

Examensarbete i geografisk informationsteknik nr 11

Undersökning och utveckling av ett mobilt GIS-system för kommunal verksamhet

Alexander Quist

Civilingenjörsutbildningen i Lantmäteri
Lunds Tekniska Högskola

Institutionen för Naturgeografi och Ekosystemvetenskap
Lunds Universitet





LUNDS UNIVERSITET
Lunds Tekniska Högskola

Undersökning och utveckling av ett mobilt GIS-system för kommunal verksamhet

EXTM05 Master uppsats, 30 hp
Civilingenjörutbildningen i Lantmäteri

Författare:
Alexander Quist

Handledare:
Lars Harrie
Institutionen för Naturgeografi och Ekosystemvetenskaper

Juni 26, 2013

Opponent: Martin Fredriksson
Examinator: Andreas Persson

Copyright © Alexander Quist, LTH

Institutionen för Naturgeografi och Ekosystemvetenskaper
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 11
Tryckt av E-tryck, E-huset, 2013

Förord

Denna rapport är resultatet av ett examensarbete utfört av Alexander Quist.

Examensarbetet är den sista delen i min femåriga utbildning till civilingenjör inom lantmäteri vid Lunds tekniska högskola, med inriktning mot GIS (geografiska informationssystem).

Examensarbetet utfördes i samarbete med Institutionen för naturgeografi och ekosystemvetenskaper vid Lunds universitet samt Kalmar kommun.

Jag vill tacka alla på samhällsbyggnadskontoret i Kalmar för det mycket trevliga bemötandet inklusive all positiv uppmuntran, samt för den tillgång av teknisk utrustning med lån av kontor och dator som jag har haft tillgång till. Jag vill speciellt tacka Sabina Bernhard och Andreas Wendel som har hjälpt mig få fram de resurser som jag har behövt samt svarat på mina stundtals otaliga frågor.

Jag vill även tacka min handledare i Lund, Lars Harrie, för att han har haft tålamod med mig samt gett mig viktig feedback.

26 juni 2013

Alexander Quist

Abstract

Swedish municipalities today hold large amounts of geographical information, information that is necessary to manage the municipality. In the past all geographical information was stored in paper format, but it has now been digitalized and stored in computers. This thesis has focused on how to take the next step, from the computer to the mobile phone. Today's technological advances in the mobile market have given us an entirely new platform to spread information on. Smartphones are like small laptops that have access to the Internet almost anywhere they are.

This project has focused on how to develop a tool for surveyors, a tool that allows them to reach the information they need in the field. This tool shall have the ability to retrieve information, but also send information to the server. All this will be done from the user's smartphone; it is therefore a mobile application that has been investigated. Although the project is aimed at surveyors it is intended that one should take knowledge from the project, knowledge that can then be applied to other departments within the municipality. The project will therefore present more than one possible solution.

The first step was to collect comprehensive information and conditions about the project. This was followed by interviews with some of the future users. A needs analysis was carried out and their opinions were recorded, which was then used to compile a list of requirements. These specifications determine the requirements for the final product. The requirements are then assigned a respective priority level (between one and three). First priority requirements are a must for the final product's approval, while the second and third are only desired if possible.

After the needs analysis and the priority level is set the next step was to examine already existing similar systems to get knowledge for what is possible to do on a mobile device. It was also examined whether or not there were any already finished products that could be useful in this case. Both apps and so called web apps were studied. A web app is a website that looks and acts like a regular app.

A thorough examination was carried out of the technical background to investigate which solutions that was possible and what they required to work. Some solutions were discarded after further studies while some were studied more throughout. In the end three different solutions where chosen for the server side and three for the client side. Server side is where all the information is located, and the client side is where the user is located. The various solutions were presented, how they interact with each other and their pros and cons. It is also discussed in which situation each solution is most appropriate, and which solution that is most suitable for this very project.

In the end a solution was selected where the server side use both ArcGIS Server and ArcGIS Online, two solutions that Kalmar municipality already has licenses for. On the client side an already finished app was chosen, called ArcGIS Collector. It is free to use. It is an easy

solution since it's using an existing app, but it also works very well in general. You get more flexibility when developing your own app, but it requires time and money that is not quiet justified in this case.

Sammanfattning

Svenska kommuner innehar i dag stora mängder geografisk information, information som är nödvändig för att sköta kommunen. Förr i tiden var all geografisk information i pappersformat, sedan digitalerades den och fördes in i datorn. Detta examensarbete har fokuserat på hur man kan ta nästa steg, från datorn till mobiltelefonen. Dagens tekniska framsteg på den mobila marknaden har gett oss en helt ny plattform att sprida information via. Smartmobiler är som små bärbara datorer som har tillgång till internet nästan var de än befinner sig.

Detta projekt har fokuserat på hur man kan ta fram ett hjälpmedel för mätingenjörer, ett hjälpmedel som gör att de når den information som de behöver även i fält. Detta hjälpmedel ska dels ha möjligheten att hämta information, men den ska även skicka iväg information till en server. Allt detta ska göras från användarens smartmobil, det är m.a.o. en "app" (mobile application) som har undersökts. Trots att projektet är riktat mot mätingenjörer är det tänkt att man ska ta lärdom av projekt och därmed kunna appliceras det på andra avdelningar inom kommunen. Projektet kommer därför att presentera mer än en möjlig lösning.

Först samlades övergripande information om projektet in samt information om kommunens förutsättningar. Därefter genomfördes intervjuer med några av de framtida användarna. En behovsanalys genomfördes och deras synpunkter dokumenterades, vilka senare användes för att sammanställa en s.k. kravspecifikation. Kravspecifikationen bestämmer vilka krav man har på slutprodukten. De uppställda kraven tilldelas sedan varsin prioriteringsgrad (mellan ett och tre). Första prioritet är krav som är ett måste för slutprodukten godkännande, medan andra och tredje bara är önskvärda om möjligt.

Efter detta undersöktes liknande system för att få en uppskattning om vad man har möjlighet att utföra mobilt. Det undersöktes även om det fanns några redan färdiga produkter som skulle vara användbara i detta fall. Både appar och s.k. webbappar studerades. En webbapp är en hemsida som ter sig som en vanlig app.

Det genomfördes en grundlig undersökning av den tekniska bakgrunden, vilka lösningar som var möjliga och vad dessa skulle kräva. Efter att noggrant ha undersökt de olika alternativen valdes sedan vilka lösningar som var värda att gå vidare med. I slutänden valdes tre olika lösningar för serversidan och tre olika för klientsidan. Serversidan är där all information befinner sig, och klientsidan är där användaren befinner sig. De olika lösningarna presenterades, dels hur de fungerar med varandra men även deras för- och nackdelar. Det diskuteras även i vilken situation som de olika lösningarna är mest lämpade, samt vilken lösning som är bäst för detta projekt.

I slutänden valdes en lösning på serversidan där man använder både ArcGIS Server och ArcGIS Online, två lösningar som Kalmar kommun redan har licens för. På klientsidan valdes en färdig app som heter ArcGIS Collector. Den är gratis att använda. Det är en enkel och bra lösning då man använder en befintlig app, som även ansågs ha en god användningsförmåga i

övrigt. Fördelen med att utveckla sin egen app är att man får större flexibilitet, detta kräver dock både tid och pengar vilket inte kändes motiverat i detta fall.

Förkortningar

AGO – ArcGIS Online

En webbservice för att publicera kartor på internet.

API – Application programming interface

Ett protokoll som används för att kommunicera med mjukvaran.

DBMS – Database management system

Program som används för att kommunicera med en databas.

DMZ – Demilitarized zone

Ett separat och kontrollerat nätverk som agerar mellanhand mellan ett internt nätverk och omvärlden.

FME – Feature manipulation engine

Program som innehåller flertalet verktyg för att transformera och översätta geografisk information.

GIS – Geografiskt informationssystem

Ett system för att t.ex. skapa, analysera, förvalta och förmedla geografisk information.

GPS – Global positioning system

Ett navigeringssystem som m.h.a. satelliter beskriver användarens position.

HTML – HyperText Markup Language

Är det primära programmeringsspråket som används för att skapa hemsidor.

RDBMS – Relational database management system

Program som används för att kommunicera med en relationsdatabas.

SDK – Software development kit

Innehåller ett antal verktyg som möjliggör utvecklingen av mjukvara för t.ex. något specifikt programvarupaket eller operativsystem.

Innehåll

1. Inledning.....	1
1.1. Bakgrund.....	1
1.2. Problemställning.....	2
1.3. Syfte	2
1.4. Metod	2
1.5. Målgrupp	3
1.6. Disposition	3
2. Kalmar kommun	4
2.1. Mätavdelningen.....	4
2.2. GIS-avdelningen.....	4
2.3. Kommunikationen mellan mätavdelningen och GIS-avdelningen	5
2.4. Baskartan	5
2.5. Stomnätspunkter/Stomnätsskisser	5
2.6. Förvaltning och spridning av geografisk information.....	6
2.7. Illustration.....	6
3. Behovsanalys.....	8
3.1. Intervjuer	8
3.2. Slutsats.....	9
3.2.1. Ej genomförbara krav.....	9
3.2.2. Beslut	9
4. Kravspecifikation	10
4.1. Krav	10
4.2. Förklaring.....	10
4.2.1. Första prioritet	10
4.2.2. Andra prioritet	11
4.2.3. Tredje prioritet.....	11
5. Liknande system	12
5.1. Felanmälan – Kalmar kommun.....	12
5.2. iSurvey – Örebro kommun.....	13
5.3. Parkering – Helsingborg kommun	14

5.4. ArcGIS webblösningar.....	15
5.4.1. Mobile template	15
5.4.2. Geolocation.....	16
5.5. CarryMap Observer	17
5.6. GeoMobile for ArcGIS.....	18
5.7. ArcGIS appar	19
5.7.1. ArcGIS (app)	19
5.7.2. ArcGIS Collector	20
5.8. Sammanfattning	21
6. Teknisk bakgrund	22
6.1. Mobila klienter.....	23
6.1.1. Smartmobiler	23
6.1.2. Läsplattor	24
6.1.3. Andra alternativ	24
6.2. Operativsystem.....	24
6.2.1. iOS	25
6.2.2. Android.....	25
6.2.3. Övriga	25
6.3. Appar	25
6.3.1. "Klassiska" appar.....	25
6.3.2. Webbappar	26
6.4. GIS-program.....	26
6.4.1. ArcGIS.....	26
6.4.2. Utveckla en egen app med ArcGIS.....	28
6.4.3. Öppen källkod	30
6.5. Lagring av geografiska data	31
6.5.1. Spatial databas	32
6.5.2. Molnbaserade tjänster.....	32
6.6. Säkerhet	33
6.6.1. Intranät kontra internet.....	33
6.6.2. Inloggningssystem.....	34
6.6.3. Säker förvaltning	35

6.6.4. Mobil säkerhet	35
7. Licensfrågor på programvara	38
7.1. Proprietära program.....	40
7.2. Öppet källkodsprogram	40
7.2.1. Licenser	41
8. Licensfrågor på data	43
8.1. Inledning	43
8.2. Digital rights management	43
8.3. Fria data	45
8.3.1. Creative commons	45
8.3.2. ODbL.....	47
8.3.3. VGI-data	47
8.3.4. OpenStreetMap	48
9. Genomgång av lösningsalternativ	51
9.1. Hårdvara	51
9.2. Mjukvara	51
9.3. Val av lösningsalternativ	52
9.3.1. Serversidan.....	52
9.3.2. Klientsidan.....	52
9.4. Lösning 1 – ArcGIS Online	53
9.4.1. Funktionalitet.....	53
9.4.2. Åtkomst.....	55
9.4.3. Implementation	56
9.5. Lösning 2 – ArcGIS Server	56
9.5.1. Licens.....	57
9.5.2. Databas	57
9.5.3. Funktionalitet.....	57
9.5.4. Åtkomst.....	59
9.5.5. Implementation	60
9.6. Lösning 3 – Hybriden	60
9.6.1. Funktionalitet.....	61
9.6.2. Åtkomst.....	62

9.6.3. Implementation	63
10. Funktionalitet	64
10.1. ArcGIS (app)	64
10.1.1. Användargränssnitt.....	64
10.1.2. Verktyg	66
10.1.3. Implementerat	66
10.2. ArcGIS Collector	66
10.2.1. Användargränssnitt.....	67
10.2.2. Verktyg	69
10.2.3. Implementerat	69
10.3. Webbapp	69
10.3.1. Automatiskt genererad	69
10.3.2. Att skapa en egen	69
10.3.3. Användargränssnitt.....	70
10.3.4. Verktyg	70
10.3.5. Implementerat	71
11. Val av tekniska lösningar	73
11.1. Lösningar – Serversidan.....	73
11.1.1. Lösning 1 – ArcGIS Online	73
11.1.2. Lösning 2 – ArcGIS Server	73
11.1.3. Lösning 3 – Hybriden.....	74
11.1.4. För- och nackdelar	74
11.2. Lösningar – Klientsidan	75
11.2.1. För- och nackdelar	79
11.3. Slutprodukten	80
11.3.1. Serversidan.....	80
11.3.2. Klientsidan.....	81
12. Diskussion	82
12.1. Vidareutveckling	82
12.2. Vidareutveckling mot fria data	83
13. Slutsatser	85

Tidigare examensarbete

Referenser

Bilaga A – Källkod webbapp

Figurlista

2.1: Baskartan	5
2.2: Stomnätsskiss	6
2.3: Kommunikationsskiss	7
5.1: Felanmälan Kalmar kommun (Kalmar kommun, 2013)	12
5.2: iSurvey (Ekstedt och Endoff, 2012)	13
5.3: iSurvey (Helsingborg kommun, 2013)	14
5.4: Mobile Template (ArcGIS, 2013b)	15
5.5: Geolocation API (ArcGIS, 2013c)	16
5.6: CarryMap Observer for iPhone (CarryMap, 2013)	17
5.7: GeoMobile for ArcGIS for iPhone (GeoMobile for ArcGIS, 2013)	18
5.8: ArcGIS for iPhone (ArcGIS, 2013d)	19
5.9: ArcGIS Collector (ArcGIS, 2013e)	20
6.1: Server och klient	22
6.2: Antalet användare av smartmobiler i Sverige, Think with Google (2013)	24
6.3: DMZ	34
7.1: Olika typer av licenser	42
8.1: DRM	44
8.2: De sex Creative Commons-licenserna	46
8.3: OpenStreetMap	48
8.4: Att skapa ny data för OpenStreetMap (OpenStreetMap, 2013b)	49
9.1: Konfigurera bild	54
9.2: Lösning 1 – ArcGIS Online	55
9.3: Lösning 2 – ArcGIS Server	58
9.4: Exempel på en hybridlösning	60
9.5: Lösning 3 – Hybriden	62
10.1: ArcGIS (app) – Översikt	64
10.2: ArcGIS (app) – Editering	65
10.3: ArcGIS Collector – Översikt	67
10.4: ArcGIS Collector – Editering	68
10.5: Den utvecklade webbappen	70
10.6: Kommunikationsskiss webbappen	72

1. Inledning

1.1. Bakgrund

Att ha tillgång till information var vi än befinner oss har i dag blivit en självklarhet. Detta är dock ett relativt nytt behov. Detta "beroende" av information är något som har ökat i och med smartmobilens introduktion. Detta projekt är gjort i samarbete med Kalmar kommun och kommer därmed utföras ur kommunens synvinkel. De som arbetar på Kalmar kommun har i arbetsuppgift att "driva" kommunen. Då kommunen i sig är en fysisk plats leder detta till en hel del fältarbete. Detta projekt kommer att fokusera på hur man kan underlätta detta fältarbete genom att göra mer information tillgänglig mobilt. Att hitta en lösning som passar för alla inom kommunen är omöjligt. Detta projekt har därför valt att fokusera på mätningenjörerna.

Mätningenjörer är ansvariga för att både mäta in och sätta ut detaljer. De har exempelvis som arbetsuppgift att mäta in alla nya centrala fastigheter. Men det är även de som är där och mäter ut fastigheten innan den skapas, dvs. placerar ut gränserna. Till sin hjälp har de en rad olika mätinstrument, t.ex. GPS-mottagare och totalstationer.

De mätinstrument som brukas av mätningenjörer är väldigt avancerade, men det är fortfarande en papperskarta som de tar med sig ut i fält. I många fall är detta tillräckligt, men det kan uppstå problem om något oväntat inträffar. Detta beror på att papperskartan är begränsad till de val av information som man valt redan innan man åkte ut i fält. Att man idag fortfarande bär med sig en papperskarta är aningen primitivt. Papperskartan har två syften. Det är dels en vägvisare men det är även ett "anteckningsblock". Med vägvisare menas att det är ett hjälpmedel för att tolka det man ser, men det är även ett viktigt verktyg för att lokalisera det man inte ser. Exempel på detta kan vara fastighetsgränser eller nergrävda avloppsrör. Med anteckningsblock menas att mätningenjören skissar det som de har observerat/mätt i fält, kompletterat med mätvärden eller andra kommentarer.

Dagens teknik när det gäller smartmobiler och läsplattor har dock gett oss en ny plattform att sprida information via. Plattformens resurser är självklart väldigt begränsade, men man kan ändå se de som begränsade datorer. Det är inte önskvärt att mätningenjörerna ska ha en laptop med sig, en smartmobil är därför ett enklare alternativ. Smartmobiler och läsplattor kan nå geografisk information om de har tillgång till internet, detta kräver dock en mellanhand. En mellanhand kan vara ett program som man installerar på sin mobil, men det kan även vara en hemsida som är specialbyggd just för detta ändamål. Mellanhandens roll är att hämta information från databasen och sedan omvandla den till en karta.

Läsplattor har en mycket större skärm än smartmobiler och är därmed mer lämpade för att presentera en karta. Att man väljer det ena behöver dock inte betyda att man utesluter det andra. Smartmobiler och läsplattor är i grund och botten väldigt lika, de bygger på samma teknik och operativsystem. Exempelvis om man har ett system som är anpassat för

smartmobiler betyder detta med största sannolikhet att det även är kompatibelt med läsplattor. Efter Kalmar kommuns önskemål har smartmobiler valts som plattform. Anledningen till detta är att alla mätingenjörer i dag redan har en smartmobil som arbetstelefon.

1.2. Problemställning

Papperskartor är enkla och bra på många sätt, men det kan behövas ett komplement som gör att man kan nå geografisk information även ute i fält. Detta för att man exempelvis ska nå kompletterande information som man inte fick med sig, t.ex. någon extern skiss. Det kan även vara information om den papperskarta som man har framför sig, vilken noggrannhet är t.ex. en elstolpe inmätt med.

Förutom att hämta information är det även önskvärt att använda det nya systemet som ett anteckningsblock där ens "anteckningar" automatiskt hamnar i kommunens databas.

Man behöver alltså skapa någon form av system för smartmobiler som har tillgång till Kalmar kommuns geografiska information.

1.3. Syfte

Att utveckla en mobil kartapplikation som har möjlighet att läsa från en kartdatabas. Men även möjligheten att skriva till databasen ska undersökas, dvs. lägga till eller ändra existerande punkter. Något som gör att man vidare måste undersöka säkerheten i systemet.

Det ska också undersökas möjligheten att bygga vidare på projektet. Även om det bara är en mindre grupp människor inom kommunen som kommer att använda slutprodukten, kan man tänka sig att det finns andra med liknande behov. Exempelvis trafikavdelningen kanske har behov av en liknande lösning, fast med en annan grundkarta.

Sammanfattningsvis är syftet att:

- Skapa ett hjälpmedel för att nå kommunens kartor i användarens smartmobil.
- Kunna ändra/uppdatera en karta från användarens smartmobil.
- Undersöka hur man kan ta lärdom av projektet och bygga vidare på det.

1.4. Metod

Det system som ska utvecklas är en "app", dvs. en mobilapplikation. En app är smartmobilernas motsvarighet till ett datorprogram.

Första fasen i utvecklingen blir att upprätta en kravspecifikation med hjälp av dem som ska använda det färdiga systemet (mätavdelningen och GIS-avdelningen). Det är viktigt att undersöka och rangordna de uppställda kraven. Vissa krav är ett måste för slutproduktens

godkännande, medan andra endast är önskvärda om tiden tillåter. Det kommer även finnas krav som kommer bli omöjliga att genomföra, och de är då viktiga att identifiera så snabbt som möjligt.

I andra fasen undersöks om det finns några liknande system i nuläget, samt studera vilka olika tekniker de använder. Efter att ha studerat de tillgängliga teknikerna noggrant ska ett val av system göras, dvs. vilka tekniker som systemet ska utvecklas med. Enligt kommunens önskan ska det främst undersökas hur man på bästa sätt kan lösa problemet med deras nuvarande programvara, dvs. ArcGIS som de har licens på. Möjligheten att i stället använda sig av program med öppen källkod ska dock också undersökas.

Tredje fasen består av själva utvecklingen av produkten. Här behöver man även se över de uppställda kraven på slutprodukten. Vilka krav som det inte finns tid att uppfylla, samt vilka krav som har visat sig vara omöjliga att uppfylla.

I fjärde och sista fasen kommer slutprodukten att utvärderas, hur pass produkten lever upp till de uppställda kraven.

1.5. Målgrupp

Denna uppsats är riktad till personer som har studerat eller arbetat med GIS. Läsaren förväntas ha en insikt i hur ett GIS-system är uppbyggt samt kunskap om grundläggande termer.

1.6. Disposition

Rapporten inleds med en grundläggande beskrivning av projektet. Därefter följer en förklaring av arbetssättet inom GIS-avdelningen (Kalmar kommun) och deras behov. Efter det följer behovsanalysen och kravspecifikationen som ligger till grund för projektet. Sedan undersöks liknande existerande system. Den tekniska bakgrunden undersöks, vilka tillgängliga tekniker som man kan använda och hur de fungerar. Licensfrågor för programvara och data undersöks. Därefter diskuteras de tekniska lösningar som övervägts i projektet, samt deras för- och nackdelar. Detta följs av motivering av vilka lösningar som valdes, samt diskussion av processen och slutprodukten. Rapporten avslutas med slutsatser.

2. Kalmar kommun

Sveriges kommuner har mycket gemensamt vad gäller GIS, då grundtanken är i stort sett densamma. Man vill på bästa sätt kartlägga kommunen och dess invånare, vad som gäller för en kommun gäller därför även för andra. Det finns dock skillnader, det beror främst på kommunens storlek och deras tekniska val. Kommunens storlek bestämmer dels behovet av att ha bra information, men det bestämmer även budgeten. Med tekniska val menas t.ex. vilken programvara som används och hur uppdaterad den är. Detta projekt fokuserar på Kalmar kommun. Flertalet slutsatser är dock generella och gäller även andra kommuner och företag.

GIS är något som är på stark frammarsch inom Sveriges kommuner, vissa kommuner har kommit längre än andra. Många små kommuner är fortfarande i startgroparna medan större kommuner redan har väl utvecklade system. Kalmar kommun kan klassas som en mellanstor kommun, men de har trots detta ett väl utvecklat GIS-system.

I projektet ska det undersökas hur man på bästa sätt kan sprida kommunens geografiska information till mobila enheter, detta gäller främst kommunens egen baskarta. Det ska även undersökas möjligheten att komplettera denna lösning med stornätsskisser samt enklare verktyg för editering. Det sistnämnda är främst för att underlätta kommunikationen mellan mätavdelningen och GIS-avdelningen.

2.1. Mätavdelningen

Det är mätavdelningens arbete att mäta in alla förändringar i kommunen, som sedan ska uppdateras i kommunens databas. De mäter först i fält och lämnar sedan över sin data till en editör, som uppdaterar databasen. Editörerna behöver inte tillhöra varken mätavdelningen eller GIS-avdelningen. Vid skapande av nya fastigheter eller nybyggnationer är det även mätavdelningen som mäter ut (placerar) de nya gränserna.

Det är på mätavdelningens önskemål som detta projekt har initierats. De har uttryckt önskemål för ett verktyg som kan användas i fält för att nå baskartan, dels för att ta del av kompletterande information men även för att editera den.

2.2. GIS-avdelningen

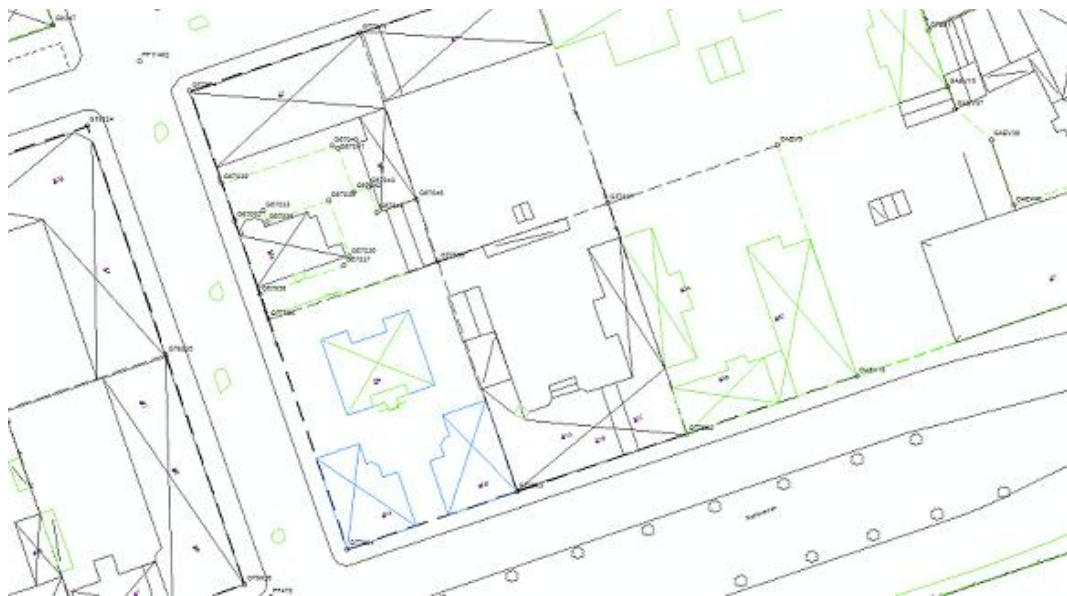
GIS-avdelningens syfte är att förvalta, uppdatera och förmedla den geografiska information som kommunen besitter. Detta eftersom det krävs speciell utbildning och erfarenhet för att bearbeta geografisk information i dess råa form. Deras arbetsuppgifter kan vara allt från att tillhandahålla en mall till att leverera en komplett tjänst eller karta. De ansvarar även för kommunens GIS-programvara.

2.3. Kommunikationen mellan mätavdelningen och GIS-avdelningen

Den kommunikation som ska utredas närmre är den mellan avdelningarna när en mätingenjör i fält inser att kartan är felaktig. För det första måste man då rapportera detta till GIS-avdelningen, något som enkelt kan glömmas bort. När felet väl är rapporterat måste GIS-avdelningen kontrollera om detta stämmer, och det innebär ibland att de måste åka ut på plats. Först efter denna kontroll kan de ändra i kartan. Målet är att undersöka om man kan underlätta denna process, t.ex. genom att låta mätingenjören infoga en bild i kartan.

2.4. Baskartan

Baskartan är upprättad och uppdateras av Kalmar kommun. Den är en visualisering av kommunens egna geografiska information. Det finns ett antal mallar och regler för hur kartan ska visas.

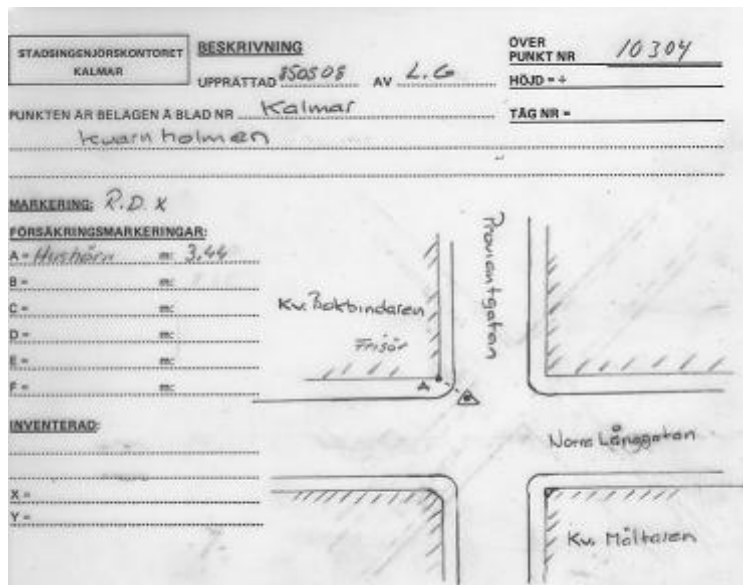


Figur 2.1: Baskartan

I figur 2.1 kan man se att olika linjer har olika färg, detta visar hur de är inmätta. Det är därför viktigt att kartan i mobilen ser identisk ut med hur den ser ut i datorn, för att mätningenjörerna ska ha möjlighet att tolka informationen. Det är väldigt viktigt för dem att veta hur en detalj är inmätt, för det i sin tur bestämmer hur noggrant bestämd den är.

2.5. Stomnätspunkter/Stomnätsskisser

Stomnätspunkter är fysiska punkter som är väl bestämda (deras koordinater är inmätta med god noggrannhet), därför använder man dem som utgångspunkt vid mätningar. För att man lättare ska hitta dem finns det *stomnätsskisser* (figur 2.2), som främst innehåller en karta över var punkten befinner sig, men den kan även innehålla punktens höjd.



Figur 2.2: Stomnätsskiss

2.6. Förvaltning och spridning av geografisk information

Den geografiska informationen är lagrad i kommunens intranät, den kan enkelt nås av de kommunanställda (med tillräckliga rättigheter). Det finns även en webbportal som man kan använda för att hämta geografisk information (också bara tillgänglig för datorer i intranätet).

ArcGIS

Kalmar kommun har en kommunlicens på ArcGIS, som innehåller flertalet ArcGIS-produkter. Det är främst ArcGIS som används för att arbeta med den geografiska informationen.

Öppen källkod

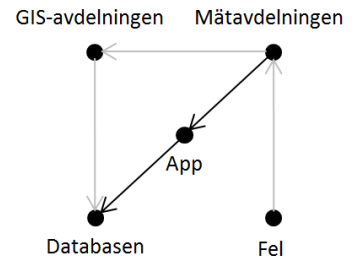
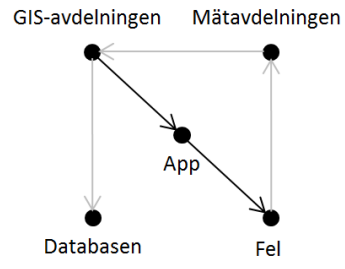
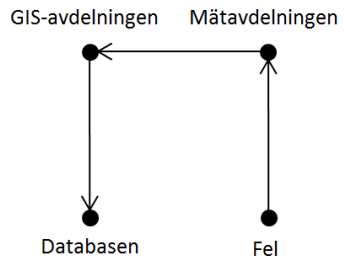
Kommunen använder inte i dag någon programvara med öppen källkod.

