


# Offentlig Samkjøringsplattform

## Forprosjekt Incentivordning

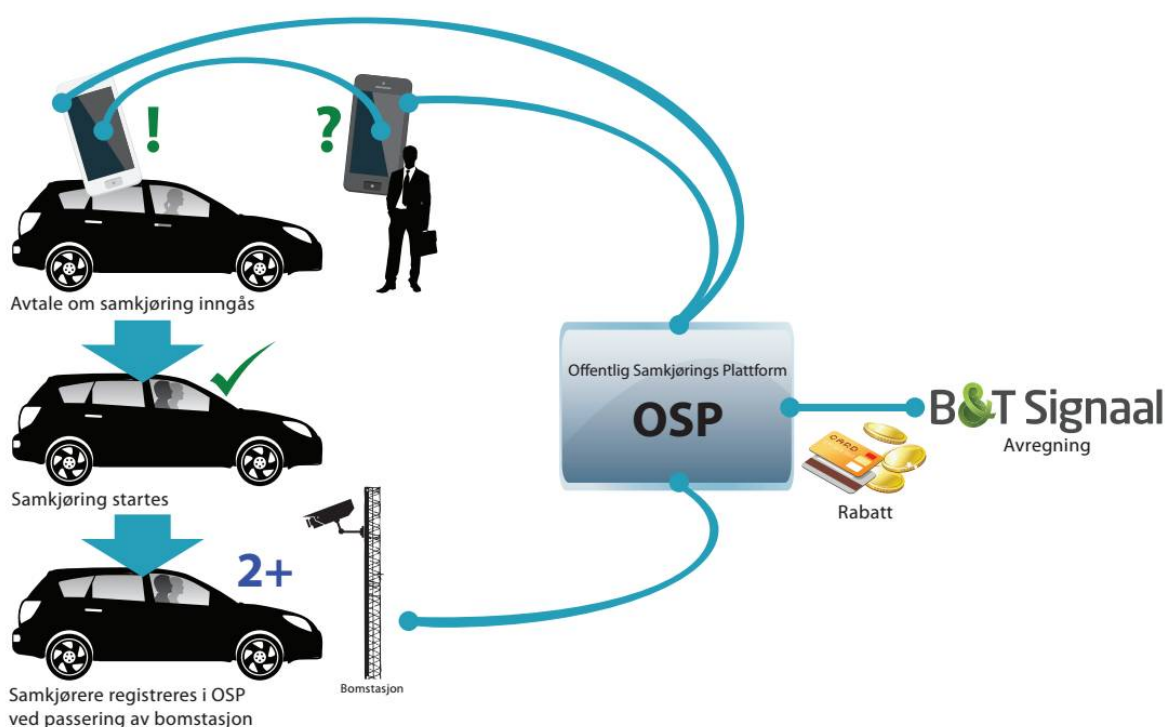
### Dokumenthistorikk:

<i>Versjon</i>	<i>Initialer</i>	<i>Dato</i>	<i>Kommentar</i>
0.1	AEH	2015-05-10	Utkast / disposisjon
1.0	HSR	2015-07-03	Første versjon



# 1 Om forprosjektet

Denne rapporten er resultat av et forprosjekt for å spesifisere teknisk løsning for incentivordning for samkjøring. Hensikten er at sjåfør skal kunne gis rabatt på betaling av bompenger og i parkeringshus dersom han eller hun har hatt med minst en passasjer på turen sin. Rapporten inneholder også et utkast til prosjektplan og et overslag på kostnader. Som utgangspunkt for denne løsningen har vi brukt det eksisterende systemet OSP (Offentlig SamkjøringsPlattform), som utvides med en incentivordning. Det forutsettes at appene som skal benytte systemet kobler seg opp mot denne utvidelsen av OSP for validering av samkjøringer.



Vi har i forprosjektet utarbeidet en beskrivelse av en grunnleggende sporings- og registreringsløsning for OSP. Det understrekes at vårt forslag ikke er endelig og at det er rom for endringer og detaljering av spesifikasjoner i det videre arbeidet. Forslaget som presenteres, med prosjektplan og estimater, er et arbeid som må videreføres i en planleggingsfase. Det er også i denne fasen man vil se det totale omfanget av arbeidet. Estimatenes er tatt frem etter beste evne og gir en pekepinn på størrelsen for et slikt prosjekt.

Systemet fungerer i grove trekk ved at samkjøringsapplikasjonene som kjører på brukernes telefoner rapporterer til OSP sin incentivordning (fra nå OSP-I) når de passerer "hotspots". Et hotspot er et geografisk punkt som typisk er en bomstasjon eller inngangen til et parkeringshus. OSP-I sammenstiller så en liste med registrerte samkjøringer som tilgjengeliggjøres for bomselskapet eller parkeringsselskapet. Dette kan være både samkjøringer for enkeltkjøretøy eller for alle samkjøringer på en hotspot. På grunnlag av dette er selskapene i stand til å gi rabatter.

Bomselskapene sender liste over posisjonene (hotspots) for sine bomstasjoner til OSP-I. Baksystemene henter listen over alle bomstasjonenes posisjoner (hotspots) og holder samkjøringsappene oppdatert.

Samkjøringsapplikasjonene er selv ansvarlige for at rådata som kommer inn fra kjøretøyene (telefoner, nettbrett mm.) raffineres før de sendes til OSP-I.

Ved å bruke bilnummer som ”nøkkel” i rapportene, behøver man ikke å synkronisere noen brukerbases mellom OSP-I og bomselskapene.

Løsningen er i hovedsak rettet mot å håndtere bompasseringer, men vil fungere med alle incentivordninger som forutsetter at sjåfør og passasjer er samtidig i nærheten av et gitt punkt, for eksempel innkjøringen til et parkeringshus.

Rapporten inneholder også plan og estimater for å koble den første samkjøringsappen på systemet. Teknisk beskrivelse av hvordan dette er tenkt løst er også en del av rapporten.

Vi legger opp til et prosjekt som kan ferdigstilles til sommeren 2016. For at dette skal være mulig har vi en prosjektplan som vil levere en beta-/testversjon av både samkjøringsapplikasjon og OSP-I plattform klar for testing til 1. januar 2016.

