

Publicering av geografiska data på webben

- En utvärdering av programsystem med fokus på öppen källkod

Rickard Ingesson & Gabriella Olsson

Civilingenjörsutbildningen i Lantmäteri

Lunds Tekniska Högskola

Institutionen för Naturgeografi och Ekosystemvetenskap

Lunds Universitet





LUNDS UNIVERSITET
Lunds Tekniska Högskola

Publicering av geografiska data på webben

- En utvärdering av programsystem med fokus på öppen källkod

EXTM05 Masteruppsats, 30 hp

Civilingenjörsutbildningen i Lantmäteri

Rickard Ingesson & Gabriella Olsson

Handledare:

Lars Harrie

Institutionen för Naturgeografi och Ekosystemvetenskap

juni 12, 2019

Opponent: Isabelle Andersson

Examinator: Andreas Persson

Copyright © Rickard Ingesson & Gabriella Olsson, LTH

Institutionen för Naturgeografi och Ekosystemvetenskap

Lunds universitet

Sölvegatan 12

223 62 Lund

Telefon: 046-222 30 30

Fax: 046-222 03 21

Hemsida: <http://www.nateko.lu.se>

Examensarbete i geografisk informationsteknik nr 27

Tryckt av E-tryck, E-huset, 2019

Sammanfattning

I de flesta samhällsprojekt finns det ett behov att visualisera geografiska data i en karta och i de flesta fall görs detta i en interaktiv karta på en dator, läsplatta eller mobil. För att kunna publicera sina geografiska data på webben behövs ett antal komponenter. Dessa komponenter köps i många fall med en användarlicens från ett företag som tillhandahåller kunden support och annan information om hur produkten kan användas. Ett alternativ till dessa är produkter som är utvecklade med öppen källkod som oftast distribueras helt eller delvis gratis till användaren. Dessa produkter kan i jämförelse med kommersiella produkter innebära ett större ansvar för användaren i avseende på support och utvecklingstid.

Det finns många för- och nackdelar med användningen av program med öppen källkod. I den här rapporten förklaras begreppet öppen källkod samt vilka licenstyper som är viktiga att känna till för att kunna utveckla egna system inom GIS. Den här rapporten syftar även till att utvärdera användningen av öppen källkod som ett alternativ för kommersiella produkter i avseende publicering av geografiska data på webben. Det finns även ett behov av att utvärdera processen från data till färdig produkt då både många kommuner och företag hanterar och producerar stora mängder geografiska data. Att kunna visualisera dessa effektivt är en viktig aspekt, där det finns mycket tid att spara om det finns möjlighet för effektivisering.

Rapporten består av tre huvuddelar, en teoretisk bakgrund för problemområdet, en intervjustudie för publiceringsprocessen och synen kring öppen källkod för ett antal utvalda kommuner och företag. Tredje delen av fallstudien redovisar en avslutande implementation där den egenimplementerad lösningen, byggd med öppen källkod, jämförs mot ett kommersiellt program, Carto.

Resultatet från studien visar att en lösning med öppen källkod för publicering av geografiska data är ett klart alternativ till lösningar av proprietär programvara. Däremot finns det en del betydande frågor som ska beaktas innan ett sådant system byggs, helt eller delvis, av komponenter med öppen källkod. I intervjustudien framkom att, både de som arbetar med öppen källkod eller personer som arbetar med kommersiella produkter, är väldigt positiva till den generella användningen av öppen källkod. Skillnader i mängden öppen källkod i organisationen har främst att göra med den generella inställningen till att arbeta med öppen källkod och om det finns ett stöd eller krav från exempelvis ledningen eller inte. En annan viktig aspekt är organisationens storlek och medarbetarnas kompetens, då utveckling med öppen källkod är mer tidskrävande vid uppstart än kommersiella produkter. System med öppen källkod tenderar också att mer bli personberoende och känsliga för personalomsättningar i organisationen. En stor fördel med system av öppen källkod är dock att de blir mer flexibla i designen och funktionaliteten samt att utvecklaren har full kontroll över systemet.

Sammantaget av samtliga tre fallstudier har påvisat att möjligheter för utveckling med öppen källkod är möjlig om rätt kompetens, tid och satsning finns. En intressant aspekt är att många kommersiella program i grunden bygger på bibliotek av öppen källkod vilket gör att även kommersiella företag som utvecklar proprietär programvara har vinning av vidareutvecklingen av öppen källkod.

Förord

Denna rapport är resultatet av examensarbete i geografisk informationsteknik för utbildningen civilingenjör inom lantmäteri på Lunds Tekniska Högskola. Examensarbetet är skrivet och publicerat för institutionen på Naturgeografi och Ekosystemvetenskap på Lunds universitet med ett samarbete med BIM- och GIS-avdelningen på Tyréns i Malmö.

Vi vill tacka vår handledare på Lunds universitet, Lars Harrie, för ett stort stöd, god återkoppling och noggrannhet genom hela examensarbetsprocessen

Vi vill även tacka vår handledare Måns Andersson på Tyréns för ett stort engagemang, för all hjälp under processens gång samt för att du orkat korrekturläsa rapporten ett flertal gånger. Ett tack ska också riktas till Pär Hagberg och Peter Alstorp för idén till examensarbetet och för intresset och engagemanget för vårt examensarbete. Ett tack går även till resterande medarbetare på BIM/GIS-avdelningen på Tyréns i Malmö för en trevlig tid och för att ni tog emot oss väl.

Vi vill även tacka Erik Ravhed T-kartor, Karl-Magnus Jönsson Kristianstad kommun, William Walker Ystad kommun och Ulf Minör Malmö stad för att ni ställde upp på intervjuer och gav oss nyttiga tankar, idéer och förslag till vårt examensarbete.

Lund 2019-06-12

Rickard Ingesson & Gabriella Olsson

Förkortningar

API	Application Programming Interface <i>En specifikation och ett gränssnitt mellan ett bibliotek och en programvara</i>
BIM	Byggnadsinformationsmodellering <i>Modellering i 3D för visualisering av exempelvis byggnader samt samverkan och modellering av information som kopplas till byggnadsmodellerna</i>
BSD	Berkeley Software Distribution <i>En av de vanligaste licenserna för öppen källkod</i>
CC	Creative Commons <i>En licens för text och bilder</i>
CSS	Cascading Style Sheets <i>Stilsättningsmall och standard som beskriver hur objekt ska presenteras grafiskt</i>
DBMS	Database Management System <i>Databashanterare hanterar teknisk kommunikation av data mellan användare och databas</i>
DTD	Document Type Definition <i>I DTD-filen specificeras vilka element som ska finnas med i XML-dokumentet</i>
EPSG	European Petroleum Survey Group <i>En samling standarder för koordinatsystem och koordinattransformationer</i>
ETL-verktyg	Extract Transform Load-verktyg <i>Verktyg för hantering och transformering av data</i>
GIS	Geografiska Informationssystem <i>Används för visualisering och analys av geografiska data</i>
GitHub	GitHub <i>Community och delningstjänst för källkod med mera</i>
GML	Geography Markup Language <i>XML-baserad kodningsstandard för geografisk information</i>
GeoJSON	Geographic JavaScript Object Notation <i>Spatial utvidgning av formatet JSON för att kunna hantera geografiska data</i>
GPL	General Public Licence <i>En licensform för öppen källkod</i>
HTML	Hypertext Markup Language <i>Ett dataspråk använt för visualisering av bilder och text på internet</i>
IETF	Internet Engineering Task Force <i>En organisation som berör utvecklingen internetarkitekturen</i>
INSPIRE	Infrastructure for Spatial Information in Europe <i>Ett EU-direktiv som syftar till en gemensam infrastruktur för geografiska data inom EU</i>
JSON	JavaScript Object Notation <i>Ett enkelt textbaserat format för datautbyte som påminner om XML</i>

LGPL	Lesser General Public License <i>En licensform för öppen källkod</i>
OGC	Open Geospatial Consortium <i>Ett konsortium som bidrar med fria standarder för hantering av geospatial data.</i>
OSD	Open Source Definition <i>En definition av öppen källkod</i>
OSGeo	The Open Source Geospatial Foundation <i>En organisation som verkar för att stötta utvecklingen av spatial teknologi med öppen källkod</i>
OSI	Open Source Initiative <i>En ideell stiftelse som verkar för att sprida kunskapen om öppen källkod</i>
QGIS	QGIS <i>GIS-programvara baserad på öppen källkod</i>
RIGES	Regional Innovativ GIS- och E-tjänst Samverkan <i>EU-projekt vars mål är att förenkla och skapa e-tjänster för översiktsplanering och bygglovsansökningar</i>
SDE	Spatial Database Enginge <i>Spatial databas producerad av ESRI</i>
SFSQL	Simple Features for SQL <i>Spatial utvidning av frågespråket SQL.</i>
SQL	Structured Query Language <i>En standard för att ställa logiska frågor för att hämta ut data ur databaser</i>
W3C	World Wide Web Consortium <i>Ett konsortium som utvecklar tekniska standarder för webben</i>
WFS	Web Feature Service <i>Nedladdningstjänst, används om användaren begär den önskade data via tjänsten som vektordata</i>
WMS	Web Map Service <i>Visningstjänst, en standard som definierar en karta som bildformat</i>
XML	eXtensible Markup Language <i>Textbaserad kodningsstandard för utbyte av data</i>
XSD	XML Schema Definition <i>Ett schema för hur XML-dokumentet ska presenteras</i>

Figurförteckning

- Figur 1.1 Generellt flödesschema för publicering av geografiska data på webben
- Figur 2.1 Malmö stads interaktiva karta för medborgarna
- Figur 2.2 Malmö stads solkarta
- Figur 2.3 En illustration av en höjdmodell över delar av Kristianstad kommun
- Figur 2.4 Walk of Film, en interaktiv karta i Ystad kommun
- Figur 2.5 Exempel från en av T-Kartors tjänster i London
- Figur 2.6 Översiktlig beskrivning och flödesschema av Tyréns GeoBIM-portal
- Figur 3.1 Geodataportalen, en samlingsportal för geodata i Sverige
- Figur 5.1 En karta producerad i Carto för ett testprojekt i GeoBIM
- Figur 5.2 ESRI's Sandbox, borrhål i 3D
- Figur 5.3 ESRI's Sandbox, borrhål i 2D-karta
- Figur 7.1 Egengjort flödesschema för publiceringsprocessen på Tyréns
- Figur 7.2 Egengjort flödesschema för publiceringsprocessen på Malmö stad
- Figur 7.3 Egengjort flödesschema för publiceringsprocessen på T-kartor Kristianstad
- Figur 7.4 Egengjort flödesschema för publiceringsprocessen på Kristianstad kommun
- Figur 7.5 Egengjort flödesschema för publiceringsprocessen på Ystad kommun
- Figur 9.1 Översiktligt flödesschema av den egna implementationen
- Figur 9.2 Översiktlig bild för tillvägagångssätt för att hämta data till klienten
- Figur 9.3 Översiktlig beskrivning av filstrukturen i klienten
- Figur 9.4 Egengjord figur av filstrukturen i den egna klienten
- Figur 9.5 Översiktlig startvy av den egna implementationen
- Figur 9.6 Illustration över byte av bakgrundskarta
- Figur 9.7 Översiktlig figur över hur punkternas läge flyttas om de överlappar
- Figur 9.8 Popup-fönster för en given punkt i kartan
- Figur 9.9 Innehållet i den första klickbara länken för borrhål 12B
- Figur 9.10 Legenden visar de objekt som visas i kartan. I detta fall sex olika objekt
- Figur 9.11 Legenden visar de objekt som visas i kartan. I detta fall endast tre olika objekt

Tabellförteckning

- Tabell 4.1 Jämförelse av olika licensformer
- Tabell 4.2 Jämförelse av öppen källkod och kommersiella programvaror
- Tabell 9.1 Utvärdering av tekniska krav
- Tabell 9.2 Utvärdering av krav på säkerhet
- Tabell 9.3 Utvärdering av krav på layout och stilsättning

Innehållsförteckning

1	Introduktion.....	1
1.1	Bakgrund.....	1
1.2	Problembeskrivning.....	1
1.3	Syfte	3
1.4	Angreppssätt.....	4
1.5	Avgränsningar	4
1.6	Disposition	5
2	Tillämpningar och tekniska lösningar av geografiska data på webben	6
2.1	Samhällsbyggnad.....	6
2.2	Malmö stad.....	6
2.3	Kristianstad kommun	8
2.4	Ystad kommun	9
2.5	T-kartor Kristianstad	9
2.6	Tyréns - GeoBIM	10
3	Geografiska data på webben.....	12
3.1	Geografiska data.....	12
3.2	Inspire.....	12
3.3	Tillgång till geografiska data.....	13
3.4	Publicering av geografiska data	14
3.4.1	Reglering kring publicering av geografiska data.....	15
3.5	Tekniker kopplade till geografiska data på webben	15
3.5.1	Hypertext Markup Language.....	15
3.5.2	JavaScript	15
3.5.3	eXtensible Markup Language.....	16
3.5.4	GML.....	16
3.5.5	GeoJSON.....	16
3.5.6	Open Geospatial Consortium	17
3.6	Databaser.....	17
4	Öppen källkod	19
4.1	Licenser och upphovsrätten.....	19
4.2	Öppen källkod	19
4.3	Copyleft-licenser	20
4.3.1	GNU GPL.....	21
4.3.2	GNU LGPL	21

4.4 BSD	21
4.5 Apache.....	22
4.6 Jämförelse av licensformer för mjukvaror	22
4.7 Creative Commons.....	23
4.8 Fördelar och nackdelar med öppen källkod.....	23
5 Kartverktyg.....	28
5.1 Verktyg för bearbetning av geografiska data	28
5.1.1 FME.....	28
5.2 Kommersiella program för server- och klientsidan.....	28
5.2.1 Carto.....	28
5.2.2 ArcGIS Online.....	29
5.2.3 Mapbox.....	31
5.3 Serverprogram med öppen källkod	31
5.3.1 Geoserver.....	31
5.3.2 Mapserver.....	32
5.3.3 QGIS-server.....	32
5.4 Klientprogram med öppen källkod.....	32
5.4.1 OpenLayers	33
5.4.2 Leaflet.....	34
5.4.3 Open ePlatform.....	34
5.4.4 MapGuide.....	34
5.4.5 QGIS – QGIS Web Client	34
6 Fallstudie	36
6.1 Upplägg	36
6.2 Val av respondenter för intervjustudie	36
7 Fallstudie del I - Publiceringsprocesser.....	37
7.1 Bakgrund	37
7.2 Publiceringsprocess - Tyréns AB	37
7.4 Publiceringsprocess - T-Kartor Kristianstad	39
7.5 Publiceringsprocess - Kristianstad kommun	41
7.6 Publiceringsprocess - Ystad kommun	42
7.7 Jämförelse av publiceringsprocesser	43
8 Fallstudie del II – Intervjustudie om öppen källkod.....	45
8.1 Intervju med Ulf Minör, stadsbyggnadskontoret Malmö stad.....	45
8.2 Intervju med Erik Ravhed, T-kartor Kristianstad.....	45
8.3 Intervju med Karl-Magnus Jönsson, Kristianstad kommun	46

8.4 Intervju med William Walker, Ystad kommun	47
8.5 Sammanställning av för- och nackdelar med öppen källkod från intervjuer.....	48
9 Fallstudie del III - Egen implementation.....	50
9.1 Kravspecifikation för visualiseringssteget	50
9.1.1 Bakgrund	50
9.1.2 Tekniska krav	50
9.1.3 Krav på säkerhet.....	50
9.1.4 Krav på layout och stilsättning.....	51
9.2 Utformning av publiceringsprocessen.....	51
9.3 Val av kart-server.....	52
9.4 Val av klient	53
9.4.1 Val av uppbyggnad av klient utifrån kravspecifikation.....	53
9.4.2 Andra aspekter vid val av klient.....	54
9.5 Val av bakgrundskarta.....	55
9.6 Teknisk beskrivning av klient	55
9.6.1 Kluster	57
9.6.2 Generalisering	58
9.6.3 Dynamisk stilsättning utifrån vilken data som läses in	58
9.6.4 Dynamisk legend.....	58
9.7 Användaraspekt – slutanvändare.....	59
9.8 Jämförelse av egen implementation gentemot Carto.....	62
10 Diskussion	65
10.1 Egen lösning	65
10.2 Metodval.....	65
10.3 Öppen källkod	66
10.4 Publiceringsprocesser.....	67
11 Slutsatser	69
Referenser.....	71

1 Introduktion

1.1 Bakgrund

I modern samhällsplanering används idag Geografiska Informationssystem, GIS, inom de flesta områden för visualisering och analys. Ett bra sätt att visualisera geografisk information är genom att publicera data på webben i både 2D- och 3D-kartor, vilket brukar benämnas webb-GIS. Området har ökat i intresse inom många branscher de senaste åren, vilket bland annat beror på att tekniken och verktygen för att hantera stora mängder data har förbättrats.

Framställning av kartor på webben är idag ett utbrett område både för statiska och interaktiva kartor, där utvecklingen av interaktiva webbaserade kartor gått fort framåt. Bara för några år sedan var tekniken långt ifrån där den är idag och kostnaden för framtagning av webbkartor var väldigt hög. Idag däremot är webbkartor ett stort och utvecklat område som varje individ med en webbläsare har tillgång till. Kartorna har gått från statiska till interaktiva, med hög funktionalitet anpassade och specifika till sitt användningsområde [1, s.306].

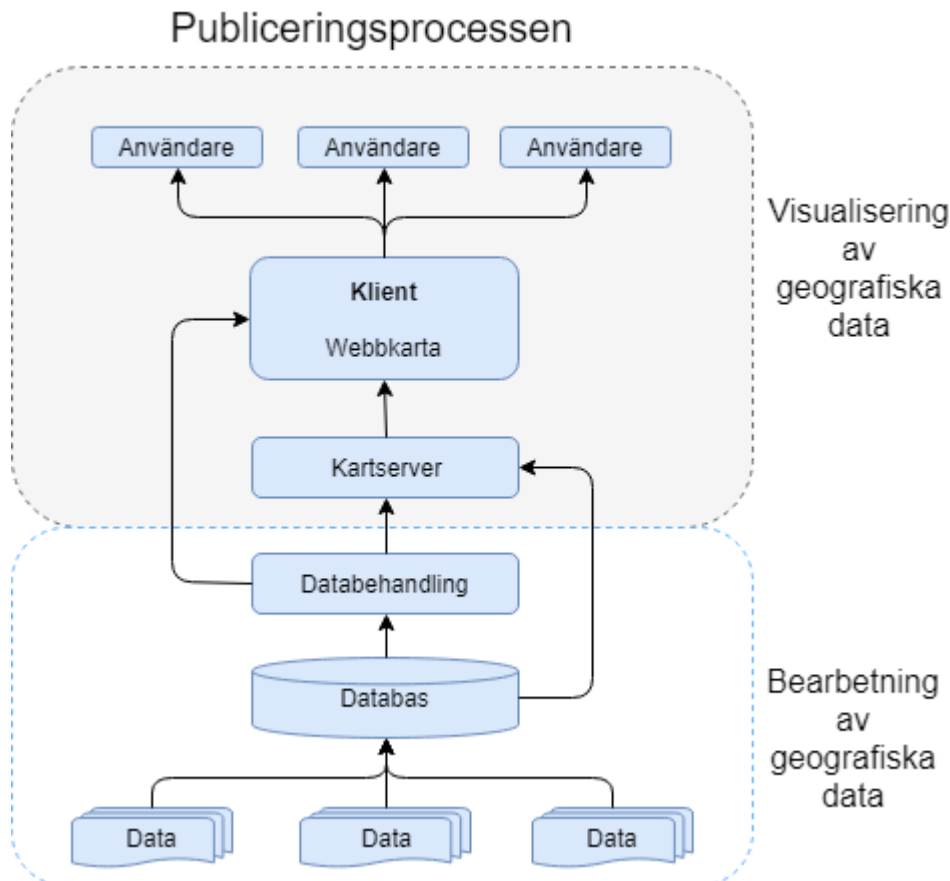
Det finns ett antal programvaror och system för att framställa kartor på webben, både kommersiella samt program utvecklade med öppen källkod. Kommersiella programvaror, det vill säga programvaror som har framställts av ett företag med vinstintresse, kräver oftast för full funktionalitet att programmet köps in med en användarlicens för att kunna nyttja tjänsten. En kommersiell karttjänst kan alltså innebära en stor kostnad men kanske även ger stor funktionalitet och god produktion i slutändan. Jämförelsevis distribueras program med öppen källkod oftast gratis, eller delvis gratis med vissa begränsningar. Villkoren för programvaran beror på vilken licens programmet publiceras under, där det finns olika licenser för öppen källkod, som anger villkoren för hur det specifika programmet får användas (se mer under kapitel 4) [1, s.31]. Denna avvägning mellan kostnad, funktion och eventuella begränsningar måste göras när det ska väljas vilken programvara eller system som ska användas i verksamheten.

I denna studie utvärderas ett antal program med öppen källkod som ett ersättningsalternativ till ett kommersiellt program, Carto, som används för publicering av webbkartor på många företag idag, däribland konsultbolaget Tyréns. Tanken bakom studien är att det ska leda till att öppen källkod, för publicering av geografiska data på webben, kan utvärderas som ett alternativ till Carto och liknande kommersiella kartverktyg.

1.2 Problembeskrivning

Publiceringsprocessen för en webbkarta, från kunddata till färdig produkt, innehåller idag flera flaskhalsar och manuella moment som är relativt likformade där många utvecklare måste lägga värdefull tid. Det finns därför ett behov att effektivisera denna process genom att, i den mån det går, automatisera processen.

Publiceringsprocessen skiljer sig åt mellan olika företag och organisationer beroende på vilka produkter och system som används samt vad det slutliga syftet är. En generalisering av publiceringsprocessen visas i figur 1.1. Publiceringsprocessen har i figuren delats in i två delar, bearbetning och visualisering av geografiska data på webben, här i rapporten benämnda som bearbetnings- och visualiseringssteget.



Figur 1.1 Generellt flödesschema för publicering av geografiska data på webben [2, s.218–222].

Processen, som inleds med bearbetningssteget, börjar oftast med en datainsamling från en eller flera aktörer. Data kan exempelvis tillhandahållas från Trafikverket som har information om vägar, Lantmäteriet som lagrar data om fastighetsregistret eller andra typer av data i enskilda projekt från någon annan fristående aktör. Data från alla dessa aktörer lagras sedan i en eller oftast flera databaser och hanteras sedan via databashanterare. För att förbereda data innan publicering genom att exempelvis klippa ut ett visst område, konvertera till en viss typ av format, byta eller lägga till attribut eller annan metadata så används ofta någon typ av databearbetningsprogram, så kallat ETL-verktyg. Ett exempel på ett sådant verktyg är FME (avsnitt 5.1.1) som i sin tur exporterar data vidare till någon typ av programvara för visualisering.

Efter bearbetning av data övergår data i detta exempel (figur 1.1) till visualiseringssteget. Steget börjar i form av ett kartserverprogram exempelvis Geoserver, där även stilsättning av data kan utföras. Data hämtas sedan från kartserverprogrammet av ett klientprogram där den slutliga stilsättningen och funktionaliteten implementeras till en färdig tjänst. Slutligen distribueras en

färdig webbkarta i klienten till användarna, vilket oftast sker genom en HTML-länk. Den publicerade kartan eller tjänsten ger användaren möjlighet att interagera med data i form av klickbarhet, pop-up och liknande funktioner [2, s.117–122, s.218–222].

Många företag, däribland Tyréns, använder idag den kommersiella produkten Carto för att visualisera och publicera projektspecifika geodata i webbkartor. Produkten har vissa begränsningar, exempelvis krävs det att utvecklaren manuellt går in och implementerar vissa delar för hur layouten ska designas och vilka funktioner den färdiga webbkartan till kunden ska ha. Dessa moment hade varit önskvärda för Tyréns att automatisera och/eller generera med en smidig standardmodell i ett led för att effektivisera processen. För att hitta en så optimal publiceringsprocess som möjligt genomförs en jämförelse av lösningar på ett antal kommuner och företag. Denna rapport berör hela publiceringsprocessen översiktligt men fokus läggs på visualiseringssteget. Att fokus läggs på visualiseringen är för att det primära syftet är att hitta en alternativ lösning till kommersiella programvaror för visualisering av geografiska data på webben, med utgångspunkt från Carto.

Förutom manuella moment innebär Carto också en stor kostnad då det är en kommersiell programvara. Kommersiella programvaror riskerar alltid att bli dyrare eller begränsas på andra sätt. Det är därför även intressant att utvärdera öppen källkod som ett alternativ till kommersiella lösningar. För att kunna jämföra olika kommersiella och öppna källkodsalternativ för webbkartor måste många faktorer vägas in, exempelvis licenskostnader, vilka varierar för många kommersiella program och lösningar idag. Licenskostnaden måste i sin tur jämföras mot den funktionalitet som tjänsten erbjuder. Ett vanligt problem är att tjänster som många företag har bundit upp sig på alltid medför en risk för förändring med avseende på prisnivåer och/eller begränsningar i funktionalitet. Utifrån vetskapen att en tjänst med öppen källkod i vissa fall riskerar att bli kommersiell måste även denna aspekt beaktas.

Det är också väsentligt att utreda hur systemet hanterar kunder och sina egna data på ett säkert sätt. Möjlighet för hög integritet är en viktig punkt i valet av lösning då kundernas data kan vara sekretessbelagd eller hemlig av andra orsaker. Förutom skyddet för kunderna är en annan betydelsefull aspekt att företaget själv har möjlighet att säkra sin egen verksamhet. Det är betydande att ha möjlighet att skydda sin data enligt standarder och snabbt uppdatera möjliga säkerhetsbrister i ett system. Förutom de tidigare nämnda punkterna är det alltid viktigt för ett program med öppen källkod att det finns en stor intresseorganisation (*community*) bakom som kan driva projektet för att få en framtidssäker och hållbar lösning. Det är också av väsentlighet att det följer givna standarder och att funktionaliteten håller den nivå som krävs av branschen.

Utifrån de ovan angivna aspekter jämförs, som tidigare nämnts, ett antal program med öppen källkod mot kommersiella alternativ med utgångspunkt av kartverktyget Carto.

1.3 Syfte

Examensarbetets främsta syfte är att utvärdera om program utvecklade med öppen källkod är ett likvärdigt alternativ till det kommersiella program Carto för att publicera geografiska data

på webben. Studien syftar även till att utvärdera fördelar och nackdelar med öppen källkod för generering av webbkartor. Rapporten syftar även till att utvärdera i vilken utbredning öppen källkod används i praktiken och om det förekommer olika mentaliteter eller ideologier kring öppen källkod i olika verksamheter.

Studien syftar även till en teknisk utvärdering av öppen källkod genom att utvärdera olika programvaror och system för visualisering av geografiska data i en webbmiljö. Detta görs genom en fallstudie där olika kartverktyg med öppen källkod testas och utvärderas i jämförelse mot det kommersiella programmet Carto. Utvärderingen utgår från uppsatta krav för visualiseringssteget i form av en kravspecifikation vilken är framtagen i samarbete med Tyréns. Kravspecifikationen utgår från önskvärd funktionalitet i en webbkarta.

Ett ytterligare syfte är att utvärdera publiceringsprocessen från data till färdig webbkarta och kartlägga alternativa lösningar. Fokus läggs framförallt på visualiseringssteget och ett mål är därför att utvärdera hur olika kommuner och företag gör för att publicera sina data på webben.

1.4 Angreppssätt

Första delen av rapporten består av en litteraturstudie för publicering av geografiska data på webben. Denna litteraturstudie behandlar bland annat olika typer av licensformer för öppen källkod. Teoridelen tar även upp olika programvaror, såväl kommersiella som programvaror med öppen källkod och innehåller en genomgång av fördelar och nackdelar med öppen källkod.

Litteraturstudien följs av en fallstudie bestående av tre olika delar, där de två första delarna består av intervjustudier. Första delen av intervjustudien är en processkartläggning avseende förekommande publiceringsrutiner, från data i databas till färdig karta, på några utvalda kommuner och företag. Dessa rutiner används sedan för en jämförelse och utvärdering av den publiceringsprocess som råder på Tyréns idag. Denna undersökning leder till en framtagning av standardrutiner för en effektiv publicering av geodata på webben. Den andra delen av intervjustudien syftar till att utvärdera fördelar och nackdelar med öppen källkod. Slutligen genomförs en testimplementering i visualiseringssteget för en vald lösning med öppen källkod som resulterar i en utvärdering och jämförelse av Carto som publiceringstjänst. Utvärderingen utförs från en kravspecifikation, där funktionaliteten i kartverktyget Carto jämförs gentemot möjliga program med öppen källkod med avseende på problemställningarna i avsnitt 1.2 ovan.

