

# Att öppna datakällor för tredjeparts- utveckling

- rekommendationer för  
kollektivtrafikbranschen

DANIEL RUDMARK

2013-01-30

# Innehåll

Att öppna datakällor för tredjeparts-utveckling .....	1
Vad är ett öppen datakälla och vilken nytta kan det skapa? .....	3
Utmaningar vid utformning av API:ers funktionalitet .....	4
Process för framtagning av öppna API:er i kollektivtrafikbranschen .....	4
Case: Trafikverket Realtidsinformation Resenär .....	18
Kartlägg nuläge och önskad framtid .....	18
Intervjuer med Trafikverket .....	18
Intervju med Tredjepartsutvecklare .....	19
Workshop .....	19
Genomförbarhet och teknisk lösning .....	21
Arkitektur .....	21
Funktionalitet .....	21
Feedback på specifikationen .....	22
Utveckling, test och driftsättning .....	22
Lansering och utvärdering .....	24

## Vad är ett öppen datakälla och vilken nytta kan det skapa?

Under senare år har *öppna datakällor* blivit en allt mer populär ansats för att öka spridningen av information från kollektivtrafiken. Genom att ex. tidtabellinformation och realtidsinformation blir tillgängliga för *tredjepartsutvecklare* utanför den egna organisationen kan nya innovativa tjänster skapas. Det finns ingen entydig definition på vad en öppen datakälla är men typiskt menas att 1) den saknar *exklusivitet* – d.v.s. alla intresserade parter har möjlighet att använda datakällan och 2) den är *möjlig att återanvända i nya tjänster* helt utan eller endast med mindre begränsningar. Tekniskt kan en öppen datakälla vara allt från en publicerad Excel-fil till en databas till ett API. Inom kollektivtrafikbranschen används företrädesvis *öppna API:er* för att tillgängliggöra information till tredje part.

API står för Application Programming Interface möjliggör för en programmerare att använda rutiner eller hämta data som ligger utanför den kod man skrivit själv. Traditionellt har ett API varit en ingång till olika kodbibliotek på ex. programmerarens egen dator men med internets intåg möjliggörs även anrop till kod som fysiskt kan ligga någon helt annanstans. Med *öppet* API menas att det är tillgängligt i princip för vem som än är intresserad och därmed helt saknas eller finns mycket liten exklusivitet avseende tillgången till API:et<sup>1</sup>. Genom att tillhandahålla ett API mot organisationens data kan API:et ligga till grund för nya typer av tjänster och idag har ledande mjukvaruplattformar från ex. Google, Facebook, Twitter, Salesforce och Dropbox egna API:er – med olika grad av öppenhet – där de uppmanar utvecklare att bygga nya innovativa tjänster.

En typ av tjänster som kan skapas är sådana organisationen själva inte har möjlighet att ta fram. I Sverige tillhandahåller ex. Trafikverket dynamisk väginformation till externa parter via ett öppet API som stödjer standarden DATEX II. En av deras kunder är den holländska bilnavigatortillverkaren Tom-Tom. Tom-Tom samkör informationen från Trafikverket med egeninsamlad information (kring ex. hastighet vid en olyckplats) via en egenutvecklad algoritm och kan därmed ge ett beslutsunderlag till chauffören som Trafikverket själva inte kan leverera.

En annan typ av tjänst är spridning av information till nya plattformar och sammanhang. Storstockholms Lokaltrafik utvecklar gentemot sina resenärer idag endast en vanlig webbplats samt en mobil webbplats. Alla plattformsspecifika reseplanerare eller realtidsappar (ex. för iPhone, Android eller Windows Phone) utvecklas istället av tredjepartsutvecklare som får data via plattformen Trafiklab. Idag finns flera mycket populära appar (200 000+ nedladdningar) med höga användarbetyg.

En ytterligare typ av tjänst är sådana som når en liten nischgrupp, och som det är svårt för API-tillhandahållaren att själv motivera en utveckling för. Tjänsten blir dock fullt genomförbar för en extern aktör att implementera via ett API eftersom API-ägarens marginalkostnad för nya tjänster är mycket låg. Ett sådant exempel är polisens initiativ "Nattknappen". Där kan personer ringa in mellan 23:30 och 03:30 på natten om den inringande känner sig otrygg på väg hem. Känslan av trygghet ökar då man pratar med någon i telefon och den man pratar med på Nattknappen kan också snabbt agera om något trots allt skulle hända. I de system som används av Nattknappens volontärer finns en koppling till kollektivtrafikens API:er vilket möjliggör att polisens kan guida den inringande över telefon i kollektivtrafiksystemet.

---

<sup>1</sup> Implicit avses också att API:et är webbaserat API, tillgängligt för de flesta olika plattformar som ex. SOAP eller REST.

