

Geografiska sökningar och POI

Hitta Jämför Vård: Sökförbättringar 2



Innehållsförteckning

| | |
|--|-----------|
| 1. Bakgrund | 2 |
| 1.1. Introduktion..... | 2 |
| 1.2. Nuvarande stöd | 2 |
| 2. Geocoding | 3 |
| 2.1. Beskrivning..... | 3 |
| 2.2. Användning i HJV..... | 3 |
| 2.3. Leverantörer | 3 |
| 2.3.1. Google | 4 |
| 2.3.2. MapQuest..... | 5 |
| 2.3.3. Microsoft..... | 5 |
| 2.3.4. Prisjämförelse..... | 5 |
| 2.4. Geometrisk data vid uppslag..... | 5 |
| 3. Implementering..... | 7 |
| 3.1. Översikt | 7 |
| 3.2. Söklogik..... | 8 |
| 3.3. Sökgränssnitt..... | 8 |
| 3.4. Automatisk igenkänning av termer..... | 9 |
| 3.5. Val av teknisk strategi | 9 |
| 3.5.1. Strategi 1: Merparten logik på klientsidan (rekommenderas) | 10 |
| 3.5.2. Strategi 2: Merparten logik på serversidan | 12 |
| 3.6. Genomförande i faser..... | 12 |
| 3.6.1. Fas 0: Förberedande aktiviteter | 12 |
| 3.6.2. Fas 1: Geocoding för områden och kommuner..... | 13 |
| 3.6.3. Fas 2: Implementering av POI..... | 14 |
| 3.6.4. Fas 3: Geocoding för regioner och län (aktuellt på längre sikt) | 14 |
| 4. Aktiviteter och estimat | 15 |

1. Bakgrund

1.1. Introduktion

Inom projektet *HJV: Sökförbättringar 2* utreds möjligheterna att utöka stödet för geografiska sökningar genom att implementera POI-funktionalitet. Den här rapporten beskriver föreslagna tillvägagångssätt att förbättra stödet för geografiska sökningar med en utgångspunkt från stödet för geografiska sökningar som finns i HJV idag.

1.2. Nuvarande stöd

HJV har idag ett begränsat stöd för geografiska sökningar. Användare kan söka på tre typer av nyckelord som representerar geografiska områden i Sverige;

- **Län eller region**
Varje mottagning kan ha ett län eller region angivet i HJV. Datat härstammar från HSA där ett värde återfinns för varje mottagning.
Exempel: "Stockholms län"
- **Kommun**
Varje mottagning kan ha en kommun angiven i HJV. Datat härstammar från HSA där ett värde återfinns för varje mottagning.
Exempel: "Stockholm"
- **Områden**
Varje mottagning kan ha en lista med områden angivna i HJV. Det vanligaste scenariot är att en mottagning enbart har ett område definierat även om det är tekniskt möjligt att definiera flera (enbart exempel med ett område har hittats). Datat härstammar från HSA där ett värde återfinns för varje mottagning, vilket förklarar att ingen mottagning med fler än ett värde hittades.
Exempel: ["Kungsholmen"]

Kopplingen mellan mottagning och geografiskt område sker i dagens lösning alltså genom kategorisering; Om en mottagning innehåller värdet "Kungsholmen" i HSA-fältet "Location" kopplas denna till kategorin "kungsholmen" tillsammans med alla andra mottagningar med samma värde.

De geografiska koordinater som finns definierade för varje mottagning används alltså inte när geografiska sökningar mot *län/region*, *kommun* eller *område* utförs.

2. Geocoding

2.1. Beskrivning

Möjligheterna till geografiska sökningar i HJV är i dagens lösning begränsade till fördefinierade områden, kommuner och län. Detta begränsar användarens möjlighet att hitta rätt resultat och innebär även ett krav på redaktionellt underhåll av mottagningars data i HSA.

Ett problem med dagens lösning är att HJV inte klarar av att koppla namn på många vanliga platser och områden till geografiska punkter eller figurer. Processen att översätta ett namn på en plats till en uppsättning geografiska koordinater kallas för *Geocoding* och är en central del i geografiska söktjänster.

2.2. Användning i HJV

Genom implementering av *Geocoding* i HJV kan sökningar på godtyckliga platser, adresser eller områden översättas till geografiska koordinater som sedan kan användas för att konstruera geografiskt medvetna sökkriterier eller sorteringsmekanismer.

2.3. Leverantörer

En förutsättning för att implementera *Geocoding* i någon applikation är att ha tillgång till en databas över geografisk data. Det rekommenderade tillvägagångssättet är att utnyttja en extern leverantör till detta. Sådana tjänster är vanlig och tillgången på leverantörer är stor. Prismodellerna kan variera men baseras i regel på antalet uppslag per tidsenhet, där vissa leverantörer även erbjuder ett begränsat antal uppslag utan kostnad.

Ett urval av leverantörer är; *SmartyStreets, MelissaData, ServiceObjects, Geocoder.us, IntelligentSearch, Strikelron, Google, IntelligentSearch, TAMU, Bing, Yahoo, Cdyne, Gisgraphy, Nominatim, OpenCage, Placefinder, MapQuest, Loqate, Data Science Toolkit, Geocodio, Mapbox och Here.*

Två av dessa leverantörer har undersökts mer än andra i samband med den här rapporten; Google och MapQuest.

