



# GO FER

## Godstransportfremkommelighet på egnede ruter

### L4.0 Demonstratorer

Versjon 1.0  
august 2013



## Revisjoner

Versjon	Forfatter(e)	Beskrivelse av innhold	Dato
Versjon 1.0	Solveig Meland, Thomas Engen, Odd Andre Hjelkrem, Cato Mausethagen, Ola Martin Rennemo, Kenneth Sørensen, Jo Skjermo	Dokumentasjon av aktiviteter i GOFER Arbeidspakke 4	2013-08-19

## Innhold

<b>REVISJONER .....</b>	<b>2</b>
<b>INNHold .....</b>	<b>I</b>
<b>FIGUROVERSIKT .....</b>	<b>III</b>
<b>TABELLOVERSIKT .....</b>	<b>III</b>
<b>1 GODSTRANSPORTFREMKOMMELIGHET PÅ EGNEDE RUTER, GOFER .....</b>	<b>1</b>
1.1 BESKRIVELSE AV GOFER .....	1
1.2 INNLEDNING OG BAKGRUNN FOR DEMONSTRATORENE I GOFER .....	2
<b>2 “LIVE” DEMONSTRATOR OSLO – TRONDHEIM .....</b>	<b>3</b>
2.1 FUNKSJONALITET .....	3
2.2 HENDELSSEFORLØP .....	3
2.3 TEKNISK UTSTYR .....	4
2.4 DATASYSTEM .....	4
2.5 GRAFISK BRUKERGRENSESNITT, KLIENT .....	14
2.6 SJÅFØRENES OPPGAVER I DEMONSTRATOREN .....	14
2.7 AUTOMATISK LOGGING .....	20
2.8 GJENNOMFØRING AV TESTEN .....	20
<b>3 TUNGBILSIMULATOR I TRONDHEIM .....</b>	<b>22</b>
3.1 FUNKSJONALITET .....	22
3.2 PRAKTISKE FORHOLD OG FORBEREDELSE .....	24
3.3 GJENNOMFØRING AV TESTEN .....	26
<b>4 DEMONSTRATOR OSLO .....</b>	<b>29</b>
4.1 MODELLEN .....	29
4.2 MÅL OG OMFANG .....	30
4.3 INNSAMLING OG TILRETTELEGGING AV DATA .....	31
4.4 UTVIKLING AV BASISMODELL .....	33
4.5 FEILSØKING .....	34
4.6 KALIBRERING AV MODELLEN .....	34
4.7 SCENARIER OG AKTUELLE RESULTATUTTAK .....	37
4.8 RUTINER SOM ER UTVIKLET I FORBINDELSE MED ETABLERING AV SIMULERINGSMODELLEN .....	38
<b>REFERANSELISTE .....</b>	<b>39</b>
<b>VEDLEGG 1 DATABASE FOR LIVE-DEMONSTRATOREN .....</b>	<b>1</b>
<b>VEDLEGG 2 ETABLERING AV TRANSPORTNETT OG RUTER I GOFER LIVE-DEMONSTRATOR .....</b>	<b>5</b>
<b>VEDLEGG 3 IMPLEMENTERING AV FARTSMODELLEN I LIVE-DEMONSTRATOREN .....</b>	<b>9</b>
<b>VEDLEGG 4 APPLIKASJON FOR ADMINISTRATIV BRUK I LIVE-DEMONSTRATOREN .....</b>	<b>12</b>



## Figuroversikt

FIGUR 1: OMBORDENHET I "LIVE"-DEMONSTRATOREN .....	4
FIGUR 2: GOFER SYSTEMARKITEKTUR OG DATASYSTEM .....	6
FIGUR 3: INFORMASJONSFLYT I "LIVE"-DEMONSTRATOREN .....	8
FIGUR 4: KOMMUNIKASJONSSEKVENSS FØR OPPSTART AV OPPDRAG .....	10
FIGUR 5: KOMMUNIKASJONSSEKVENSS MENS OPPDRAG PÅGÅR .....	11
FIGUR 6: STANDARD SKJERMBILDE MED KJØRETØYPOSISJON AVGITT PÅ KARTBAKGRUNN .....	14
FIGUR 7: MENYVALG .....	14
FIGUR 8: SKJERMBILDE, ANGI KJØRERUTE .....	15
FIGUR 9: SKJERMBILDE, ANGI LASTVEKT .....	15
FIGUR 10: SKJERMBILDE, ANGI LEVERINGSSTED .....	16
FIGUR 11: SKJERMBILDE, OPPSUMMERING AV OPPSTART-INFORMASJON .....	16
FIGUR 12: SKJERMBILDE, VIKTIGE VEGMELDINGER FOR VALGT RUTE .....	16
FIGUR 13: STANDARD SKJERMBILDE MED "VARSLER HENDELSE" .....	16
FIGUR 14: SKJERMBILDE, VARSLER OM SPESIELLE KJØREFORHOLD .....	17
FIGUR 15: MELDINGSINNHOOLD OG UTLØPSTID FOR GOFER-MELDINGENE .....	17
FIGUR 16: SKJERMBILDE, MOTTA MELDING FRA SJÅFØR .....	17
FIGUR 17: SKJERMBILDE, TILLEGGSTEKST GOFER-MELDING .....	18
FIGUR 18: SKJERMBILDE, MOTTA VEGMELDING .....	18
FIGUR 19: SKJERMBILDE, TILLEGGSTEKST VEGMELDING .....	18
FIGUR 20: SKJERMBILDE, REGISTRÉR HVILETID .....	18
FIGUR 21: SKJERMBILDE, VARSEL OM ANBEFALT KJØRERUTE OG Plass PÅ TERMINAL .....	19
FIGUR 22: STANDARD SKJERMBILDE MED "AVSLUTT OPPDRAG" .....	19
FIGUR 23: SKJERMBILDE, BEKREFT SLUTT PÅ TEST-TUR .....	19
FIGUR 24: HELE VEGNETT I SIMULATOREN I RØDT. REGISTRERINGSSTREKNING MERKET BLÅTT .....	22
FIGUR 25: VISUELL FRAMSTILLING I KJØRESIMULATOREN – E6 VED SLUPPEN, RETNING MOT SENTRUM .....	23
FIGUR 26: VISUELL FRAMSTILLING I KJØRESIMULATOREN – VISNING I FORSØKENE, TRONDHEIM SENTRUM MED FØRERKABINEN .....	23
FIGUR 27: VISUELL FRAMSTILLING I KJØRESIMULATOREN – SCANERIO MED PRIORITERINGSTILTAK I MELLOMRUSH-PERIODE .....	24
FIGUR 28: VISUELL FRAMSTILLING I KJØRESIMULATOREN – SCANERIO MED PRIORITERINGSTILTAK I LAVTRAFIKKPERIODE .....	24
FIGUR 29: AVGRENSNING AV MODELLOMRÅDET I OSLO .....	30
FIGUR 30: OVERSIKT OVER STREKNINGER MED REGISTRERTE GJENNOMSNITTSHASTIGHETER .....	32
FIGUR 31: RESULTAT AV KALIBRERING AV GJENNOMSNITTSHASTIGHETER .....	35
FIGUR 32: RESULTAT AV KALIBRERING AV TRAFIKKVOLUM. SØYLENE VISER GEH FOR HVERT NIVÅ-1 TELLEPUNKT. DEN RØDE STREKEN INDIKERER KALIBRERINGSMÅLET PÅ GEH LIK 5 .....	36

## Tabelloversikt

TABELL 1: SENTRALE BEGREPER I GOFER .....	5
TABELL 2: DE FIRE TEST-SCENARIENE I KJØRESIMULATOREN .....	23
TABELL 3: OVERSIKT OVER TELLEPUNKT .....	32
TABELL 4: GJENNOMSNITTLIG ESTIMERT HASTIGHET I DE 9 OMRÅDENE, MED 95 % KONFIDENSINTERVALL .....	33



# 1 Godstransportfremkommelighet på egnede ruter, GOFER

## 1.1 Beskrivelse av GOFER

### Mål og prosjektidé

Hovedmål med GOFER-prosjektet er å bidra til reduserte miljø- og klimautslipp, køproblemer, ulykker og operatørkostnader for godstransport, gjennom å ta i bruk nye samarbeidsformer og teknologiske løsninger. Målet i GOFER er å etablere løsninger som muliggjør styring og regulering av tung godstransport, på samme måte som flykontrollen opererer flytrafikken.

### Muligheter og behov

I prosjektets første fase ble hovedtyngden av arbeidet rettet mot å kartlegge mulighetene for å få til å etablere et demonstratorprosjekt. Aktivitetene omfattet identifisering av brukerbehov og -krav, og mulige samarbeidsmodeller. Resultatene fra dette arbeidet er dokumentert i L1.0 Behovsanalyse og samarbeidsmodell (GOFER, 2010a).

### Demonstratorer

Hovedaktivitetene i prosjektet siste fase er knyttet til tre ulike demonstratoraktiviteter:

#### *"Live"-demo:*

Vinteren 2011-12 deltok tungbiler fra Bring i en ti-ukers live-demonstrator på strekningen Oslo-Trondheim. Testen var basert på datasystemer utviklet i prosjektet, og inkluderte bl.a. utveksling av sjåfør-initierte meldinger knyttet til forhold langs kjøreruta, beregning av ankomsttidspunkt og evt. forsinkelser til terminal ved hjelp av en nyutviklet Fartsmodell, og anbefaling av rutevalg i Trondheim.

#### *Tungbilsimulator:*

I mangel på mulighet til å gjøre studier av prioriteringstiltak ute i trafikken, er kjøresimulator benyttet for å studere mulige effekter av prioriteringstiltak for tungbiler som adgang til kollektivfelt og prioritering i lyskryss. Testene er gjort med oppdatert vegnett og trafikksituasjon for Trondheim.

#### *Simuleringsmodell for Alnabru-området:*

For å studere mulige fullskala-effekter av og nødvendige forutsetninger for etablering av et GOFER-system, er det etablert en simuleringsmodell for Alnabruområdet med tilfarer. Disse aktivitetene inngår i et doktorgradsarbeid, og vil pågå etter at GOFER-prosjektet er avsluttet.

Evaluering av demonstratorene er dokumentert i L3.0 Evaluering (GOFER, 2013b). Videoer fra "live"-demonstratoren og test i tungbilsimulatoren ligger på prosjektets nettside.

Systemarkitektur og datamodell for live-demonstratoren er dokumentert i L2.0

Systemarkitektur og datamodell (GOFER, 2013a).

### Videre anvendelse

I L5.0 Videre anvendelse (GOFER, 2013c) er erfaringer med konsept og løsninger, teknologi og rammeverk oppsummert og benyttet som grunnlag for å identifisere eventuelle suksesser,

potensiale og muligheter for forbedring, videreutvikling og videre anvendelser av idéer og løsninger fra GOFER-prosjektet.

### **Prosjektorganisering**

Prosjektperiode er 2009-2012. Prosjektet støttes av Norges forskningsråd gjennom SMARTRANS-programmet. ITS Norge ([www.itsnorge.no](http://www.itsnorge.no)) er prosjekteier, og følgende aktører inngår i konsortiet: SINTEF Teknologi og samfunn (Prosjektleder), NTNU, Bring, CargoNet, Statens vegvesen, Triona, Q-Free, Logistikk- og Transportindustriens Landsforening (LTL) – nå NHO Logistikk og Transport, Oslo kommune Bymiljøetaten, Bergen kommune, Trondheim kommune, Trondheimsfjordens Interkommunale Havn IKS, Rogaland Fylkeskommune.

## **1.2 Innledning og bakgrunn for demonstratorene i GOFER**

GOFER-prosjektet gjennomfører demonstratorer i Trondheim og Oslo. Dette notatet gir en kort oversikt over de ulike delene av disse demonstratorene.

GOFER-demonstratoren skal ikke primært være en test av teknologi, men en demonstrasjon av tjenester og funksjonalitet. Dette har vært en viktig premiss for de prioriteringene og avgrensningene som er gjort under veis i arbeidet med å utforme demonstratoren.

Erfaringene viser at det kan være vanskelig å demonstrere nye prioriteringstiltak "live" ute i trafikken. Dette viste seg tidlig å være tilfelle for Oslo-demonstratoren, som ble besluttet gjennomført som en simuleringsaktivitet. Denne beskrives i kapittel 4.

I Trondheimsområdet var det tanken å gjennomføre reelle prioriteringstiltak for tungbiler i lyskryss og kollektivfelt. Å få tillatelse til å slippe tungbiler til i kollektivfelt viste seg imidlertid å være umulig innenfor tidsrammene for dette prosjektet. Samtidig viste det seg at det å gjennomføre reell tungbilprioritering i lyskryss ville koste uforholdsmessig mye. For likevel å kunne studere effekter av konkrete prioriteringstiltak for tungbiler, benyttes tungbildelen av kjøresimulatoren ved SINTEF. Denne delen av demonstratoren beskrives i kapittel 3.

Hovedidéen i GOFER er å etablere et kooperativt system for informasjonsutveksling med og styring av tungtrafikk. Dette gjennomføres i "live"-delen av demonstratoren, som gjennomføres med deltakelse av Bring-biler og sjåfører som kjører mellom Oslo og Trondheim. Denne delen av demonstratoren beskrives i kapittel 2.



## 2 “Live” demonstrator Oslo – Trondheim

Denne demonstratoren inkluderte åtte av Brings biler som jevnlig kjørte strekningen fra Oslo til Trondheim.

### 2.1 Funksjonalitet

Demonstratoren inkluderte:

- kommunikasjon mellom sjåfører, transportutøver og baksystem, med oppdatert informasjon om forhold langs valgt rute
- beregning av forventet ankomsttid, basert på Fartsmodellen
- tenkt booking av ressurser (plass/slottid på terminal)
- styring/anbefaling av rutevalg basert på kjøretøy-/lastinformasjon og informasjon om reisemål i Trondheim
- distribusjon av sjåfør-initierte meldinger (heretter kalt GOFER-meldinger) til ”vegmeldinger for yrkessjåfører”
- informasjon til terminal om biler under veis
- nedlasting av GPS-data for bl.a. beregning av miljørelaterte forhold (evt. også utslippsdata rett fra motoren, for sammenligning av resultater)

### 2.2 Hendelsesforløp

Dreiebok ble utarbeidet i samarbeid mellom SINTEF og TRIONA, Q-Free, SVV og Bring. Her beskrives kort og punktvis hovedpunktene i den praktiske demonstratoren. Hoveddelen omfattet langtransport som kom inn til Trondheim fra Oslo.

Hovedfokus var på informasjonsutveksling under veis, og styring av kjørerute i Trondheim. I tillegg ville det vært ønskelig å inkludere prioritering i lyskryssene i Elgeseter gate, tilgang til kollektivfelt utenom rush, og tildeling av slot-tid ved terminal. Prioritering i lyskryss og tilgang til kollektivfelt viste seg imidlertid å være urealistisk å få til ute på veien. Dette ble derfor gjennomført som forsøk i tungbilsimulator i stedet, se kapittel 3. Tildeling av slot-tid ved terminal krever et baksystem på terminalsiden som ennå ikke eksisterer. Dette ble imidlertid inkludert i live-demonstratoren med automatisk genererte meldinger som eksempel på hvordan et slikt system kan fungere i praksis.

Hovedpunktene i hendelsesforløpet i demonstratoren var:

- Kjøretøy registreres inn i demonstratoren ved avreise fra Oslo.
- Forventet ankomsttid (inkludert pålagt hviletid) beregnes fortløpende under veis
- Sjåfør og terminal i Trondheim får tilgang til informasjon som oppdateres fortløpende om transporten og forhold langs kjøreruta.
- Sjåfør kan sende og motta meldinger om trafikkrelaterte forhold under veis
- Kjøretøy ankommer demonstratorområdet i Trondheim:
- det gis beskjed om hvilken kjørerute som skal benyttes/anbefales i Trondheim. Inkluderte også tildeling av slot-tid på terminalen ut fra beregnet ankomsttid.
- Kjøretøy ankommer terminal/kunde, og kvitteres ut av demonstratoren.

## 2.3 Teknisk utstyr

I tilknytning til de ulike fasene i hendelsesforløpet, kan en se for seg ulike former for informasjonsutveksling mellom kjøretøy/sjåfør, terminal og vegkantutstyr/baksystem. Hvilket utstyr som skulle benyttes i disse prosessene, hvilken informasjon som skulle utveksles og hvordan informasjonsutvekslingen skulle foregå, ble vurdert i forhold til relevans, ressursbruk og risiko knyttet til de ulike løsningene.

### 2.3.1 Ombordenhet

Alle kjøretøyene i demonstratoren ble utstyrt med en Samsung Galaxy Tab. Det var lenge tanken å i tillegg inkludere CVIS-utstyr i demonstratoren. CVIS inngikk bl.a. i Trondheimsdemonstratoren i EU-prosjektet SMARTFREIGHT, der potensialet ved bruk av denne teknologien ble vist med suksess. Ettersom GOFER-demonstratoren ikke skulle være en test av teknologi, men en demonstrasjon av tjenester og funksjonalitet, ble bruk av CVIS i GOFER til slutt skrinlagt. Dette skyldtes at fokus i prosjektet har vært på teknologiavhengige løsninger. Bruk av to ulike typer teknologi i demonstratoren ville gjøre gjennomføringen mer komplisert og ressurskrevende enn nødvendig. Valget falt på Samsung Galaxy Tab fordi dette er hyllevarer, mens CVIS-utstyr kun var tilgjengelig i svært begrenset omfang (2-3 komplette sett med utstyr), og da som prototype.



Figur 1: Ombordenhet i "live"-demonstratoren

### 2.3.2 Server

Server ble satt opp, bestående av standard PC hovedkort, redundante diskere og rundt 20 GB RAM. Vi brukte Windows 2008 Server operativsystem, PostGIS geodatabase, Knopflerfish server for sjåføragerter, og Tomcat servletcontainer for administrative rutiner. Vi var ikke på noe tidspunkt i nærheten av å overbelaste systemet.

## 2.4 Datasystem

Demonstratoren ble realisert med utgangspunkt i datamodell og arkitektur fra prosjektets Arbeidspakke 2 Systemarkitektur og datamodell (AP2).

Datamodellen ble overført til et databaseskjema, med noen modifiseringer og utvidelser. Databasen som ble benyttet i demonstratoren er en mer komplett datamodell enn den i AP2, selv om noen integritetsregler falt ut, og enkelte detaljer bør endres.

Ikke all funksjonalitet som ble beskrevet i AP2, ble inkludert i demonstratoren.

Under tilrettelegging og redigering av grunnlagsdata, ble det benyttet programvare fra andre prosjekter:

- Vegnett ble hentet fra NVDB på vanlig måte, men for å legge til rette for test av fartsmodellen, ble vegnettet oppdelt (segmentert) og komplettert med de samme funksjonene som ble utviklet der.

- I demonstratoren er kjørerutene statiske. De ble definert ved hjelp av kollektivrouteutleggingen i TNext, som igjen har lånt kode fra Storbysamarbeidet og DoIT.
- Generelt vedlikehold av tabeller ble utført via en web-applikasjon, som i hovedsak består av kode fra Dakat-prosjektet.
- Fartsmodellen ble implementert som tilleggsfunksjoner til denne.
- Veimeldingstjenesten baserte seg på informasjonsplattformen TRIP, utviklet av TRIONA, og kode fra WiseCar-prosjektet.