1.5 Avgränsningar

Fallstudien behandlar den kommersiella programvaran Carto som utgångspunkt och jämförs sedan med ett antal program med öppen källkod. För att göra studien mer omfattande hade det varit av väsentlighet att jämföra fler programvaror. Studien berör främst visualiseringssteget, vid val av kartverktyg behöver andra faktorer också beaktas. Studien är främst granskad från utvecklarens perspektiv, men även vissa kunders perspektiv har utvärderats för att få en så komplett bild som möjligt över problemområdet. För att utvärdera publiceringsprocessen och

få en bild av de åsikter som finns angående öppen källkod har ett antal kommuner och företag valts ut och intervjuats. För att få en mer heltäckande bild kunde studien inkludera fler intervjuade organisationer.

1.6 Disposition

Kapitel 2 innehåller ett antal tillämpningsområden för publicering av geografiska data på webben. De exempel som tas upp i detta kapitel är tagna från de utvalda företag och kommuner som intervjuas i fallstudien. Kapitel 3 beskriver den teoretiska bakgrunden för geografiska data. Avsnittet tar upp reglering kring publicering av geografiska data och de grundläggande formaten för webb-GIS. Kapitel 4 behandlar även databaser och berör de olika systemen för att lagra geografiska data och hur data ska förberedas för att kunna användas i senare led för publicering och analys. Kapitel 4, öppen källkod, redogör för de olika licensformer som finns för att använda och utveckla kartverktyg med öppen källkod. Fokus läggs på de licensformer som används i de programvaror som berörs i denna rapport. Kapitel 5 behandlar även fördelar och nackdelar för öppen källkod. Kapitel 5, kartverktyg, undersöker programvaror, både de som har öppen källkod men även kommersiella program som används för att publicera geografiska data på webben. Kapitel 5 beskriver både server- och klientprogram. Kapitel 6 inleder fallstudien och beskriver upplägget för denna. Kapitel 7 och 8 är första och andra delen av fallstudien bestående av en intervjustudie, där första delen tar upp publiceringsprocesser och den andra behandlar öppen källkod. Kapitel 9 är den tredje och sista delen av fallstudien och beskriver en egen lösning och implementation. Rapporten avslutas med kapitel 10 diskussion och kapitel 11 slutsatser.

2 Tillämpningar och tekniska lösningar av geografiska data på webben

2.1 Samhällsbyggnad

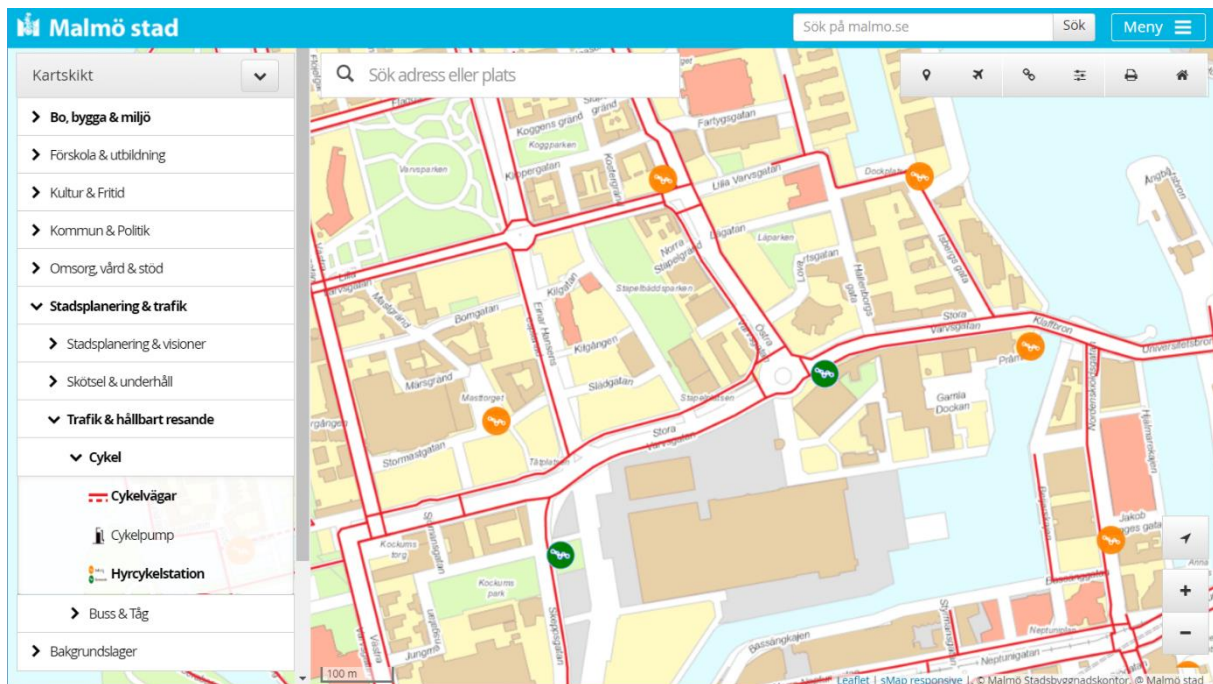
I både stora och små samhällsbyggnadsprojekt är vikten av att kunna presentera sina data på ett ändamålsenligt sätt stor. Om ett företag exempelvis ska konstruera en ny byggnad måste ett stort antal faktorer först beräknas, bland annat markförhållanden. En karta som visualiserar dessa markförhållanden är då ett betydande och användbart verktyg för byggprocessen redan i projekteringsstadiet, då kartan kan visualisera byggnadsområdet. Ju tidigare i ett projekt fel och brister upptäcks desto billigare blir själva slutprojektet, och om alla faktorer i en karta behandlas på ett korrekt sätt finns det stora pengar att spara [1, s.24].

Geografiska data och visualisering av dessa är således en förutsättning för alla typer av samhällsprojekt och för att samhället ska fungera. Det är med andra ord av intresse att utreda olika lösningar för hur publicering av geografiska data görs idag. Under avsnitt 7, intervjustudie publiceringsprocesser, genomförs en utvärdering av olika publiceringsprocesser, på kommuner såväl som hos kommersiella aktörer. Då publiceringsrutiner skiljer sig åt på olika kommuner och företag är det av intresse att jämföra dessa rutiner.

Under nedanstående avsnitt presenteras exempel från olika tillämpningsområden för publicering och visualisering av geografiska data i en webbmiljö. Det organisationer som utvärderas är Malmö stad, Ystad kommun, Kristianstad kommun, T-kartor och Tyréns. Dessa aktörer är ett genomgående exempel i rapporten då det är dessa parters lösningar som utvärderas under kapitlen 7 och 8 i rapporten. Det är av intresse att jämföra hur en stor organisation som Malmö stad fungerar jämfört med en mindre kommun som Ystad. Det är även intressant att se om lösningar skiljer sig mellan kommuner och kommersiella företag.

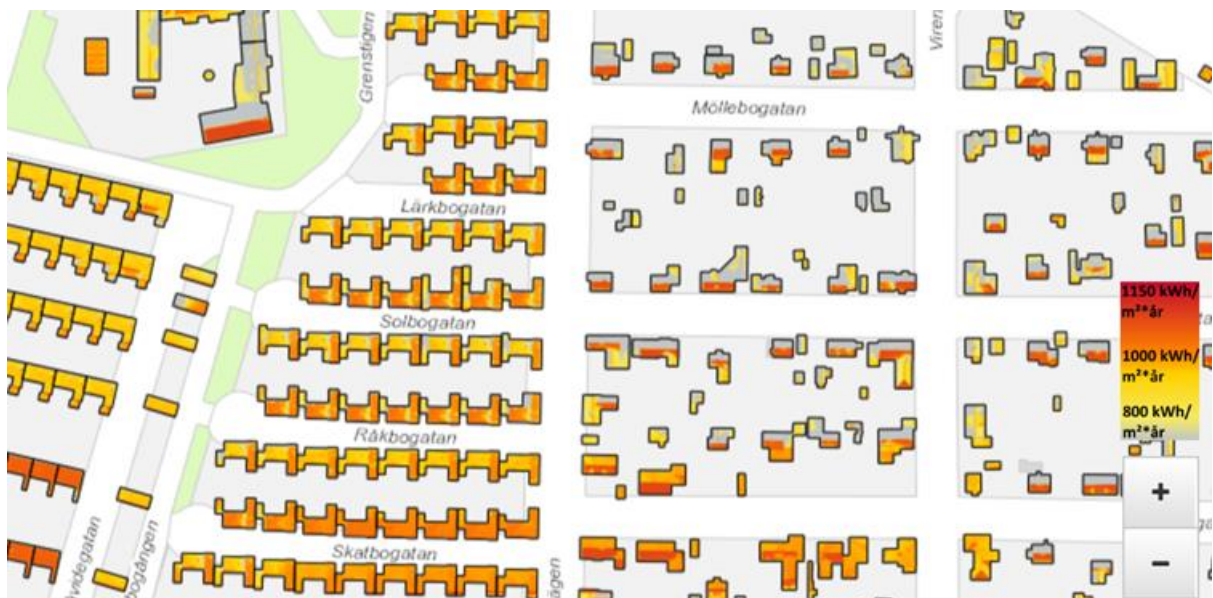
2.2 Malmö stad

Många kommuner publicerar en stor mängd av sin geografiska data på webben där vissa data är distribuerat fritt och andra delar i begränsad form genom exempelvis en betaltjänst. Om Malmö stad sätts som exempel tillhandahåller de medborgarna en stor mängd geografiska data i form av interaktiva kartor på sin hemsida. Dessa interaktiva kartor erbjuder medborgarna allt från kommunens anläggningar, skolor, busshållplatser, detaljplaner och erbjuder även en solkarta. Exempel på dessa kartor kan ses i figurerna 2.1 och 2.2.



Figur 2.1 Malmö stads interaktiva karta för medborgarna. Här är lagren för cykelvägar och hyrcykelstationer aktiverade [3]

Solkartan är en tjänst där en fastighetsägare själv kan gå in och undersöka vilken mängd solinstrålning som faller på taket på den egna fastigheten. Denna tjänst, som från början är utvecklad av Tyréns, gynnar medborgarna genom att göra det möjligt att se potentialen i att investera i solceller på sitt eget hus [4].



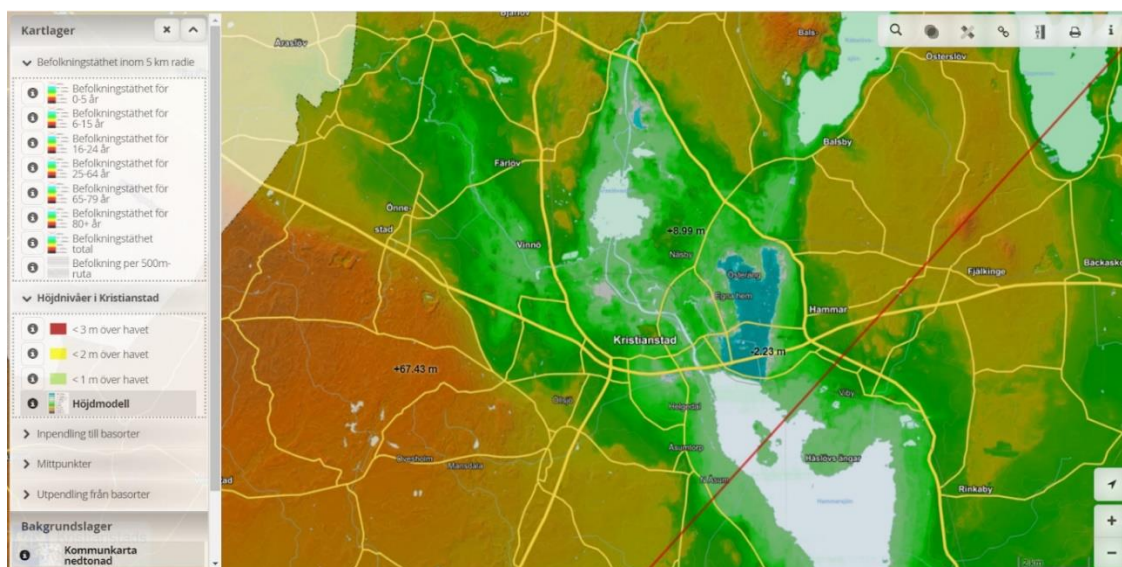
Figur 2.2 Karta över ett bostadsområde i Malmö där mängden sol mätt i kWh/m²*år kan observeras [5]

Malmö är en av de kommuner som tidigare valt en plattform med öppen källkod för sina webbkartor. De webbkartor som är publicerade på kommunens hemsida just nu använder Leaflet, vilket är ett gränssnitt baserat på öppen källkod (avsnitt 5.4.2), och sMap. Ramverket sMap används för publicering av webbkartor och är ett samarbete mellan Malmö, Helsingborg, Lund och Kristianstad. Ramverket bygger, som tidigare nämnts, på gränssnittet Leaflet [6]. Samarbetet startade 2009 och avslutades några år senare. Två liknande samarbeten mellan kommuner i Sverige där systemet är konstruerat med öppen källkod är exempelvis plattformarna, eSamverkan och Riges (avsnitt 5.4.3) [7][8]. En djupare inblick i vilka lösningar Malmö stad har för att publicera sin data finns i avsnitt 7.3, publiceringsprocess för Malmö stad.

2.3 Kristianstad kommun

Liksom för Malmö stad publicerar Kristianstad kommun en hel del kommunspecifika data för sina medborgare. Kommunen tillhandahåller tre publika webbkartor för medborgarna, Kristianstadkartan, Strategikartan och planeringskartan. Kristianstadkartan ger användaren möjlighet att hitta en fastighet eller andra punkter av intresse, exempelvis bibliotek, skolor eller badhus. Planeringskartan redovisar planeringsunderlag för kommunen och tillhandahåller även samrådsförslag. Kartan är tänkt att vara ett komplement till kommunens översiktliga planering.

Strategikartan, vilken distribueras till medborgarna via kommunens hemsida, däremot innehåller analyser av olika slag med syftet att vara ett underlag för kommunens olika beslut. Ett exempel på Kristianstad kommuns strategikarta kan ses i figur 2.3 nedan. I figuren illustreras en höjdmodell över en del av kommunen. Eftersom Kristianstad är speciellt med tanke på att stora delar av kommunen ligger på en marknivå på mindre än en meter över havsnivå är höjdmodellen intressant ur exempelvis byggnadssynpunkt [9].

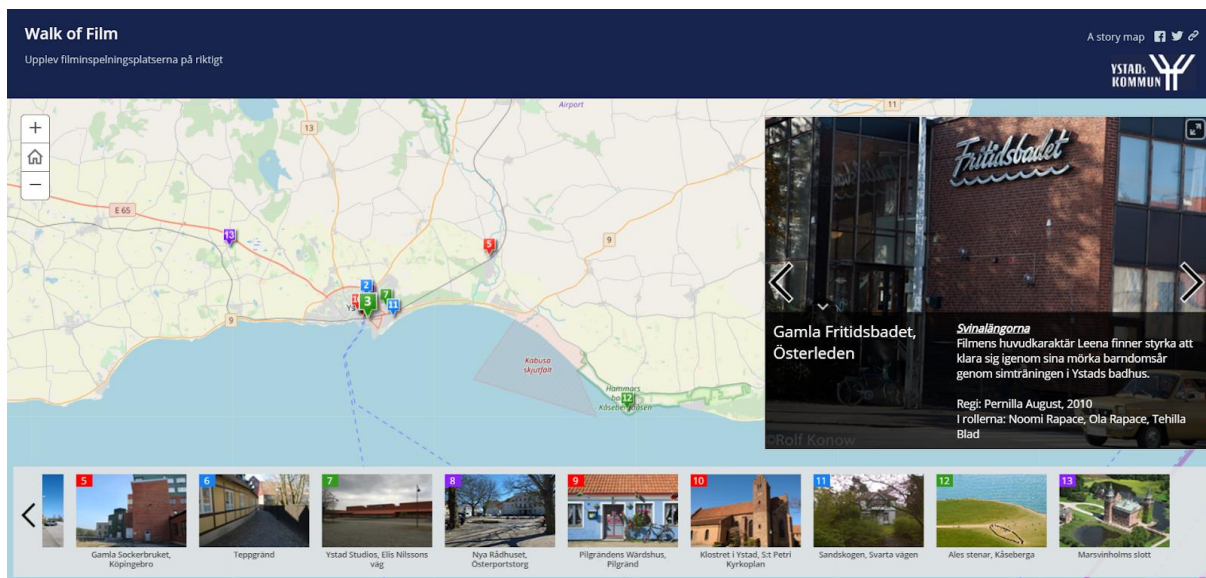


Figur 2.3 En illustration av en höjdmodell över delar av Kristianstad kommun [10].

Kristianstad var, som tidigare nämnts, en del av samarbetet sMap som byggde på användandet av öppen källkod. Kristianstad kommun har efter samarbetet valt att behålla användningen av öppen källkod och använder öppen källkod till största del i verksamheten, vilket går att läsa mer i avsnitt 7.5 och 8.3.

2.4 Ystad kommun

Ystads kommun är en relativt liten kommun i södra Skåne med cirka 30 000 invånare [11]. Trots det producerar Ystad en del egna webbkartor, dels kommunkartor men även kartor för att främja turismen i kommunen. Kommunkartan är interaktiv, liksom kommunkartorna i Malmö stad och Kristianstad kommun, och distribueras på webben med hjälp av ArcGIS online (avsnitt 5.2.2). För att främja turismen har kommunen skapat en karta så kallad walk of film, se figur 2.4, som visar olika platser i och runt Ystad som använts för filminspelningar [12]. Det kan observeras att Ystad, till skillnad från Kristianstad, använder en kommersiell lösning för sina webbkartor. I avsnitt 7.6, publiceringsprocess för Ystad kommun, och 8.4 intervju med William Walker går det att läsa mer om vilka lösningar som valts i kommunen och hur kommunen ser på fördelar och nackdelar med öppen källkod.



Figur 2.4 Walk of Film, en interaktiv karta som visar olika inspelningsplatser för filmer som har producerats runt om Ystad, kartan produceras av Ystads kommun med hjälp av ArcGIS-online [13] [14].

2.5 T-kartor Kristianstad

T-Kartor är ett GIS-företag som ursprungligen etablerades i Kristianstad där också huvudkontoret ligger, men har idag även kontor på flera andra platser i världen. T-kartor utvecklar flera olika produkter, däribland kartor, system och andra geospaciala lösningar för

flera försvarsorganisationer bland annat amerikanska och norska militären. Företaget har även arbetat med navigationslösningar inom kollektivtrafiken i städer som London, New York, Paris och Toronto, se figur 2.5. T-Kartor använder både kommersiella och lösningar med öppen källkod i sina produkter och har stor kompetens inom området [15] [16].

I avsnitt 7.4 och 8.2 går det att läsa mer om vilka lösningar som används på T-kartor, och utvecklaren Erik Ravheds syn på fördelar och nackdelar med öppen källkod.



Figur 2.5 T-Kartor har bland annat varit med och utvecklat karttjänster för kollektivtrafiken i flera storstäder, här en bild från London [16].

2.6 Tyréns - GeoBIM

När det ska genomföras ett stort byggnadsprojekt är det av stor väsentlighet att veta hur marken som ska bebyggas ser ut under markytan. Denna geotekniska undersökning ligger sedan till grund för vilka åtgärder som behövs för att exempelvis stabilisera marken, eller schakta. Andra viktiga aspekter för ett byggnadsprojekt kan exempelvis vara geohydrologi, vilket bland annat behandlar grundvattennivåer och geofysik, som till exempel kartlägger eventuella naturresurser [17]. Miljöpåverkan är centralt i såväl planerings-, projekterings-, som byggnadsfasen. Det är med andra ord essentiellt att känna till miljöförhållanden under markytan och kännedom om eventuella miljögifter i marken då dessa kan ha en signifikant påverkan på byggnadsprojektet i form av saneringskostnader eller andra åtgärder. [18].

Undersökningarna, som tidigare nämnts, är inga nya tekniker i sig, dessa har genomförts även tidigare. Problemet har tidigare varit att de geotekniska undersökningarna varit knutna till specifika projekt, vilket lett till att dokumentation och filer varit utspridda mellan olika parter. GeoBIM är en portal, utvecklad av ett antal medarbetare på Tyréns, som samlar geoteknisk, geofysisk och miljögeoteknisk information i en och samma webbmiljö. Portalen innehåller bland annat en 3D-modell som visar den tekniska undersökningen, en öppen databas som samlar all data och en webbkarta där all data tydligt visualiseras i en interaktiv 2D-karta. En översiktlig illustration av Geo-BIM portalen visas i figur 2.6.

På grund av att det är en webbaserad tjänst där all data samlas i en molntjänst underlättar denna tjänst för såväl i projekteringsfasen som i byggnadsfasen [19]. I avsnitt 7.2, publiceringsprocess på Tyréns, går det att läsa mer om flödet från data till webbkarta i GeoBIM-projektet.



Figur 2.6 Översiktlig beskrivning och flödesschema av Tyréns GeoBIM-portal [19].

3 Geografiska data på webben

3.1 Geografiska data

Geografiska data är alla data som går att knyta till ett geografiskt läge som exempelvis punkter, linjer och polygoner som representerar fysiska objekt som hus, vägar och markområden. Geografiska data kan även vara data som exempelvis personlig information, som adress, telefonnummer etcetera. Även denna typ av data går indirekt att koppla till ett läge.

För att bättre visualisera geografiska data används oftast någon form av konceptuell modell av verkligheten, exempelvis en karta. När geografiska data ska visualiseras måste alltid vissa avvägningar göras eftersom en karta aldrig helt stämmer överens med verkligheten. En karta är endast en modell av verkligheten, vilket innebär att det måste bestämmas vad det är som ska visas och rangordna detta i vad som är viktigast. Ställningstagandet innebär att kartkonstruktören värderar vad som är viktigast att visualisera för just den kartan som ska produceras. För en vägmappa är det uppenbart att vägarna är det mest centrala att visualisera, för en geologisk karta är den geologiska jordlagren av större väsentlighet att kunna presentera på ett tydligt sätt. I praktiken eftersträvas sedan att modelleringen av punkter och objekt med liknande egenskaper associerar till en typ av kategori som sedan brukar visas i teckenförklaringen i kartan [1, s.24].

3.2 Inspire

De flesta länder i världen skapar och använder idag geografiska data, som exempelvis data över vägnät, befolkningstäthet och markanvändning. Alla dessa länder har även egna metoder och standarder för att beskriva dessa data. Den nationella lösningen leder till problem till exempel vid naturkatastrofer eller naturfenomen som inte har några landsgränser och sällan drabbar ett enskilt land. Vid sådana fall påverkas många länder antingen direkt eller indirekt. Ett exempel på en indirekt påverkan var vulkanutbrottet på Island 2010 som drabbade flygtrafiken i stora delar av Europa. Översvämningar, vulkanutbrott eller jordbävningar drabbar med andra ord oftast mer än bara ett land och därför finns det ett stort behov av en samlad standard för att enkelt och snabbt få tillgång till den nödvändiga informationen [20].

Inspire, *Infrastructure for Spatial Information in Europe*, är ett EU-direktiv vilket började gälla 2007 och förväntas vara färdigutvecklat 2021. Dess syfte är att medlemsländerna ska kunna samverka under en gemensam standard för hela EU. Med hjälp av denna gemensamma infrastruktur för geografiska data eftersträvas en möjlighet för att enkelt kunna utnyttja den offentliga geografiska data som finns i EU:s medlemsländer. Tanken bakom Inspire är även att effektivisera delningen och hanteringen av geografiska data mellan medlemsländerna [20].

Bestämmelserna för Inspire-direktivet är antagna som beslut av EU-kommissionen vilket innebär att de är direkt bindande för samtliga medlemsländer i EU. I Sverige är direktivet styrt av Lag (SFS 2010:1767) om geografisk miljöinformation och Förordning (SFS 2010:1770) om

geografisk miljöinformation. Lagen (SFS 2010:1767) går ut på att skapa ett sammanhängande system, så kallad infrastruktur, vilken underlättar möjligheten att utnyttja den offentliga geodata som finns. Geodata ska göras tillgänglig för alla genom att samtliga länder har en skyldighet att göra det möjligt för allmänheten att söka, titta på och ladda ned geografiska data. Det innebär även att denna hantering av geografiska data ska vara gratis. Förordning (SFS 2010:1770) behandlar bland annat informationsansvaret och det nationella och europeiska samarbetet. Förordningen reglerar att det i Sverige är Lantmäteriet som ansvarar för den svenska infrastrukturen av geografiska data [21].

I ett led för att genomföra Inspire-direktivet i Sverige har Lantmäteriet, Geodatarådet och andra berörda myndigheter tagit fram en geodatastrategi för hur geografiska data skall samordnas och lättare bli tillgängligt för allmänheten, företag och andra aktörer som använder geografisk information. Den nationella geodatastrategin i Sverige för 2016–2020, som ska se till att landet har en hållbar infrastruktur för geografiska data, delas översiktligt in i fyra målområden [22]:

1. Geodata är öppna

Målet syftar till att geografiska data i så hög utsträckning som möjligt ska vara avgiftsfri och åtkomlig för både privata och offentliga verksamheter. Detta för att bidra till en innovativ samhällsutveckling där tillgången till geografiska data ska var gratis och lättåtkomlig [22].

2. Geodata är användbara

Målets syfte är att geografiska data ska vara lätt att använda, det vill säga följa de aktuella standarder för geodata som finns, både nationella och internationella. Det innebär också att data ska vara lätt att konvertera mellan olika system och uppfylla de krav som ställs inom respektive bransch. Den ska även vara enhetlig och alltid uppdaterad av den ansvariga myndigheten [22].

3. Geodata är tillgängliga

För att öka och underlätta användningen av geografiska data krävs det att det finns olika typer av webbtjänster och andra applikationer, som är lätta att använda både för allmänheten och utvecklande företag. Tjänsterna ska utöver det vara väl anpassade och integrerade i tiden för att möta efterfrågan [22].

4. Samverkan är välutvecklad

För att driva samhällsutvecklingen framåt krävs mycket samarbete och dialog mellan de olika myndigheterna och andra berörda aktörer. Målet syftar till att underlätta detta samarbete och kunskapsutbyte mellan dessa aktörer för att dra nytta av varandras kunskaper och utveckla en innovativ och modern geodatastrategi [22].

3.3 Tillgång till geografiska data

För att uppfylla det tredje målet för Sveriges nationella geodatastrategi, att geografiska data ska vara tillgänglig för alla, finns det ett antal utvecklade tjänster. Tre av dessa tjänster är Geodataportalen, Inspire-geoportalen och Lantmäteriets Kartsök.

En tjänst som har utvecklats för enklare få tillgång till alla samlade geografiska data i Sverige är geodataportalen, figur 3.1. Tjänsten samlar all geografiska data från myndigheter där användare enkelt kan välja söka, visualisera, ladda ner och hitta information om geografiska data som finns tillgängligt från myndigheter i Sverige [23].



Figur 3.1 Geodataportalen, en samlingsportal för geografiska data i Sverige [24].

Inspire-geoportal är en annan samlingsportal för geografiska data men är en portal som täcker över hela Europa, där den svenska geodataportalen är en del. Tjänsten innehåller länkar till den geodata som varje delaktigt land tillhandahåller i enlighet med Inspire-direktivet. Portalen innehåller också funktioner, för medlemsländer och leverantörer av geodata, för att undersöka om data, tjänster och metadata uppfyller de tekniska riktlinjerna i Inspire. Tjänsten är också en informationsportal för lagstiftning annan relevant information gällande geodata i Europa både nationellt och globalt [25].

Lantmäteriets Kartsök är en tredje karttjänst som är fri att använda. För privat bruk är det fritt att ladda ner och använda alla bilder och kartor i tjänsten, men det bör understrykas att fastighetsgränser i kartan inte är juridiskt bindande och noggrannheten på dessa kan variera betydligt. För att sprida och publicera kartor och material från tjänsten vidare krävs medgivande från Lantmäteriet förutom den data som klassificeras som öppen det vill säga historiska och topografiska kartor utan fastighetsgränser, denna data sprids under en *creative commons* licens (CC0 1.0) vilket behandlas mer i avsnitt 4.7 [26].

