

Nærværende rapport om de grønlandske kommuners IT system - Winformatik, er en del af det analyse materiale der vil blive inddraget i "Beslutningsgrundlag for igangsættelse af fase 2 af Pisariillisaaneq-projektet"

Analyse af Winformatik

Arbejdspakke 1.9.1.3.2 Snitflader
Winformatik

Torben Porsgaard

Indholdsfortegnelse

1	Versionering	4
2	Omfang.....	4
2.1	Begrænsninger	4
2.2	Læsers forudsætninger.....	4
3	Indledning.....	5
4	Winformatiks historie	6
5	Overblik over Winformatik	6
5.1	Databasen	7
5.1.1	Afhængigheder på databaseniveau	7
5.2	Afhængigheder på tværs af Moduler	8
5.2.1	Analyse eksempel af et modul – Social - Pension	10
5.3	Integration og eksterne afhængigheder	12
5.4	Design og arkitektur.....	14
5.4.1	Arkitektur.....	15
5.4.2	Design.....	15
5.5	Borgerservice	18
5.6	Folkeregister	19
5.7	Brugerstatistik.....	19
6	Dokumentation	20
6.1	Kravspecifikation.....	21
6.2	Design og arkitektur dokumentation	21
6.3	Teknisk dokumentation og kode dokumentation.....	22
6.3.1	Teknisk dokumentation	22
6.3.2	Kode dokumentation	23
6.4	Brugervejledning.....	23
6.5	Vurdering af systemets dokumentation	23
7	Sikkerhed	24
7.1	Sikkerhedsmodel	24
7.2	Teknisk sikkerhed	25
7.3	Fysisk sikkerhed	26
7.4	Backuprutiner	26
7.5	Vurdering af sikkerhedsmodellen og fysisk sikkerhed	26
8	Udviklingsmetoder og værktøjer.....	26

8.1	Udviklingsprocesser.....	26
8.2	Systemudrulning.....	27
8.3	Udviklingsværktøjer.....	28
8.4	Test.....	28
8.4.1	Udviklings-, test- og produktionsmiljø.....	29
8.5	Vurdering af udviklingsmetoder og værktøjer.....	29
9	Logning.....	30
9.1	Teknisk og performance logning.....	30
9.2	Brugerhandlinger der logges.....	30
9.3	Historisk data.....	31
9.4	Lovmæssige krav.....	31
9.4.1	Lov om offentlige myndigheders registre.....	31
9.4.2	Sikkerhedsbekendtgørelsen og sikkerhedsvejledningen.....	32
9.4.3	CPR.....	32
9.4.4	Arkivloven.....	32
9.4.5	Overholdelse af gældende lovkrav.....	32
9.5	Vurdering af systemets logning.....	34
10	Teknik og infrastruktur.....	34
10.1	Virtualisering.....	35
10.2	Latency.....	36
10.2.1	Postback strategi.....	36
10.2.2	Virtualisering.....	36
10.3	Road-map for teknik og kode.....	37
10.3.1	RDS (Remote Data Services).....	37
10.3.2	Klassisk ASP.....	37
10.4	Krav til platform.....	38
10.5	Vurdering af teknik og infrastruktur.....	38
11	Brugervenlighed.....	39
11.1	Idéen bag brugergrænsefladen.....	39
11.2	Sprogversionering.....	39
11.3	Vurdering af brugergrænsefladen.....	40
12	Winformatik i et åbent marked.....	40
12.1	Winformatik som ERP-system.....	41
12.1.1	Økonomistyring.....	41

12.1.2	Forretningsprocesser	41
12.1.3	Ledelsesinformation	41
12.1.4	Organisationsunderstøttelse	41
12.1.5	Kimik iT's vurdering	42
13	Bidragssydere	43
14	Bibliografi	44
15	Begreber	45
16	Figuroversigt	47
17	Index	48

1 Versionering

Dato	Ændring	Udført af
13. marts 2013	Dokument oprettet.	TP
9. maj 2013	Ændringer ifm. første review af Kimik iT	TP
13. maj 2013	Korrektur	TP
23. maj 2013	Ændringer ifm. andet review af Kimik iT, AHOF, LGM og OPK	TP
11. juni 2013	Indskærpning af CPR lovgivning i 9.4.3	TP

2 Omfang

Denne rapport har overordnet set til formål at afdække afhængigheden af Winformatik og undersøge, hvordan systemet ud fra et design- og arkitekturmæssigt synspunkt er opbygget. Rapporten skal indgå som en del af den samlede analyse og vil dermed også blive inddraget i "Beslutningsgrundlag for igangsættelse af fase 2 af Pisariillisaaneq-projektet".