## Innholdsfortegnelse

1	Om forprosjektet.....	2
2	Innledning.....	6
2.1	Formål .....	6
2.2	Bakgrunn .....	6
3	Aktører .....	7
3.1	Leverandør av samkjøringsapplikasjon.....	7
3.2	Leverandør av Offentlig Samkjøringsplattform Incentiv (OSP-I).....	7
3.3	Bomselskap .....	7
4	Begreper .....	8
4.1	OSP-I.....	8
4.2	Samkjøringsapp.....	8
4.3	Hotspots.....	8
5	Løsning og virkemåte.....	9
5.1	Hotspots.....	10
6	Krav til klienter .....	11
6.1	Klient/app.....	11
7	Utvikling av testklient .....	12
7.1	Eksisterende funksjonalitet.....	12
7.2	Forslag til løsning.....	12
7.3	Kobling til OSP-I.....	13
7.4	Prosjektgjennomføring .....	13
8	Risikovurderinger.....	14
8.1	Krav til bruker og teknologi .....	14
8.2	Falske samkjøringer .....	14
8.3	Forsinkelse og usikkerhet i posisjonsavlesning.....	14
9	Testfaser .....	15
10	Tidsplan.....	17
10.1	Testversjon OSP-I .....	17
10.2	Utvikling av testklient .....	18
10.3	Testing mot betaversjon og videre utvikling.....	19
10.4	Produksjonssetting.....	19
10.5	Estimert kostnad.....	20

---

11	Vedlegg 1: Validering av passeringer av hotspots under samkjøring .....	21
11.1	Krav til appene .....	21
12	Vedlegg 2: Iterasjonsplan.....	23
13	Vedlegg 3: Format på innrapportering .....	24
14	Vedlegg 4: Rapporteringsgrensesnitt: .....	25
15	Vedlegg 5: Distribusjon av hotspot-listen .....	27
16	Vedlegg 6: Vedlikehold av hotspot-listen .....	28

## 2 Innledning

### 2.1 Formål

Formålet med dette dokumentet er å spesifisere en teknisk løsning for en incentivordning for samkjøring som er tenkt benyttet av bomselskaper og parkeringshus. Dokumentet inneholder teknisk spesifisering og en overordnet plan for utvikling av løsningen med kostnadsramme og viktige milepæler. Rapporten er resultat av forprosjekt for implementasjonen av Offentlig SamkjøringsPlattform – Incentivordning.

### 2.2 Bakgrunn

Statens Vegvesens prosjekt ”Spontan Samkjøring” tar sikte på utnytte ledig passasjerkapasitet i privatbiler, samt å samordne transport mellom privatbiler, taxi og buss/bane. Prosjektet har testet ut flere mobile samkjøringsapplikasjoner (apper) som kobler sammen sjåfører og passasjerer basert på destinasjon og tidspunkt. Målet med prosjektet er å redusere antall personbiler på belastede strekninger og dermed bedre fremkommelighet og luftkvalitet.

Triona er leverandør av Offentlig SamkjøringsPlattform (OSP). Dette er et system som lar ulike samkjøringsapplikasjoner kommunisere med hverandre og dele informasjon om privatbiler, buss og bane slik at sjåfører og passasjerer kan deles på tvers av sluttbrukerens foretrukne applikasjon. På den måten søker man å lage et økosystem der man sømløst kan tilby og motta skyss uavhengig av applikasjon som benyttes.

OSP Incentiv (OSP-I) vil komme som en utvidelse av OSP. Hensikten med OSP-I er å tilby en plattform for innrapportering av samkjøringer slik at disse kan brukes som grunnlag for belønningsordninger. Den første versjonen av OSP-I skal brukes for å beregne rabatt i bomstasjoner og parkeringshus.

OSP-I vil kunne fungere som en frittstående modul og vil derfor ikke være avhengig av at apptilbyderen også deler turer via OSP. For å verifisere systemet og integrasjonen vil vi også utvikle støtte for OSP-I i en eksisterende samkjøringsapp som en del av prosjektet.

## **3 Aktører**

### **3.1 Leverandør av samkjøringsapplikasjon**

Carma og HentMEG er relevante samkjøringsløsninger i OSP sammenheng. Det finnes også en rekke andre som man må kontakte for å gi informasjon om OSP-I. Det fins en rekke samkjøringsapplikasjoner i markedet og undersøkelser rundt dette vil ikke være en del av prosjektets omfang, men en aktivitet som bør gjøres i det videre arbeidet.

Det vil være opp til leverandørene å implementere støtte for å kunne bruke OSP-I fra sine egne applikasjoner. Rapportering til OSP-I vil kunne gjøres i et tjenestegrensesnitt som utvikles som en del av løsningen. Forprosjektet har fokusert på å definere lavest mulig terskel for applikasjoner mot OSP-I. Vi tror at et av suksesskriteriene er at flest mulig samkjøringsapplikasjoner benytter systemet og dermed en stor brukermasse.

### **3.2 Leverandør av Offentlig Samkjøringsplattform Incentiv (OSP-I)**

Triona AS har tidligere implementert OSP og denne rapporten med prosjektplan og kostnadsramme forutsetter at implementasjonen av OSP-I blir utført av Triona og er en videreføring av tankegangen bak OSP.

### **3.3 Bomselskap**

Alle selskaper som registrerer bompasseringer, lagrer disse hos seg og bruker dette som grunnlag for fakturering. I dette prosjektet forutsettes det at denne jobben gjøres av B&T Signaal. Det anbefales også at man går ut med informasjon til alle andre aktuelle aktører, slik flest mulig har mulighet til å gjøre nødvendige tilpasninger for koble seg opp mot systemet senere.

## 4 Begreper

### 4.1 OSP-I

OSP-I er tjenesten som tar imot data fra samkjøringsappenes baksystemer, sammenstiller disse og tilgjengeliggjør bekreftede samkjøringer for bomselskapet. Tjenesten har også ansvaret for å holde databasen med hotspots oppdatert og pushe denne til appenes baksystemer dersom hotspotene endres. Tjenesten vil bli lagd som en skytjeneste som kjører på Microsoft Azure.

### 4.2 Samkjøringsapp

En mobil applikasjon som er installert på samkjørerens mobile enhet, i de aller fleste tilfeller en smarttelefon. Appene brukes for å planlegge og gjennomføre samkjøringer, samt å rapportere passeringer av hotspots.

Det forutsettes at samkjøringsappene kan samle inn data og raffinerer dem før de sendes videre til OSP-I. Dette må være et krav for å holde mengden data inn mot tjenesten på et minimum.