3.4 Publicering av geografiska data

1 juli 2018 slutade Lantmäteriet trycka papperskartor för att satsa på digitala kartor eftersom efterfrågan på tryckta kartor drastiskt minskat. Däremot är deras kartinformation mer efterfrågade än någonsin [27]. Detta ger en indikation på att publicering av geografiska data på webben är enormt viktigt och efterfrågat. Tillsammans med vetskapen av att kartan idag är en central del för många, då alla människor med en mobil har tillgång någon typ av digital karta, kan det inte nog betonas vikten av publicering av geografiska data.

3.4.1 Reglering kring publicering av geografiska data

I takt med att tekniken för publicering av kartor på webben blir enklare och mer utbredd är det också viktigt att understryka vikten av att ta hänsyn till olika typer av lagar och regler som finns kring publicering av geografiska data. Det finns framförallt tre aspekter som behövs skyddas vid - publicering, personlig integritet, nationell säkerhet och upphovsrätt. I denna rapport kommer främst upphovsrätten och undantag från denna, i form av olika licenser, att beröras.

Upphovsrätten regleras i *upphovsrättslagen för litterära och konstnärliga verk* (SFS 1960:729) och definierar att den som skapar en karta är upphovsman till den. Generellt gäller det att en karta inte kan användas utan tillstånd från upphovsmannen, dock med vissa inskränkningar, vilket behandlas mer i avsnitt 4.1 licenser. Upphovsrätten är uppdelad i två delar, den ekonomiska och den ideella delen. Den ekonomiska delen innebär helt enkelt att den som har skapat kartan ska få betalt för sitt jobb medan den ideella rätten är definierad som rätten till att bli omnämnd [28].

3.5 Tekniker kopplade till geografiska data på webben

För att presentera geografiska data på webben används en del olika filformat, standarder och datorspråk beskrivna nedan.

3.5.1 Hypertext Markup Language

HyperText Markup Language, HTML, är ett dataspråk använt för visualisering av bilder och text på internet. HTML-dokument, ofta benämnda som webbsidor, består av ett antal taggar (element) som beskriver innehållet i dokumentet. Fördelen med HTML är att det enkelt kan produceras genom ett enkelt program för textredigering exempelvis Notepad [29, s.16].

3.5.2 JavaScript

JavaScript är ett objektorienterat programmeringsspråk som är utformat för att användas tillsammans med HTML. JavaScript fungerar inte som ett fristående programmeringsspråk utan är utvecklat, med omfattande funktioner, för att utvidga HTML-kodningen. JavaScript är bland annat användbart för att manipulera text och bilder men även för ett stort antal andra problem och tillämpningar. Detta görs på klientens sida, vilket innebär att när en användare laddar en webbsida skickas JavaScript-koden till klientens dator som läser av koden och sedan presenterar innehållet i webbsidan [30].

3.5.3 eXtensible Markup Language

eXtensible Markup Language, XML, är en välkänd och mycket använd standard som underlättar framställning och utbyte av data. XML har utvecklats av *World Wide Web Consortium*, W3C, som är ett konsortium som bland annat arbetar med att utveckla tekniska standarder för webben. XML spelar en stor roll för kartproduktion och GIS-tillämpningar på webben [29, s.27].

Ett XML-dokument bör följa en viss struktur, med denna struktur menas att exempelvis taggar, attribut är i rätt ordning eller att attributen är att rätt datatyp. Vad som är rätt definieras och specificeras antingen i en DTD, Document Type Definition, eller XSD, XML Schema Definition [29, s.33]. I båda fallen specificeras vilka element som ska finnas med i XML-dokumentet, deras struktur och en mängd andra specifikationer [31, s. 79].

3.5.4 GML

GML, *Geography Markup Language*, är en XML-baserad kodningsstandard för geografisk information som utvecklats av *Open Geospatial Consortium* (se avsnitt 3.5.6). Likt XML-kodningen representeras GML i form av text, vilket är enkelt att skriva och använda. Skillnaden är att GML beskriver geografisk information och objekt [32]. Ett av standardens främsta mål är att erbjuda ett språk för att uttrycka geografiska funktioner och objekt på ett sätt som enkelt kan delas på internet. Detta erbjuds genom att GML lägger på ytterligare struktur till XML som exempelvis regler om användningen av attribut men reglerar även klassificeringen av element till de två grupperna objekt och egenskaper [33].

En huvudsaklig anledning till att GML används så brett som det gör är att det bygger på XML och därför är enkelt att använda och modifiera. En grundläggande del av ett geografiskt informationssystem är funktionen att kunna referera till ett geografiska punkter. Dagens version av GML gör det möjligt att referera till alla referenssystem som är definierade av *European Petroleum Survey Group*, EPSG. Precis som för XML använder GML sig av ett dokument, GML Schema, som beskriver hur och vad dokumentet ska innehålla. GML kan även användas för icke-spatiala data och enkelt transformeras till andra filformat [34].

3.5.5 GeoJSON

JSON är ett enkelt textbaserat format för datautbyte som påminner om XML. GeoJSON är en geometrisk utvidgning av formatet JSON, *JavaScript Object Notation*, och används för att överföra geografiska data. Formatet gör det möjligt att beskriva exempelvis punkter, linjer och polygoner i ett koordinatsystem, vilket gör det till ett bra format för överföring av vektordata. Formatet GeoJSON följer den öppna standarden (RFC 7946) som är uppsatt av organisationen IETF, *Internet Engineering Task Force* [35].

3.5.6 Open Geospatial Consortium

Eftersom geografiska data kan representeras i flera olika format och koordinatsystem ökar behovet av gemensamma standarder. *Open Geospatial Consortium, OGC*, är ett internationellt konsortium bestående av bland annat företag och forskningsorganisationer, som är engagerade i att bidra med fria standarder för hantering av geospatial data. Dessa standarder har som syfte att förbättra delningen av geografiska data och är fritt tillgängliga för alla att använda [36].

Standarderna är en samling dokument som specificerar uppbyggnaden av källkod. Användarna kan sedan utgå ifrån dessa dokument för att skapa öppna gränssnitt i sina tjänster och produkter. Syftet med detta konsortiums standarder är att möjliggöra en överensstämmande och sammanhållen standard för geografisk datalagring, men även för tjänster och applikationer [37].

OGC har bland annat utvecklat standarder för olika karttjänster på webben som exempelvis visningstjänster (WMS) och nedladdningstjänster (WFS) [37]. En Web Map Service, WMS, är en internationell standard, vilken definierar en karta som ett bildformat, exempelvis PNG [38]. För att hämta vektordata till webbkartor används tjänsten Web Feature Service, WFS, där klienten det vill säga användaren, begär den önskade data via tjänsten som sedan returneras som ren vektordata utan stilsättning tillbaka till användaren [29, s.113].

3.6 Databaser

För att lagra geografiska data kan data antingen sparas direkt till fil lokalt i systemet eller mot en databas. I stora system, där stora mängder data måste lagras, används nästan uteslutande databaser för lagring. För att kunna kommunicera och hämta data från databasen till användarprogrammet behövs någon form av databashanterare, *Database Management System, DBMS* [1, s.163].

Databashanterarens huvuduppgift är sköta den tekniska kommunikationen av data mellan användare och databas. De viktigaste kraven för ett bra system är hög driftsäkerhet samt god driftsekonomi, det vill säga hög säkerhet och korta svarstider av datatransaktioner samt på ett effektivt och korrekt sätt kunna hantera många förfrågningar och användare [1, s.165f]. Det finns olika typer av databashanterare och generellt sätt inom IT så är relationsdatabaser och NoSQL-databaser, "*not only SQL*", de vanligaste typerna av databashanterare [39].

NoSQL är en samling av databaser med det gemensamt att de inte är relationella. Det finns många olika typer av NoSQL-databaser som hanterar lagring på lite olika sätt, de vanligaste är nyckel-värdeslagring, kolumn-orienterad lagring, grafbaserad lagring och dokumentbaserad lagring. I vissa tillämpningar är denna typ av databas mer flexibel och snabbare än den relationella exempelvis vid hantering av stora mängder ostrukturerade data som exempelvis bilder, e-post och sociala medier [40].

Relationsdatabaser är den vanligaste typen av databaser som används för datalagring inom IT. Enkelt beskrivet så är denna databas konstruerad som tabeller och relationer mellan dessa. Varje rad i tabellen innehåller data som är unik för attributen som definieras i kolumnerna [40].

Fördelarna med relationella databaser är att de är enkla i sin uppbyggnad, det är lätt att kombinera tabeller, hantera data, utföra komplexa sökningar vilket fungerar bra för många typer av strukturerade data [41, s.66].

För att söka efter och hantera data i relationsdatabaser används det standardiserade dataspråket SQL, *Structured Query Language*. För att kunna hantera spatiala data och utföra rumsliga sökningar har OGC även utvecklat en standard SFSQL som utökar SQL [42].

Den vanligaste databashanterare inom GIS är objektrelationella databaser som är en kombination av relations- och objektsdatabas. En objektrelationell databas är precis som de relationella enkelt förklarar uppbyggt av tabeller och relationer mellan tabellerna men kan förutom de primära datatyperna, exempelvis text, heltal och datum, också lagra mer komplexa datatyper och funktioner exempelvis för att hantera spatiala data [43, s.9]. Många stora databaser, både utvecklade med öppen källkod exempelvis PostgreSQL och kommersiella exempelvis Oracle, IBM DB2, och SQL Server, tillhandahåller även tillägg som utvidgar den relationella databasen till en objektrelationell databas för att kunna hantera mer komplexa datatyper [44, s.34].

För att kunna lagra spatiala objekt inom GIS används ofta spatiala databaser som vanligen är en utvidgning av objekt-relationella som har anpassats för att lagra, söka, hantera och analysera geografiska data på ett effektivare sätt [43, s.5]. De stora fördelarna med spatiala databaser är att det både kan utföras snabba sökningar, men även bearbeta data med SQL precis som med relationella databaser. Utöver det tillåter en spatial databas hantering av spatial data och utförandet av spatiala funktioner och analyser på dessa. De spatiala databaserna klarar förutom de vanliga datatyperna även av att hantera geometri som exempelvis punkt, linje och polygon [1, s.172f].

PostgreSQL är den mest använda objekt-relationella databasen med öppen källkod [45]. Databasen är publicerad under licensen "*PostgreSQL License*" som i princip går att jämställa med licensen BSD (se avsnitt 4.4) vilket gör att den kan integreras med proprietär programvara [46]. PostgreSQL skapades ursprungligen som ett projekt för att utveckla en databas med öppen källkod på universitet Berkeley i Kalifornien 1986 men drivs nu av ett community med sponsorer från flera välkända företag som exempelvis Google och IBM. Databasen är väldokumenterad och går att köra på alla de vanligaste plattformarna och en stor fördel är att den går att utöka med det spatiala tillägget PostGIS [47].

PostGIS utökar PostgreSQL med spatiala funktioner och innebär en möjlighet att ställa spatiala frågor med hjälp av SFSQL och indexering med hjälp av R-Tree [48]. PostGIS släpps även den med öppen källkod men distribueras till skillnad från PostgreSQL under licensen GPLv2, (läs mer under avsnitt 4.3.1) [48]. PostGIS är utvecklat i programmeringsspråket C och integrerar med flertalet andra kända bibliotek inom GIS, också utvecklade i C och C++, som exempelvis GDAL och GEOS [49]. PostGIS stöds av många andra programvaror inom GIS som exempelvis ArcGIS, Geoserver och Safe FME [43, s.15f].

4 Öppen källkod

4.1 Licenser och upphovsrätten

Den som skapar ett konstnärligt eller litterärt verk har enligt den svenska lagen om *upphovsrätt till litterära och konstnärliga verk* (SFS 1960:729) 1§ upphovsrätt. Upphovsrätt speciellt för datorprogram beskrivs i 1§ andra punkten. För att avsäga sig delar eller hela upphovsrätten kan utvecklaren ge ut en så kallad avtalslicens enligt 42§. Denna licens ger andra användare rätt att använda och dela ett verk enligt uppsatta villkor i licensen.

4.2 Öppen källkod

Öppen källkod avser program där källkoden är öppen och tillgänglig för andra. Källkoden är den del av ett program som de flesta användare inte kommer i kontakt med utan det är den kod som bygger upp programmet. Om denna källkod ändras kan exempelvis programmets utseende eller funktionalitet förändras [50].

Kopplat till upphovsrätten, i föregående avsnitt, medför öppen källkod ett avsteg från denna upphovsrätt. Det innebär att upphovsmannen avsäger sig rätten till sitt verk. Distributören av öppen källkod producerar dessa produkter utan att nödvändigtvis vara intresserade av att tjäna pengar på distributionen. Syftet kan istället vara att det är viktigare att tekniken fortsätter utvecklas. Öppen källkod medför med andra ord att kod kan återanvändas av flera användare och därför kan fokus istället läggas på utveckling och produkterna som framställs med den utvecklade tekniken [50].

Intresset för öppen källkod har funnits ungefär lika länge som mjukvaruprogram utvecklats. I slutet av 1990-talet sågs dock ett ökat intresse av att kunna dela med sig av källkoden. Till följd av intresseökningen skapades 1998 OSI, *Open Source Initiative*, vilket är en ideell stiftelse som verkar för att sprida kunskapen om öppen källkod. Stiftelsen syftar även till att sprida vikten av öppen källkod som koncept. OSI har tagit fram OSD, *Open Source Definition*, vilken definierar vad öppen källkod är [51].

OSD definierar att "*öppen källkod inte endast innebär att källkoden är tillgänglig*". Källkoden ska, förutom att vara öppen, följa en del andra kriterium. Dessa kriterier definieras enligt OSI i tio punkter, de viktigaste beskrivs nedan [52].

- *Fri vidaredistribution*

Första punkten innebär att licensen inte på något sätt får begränsa vidaredistribution. Det innebär att licensen exempelvis inte ska begära någon avgift om källkoden används som en komponent i ett annat program vars syfte är att säljas.

- *Källkod*

Detta krav innebär att programmet som distribueras måste innehålla källkod och måste tillåta distribution av denna. Om en del inte innehåller källkod måste det istället finnas en väl dokumenterad metod för denna källkod kan nås.

- *Licensen får inte vara specifik för en produkt*

Denna punkt definierar att de rättighet som följer ett visst program inte får bero på att programmet i sig är en del av en viss mjukvara. Om en del av programmet tas bort eller distribueras under villkoren för programmets original-licens ska alla parter som programmet distribueras till ha samma rättigheter som de rättigheter som programmet hade före distribuering.

För att uppfylla kravet på en fri programvara måste en användare även ha alla de nedan nämnda friheterna. Det är av stor vikt att poängtera att en fri programvara inte är likställt med icke-kommersiell. Ett fritt program är även tillgängligt för kommersiell användning och distribution [53].

Frihet noll

- *Använda programmet fritt enligt eget önskemål*

Frihet ett

- *Studera källkoden och mjukvarans funktionalitet och modifiera det enligt eget önskemål*

Frihet två

- *Kopiera och dela programmet vidare till andra användare*

Frihet tre

- *Modifiera och dela programmet vidare i den nya formen*

4.3 Copyleft-licenser

Copyleft är en samlingsterm för ett antal licenser med öppen källkod, exempelvis GPL och LGPL som beskriv närmare i avsnitt 4.3.1 och 4.3.2. Copyleft-licenser är definierade utifrån tidigare nämnda fyra punkter som handlar om vilka rättigheter en användare har för källkoden eller mjukvaran i ett program med en öppen källkod [53].

När en utvecklare distribuerar ett program med öppen källkod så finns det, som tidigare nämnts, alltid en risk att någon utvecklare modifierar programmet och sedan distribuerar den under en kommersiell licens. För att motverka detta och försäkra sig om att ett program med öppen källkod fortsätter vara fri är att distribuera det under den generella licensen copyleft. Alla program där hela eller delar innehåller källkod med en copyleft-licens kommer automatiskt fortsätta att vara fritt och under samma copyleft-licens som den ursprungligen utvecklats under [54].

4.3.1 GNU GPL

General public license, vanligen förkortad till GPL, är en copyleft-licens för så kallad fri programvara. Licensen utvecklades från början av Richard Stallman som en del i GNU-projektet från 1984 för att främja fria programvaror [55].

Licensen är definierad utifrån fyra punkter som varje användare av fri programvara ska ha rätt till (avsnitt 4.2). Detta innebär inte nödvändigtvis att programmet distribueras utan kostnad utan handlar främst om att användaren har rätt att använda, kopiera, modifiera och vidare distribuera kopior av källkoden och programmet utan tillstånd så länge som det fortsatt sprids under samma licensform. Utvecklare kan välja att distribuera det fritt eller mot en avgift för distributionskostnader [56].

Den viktigaste punkten som ska kännas till om GPL är att ett program som distribueras under GPL fortsätter vara fritt under denna licens, det vill säga, det är inte möjligt att modifiera ett program med GPL-licens och distribuera det under en kommersiell licens. Ändringar för privat bruk behöver dock inte göras offentliga för andra. Den som får programmet har rätt till källkoden men även att fritt distribuera denna kod [57].

4.3.2 GNU LGPL

LGPL är en annan GNU-licens. L står för "*lesser*" och ger en antydning om att LGPL inte garanterar full frihet vid användning av programvaran. Licensen ger endast friheten att modifiera koden för komponenter licensierade enligt licensen inte för övriga komponenter.

Största skillnaden mellan GPL och LGPL är att det är tillåtet att i ett nytt program använda och inkludera kod och program under LGPL-licensen utan att det nya programmet blir LGPL. Detta ger en frihet i exempelvis ett kommersiellt syfte där det kan dras nytta av bibliotek eller kod under LGPL utan att nödvändigtvis ändra licensformen på det kommersiella programmet eller bryta mot några licensregler [58].

4.4 BSD

Berkeley Software Distribution, BSD, är en av de vanligaste öppna licenserna som inte är copyleft. Licensformen innebär att utvecklare som vill vidare distribuera sin källkod eller mjukvara gratis för andra användare kan göra detta under denna licensform. Användarna har sedan rättighet att använda, kopiera, modifiera och sprida vidare mjukvaran under en öppen eller kommersiell licens.

Den senaste versionen av Berkeley-licensen, "BSD-3-Clause", definieras sammanfattningsvis enligt tre punkter som användare av källkoden måste uppfylla:

- Om källkoden vidare Distribueras så ska en kopia av ursprungliga villkor och ansvarsfriskrivning bifogas med källkoden.
- Även om källkoden distribueras digitalt måste en lista över ansvarsfriskrivning och villkoren skickas med den digitala kopian eller genereras från den digitala kopian.
- Upphovsrättshavarens namn eller andra personer som bidragit får användas för marknadsföring eller liknande syften utan dennes godkännande [59].

4.5 Apache

Apache är en annan licens för öppen källkod som ges ut av ASF, *Apache Software Foundation*. Som för övriga licenser tillåter Apache användaren att studera, distribuera, modifiera och distribuera den modifierade programvaran. Detta är tillåtet både för privat men även för kommersiellt syfte. De programvaror som släpps av ASF släpps under Apache-licens men det finns även andra programvaror, inte släppta av ASF, som använder licensen [60].

Apaches senaste version 2.0 är kompatibel med GPL version 3. Det innebär att dessa två licenser kan bli kombinerade i en programvara, med förutsättning att det resulterande mjukvaran är av licenstyp GPL3. GPL går med andra ord inte att inkludera i en Apache2-licens [61].

4.6 Jämförelse av licensformer för mjukvaror

I tabell 4.1 visas en jämförelse av tidigare nämnda licenser för att få en inblick i hur de skiljer sig.

Tabell 4.1, Jämförelse av olika licensformer för mjukvaror [62]

Tillåtet att:	GPL	LGPL	BSD	Apache
Använda och studera källkoden	Ja	Ja	Ja	Ja
Kopiera och vidare distribuera källkoden	Ja	Ja	Ja	Ja
Modifiera och vidare distribuera under samma typ av licens	Ja	Ja	Ja	Ja
Modifiera och vidare distribuera under annan typ av licens	Nej	Nej*	Ja	Ja**

* Program eller delar under licensen LGPL kan användas i program under annan typ av licens utan att hela programmet måste släppas under LGPL. De delar som har används med LGPL fortsätter dock att ligga under denna licens.

** Förutsatt att det tillkännages vilka delar som är Apache.

4.7 Creative Commons

Denna licens är en licens för att sprida text och bild. För att förenkla för andra användare att sprida verk, utan att behöva kontakta upphovsrättshavaren kan upphovsmannen redan vid skapandet av sitt verk sätta en slags stämpel för hur denna godkänner att verket distribueras. Ett licenssystem som har skapats för detta syfte är *creative commons*, CC, som är definierat enligt samma fyra friheter som GNU GPL [63].

Dessa friheter har resulterat i den internationella licens CC där varje upphovsman enkelt kan välja hur verket ska spridas. Det kan spridas enligt fyra kategorier [64]:

CC BY - Erkännande. Denna licensform innebär att licenstagaren får distribuera, kopiera och bearbeta verket endast om erkännande ges till upphovsmannen.

CC SA - Dela lika. Om bearbetningar görs får dessa bearbetningar endast distribueras vidare under en licens som inte är mer restriktiv än original-licensen.

CC ND - Inga bearbetningar. Denna licensform tillåter endast distribution, kopiering och visning av originalet. Inga bearbetningar får göras.

CC NC - Icke-kommersiellt. Här är samma förutsättningar som för CC BY men det inte för kommersiella ändamål.

Dessa fyra grundvillkor kan kombineras i sex möjliga kombinationer [65]. Ett välkänt exempel på tjänst som släpper under creative commons-licens är *OpenStreetMap* som släpper sina kartor och sin dokumentation under denna licens [66].

4.8 Fördelar och nackdelar med öppen källkod

Öppen källkod är ett brett och omdebatterat område både för GIS-produkter men även för andra mjukvaror och det förekommer mycket åsikter om och hur öppen källkod ska användas och som för alla debatter förekommer en rad för- och nackdelar. Nedanstående avsnitt berör diskussionen kring användandet av öppen källkod och kommersiella produkter.

Det finns en rad olika organisationer som verkar för användandet och utvecklingen av öppen källkod, däribland OSGeo som har stöd av ett stort antal utvecklare som jobbar med produkterna. Organisationer som dessa menar på att utveckling av mjukvaror blir bäst då de utvecklas med öppen källkod [67]. En kommersiell programvara däremot skiljer sig från programvaror med öppen källkod på fler områden än att de kommersiella tjänsterna har ett vinstsyfte. En tjänst med öppen källkod måste förlita sig på att någon person eller organisation, indirekt utan ersättning, måste implementera nya delar för att utvecklingen ska gå framåt. En kommersiell tjänst bör vara vinstdrivande, därför ligger det i utvecklarens intresse att vara uppdaterad så att tjänsten fortsätter användas. En annan skillnad mellan de kommersiella tjänsterna och de med öppen källkod är tillgången till support och hjälp menar Singh, Bansal och Jha [68]. Användare som betalar för en tjänst förväntar support och hjälp om programvaran

inte motsvarar de förväntningar som finns. En tredje och betydelsefull aspekt är säkerheten på programvaran. Vissa programvaror med öppen källkod förhandsgranskas inte innan de släpps, vilket ett kommersiellt bolag gör för sina produkter. Detta kan medföra säkerhetsrisker för användare av program med öppen källkod.

I en rapport från CASCADOSS, som är en organisation som verkar för utveckling av öppen källkod för GIS-applikationer, tas det upp ett antal punkter som är viktiga att tänka på när en programvara eller ett system med öppen källkod ska utvärderas gentemot ett kommersiellt alternativ. För eventuell användning av öppen källkod i ett företag eller verksamhet bör det exempelvis beaktas vad det är för funktioner som eftersträvas. Möjlighet till support är en annan aspekt då kommersiella alternativ ofta erbjuder en support som en del av avtalet. I rapporten framförs det att en del projekt med öppen källkod också erbjuder support men i många fall innebär öppen källkod att användaren blir beroende av att projektet innehåller kompetenta utvecklare som för projektet framåt och delar med sig av kunskapen i forum eller liknande [69, s.32].

Kostnaden är en viktig faktor i val av programvara eller system och är i många fall den avgörande faktorn för beslutet. Kostnadsaspekten för en kommersiell programvara är licenskostnaden. Med öppen källkod innebär kostnaden att de distributionsvillkor som följer med den öppna källkoden stämmer överens med verksamhetens syften, mål och ambitioner. En del öppna källkodlicenser kräver att ändringar i källkoden distribueras vidare. I sådana fall måste det övervägas vilken inverkan det har på verksamheten. En annan kostnad som riskerar att följa med den öppna källkoden är om licensen kräver vidareutveckling och distribution av denna [69, s.33]. Ebert [70] menar också på att det är ett vanligt misstag att tro att programvaror med öppen källkod är billigare än kommersiella alternativ. Ebert betonar att det i många fall stämmer för de kortsiktiga kostnaderna men inte nödvändigtvis för de långsiktiga. De långsiktiga kostnaderna består bland annat i utvärderingskostnader, där organisationerna måste lägga pengar på att förstå de olika projekten och ta ställning till vilket projekt som passar verksamheten bäst. Andra faktorer som måste beaktas är till exempel licenstyp men även vilka förutsättningar det finns på företaget för underhåll och vidareutveckling av projektet [70].

Steiniger och Bocher skriver att problem för användare på grund av så kallade strategiska beslut av producenter av kommersiella GIS-program är kända men även svåra att påvisa. Författarna menar på att de stött på problem hos kommersiella aktörer i exempelvis förlust av support för gamla komponenter om användaren inte byter till nya tillgängliga komponenter. Ett annat problem som identifierats med kommersiella alternativ är förändringar av licenskostnader eller i reglerna kring distributionen av licensen, vilket innebär att producenten exempelvis höjer licenskostnaden eller begränsar antalet användare per inköpt licens. En sådan restriktion i användandet är inte möjligt med licenser med öppen källkod [62, s.17]. Ven, Verelst och Mannaert [71] skriver att en del organisationer därför väljer att använda öppen källkod för att känna en större frihet. Att använda kommersiella produkter innebär att en organisation blir beroende av dess produkt och den service som erbjuds. Eventuella felkorrigeringar av programvaran kan vara långsam och om det önskas ytterligare funktionalitet ligger ansvaret hos den kommersiella aktören. Författarna skriver att i princip vem som helst kan erbjuda stöd och support för programvaror med öppen källkod. Stora projekt med öppen källkod stöds ofta av en

stor organisation och författarna menar därför på att det ökar tillgängligheten till support och på så sätt minskar beroendet av en enda aktör. Författarna trycker dock på att användningen av öppen källkod inte ger total frihet eftersom projekt med öppen källkod trots allt är begränsade i antal [71].

Steiniger och Bocher, [62, s.17] betonar även att det finns fler fördelar med öppen källkod förutom avsaknaden av licenskostnaden. Fördelarna anses speciellt stora om projekten är omfattande, med ett stort stöd bakom de öppna standarderna, vilket innebär en viss trygghet för användaren om svårigheter uppstår. Det är även till stor fördel om licenstyperna medger fri användning av programvaran eller koden. Med fri användning menas i detta fall att programvaran eller koden får användas hur som helst, både privat men även för kommersiella aktörer. Som en motsättning till friheten menar Singh, Bansal och Jha [68] på att utveckling med öppen källkod att den friheten öppen källkod medför också innebär att ett stort ansvar hos utvecklaren. Ansvar kan med andra ord innebära att öppen källkod inte blir för alla utan endast är tillgängligt för dem med kompetens inom programmering.

Det finns en viss risk att organisationer som är intresserade av att sänka sina licenskostnader integrerar programvaror med öppen källkod i sina verksamheter utan att tänka på de så kallade dolda kostnaderna skriver Nagy, Yassin och Bhattacharjee [72]. Författarna diskuterar olika barriärer som en organisation kan känna inför valet att integrera öppen källkod i sin verksamhet. Författarna beskriver bland annat att det i organisationer upplevs en kunskapsbarriär angående öppen källkod. Trots alla dessa fördelar med användningen är det fortfarande ett stort antal organisationer som betalar stora licenskostnader för en kommersiell programvara vilket författarna främst menar beror på att organisationerna inte vet att dessa alternativ finns för just den egna organisationens användningsområde. I andra fall kan kunskapen om dessa program finnas inom organisationen men kunskapsgapet består då av okunskap för själva användningen av dessa produkter med öppen källkod. Det kan även saknas kunskap i hur dessa produkter kan integreras och anpassas till de egna tillämpningarna. En lösning på denna kunskapsbarriär skulle för många organisationer kunna bestå av en övervakning av öppen källkod, för att på så sätt hållas uppdaterade om vilka projekt som finns. Även utbildning av personal skulle minska denna kunskapsbarriär. En idé som författarna lyfter fram är att organisationer kan överlämna utveckling och underhåll av produkterna med öppen källkod till en konsult för att på så sätt få en viss säkerhet och lösningen blir då mindre personberoende [72].