En del begreper er brukt i dokumentasjonen av GOFER-prosjektet. For å kunne presist kunne beskrive systemet er det nødvendig og definere hva som menes med disse. En del sentrale begreper er forklart i Tabell 1.

Tabell 1: Sentrale begreper i GOFER

Begrep	Betydning
Baksystem	Server og tilhørende funksjonalitet som tar i mot og formidler data
Digital Assistent	Også kalt klient, enhet som er montert i bil som fungerer som grensesnitt mellom sjåfør og baksystem, i demonstratoren en Samsung Galaxy TAB
Aktør	En enhet i systemet. Aktøren kan være en person, en organisasjon, en datamaskin eller en ressurs, som for eksempel en terminal
Rolle	Oppgave eller et sett med rettighet en aktør har i systemet.
Agent	Agenten er et dataprogram som handler på vegne av aktører i GOFER-systemet. Aktører kommuniserer alltid med agenter, agentene kommuniserer deretter med database og andre ressurser på vegne av aktøren
Vegmelding	En melding fra vegloggen slik systemet er implementert i dag. Hentes via dataskyen TRIP, driftet av Triona AS
Fartsmodell	Forskningsrådsprosjekt gjennomført i regi av SINTEF tidligere. Basert på farts- og posisjonsdata fra et større antall tungbiler over en periode, har man utarbeidet en modell for kjøretid på en veglenke som tar med vekt på kjøretøyet i beregningene.
Veglenke	En matematisk beskrivelse av vegnettet basert på linjer mellom noder.
Rute	Et nettverk av veglenker fra et startpunkt til en destinasjon.
Kjøretid	Beregnet tid det tar å kjøre en gitt rute, uten at pauser er tatt med i beregning

#### 2.4.1 Resonnement omkring arkitektur

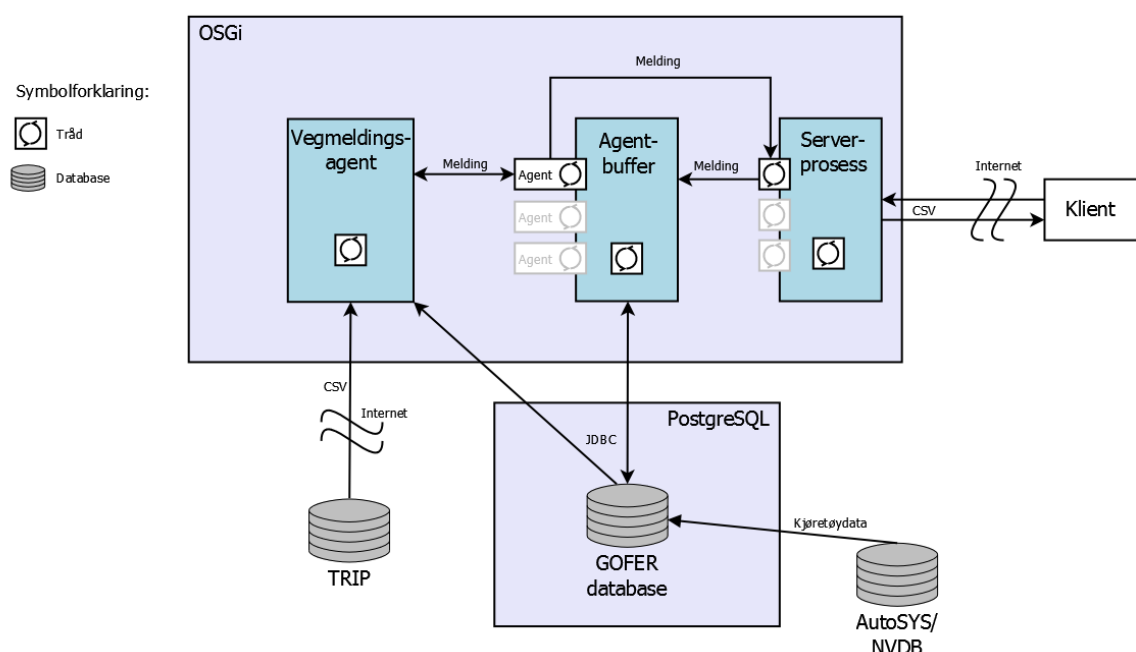
Det er naturlig nok ingen "riktig" måte å implementere et system som dette på, så spesifikke designvalg ble tatt før programmeringen begynte. Bakgrunn for de valgene som har blitt gjort her kommer av en del av begrensningene som ligger i http-protokollen, det er nemlig ikke mulig å adressere en enhet direkte om den ikke har fast IP. Derfor er all kommunikasjon nødt til å initieres av klienten, etter som serveren er den eneste aktøren i systemet som har en fast IP adresse.

Strukturen på systemet er inspirert av hvordan telefonsystemet fungerer. Der en tjeneste muliggjør kommunikasjonen, data sendes derfra inn til en sentral som sørger for at det opprettes kommunikasjon mellom riktige abonnemeter i systemet. Agenter henter deretter data på vegne av klienten fra en database. Dette gjør det mulig å ha en kompakt protokoll, samtidig som klientene er forholdsvis nært databasen.

Den sentraliserte løsningen betyr også, som det står i kapittelet "komponentperspektiv" i AP2, at vedlikehold og samordning av aktører kan gjøres enkelt og pålitelig.

Agentene kan opprettes og avvikles dynamisk mens systemet kjører. Så lenge en klient er logget på i systemet og sender data vil agenten leve. Etter hvert som flere klienter kommer til systemet vil det opprettes flere agenter. Klientene trenger heller ikke være knyttet til en type aktører i systemet, slik den her er knyttet til klientene i bilen, den kan like gjerne handle på vegne av andre aktører i systemet, slik som kunder, transportør eller transportmyndigheter. Agenten er i praksis en digital representant for den som ønsker å gjøre noe i systemet.

## 2.4.2 Hovedkomponenter



Figur 2: GOFER systemarkitektur og datasystem

Systemet (Figur 2) består i hovedsak av fem overordnede komponenter, om man bare ser på serversiden og på meldingsutvekslingen. Disse er:

1. Klient – Enhet som er montert i kjøretøy og som formidler informasjon fra GOFER-systemet til baksystemet.
2. Servlet (Serverprosess) – Er kontaktpunktet til klientene. Servleten er ansvarlig for å ta i mot meldinger og formidle disse videre til AgentPool
3. AgentPool (Agent-buffer) – Holder styr på hvilken aktør som kommuniserer med hvilken agent
4. Agent – Agent er en prosess som handler på vegne av aktøren som sender forespørsel, den er ansvarlig for å holde databasen oppdatert med riktig data fra klienten, samt å formidle nødvendige beskjeder til klientene
5. GOFER database – Database der alle nødvendige data for GOFER-systemet lagres, blant annet posisjonslogg fra klientene, samt oppdragsdefinisjoner, lastvekt og lignende.

### 2.4.3 Datautveksling

All kommunikasjon mellom klient og server foregår ved hjelp av http-protokoll og klienten initierer all kommunikasjon ved å generere en http – post. Post meldinger er en del av http-standarden og metoden som blant annet nettlesere bruker for å utveksle meldinger over nettet. Denne meldingen tas i mot av servleten som sender den videre til agentpool der den blir sendt videre til riktig agent for tolkning av innholdet.

Datainnholdet i post meldingen er en strengt formattert CSV-fil (Comma Separated Values) som agentene kan tolke. Meldingen må være riktig formattert, skilletegnet mellom de forskjellige verdiene må være et semikolon. I tillegg må hver enkelt linje avsluttes med et newline (n) tegn. Meldingene har også noen obligatoriske felter, disse er

- Timestamp – tidspunktet da meldingen ble generert
- ID – unik identifikator for avsender av meldingen
- Type – hva slags type melding det er snakk om

En CSV-fil kan inneholde de datatyper man ønsker, så lenge man lager mekanismer for å håndtere disse, både på klientsiden og på serversiden.

Når meldingen kommer frem til AgentPool blir den rutet videre til riktig agent. Hver klient skal ideelt sett ha en egen agent som kjører på serveren, men dette er ikke alltid tilfelle. For å frigjøre dataressurser vil agenter etter hvert utløpe og bli terminert om de ikke mottar data. Alle nødvendige data for å gjenskape tilstanden agenten hadde før den ble terminert vil imidlertid være lagret i GOFER-databasen. I GOFER-demonstratoren har man to typer agenter, en agent som administrerer kommunikasjon med TRIP (Transport Related Information Platform) som driftes av Triona, samt Agenter som handler på vegne av aktører i systemet, slik som de mobile klientene som står i kjøretøyene. I praksis kan man ha så mange typer agenter man ønsker i systemet.

### 2.4.4 Informasjonsflyt

GOFER-demonstratoren er et kooperativt system, med informasjonsutveksling mellom sjåfører, terminal og myndigheter. Figur 3 illustrerer dette.

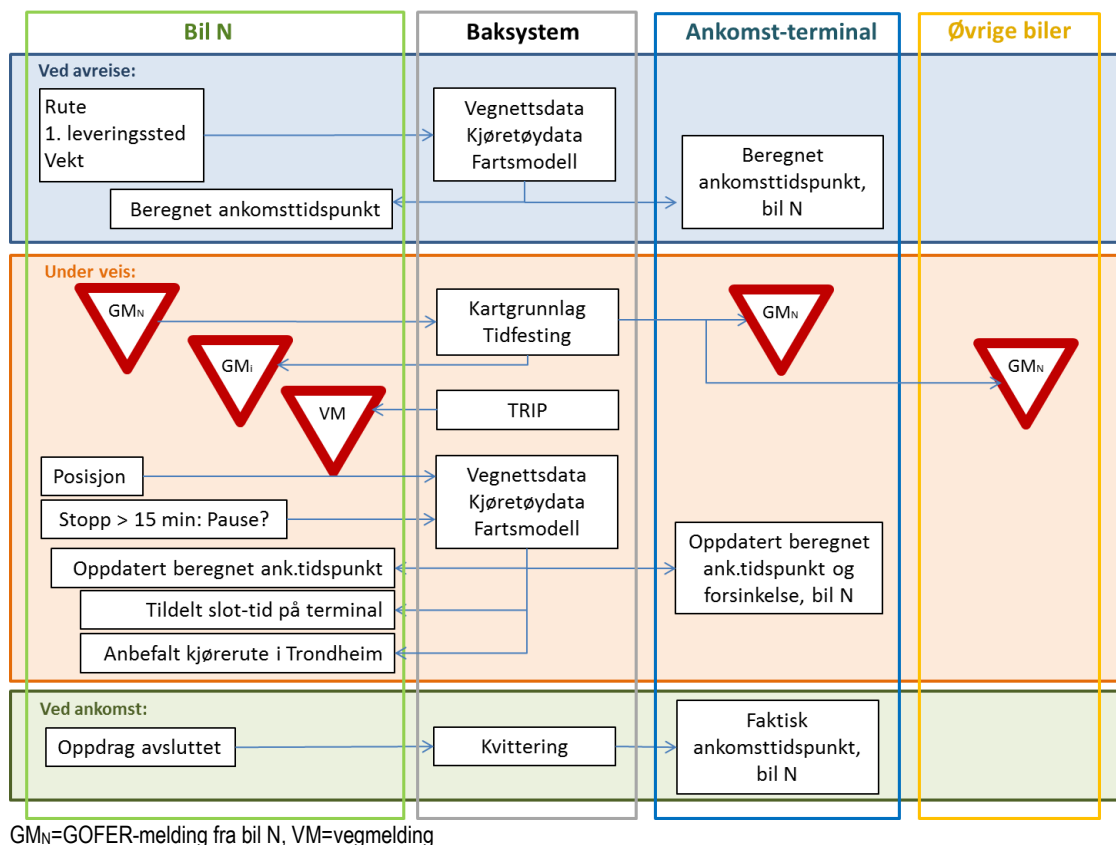
#### *Informasjon fra terminal eller transportselskap til sjåfør før avreise*

Demonstratoren inkluderer ikke reell datainput fra terminal eller transportselskap. Det vil imidlertid være mulig å tilrettelegge funksjonalitet for å hente relevant informasjon om kjøreoppdraget direkte fra de bedriftsinterne datasystemene og inn i GOFER-systemet. I GOFER-demonstratoren la sjåføren inn denne informasjonen manuelt før oppstart.

#### *Til terminal*

Hvilken type informasjon som vil være nyttig for ankomst-terminalen, framkom i samarbeid med ansatte ved Bing Logistics' terminal i Trondheim:

- Beregnet ankomsttid og evt. forsinkelse for hvert kjøretøy
- Forhåndsdefinerte meldinger som er sendt fra bilene
- Klokkeslett for når oppdraget ble avsluttet (for å unngå å ringe til sjåfører i løpet av den pålagte hviletiden etter at kjøretøyet er parkert)



Figur 3: Informasjonsflyt i "live"-demonstratoren

## 2.4.5 Generell meldingsutveksling

### Tilstander:

For å forstå strukturen i GOFER arkitekturen er det viktig å ha oversikt over de forskjellige tilstandene systemet har og hvilke meldinger som går i systemet i de forskjellige tilstandene. Tilstandene er styrt av klienten, eller egentlig kjøretøy/ fører. De viktigste tilstandene er:

- Pending – venter på å bli tildelt et oppdrag
- Accepted – oppdrag akseptert men ikke påbegynt
- Under delivery – har påbegynt oppdraget
- Oppdrag utført

Når en klient logger seg på nettet og spør om data vil den være i tilstanden *pending*. Dette betyr at kjøretøyet ikke har blitt tildelt noe oppdrag enda. Det vil til en hver tid ligge oppdrag, eller transporter som det kalles i GOFER systemet, tilgjengelig for klienten i databasen. Oppdrag er alltid knyttet opp mot identifikatoren til en spesifikk klient. Oppdrag tildeles alltid på kjøretøy-/klientnivå og man kan derfor ikke velge oppdrag tiltenkt andre kjøretøy.

Når et oppdrag er valgt går systemet inn i tilstand nummer to *accepted*. Dette er gjerne før bilen lastes og klargjøres til transporten. Når bilen er klar til å starte transportoppdraget setter sjåføren systemet i "under leveranse"-modus. Når dette gjøres sendes nøkkelinformasjon slik som lastvekt til serveren slik at kjøretiden kan beregnes ved hjelp av fartsmodellen. Når bilen er underveis kan det gå fire typer meldinger fra kjøretøyet. Disse er:

- Statusmeldinger
- Logmeldinger
- Sjøførmeldinger
- Forespørsel om vegmeldinger

Statusmeldinger er meldinger som endrer tilstanden til klientene. Disse sendes kun ut før oppdraget starter for å endre fra *pending* til *accepted*, fra *accepted* til *under delivery* og fra *under delivery* til oppdrag fullført.

Når oppdraget er i gang vil klientene hvert femte sekund sende ut en logmelding, denne inneholder nøkkeldata slik som GPS-posisjon, antall satellitter som er synlig for GPS-antenne, fart, retning og gjenstående hviletid, for å nevne noen. Disse dataene lagres i GOFER-systemet for analyse etter at prosjektet er ferdig. I tillegg benyttes dataene til å lage et estimat for gjenværende kjøretid som legges i svarmeldingen som går tilbake til klienten.

I tillegg å sende loggmeldinger kan klienten også sende ut sjåførmelding, disse er forhåndsdefinerte meldinger som angår sjåføroppgaven, slik som vanskelige vegforhold, stengte veger og behov for kjetting. Disse er spesifikt tiltenkt tungbilsjøfører og det er sjåføren selv som sender disse, basert på sine egne vurderinger om situasjonen. Meldingene går inn i databasen og gjøres tilgjengelig for andre sjåfører.

Den siste meldingstypen er forespørsel om vegmeldinger, der klienten ber serveren sende vegmeldinger til klienten. Vegmeldinger hentes fra TRIP hvert femte minutt og gjøres tilgjengelig til klientene. Sjøførmeldinger og vegmeldinger har samme format og sammenstilles i samme tabell i databasen, slik at når en klient ber om vegmeldinger mottar denne en liste av både veg- og sjåførmeldinger.

Når oppdraget er fullført kvitterer sjåførene for dette vi av *oppdrag fullført* melding som setter systemet tilbake til *pending*.

I GOFER-terminologien har de forskjellige meldingene følgende navn:

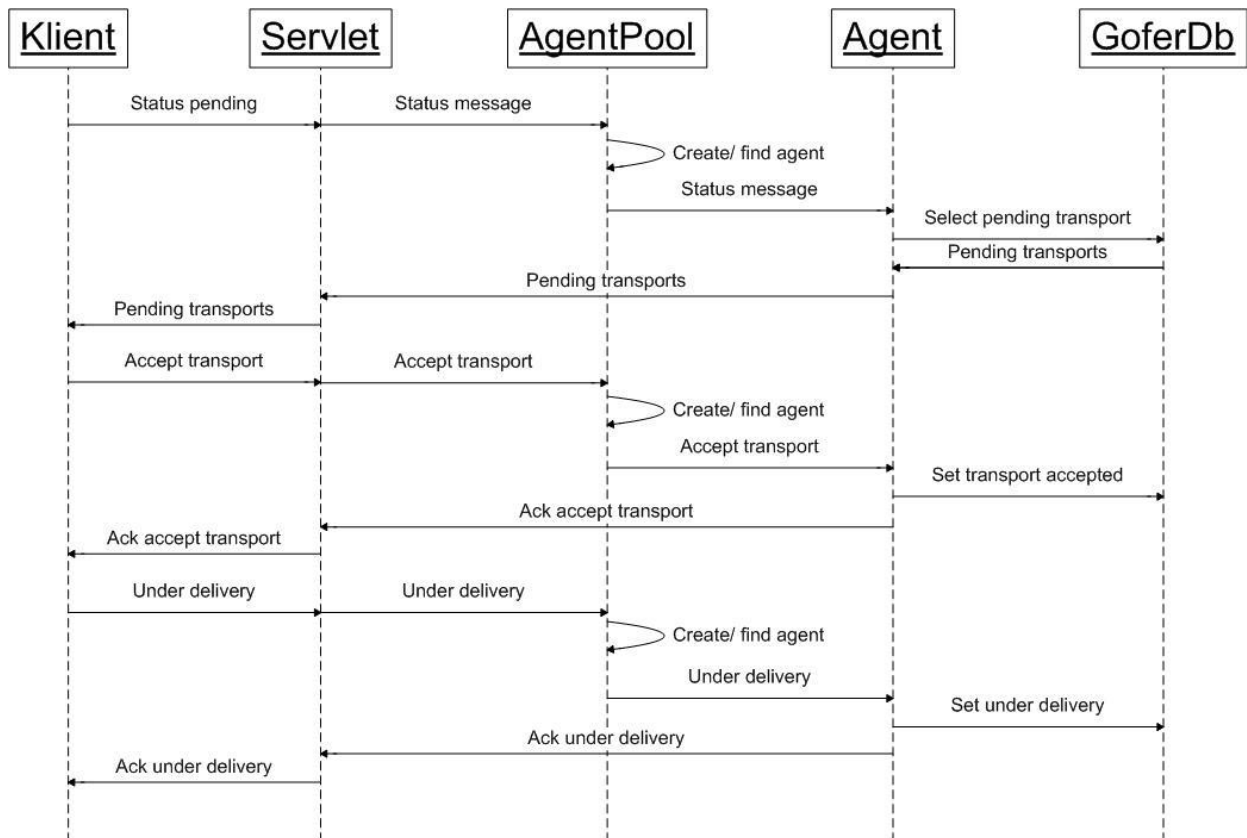
- Request – generisk forespørsel fra klienten, alle meldinger fra klienten er request
- Response – Svar på request, alle meldinger som går fra serveren ut til klienter er responser, dette er i henhold til http standarden
- StatusMessage – Meldinger som endrer tilstanden til klienten, det finnes fire forskjellige tilstander, disse sendes i klartekst til serveren via en statusMessage for å endre tilstand. Disse er
  - Pending – avventer oppdrag
    - Accept – Godta oppdrag
    - Under delivery – oppdraget underveis
    - Finished – oppdrag fullført
- LogMessage – Melding som inneholder data som skal lagres i databasen. Inneholder blant annet GPS posisjon for kjøretøy
- DriverMessage – Melding for å rapportere om forhold som kan påvirke fremkommeligheten til kjøretøyet. Sendes av klient /sjåfør.