## Utmaningar vid utformning av API:ers funktionalitet

På många sätt innebär arbetet med att ta fram ett öppet API samma utmaningar som andra IT-utvecklingsprojekt. Att säkerställa budget och resurser, ha en god kvalitetssäkring samt en klarhet i vad som ska utvecklas för att projektet ska leda till önskade effekter. Denna rapport beskriver dock särskilda aspekter som uppstår när en organisation står inför att öppna datakällor för tredjepartsutveckling.

Vid framtagande av ett öppet API finns mer specifika omständigheter som gör processen mer svårnavigerad (än ex. i ett organisationsinternt utvecklingsprojekt) och vars hantering får stor inverkan på den innovation som kan komma att ske:

- *Organisatoriska dimensioner*: Processen att öppna upp en organisations information för utomstående och ofta okända aktörer att skapa helt nya tjänster där organisationen har litet eller inget inflytande över utformningen är ofta en tämligen lång och omfattande process av förankring.
- *Funktionella dimensioner*: Typiskt stöder ett API (eller en del av ett API – ex. en metod) ett användningsfall (ex. "Sök parkering inom ett visst område"). Detta gör det enkelt för utvecklare att implementera nya tjänster – men om API:et saknar stöd för andra relevanta användningsfall ("sök parkering baserat på adress") kan innovativa tjänster utebli – eller leda till att utvecklare istället skrapar data från ex. webbsidor. En alltför bred ansats (med komplicerad "rådata" som kräver djup verksamhetskunskap och mycket arbete innan en tjänst kan driftsättas) kan å andra sidan göra att färre aktörer ger sig i kast med att utveckla tjänster.
- *Tekniska dimensioner*: Rent tekniskt kan ett API implementera en viss funktionalitet på en mängd olika sätt. Aspekter som ex. koordinatsystem, dataformat och dataöverföringsprotokoll behöver passa målgruppen i största möjliga mån.
- *Användardimensioner*: Med användare menas här gruppen tilltänkta (eller befintliga) utvecklare. Dessa kan vara svåra att identifiera för att kvalitetssäkra en viss lösning då de befinner sig helt utanför organisationen och det är svårt att veta vilka tjänster dessa kan komma att utveckla.

Nedan följer en generell process systematiskt adresserar frågeställningar ovan och som typiskt uppstår när en organisation initierar ett arbete att öppna sina datakällor för tredjepartsutveckling.

## Process för framtagning av öppna API:er i kollektivtrafikbranschen

Processen nedan baseras på flera lyckosamma projekt i arbetet med öppna datakällor för tredjepartsutvecklare och forskning på Viktoria Swedish ICT. Observera att processen är tänkt att användas som ett *stöd* i arbetet med att öppna datakällor och som ett kompletterande stöd till ev. befintliga projektmodeller i denna typ av projekt.

Steg	Beskrivning	Kommentar
<b>1. Kartlägg nuläge</b>		
<b>1.1 Innovativa tjänster i segmentet</b>	<p>Det första steget handlar om att skapa en bild över nuläget, främst med inriktning på befintliga innovativa tjänster inom branschen/segmentet som organisationen jobbar inom. På så sätt kan de möjligheterna med tredjepartsutveckling bli tydligare och användas som motvikt mot organisatoriska <i>risker</i> som identifierats inom projektet. Inom kollektivtrafik finns en mängd innovativa tjänster att inspireras av.</p>	<p><i>Källor för inspiration</i></p> <p>Internetsökning.</p> <p>Sökning på marknadsplatser för applikationer (ex iTunes, Google Play)</p> <p>Deltagande i branschkonferenser</p> <p>Trafiklab.se</p> <p><i>Bra exempelorganisationer i Sverige:</i></p> <p>Storstockholms Lokaltrafik</p> <p>Trafikverket (Väg)</p> <p>Samtrafiken</p> <p>Västrafik</p> <p>Skånetrafiken</p>

<p>1.1.1 Utvecklare av innovativa tjänster</p>	<p>Vilka typer av aktörer utvecklar intressanta tjänster för segmentet?</p>	<p>Ex. på egenskaper hos utvecklarna är om tjänsterna är kommersiella / gratis, om det finns många tjänsteleverantörer eller om tjänsteutvecklingen domineras av några få aktörer.</p>
<p>1.1.2</p>	<p>Vilka (om någon) affärsmodeller bygger tjänsterna på?</p>	<p>Sker det ekonomiska transaktioner mellan organisationen och tredjepartsutvecklaren? (ex. biljettförsäljning)</p> <p>Vilka nyttjandevillkor är nödvändiga att ställa på informationen för att möjliggöra tjänsteutvecklingen?</p>
<p>1.1.3</p>	<p>Hur föds tjänsterna med information?</p>	<p>Ex. Egenutvecklat API, Standarder, Rådata</p>
<p>1.2</p>	<p>Kartlägg organitoriska risker</p>	

1.2.1	Kartläggning av interna organisatoriska risker	Typiska risker inom kollektivtrafikbranschen är:  <i>Varumärke</i> – vad gäller ang. avsändare? Skall ex. organisationen stå med som avsändare av informationen?  <i>Vem har ansvar för att informationen är korrekt?</i>  <i>Vad händer om en utvecklare ändrar inställningar, ex. bytesmarginal?</i>  <i>Hur motiveras kostnaden för satsningen?</i>  <i>Klarar bakomliggande system alla anrop?</i>  <i>Hur stänga av en illvillig utvecklare?</i>
1.2.2	Hantering av organisatoriska risker hos befintliga tjänster	Hur befintliga tjänster löst riskerna identifierade ovan? Undersök hur tredjepartsprogram är utformade avseende licensvillkor, anmälningsprocess etc.