En användare som övergår från kommersiella produkter till öppen källkod ska inte förvänta sig att det är helt gratis [62, s.17]. Utveckling och användning av öppen källkod medför kostnader i form av utbildning och utveckling istället och kräver ett specifikt kunnande hos utvecklaren. Steiniger och Bocher skriver även att en annan nackdel med öppen källkod är att projekten ibland saknar kontinuitet och långsiktig planering. En risk med öppen källkod är att intresset för projektet avtar och utvecklingen avstannar, vilket kan medföra problem för en verksamhet som använder den öppna källkoden. Slutligen skriver författarna att det i vissa projekt med öppen källkod tillhandahåller bristande dokumentation av projektet, då en del utvecklare innehar ett större intresse i själva utvecklingen än för dokumentationen för andra utvecklare. Författarna menar på att det för utvecklaren saknas egen vinning, i form av exempelvis meriter för arbete eller liknande, att dokumentera vilket leder till att dokumentationen ofta blir

bristfällig. Problemen som en bristande dokumentation leder till är att det blir otillgängligt och svårhanterat för andra utvecklare, speciellt nybörjare. För att vidareutveckling ska kunna ske är en god dokumentation av källkoden med andra ord avgörande [62, s.17].

En annan debatt kring att integrera programvaror med öppen källkod i verksamheter är säkerheten med avseende på öppen källkod. Säkerhetsaspekten måste hanteras mer noggrant om öppen källkod ska användas gentemot kommersiella programvaror eftersom dessa betraktas som mindre säkra. Enligt Damiani, Ardagna och Ioini, [73, s.98] har debatten inte lett till någon slutgiltig slutsats och menar på att det är högst osannolikt att den gör det inom en snar framtid. En av de viktigaste frågorna i debatten om säkerheten kring öppen källkod behandlar faktumet att källkoden synliggörs för alla. Eftersom källkoden är synlig för alla innebär det också i vissa fall att alla säkerhetsegenskaper programvaran innehar är synliggjord, vilket innebär att både utvecklare men även andra aktörer som inte nödvändigtvis har gott syfte. Tillgången till koden kan medföra att en angripare av programvaran kan upptäcka och på så sätt utnyttja säkerhetsbrister i programvaran. Däremot, i motsats till säkerhetsrisken, medför synligheten av koden även att så kallade försvarare av koden har kontroll över den. Öppen källkod backas vanligen av ett stort community av utvecklare som värnar om den öppna källkoden och verkar för vidareutvecklingen. Dessa utvecklare kan också dra nytta av tillgången till koden och söka efter och identifiera säkerhetsbrister i koden och på så sätt använda bristerna till att förstärka säkerheten av programvaran. Generellt innebär det stora antalet utvecklare att säkerheten blir stor med öppen källkod då koden ofta först granskas av en annan utvecklare och sedan även genomgår en valideringsprocess [73, s.98f].

Eftersom det finns en rad fördelar med öppen källkod, är det intressant att undersöka varför det inte är fler organisationer som integrerar öppen källkod i sina verksamheter. Goode skriver [74] att det ofta finns ett stort antal företag och organisationer som har funderat på att integrera öppen källkod i sina verksamheter. Det är dock bara ett litet antal av dessa organisationer som faktiskt tagit tag i saken och integrerat den öppna källkoden i verksamheten. Goode skriver att detta till största del beror på att många företag inte vet vilka projekt med öppen källkod som finns tillgängliga för just deras tillämpningsområde. Den andra stora anledningen till att vissa organisationer inte tar sig an öppen källkod menar Goode beror på att många företag saknar stöd från ledningen. I många fall kan ledningen se en risk i att mista supporten som följer de kommersiella produkterna och att supporten i sådana fall måste skötas inom den egna organisationen. Goode skriver också att två andra barriärer som kan upplevas för företag att integrera öppen källkod i sina verksamheter är tidsbrist och brist på kompetens. Organisationer kan uppleva att de inte innehar den kompetens på arbetsplatsen som det krävs att använda och utveckla med öppen källkod. I många fall finns det inte heller tid avsatt för denna typ av utveckling eftersom det inte finns någon stöttning från ledningen. Goode menar på att en förutsättning är att det finns tydliga riktlinjer från ledningen för att utveckling och användning ska vara möjlig och lyckad i en organisation [74].

I tabell 4.2 har det gjorts en sammanställning av fördelar och nackdelar för kommersiella samt programvaror med öppen källkod. Tabellen utgår från tidigare litteratur i avsnittet och sammanställer de viktigaste punkter som berörs.

Tabell 4.2, Jämförelse av olika öppen källkod och kommersiella programvaror

	Öppen källkod		Kommersiella produkter	
	Fördelar	Nackdelar	Fördelar	Nackdelar
Support	Hjälp finns i form av forum i stora projekt	Kräver att det finns ett stort community	Ingår oftast i avtalet	Riskerar att försvinna på äldre versioner av produkten
Dokumentation	Oftast väldokumenterat	Mängden och kvaliteten varierar		
Kostnad	Licenskostnad undviks	Kräver pengar i form av tid för utveckling och underhåll	Undviker utvecklingskostnader	Riskerar att höjas
Funktionalitet	Kan skapa den funktionalitet som önskas	Kräver att utvecklaren håller produkten uppdaterad	Stora utvecklare vet vad branschen efterfrågar	Funktioner kan bli för generella så egna anpassningar måste göras
Uppstartstid	-	Kräver en del utvecklingstid innan användning	Kan användas direkt	-
Kompetenser	-	Kräver hög kompetens och mycket tid	Kräver inte lika hög kompetens	-
Säkerhet	Stora projekt valideras av ett community	Säkerhet varierar mellan projekt	Ett företag står bakom produkten och är ansvariga	-

5 Kartverktyg

Nedan presenteras ett antal kommersiella kartverktyg samt kartverktyg med öppen källkod som berörs och används i denna rapport. Inspiration i urvalet har främst hämtats från OSGeo, *Open Source Geospatial Foundation*, vilket är en organisation som verkar för att tillhandahålla, stötta och främja utvecklingen av spatial teknologi med öppen källkod och listar de vanligaste projekten med öppen källkod [75]. Valda programvaror i avsnittet är även presenterade utifrån vilka programvaror som använts på de intervjuade organisationerna i fallstudien (kapitel 7 och 8). Avsnittet följer strukturen i figur 1.1 och utvärderar programvaror för publiceringsprocessen och inleds med programvaror för bearbetning följt av server- och klientprogram.

5.1 Verktyg för bearbetning av geografiska data

5.1.1 FME

Feature Manipulation Engine (FME) är ett ETL-verktyg, lanserat av *Safe Software* som används för att bearbeta geografiska data [76]. Programvaran kan exempelvis konvertera geografiska data mellan olika referenssystem, transformera data mellan olika filformat eller behandla data på flertalet andra sätt med hjälp av alla de funktioner som programmet erbjuder. ETL-verktyget FME är ett kraftfullt verktyg för att kunna skapa, transformera och extrahera data för att uppnå de krav på data som ställs av användaren [77].

FME kan hantera över 350 olika format, där programmet till stor del är flödesbaserat. Det innebär att användaren länkar relationer mellan läsare, konverterare och skrivare i programmet för att manipulera, konvertera och transformera data till önskat format. Flödesstrukturen medför en väldigt överskådlig bild för användaren som snabbt kan hantera stora mängder data utan att behöva skriva stora mängder kod [78].

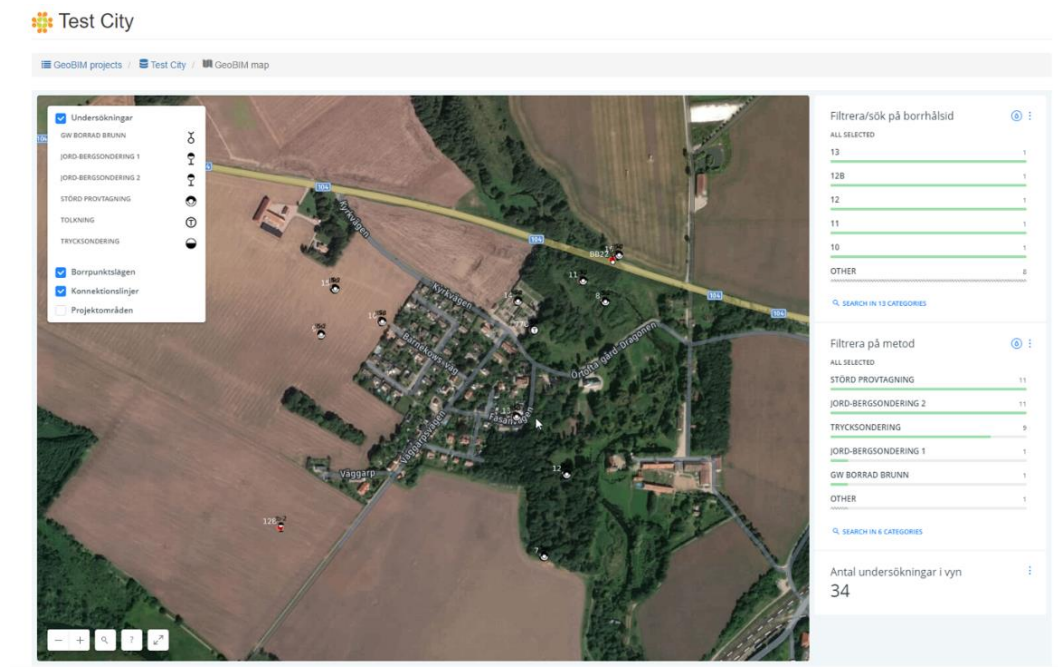
5.2 Kommersiella program för server- och klientsidan

5.2.1 Carto

Fallstudien i denna rapport använder funktionaliteten i Carto som utgångspunkt i jämförelsen av programvaror med öppen källkod. Denna programvara är ett verktyg för att publicera sina data i en webbkarta och är både ett server- och klientprogram, och som används av många företag däribland Tyréns. Carto är en tjänst där användaren prenumererar per företag i molnet.

För att stilsätta data och lager som ska användas i kartan krävs en del tidskrävande manuella moment. För att stilsätta en karta i Carto används Cartos eget kodningsspråk, CartoCSS. CartoCSS är en variant av CSS, *Cascading Style Sheets*, som är en stilsättningsmall som beskriver hur kartan ska presenteras. Koden definierar exempelvis teckensnitt, teckenstorlek

eller färg som ska användas. I avsnitt 9.1 presenteras en kravspecifikation, mestadels framtagen utifrån de funktioner och behov som Carto tillfredsställer i det tillämpningsområdet som det används för idag, det vill säga i tjänsten GeoBIM [79]. I figur 5.1 visas en karta producerad i Carto utifrån ett antal testpunkter.



Figur 5.1 En egenproducerad karta producerad i Carto för ett testprojekt i GeoBIM, data given av Tyréns

5.2.2 ArcGIS Online

En av de mest använda programvarorna inom GIS är ArcGIS som är utvecklat och producerat av företaget ESRI. ESRI, som är ett av de största företagen inom geografisk information, erbjuder även plattformen ArcGIS online som används för publicering av kartor på webben. Programmet bygger, precis som tidigare beskrivna gränssnitt, på JavaScript och det finns även här möjlighet att importera och använda ett stort antal bakgrundskartor efter vilket projekt som skapas.

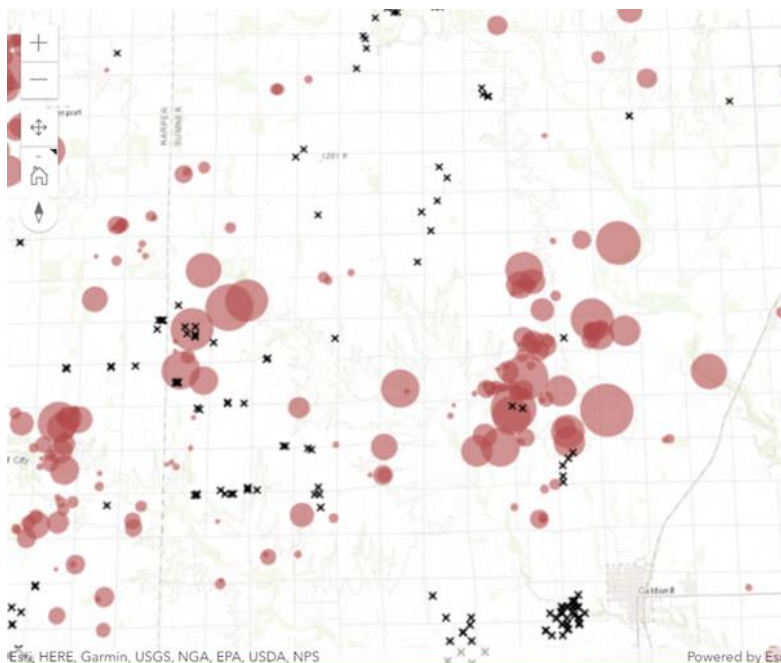
ArcGIS online erbjuder ett antal olika tekniker och möjligheter att utveckla ett effektivt och väl fungerande gränssnitt på de flesta plattformar med anpassad design efter vilken produkt användaren har. Gällande publicering av webbkartor så kan användaren, precis som för tidigare program, publicera kartor på webben med hjälp av HTML, JavaScript och CSS. ArcGIS online är liksom Carto ett verktyg som består av både ett klient- och serverprogram.

Arcgis API, *Application programming Interface*, för JavaScript finns på alla de vanligaste förekommande operativsystemen som exempelvis Windows, iOS, Android, MacOS och Linux [80]. En smidig funktion som går att utveckla med hjälp av ArcGIS API för JavaScript är en kartvy där användaren enkelt med ett knapptryck kan scrolla från en 2D karta till 3D-vy till

exempel för att visa borrhål för undermarksdata. Figur 5.2 nedan är ett exempel på en karta med undermarksdata där användaren enkelt kan välja att växla mellan en “vanlig” 2D-karta, och som visas här, en 3D-vy över borrhålen, det går även att rotera vyn i 360 grader i 3D för olika snedställda projektioner [81].



Figur 5.2 Exempel från ESRI's Sandbox över borrhål i 3D utvecklat med ESRI's API för JavaScript [81]



Figur 5.3 Exempel på samma karta som i figur 5.2, men här visad som en 2D-karta [81].

5.2.3 Mapbox

Ett annat gränssnitt för att visualisera geografisk information på internet är Mapbox. Mapbox erbjuder ett antal verktyg för att integrera geografiska data i hemsidor, appar och andra webbaserade applikationer på ett snyggt sätt. Mapbox använder delvis ett bibliotek av JavaScript med öppen källkod men har också vidareutvecklat andra delar för högre funktionalitet [82].

Mapbox startade som ett företag helt baserat på öppen källkod 2010 och fortfarande utvecklas många av deras moduler med öppen källkod men en del plattformar som används för stilsättning och publicering av kartor, likt Mapbox studio, är ett kommersiellt verktyg. Mapbox fungerar som en utvecklingsplattform för många nya programmoduler som släpps under licensen BSD som sedan kan användas av utvecklade företag i hela branschen [83][84][85].

Mapbox är också en stor tillverkare och distributör av många olika typer av kartor som går att specialdesigna och stilsätta efter syfte och tjänst. Deras kartor går även att bädda in i många andra typer av gränssnitt [83].

5.3 Serverprogram med öppen källkod

Ordet server kan syfta på många olika begrepp inom IT men innebär i grunden ett system som hjälper en så kallad klient och kan exempelvis vara en webbserver som tar emot förfrågningar från HTML-sidor via en webbläsare. I avsnitt 5.3 beskrivs de vanligaste plattformar inom GIS med öppen källkod som tillhandahåller spatiala data med hjälp av WMS och WFS.

5.3.1 Geoserver

Geoserver är en fri mjukvara, det vill säga en server som både är gratis men även innehåller öppen källkod. Denna klientserver tillåter användarna delning, bearbetning och redigering av spatiala data. Geoserver erbjuder visualisering av spatiala data på alla de populära webbaserade kartorna, så som Google Maps eller OpenStreetMap, men också egna data. Geoserver kan även en kopplas till andra GIS-programvaror som till exempel ArcGIS.

Målet med Geoserver är att fungera som en sammankopplande punkt inom en infrastruktur med öppen källkod. Det kan dras en liknelse till HTTP-server vilken är en server som tillåter fri publicering av HTML. GeoServer syftar till att göra detsamma för geografiska data. GeoServer är kompatibel för många av OGC-standarderna så som exempelvis WMS, WFS och WCS. Geoserver ligger under licensen GNU GPL [86].

5.3.2 Mapserver

En annan klientserver med öppen källkod är MapServer, vilken skapades i mitten av 90-talet på *University of Minnesota – Twin Cities* USA. Projektet, vilket är en plattform för att publicera geografiska data på webben, grundade sig i att NASA behövde ett sätt för att göra sina satellitbilder tillgängliga för allmänheten. MapServer-projektet blev ett av grundprojekten till organisationen OSGeo [87].

MapServer distribueras under licensen MIT, som är liknande licensen BSD (avsnitt 4.4), via GitHub, en plattform för att dela och ladda upp källkod. MapServer erbjuder en avancerad webbkarta med en rad olika funktioner och stilsättningsmöjligheter. Plattformen stödjer även de flesta OGC-standarderna och är kompatibel med de vanligaste operativsystemen. MapServer innehar en god kompatibilitet med de andra komponenter av öppen källkod såsom exempelvis GDAL (ett verktyg för transformering av data [88]) och PostGIS [89].

5.3.3 QGIS-server

En tredje server med öppen källkod är QGIS-server, vilken används av exempelvis Kristianstad kommun, (avsnitt 7.5 och 8.3). Eftersom QGIS-desktop och server använder samma visualiseringsbibliotek medför det en hög kompatibilitet mellan dessa komponenter vilket innebär att kartorna innehar samma utseende när de publiceras på webben [90].

5.4 Klientprogram med öppen källkod

Intresset för öppen källkod är stort. Dels ur kostnadsperspektiv men även den stora flexibiliteten att designa ett eget system för sitt specifika ämnesområde gör det extra intressant. Ett dilemma kring öppen källkod är att vissa av de programvaror och gränssnitt som distribueras med öppen källkod har vissa begränsningar i licensen. I kravspecifikationen för visualiseringssteget under de tekniska kraven (avsnitt 9.1.2 punkterna 1–5) definieras några av de kraven som bör övervägas ur licenssynpunkt, i den egna implementationen.

Nedanstående avsnitt kartlägger de vanligaste tjänsterna med öppen källkod för att publicera geografiska data på webben. Tjänsterna som granskas i avsnitt 5.4 är de som bäst uppfyller grundläggande förutsättningar för ett kartverktyg utifrån en diskussion med Tyréns.

Kartverktyget ska kunna:

1. Publicera en webbkarta
2. Ska publiceras under öppen källkod
3. Stödja operativsystemet Windows
4. Stödja grundläggande funktionalitet som exempelvis zooma, egen stilsättning av data etcetera.
5. Importera bakgrundskartor, exempelvis ortofoto

6. Importera vektordata i nya lager

7. Stödja de vanligaste formaten och standarder för branschen

Förutsättningarna är framtagna för att kunna sälla bland alla de verktyg som finns för att publicera geografiska data på webben. Inspiration i urvalet har främst hämtats från OSGeo, vilka listar de vanligaste projekten med öppen källkod [75]. Ovanstående förutsättningar framkom även ur de förhållande som fallstudien gjordes under. Syftet är att hitta en alternativ programvara med öppen källkod, vilket mynnade ut i punkt ett, två och tre. Produkten ska vara en interaktiv karta som följer vanliga standarder, vilket i sin tur formade de fyra sista punkterna. Det har dessutom valts att studera kartverktyg som används av Malmö stad, T-kartor, Ystad kommun, Kristianstad kommun och Tyréns.

5.4.1 OpenLayers

OpenLayers är ett gränssnitt bestående av ett relativt stort bibliotek av JavaScript under licensen BSD 2-clause, vilket gör att koden kan integreras i proprietär programvara. Källkoden finns fritt tillgänglig för nerladdning från GitHub. Detta gränssnitt används främst för publicering av interaktiva och dynamiska kartor på webben. OpenLayers bibliotek är objektorienterat där objekt skapas av klasser definierade i OpenLayers API. Exempel på objekt är map, layer, style och liknande. OpenLayer tillhandahåller en katalogstruktur i sitt bibliotek där alla objekt och tillhörande funktioner först definieras som en huvudklass (ol.) följt av underklasser till denna [91].

OpenLayers har ett stort utbud av funktioner som gör det till ett relativt komplett kartverktyg för att skapa karttjänster men kräver också att utvecklaren är väl insatt i användningen av JavaScript. Det är ett väldigt flexibelt gränssnitt då det har support för många GIS-standarder och integrerar väl ihop med många plugin-bibliotek från andra utvecklare som exempelvis Google Maps och Mapbox. Det stödjer import av bakgrundskartor från tjänster som OSM, Bing och Mapbox. OpenLayers har ett väldokumenterat bibliotek över funktionerna med många praktiska exempel på tekniker för att skapa dynamiska och interaktiva webbkartor [92] [93]. OpenLayers går att köra på plattformarna Linux/Unix, Mac OSX, Windows, Android, IOS och stödjer format som GML, KML och GeoJSON [94].

I takt med att tekniken utvecklas ställs allt högre krav på att kunna visualisera geografiska data även i 3D. Eftersom OpenLayers är byggt på JavaScript är det lätt att integrera med andra bibliotek exempelvis har OpenLayers stöd för Cesium och Three som är två andra JavaScript-bibliotek. Cesium används främst för att visualisera kartan i 3D och Three för att skapa animerade objekt i 3D-miljön, det skulle exempelvis kunna vara cylindrar som illustrerar borrhållslägen i förhållande till marknivån [95] [96] [97].

5.4.2 Leaflet

Ett annat frekvent använt JavaScript-bibliotek med öppen källkod för interaktiva kartor är Leaflet som distribueras fritt under en BSD-licens. Leaflet används som byggsten i många olika tillämpningsområden exempelvis i sMap (beskrivs i avsnitten 7.5 och 8.3) [98].

För att använda Leaflet krävs en viss kunskap om JavaScript. Fördelen med Leaflet är dock att det finns en stor och omfattande dokumentation, och ett bibliotek av funktioner som användaren kan nyttja. Leaflets bibliotek av JavaScript är endast på 38 KB vilket är betydligt mindre än många andra bibliotek, vilket gör det effektivt men också att funktionaliteten kan bli något begränsad i mer komplexa projekt. Det väldokumenterade biblioteket underlättar användningen och medför till viss del att Leaflet anses vara lättare att använda som nybörjare jämfört med exempelvis OpenLayers. Leaflet innebär en stor frihet och flexibilitet eftersom Leaflet är ett JavaScript-bibliotek är det lätt att integrera med andra funktioner och bibliotek skrivna i JavaScript [99].

5.4.3 Open ePlatform

Open ePlatform utvecklades som en del av ett EU-projekt, RIGES, *Regional Innovativ GIS- och E-tjänst Samverkan*, mellan ett antal kommuner i mellersta Sverige. RIGES är ett projekt vars mål var att göra det möjligt för bygglovsansökningar på internet. Utöver bygglovshantering strävas det efter att underlätta informationssökning och användning av de berörda kommunernas kartverktyg. För att möta behovet av dessa elektroniska tjänster utvecklades Open ePlatform i Java med hjälp av ramverket OpenHierarchy [7].

Tjänsten är främst till för publicering av kartor i webbläsaren och stödjer andra kända format, standarder och komponenter med öppen källkod som exempelvis OpenLayers, WMS, WFS och GeoJSON [100].

5.4.4 MapGuide

MapGuide är en annan plattform med öppen källkod. Denna webb-plattform möjliggör för utveckling och distribution av webbkartor och andra GIS-analyser och beräkningar i en webbmiljö. MapGuide stödjer de flesta filformat, databaser och standarder [101]. MapGuide går under licensen LGPL sedan den släpptes 2005, och har även varit en del av Open Source Geospatial Foundation (OSGeo) sedan 2006 [102].

5.4.5 QGIS – QGIS Web Client

Ett av de ledande och mer utvecklade programmen med öppen källkod för geografisk information är QGIS. Programmet har flera applikationer för olika delsyften och innehar en god

kompabilitet mellan varandra. Dessa delapplikationer är exempelvis QGIS Server, QGIS Desktop och QGIS Web Client. QGIS Desktop används för att skapa, bearbeta, analysera och publicera geografiska information och QGIS server möjliggör användaren att koppla upp till en visningstjänst (WMS).

QGIS Web Client är en applikation för att kunna publicera data från QGIS Desktop och andra källor på webben. En av fördelarna med QGIS Web Client är att data och lager som har tagit fram i QGIS Desktop enkelt kan integreras till webbklienten utan att data behöver omarbetas allt för mycket. Alla QGIS-projekt släpps under licensen GNU GPL (avsnitt 4.3.1) [103].

6 Fallstudie

6.1 Upplägg

Fallstudien har delats upp i tre delar, där de två första delarna består av en intervjustudie och den tredje av en egen implementation av publiceringsprocessen.

Intervjustudien har behandlat två huvudfrågor. Den första huvudfrågan presenteras i kapitel 7, där de intervjuade parterna har svarat på frågor kring organisationens publiceringsprocess. Detta görs dels för att se om det finns effektiviseringsmöjligheter i flödet men också för att få en bild av vilka olika komponenter som används för webbkartor. På så sätt kan det utvärderas till hur stor del öppen källkod används i praktiken. Den andra huvudfrågan behandlar hur dessa organisationer ser på fördelar och nackdelar med användning och utveckling med öppen källkod, vilket presenteras i kapitel 8.

Intervjustudien används även som underlag till del tre av fallstudien, en egen implementation. Fallstudie del tre redovisar även en kravspecifikation gjord utifrån material från intervjustudien. Kravspecifikationen används för utvärdering av den egna lösningen.

6.2 Val av respondenter för intervjustudie

Inför intervjustudien utvärderades vilka kommuner och företag som skulle vara av intresse för studien. Det fanns ett intresse från Tyréns sida att jämföra företag, gentemot kommuner för att utreda om skillnader finns i tankarna kring öppen källkod mellan kommersiella aktörer och offentliga verksamheter. Det valdes ett företag, T-kartor i Kristianstad, för att jämföras mot Tyréns. Resterande respondenter representerar tre kommuner, av olika storlek och användande av öppen källkod. Utifrån dessa aspekter resulterade de intervjuade kommunerna i Malmö stad, Kristianstad kommun samt Ystad kommun. I nedanstående kapitel presenteras en sammanställning av samtliga intervjuer som genomförts.

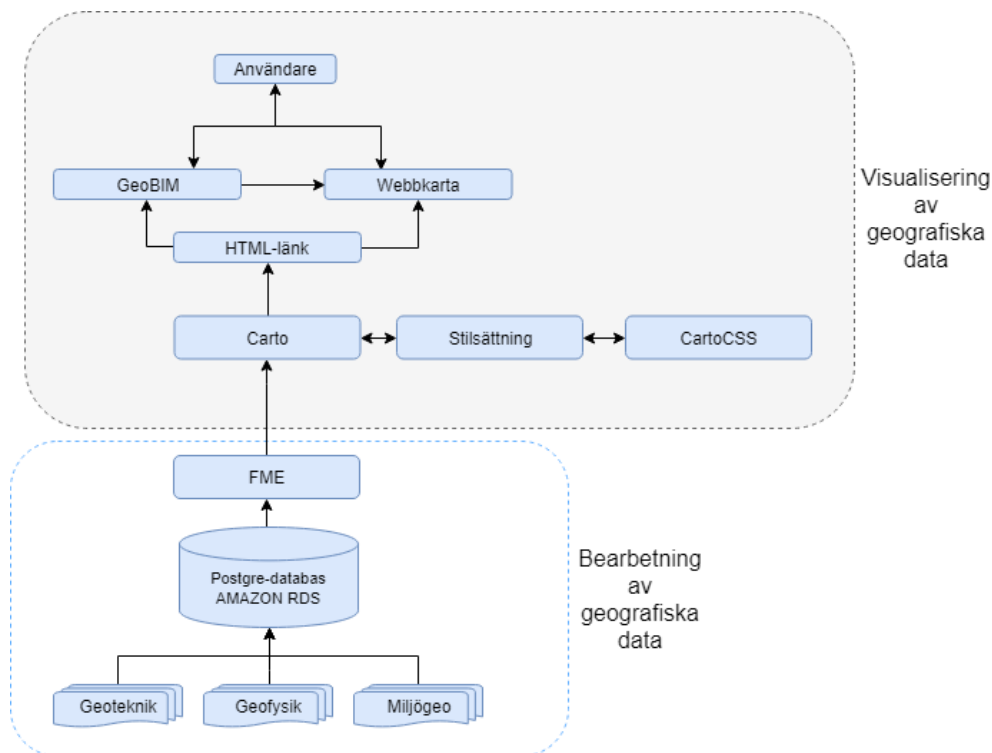
7 Fallstudie del I - Publiceringsprocesser

7.1 Bakgrund

För att kunna jämföra hur väl Carto fungerar idag gentemot andra lösningar har en kartläggning över olika publiceringsprocesser gjorts genom en intervjustudie. Intervjuade parter är Tyréns, Malmö stad, Kristianstad kommun, Ystad kommun och T-kartor i Kristianstad. Bakgrunden till denna del av intervjustudien är att visa på hur de olika parternas publiceringsprocesser går till, vilka lösningar de har valt samt vilka program och system som används för att publicera geografiska data på webben. Tanken är också att kartlägga varför olika lösningar har valts för att kunna utvärdera Carto som publiceringsverktyg.