Som tidligere forklart initieres all kommunikasjon av klientene og alle data som går ut til klientene fra serveren er respons på en request Ved å gjøre det på denne måten er systemet

skalerbart i og med at man ikke trenger å vite om hvor mange kjøretøy som befinner seg på veien til en hver tid. Så lenge de har en gyldig id, transport og sender data til serveren vil de være inne i systemet.

#### 2.4.6 Kommunikasjonssekvens

Når et oppdrag startes opp, melder kjøretøyet seg inn i systemet som ledig. Om det finnes ledige oppdrag tilgjengelig for kjøretøyet vil disse mottas av klient slik at sjåføren kan velge det oppdraget som stemmer med de oppgavene han eller hun er tildelt. Når oppdraget velges velger også sjåføren mellom de forskjellige rutealternativene som finnes for oppdraget, dersom det er flere. For demonstratoren er dette begrenset til om sjåføren skal kjøre Østerdalen eller E6 Gudbrandsdalen fra Oslo til Trondheim. Når bilen er ferdig lastet og klar til å påbegynne transporten meldes dette, sammen med vekten på lasten inn til GOFER-systemet. Informasjonen om vekt benyttes til å beregne kjøretiden. De forskjellige meldingene som sendes og hvordan disse utveksles mellom komponentene i systemet er gjengitt i Figur 4.

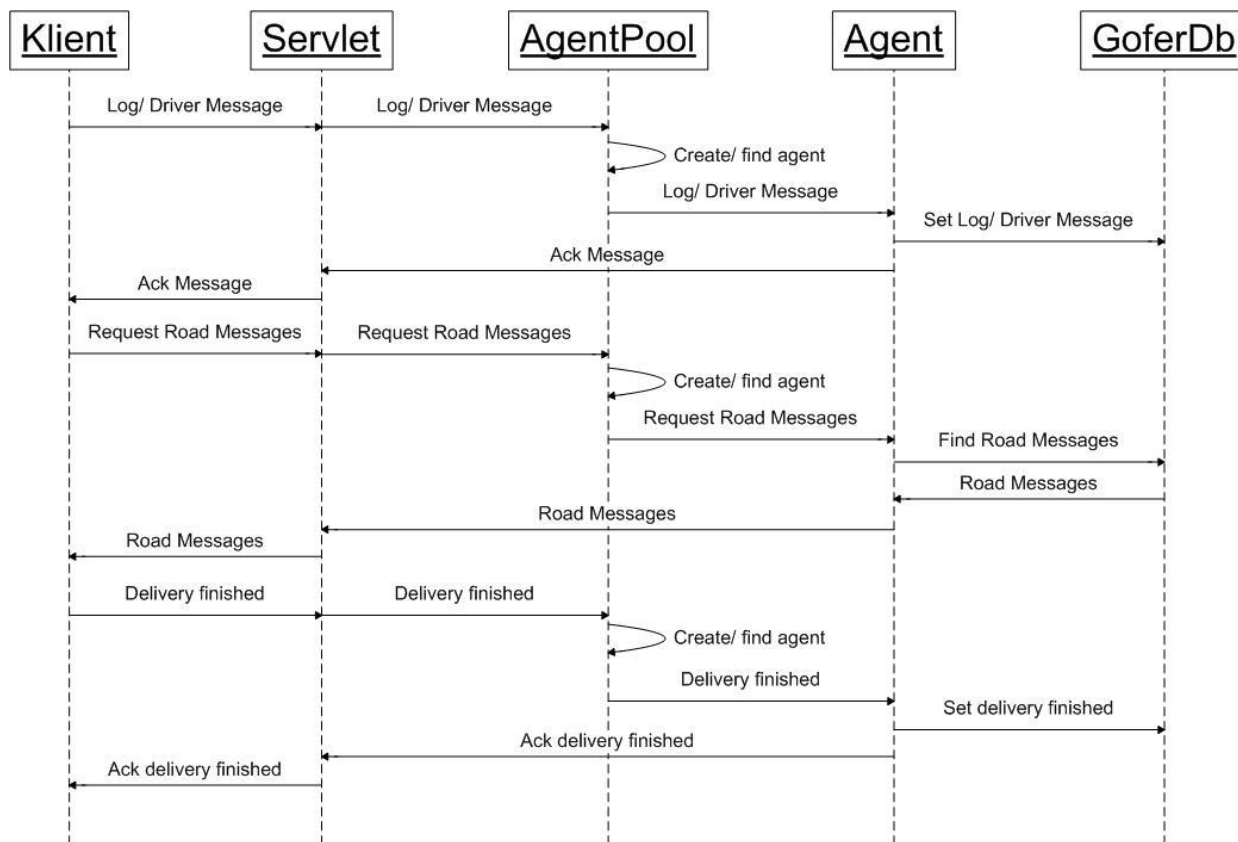


Figur 4: Kommunikasjonssekvens for oppstart av oppdrag

Når oppdraget er akseptert bytter systemet, som tidligere forklart, kontekst eller tilstand og går over i oppdragsmodus som har sin egen meldingsflyt. I denne tilstanden utveksles stort sett bare "LogMessages" eller loggemeldinger, som inneholder data om kjøretøyet og som sendes og lagres i databasen. Disse dataene sendes fra klienten hvert femte sekund. I denne tilstanden har dessuten sjåførene mulighet til å sende sjåførmeldinger. Disse håndteres på



samme måte av systemet som loggemeldinger. I tillegg sendes det også periodisk forespørsler om vegmeldinger. Sekvensen på de forskjellige meldingene er gjengitt i Figur 5.



Figur 5: Kommunikasjonssekvens mens oppdrag pågår

Når oppdraget er utført går klienten i "pending" eller avventer modus. Den eneste meldingen som er tilgjengelig i denne tilstanden er melding om at klienten kan ta i mot oppdrag.

## 2.4.7 Applikasjoner

Applikasjonene som ble benyttet i GOFER var delvis gjenbruk av applikasjoner som var utviklet i andre prosjekter/sammenhenger, og delvis nyutviklet eller videreutviklet for å dekke behovene i GOFER-demonstratoren.

### Videreutvikling av eksisterende applikasjoner

*Allerede eksisterende funksjonalitet/applikasjoner som ble videreutviklet og inngikk i etablering/drift av GOFER-demonstratoren:*

- Bibliotek for å hente ut data fra TRIP til CVIS<sup>1</sup> (SMARTFREIGHT)
- Tracking-applikasjon, Triona
- Karttjeneste basert på Open Street Map<sup>2</sup> (OpenSource)
- TNext for generering av rute-definisjoner (TNext-prosjektet for NTP/SVV)

<sup>1</sup> <http://www.cvisproject.org/>

<sup>2</sup> <http://www.openstreetmap.org/>

- DAKAT-komponenter for vedlikehold av grunnlagsdata (DAKAT-prosjektet for SVV)
- Avstand til den mest relevante vegmelding for deg (fra vegloggen eller P4) (ITS TestBeds<sup>3</sup>)
- Generell aksessapplikasjon fra SMARTFREIGHT

### Nyutviklede applikasjoner

- Spørrefunksjonalitet basert på posisjon og gjenstående del av kjørerute, for å gi beskjeder under veis (f.eks. vegmeldinger eller anbefalt kjørerute)
- Databasemodell bl.a. for bestillingsfunksjoner (f.eks. slot-tid) og håndtering av kjøretøyinformasjon.
- Vegmeldinger som er relevante for deg – basert på CVIS
- GeoFencing-funksjon knyttet til booking og ruteanvisning

#### 2.4.8 Estimert ankomsttid

Forventet ankomsttid estimeres ved oppdragets oppstart og skrives til databasen. Denne verdien endres imidlertid ikke underveis, men siden bilen sender sin posisjon til baksystemet hvert femte sekund har man til en hver tid god oversikt over hvor bilen er. Dette benyttes til å beregne til å sende en oppdatert estimert ankomsttid kontinuerlig, som ikke lagres. Estimert kjøretid er beregnet ved hjelp av resultatene fra fartsmodellprosjektet, ikke kun ut i fra fartsgrenser. Denne modellen tar inn i betraktningene hvordan bilen påvirkes av lastvekten og hvordan sjåførene velger fart ut i fra terreng, slik som i stigninger og svinger. I tillegg legges det til 45 minutter hviletid i estimatet etter som turen fra Trondheim er så lang at en lang pause er påkrevd. Systemet sjekker underveis om sjåføren har pause ved at den registrerer at bilen står stille og logge tiden. Sjåføren må selv bekrefte om det er pause eller bare trafikale problemer, for eksempel.

Ut i fra denne informasjonen kan systemet til en hver tid vite om bilen er forsinket, ganske enkelt ved å sjekke at nåtidspunkt pluss gjenværende reisetid og hviletid er et tidspunkt etter det som var estimert ankomsttid ved oppdragets start. Dette kan brukes til å endre ressursallokering i form av lasteramper og terminaler, og ikke minst i form av menneskelige ressurser.

#### 2.4.9 Etablere rute

Nettverket blir delt inn på samme måte som nettverket i Fartsmodellprosjektet (Tørset m.fl., 2011), slik at man slipper å introdusere flere usikkerhet eter angående beregningenes gyldighet.

Se databaseskjemaet for GOFER, gjengitt i Vedlegg 1, og Vedlegg 2, der etablering av transportnett og ruter beskrives nærmere.

Etablering av rute involverte følgende tabeller: *route*, *routeroadlink*, *roadlink*, som blir forkortet til hhv. r, rrl og rl i det etterfølgende.

---

<sup>3</sup> <http://www.itstestbeds.org/>

En GOFER-rute er en ordnet liste av veglenker, som hentes slik:

```
Datasett rute:  
select id, route, nr, roadlink from roadroutelink where route = :rutenr order  
by nr;
```

Parameter *rutenr* angir den valgte ruten.

Det å gå langs ruten betyr ganske enkelt å lese denne listen rad for rad, slik den blir hentet. Hent lenkedetaljene slik:

```
Datasett lenke:  
select * from roadlink where id = :rute.lenkenr;
```

#### 2.4.10 Implementering av Fartsmodellen

Vegnettet ble delt inn i lenker etter samme kriterier som i Fartsmodellprosjektet:

- En rute defineres som lenkeliste fra start til slutt.
- Kjøretøy beskrives med totalvekt, motoreffekt med utnyttelsesgrad, frontareal m. friksjonskoeffisient og rullestand.
- Utnyttelsesgrad på motor er noe høyere ved minkende fart i motbakke, enn under vanlig akselerasjon.
- Reisetiden lagres pr lenke og transport.
- For hver lenke er inngangsfarten lik forrige lenkes utgangsfart, eller 0 ved oppstart.

Forløpet er beskrevet nærmere i Vedlegg 3.

#### 2.4.11 Applikasjon for administrativ bruk

Komponenter fra DAKAT-prosjektet<sup>4</sup> ble brukt til å sette opp en liten applikasjon for innlegging av kjøretøy, aktører, rutenavn, tilhengere m.m., og til å teste fartsmodellen på gitte transporter.

Koden for å kjøre Fartsmodellen på transportene var også inkludert. Noen eksempler på brukergrensesnittet er vist i Vedlegg 4.

#### 2.4.12 Utvikling av datasystemer

TRIONA stod for utvikling av Web-tjeneste mot terminal, med info om de aktuelle kjøretøyene, og tilrettelegging for formidling av vegmeldinger via TRIP til GOFER-systemet. SINTEF stod for utvikling av tjenester til klient (Samsung Galaxy Tab), installering av Samsung Galaxy Tab i testkjøretøyene, oppfølging mht. drift/service på dette utstyret, og drifting av server.

---

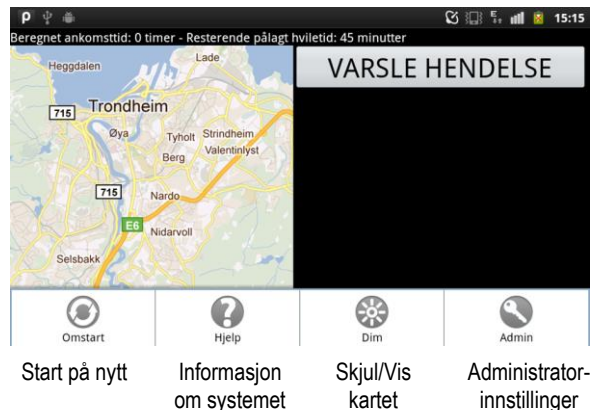
<sup>4</sup> Datakatalog for Statens vegvesen, med oversikt over alle objektene som utgjør kjernen i Nasjonal Vegdatabank (NVDB)

## 2.5 Grafisk brukergrensesnitt, klient

Google Maps ble benyttet som kartunderlag. Lisensbegrensninger gjorde det ikke mulig å vise *ruter* på kartet, men meldinger og kjøretøyposisjon kunne vises på denne kartbakgrunnen.



Figur 6: Standard skjerm bilde med kjøretøyposisjon avgitt på kartbakgrunn



Figur 7: Menyvalg

Kjøretøyet ble identifisert på kartet med en rød prikk (se Figur 6).

Symboler ble benyttet for å identifisere informasjonskilde for meldinger til sjåfør:

- Vegmeldinger ble angitt på kartet med skiltet "Annen fare"
- GOFER-meldinger ble angitt på kartet med GOFER-logoen

Innholdet i meldingene ble vist på høyre halvdel av skjermen når kjøretøyet var innen 5 km fra stedet meldingen var knyttet til. Under symbolet ble det angitt hvor langt det var fram til dette stedet.

- I de forhåndsdefinerte GOFER-meldingene ble meldingsinnholdet formidlet kun med symbol, uten forklarende tekst. Disse meldingene ble vist på grønn bakgrunn.
- I denne demonstratoren ble "Annen fare"-symbolet benyttet for alle de ordinære vegmeldingene, uavhengig av innhold. I et virkelig system vil det være naturlig å benytte symbolbruk som samsvarer med meldingsinnholdet også for denne typen meldinger. Friteksten som er knyttet til vegmeldingene var tilgjengelig for sjåførene så lenge bilen står i ro, ved klikk på skiltsymbolet for meldingen. Vegmeldingssymbolene ble vist på hvit bakgrunn

Kartvisningen kunne tas bort/hentes fram via et meny-valg (Figur 7), for å redusere "lysforurensning" i førerhytta ved kjøring i mørke.

## 2.6 Sjåførenes oppgaver i demonstratoren

Testen omfattet alle turer fra Oslo til Trondheim i løpet av testperioden, og det utleverte utstyret skulle slås på og benyttes som vist i instruksene, for alle disse turene. Utstyret skulle ikke slås av eller benyttes til andre aktiviteter mens turen fra Oslo til Trondheim pågikk. Dersom sjåførene var lojale mot testopplegget i demonstratorperioden, skulle de få beholde ombordenheten (Samsung Galaxy Tab) etter ferdig test. Samtidig med at ombordenheten ble montert i bilen, fikk hver sjåfør en gjennomgang av oppgavene og en detaljert

informasjonsbrosjyre med trinn-for-trinn beskrivelse. Hovedtrekkene er gjengitt i det følgende.

### 2.6.1 Ved oppstart fra Oslo

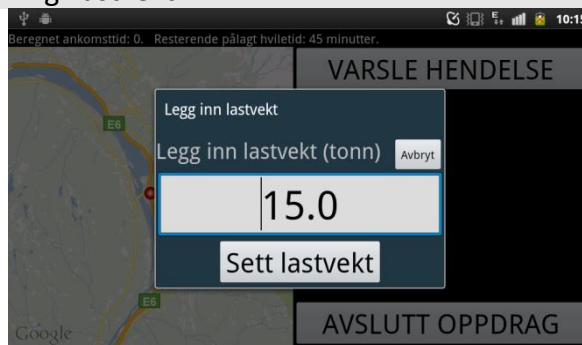
#### Angi kjørerute mellom Oslo og Trondheim



Ettersom testen var begrenset, ble formelverket for Fartsmodellen bare kodet opp for vegnettet på rutene mellom Oslo og Trondheim via hhv. Østerdalen og Gudbrandsdalen.

Figur 8: Skjerm bilde, angi kjørerute

#### Angi lastvekt



Lastvekten inngikk i beregningen av forventet ankomsttid til Trondheim. Den ble lagt inn manuelt av sjåføren. I et fullt utviklet system kan en se for seg at denne informasjonen kan hentes ut fra fraktinformasjonen hos transportøren, eller fra datasystemene i kjøretøyet.

I tillegg inngikk også informasjon om kjøretøyets motoreffekt, men denne informasjonen ble hentet direkte fra en "mini-utgave" av Autosys (kjøretøyregisteret), og sjåførene trengte derfor ikke å legge inn denne informasjonen.

Figur 9: Skjerm bilde, angi lastvekt

#### Angi hvor i Trondheimsområdet første leveringssted ligger



Igjen på grunn av at testen var begrenset, ble ikke formelverket for Fartsmodellen kodet opp på et fullt vegnett i Trondheimsområdet. Alternativene på skjermen representerte de vanligste leveringsområdene, og sjåførene ble bedt om å velge blant disse. Om ingen av områdene passet, kunne sjåføren velge "Annet".

Figur 10: Skjerm bilde, angi leveringssted

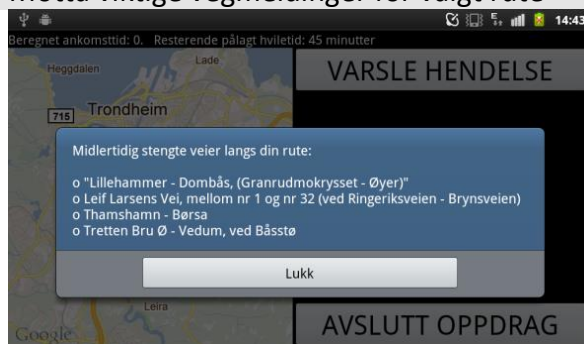
### Motta oppsummering av oppstart-informasjon



Før oppstart fikk sjåførene så opp et skjermbilde med den informasjonen de hadde lagt inn for den aktuelle turen. De kunne så velge å gå tilbake for evt. å korrigere feil i denne informasjonen, eller velge "start oppdrag" for å logge seg på GOFER-systemet.