Vid val av leverantör bör vissa saker tas i åtanke, dels för att förenkla implementering men även säkra lösningens framtid;

- Standardmässigt API
- Välskriven och utförlig dokumentation
- Finansiellt stabil leverantör som förväntas finnas kvar i framtiden (att göra om implementeringen vid eventuell nedstängning av tjänsten innebär kostnader)

Valet av leverantör kan även påverkas av vilken teknisk lösning som väljs för att realisera *Geocoding* i HJV. Beroende på huruvida uppslagen väljs att implementeras inom KnockoutJS-applikationen i användarens webbläsare eller inom HJV API kan vissa prismodeller eller utbud av tjänster bli mer relevanta vid valet av leverantör.

2.3.1. Google

Google tillhandahåller geocoding-uppslag genom *Google Geocoding API*. Vid implementering av autocomplete-funktionalitet används dock med fördel i stället *Google Places API* som ger tillgång till samma data men genom ett mer autocomplete-vänligt API. *Google Places API* utnyttjas enklast genom att inkludera ett tillägg i det befintliga JavaScript-bibliotek som i dag används av HJV för att visa kartor från Google Maps.

2.3.1.1. Prismodell

Google Maps API tillämpar en prismodell där användning som underskrider 25 000 kartladdningar per dag förblir gratis. Värt att notera är dock att Google tillämpar speciella regler för webbsidor som kan anses falla inom ramen för "offentlighetens intresse", så det är möjligt att restriktionen på 25 000 kartladdningar inte är aktuella för HJV;

Note: Applications deemed in the public interest (as determined by Google at its discretion) are not subject to these usage limits. For example, a disaster relief map is not subject to the usage limits even if it has been developed and/or is hosted by a commercial entity. In addition, qualifying nonprofit organizations who demonstrate a need for increased geocoding limits or internal usage of the Google Maps API may apply for a Google Maps API for Work license through the Google Earth Outreach grants program.

<https://developers.google.com/maps/documentation/javascript/usage>

Vid användning av *Google Places API* som ett tillägg till JavaScript-biblioteket för Google Maps täcks användningen av *Google Places API* inom samma begränsningar som *Google Maps API*.

Om *Google Places API* i stället används enskilt är begränsningarna annorlunda; Maximalt 1 000 frågor per dygn. Dock tillämpas begränsningen då i stället på IP-basis i stället för per applikation.

Exakt vilken av dessa restriktioner som är aktuella vid en implementering mot HJV är inte helt klar och beror på hur autocomplete-funktionaliteten implementeras. Ingen av gränserna för gratis bruk bör dock utgöra en begränsning för lösningen.

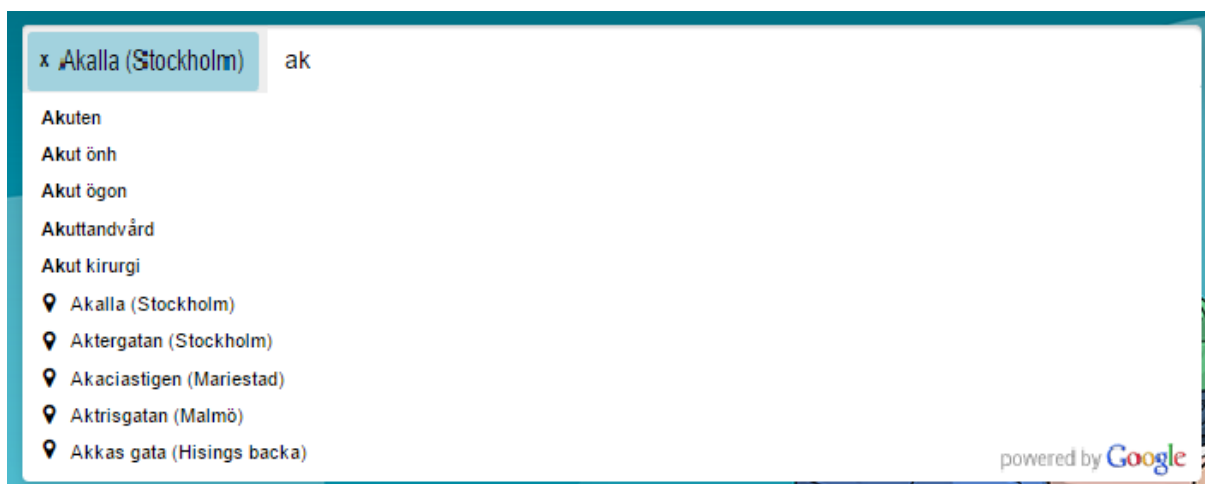
2.3.1.2. Begränsningar

Google tillhandahåller generösa användningsgränser, men ställer samtidigt krav på att Google tillskrivs i alla sammanhang där deras tjänster används. Vid användning av Google Maps är visningen av den inkluderade Google-logotypen obligatorisk, oavsett om tjänsten används gratis eller genom någon av deras betalningsmodeller.

Vid implementering av Geocoding innebär detta att en "Powered by Google"-logotyp behöver läggas till när autocomplete-vyn visas på startsidan för HJV. När en sökning har genomförts behöver logotypen inte längre visas, eftersom en motsvarande logotyp då är synlig i kartvyn.

Note: When you display predictions from Place Autocomplete, you must include the 'powered by Google' logo. This logo is included in the results list by default, as shown in the above example.

<https://developers.google.com/maps/documentation/javascript/places-autocomplete>



Figur 1 – Autocomplete-vy med Google-logotyp

Vidare innebär en implementering av geocoding-autocomplete baserad på Google Places API begränsningar för vilken kartleverantör som kan användas;

Hint: You can use Place Autocomplete even without a map. If you do show a map, it must be a Google map.