FME

FME är en förkortning av *Feature Manipulation Engine* och är ett hjälpmedel för att bearbeta geografisk data, eller manipulera som namnet antyder. Kalmar kommun har i dag licens på FME och använder det till ett antal olika tjänster, men FME anses inte vara applicerbart i detta projekt.

2.7. Illustration

Syftet med det system som ska utvecklas är att underlätta arbetet för mätavdelningen och GIS-avdelningen. För GIS-avdelningen innebär detta att de lättare ska kunna avgöra om en punkt i kartan ska ändras eller inte. Den nya lösningen ska ge de koordinater, beskrivning och bild. För mätavdelningen underlättas deras arbete genom att de får tillgång till baskartan i sin smartmobil, något som de inte har tillgång till i nuläget. Se figur 2.3 för hur den tänkta appen ska användas för att skapa nya kommunikationsvägar.



Figur 2.3: *Kommunikationsskiss*

3. Behovsanalys

För att fastställa behovet av mobila kartlösningar genomfördes intervjuer med representanter för de olika avdelningarna som ska använda slutprodukten, varje avdelning hade en representant. De avdelningar som var representerade var *mätavdelningen*, *GIS-avdelningen* och kommunens *lantmäteri*. Författaren ser även möjligheter till att bygga vidare på projektet för att involvera t.ex. *trafikavdelningen*, *miljöavdelningen* och *bygglovsavdelningen*. Detta är dock något som anses ligga i framtiden då deras behov är väldigt annorlunda från de övriga avdelningarna.

Först samlade författaren in övergripande information om projektet. Därefter hölls tre separata intervjuer med de tre representanterna. Vid dessa intervjuer fick representanterna först information om hur författaren såg på projektet, sedan diskuterades till vilken grad de delade denna bild. De fick komma med egna åsikter, de ombads förklara vad som var extra viktigt ur deras (avdelningens) synvinkel och varför. Dessa åsikter antecknades och återfinns nedan.

3.1. Intervjuer

De olika representanterna frågades ut om de ser ett behov av projektet, men även vad de anser vara viktigt för projektets framgång. Detta är något som ligger till grund för kravspecifikationen. Även författarens egna åsikter finns redovisade.

Mätavdelningen

- Slutprodukten kan bli mycket användbar i fält.
- Vill främst ha tillgång till baskartan.
- Det är mycket viktigt att färgerna för baskartan är desamma som i datorn, då de representerar hur exakt inmätt en detalj är (eller rent utav digitaliserad).
- Vill om möjligt ha tillgång till stomnätsskisser, som finns som externa bilder.
- Det ska vara så användarvänligt som möjligt då det ibland kan vara tuffa förhållanden i fält.
- Möjlighet att editera i fält.

GIS-avdelningen

- Anser att slutprodukten kan underlätta både för dem och mätingenjörerna.
- Systemet skall utvecklas för smartmobiler.
- Tycker att det är önskvärt om man kan nå stomnätsskisser.
- Man ska ha möjlighet att skapa punkter i kartan. Till dessa punkter ska det sedan finnas möjlighet att mata in en kort beskrivning och att bifoga en bild.

Lantmäteriet

- Anser sig bara ha nytta av slutprodukten om den innefattar förrättningsakter.
- Vill ha möjligheten att nå förrättningsakter.

Författarens

- Anser att projektet kommer att leda till något lönsamt för uppdragsgivaren.
- Någon form av säkerhetssystem måste upprättas, t.ex. att en användare måste logga in för att få tillgång till informationen.
- Systemet ska via smartmobilens GPS hitta sin position på kartan.

3.2. Slutsats

3.2.1. Ej genomförbara krav

- Mätavdelningens krav på att ha möjlighet att editera i fält. Smartmobiler är bra på många sätt, men dess pekskärm får anses vara aningen "klumpig". Det går inte att få samma precision som med en datormus, eftersom fingret är så pass stort i jämförelse med hur liten pekskärmen är. Detta kan även leda till ett förhastat beslut, därför är det önskvärt att alla ändringar kontrolleras innan de genomförs. För att detta krav ska implementeras måste man även upprätta ett säkerhetssystem, som gör det möjligt att backa/ångra ändringar.
- Lantmäteriets krav på att ha tillgång till förrättningsakter i fält kommer inte att inkluderas. Ett stort problem är att de är svåra att tillhandahålla p.g.a. juridiska skäl, men även för att de i nuläget är sparade i ett format som inte stöds av smartmobiler (DjVu-format, "déjà vu"). Det går dock att kringgå detta problem om man installerar ett extern program, men det är då något som varje användare skulle behöva installera på egen hand. Säkerhetskraven på systemet skulle dock bli enorma då det är väldigt känslig information som berörs. Om systemet skulle leda till ett säkerhetshål kan Kalmar kommun bli fällida enligt bl.a. Personuppgiftslagen (PuL).

3.2.2. Beslut

Mätavdelningen och GIS-avdelningen får anses ha behov av projektet, och det skall därmed genomföras. Detta är skäl nog, eftersom det är främst för dem som projektet genomförs.

4. Kravspecifikation

Detta kapitel hanterar de krav som är uppställda på slutprodukten. Kraven är uppställda i enlighet med den behovsanalys som genomförts. De krav som inkom har rangordnats i *första, andra* och *tredje prioritet*. *Första prioritet* innebär att kraven måste ha genomförts för att projektet ska ses som godkänt. *Andra-* och *tredje* prioritetens krav är önskvärda och ska därmed försöka implementeras, men de är inte ett måste.

4.1. Krav

Första prioritet

- 1.1 - Man skall nå baskartan
- 1.2 - Systemet skall utvecklas för smartmobiler
- 1.3 - Kartan skall se identisk ut i mobilen som i datorn
- 1.4 - Kartan skall skyddas av ett säkerhetssystem

Andra prioritet

- 2.1 - Man skall ha möjligheten att lägga till punkter i kartan, med beskrivning och ev. foto
- 2.2 - Systemet ska via smartmobilens GPS hitta sin position på kartan
- 2.3 - Det skall vara så användarvänligt som möjligt

Tredje prioritet

- 3.1 - Man skall kunna klicka på stornätspunkter i kartan och få upp stornätsskisserna

4.2. Förklaring

4.2.1. Första prioritet

1.1 - Man skall nå baskartan

Detta är huvudmålet med projektet, dvs. att man enkelt ska nå baskartan via sin smartmobil.

1.2 - Systemet skall utvecklas för smartmobiler

Valet av smartmobiler som plattform beror på att alla anställda (på samhällsbyggnadskontoret i Kalmar) i dag har en smartmobil som arbetstelefon. Inköpskostnaden blir därmed minimal i jämförelse med om man skulle köpa in läsplattor till alla tänkbara användare.

1.3 - Kartan skall se identisk ut i mobilen som i datorn

Mätingenjörerna anser att det är väldigt viktigt att man direkt i kartan kan se vilken noggrannhet som gäller. Författaren anser även att det är viktigt att kartan ser identisk ut i mobilen som i datorn, för att de som nyttjar systemet ska känna igen sig när de byter där

emellan. Användarna är vana vid hur kartan "ska" se ut, vilket kan leda till ett onödigt distraktionsmoment.

1.4 - Kartan skall skyddas av ett säkerhetssystem

Chansen att utomstående försöker få tillgång till systemet får anses vara låg, men någon form av säkerhetssystem måste ändå implementeras.

4.2.2. Andra prioritet

2.1 - Man skall ha möjligheten att lägga till punkter i kartan, med beskrivning och ev. foto

Mättingenjörerna ska ha möjligheten att göra en "anteckning" i kartan åt GIS-avdelningen. De skapar en punkt i kartan med en kort beskrivning och en bild som "bevis". Det kan t.ex. vara en stomnätspunkt som blivit överasfalterad. Den ska därmed tas bort ur kartan. GIS-avdelningen anser att detta är viktigt då det underlättar kommunikationen mellan dem och mätavdelningen, men det kan även minska deras eget arbete. Detta beror på att de måste dubbelkolla "verkligheten" innan de gör förändringar i databasen, en bild och beskrivning kan användas som substitut för att åka ut på plats.

2.2 - Systemet ska via smartmobilens GPS hitta sin position på kartan

Med ett knapptryck ska man ha möjlighet att hitta sin position och automatiskt zooma in på den. Detta underlättar dels för att hitta sin egen position, men troligtvis även det område som man vill undersöka då man i fält ofta befinner sig just på det område som man är intresserad av.

2.3 - Det skall vara så användarvänligt som möjligt

Detta är alltid väldigt viktigt när man utvecklar ett system, men ibland får användarvänligheten ta ett steg tillbaka för funktionaliteten. Anledningen till att detta anses vara andra prioritet och inte första beror på att systemet som ska skapas inte kommer att skapas helt från grunden, man är därför ofta låst till vissa lösningar.

4.2.3. Tredje prioritet

3.1 - Man skall kunna klicka på stomnätspunkter i kartan och få upp stomnätsskisserna

Detta krav kan bli svårt att genomföra, men det beror helt på vilka tekniska lösningar som man väljer att basera systemet på. Skisserna kan även bli aningen svårlästa på en liten skärm, skisserna är uppskattningsvis mellan fyra till sex gånger större än ytan på smartmobilens skärm.

5. Liknande system

I detta kapitel kommer liknande system som undersökts att redovisas. Systemen har undersökts hur de fungerar och vilka tekniker de använder. Det har även undersökts hur de uppfyller de i avsnitt 4.1 uppställda kraven.

5.1. Felanmälan – Kalmar kommun



Figur 5.1: Felanmälan Kalmar kommun (Kalmar kommun, 2013)

Kalmar kommun har i dag en hemsida där invånarna kan skicka in en felanmälan, det kan vara t.ex. en trasig gatulampa eller ett hål i vägen (Kalmar kommun, 2013). Användaren kan skapa punkter i kartan, samt fylla i informationsfält och bifoga bilder. Hemsidan är en s.k. webbapp, dvs. en hemsida som ter sig som en mobilapp. Eftersom det är en webbapp fungerar den lika bra i en dator som i en smartmobil.

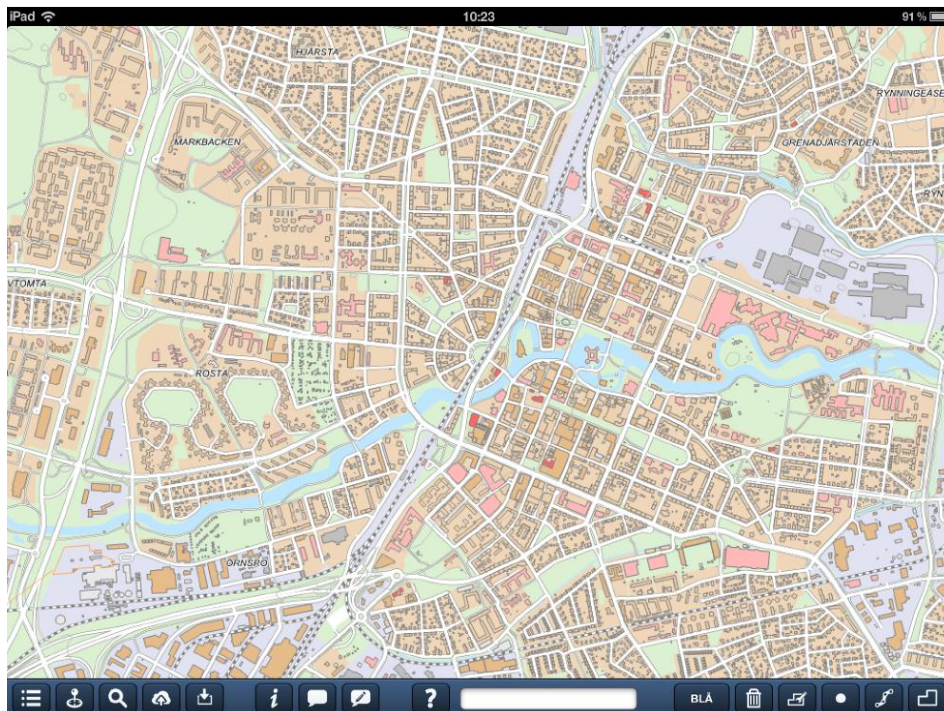
Webbsidan använder sig av *HTML5*, *JavaScript* och *Dojo*. Den geografiska informationen sprids m.h.a. ArcGIS Server. Användarna kan inte se de skapade punkterna.

Serviceförvaltningen kan sedan via intranätet se de skapade punkterna. (Wendel, 2013)

Denna tjänst uppfyller följande krav ur kravspecifikationen:

1.1	1.2	1.3	1.4	2.1	2.2	3.1
✓	✓	✓		✓		

5.2. iSurvey – Örebro kommun



Figur 5.2: iSurvey (Ekstedt och Endoff, 2012)

iSurvey är en produkt för Örebro kommun, den togs fram i ramen av ett examensarbete. Målet var att ersätta papperskartan helt. Bortsett från att man kan se kommunens karta (deras geografiska information) kan man även rita och mäta i kartan. Det som man ritar kan man sedan välja att exportera till databasen. *iSurvey* är en webblösning och är ämnad att användas på en läsplatta, den fungerar även på smartmobiler men det är inte rekommenderat. (Ekstedt och Endoff, 2012)

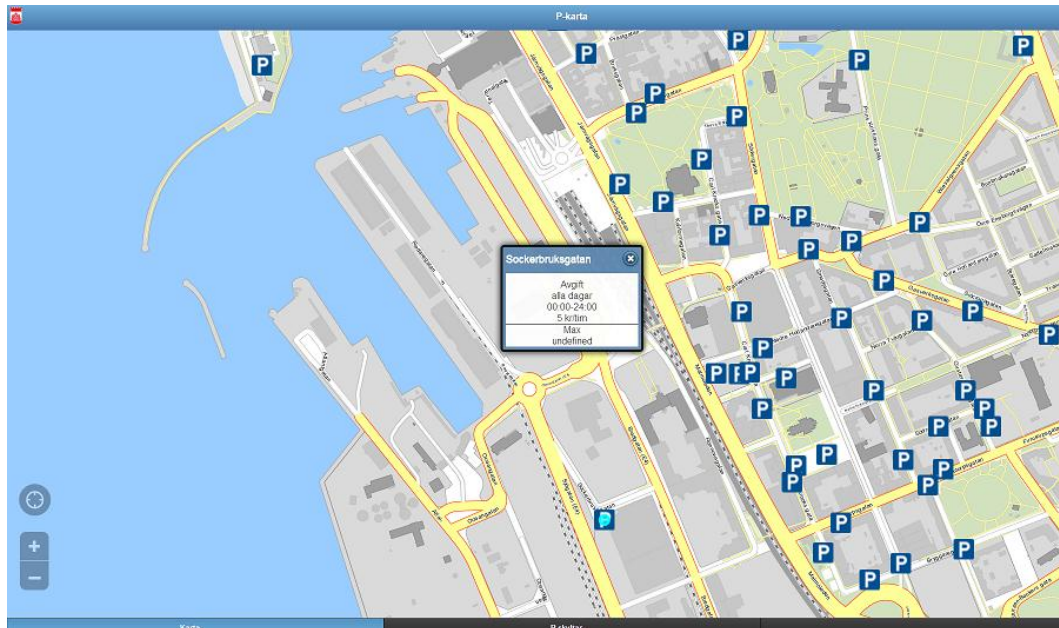
Både server- och klientsidan bygger på lösningar med öppen källkod, dvs. tekniska lösningar som är gratis att använda och fria att modifiera. Exempel på öppen källkod som har använts är *OpenLayers*, *Sencha Touch* och *jQuery mobile* (Ekstedt och Endoff, 2012). Eftersom *Sencha Touch* är licenserat under *GNU GPL* kommer även *iSurvey* att vara det. Det blir därmed fritt för alla att hämta hem, modifiera och återanvända källkoden.

Uppfyller kraven:

1.1	1.2	1.3	1.4	2.1	2.2	3.1
✓		✓		✓ *	✓	✓

* ej foto.

5.3 Parkering – Helsingborg kommun



Figur 5.3: Parkering (Helsingborg kommun, 2013)

Man kan via Helsingborg kommuns hemsida hitta denna webblösning, den visar alla kommunala parkeringsplatser inom kommunen. Om man klickar på en ikon i kartan får man upp ytterligare information om parkeringen, exempelvis när man måste betala och hur mycket. Men det finns även information om de olika formerna av parkeringsplatser. Webblösningen använder sig av ett antal lösningar med öppen källkod, t.ex. *OpenLayers*, *jQuery* och *Proj4J* (Helsingborg kommun, 2013).

Uppfyller kraven:

1.1	1.2	1.3	1.4	2.1	2.2	3.1
	✓	✓			✓	✓

5.4. ArcGIS webblösningar

Via ArcGIS hemsida kan man nå flera exempel på lösningar och färdiga lösningar (*ArcGIS, 2013a*), samtliga är framtagna av ESRI (företaget som utvecklar ArcGIS). Lösningarna är antingen i form av en webblösning eller en riktig mobilapp. Alla lösningar är baserade på att informationen på server-sidan sprids antingen via ArcGIS Online eller ArcGIS server.

De exempel som går att finna via deras hemsida är oftast väldigt "smala". Exempelen är i regel skapade för att endast belysa **en** egenskap. Det finns alltså ingen komplett lösning. Tanken är mer att man ska plocka de egenskaper som man behöver från deras "smörgåsbord" för att sedan bygga ett system själv.

5.4.1. Mobile template



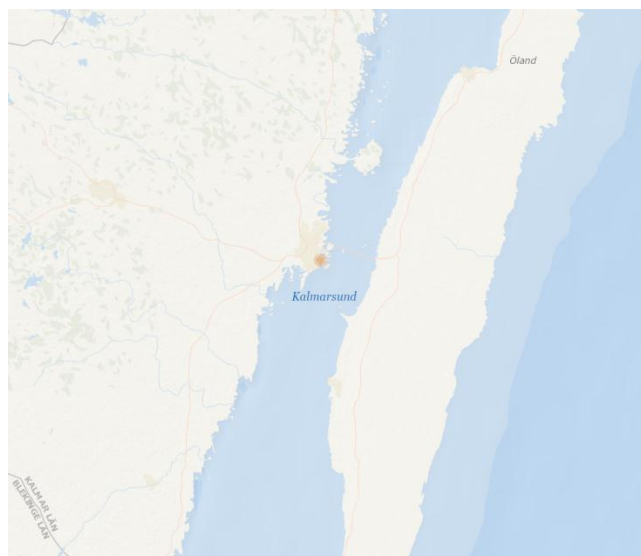
Figur 5.4: *Mobile Template (ArcGIS, 2013b)*

Exemplet *Mobile Template* visar hur man bygger ett system som fungerar på "alla" typer av plattformar (datorer, smartmobiler och läsplattor) (*ArcGIS, 2013b*). Systemet ändrar storlek beroende på vilken typ av plattform man använder. Den kommer även att rotera bilden när en smartmobil eller läsplatta byter riktning, dvs. när den byter mellan *stående* eller *liggande* (engelskans *portrait* eller *landscape*). Detta beror på att webbappen är tillräckligt smart för att detektera vilken typ av plattform (och ev. dess riktning) användaren har. Det är även möjligt att detektera användarens operativsystem och webbläsare.

Uppfyller kraven:

1.1	1.2	1.3	1.4	2.1	2.2	3.1
		✓				

5.4.2. Geolocation



Figur 5.5: Geolocation API (ArcGIS, 2013c)

Detta exempel visar hur man använder enhetens inbyggda GPS för att hitta användarens position. När positionen är funnen zoomar kartan in på dess läge. Smartmobiler har i regel en inbyggd GPS-mottagare som kan användas i detta fall, men vissa läsplattor samt alla datorer saknar dock detta. Om systemet körs på en enhet som saknar GPS-mottagare (t.ex. en dator) bestäms deras position till den plats där dess internetuppkoppling befinner sig (t.ex. användarens router), något som i regel orsakar många meters fel. Vid test från kommunens datorer uppmättes ett fel på cirka 650m. Det är därför viktigt att använda en smartmobil eller läsplatta med GPS-mottagare.

Uppfyller kraven:

1.1	1.2	1.3	1.4	2.1	2.2	3.1
					✓	

5.5. CarryMap Observer



Figur 5.6: CarryMap Observer for iPhone (CarryMap, 2013)

Denna app finns tillgänglig för iOS och Android och är tänkt att användas med ArcGIS. För att få tillgång till sin karta behöver man först exportera den, detta görs m.h.a. ett tilläggsprogram för ArcGIS (som är utvecklat för CarryMap). Detta genererar en cmf-fil som man sedan kan föra över till sin smartmobil (CarryMap, 2013). För- **och** nackdelen med detta är att alla användare har en lokal kopia av kartan. Fördelen är att systemet även fungerar utan en internetuppkoppling. Nackdelen är att synkroniseringar av ändringar och uppdateringar blir svårare, informationen måste manuellt exporteras från smartmobilen till datorn.

Uppfyller kraven:

1.1	1.2	1.3	1.4	2.1	2.2	3.1
✓ *	✓	✓		✓ **	✓	

* endast kopia.

** appen har möjlighet, men det beror på servern.

5.6. GeoMobile for ArcGIS



Figur 5.7: *GeoMobile for ArcGIS for iPhone (GeoMobile for ArcGIS, 2013)*

Denna app finns tillgänglig för iOS och Android. För att nå din geografiska information krävs det att du skapar en fil som beskriver din tjänst. Detta innebär t.ex. var din geografiska information befinner sig, hur den ska visas och hur den kan användas. Detta leder till att man får ett väldigt flexibelt system. (*GeoMobile for ArcGIS, 2013*)

Det finns även en annan app från samma tillverkare med namnet *GeoMobile for ArcGIS Online*, den är i nuläget dock bara kompatibel med iPad.

Uppfyller kraven:

1.1	1.2	1.3	1.4	2.1	2.2	3.1
✓	✓	✓		✓*		

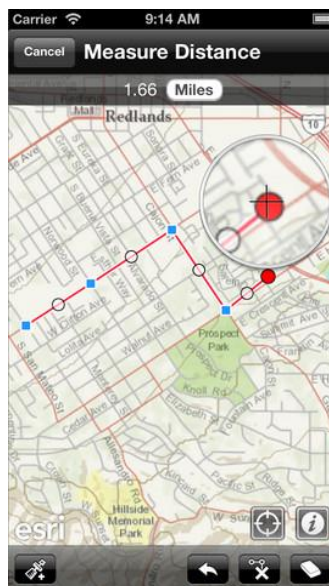
* ej foto.

5.7. ArcGIS appar

ESRI har i nuläget två färdiga appar tillgängliga för iOS (operativsystemet för iPhone/iPad); *ArcGIS* och *ArcGIS Collector*. Båda är gratis att ladda ner och att använda. Fördelen med detta är att man slipper utveckla ett system för klienten, men nackdelen är att man blir väldigt låst då det är ett färdigt paket som man installerar. Trots att klienten är "låst" finns det dock fortfarande möjlighet att på serversidan påverka hur informationen ska spridas och hur den ska tolkas av klienten. Man behöver alltså inte vara så låst som det först kan verka.

Det finns båda för- och nackdelar med att använda en färdig lösning. En fördel i detta fall är att ESRI är ett mycket större företag än GIS-avdelningen på Kalmar kommun. Detta leder till att de kan lägga mer resurser på utveckling, de kan även se till att produkten blir väl testad. En nackdel med att välja en färdig lösning är som nämnts tidigare att man är låst till deras val, man får dålig flexibilitet.

5.7.1. ArcGIS (app)



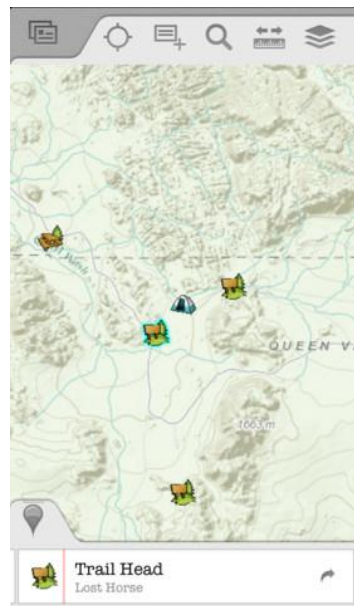
Figur 5.8: *ArcGIS for iPhone (ArcGIS, 2013d)*

Denna app finns tillgänglig för iOS, Android, Windows Mobile och Windows Phone. Den går under det aningen tvetydiga namnet *ArcGIS*, för att undvika missförstånd har författaren valt att kalla den *ArcGIS (app)*. Appens syfte är att man enkelt ska hitta och sprida kartor, men den har även enklare verktyg för editering. Man kan t.ex. skapa, ändra och ta bort punkter i kartan. (*ArcGIS, 2013d*)

Uppfyller kraven:

1.1	1.2	1.3	1.4	2.1	2.2	3.1
✓	✓	✓ *	✓	✓	✓ *	✓

5.7.2. ArcGIS Collector



Figur 5.9: ArcGIS Collector (ArcGIS, 2013e)

ArcGIS Collector gör det möjligt för användaren att nå sina kartor som de skapat på ArcGIS Online. Syftet med appen är att underlätta fältarbete, t.ex. kan man enkelt skapa/editera punkter, men den innehåller även mätverktyg och möjligheten att få en vägbeskrivning (ArcGIS, 2013e). ArcGIS Collector finns tillgänglig för iOS och Android.

Uppfyller kraven:

1.1	1.2	1.3	1.4	2.1	2.2	3.1
✓	✓	✓ *	✓	✓	✓ *	✓

* appen har möjlighet, men det beror på servern.

5.8. Sammanfattning

Krav:

1.1 - Man skall nå baskartan

1.2 - Systemet skall utvecklas för smartmobiler

1.3 - Kartan skall se identisk ut i mobilen som i datorn

1.4 - Kartan skall skyddas av ett säkerhetssystem

2.1 - Man skall ha möjligheten att lägga till punkter i kartan, med beskrivning och ev. foto

2.2 - Systemet ska via smartmobilers GPS hitta sin position på kartan

2.3 - Det skall vara så användarvänligt som möjligt

3.1 - Man skall kunna klicka på stomnätspunkter i kartan och få upp stomnätsskisserna

Krav **1.1** har tolkats som "möjligheten att använda sin egen geografiska information i kartan.

Krav **2.3** har valts att bortse ifrån, då det är en tolkningsfråga.

Krav **3.1** har tolkats som "möjligheten att klicka på punkter i kartan och få upp information om punkten".

Tabell 5.1: Liknande system

	1.1	1.2	1.3	1.4	2.1	2.2	3.1
Felanmälan	✓	✓	✓		✓		
iSurvey	✓		✓		✓	✓	✓
Parkering		✓	✓			✓	✓
För alla skärmar			✓				
Geolocation						✓	
CarryMap Observer	✓	✓	✓		✓	✓	
GeoMobile for ArcGIS	✓	✓	✓		✓		
ArcGIS (app)	✓	✓	✓	✓	✓	✓	✓
ArcGIS Collector	✓	✓	✓	✓	✓	✓	✓

Vid en snabb överblick ser de två sista alternativen ut att vara perfekta för detta projekt, men det är inte hela sanningen. Att de har möjlighet att uppfylla alla krav medför inte att de gör det på ett bra sätt. De första fyra lösningarna visar även på potentialen med webblösningar, då de är enklare att modifiera och bygga själva. Detta betyder att om man kan hitta flera exempel som var och en uppfyller minst ett krav, kan man sedan använda de för att bygga ett komplett system som uppfyller alla. "Mobile template" och "Geolocation" är exempel på sådana mindre system.

6. Teknisk bakgrund

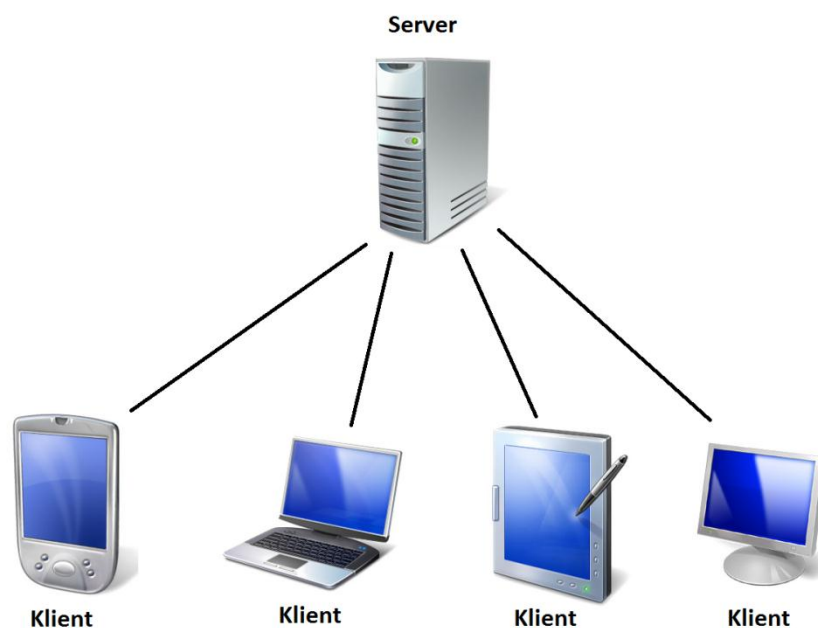
Syftet med detta kapitel är att ge läsaren en inblick i tekniken bakom projektet, dvs. tekniska förutsättningar och förslag på lösningar. Tekniken kan delas in i två olika undergrupper; hårdvara och mjukvara.

Med hårdvara menar man de fysiska delarna av systemet, saker som man kan ta på. I detta kapitel diskuteras smartmobiler och läsplattor, andra exempel på hårdvara kan vara t.ex. en datorskärm eller en processor.

Mjukvara syftar ofta på datorprogram. Men det behöver inte vara ett fullständigt program, det kan även vara en mindre lösning. Det behöver inte heller vara specifikt menat till datorer. Skillnaden mellan hårdvara och mjukvara handlar i grund och botten om den fysiska aspekten. Till skillnad från hårdvaran är inte mjukvaran något som man kan ta på, utan den är något som finns lagrat på en hårddisk.

När man diskuterar mjukvara i nätverkssammanhang är det väldigt viktigt att man delar upp problemet i två delar, server och klient.