Steg	Beskrivning	Kommentar
2 <i>Önskad framtid</i>		
2.1	Interagera med potentiella/befintliga tjänsteutvecklare	<p>Vår erfarenhet är att en gemensam workshop med tjänsteutvecklare är att föredra. Förutom att involverade parter få utveckla sina synpunkter kan även en sådan aktivitet vara viktig för förankra projektet. Vår erfarenhet är också att riskerna med tredjepartsutveckling upplevs som mindre när dessa varit med i processen.</p> <p>Genom ett enskilt möte med tjänsteutvecklare möjliggörs en fördjupad dialog men möjligheterna att förankra en lösning hos alla parter minskar.</p> <p>Det minst resurskrävande är att samla in behov och önskemål via ett on-lineforum – men erfarenheter ger att feedbacken där kan bli begränsad och fragmenterad.</p>

## 2.2

## Interagera med befintliga API-tillhandahållare

Goda exempel i Sverige som ex. SL, Samtrafiken och Västtrafik som framgångsrikt arbetat med öppna datakällor och tredjepartsutvecklare kan avsevärt förenkla förankringen.

Dessa exempel kan ex. visa på hur man kan hämta hem de kostnader som ett öppet API medför genom en mycket bredare tjänsteportfölj mot resenär, utvecklat av tredjepartsaktörer.

Steg	Beskrivning	Kommentar
3	Genomförbarhet och principiell lösning	
3.1	Arkitektur	<p>Finns den eftersökta informationen tillgänglig i databaser för tredjepartsutveckling?</p> <p>Hur ofta uppdateras informationen? Om det ex. rör sig om realtidsinformation behöver sannolikt någon form av begränsning av anrop göras. För mer statisk information är detta däremot troligen inget problem.</p> <p>Innebär tjänsten att tredjepartsutvecklare ska läsa och/eller skriva information? Om tjänsten innebär att skrivning behöver ske, hur verksamhetskritisk är den informationen som skrivs? Hur hanteras användare om det är verksamhetskritisk information? En variant är att ha ett testsystem tillgängligt för samtliga intresserade användare och ett produktionssystem där endast godkända användare får tillgång.</p>

3.2	Affärsmodeller	Hur ser affärsmodellen ut? Ligger vinsten i att ha många mindre utvecklare? Är det viktigt att attrahera stora bolag/partners? Sker några ekonomiska transaktioner mellan parterna (ex. procent på försäljning via API:et)?
3.3	Funktionalitet	<p>Vilken funktionalitet ska erbjudas?</p> <p>I det fall tjänsterutvecklingen ska fokuseras på att sprida befintliga information (ex. webbaserade) i fler kanaler kan sådana befintliga tjänster med fördel användas som utgångspunkt. Denna funktionalitet har traditionellt varit det vanligaste inom kollektivtrafikbranschen.</p> <p>Om man däremot vill åstadkomma helt nya typer av tjänster bör organisationen överväga att erbjuda mer av obearbetad rådata.</p>

## 3.4

## Fortsatt dialog med tredjepartsutvecklare

Att fortsätta dialogen med tredjepartsutvecklare under denna och senare faser, ex. via ett offentligt webbforum, kan öka bidrag till projektet på två sätt:

1. Feedback och kvalitetssäkring av förslag på lösningar från organisationen
2. Att hålla intresset levande samt informera tredjepartsutvecklare om projektets status.

Steg	Beskrivning	Kommentar
4	Utveckling och test	
4.1	Målgrupp för tjänstutveckling	
4.1.1	Företag	<p>För företag ger tidigare erfarenheter att det är viktigt att tydligt definiera ansvarsförhållanden och avtal.</p> <p>Tekniskt kan det vara viktigt att erbjuda data antingen som rådata då denna erbjuder större värdetvårlingspotential än ex. mer aggregerad data alt. följa en vedertagen standard då detta möjliggör att tjänsten kan flyttas mellan olika marknader. Vanliga standarder inom segmentet är SIRI och GTFS.</p>

## 4.1.2

## Icke marknadsdrivna aktörer

Den forskning som bedrivits på Viktoria Swedish ICT har funnit att följande fyra principer avsevärt ökar chansen för att icke marknadsdrivna börjar och fortsätter arbete med API:erna.