2.3.2. MapQuest

Till skillnad från Google har MapQuest inga krav på logotyp i samband med användning av deras tjänster. Däremot är deras tjänst märkbart dyrare än motsvarigheten från Google. Vid implementering av Geocoding baserat på tjänster från MapQuest kommer sannolikt deras Nominatim API att utnyttjas; <http://open.mapquestapi.com/nominatim/>

2.3.3. Microsoft

Bing Maps har liknande villkor som Google Maps; Logotyper och marknadsföring kan förekomma. Kvoterna för gratis användning är lägre än hos Google Maps, men det framkommer även bland deras villkor att vissa organisationer som klassificeras som non-profit (exakt definition bör undersökas närmare om det är aktuellt) inte omfattas av dessa begränsningar på samma sätt. Microsoft tillhandahåller tillgång till ett API för att göra geocoding-uppslag som en del av Bing Maps.

2.3.4. Prisjämförelse

| Leverantör | Fri kvot | Kostnad per uppslag |
|------------|-------------------------------------|--|
| Google | 25 000 per dag (~750 000 per månad) | \$0.0005 |
| MapQuest | 15 000 per månad | \$0.0066-\$0.0015 beroende på prisplan |
| Bing | 125 000 per år | |

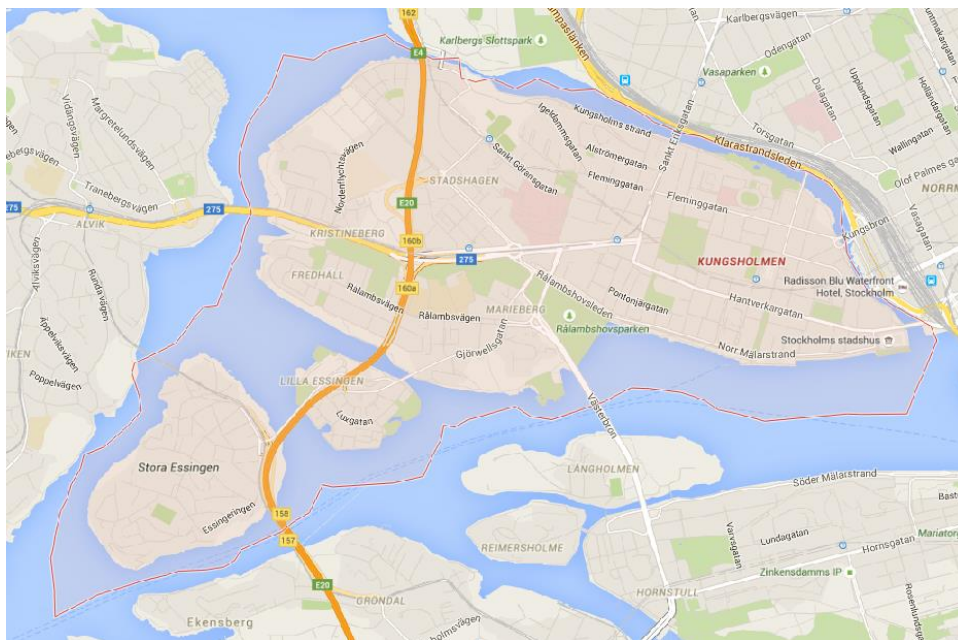
Figur 2 – Prisjämförelse

2.4. Geometrisk data vid uppslag

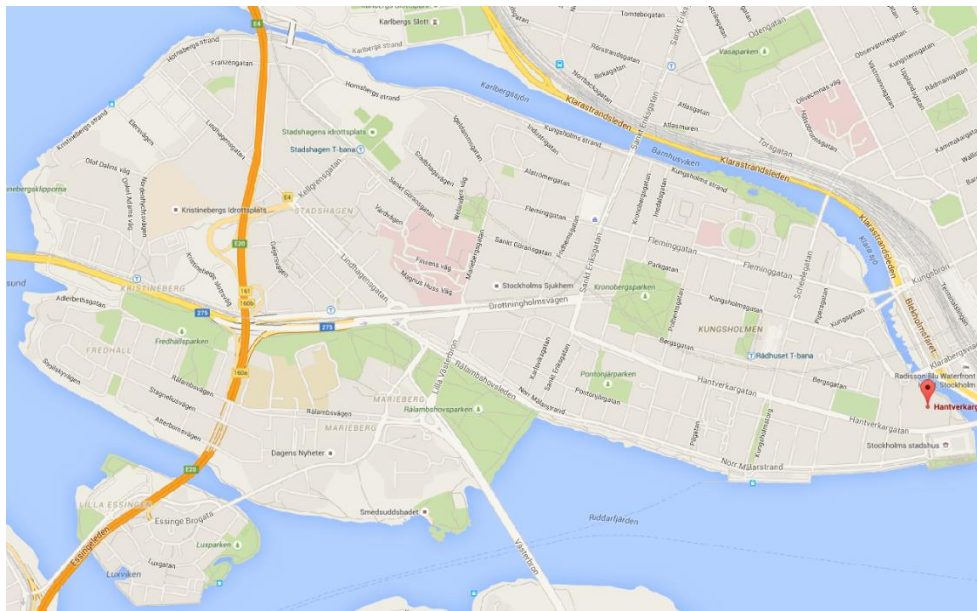
Resultatet från ett uppslag kan variera beroende på vilken typ av geografisk data som sökningen resulterar i. Det exakta utbudet av geografisk data kan skilja beroende på leverantör, men är sannolikt

till stora delar jämförbart. Inom rapporten används *Google Geocoding API*, *Google Maps API* samt *Google Places API* som referens.