2.1 Begrænsninger

Formål, omfang og begrænsninger for denne rapport er som udgangspunkt at finde i arbejds pakken: Arbejdspakke 1.9.1.3.2 Snitflader Winformatik.

Ud over de rammer og begrænsninger der er beskrevet i arbejds pakken, er følgende emner ikke dækket i indeværende rapport, da de er fravalgt i denne omgang:

- Der er ikke foretaget interviews af enkelte brugere
- Der er ikke foretaget interviews af repræsentanter for kunderne
- Der er ikke foretaget en komplet gennemgang af de enkelte moduler i forhold til funktionalitet
- Der er ikke foretaget test/fejlfinding
- Der er ikke foretaget nogen form for vurdering af, hvorvidt enkelte moduler lever op til kundens krav
- Juridiske overvejelser tager udgangspunkt i almindelig tilgængeligt materiale
- Der er ikke konsekvent forsøgt at verificere informationer indsamlet fra leverandøren ved tredjepart
- Alle vurderinger, der udelukkende er foretaget af leverandør, er markeret som sådan
- Der er i rapporten ikke draget nogen form for konklusioner på baggrund af det skrevne
- I det omfang der er foretaget vurderinger, er det tilstræbt, at disse er så objektive som muligt, og med det formål at tillægge rapporten en tolkning af den information der allerede er givet - oftest for at fortætte de til tider lidt tekniske informationer til noget mere almindeligt forståeligt.

2.2 Læsers forudsætninger

Det har været hensigten at lægge rapportens tekniske niveau, så de fleste vil kunne både læse og forstå de væsentligste dele af rapporten. Dog er informationer af mere teknisk karakter inkluderet hvor påkrævet.

For at lette læsbarheden er de fleste forkortelser defineret ved første anvendelse, desuden er begreber der ikke i sig selv er deskriptive, beskrevet i afsnit 0

Begreber. Der findes i afsnit 17 et index over anvendte begreber.

Emner af mere teknisk karakter er kort forklaret for på den måde at sikre at informationer og/eller vurderinger forstås korrekt, entydigt og i den rette kontekst.

Der er inkluderet en bibliografi (anvendt standard IEEE Citation <http://www.ieee.org/documents/ieeecitationref.pdf>), der anvendes til de referencer der bruges i dokumentet. Dette er primært elektroniske dokumenter som bl.a. dokumentation fra Kimik iT. Det anbefales, at læser har adgang til disse dokumenter.

3 Indledning

I bestræbelserne for at indføre et fællesoffentligt ERP-system i Grønland, er en analyse af de systemer den offentlige administration anvender i dag et afgørende skridt i forhold til at kunne træffe fornuftige beslutninger om fremtidens IT systemer i Grønland.

De erfaringer, gode som dårlige, der de sidste ~25 år har ophobet sig om udvikling, anvendelse og drift af et system som Winformatik, skal naturligvis anvendes til bedre at kunne se frem mod fremtidens IT systemer - uanset hvad systemet hedder og hvem der nu engang måtte levere det.

Nærværende rapport tager udgangspunkt i Winformatik, systemet der stort set har været de facto standard økonomi/sags-system for de grønlandske kommuner de sidste ~25 år. Målsætningen har været at udarbejde et generelt overblik over systemets aktuelle tilstand. Dette overblik skulle dække så bredt et spekter som muligt, således at både tekniske og procesrelaterede aspekter ved systemet blev gennemgået. Desuden har det været vigtigt at identificere systemets afhængigheder, både eksterne og interne, for på den måde at kunne adskille og isolere risici i forbindelse med en evt. helt eller delvist systemskift.

Det har været vigtigt at selve gennemgangen af Winformatik var så objektiv som muligt. Derfor findes der heller ingen deciderede konklusioner i rapporten, der drages heller ingen konkrete erfaringer – og det på trods af, at et erklæret mål må være at høste erfaringer.

Kort sagt skal nærværende rapport blot danne baggrund for at man lettere kan erkende erfaringer, udlede konklusioner og i sidste ende tage gode beslutninger om hvordan fremtidens IT understøttelse i Grønland ser ud.