*Enkel tillgång.* Det är viktigt att minimera "time-to-first-request", d.v.s. den tid det tar för en utvecklare att genomföra ett första anrop.

*Förstå innehåll.* För att en utvecklare ska kunna göra en bedömning av vad som går att göra eller inspirera till nya tjänster är det viktigt att underlätta förståelsen av innehållet.

*Enkel integration med utvecklingsverktyg.* När väl en utvecklare gjort ett första anrop och förstått vad som är möjligt börjar integrationen med utvecklarens egna verktyg. Där är det viktigt underlätta det arbetet genom använda enkla standarder (ex. REST), format (ex. JSON, XML) och koordinatsystem (ex. WGS84).

*Minimera arbete med att produktionssätta tjänsten och hålla tjänsten i drift.* Efter att tjänsten är utvecklad behöver den gå i produktion för att tillföra användarnytta. Ett sätt att göra detta är tillåta att alla klienter får gå direkt mot API:et, utan mellanlagring. Ett annat sätt är ta fram en enkel mekanism för att prenumerera på förändringar. På sätt behöver minimalt med tid läggas på att identifiera vilka ändringar som skett sen senaste anropet.

## 4.2

## Test

Som i alla IT-utvecklingsprojekt är genomgripande tester av API:erna av stor vikt. I flera fall har vi sett hur API:er med bristande kvalitet snabbt överges av tredjepartsutvecklare.

För att skapa en helhetsupplevelse för en användare måste samma information vara konsekvent över alla kanaler. I tidigare projekt har det visat sig framgångsrikt att testa mot befintliga tjänster i största möjliga grad. Om ex. organisationen idag har en webbaserad tjänst som utför motsvarande funktion som API:et innehåller utgör detta en bra jämförelsegrad för tester.

På sikt är det en stark rekommendation att själva använda de datakällor som organisationen exponerar i sina egenutvecklade tjänster. På sätt stärks tillförlitligheten både internt och externt och eventuella brister och felaktigheter hittas snabbare.

Steg	Beskrivning	Kommentar
5 Lansering	För att skapa kännedom och intresse kring att datakällan släppts är det viktigt att genomföra en genomtänkt lansering.	<p><i>Registrera datakällan i kända kataloger.</i> I dag finns flera katalogtjänster där öppna datakällor listas. Ex. på såna tjänster är <a href="http://opnadata.se">opnadata.se</a>, Svenska API-katalogen och Programmable Web. Dessutom finns en del standardspecifika kataloger – ex. finns en förteckning hos Google över alla öppna GTFS-leverantörer.</p> <p><i>Kontakta potentiella tjänsteutvecklare.</i> Detta kan vara redan kända aktörer, aktörer som medverkat i framtagandeprocessen på något sätt och/eller att meddela i relaterade elektroniska forum (ex. utvecklarcommunities) att källan finns tillgänglig.</p> <p>I vissa fall genomförs även <i>tävlingar</i> för att öka intresset kring datakällorna. Ex. genomfördes West Coast TravelHack 2011 för att bl.a. lansera <a href="http://Trafiklab.se">Trafiklab.se</a> och Stockholms Stad anordnade Open Stockholm Award i samband med att deras satsning på öppna datakällor lanserades. Detta är generellt ett mycket bra sätt att locka aktörer men kräver också en betydande insats från organisationen med datakällan.</p>

6 Utvärdering	Hitta framtida förbättringsområden	<p>Efter att en öppen datakälla lanserats är det viktigt att få feedback från användarna. Rena buggar rapporteras (oftast) in via elektroniska forum men synpunkter av ex. hur krångligt något är – som kan vara mycket viktiga för hur API:erna kommer i användning – yttrar sig sällan i sådana forum.</p> <p>En rekommendation om organisationen är intresserad av att förstå vad tredjepartsutvecklare brottas med är att anordna <i>utvecklarträffar</i>. Där kan man under avslappnade former informera om vad som är på gång och få möjlighet att prata med utvecklare och vad de ser som möjliga förbättringsområden.</p>

## Case: Trafikverket Realtidsinformation Resenär

Under 2012 har Trafikverket i samarbete med Viktoriainstitutet och Samtrafiken tagit fram och publicerat två stycken API:er för tågdata. Syftet har varit att undersöka hur programmatiska gränssnitt mot tredjepartsutvecklare bör designas för att underlätta att nya tjänster skapas men också att förstå målgruppen och dess behov samt att skapa en relation och dialog med dessa.