7.2 Publiceringsprocess - Tyréns AB

För att kunna utvärdera möjliga alternativ till Carto genomfördes en kartläggning över publiceringsprocessen för GeoBIM-portalen hos Tyréns. I figur 7.1 visas ett översiktligt flödesschema över hur data i ett normalt GeoBIM-projekt publiceras på webben. GeoBIM hanterar data främst från de tre områdena geoteknik, geofysik och miljögeoteknik. Rådata, som kommer in från en användare av GeoBIM-tjänsten, laddas först upp till en PostgreSQL-databas via portalen i GeoBIM. Nuvarande lösning använder en relationsdatabas utan det spatiala tillägget PostGIS men även en spatial databas hade varit möjlig. En spatial databas hade kunnat hantera geometri på ett annat sätt, se mer i avsnitt 3.6.



Figur 7.1 Flödesschema för publiceringsprocessen i GeoBIM på Tyréns

Tyréns använder ett egenutvecklat standardskript i FME för att behandla data från databasen till en lämplig utformning för att kunna publicera i en karta. Data som har behandlats exporteras sedan vidare med hjälp av en så kallad skrivare i FME som ett färdigt dataset till Carto. Arbetet i FME är idag automatiskt när standardskripten väl har utvecklats.

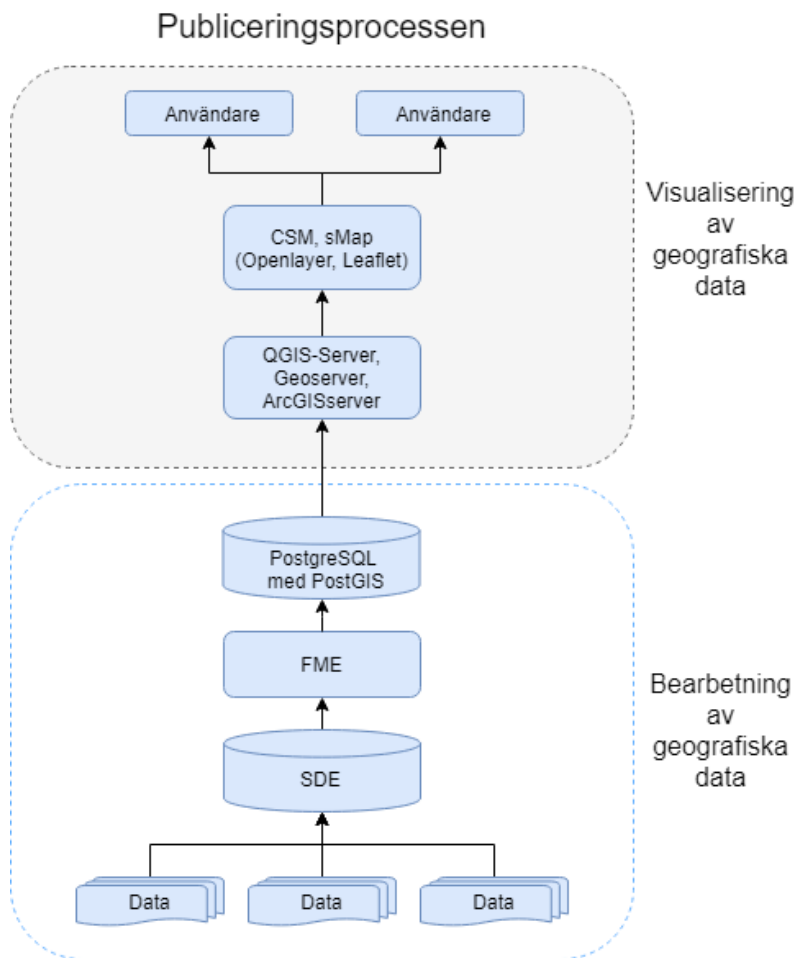
Carto (avsnitt 5.2.1) läser sedan in datasetet vilket går att stilsätta efter eget önskemål. Stilsättningen görs antingen genom färdiga funktioner i Carto eller med hjälp av egna skript i Carto-CSS. Stilsättningen är ett tidskrävande arbete och utformas idag manuellt men har effektiviserat med hjälp av egenskivna skript för stilsättningen för de olika lagren. Det hade dock varit önskvärt med en mer automatiserad process, med standardval för olika typer av data för att göra detta steg mer automatiserat.

För att få en färdig produkt används främst bakgrundskartor från Cartos standardutbud, till exempel ortofoto men Carto stödjer även import av egna bakgrundskartor bland annat med hjälp av tjänster som WMS och WMTS. Om standardfunktionerna inte är anpassade till det tillämpningsområde som kartan ska användas till finns det möjlighet för att bygga egen funktionalitet i kartan med hjälp av Carto JavaScript.

Slutligen kan den färdiga kartan publiceras med hjälp av en HTML-länk, som exporteras till GeoBIM-portalen eller till kunden direkt. Carto har flertalet inställningar för att begränsa rättigheterna till kartan, den kan publiceras publikt, via länk med lösenordskyddad inloggning specifik för en karta eller privat för redigeringsbehöriga. Säkerhetsaspekten har stor vikt i valet av lösning då kundernas data kan vara hemlig.

7.3 Publiceringsprocess - Malmö stad

Stadsbyggnadskontoret på Malmö stad använder idag främst Esri:s produkter som ArcGIS för att hantera primärkartan. För att anpassa, transformera och manipulera data till den aktuella tjänsten och det specifika området används främst ETL-verktyget FME (avsnitt 5.1.1) för att exportera data från Esri:s geodatabas till PostGIS. Servrarna som används för visualiseringen är ArcGIS-server, Geoserver och QGIS-server. ArcGIS-server används för primärkartan och Geoserver främst för sMap-projektet. I figur 7.2 illustreras ett flödesschema för publicering av geografiska data på stadsbyggnadskontoret på Malmö stad.

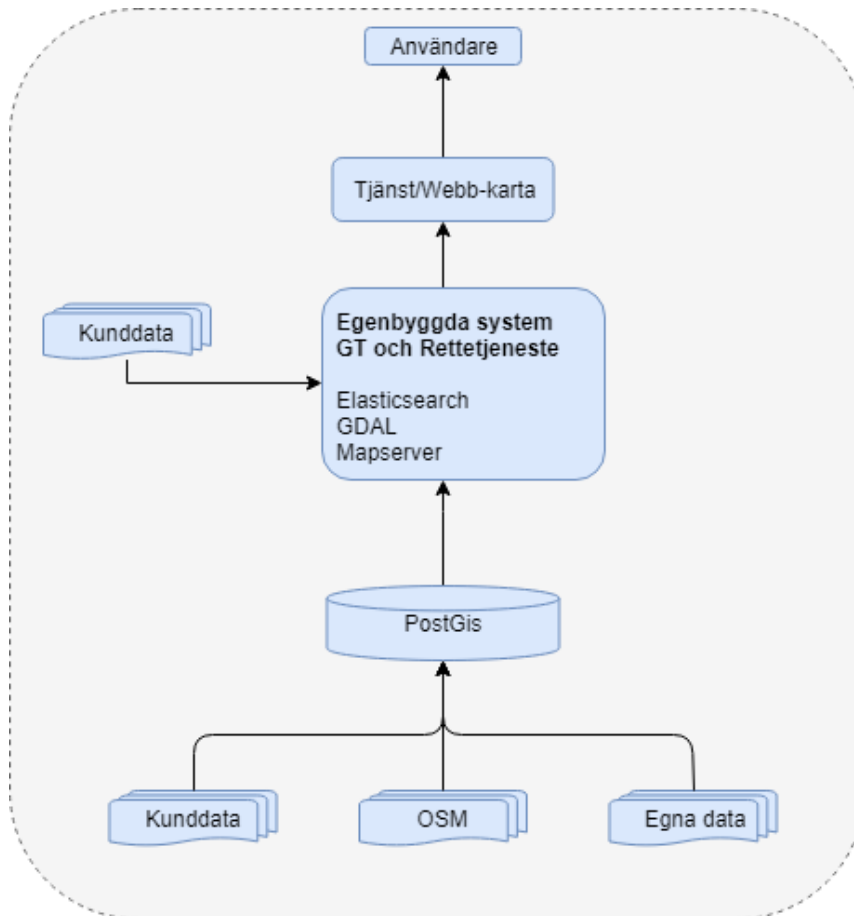


Figur 7.2 Egengjort flödesschema för publiceringsprocessen på stadsbyggnadskontoret på Malmö stad

Stadsbyggnadskontoret på Malmö stad befinner sig just nu i en övergångsperiod från det tidigare samarbetssystemet sMap (avsnitt 2.2). sMap används fortfarande, bland annat för visualisering av webbkartor för kommunspecifika data på kommunens hemsida. Ramverket sMap utvecklades som ett samarbetsprojekt mellan kommunerna Malmö, Helsingborg, Kristianstad och Lund. SMap utvecklades internt med OpenLayers och senare externt med Leaflet. Javaskriptbiblioteket Leaflet (se avsnitt 5.4.2) som vid tiden för projektet bedömdes vara ett lättarbetat gränssnitt med den stora fördelen att vara responsivt, vilket innebär att layouten anpassas utifrån storlek på skärmen och ger ett optimalt utseende för en specifik skärm. Eftersom sMap ansågs vara för personberoende och känsligt för personalomsättningar valdes det på stadsbyggnadskontoret i Malmö att släppa fortsatt utveckling av ramverket och istället succesivt övergå till det kommersiella programmet Cartesia Spatial Map (CSM), som utvecklas av Sweco Danmark och anpassas för svensk användning av Sokigo.

7.4 Publiceringsprocess - T-Kartor Kristianstad

På T-kartor finns inget generellt flöde från data till färdig webbkarta, utan det kan se lite olika ut beroende på vilken tjänst som utvecklas. I figur 7.3 har ett generaliserat flöde sammanställts för att visa hur data går från data till en färdig tjänst.



Figur 7.3 Egengjort flödesschema för publiceringsprocessen på T-kartor Kristianstad

T-kartor använder olika typer av datakällor, vilka bland annat kan vara kundernas egna data eller fria data från OpenStreetMap. I vissa fall är företaget även själva ute och samlar in data till sina produkter. Denna data lagras i framför allt PostGIS-databaser men i många fall hanterar inte T-kartor data utan ger användaren möjlighet att importera egna data i de tjänster som produceras av T-kartor. T-Kartor utvecklar ett antal egna tjänster och produkter, varav två av dessa är GT och Rettetjeneste.

I figur 7.3 ovan har det illustrerats hur dessa egenproducerade tjänster är uppbyggda. GT är en tjänst där kunden kan ladda in egna data i systemet. I tjänsten kan sedan sökningar och andra databehandlingar utföras med hjälp av programmen Elasticsearch, GDAL och Mapserver. Elasticsearch är en sök- och analysmotor som är byggd på öppen källkod och används i T-kartors fall till spatiala sökningar av data [104]. GDAL är ett bibliotek för raster och vektorformat som även det är uppbyggt av öppen källkod. GDAL tillhandahåller en mängd olika och användbara verktyg för att bearbeta och transformera data. Eftersom GDAL saknar gränssnitt använder T-kartor biblioteket kombinerat med Mapserver [89]. Stilsättningen för

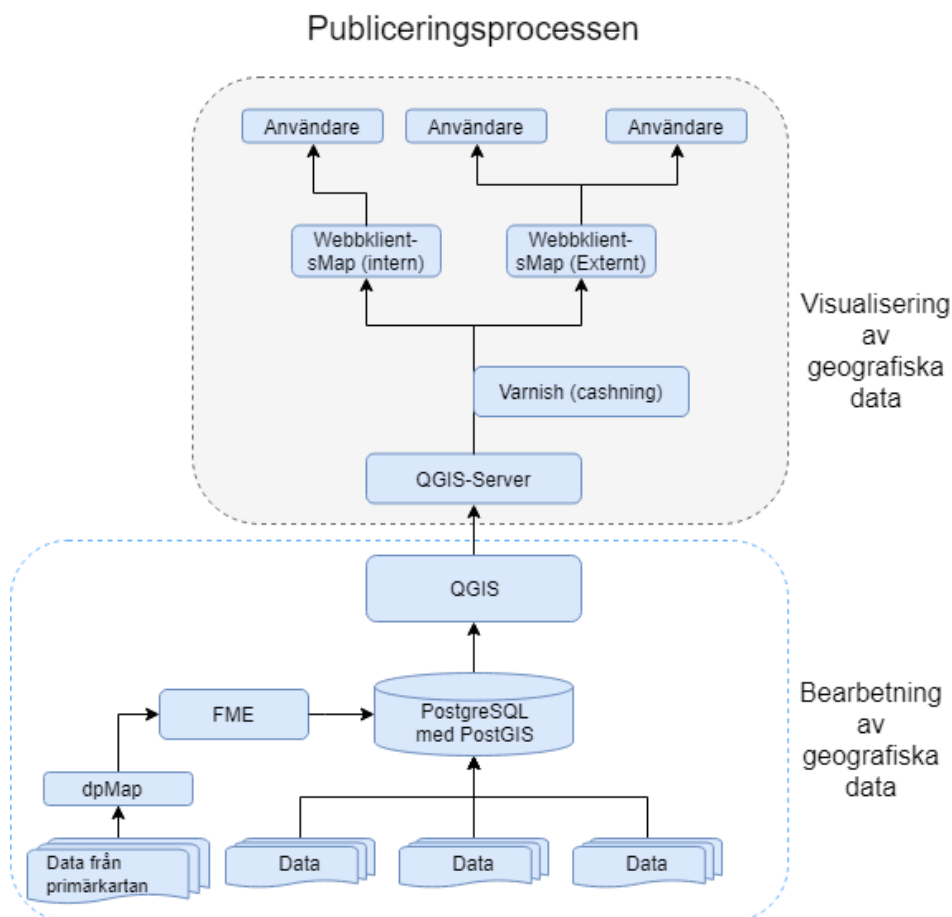
webbkartorna sköts med hjälp av Mapserver, däremot är det inte alla gånger kunden publicerar kartorna på webben men om det görs används Mapserver för själva publiceringen. Själva flödet i T-kartors egenproducerade tjänster GT och Rettetjeneste innefattar en helt automatisk process och därmed behövs inga manuella moment utan kunderna kan direkt aktivera data för att publicera dessa på webben om så önskas.

7.5 Publiceringsprocess - Kristianstad kommun

Kristianstad kommun innehar, likt andra kommuner, en hel del egna data som verksamheten själva är ute och samlar in och lagras i en PostgreSQL-databas med det spatiala tillägget PostGIS. PostGIS är en databas med öppen källkod och har i kommunen ersatt den tidigare använda kommersiella databasen Oracle. För att ladda ned och uppdatera all data används ETL-verktyget FME (avsnitt 5.1.1) för att exempelvis hämta förändringar i primärkartan. Dessa data läses in från ett system kallat dpMap, utvecklat av Digpro och görs varje natt. dpMap är ett system som riktar till kommunal karthantering som samlar all kommunala data från olika verksamhetssystem som exempelvis vägar, fastighetsindelning eller annan information från fastighetsregistret. dpMap är en kommersiell produkt som Kristianstad kommun beslutat att köpa in [105].

Databehandling och stilsättning sker i det öppna källkodsprogrammet QGIS (avsnitt 5.4.5). Tidigare har även GeoServer använts för publicering av kartorna men det har nu nästan frångåtts helt. Idag används istället QGIS-server vilket är mer kompatibelt med QGIS och tillåter återanvändning av stilsättningen gjord i programmet. Övergången från GeoServer till QGIS-server innebär att det nu undviks dubbelarbete med både stilsättning i QGIS och i GeoServer. Denna typ av lösning är även mer användarvänlig då det inte krävs några programmeringskunskaper för lösningen.

Mellan QGIS och den slutgiltiga klienten används ett lager för att öka prestandan och avlasta servern. Prestandaförbättringarna görs genom cashning, med hjälp av Varnish och programmeringsspråket PHP. Lösningen känner igen anrop och på så sätt inte skapar nya kartor om dessa anrop gjorts tidigare under dagen. Cachen rensas varje natt och är sedan förgenererat på morgonen. Den slutgiltiga klienten innefattar ramverket sMap, vilket är ett egenutvecklat ramverk byggt på Leaflet (avsnitt 5.4.2) där kartorna slutligen presenteras på webben. Ett översiktligt flödesschema över publiceringsprocessen på Kristianstad kommun illustreras i figur 7.4.



Figur 7.4 Egengjort flödesschema för publiceringsprocessen på Kristianstad kommun

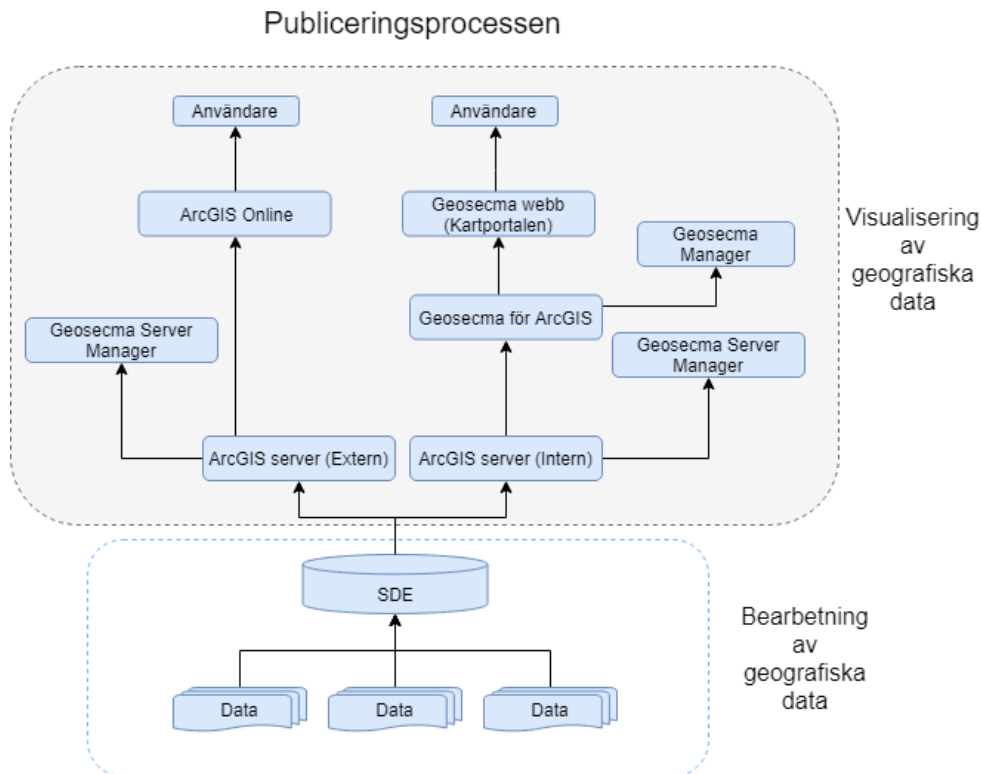
Största delen av Kristianstad kommuns flöde från data till karta är baserat på öppen källkod och endast ett fåtal kommersiella produkter används. Karl-Magnus Jönsson berättar att de på avdelningen stegvis övergått från kommersiella produkter de senaste åren. Detta då det är svårt att ersätta system med öppen källkod fullt ut på en gång. De har exempelvis övergått från en Oracle-databas till nuvarande PostGIS-databas. Det ringa användandet av kommersiella produkter sträcker sig i nuvarande system endast till tidigare nämnda datainköp via dpMap.

Flödet medför en del manuellt arbete vilket kan ses som en nackdel i systemet. De manuella momenten omfattar bland annat att sätta upp rutiner för att hämta data och stilsätta i QGIS. För att publicera data eller lager i webbkartan behöver även en del funktioner så som exempelvis zoomnivåer ställas in. Detta manuella moment är visserligen ett engångsarbete men medför trots allt en del arbete. En önskvärd funktion, enligt Karl-Magnus Jönsson, hade varit en eventuell konfiguration i systemet för att undgå detta moment.

7.6 Publiceringsprocess - Ystad kommun

Ystad kommun använder i huvudsak kommersiella produkter i den kommunala verksamheten från ESRI och S-GROUP solutions, dotterbolag till ESRI. Som indata används dels egna data som kommunen själva samlar in men data hämtas även från externa källor, däribland

Lantmäteriet, Länsstyrelsen och SCB. All data hanteras och lagrad i en ArcSDE-databas, vilket är en databas tillhandahållen av ESRI. Ett flödesschema över publiceringsprocessen på Ystad kommun har sammanställts i figur 7.5.



Figur 7.5 Egengjort flödesschema för publiceringsprocessen på Ystad kommun

Ystad kommun har, likt Kristianstad kommun och många andra kommuner, två system bestående av ett internt och ett externt system. Dessa system är uppbyggda på liknande sätt och i samma ramverk, skillnaden är i vad som blir publikt eller inte. Det interna systemet används i många tillämpningar i hela den egna verksamheten inom ett antal olika förvaltningar. Det externa systemet däremot används i synnerhet för att publicera data för medborgare och företag i form av kartor, storymaps och appar. För att publicera data i det externa systemet används ArcGIS-online, avsnitt 5.2.2. Både det interna och det externa systemet använder ESRI:s server ArcGIS-server.

I den interna verksamheten används S-GROUP solutions produkt Geosecma som är en anpassning av ArcGIS för olika ämnesområde. Ystad kommun använder en version anpassad för kommunal verksamhet.

7.7 Jämförelse av publiceringsprocesser

Publiceringsprocesserna på de tillfrågade kommuner och företag anses svåra att jämföra. Hur processen för data till färdig webbkarta genomförs skiljer sig för samtliga respondenter. En

generell observation utifrån intervjustudien är att publiceringsprocessen i de flesta fall skiljer sig beroende på tillämpningsområde. För kommuner kan de olika tillämpningsområden vara exempelvis det interna och det externa, för T-kartor bestod skillnaderna i publiceringsprocessen i vilket projekt processen tillhörde. Processerna skiljer sig även i användningen av öppen källkod, där exempelvis Kristianstad kommun och T-kartor nästan uteslutande använder öppen källkod medan Ystad är dess motpol och Malmö någonstans mitt emellan.

Trots att publiceringsprocesserna skiljer sig och därmed är svåra att jämföra kan likheter för det generella flödet ses. Alla geografiska data som ska publiceras lagras i en databas för att sedan manipuleras på något sätt med ett ETL-verktyg eller GIS-programvara. Samtliga geografiska data genomgår sedan någon typ av stilsättning innan de slutligen publiceras i en karta.

8 Fallstudie del II – Intervjustudie om öppen källkod

8.1 Intervju med Ulf Minör, stadsbyggnadskontoret Malmö stad

Den 1 mars 2019 hölls en intervju med Ulf Minör på stadsbyggnadskontoret på Malmö stad. Intervjun syftade till att utreda hur en stor organisation som Malmö stad ser på publicering av geografiska data på webben. Under intervjun berättade Ulf Minör att stadsbyggnadskontoret på Malmö stad tidigare haft ett samarbete med Kristianstad, Lund och Helsingborg i ett projekt som kallades sMap.

Ulf Minör menar på att anledningen till att samarbetet och användningen av sMap frångicks var för att minska personberoendet. Med en proprietär programvara tar företaget bakom programmet ansvar för utveckling. Tidigare har det använts olika programvaror och system på exempelvis fastighetskontoret, väg- och gatukontoret och stadsbyggnadskontoret, vilket splittrat kommunen och medfört en del dubbelarbete och onödig lagring av data. Idag satsas det istället på att alla ska använda CSM.

En erfarenhet av öppen källkod är, enligt Ulf Minör, att lösningen varit flexibel. Han anser även att uppdateringar och systemåtgärder sker snabbt om lösningen är egenutvecklad eftersom det då inte behövs vänta på att leverantören åtgärdar problem eller brister. sMap-samarbetet bestod av en eller flera kunniga från varje kommun vilket gav en viss säkerhet i eventuellt bortfall av personal men egenutveckling tenderar trots det till att vara känsliga för personalomsättningar.

Ett annat problem med användandet av öppen källkod är risken att intresset för projektet avtar med tiden. Ett sådant fall, ett så kallat dött projekt, medför stora problem i längden. Om projekten inte uppdateras riskerar de till slut att inte längre vara kompatibla med andra komponenter exempelvis webbläsare. Ulf Minör nämner dock att det kommersiella program Catesio Spatial Map, CSM, som används på Malmö stad idag är byggt på det öppna gränssnittet OpenLayers (avsnitt 5.4.1) och i nuläget är byggt på OpenLayers 2 (dock är en uppdatering med version 4 på gång). Idag finns redan OpenLayers v.5.3, vilket ger en indikation om resonemanget att det krävs ständig uppdatering om det i en verksamhet ska utvecklas system med öppen källkod. Ulf Minör säger avslutningsvis att han tror att öppen källkod är här för att stanna och ser ljus på framtiden med avseende på öppen källkod.

8.2 Intervju med Erik Ravhed, T-kartor Kristianstad

En intervju hölls den 6 februari 2019 med Erik Ravhed, utvecklare på T-kartor i Kristianstad. Intervjun syftade till att undersöka hur och vilka lösningar T-kartor använder för att publicera webbkartor. Det framkom ur intervjun att T-kartor till största del använder egenutvecklade system byggda på en grund av öppen källkod. Erik Ravhed menade på att det är en stor fördel från deras sida att använda egenbyggda system eftersom det ger möjlighet att bygga in den funktionalitet som efterfrågas samt större kontroll på att det fungerar som de tänkt. Det framkom även att det tidigare använts kommersiella produkter men i brist på fungerande

support och en produkt som inte höll det mått som krävdes valdes det att ersätta produkten. T-kartor startade därför en utveckling av en egen tjänst bestående av olika komponenter med öppen källkod.

Vid val av programlösning är det viktigast att väga licenskostnaderna mot eventuella utvecklingskostnader. Erik Ravhed trycker, trots sin positiva syn på öppen källkod, på att en egenutvecklad produkt inte behöver vara lönsam i varje fall. Erik Ravhed menar på att utvecklingskostnaderna kan bli för höga, vilket medför att det är viktigt att planera den bästa lösningen inför varje projekt. Om lösningar med öppen källkod väljs är det viktigt att det finns aktivitet inom projektet, vilket innebär att det finns många utvecklare som uppdaterar projektet efter hand och hanterar buggar, nya funktioner och uppdaterat till ny teknik. Annars riskerar företaget att själva få stå för stora delar av utvecklingskostnaderna vilket kan bli väldigt kostsamt.

Erik Ravhed menar på att den största fördelen med användningen av öppen källkod är att de som utvecklare har ansvaret på deras sida, till exempel beskriver han att om det om det blir något fel kan de som utvecklare själva uppdatera koden. Dessa buggfixar kan med fördel göras direkt utan att behöva vänta på eventuella buggfixar från leverantören. Erik Ravhed menar dock på att det egna ansvaret även kan vara en nackdel ifall det inte finns tillräckligt med kompetens inom företaget. Eftersom öppen källkod till största del utvecklas ideellt av olika intresseorganisationer kan användningen av öppen källkod utan tillräcklig kompetens och avsaknaden av support leda till att projekt stannar av eller liknande konsekvenser.

Erik Ravhed ser ljust på framtiden avseende öppen källkod och menar på att anledningen till att det inte används till så stor del i alla företag och kommuner mest beror på att många anser att det är tryggast att använda system som redan finns på företaget. Där det sedan tidigare finns både kompetens men också en kultur av att används vissa kommersiella produkter som medarbetarna är nöjda med. En annan stor fördel är också tryggheten med support från utvecklaren. Utveckling av öppen källkod kan innebära en stor risk men också medföra stora fördelar om en organisation vågar satsa på utveckling av en lösning helt eller delvis med öppen källkod.