Figur 11: Skjerm bilde, oppsummering av oppstart-informasjon

### Motta viktige vegmeldinger for valgt rute

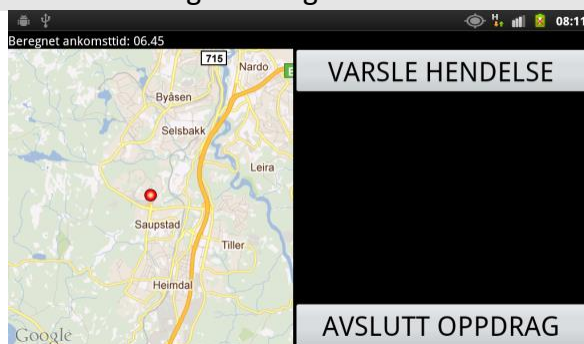


Evt. viktige vegmeldinger for valgt kjørerute ble så vist på skjermen, og sjåføren kunne evt. vurdere om igjen rutevalget (Østerdalen eller Gudbrandsdalen)

Figur 12: Skjerm bilde, viktige vegmeldinger for valgt rute

## 2.6.2 Under veis fra Oslo til Trondheim

### Sende melding til kolleger



Figur 13: Standard skjermbilde med "Varsle hendelse"

Ved å velge "Varsle hendelse" på det standard skjermbildet, fikk sjåføren opp et eget skjermbilde for varsling av spesielle forhold langs veien.



De ulike meldingene ble identifisert kun med et symbol, uten forklarende tekst. Dette ble gjort for at operasjonen med å sende melding skulle kreve så lite oppmerksomhet fra føreren som mulig. Ved å velge ett av ikonene, ble melding "lagt igjen" til de sjåførene som fulgte etter på samme vegstrekning.

Figur 14: Skjerm bilde, varsle om spesielle kjøreforhold

Innholdet i meldingene var basert på forslag fra sjåførene. Hvor lenge hver meldingstype lå ute, varierte avhengig av en skjønsmessig vurdering av hvor varig de ulike forholdene ble ansett å være. Figur 15 viser hvilke meldinger som var inkludert i demonstratoren, og utløpstiden for hver av dem.



Figur 15: Meldingsinnhold og utløpstid for GOFER-meldingene

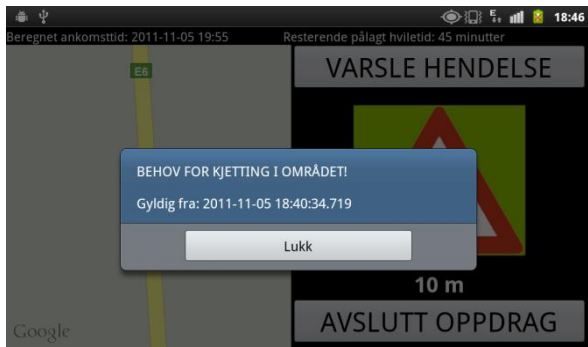
#### Motta meldinger fra andre sjåførere



Figur 16: Skjerm bilde, motta melding fra sjåfør

Meldinger fra andre sjåførere i GOFER-testen ble angitt i kartet med GOFER-logoen. Disse meldingene ble vist med grønn bakgrunn for skiltsymbolet.

Under skiltsymbolet ble det angitt avstand til der meldingen ble sendt fra. Varslene ble vist når dette lå innen 5 km foran bilen.



Figur 17: Skjerm bilde, tilleggstekst GOFER-melding

Enkel forklarende tekst kunne hentes fram ved å peke på skiltsymbolet.

GOFER-meldingene hadde ulik "utløpstid", dvs. hvor lenge de ble vist i systemet, se Figur 15.

### Motta ordinære vegmeldinger



Figur 18: Skjerm bilde, motta vegmelding

Ordinære vegmeldinger ble angitt i kartet med "Annen fare"-skilt. Disse meldingene ble vist med hvit bakgrunn for skiltsymbolet.

Under skiltsymbolet ble det angitt avstand til "start-stedet" for vegmeldingen. Varslene ble vist når dette lå innen 5 km foran bilen.



Figur 19: Skjerm bilde, tilleggstekst vegmelding

Forklarende tekst kunne hentes fram ved å peke på skiltsymbolet. Av sikkerhetshensyn ble ikke den fulle teksten for vegmeldingene gjort tilgjengelig mens bilen var i fart. Vegmeldingene hadde "utløpstid" hentet fra vegmeldingssystemet.

### Registrere hviletid



Figur 20: Skjerm bilde, registrer hviletid

Etter stopp på mer enn 15 minutter, spurte systemet om stoppet inngikk i den pålagte hviletiden, eller om det skyldtes andre forhold. Dette ble gjort for å korrigere beregnet ankomsttidspunkt med hvor mye av den pålagte hviletiden som til en hver tid var benyttet.



## Motta varsel om anbefalt rute i Trondheim og booket plass på terminal



Ettersom det ikke fantes noe eget baksystem for angivelse av anbefalt kjørerute inn mot bestemmelsesstedet i Trondheim, var disse meldingene forhåndsdefinerte. De varierte ut fra leveringssted i Trondheimsområdet, f.eks.:

- Brattøra: Osloveien
- Sluppen: E6 mot sentrum
- Lade: Omkjøringsveien

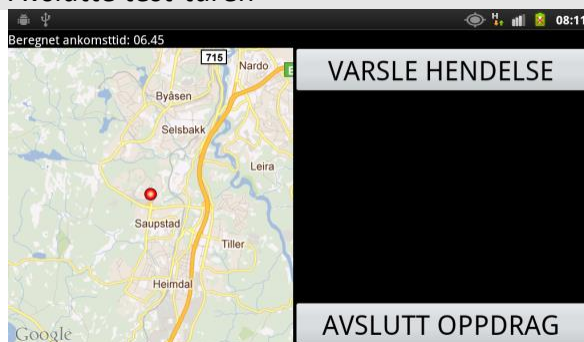
Figur 21: Skjerm bilde, varsel om anbefalt kjørerute og plass på terminal

Booket plass på terminal eller hos leverandør var heller ikke reell, men eksempel på en funksjon som kan inkluderes i et GOFER-system.

### 2.6.3 Ved ankomst til første leveringssted i Trondheim

Dette var en begrenset test, uten et fullt utviklet system for å følge bilene til flere leveringssteder. Test-turen ble derfor avsluttet ved første "stoppested" i Trondheimsområdet.

#### Avslutte test-turen



Figur 22: Standard skjerm bilde med "Avslutt oppdrag"

Ved å velge "Avslutt oppdrag" på det standard skjerm bildet, fikk sjåføren opp Skjerm bilde for å bekrefte av turen var over.



Figur 23: Skjerm bilde, bekreft slutt på test-tur

Dette skjerm bildet ga en oppsummering av hvilken rute bilen hadde fulgt, hvor bestemmelsesstedet lå i Trondheim, og hvor lang tid som var gått siden turen startet i Oslo.

## 2.7 Automatisk logging

### 2.7.1 Fra ombordenheten

Data ble logget fortløpende hvert 5. sekund til fil fra det innmonterte kommunikasjonsutstyret. Loggefrekvensen ble valgt ut fra hensyn til datakapasitet i mobilnettet. Følgende ble logget:

- Posisjon (XYZ)
- fart
- retning
- location provider
- tilkoblingstype

I tillegg er datamodellen forberedt for logging av følgende typer data:

- nøyaktighet posisjon (antall satellitter)
- gyro
- akselerometer
- lysforhold i førerhytta
- signalstyrke

Loggedataene for hver enkelt tur kunne kobles mot informasjon som ble gitt ved turens oppstart, om bl.a. kjøretøyets totalvekt og destinasjon, og med informasjon hentet fra Autosys om kjøretøyet og hengeren.

### 2.7.2 Fra kjøretøy

Det ble vurdert å hente ut driftsdata direkte fra enkelte av kjøretøyene. Dette ville være data som bl.a. beskriver miljø- og energirelaterte forhold. På grunn av begrensninger i oppsettet i de aktuelle kjøretøyene, var dette ikke mulig å gjennomføre i GOFER.

## 2.8 Gjennomføring av testen

### 2.8.1 Testkjøretøyene

Bring stilte til rådighet biler og sjåfører som trafikkerte strekningen Oslo – Trondheim på fast basis. Bring plukket ut 10 biler som skulle inngå i testen, med bl.a. registreringsnummer for trekkvogn og henger, og sjåførnavn.

Høyde på kjøretøyene avhenger av hengerne som benyttes. Disse ligger inne i Autosys, med bl.a. opplysninger om høyde. Statens vegvesen leverte informasjon fra Autosys om de aktuelle kjøretøyene og hengerne.

Bilene kommer til Trondheim på kvelds-/nattestid annenhver natt før virkedager. Sjåførene tar ut sin hviletid i Trondheim før retur til Oslo. Ombordenhetene ble montert mens bilene var i Trondheim.

### 2.8.2 Testperiode

Testen Oslo - Trondheim startet i midten av november 2011, og pågikk ut januar måned 2012.

### **2.8.3 Dokumentasjon**

Testen ble dokumentert i form av en kort video. Videoen ble laget i både norsk og engelsk versjon, og begge er tilgjengelig for nedlasting på prosjektets hjemmeside.

### **2.8.4 Evaluering**

Evaluering av testen fokuserer i stor grad på nytteverdi og potensiale for videreutvikling av de ulike informasjonstypene som ble formidlet mellom de involverte aktørene.

Evalueringen baseres på skriftlige intervju med sjåførene, og muntlige tilbakemeldinger fra de øvrige aktørene. Statistikk fra driften av testen inngår også i evalueringen.

Resultatene fra evalueringen er presentert i L3.0 Evaluering (GOFER, 2013b).

### 3 Tungbilsimulator i Trondheim

Bruk av simulator var et supplement til ”live”-demonstratoren, og dekket funksjonalitet det var vanskelig å få til ute på veien.

Testen ble gjennomført med syv erfarne sjåfører fra Bring, og omfattet kjøring på oppdatert Trondheims-vegnett, fra City Syd til Nidarosdomen, med kodete lyskryss og kollektivfelt. Trafikkbelastning ble hentet fra trafikktelegger og registreringer i vegnettet.

Testen ble gjennomført over to dager i januar 2012.

#### 3.1 Funksjonalitet

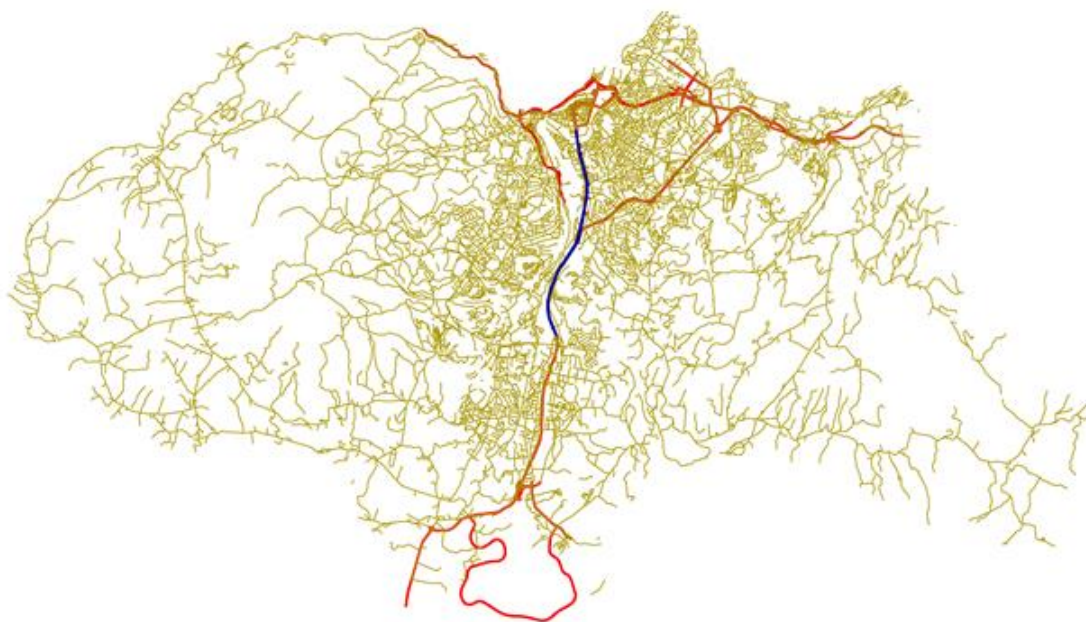
Testen fokuserte på effekter av:

- Prioritering i kollektivfelt i mellomrush-perioden
- Prioritering i lyskryss i lavtrafikkperiode (natt)

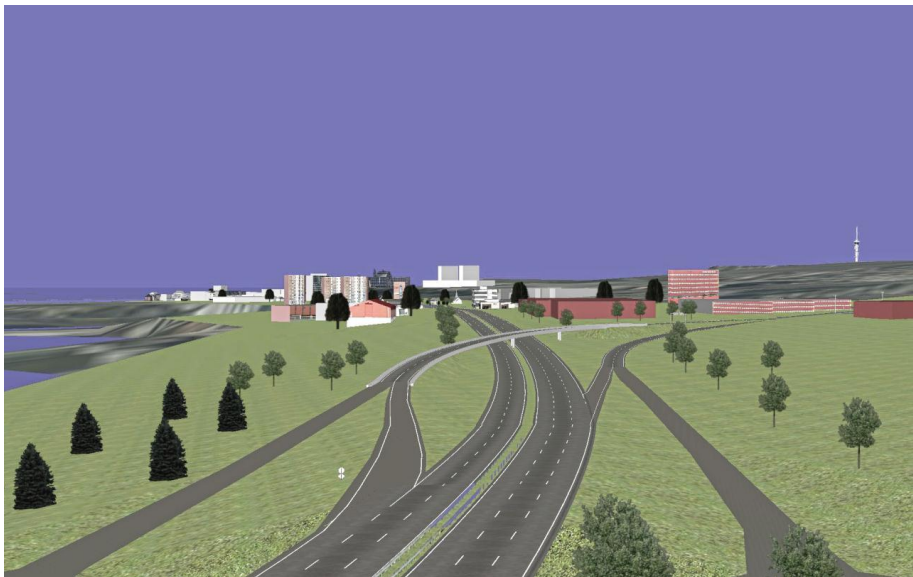
Hver sjåfør kjørte teststrekningen i alt fire ganger, under ulike forhold:

1. På dagtid, med trafikk som tilsvarte mellomrush-perioden:
  - a) med dagens trafikkregler, der tungbiler ikke kan benytte kollektivfelt
  - b) med endrede regler, der tungbiler kan benytte kollektivfelt
2. På nattetid, med lav trafikkbelastning (ikke tilgang til kollektivfelt):
  - a) med dagens lysregulering i kryssene
  - b) med "grønn bølge" for tungbiler i kryssene

Rekkefølgen på disse fire ulike settene med betingelser varierte fra sjåfør til sjåfør.



Figur 24: Hele vegnett i simulatoren i rødt. Registreringsstrekning merket blått



Figur 25: Visuell framstilling i kjøresimulatoren – E6 ved Sluppen, retning mot sentrum



Figur 26: Visuell framstilling i kjøresimulatoren – visning i forsøkene, Trondheim sentrum med førerkabinen

Tabell 2: De fire test-scenariene i kjøresimulatoren

Situasjon	Eksterne forhold	Uten prioritering	Med prioritering
Mellom rush	Dagtid, mye trafikk	Normale trafikkregler	Tilgang til kollektivfelt
Lavtrafikk	Nattestid, lite trafikk	Normale trafikkregler	Prioritering i lyskryss (grønn bølge)



Figur 27: Visuell framstilling i kjøresimulatorene – scaneriom med prioriteringstiltak i mellomrush-periode



Figur 28: Visuell framstilling i kjøresimulatorene – scanerio med prioriteringstiltak i lavtrafikkperiode

## 3.2 Praktiske forhold og forberedelser

### 3.2.1 Valg av teststrekning

Vegstrekningen som inngikk i testen startet ved City Syd. I de opprinnelige planene var tanken å føre teststrekningen helt fram til terminalen på Brattøra, men dette ble senere omgjort, og teststrekningen ble avsluttet ved krysset mellom Prinsens gate og Bispegata, dvs. ved avkjøring til Nidarosdomen. Beslutningen om å stoppe der var bl.a. basert på følgende vurderinger og forhold:

- Ingen av prioriteringstiltakene som skulle studeres, hadde effekt i Midtbyen
- Kortere strekning ville gi redusert tid i testkjøringene, særlig på dagtid-scenariene (med/uten kollektivfelt)
- Kortere strekning ville gi redusert risiko for simulatorsyke (som øker dramatisk med svinger)
- Det var ikke kodet opp bygninger langs Kjøpmannsgata, og å gjøre dette ville kreve ekstra ressurser

### 3.2.2 Vegnett- og kryssbeskrivelse, skilting

Vegnettsbeskrivelsen ble i hovedsak utarbeidet av Statens vegvesen. I vegnettsbeskrivelsen inngikk koding av vegnettet, lyskryssene og skilting langs teststrekningen. Der det fantes ferdig kodet vegnett, ble dette benyttet. Det krevdes noe tilpasning av den eksisterende vegnettskodingen for å sikre framkommelighet for tungbiler i kryssene.

Eksisterende modeller over veinettet var delvis mangelfulle. Flere kryss manglet fullstendig i modellen for den påkrevete strekningen. Noen kryss eksisterte kun i visuell form, og manglet logisk form, som var påkrevet for at simuleringer av trafikk på veinettet skulle kunne skje. Dette gjorde at sidetrafikk ikke kunne ledes inn på veinettet ved flere nøkkelområder.

I tillegg manglet de aller fleste kryss både visuell og logisk oppsett av trafikklys. For dette forsøket var visuell representasjon og logisk oppbygging av lyskryss av stor viktighet. En fullstendig gjennomgang av den aktuelle delen av modellen ble utført, og flere manglende deler av det visuelle og det logiske veinettet ble lagt til.

Fokus for arbeidet med det logiske veinettet var på områder rundt kryss, mens det visuelle veinettet ble gjennomgått fullstendig for hele den aktuelle delen av modellen. Fortau ble lagt til hvor aktuelt, og teksturer i veibanen ble tilpasset for å optimalisere visuell gjengivelse av strekningen. I tillegg ble flere versjoner med veimerking og skilting lagt til, da disse avvek noe i hvert scenario med hensyn på bruk av kollektivfelt.

### 3.2.3 Registrering og koding av trafikkforhold

Det ble gjennomført registreringer av kjøretid, kombinert med videoopptak av trafikksituasjonen. Dette ble gjort av Statens vegvesen.

Registreringene dekket trafikk i retning sentrum, fra lyskryss ved Siemens til kryss Bispegata. Kjørefart som ble benyttet ved registreringene var fartsgrensen, hvis mulig, ellers ”go with the flow”.

Registreringene dekket følgende situasjoner:

- Mellomrush, med og uten prioritering i kollektivfelt. Det ble gjort samtidige registreringer av kjøretid i og utenfor kollektivfelt. Registreringene ble gjort mellom kl 12 og 14, med el-bil i kollektivfelt og ordinært kjøretøy i det andre feltet.
- Lavtrafikkperiode, med registreringer av kjøretid i situasjon med vanlig faseplan for natt i lyskryssene. Registreringene ble gjort mellom kl 00 og 02.

I arbeidet med koding av trafikksituasjon for hvert scenario ble faseplaner for lyskryssene benyttet for initial for å lage den initiale situasjonen Fra 2010 av har derimot disse faseplanene, for flere av lyskryssene i strekningen, blitt utbyttet med et Spot Utiopia system som dynamisk samkjører flere påfølgende kryss, samt støtter prioritering av busstrafikk.