Gemensamt för alla typer av träffresultat är att de kan representeras som en punkt på en karta. Vissa typer av resultat kan innehålla ytterligare information i form av ett antal geografiska punkter som tillsammans utgör ett geografiskt område, exempelvis "Kungsholmen" (se nedan).



Figur 3 – Geografiskt område som representerar sökresultatet "Kungsholmen"



Figur 4 – Geografisk punkt som representerar sökresultatet "Hantverkargatan 2D"

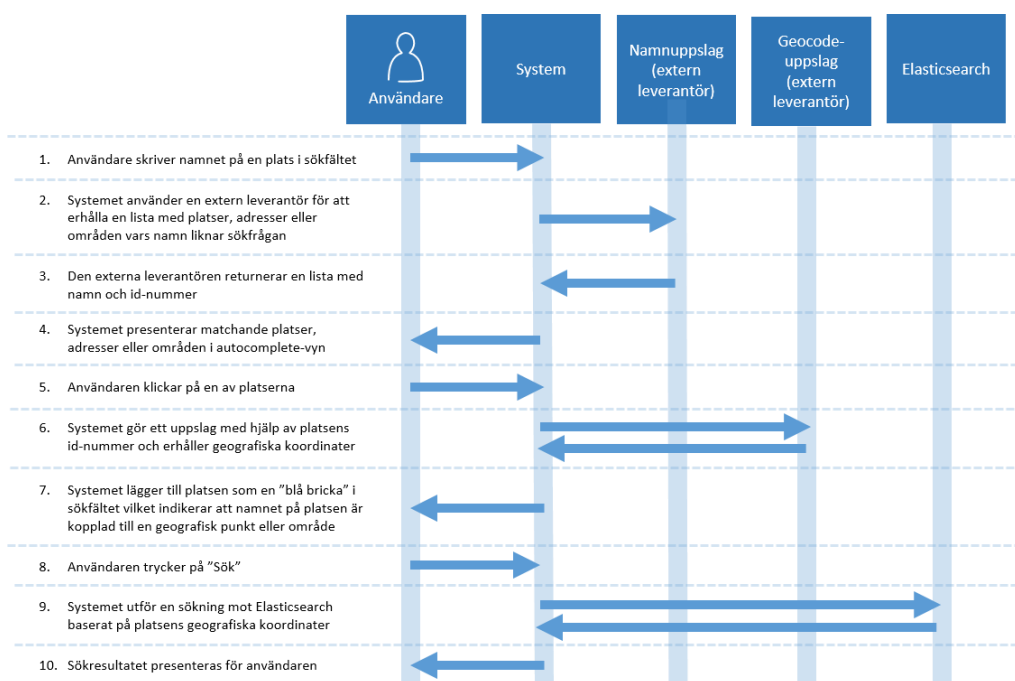
3. Implementering

3.1. Översikt

En realisering av geografiska sökningar i HJV omfattar följande två aktiviteter som kräver kommunikation med en extern geocoding-leverantör;

- **Namnuppslag**
En sökning efter platser, adresser eller områden med namn som liknar sökfrågan. Resultatet används för att presentera en lista med klickbara sökförslag för användaren (autocomplete).
- **Geocode-uppslag**
Omvandling av ett exakt namn (eller unikt ID-nummer) på en plats, adress eller område till geografiska koordinater. Används för att konstruera en geografiskt medveten sökfråga till Elasticsearch.

Dessa två uppslag kan endera göras inom ett och samma API-anrop eller genom två separata anrop. Utformning av tjänster och API kan skilja mellan leverantörer. Nedan tecknas dessa två interaktioner separat.



Figur 5 – Sekuensdiagram över interaktioner mellan användare, system och geocode-leverantör

En övergång till att använda varje mottagnings geografiska koordinater i högre grad ställer större krav på denna datas riktighet. Det kan därför bli nödvändigt för SLL att bedriva ett redaktionellt arbete där felaktig data identifieras och rättas till.

3.2. Söklogik

När namnuppslag efter platser, adresser eller områden genomförs kan två typer av data returneras;

- **Områden (område, stadsdel, kommun, län)**
Representeras geografiskt av ett område som begränsas mellan ett antal punkter
- **Punkter (adress, POI)**
Representeras geografiskt av koordinater

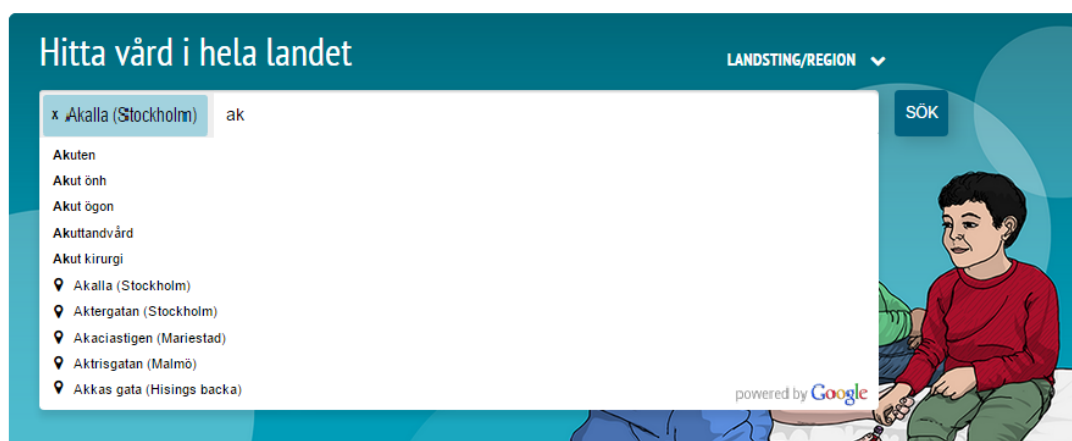
Den geografiska representationen av denna data används sedan för att konstruera en sökfråga som kan tolkas och utföras av Elasticsearch. Frågan kan formuleras på olika sätt och således returnera flera olika typer av resultat. Vilken söklogik som är önskvärd inom HJV behöver därför bestämmas i förväg.