### Kartlägg nuläge och önskad framtid

#### Intervjuer med Trafikverket

Under februari 2012 genomfördes ett tiotal intervjuer med olika aktörer med kunskap om tåginformation och dess distribution till externa parter vilket gav en fördjupad bild av de organisatoriska risker som fanns i dagens situation. Redan vid projektets start fanns ett antal populära tjänster som användes av en stor användargrupp och som var framtagna av tredjepartsutvecklare. De problem som identifierats var:

- 1) Datatillförseln till tjänsterna skedde via skrapning vilket både innebär risker för de utvecklade tjänsterna (att de går ner vid förändringar av ex. webbsidor) och Trafikverket (att systemen riskerar gå ner vid kraftigt ökad trafik). Skrapningen skedde dels mot olika webbsidor och dels via en icke publik webbtjänst mot Trafikverkets system Orion som kunde nås via JavaScript-anrop.
- 2) Det saknades en relation mellan Trafikverket och tredjepartsutvecklarna. Detta gjorde dels att Trafikverket inte visste vilken information som skulle levereras och dels att man inte kunde "stänga av" en användare som lastade ner information eller gjorde något olagligt med informationen.

Utmaningarna som identifierades var

- 1) I Trafikverkets uppdrag ligger att se till att resenären får den information som denne behöver. I det uppdraget ligger också att inte skapa individanpassade tjänster utan lämna detta till marknadsaktörer. Detta innebär att man inte får störa marknaden genom att själva tillhandahålla tjänster som marknaden kan skapa. Eftersom det finns marknadsaktörer som vidareförädlar information mot tjänsteutvecklare finns en risk om man själva gör detta genom ett mer specifikt API.
- 2) Vilka format och standarder är lämpliga för att tillgängliggöra tåginformationen på?

Under intervjuerna beskrevs också de befintliga strukturerna för att tillhandahålla information för väg. Där fanns idag dels en tjänst – Lastkajen – där företag kunde köpa statisk vägdata som ex. olika vägdata-baser. Dessutom tillhandahölls dynamisk väginformation via standarden DATEX II (olyckor, väglag etc.), dock utan kostnad. Båda tjänsterna krävde skriftliga avtal för tillgång där ansvar, villkor och motprestationer definierades.

Det fanns dock en medvetenhet om att dessa befintliga strukturer kanske inte motsvarade alla tredjepartsutvecklares krav och önskemål.

## Intervju med Tredjepartsutvecklare

Nästa steg var att intervjua de tjänsteutvecklare som trots att det i dagsläget inte fanns något officiellt API ändå tagit fram populära tjänster. Totalt genomförde sex st intervjuer via Skype och personligen. Vi hittade tjänsterna genom att söka igenom olika marknadsplatser för smartphones (iTunes, Google Play och Windows Marketplace) samt genom att fråga de vi pratade med om de kände till ytterligare personer som utvecklade liknande tjänster. Gensvaret var oväntat positivt – applikationerna utvecklades mestadels på fritid då personerna hade ordinära utvecklarjobb dagtid – men trots detta var majoriteten villig att ställa upp på en intervju (kvällstid).

Intervjuerna kretsade kring vissa teman: vad hade man byggt, hur kom det sig att man börjat, hur man hämtar data och hur man skulle vilja se en framtida lösning. Alla tjänster hade ett gemensamt tema – att visa den information som Trafikverket tillhandahåller idag (på ex. trafikverket.se och stationer) i nya sammanhang. Detta kunde vara en mobil webbplats (som inte fanns vid det tillfället då den utvecklingen började), prenumerera på info om ett tåg via SMS samt tåginformation för olika typer av smartphone-plattformar.

Bland orsakerna till varför man börjat utveckla tjänster kunde två teman utkristalliseras. Dels fanns de som varit inbitna tågpendlare och upplevt frustration över att den trafikinformation som fanns inte fungerade på ett bra sätt – att det helt enkelt fattades bra tjänster. Baserat på egenupplevda behov skapade man därför en tjänst som uppfyllde kraven och gjorde den tillgänglig för andra användare via en webbplats eller marknadsplats för smartphone-applikationer. Att sedan tjänsten blev så pass uppskattad av andra användare kom som en överraskning då tjänsten från början bara var tänkt för eget bruk. Bland dessa fanns en ganska djup kunskap kring Trafikverkets information och uttryckte därför önskemål om dataleverans i så "rå" form möjligt för att kunna göra nya spännande lösningar. Om information var för aggregerad upplevdes detta som hindrande i den processen.

Det andra temat var de som kommit att utveckla mot just tågdata som mer av en slump. Utvecklaren ville lära sig utveckla på en ny plattform eller bara hitta ett eget projekt att arbeta med och hittade då några av de API:er som redan fanns tillgängliga (baserade på skrapad data), och eftersom dessa vara enkla att förstå och använda föll helt enkelt valet på dessa. Bland dessa utvecklare var önskemålen annorlunda. Snarare att ha ett enkelt API som var lätt att förstå och använda – alltså liknande de som redan använts – dock med fullständiga informationsmängder (ex. saknades vissa stationer i något API) och med stabil och långsiktig drift.