- Serversidan är där all information finns lagrad, problemet är hur man ska sprida denna information. Informationen ska endast spridas till dem som har rätt att ta del av den. Det handlar dock inte bara om att skicka iväg information, man ska också ha möjligheten att ta emot information från klienten.
- Klientsidan är där som användaren befinner sig, dvs. han eller hon som använder systemet för att hämta information. I detta fall är det en lösning för smartmobiler/läsplattor som ska undersökas.



Figur 6.1: Server och klient

6.1. Mobila klienter

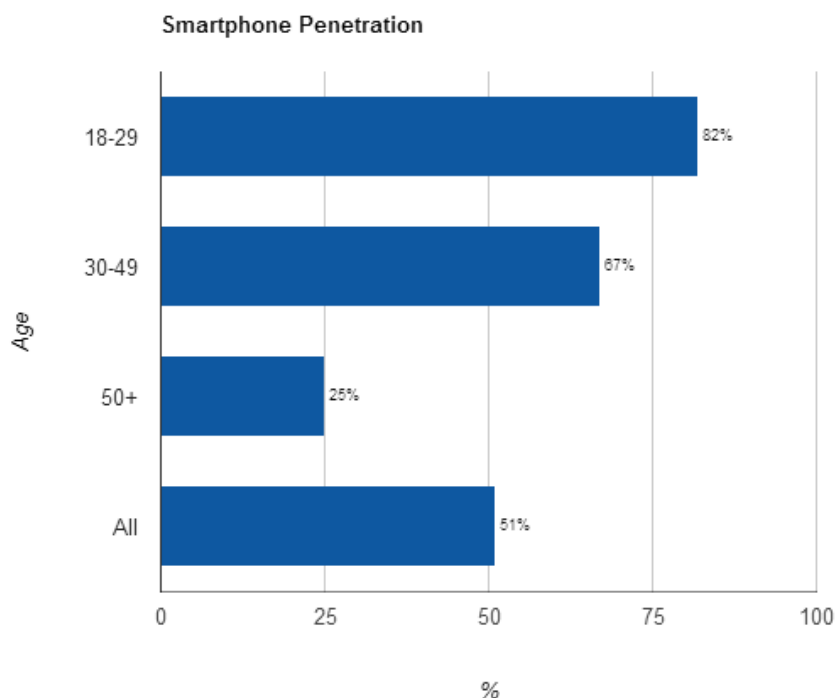
6.1.1. Smartmobiler

En smartmobil (engelskans *smartphone*) är som namnet antyder en "smartare" mobiltelefon, den har högre prestanda och fler funktioner än en traditionell mobiltelefon. Med högre prestanda menas t.ex. att processorn är kraftfullare och att skärmen är större, något som i sin tur öppnar upp möjligheten för mer avancerade funktioner. Skärmen är en s.k. pekskärm, den kan registrera när och var en användare rör vid skärmen. I stort sett alla smartmobiler har även en inbyggd GPS-mottagare, vilket möjliggör att man kan följa sin position på en karta.

Listan på möjligheter med smartmobiler är lång, författaren anser dock att den viktigaste funktionen är att man ständigt kan vara uppkopplad mot internet. Detta i sig var knappast något som smartmobilerna var först med, men det var de som gjorde att det fick dess stora genomslag. Enligt författaren revolutionerade smartmobilerna internet i mobilen av två anledningar:

- Det första är att en smartmobil har förmågan att tolka och visa en hemsida lika väl som en dator, fast på en mindre skärm. De tidigare mobiltelefonerna som hade tillgång till internet kunde ofta bara visa hemsidor korrekt om de var specialbyggda för mobiltelefoner.
- Det andra är introduktionen av *appar*, dvs. ett program som man laddar ner och installerar på sin smartmobil. Detta är analogt med ett datorprogram för en dator. Det viktigaste var att alla fick möjligheten att utveckla sina egna appar. Det var inte begränsat till mobiltelefonverkarna.

Antalet användare av smartmobiler i Sverige är stort och det fortsätter att öka. Enligt siffror från 2012 är Sverige på tredje plats i världen i antalet smartmobiler per capita (*Communities Dominate Brands, 2013*). Ungefär varannan svensk äger en smartmobil, men i åldern 18-29 är siffran fyra av fem (*Think with Google, 2013*). Detta medför att smartmobiler är en plattform som folk är vana vid, vilket i sin tur leder till att företagen inte behöver investera lika mycket tid och pengar på att utbilda sin personal för att använda det nya systemet.



Figur 6.2: Antalet användare av smartmobiler i Sverige, Think with Google (2013)

6.1.2. Läsplattor

En läsplatta är i grund och botten som en stor smartmobil fast utan de klassiska telefontjänsterna (som att ringa och skicka textmeddelanden). De använder i regel samma operativsystem som andra smartmobiler och kan därmed köra samma appar (program). Den mest kända läsplattan är idag iPad, den är utvecklad av Apple och använder samma operativsystem som iPhone. Med andra ord fungerar program som är ämnade för iPhone även för iPad (om de inte berör de telefontjänsterna som saknas).

Fördelen som läsplattan har över smartmobilen är dess storlek, den stora skärmen gör det mycket lättare att läsa texter och se små detaljer.

6.1.3. Andra alternativ

Det finns andra alternativ än smartmobiler och läsplattor, t.ex. tåliga handdatorer (engelskans *rugged computers*) som är anpassade för att användas i fält. Det har dock valts att inte undersöka detta närmre i projektet.

6.2. Operativsystem

Operativsystemet är den mjukvara som driver ett system, det agerar mellanhand mellan de installerade programmen och hårdvaran. Exempel på operativsystem för datorer är Microsoft Windows, Linux och OS X (Mac OS X). Denna rapport är dock fokuserad på operativsystem för smartmobiler.

Anledningen till att diskutera operativsystem är att de påverkar hur ett program fungerar (eller om det fungerar alls). Ett operativsystem är egentligen ett program, vilket medför att det i grund och botten är uppbyggt av kod. Det är viktigt att veta vilket språk det är som gäller för det valda operativsystemet, för man måste anpassa sig därefter. Två människor som pratar olika språk kommer (troligen) inte att förstå varandra, vilket är detsamma för datorer. Ett program som är skrivet för OS X och med dess regler kommer därmed inte att fungera i Microsoft Windows, och vice versa.

6.2.1. iOS

Det är utvecklat/utvecklas av Apple, ursprungligen för *iPhone*. När det släpptes användes det dock av *iPod* och i dag återfinns det även i *iPad* och *Apple TV* (Apple, 2013a). Apple använder alltså *iOS* till alla sina mobila enheter. *iOS* återfinns endast i Apples egna produkter, då de inte lämnat ifrån sig rätten att använda operativsystemet.

6.2.2. Android

Ursprungligen utvecklat av ett mindre företag, men de blev senare uppköpta av Google som i dag driver utvecklingen vidare. *Android* är i dag en stor gemensam plattform för många olika mobiltelefonstillverkare som exempelvis Sony, HTC, Samsung, LG och Huawei.

6.2.3. Övriga

Andra operativsystem för smartmobiler är t.ex. Symbian, Blackberry OS, Windows Mobile och Bada. Fler är även under utveckling. Eftersom ungefär nio av tio svenskar använder antingen *iOS* eller *Android* (Think with Google, 2013) kommer dessa övriga inte att undersökas närmare.

6.3. Appar

Ordet *app* kommer från engelskans *mobile application*, vilket är ett "program för mobila enheter". En app är för smartmobilen vad ett datorprogram är för datorn. Den klassiska bilden av en app är ett program som man laddar ner och installerar på sin enhet. På senare tid har det uppkommit en hybrid mellan en klassisk app och en hemsida, en s.k. *webbapp*.

6.3.1. "Klassiska" appar

I takt med att antalet smartmobilanvändare ökar har även antalet aktörer på marknaden ökat. Företagen konkurrerar bl.a. om vilket operativsystem som är "bäst". I nuläget är *iOS* och *Android* de två största (Strategy analytics, 2013). Det är dock på apparnas marknad som den hårdaste konkurrensen utspelar sig. Som tidigare nämnts är möjligheterna med appar i stort sett oändliga, vilket medför att det finns i stort sett oändligt många appar. Appar säljs på internet via digitala affärer (även kallade marknader) som ofta är förinstallerade på

smartmobilerna. Användaren är därför oftast bara ett klick därifrån. Fördelen med digitala affärer är att de är mycket lättare att slå sig in på än den konventionella (fysiska) marknaden, då nya aktörer bara behöver ett minimalt startkapital. All denna konkurrens har gjort att kvalitén har trissats upp och priset ner. Appar till iOS går att ladda ner via *App store*, Androids motsvarighet är *Google Play*.

6.3.2. Webbappar

En webbapp är en hemsida som ter sig som om den vore en riktig app. Ett otränat öga skulle därför kanske inte kunna skilja dem åt vid ett blindtest. De tekniska framsteg som skett på webben de senaste åren har gett oss möjligheten att anpassa hemsidor för just smartmobiler, men det har även gett oss möjligheten att nå smartmobilens hårdvara som t.ex. dess GPS-mottagare.

Fördelar med webbappar är t.ex. att:

- Man slipper installera ett program. Det är en fördel rent tids- och bekvämlighetsmässigt, men även säkerhetsmässigt.
- Det är enkelt att uppdatera systemet (hemsidan). Man behöver inte skicka ut en ny version av appen till alla användare.
- En hemsida fungerar på alla operativsystem.

Nackdelar med tekniken är t.ex. att

- Det blir aningen långsammare då allt måste hämtas från webben, inget är sparat på mobilen.
- Man får inte samma rättigheter som ett installerat program, webbappar i iOS tillåts t.ex. inte att nå kameran direkt (men det går att kringgå genom att installera en app som agerar som mellanhand, ett exempel på detta är appen *Picup* (Picup, 2013)).

6.4. GIS-program

Ett GIS-program kan användas för att lagra, bearbeta, analysera och visualisera geografisk information. Ren geografisk information är oftast enklast att förstå i form av en karta. Det är först därefter som man kan börja tolka och arbeta med informationen. Det är dock inte bara ett redskap för editering utan det innehåller även flera kraftfulla verktyg och funktioner. Det finns t.ex. flertalet rumsliga analyser och beräkningar.

6.4.1. ArcGIS

ArcGIS är ett samlingsnamn för en grupp av GIS-program, utvecklade av ESRI. ESRI är idag en av världens största utvecklare och återförsäljare av GIS-program (*All Points Blog*, 2011). Det finns i dag många olika ArcGIS-program och de är utvecklade för många olika plattformar, även om det främst är för datorer (som kör Microsoft Windows). De flesta som är bekanta

med ArcGIS tänker nog främst på ArcGIS Desktop (ArcMap och ArcCatalog) när de hör namnet ArcGIS. I detta projekt har det dock valts att fokusera främst på fyra ArcGIS-produkter: ArcGIS Online, ArcGIS Server, ArcGIS (app) och ArcGIS Collector. *Online* och *Server* hanterar server-sidan medan "*app*" och *Collector* agerar på klient-sidan.

ArcGIS Online

ArcGIS Online (även kallat *ArcGIS.com* eller *AGO*) är en "*moln*"-tjänst (engelskans *cloud*), vilket betyder att allt material som man arbetar med lagras i molnet (på internet). Detta är till skillnad från det traditionella arbetssättet där man arbetar med allt material lokalt (på sin egen dator/server). Fördelen med molntjänsten är att alla lätt kan nå dina kartor, förutsatt att de har en internetuppkoppling och tillräckliga rättigheter. Nackdelen är att man lämnar ifrån sig sin information, samt att man skapar en kopia som man arbetar mot i stället för den riktiga databasen. I molntjänsten finns det även möjlighet att nå andra användares kartor, samt rentutav återanvända dem (om de valt att dela dem med allmänheten). Det finns även flertalet färdiga bakgrundskartor att välja mellan. För att sammanfatta är fokus med AGO att enkelt skapa och sedan sprida sina kartor, men även att upptäcka vad andra har publicerat.

En annan fördel med AGO är att det är väl integrerat med ArcGIS (app) och ArcGIS Collector. Detta medför att det är väldigt enkelt att sprida sina kartor till smartmobiler och läsplattor via AGO.

AGO tar betalt via *krediter*, vissa av dina handlingar resulterar i att du måste betala krediter. När dina krediter har tagit slut måste du ladda ditt konto igen. Detta går att likna vid ett kontantkort för mobiltelefoner, då vissa handlingar resulterar i att pengar dras från ditt kort. Det är svårt att uppskatta antalet krediter som kommer att konsumeras även med bra bakgrundsfakta. Kalmar kommun har dock i dag en kommunlicens som ger dem tillgång till relativt många krediter (samt många användarkonton).

Man kan på deras hemsida få information om ungefär hur många krediter ett typiskt system använder. Det finns även ett verktyg (*ArcGIS, 2013f*) där användaren matar in information om det nya systemet för att få en bättre uppskattning, exempel på information som matas in är t.ex. antalet användare och lagringsutrymme på servern. En grov uppskattning med detta verktyg visar att antalet krediter som systemet använder är långt under vad kommunlicensen ger tillgång till, då antalet användare av slutprodukten är relativt få (och trafikmängden blir därmed låg).

ArcGIS Server

ArcGIS Server är en lösning för att driva en egen server med sin egen information. Till skillnad från AGO lämnar aldrig informationen kommunens intranät. Detta är en fördel ur säkerhetssynpunkt, då man ansvarar för sin egen information. Om man laddar upp sin data till en molntjänst måste man lita på att den tjänsten är säker. En annan fördel att man har

möjligheten att arbeta skarpt mot databasen, dvs. man arbetar hela tiden mot originalet. Om man använder en molntjänst måste man kopiera sin information till dess server, och man arbetar därför alltid mot en kopia. Då du driver din egen server (till skillnad från AGO) får du större flexibilitet och fler valmöjligheter, men även kraftfullare lösningar.

ArcGIS Server är kompatibelt med ArcGIS (app), något som kan underlätta att sprida sin information mobilt.

ArcGIS (app)

Som har diskuterats i kapitel 5 är syftet med denna app att agera portal för AGO i användarens smartmobil. Genom att man loggar in med sitt AGO-konto kommer man enkelt åt sina kartor. Det går även att hitta kartor från andra användare, precis som i AGO. Appen innehåller även enkla mätverktyg och ett enkelt sökverktyg. Den har även stöd för enkel datainsamling och editering. (*ArcGIS, 2013d*)

Appen är främst ämnad för att användas tillsammans med AGO, men den har även stöd för att ansluta sig direkt mot ArcGIS Server. Fördelen med detta är att man gör sig helt oberoende av AGO och slipper då bekymra sig om kreditkostnader.

ArcGIS Collector

Denna app har vid en första anblick väldigt mycket gemensamt med ArcGIS (app). De stora skillnaderna är att ArcGIS Collector inte är kompatibelt med ArcGIS Server samt dess målgrupper. Medan ArcGIS (app) är aningen mer riktad mot gemene man är ArcGIS Collector riktad mot mätingenjörer (eller liknande) som arbetar i fält. Detta är något som återspeglas i användargränssnittet, vilka knappar och vilken information som finns tillgängliga (och var). Det är tänkt att vara ett arbetsredskap i fält. Eftersom den har samma målgrupp som detta projekt kan man lätt tro att det skulle vara en perfekt lösning, men det faktum att man är låst till både AGO och en färdig app ger inte mycket flexibilitet.

Det kan tyckas konstigt att en app som är riktade mot professionella användare (mätingenjörer) är låst till AGO, då många professionella användare säkerligen önskar använda information från sin egen server. Detta problem går dock att lösa genom att man infogar en karttjänst från ArcGIS Server i AGO, se mer om detta i kapitel 9.

6.4.2. Utveckla en egen app med ArcGIS

App till smartmobil

På ArcGIS hemsida finns det hjälpfiler, exempel och guider för att underlätta utvecklingen av egna appar. För smartmobiler finns det ett s.k. *SDK* (software development kit) som man kan ladda ner från ESRI:s hemsida, helt gratis. Man kan då tillverka sina egen appar som kan

kommunicera med AGO eller ArcGIS Server. I nuläget finns det SDK för ArcGIS att ladda ner till iOS, Android och Windows Phone (ESRI, 2013).

Bortsett från ESRI:s SDK för ArcGIS behöver man även ett speciellt SDK för det valda operativsystemet, dvs. iOS eller Android SDK. Man behöver även en utvecklingsmiljö, ett program som kan hjälpa dig skriva, kontrollera och testa din kod.

iOS

För att man ska ha möjligheten att utveckla appar till iOS behöver man först en Mac-dator, då kan man (gratis) ladda ner programvara för utveckling. Här krävs det dock att man betalar en årsavgift för att göra sina appar tillgängliga på *App store*, vilket är enda sättet att förflytta sina appar från datorn till mobilen. Apparna skrivs i språket *Objective-C*. (Apple, 2013b)

Android

Programvaran som krävs för att utveckla appar till Android är gratis. Till skillnad från den för iOS fungerar den dock på de flesta operativsystemen. Vill man publicera sina appar på *Google Play* måste man betala en engångsavgift. Det är dock inte ett måste; till skillnad från iOS kan man nämligen med Android föra över appen direkt från datorn till mobilen. Appar till Android skrivs oftast i en specialanpassad version av Java. (Google, 2013)

Webbapp

För webbappar finns det inget SDK, men ESRI har släppt något liknande som kallas *API* (*Application programming interface*). Ett API är ett protokoll som förklarar reglerna för hur systemet kan kommunicera med ett annat system, detta är ett måste om man inte har tillgång till källkoden. Det förklarar *vad* man kan göra och *hur* man *måste* göra det. För att använda ett kommando behöver man veta exakt vad kommandot heter och vad det förväntar sig för information (indata).

Eftersom en webbapp är mer generell än t.ex. en iPhone-app finns det flera sätt att skapa dem på, samt flera programmeringsspråk att välja på. Ett av de mer populära språken för detta ändamål är *JavaScript*. Ren JavaScript räcker inte för att skapa en webbapp, men det är den bas som man sedan kan bygga vidare på (för att kalla på andra skript/funktioner). Medan JavaScript är det skriptspråk som man baserar sin app på är *HTML* det språk som är grunden för alla hemsidor. Ren HTML-kod kan bara utföra enklare uppgifter och lösningarna blir då oftast inte de mest optimala. Det är därför som man tar hjälp av andra skriptspråk som exempelvis JavaScript. (acunetix, 2013)

6.4.3. Öppen källkod

Här följer information om några vanliga öppen källkodslösningarna inom webbgis. Mer information om öppen källkod och dess licenser följer i kapitel 7.

OpenLayers

OpenLayers används för att skapa dynamiska kartor till webbsidor (klientsidan). Den hämtar sin information via protokollen *Web Map Service* (WMS) eller *Web Feature Service* (WFS), dvs. den standarden som är satt av OGC (OGC, 2013). OpenLayers fungerar därför med flera serverprogram, t.o.m. ArcGIS Server. OpenLayers är skrivet i JavaScript, och är licenserat under *BSD*. (OpenLayers, 2013)

jQuery

jQuery är ett bibliotek av olika skript för JavaScript, det användas för att förenkla utvecklingen av en webbsida på klientsidan. jQuery utvecklas av *jQuery foundation*. De driver även fyra fristående utvecklingsprojekt; *jQuery user interface*, *jQuery mobile*, *jQuery js unit testing* och *jQuery Sizzle*. jQuery och dess underprojekt är släppta under licensen *MIT*. (jQuery, 2013)

För detta projekt är det främst *jQuery mobile* som är intressant, som namnet antyder är det en specialanpassad version av jQuery för smartmobiler. Det är utvecklat för att stödja alla olika typer av smartmobiler och operativsystem, det är också därför optimerat för att användas med pekskärm. Eftersom en smartmobils resurser är begränsade är även *jQuery mobile* utvecklat för att dra så lite processorkraft som möjligt. Det går även att använda för att skapa ett grafiskt användargränssnitt som är anpassat för smartmobiler. (*jQuery mobile*, 2013)

Dojo

Dojo är en extra verktygslåda till JavaScript och tillhandahåller ett antal färdiga verktyg. Det är baserat på JavaScript och är utvecklat för att användas tillsammans med det. Dojo använder öppen källkod och är licenserad under *BSD* och *Academic Free License*. (Dojo, 2013)

Det är möjligt att skapa "lyssnare" som lyssnar på användarens handlingar, och sedan reagerar på ett förbestämt sätt. Man kan t.ex. skapa en lyssnare för ett speciellt lager där lyssnaren reagerar när användaren klickar på lager. Då detta sker kallar man på ett javaskript som hämtar och visar information om den punkt som användaren klickat på. Utöver ovanstående exempel kan lyssnare även lyssna på andra handlingar än klickningar, t.ex. då användaren har skapat en punkt.

Proj4J

Proj4J är ett bibliotek av olika Javaskript, det används vid transformationer av koordinater från ett koordinatsystem till ett annat. Kärnan i Proj4J är baserat på *Proj4*, men det använder Java som språk istället för C. Det är släppt under licensen *Apache license*. (*Proj4J, 2013*)

GeoServer

GeoServer är ett program som används på serversidan för att sprida geografisk information, användaren kan både visualisera och editera informationen. Det sprider sin information via t.ex. WMS och WFS (som har nämnts tidigare under *OpenLayers*). Informationen kan spridas i många olika format (t.ex. jpeg, pdf och kml) och till olika typer av klienter (t.ex. Google Maps och ArcGIS). *OpenLayers* är integrerat i GeoServer för att man enklare ska kunna visualisera sin information. GeoServer kan sprida data från t.ex. shapefiler, PostGIS och ArcSDE. Det är licenserat under *GNU GPL*. (*GeoServer, 2013*)

MapServer

MapServer har mycket gemensamt med GeoServer då de utför ungefär samma uppgift. Det använder sig också av de uppställda OGC-standarderna för att sprida sin information (t.ex. WMS och WFS). Precis som GeoServer kan den sprida information från t.ex. shapefiler, PostGIS och ArcSDE. Den är licenserad under *MIT license*. (*MapServer, 2013*)

PostgreSQL och PostGIS

PostgreSQL är ett objektrelationellt databassystem (ORDBMS). Det är licenserat under *PostgreSQL license*, en licens som liknar *BSD* och *MIT*. (*PostgreSQL, 2013*)

PostGIS är ett tillägg till PostgreSQL som möjliggör skapandet av en spatial databas i PostgreSQL, vilket medför att man kan ställa rumsliga frågor till databasen. PostGIS är licenserat under *GNU GPL*. (*PostGIS, 2013*)

6.5. Lagring av geografiska data

När man ska lagra stora mängder data är det fördelaktigt att göra detta i en databas. Informationen blir då lagrad på en plats och på ett enhetligt sätt (då databasen har regler för hur dess data ska vara utformad).

Två populära former av databaser är relationsdatabaser och objektorienterade databaser, men det finns även en hybrid av de två som kallas *objektrelationella databaser*. En klassisk relationsdatabas är uppbyggd av rader och kolumner där man kan lagra värden (siffror eller text). Skillnaden mellan detta och en objektrelationell databas är att du även kan mata in

objekt som värden, dvs. man länkar till ett objekt. Detta objekt kan innehålla ett eller flera värden, vilka i sin tur kan vara andra objekt. (Brinkhoff och Kresse, 2012. s.80)

Problemet med vanliga filer är att de bara kan användas av en användare åt gången, men detta problem går att kringgå genom att man lagrar sin information i en databas med ett *database management system (DBMS)*. Flera användare kan läsa informationen även om den inte är lagrad i en databas med ett DBMS, men bara en kan editera den åt gången. Det är m.a.o. ett måste att sprida sin information m.h.a. ett DBMS om man ska ha editeringsmöjligheter. Om man använder ett DBMS kan man även ställa frågor till databasen och få svar. Om vi t.ex. har en databas med alla anställda inom kommunen kan vi be databasen hämta all information om alla med namnet "Peter". (Brinkhoff och Kresse, 2012. s.63)

6.5.1. Spatial databas

En spatial databas är en form av databas som är specialdesignad för att lagra geografisk information. Den lagrar därför informationen på ett mer lämpat sätt och den gör att man kan ställa rumsliga frågor. Traditionella relationsdatabaser saknar möjligheten att lagra geometri. För att beskriva geometri behöver man använda sig av ett geometriobjekt, det är annars omöjligt att beskriva alla olika typer av geometrier (t.ex. punkter, linjer och polygoner). Det finns flera objektrelationella databaser som har adderat en datatyp för geometri, dessa databaser kallas för spatiala databaser. (OpenGeo, 2013)

6.5.2. Molnbaserade tjänster

Molnlagring är en *molnbaserad tjänst*, det betyder att man använder ett datormoln (en grupp av datorer) för att utföra sitt arbete. Molntjänster kan tillhandahållas av samma företag eller kommun som använder dem, men det vanligaste är att man hyr in molntjänster utifrån (från företag som är specialiserade på det). Användarnas arbetsuppgifter sprids automatiskt ut i datormolnet, arbetsbelastningen på systemet blir därmed jämnt fördelad (NIST, 2011). När det gäller molnlagring innebär detta att användarens data kan hamna på flera olika datorer inom molnet, detta är dock inget som användaren märker av. Ett exempel på en populär molnbaserad tjänst för lagring av information är Dropbox (Dropbox, 2013). Molnbaserade tjänster är dock mer än molnlagring, det innebär även att man tillhandahåller verktyg och hjälpmedel (NIST, 2011). Oftast är dessa verktyg till för att behandla den data som finns lagrad i molnet.

AGO använder sig av vad de kallar för *Cloud GIS*, som är en molnbaserad tjänst (ArcGIS, 2013g). Det innebär att man förutom att lagra information även har tillgång till GIS-verktyg för att arbeta med sin data. Tanken är att man ska ha möjligheten att arbeta med sin data var man än befinner sig, då all data och alla verktyg finns tillgängliga i molnet. Användaren behöver inte ens installera någon mjukvara.

6.6. Säkerhet

För att skapa ett säkert system är det mycket som man ska ha i åtanke, det första problemet är det säkerhetssystem som användaren kommer att använda för att logga in (för att ta del av systemet och dess tjänster). Det är viktigt att utomstående inte kan ta sig in olovligen, då kan de t.ex. modifiera den existerande informationen eller kopiera den till sin egen dator.

Innebörden av ett säkert system är dock inte begränsad till hur man kommer åt informationen, det handlar även om att den ska förvaltas på rätt sätt. Informationen ska lagras i en säker databas som genomgår regelbundna säkerhetskopieringar. Det är också viktigt att de som uppdaterar informationen gör detta på ett korrekt sätt, vissa handlingar kan vara omöjliga att ångra.

6.6.1. Intranät kontra internet

Intranätet är Kalmar kommuns interna nätverk, det är en serie av hoppkopplade nätverk som är avskärmade från omvärlden. De kommunanställda har varsitt användarnamn och lösenord som de använder för att nå nätverket. Varje användarkonto kan tilldelas olika rättigheter, och får därmed tillgång till olika mycket data och programvaror. I intranätet finns stora mängder information tillgänglig, som då enkelt kan nås av de kommunanställda som har tillgång till den. Intranätet går inte att nå via internet (om det inte finns en tjänst skapad för det), men datorer i intranätet har fortfarande möjlighet att nå internet och omvärlden.

Ett intranät är knappast något som är unikt för Kalmar kommun, utan det är något som finns för de flesta kommuner och företag i Sverige.

Brandvägg

En brandvägg är ett program som används för att skydda ett nätverk eller en enskild dator, i detta projekt är det dock brandväggar för hela nätverk som diskuteras. En brandvägg för ett nätverk placeras i nätverkets "ingång", i den eller de routers som sprider internet. Informationen kommer in i nätverket i form av paket, dess innehåll kontrolleras av brandväggen (*SearchSecurity, 2007*). Det kontrollerar t.ex. vem som är avsändaren och mottagare, och om de är säkra källor.

DMZ

DMZ (*demilitarized zone*) är ett isolerat nätverk som agerar som mellanhand mellan ett nätverk och omvärlden (internet). När man publicerar tjänster över internet medför detta en säkerhetsrisk. För att en server ska ha möjlighet att skicka och ta emot information från internet måste den göra sig tillgänglig för kommunikation, denna dator kan då användas som en baddörr in i nätverket. Det är därför vanligt att man placerar alla webbtjänster på servrar

inom DMZ. Detta medför att även om säkerheten i en serverdator blir kompromissad kommer inte inkräktaren åt intranätet. (TechRepublic, 2005)

Som figur 6.3 visar skyddas DMZ av en brandvägg från omvärlden, men även intranätet skyddas i sin tur av en brandvägg från DMZ.



Figur 6.3: DMZ

6.6.2. Inloggningssystem

Det finns många olika sätt att logga in i ett system, det mest klassiska är via ett användarnamn och lösenord.

Användarnamn och lösenord

Varje användare måste ha ett eget användarkonto med ett eget lösenord. Fördelen med att alla användare har varsitt konto är att man har möjligheten att kontrollera användarna i systemet, t.ex. om en anställd slutar kan dennes konto tas bort.

Ett enkelt alternativ att implementera är att alla användare använder samma användarnamn och lösenord. Säkerheten är dock låg då risken för att användarnamn och lösenord läcker ut är stor. Det är även arbetsamt att byta lösenord då alla användare måste informeras.

Filer som nycklar

Användaren måste i förväg ladda ner en fil till sin enhet som fungerar som nyckel till systemet. Fördelen med denna lösning är att användaren slipper logga in varje gång den ska använda systemet. Nackdelen är att användaren måste få tillgång till säkerhetsnyckeln och installera den, något som alla användare kanske inte är vana vid. Man kan även använda ett system där varje användare genererar en egen nyckel, varje nyckel förknippas sedan med användaren på serversidan. Detta ger en bättre kontroll över de enskilda användarna.

(Weiske, 2013)

ArcGIS's säkerhetssystem

Både AGO och ArcGIS Server har inbyggda inloggningssystem, en administrator för tjänsten kan i båda fallen bjuda in användare. I AGO får användaren möjlighet att skapa sitt eget

konto efter det att de blivit inbjudna i organisationen, i ArcGIS Server skapar administratören kontot.

För båda lösningarna kan administratören skapa olika "roller", som de sedan kan tilldela användarna. Detta bestämmer vad användaren har rätt att göra. Rollerna kan vara av tre olika typer *User*, *Publisher* och *Administrator* (i stigande rangordning).

6.6.3. Säker förvaltning

Det är viktigt att informationen förvaras i en säker databas som man genomför regelbundna säkerhetskopieringar på, men det är också viktigt att personalen är rätt utbildad. Det är viktigt att den som genomför en ändring eller uppdatering i databasen gör detta på ett korrekt sätt. Detta innebär dels att de arbetar mot databasen på ett korrekt sätt, men även att de ändringar som de genomför faktiskt är korrekta till att börja med. Kalmar kommun har ett antal editörer som alla har varsitt område som de ansvarar för, för att det ska bli rätt. Någon kanske t.ex. ansvarar för alla vägar. Det är endast de utnämnda editörerna som har rätt att editera i databasen, och de har endast rätt att editera precis det som de är ansvariga för (t.ex. endast vägar) (*Bernhard, 2013*).