8.3 Intervju med Karl-Magnus Jönsson, Kristianstad kommun

Den 8 februari 2019 genomfördes en intervju med Karl-Magnus Jönsson från Kristianstad kommun, där han arbetar som GIS-utvecklare. Karl-Magnus Jönsson är utöver det ordförande i QGIS-Sverige. Intervjun syftade till att undersöka hur Karl-Magnus Jönsson och GIS-enheten på Kristianstad kommun resonerar kring användandet av program med öppen källkod samt vilka system som används i verksamheten.

Kristianstad kommun använder nästan uteslutande program och komponenter med öppen källkod i sina kartsystem. Webblienten där kommunen publicerar och arbetar med sina kartor är från början ett samarbetsprojekt, sMap, som utvecklades tillsammans med Helsingborg, Malmö och Lunds kommun. Tjänsten finns i en extern och intern del som dagligen används inom stora delar av verksamheten i kommunen. Investeringen i kartprodukten kan motiveras

genom att den kan integreras med kommunens verksamhet, e-tjänster samt generera specifik och tematisk information utan kommersiella inslag till medborgarna och andra intressenter.

Karl-Magnus Jönsson anger att användandet av öppen källkod innebär en frihet att påverka vilka funktioner och komponenter som ska finnas i tjänsten. Användandet av öppen källkod innebär även en bra och tät kommunikation med användarna av tjänsten eftersom utvecklingen anpassar programmet efter användarnas behov. En annan fördel med program med öppen källkod är att det undviks licenskostnader och långa avtal. Slutligen anger Karl-Magnus Jönsson att det är fördelaktigt att inte tjänsten låses till att använda vissa typer av komponenter som bara är kompatibla med andra systemkomponenter, utan en del kan bytas ut utan inverkan på övriga komponenter. Fördelarna får till viss del kompenseras med mer kompetent personal och högre utvecklingskostnader. Friheten innebär även ett stort ansvar genom att exempelvis alltid behöva finnas tillgängliga för support internt inom kommunen. På grund av att egenutvecklade produkter med öppen källkod saknar support måste det lösas sådana problem internt, vilket kan göra systemen personberoende till kunnig personal.

Karl-Magnus Jönsson är nöjd med hur Kristianstad har lyckats med användningen av system med öppen källkod i sina karttjänster och menar på att en framgångsfaktor för deras del har varit intresserad personal och ledning med en bra kompetens inom geografisk information. Utöver det är Kristianstad en lagom stor kommun för att lyckats samordna systemen samt innehar en historia och kultur av att vilja testa nytt. En annan viktig aspekt enligt Karl-Magnus Jönsson är att utvecklandet av nya produkter medför ett intressantare och roligare arbete för medarbetarna, vilket kan vara en viktig faktor för att behålla personal, och få mer engagerade och drivna arbetskamrater med mycket personligt engagemang.

Karl-Magnus Jönsson tror starkt på en fortsatt användning av program med öppen källkod i framtiden men resonerar utifrån tidigare nämnda nackdelar exempelvis personberoende och en svaghet i att supporten måste skötas internt. En möjlig lösning i framtida projekt är att använda sig av en konsult som organisatör, utvecklare och support för projekt eller systemdelar. Denna typ av lösning skulle medföra en minskad känslighet i systemet och innebära en drivande part i samarbetsprojekt. Karl-Magnus Jönsson förmodar att anledningen bakom ett stort användande av kommersiella produkter i många andra kommuner och organisationer är att de kommersiella programmen ger snabbt resultat och en produkt som fungerar direkt. En annan faktor till kommersiell användning kan vara en typ av kultur på arbetsplatsen och en erfarenhet av att använda vissa produkter som gör att det blir naturligt att fortsätta använda dessa.

8.4 Intervju med William Walker, Ystad kommun

Den 13 februari 2019 hölls en intervju med William Walker som arbetar på GIS-enheten på Ystad kommun. Intervjun syftade till att utvärdera hur en liten organisation som Ystad kommun ser på problemställningen med öppen källkod. Intervjun syftade även till att göra en jämförelse med en organisation som i huvudsak använder kommersiella produkter på GIS-avdelningen.

William Walker berättar att Ystad kommun har använt S-GROUP solutions produkt Geosecma sedan 2011 och sedan 2017 även en så kallad kommunlicens med ESRI:s produkter. Avtalet är löpande och innehar en lång uppsägningstid vilket gör att framtiden, med avseende på produkt-

och lösningsval inte ser ut att vara öppen källkod. Då ESRI:s lösningar även används inom andra förvaltningar i kommunen finns det inte incitament till att byta val av lösning då det, enligt William Walker, skulle ställa till med mer problem.

William Walker trycker på att det är lätt att komma igång med produkterna som används idag och att det inte behövs någon större IT- eller programmeringsbakgrund för att snabbt komma igång. William Walker menar på att det måste finnas rätt kompetens för att kunna utveckla under öppen källkod. I Ystad kommun arbetar det just nu endast två personer på GIS-enheten och har som det är redan ett stort ansvarsområde vilket medför att det inte finns den tid eller den kompetensen till att starta en utveckling med öppen källkod.

Med dagens lösning upplevs inga större begränsningar för de tillämpningar det används för förutom att den kommersiella lösningen medför en specifik stilsättning. Begränsningen innebär även att det idag inte är möjligt att själv förändra utseendet med hjälp av JavaScript och sedan använda ESRI:s servrar igen, i så fall krävs en annan egen server.

William Walker tror att anledningen till att användningen av öppen källkod skiljer sig mellan olika kommuner beror på skillnaden i ledning, kompetens och intresse. Förutom intresse och duktiga medarbetare menas det på att det från kommunens sida måste satsas på GIS inom en kommun för att det ska finnas resurser till vidareutveckling.

8.5 Sammanställning av för- och nackdelar med öppen källkod från intervjuer

Samtliga respondenter har kraftfullt betonat att användningen av öppen källkod kräver kompetent personal och resurser till utveckling. Utöver det krävs ett visst intresse på arbetsplatsen samt hos ledningen för att det ska vara lönsamt att starta egen utveckling med öppen källkod. I vissa fall kan detta intresset hos ledningen även förekomma som ett krav på användning av öppen källkod. Utifrån intervjustudien har det övervägande framkommit positiva synpunkter, tankar och idéer på användandet och utvecklingen av och med öppen källkod men det har även framkommit ett antal nackdelar. Nedan listas de största fördelar och nackdelar som framgått genom intervjustudien.

Fördelar:

- Öppen källkod innebär en stor frihet i att kunna utveckla och skapa ett system efter egna önskemål, ambitioner och mål.
- Öppen källkod medför en stor frihet att välja komponenter som integrerar på ett bra sätt och uppfyller utvecklarens krav.
- Med öppen källkod undviks licenskostnader och långa avtal.
- För kommuner och myndigheter undviks offentlig upphandling.
- Användningen av öppen källkod möjliggör en tät kommunikation med användarna. Detta medför att buggar, önskemål och utökad funktionalitet kan utföras snabbt och i kommunikation inom den egna organisationen, utan att behöva vänta på en uppdatering från den kommersiella tillverkaren.

Nackdelar:

- Öppen källkod innebär ett stort ansvar för verksamheten eftersom support saknas.
- Utveckling med öppen källkod kräver hög kompetens.
- Utveckling med öppen källkod tenderar till att bli personberoende vilket innebär att verksamheten blir bräcklig om personal försvinner.
- Utveckling av system med öppen källkod innebär oftast längre utvecklingstid innan systemet kan användas medan kommersiella produkter oftast är färdiga för produktion direkt.
- Det finns en risk i användningen av öppen källkod att källkodsprojektet avstannar och inte uppdateras. Detta kan medföra att kompatibilitet med andra filtyper och program försvinner i takt med att tekniken utvecklas.
- Användning av öppen källkod innebär att personal och användare av det egna systemet måste utbildas, vilket kan leda till en högre utbildningskostnad både i uppstart men också över tid.

9 Fallstudie del III - Egen implementation

9.1 Kravspecifikation för visualiseringssteget

9.1.1 Bakgrund

Nedanstående avsnitt specificerar de krav som är framtagna för den egna lösningen i nästkommande avsnitt. Kravspecifikationen ligger även till grund för en utvärdering av den egna lösningen. Kraven är framtagna utifrån de behov som Tyréns har idag för att publicera sin data på webben, avsnitt 2.6. I GeoBIM-portalen publiceras kartan i form av en HTML-länk som är lösenordskyddad för att skydda kundens data. En annan viktig aspekt för framtagandet av kravspecifikationen är automatisering, framförallt stilsättningen. I nuvarande process sker stilsättning manuellt. Kravspecifikationen är indelad i tekniska krav, säkerhetskrav samt krav på layout och stilsättning.

9.1.2 Tekniska krav

T1. *Bakgrundskartan ska kunna vara ett ortofoto.*

T2. *Lösningen ska kunna hantera minst fem vektorlager från projektspecifika data i form av punkter.*

T3. *Lösningen ska kunna hantera stora mängder data, (> 10 GB).*

T4. *Stöd för flera användare, så att dessa kan delas på flera kontor.*

T5. *Inga begränsningar i antalet visningar.*

T6. *Tjänsten ska generera grundläggande funktionalitet såsom tillåtelse för zoomning och klickning.*

T7. *Export och import av data till och från FME är möjlig.*

T8. *Lösningen ska kunna hantera nationella och lokala koordinatsystem för Sverige.*

9.1.3 Krav på säkerhet

S1. *Den färdiga kartan ska vara skyddad, med exempelvis lösenord eller liknande skydd.*

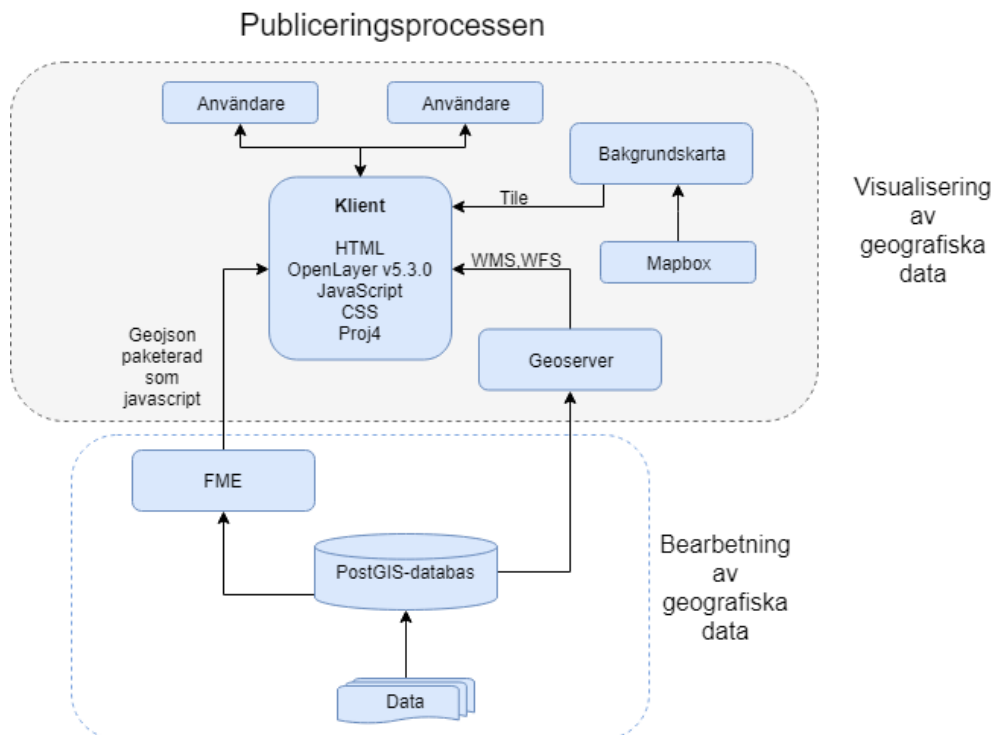
S2. *Användarna ska endast komma åt kartan i sitt eget projekt.*

9.1.4 Krav på layout och stilsättning

- L1. Programmet ska stödja automatisk textsättning för punktobjekt.
- L2. Det ska gå att ändra färger på objekt i kartan.
- L3. Teckensnitt ska vara justerbart och valbart.
- L4. En dynamisk legend för visuella lager i kartan ska finnas.
- L5. Stilsättningen ska kunna sättas som "default" för en mer automatiserad stilsättning
- L6. Stilsättning ska kunna sättas som dynamisk beroende på vilka data som kommer in
- L7. Kartan ska ha hög läsbarhet

9.2 Utformning av publiceringsprocessen

Den egna lösningen består av två delar. Ett specifikt fall och ett mer generellt fall vilka båda resulterat i en klient bestående av en JavaScript-klient. Klienten bygger på ramverket OpenLayers (5.4.1). Utöver det har ramverket proj4 (beskrivs närmre i avsnitt 9.4.1) använts samt JavaScript (3.5.2) och CSS. Ett översiktligt flödesschema för hur fallstudien genomförts visualiseras i figur 9.1.



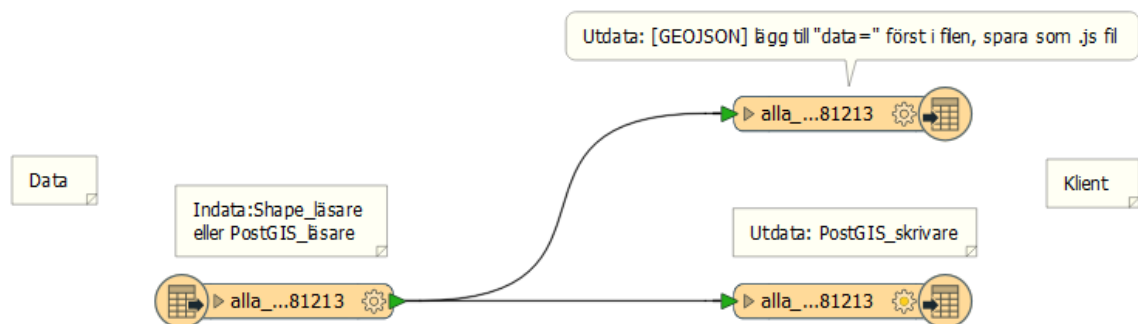
Figur 9.1 Översiktligt flödesschema av den egna implementationen

Fallstudien har utgått från data i en PostGIS-databas, för att i första fallet importeras till ETL-verktyget FME. FME exporterar sedan data som en GeoJSON-fil paketerad som en JavaScripts-fil till klienten som läser in filen, hanterar och stilsätter punkterna och publicerar dessa i kartan.

Om data importeras som lokal fil till klienten så läser OpenLayers denna fil i GeoJSON-format (eller annat format som specificeras i OpenLayers-funktionen). För att importera GeoJSON på ett enkelt och smidigt sätt deklarerar GeoJSON-filen som en global variabel (var data= {...}). Filen sparas sedan som ett JavaScript, vilket kan hanteras smidigare i webbläsaren. I figur 9.2 illustreras hur genereringen av denna JavaScript-fil går till.

För GeoServer-fallet har det gjorts en koppling mellan PostGIS-databasen och GeoServer (5.3.1). Lagret läses sedan in antingen som WMS eller WFS till klienten beroende på hur automatiserad stilsättningen ska vara. Om filen läses in som WFS till klienten sker stilsättningen automatiskt i klienten. Läses den däremot in som WMS måste stilsättningen ske i GeoServer.

Bakgrundkartan som används i fallstudien (för båda fallen) läses in som en Tile med hjälp av OpenLayers från Mapbox (5.2.3). Figur 9.2 visar även den alternativa metoden där data istället kan exporteras från FME till en PostGIS-databas där denna data sedan kan exporteras till exempelvis GeoServer.



Figur 9.2 Översiktlig bild för att illustrera olika tillvägagångssätt för att hämta data till klienten från FME

9.3 Val av kart-server

Kart-serverns uppgift i denna fallstudie är att kunna importera data från en databas men också kunna läsa in lokala filer för att sedan kunna leverera data till klienten via en WFS-tjänst. En av de mest använda serverna som bygger på öppen källkod är GeoServer. Denna server bidrar till att kraven T2 uppfylls samt att det finns en möjlighet att stilsätta text och geometrier i vektorlagren enligt krav L2, L3, L6 och L7. Import från GeoServer sker i klienten via WFS för

att sedan stilsättas via OpenLayers "style" i klienten. Data kan även stilsättas direkt i GeoServer via en SLD-fil. Det ska dock betonas att Geoserver inte används om FME väljs som import till klienten, se figur 9.1.

9.4 Val av klient

Klienten består av flera komponenter men bygger i huvudsak på OpenLayers, vilken anses bäst uppfylla kravspecifikationen (avsnitt 9.1). Även andra aspekter är iakttagits när val av klient gjorts, vilka presenteras i avsnitt 9.4.2.

9.4.1 Val av uppbyggnad av klient utifrån kravspecifikation

I OpenLayers finns möjlighet att skapa ett fritt antal vektorlager från olika datakällor, som i den egna implementationen görs lokalt från en GeoJSON-fil alternativt via WMS och WFS-tjänster. Denna aspekt uppfyller krav T2.

OpenLayers erbjuder även möjlighet för generalisering av objekt, genom exempelvis möjlighet att sätta en "offset" mellan punkter och text samt möjlighet att generera kluster för att samla ihop närliggande punkter. Dessa två möjligheter ökar läsbarheten i kartan vilket bidrar till att krav L7 uppfylls. Text, färger och flertalet andra stilfunktioner kan hanteras av klassen "Style" i OpenLayers vilket väl uppfyller kraven L1-L3. Utifrån tidigare nämnda stilfunktion som OpenLayers erbjuder finns även möjlighet att skriva funktioner där attribut i data kan kopplas till specifik stilsättning. OpenLayers erbjuder även flexibilitet i att implementera egna JavaScriptfunktioner vilket sammantaget med "style" genererar att kraven L4-L6 uppfylls. OpenLayers stödjer även dynamisk funktionalitet exempelvis zoom, pop-up och interaktion med geometrier i kartan såsom exempelvis klickbarhet och "hover" vilket bidrar till att krav T6 uppfylls.

Sammanfattningsvis uppfyller OpenLayers ensamt Krav T2 och L1-L8 men eftersom OpenLayers är ett JavaScriptbibliotek kan OpenLayers även lätt kopplas ihop med andra funktioner i JavaScript samt sammankopplas med andra JavaScript-bibliotek såsom Proj4 och Cesium. OpenLayers tillhandahåller ett antal fördefinierade referenssystem men tyvärr saknas en stor mängd referenssystem. För att kunna uppfylla krav T8 det vill säga att kunna använda och transformera befintligt referenssystem (standard är WGS 84) till nationella eller lokala referenssystem som exempelvis SWEREF99 13 30 i OpenLayers, krävs att dessa system först definieras. För detta tillämpningsområde finns ett Javascript-bibliotek, *proj4*, som kan integreras med OpenLayers där det kan definieras och transformera referenssystem på ett relativt smidigt sätt. I den egna implementationen har proj4 använts för transformation mellan WGS 84 och det lokala referenssystemet EPSG:3008 (SWEREF99 13 30), eftersom punkterna var angivna i detta koordinatsystem.

För att uppfylla kraven på säkerhet S1 kan en .htaccess-fil läggas till klienten vilket gör den möjlig att lösenordskyddas. Eftersom alla inkluderade filer är av öppen källkod äger administratören rättigheter att dela klienten vilket även uppfyller krav S2 och T4.

Klienten hämtar bakgrundskartor, ett ortofoto och en bakgrundskarta kallad *streets*, från Mapbox vilket uppfyller krav T1 men även ger möjlighet för användaren att byta bakgrundskarta vid önskemål. Bakgrundskartorna går enkelt att byta ut mot andra kartleverantörer exempelvis OpenStreetMap. Mapbox har en begränsning på 50 000 fria visningar innan det tas ut en mindre avgift, vilket innebär att krav T5 inte är uppfyllt men används istället OpenStreetMap så är kravet uppfyllt. Valet av bakgrundskarta motiveras i avsnitt 9.4.

ETL-verktyget FME stödjer export till PostGIS-databaser samt kan exportera lokala filer i flertalet format, vilket gör det möjligt att läsa in data till en klient från en PostGIS-databas men också från exempelvis GeoServer eller via lokala filer. OpenLayers har stöd för att läsa data via nedladdningstjänster som WMS och WFS. Eftersom klienten inte stödjer export från klienten till FME så uppfylls krav T7 endast delvis. Data kan lagras både i PostGIS, lokala filer och GeoServer.

Sammantaget uppfyller klienten kraven T1, T2, T4-T6, T8, S1, S2, L1-L7 och delvis T7.

9.4.2 Andra aspekter vid val av klient

Bakgrunden till varför OpenLayers valts är förutom att det uppfyller de uppsatta krav som finns väl att det framkommit mycket positiva tankar och åsikter om JavaScript-biblioteket från intervjustudien. Utifrån intervjustudien framkom en rad olika fördelar och nackdelar med användandet av öppen källkod (avsnitt 8.5). Samtliga respondenter, oberoende på användningsgraden av öppen källkod i verksamheten, hade ett stort intresse för och såg positivt på framtiden med avseende på öppen källkod. Utifrån intervjustudien har det även observerats att det går att utveckla det mesta med hjälp av öppen källkod, med förutsättningen att det finns kompetens och tid.

Erik Ravhed från T-kartor berättade att gränssnittet bakom deras system bygger på OpenLayers. Erik Ravhed menade på att OpenLayers medför en betydligt högre funktionalitet jämfört med exempelvis Leaflet. Karl-Magnus Jönsson från Kristianstad kommun framförde samma mening trots att Kristianstad kommun utvecklat i Leaflet. Som svar till varför kommunen valt att utveckla i Leaflet trots att OpenLayers anses bättre var att utvecklarna i projektet hade arbetat med Leaflet tidigare och därav valdes Leaflet samt att det vid tiden för uppstart av projektet stödde responsiv design bättre.

Utifrån egna erfarenheter har det upplevts att OpenLayers tillhandahåller massvis med exempel på funktioner på OpenLayers hemsida. OpenLayers stöds även av ett stort nätverk på olika forum där utvecklare ger tips och hjälper andra utvecklare, vilket även det medför en stor fördel vid informationssökning av funktioner eller felsökning. Detta ansågs viktigt då en egen utveckling skulle göras. Trots att exemplen inte direkt är applicerbara på den egna koden så kan inspiration och tips hämtas från dessa forum.

9.5 Val av bakgrundskarta

I implementationen har det valts att använda två bakgrundskartor från Mapbox. Detta för att tillgodose krav T1. De två kartorna som används i klienten är *satellite* och *streets* där den först nämnda är standardkartan. Användaren har dock möjlighet att byta bakgrundskarta till *streets* via en knapp i kartan. Mapbox valdes på grund av den enkelhet som *satellite* erbjöd. Kartan är en satellitbild utan annan information i form av text eller vektorlager i form av exempelvis vägnät. Valet av bakgrundskarta går med andra ord även att koppla till krav L7 då enkelheten medför en hög läsbarhet av kartan utan överflödig information. För att använda Mapbox bakgrundskartor krävs att det skapas ett konto för att få tillgång till den så kallade "access token", vilket är en nyckel för att få åtkomst till kartan. Kontot är gratis att skapa och gäller utan tidsbegränsning. En nackdel med användandet av Mapbox är däremot att det finns en begränsning i antalet visningar av kartan. Gränsen för fri visning går upp till 50 000 vilket ansågs tillräcklig. Det ska betonas att vilken bakgrundskarta som helst hade varit möjlig att använda men Mapbox ansågs passa bäst för detta tillämpningsområde.

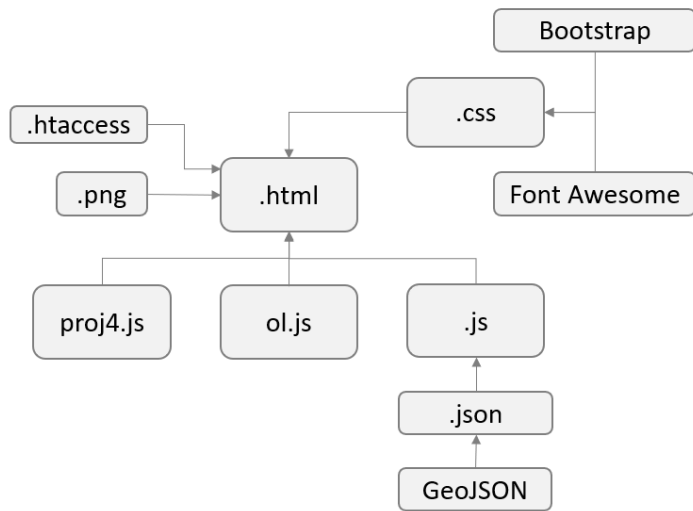
9.6 Teknisk beskrivning av klient

Klienten är utvecklad för Google Chrome eftersom webbläsaren har hög funktionalitet och bra prestanda för den här typen av klienter, men klienten är tillämpbar på i princip alla andra webbläsare. Klienten som är utvecklad är en så kallad klientapplikation vilket innebär att koden exekverar koden på klientsidan och inte på serversidan.

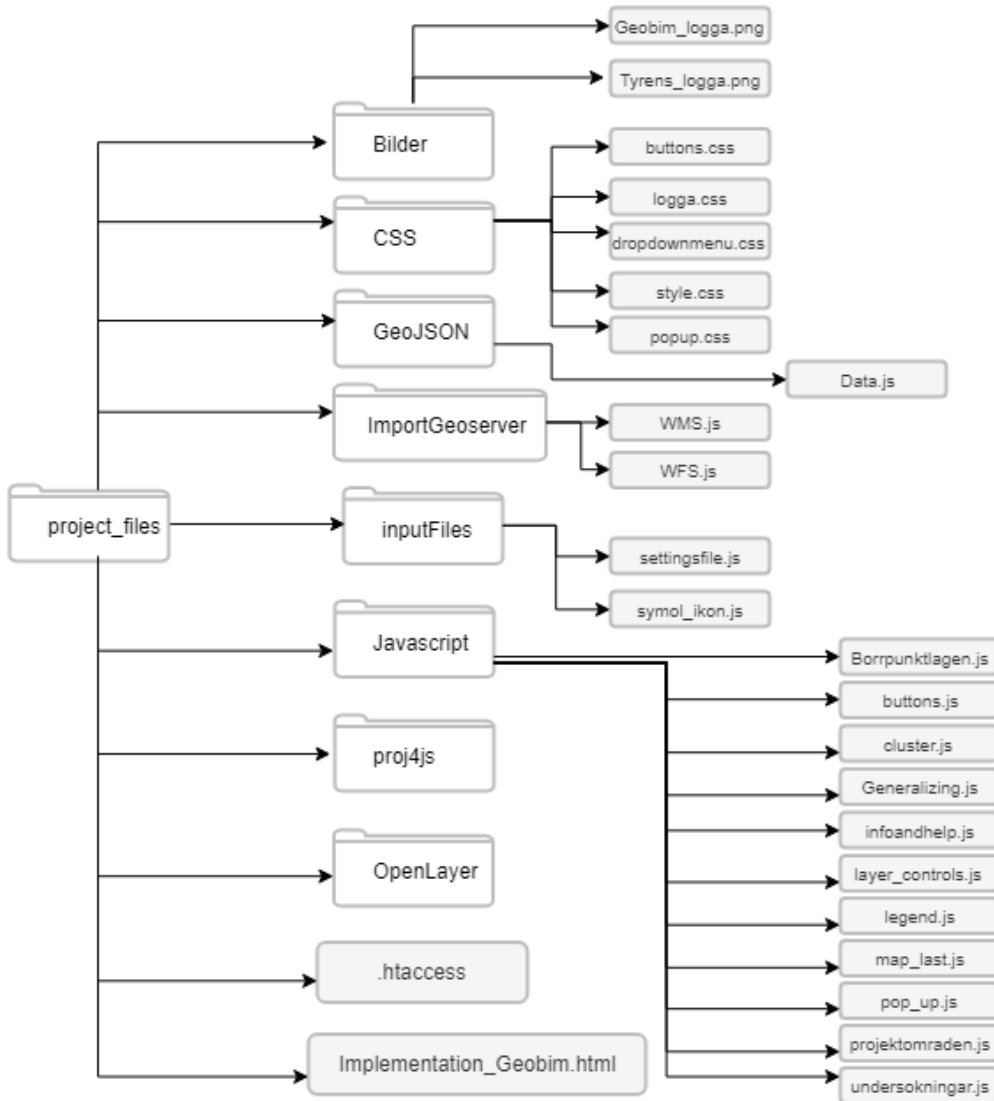
Klienten är uppbyggd av ett antal filformat vilket visas i figur 9.3 och en mer fördjupad bild över filstrukturen för klienten redovisas i figur 9.4. Klienten hanterar främst formaten HTML, CSS och JavaScript, där HTML-filen sätter samtliga objekt i klienten. Klienten anropas från HTML-filen, vilken har valts att göras enkel genom att endast representera de objekt som JavaScript och CSS-filerna sedan kan referera till. Detta har gjorts av säkerhetsskäl då en användare kommer åt HTML-filen och en grundläggande idé är att användaren inte ska ha någon tillgång till att förändra klienten.