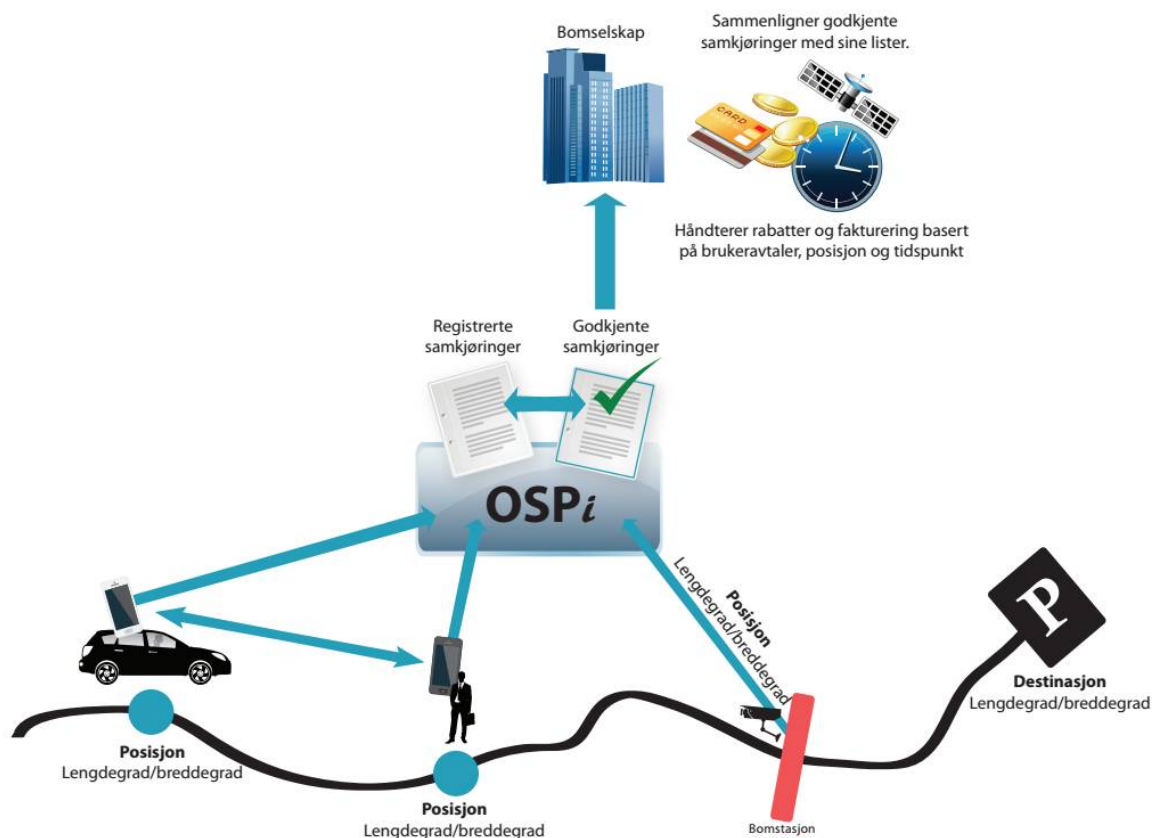
### 4.3 Hotspots

For at OSP-I skal fungere som tiltenkt så må appene rapportere når de har ferdigstilt en samkjøring. Rapporten må inneholde interessante geografiske punkter som har blitt passert og tidspunkt. I første omgang vil dette være bomstasjoner, men senere også når de ankommer parkeringshus eller når de passerer andre punkter av interesse. En kontinuerlig rapportering av posisjon vil føre til store mengder rapporteringer og generere unødvendig store mengder datatrafikk.

Dette løses ved at det etableres rapporteringspunkter, også kalt hotspots. En typisk hotspot vil være definert ved posisjonen til en bomstasjon. Leverandørens applikasjoner må holde sitt register over hotspots oppdatert ved hjelp av søking mot tjenester på OSP-I.



## 5 Løsning og virkemåte



Løsningen har tre hovedkomponenter. Den består av

1. Samkjøringsapper som brukes av samkjørerne for å planlegge og gjennomføre samkjøringene. Disse vil også filtrere og kvalitetssikre data før disse videresendes til OSP-I.
2. OSP-I sammenstiller data fra appene og genererer en liste av bekreftede samkjøringer. Denne listen sendes så videre til bomselskapet.
3. Bomselskapet som sammenstiller listen med sin egen liste over bompasseringer og bruker dette til å gi rabatter.

For å være helt sikker på at en samkjøring fant sted gjennom en bomring kan brukeren spores ved hjelp av GPS på telefonen. Dette kan gjøres på forskjellige måter. OSP-I prosjektet tar ikke sikte på å definere hvordan hver enkelt appleverandør håndterer verifisering av samkjøringer, men vi har gått gjennom noen mulige løsninger og presenterer dette i kapittel **Feil! Fant ikke referanseilden..** Vi presenterer også en teknisk løsning for hvordan dette skal gjøres i samkjøringsapplikasjonen som er tiltenkt som den første testklienten. Dette er beskrevet i kapittel 7.

Det vi legger opp til er at det vil finnes forskjellige hotspots som det skal rapporteres om har blitt passert eller ikke. I den første versjonen vil dette være bomstasjoner, men man kan tenke seg at dette senere kan utvides til flere områder; f.eks. parkeringshus.

## 5.1 Hotspots

Hotspots defineres som et punkt med en gitt posisjon. Hver hotspot må ha en unik identifikator. Ved innrapportering av passeringer, er det hotspotens identifikator som innrapporteres, ikke posisjonen.

For å kunne flytte/nedlegge/opprette nye bomstasjoner, bør disse ha en definert gyldighetsperiode, slik at man periodisk kan distribuere oppdateringer.

Hvis en hotspot ligger i nærheten en annen veg, for eksempel hvis en bomstasjon ligger ved et kryss eller på en rampe, vil OSP-I kunne få rapportert inn ”falske” data, dvs. samkjøring som var i nærheten av en bomstasjon, men som ikke passerte gjennom denne. Dette vil ikke ha noen konsekvens for incentivordningen, da bilen i dette tilfellet ikke vil være registrert med passering hos bomselskapet.

## 6 Krav til klienter

Det forutsettes at kommunikasjonen mellom samkjøringsappenes kommunikasjon er forsvarlig sikret og håndtert slik at det ikke er lett å forfalske samkjøringer. Det forutsettes også at både sjåførens og passasjerenes apper kan registrere en passering av hotspot. Det er likevel bare krav til at sjåførens samkjøringsrapport må inneholde en passering når listene oversendes bomselskapene og rabatt beregnes. Garantien for at samkjøringen fant sted til rett tidspunkt kommer fra kvittering på når passasjer gikk inn og ut av bilen.

### 6.1 Klient/app

Klienten holder en liste med hotspots og lagrer passeringen av slike i samkjøringsrapporten. Lista med hotspots vil være mulig å oppdatere ved å sjekke (pull) etter endringer ved konfigurerte mellomrom. På den måten trenger man ikke å gjøre en ny release av appen dersom det feks kommer nye bomstasjoner.

For å oppnå en tilfredsstillende nøyaktighet på posisjonen så må GPS på telefonen være påslått, noe som ikke er mulig å gjøre automatisk. Brukeren blir derfor bedt om å skru på GPS dersom denne ikke allerede er på når han går inn i bilen

GPS kan skrues av automatisk i det man går ut av bilen. Dette bør være konfigurerbart for brukeren.

Telefonen har en liste med hotspots lagret og vil sende rapporter når man er ferdig med samkjøringen. Dersom det ikke er datadekning på telefonen i det man avslutter kjøringen så kan disse settes i kø slik at de sendes i det man får dekning.

Appen bør si fra når samkjøringen er avsluttet. Dette kan enten gjøres manuelt ved at brukeren (sjåfør eller passasjer) avslutter denne i appen, eller så kan appen selv avslutte samkjøringen når den oppfatter at destinasjonen er nådd. Appen bør også implementere en timeoutfunksjon slik at ikke samkjøringen fortsetter i alle evighet dersom destinasjonen ikke nås eller at brukeren glemmer å avslutte appen manuelt. Dette er viktig for den av samkjørerne som oppnår rabatt, slik at samkjøringen blir gyldig hos bomselskapene.