På grunn av dette ble flere av lyskryssene i simulatoren kodet til å kunne settes med faseplan valgt av simulatorens operatør. Initial oppførsel til lyskryssene ble satt til å vise grønt lys til sjåfør, mens operatøren kan sette at lyset skulle skifte til rødt når en sjåføren passerer et gitt punkt før krysset, og økt trafikk fra sidevei ble dermed ledet inn på veinettet. Det er verdt å merke seg at en sjåfør delvis selv kunne påvirket farten til trafikken rundt seg, ut i fra hvor

aggressivt han kjørte, og kunne dermed i teorien ha påvirket situasjonen i disse kryssene. I praksis var dette derimot ikke tilfelle, da tungbilen i simulatoren ikke var satt opp med tilstrekkelig motorkraft til at situasjonen rundt lysskifte i kryss ble påvirket.

I arbeidet med definerings av trafikksituasjon for hvert scenario, ble videoopptakene brukt som "fasit". Flere opptak av den virkelige trafikksituasjonen ble gjort for hvert scenario, over flere dager og evaluert av Statens Vegvesen. Ut i fra disse ble bl.a. omfang av trafikk som kommer inn fra sideveiene, og hvilke faseplan operatøren benyttet på spot-styrte lyskryss bestemt for de forskjellige scenario.

### 3.2.4 Selve simulatoren

SINTEF var ansvarlig for arbeidet med og i selve tungbilsimulatoren.

#### *Kjøreopplevelse:*

For å få tilfredsstillende bevegelser i førerstedet i tungbilsimulatoren ble det investert i en ny "Advanced" programmodul for simulatoren. Før gjennomføring av forsøkene ble det derfor utført kalibrering, testing og justering av simulatoren med denne nye modulen. I denne prosessen deltok elever ved kjøreskolelærer-linjen ved HiNT som testpersoner.

#### *Uttak av loggedata:*

Det ble lagt til rette for uttak av GPS-data for teststrekningen. Det var også planlagt å ta ut beregnede tall for drivstofforbruk og utslipp. Det viste seg imidlertid å være vanskelig å få tatt ut så detaljerte data som ønskelig. Resultater mht. miljøeffekter som inngår i evalueringen av testen i simulatoren, er derfor basert på posisjonsdata og formelverk som beskriver forhold mellom hastighet og drivstofforbruk.

## 3.3 Gjennomføring av testen

SINTEF var ansvarlig for rekruttering av testsjåfører, utarbeiding av undersøkelsesopplegg, gjennomføring av testene, og dokumentering av disse.

### 3.3.1 Testsjåfører

Det var først tanken å rekruttere noen av Brings ordinære sjåfører mens de var i Trondheim. For å unngå evt. konflikter med pålagte hviletiden og evt. problemer som følge av "simulator-syke", ble denne løsningen skrinlagt. Sjåførene i simulatortesten ble rekruttert fra Bring. De hadde lang fartstid som tungbilsjåfører, men hadde ikke lenger sitt daglige virke på veien.

### 3.3.2 Timeplan

Testen ble gjennomført over to dager, med fire sjåfører hver dag – to før og to etter lunsj. De to sjåførene som var satt opp i samme tidsperiode, alternerte på å kjøre i simulatoren. Før testkjøringen startet, fikk hver person en "oppvarmingsrunde" på ca. 10 minutter.

Scenariene ble satt opp i ulik rekkefølge for testsjåførene, slik at det ikke skulle bli noen systematiske rekkefølge-effekter i testen. For at deltakerne skulle kunne vurdere effekten av



de to prioriteringstiltakene uten å blande dem sammen, ble imidlertid hhv. de to mellomrush-scenariene og de to lavtrafikk-scenariene gjennomført samlet. Det varierte om det var mellomrush- eller lavtrafikkscenariene som ble kjørt først, og likeledes om det var scenariet med eller uten prioritering som kom først av de to.

Etter hvert par av scenarier fikk deltakerne presentert en serie spørsmål knyttet til kjøreopplevelsen og prioriteringstiltakene som de hadde testet ut i kjøresimulatoren. Svarene på disse spørsmålene er presentert i L3.0 Evaluering (GOFER, 2013b).

### 3.3.3 Logging

Simulatordata ble logget 20 ganger i sekundet, og følgende ble lagret:

- Tid gått fra scenariet startet, i sekunder
- Fartsgrense på den aktuelle strekningen
- Bilens fart i km/t (negativ verdi ved rygging)
- X-koordinat tilsvarende UTM32
- Y-koordinat tilsvarende UTM32
- Avstand til forankjørende kjøretøy i samme felt (meter)
- Avstand fra bilens venstre side til kjørefeltets linjedeler (meter)
- Avstand fra bilens venstre side til vegbanens ytterkant (meter)
- Gasspedalens posisjon mellom [0 (ikke pådrag), 1 (maks pådrag)]
- Kraft som er påført bremsepedalen (newton)
- Rattets posisjon mellom [-1, 1]
- Utkjørt distanse fra scenariets start (meter)
- Avstand i luftlinje til forankjørende kjøretøy
- Kjøretøyets gir
- Turtall
- X-koordinat tilsvarende UTM32 for forankjørende kjøretøy
- Y-koordinat tilsvarende UTM32 for forankjørende kjøretøy
- Kjøretøyets rotasjonsvinkel rundt Y-aksen
- Forankjørende kjøretøys fart i km/t
- Avstand mellom to angitte kjøretøy langs veiens senterlinje (meter)

### 3.3.4 Dokumentasjon

Testen ble dokumentert i form av en kort video. Videoen ble laget i både norsk og engelsk versjon, og begge ligger tilgjengelig for nedlasting på prosjektets hjemmeside.

### 3.3.5 Evaluering

Evaluering av testen i tungbilsimulatoren fokuserer på hvilke effekter prioriteringen i scenariene kan gi mht.:

- Tidsbruk på teststrekningen
- Energiforbruk på teststrekningen - basert på GPS-baserte fartsdata fra simulatorkjøringen og formelverk fra Artemis-prosjektet
- Utslipp på teststrekningen - basert på GPS-baserte fartsdata fra simulatorkjøringen og formelverk fra Artemis-prosjektet
- Støy på teststrekningen - basert på endringer i antall stopp/start langs traséen.

I tillegg inkluderer evalueringen sjåførenes vurdering av hvordan de opplevde testen og bruk av kjøresimulator for å studere denne typen prioriteringstiltak.

Resultatene fra evalueringen er presentert i L3.0 Evaluering (GOFER, 2013b).

## 4 Demonstrator Oslo

I vegnettet som finnes rundt Alnabru i dag, er det ikke mulig å prioritere én kjøretøytype fremfor en annen ved for eksempel feltbruk og signalkryss. Det ble derfor relativt tidlig i prosjektperioden bestemt at demonstratoren i Oslo skulle være en simuleringsmodell. Målet med bruk av simuleringsmodellen er å beregne effekter av et styringssystem som GOFER for hele trafikkstrømmer, i motsetning til de to andre demonstratorene, som fokuserte på et utvalg tungbiler. Demonstrator Oslo komplementerer dermed kjøresimulator-testen og live-demonstratoren, ved å inkludere effekter på en større skala.

Arbeidet med simuleringsmodellen inngår i et doktorgradsarbeid ved NTNU, og er utført av stipendiat Odd Andre Hjelkrem.

### 4.1 Modellen

Modellen er bygd opp i simuleringsverktøyet Aimsun. Det er en mikroskopisk modell, som vil si at de dynamiske egenskapene til hvert kjøretøy i modellen blir oppdatert med et fast tidsintervall, for eksempel hvert sekund.

Dynamikken til kjøretøyene er bestemt av atferdsmodeller som tilordner hastighet og posisjon til hvert kjøretøy, basert på dynamiske egenskaper til kjøretøyene/trafikanter i nærheten. Hvert kjøretøy forholder seg også til vegnettet gjennom bestemte parametere, for eksempel fartsgrense, stigning, kurvatur og svingebevegelser i kryss. Dette er faste parametere som relativt enkelt kan hentes fra databaser som NVDB. Atferdsparametrene er stokastiske, som innebærer at atferden til trafikantene har en viss variasjon. Grunnen til dette er at en vil få en mer realistisk simuleringsmodell ved å unngå at alle trafikanter oppfører seg likt. I tillegg til de stokastiske atferdsparametrene, er det også for hvert kjøretøy definert et sett av tekniske parametre som bestemmer for eksempel kjøretøyets vekt, størrelse, motortype, osv.

Det er en del faste data som blir beregnet av simuleringsmodellen, for eksempel posisjon, hastighet, hastighetsendring, reisetid, utslipp og energiforbruk. Dette blir lagret for hvert tidsintervall, i tillegg til at det er mulig å ta ut data for aggregerte tidsintervall eller for større områder i modellen. Modellen kan også benyttes til å demonstrere bestemte funksjoner gjennom animasjoner av trafikkavviklingen.

Arbeidet med å etablere modellen er utført etter retningslinjene anbefalt i Federal Highway Administrations håndbok "Guidelines for Applying Traffic Microsimulation Modelling Software" (USDOT, FHA, 2003). Retningslinjene består av følgende punkter:

- Identifikasjon av mål og omfang
- Innsamling og tilrettelegging av data
- Utvikling av basismodell
- Feilsøking
- Kalibrering
- Analyse av alternativ

De fem første punktene er beskrevet i dette kapitlet, mens analysene er presentert i L3.0 Evaluering (GOFER, 2013b).

## 4.2 Mål og omfang

I prosjektplanen til GOFER står det at ”Målet i GOFER er å etablere løsninger som muliggjør kontroll og regulering av godstransport i byområder på samme måte som vi kjenner fra flykontrollen.” Dette skal demonstreres i tre parallelle demonstratorer med forskjellig innhold. Mens to av demonstratorene har fokus på å demonstrere effekter som påvirker tunge kjøretøy, skal demonstratoren i Oslo i tillegg se på effekter på den øvrige trafikkstrømmen. Videre er det beskrevet konkrete mål for Oslo-demonstratoren og omfanget av modellen.

### Mål:

Målene for Oslo-demonstratoren kan oppsummeres i følgende delmål:

- Samle inn data fra aktuelle datakilder i modellområdet.
- Etablere en basismodell i Aimsun.
- Kalibrere modellen til de innsamlede data.
- Kjøre simuleringer med forskjellige scenarier for å vise effekter av GOFER

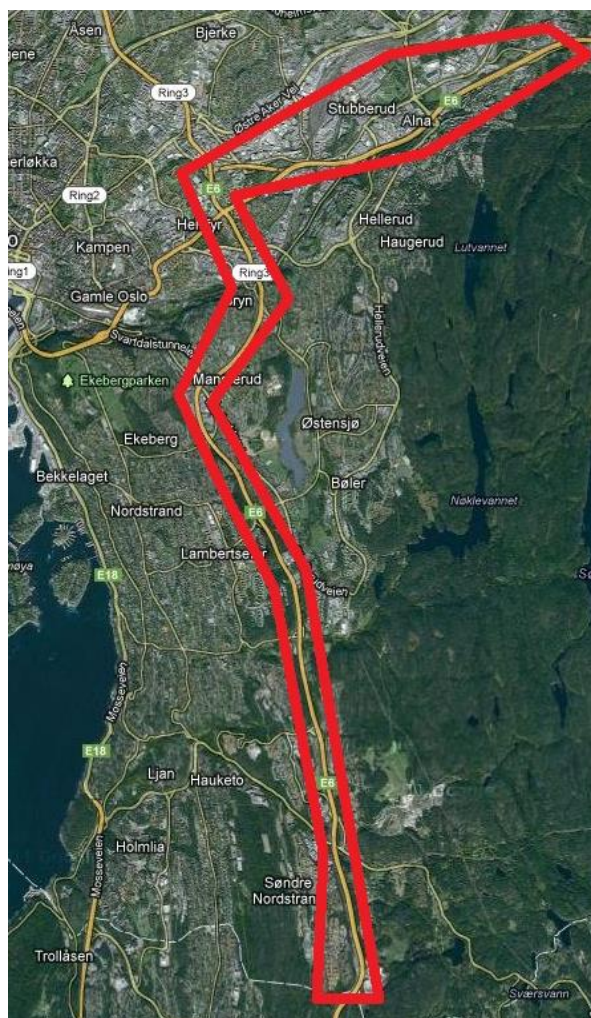
### Omfang:

Vegnettet i modellområdet er i stadig endring. Selv om det kunne være interessant å inkludere planene i modellen, er det først og fremst viktig å demonstrere GOFER-funksjonaliteten. Derfor baserer modellen seg på dagens vegnett.

Fysisk er modellområdet avgrenset til å omfatte E6 fra kommunegrensen i sør til kommunegrensen i øst. I tillegg er vegnettet i Alnabruområdet inkludert. Figur 29 viser en skisse av hvor grensene til modellen går. Det omliggende vegnettet er ellers ikke med i modellen, men av- og påkjøringsramper er inkludert. Dette innebærer at en stor del av modellen består av flerfelts motorveg.

For å begrense arbeidsmengden med å samle inn data og kalibrere modellen, er det kun etablert en simuleringsmodell for tidsperioden 07.00 til 09.00. Modellen skal demonstrere effekter av GOFER-funksjonaliteten i retningen inn mot Alnabru fra den sørlige ankomsten på E6.

Tiltak som påvirker trafikantene i modellen vil i varierende grad føre til at noen trafikanter vil velge å kjøre en annen veg. Fordi simuleringsmodellen inneholder et avgrenset område, er det *ikke mulig å simulere endringer i rutevalg*. Dersom det for eksempel skal simuleres en hendelse i vegnettet som forårsaker køer, vil det i virkeligheten være noen



Figur 29: Avgrensning av modellområdet i Oslo

kjøretøy som velger å kjøre en annen veg. For å få med endringer i rutevalgene, må det først gjøres beregninger i makromodeller for Oslo-området, deretter må det gjøres nye beregninger i Contram, før resultatene importeres på nytt inn i simuleringsmodellen. Denne prosessen er gjennomførbar, men tidkrevende. For å begrense omfanget på arbeidet med å etablere selve modellen, er dette ikke gjennomført i dette prosjektet. Beregningene vil dermed ikke ta hensyn til endring i rutevalg i modellen. Det er spesielt viktig å tenke på når en vurderer resulterende effekter for andre trafikanter i modellen.

### 4.3 Innsamling og tilrettelegging av data

For å kunne kalibrere en simuleringsmodell, må en ha data å kalibrere til. Det innebærer at en må kunne beskrive trafikkstrømmene i modellområdet ved trafikkvolum, hastigheter og generell kjennskap til trafikksituasjonen. En kalibrert modell er en representativ modell av trafikksituasjonen for den tidsperioden en har data å sammenligne med.

På grunn av at det er tidkrevende og kostbart å gjennomføre trafikktegninger av større omfang, er det hovedsakelig benyttet de tellinger og registreringer som allerede finnes. Bare for noen få viktige områder er det gjort supplerende datainnsamling. Dette innebærer at den kalibrerte modellen ikke nødvendigvis vil være en eksakt kopi av trafikksituasjonen i modellområdet, men et eksempel på hvordan trafikksituasjonen kan være.

Ved kalibrering av en simuleringsmodell er det ikke hensiktsmessig å benytte data fra forskjellige tidsperioder. Hovedgrunnen til dette er at det kan være forskjeller i trafikkstrømmene fra dag til dag, som gjør det umulig å tilpasse modellen til de innsamlede data. Derfor er det data fra samme tidsperiode for tellepunktene i kategorien nivå 1 som er benyttet i kalibreringen. Fordi tellepunktene på nivå 1 ikke dekker store nok deler av modellområdet, er det også samlet inn data fra andre korttidstegninger. De er derimot ikke samlet inn i samme tidsperiode, og er behandlet som sekundære data i kalibreringen.

Det er videre gitt en oversikt over de data som er samlet inn og benyttet til kalibreringen av modellen, kategorisert i trafikktegninger og reisetider.

#### *Data fra tellepunkt:*

De tellepunktene som er på nivå 1 er Mortensrud, Manglerud, Ulvensplitten, Teisen og Jerikoveien, samlet inn i perioden 14.03.11 til 19.04.11. Det er i tillegg benyttet data fra tellinger på Strømsveien og Tvetenveien, rundkjøringer på Tvetenveien ved E6, og rampetellinger langs E6. Informasjon om tellepunktene er oppsummert i Tabell 3.

Tabell 3. Oversikt over tellepunkt

Tellepunkt	ID	Dato	Oppløsning
<b>Nivå 1-punkt:</b>			
Mortensrud	318	14/3-11 til 19/4-11	5 minutt
Manglerud	307	14/3-11 til 19/4-11	5 minutt
Ulvensplitten	61, 62 og 63	14/3-11 til 19/4-11	5 minutt
Teisen	79	14/3-11 til 19/4-11	5 minutt
Jerikoveien	58, 59 og 60	14/3-11 til 19/4-11	5 minutt
<b>Korttidstillinger:</b>			
Nedre Kalbakkvei	300362	6/4-10 til 21/4-10	1 time
Tvetenveien	300087	17/3-10 til 16/4-10	1 time
Rundkjøring ved E6 Tvetenveien	-	22/3-11 og 29/3-11	15 minutt
Rampe Klemetsrud	-	10/1-12	15 minutt
Rampe Mortensrud	-	31/1-12	15 minutt
Rampe Skullerud	-	1/2-12 og 7/2-12	15 minutt

#### Reisetider:

For å beregne reisetider gjennom modellområdet er det benyttet registreringer gjort med GPS av BRING sine distribusjonsbiler i november 2010.

GPS-koordinatene er hentet inn i ArcGIS for viderebehandling. Det er deretter laget en linje mellom to etterfølgende punkter og beregnet hastighet ut ifra lengden på linjen og tiden mellom GPS-registreringene i hver ende av linjen. Fordi lengden på linjen i noen tilfeller vil være lengre enn den faktiske avstanden mellom punktene, vil i noen tilfeller føre til en undervurdering av hastigheten. Allikevel vil ikke denne feilen være for stor, da det er hentet ut gjennomsnittshastighet for tilnærmet rette strekninger. Denne metoden gjør at det ikke er hensiktsmessig å bruke reisetid for hver linje, da hver linje har et unikt start- og slutt punkt. Det er derfor beregnet gjennomsnittlig hastighet innenfor hvert område, som blir benyttet videre i kalibreringsprosessen.

Modellområdet er delt inn i 9 områder, som vist i Figur 30. GPS-registreringer som er gjort innenfor de 9 områdene, er tatt med i beregningen av gjennomsnittshastighet. Det er kun brukt reisetider i perioden 07.00 til 09.00 på morgenen. For å kunne si noe om spredningen på reisetidene er det beregnet et 95 % konfidensintervall for gjennomsnittlig



Figur 30: Oversikt over strekninger med registrerte gjennomsnittshastigheter

hastighet  $v$ . Fordi variansen til alle reisetider er ukjent, er den estimert fra data. Konfidensintervallet for  $v$  kan dermed uttrykkes på følgende måte, gitt at vi antar en normalfordeling av reisetider:

$$v = \bar{v} \pm t_{\alpha/2} \frac{s}{\sqrt{n}}$$

Her er  $\bar{v}$  beregnet gjennomsnittlig hastighet,  $t_{\alpha/2}$  er kritisk verdi i  $t$ -fordelingen med  $\alpha$  lik 0,05,  $s$  er beregnet standardavvik og  $n$  er antall registreringer. Tabell 4 viser gjennomsnittlig estimert hastighet gjennom de 9 områdene.