## Workshop

Resultatet av intervjurundan låg till grund för en workshop som genomfördes i Stockholm den 19 april i Stockholm. På workshopen medverkade nio representanter för Trafikverket, fyra stycken befintliga tredjepartsutvecklare för tågrelaterade tjänster samt två representanter från Samtrafiken och en från Viktoriainstitutet. Syftet med workshopen var att diskutera problemen idag, både från Trafikverkets och tredjepartsutvecklarnas perspektiv.

Förmiddagen inleddes med alla inblandade fick ge sina perspektiv. Projektledningen/ Trafiklab inledde med sina erfarenheter av frågorna och hur Trafiklab använts för att skapa intresse kring trafikdata och skapa relationer med tredjepartsutvecklare. Därefter presenterade Trafikverket sin bild, sitt övergripande uppdrag (med innehållande begränsningar i vad som går att göra) samt de två befintliga gränsytorna för att

vidarebefordra väginformation till externa aktörer (Lastkajen och DATEX II). I nästa del av workshopen fick de inbjudna tredjepartsaktörerna (Teodor Storm, Erik Pettersson, Anders Granåker samt Tommy Andersson som representant från SJ) ge sin bild av situationen. De beskrev vad de gjort för tjänster, varför de börjat och vad de skulle vilja se i en API-lösning. Slutligen pratade SL om sin strategi (se ovan) och hur den gått att genomföra m.h.a. Trafiklab. SL menade det fanns flera smarta och duktiga personer utanför den egna organisationen och därmed blev API:et en strategisk kanal för sprida sin information. Presentationspaket avslutades genom att en konsult med fokus på API:er fick beskriva sin syn på vad ett API är och vad som är viktigt när man designar ett sådant.

På eftermiddagen ägnades tiden åt att diskutera situationen. Utgångsläget var inte att frågan skulle gå att lösa under en så kort tid – men däremot att skapa en bättre gemensam bild över situationen och de behov som olika aktörer upplever.

Övningen hade två moment: 1) *Identifiera de grupper som är intresserade av maskinläsbar tågdata och de behov och betalningsvilja som finns.* Den övningen skedde i blandade grupper med representanter från både Trafikverket och tredjepartsutvecklare. Följande segment/behov identifierades:

#### Marknadsdrivna aktörer

- Kan betala
- Kräver SLA:er
- Kräver support
- Vill ha lösningar liknande Trafikverkets befintliga (Lastkajen och DATEX II)

#### Icke-kommersiella aktörer

- Förväntar sig gratis tillgång
- Låga trösklar
- Klarar sig i större grad utan supporttid, ex. via community, FAQ
- Ex. ensamutvecklare, forskare

2) *Identifiera konsekvenserna av att inte möta behoven samt Trafikverkets utmaningar i att möta behoven.* Övningen var uppdelad så att tredjepartsutvecklarna fick diskutera den första delen av frågan och Trafikverket den andra delen. Risker med dagens lösning

- Risk att dagens gratis-API:er (tillhandahålls av andra tredjepartsutvecklare på ideell basis) försvinner – många användare får ingen information längre
- Önskat format: Allt utom HTML (som i dagens skrapningslösningar), helst JSON/XML
- Mismatch i informationen mellan olika kanaler (TrV:s hemsida, API:er, skyltar)
- Ingen bra dialogmekanism

#### Upplevda problem hos Trafikverket

- Risk att små utvecklare tröttnar – då dessa måste utveckla mycket själva för att komma igång

- Ingen feedback till Trafikverket – vet inte om API:erna är bra
- Tredjepartare får gissa om hur data fungerar – kan bli fel i onödan
- Risk med olika information i olika kanaler p.g.a. olika datakällor
- Vilka format skall användas?

## Genomförbarhet och teknisk lösning

Efter att workshopen hållits och summerats stod projektet vid en beslutspunkt: skulle några pilot-API:er tas fram och hur skulle de i.s.f. se ut? Trafikverket ville utveckla kunskap om området och baserat på workshopens utfall bestämdes att man skulle pröva att ta fram pilot-API:er.

### Arkitektur

Som underliggande datakälla från Trafikverket valdes systemet *Orion* av två skäl: 1) Detta var det system som låg till grund för dagens skrapningslösningar och innehöll därmed den data som befintliga utvecklare hade nytta av 2) systemet var byggt för att enkelt integreras in med andra system (bra ur ett pilotperspektiv) och var dimensionerat för att hantera mycket trafik. Orion skulle därför användas som källa och tillgängliggöras via Trafiklabs API-plattform. Genom Trafiklab kom mycket eftersökt funktionalitet gratis – exempelvis fanns inbyggd användarhantering, cachningsfunktionalitet och möjlighet att styra hur många anrop en viss användare hade rätt till.