For å få inn en begrenset datamengde til OSP-I er det nødvendig at samkjøringsappene bare sender inn validerte og godkjente samkjøringer. Det stilles derfor en del krav til appene som vi også ønsker å utforske i prosjektet. For å utarbeide en fullstendig liste og en metode for å integrere appene mot OSP-I er vi avhengig av å kjøre et prosjekt i parallell til OSP-I, som går på å integrere en eksisterende samkjøringsapplikasjon til systemet. Dette er beskrevet i neste kapittel.

## 7 Utvikling av testklient

Dette kapitlet definerer utviklingsløp for en samkjøringsapplikasjon. Vi bruker HentMEG som et eksempel i rapporten, men vi vil tro at kravene og tilpasningene er tilsvarende for andre løsninger.

### 7.1 Eksisterende funksjonalitet

I HentMeg setter sjåfør og passasjerer hvor de ønsker å reise fra og til, og planlagt tidspunkt for avreise.

På bakgrunn av dette matcher HentMeg baksystemet sjåfør og passasjer, og avtale inngås ved at begge bekrefter reiseplan med avtalt av- og på-stigning. Når passasjerer går inn i bilen bekreftes dette ved at en trykker på en knapp i HentMeg appen. Det sendes da en melding fra systemet til den andre telefonen, slik at denne også oppdaterer sin status.

Når passasjerer forlater bilen håndteres dette på samme måte.

#### 7.1.1 Krav til funksjonalitet i eksisterende løsning

For å kunne sikre at samkjøring har funnet sted og at passering har forekommet er det en del funksjonalitet som må utvikles i den eksisterende løsningen som må på plass. De følgende punktene lister opp funksjonalitet som ikke finnes i dag, men som kreves for å dekke kravene til OSP-I:

- **Ingen autentisering ved av- og på-stigning**  
Sjåføren kan i teorien gjennomføre en samkjøring og fortelle systemet at passasjerer går på og av uten at vedkommende er tilstede i bilen.
- **Løsningen har ikke kjennskap til hotspots**  
HentMeg kjenner kun startpunkt og stoppunkt for planlagt samkjøring, og kan derfor ikke vite hvilke hotspots som kan ha blitt passert
- **Ingen automatisk detektering av at passasjerer går ut av bilen**  
Hvis ingen av partene husker å bekrefte dette, vil ikke systemet vite når samkjøringen ble avsluttet.
- **Ingen sikkerhet i protokollen mellom app og baksystem**  
Det er ikke komplisert å manipulere samkjøring hvis man kjenner (eller sniffer) protokollen

### 7.2 Forslag til løsning

Vi tilstreber å få til en løsning hvor man ikke er avhengig at både sjåfør og passasjerer har HentMeg aktiv under kjøreturen. Det er i første versjonen bare sjåføren som har en økonomisk fordel av samkjøringen og krav til registrering av hotspots ligger derfor på henne.

#### 7.2.1 Autentisering ved av- og på-stigning

Det sendes allerede melding til de andre enhetene når en av enhetene signalerer at passasjerer går på eller av. Når dette skjer kan enheten kvittere ved å sende sin nåværende posisjon. På denne måten kan det sjekkes at enhetene er i rimelig nærhet av hverandre.

### 7.2.2 Håndtering av hotspots

HentMeg må periodisk laste ned liste over hotspots. På sjåførens enhet krever vi at GPS er på og aktiv, og detekterer når vi er i nærheten av en hotspot. Da sendes det en melding til baksystemet som oppdaterer listen over hotspots som har eller kan ha blitt passert.

Samtidig sendes et signal (GCM) til passasjerens enhet for å bekrefte posisjon. På denne måten kan det sjekkes at enhetene er i rimelig nærhet av hverandre ved passering.

### 7.2.3 Detektering av avsluttet samkjøring

En samkjøring regnes som avsluttet når:

- Det bekreftes at den siste passasjereren går ut av bilen.
- Sjåføren avslutter HentMeg
- Passasjerens app ikke lenger er i nærheten av sjåførens app  
Dette kan verifiseres ved at HentMeg baksystemet periodisk sender signal om å bekrefte posisjon til alle samkjørende enheter.

### 7.2.4 Forbedre sikkerheten.

Kommunikasjonen bør gå via https. Vi bør også implementere sikkerhet i protokollen.

## 7.3 Kobling til OSP-I

Når en samkjøring avsluttes, sendes informasjonen om samkjøringen til OSP-I.

Det rapporteres inn bilnummer, liste over hotspots man har vært i nærheten av med tidsstempel, samt samkjøringens start- og stopp-tidspunkt.

## 7.4 Prosjektgjennomføring

Vi anbefaler at dette prosjektet kjøres i parallell med utviklingen av OSP-I, slik at man hele veien har en testklient tilgjengelig. Man vil på denne måten også kunne luke ut «barnesykdommer» hos OSP-I før man går ut med en bredere testing.

I tillegg anbefaler vi at det kjøres en fase med brukertesting av denne samkjøringsapplikasjonen for å finne ut hva man kan gjøre bedre for å få en enklest mulig applikasjon for sluttbrukerne.

Prosjektplanen med estimater er beskrevet i kapittel 10 sammen med prosjektplan for OSP-I.

## 8 Risikovurderinger

### 8.1 Krav til bruker og teknologi

For at OSP-I skal kunne bli en suksess så settes det en del krav både til brukeren og til teknologien. Telefonen må være påslått, den må ha tilgang til datatrafikk og posisjoneringssystemet må fungere godt nok. Det er en helt klar forutsetning at brukeren behersker bruk av smarttelefon.

Dersom noe av dette feiler så vil ikke samkjøringen bli registrert og sjåføren vil gå glipp av rabatten selv om han eller hun har gjort alt riktig.

En senere versjon av OSP-I bør inneholde muligheten til at samkjørerne selv kan sjekke hvilke bompasseringer som er gjort og hva som er registrert hos bomsekskapet.

### 8.2 Falske samkjøringer

Ettersom OSP-I åpner for at det blir en økonomisk gevinst ved å ta på en passasjer så vil det være nærliggende å tro at det vil bli attraktivt å prøve og lure systemet.

Vi kan delvis sikre oss mot dette ved å kreve at appene oppretter en felles brukerbase med OSP-I, som igjen synkroniserer sin brukerbase med bomsekskapet. I tillegg kan det kreves å kunne dokumentere at samkjøring har funnet sted ved at passasjerene også rapporterer bompasseringer.

Vi kan likevel ikke sikre oss mot at noen tar med seg dobbelt sett med telefoner for å lure systemet til å tro at det fantes en passasjer i bilen.

Det finnes også programvare som tillater emulering av posisjonsdata slik at telefonen kan lures til å rapportere til at den befinner seg et sted der den egentlig ikke er.