4 Winformatiks historie

Winformatik er udsprunget på baggrund af en konkret analyse af de behov de grønlandske kommuner stod over for i slutningen af firserne, da anvendelsen af IT i det grønlandske samfund var en realitet og nødvendighed.

Med udgangspunkt i de opgaver der skulle løses, blev Winformatik trinvist opbygget og udviklet. Udviklingen foregik i tæt samarbejde med fokusgrupper fra kommunerne og med de grønlandske kommuner som eneste målgruppe for systemet. Således burde både processer og anvendt terminologi være tæt knyttet til kommunernes. Efterfølgende har systemet relativt smertefrit gennemgået en række forandringer foranlediget af indførslen af selvstyret, et årtusindskifte og en omfattende kommuneomlægning.

Winformatik ejes af kommunerne, og det har til alle tider har været disse der som eneste målgruppe og systemejere suverænt har bestemt hvor mange ressourcer der har været til rådighed i relation til udviklingen af systemet og de hertil knyttede rådgivnings- og konsulenttydelser.

Systemet var baseret på teknologier, der på daværende tidspunkt både var fremsynede og moderne. Baseret på en relationel database fra Oracle, startede produktet med klienter udviklet i Oracle Forms. Oracle Forms blev dog løbende udskiftet til fordel for et web interface, baseret på Microsoft produkter (ASP). Udskiftningen blev indledt omkring 1997 og forløb over en række år. Efterfølgende har systemet været under stadig udvikling, og i dag er systemet på vej til at være fuldt udviklet i ASP.NET. Omskrivningen fra ASP til ASP.NET forventes at være færdig i 2014.

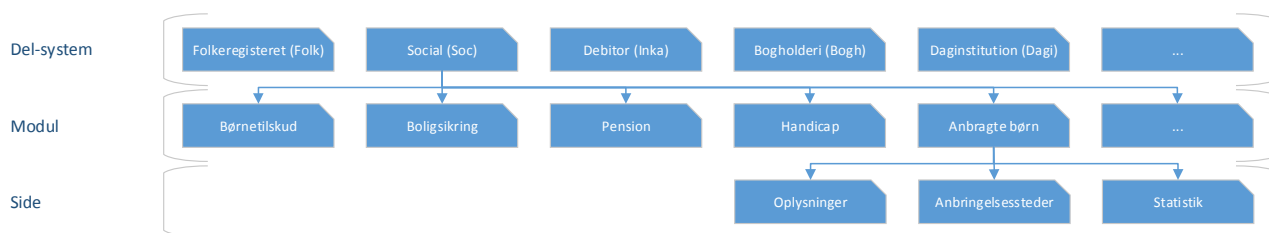
Som et overordnet mantra for Kimik iT, i forhold til at udvikle IT løsninger til det grønlandske samfund, vægtes kontinuitet, simplicitet, driftbarhed og økonomisk tilgængelige løsninger. Det vægtes således, da det er her udfordringerne i langt højere grad ligger end i den tekniske implementering. Årsagen hertil skal findes i en årelang erfaring med at udvikle systemer til kommunerne i Grønland.

5 Overblik over Winformatik

Overordnet set er Winformatik bygget op omkring et side begreb. Det skal forstås således, at de forskellige opgaver/funktioner er samlet på én side, der så løser den ene opgave. Brugeren har mulighed for at gå til en given side ved at vælge den i en serie af valgmener eller indtaste sidenummeret og gå direkte til siden. Mere information om selve brugergrænsefladen kan læses i afsnit 11, hvor også anvendte begreber defineres.

Winformatik er logisk set opdelt i tre niveauer: Del-system, Modul og Side. Således har systemet et antal del-systemer, der hver i sær har et antal moduler. Disse moduler har så et antal sider, der udgør modules funktionelitet. Opdelingen i moduler er ikke helt så tydelig (og er heller ikke konkret til stede i systemet) som opdelingen i del-systemer og sider.

I Figur 1 ses den generelle opbygning af systemet. I figuren er det kun en delmængde af del-systemerne og modulerne der vises – konkret er det Social/Anbragte børn der er vist til fulde.