HTML-filen arbetar nära tillsammans med CSS-filerna, som stilsätter samtliga objekt i HTML-filen med avseende på bland annat storlek, färg och placering. För ytterligare stilsättning och en del symboler och ikoner har *Bootstrap* [106] och *Font Awesome* [107] använts. Dessa länkas in i HTML-filens "header" vilket ger full åtkomst till de tillhandahållna klasserna och kan anropas från andra CSS-filer. Referenserna till biblioteken OpenLayers och proj4 placeras även dessa i *headern* och kan därför användas i resterade JavaScript-filer.

Utöver dessa bibliotek av CSS och JavaScript har det skrivits en rad olika egna JavaScript-filer vilka refereras till i HTML-filens "body". För att begränsa åtkomsten till klienten placeras en fil (av typen .htaccess) i projektmappen tillsammans med html-filen. Denna filtyp genererar lösenordskyddade HTML-filer i samma mapp. Till sist används också ett par bilder i png-format för loggorna i kartan. Dessa refereras från HTML-filen genom en .



Figur 9.3 översiktlig beskrivning av filstrukturen i klienten



Figur 9.4 Egengjord figur av filstruktur i den egna klienten

Klienten består av massvis av funktioner som tillsammans krävs för att klienten ska fungera. Nedan listas de övergripande funktioner som implementerats i klienten och i nästkommande avsnitt kommer det mer specifikt förklaras de största och mest omfattande funktionerna för klienten.

De övergripande funktioner som implementerats i klienten:

- klusterfunktion för att samla närliggande punkter
- en knapp för att byta bakgrundskarta
- en generaliseringsfunktion som flyttar överlappande punkter i in-zoomat läge
- automatisk symbolsättning av punkter baserad på attribut
- dynamisk legend
- popup-fönster med länkar till geoteknisk information då undersökningspunkterna blir klickade.
- interaktion på klick och hover för undersökningspunkterna
- tända och släcka lager från legenden
- knappar för hjälp och information
- dynamisk zoomning
- lösenordsskydd
- WMS/WFS import av punkter via GeoServer
- standardstilsättning av objekt som saknar symbol.

9.6.1 Kluster

Klusterfunktionen är till för att öka läsbarheten i kartan och samla ihop närliggande punkter. Kluster är från grunden en inbyggd funktion i OpenLayers som kopplas till en vektorkälla och som sedan får modifieras för att passa det specifika syftet för funktionen. Funktionen har ett minsta avstånd i antal pixlar mellan kluster som i klienten har sats till 50 pixlar. Detta avstånd samlar ihop alla punkter innanför denna diameter till ett kluster. Detta görs dock endast om området består av fler än en punkt. Klustren förändras utifrån kartans upplösning, då avståndet mellan punkter på skärmen förändras då skalan i kartan förändras. Detta medför att antalet punkter i varje kluster uppdateras vid varje zoomsteg.

Klustren stilsätts som en cirkel där storleken på dessa cirklar anpassas dels utifrån zoom-nivå men även ifrån ett viktad värde baserat på antalet punkter i klustret. Om klustren innehåller mindre än en punkt så visas istället typikonen för punkten. Utöver den dynamiska storleken har det även implementerats en dynamisk textsättning för klustren som redovisar hur många punkter varje kluster innehåller. Kod som rör klusterfunktionen är implementerad i filen (cluster.js), se figur 9.4.

9.6.2 Generalisering

Generaliseringsfunktionen är, liksom klusterfunktionen, implementerad för att öka läsbarheten i kartan. Funktionen är implementerad i en av JavaScript-filerna (*Generalizing.js*), se figur 9.4. Detta görs genom att flytta överlappande punkter i kartan och istället placera dessa i en cirkel runt deras gemensamma borrhpunkt. Funktionen bygger på att hitta punkter med samma ID för att sedan sätta en placering med en vinkel beroende på vilken punkt i ordningen det är. Mellan punkterna som flyttats och deras gemensamma borrhpunkt har sedan linjer skapats för att ytterligare förtydliga att de ursprungligen är samma punkt men att det är olika undersökningstyper och att de därav tidigare överlappat. Generaliseringsfunktionen aktiveras när klusterfunktionen inaktiverats, vilket sker vid en specifik zoom-nivå. Valet gjordes för att öka läsbarheten i kartan, då det i ut-zoomat läge inte behöver synas lika mycket information som i in-zoomat läge.

9.6.3 Dynamisk stilsättning utifrån vilken data som läses in

För att göra stilsättningen så automatiserad som möjligt har det implementerats en dynamisk stilsättning i JavaScriptet (*undersökningar.js*, *borrpunktslägen.js*, *settingsfile.js* och *symbols_ikon.js*) för de punkter som kommer in (*data.js*) se figur 9.4. I klienten har det kopplats symboler till namnet på en undersökning. Symbolerna i sin tur refererar till en databas där de är lagrade. Detta innebär att samtliga punkter med ett specifikt namn får samma symbol. I JavaScriptet, som stilsätter punkterna, itereras samtliga punkter igenom och matchas mot en referens till en symbol. En fördel med denna typ av lösning är att det enkelt att lägga till en symbolreferens, vilket även innebär att klienten är tillämpbar för andra typer av punkter så länge en fil skapas där det kopplas symbol till namn. Om det kommer in ett attribut som saknar symbol kommer denna punkt sättas som en standardstil.

9.6.4 Dynamisk legend

Legenden i klienten visar en ikon och en förklarande text för vilka symboler som visas i kartan. Legenden är implementerad i JavaScriptet (*legend.js*, *map_last.js* och *symbol.js*), se figur 9.4, så att den uppdaterar efter vad som visas på användarens skärm. Denna funktionalitet innebär att om lagren är urklickade så visas de inte heller i legenden. Om checkboxen däremot är klickad syns ikoner beroende hur många punkter som syns på skärmen vid just detta tillfälle.

Funktionen är uppbyggd genom att legenden är kopplad som en lyssnare till en ”*dropdownmenu*” i HTML-filen, vilken lyssnar på händelser i JavaScriptet. Utifrån vad som visas på skärmen ger funktionen tillbaka en färdig tabell som klistras in i ”*dropdownmenu*” i klienten. För att möjliggöra en uppdaterad legend efter vad som visas används funktioner i OpenLayers för att hålla koll på upplösning och vilka punkter som visas på skärmen. För de punkter som visas kontrolleras attributen och hämtar text och länkar in symbolerna för dessa

punkter. Texten och länkarna till symbolerna sparas i en lista som importeras till legenden. All denna funktionalitet genomförs varje gång kartan är färdigrenderad för att fastställa att legenden hålls uppdaterad.

9.7 Användaraspekt – slutanvändare

Nedanstående avsnitt presenterar hur resultatet från fallstudiens tredje del ser ut ur ett användarperspektiv. Avsnittet går igenom utseende, design, uppbyggnad och funktionalitet av den implementerade klienten. Klienten är en del av Tyréns GeoBIM-projekt som syftar till att samla data från geotekniska undersökningar och göra det enklare för personer som arbetar i branschen att få lättillgänglig information.

Figur 9.5 illustrerar hur startvyn ser ut för klienten då HTML-filen körs. Om en användare önskar en annan bakgrundskarta kan detta fås genom att klicka på knappen "L" i nedre vänstra hörnet. Vilken bakgrundskarta som visas då knappen "L" blir klickad illustreras i figur 9.6.

Designen har främst valts för att vara enkel och lätthanterlig och med få objekt. I startvyn representeras punkterna som kluster om fler än ett objekt ligger nära varandra. Om användaren istället vill se samtliga punkter kan användaren zooma in genom att antingen klicka på plus-knappen i nedre vänstra hörnet eller använda rullhjulet på musen. I figur 9.5 kan det även observeras en vit kvadrat. Denna kvadrat är projektområdet och är ett lager baserat på utbredningen av samtliga punkter för projektet. Denna kan användaren välja att släcka genom att klicka ur projektområdet i rullmenyn i övre vänstra hörnet. I rullmenyn har användaren även möjlighet för att välja om övriga data ska visas i kartan.

Användaren kan även interagera med andra funktioner så som exempelvis info och hjälp via knapparna i övre högra hörnet.



Figur 9.5 Översiktlig startvy av den egna implementationen



Figur 9.6 Illustration över byte av bakgrundskarta

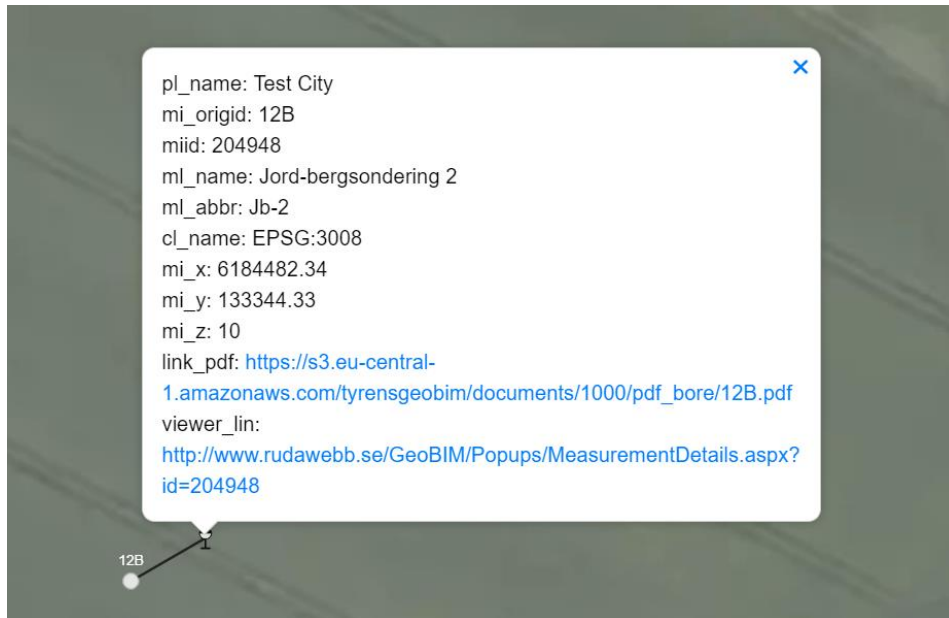
Om användaren zoomar in i kartvyn visualiseras borrpunkterna som i figur 9.7. I figuren kan användaren se vilken typ av undersökning som har utförts som centreras via linjer till borrpunktsläget. Om användare för musen över en undersökning byter denna stil för att visualisera att undersökningspunkten är klickbar.



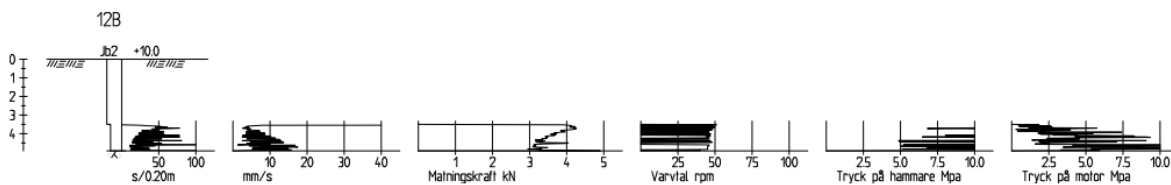
Figur 9.7 Översiktlig figur över hur punkternas läge flyttas om de överlappar

Punkterna innehåller även andra data förutom vilken borrpunkt de tillhör. Denna information kan användaren få genom att klicka på undersökningspunkten. Dessa data presenteras genom ett popup-fönster. Informationen består exempelvis av ett ID för borrhålet, information om vad

det är för typ av undersökning och länkar till viktiga dokument för undersökningen. Dessa länkar är klickbara och öppnas upp i en ny flik i webbläsaren. Figur 9.8 visar hur data presenteras i popup-fönstret och i figur 9.9 illustreras innehållet för den klickbara länken för samma borrhpunkt.



Figur 9.8 Popup-fönster för en given punkt i kartan



Figur 9.9 Innehållet i den första klickbara länken för borrhpunkt 12B

För att enkelt kunna se vilka typer av undersökningar som finns i den aktuella kartvyn på användaren skärm så kan användaren placera muspekaren över rullmenyn och få fram en legend över undersökningstyper och lager. Legendens uppdateras dynamisk utifrån aktuell information. Figur 9.10 illustrerar hur legenden representeras när samtliga punkter är synliga i kartan och figur 9.11 visar en in-zoomad bild av kartan och vilka objekt som representeras i kartan i detta fall.



Figur 9.10 Legenden visar de objekt som visas i kartan. I detta fall sex olika objekt.



Figur 9.11 Legenden visar de objekt som visas i kartan. I detta fall endast 3 olika objekt.

9.8 Jämförelse av egen implementation gentemot Carto

För att kunna utvärdera den egna implementationen gentemot Carto har det sammanställts tre tabeller (tabell 9.1–9.3) som utgår från kraven i kravspecifikation i avsnitt 9.1. Tabellerna visar vilka krav som är uppfyllda för respektive lösning idag.

Tabell 9.1 Utvärdering av tekniska krav

Krav	Egen	Carto
T1. Bakgrundskartan ska kunna vara ett ortofoto	✓	✓
T2. Lösningen ska kunna hantera minst fem vektorlager från projektspecifika data	✓	✓
T3. Lösningen ska kunna hantera stora mängder data, (> 10 GB)	**	*
T4. Stöd för flera användare, så att dessa kan delas på flera kontor.	✓	*
T5. Inga begränsningar i antalet visningar	***	✓
T6. Tjänsten ska generera grundläggande funktionalitet såsom tillåtelse för zoomning och klickning	✓	✓
T7. Export och import av data till och från FME är möjlig	****	✓
T8. Lösningen ska kunna hantera nationella och lokala koordinatsystem för Sverige	✓	✓

* Begränsat enligt licenstyp.

** Varierar beroende på vilken lagringstjänst som väljs. Inte undersökt i denna studie.

*** Beroende på vald bakgrundskarta. Mapbox ger fria visningar upp till 50 000 därefter tas en avgift.

**** Import från FME är möjlig om data paketeras som JavaScript eller laddas upp till GeoServer. Export till FME är inte implementerad eller undersökt.

Tabell 9.2 Utvärdering av krav på säkerhet

Krav	Egen	Carto
S1. Den färdiga kartan ska vara skyddad, med exempelvis lösenord eller liknande skydd.	✓	✓
S2. Användarna ska endast komma åt kartan i sitt eget projekt	✓	✓

Tabell 9.3 Utvärdering av krav på layout och stilsättning

Krav	Egen	Carto
L1. Programmet ska stödja automatisk textsättning för punktobjekt	✓	✓
L2. Det ska gå att ändra färger på objekt i kartan	✓	✓
L3. Teckensnitt ska vara justerbart och valbart	✓	✓
L4. En dynamisk legend för visuella lager i kartan	✓	-
L5. Den totala stilsättningen ska kunna sättas som "default" för en mer automatiserad stilsättning	✓	-
L6. Stilsättning ska kunna sättas som dynamisk beroende på vilka data som kommer in	*	-
L7. Kartan ska ha hög läsbarhet	✓	✓

* Möjligt men i nuläget inte fullt implementerad för alla typer av data.

I tabellerna 9.1–9.3 kan det observeras en del skillnader mellan den egna implementationen och programmet Carto. I tabell 9.1 kan fyra skillnader noteras, det vill säga för krav T3, T4, T5 och T7. Hur mycket data som kan lagras i tjänsten är främst beroende på vilken lagringstjänst som väljs, men har inte undersökts närmare i denna studie. För Carto begränsas mängden data som kan användas för att göra kartor utifrån vilken licenstyp användaren har. Ju mer pengar som betalas desto mer data kan Carto hantera, vilket medför skillnaden i krav T3. Liknande resonemang gäller för krav T4, det är licensen som avgör om det finns stöd för flera användare. För den egna klienten så har utvecklaren full kontroll över hur programvaran får delas mellan användare. För krav T5 hade det inte behövt vara en skillnad eftersom det finns tillgängliga bakgrundskartor som inte innehar någon begränsning i antalet visningar. Då 50 000 visningar ansågs tillräckligt för tillämpningen valdes denna bakgrundskarta. Bakgrundskartorna är även väldigt enkla att byta ut i klienten vilket innebär att en bakgrundskarta utan begränsningar i antalet visningar lätt kan användas istället. I Carto finns möjlighet att både importera och exportera data i olika filformat till FME vilket uppfyller krav T7. I den egna klienten finns möjlighet att importera data från FME men export av data är inte implementerad, vilket endast delvis uppfyller krav T7.

För kraven på säkerhet observerades inga skillnader mellan den egenutvecklade klienten och Carto. Däremot noterades ett par skillnader i layout och stilsättning i kraven L4, L5 och L6. Krav L4 innebär ett krav på en dynamisk legend, vilket inte finns i Carto idag. I den egna klienten har detta implementerats på så sätt att legenden dels bara tar med de attribut som faktiskt finns i kartan men även att den endast visar det som är visuellt på skärmen i varje specifikt tillfälle, vid exempelvis in-zoomning och liknande, (se avsnitt 9.6.4). För den aktuella tillämpningen av Carto sker inte stilsättningen dynamisk utan detta får manuellt ställas in i programmet vilket inte uppfyller L5 och L6. För den egna klienten så sker stilsättningen dynamiskt efter vilken data som kommer in, detta styrs via attributen i indata och går att anpassa efter olika projekt vilken uppfyller L6. Om attributen inte finns så sätts stilen till en standardstil som uppfyller L5. Den egna lösningen är anpassad efter den aktuella tillämpningen i GeoBIM så för andra indata skulle den behöva anpassas ytterligare vilken gör att L6 inte kan anses helt uppfyllt för alla datatyper.

10 Diskussion

10.1 Egen lösning

Klienten som utvecklats i fallstudien har främst utgångspunkt från den lösning som ligger till grund för Tyréns GeoBIM-tjänst, Carto. Detta genomfördes främst för att visa på möjligheten att kunna utföra likartade moment med en lösning i öppen källkod som för en kommersiell programvara. En stor del i fallstudien var också att visa möjligheten för att kunna generera projektspecifika kartor med ett så automatiskt flöde som möjligt, från data till färdig webbkarta, genom att minska stora återkommande manuella moment. Detta flöde innebär med andra ord att kunna hämta data, stilsätta data, generera en dynamisk legend och karta som anpassar sig efter vilken data som importerar till klienten. Det ska poängteras att den egna lösningen i denna studie främst är utvecklad som en klientapplikation och den kommersiella programvaran Carto har flera andra delverktyg och funktioner som inte jämförs i denna studie.

Resultatet efter den uppsatta kravspecifikation visar att både den kommersiella och lösningen med öppen källkod är relativt lika och båda två uppfyller kraven som ställs väl (se tabell 9.1–9.3). En fördel med lösningen byggd på öppen källkod är flexibiliteten i designen och möjligheten att kunna generera ett mer automatiserat flöde vilket i sin tur får vägas mot tiden att implementera denna lösning. Det ska dock poängteras att klienten som är producerad i nuläget endast hanterar punkter. Utveckling av en egen produkt är tidskrävande men för andra typer av tillämpningar med exempelvis mer komplexa geometrier hade implementationen krävt betydligt mer tid. En stor fördel med den egenutvecklade lösningen är att nu när “ramverket” är implementerat är det mindre tidskrävande att lägga till ytterligare funktionalitet eller att generera en mer generell lösning för andra typer av data, det kräver dock i nuläget att data i olika projekt är av samma format och uppbyggnad. Klienten kräver i nuläget en GeoJSON paketerad som JavaScript-fil eller att data importerar via WMS/WFS från GeoServer. För att automatisk kunna generera webbkartor i den implementerade klienten krävs det även en god kännedom av indata hos utvecklaren, då stilsättning och dynamiska kartfunktioner till hög grad är kopplade till attribut och länkar i den data som importerar.

En stor fördel att utveckla klienten med öppen källkod är flexibiliteten att kunna designa alla funktioner och layout precis efter vad som efterfrågas. En annan fördel med just OpenLayers är att det stödjer integration med många andra kända bibliotek såsom bland annat Proj4, GDAL, Cesium och Mapillary, vilket gör det lätt att vidareutveckla klienten med högre funktionalitet och integration exempelvis 3D-visualisering av borrhål.

10.2 Metodval

För att utvärdera olika alternativ till Carto valdes att fördjupa sig i en ändamålsenlig lösning av de program, med öppen källkod, som av litteraturstudien och intervjuerna visat sig vara lämpliga för ändamålet. Metoden valdes för att bättre visualisera programstruktur, för de valda

programmen, samt visa på för- och nackdelar kring utveckling med öppen källkod under implementering. En ytterligare anledning var att undersöka lösningar för ett mer automatiskt flöde från indata till färdigt webbkarta, främst med fokus på automatisk stilsättning.

Intervjustudien bestod av fyra intervjuer där det gjordes ett urval av stora och små kommuner samt ett företag. För en mer omfattande studie hade fler intervjuer behövts där det även kunde inkluderas myndigheter och andra aktörer i branschen. Trots ett fåtal intervjuer representerade intervjuerna en enhetlig och samstämmig bild över för och nackdelar av öppen källkod, vilka även överensstämde med det som framtagits ur litteraturen i avsnitt 4.8. Materialet från intervjustudien gav med andra ord en detaljerad bild över problemområdet.

Sammantaget gav litteraturstudien och intervjustudien ett bra underlag för val av egen lösning. Underlaget från föregående studier visade på fördelar och nackdelar med olika typer av lösningar med öppen källkod gentemot kommersiella alternativ. Studierna resulterade i underlag för vilka komponenter, med öppen källkod, som integrerades i klienten. Sammantaget har den kvalitativa intervjustudien, litteraturstudien och den praktiska implementationen inneburit att problemområdet utvärderats med hjälp av flera olika angreppssätt och därigenom medfört en bred inblick i problemområdet.

10.3 Öppen källkod

Studien syftar till att utreda fördelar och nackdelar med öppen källkod, vilket utvärderats både genom litteratur men även genom den intervjustudie som genomförts. För att få en egen bild av användningen och praktisk erfarenhet av öppen källkod genomfördes även en egen implementation. Det syftade också att, genom implementationen, bättre kunna redovisa fördelar och nackdelar med öppen källkod som inte presenteras på ett teoretiskt plan. Det observerades under intervjustudien att de fördelar och nackdelar som framkommit ur denna stöds av de punkter som även diskuteras i litteraturen om öppen källkod (avsnitt 4.8). Det kan funderas över vilka faktorer och anledningar som ligger bakom användandet av kommersiella programvaror eller om en verksamhet istället satsar på öppen källkod. En observation som gjorts genom studien är att det generellt finns ett stort intresse för öppen källkod och att det finns en viss nyfikenhet för ämnet. Problemen ligger vanligtvis i brist på resurser och kompetens och därför kan det vara svårt att som liten verksamhet att starta en utveckling med öppen källkod. Däremot är T-kartor ett exempel på en liten organisation som trots sin ringa storlek utvecklar stora delar med öppen källkod.

Vid val av lösning i en verksamhet är det viktigt att ha ett långsiktig plan för hur en klient eller liknande produkt kan komma att användas. Frågor som måste besvaras är exempelvis om produkten är tillämpbar för många projekt i organisationen framöver. Det måste även funderas över om det blir kostnadseffektivt att utbilda personalen för att hantera ramverket eller om det måste rekryteras fler medarbetare med kompetens för utveckling. Det behöver även göras någon typ av kostnadsberäkning som jämför kostnaden för egen utveckling gentemot en licenskostnad både på kort men framför allt på lång sikt.

En stor anledning till att öppen källkod väljs istället för kommersiella alternativ kan vara friheten i att kunna välja och skapa ett system precis utifrån de önskemål som finns i verksamheten. Utveckling med öppen källkod är delvis begränsat i den mån att många lösningar är anpassade för andra komponenter med öppen källkod. Detta kan medföra en begränsning i att komponenter blir svårintegrerade med varandra. Det måste även beaktas vilken licens som komponenterna med öppen källkod är distribuerade under. Komponenter som ska användas och integreras i nya produkter som ska distribueras kommersiellt måste tillåta detta i licensen för att inte bryta mot upphovsrättslagen.

En annan observation som gjordes i intervjustudien är att det inom verksamheter där öppen källkod används, till stor del, finns en viss anda som gemensamt verkar för utvecklingen av öppen källkod. För att utveckling med öppen källkod ska vara ett vinnande koncept i en verksamhet krävs det, förutom kompetens, ett visst intresse och nyfikenhet för ämnet hos samtliga medarbetare i verksamheten. Eftersom utveckling med öppen källkod är en ekonomisk chansning, då risken finns att det projektet som används blir ett så kallat dött projekt är det avgörande för en verksamhet att det finns stöd från ledningen. För kommuner innebär det en politisk fråga då det måste finnas satsningar och stöd från kommunledningens sida.

Under implementeringen har egna erfarenheter av öppen källkod utvärderats. Precis som litteraturen och intervjustudien påpekat är öppen källkod tidskrävande arbete och kräver kompetens. OpenLayers som JavaScript-bibliotek har upplevts väldokumenterat med flertalet olika forum där utvecklare ger tips och idéer. En fördel som har upplevts under arbetet med öppen källkod är att det finns stora möjligheter att designa funktioner och layout i program som utvecklas specifikt efter vilken data och syfte som programmet har. Detta kunde specifikt observeras när jämförelsen av den egna implementationen gjordes mot Carto i tabell 9.3. Den egna lösningen medförde en större frihet i layout och design, där Carto istället endast erbjuder en fördefinierad stil för exempelvis legenden. En nackdel som upplevts gentemot kommersiella alternativ är att kommersiella alternativ ofta kombinerar server- och klientsidan och skapar en komplett produkt. Det medför att när utveckling ska ske med öppen källkod behöver flera komponenter samverkas för att skapa en klient-serverlösning.

10.4 Publiceringsprocesser

Under intervjustudiens första del undersöktes publiceringsprocessen från geografiska data till färdig webbkarta hos de intervjuade parterna. Resultatet i denna fråga var att dessa flöden är svåra att jämföra eftersom de skiljer sig beroende på tillämpningsområde. Publiceringsprocesserna utvärderades för att ta reda på om det finns effektiviseringsmöjligheter för att i ett senare led kunna jämföra mot den publiceringsprocess som råder på Tyréns idag. Syftet med utvärderingen var även att ta reda på vilka komponenter som används på de olika organisationerna för att på så sätt ta reda på till hur stor del öppen källkod används i praktiken.

Angående effektivisering tyder både lösningen på T-kartor i Kristianstad samt vår egen lösning på att det är möjligt att få ett relativt automatiskt flöde från geografiska data till färdig

webbkarta. Det som är svårast att få helt automatiserat är stilsättning av den data som varierar från projekt till projekt. På T-kartor hade de utvecklat en produkt som skapade en webbkarta automatiskt utifrån valda data. Vår egenutvecklade klient kräver inte heller manuella moment efter import av data till klienten. Detta tyder på att effektiviseringsmöjligheter för Tyréns är möjliga att utveckla. För att få en mer heltäckande bild av automatisering hade dock en större och mer omfattande intervjustudie behövts.

Andra anledningen till att titta på publiceringsprocesserna var för att se utbredningen av öppen källkod i praktiken. Eftersom intervjustudien var begränsad till ett fåtal kommuner och företag kan det inte ge en heltäckande bild över Sverige. Däremot kunde det under intervjustudien observeras att även om en organisation inte använder öppen källkod i sin verksamhet råder det ändå en stor nyfikenhet och en generell positiv syn på ämnet.

11 Slutsatser

Examensarbetets främsta syfte var att utvärdera om öppen källkod är ett likvärdigt alternativ till det kommersiella program Carto som används på Tyréns idag för att publicera geografiska data på webben. Resultaten från den egna implementationen visar på att det finns möjligheter att ersätta Carto som publiceringsverktyg och istället använda öppen källkod.