Säker databas

ACID-modellen brukar användas för att beskriva en säker databas. *ACID* står för *Atomicity*, *Consistency*, *Isolation* och *Durability*.

Atomicity – När en användare försöker göra en ändring i databasen ska antingen hela ändringen träda i kraft, eller inte alls. Det vore fatalt om användarens dator kraschar mitt under uppdateringen, detta skulle då leda till en halv uppdatering. Det skulle i sin tur leda till en korrupt databas.

Consistency – Bara data som följer databasens uppsatta regler ska skrivas in i databasen. Om en del av uppdateringen bryter mot någon regel kommer hela uppdateringen att misslyckas.

Isolation – Varje ändring i databasen sker isolerat från andra. Om två användare försöker ändra i databasen samtidigt ska de inte påverka varandra.

Durability – Betyder att en ändring som en gång har tagits in i databasen ska förbli sparad. Detta krav följs m.h.a. regelbundna säkerhetskopieringar och loggning av användarnas aktivitet.

(*Chapple, 2013*)

6.6.4. Mobil säkerhet

Marknaden för smartmobiler och dess program har vuxit explosionsartat de senaste åren, i spåren av detta har det även uppstått säkerhetsrisker. Desto fler användaren ett

operativsystem har, desto mer har en hackare att vinna på att knäcka systemet. Precis som för datorer är det en evig kamp mellan de som försöker bryta sig in och de som försöker hålla de ute.

Eftersom mobila operativsystem är relativt nya i jämförelse med de för datorer är säkerheten inte på samma nivå än. Ett annat problem är också folks inställning och kunskap. Många i Sverige är rädda för att få sabotageprogram (t.ex. virus) i sin dator, men det är få som installerar antivirusprogram på sina smartmobiler. Det lär hända mycket på säkerhetsfronten för mobiltelefoner, förhoppningsvis kommer även människors inställning att förändras.

För att lösa säkerhetsfrågan finns det även förslag på att använda t.ex. textmeddelanden (sms). När användaren loggar in med sitt lösenord får de ett textmeddelande till sin smartmobil, meddelandet innehåller en kod som användaren matar in. Med denna form av dubbel identifiering kan användaren visa att han eller hon är den som de påstår, och att de använder den registrerade smartmobilen. (*Abraham m.fl., odaterad*)

Mobil inloggning

Trots att smartmobiler inte är lika säkra som datorer börjar det dock komma fler och fler "känsliga" tjänster till smartmobiler. Ett bra exempel på en sådan tjänst är Swedbank som har släppt en app för bankaffärer. Ett säkerhetshål in i bankens system skulle vara förödande. För att öka säkerheten i systemet använder de sig av *BankID*, en tjänst som används för att identifiera användaren. (*Swedbank, 2013*) Även om *BankID* inte är aktuellt för detta projekt ger det hopp för smartmobiler som plattform, eftersom t.o.m. bankerna vågar använda det.

Sabotageprogram

Sabotageprogram är ett program som installeras på användarens dator (ofta utan deras vetskap) och sedan utför uppgifter i hemlighet. Detta kan t.ex. vara att förstöra datorns mjukvara, spionera på användaren eller visa reklam (*IST, 2013*). Det som medför den största säkerhetsrisken i detta projekt är om ett sabotageprogram spionerar på användaren och kopierar deras användarnamn och lösenord. Den mest välkända formen av sabotageprogram är virus, virus används främst för att förstöra mjukvara.

BYOD

BYOD står för *Bring your own device* och innebär att de anställda tar med sina privata enheter till jobbet (*Sophos, 2013*). *BYOD* är idag på stark frammarsch, detta beror på att dagens mobila enheter blir mer och mer kraftfulla. Smartmobiler och läsplattor blir kraftigare och kraftigare, men även bärbara datorer blir mindre och mindre. En anställd önskar kanske exempelvis att ta med sin privata iPad till jobbet, för ha som hjälpmedel vid

presentationer. I vissa fall hjälper arbetsgivaren till att finansiera de anställdas *BYOD*-enheter (*Sophos, 2013*).

Nackdelen med *BYOD* är att en utomstående enhet kan ta med sig sabotageprogram in i nätverket (*Sophos, 2013*). Ett företags nätverk är i regel väldigt skyddat av brandväggar och liknande, och de enskilda datorerna har skyddsprogram installerade. Men de anställdas personliga enheter har inte företaget någon kontroll över. Vissa företag har därför valt att förbjuda *BYOD*, men risken är att de anställda ändå tar med sig sina enheter i smyg. Det bästa är därför att planera för det från början, och försöka göra systemet så säkert som möjligt.

7. Licensfrågor på programvara

Detta kapitel kommer att diskutera de olika licensfrågor som rör s.k. proprietära program samt öppna källkodsprogram. För att ge en bild av hur de hör ihop bör man först förklara s.k. gratis, fri och kommersiell programvara.

Gratis – Som namnet antyder är detta en programvara som är gratis att införskaffa, frivilliga extraavgifter kan dock tillkomma senare.

Fri – Betyder att programmets källkod finns tillgänglig för vem som helst att ladda ner, modifiera och sprida. Ordet fri kommer från ordet frihet, inte gratis.

Kommersiell – Ett program som har skapats för att tjäna pengar, oftast genom att sälja kopior av programmet.

Öppen källkod – Betyder att källkoden är ”synlig”, man har på så vis möjlighet att studera den.

Proprietär – Kallas även ”stängd källkod”, dvs. källkoden är helt dold. Utvecklingen drivs därför endast av ägaren, inga utomstående kan utveckla programmet.

I de flesta fall brukar *fri* och *gratis* programvara användas tillsammans, och *kommersiell* programvara brukar vara *proprietär*. Detta är dock inte alltid fallet, det finns således öppna källkodsprogram som kostar pengar samt proprietära som är gratis.

När det gäller säkerheten för program med öppen och stängd källkod råder det oenigheter (*IBM, 2012*). De som förespråkar proprietära program menar att säkerheten minskar om man släpper källkoden, då en potentiell inbrottstjuv kan studera koden efter säkerhetshål. På andra sidan av diskussionen menar de som företräder öppen källkod att säkerheten ökar i och med att koden granskas av fler ögon. De menar därför att säkerhetsluckor både upptäcks och lagas snabbare. Ett rimligt antagande är att det finns fler som faktiskt vill hjälpa programmet att växa än de som försöker förstöra det (*IBM, 2012*).

Fri och *öppen källkod* är två väldigt snarlika begrepp men har inte samma innebörd. . *Fri* programvara betyder kortfattat att användaren har stora friheter med programvaran, medan *öppen källkod* kortfattat bara säger att källkoden är läsbar (*Gnu, 2013c*). Både uttrycken myntades ungefär samtidigt på mitten av 80-talet, men den *fria* rörelsen har redan från början varit tydlig med att de inte vill bli förväxlade med *öppen källkod*. Det finns program som uppfyller kraven för både *fri* och *öppen källkod*, de kallas då *FOSS (Free and open-source software)*.

Den fria rörelsen startades av Richard Stallman. Han skapade först det fria operativsystemet *GNU* som ett alternativ till de allt fler proprietära operativsystemen. Sedan skapade han *FSF (Free Software Foundation)*, en förening som än idag driver kampen för fri programvara. Han skapade även *Copyleft*, en friare form av upphovsrätt.

Richard Stallman skrev även fyra grundläggande regler för vad fri programvara är (*Gnu, 2013c*):

- *Freedom 0: The freedom to run the program for any purpose.*
- *Freedom 1: The freedom to study how the program works, and change it to make it do what you wish.*
- *Freedom 2: The freedom to redistribute copies so you can help your neighbor.*
- *Freedom 3: The freedom to improve the program, and release your improvements (and modified versions in general) to the public, so that the whole community benefits.*

Krav nummer 1 och 3 är omöjliga att genomföra utan tillgång till källkoden. Enbart *öppen källkod* är dock inte tillräckligt för att uppfylla alla krav, då ett program med *öppen källkod* kan ha restriktioner på hur mycket en användare får modifiera eller distribuera programmet.

Upphovsrätt

Upphovsrätt (engelskans *copyright*) ger skaparen äganderätt över sitt verk med en rättslig tyngd bakom sig. Detta medför att skaparen får rätt att bestämma hur deras verk ska hanteras. Detta gäller främst hur det ska distribueras, men man styr även över dess utveckling (*WIPO, odaterad*). Om någon bryter mot upphovsrätten blir de således juridiskt skadeståndsskyldiga.

Det finns dock undantag där upphovsrättskyddet inte gäller (*Klang, 2010*). Exempelvis får användare av "*mindre omfattande verk*" kopiera materialet för privat bruk, där privat bruk även innefattar familj och vänner. Det finns även "*citaträtt*" vilket innebär att man får citera ett upphovsrättskyddat material. Det är dock reglerat i lag att man måste följa *god sed* och endast citera i den omfattning som är befogat. Med *god sed* menas att man ska citera på ett sådant sätt att den ursprungliga författaren ej kränks eller att dennes budskap förvrängs.

Copyleft

Copyleft är en ordvits på *copyright* (upphovsrätt), då det syftar till att motverka det. Medan upphovsrätt i regel används för att säkra ägarens rätt används *copyleft* för att säkra *användarnas* rätt (dvs. användarens rätt att modifiera och sprida verket) (*GNU, 2013a*). Trots att *Copyleft* finns för att motverka den traditionella upphovsrätten är även detta en form av upphovsrätt, dock är det en form som ger användarna *mer* frihet. De licenser som stödjer *Copyleft* är de som stödjer fri programvara, och ett program som är licenserat under en licens som stödjer *Copyleft* kan inte göras "*ofri*" (proprietär).

7.1. Proprietära program

Proprietära program tillåter inte att användare ser källkoden och därmed inte heller har möjlighet modifiera den. Detta är alltså motsatsen till *fri* programvara. På så vis hålls källkoden hemlig då den version av programmet som släpps till användaren inte är läsbar för människor (binärspråk) (*LINFO, 2005*). För att kunna hålla koden dold måste man dock ha legalt skydd som t.ex. upphovsrättsskydd. Proprietära program kan även innehålla avtal som användaren måste godkänna för att få tillgång till programmet, avtal som har som syfte att begränsa användningen av programmet. Exempel på restriktioner kan vara att man inte får kopiera programmet eller att man bara får installera det på ett begränsat antal datorer.

De flesta proprietära program är kommersiella program som säljs för att företaget ska tjäna pengar, men det finns även program av denna typ som är gratis. Ett exempel på ett proprietärt program som är gratis är *Skype* (*Skype, 2013*). Skype livnär sig på de frivilliga betaltjänster som programmet erbjuder men i övrigt är programmet gratis. Det har på senare tid blivit populärt med proprietära program som är gratis, inte minst för smartmobiler. Detta beror på att branschen har blivit bättre på att tjäna pengar på gratisprogram, detta sker främst via reklam eller frivilliga avgifter som ger användaren fördelar.

7.2. Öppet källkodsprogram

Öppen källkod (engelskans *Open source*) betyder att koden finns tillgänglig för alla. Man kan dels undersöka koden för att se hur någon annan har löst ett problem men även använda koden för att själva bygga om/på programmet. Vid öppen källkod har man även rätt att sprida sin modifierade version av programmet vidare (*Open Source Initiative, 2013a*).

Möjligheten för vem som helst att ändra i koden har gett upphov till att det finns många olika färdiga lösningar. Chansen är alltså hög att det som man letar efter redan finns då det finns många varianter av lösningarna. En annan fördel med öppen källkod är att de nästan alltid gratis att använda.

Öppen källkod förekommer oftast när det gäller datorprogram. Några exempel på detta är Mozilla Firefox, GIMP och Open Office. Konceptet/formatet har även varit väldigt framgångsrikt när det gäller operativsystem (t.ex. Linux och Android) samt programmeringsspråk (t.ex. PHP och Python).

Begreppet öppen källkod förekommer främst inom datorvärlden, men en mer oortodox open source-lösning var OpenCola. Det var en cola som använde sig av samma princip dvs. att man via deras hemsida kunde få tillgång till deras recept (*OpenCola, 2000*). Som med open source för datorer medförde detta att användarna började experimentera och driva utvecklingen framåt och att i takt med att produkten utvecklades släpptes nya versioner. Receptet går än i dag att finna på internet, även om OpenCola-projektet är nerlagt (*wikiHow, 2013*).

7.2.1. Licenser

Program med öppen källkod har olika licenser vilka bestämmer vad man får och inte får göra med koden. Man kanske t.ex. får lov att ändra i koden för privat bruk men inte sälja det nya programmet vidare. Den mest använda licensen är *GNU GPL*, men ett annat välanvänt exempel är *BSD* (*Open Source Resource Center, 2013*).

Genom att ägaren licenserar sitt program har han eller hon möjlighet att ge användaren rättigheter som i normala fall endast finns för upphovsrättsägaren, som t.ex. att modifiera och kopiera verket. Licenser används dock även för att upprätta vilka skyldigheter som användaren har, dvs. trots att man får kopiera programmet får inte detta ske hur som helst. Ett exempel på detta är vilken typ av licens som det härledda verket ska släppas under, många fria program använder licenser som kräver att även modifierade verk ska släppas under en fri licens (se *Copyleft*).

För att en licens ska klassas som *öppen källkod* måste den uppfylla vissa kriterier (*Perens, 2007*). Nedan följer några exempel på sådana krav, som t.ex. att:

- användarna måste ha tillgång till källkoden. Det inte är lagligt att hindra detta via olika tekniska lösningar eller kostnader.
- man får kopiera eller sälja programmet vidare utan att fråga tidigare författare.
- programmet ska vara öppet för alla och inte bara en utvald grupp.

GNU GPL

GNU GPL (*GNU General Public License*) är en stark företrädare för "fri programvara", där fri har sitt ursprung i ordet frihet snarare än gratis. Källkoden ska m.a.o. vara tillgänglig för vem som helst. Man ska ha friheten att dels läsa och modifiera källkoden, men man får även lov att sprida det vidare utan att be tidigare författare om lov. (*GNU, 2013b*)

En produkt som registreras under licensen GNU GPL är för alltid låst till den. Detta innefattar även senare versioner av programmet, men det gäller även andra program som vill använda sig av kod från det licensierade programmet (*GNU, 2013b*). Detta är eftersom det är kompatibelt med *Copyleft*. Om program *A* vill använda sig av delar av (eller hela) program *B* som är licensierad under GNU GPL måste även program *A* släppas under GNU GPL (eller annan liknande licens som är kompatibel med GNU GPL).

BSD

BSD (*Berkeley Software Distribution*) har mycket gemensamt med GNU GPL. Det som skiljer dem åt är vilka rättigheter man har när det gäller att sprida sitt verk vidare. Till skillnad från GNU GPL kan program som är licensierade under BSD implementeras i kommersiella program (*FreeBSD, 2013*). Det blir alltså lättare att tjäna pengar på sin mjukvara. Fördelen med BSD är att det inte är lika strikt som GNU GPL. Om man som utvecklare vill driva ett projekt som man på sikt hoppas kan bli en ny standard kan således BSD vara fördelaktigt. Detta eftersom

företag då kan använda din kod utan att slutprodukten blir låst till en viss form av licens.

Andra licenser

Andra exempel på populära licenser är *Academic Free License (Open Source Initiative, 2013b)*, *Artistic License (Open Source Initiative, 2013c)*, *Apache license (Apache Software Foundation, 2004)* och *MIT license (Open Source Initiative, 2013d)*. Dessa licenser är ej kompatibla med *Copyleft*.

Figur 7.1 visar några av de vanligaste licenserna och jämför vad de ger användaren för rättigheter. Man kan exempelvis se att *GNU GPL* är den som ger användaren flest friheter (enligt de uppställda kraven).

<h3>Comparison of the Open Source Licences</h3> <p>The bullets mark if the the licence explicitly states the item in question. Implicit items are not marked by this chart</p>		Must distribute license with binray or source	Cannot use contributors name to endorse	There has to be a notification for changed files	Any change must distributed in source form	Lets you provide warrenty if you want to, normally no	Lets you explicitly charge for providing warrenty or guranteee or transfer of code	All derivative work must be under the same license	Must show License when Run from command line	Non derivative works can have different license	May exclude countries where there is a contradiction with patent in that ocuntry	Must describe any deviation due to regulation
Apache License 2.0	●	●	●	●	●	●		●				
Common Development and Distribution License	●		●	●				●				
GNU General Public License (GPL)	●		●	●	●		●	●	●	●	●	
GNU Library General Public License (LGPL)	●		●	●	●		●			●	●	
Microsoft Public License (Ms-PL)	●	●										
Microsoft Reciprocal License (Ms-RL)	●	●								●		
Mozilla Public License 1.1 (MPL)	●		●	●								●
New BSD License	●	●										
The MIT License	●											

Figur 7.1: Olika typer av licenser (dholm, 2011)

8. Licensfrågor på data

Detta kapitel kommer att diskutera frågor och lösningar rörande licensfrågor för data.

8.1. Inledning

Licensfrågor för data och programvara är väldigt lika då programvara mer eller mindre är data. Det finns t.ex. licenser för data som är inspirerade av dem för programvara. Även tanken om "fri" tillgång till material återfinns hos både programvara och data.

Ett exempel på hur man kan skydda sin geografiska information är med *GeoDRM (Geospatial Digital Rights Management Reference Model)*, även kallad *ISO 19153* (ISO-standarder skapas av *International Organization for Standardization*) (*Danko D. M. m.fl., 2012 s.462*). Den bestämmer vad användaren får eller inte får göra med det licenserade materialet, men den bestämmer även vad användaren kan förvänta sig av ägaren. Exempel på krav som användaren kan ställa på ägaren är regelbundet underhåll och s.k. "buggfixar".

Det norska *Kartverket* släppte i början av 2013 sin information över platsnamn. Denna information innehåller namn och plats på cirka 950 000 platser i Norge. Denna information har varit tillgänglig redan tidigare, men då mot betalning (*Kartverket, 2013*). Informationen är licenserad under en *Creative Commons*-licens, *CC BY 3.0 NO (OpenStreetMap, 2013a)*.

Att släppa informationen fri/gratis hjälper allmänheten, men tyvärr "finns det inga gratisluncher". Även om tillgången till information är gratis för användaren betyder inte detta att den är gratis att producera. När norska *Kartverket* valde att släppa delar av sin information gratis så innebar detta en direkt förlust i inkomst. Samma förlust av inkomst skulle ske om svenska *Lantmäteriet* släppte sina kartor gratis. Då *Lantmäteriet* till största delen är självförsörjande, dvs. inte finansieras av skattepengar (*Lantmäteriet, 2013*). Skulle en gratis tillgång av kartorna ge en betydande inkomstförlust.

8.2. Digital rights management

Digital rights management (DRM) används för att skydda rättighetshavare på internet. Det är inte ett alternativ till upphovsrätten utan ett hjälpmedel för att upprätthålla det.

Anledningen till att man har valt att skapa DRM är att det är väldigt enkelt att kopiera information över internet. När man pratar om DRM brukar man oftast syfta på de tekniska lösningar som skyddar produkten, men det finns även det juridiska skyddet som kan träda i kraft. Exempel på tekniska lösningar för DRM är *digital vattenmärkning, kryptering*, eller att man behöver koppla upp sig mot en *licensserver* för att få tillgång till ens material (*Becker m.fl., 2003*).

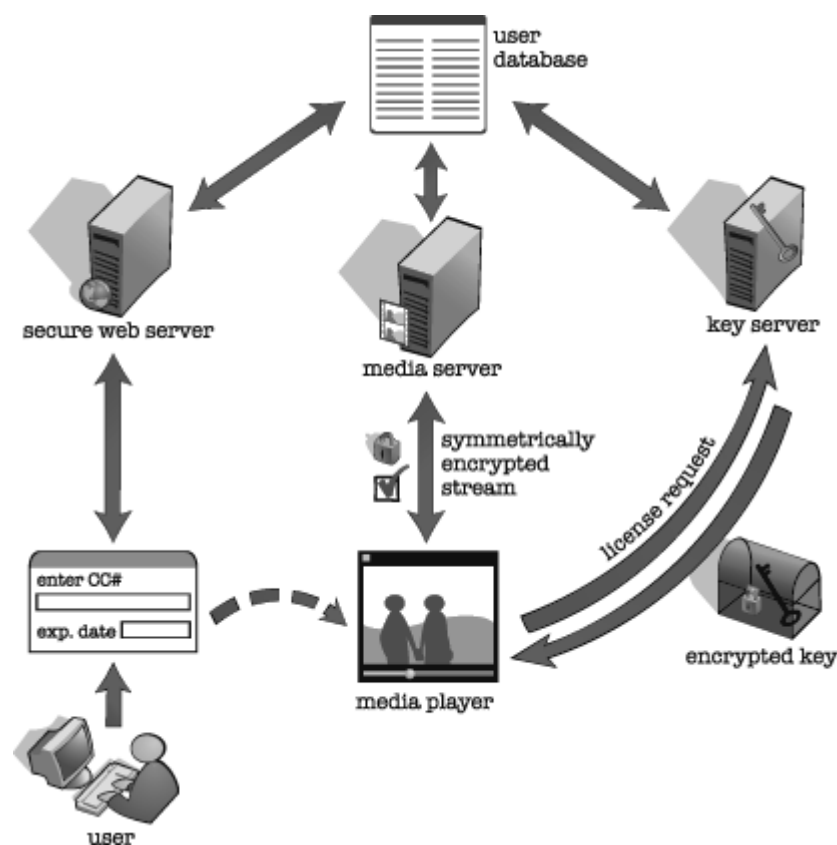
Free Software Foundation (FSF) har valt att kalla DRM för *Digital restriction management*. Anledningen till detta är att de anser att DRM bara är ett verktyg för att sätta restriktioner

på användaren. Detta bryter mot den ideologi som FSF står för, nämligen yttrandefrihet och fri tillgång till information. (*Defective by Design, 2013*)

DRM används idag främst av musik-, film och spelindustrin, men det går även att använda för många olika typer av digitalt material som t.ex. kartor.

Figur 8.1 visar ett exempel på ett DRM-system för en film, förloppet kan tänkas ske ungefär såhär:

1. Användaren försöker spela upp en film.
2. Systemet kräver då att användaren loggar in med användarnamn/lösenord.
3. Systemet kontrollerar det givna användarnamnet och lösenordet mot databasen.
4. När användaren är inloggad kan den be systemet om att skickas en nyckel för en specifik film som används för att dekryptera filmen (dvs. låsa upp den).
5. Om användaren har rätt att se filmen skickas nyckeln till användaren.
6. Användaren kan nu se filmen.



Figur 8.1: DRM (*webreference, 2003*)

DRM är ett verktyg för att säkra ett verk mot *piratkopiering* (illegal kopiering och användning), DRM har dock mött väldigt stark kritik de senaste åren. Anledningen till detta är att DRM försvårar användningen av produkten, även om användningen sker på ett legalt sätt. Den industri som har fått mest kritik är troligen spelindustrin, i sin kamp mot piratkopiering har de tagit till kraftigare och kraftigare medel. Det senaste verktyget som de

har använt sig av är att användarna måste vara konstant uppkopplade mot deras server för att kunna spela, m.h.a. detta kan spelföretagen kontrollera exakt vem som har tillgång till deras spel. Detta system har orsakat klagomål då:

- Användaren måste vara konstant uppkopplad mot internet för att spela. Detta innebär att man inte kan använda sig av sitt spel om man befinner sig på en plats som inte har tillgång till internet. man konstant arbetar mot spelföretagets server leder detta till att spelet "hackar" (går långsamt) om man har en dålig internetuppkoppling, eller om spelservern är överbelastad.
- användarna kan inte spela spelet medan spelföretaget genomför underhåll på servern.
- spelet slutar att fungera om spelföretaget en dag väljer att stänga ner alla servrar. Det låter konstigt, men det vore samtidigt konstigt om spelföretaget förväntades ha sina servrar igång för *all framtid*. Även om man bortser från de ekonomiska skälen är konkurs något att oroa sig för.

Ett nyligen (mars 2013) uppmärksammat fall är spelet *Sim City*, där spelet krävde att spelarna var konstant uppkopplade mot deras server. Deras DRM var uppbyggd så att alla uträkningar som krävdes utfördes av spelservern, det blev därför omöjligt att spela spelet utan en internetuppkoppling och ett legalt exemplar av spelet (*Wired, 2013*). De problem som uppstod var de som nämndes ovan. Det i särklass största problemet var dock att spelservrarna inte klarade av att genomföra alla beräkningar till dem hundratusentals som försökte spela. Spelare lyckades inte logga in, och de som väl lyckades logga in kunde oftast inte spara sina spel. Det tog flera dagar innan spelet fungerade som det var tänkt och all mediauppståndelse gav *DRM* mycket negativ uppmärksamhet.

8.3. Fria data

Den mest kända licensen för *fri* data är *Creative commons*, men det finns självklart fler. När det gäller fri geografisk information är *OpenStreetMap* i fronten, de tillhandahåller fri geografisk information som är skapad av volontärer (s.k. *VGI*-data). *OpenStreetMap* använder sig av en fri licens som heter *ODbL*.

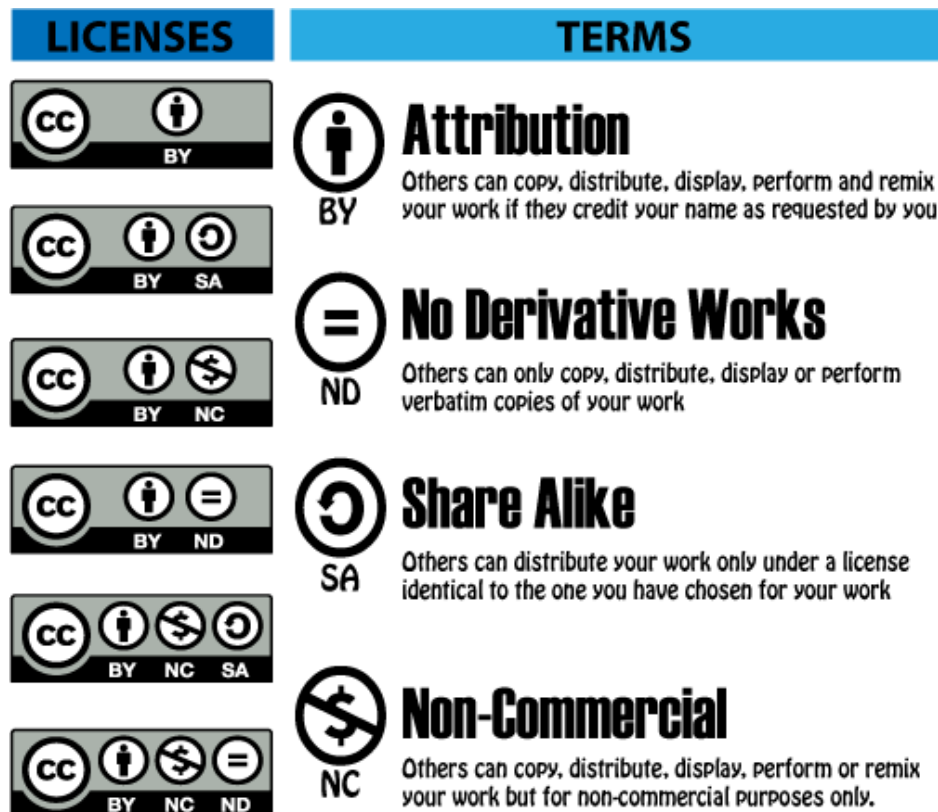
8.3.1. Creative commons

Creative Commons (CC) är en ideell organisation, deras namn är dock mer förknippat med deras sex olika licenser. Deras första licens släpptes 2002 och var till större delen inspirerad av mjukvarulicensen *GNU GPL*, CCs licens var dock ämnad för mer än bara mjukvara. CC har en liknande ideologi som *Copyleft*; man använder upphovsrätten för att stärka användarnas rätt till (att använda) innehållet, snarare än tvärtom. (*Creative Commons, 2013*)

CC har idag sex olika licenser som alla mer eller mindre följer samma "fria" ideologi. Det som samtliga licenser har gemensamt är att man måste vara försiktig när man ska sprida verket

vidare, man måste vara tydlig med vem den ursprungliga författaren är. Egenskaper som skiljer de olika licenserna åt är om:

- användaren får modifiera verket.
- verket måste släppas under samma/liknande licens.
- verket får släppas som del i ett kommersiellt program.



Figur 8.2: De sex Creative Commons-licenserna (Shihaam 2011)

CC licenser används av t.ex. *Flickr* och *Wikipedia* (Creative Commons, 2013). Detta betyder exempelvis att man får använda bilder från *Flickr*, om man är tydlig med vem "författaren" är till bilden. Ägaren till bilden på *Flickr* måste dock ha valt att andra ska få ta del av bilden, dvs. det är möjligt att använda sig av en striktare licens. *Wikipedia* använder en licens som ger användarna rätt att modifiera och bygga på deras verk, samt använda informationen s kommersiellt. Informationen är dock låst till sin licens vilket i detta fall innebär att alla härledda verk måste använda samma typ av licens.

Licenserna är oåterkalleliga vilket betyder att upphovsrättshavaren inte kan byta licens för att stoppa spridningen av sitt verk (Klang, 2010). Han eller hon kan självklart stoppa sina egen spridning av verket, men de verk som redan är i omlopp går inte att återkalla.

Upphovsrättshavaren har dock möjlighet att välja att licensera materialet under mer än en licens. Om verket är släppt under en CC-licens som inte tillåter kommersiellt bruk kan upphovsrättshavaren fortfarande välja att licensera verket under en ny licens som tillåter det

(Klang, 2010). De restriktioner som CC-licensen medför gäller bara för användarna och inte för upphovsrättshavaren.

8.3.2. ODbL

ODbL (Open Database License) är en licens från den ideella organisationen *Open Knowledge Foundation*. Licensen är mest känd för att den används av *OpenStreetMap*, en karttjänst som liknar *Google Maps*. ODbL är en licens för databaser, till skillnad från CC som är en licens för enskilda verk. Detta är anledningen till att *OpenStreetMap* valde att gå ifrån sin gamla licens (CC) och byta till ODbL, de efterfrågade ett skydd för sina databaser.