Figur 1 Generel opbygning af Winformatik

Der findes desuden en anden opdeling der kaldes Kategori. Denne opdeling er udelukkende anvendt i adgangskontrol sammenhænge, hvor den giver mulighed for at tildele rettigheder til en delmængde af et modul, som kaldes en kategori. En kategori kunne eksempelvis indeholde alle sider undtagen dem der kunne foretage udbetaling til borgere.

Når man designer databaser, er det typisk ikke anbefalet praksis at lade brugergrænsefladen diktere tabellerne i databasen. I Winformatiks tilfælde er der derfor ikke et 1-til-1 forhold mellem sider og tabeller i relation til detaljesiderne. Det er ofte den logiske domænemodel af problemområdet, der er afgørende for hvordan databasen designes. Desuden er der naturligvis altid en række databasetekniske forhold, der også spiller en væsentlig rolle, når databasen designes.

5.1 Databasen

Afvendelsen af en relationel database har, siden ideen om det IT system der sidenhen skulle blive til Winformatik, været en vigtig og central del af systemet. Og denne database har lige siden starten været en Oracle database. I dag kører systemet på den nyeste hovedversion; Oracle 11g [1].

For hver instans (en komplet og samlet installation), findes der én database, der rummer alle data i de tilhørende del-systemer. Der er til databasen defineret en række schemas, der hver især dækker over en delmængde af de tabeller, views og procedurer (defineret i packages), der er i databasen. Hvert schema har en databasebruger tilknyttet, der deler navn med dette schema.

Databasens opbygning er nært forbundet med disse schemas, der i øvrigt er navngivet med 3-4 bogstavsforkortelser for de del-systemer, de dækker over eksempelvis: Budg, Bogh, Folk, Info, Led, Lon, Udd, Skat osv. Der er både fordele og ulemper (kompleksitet og vedligehold) ved de mange schemas, men på trods af ulemperne, giver de en vis adskillelse, som er beskrevet nærmere i næste afsnit.

Anvendelsen af stored procedures samlet i packages er en fremgangsmåde der i højere grad anvendes i systemet. De umiddelbare fordele er performance, der altid vil være bedre for statisk SQL end dynamisk, da det giver databasen mulighed for at lave en execution plan. Forudsætningen er, at forespørgslen (query) er den samme, og det er kun parametrene, der ændrer sig. En anden fordel er, at disse procedurer kan samles i en package, som man så kan give grants til på tværs af del-systemer. Ulemperne er som udgangspunkt at forretningslogik i stor udstrækning vil være indlejret i databasen.

Der er udarbejdet et dokument ved navn Database_conventions [2], der beskriver retningslinjer for opbygning og navngivelse af database elementer.

5.1.1 Afhængigheder på databaseniveau

I Winformatik databasen er der for hvert del-system defineret ét schema. I udgangspunktet er der således ingen afhængighed mellem del-systemerne på database niveau. Men da det naturligvis er nødvendigt for f.eks. social del-systemet at slå op i folkeregister del-systemet, er der defineret såkaldte grants, så andre schemas har adgang til udvalgte dele (tabeller, views, packages) af det konkrete schema. I realiteten betyder dette, at der for hovedparten af de udvalgte dele er adgang til f.eks. folkeregister del-systemet fra langt de fleste andre schemas. Dette betyder, at der på trods af denne opdeling, i form af adskilte schemas, stadig i praksis er en relativ stor afhængighed på tværs af del-systemerne – set ud fra en database synsvinkel.

En komplet liste over grants mellem de forskellige schemas i databasen (del-moduler) kan ses i dokumentet; Winformatik grants.xlsx [3]

5.2 Afhængigheder på tværs af Moduler

Som nævnt i afsnit 5.1.1 ovenfor giver oversigten over database grants ikke nødvendigvis et klart billede over, hvilke afhængigheder der er på tværs af del-systemer og moduler. Selvom der er grants til et modul, er der ikke nødvendigvis en logisk afhængighed til det modul. Modsat kan man dog ikke sige, at der kan være afhængighed uden database grants, i det database grants skal ses som en bruttoliste over mulige logiske afhængigheder – som så udgør nettolisten.