Ett annat syfte var att utvärdera fördelar och nackdelar med öppen källkod för webbkartor. Slutsatsen till detta syfte är att det finns en rad fördelar men även nackdelar med användningen av öppen källkod, där det främst är kompetens och utvecklingstid som avgör om öppen källkod används i verksamheten eller inte. Ett annat syfte var att utvärdera i vilken utbredning öppen källkod används i praktiken och om det förekommer olika mentaliteter eller ideologier kring öppen källkod i olika verksamheter. Utifrån intervjustudien observerades att användningen av öppen källkod skiljer mellan olika typer av verksamheter. En stark koppling till användandet är att det måste finnas ett visst intresse för öppen källkod om det ska användas i verksamheten. För kommuners del kräver det politiska insatser där det satsas på respektive förvaltning som hanterar geografiska data för att det ska finnas möjlighet för utveckling. Det observerades stora skillnader mellan exempelvis Kristianstad kommun där det berättades att många som arbetade på GIS-avdelningen hade ett stort intresse för öppen källkod och det fanns en stor ”anda” att verka för öppen källkod. Motsättningsvis fanns det mindre utrymme för utveckling i Ystad kommun eftersom resurserna inte fanns i lika stor utsträckning. Dels för att kommunen inte är lika stor men även för att det inte fokuseras på GIS-utveckling inom kommunen i lika stor utsträckning. Medarbetarnas arbetsuppgifter i Ystad fokuseras i högre grad på daglig kommunal hantering av geografiska data, vilket medför att det inte finns lika mycket tid för utveckling. Slutsatsen är utifrån detta att det finns en viss ”mentalitet” i verksamheter som påverkar verksamheternas val av lösningar.

Rapporten syftade även till en teknisk utvärdering av öppen källkod genom att utvärdera olika programvaror och system för visualisering av geografiska data i en webbmiljö. Den tekniska utvärderingen gjordes i jämförelse mot det kommersiella programmet Carto. Denna utvärdering har gjorts utifrån uppsatta krav för visualiseringssteget i form av en kravspecifikation. Öppen källkod kan sammanfattningsvis vara ett bra alternativ till kommersiella produkter. Allt det som erbjuds i kommersiella produkter går att utveckla med öppen källkod. Utöver det tillför utveckling med öppen källkod en stor flexibilitet och på så sätt kan produkten bättre tillämpas efter exakta önskemål vilket är en stor fördel. Problemet med öppen källkod är att det oftast behöver kopplas ihop flera tjänster för att få en fungerande produkt och det kan då upplevas enklare att använda en befintlig kommersiell produkt, särskilt om det inte finns tillräcklig kompetens i verksamheten.

Ett sista syfte var att utvärdera publiceringsprocessen från data till färdig webbkarta. Utvärderingen genomfördes genom intervjustudien där det upptäcktes att det finns många olika lösningar för hur publicering av geografiska data utförs i praktiken. Generellt är automatisering eftersträvat då allt i slutändan handlar om tid. En slutsats som dragits är att hur publiceringsprocessen ser ut beror på vilket specifikt tillämpningsområde som processen syftar till, vilket innebär att det inom en verksamhet kan det finnas ett flertal olika processer beroende

på projekttyp. Det kan också vara så att det finns en rad andra processer i verksamheten men som inte mynnar ut i en webbkarta utan i dess fall hanteras data på andra sätt.

Den sammanlagda studien resulterade i en att öppen källkod är ett fullt möjligt alternativ till kommersiella produkter för att publicera geografiska data på webben. Nyckelfaktorer som varje organisation inför en sådan satsning noggrant måste utvärdera är lönsamheten både på kort och lång sikt vad gäller utvecklingstid mot licenskostnader och andra tidigare nämnda faktorer.

Referenser

- [1] Harrie L. Geografisk informationsbehandling. Lund: Studentlitteratur; 2013.
- [2] Peng Z, Tsou M. Internet GIS. Hoboken (New Jersey): Wiley; 2003
- [3] [Internet]. Kartor.malmo.se. 2018 [Citerad 17 december 2018]. Hämtad från: https://kartor.malmo.se/rest/leaf/1.0/?config=../configs-1.0/malmo_atlas.js
- [4] Webbkartor [Internet]. Malmo.se. 2018 [Citerad 27 november 2018]. Hämtad från: <https://malmo.se/Stadsplanering--trafik/Kartor--geografisk-information/Webbkartor.html>
- [5] Solguiden [Internet]. Malmo.se. 2018 [Citerad 27 november 2018]. Hämtad från: <https://malmo.se/Bo-bygga--miljo/Bygga-nytt--bygga-till/Solguiden.html>
- [6] Presentation sMap Möjligheter med opensource Gemensam utveckling av webbkartor med opensource Ett samarbete mellan Malmö, Helsingborg, Lund och. - ppt ladda ner [Internet]. Slideplayer.se. 2018 [Citerad 27 november 2018]. Hämtad från: <https://slideplayer.se/slide/2468990/#>
- [7] Open ePlatform - Om plattformen [Internet]. Oeplatform.org. 2018 [Citerad 26 november 2018]. Hämtad från: <http://www.oeplatform.org/omplattformen>
- [8] Om eSamverkan | Esamverkan [Internet]. Esamverkan.nu. 2018 [Citerad 4 december 2018]. Hämtad från: <https://esamverkan.nu/om-esamverkan/>
- [9] Kartor [Internet]. Kristianstad.se. 2019 [Citerad 28 januari 2019]. Hämtad från: <https://www.kristianstad.se/sv/kommun-och-politik/kommunfakta/kartor/>
- [10] [Internet]. Kartor.kristianstad.se. 2019 [Citerad 28 januari 2019]. Hämtad från: <https://kartor.kristianstad.se/strategikartan/>
- [11] Ystads Kommun - Statistik om Ystad [Internet]. Ystad.se. 2019 [Citerad 28 januari 2019]. Hämtad från: <https://www.ystad.se/kommun--politik/ystads-kommuns-organisation/kommunfakta---statistik-och-fakta-om-ystad/>
- [12] Ystads Kommun - Kartor [Internet]. Ystad.se. 2019 [Citerad 28 januari 2019]. Hämtad från: <https://www.ystad.se/kommun--politik/ystads-kommuns-organisation/kommunal-service/kartor/>
- [13] Story Map Tour [Internet]. Ystadskommun.maps.arcgis.com. 2019 [Citerad 28 januari 2019]. Hämtad från: <http://ystadskommun.maps.arcgis.com/apps/MapTour/index.html?appid=bc40f62c3c0e4ab7ac9e32cc29524572>
- [14] Ystads Kommun - Walk of Film [Internet]. Ystad.se. 2019 [Citerad 28 januari 2019]. Hämtad från: <https://www.ystad.se/kultur/film-i-ystad/walk-of-film/>
- [15] history [Internet]. T-Kartor. 2019 [Citerad 28 januari 2019]. Hämtad från: <http://www.t-kartor.com/about/>
- [16] City Customers [Internet]. T-Kartor. 2019 [Citerad 28 januari 2019]. Hämtad från: http://www.t-kartor.com/citywayfinding/customers/#Paris_STIF

- [17] Geoteknisk undersökning - SGI [Internet]. Swedgeo.se. 2018 [Citerad 27 november 2018]. Hämtad från: <http://www.swedgeo.se/sv/vagledning-i-arbetet/falt--och-laborariemetoder/geotekniska-undersokningar/>
- [18] Vad är geoteknik och miljögeoteknik - SGI [Internet]. Swedgeo.se. 2018 [Citerad 27 november 2018]. Hämtad från: <http://www.swedgeo.se/sv/kunskapscentrum/om-geoteknik-och-miljogeoteknik/geoteknik-och-markmiljo/>
- [19] GeoBIM Portalen - GeoBIM Portal [Internet]. Geobim.se. 2018 [Citerad 12 november 2018]. Hämtad från: <https://www.geobim.se/geobim/geobim-portalen/>
- [20] Inspire [Internet]. Geodata.se. 2018 [Citerad 12 november 2018]. Hämtad från: <https://www.geodata.se/styrande/inspire/>
- [21] Lag och förordning [Internet]. Geodata.se. 2018 [Citerad 12 november 2018]. Hämtad från: <https://www.geodata.se/styrande/lag-och-forordning/>
- [22] Geodatastrategin [Internet]. Geodata.se. 2018 [Citerad 15 november 2018]. Hämtad från: <https://www.geodata.se/styrande/nationell-geodatastrategi/?faq=f130>
- [23] Geodataportalen [Internet]. Geodata.se. 2018 [Citerad 15 november 2018]. Hämtad från: <https://www.geodata.se/anvanda/geodataportalen/?faq=019b>
- [24] Geodata - Lantmäteriet [Internet]. Geodata.se. 2018 [Citerad 19 november 2018]. Hämtad från: <https://www.geodata.se/geodataportalen/srv/swe/catalog.search#/search?resultType=swe-details&type=dataset%20or%20series&from=1&to=20>
- [25] Centre E. INSPIRE Geoportal [Internet]. Inspire-geoportal.ec.europa.eu. 2018 [Citerad 19 november 2018]. Hämtad från: <http://inspire-geoportal.ec.europa.eu/index.html>
- [26] Hjälp och tips till Kartsök och ortnamn [Internet]. Lantmateriet.se. 2018 [Citerad 15 november 2018]. Hämtad från: <https://www.lantmateriet.se/sv/Kartor-och-geografisk-information/Kartor-flygbilder-och-ortnamn/Om-Kartsok-och-ortnamn/?faq=aafb>
- [27] Lantmäteriet slutar med papperskartor – satsar på digitala [Internet]. Lantmateriet.se. 2018 [Citerad 19 november 2018]. Hämtad från: <https://www.lantmateriet.se/sv/nyheter-och-press/Pressmeddelande/2017/lantmateriet-slutar-med-papperskartor--satsar-pa-digitala/>
- [28] Viktigt att veta om upphovsrätt - Riksarkivet [Internet]. Riksarkivet. 2018 [Citerad 19 november 2018]. Hämtad från: <https://riksarkivet.se/upphovsratt>
- [29] Stefanakis Emmanuel, 2015, Web mapping & Geospatial Web Services, edition 1.0, ISBN: 1513757559)
- [30] Brooks DR. Guide to HTML, JavaScript and PHP. [Elektronisk resurs] : For Scientists and Engineers [Internet]. London: Springer London, 2011.; 2011 [Citerad 2018 Dec 18]. Hämtad från: <http://ludwig.lub.lu.se/login?url=http://search.ebscohost.com.ludwig.lub.lu.se/login.aspx?direct=true&db=cat01310a&AN=lovisa.004192803&site=eds-live&scope=site>
- [31] Green D, Bossomaier T. Online GIS and spatial metadata. London: Taylor & Francis; 2002

- [32] Geography Markup Language | OGC [Internet]. Openeospatial.org. 2018 [Citerad 12 november 2018]. Hämtad från: <http://www.openeospatial.org/standards/gml>
- [33] Burggraf DS. Geography Markup Language. Data Science Journal. 2006; 5:178–203. DOI: <http://doi.org/10.2381/dsj.5.178>
- [34] GML Introduction [Internet]. W3.org. 2018 [Citerad 15 november 2018]. Hämtad från: <https://www.w3.org/Mobile/posdep/GMLIntroduction.html>
- [35] RFC 7946 - The GeoJSON Format [Internet]. Tools.ietf.org. 2018 [Citerad 13 december 2018]. Hämtad från: <https://tools.ietf.org/html/rfc7946>
- [36] Welcome to The Open Geospatial Consortium | OGC [Internet]. Openeospatial.org. 2018 [Citerad 12 november 2018]. Hämtad från: <http://www.openeospatial.org/>
- [37] OGC® Standards and Supporting Documents | OGC [Internet]. Openeospatial.org. 2018 [Citerad 15 november 2018]. Hämtad från: <http://www.openeospatial.org/standards>
- [38] OpenGIS Web Map Server Implementation Specification, dokument OGC 06–032, version 1.3.0, Web Map Service | OGC [Internet]. Openeospatial.org. 2018 [Citerad 19 november 2018]. Hämtad från: <http://www.openeospatial.org/standards/wms>
- [39] DB-Engines Ranking per database model category [Internet]. Db-engines.com. 2019 [Citerad 22 Februari 2019]. Hämtad från: https://db-engines.com/en/ranking_categories
- [40] Leavitt N. Will NoSQL Databases Live Up to Their Promise?. Computer. 2010;43(2):12-14.
- [41] Kresse W, Danko D. Springer Handbook of Geographic Information [recurso electrónico] \$. Alemania: Springer Healthcare Ltd; 2012.
- [42] OGC, OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option, John R. Herring, OGC 06-104r4 Version: 1.2.1, 2010
- [43] Obe R. O, and Hsu, L. S., 2011. PostGIS in action, Chapter 1: Learning PostGIS
- [44] Iuliana BOTHERA, Anda VELICANU, Adela BARA. Modeling Spatial Data within Object Relational-Databases. Database Systems Journal [Internet]. 2011 [Citerad 7 maj 2019]. Hämtad från: <http://search.ebscohost.com.ludwig.lub.lu.se/login.aspx?direct=true&db=edsdoj&AN=edsdoj.b939470728c242fdabf24a582acd09b8&site=eds-live&scope=site>
- [45] Historical trend of relational DBMS popularity [Internet]. Db-engines.com. 2019 [Citerad 25 Februari 2019]. Hämtad från: https://db-engines.com/en/ranking_trend/relational+dbms
- [46] PostgreSQL: License [Internet]. Postgresql.org. 2019 [Citerad 25 februari 2019]. Hämtad från: <https://www.postgresql.org/about/licence/>
- [47] PostgreSQL: About [Internet]. Postgresql.org. 2019 [Citerad 25 februari 2019]. Hämtad från: <https://www.postgresql.org/about/>

- [48] Refrations Research : PostGIS Technical Overview [Internet]. Refrations.net. 2019 [Citerad 25 Februari 2019]. Hämtad från: <http://www.refrations.net/products/postgis/technicaloverview/> [PostgreSQL5] <https://postgis.net/>
- [49] Developers P. PostGIS — Development [Internet]. Postgis.net. 2019 [Citerad 25 Februari 2019]. Hämtad från: <https://postgis.net/development/>
- [50] What is open source? [Internet]. Opensource.com. 2018 [Citerad 20 november 2018]. Hämtad från: <https://opensource.com/resources/what-open-source>
- [51] History of the OSI | Open Source Initiative [Internet]. Opensource.org. 2018 [Citerad 20 november 2018]. Hämtad från: <https://opensource.org/history>
- [52] The Open Source Definition | Open Source Initiative [Internet]. Opensource.org. 2018 [Citerad 20 november 2018]. Hämtad från: <https://opensource.org/docs/osd>
- [53] Definitionen av fri programvara - GNU Projektet - Free Software Foundation [Internet]. Gnu.org. 2018 [Citerad 20 november 2018]. Hämtad från: <https://www.gnu.org/philosophy/free-sw.sv.html>
- [54] What is Copyleft? - GNU Project - Free Software Foundation [Internet]. Gnu.org. 2018 [Citerad 20 november 2018]. Hämtad från: <https://www.gnu.org/copyleft/copyleft.html>
- [55] Overview of the GNU System- GNU Project - Free Software Foundation [Internet]. Gnu.org. 2019 [Citerad 14 januari 2019]. Hämtad från: <https://www.gnu.org/gnu/gnu-history.html>
- [56] What is free software? - GNU Project - Free Software Foundation [Internet]. Gnu.org. 2019 [Citerad 14 januari 2019]. Hämtad från: <https://www.gnu.org/philosophy/free-sw.html>
- [57] A Quick Guide to GPLv3- GNU Project - Free Software Foundation [Internet]. Gnu.org. 2019 [Citerad 14 januari 2019]. Hämtad från: <https://www.gnu.org/licenses/quick-guide-gplv3.html>
- [58] GNU Lesser General Public License v3.0- GNU Project - Free Software Foundation [Internet]. Gnu.org. 2019 [Citerad 14 januari 2019]. Hämtad från: <https://www.gnu.org/licenses/lgpl-3.0.en.html>
- [59] Sinclair A. License Profile: BSD. International Free and Open Source Software Law Review [Internet]. 2010 [Citerad 2018 november 20] ;(Issue 1):1. Hämtad från: <http://ludwig.lub.lu.se/login?url=http://search.ebscohost.com/ludwig.lub.lu.se/login.aspx?direct=true&db=edshol&AN=edshol.hein.journals.ifosslr2.4&site=eds-live&scope=site>
- [60] Licenses [Internet]. Apache.org. 2019 [Citerad 15 januari 2019]. Hämtad från: <http://www.apache.org/licenses/>
- [61] Apache License v2.0 and GPL Compatibility [Internet]. Apache.org. 2019 [Citerad 15 januari 2019]. Hämtad från: <http://www.apache.org/licenses/GPL-compatibility.html>
- [62] Steiniger S, Bocher E. An overview on current free and open source desktop GIS developments. International Journal of Geographical Information Science. 2009;23(10):1345-1370.

- [63] Understanding Free Cultural Works - Creative Commons [Internet]. Creative Commons. 2018 [Citerad 03 december 2018]. Hämtad från: <https://creativecommons.org/share-your-work/publicdomain/freeworks/>
- [64] Licensing types - Creative Commons [Internet]. Creative Commons. 2018 [Citerad 03 december 2018]. Hämtad från: <https://creativecommons.org/share-your-work/licensing-types-examples/>
- [65] About The Licenses - Creative Commons [Internet]. Creativecommons.org. 2018 [Citerad 03 december 2018]. Hämtad från: <https://creativecommons.org/licenses/>
- [66] OpenStreetMap [Internet]. OpenStreetMap. 2018 [Citerad 03 december 2018]. Hämtad från: <https://www.openstreetmap.org/copyright>
- [67] About OSGeo - OSGeo [Internet]. OSGeo. 2019 [Citerad 20 februari 2019]. Hämtad från: <https://www.osgeo.org/about/>
- [68] Singh A, Bansal R, Jha N. Open Source Software vs Proprietary Software. International Journal of Computer Applications. 2015;114(18):26-31.
- [69] Open Source Legal Issues, 2008. Report from the CASCADOSS project. Delivery 2.1
- [70] Ebert C. Open Source Drives Innovation. IEEE Software. 2007;24(3):105-109.
- [71] Ven K, Verelst J, Mannaert H. Should You Adopt Open Source Software?. IEEE Software. 2008;25(3):54-59.
- [72] Nagy D, Yassin A, Bhattacharjee A. Organizational adoption of open source software. Communications of the ACM. 2010;53(3):148.
- [73] Damiani E, Ardagna C, Ioini N. Open source systems security certification. New York: Springer; 2009.
- [74] Goode S. Something for nothing: management rejection of open source software in Australia's top firms. Information & Management. 2005;42(5):669-681.
- [75] Projects Archive - OSGeo [Internet]. OSGeo. 2018 [Citerad 03 december 2018]. Hämtad från: <https://www.osgeo.org/projects/>
- [76] FME [Internet]. Esri.se. 2018 [Citerad 20 november 2018]. Hämtad från: <http://www.esri.se/Produkter/ovriga-produkter/fme>
- [77] FME | Data Integration Platform | Safe Software [Internet]. Safe.com. 2018 [Citerad 20 november 2018]. Hämtad från: <https://www.safe.com/how-it-works/>
- [78] [Internet]. Cdn.safe.com. 2019 [Citerad 9 maj 2019]. Hämtad från: <https://cdn.safe.com/resources/brochures/SafeSoftware-FactSheet.pdf>
- [79] Category: Building Maps [Internet]. Carto.com. 2018 [Citerad 04 december 2018]. Hämtad från: <https://carto.com/help/building-maps/>
- [80] APIs, SDKs and Apps | ArcGIS for Developers [Internet]. Developers.arcgis.com. 2018 [Citerad 18 december 2018]. Hämtad från: <https://developers.arcgis.com/documentation/core-concepts/apis-sdks-apps/>

- [81] ArcGIS API for JavaScript Sandbox [Internet]. Developers.arcgis.com. 2018 [Citerad 18 december 2018]. Hämtad från: <https://developers.arcgis.com/javascript/latest/sample-code/sandbox/index.html?sample=scene-local>
- [82] Web applications [Internet]. Mapbox. 2019 [Citerad 18 februari 2019]. Hämtad från: <https://docs.mapbox.com/help/how-mapbox-works/web-apps/>
- [83] Overview [Internet]. Mapbox. 2019 [Citerad 18 februari 2019]. Hämtad från: <https://docs.mapbox.com/help/how-mapbox-works/>
- [84] [Internet]. 2019 [Citerad 18 februari 2019]. Hämtad från: <https://docs.mapbox.com/studio-manual/overview/><https://docs.mapbox.com/studio-manual/overview/>
- [85] About [Internet]. Mapbox. 2019 [Citerad 18 februari 2019]. Hämtad från: <https://www.mapbox.com/about/>
- [86] About - GeoServer [Internet]. Geoserver.org. 2018 [Citerad 04 december 2018]. Hämtad från: <http://geoserver.org/about/>
- [87] Welcome to MapServer — MapServer 7.2.1 documentation [Internet]. Mapserver.org. 2019 [Citerad 18 februari 2019]. Hämtad från: <https://mapserver.org/index.html>
- [88] GDAL: GDAL - Geospatial Data Abstraction Library [Internet]. Gdal.org. 2019 [Citerad 15 februari 2019]. Hämtad från: <https://www.gdal.org/>
- [89] About — MapServer 7.2.1 documentation [Internet]. Mapserver.org. 2019 [Citerad 18 februari 2019]. Hämtad från: <https://mapserver.org/about.html#about>
- [90] QGIS as OGC Data Server [Internet]. Docs.qgis.org. 2019 [Citerad 20 februari 2019]. Hämtad från: https://docs.qgis.org/2.8/en/docs/user_manual/working_with_ogc/ogc_server_support.html
- [91] OpenLayers - Welcome [Internet]. Openlayers.org. 2018 [Citerad 27 november 2018]. Hämtad från: <http://openlayers.org/>
- [92] OpenLayers - Welcome [Internet]. Openlayers.org. 2019 [Citerad 20 Februari 2019]. Hämtad från: <https://openlayers.org/>
- [93] OpenLayers - 3rd party [Internet]. Openlayers.org. 2019 [Citerad 20 Februari 2019]. Hämtad från: <https://openlayers.org/3rd-party/>
- [94] OpenLayers - OSGeo [Internet]. OSGeo. 2019 [Citerad 20 Februari 2019]. Hämtad från: <https://www.osgeo.org/projects/openlayers/>
- [95] OpenLayers - 3rd party [Internet]. Openlayers.org. 2019 [Citerad 11 April 2019]. Hämtad från: <https://openlayers.org/3rd-party/>
- [96] three.js - Javascript 3D library [Internet]. Threejs.org. 2019 [Citerad 11 April 2019]. Hämtad från: <https://threejs.org/>
- [97] About CesiumJS - Open Source 3D Globe and Maps [Internet]. Cesiumjs.org. 2019 [Citerad 11 April 2019]. Hämtad från: <https://cesiumjs.org/about/>

- [98] Leaflet — an open-source JavaScript library for interactive maps [Internet]. Leafletjs.com. 2018 [Citerad 27 november 2018]. Hämtad från: <https://leafletjs.com/>
- [99] Leaflet - OSGeo [Internet]. OSGeo. 2019 [Citerad 18 februari 2019]. Hämtad från: <https://www.osgeo.org/projects/leaflet/>
- [100] Open ePlatform - Teknik [Internet]. Oeplatform.org. 2018 [Citerad 26 november 2018]. Hämtad från: <http://www.oeplatform.org/teknik>
- [101] MapGuide Project Home | MapGuide Open Source [Internet]. Mapguide.osgeo.org. 2019 [citerad 11 januari 2019]. Hämtad från: <https://mapguide.osgeo.org/>
- [102] About | MapGuide Open Source [Internet]. Mapguide.osgeo.org. 2019 [citerad 11 januari 2019]. Hämtad från: <https://mapguide.osgeo.org/about.html>
- [103] Applications [Internet]. Qgis.org. 2019 [Citerad 15 februari 2019]. Hämtad från: <https://www.qgis.org/en/site/about/features.html#qgis-server>
- [104] Elasticsearch: RESTful, Distributed Search & Analytics | Elastic [Internet]. Elastic.co. 2019 [Citerad 15 februari 2019]. Hämtad från: <https://www.elastic.co/products/elasticsearch>
- [105] dpMap - Digpro [Internet]. Digpro.se. 2019 [Citerad 15 februari 2019]. Hämtad från: <https://www.digpro.se/sv/vara-produkter/stat-kommun/dpmap-sv-se-4>
- [106] Mark Otto a. Bootstrap [Internet]. Getbootstrap.com. 2019 [Citerad 7 maj 2019]. Hämtad från: <https://getbootstrap.com/>
- [107] Font Awesome 5 [Internet]. Fontawesome.com. 2019 [Citerad 7 maj 2019]. Hämtad från: <https://fontawesome.com/>

Institutionen av naturgeografi och ekosystemvetenskap, Lunds Universitet

Student-examensarbete (seminarieuppsatser) i geografisk informationsteknik.

Uppsatserna finns tillgängliga på institutionens geobibliotek, Sölvegatan 12, 223 62 LUND. Serien startade 2010. Hela listan och själva uppsatserna är även tillgängliga på LUP student papers och via Geobiblioteket (www.geobib.lu.se).

Serie examensarbete i geografisk informationsteknik

1. Patrik Carlsson och Ulrik Nilsson (2010) Tredimensionella GIS vid fastighetsförvaltning
2. Karin Ekman och Anna Felleson (2010) Att välja grundläggande karttjänst - Utveckling av jämförelsemodell och testverktyg för utvärdering
3. Jakob Mattsson (2011) Synkronisering av vägdatabaser med KML och GeoRSS - En fallstudie i Trafikverkets verksamhet
4. Patrik Andersson and Anders Jürisoo (2011) Effective use of open source GIS in rural planning in South Africa
5. Nariman Emamian och Martin Fredriksson (2012) Visualisering av bygglovsärenden med hjälp av Open Source-verktyg - En undersökning kring hur man kan effektivisera ärendehantering med hjälp av en webbapplikation
6. Gustav Ekstedt and Torkel Endoff (2012) Design and Development of a Mobile GIS Application for Municipal FieldWork
7. Karl Söderberg (2012) Smartphones and 3D Augmented Reality for disaster management - A study of smartphones ability to visualise 3D objects in augmented reality to aid emergency workers in disaster management
8. Viktoria Strömberg (2012) Volymberäkning i samhällsbyggnadsprojekt
9. Daniel Persson (2013) Lagring och webbaserad visualisering av 3D stadsmodeller - En pilotstudie i Kristianstad kommun
10. Lisette Danebjer och Magdalena Nyberg (2013) Utbyte av geodata - studie av leveransstrukturer enligt Sveriges kommuner och landstings objekttypskatalog
11. Alexander Quist (2013) Undersökning och utveckling av ett mobilt GISsystem för kommunal verksamhet
12. Nariman Emamian (2014) Visning av geotekniska provborrningar i en webbmiljö
13. Martin Fredriksson (2014) Integrering av BIM och GIS med spatiala databaser – En prestandaanalys
14. Niklas Krave (2014) Utveckling av en visualiseringsapplikation för solinstrålningsdata
15. Magdalena Nyberg (2015) Designing a generic user interface for distribution of open geodata: based on FME server technology
16. Anna Larsson (2015) Samredovisning av BIM- och GIS-data

17. Anton Lundkvist (2015) Development of a WEB GI System for Disaster Management
18. Ellen Walleij (2015) Mapping in Agricultural Development – Introducing GIS at a smallholders farmers’ cooperative in Malawi
19. Frida Christiansson (2016) Lagring av 3D - geodata - en fallstudie i Malmö Stad
20. Lisette Danebjer (2016) Methodology for creating and modifying distributed topologically structured geographical datasets
21. Jeanette Dunn Ekelund (2016) En jämförelse av algoritmer och resultat för flödesberäkning i QGIS/GRASS och ArcGIS
22. Ebba Gröndahl och Frida Thorman (2016) Verksamhetens optimala läge i staden och hur de är lokaliserade idag
23. Gunnar Rolander (2017) Data transformation using linked data ontologies
24. Måns Andersson och Moa Eklöf (2017) Stilsättning av geografiska data
25. Josefine Axelsson (2018) Automatisering av bygglovsansökningsprocessen med stöd av BIM och GIS
26. Leonard B. O. Berge (2018) Uppdatering och visualisering av stadsmodell med stöd av konverterade BIM-modeller
27. Rickard Ingesson & Gabriella Olsson (2019) Publicering av geografiska data på webben : En utvärdering av programsystem med fokus på öppen källkod