ODbL bestämmer vad användaren får och inte får göra med databasen. Grundtanken med licensen är som för många andra "fria" licenser att säkra användarnas rätt till materialet, men den används samtidigt för att kontrollera den. Licensen bestämmer t.ex. till vilken grad användaren kan kopiera och återanvända materialet, samt hur man hänvisar tillbaka till dess rätte ägare. Material som är licenserat under *ODbL* får inte modifieras och sedan publiceras i någon annan licens än *ODbL* (eller licenser som är kompatibla med det). Detta är för att informationen inte ska bli "ofri" på något sätt. Även om en databas är licenserad med *ODbL* kan innehållande verk vara licenserade under andra licenser (*Open Data Commons, 2013*).

8.3.3. VGI-data

VGI (*volunteered geographic information*) är som namnet antyder geografisk information som är skapad av volontärer. Precis som med *Wikipedia* är det användarna som både skapar och granskar informationen. Skapandet av information sker t.ex. när en användare ger sig ut i fält med en GPS-mottagare och mäter in en detalj, t.ex. går längs en cykelväg (*Goodchild, 2007*). Det kan även vara information som t.ex. var en viss affär eller gata befinner sig.

Det finns en del olika teorier om varför folk väljer att hjälpa till med VGI. En teori är att man upplever att man gjort en insats för allmänheten och att man därmed har gjort något bra (*Goodchild, 2007*). Detta behov att hjälpa allmänheten kan vara extra starkt i svårt drabbade områden. Det är även ett sätt att synas på då alla ändringar bokförs, det är alltså möjligt för andra att se att det är precis du som har ändrat i kartan. Om det är en väg som saknas kan man som användare mäta in den, sedan i framtiden kan man se den vägen och tänka på sin insats.

Det finns dock ett antal problem med att använda sig av allmänheten för att samla in information (*Longley m.fl., 2011*), det mest uppenbara är att man inte vet vilken utrustning eller utbildning de olika användarna har. Utöver detta så finns det även andra problem som t.ex. att:

- allmänheten inte har rätt att beträda alla platser, t.ex. militära områden.

- människor ofta favoriserar områden som de rör sig i. Detta leder t.ex. till att fattiga områden inte blir kartlagda, då deras invånare inte har råd med den nödvändiga utrustningen.
- alla länder är inte lika öppna med sin geografiska information. Ex på detta är länder som ej skulle acceptera att man märker ut var deras militära inrättningar befinner sig.
- viss data kan utnyttjas i fel syfte. Om man t.ex. har en karta över alla handikappade i ett område kan detta utnyttjas av inbrottstjuvar.

8.3.4. OpenStreetMap

OpenStreetMap (OSM) är världens ledande karttjänst som består av VGI-data. Projektet påbörjades 2004 som ett experiment då dess skapare Steve Coast ville skapa ett alternativ till de kommersiella karttjänsterna. Ursprungligen var det en väldigt simpel tjänst där användare kunde hjälpa till att rita kartan men med åren har det vuxit till en mycket avancerad karttjänst. Det ursprungliga syftet var att kartlägga vägar men det innehåller idag all möjlig information som t.o.m. markanvändning och busslinjer. (Bennett, 2010)



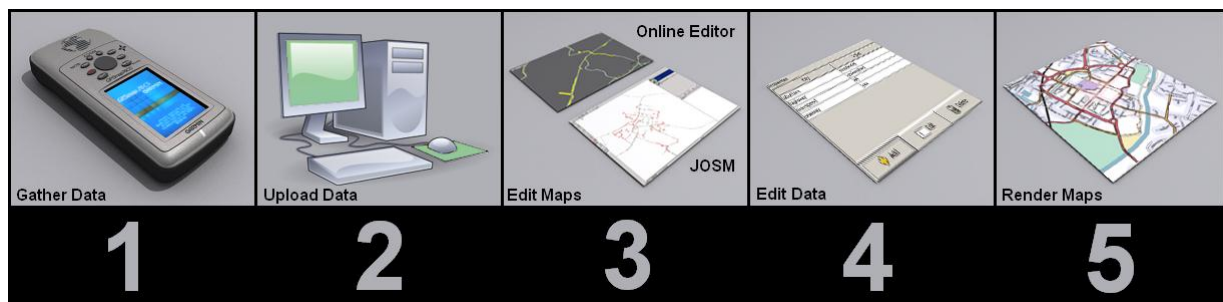
Figur 8.3: OpenStreetMap

Vilken information som finns kartlagd är väldigt olika beroende på var man befinner sig. Större städer i västvärlden är ofta väldigt väl kartlagda, ofta t.o.m. mer detaljerade än andra tjänster som t.ex. i *Google Maps*. Detta beror på att det finns fler entusiaster som kan hjälpa till i större städer (Bennett, 2010). Mindre städer är därför ofta sämre kartlagda, samt så är det ett stort problem i fattiga länder som inte har råd med den utrustning som behövs. Trots detta anses OSM som en bra lösning för krisdrabbade länder som t.ex. när ett land har drabbats av en miljökatastrof eller befinner sig i krig. Användare med en GPS-mottagare kan då hjälpa till att kartlägga områden som kanske inte tidigare finns med på kartan alternativt kartlägga förändringar i landskapet. Volontärarbete i krisdrabbade länder är inget nytt (t.ex. *läkare utan gränser*), men nu börjar det sprida sig till att även omfatta GIS (Bennett, 2010).

OSM har kallats för den geografiska informationens motsvarighet till *Wikipedia*. Precis som för Wikipedia har även OSM fått mycket kritik för deras bristande precision då de användarna som hjälper till att uppdatera OSM oftast är GIS-amatörer med GPS-mottagare med dålig precision. Studier i England visade på ett medianfel på cirka 6 meter (*Bennett, 2010*). Det är därför omöjligt att använda OSM för mer detaljerade uppgifter men för enklare navigering duger systemet gott. Studien i England ville snarare belysa problemet med att många områden inte är kartlagda alls snarare än att de är kartlagda med låg noggrannhet.

Det typiska förloppet för när man ska samla in den geografiska informationen kan ses i figur 8.4. För att exemplifiera förloppet väljer vi en användare som ska mäta in ett antal vägar:

1. Användaren ger sig ut i fält för att samla in information. Denna information består delvis av inmätta punkter från en GPS-mottagare, men även allmän information om området. Allmän information är t.ex. vägarnas namn.
2. Den insamlade informationen laddas upp på till OSMs server, detta gäller främst de inmätta punkterna. Vägnamnen behandlas senare.
3. Informationen editeras, i detta exempel innebär detta att de inmätta punkterna ska sättas ihop till vägar (linjer).
4. Här matas all ytterligare information in, t.ex. så namnges vägarna.
5. Slutligen målas den färdiga kartan upp. Det är möjligt för användaren att skapa en egen specialanpassad karta om de vill, t.ex. en Kalmarkarta med enbart cykelvägar.



Figur 8.4: Att skapa ny data för OpenStreetMap (*OpenStreetMap, 2013b*)

Ett alternativ till detta klassiska arbetssätt är att man tar hjälp av befintliga kartor och flygbilder för att lägga till information i OSM. Man använder då främst flygbilder för att rita upp den nya kartan och därefter den befintliga kartan för att t.ex. fylla i information som gatunamn (*OpenStreetMap, 2013b*).

Den geografiska informationen i OSM kommer ursprungligen från volontärer, på senare år har de dock även fått information donerad till sig. Denna information kommer från t.ex. företag, kommuner eller statliga verk. Ett av hundratals exempel på donerad information är tidigare nämnda norska platsnamn som donerats från *Kartverket* (*OpenStreetMap, 2013a*).

OSM använder två olika licenser för sin information. Olika delar av informationen använder olika licenser, det är inte så att all information täcks av båda licenserna. Ursprungligen

användes *Creative Commons Attribution-ShareAlike 2.0 (CC-BY-SA)*, men man har de senaste åren arbetat med att porta över all information till den nya licensen *Open Data Commons Open Database License (ODbL)* (*OpenStreetMap, 2013c*). De två licenserna är dock väldigt lika varandra, de är båda väldigt "fria". Man får t.ex. kopiera, modifiera och distribuera sin modifierade version, kravet är att man skyltar med att informationen kommer från OSM.

Den 12 september 2012 meddelande OSM att de var klara med flytten från *CC-BY-SA*-licensen till *ODbL* (*OpenStreetMap, 2012*). Denna nya licens gäller dock bara deras egna material. OSM har fått geografisk information donerad till sig av företag och länder, vilken licens den är i kan bara upphovsrättshavaren bestämma.

Det är fritt fram att använda sig av data från OSM då OSMs syfte är att tillhandahålla fri geografisk information. Ska man integrera informationen i sin karta behöver man dock vara vaksam för licensfrågor. OSM är licenserad under en fri licens som kräver att informationen ska fortsätta vara fri. Detta kan medföra att om man integrerar deras information i sitt system kan man bli tvungen att licensera även sin egen data som fri (en fri licens som är kompatibel med *ODbL*). Man kan exempelvis använda sig av OSM som baskarta utan att behöva licensera om sin data, men om man ska koppla ihop sin data med den från OSM kräver detta en fri licens (*Open Data Commons, 2013*).

Humanitarian OpenStreetMap Team

HOT är ett projekt som syftar till att använda OpenStreetMap för att hjälpa krisdrabbade platser i världen. En av deras mest kända insatser var vid jordbävningen på Haiti 2010. På bara 48 timmar hade större delar av Haitis huvudstad kartlagts av ungefär 600 volontärer via OpenStreetMaps hemsida. Detta arbete skedde till största delen från utanför Haiti m.h.a. kartor och flygbilder, men någon månad efter jordbävningen åkte även volontärer till Haiti med GPS-utrustning för att kartlägga området (*HOT, 2013*).

9. Genomgång av lösningsalternativ

Detta kapitel kommer att förklara de tekniska val som genomförts, dvs. tekniken bakom slutprodukten. Mer än en möjlig lösning kommer att presenteras.

9.1. Hårdvara

Läsplattor och smartmobiler har valts som plattform för systemet, men smartmobiler har haft högsta prioritet. Anledningen till detta är att mätingenjörerna redan har smartmobiler som arbetstelefoner, inköpskostnaden blir därmed noll. En läsplatta är inte lämpad för detta projekt, p.g.a. att systemet är tänkt att ses som ett komplement. Det nya systemet är inte tänkt att används vid varje uppdrag, utan endast vid behov. Det är därför inte önskvärt att mätingenjörerna ska bära med sig en läsplatta, ifall de behöver nå systemet.

Systemet kommer att vara kompatibelt med läsplattor trots att det är optimerat för smartmobiler. Enda nackdelen är att knappar och texter kan uppfattas som för stora, då de är anpassade för en mindre skärm. Detta får anses vara ett minimalt problem då väldigt få användare förväntas använda läsplattor.

Andra alternativ på hårdvara är bärbara datorer (*laptop*) eller s.k. tåliga handdatorer (*rugged computer*). Problemen med detta är att:

- Det blir ännu ett verktyg att bära med sig, till skillnad från smartmobilen som man redan bär med sig.
- Man behöver införskaffa ny hårdvara. Det blir därmed dyrare, men det ökar även installationstiden då det inte sker på en redan fungerande enhet.

Andra nackdelar med tåliga datorer är att:

- Det försvårar utvecklingen. Marknaden för smartmobiler och läsplattor är i dag väl utvecklad, och därmed också verktygen som används för utveckling.
- Det blir en längre inlärningsfas, då inte alla är bekanta med dem. Detta till skillnad från smartmobiler som majoriteten av alla redan har använt.

9.2. Mjukvara

När det gäller mjukvaran anser författaren att det inte finns en uppenbar lösning. Därför finns tre olika lösningar redovisade. Dessa är uppdelade efter hur informationen hanteras på serversidan. Detta sker via ArcGIS Online, ArcGIS Server eller en hybrid av de två. För varje serverlösning presenteras sedan olika förslag på klient.

Valet av lösning beror på många olika faktorer, t.ex. antalet användare, målgrupp, tidsram för utvecklingen och personliga preferenser. Om ett liknande projekt hade genomförts för en annan avdelning inom Kalmar kommun hade valet av lösning kanske bli annorlunda. Även

om ett identiskt projekt som detta skulle genomföras för ett antal olika kommuner i Sverige skulle resultaten bli olika.

För de tre valda serverlösningarna har det fokuserats på hur man kan lösa funktionaliteten och åtkomsten av informationen. Funktionaliteten har delats in i fyra delproblem: *Bakgrundskarta, editeringsmöjligheter, stomnätsskisser och säkerheten*. Slutligen förklaras vad som har implementerats och testats för varje lösning.

9.3. Val av lösningsalternativ

I projektets början undersöktes flera olika tekniker, men det är främst ArcGIS-lösningar som har valts att gå vidare med. Detta beror på att Kalmar kommun uttryckte speciellt intresse för detta, eftersom det är ArcGIS som de använder i dag. Fördelen med en ArcGIS-lösning är att det underlättar integrationen då de i dag har ett befintligt ArcGIS-system, de olika delarna av systemet får enklare att kommunicera med varandra. Det är även delvis en kostnadsfråga. ArcGIS-paketet är för dyrt för många små aktörer, de väljer därför en billigare lösning. Kalmar kommun har redan i dag en kommunlicens på ArcGIS. Merkostnaden för licens för detta nya system blir därmed noll.

9.3.1. Serversidan

På serversidan har det valts att undersöka ArcGIS Online och ArcGIS Server vidare. Tre lösningsförslag kommer att presenteras i detta kapitel; en lösning som använder AGO, en som använder ArcGIS Server och en som använder båda. Anledningen till detta är att alla tre lösningarna har sina fördelar, vilken lösning som är optimal beror helt på omständigheterna.

Anledningen till att dessa två lösningar (AGO och ArcGIS Server) har valts framför t.ex. *MapServer* och *GeoServer* är att Kalmar kommun idag har ett väl integrerat ArcGIS-system. Den geografiska informationen är anpassad för att användas med ArcGIS Desktop, det blir därmed enkelt att sprida sin information till ArcGIS Server direkt från ArcGIS Desktop. Det finns redan idag har ett antal webbtjänster som konsumerar karttjänster från ArcGIS Server. Både GIS-avdelningen som driver tjänsterna och IT-avdelningen som driver serverdatorn har därmed erfarenhet av ArcGIS Server.

9.3.2. Klientsidan

De tre klientlösningar som valts att gå vidare med är ArcGIS (app), ArcGIS Collector och en egenutvecklad webbapp. Serverlösningar 1 och 3 kan använda sig av alla tre klienterna, lösning 2 (ArcGIS Server) kan endast använda sig av ArcGIS (app) eller webbappen.

Det finns flera alternativ för att nå kartor från AGO, men i detta projekt har det valts att fokusera på tre av dem. Det finns andra färdiga appar på marknaden som man kan använda, men detta känns osäkert. De två färdiga ArcGIS-apparna utför sin uppgift väl godkänt.

Eftersom det är ett företag som står bakom dem vet man att man kan förvänta sig support och regelbundna uppdateringar. Detta är inte fallet för flertalet andra appar på marknaden. Men även om man bortser från detta faktum har författaren inte hittat någon annan app som uppfyller kraven bättre (jft tabell 5.1).

Att välja en webbapp som lösning är dock väldigt ospecifikt, det finns otaliga sätt att utveckla en på. Några exempel på lösningar är de som har diskuterats i avsnitt **6.4.3**.

9.4. Lösning 1 – ArcGIS Online

En ren ArcGIS Online-lösning innebär att all geografiska information finns lagrad i "molnet" (dvs. på ArcGIS.com). Man måste därför ladda upp all information som man vill använda.

Krav **1.1** (att nå baskartan) går att lösa genom att man laddar upp den geografiska informationen, därefter kan man använda den i sin karta. Nackdelen med detta är att man då arbetar mot en kopia av sin information. För större system är detta inte en bra lösning. Att lagra kommunens geografiska information på AGO skulle även kosta väldigt många krediter.

En fördel med att publicera sin karta via AGO är att den blir lättillgänglig, du låser dig inte heller till en typ av klient. Om t.ex. person A föredrar *ArcGIS (app)* och person B föredrar *ArcGIS Collector* är det inte ett problem. De kan utan problem installera olika klienter och koppla upp sig mot samma server.

9.4.1. Funktionalitet

Bakgrundskarta

Det finns ett antal förinställda bakgrundskartor som man kan välja mellan, exempelvis OpenStreetMap, Microsoft Virtual Earth och Bing Maps.

Editeringsmöjligheter

För att skapa ett editerbart ritlager behöver man en *Feature service*, som gör att man kan modifiera dess egenskaper (features). Man skapar en sådan tjänst genom att först ladda upp det lager som ska vara editerbart, sedan väljer man "Skapa tjänst" på AGO. Efter det öppnar man en karta i AGO och infoga sin nya *Feature service*.

Om man ska skapa en ny karta med editeringsmöjligheter (där användarna har möjlighet att lägga till egna punkter) är det rekommenderat att använda ett tomt lager som grund. Det är viktigt att det nya lagret använder rätt koordinatsystem, samt att det har textfält för den information som man vill samla in (t.ex. namn, beskrivning och datum).

Stomnätsskisser

Kalmar kommun använder sig i dag av dynamiska länkar för att nå stomnätsskisserna via ArcGIS Desktop. Ett liknande system finns för AGO. Figur 9.1 visar hur man kan använda punktens idnummer för att generera en länk till bilden, samt en rubrik. Vi kan anta att det finns en testpunkt i kartan med idnummer 35. När vi klickar på denna punkt kommer en ruta visas med titeln "Bild {Id}", dvs. kommer det att stå "Bild 35". Länken till bilden fungerar på samma sätt, systemet kommer att försöka visa bilden "http://www.google.com/35.jpg".

Nackdelen är att stomnätsskisserna måste publiceras på internet för att bli tillgängliga mobilt. Detta må vara enkelt att genomföra, men problemet är då att de blir tillgängliga för alla. För att nå bilderna krävs dock att man känner till dess adresser. Det går att länka till bilderna direkt på intranätet, men då kan de bara nås via Kalmar kommuns datorer.



Figur 9.1: Konfigurera bild

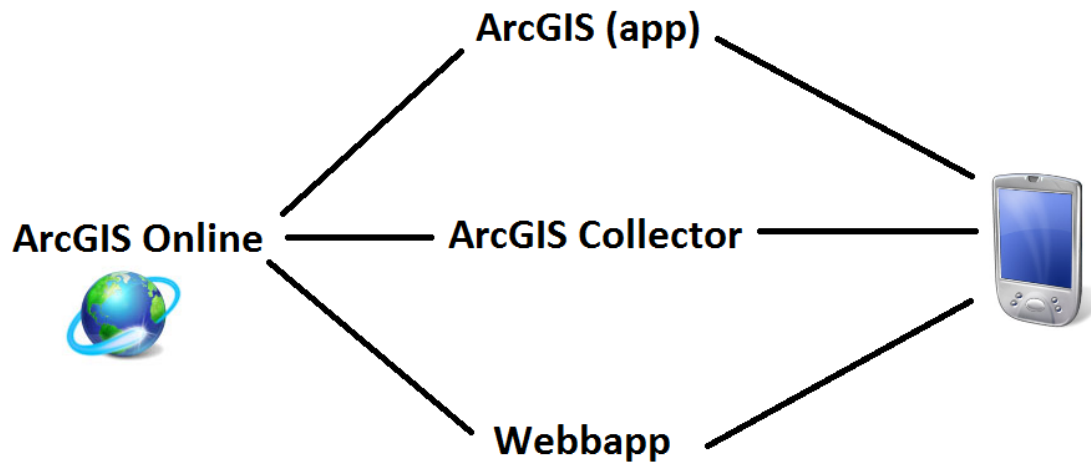
Säkerhet

Kalmar kommun har en kommunlicens på ArcGIS, någon som ger dem tillgång till en egen *Organisation* på AGO. Kommunens anställda kan skapa egna användare i *organisationen*, med ett eget användarnamn och lösenord. En administratör för organisationen kan även skapa olika roller som man sedan kan tilldela användarna, vilket avgör användarnas rättigheter. Vissa får endast läsrättigheter medan andra får publiceringsrättigheter (att publicera nya och ändra befintliga kartor).

För varje karta kan man bestämma vilka den ska delas med, här är det lämpligt att man bara delar den med sin organisation. Andra alternativ är att man håller den privat (endast för ägaren), eller att man tvärtom delar den med alla (kräver inget AGO-konto).

9.4.2. Åtkomst

Informationen från AGO når man antingen via en färdig lösning som ArcGIS (app) och ArcGIS Collector, eller via en webbapp. Det krävs att användaren har ett AGO-konto inom kommunens organisation för att den ska få tillgång till informationen.



Figur 9.2: Lösning 1 – ArcGIS Online

ArcGIS (app)

Denna app är tänkt att främst användas med AGO och det fungerar därmed väldigt smidigt. Första gången användaren startar appen får han eller hon logga in med sitt AGO-konto, användarnamn och lösenord sparas sedan. När användaren väl är inloggad har den tillgång till alla kartor som den har tillgång till i datorn.

ArcGIS Collector

Även denna app är tänkt att användas med AGO och fungerar därmed analogt med ArcGIS (app).

Webbapp

Det är möjligt att skapa sin egen webbapp som hämtar information från AGO, men det enklaste är att generera en automatiskt. Detta görs via **Dela → Skapa en webbapplikation**, därefter väljer man en av de färdiga mallarna. Nackdelen är att man har väldigt liten inverkan på den färdiga webbappen då mallen bestämmer allt. Fördelen är att det går väldigt snabbt och att webbappen sparas på AGO (man slipper publicera den via en egen hemsida).

9.4.3. Implementation

Första steget i implementeringen av denna lösning var att skapa en *Feature service*. Denna tjänst skapades på samma sätt som beskrivs närmre under rubriken *Editeringsmöjligheter*. Efter detta skapades en ny karta i AGO, med en färdig bakgrundskarta från AGO. Editeringstjänsten infogades sedan i kartan. För att testa möjligheten att länka till stomnätsskisser ändrades inställningarna för det editerbara lagret via menyn "*Konfigurera popup-ruta*", detta beskrivs under rubriken "*Stomnätsskisser*". Sist kontrollerades säkerhetsinställningarna, kartan valdes att bara delas med *organisationen*.

När kartan var redo på serversidan påbörjades testningen av de olika klienterna. Testkartan kunde enkelt nås via ArcGIS (app) och ArcGIS Collector genom att man loggade in med ett AGO-konto. För testande av webbappen användes en automatiskt genererad webbapp. Olika punkter skapades, togs bort och ändrades via de olika klienterna. Det testades även hur de hanterade att bifoga bilder till de olika punkterna. Även andra färdiga verktyg testades, som exempelvis att visa användarens position på kartan och att få vägbeskrivning.

9.5. Lösning 2 – ArcGIS Server

Till skillnad från lösning 1 använder denna lösning endast geografisk information från ens egna nätverk. Man använder sig av ArcGIS Server för att starta en tjänst som sedan tillhandahåller den geografiska informationen. Den geografiska information som ska spridas behöver inte befinna sig på samma dator som ArcGIS Server-tjänsten, men serverdatorn måste ha tillgång till den. För större GIS-system är det vanligt att informationen befinner sig i på mer än en dator, och i mer än en databas.

Denna lösning har flera fördelar i jämförelse med lösning 1, t.ex. att:

- Man inte skapar en kopia av sin information, dvs. laddar upp den på AGO.
- Alla ändringar som man genomför i sin app uppdateras direkt i databasen (eftersom man inte jobbar mot en kopia).
- Man får enklare integration med ArcGIS Desktop.
- All information (t.ex. bilder) som laddas upp av användarna hamnar på ens egen server.
- Man får en kraftfullare slutprodukt.

Nackdelen med denna lösning gentemot lösning 1 är att den är "dyrare" att utveckla och driva. Detta innebär t.ex. att:

- Man behöver införskaffa en ArcGIS Server licens om man inte redan äger en. Detta anses dock inte vara en stor nackdel då AGO inte heller är gratis.
- Det kräver att man har en egen serverdator. Om man inte redan har en existerande server måste man köpa in detta. Ett annat problem är de mantimmar som krävs för

att konfigurera servern. Detta gäller även om det är en existerande server. Man måste ge servern tillgång till den information och de programvaror som krävs, men man måste även se till att omvärlden har möjlighet att nå servern. Servern kräver även regelbundet underhåll.

9.5.1. Licens

ArcGIS Server finns i tre olika licenser: *Basic*, *Standard* och *Advanced*. *Basic* är den version som har minst antal verktyg och funktioner, medan *Advanced* är den som har högst. För att genomföra webb- och mobileditering krävs en *Standard* eller *Advanced* licens.

9.5.2. Databas

Vid test i detta projekt har informationen lagrats i en geodatabas, skapad m.h.a. ArcGIS Desktop och Microsoft SQL Server. Microsoft SQL Server valdes eftersom det var relativt lättinstallerat, någon djupare studie av olika DBMS har inte genomförts.

Geodatabas

Geodatabas är en spatial databas som är skapad (och använd) av ArcGIS. Geodatabasen finns i tre olika typer *File Geodatabase*, *Personal Geodatabase* och *ArcSDE Geodatabase*. Vissa ArcGIS-verktyg kräver att den information som man bearbetar befinner sig i en geodatabas. Detta medför att man t.ex. kan bestämma regler och relationer för sin information. *ArcSDE* är den mest kraftfulla av de tre, men ofta är någon av de andra två tillräcklig.

Det är främst *ArcSDE Geodatabase* som har använts i detta projekt, då det är denna typ av databas som använder sig av ett RDBMS (Relational database management system). De andra två typerna är klassiska fildatabaser och kan därför bara användas av en användare åt gången. De RDBMS som ArcGIS stödjer är DB2, Informix, Oracle, SQL Server, PostgreSQL och SQL Server Express. (*ArcGIS, 2013h*)

9.5.3. Funktionalitet

Bakgrundskarta

Bakgrundskartan för tjänsten bör publiceras som en egen kartservice via ArcGIS Server. Detta är enkelt gjort då Kalmar kommuns geografiska information redan är väl integrerad med ArcGIS. Kalmar kommun har ett antal olika "mallar" (lyr-filer) för att visualisera deras geografiska information. När man väl har valt karta kan man enkelt sprida den m.h.a. ArcGIS Desktop.

Den geografiska informationen för bakgrundskartan behöver inte vara lagrad i en ArcSDE databas, då den ej ska editeras.

Editeringsmöjligheter

De editerbara lagren ska publiceras separat från bakgrundskartan, när man ska publicera sin karta lägger man sedan ihop dem. Samtliga lager som ska publiceras som editerbara måste befinna sig i en (och samma) spatiala databas.

Att publicera en karttjänst med editeringsmöjligheter via ArcGIS Desktop går till på samma sätt som för bakgrundskartan, skillnaden är att man väljer till *Feature access*. Man ger då tjänsten tillgång till lagrets egenskaper (features).

Stomnätsskisser

Åtkomsten till stomnätsskisserna går att lösa på flera olika sätt, då detta är en lösning som man till stor del måste själv utveckla. Det går därför att bestämma exakt vilken information som ska visas och hur. En enkel lösning är att man skriver ett skript som genererar och visar *HTML*-kod. Detta sker för varje punkt som användaren klickar på, koden infogas sedan i ett tomt fönster och visas för användaren. Detta leder till att man kan blanda t.ex. text, bilder och länkar hur man önskar. Om man använder sig av en webblösning finns det även tilläggsprogram eller insticksmoduler som man kan använda för att enklare visa sina bilder.

Bilderna måste vara publicerade på internet för att nås utanför kommunens intranät (precis som för lösning 1).

Säkerhet

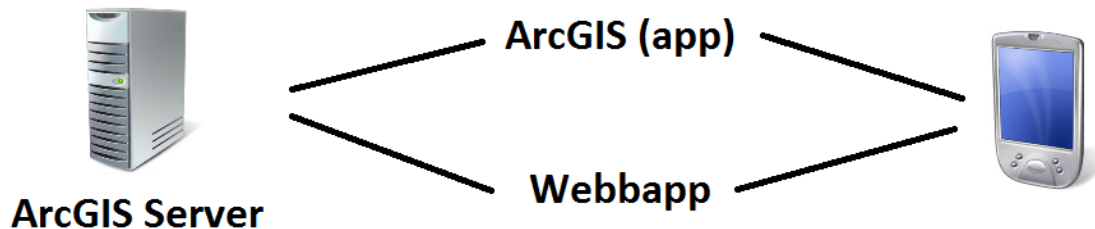
I ArcGIS Server kan man bestämma vilka tjänster som ska vara lösenordsskyddade och vilka som ska vara öppna för alla. Med lösenordsskyddat menas att användaren måste logga in med ett användarnamn och ett lösenord. Det går att skapa flera användare, alla kan ha varsin inloggning. Varje användare kan sedan även tilldelas olika roller, och därmed få olika rättigheter. Vissa användare kan då t.ex. bara läsa information, medan andra kan skriva information till servern och publicera tjänster.

Det går även att använda kommunens befintliga inloggningsserver för att ge åtkomst till kartan, i ArcGIS Server kan man då tilldela en ny roll för de som man vill ska få nå kartan. Ett exempel på detta är att skapa en ny roll som heter "Mätavdelningen" och tilldela den till alla användare i avdelningen, sedan ställer man in i servern att endast de med rollen "Mätavdelningen" ska få tillgång till kartan. Användarna kan då enkelt använda samma användarnamn och lösenord som till resten av kommunens system och man slipper skapa nya användare.

Om man väljer att nå sin information via en webbapp på klientsidan kan man även skriva sitt eget säkerhetssystem, eller installera ett färdigt.

9.5.4. Åtkomst

Ansiktet utåt är den server som kör ArcGIS Server och det är (bara) den som användaren behöver koppla upp sig mot. Serverdatorn kommer sedan i sin tur att hämta information från andra datorer i nätverket, men det är inget som användaren behöver tänka på.



Figur 9.3: Lösning 2 – ArcGIS Server

ArcGIS (app)

När man startar en server med ArcGIS Server startas även en mobiltjänst, som man kan ansluta sig till via ArcGIS (app). En karta (eller annan tjänst) som publiceras på servern blir dock inte automatiskt tillgänglig för ArcGIS (app), för att ta del av dessa tjänster måste man starta en ny mobiltjänst specifikt för detta. Detta görs dock inte på samma sätt som för normala tjänster, utan man använder sig av språket *JSON*. Man använder *JSON*-kod för att beskriva den nya tjänsten: vad som ska ingå, hur det ska visas och vad man kan göra.