En föreslagen söklogik är enligt följande;

- **Områden som ingår i sökfrågan** utgör *filter* och påverkar träffbilden på samma sätt som geografiska söktermer så som "Kungsholmen" påverkar träffbilden i HJV idag; Enbart mottagningar som ligger inom området returneras. Om mer än ett områden anges tillämpas OR-logik mellan områdena inbördes.
- **Punkter som ingår i sökfrågan** påverkar *relevansen* hos träffresultaten. Att en sökfråga innehåller en punkt (adress, POI) innebär alltså inte att antalet träffar blir färre, utan påverkar enbart rangordningen i resultatlistan. Mottagningar som har ett kort avstånd till den angivna punkten rankas högre än mottagningar som ligger längre bort. Om mer än en punkt anges kan någon av följande strategier tillämpas;
 - Mottagningar som ligger nära någon av punkterna rankas högre än mottagningar som ligger längre bort
 - Mottagningar som ligger nära punkternas geografiska mittpunkt rankas högre än mottagningar som ligger längre bort

3.3. Sökgränssnitt

Gränssnittet för sökfunktionen i HJV behöver förändras för att tillgängliggöra den nya funktionaliteten tillgänglig. I sökfältet behöver platser, adresser och områden markeras på samma eller liknande vis som kategorier markeras blåa i dagens lösning. För att presentera geografiska sökförslag behöver även autocomplete-vyn anpassas.



Figur 6 - Exempel på utökad autocomplete-vy

Eftersom de geografiska sökförslagen har en annan datakälla än förslagen på kategorier/vård behöver de presenteras separat. Det finns ingen enkel metod för att jämföra relevans hos dessa med varandra. För att visa all information i autocomplete-vyn kan antalet sökförslag som visas även behöva justeras - alternativt om sökförslagen på mottagningar döljs.

3.4. Automatisk igenkänning av termer

I dagens lösning implementeras *automatisk igenkänning* för kategorier och nyckelord i sökfältet. Automatisk igenkänning innebär att applikationen självmant ersätter termer i sökfältet som har exakta motsvarigheter bland sökförslagen.

Det finns scenarion när sådan funktionalitet ställer till problem; Exempelvis när det finns sökförslag som har "frivilliga komponenter" så som i en adress. "Hantverkargatan" och "Hantverkargatan 4D" är båda giltiga geografiska sökförslag. Det fundamentala problemet med *automatisk igenkänning* blir synligt i situationer där användaren ämnar skriva in den fullständiga adressen "Hantverkargatan 4D" i sökfältet. En *automatisk igenkänning* kommer då göras redan efter "Hantverkargatan" vilket förhindrar användaren att ange den fullständiga adressen.

Det är därför rekommenderat att undvika den här typen av funktionalitet eftersom det är för komplext att göra den så pass intelligent så att funktionen gör mer nytta än skada.

3.5. Val av teknisk strategi

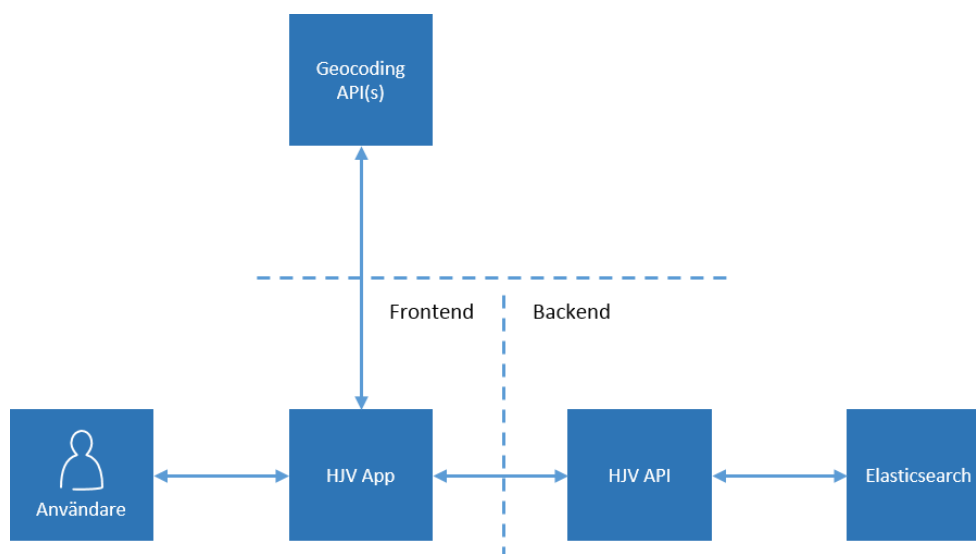
Den tekniska lösningen för att implementera Geocoding i HJV kan tillämpas efter två olika strategier där merparten logik endera utförs på klientsidan eller på serversidan.

I dokumentationen för *Google Maps API* finns en övergripande beskrivning av dessa två modeller (<https://developers.google.com/maps/articles/geocodestrat>) samt rekommendationer kring vilken strategi som lämpar sig bäst i olika scenarion.