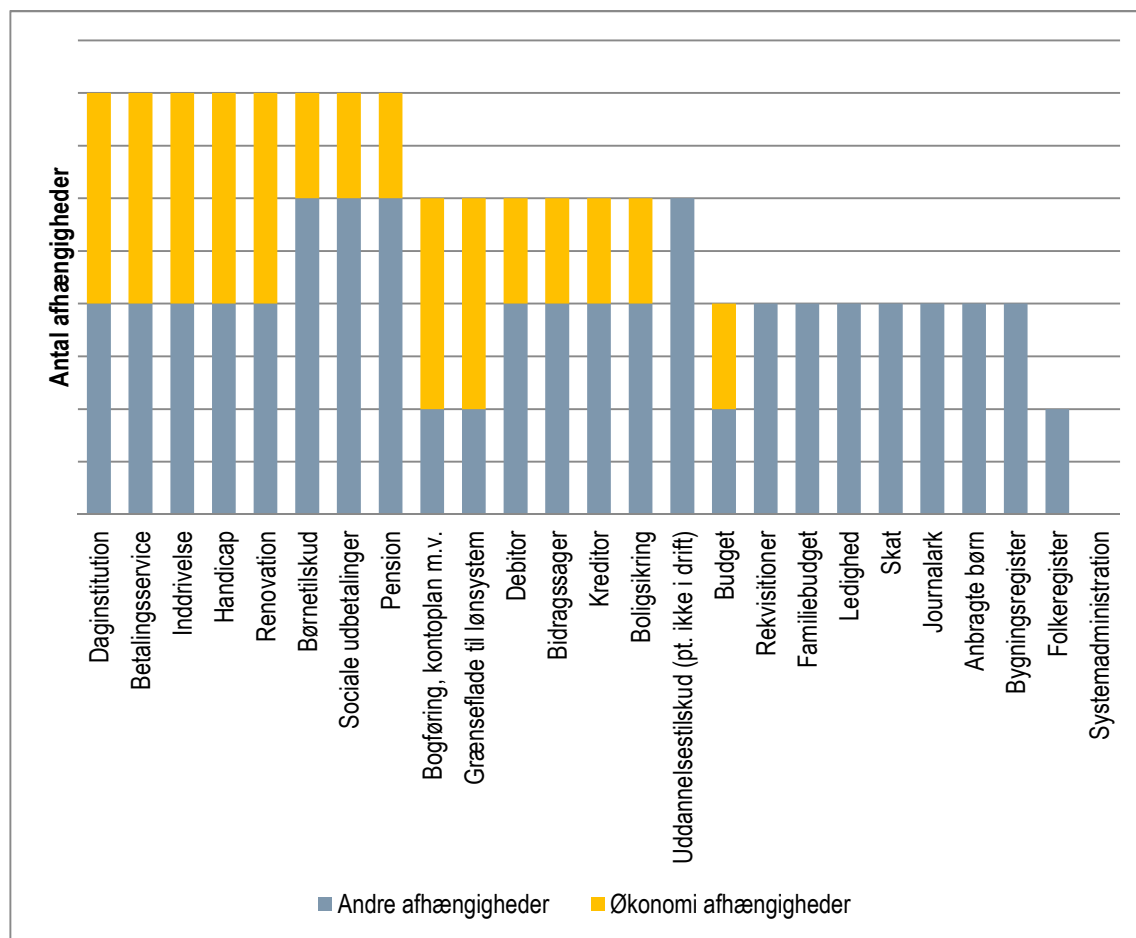
I Figur 2 ses en tabeloversigt over logiske afhængigheder. De syv moduler, der er vist på højkant i kolonnerne er dem, der er afhængigheder til fra andre moduler. De er ligeledes markeret med kursiv i modul-kolonnen.

Delsystem	Modul	Bogføring	Budget	Folkeregister	Systemadmin.	Debitor	Ledighed	Skat
Bogh	<i>Bogføring, kontoplan m.v.</i>		X		X	X		
Bogh	Rekvitioner			X	X			
Bosc	Familiebudget			X	X			
Budg	<i>Budget</i>	X			X			
Dagi	Daginstitution	X		X	X	X		X
Folk	<i>Folkeregister</i>				X			
Info	<i>Systemadministration</i>							
Inka	<i>Debitor</i>	X		X	X			
Inka	Bidragssager			X	X	X		
Inka	Betalingservice	X			X	X		
Inka	Inddrivelse	X		X	X	X		
Krdi	Kreditor	X		X	X			
Led	<i>Ledighed</i>			X	X			
Lon	Grænseflade til lønsystem	X			X	X		
Skat	<i>Skat</i>			X	X			
Soc	Børnetilskud	X		X	X			X
Soc	Sociale udbetalinger	X		X	X		X	
Soc	Boligsikring	X		X	X			
Soc	Journalark			X	X			
Soc	Pension	X		X	X			X
Soc	Handicap	X	X	X	X			
Soc	Anbragte børn			X	X			
Tekn	Renovation	X		X	X	X		
Tekn	Bygningsregister			X	X			
Udd	Uddannelsestilskud (pt. ikke i drift)			X	X			X