JSON

JSON (JavaScript Object Notation) är ett språk som används för att "transportera" data, t.ex. mellan en server och en webbapp. Informationen omvandlas till *JSON* på serversidan och skickas sedan till klienten där den packas upp. Detta så att servern och klienten kan kommunicera på ett enhetligt sätt. *JSON* är ursprungligen baserat på *JavaScript*, men det kan ändå användas av andra språk. Det är designat så att det är lättförståeligt för både människor och datorer. (*JSON, 2013*)

Här är ett exempel på en enkel tjänst, den innehåller bara en bakgrundskarta och ett editerbart lager. De två tjänsterna är publicerade tjänster från ArcGIS Server.

```
{
  "version": "1.1",
  "operationalLayers": [
    {
      "url": "http://192.168.1.66:6080/arcgis/rest/services/Upload/FeatureServer/0",
      "visibility": true,
      "opacity": 1,
      "title": "Edit"
    }
  ],
}
```

```
"baseMap":{
"baseMapLayers":[
{
"url":"http://192.168.1.66:6080/arcgis/rest/services/Kalmarkarta3/MapServer
"
}
],
"title":"Kalmar"
}
}
```

Webbapp

Användaren når informationen via en hemsida, man slipper installera ett program. En sådan hemsida kan ändå vara relativt kraftfull, då de använder sig av avancerade skript. De skript som har använts i detta projekt är av typen *JavaScript*, *jQuery* och *Dojo*.

En webbapp är den mest kraftfulla lösningen, då man bygger sitt system från grunden. Men det är även den mest krävande att utveckla.

9.5.5. Implementation

För att publicera tjänster till ArcGIS Server användes ArcGIS Desktop. För att starta, stoppa eller ändra de publicerade tjänsterna användes *ArcGIS Server Manager*. Bakgrundskartan, det editerbara lagret och stornätspunkterna publiceras som tre olika tjänster. Bakgrundskartan och stornätspunkterna publiceras som vanliga karttjänster. För att skapa det editerbara lagret behövs en spatial databas (som har diskuterats under rubriken *”Editeringsmöjligheter”*). För test i projektet användes en ArcSDE databas, den skapades m.h.a. ArcGIS Desktop i en Microsoft SQL databas. Ett tomt lager placerades sedan i databasen och blev grund för den nya editeringstjänsten.

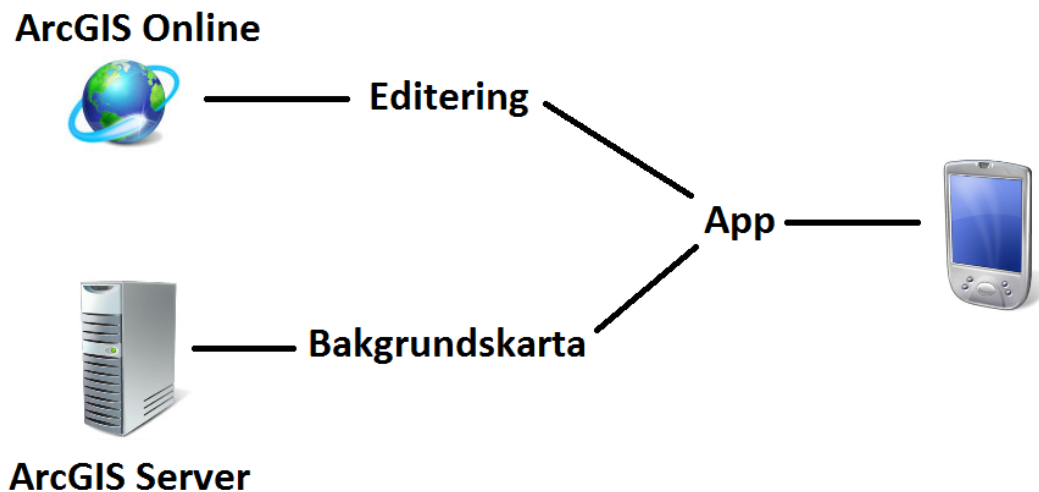
När all information fanns redo på servern påbörjades implementationen av klienterna, först var ArcGIS (app). Den har möjlighet att koppla upp sig direkt mot ArcGIS Server, men för att kartan ska bli tillgänglig måste den publiceras separat för mobilt bruk. *JSON*-kod utvecklades som beskrev den nya tjänsten. Efter ArcGIS (app) undersöktes möjligheten att använda en webbapp som klient. Den webbapp som användes vid testande utvecklades av författaren och var baserad på de exempel som finns på ArcGISs hemsida. Webbappen är främst uppbyggd med *JavaScript*, *Dojo* och *jQuery*.

När de båda klienterna var installerade påbörjades testfasen, den följde samma mönster som för lösning 1. Det testades t.ex. hur väl de hanterade att skapa, ta bort och ändra punkter samt bifoga bilder.

9.6. Lösning 3 – Hybriden

Lösningen använder både AGO och ArcGIS Server för att sprida informationen. Fördelen är att de två lösningarna kan komplettera varandra, då de har olika för- och nackdelar kan de

täcka upp för varandra. Som exempel på detta kan vi ta lösning 1 – ensam uppfyller den inte alla de uppställda kraven, men om man komplementerar den med en bakgrundskarta från ArcGIS Server gör den det.



Figur 9.4: Exempel på en hybridlösning

Figur 9.4 visar hur arbetet kan fördelas mellan de två servrarna, observera att detta bara är **en av många** olika varianter. Informationen behöver inte delas mellan de två precis som i bilden. Det kan t.o.m. vara så att all information sprids från ArcGIS Server. Med detta menas att all information *ursprungligen* sprids från ArcGIS Server, men att den sedan passerar AGO.

Fördelen med att sprida sin karta via AGO är att den bli mer lättillgänglig än om man använt ArcGIS Server. Det är dock inte önskvärt att man ska lagra all sin information på AGO. Men det finns möjlighet att skapa kartor på AGO med karttjänster från ArcGIS Server. Man kan starta en tjänst via ArcGIS Server som tillhandahåller en karta, den tjänsten infogassedan i en karta på AGO. Kartan på AGO får då tillgång till informationen från ArcGIS Server trots att den inte lagras på AGO.

9.6.1. Funktionalitet

Funktionaliteten för systemet är densamma som för de övriga två lösningarna, på varje punkt kan man välja mellan antingen lösning 1 eller lösning 2.

Bakgrundskarta

Om man har en befintlig karta som man vill använda som bakgrundskarta gör man detta via ArcGIS Server. Om man inte har en bakgrundskarta har AGO flera baskartor att välja mellan.

Editeringsmöjligheter

För att få editeringsmöjligheter behöver man en *Feature service*. Detta löser man enklast m.h.a. AGO, då det inte kräver en egen server. Nackdelen med detta är att allt material som användarna lägger till sparas på AGO. Det går att exportera data från AGO, men bara till viss del. Det går exempelvis inte att exportera bifogade filer.

Att starta en *Feature service* m.h.a. ArcGIS Server kräver mer arbete än för AGO, men fördelen är att man slipper betala krediter för sin tjänst. ArcGIS Server rekommenderas därför för större projekt, där det lönar sig att lägga extra tid i början av projektet (för att spara krediter).

Stomnätsskisser

Hur man når stomnätsskisserna beror på hur åtkomsten till systemet sker. Om man sprider sin information via AGO når man skisserna på samma sätt som i lösning 1, och om man använder sig av ArcGIS Server sker detta i enlighet med lösning 2.

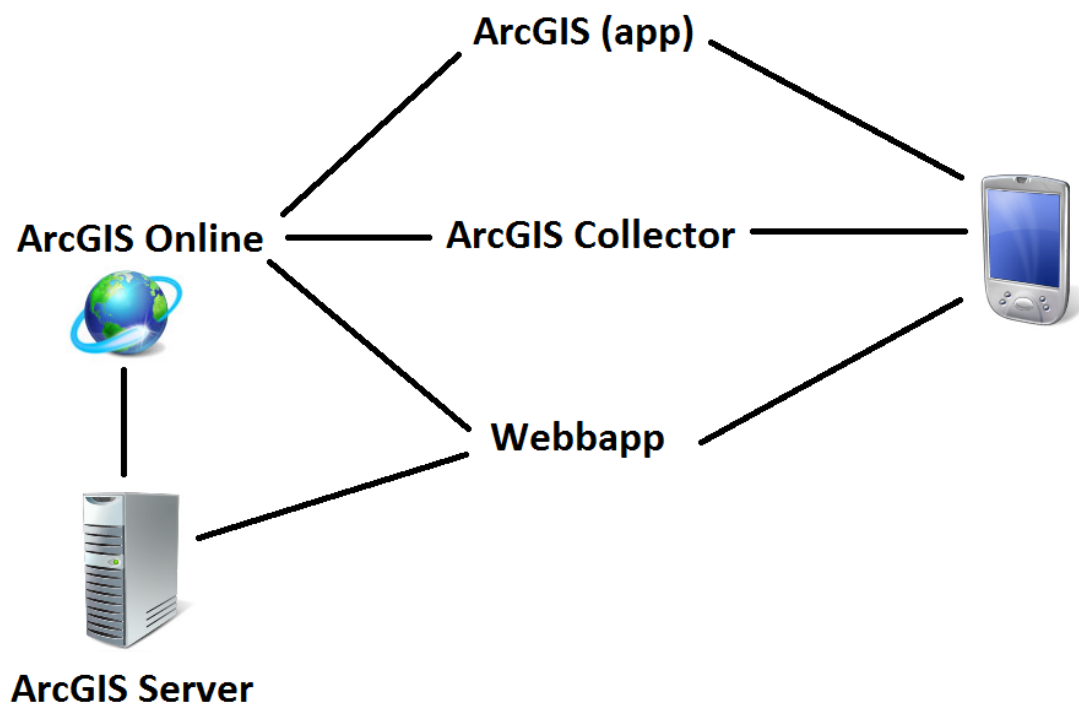
Säkerhet

Det går att bestämma i både AGO och ArcGIS Server vem som ska ha rätt att ta del av vilken information. För mer om detta läs punkten om *Säkerhet* för lösning 1 och lösning 2.

9.6.2. Åtkomst

Figur 9.5 visar hur åtkomsten till system sker, den har mycket gemensamt med åtkomsten för lösning 1 (figur 9.2). Faktum är att åtkomsten sker analogt för ArcGIS (app) och ArcGIS Collector. Skillnaden mellan de två lösningarna är att man hämtar kompletterande information från ArcGIS Server. Detta påverkar inte hur man når information.

Kommunikationen mellan AGO, ArcGIS Server och andra eventuella databaser behöver inte användaren bry sig om.



Figur 9.5: Lösning 3 – Hybriden

En webbapp behöver inte "koppla upp" sig mot en server som de andra två apparna. När man använder en webbapp kan man därför handplocka den information som man behöver och infoga den, oberoende av om den ursprungligen kommer från AGO eller ArcGIS Server.

För mer information om hur åtkomsten till AGO sker, läs punkten om *Åtkomst* för lösning 1.

Via ArcGIS Server

Man kan även byta plats på AGO och ArcGIS Server. Detta betyder att man ansluter till ArcGIS Server och sedan hämtar kompletterande information från AGO. Detta anses dock vara en sämre lösning i de flesta fall. Enda situationen där författaren tror att detta kan vara att föredra är när man har all sin information via ArcGIS Server (via ArcGIS (app)) och man behöver ett editerbart lager från AGO, som redan editeras av andra AGO-tjänster.

9.6.3. Implementation

Som grund för test av denna lösning användes en ny karta på AGO, i den infogades bakgrundskarta och ett editerbart lager från ArcGIS Server (samma två tjänster som används vid test av lösning 2). Resultatet är en AGO-karta som hämtar all sin information från ArcGIS Server. De tre klienterna genomgick sedan samma testningsprocedurer som tidigare.

10. Funktionalitet

Medan förra kapitlet var fokuserat på tekniken bakom handlar detta kapitel om vad den framtida användaren kommer att se och göra. De tre klientlösningar som har valts är ArcGIS (app), ArcGIS Collector och en webbapp.

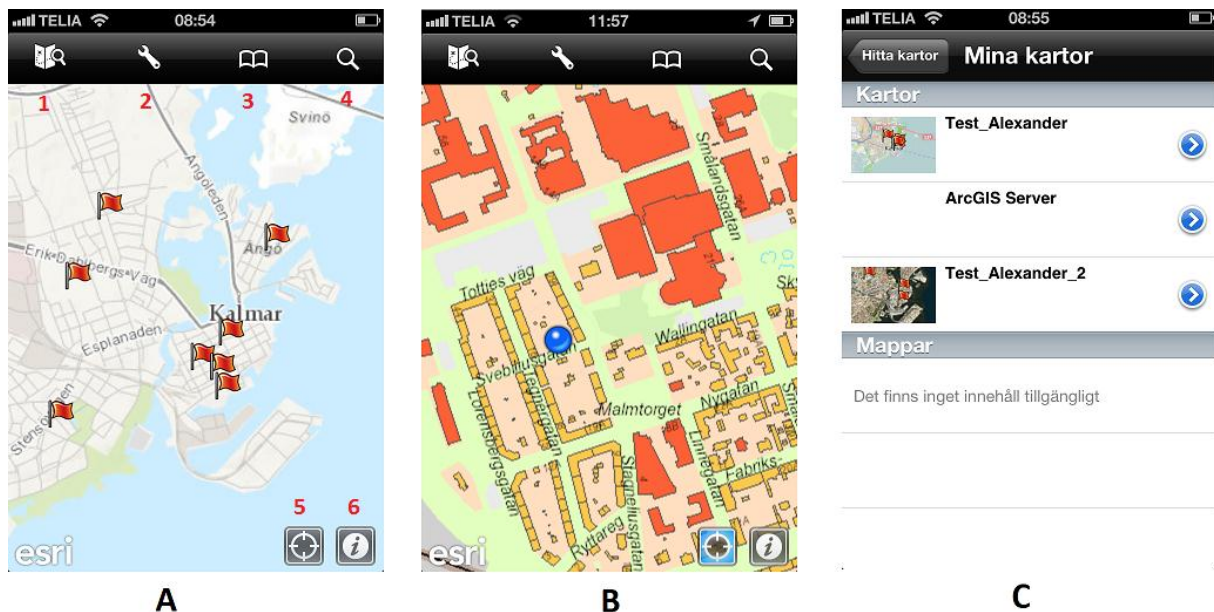
10.1. ArcGIS (app)

Denna app har först och främst valts eftersom den uppfyller de uppställda kraven väl (kravspecifikationen). Appen är även lätt att använda tillsammans med andra ArcGIS-produkter, då de tillhör samma familj.

Det faktum att appen tillhör ArcGIS-familjen underlättar integrationen med andra ArcGIS-produkter, men det ökar även tryggheten och säkerheten. Med trygghet menas att man vet att det är ett företag som står bakom appen, och att den kommer att vidareutvecklas. Annars kanske systemet slutar att fungera en dag. Med säkerhet menas att man vet att informationen är säker. Ska man lämna ut användarnamn och lösenord för sin data måste man kunna lita på sin programvara.

Åtkomsten till den geografiska informationen är först och främst tänkt att nås via AGO, men som nämnts tidigare stödjer även appen att man ansluter direkt mot ArcGIS Server. Det enklaste sättet är dock via AGO.

10.1.1. Användargränssnitt



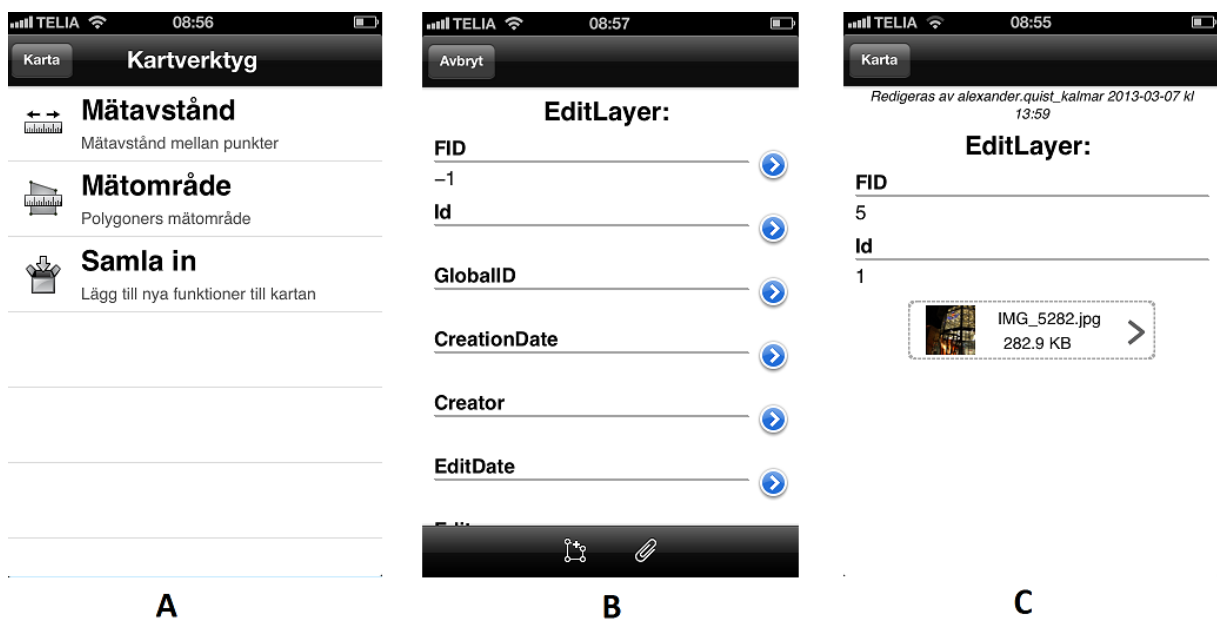
Figur 10.1: ArcGIS (app) – Översikt

A. En karta som använder sig av lösning 1 (avsnitt 9.4).

Knappar:

1. Visar dina kartor, se **C**.
2. Verktyg, se **A** i nästa bild (figur 10.2) – möjligheten att mäta avstånd och ytor, samt skapa punkter (om editeringsmöjligheter finns).
3. Bokmärken.
4. Sök.
5. Visa användarens position på kartan.
6. Visar information om kartan, men innehåller även inställning som t.ex. byte av bakgrundskarta.

- B.** En karta som använder sig av lösning 2, det är m.a.o. Kalmar kommuns egen geografiska information. Man kan även se att användarens position är markerad på kartan.
- C.** Visar vilka kartor som användaren har tillgång till, i detta fall är appen uppkopplad mot AGO.



Figur 10.2: ArcGIS (app) – Editering

- A.** De tillgängliga verktygen, man kan se att kartan har editeringsmöjligheter då vi kan välja att lägga till funktioner (dvs. punkter).
- B.** Menyn för att skapa nya punkter. Man kan se de informationsfält som tillhör punkten, dock är endast "Id" editerbart i detta exempel. De två knapparna i botten används för att välja punktens position samt för att bifoga filer.

- C. Informationen för en punkt som användaren har skapat. Vi kan även se att det finns en bifogad bild. Användaren kan välja att ladda ner bilden genom att klicka på den.

10.1.2. Verktyg

Vilka verktyg som är tillgängliga i appen beror på vad man har ställt in på serversidan. Det går att stänga av verktyg som man inte vill göra tillgängligt för användaren. Fördelen med detta är t.ex. att en användare inte ska ha tillgång till fler verktyg än han eller hon behöver (för att inte skapa förvirring), men det är också en kostnadsfråga då vissa verktyg kostar krediter att använda (om man sprider sin information via AGO).

Exempel på verktyg är:

- mätverktyg
- byte av bakgrundskarta
- skapa, uppdatera och radera punkter
- bifoga filer till punkterna
- skapa och använda bokmärken
- sökverktyg
- visa användarens position
- visa eller dölja lager
- visa information om de lager som finns i kartan

10.1.3. Implementerat

ArcGIS (app) har testats med alla de tre föreslagna serverlösningarna, för mer information om detta se kapitel 9. Detta innebär att den har testats direkt mot både AGO och ArcGIS Server. Vid test av systemet testades främst hur den uppfyller de uppställda kraven, exempelvis hur den skapade punkter och hanterade bifogade bilder. Alla verktygen testades, även sådana som inte är relevanta för uppgiften (t.ex. byte av bakgrundskarta och mätverktyg). Allt för att skapa sig en bra bild över hur programmet i stort fungerar. Även det estetiska vägdes in, hur bra ser användargränssnittet ut och hur lätt är det att arbeta med.

10.2. ArcGIS Collector

Denna app har i stort sett samma funktionalitet som ArcGIS (app), fast de har olika användargränssnitt. Anledningen till varför denna app har valts som lösning är därför samma som för ArcGIS (app), d.v.s. att:

- Den uppfyller de uppställda kraven.
- Den är lättanvänd.
- Den är väl integrerad med AGO.

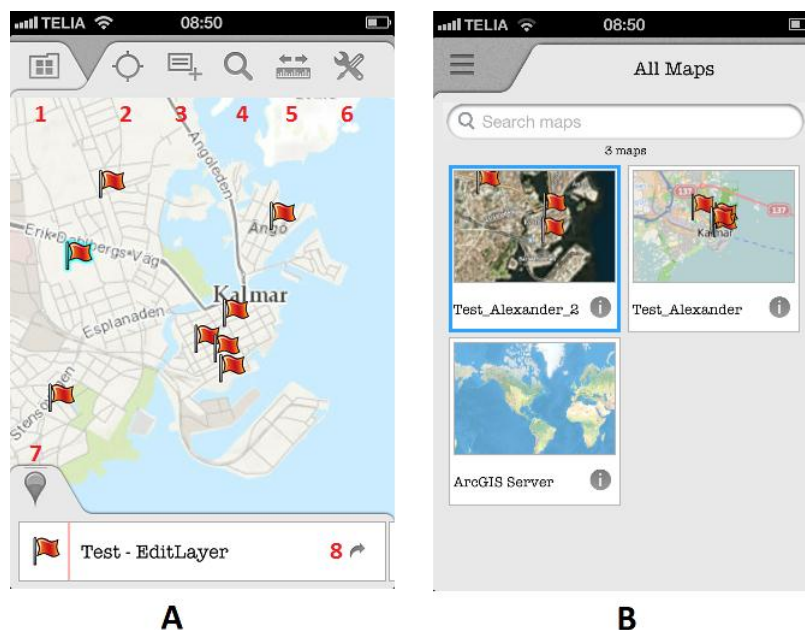
- Den har bra säkerhet, då den tillhör samma familj som serverlösningarna (ArcGIS).

Två andra fördelar med denna app är att:

- Den är utvecklad för att samla in data i fält.
- Den har fullt stöd för funktionen "Editor tracking". Editor tracking innebär att systemet bokför vem som har ändrat informationen och när.

ArcGIS Collector kan endast hämta information från AGO. Om man har information som man vill sprida från ArcGIS Server måste detta ske via en karta på AGO.

10.2.1. Användargränssnitt



Figur 10.3: ArcGIS Collector – Översikt

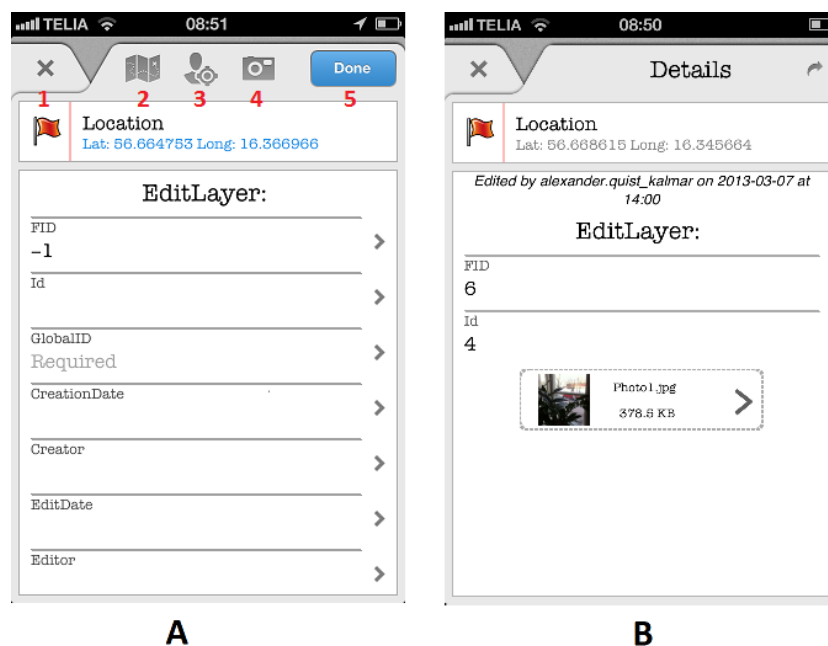
A. En karta som använder sig av lösning 1 (från förra kapitlet).

Knappar:

1. Visar dina kartor, se B.
2. Visa användarens position på kartan.
3. Skapar en ny punkt i kartan.
4. Sök
5. Mätverktyg
6. Visar dolda verktyg, i vårt fall innebär det verktygen byta bakgrundskarta samt visa information om kartan.
7. Används för att dölja informationsraden i botten.

8. Visar ytterligare information om punkten.

- B.** Visar vilka kartor som användaren har tillgång till. I bilden är appen uppkopplad mot AGO.



Figur 10.4: ArcGIS Collector – Editering

- A.** Menyn för att skapa en ny punkt. Vi kan se de informationsfält som tillhör punkten, endast "Id" är editerbart i detta exempel.

Knappar:

1. Avbryt
2. Välj vart på kartan punkten ska placeras.
3. Placera punkten på användarens nuvarande position.
4. Bifoga en bild.
5. Skapa punkten.

- B.** Information om en punkt som användaren har skapat. Vi kan även se att det finns en bifogad bild. Användaren kan välja att ladda ner bilden genom att klicka på den.

10.2.2. Verktyg

Verktygen för denna app är i stort sett identiska med de för ArcGIS (app), även här är det serversidan som bestämmer vilka verktyg som finns tillgängliga (se **10.1.2**). Förutom de verktyg som nämnts för ArcGIS (app) har även denna app möjligheten att ge vägbeskrivning mellan två punkter.

10.2.3. Implementerat

ArcGIS Collector har testats mot lösning 1 och 3, av de (i kapitel 9) föreslagna serverlösningarna. Själva testningen gick till på samma sätt som för ArcGIS (app), som diskuterats i avsnitt **10.1.3**.

10.3. Webbapp

Både för- och nackdelen med en webbapp är att man måste utveckla den själv. Nackdelen är att det tar tid och kräver viss expertis. Fördelen är att man kan utforma sitt system precis efter behov.

Eftersom webbappen i grund och botten är en hemsida behöver inte användaren installera ett program för att använda tjänsten. Vilket leder till att systemet blir kompatibelt med flera operativsystem. Åtkomsten av information för webbappen fungerar ungefär som den för de andra lösningarna. När man använder sig av de två färdiga apparna hämtar man "färdiga" kartor som är skapade på servern. Men när man använder sig av en webbapp är det i klienten (webbappen) som de olika tjänsterna svetsas samman till en karta.

10.3.1. Automatiskt genererad

Det är på AGO möjligt att skapa webbappar för sina kartor. Man har ett antal mallar att välja mellan för att styra hur ens webbapp blir i slutänden. Nackdelen med detta är dock att det bara finns en mall som är anpassad för smartmobiler och läsplattor. Det är därför att rekommendera att man gör sin egen webbapp om man ska använda den med smartmobiler.

10.3.2. Att skapa en egen

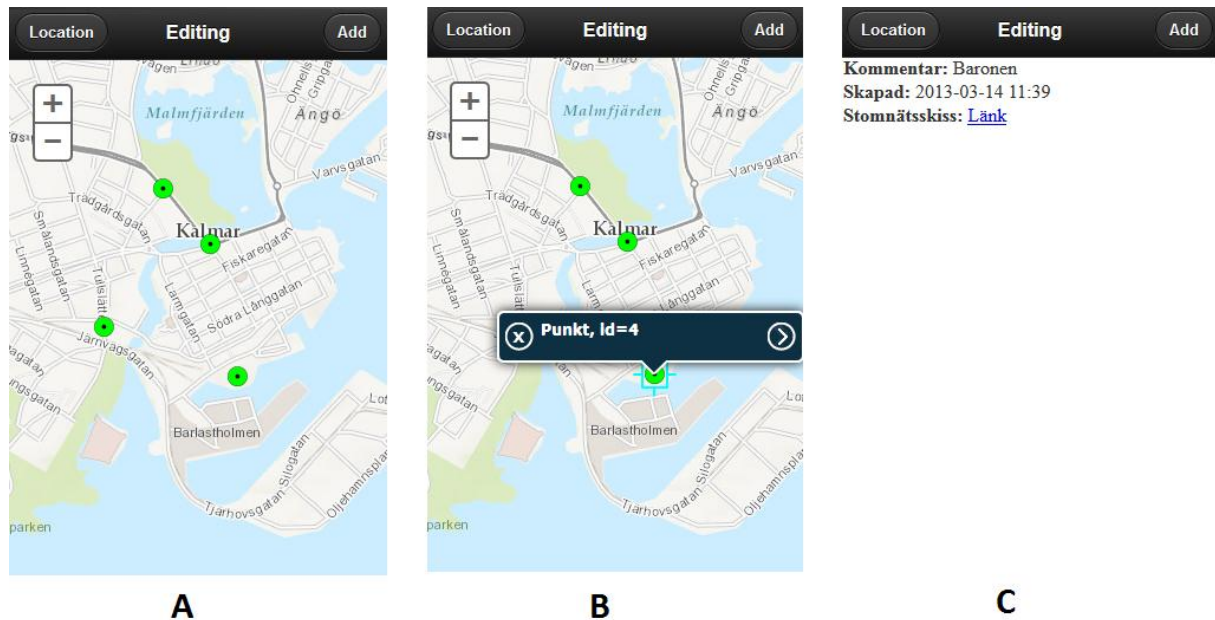
Det finns många olika sätt för att skapa en webbapp, man kan använda sig av många olika språk och tillägsprogram/extra verktygslådor. I detta projekt har det skapats en webbapp baserad på *HTML*, *JavaScript*, *Dojo* och *jQuery*. Den webbapp som har skapats är baserad på några av de exempel som finns på ArcGIS hemsida. Deras kod har kombinerats och sedan modifierats för att passa just detta ändamål.

Den app som har skapats uppfyller alla de uppställda kraven. Med viss reservation för krav **2.1**, då den inte har möjlighet att bifoga bilder till de skapade punkterna. Det är något som är

fullt möjligt att genomföra, men problemet hann inte lösas inom projektets tidsram.

10.3.3. Användargränssnitt

Figur 10.5 visar en bild på den webbapp som har utvecklats i projektet.



Figur 10.5: Den utvecklade webbappen

- Så här ser webbappen ut när den startar. Bakgrundskartan är från AGO, punkterna i kartan är från ArcGIS Server. *Location* zoomar in på användarens position och *Add* låter användaren skapa en ny punkt.
- Här har användaren klickat på en av punkterna i kartan, har man valet mellan att avmarkera punkten eller se ytterligare information (se C). Texten har valts till "Punkt, id=4" (där fyran är ett dynamiskt tal), detta är lätt att ändra till vad man önskar.
- Detta är den information som visas när användaren klickar på ">" i föregående ruta.