Den här rapporten kommer fram till en klientbaserad lösning är mest fördelaktig. Applicerar man Googles rekommendationer på de användningsfall som är aktuella för HJV kommer man fram till samma slutsats. Det finns dock fördelar med en server-baserad lösning vilket tas upp längre fram.

3.5.1. Strategi 1: Merparten logik på klientsidan (rekommenderas)

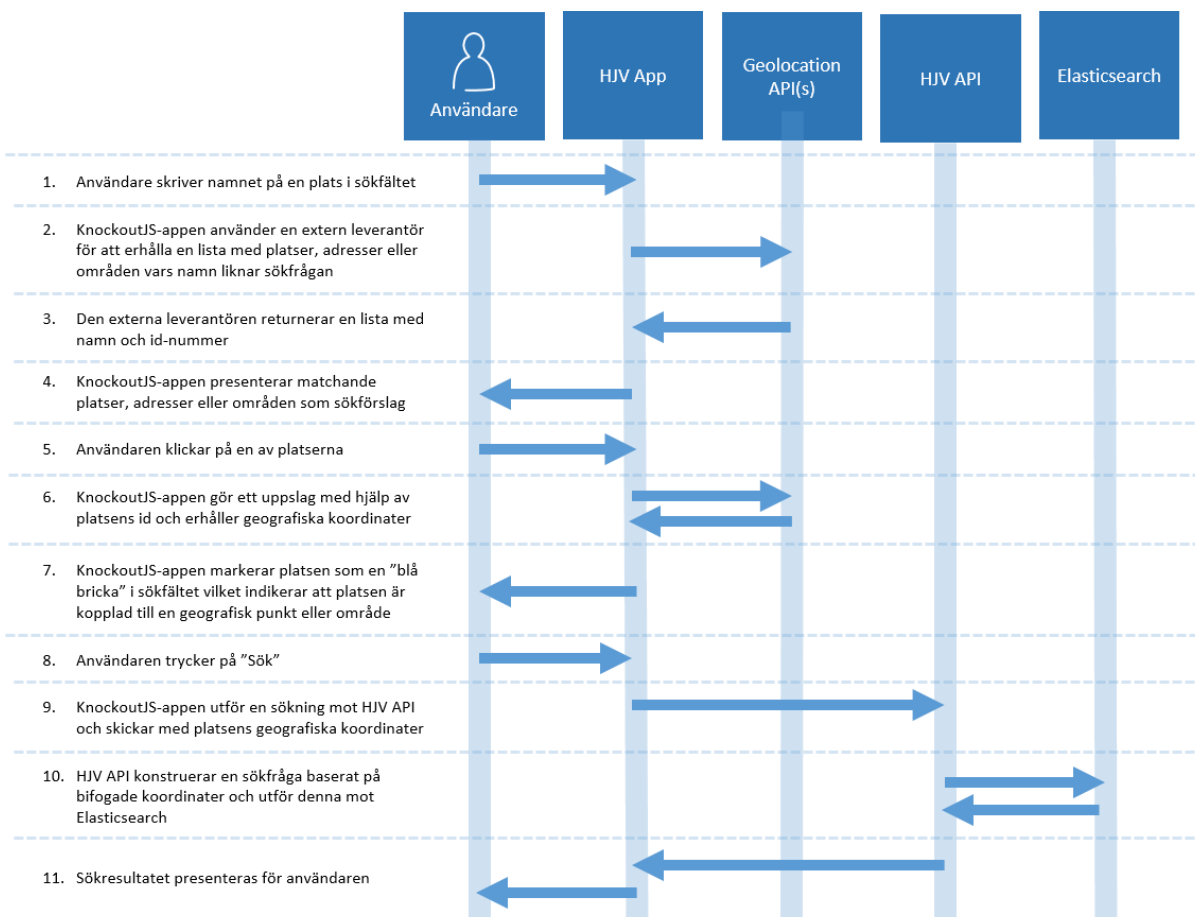
Att implementera merparten logik på klientsidan innebär att den befintliga KnockoutJS-applikationen (benämnd "HJV App") utökas till att i en högre grad ansvara för att konstruera sökfrågor.



Figur 7 – Illustration över kommunikationen med vald geocoding-leverantör när merparten logik utförs på klientsidan

Innan en sökning skickas vidare till HJV API ansvarar HJV App vid denna strategi för att;

- 1. Tolka användarens söktext och identifiera eventuell geografisk kontext**
Användarens söktext analyseras genom att anropa en extern geocoding-leverantör. Söktexten kan sedan ersättas med en geografisk medveten sökterm automatiskt genom igenkänning likt dagens lösning eller först efter att användare väljer ett geografiskt sökförslag i autocomplete-vyn (rekommenderad lösning).
- 2. Konstruera en sökfråga till HJV API**
En sökfråga till HJV API konstrueras sedan baserat på valda geografiska söktermer. Implementeringen av HJV API behöver därför förändras till att kunna ta emot geografiska punkter eller områden som sökkriterier. Till HJV API behöver alltså enbart de uppslagna koordinaterna som representerar valda punkter och områden skickas med.