### Funktionalitet

Nästa viktiga fråga gällde hur informationen skulle tillhandahållas. Här stod projektet inför ett dilemma med motstridiga krav. Å ena sidan, företrätt av utvecklare drivna av problemlösning och kommersiella aktörer, ville ha så mycket information som möjligt för att sedan själva behandla informationen och göra mer specialiserade tjänster. Å andra sidan var de utvecklare som snarare drevs av lärande, ett komplement till en befintlig tjänst eller att utveckla något på en specifik plattform inte så intresserade av att lägga avsevärt med tid på att förstå och processa om datamängden samt sätta upp en egen serverlösning för att utveckla en tjänst. Dessa ville istället ha ett enkelt "out-of-the-box" API.

Som lösning på problemet skissades på två API:er.

- *TrainExport* motsvarade allt innehåll i Orion som var möjligt att sprida vidare och som kunde filtreras baserat på valda parametrar. Detta innebar också att det gavs tillgång till termer som normalt sett inte en resenär har tillgång till och som kan vara svåra att förstå. Ett exempel på detta är "Teknisk avgångstågtid" som är den faktiska avgångstid ett tåg beräknas lämna stationen (vilken inte syns på realtidskyltar etc.). Ett viktigt krav som TrainExport inte kunde klara var att prenumerera på förändringar. Detta berodde på att datamängden i Orion inte klarade av detta, eftersom det inte fanns någon tidsstämpel som indikerade när en viss post ändrats.
- *TrainInfo* var istället ett enkelt API med stöd för de vanligaste användningsfallen och en delmängd av TrainExport. Idéen med API:et var det inte skulle krävas någon vidare behandling av informationen och att det skulle vara relativt självinstruerande.

## Feedback på specifikationen

Även om specifikationen grundades i intervjuer/workshopen som genomförts och projektets resurser ville projektet säkerställa att lösningen motsvarade utvecklarnas behov och få feedback på tänkt upplägg. Intervjuerna och workshopen hade bara involverat en dryg handfull intressenter och det fanns risk att dessa få aktörer hade snedvridit kravbilden. För detta ändamål lades specifikationen upp på ett öppet diskussionsforum på Google där både tidigare inblandade och alla övriga intresserade bjöds in för att komma med synpunkter.

Bland de synpunkter/önskemål som kom in fanns stöd för olika typer av mer tekniska standarder (inte standarder för tågdata), en tidsstämpel för när data senast var uppdaterat (som inte gick att stödja enl. diskussionen ovan), frågor om innehållet i informationsmängden samt feedback på vilka användningsfall som borde stödjas. Även om det gavs något mindre feedback än förväntat var gruppen också ett bra sätt hålla intressenter informerade om status, ev. problem som projektet stött på och tänkt lansering. På det sättet kunde dialogen hållas levande även under utvecklingsfasen.

## Utveckling, test och driftsättning

Under sommaren och första delen av hösten togs API:erna fram. Den underliggande plattformen (Orion) var byggd för enkelt kunna integreras in i nya tjänster via XML och utvecklingsarbetet skedde på tre ställen i systemets arkitektur:

- 1) De xml-frågor mot Orion som hämtade relevant information togs fram av Trafikverket
- 2) Den specifikation som beskrev hur informationen skulle mappas mellan Trafikverket och Trafiklab togs fram av Samtrafiken
- 3) Baserat på specifikationen genomförde sedan Samtrafikens API-leverantör ApiGee själva mappningen mellan Trafikverket och Samtrafiken.

Tidigare forskning från Viktoriainstitutet och erfarenheter från Trafiklab har visat att API:er som uppfyller fyra kriterier har större chans att komma i användning. Dessa tillämpades i Trafikverkets API:er och yttrade sig på följande sätt:

- 1) *Enkel tillgång.* I Trafiklab går utvecklaren igenom tre steg för att kunna göra ett första anrop:
  - a. Registrera sig. Detta implementerades genom att använda den slimmade registreringsprocessen där man kan använda befintliga konton på Facebook och GitHub (webbplats för programmerare att hantera källkod) eller uppge sin e-postadress och välja ett lösenord.
  - b. Alla anrop måste skicka med en API-nyckel vilken är knuten till en enskild utvecklare (en utvecklare kan dock begära flera nycklar). En nyckel fås genom att användaren skapar ett projekt och väljer vilka API:er som ska ingå.
  - c. Anropa API:et med API-nyckeln. För att underlätta hur själva anropet görs finns både exempelänkar för varje metod och en tutorial som visar hur man enklast löser och en nyckel