Vår vurdering er likevel at problemet med falske samkjøringer neppe blir særlig stort ettersom det skal en del innsats til for å jukse. Dette er likevel et område vi må belyse ytterligere gjennom prosjektet. En risikoanalyse av dette bør foreligge som en del av prosjektets leveranse.

### 8.3 Forsinkelse og usikkerhet i posisjonsavlesning

Når GPS aktiveres går det som regel noe tid før den er i stand til å rapportere posisjon. Hvis brukeren slår på GPS i det han eller hun går inn i bilen, er det ikke sikkert at GPS er klar før bilen passerer et hotspot. Man vil da ikke få registrert passeringen og sjåføren vil derfor heller ikke få rabatt for den.

Posisjonsavlesninger kan også ha en grad av usikkerhet. Dersom GPS-signal ikke er tilgjengelig så vil de fleste enheter kunne bruke trådløse nettverk og 3G/4G-nett for å estimere en posisjon, men nøyaktigheten her er mye større enn hvis GPS benyttes.

På Android-enheter inneholder en posisjonsrapportering også data om hvordan posisjonen ble innhentet, dette er ikke tilfelle på iOS-enheter.

## 9 Testfaser

Testing av systemet er en sentral del av utviklingen. Fysiske tester ute på vegen er dyrt og tidkrevende. Det legges derfor opp til et omfangsrikt testregime for å kunne simulere bruk og innsamling av data. Dette vil være kostdrivende ved første versjon, men gjenbruk av testene vil sørge for at vi til enhver tid har kontroll over nye versjoner og leveranser.

Den tekniske implementasjonen av systemet bør testes stegvis.

1. Første teststeg i systemet er **enhetstester** som tester de enkelte funksjonelle kravene i OSP-I mens det utvikles. Dette gjøres av utvikler, og skal fange opp enkle funksjonsfeil.
2. Deretter vil det bli bygget **automatiserte integrasjonstester**. Disse skal teste at helheten i systemet (OSP-I) fungerer. For å gjøre dette, bruker man mock-rapportering og mock-servicer. Disse testene skal håndtere kjente use-caser og kjente feilkilder, og vil bygges etter hvert som funksjonalitet er på plass, og videreutvikles etter hvert som videre testing avdekker nye problemer. Disse testene bygges også av utviklerne, men benytter i større grad innspill fra andre involverte.

Mock-rapportering er testkode som opptrer som en samkjøringsapp ville gjort. Denne testkoden vil kunne teste alle grenseverdier man setter opp i systemet. For eksempel, rapporterer man data fra «sjåfører» og «passasjerer» som akkurat kvalifiserer til samkjøring, og man rapporterer data som akkurat ikke kvalifiserer. Man kan feilrapportere inn i OSP-I ved å rapportere på hotspots som ikke finnes, samtidig som man kan teste hvordan OSP-I fungerer ved å sende data mange ganger, slik at man kan simulere dårlig implementering fra appleverandørenes side. I dette testmiljøet, vil man lagre data som ellers, slik at man kan se hvordan systemet takler alle typer input.

Mock-servicer vil kunne rapporteres til på samme måte som man vil rapportere til hotspot-leverandører senere, slik at man kan sikre at hotspot-backsystemene mottar riktig data om riktige passeringer på riktig måte.

I dette systemet vil man også kunne simulere feil som oppstår i OSP-I senere, og bruke det til å finne ut hva og hvorfor ting feiler. Ved å bygge tester som simulerer feilene som oppstår senere, sørger man for at de ikke oppstår igjen, og samtidig at nye rettinger ikke ødelegger det som fungerte tidligere.

3. Man åpner en **testservice** for alle appleverandører slik at de kan teste OSP-I-APIer og gjøre justeringer for å kunne rapportere til OSP-I. Denne testservicen rapporterer ikke til noen hotspots, men tar vare på dataen og logger, slik at man kan bruke det til å vurdere kvaliteten på data som kommer inn fra appleverandørene. Denne kvaliteten vil være et viktig kriterium for å vurdere tilgang til OSP-I i produksjon. Dette kan gjøres allerede når man har en prototyp klar. Slik kan man gi appleverandørene tid til å tilpasse seg systemet og til å bli i stand til å levere kvalitetsdata.
4. Man starter opp testing mot OSP-I med **en eller flere APP-er**. Her må man sørge for at valgte apper er i stand til å levere brukbar data. Man vil kjøre igjennom hotspots med en og flere telefoner (helst av flere typer og modeller) aktive i bilen for å teste funksjonalitet og datakvalitet. Disse testdataene vil kunne brukes til å justere innstillinger i OSP og vurdere

hvor grenseverdier for hva som skal godkjennes som samkjøringer bør være. Disse dataene kan bruke nevnte testservice, og skal ikke rapporteres til hotspot. Denne fasen vil kunne gi noen nye admin-krav til OSP-I.

5. **Fullskala testing** med testgrupper av samkjørere. Her skal hele produksjonssystemet benyttes, og man bruker det som det skal brukes. Her må også rapporteringsmodulen til B&T være på plass, og de må involveres. Dette er akseptansetestingen i systemet og resultatene vil avgjøre når OSP-I kan gå i produksjon.



## 10 Tidsplan

Løsningen bør utvikles iterativt. Det er mange muligheter når man skal spesifisere detaljene ved et slikt prosjekt fremover og det kan bli mange aktører som skal kobles på systemet med forskjellige roller. Prosjektet vil kreve involvering fra flere interessenter i en tidlig fase og det bør settes av ressurser og tid til å håndtere dette. Et godt planleggingsprosjekt som ender i et tydelig mål bilde er et viktig suksesskriterie for incentivordningen. Dette dokumentet dekker den tekniske løsningen og estimatene dekker ikke krav til innsats utenfor OSP-I løsningen.

Vi anbefaler at systemet utvikles med smidige utvikling og at kunden er delaktig i prosessen. Dette vil sørge for at man hele tiden kan korrigere retningen på utviklingen etter hvert som man har flere erfaringer å basere valgene på. Dette vil også sørge for at man får rett funksjonalitet, i stedet for at man binder seg til det man tror ved prosjektets oppstart.

Vi foreslår et utviklingsløp som resulterer i en beta/test-versjon. Videre prosjektløp bygger videre på resultater fra resultater med testing mot denne betaversjonen.