Figur 8 – Sekvensdiagram över interaktioner mellan delsystem när merparten logik implementeras på klientsidan

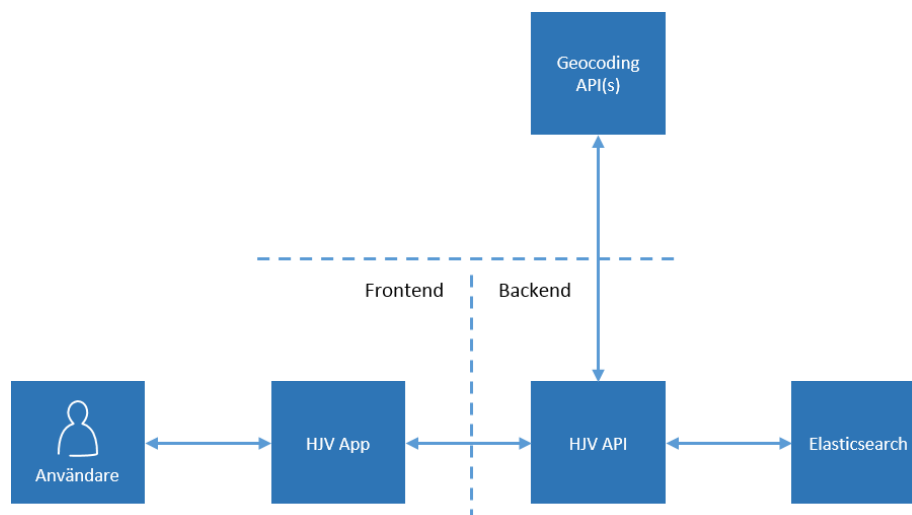
En fördel med att bygga efter en klientbaserad strategi är att belastningen på applikationen HJV och även leverantören av geodata blir distribuerad i stället för centraliserad. Det innebär lägre belastning på infrastruktur som drifvar HJV. Det kan även påverka kostnaden för geocoding-uppslag beroende på vilken leverantör som väljs. En del leverantörer begränsar användningen av deras tjänster på IP-basis i stället för per applikation (exempelvis Google i vissa fall) medan andra utgår från applikationens totala belastning (MapQuest).

En klientbaserad strategi innebär även att varje konsument fritt kan välja geocoding-leverantör, eftersom implementeringen i HJV API kommer att vara generisk och oberoende av geocoding-leverantör.

3.5.2. Strategi 2: Merparten logik på serversidan

Det är också möjligt att lägga merparten logik i HJV API. De konkreta fördelarna med detta är att uppgifterna för HJV App blir simplare. HJV App, och eventuella framtida konsumenter/klienter, kan kommunicera direkt mot HJV API utan att integrera mot externa tjänster för att erbjuda geografiska sökningar.

I praktiken innebär en sådan lösning att HJV API agerar som en proxy mellan HJV App och Geocoding API. Gränssnittet mot HJV App förenklas till priset av att komplexiteten i lösningen som helhet ökar. Inom HJV API behöver i en sådan situation en eller två ytterligare tjänster implementeras för att tillhandahålla geografiska uppslag.



Figur 9 - Illustration över kommunikationen med vald geocoding-leverantör när merparten logik utförs på serversidan

En serverbaserad strategi medför att samtliga frågor till geocode-leverantören kommer att härstamma från samma maskin. Det innebär att kostnaden för geocode-uppslag centraliseras till HJV API. Eventuella framtida konsumenter slipper att själva upphandla en leverantör av geodata.

I vissa fall kan en sådan lösning påverka vilka prismodeller från respektive leverantör som är intressanta. Eventuella gratis-tjänster med begränsning i användning baserat på IP som vissa leverantörer erbjuder kanske inte kan användas eftersom trafikvoten från HJV API riskerar att överskrida eventuella gränser.

3.6. Genomförande i faser

3.6.1. Fas 0: Förberedande aktiviteter

För att underlätta implementering av Geocoding i HJV rekommenderas ett initialt fokus på omskrivning av utvalda delar. Först bör *Minska resursanvändning och öka prestanda vid sök* påbörjas följt av en successiv anmaning av *Migrering mot lager-orienterad arkitektur* beskrivna i rapporten *Förberedelser inför vidareutveckling*.

3.6.2. Fas 1: Geocoding för områden och kommuner

Inom HJV finns en stor komplexitet kring hantering av kategorier och verksamhetskoder. En del verksamhetskoder används enbart av mottagningar i vissa län eller regioner vilket innebär att de delar av söklogiken som berör verksamhetskoder behöver presenteras och utföras olika baserat på vilket län användaren ämnar söka i.

Den här logiken är utvecklad ovanpå den nuvarande lösningen där kategorier snarare än koordinater används för att genomföra geografiska sökningar. Att ersätta den nuvarande implementeringen av sökningar på regioner och län med en koordinatbaserad lösning blir därför ett omfattande arbete. I stället föreslås det att nuvarande lösning för sökningar på län och regioner bevaras.

Inom fas 1 läggs i stället fokus på att ersätta den nuvarande implementeringen av *områden*. I dagsläget kategoriseras mottagningar inom områden baserat på data från HSA. Denna lösning kan med fördel ersättas med en koordinatbaserad indelning som uppnås efter implementering av Geocoding med en extern leverantör.

3.6.2.1. Preliminär omfattning

Fasen förväntas omfatta följande områden;

- Bevaring av nuvarande söklogik kring län och region
- Ersätt söklogik kring områden med Geocoding-baserad logik
- Bygg om KnockoutJS-applikation till att;
 - Integrera mot extern Geocoding-leverantör
 - Hantera intern modellering, lagring och hantering av geodata
 - Presentera geodata i sökfält, autocomplete och eventuellt karta
 - Refaktorisering av nödvändiga delar
- Bygg om API till att;
 - Stöd för att filtrera resultat inom begränsade områden
 - Refaktorisering av nödvändiga delar

3.6.2.2. Begränsningar

Sökförslag på län eller regioner kommer att presenteras tillsammans med förslag på kategorier och verksamhetskoder likt dagens lösning. Lösningen kommer förmodligen att resultera i att två separata listor med sökförslag presenteras; En lista med sökförslag på kategorier, län eller regioner. En lista med sökförslag på områden.