Tabell 4. Gjennomsnittlig estimert hastighet i de 9 områdene, med 95 % konfidensintervall

Område	Gjennomsnittlig hastighet
1	34,8 ± 33,9
2	45,5 ± 11,5
3	58,8 ± 4,8
4	62,2 ± 7,9
5	60,1 ± 11,2
6	47,9 ± 31,1
7	74,2 ± 9,7
8	86 ± 5
9	82,7 ± 7

#### 4.4 Utvikling av basismodell

Begrepet basismodell er i denne sammenhengen definert som en ukalibrert modell. Det vil si at vegnettverk og trafikkmatrise er på plass i modellen, men den er ikke kalibrert mot tellinger og hastighetsregistreringer. Dette kapitlet beskriver hva som er gjort for å etablere basismodellen.

##### *Vegnettverk:*

Datakilden for vegnettverket er Nasjonal Vegdatabank (NVDB). Informasjon om veggeometrien kan importeres automatisk inn i simuleringsmodellen. Da importeres alle attributter som finnes på hver lenke. Selv om importeringen går raskt, kreves det mye redigering i ettertid. Delvis fordi nettverket må feilsøkes, men også fordi nettverket må stemme overens med det nettverket som er grunnlaget for trafikkmatrisen som senere skal importeres i modellen.

##### *Kjøretøytyper:*

Det er benyttet to ulike kjøretøytyper i simuleringsmodellen, lette kjøretøy og tunge kjøretøy. Hver kjøretøytype beskrives gjennom et sett med parametere i Aimsun. Det finnes per i dag lite informasjon om hvilke verdier parameterne skal ha. Dette er parametere som bør estimeres på grunnlag av empiriske undersøkelser. Spesielt for tunge kjøretøy er det en mangel på slike undersøkelser. I mangel på bedre alternativer er det benyttet de samme parameterverdiene som ble benyttet i PRINT-prosjektet, som er delvis basert på standardverdier i Aimsun.

#### *Etterspørselsmatrise:*

Matrisen er hentet fra CONTRAM-modellen som er utviklet for Oslo. Den baserer seg på arealbruk og trafikk tall for 2003. Ideelt sett burde matrisene vært hentet fra en modell som baserer seg på et senere årstall, men det finnes ingen operative nyere modeller. Det er hovedsakelig to utfordringer knyttet til å bruke en matrise beregnet for 2003. Den første er trafikkmengden mellom hver sone, som kan antas å ha økt noe siden 2003. Den andre er reisemønsteret i det aktuelle modellområdet. I løpet av 8 år kan det tenkes at det har forandret seg på grunn av flere store vegprosjekter, både i og utenfor. Med tanke på det snevre modellområdet, så kan det antas at endringer i reisemønsteret ikke har så mye å si for modellen, så lenge trafikkvolumene inn og ut av modellen kalibreres.

#### **4.5 Feilsøking**

Etter at basismodellen ble etablert, ble det foretatt en feilsøking av vegnettverket og trafikkmatrisene. Dette ble gjort ved inspeksjon av attributter på veglenker og visuell sjekk av simuleringen. De viktigste hovedpunktene som ble kontrollert er:

- Geometri på vegnettverk sammenlignet med importert bakgrunnskart.
- Z-koordinat i kopling mellom lenker.
- Fartsgrenser.
- Lenkeparametere i rundkjøringer.
- Lenkeparametere på ramper.
- Svingebevegelser i vegkryss.
- Sonetilknytninger

#### **4.6 Kalibrering av modellen**

Det siste steget i etableringen av en simuleringsmodell, er å gjennomføre kalibrering. Det er gjort ved å tilpasse modellen til de innsamlede grunnlagsdata. Det er en tidkrevende prosess, fordi det er flere kriterier som skal oppfylles samtidig. En tilpassing til trafikkvolum kan føre til at reisetidene ikke stemmer overens, og motsatt. Det blir derfor en iterativ prosess som er ferdig når kalibreringsmålene er nådd. Det er videre beskrevet hvilke kalibreringsmål som er benyttet, samt hvordan kalibreringen er gjennomført.

#### *Kalibreringsmål:*

Håndboken "Guidelines for Applying Traffic Microsimulation Modelling Software" (USDOT, FHA, 2003) fra Federal Highway Administration (FHA) bruker størrelsen GEH<sup>5</sup> som et mål på avvik mellom simulert og registrert trafikkvolum. GEH defineres som

$$GEH = \sqrt{\frac{2(E - V)^2}{(E + V)}}$$

hvor E er simulert volum og V er registrert volum. Kravet til en kalibrert modell er at GEH skal være mindre enn 5 for 85 % av alle sammenlignbare trafikkvolum, samt mindre enn 4 for

---

<sup>5</sup> Etter trafikkforskeren Geoffrey E. Havers, som "fant opp" formelen (<http://en.wikipedia.org/wiki/GEH>)



det totale registrerte trafikkvolum. Samtidig skal over 85 % av de simulerte reisetider ha mindre enn 15 % avvik fra registrerte reisetider.

#### *Trafikkstrømmer inn og ut av modellen:*

Det første som ble gjort i kalibreringen var å kontrollere trafikkstrømmene i randen av modellområdet mot registrerte tellinger. Det er hovedsakelig gjort med telldata fra tellepunktene på Ulvensplitten, Teisen og Jerikoveien. Uansett hvor mye en endrer på modellen, så vil ikke endringene påvirke de trafikkstrømmene som kommer inn i modellen. Derfor ble nivået på trafikkvolum inn i modellen justert til nivået på registrerte volum.

#### *Rampeatferd:*

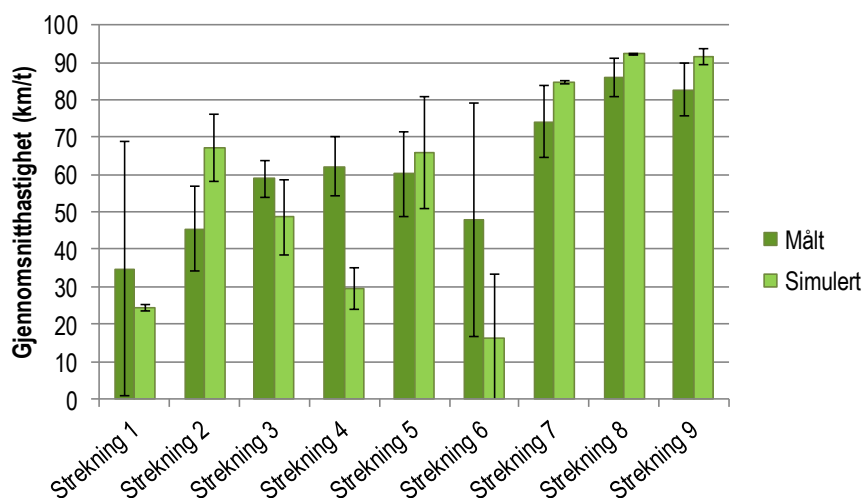
Modellparameterene som styrer atferden på av- og påkjøringsramper er "Lane Changing Cooperation", som beskriver hvor tilbøyelig et kjøretøy er til å gi plass til et kjøretøy som foretar et feltbytte. En lav verdi på denne parameteren vil føre til at få kjøretøy slipper inn fra en rampe, mens en høy verdi vil tilsvare en fletteatferd. Denne parameteren ble benyttet i kalibreringen for å få justere reisetid og trafikkvolum på flerfeltsveger med ramper.

#### *Atferd i rundkjøringer:*

I en så begrenset modell kan atferden i de få kryssene påvirke resultatet. Det er derfor viktig at atferden i rundkjøringene blir så riktig som mulig.

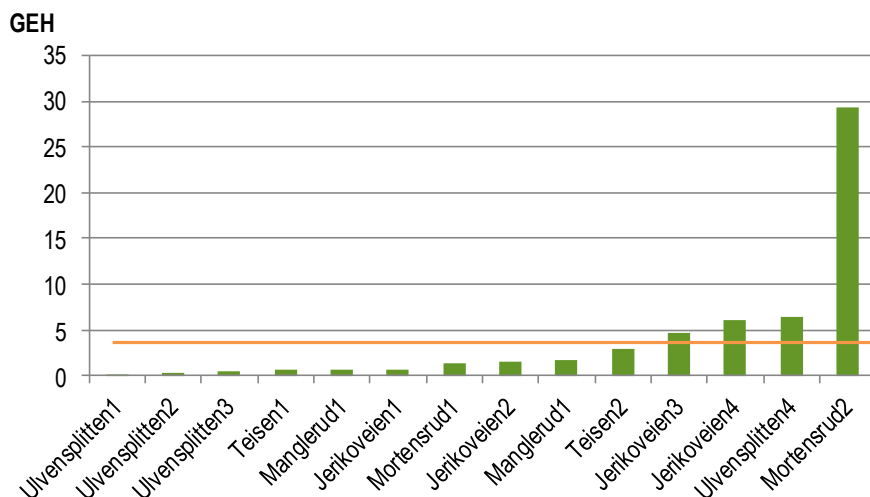
#### *Resultat av kalibrering:*

Kalibreringen er en iterativ prosess hvor modellen må kjøres etter hver endring i modellen. Det er ikke alltid like enkelt å forutsi hva virkningen av en endring blir, slik at en ender opp med å gå noen skritt frem og noen skritt bakover på veg mot målet. Med kalibreringsmålene som er presentert tidligere, er det lettere å stoppe kalibreringen på riktig tidspunkt og si at modellen er god nok. En må allikevel være klar over at det finnes mange ulike konfigurasjoner av modellen som vil føre til at kalibreringsmålene blir oppfylt. Derfor er det viktig å utføre visuelle inspeksjoner av animasjonen for å forsikre seg om at avviklingen ser tilfredsstillende ut.



Figur 31: Resultat av kalibrering av gjennomsnittshastigheter

Figur 31 og Figur 32 viser resultatet av kalibreringen i form av måloppnåelse av kalibreringsmål. De fleste gjennomsnittshastighetene fra simuleringen er innenfor ett standardavvik fra de målte reisetidene. De strekningene som skiller seg ut, er strekning 4 og 6 som ligger ved henholdsvis Ryen og Ulvensplitten. På grunn av den store spredningen i målinger på strekning 6 er simuleringene innenfor ett standardavvik. Strekning 4 har derimot litt for lav hastighet i forhold til målingene.



Figur 32: Resultat av kalibrering av trafikkvolum. Søylene viser GEH for hvert nivå-1 tellepunkt. Den røde streken indikerer kalibreringsmålet på GEH lik 5

Figur 32 viser GEH for hvert nivå-1 punkt. GEH representerer avviket mellom registrert og simulert trafikkvolum. Kravet fra FHA er at 85 % av målepunktene skal ha en GEH under 5. Figuren viser at 3 av de 14 målepunktene har GEH over 5. To av punktene er så vidt over 5, som kan forventes. Det tredje avviket er derimot altfor høyt, og viser at modellen ikke klarer å reprodusere de riktige trafikkstrømmene. Dette punktet ligger i nordgående retning ved Mortensrud, og det er simuleringene som har for høyt nivå på trafikkvolumet i forhold til registrert trafikkvolum.

Grunnen til avviket på Mortensrud, er at modellen overestimerer avviklingen på motorveg med ramper. Selv om gjennomsnittshastighetene er lave og dermed i tråd med registrerte hastigheter, er trafikkvolumet for høyt, spesielt på Mortensrud. I løpet av kalibreringsprosessen ble modellen justert flere ganger for å senke trafikkvolumet, men det førte kun til at gjennomsnittshastigheten ble for lav. Av praktiske grunner ble det besluttet å akseptere et høyt nivå på trafikkvolumet, siden hastighetene var akseptable.

Total GEH for alle nivå-1 punktene i modellen er beregnet til å være 5,2. Kravet fra FHA er at denne verdien skal være under 4. Hovedgrunnen til at total GEH er så høy er den høye verdien på GEH på Mortensrud. På grunn av at det ikke var mulig å endre på denne ene verdien, ble det besluttet at den totale verdien på GEH er akseptabel.

Alt sett under ett tilsier at den kalibrerte modellen er akseptabel med tanke på gjennomsnittshastigheter og trafikkvolum. Visuelle inspeksjoner av animasjonene ga heller ingen grunn til å tvile på den konklusjonen.

## 4.7 Scenarier og aktuelle resultatuttak

Demonstratoren i Oslo er tenkt å skulle kunne gi indikasjoner på størrelsesorden på totaleffekter av et kontrollsystem for alle involverte kjøretøy i modellområdet. Et slikt system vil kunne bestå av både regulerende og prioriterende tiltak for tungtransporten, og begge disse aspektene er forsøkt inkludert i Oslo-demonstratoren.

### 4.7.1 Prioriterende tiltak

I testen i tungbilsimulatoren ble det testet ut effekter av prioritering hhv. ved hjelp av grønn bølge og adgang til å benytte kollektivfelt. Da modellområdet som Oslo-demonstratoren stort sett inneholder motorveg og ingen lyskryss, er det irrelevant å gi prioritering til tunge kjøretøy ved grønn bølge i simuleringmodellen. Av prioriteringstiltakene er det derfor kun det å gi tungbiler adgang til egne felt, som er inkludert.

Å endre på feltbruken vil i noen tilfeller føre til at situasjonen for personbilene blir forverret, noe som sannsynligvis vil føre til en endring i reisemønsteret som denne modellen ikke klarer å fange opp. Det er derfor kun fokusert på bestemte strekninger i modellområdet, for å sammenligne trafikksituasjonen til personbiler og tungbiler ved ulike former for feltbruk. For å studere ulike varianter av endringer i feltbruk., er det etablert tre ulike scenarier som beskrives kort nedenfor. Modellen er etablert med data for perioden kl 07.00-09.00, og alle scenariene gjelder dermed for morgenrush-perioden. Alle endringer i feltbruk er gjort på strekningen langs E6 inn mot Alnabru fra sør, mellom Klemetsrud og Ulvensplitten.

#### *Omdefinering av eksisterende høyre felt til tungbilfelt:*

Det første scenariet med prioritering er å gjøre om det eksisterende høyrefeltet langs E6 om til et tungbilfelt. Det vil dermed være en nesten sammenhengende rute fra Klemetsrud i sør til Ulvensplitten i nord hvor tungbiler får sitt eget felt. I området rundt av- og påkjøringsramper vil det oppstå konfliktsituasjoner hvor personbiler må krysse tungbilfeltet for å komme til/fra venstrefeltet. Det vil derfor oppstå noe forsinkelse for tungbilene ved ramper.

#### *Omdefinering av eksisterende venstre felt til tungbilfelt:*

Det andre scenariet er å gjøre om det eksisterende venstrefeltet til tungbilfelt. Ruten fra Klemetsrud til Ulvensplitten er da fullstendig sammenhengende siden konfliktpunktene ved ramper er borte. Begge scenariene med omdefinering av eksisterende felt til tungbilfelt vil være dramatiske for personbilene, som får halvert kapasiteten langs omtrent hele ruten.

#### *Nytt høyre felt for tungbiler:*

Det tredje scenariet er å legge til et ekstra felt til høyre for eksisterende veg. Kapasiteten for personbilene blir dermed opprettholdt feltmessig, samtidig som tungbilene får et eksklusivt felt. Problemet er også her at det oppstår konfliktpunkter ved rampene, hvor personbiler må krysse tungbilfeltet for å komme ut på/av motorvegen.

### 4.7.2 Regulerende tiltak

I grunn-idéen for GOFER ligger bruk av egne oppstillingsområder for tungbiler i utkanten av byområdene som venteplass for kjøretøy som får beskjed om å vente med å fortsette sin ferd

inn mot sin destinasjon i byområdet. En kan se for seg at et mulig styringsregime med regulering av hvor mange/hvor stor andel av tunge kjøretøy som får benytte vegnettet i morgenrush-perioden. Det er derfor gjennomført simuleringer med varierende andel tungbiler i modellen. I tillegg til dagens situasjon, der samtlige tungbiler får slippe til i transportsystemet, er det gjort beregninger med fem alternative nivå på tungbiltrafikken, der hhv. 20 %, 40 % 60 % 80 % og 100 % av tungbilene holdes tilbake.

#### 4.7.3 Evaluering av scenariene

Det er hovedsakelig tre hovedtyper av effekter en kan beregne i modellen:

- Tidsbruk, som er reisetider for kjøretøyene i modellen.
- Trafikkflyt, som kan beskrives av hastigheter og akselerasjoner.
- Miljø, som beskrives ved drivstofforbruk og utslipp av klima- og miljøgasser.

Det er mulig å beregne effekter for både utvalgte kjøretøytyper og for alle sett under ett.

##### *Prioriterende tiltak:*

Det gjøres beregninger for dagens situasjon og hvert av de øvrige scenariene. Resultatene tas ut for strekningen mellom Klemetsrud og Ulvensplitten, hvor de ulike feltbruksmåtene er utprøvd. For hvert beregningsalternativ tas det ut beregnet gjennomsnittlig hastighet og forsinkelse, totalt og for hhv. tunge kjøretøy og personbiler.

##### *Regulerende tiltak:*

Det gjøres beregninger for dagens situasjon og hvert av de øvrige scenariene. Resultatene tas ut for hele modellområdet. For hvert beregningsalternativ tas det ut beregnet gjennomsnittlig hastighet og forsinkelse, totalt og for hhv. tunge kjøretøy og personbiler.

#### 4.8 Rutiner som er utviklet i forbindelse med etablering av simuleringsmodellen

I forbindelse med etablering av simuleringsmodellen er det utviklet en rekke script og metoder for håndtering av grunnlagsdata og bearbeiding og presentasjon av resultater, som vil kunne ha nytteverdi i andre sammenhenger:

- Tilrettelegging for import av vegnettverk fra Elveg.
- Script for import av bakgrunnskart til modellen.
- Script for tilrettelegging av telldata på ulikt format.
- Script for import av matriser fra CONTRAM.
- Script for uttak av data fra simuleringsmodell.
- Script for visning av hastighetsforskjeller mellom simulerte scenarier.
- Metodikk for å beregne reisetider fra GPS-data med lav oppløsning.
- Bruk av retningslinjer for å bygge opp og kalibrere simuleringsmodeller.
- Script for å benytte forbruksberegninger fra Aimsun på GPS-data.
- Metodikk for grafisk framstilling av data på ulikt format.
- Metodikk for å benytte fartsmodelldata som input til SEMBA.

## Referanseliste

Federal Highway Administration (2003): *Traffic Analysis Toolbox: Guidelines for applying traffic microsimulation modeling software*. U.S. Department of Transportation, Federal Highway Administration, Research, Development, and Technology

GOFER hjemmeside: [www.sintef.no/GOFER](http://www.sintef.no/GOFER)

GOFER (2010a): *L1.0 Behovsanalyse og samarbeidsmodell*, Versjon 1.1, 2. juli 2010

GOFER (2010b): *L0.1 Prosjektplan*, Versjon 2.0, 20. desember 2010

GOFER (2013a): *L2.0 Systemarkitektur og datamodell*, Versjon 1.0, august 2013

GOFER (2013b): *L3.0 Evaluering*, Versjon 1.0, august 2013

GOFER (2013c): *L5.0 Videre anvendelse*, Versjon 1.0, august 2013

Natvig, M.K., H. Westerheim, T.K. Moseng, A. Vennesland, (2009): *ARKTRANS The multimodal ITS framework architecture* Version 6. SINTEF Report A12001, SINTEF, Norway. <http://arktrans.no/english>

Tørset, T., A. Aakre, V. Børnes, O.M. Rennemo (2011): *Fartsmodell for næringslivets transporter*. Datagrunnlag og dokumentasjon av modell. SINTEF Rapport A17524, SINTEF, Trondheim.