- 2) *Förstå innehåll.* I Trafiklab görs detta genom en tydlig dokumentation av API:erna och dess metoder samt en *API-konsol*. Genom API-konsolen kan en utvecklare (eller en icke-utvecklare) utforska innehållet i API:et utan att behöva skriva någon programkod eller ens registrera sig för att få en nyckel. Utöver detta finns ett antal *exempellänkar* som beskriver hur ett anrop bör gå till. Därmed kan användaren bara klistra in dessa exempel i en browser, lägga till sin nyckel och testköra anropet. Det finns även för vissa API:er även en *tutorial* där det i detalj beskrivs hur de vanligaste användningsfallen implementeras i en tjänst och i vilken ordning API-anropen bör köras. Slutligen måste det dessutom finnas en tydlig *dokumentation*, helst fri från facktermer, som på ett tydligt sätt beskriver API:er, metoder och parametrar.
- 3) *Enkel integration med utvecklingsverktyg.* När väl en utvecklare gjort ett första anrop och förstått vad som är möjligt börjar integrationen med utvecklarens egna verktyg. Inom de projekt som författarna varit inblandade i har REST som överföringsprotokoll varit mest önskvärt p.g.a. att underlättar förståelse av API:ets innehåll (man kan klistra in en URL i en browser och testa en funktion), att protokollet är mindre "pratigt" än ex. SOAP vilket lämpar sig väl vid mobila tillämpningar samt att de flesta utvecklingsmiljöer väl stödjer detta protokoll. Vad gäller format brukar JSON och XML vara de mest eftersökte och båda stöds i API:et. Slutligen är även koordinatsystem en viktig komponent i kollektivtrafik-API:er. I princip alla vanliga plattformar har WGS 84 som koordinatsystem och det är därför önskvärt att koordinater representeras i detta system. I Trafikverkets fall var detta dock inte möjligt och därför fanns tydlig information hur konvertering från deras koordinatsystem (SWEREF90) med bl.a. referenser till kodbibliotek baserade på öppen källkod som kunde användas.
- 4) *Minimera arbete med att produktionssätta tjänsten och ha den i drift.* I detta och tidigare arbete har vi sett två vanliga konfigurationer av produktionssatta lösningar.
  - a. I den ena laddar utvecklaren själv hem information från API:et låter sedan klientapplikationerna hämta information från utvecklarens egna server. I det fallet är det bästa alternativet att kunna prenumerera på uppdaterad information. Om ex. ett tåg då blir försenat skickas en signal från API:et till utvecklarens server så att dennes information alltid är uppdaterad. En annan lösning är att göra det enkelt att hämta den information som ändrats sedan en viss tidpunkt. Utvecklaren kan då vid jämna mellanrum (ex. 1 gg/minut) bara få den ändrade informationen och uppdatera informationen på sin sida. Tyvärr var inget av detta möjligt i Trafikverkets API:er eftersom den underliggande datamängden inte stödde detta och förblev därmed en känd brist i lösningen.
  - b. I den andra vanliga konfigurationen går utvecklarens applikationer direkt mot API:et. Detta genererar mer trafik hos API-leverantören men minskar samtidigt risken för felkällor. I en sådan lösning finns ett särskilt behov att ha kontroll över antalet anrop som en utvecklare får göra – men samtidigt ha en smidig process för att enkelt uppgradera användaren. I Trafiklab får alla som registrerar sig 30 antal anrop/minut och 10 000 antal anrop/månad. Detta hjälper till i processen att kontrollera den trafikvolym som API-servern behöver behandla och räcker långt för ett utvecklingsscenario men man kan slå i taket

i en produktionslösning. Genom att skicka in ett webbformulär på Trafiklab och beskriva tjänsten uppgraderas dock utvecklaren på ett enkelt sätt och behöver inte sätta upp en egen server.

En av grundpelarna i arbetet var att samma information som finns på Trafikverkets webbtjänst Läget i Trafiken också skulle finnas i API:erna. Därför genomfördes *testerna* med denna utgångspunkt – ett vanligt scenario inom kollektivtrafiken. Genom att i testerna jämföra de svar som kom från API:erna med de som fanns på webbplatsen kunde man säkerställa att API:erna returnerade rätt information.

## Lansering och utvärdering

I oktober 2012 lanserades API:erna på Trafiklab. Denna aktivitet skedde främst genom att registrerade medlemmar på Trafiklab fick ett mejl om att det lanserats två nya API:er. Dessutom meddelades de utvecklare som deltagit projektet och Sveriges största API-blogg, mashup.se, skrev om lanseringen.

Den sista aktiviteten var en utvärdering. Denna skedde genom intervjuer med 7 st utvecklare, skanning av information i sociala medier (främst Twitter) samt genom individuella möten med utvecklare (ex. vid Trafiklabs meetup den 6 december då Trafiklab bjöd in utvecklare för att informera och diskutera framtida och nyligen genomförda insatser.

Generellt fick API:ert TrainInfo högt betyg. Alla intervjuade utvecklare som testat API:et menade att det var mycket enkelt att förstå och använda och mötte deras behov. TrainExport ansågs dock ha en högre förbättringspotential. Då TrainExport genererar stora mängder information och det idag inte gick att prenumerera eller fråga efter information som ändrats sedan senaste anrop var API:et svårare att använda. Detta identifierades tidigt i projektet men då det kräver förändringar i underliggande system (Orion) är detta en punkt för Trafikverket att ta med sig i framtida arbete.