Funktionen *automatisk igenkänning av sökord* likt vad som används i dag kommer inte att implementeras, eftersom en sådan innebär begränsningar i vissa situationer (beskrivs under rubriken *Automatisk igenkänning av termer*).

3.6.2.3. Förväntat resultat

Efter fasens genomförande förväntas följande;

- Bibehållen funktionalitet kring län- och regionspecifika verksamhetskoder
- Möjlighet att söka på godtyckliga geografiska områden och enbart få resultat inom dessa
- Presentation och valmöjlighet av sökförslag på områden i autocomplete-vyn
- Möjlighet att lägga till och ta bort områden från sökfältet

3.6.3. Fas 2: Implementering av POI

I en andra etapp byggs lösningen ut till att även möjliggöra sökningar utifrån geografiska punkter, här benämnt *POI*. *POI* skiljer fundamentalt från områden i och med att de representeras av en punkt i stället för ett begränsat område. Exempel på sådana punkter kan vara "Sveavägen 59, Stockholm" eller "Storkyrkan, Stockholm".

Sökningar baserat på punkter förutsätter att en logik har bestämts i förväg som definierar hur sökfrågan ska utformas. Logiken kan exempelvis utformas till att enbart hämta resultat som ligger inom en bestämd radie till punkten. Alternativt kan resultaten sorteras baserat på avståndet till den givna punkten och därmed inte innebära något *begränsande* sökkriterium utan enbart påverka presentationen. En beskrivning av lämplig logik finns under rubriken *Söklogik*.

Inom *fas 2* kommer även den nuvarande implementeringen av *Nära mig* att skrivas om till att följa samma struktur som *POI*. Samma sökstrategi som väljs för *POI* kommer därmed att användas för *Nära mig*.

3.6.3.1. Preliminär omfattning

- Val av söklogik
- Implementering i HJV App till stora delar baserat på lösningen för *områden*
- Implementering i HJV API beroende på val av sökstrategi. Eventuellt införande av mer flexibla val av sortering.
- Omskrivning av *Nära mig* till att använda samma strukturer som *POI*
- Refaktorisering av utvalda delar

3.6.3.2. Förväntat resultat

Efter fasens genomförande förväntas följande;

- Möjlighet att söka på godtyckliga *POI* och få resultat i närheten av dessa

3.6.4. Fas 3: Geocoding för regioner och län (aktuellt på längre sikt)

I ett sista skede eftersträvas att förenkla implementeringen av HJV API till att hantera alla typer av geografiska sökningar på samma sätt. En sökning på "Kungsholmen" ska inom HJV API gå till på samma sätt som en sökning på "Skåne län".

Skälen bakom denna målsättning är att förenkla strukturen och lösningsarkitekturen inom HJV API. För att genomföra en sådan förenkling krävs dock att andra delar av applikationen modelleras om. Den nuvarande lösningen där viss söklogik enbart är tillgänglig inom vissa län behöver byggas om efter en modell där söklogiken är homogen oberoende av valda län eller regioner.

Problemet härstammar i att kopplingarna mellan mottagningar och vissa verksamhetskoder är ojämn över landet. Inom vissa län har det uppförts mer detaljerade verksamhetskoder som används för viss typ av vård, som i andra län inte används över huvud taget, trots att vårdutbudet är likartat.

Den underliggande datamängden som kan användas för kategorisering av vård skiljer därmed mellan landsting. Det ligger en utmaning i att modellera och presentera detta för användaren utan att skapa otydlighet. Den nuvarande lösningen har vissa brister i sin presentation mot användaren samtidigt som den bakomliggande modelleringen skapar komplexitet i lösningen. Om en bättre lösning på detta problem kan hittas finns det stora möjligheter att förenkla lösningsarkitekturen inom HJV API.

4. Aktiviteter och estimat

Följande estimat är baserad på en uppskattning kring vilka delar områden inom HJV App och HJV API som kommer att behöva vidareutvecklas. För mer detaljerade estimat behöver individuella utvecklingsuppgifter identifieras och estimeras individuellt.

Estimaten förutsätter att viss refaktorisering genomförs i förväg, beskrivet inom Fas 0.

Den tredje fasen exkluderas från estimaten då dess omfattning behöver utredas närmare innan eventuell estimering kan ske.

| | Utveckling frontend | Utveckling backend | Test | Projektledning |
|-----------|---------------------|--------------------|------|----------------|
| Fas 1 & 2 | 300 h | 200 h | 75 h | 75 h |



About Capgemini

With almost 145,000 people in over 40 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2014 global revenues of EUR 10.573 billion.

Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want.

A deeply multicultural organization, Capgemini has developed its own way of working, the [Collaborative Business Experience™](#), and draws on [Rightshore®](#), its worldwide delivery model.



Name Name
Function
Function
Phone:00 00 00 00 00
email@capgemini.com

Name Name
Function
Function
Phone:00 00 00 00 00
email@capgemini.com

Name Name
Function
Function
Phone:00 00 00 00 00
email@capgemini.com

Name Name
Function
Function
Phone:00 00 00 00 00
email@capgemini.com

www.capgemini.com



The information contained in this document is proprietary.
Copyright © 2015 Capgemini. All rights reserved.
Rightshore® is a trademark belonging to Capgemini.