## Vedlegg 1 Database for Live-demonstratoren

GOFER er et sentralisert system bygget rundt en felles database. Muligheter og begrensninger er i stor grad bestemt av strukturen i denne, og er derfor verdt nærmere omtale. Se også AP2 for en introduksjon til datamodellen.

Her følger en detaljering av hvordan datamodellen ble realisert som en database i demonstratoren, tabell for tabell.

En fremstilling av det komplette skjemaet ligger i et eget dokument.

Disse bør leses i sammenheng.

Merk at bruk av fremmednøkler ikke er konsekvent, og bør revideres og kompletteres i et endelig system.

### V1.1 Actor

Aktør-begrepet ble realisert som en enkelt tabell, med alle nødvendige data for alle aktører inkludert.

### V1.2 Role

Inneholder alle rolledefinisjoner i systemet, og er sentral når grupper av aktørgrupper skal behandles. Det skjer typisk under fylling av nedtrekksmenyer i dialoger for initiering av transportoppdrag.

### V1.3 ActorRole

Kobler sammen Actor og Role slik at hver rad i Role kan kobles til flere rader i Actor, og slik at hver rad i Actor kan kobles til flere rader i Role. D.v.s. en mange-til-mange-relasjon.

Eksempel:

Alle sjåfører er knyttet til rollen «Driver». Samtidig kan selvstendig sjåfør kan også være knyttet til rollen «Transporter»

Aktøren «COOP» er knyttet til rollene «Customer» og «Terminal operator».

### V1.4 Message

Meldinger mellom aktører i systemet legges her. «Fromactor» angir avsender mens «toactor» angir mottager. Kolonnen er knyttet til tabellen Actor. Alternativt kan en mottagergruppe angis ved å bruke kolonnen «torole», som peker på en rad i Role.

Hver melding er knyttet til MessageType og MessageStatus.

### V1.5 RoadMessage

Mottakstabell for meldinger fra TRIP. Nye meldinger konverteres til rad i Message.

### V1.6 MessageType

Angir hva slags meldingstype det er snakk om: Sjåførmelding, stengningsmelding m.m.

### V1.7 MessageStatus

Definerer meldingens livssyklus: Aktiv, utgått m.m. Nødvendig innhold i de siste to tabellene defineres av detaljene i meldingshåndteringen, og mange utvidelser er tenkelige.





### **V1.8 Position**

Angir koordinater som log, lat og altitude, men også med peker til RoadNode og RoadLink. Tabellen brukes av Actor og Message, og gjør det mulig å knytte terminaler og meldinger direkte til transportnettverket. Noe som i sin tur gjør det mulig å søke effektivt etter meldinger, rasteplasser, terminaler m.m. langs definerte ruter.

### **V1.9 Restriction**

Angir en begrensning på transporter som kan passere et punkt eller lene, og er knyttet til Position. Brukes til å forkaste uegnede lenker under definering eller søk etter transportruter.

### **V1.10 Instruction, InstructionType**

Definerer forbud eller påbud knyttet til angitt rute, med gyldighetsbegrensninger som går på dato, klokke, dagtype, kjøretøytype, lasttype, euroklasse, vekt, lengde.

Tilhørende rutedefinisjon for et forbud kan f.eks. være alle transportlenkene i et bysenter, mens påbudsruiter kan være sammenhengende traseer som spesielle transporter må bruke.

### **V1.11 RoadNode, RoadLink**

Hver rad i RoadLink er knyttet til to rader i RoadNode, franode og tilnode. I tillegg inneholder RoadLink informasjon om vegident, lengde, bredde, ÅDT, stigning, radier, fartsgrense m.m., slik at fartsmodellen kan implementeres. (Se Vedlegg 3). Navn på kolonnene er stort sett lånt fra TNext-prosjektet og fartsmodellprosjektet for å forenkle etableringen av rutedefinisjoner. Se beskrivelse av fartsmodellen i Vedlegg 3.

RoadNode inneholder stort sett posisjon, nummer og eventuelt navn.

Både RoadNode og RoadLink inneholder geometrikolonner, slik at de kan presenteres i et kart.

### **V1.12 Route, RouteRoadLink**

En rute i Route er en samling lenker som strekker seg fra en RoadNode til en annen, men det kan også være en uordnet samling av rader fra RoadLink, og da brukes ruten bare til å angi områder knyttet til kjørerestriksjoner.

Hver RoadLink kan inngå i flere ruter. Dette mange-til-mange-forholdet mellom Route og RoadLink realiseres av tabellen RouteRoadLink. I denne ligger også en geometrikolonne, som egentlig er unødvendig, men som forenkler presentasjonen av enkeltruter i kartet.

### **V1.13 TravelTime, TravelTimeLink**

Hver rad i TravelTime representerer en kjøretidsberegning basert på Fartsmodellen. Den er knyttet til kjøretøy, tilhenger, og rute, og med oppgitt nyttelast.

Kjøretiden er aggregert fra tilhørende rader i TravelTimeLink. Hver av dem er i tillegg knyttet til en rad i RouteRoadLink. For enklere bruk er også den underliggende id fra RoadLink tatt med.

Hver rad i TravelTimeLink er også påført inngående, utgående og midlere hastighet, reisetid etter fartsmodellen, og reisetid etter fartsgrensen.

Reisetidene oppgitt i TravelTime er altså en aggregering av disse.

#### **V1.14 Vehicle, Trailer**

Vehicle inneholder kjøretøydefinisjonene, med tilstrekkelige data for fartsmodellen. Hvert kjøretøy kan påkobles 0 eller 1 tilhenger, Trailer.

#### **V1.15 VehicleUnit**

Kjøretøyenhetene er beskrevet i VehicleUnit, og er i demonstratoren fast oppkoblet mot kjøretøy og sjåfør. Denne koblingen er i utgangspunktet fleksibel, og vi ser for oss at denne kan etableres under definering av et transportoppdrag.

#### **V1.16 Assignment, Cargo, CargoHazardGroup**

Hver rad i denne representerer ett transportoppdrag, med kobling til lastdefinisjon (Cargo), fareklasse (CargoHazardGroup), transportør, avsender, mottager, kunde, starttidspunkt og leveringstidspunkt.

#### **V1.17 Transport, AssignmentTransport**

Hver rad i Transport beskriver transporten mellom 2 terminaler i transportnett, som utføres av ett kjøretøy, og eventuelt en tilhenger.

Hvert oppdrag (Assignment) kan bestå av et vilkårlig antall transporter. Samtidig kan en enkelt transport bidra i flere transportoppdrag. Disse koblingsmulighetene realiseres av AssignmentTransport.

Containere kan defineres, og knyttes til hver enkelt transport.

#### **V1.18 Positionlog**

Denne inneholder posisjonsloggene fra OBE'ene med øyeblikksverdier for hastighet og akselerasjon, og er samtidig knyttet til kjøretøy, transport og veglenke. Sistnevnte er nyttig ved lenkevis etterkontroll av fartsmodellen.

## Vedlegg 2 Etablering av transportnett og ruter i GOFER Live-demonstrator

Her følger en kortfattet gjengivelse av fremgangsmåte for etablering av transportnett og ruter i GOFER demonstrator. Nettverket blir delt inn på samme måte som nettverket i Fartsmodellprosjektet, slik at man slipper å introdusere flere usikkerheter angående beregningenes gyldighet.

Nettverk og ruter lastes til slutt inn i GOFER databasen (Beskrevet i Vedlegg 1), som er basert på bruk av "[Postgres Plus Standard Server](#)" fra Enterprise DB, og bruk av "NVDB analyse" og "TNext" i ArcMap.

*Oversikt:*

- Uttak av grunnlagsdata fra NVDB
- Konvertere data fra ArcGis10 til ArcGis9 geodatabaser
- Generere fartsmodellenker
- Definere ruter
- Eksport til shapeformat
- Import av shapefiler til PostGis

### V2.1 Uttak av grunnlagsdata fra NVDB

Installere NVDB analyse for ArcMap10 fra denne lenken:

[ftp://NVDB:NVdb\\_08@80.232.45.122/Prod10\\_01/NVDB%20Client%20API%20-%20Additions/Build\\_5583\\_168\\_NVDB%20Client%20API%20-%20ArcMap.msi](ftp://NVDB:NVdb_08@80.232.45.122/Prod10_01/NVDB%20Client%20API%20-%20Additions/Build_5583_168_NVDB%20Client%20API%20-%20ArcMap.msi)

Hentet data fra NVDB via "start NVDB klient".

Definerte datauttak med følgende egenskaper:

*Hvor:*

(Akershus + Hedmark + Oppland + Oslo + Sør-Trøndelag +Telemark) union  
( EV6 + RV3 + RV706)

Tillegg v5:

FV902, FV885, FV715, FV868, RV704.

*Hva:*

Vegreferanse og kommune er standard innhold.

Legg til:

- Fartsgrense (105)
- Feltstrekning (616)
- Kurvatur horisontalelement (639)
- Kurvatur vertikalpunkt (642)
- Vegbredde (583)

I tillegg ble følgende inkludert som interessant tilleggsinformasjon:

- ATK-punkt (162)
- ATK-strekning (775)
- Trafikkmengde (583)

NVDB analyse vil etter nedlasting pakke ut prosjektfilene til en geodatabase, og vise innholdet i kartet.

## V2.2 Konvertere data fra ArcGis10 til ArcGis9 geodatabase

*Dette utgår etter full overgang til ArcGis10:*

Kode for å produsere veglenker etter Fartsmodellens krav, kjører for øyeblikket ikke på ArcGis10, mens NVDB analyse ikke kjører på ArcGis9.

Ny tom geodatabase ble initiert med ArcCatalog9, og kopiert inn på maskin med ArcGis10.

Her ble alle vanlige tabeller kopiert over fra 10-versjon til 9-versjon.

Vegnettet (vanligvis kalt "VEGNETT 50000" i databasen) kunne ikke kopieres/eksporteres over fra v10 til v9, p.g.a. endringer i måten høydedata representeres på. Derfor gikk denne overføringen via shapefil.

## V2.3 Generere fartsmodellenker

*Aktiver fartsmodell-funksjoner:*

TNExt-modulen har en egen "fartsmodell-bryter". Den aktiveres på følgende måte:

- Initier nytt transportmodellprosjekt
- Legg til vegnettet som lag fra geodatabasen.
- Definer lag, og pek ut den nye geodatabasen som den som skal brukes.
- Definer transportmodell-lag som vanlig.
- Avslutt ArcMap
- Start Ms Access, og let opp tabellen "SintefTmConfig"
- Legg til ny rad i tabellen: Id = største id + 1, dataset = 1, paramname = 'Fartsmodell', paramvalue = 'True'
- Restart ArcMap-prosjektet

*Lag Fartsmodell lenkedefinisjoner:*

- Velg lag på nytt. Lag for veglenker er nå ikke lengre gyldig. Legg til nytt lenkelag på vanlig måte. Det får navnet "FmVeglenke" som standard.
- Velg "Importer vegnett" i menyen. Dette starter en prosess som fyller lenke- og node-tabellene med nye data, basert på valgt vegnett og tilgjengelige fagdata, slik at fartsmodellen kan bruke dem.

## V2.4 Definere ruter

En GOFER-rute er en liste over veglenker, og er slik sett det samme som en kollektivrute. Åpne dialogen for kollektivrutedefinisjoner, og definer rutene:

- Oslo-Trondheim via Østerdalen og NAV
- Trondheim – Oslo via NAV og Østerdalen
- Oslo-Trondheim via Gudbrandsdalen og NAV
- Trondheim – Oslo via NAV og Gudbrandsdalen

I tillegg kommer delstrekningene fra Sluppen til Piren via Trondheim sentrum, begge veier. Innholdet i kollektivlenketemaet dekker innholdet i GOFER-tabellen "routeroadlinks".

## V2.5 Eksport til shapeformat

En del av attributtene på vegnode, veglenkeFm og kollektivlenke er uinteressante, og kan med fordel fjernes ved eksport. Det gjelder særlig for de attributtene som er spesifikke for transportmodeller. Det gjør datasettene mere håndterlige og oversiktlige.

For FmVeglenke **utelates** disse:

Linje, editstatus, editstes, geomstatus, synlig, tid, t, ablinktype, abjurcode, ablinkcap, abcapind, balinktype, bajurcode, balinkcap, bacapind, tellepunkt, description, behold, effektid, abspeed, baspeed, abspeed\_light, abspeed\_heavy, baspeed\_light, baspeed\_heavy, baspeedlimit

For Node **utelates** følgende:

Linje, nr, sentroide, kjerne, editstatus, editstes, synlig

## V2.6 Import av shapefiler til PostGis

*Forbered:*

Kopiere de aktuelle shape-filene til PostGreSql bin-katalog.

I PgAdminIII, åpne databasen som er aktuell for import. (slipper å logge inn?)

*Lag sql-fil:*

Åpne kommandolinjevinduet for maskinen, gå til PostGres bin-katalog.

Lag sql-fil med shp2pgsql.exe, som følger med i PostGIS-installasjonen, ved å skrive følgende i konsollvinduet:

```
shp2pgsql -I -S -i -d -g geom <shapefilnavn_uten_filtype> <tabell_for_mottak_av_data> > <kommandofilnavn>
```

Eksempel:

```
shp2pgsql -I -S -i -d -g geom kollektivlenke routeroadlink > routeroadlink.sql
```

En grei forklaring på bruk av shp2pgsql finnes her: <http://man.cx/shp2pgsql>

Ta med kolonne OBJECTID i lenketabellen, da det er denne som er knyttet mot kollektivlenke veglenke.

*Kjør sql-fil:*

I det samme konsollvinduet, skriv deretter:

```
Psql -h <vertsnavn_eller_ip> -p <portnummer> -d <databasenavn> -f <kommandofilnavn>
```

Greit forklaring på kommandoen ligger her: <http://man.cx/psql>

Eksempel:

```
psql -h 129.241.253.21 -p 8088 -d GOFER -f routeroadlink.sql
```

*Korreksjon – feil i datagrunnlag:*

Kjøring av sql-fil for veglenke feilet.

Problemet viste seg å være den tekstlige angivelsen av talltypen double, hvis den har verdien -1.

Denne ble lagret som '-1.#IND00000e+000', som ikke kan tolkes av databasen / psql.

Bruk søk og erstatt i Notepad eller lignende, og erstatt '#IND' med en tom streng.

*Korreksjon – lenkerrekkefølge:*

Når TNExt produserer kollektivruter, legges lenkerrekkefølgen på kollektivnodene.

I GOFER skal rekkefølgen ligge i kolonne "nr" i routeroadlink.

Dette løses ved import av kollektivlenke shapefil ved at dennes "objectid" -kolonne blir endret til "nr" etter importen. Det kan gjøres fordi kollektivlenkene lagres i den rekkefølgen de er ment å kjøres av TNExt. (Alternativet er å lenke inn kollektivnode-temaet ved eksport, og bruke kolonnen "nr" fra dette)

*Korreksjon – roadlink – roadnode:*

Tabell roadnode

Primærnøkkel kolonne "gid" -> "id".

"objectid" utgår

"linje" utgår

Legg til kolonner "lat", "lon", "altitude", alle [double]

Tabell "roadlink"

Primærnøkkelkolonne "gid" -> "id".

GOFER antar at node-referanser henviser til node.id, og ikke HNr, som er utgangspunktet.

Det ordnes slik:

Legg til kolonne "fromnode", og fyll med data:

"update roadlink set fromnode = (select id from roadnode where hnr = roadlink.anode);"

Legg til kolonne "tonode", og fyll med data:

"update roadlink set tonode = (select id from roadnode where hnr = roadlink.bnode);"

*Korreksjon – routeroadlink:*

Primærnøkkelkolonne "gid" -> "id".

Innfører og oppdaterer "roadlinkid":

"update roadroutelinks set roadlinkid = (select id from roadlink where objectid=roadroutelinks.veglenke"

Innfør og oppdater "roadlinkdir" ved å endre navn på "veglenkere".

Innfør og oppdater "routeid" ved å endre navn på "rute".

Koble routeroadlinks til roadlink:

Bruke av PgAdmin / POSTGIS til å etablere forbindelsen til roadlink:

1) update routeroadlinks set roadlinkid = null;

2) update routeroadlinks set roadlinkid = (select id from roadlink where ST\_Covers(roadlink.geom, routeroadlinks.geom));

Feiler en del. Denne tar nesten alt resterende:

update routeroadlinks set roadlinkid = (select id from roadlink where

ST\_HausdorffDistance(roadlink.geom,routeroadlinks.geom) < 2) where roadlinkid is null;

Konvertere fra multiline til linestring:

Etter import av shape til pgsql, kan det være ønskelig/nødvendig å konvertere type geometri fra multiline til linestring. Se her: [http://www.postgis.org/docs/ST\\_LineMerge.html](http://www.postgis.org/docs/ST_LineMerge.html)

### Vedlegg 3 Implementering av Fartsmodellen i Live-demonstratoren

Etablering av transportnett og rutedefinisjoner er beskrevet i Vedlegg 2. Alle beregninger er basert på resultatene fra Fartsmodellprosjektet (Tørset m.fl., 2011). Se spesielt kapittel 3, «Etablering av fartsmodellen».

For å få fullt innsyn i alle detaljer i beregningen, må kildekoden leses. Se filen «AdminApp\_og\_Fartsmodell\_Kildekode.zip». Interessante filer her ligger i underkatalogen «java\no\sintef\pro\gofer\server\». Startpunktet for Fartsmodellen finnes i «GoferObjectManager.java», nærmere bestemt metoden «findRouteTravelTime(int routeId, int vehicleId, int trailerId, int payload)»

Det gir liten oversikt å forsøke å lese denne uforberedt, så her følger en oversikt, og deretter en mere detaljert pseudokode for det samme.

Vegnettet ble delt inn i lenker etter samme kriterier som i Fartsmodellprosjektet:

- En rute defineres som lenkeliste fra start til slutt.
- Kjøretøy beskrives med totalvekt, motoreffekt med utnyttelsesgrad, frontareal m. friksjonskoeffisient og rullemotstand.
- Utnyttelsesgrad på motor er noe høyere ved minkende fart i motbakke, enn under vanlig akselerasjon.
- Reisetiden lagres pr lenke og transport.
- For hver lenke er inngangsfarten lik forrige lenkes utgangsfart, eller 0 ved oppstart.