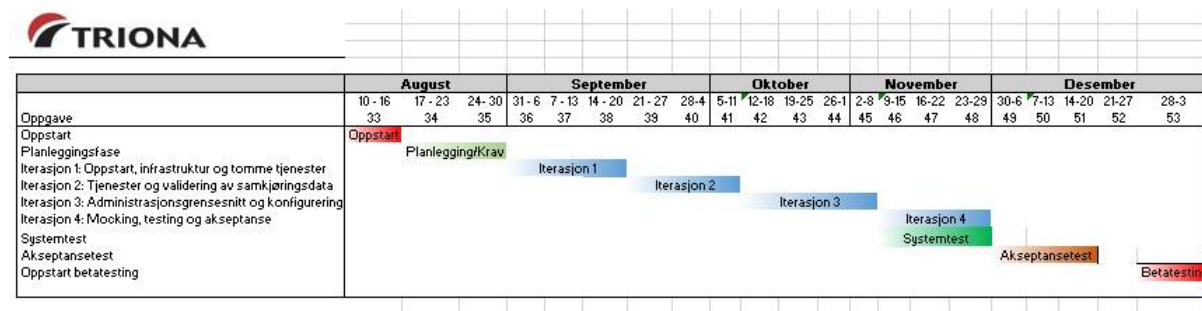
*Overordnet prosjektplan*

### 10.1 Testversjon OSP-I

Testversjonen av OSP-I vil inneholde:

- Tjeneste for uthenting av hotspotinformasjon
- Tjeneste for rapportering av samkjøringer med hotspots
- Administrasjon av hotspots og apptilgang
- Tjeneste for rapportering til aktuelt bomselskap
- Dokumentasjon av tjenestene
- Testinfrastruktur og automatiserte usecase-tester

Testversjonen av OSP-I utvikles høsten 2015. Figuren viser prosjektplan med milepæler.



Prosjektplan for testversjon

Nedenfor følger innholdet i iterasjonene og en estimert størrelse på hver iterasjon.

<b>Iterasjon 1:</b> Oppstart, infrastruktur og tomme tjenester	170 Timer
<b>Iterasjon 2:</b> Tjenester og validering av samkjøringsdata	180 Timer
<b>Iterasjon 3:</b> Administrasjonsgrensesnitt og konfigurering	250 Timer
<b>Iterasjon 4:</b> Mocking, testing og akseptanse	180 Timer
Planleggingsfase: Kravinnhenting, spesifisering, oppstart	80 Timer
<b>Totalt</b>	<b>860 Timer</b>

For detaljert informasjon om innhold i hver iterasjon og funksjonalitet som blir utviklet i hver fase, se kapittel 12.

## 10.2 Utvikling av testklient

Det anbefales at det startes et prosjekt i parallell med OSP-I, som fokuserer på å integrere en samkjøringsapplikasjon mot systemet. Løsningen er beskrevet i kapittel 7. Utviklingen av testklient som er integrert med OSP-I vil derfor også være klar i betaversjon til 1. januar 2016.

Testklienten vi har beskrevet har en del ny funksjonalitet som må på plass før sikkerhet og registrering av samkjøringer er god nok til å garantere den kvaliteten vi ønsker for rabattberegning. I tillegg er det en del oppgaver som går på selve integrasjonen mot OSP-I. Vi har delt dette i to prosjekter for å få et bilde av hvor mye en integrasjon mot OSP-I kan koste også for andre samkjøringsapper.

I kostnadsbildet har vi også lagt på et eget prosjekt for brukertesting. Vi tror at bruken av slike systemer er helt avhengig av enkelhet og sikkerhet. Sikkerheten er fokuset i den tekniske delen, mens vi ved å kjøre et prosjekt på funksjonalitet gjennom brukertesting vil kunne få en løsning som er enklere og mer tilpasset faktiske brukere.

<b>Integrasjon mot OSP-I</b>	330 Timer
<b>Utvikling av funksjonalitet for å dekke nyutvikling i HentMEG</b>	250 Timer
<b>Brukertesting</b>	200 Timer
<b>Totalt</b>	<b>780 Timer</b>

### **10.3 Testing mot betaversjon og videre utvikling**

Umiddelbart etter leveranse av testversjonen bør man gå inn i en testfase hvor man tester intensivt med brukere og utvalgte samkjøringsapper. Det betyr at man i løpet av utviklingen involverer leverandør av samkjøringsapper til å koble seg opp mot systemet og gjøre de endringene som kreves for å verifisere samkjøringene.

Når testperioden starter i 2016 må API'et for kommunikasjon med OSP-I være åpent for alle, slik at flest mulig samkjøringsapper kan koble seg på og delta i testingen. Dette kan pågå i parallell med utviklingen av komplett løsning, siden testversjonen skal inneholde den grunnleggende funksjonaliteten for OSP-I.

Prosjektteamet som har hatt ansvaret for utviklingen av OSP-I må være tilgjengelig underveis i testingen for å kunne levere nye versjoner hvis man oppdager feil. Vi foreslår at testperioden går i 2 måneder før man starter opp full utvikling igjen.

Utviklingsprosjekt for å ferdigstille den endelige versjonen bør derfor planlegges fra mars-juni. Da har man en måned med test og produksjonssetting før fullskala produksjonssetting. Det vil da allerede ha vært utført kontinuerlig testing i hele utviklingsløpet og vi mener en måned dermed vil være lang nok tid for utrulling.

### **10.4 Produksjonssetting**

Vi sikter inn på en fullskala implementasjon til 1.juli 2016. I løpet av utviklingen i starten av 2016 må man sørge for at alle samkjøringsapplikasjoner som har intensjon om å koble seg på incentivordningen involveres og gis mulighet til å tilfredsstille kravene for dette.

Det bør være et betydelig antall klienter og samkjørere som er klar til å benytte seg av løsningen ved fullskala produksjonssetting.

## 10.5 Estimert kostnad

Dette kapitlet summerer opp de forskjellige fasene og viser kostnadsoverslag på de første fasene. De senere fasene må spesifiseres underveis i prosjektet.

Fase	Tidsperiode	Omfang (timer)	Kommentar
Utvikling av OSP-I	August 15 – Januar 16	860	
Utvikling av ny funksjonalitet i HentMEG	August 15 – Januar 16	250	
Utvikling av integrasjon av HentMEG mot OSP-I	August 15 – Januar 16	330	
Brukertesting	September 15 – November 15	200	
Testfase av betaversjoner	Januar 16 – mars 16	-	Dette avhenger av eksterne testressurser. Feilretting og tilpasninger av løsning avhenger av hva testfasen resulterer i. Vi anbefaler at disse to månedene med testing dekkes opp med en blanding av ressurser fra leverandør og eksterne. Kundeinvolvering er viktig i denne fasen.
Utvikling og produksjonssetting av første versjon	Mars 16 - Juli 16	.	Dette er 4 måneder med utvikling for å tilfredsstille nye krav og feilretting fra testfasen. Størrelsen på dette prosjektet avhenger av for mange faktorer til å komme med et estimat i nåværende fase.
<b>Totalt</b>	<b>August 2015 – 1. juli 2016</b>	<b>1640</b>	Dette er et foreløpig anslag basert på spesifikasjon som er utledet av resultatet av forprosjektet. Dette må raffineres i løpet av planleggingsfasen for prosjektet.