10.3.4. Verktyg

Appen innehåller verktyg för att:

- Skapa punkter.
- Ändra en punkts attribut.
- Se information om existerande punkter.
- Zooma till användarens position på kartan.
- Enkelt länka till stomnätsskisser.

10.3.5. Implementerat

Den webbapp som har implementerats är kompatibel med alla de 3 lösningarna från kapitel 9, den är skapad m.h.a. HTML, JavaScript, jQuery och Dojo. Huvuddelen av koden till webbappen finns i bilaga A, undantaget är de externa skript som (importeras och) används.

Appen är skapad m.h.a. en textredigerare och är främst baserad på olika exempel från ESRI:s hemsida, resterande delar av koden är antingen skriven av författaren eller hämtad från andra hemsidor. All kod är hämtad från hemsidor som tillåter att man återanvänder deras exempel (dvs. deras kod). Som grund för appen användes ett exempel från ESRI, den innehöll de grundläggande funktionerna som t.ex. att använda en bakgrundskarta och att skapa punkter. Appen kodades sedan om till att använda en bakgrundskarta och ett editerbart lager från den egna serverdatorn, som körde ArcGIS Server. När användaren skapade punkter fanns det möjlighet att lägga till attribut, men det gick inte att läsa eller ändra dem när punkten väl var skapad. Nästa steg blev därför att lösa detta problem. Eftersom flertalet av de använda lösningarna inte var anpassade för mobilt bruk gick det åt mycket tid för att anpassa dem till smartmobiler (främst för den lilla skärmen).

Mycket tid under utvecklingsfasen gick även åt till att leta och studera liknande lösningar, exempelvis de som redovisats i kapitel 5. Först studeras källkoden för att se vilka olika tekniker som använts (t.ex. *jQuery* eller *Dojo*), sedan studerades även *hur* man ska använda dem. De olika lösningarna har alla egna *API* som dikterar hur de ska användas.

Bottenplattan i webbappen är (som för majoriteten av alla hemsidor) skriven i *HTML*. *HTML*-koden är indelad i två olika delar: *head* och *body*. *Head* innehåller information som exempelvis hemsidans titel, men det innehåller även de skript som används. *Body* innehåller främst hemsidans grafiska utseende, t.ex. var knapparna och kartan ska placeras.

Webbappens *body* innehåller tre olika utseenden som den växlar mellan, dessa tre används när man:

- Tittar på kartan.
- Skapar/uppdaterar en punkt.
- Inspekterar en punkt.

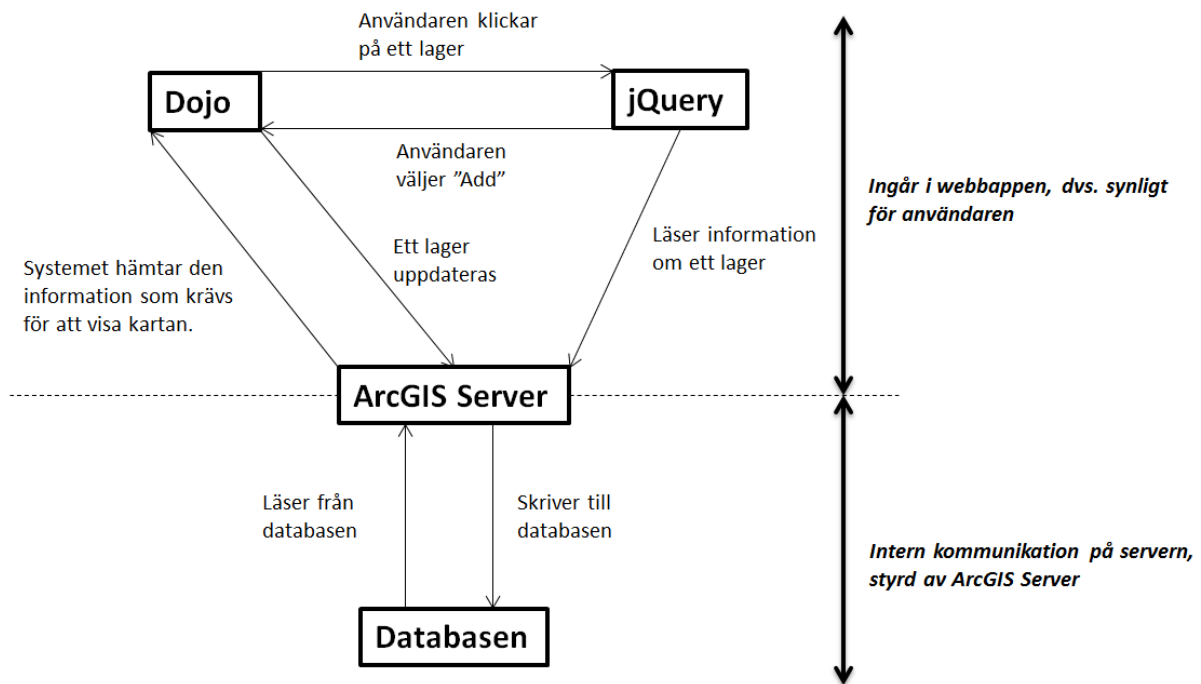
JavaScript är det skriptspråk som används för att driva webbappen (skripten finns i webbappens *head*). Vissa skript körs direkt när webbappen laddas och vissa körs vid utvalda tillfällen. Detta kan t.ex. vara när användaren trycker på en knapp eller när systemet gör något av sig själv, t.ex. när ett lager läggs till i kartan.

Dojo är det som främst används för att bearbeta den geografiska informationen. Det gör det möjligt att skapa en karta där man kan infoga sin information men även redigera den. Både *ArcGIS Online* och *Server* stödjer användning av *Dojo*.

jQuery mobile används för att utföra två uppgifter, dels att handha användargränssnittet samt dels att hämta information från databasen. I *HTML*-koden har det specificerats vilket

område som är t.ex. menyrad, men det är *jQuery* som ger den det faktiskt utseendet. *jQuery* styr utseendet på knappar och menyrader som är utanför själva kartan, i kartan används *Dojo* för detta. Detta betyder exempelvis att utseendet för formulären för att skapa en ny punkt styrs av *jQuery*, men det gäller exempelvis inte för hur en markerad punkt i kartan ser ut. Som nämndes i början av stycket används även *jQuery* för att hämta information från databasen. Detta används när användaren klickar på en punkt i kartan, då hämtas information om exakt den punkten som sedan kan visas för användaren. Informationen som visas i figur 10.5.C är skapad med ett *jQuery*-skript som genererar HTML-kod, informationen som visas är skraddarsydd.

Figur 10.6 visar förenklat hur kommunikationen sker i webbappen. ArcGIS Server utför majoriten av arbetet med att visualisera informationen, dvs. läsa från databasen och sedan omvandla det till en karta. Det är även ArcGIS Server som skriver till databasen när den får information från Dojo att skapa eller uppdatera en punkt.



Figur 10.6: Kommunikationsskiss webbappen

Webbappen uppfyller alla användarkraven (avsnitt 4.1). Dock uppfylls krav 2.1 inte fullständigt, då systemet ej kan hantera bifogade filer. Den kan varken bifoga eller visa tidigare bifogade filer. Dessa funktioner är tekniskt genomförbara men rymdes inte inom projektets tidsram.

Appen har testats regelbundet under utvecklingsfasen.

11. Val av tekniska lösningar

Detta kapitel kommer att diskutera vilka tekniska val som gjorts och varför. Kapitlet är uppdelat så att det först diskuteras lösningarna för serversidan och därefter klientsidan. Sist diskuteras den slutgiltiga lösningen.

11.1. Lösningar – Serversidan

I kapitel 9 har tre olika lösningar diskuterats för serversidan. I detta kapitel kommer det diskuteras djupare när man ska använda dem. Det gäller vilken lösning som är optimal för en specifik situation dock har även kostandsfrågan vägts in. Kostnadsfrågan är en avvägning mellan driftskostnad och utvecklingskostnad.

11.1.1. Lösning 1 – ArcGIS Online

ArcGIS är den lösning som är "enklast". Lösningen kräver ingen GIS-mjukvara samt så kräver den minst expertis hos utvecklaren. Det som krävs är ett AGO-konto, samt ett (tomt) lager som användaren kan ha som grund för sitt editerbara lager. Utvecklingskostnaden blir därmed nästan obefintlig för denna typ av system. Driftskostnaden kan dock bli hög för stora system, då användning av tjänster på AGO kostar krediter. Fördelen är dock att Kalmar kommun i dag har tillgång till cirka 37 000 krediter (*Bernhard, 2013*), betalt av deras kommunlicens. Dessa krediter kommer i sin tur att försvinna om de inte används på ett år, dvs. innan man får nya. Systemet medför ingen merkostnad om man håller sig inom ramen för ens krediter. Det är omöjligt att uppskatta hur många krediter det nya systemet skulle använda, men en grov uppskattning visar på ca 100-300, dvs. även vid en eventuell grov underskattning finns det god marginal till kommunens idag 37 000 krediter.

Att informationen lagras i molnet (på ArcGIS.com) är både en för- och nackdel. Det innebär att man inte själv behöver tillhandahålla en server, men det innebär samtidigt att man skapar en kopia av sin information. Om man sedan fortsätter att arbeta mot sina lokala filer kommer denna onlinekopia bli utdaterad, och vice versa om man bara arbetar mot molnet.

Ett annat problem är att informationen kan vara svår att exportera från molnet, i vissa fall omöjlig. För att få tillgång till sin geografiska information lokalt behöver man exportera den från molnet, men om man bara ska genomföra enklare operationer går det att genomföra direkt mot AGO. Punkter som har bifogade filer är exempelvis ett problem. I nuläget följer inte de bifogade filerna med när man exporterar punkterna.

11.1.2. Lösning 2 – ArcGIS Server

Detta är den mest avancerade lösningen av de tre men det är också den mest kraftfulla. Den kräver mest tid och expertis att utveckla och har högst utvecklingskostnad. För att detta ska löna sig krävs det att systemet ska användas av många. Detta gäller både dess funktionalitet

och dess driftskostnad. Om ett system används av få användare kan man i regel acceptera att systemet är något "sämre".

ArcGIS Server är ett bra alternativ om man vill slippa de driftskostnader som AGO har i form av krediter. Detta innebär dock inte att driftskostnaderna är låga. Det krävs en fysisk serverdator som kan sprida informationen vilken i sin tur kommer att kräva regelbundet underhåll. Denna typ av servrar finns redan i Kalmar kommun, som t.o.m. har ArcGIS Server installerat. Därför blir merkostnaden av ytterligare en tjänst på servern låg.

11.1.3. Lösning 3 – Hybriden

Denna lösning är den som är mest flexibel av de tre. Lösning 1 är bäst lämpad för mindre system och lösning 2 för större, lösning 3 täcker mellanrummet där emellan. När man skapar ett system av denna typ plockar man tjänster från AGO och ArcGIS Server. För att skapa ett mindre system plockar man då fler delar från AGO och färre från ArcGIS Server, samt vice versa för ett större system. Man kan på detta sätt anpassa sitt system efter hur många som ska använda det. Man kan på så vis hitta en bra balans mellan drifts- och utvecklingskostnader.

Utvecklingskostnaden blir relativt liten med denna lösning, detta då det är enkelt att sprida sin karta via AGO. Man kan antingen sprida den via deras färdiga mobilappar eller via en automatiskt genererad webbapp. För att publicera tjänster via ArcGIS Server krävs således en serverdator, om man dock redan har en är det relativt enkelt att publicera enskilda tjänster. Dessa tjänster samlas sedan ihop till en karta via AGO. Om man t.ex. publicerar en tjänst med en bakgrundskarta via ArcGIS Server, kan sedan flera olika kartor använda den som bakgrundskarta. Om man har planer på att publicera flera kartor m.h.a. denna lösning kan det således vara klokt skapa ett antal bastjänster.

11.1.4. För- och nackdelar

Lösning 1 – ArcGIS Online

Fördelar:

- Enklare verktyg än de andra lösningarna, bra för dem som saknar GIS-utbildning.
- Man behöver inte ha tillgång till någon GIS-programvara då allt arbete sker på ArcGIS.com.
- Det går snabbt att starta en tjänst.
- Lättillgängligt, lätt att sprida sina kartor.

Nackdelar:

- Uppfyller inte alla de uppställda kraven i kravspecifikationen (**1.1** och **1.3**).

- Om man vill lägga till sin egen information behöver man ladda upp den till ArcGIS.com.
- Kräver att man har en AGO-licens laddat med krediter.
- En del handlingar kostar krediter.

Lösning 2 – ArcGIS Server

Fördelar:

- Man behöver inte kopiera eller förflytta sin information då man kan arbeta skarpt mot databasen.

Nackdelar:

- Svårare att starta en tjänst, kräver mer tid och expertis än lösning 1.
- Kräver ArcGIS Server-licens (Standard eller Advanced).
- Kräver att man har tillgång till en fysisk serverdator som man kan sprida sin information via.

Lösning 3 – Hybriden

Fördelar:

- Lätt att nå tjänsten via AGO.
- Om man använder ArcGIS Server för att sprida sin information behöver man inte kopiera eller förflytta den.
- Enkelt att skapa och editera tjänsten via ArcGIS.com. Kan ske från vilken dator som helst, inte bara serverdatorn.

Nackdelar:

- Kräver tillgång till en fysisk serverdator som man kan sprida sin information via.
- Kräver en AGO-licens laddat med krediter.
- Kräver ArcGIS Server-licens (Standard eller Advanced).

11.2. Lösningar – Klient sidan

De klientlösningar som har valts att fokusera på är de två färdiga apparna *ArcGIS (app)* och *ArcGIS Collector*, samt att sprida sin information via en *webbapp*.

ArcGIS (app) och ArcGIS Collector har nästan samma funktioner men olika användargränssnitt. Vilken av de två lösningarna som man väljer beror därför på vilket gränssnitt som man föredrar, dvs. det beror på användarens personliga preferenser. Båda apparna är dock fortfarande under utveckling vilket gör det ännu svårare att fastställa vilken

som är den optimala. När detta projekt startade ansåg författaren att ArcGIS (app) var den bättre av de två, men efter en senare uppdatering till Collector anser författaren nu i skrivande stund att den är en bättre lösning.

Förutom användargränssnittet finns det två små skillnader mellan ArcGIS (app) och ArcGIS Collector, nämligen hur de hanterar "*Editor tracking*" samt systemkrascher. Editor tracking innebär att programmet bokför när informationen uppdateras och av vem. ArcGIS Collector stödjer denna funktion helt medan det bara verkar fungera delvis i ArcGIS (app). Författaren har fog att anta att detta är ett problem som kommer att lösas i nästa uppdatering av ArcGIS (app), men har ännu inte lyckats hitta någon information om det. Problemet med systemkrascher gäller även ArcGIS (app), av okänd anledning har det slutat att fungera vid ett antal tillfällen under testperioden. Med tanke på den ringa omfattningen av detta anses problemet för närvarande inte vara av svår art, det är dock en nackdel som man bör ha i åtanke vid val av system.

Förutom att använda sig av befintliga appar kan man även utveckla egna. Det finns guider och *SDK* att ladda ner via ArcGIS hemsida. Denna lösning har inte undersökts närmare då det är den mest komplicerade samt mest tidskrävande lösningen. Om man utvecklar en app för *iPhone* kommer den inte att fungera i en *Android*-telefon. Man behöver därför utveckla flera appar för att täcka alla olika operativsystem. Fördelen med de färdiga ArcGIS-apparna är att de redan finns till både *iOS* och *Android*, och fördelen med webbappar är att de fungerar lika bra i alla operativsystem.

Färdig lösning eller egenutvecklad

En viktig frågeställning är om man ska använda sig av en färdig app (*ArcGIS (app)* alternativt *ArcGIS Collector*) eller om man ska använda sig av en webbapp. Om man ska utveckla en lösning för en avdelning på kommun (som vid detta projekt) så kan det vara en bra och enkel lösning att använda sig av en redan utvecklad app. På så vis får man en fullgod lösning genom att endast installera appen. Om man ska utveckla ett system som ska användas av allmänheten kan det vara klokt att använda sig av en webbapp. Ett exempel på detta är Kalmar kommuns felanmälan som idag sker via en webbapp. Om det i stället hade skett via t.ex. ArcGIS Collector finns det anledning att misstänka att en del personer skulle ha dragit sig för att skicka in en felanmälan. Dels för att en del personer inte vill genomgå ett extra moment (installera appen) för att skicka in sin felanmälan, men även för att man kan anta att det finns en viss naturlig skepsis mot att installera det man inte känner till.

Den webbapp som har utvecklats i projektet har använt sig av JavaScript, jQuery och Dojo. Denna webbapp tog emot information från ArcGIS Server. Anledningen till detta val är att det är de teknikerna som de färdiga exemplen använder sig av på ArcGIS hemsida. Eftersom de utförde sin roll bra har det inte undersökts närmare vilken lösning som skulle vara den absolut bästa.

Alla de tre klientlösningarna ställdes mot varandra för att jämföra vilken som var den mest optimala. I tabell 5.2 nedan jämförs en färdig app (ArcGIS Collector) mot den egenutvecklade webbappen för att gestalta de för- och nackdelar som respektive lösning. ArcGIS (app) redovisas inte i tabellen då dess resultat är nästan identiska med ArcGIS Collector.

Tabell 5.2: Jämförelse av klienter

	Färdig lösning (ArcGIS Collector)	Egenutvecklad webbapp
1. Installation	Användaren måste installera appen.	Ingen installation krävs, användaren behöver endast gå till webbsidan.
2. Inloggning	Inloggning sker första gången, därefter är användarens inloggningsinformation sparad.	Användaren måste logga in varje gång.
3. Grafiskt användargränssnitt	Välpolerat och bra responstid.	Relativt "snyggt", men inte lika välutvecklat som ArcGIS Collector. Webbappen har även sämre responstid än ArcGIS Collector.
4. Skapa punkter	Användaren använder det färdiga verktyget för att skapa punkter. Användaren kan välja att antingen klicka på en punkt i kartan eller att skapa punkten på sin position. Användaren får även möjlighet att fylla i attribut.	Användaren väljer verktyget "Add" och klickar därefter på önskad plats i kartan, sedan har man även möjlighet att fylla i attribut.
5. Bifoga bilder	Verktyget för att skapa punkter kan bifoga bilder från mobilen, alternativt kan man ta en bild med kameran och bifoga direkt.	Eftersom en webbläsare inte har tillgång till kameran krävs installation av utomstående program för att lösa detta problem, t.ex. <i>Picup (Picup, 2013)</i> .
6. Extra verktyg som inte innefattas av kraven	Mätning, vägbeskrivning, sökning, innehållsförteckning, byte av baskarta och <i>editor tracking</i> .	-
7. Publicera liknande tjänster	Om man publicerar en tjänst till sin organisation i ArcGIS Online	Webbappen måste antingen programmeras om för att

8. Support

blir den automatiskt tillgänglig för användarna i ArcGIS Collector.

innehålla flera tjänster, alternativt att man skapar en helt ny. Det innebär i båda fallen mycket merarbete.

ESRI tillhandahåller support om man innehar en ArcGIS-licens. Det finns även hjälpdokumentation som alla har tillgång till, inkl. hjälpforum.

(Ingen hjälpdokumentation har upprättats då en annan lösning valdes som slutgiltig lösning.)

Den enda fördelen som den egenutvecklade lösningen har över ArcGIS Collector är punkt 1 i Tabell 5.2, att man slipper installera ett program. Vid upprepad användning är det dock att föredra att installera appen, dvs. det är inte bara en nackdel för det färdigutvecklade systemet. Dessutom är en installerad app snabbare och har bättre funktionalitet än en webbapp.

Det är dock sant att även den egenutvecklade webbappen har möjlighet att uppfylla alla kraven, men författaren anser inte att den gör det lika väl. En färdig lösning anses exempelvis vara bättre eftersom:

- Det är en app som man installerar på sin smartmobil. Installerade program är snabbare att använda än webbappar och ger bättre funktionalitet (t.ex. direkt tillgång till kameran). Gäller punkt 2, 3 och 5 i tabell 5.2.
- Den har fler utvecklingstimmar bakom sig. Det är många utvecklare som ligger bakom ArcGIS Collector, till skillnad från författaren som står ensam bakom den egenutvecklade lösningen. Detta leder först och främst till en produkt som utför sig uppgift bättre. Det leder även till att man kan spendera tid på att förbättra detaljer som inte är obligatoriska, t.ex. snyggare grafiskt användargränssnitt eller extra funktioner (exempelvis mätverktyg). Gäller punkt 3, 6 och 8 i tabell 5.2.
- En färdig lösning innebär att man inte behöver spendera någon tid eller några pengar på utveckling.

Efter att ha testat de två olika lösningarna (färdig/egen) såg inte författaren någon anledning till att gå vidare med den egenutvecklade webbappen. Anledningen till detta är att webbappen aldrig kan uppfylla de uppställda kraven *bättre* än en av de färdiga lösningarna, de kan maximalt anses vara jämnlika. Det ansågs dock varken ekonomiskt eller tidsmässigt försvarbart att spendera flera veckor med att utveckla en lösning som endast skulle vara jämnbra. När de två lösningarna har jämförts har webbappen räknats som om den vore helt färdig. Det faktum att den idag saknar möjlighet att bifoga filer har inte tagits hänsyn till.

Att de färdiga lösningarna delvis låser systemet anses inte vara ett problem i nuläget, det kan dock bli det i framtiden. De uppfyller alla kraven i kravspecifikationen, men det kan bli svårt att anpassa systemet för andra typer av ändamål. Det är t.ex. inte möjligt att rita linjer i

det valda systemet, och det finns ingen möjlighet att åtgärda detta. Fördelen med en egenutvecklad lösning skulle bl.a. vara att man hade haft möjlighet att programmera in den funktionaliteten. Det går dock inte att förbereda sig för alla olika typer av problem som kan tänkas uppstå, dvs. man kan inte gardera sig för eventuella framtida problem som kanske inte ens uppstår. Under projektets gång har det inte heller diskuterats att man ska utvidga kraven efter projektets gång. ArcGIS Collector har därför valts eftersom det uppfyller kraven väl, då författaren bedömer att man ej kan ta hänsyn till eventuella framtida (oförutsägbara) krav.

Om man i framtiden anser att man måste utveckla systemet utanför de uppsatta ramarna kommer man kanske bli tvungen att utveckla sin egen lösning från grunden. Fördelen med detta är dock att man tills dess har kunnat använda sig av ett gratis system (ArcGIS Collector), dvs. man har ej behövt lägga ned varken tid eller pengar på det nuvarande systemet. ArcGIS Collector är dock ett program som fortfarande är under utveckling, dess rammar flyttas därför hela tiden och det blir därmed mer och mer kraftfullt.

11.2.1. För- och nackdelar

ArcGIS (app)

Fördelar:

- Man behöver inte utveckla en egen klient.
- Väl integrerat med AGO, men även ArcGIS Server.
- Appen blir snabbare och kraftfullare än en webblösning då den är installerad.

Nackdelar:

- Man behöver installera en app för att använda tjänsten.
- Man är låst till de val som gjorts när appen utvecklades.

ArcGIS Collector

Fördelar:

- Utvecklat för att användas i fält och samla in information.
- Man behöver inte utveckla en egen klient.
- Väl integrerat med AGO.
- Appen blir snabbare och kraftfullare än en webblösning, då den är installerad.

Nackdelar:

- Man behöver installera en app för att använda tjänsten.
- Man är låst till de val som gjorts när appen utvecklades.

Webbapp

Fördelar:

- Man kan skräddarsy sin tjänst helt efter behov.

Nackdelar:

- Det krävs tid och pengar för att utveckla en egen klientlösning.
- Är den lösning som kräver mest expertis av de tre lösningarna.

11.3. Slutprodukten

Den bästa lösningen för detta projekt är att på serversidan använda en **hybridlösning** och på klientsidan **ArcGIS Collector**. Många andra lösningar är nästan lika bra, men bara den som författaren anser är mest optimal kommer att redovisas.

På serversidan valdes en hybridlösning eftersom den innehåller det bästa från de övriga två lösningarna. Det är önskvärt att sprida sin information via ArcGIS Server då all befintlig information samt alla ändringar som användarna gör stannar på kommunens egen server. För att underlätta åtkomsten av informationen är det dock fördelaktigt att man sprider den via AGO, då kan man använda sig av en färdig lösning som t.ex. ArcGIS Collector. En hybridlösning kommer att kosta krediter trots att all information sprids från ArcGIS Server. Detta anser dock inte författaren vara ett stort problem då ett system av denna storlek kommer att kosta uppskattningsvis cirka 100 kredit per år, att jämföra med de cirka 37 000 krediter som Kalmar kommun har tillgång till.

På klientsidan valdes ArcGIS Collector eftersom den är en enkel och färdig lösning. Eftersom ArcGIS Collector uppfyller de krav som är uppställda på slutprodukten känns det onödigt att utveckla en egen webbapp. Även ArcGIS (app) var en stark kandidat då de båda i stort sett har samma funktionalitet, men ArcGIS Collector ansågs ha ett enklare användargränssnitt som är mer anpassat för uppgiften. Det har flera gånger i denna rapport nämnts hur låst systemet kan bli om man väljer att använda sig av en färdig lösning, som t.ex. ArcGIS Collector. Det kan därför verka konstigt att man trots detta har valt det som slutgiltig lösning. Detta val gjordes efter grundliga test av flertalet färdiga lösningar och den egenutvecklade. Efter samtliga test anser författaren att denna "lösning" inte är ett problem för systemet, då systemet uppfyller alla de uppställda kraven väl.

11.3.1. Serversidan

I den hybridlösning som har valts kommer informationen spridas via AGO, men all information kommer att härstamma från en av kommunens servrar som kör ArcGIS Server.

Bakgrundskartan kommer att spridas från ArcGIS Server som en vanlig karttjänst. Den kommer sedan att infogas i AGO och väljas som bakgrundskarta.

Editeringslagret kommer att spridas från ArcGIS Server som en *Feature service*, och den kommer därför vara tvungen att lagra sin data i en *ArcSDE* databas.

Stomnätspunkterna bör publiceras som en egen tjänst för att underlätta skapandet av kartan då de ska vara klickbara och länka till bilder. Detta görs bäst via ArcGIS Server, men det är även möjligt att ladda upp informationen till AGO. Bilderna måste kopieras till serverdatorn då man inte kan länka till dem direkt på intranätet.

11.3.2. Klientsidan

Användaren behöver installera ArcGIS Collector på sin smartmobil samt skapa en användare i kommunens organisation. För att skapa ett konto i organisationen krävs det att man blir inbjuden av en administratör. Det är GIS-avdelningen som administrerar organisationen.

12. Diskussion

Projektet lyckades uppfylla alla de uppställda kraven i *kravspecifikationen*, undantaget för detta är *Lösning 1- ArcGIS Online*. Det svåraste med projektet har varit att utveckla en bra webbapp med allt som detta innebär. Det finns många bra exempel på ArcGIS hemsida, men att kombinera flera exempel till en optimal lösning är ingen självklarhet. Det finns inte heller ett exempel för alla typer av situationer utan man får bara en grund att bygga vidare på. Det största problemet med exemplen är dock att de oftast är anpassade för att användas på en dator, inte i en smartmobil. Exemplet fungerar kanske i en smartmobil trots att det är anpassat för datorn, men eftersom det är stort skillnad på deras skärmstorlek hamnar kanske knappar och viktig information utanför skärmen.

Den klientlösning som valts är en redan färdig lösning, något som medför både för- och nackdelar. Det är dock väldigt fördelaktigt om man har möjligheten att använda en färdig lösning då man inte behöver lägga tid eller pengar på att utveckla en egen produkt. Tyvärr finns det många gånger inte en färdig lösning att tillgå. Det är inte heller så att gratis lösningar alltid innebär att man sparar pengar. Det tar tid att utveckla sin egen lösning, tid som innebär en kostnad för beställaren. På en arbetsplats med tendens till undrebemanning kan således en färdig lösning bli den mest prisvärda, även om den inte är gratis.

I projektet användes en ArcSDE databas för att lagra informationen för editeringstjänsten, något som man i efterhand kan ifrågasätta. Valet att använda ArcSDE påverkar varken testandet eller slutprodukten, dock kan det vara klokt att tänka efter innan man väljer att använda sig av ArcSDE. ArcSDE är ett väl fungerande system vilket dock innebär att man då delvist låser sitt system till ArcGIS. ArcSDE är som namnet antyder medlem i ArcGIS-familjen. Systemet blir dock inte helt låst utan det finns andra programvaror som kan använda sig av information från en ArcSDE databas, ett exempel på detta är *MapServer (MapServer, 2013)*.

Den valda lösningen har som nackdel att den kräver en internetuppkoppling, varför systemet kan vara känsligt för arbete ute i skogen där det är dålig mobiltäckning. Eftersom författaren ansåg att detta skulle kunna vara ett betydande problem genomfördes en extra intervju med två mättingenjörer i projektets slutskede. Båda mättingenjörerna uppgav dock att de inte ansåg detta vara ett stort problem då större delen av Kalmar kommun har bra täckning. Detta system är även bara tänkt att ses som ett komplement, mättingenjörerna kan således fortfarande genomföra sitt vanliga arbete även om de inte har tillgång till systemet.

12.1. Vidareutveckling

Ett förslag på vidareutveckling av examensarbetet är att ta de lösningar som har presenterats i projektet och försöka sprida dem till hela kommunen. Man skulle exempelvis kunna presentera systemet och dess fördelar för de olika avdelningarna i kommunen, och därefter samla in förslag på vad man skulle kunna tänka sig att använda systemet till. Miljöavdelningen kanske t.ex. vill ha en tjänst där de kan visa var och när de har genomfört

sina inspektioner. När man samlat in information från de olika avdelningarna skulle man undersöka om det finns någon möjlighet att skapa en universallösning. Författaren anser att det rimligtvis bör vara svårt att skapa en tjänst som passar alla men att man ev. kan upprätta mallar och guider som underlättar att starta egna tjänster.

Något som är mycket intressant är om det finns en möjligheten för ens klient att fungera även om den tappar kontakten med internet. Ingen av de klientlösningar som har presenterats fungerar i nuläget utan internetuppkoppling. ESRI har dock uttalat sig om att de arbetar för att ta fram lösningar på detta problem för sina samtliga produkter. Alternativet till detta är att man själv utvecklar en app eller webbapp som fungerar även utan internetuppkoppling, ett exempel på detta är *iSurvey (Ekstedt m.fl., 2012)*.