Forløpet beskrives slik:

```
Forrige utgangsfart = 0

For alle lenker langs ruten:

  Hvis helning < -4%:
    Egen formel uavhengig av kjøretøy, last og foregående lenker.
  Ellers:
    Inngangsfart = forrige utgangsfart

  Ønskefart = friflytfart for gjeldende fartsgrense, eventuelt korrigert
  ned p.g.a. veggbredde eller horisontalradius.

  Hvis ønskefart < inngangsfart, brems ned til ønskefart:
    Antar jevn oppbremsing i hele lenkes lengde, uavhengig av nødvendig
    effekt.

  Hvis ønskefart > inngangsfart, akselerer opp til ønskefart hvis mulig:
    Antar jevn akselerasjon med gjeldende effektoverskudd, fra inngangsfart
    og opp til ønskefart, eller opp til den farten som gjør at motoreffekt
    og effektbehov balanserer. I effektbehovet inngår faktorer som
    rullemostand, luftmotstand, helning, vekt.

Lagre utgangsfart m.m.
```

*Pseudokode:*

Følgende kode viser beregningsforløpet med litt større detaljering:

```

Kjøretøy K = finn kjøretøy for oppdraget
fartInn = 0 // Inngående fart på lenken
For alle lenker L langs valgt rute:
    fartBasis = finnFartFartsgrense(L.fartsgrense)
    fartVegbredde = finnFartVegbredde(L.vegbredde)
    fartKurve = finnFartKurve(L.kurveradius)
    fartBasis = minimum(fartBasis, fartVegbredde, fartKurve) // Ønskefart
    Hvis (L.stigning < -4%):
        fartStigning = finnFartUtforbakke(L.stigning)
        fartUt = minimum(fartBasis, fartStigning)
    Ellers:
        fartUt = finnFartKjøretøy
                (K, L.stigning, L.lengde, fartBasis, fartInn)
    L.reisetid = 2 * L.lengde / (fartInn + fartUt)
    fartInn = fartUt
Fortsett
  
```

De enkelte funksjonene er definert slik:

```

Funksjon finnFartFartsgrense(fartsgrense)
    Hvis fartsgrense < 50
        Fart = fartsgrense
    Eller hvis fartsgrense > 80
        Fart = 84
    Eller
        Velg(fartsgrense)
            50: fart = 56
            60: fart = 67
            70: fart = 75
            80: fart = 80
            Ellers:
                Meld feil "Ugyldig fartsgrense"
        Slutt velg
    Slutt hvis
    Returner fart
  
```

```

Funksjon finnFartVegbredde(vegbredde)
    fart = (vegbredde * 10) + 10
    Returner fart
  
```

```

Funksjon finnFartKurve(kurveradius, fartsgrense)
    fart = finnBasisFart(fartsgrense)
    R = kurveradius
    Velg(fartsgrense)
        50: fart = 56 - 57000 * R-2,52
        60: fart = 67,6 - 113000 * R-1,978
        70: fart = 76,1 - 26000 * R-1,568
        80 eller større: fart = 83,2 - 14600 * R-1,387
    Ellers:
        fart = finnBasisFart(fartsgrense)
    Slutt velg
    fart = maksimum(fart, 0) // Svært små radier i forbindelse med kryss
    kan ellers gi negativ fart.
    Returner fart
  
```



```

Funksjon finnFartKjoretøy(kjoretøy, stigning, lengde, fartInn, fartMaks)
  // Finner utgående hastighet, avgrenset av fartMaks.
  // Finner også lengden inn på lenke hvor denne fart ble oppnådd. (lengdeFart)
maksFeil = 0,1
maksEffekt = 0,95 * kjoretøy.motoreffekt
Hvis stigning < 0
  maksEffekt = 0,85 * kjoretøy.motoreffekt
fInn = finnMotstand(fartInn, kjoretøy, stigning)
effektInn = fInn * fartInn
fInnReserve = (maksEffekt / fartInn) - fInn
fart = fartInn
ferdig = false
Fortsett
  fMotstand = finnMotstand(fart, kjoretøy, stigning)
  effekt = fMotstand * fart
  effektReserve = maksEffekt - effekt
  Hvis fartInn ikke kan opprettholdes:
    Fart = maksEffekt / fMotstand
  Ellers hvis effekt > maksEffekt // D.v.s. fartMaks er ikke er
opnåelig
    Fart = middelverdi(fart, fartMaksEffekt) // fordi
ulineært forløp.
    Ellers// D.v.s. at fart kan holdes eller økes:
    Hvis fart < fartMaks // d.v.s det er ønskelig å øke
farten.
    // F=m*a, a=(v-v0)/t, t=2*L/(v+v0)
    // => v^2 = v0^2 + 2*F*L/m
    fart = Sqrt(fartInn^2
+ (2*fInnReserve*lengde/k.vekt))
    Hvis fart > fartMaks:
      Fart = fartMaks
      lengdeFart=sqrt(fart^2 -
fartInn^2)*k.vekt/(2*fInnReserve)
    Ellers:
      lengdeFart = lengde
    Hvis fart>=fartMaks
      Ferdig = SANN
    Ellers hvis fart-forrigeFart<0,1 m/s
      Ferdig = SANN
    Ellers
      forrigeFart = fart
Inntil(ferdig)
Return fart + lengdeFart

```

```

Funksjon finnMotstand(fart,kjoretøy,stigning)
fRull = 0,015
fRulleMotstand = fRull * m * g * cos( stigning )
fStigningMotstand = m * g * sin( stigning )
roLuft = 1,2 kg/m3
fartLuft = 0,0
fLuftMotstand = 0,5 * roLuft * kjoretøy.cw * kjoretøy.frontAreal * (
fart + fartLuft )^2
fMotstand = fRulleMotstand + fStigningMotstand + fLuftMotstand
retur fMotstand

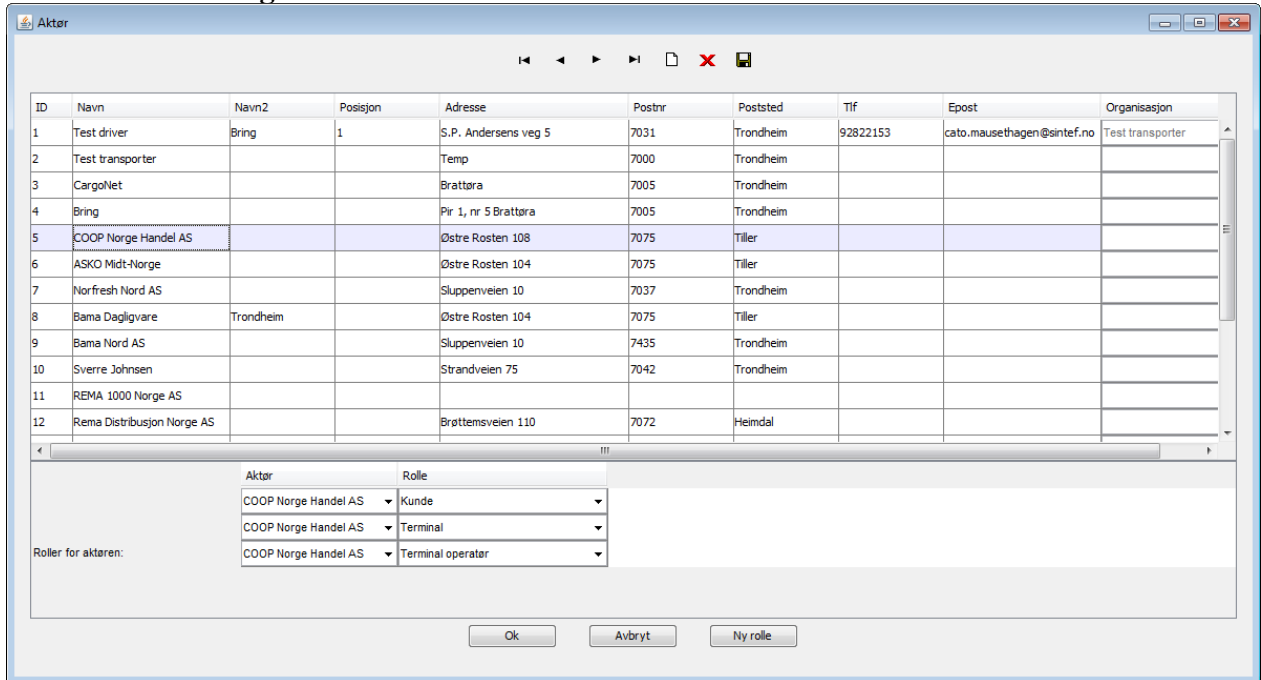
```

Fartsmodell-koden ble lagt inn i serverdelen av den administrative programvaren, beskrevet senere. Der kjørte den i en egen tråd, og startet ruteberegning på alle definerte transporter hvor dette ikke var gjort, eller hvor det hadde vært endringer siden siste kjøring.

## Vedlegg 4 Applikasjon for administrativ bruk i Live-demonstratoren

Komponenter fra DAKAT-prosjektet ble brukt til å sette opp en liten applikasjon for innlegging av kjøretøy, aktører, rutenavn, tilhengere m.m., og til å teste fartsmodellen på gitte transportere. Koden for å kjøre Fartsmodellen på transportene var også inkludert. Her følger noen eksempler på brukergrensesnittet:

### Definere aktører og roller:



ID	Navn	Navn2	Posisjon	Adresse	Postnr	Poststed	Tlf	Epost	Organisasjon
1	Test driver	Bring	1	S.P. Andersens veg 5	7031	Trondheim	92822153	cato.mausestegen@sintef.no	Test transporter
2	Test transporter			Temp	7000	Trondheim			
3	CargoNet			Brattøra	7005	Trondheim			
4	Bring			Pir 1, nr 5 Brattøra	7005	Trondheim			
5	COOP Norge Handel AS			Østre Rosten 108	7075	Tiller			
6	ASKO Midt-Norge			Østre Rosten 104	7075	Tiller			
7	Norfresh Nord AS			Sluppenveien 10	7037	Trondheim			
8	Bama Dagligvare	Trondheim		Østre Rosten 104	7075	Tiller			
9	Bama Nord AS			Sluppenveien 10	7435	Trondheim			
10	Sverre Johnsen			Strandveien 75	7042	Trondheim			
11	REMA 1000 Norge AS								
12	Rema Distribusjon Norge AS			Brøttensveien 110	7072	Heimdal			

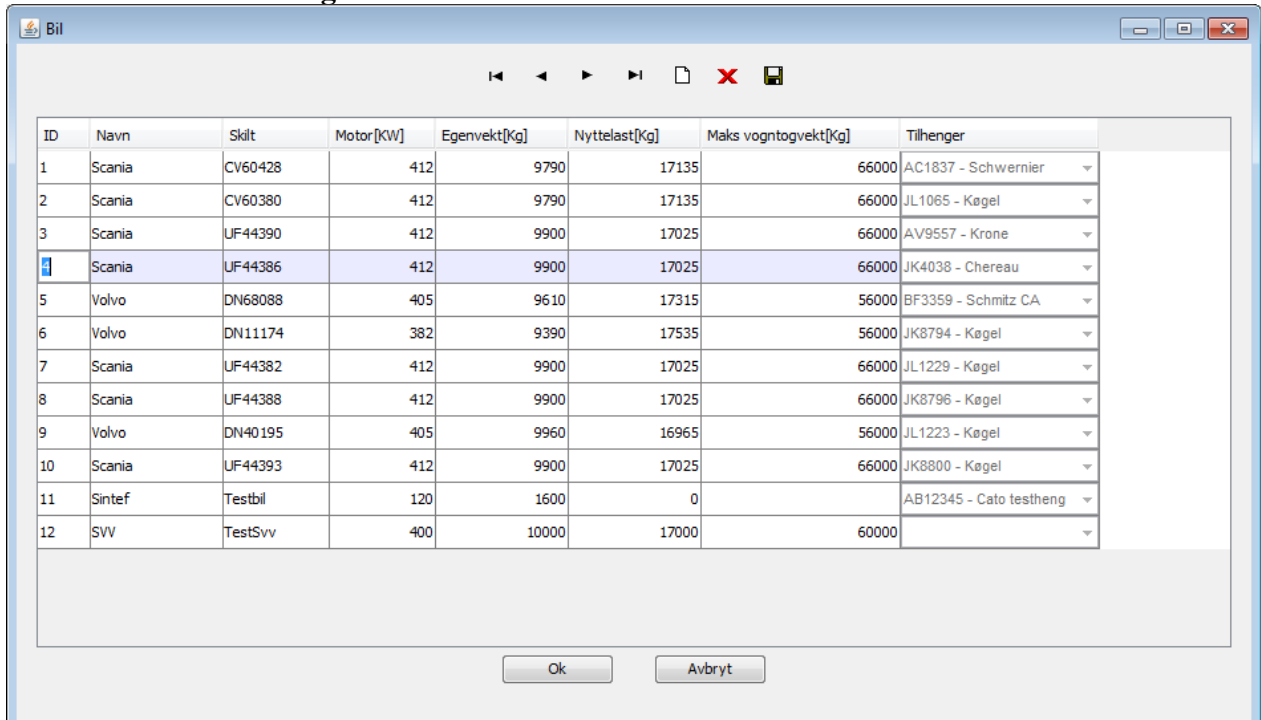
  

Aktor	Rolle
COOP Norge Handel AS	Kunde
COOP Norge Handel AS	Terminal
COOP Norge Handel AS	Terminal operatør

Roller for aktøren:

Ok Avbryt Ny rolle

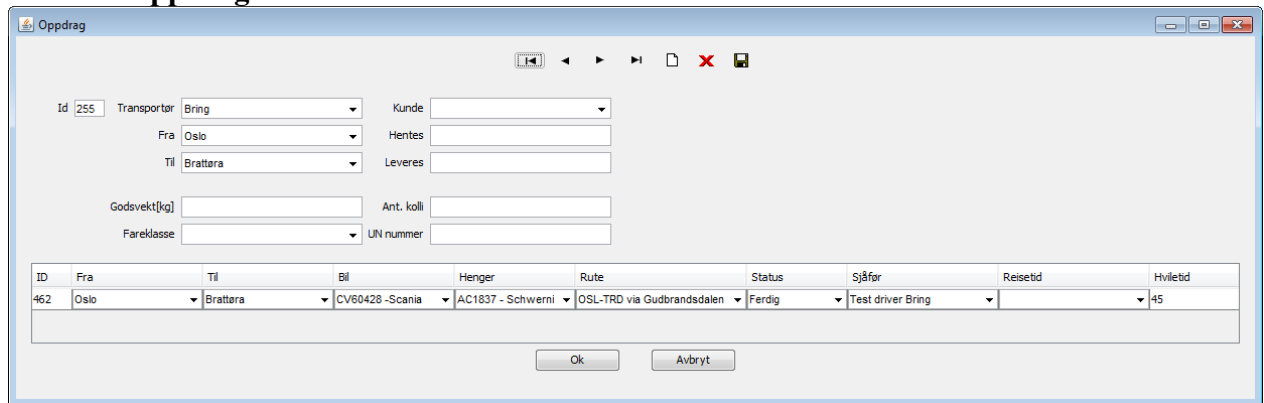
### Definere bil med tilhenger:



ID	Navn	Skilt	Motor[KW]	Egenvekt[Kg]	Nyttelast[Kg]	Maks vogntogvekt[Kg]	Tilhenger
1	Scania	CV60428	412	9790	17135	66000	AC1837 - Schwernier
2	Scania	CV60380	412	9790	17135	66000	JL1065 - Køgel
3	Scania	UF44390	412	9900	17025	66000	AV9557 - Krone
4	Scania	UF44386	412	9900	17025	66000	JK4038 - Chereau
5	Volvo	DN68088	405	9610	17315	56000	BF3359 - Schmitz CA
6	Volvo	DN11174	382	9390	17535	56000	JK8794 - Køgel
7	Scania	UF44382	412	9900	17025	66000	JL1229 - Køgel
8	Scania	UF44388	412	9900	17025	66000	JK8796 - Køgel
9	Volvo	DN40195	405	9960	16965	56000	JL1223 - Køgel
10	Scania	UF44393	412	9900	17025	66000	JK8800 - Køgel
11	Sintef	Testbil	120	1600	0		AB12345 - Cato testheng
12	SVV	TestSvv	400	10000	17000	60000	

Ok Avbryt

### Definere oppdrag manuelt:



Oppdrag

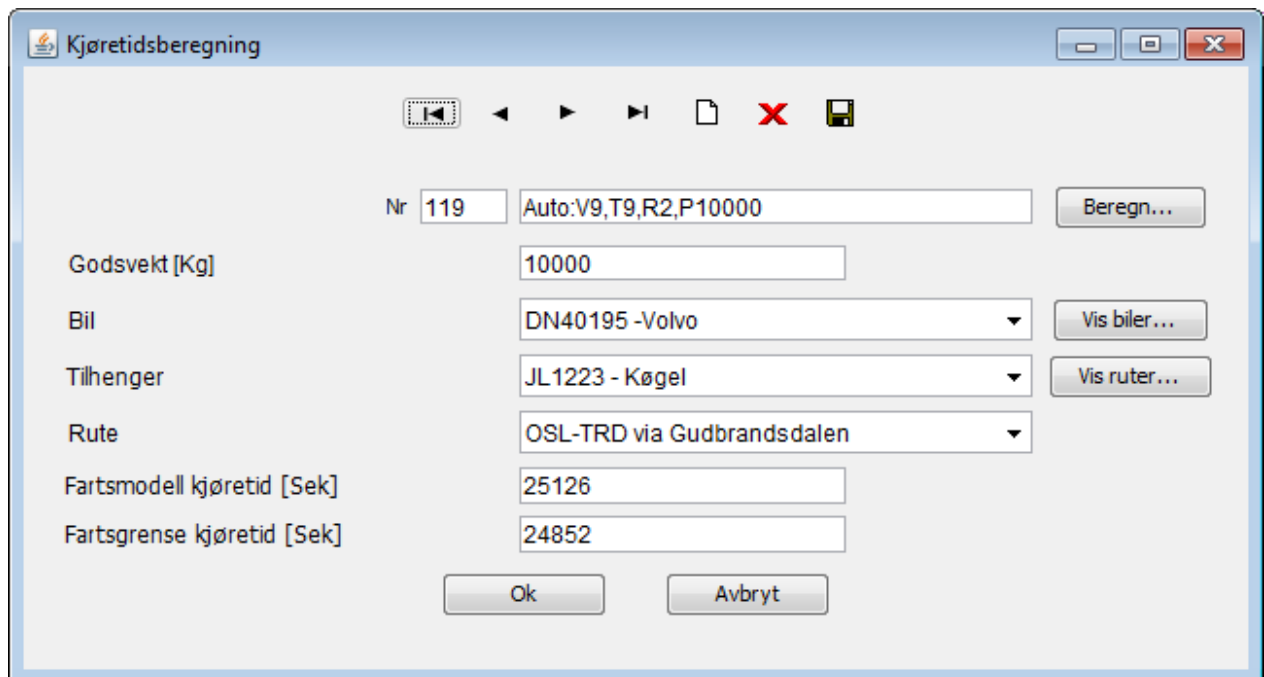
Id: 255 Transportør: Bring Kunde:   
 Fra: Oslo Hentes:   
 Til: Brattøra Leveres:   
 Godsvekt[kg]: Ant. koll:   
 Fareklasse: UN nummer:

ID	Fra	Til	Bil	Henger	Rute	Status	Sjåfør	Reisetid	Hviletid
462	Oslo	Brattøra	CV60428 -Scania	AC1837 - Schwerni	OSL-TRD via Gudbrandsdalen	Ferdig	Test driver Bring		45

Ok Avbryt

### Rapportere og eventuelt teste ny fartsmodell på transportene:

(Prefiks "Auto" angir at data er automatisk generert av serverprosess.)



Kjøretidsberegning

Nr: 119 Auto:V9,T9,R2,P10000 Beregn...

Godsvekt [Kg]: 10000

Bil: DN40195 -Volvo Vis biler...

Tilhenger: JL1223 - Køgel Vis ruter...

Rute: OSL-TRD via Gudbrandsdalen

Fartsmodell kjøretid [Sek]: 25126

Fartsgrense kjøretid [Sek]: 24852

Ok Avbryt



**Kontaktinfo:**

ITS Norge: Trond Hovland, tlf: 907 60 831, [trond.hovland@its-norway.no](mailto:trond.hovland@its-norway.no)

SINTEF: Solveig Meland, tlf: 932 11 017, [solveig.meland@sintef.no](mailto:solveig.meland@sintef.no)

Prosjektets hjemmeside: [www.sintef.no/GOFER](http://www.sintef.no/GOFER)