## 11 Vedlegg 1: Validering av passeringer av hotspots under samkjøring

### 11.1 Krav til appene

Dette er informasjonen vi må ha inn til OSP-I for å kunne validere samkjøringene og sammenfatte informasjon til hotspots. Hvordan appene håndterer disse kravene, må kunne ettergås for at de skal kunne godtas for å levere data til OSP-I. Denne godkjenningen må gjøres manuelt, men det kan gjøres på grunnlag av data som blir levert til testservicen som vil være mulig å utvikle mot for appleverandører. Følgende krav virker å være nødvendige:

1. Appsystemene må registrere start- og sluttid for samkjøringene.
2. Appsystemene må registrere hvilke hotspots som passerer under denne samkjøringen.
3. Når samkjøringen avsluttes, skal appsystemene rapportere til OSP-I.

#### 11.1.1 Appsystemene må registrere start- og sluttid for samkjøringene

##### Sjåførens og passasjerens posisjon korreleres

Både passasjerens og sjåførens posisjon registreres. Telefonens mekanismer for posisjonsbestemmelse kan benyttes. Det kan tenkes at det er andre ting enn bare GPS som vil fungere for det. Dersom app-ene ikke er nærme nok til at samkjøring kan godkjennes, sendes det en melding til sjåføren, slik at han kan be passasjerer skru på mobiltelefonen e.l. Sjåføren vil dermed ha kontroll på at samkjøringen blir gyldig før man starter. Samme prosess, eller en forenklet versjon vil måtte gjennomføres ved avslutning. Det kan gjøres ved at sjåføren velger «Avslutt», og passasjerer får en forespørsel om å avslutte. Så lenge de to registrerte posisjonene her er innenfor en viss avstand, er hele samkjøringen godkjent.

##### Passasjerer autentiserer samkjøringen uten bruk av posisjon

Dette kan gjøres ved at passasjerer via sin app bekrefter at han/hun går inn og ut av bilen. Denne bekreftelsen skjer ved at passasjerer app deler en kode med sjåføren, eller en qr-kode som sjåføren kan scanne. Tilsvarende validering må da skje ved avstigning. Her er det sjåførens ansvar og interesse at samkjøringen blir validert på riktige tider. Sjåføren vil også her kunne få bekreftelse på at samkjøringen er godkjent på app-nivå.

#### 11.1.2 Appsystemene må registrere hvilke hotspots som passerer under denne samkjøringen.

Et generelt problem med GPS-sporing er om man passerer i nærheten av en bomstasjon og ikke igjennom. Dette vil bare føre til ekstra data, og ikke få noen konsekvenser. Man vil ikke kunne få rabatt for noe man ikke har betalt for, så derfor vil dette ikke føre til problemer for brukeren eller bomselskapene. Samme type problem vil komme hvis man ikke GPS-sporer. Da vil man kanskje beregne inn flere mulige bomstasjoner, men man vil jo kun få rabatter for stasjoner man faktisk har passert. Så i begge disse tilfellene kan man ende opp med ekstra data som ikke er reelle passeringer.

##### Både sjåføren og passasjerer GPS-sporer ved bompasering

For å kunne oppnå dette, GPS-trackes både sjåfør og passasjer ved bompasering. Dette gir et ekstra sikkerhetsaspekt mot å lure systemet, siden man i tillegg til å registreres ved av- og påstigning også registreres ved hotspots under bilturen. For å oppnå dette, vil appen på en eller

annen måte måtte spore både sjåfør og passasjerer individuelt under bilturen. Det vil kreve at både sjåfør og passasjerer har telefonen på og appen aktiv under hele bilturen.

#### Kun sjåføren GPS-spores ved bompassering

Dette vil innebære samme sporingsmekanisme som for både sjåfør og passasjerer, men siden det er sjåføren som får fordelene av rabatter, vil det være i sjåførens interesse i større grad enn passasjerene. Normalt vil også sjåføren ha tilgang til lader i bilen, som vil begrense uheldige feil som skyldes at passasjerer gikk tom for batteri, glemte å ha på telefonen etc. Så lenge passasjerer er bekreftet sammen med sjåføren på påstigningstidspunktet og avstigningspunktet, er sannsynligheten for at de var sammen på passeringene imellom meget stor. Man vil i begge tilfeller kunne lure systemet ved å ha f.eks to mobiltelefoner. Ved å kjøre to biler fra samme sted til samme sted, vil man i dette tilfellet kunne lure systemet, men det virker ikke å være særlig hensiktsmessig.

#### Ruteberegning gir liste over trolige bomstasjoner

Fra- og til-sted registreres og brukes til å generere mulige/trolige bompasseringer. Dette systemet bør ta høyde for flere mulige ruter, i tilfelle en annen bompassering kan være mer hensiktsmessig grunnet normal trafikkmengde etc. Dette systemet vil trolig være mer komplisert å utvikle, men man vil samtidig oppnå større grad av personvern og trolig sikker nok validering.

#### Manuelt valg av bomstasjoner

Man kan se for seg en liste over bomstasjoner, evt. mulighet for å velge på et kart, og sjåføren velger hvilken man vil passere eller har passert avhengig av før eller etter turen. Dette vil trolig bli mer komplisert for sjåføren, men vil kanskje være det enkleste for en appleverandør.

#### **11.1.3 Når samkjøringen avsluttes, skal appsystemene rapportere til OSP-I**

Appene skal ikke rapportere hver hendelse til OSP-I, men rapportere essensiell data til OSP-I ved avslutning av samkjøringene. Disse rapportene må inneholde informasjon om sjåfør, bil og passasjerer, fra og til-tidspunkt for samkjøringen, samt liste over hotspot-passeringer. Dette er dataen OSP-I vil behøve for å velge rabattkoder og informere bomstasjonssystemene.

## 12 Vedlegg 2: Iterasjonsplan

Iterasjon	UseCase
Infrastruktur	Prosjektoppstart Rutiner for periodisk rensing av database Databaseoppsett Backup-rutiner Sette opp service-rammeverk Sette opp autentiserings-rammeverk
Mottak av samkjøringsdata fra appsystemer	Innrapporteringservice for samkjøringsapper Datamodellering Loggesystem for innrapporteringsfeil Returnere feilkoder ved feilaktige kall Lagre samkjøringsdata i database Autentisering i servicen
Validering av samkjøringer	Hente ut data fra samkjøringsdataen i et tidsintervall Matche sjåførere/biler med passasjerer på hotspot Matche i forhold til tidsdifferanser Vaske bort "ekstradata" Unittesting av alle grensetilfeller
Service for rapportering av eksisterende hotspots	Service som kan hentes data fra Datamodellering Autentisering
Rapporteringservice for samkjøringer	Service som kan hentes data fra Enkeltkall på bilnummer. per tidspunkt eller periodisk Rapportkall for periodisk informasjon per hotspot Datamodellering Autentisering
Administrasjonssystem	Datamodellering Autentisering GUI Serveroppsett for GUI Konfigurering på App-nivå Konfigurering på hotspotnivå
Mocktesting	Sette opp mock-klient for app-backsystem Sette opp mock-klient for hotspot-backsystem Definere use-caser Implementere use-caser som tester

## 13 Vedlegg 3: Format på innrapportering

Innrapportering fra samkjøringsappene skjer via REST-kall til OSP-I, med rapportdata i JSON-format.