Ett annat förslag är att man undersöker möjligheten att sprida stomnätsskisserna på ett smartare sätt. Ett exempel på detta är om man kan bifoga varje stomnätsskiss till rätt stomnätspunkt i kartan, på samma sätt som när en användare själv skapar en punkt och bifogar en bild. Att genomföra detta för hand är dock väldigt tidskrävande. Det föreslås att man försöker lösa problemet med t.ex. ett *Python*-skript eller motsvarande. Nackdelen med detta är dock att man skapar kopior av stomnätsskisserna.

Säkerheten med systemet bör undersökas vidare, detta gäller speciellt utomstående filer som stomnätsskisserna. AGO och ArcGIS Server är gjorda för att hantera geografisk information och gör det väl, det inkluderar även informationens säkerhet. En frågeställning är hur man sedan hanterar filer som krävs för ett system men som inte går att publicera med AGO eller ArcGIS Server. Stomnätsskisserna är (som nämnts ovan) ett exempel på detta, de har externa bilder som ska kopplas till punkter i kartan. Hur kan man publicera dessa "externa" filer på ett säkert sätt? Ett annat exempel vore om man skulle välja att utvidga systemet för att inkludera förrättningsakter, vilket skulle innebära mycket höga säkerhetskrav på systemet.

12.2. Vidareutveckling mot fria data

Ett annat projekt vore att skapa en publik tjänst för smartmobiler, en tjänst som skulle vara inspirerad av *OpenStreetMap* (fast endast för Kalmar kommun). Första steget i utvecklingen är att Kalmar kommun skapar en baskarta som man licensierar under en fri licens. Sedan skapar man ett system där kommunen och allmänheten tillsammans kan hjälpas åt att förbättra kartan. Denna "förbättring" kan ske på olika sätt; antingen att allmänheten bara får lämna synpunkter på kartan alternativt att man kan hjälpa till att mäta in detaljer i kartan m.h.a. smartmobilens GPS-mottagare.

Tjänsten skulle sedan även kunna fungera som kommunikationsväg mellan kommunen och allmänheten, men även invånarna emellan. Några exempel på detta är t.ex. att:

- kommunen vid översvämningar kan varna för drabbade områden samt vice versa dvs. att allmänheten kan hjälpa till att "flagga" drabbade områden.
- vid problem med kranvattnet kan kommunen markera på kartan var tillfälliga vattendunkar kommer att placeras.
- kommunen kan markera tänka avspärningar som kan uppstå vid evenemang, ett lokalt exempel är tävlingen *Iron man* vilket Kalmar kommun var värd för förra året.
- allmänheten kan skicka in sina åsikter på platser i kommunen, t.ex. ett område som de tycker bör renoveras eller restaureras.
- invånarna kan berätta för andra invånare om saker som händer i kommunen, t.ex. kosläpp, små festivaler m.m.

Den nya kartan bör licenseras under en licens som liknar den som *OpenStreetMap* använder. Då vet invånarna att informationen tillhör "alla", inte bara kommunen. Kommunen behöver inte heller oroa sig för att någon ska försöka få upphovsrätt på informationen. Men det möjliggör även att man kan integrera information från *OpenStreetMap* med sin egen.

13. Slutsatser

Projektets mål var att utveckla ett system som mätningenjörerna i kommunen skulle kunna använda för att nå kommunens geografiska information, både att hämta information och att skicka tillbaka information till servern. Ett ytterligare mål med projektet var att undersöka möjligheterna att bygga vidare på projektet.

Projektets mål får anses vara väl uppfyllt. En slutprodukt har presenterats som uppfyller de i kravspecifikationen uppställda kraven. Projektet har dessutom hjälpt till att bereda väg för liknande projekt i framtiden då flera lösningar har presenterats. Dessa lösningar får antas kunna vara behjälpliga vid liknande projekt i kommunen.

I projektets början undersöktes förutsättningarna för projektet m.h.a. intervjuer. Utvalda representanter fick berätta om deras krav och önskemål på projektet. Det är utifrån denna behovsanalys som grunden för projektet är lagd. Utifrån behovsanalysen upprättades sedan åtta krav. En del av kraven var tvungna att bli uppfyllda för att projektet skulle bli godkänt medan andra bara var önskvärda i mån av tid. Ett exempel på ett obligatoriskt krav var att kartan skulle ha samma symbologi i smartmobilen som i datorn och att man ska ha möjlighet att skapa punkter. Trots att alla krav inte var obligatoriska uppfylldes ändå samtliga åtta krav i projektet vilket får anses vara en fördel inför framtida projekt inom kommunen.

Liknande existerande system undersöktes för att ge en uppfattning av vad man kan förväntas få för resultat, samt vilka program och tekniker som man kan använda sig av. Det undersöktes dels hur det fungerade att använda sig av färdiga lösningar, dels hur det fungerar att utveckla en egen lösning. När det gällde att utveckla en egen lösning undersöktes främst lösningar med *öppen källkod*.

Eftersom lösningar med öppen källkod undersöktes blev det även nödvändigt att noggrant undersöka licenskraven för hela systemet. Detta beror på att vissa öppna källkodsprogram inte är kompatibla med varandra eller med vissa användningsområden (exempelvis kommersiellt). När licenskraven för programvaran var under granskning undersöktes även licenskraven för geografisk information, samt om man kan använda sig av fri information från exempelvis *OpenStreetMap*.

Efter att ha undersökt flera olika typer av lösningar valdes det att gå vidare med främst ArcGIS-lösningar. Anledningen till detta är att det underlättar integrationen med kommunens nuvarande GIS-system, som idag använder sig av främst ArcGIS-lösningar. På serversidan innebar detta ArcGIS Online och ArcGIS Server, och på klientsidan ArcGIS (app) och ArcGIS Collector. På klientsidan undersöktes även möjligheten att utveckla en egen webbapp m.h.a. *JavaScript*, *Dojo* och *jQuery*, dvs. en lösning som använder sig av öppen källkod.

De lösningarna som nämns i stycket ovan är de som valdes att undersökas närmare. Dessa lösningar finns redovisade i rapporten i detalj. Anledningen till detta är att man kan ta

lärdom av det som framkommit i projektet, då en lösning inte är perfekt för alla olika projekt. Det var därför viktigt för författaren att skapa en rapport som beskrev alla lösningar väl, så att man kan använda den som grundinformation till flera olika projekt. I slutänden valdes en lösning där man på serversidan använde sig av både ArcGIS Online samt ArcGIS Server, och ArcGIS Collector på klientsidan.

Med hjälp av denna rapport kan Kalmar kommun enkelt starta upp ett system som klarar av den undersökta uppgiften. Rapporten är dock inte låst till Kalmar kommun, även andra kommuner eller företag med liknande förutsättningar kan ta lärdom av projektet. Med liknande förutsättningar menas främst att de idag använder ArcGIS-produkter. Man kan även använda rapporten för att starta upp liknande system med de presenterade alternativa lösningarna, då de är förklarade nästan lika väl som den valda lösningen.

Projektet har även hjälpt till att belysa den mobila informationens roll i kommunen. Det har medfört att mobil information har fått uppmärksamhet som det tidigare inte haft. Projektet har även visat hur man enkelt kan starta upp ett nytt mobilt system, att det inte behöver vara invecklat och tidskrävande. Förhoppningsvis är detta projekt något som kommunen kommer att bygga vidare på i framtiden.

Tidigare examensarbeten

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 Felleon (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 Field Work*
- 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 *Danebjer Lisette och Nyberg Magdalena (2013) Utbyte av geodata - studie av leveransstrukturer enligt Sveriges kommuner och landstings objekttypskatalog*

Referenser

Abraham A, Knapkog S J, Sanyal S, Sanyal S, Tiwari A (odaterad), *A MULTI-FACTOR SECURITY PROTOCOL FOR WIRELESS PAYMENT- SECURE WEB AUTHENTICATION USING MOBILE DEVICES*, finns på <http://arxiv.org/ftp/arxiv/papers/1111/1111.3010.pdf>, besökt 2013-04-19

Acunetix (2013), *What Are Web Applications?*, finns på <http://www.acunetix.com/websitesecurity/web-applications/>, besökt 2013-04-22

All Points Blog (2011), *Update: Esri has 40+% of GIS Marketshare*, finns på <http://apb.directionsmag.com/entry/esri-has-40-of-gis-marketshare/215188>, besökt 2013-02-21

Apache Software Foundation (2004), *Apache License, Version 2.0*, finns på <http://www.apache.org/licenses/LICENSE-2.0.html>, besökt 2013-04-19

Apple (2013a), *What is iOS*, finns på <http://www.apple.com/ios/what-is/>, besökt 2013-04-15

Apple (2013b), *iOS Developer Program*, finns på <https://developer.apple.com/programs/ios/>, besökt 2013-02-07

ArcGIS (2013a), *Get Started with the ArcGIS APIs*, finns på <http://www.esri.com/getting-started/developers/get-started>, besökt 2013-04-15

ArcGIS (2013b), *Mobile Template*, finns på http://help.arcgis.com/en/webapi/javascript/arcgis/jstutorials/#mobile_dev, besökt 2013-01-29

ArcGIS (2013c), *Geolocation API*, finns på http://help.arcgis.com/en/webapi/javascript/arcgis/samples/exp_geolocate/index.html, besökt 2013-01-29

ArcGIS (2013d), *ArcGIS for iPhone*, finns på <https://itunes.apple.com/us/app/arcgis/id379687930?mt=8>, besökt 2013-01-29

ArcGIS (2013e), *ArcGIS Collector*, finns på <https://itunes.apple.com/us/app/collector-for-arcgis/id589674237?mt=8>, besökt 2013-01-29

ArcGIS (2013f), *Service Credits Estimator*, finns på <http://www.esri.com/software/arcgis/arcgisonline/credits/estimator>, besökt 2013-02-11

ArcGIS (2013g), *Cloud GIS*, finns på <http://www.esri.com/technology-topics/cloud-gis>, besökt 2013-04-16

ArcGIS (2013h), *Types of geodatabases*, finns på <http://resources.arcgis.com/content/geodatabases/10.0/types-of-geodatabases>, besökt 2013-03-20

Becker E, Buhse W, Günnewig D, Rump N (2003), *Digital Rights Management: Technological, Economic, Legal and Political Aspects*, finns på http://books.google.se/books?id=y_ea3Vj_IMC&source=gbs_navlinks_s, besökt 2013-04-24

Bennett (2010), *OpenStreetMap*, finns på http://books.google.se/books?id=SZfqRcPXApoC&source=gbs_navlinks_s, besökt 2013-04-23

Brinkhoff T., Kresse W. (2012), *Databases*, Springer Handbook of Geographic Information, Kresse W. och Danko D. M., sidor 61-106

Danko D. M., Fadaie K., Kresse W. (2012), *Standardization*, Springer Handbook of Geographic Information, Kresse W. och Danko D. M., sidor 393-566

CarryMap (2013), *CarryMap Observer for iPhone*, finns på <https://itunes.apple.com/us/app/carrymap-observer/id432496097?mt=8>, besökt 2013-03-18

Chapple M (2013), *The ACID Model*, finns på <http://databases.about.com/od/specificproducts/a/acid.htm>, besökt 2013-04-18

Communities Dominate Brands (2013), *Smartphone Penetration Rate by Country!*, finns på <http://communities-dominate.blogspot.com/brands/2011/12/smartphone-penetration-rates-by-country-we-have-good-data-finally.html>, besökt 2013-02-04

Creative Commons (2013), *Creative Commons*, finns på <http://creativecommons.org>, besökt 2013-04-24

Defective by Design (2013), *What is DRM?*, finns på http://www.defectivebydesign.org/what_is_drm, besökt 2013-04-24

dholm (2011), *Comparison of the Open Source Licences*, finns på <http://www.dholm.com/2011/02/16/tumblelog-110216/>, besökt 2013-05-23

Dojo (2013), *Dojo License*, finns på <http://dojotoolkit.org/license>, besökt 2013-03-21

Dropbox (2013), *Dropbox*, finns på <http://www.dropbox.com>, besökt 2013-04-16

Ekstedt G, Endoff T (2012), *Design and Development of a Mobile GIS Application for Municipal Field Work*, Lund, Sverige, Institutionen för naturgeografi och ekosystemvetenskap

ESRI (2013), *Smartmobil SDK*, finns på

<http://www.esri.com/software/arcgis/smartphones/develop>, besökt 2013-02-07

FreeBSD (2013), *Why you should use a BSD style license for your Open Source Project*, finns

på <http://www.freebsd.org/doc/en/articles/bsdl-gpl/article.html>, besökt 2013-02-28

GeoMobile for ArcGIS (2013), *GeoMobile for ArcGIS for iPhone*, finns på

<https://itunes.apple.com/ng/app/geomobile-for-arcgis/id459112762?mt=8>, besökt 2013-03-18

GeoServer (2013), *What is GeoServer*, finns på

<http://geoserver.org/display/GEOS/What+is+Geoserver>, besökt 2013-04-16

Gnu (2013a), *What is Copyleft?*, finns på

<http://www.gnu.org/copyleft/>, besökt 2013-04-19

Gnu (2013b), *GNU GPL Licence v3.0*, finns på

<http://www.gnu.org/licenses/gpl-3.0.html>, besökt 2013-02-28

Gnu (2013c), *What is free software?*, finns på

<http://www.gnu.org/philosophy/free-sw.html>, besökt 2013-05-03

Goodchild M F (2007), *Citizens as Sensors: The World of Volunteered Geography*, finns på

http://ncgia.ucsb.edu/projects/vgi/docs/present/Goodchild_intro.pdf, besökt 2013-04-23

Google (2013), *Developer Registration*, finns på

<http://support.google.com/googleplay/android-developer/answer/113468?hl=en>, besökt 2013-02-21

Helsingborg kommun (2013), *Parkering*, finns på

<http://kartor.helsingborg.se/webbapp/parkering/>, besökt 2013-04-16

HOT (2013), *The Humanitarian OpenStreetMap Team*, finns på

<http://hot.openstreetmap.org/>, besökt 2013-05-05

IBM (2012), *Who says open source software is less secure?*, finns på

https://www.ibm.com/developerworks/community/blogs/6e6f6d1b-95c3-46df-8a26-b7efd8ee4b57/entry/is_open_source_software_less_secure230?lang=en, besökt 2013-04-26

IST (2013), *Viruses, Spyware and Malware*, finns på

<http://ist.mit.edu/security/malware>, besökt 2013-04-18

jQuery (2013), *jQuery*, finns på

<http://jquery.com/>, besökt 2013-04-16

jQuery mobile (2013), *jQuery mobile*, finns på <http://jquerymobile.com/>, besökt 2013-04-16

JSON (2013), *JSON*, finns på <http://json.org/>, besökt 2013-03-21

Kalmar kommun (2013), *Felanmälan inom Kalmar kommun*, finns på <http://gis.kalmar.se/felanmal/>, besökt 2013-01-28

Kartverket (2013), *Kartverket frigir stedsnavndata*, finns på <http://www.kartverket.no/nyheter/stedsnavn>, besökt 2013-04-30

Klang M (2010), *Copyright – Copyleft*, finns på http://books.google.se/books?id=wFNteLZED0gC&dq=odbl&source=gb_s_navlinks_s, besökt 2013-04-25

Lantmäteriet (2013), *Kort fakta*, finns på <http://lantmateriet.se/Om-Lantmateriet/Om-oss/Korta-fakta/>, besökt 2013-05-09

LINFO (2005), *Proprietary Software Definition*, finns på <http://www.linfo.org/proprietary.html>, besökt 2013-04-26

Longley P, Goodchild M, Maguire J, Rhind D (2011), *Geographic information systems & science*, finns på http://books.google.se/books?id=zyFR8uxHM9oC&dq=volunteer+geographic+information&source=gb_s_navlinks_s, besökt 2013-04-25

MapServer (2013), *MapServer*, finns på <http://mapserver.org/about.html>, besökt 2013-04-16

NIST (2011), *NIST Definition of Cloud Computing*, finns på <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>, besökt 2013-04-16

OGC (2013), *Open Geospatial Consortium*, finns på <http://www.opengeospatial.org/>, besökt 2013-04-16

Open Data Commons (2013), *Open Database License (ODbL) v1.0*, finns på <http://opendatacommons.org/licenses/odbl/1.0/>, besökt 2013-04-26

Open Source Initiative (2013a), *The Open Source Definition*, finns på <http://opensource.org/osd>, besökt 2013-03-04

Open Source Initiative (2013b), *Academic Free License*, finns på <http://opensource.org/licenses/AFL-3.0>, besökt 2013-04-16

Open Source Initiative (2013c), *Artistic License 2.0*, finns på <http://opensource.org/licenses/artistic-license-2.0>, besökt 2013-04-19

Open Source Initiative (2013d), *The MIT License (MIT)*, finns på <http://opensource.org/licenses/MIT>, besökt 2013-04-19

Open Source Resource Center (2013), *Open Source License Data*, finns på <http://osrc.blackducksoftware.com/data/licenses/index.php>, besökt 2013-02-28

OpenCola (2000), *openCOLA*, finns på <http://web.archive.org/web/20000815235029/http://www.opencola.com/>, besökt 2013-02-27

OpenLayers (2013), *OpenLayers*, finns på <http://openlayers.org/>, besökt 2013-04-16

OpenGeo (2013), *Spatial Database Tips and Tricks*, finns på <http://workshops.opengeo.org/postgis-spatialdbtips/introduction.html#introduction>, besökt 2013-04-22

OpenStreetMap (2012), *OpenStreetMap data license is ODbL*, finns på <http://blog.osmfoundation.org/2012/09/12/openstreetmap-data-license-is-odbl/>, besökt 2013-04-26

OpenStreetMap (2013a), *Potential Datasources*, finns på http://wiki.openstreetmap.org/wiki/Potential_Datasources, besökt 2013-04-30

OpenStreetMap (201b), *Att skapa information för OpenStreetMap*, finns på http://wiki.openstreetmap.org/wiki/Press_Kit, besökt 2013-05-03

OpenStreetMap (2013c), *Copyright and license*, finns på <http://www.openstreetmap.org/copyright>, besökt 2013-04-24

OSGeo (2013), *Proj4J*, finns på <http://trac.osgeo.org/proj4j/>, besökt 2013-04-16

Perens B (2007), *The Open Source Definition*, finns på http://kojp.mtf.stuba.sk/mironovova/lib/exe/fetch.php/2_semester:projects:glossary:0708:glossary_2sem_0708_aia4_srokamichal_osdtext.pdf, besökt 2013-05-07

Picup (2013), *Picup*, finns på <http://picupapp.com/>, besökt 2013-03-04

PostGIS (2013), *PostGIS*, finns på <http://postgis.net/>, besökt 2013-04-16

PostgreSQL (2013), *PostgreSQL*, finns på <http://www.postgresql.org/about/>, besökt 2013-04-16

SearchSecurity (2007), *firewall*, finns på

<http://searchsecurity.techtarget.com/definition/firewall>, besökt 2013-04-18

Shihaam (2011), *De sex Creative commons-licenserna*, finns på

<http://education-copyright.org/creative-commons/>, besökt 2013-05-03

Skype (2013), *Skype*, finns på <http://www.skype.com>, besökt 2013-04-18

Strategy analytics (2013), *Android and Apple iOS Capture a Record 92 Percent Share of Global Smartphone Shipments in Q4 2013*, finns på

<http://blogs.strategyanalytics.com/WSS/post/2013/01/28/Android-and-Apple-iOS-Capture-a-Record-92-Percent-Share-of-Global-Smartphone-Shipments-in-Q4-2012.aspx>, besökt 2013-02-21

Sophos (2013), *BYOD Risks & Rewards*, finns på [http://www.sophos.com/en-us/security-](http://www.sophos.com/en-us/security-news-trends/security-trends/byod-risks-rewards/what-byod-means-for-business.aspx)

[news-trends/security-trends/byod-risks-rewards/what-byod-means-for-business.aspx](http://www.sophos.com/en-us/security-news-trends/security-trends/byod-risks-rewards/what-byod-means-for-business.aspx), besökt 2013-04-19

Swedbank (2013), *Swedbank för iPhone, iPod touch och iPad*, finns på

<https://itunes.apple.com/se/app/swedbank/id344161302>, besökt 2013-04-18

TechRepublic (2005), *SolutionBase: Strengthen network defenses by using a DMZ*, finns på

<http://www.techrepublic.com/article/solutionbase-strengthen-network-defenses-by-using-a-dmz/5756029>, besökt 2013-04-18

Think with Google (2013), *Antalet användare av smartmobiler i Sverige*, finns på

<http://www.thinkwithgoogle.com/mobileplanet/en/>, besökt 2013-02-04

webreference (2003), *DRM*, finns på

<http://www.webreference.com/multimedia/video/5.html>, besökt 2013-05-03

Weiske C (2013), *Using SSL Client Certificate with PHP*, finns på

<http://cweiske.de/tagebuch/ssl-client-certificates.htm>, besökt 2013-04-18

wikiHow (2013), *4 ways to make OpenCola*, finns på

<http://www.wikihow.com/Make-OpenCola>, besökt 2013-02-27

WIPO (odaterad), *Understanding copyright and related rights*, finns på

http://www.wipo.int/export/sites/www/freepublications/en/intproperty/909/wipo_pub_909.pdf, besökt 2013-04-19

Wired (2013), *SimCity Blackout Is Just One More DRM Disaster*, finns på

<http://www.wired.com/gamelife/2013/03/simcity-outage/>, besökt 2013-04-25

Personlig kommunikation

Bernhard (2013), Sabina Bernhard, GISsamordnare, Kalmar kommun, kontinuerlig kontakt mellan 2013-01 – 2013-05

Wendel (2013), Andreas Wendel, dataingenjör, Kalmar kommun, kontinuerlig kontakt mellan 2013-01 – 2013-05

Bilaga A – Källkod webbappen

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=7,IE=9">
<!-- Sets whether a web application runs in full-screen mode. -->
<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-
scalable=0">
<!-- Sets the style of the status bar for a web application. -->
<meta name="apple-mobile-web-app-status-bar-style" content="black-translucent">
<title>Mobile Gallery</title>

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.0/jquery.mobile-1.3.0.min.css" />
<link rel="stylesheet" href="http://serverapi.arcgisonline.com/jsapi/arcgis/3.3/js/esri/css/esri.css" />

<style>
html, body {
  height: 100%;
  margin: 0px;
  padding: 0px;
  width: 100%;
}

.ui-content {
  padding: 0 !important;
}

#map {
  height: 100%;
  width: 100%;
  position: absolute;
  z-index: 0;
}

</style>

<script src="http://code.jquery.com/jquery-1.9.0.min.js"></script>
<script src="http://code.jquery.com/mobile/1.3.0/jquery.mobile-1.3.0.min.js"></script>
<script src="http://serverapi.arcgisonline.com/jsapi/arcgis/3.3compact"></script>

<script>
dojo.require("esri.map");
dojo.require("esri.dijit.AttributeInspector-all");
dojo.require("esri.tasks.locator");
dojo.require("dijit.form.ComboBox");
dojo.require("esri.dijit.PopupMobile");

var map,
  locator,
```

```

    citizenRequestLayer;

function init() {
    //This sample requires a proxy page to handle communications with the ArcGIS Server services. You
    will need to
    //replace the url below with the location of a proxy on your machine. See the 'Using the proxy page'
    help topic
    //for details on setting up a proxy page.
    // esri.config.defaults.io.proxyUrl = "/proxy";

    var popup = new esri.dijit.PopupMobile(null, dojo.create("div"));
    map = new esri.Map('map', {
        basemap : 'topo',
        center : [16.3, 56.6],
        zoom : 12,
        infoWindow: popup
    });

    var template = new esri.InfoTemplate();
    template.setTitle("<b>Punkt, id=${OBJECTID}</b>");
    template.setContent("<b>Kommentar:</b> ${Kommentar}<br><b>Skapad:</b>
    ${created_date:DateFormat}<br><b>Stomnätsskiss:</b> <a
    href=http://www.kalmar.se/${OBJECTID}.jpg target=&quot;_blank&quot;>Länk</a>");
    //add the citizens request layer to the map as a feature layer
    citizenRequestLayer = new
    esri.layers.FeatureLayer("http://localhost:6080/arcgis/rest/services/Upload/FeatureServer/0", {
        mode : esri.layers.FeatureLayer.MODE_ONDEMAND,
        infoTemplate: template,
        outFields : ["*"]
    });
    dojo.connect(citizenRequestLayer, 'onClick', function (evt) {
        map.infoWindow.setFeatures([evt.graphic]);
    });
    locator = new
    esri.tasks.Locator("http://sampleserver1.arcgisonline.com/ArcGIS/rest/services/Locators/ESRI_Geoc
    ode_USA/GeocodeServer");
    map.addLayers([citizenRequestLayer]);
    map.infoWindow.resize(185, 100);

    dojo.connect(map, 'onLoad', function (evt) {
        $(document).ready(jQueryReady);
    });
}

function jQueryReady() {
    dojo.connect(map, 'onLayersAddResult', initEditor);
}

function initEditor(results) {
    var layerInfo = [
        {
            'featureLayer' : citizenRequestLayer,

```

```

'showAttachments':true,
'showDeleteButton':true,
'isEditable' :true,
'fieldInfos' :[
    {
        'fieldName':'OBJECTID',
        'label' :'OBJECTID',
        'isEditable':false
    },
    {
        'fieldName':'created_user',
        'label' :'created_user',
        'isEditable':false
    },
    {
        'fieldName':'created_date',
        'label' :'created_date',
        'isEditable':false
    },
    {
        'fieldName':'last_edited_user',
        'label' :'last_edited_user',
        'isEditable':false
    },
    {
        'fieldName':'last_edited_date',
        'label' :'last_edited_date',
        'isEditable':false
    },
    {
        'fieldName' :'Kommentar',
        'label' :'Kommentar',
        'isEditable' :true,
        'stringFieldOption':esri.dijit.AttributeInspector.STRING_FIELD_OPTION_TEXTAREA
    }
]
];
var attInspector = new esri.dijit.AttributeInspector({
    layerInfos:layerInfo
}, "attributeDiv");

dojo.connect(attInspector, "onAttributeChange", function (feature, fieldName, newFieldValue) {
    feature.attributes[fieldName] = newFieldValue;
    feature.getLayer().applyEdits(null, [feature], null);
});

$.each(results, function (i, result) {
    var infos = result.layer.renderer.infos;
    $.each(infos, function (j, info) {
        var listItem = $('<li/>').on('click', function (e) {
            map.setMapCursor('pointer');
        });
    });
});

```



```

    addFeature(info.label);
  });
  listItem.attr('data-theme', 'a');
  var listContent = [];
  listContent.push("<a href='#page'>" + info.label + "</a>");
  listItem.append(listContent.join(""));
  $('#featureList').append(listItem);
  });
});
}

```

```

function addFeature(item) {
  var itemTemplate = getType(item);
  var mapclickHandler = dojo.connect(map, 'onClick', function (evt) {
    //only capture one click
    dojo.disconnect(mapclickHandler);
    var currentDate = new Date();
    var newAttributes = dojo.mixin({}, itemTemplate.prototype.attributes);
    newAttributes.requestdate = Date.UTC(currentDate.getUTCFullYear(),
    currentDate.getUTCMonth(), currentDate.getUTCDate(), currentDate.getUTCHours(),
    currentDate.getUTCMinutes(), currentDate.getUTCSeconds(), 0);
    var newGraphic = new esri.Graphic(evt.mapPoint, null, newAttributes);
    citizenRequestLayer.applyEdits([newGraphic], null, null, function (adds) {
      var query = new esri.tasks.Query();
      var res = adds[0];
      query.objectIds = [res.objectId];

      citizenRequestLayer.queryFeatures(query, function (result) {
        if (result.features.length > 0) {
          var feat = result.features[0];
          var pt = esri.geometry.webMercatorToGeographic(feat.geometry);
          locator.locationToAddress(pt, 100, function (candidate) {
            var address = [];
            var displayAddress;
            if (candidate.address) {
              if (candidate.address.Address) {
                address.push(candidate.address.Address);
              }
              if (candidate.address.City) {
                address.push(candidate.address.City + ",");
              }
              if (candidate.address.State) {
                address.push(candidate.address.State);
              }
              if (candidate.address.Zip) {
                address.push(candidate.address.Zip);
              }
              displayAddress = address.join(" ");
            } else {
              displayAddress = 'No address for this location';
            }
          });
        }
      });
    });
  });
}

```

/*

```

        //display the geocoded address on the attribute dialog.
        $.mobile.changePage("#attributeDialog", null, true, true);
        $('#currentAddress')[0].textContent = displayAddress;
        */
    });
}
});
});
}

function getType(item) {
    var returnType = null;
    $.each(citizenRequestLayer.types, function (index, type) {
        if (type.name === item) {
            returnType = type.templates[0];
        }
    });
    return returnType;
}

//use the geolocation api to get the current location
function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(zoomToLocation, locationError);
    }
}

function locationError(error) {
    switch (error.code) {
        case error.PERMISSION_DENIED:
            console.log("Location not provided");
            break;
        case error.POSITION_UNAVAILABLE:
            console.log("Current location not available");
            break;
        case error.TIMEOUT:
            console.log("Timeout");
            break;
        default:
            console.log("unknown error");
            break;
    }
}

function zoomToLocation(location) {
    var pt = esri.geometry.geographicToWebMercator(new
    esri.geometry.Point(location.coords.longitude, location.coords.latitude));
    map.centerAndZoom(pt, 22);
}

```

```

dojo.ready(init);
</script>
</head>

<body>
<div data-role="page" id="page">
  <!-- header -->
  <div data-theme="a" data-role="header" data-position="fixed">
    <h3>Editing</h3>
    <a href="#" data-role="button" onclick="getLocation();">Location</a>
    <a href="#featureDialog" data-role="button" class="ui-btn-right">Add</a>
  </div>

  <!-- content -->
  <div data-role="content">
    <div id="map"></div>
  </div>
</div>

<!--Attribute Insepctor Dialog-->
<div data-role="page" id="attributeDialog" data-theme="b">
  <!-- header -->
  <div data-role="header" data-nobackbtn="true">
    <h1>Collect</h1>
  </div>

  <div data-role="content" data-inset="true">
    <div>
      <div id="currentAddress" style="font-weight:bold;text-align:center;width:100%;"></div>
    </div>
    <div id="attributeDiv"></div>
    <a href="#mapPage" data-role="button">Finish</a>
  </div>
</div>

<!--Feature Collection Dialog-->
<div data-role="page" id="featureDialog" data-theme="b">
  <div data-role="header" data-position="fixed">
    <h1>Feature Type</h1>
  </div>
  <div data-role="content">
    <ul id="featureList" data-role="listview" data-inset="true"></ul>
  </div>
</div>
</body>
</html>

```