Alle tidspunkt rapporteres i GMT

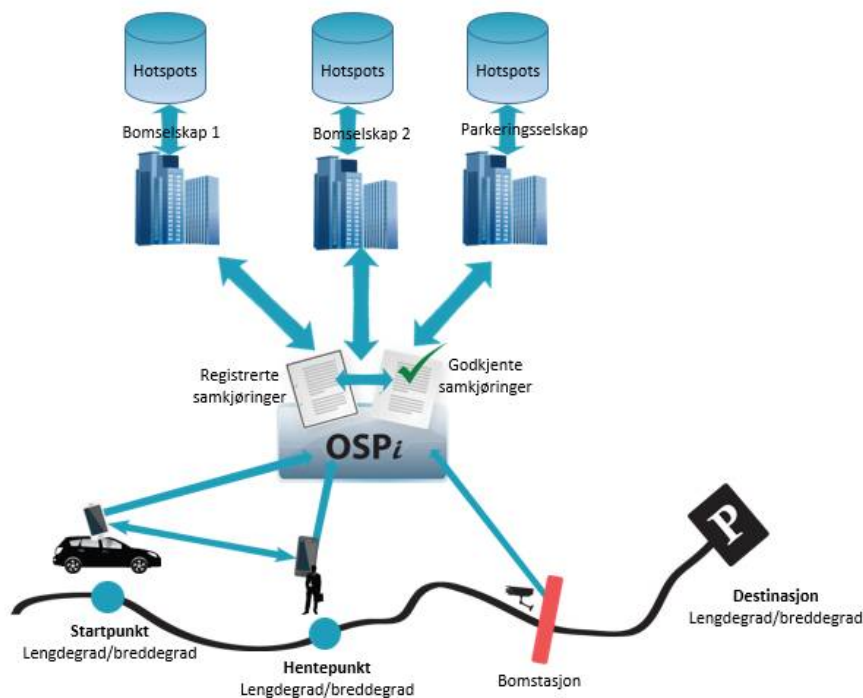
PUT /ridesharing-report

```
{
  "system": string,           // back-end system id e.g. "carma" or "hentmeg"
  "type" : string,           // "driver" or "passenger"
  "vehicle" : string,        // car registration number
  "timestamp" : string,      // ISO 8601 GMT: ex:"2015-06-11T15:08:23:02Z",
  "location" : string,       // hotspot name/id
  "userid" : string,         // user id
  "passengers" : [           // list of passengers if type is "driver"
    -{
      "userid" : string,     // user id for passenger
    }
  ]
}
```



## 14 Vedlegg 4: Rapporteringsgrensesnitt:

### Rapporteringsgrensesnitt:



Samkjøringsappene rapporterer hotspotpasseringer til OSP-I som en del av samkjøringsloggen. Passeringer skal kunne rapporteres for alle sjåførere og alle passasjerer individuelt. Samkjøringsappene bør være i stand til å rapportere kun én gang per person per hotspotpassering på en samkjøring. Appene må derfor ha rutiner som filtrerer data. Når og hvordan rapporten skal sendes til OSP-I vil variere. Dette må vurderes manuelt før en samkjøringsapp blir godkjent for å rapportere til OSP-I.

Appene må rapportere følgende data for hver passering: *SystemId* som er Id til App-systemet, slik at OSP-I kan identifisere kilden. *Type* som kan være sjåfør eller passasjer. *VehicleId* som er bilnummeret. *Timestamp* som er tidspunktet for registrering. *HotspotId* som sier hvilken hotspot registreringen gjelder. *UserId* som er brukeren (sjåføren eller passasjerene). Disse feltene er felles uavhengig av type. Sammen med sjåføren må i tillegg UserId-ene til passasjerene rapporteres. Ved hjelp av det, samt passasjerenes *VehicleId*, kan vi korrelere passasjerer med sjåføren og bekrefte samkjøringer.

Samkjøringsappene må kunne hente ut informasjon om hotspots fra OSP-I. Disse dataene bruker de til å stedfeste og knytte posisjonsdata fra sjåførene og passasjerene til det enkelte hotspot. OSP-I skal ikke stedfeste rapporterte data, og krever derfor *HotspotId* og ikke geografiske data inn.

De enkelte deltakerene i incentivordningen skal ha mulighet til å etterspørre gyldige samkjøringsdata for sine hotspots.

Samkjøringsdata etterspørres for en liste over en eller flere hotspots i et gitt tidsintervall.

Informasjonen som returneres er:

*HotspotId* siden det er mest sannsynlig at enkelte systemer behandler data fra forskjellige hotspots.

*VehicleId* for å identifisere kjøretøy.

Sjåførens *UserId* for å kunne rette incentivene til riktig bruker. Det er jo tenkbart at flere sjåfører kan benytte samme kjøretøy.

*Timestamp* for passeringen (passeringstidspunktet til sjåføren dersom det er avvikende tider).

*PassengerCount* for antall godkjente passasjerer.

Dersom man innfører ekstra incentivordninger for passasjerer, vil man i tillegg til antallet passasjerer måtte sende liste med *UserId* for passasjerer.

Det kan eventuelt implementeres en rapporteringsservice som periodisk henter ut og sender liste over alle godkjente samkjøringer i perioden til deltakerne.

## 15 Vedlegg 5: Distribusjon av hotspot-listen

For å unngå å måtte ut med ny versjon av samkjøringsappene når det skjer endringer i listen av hotspots, bør appene periodisk hente oppdatert liste over hotspots. For å sikre konsistens, og for å ha et felles distribusjonspunkt for denne listen, foreslår vi at OSP-I benyttes til å distribuere den offisielle listen over hotspots. Det forutsettes at appene henter oppdatert liste fra OSP-I minst en gang i døgnet.

Listen hentes fra OSP-I via et REST-kall

GET /hotspots

JSON response:

```
[
  _{
    "hotspot": string,           // unique name for this hotspot
    "latitude" : number,        // hotspot latitude
    "longitude" : number,       // hotspot longitude
    "speedlimit" : number,      // speed limit at hotspot in km/h
    "validfrom" : string,       // ISO 8601 GMT: ex:"2015-12-31T23:00:00Z",
    "validto" : string,         // ISO 8601 GMT: ex:"2016-12-31T23:00:00Z",
  }
]
```

## 16 Vedlegg 6: Vedlikehold av hotspot-listen

Hotspots i OSP-I's database har en referanse til hvem som "eier" hver hotspot.

De enkelte deltakerene i incentivordningen vedlikeholder sine hotspots ved å laste opp oppdatert liste til OSP-I. Å deaktivere en hotspot kan gjøres med umiddelbar virkning, da OSP-I ikke vil rapportere samkjøringer for deaktiverte hotspots. Å legge til nye hotspots må gjøres i god tid før de skal aktiveres, da oppdatert liste må være distribuert til alle appene før disse vil kunne rapportere passeringer.