Figur 2 Et overblik over modulers afhængigheder

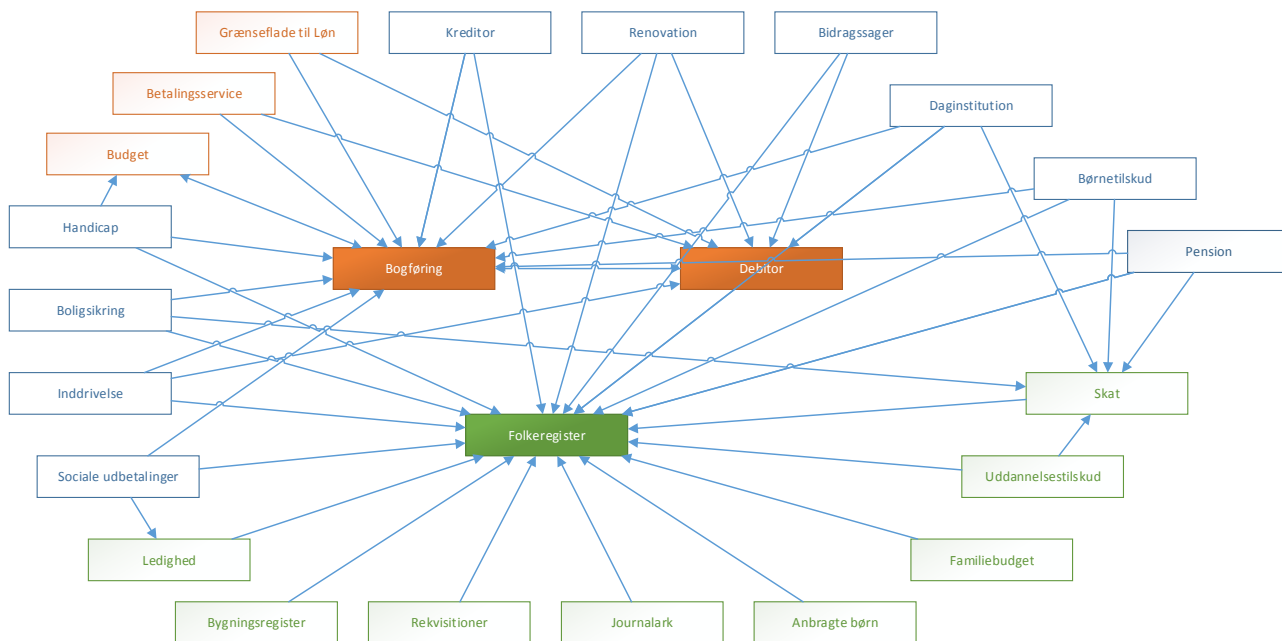
Ud fra fordelingen af markeringer, fremgår det tydeligt, at det primært er bogføring, debitor, folkeregister og systemadministration, der findes afhængigheder til. Der er ingen indikation af, hvor mange sider der er afhængigheder

fra for hvert modul, og hvor i afhængigheden består. Derfor er det ikke umiddelbart muligt at afgøre, hvor stor den markerede afhængighed er.



Figur 3 Modulernes afhængigheder

Hvis de enkelte moduler skal kategoriseres som "økonomimoduler" eller "sagsmoduler", kan Figur 4 anvendes (alternativt Figur 3). Figur 4 viser en farvekodet (for at indikere primær afhængighed) oversigt over modulerne og deres afhængigheder. Figuren viser moduler med primær afhængighed til folkeregister med grøn kant og moduler med primær afhængighed til økonomimoduler med orange kant. Moduler, der har afhængigheder på tværs, er markeret med blå kant. Det giver desværre ikke et entydigt og klart billede, men er det nærmeste man kommer en sondring mellem de to begreber. En entydig kategorisering forudsætter en ligeledes entydig definition af de to kategorier.



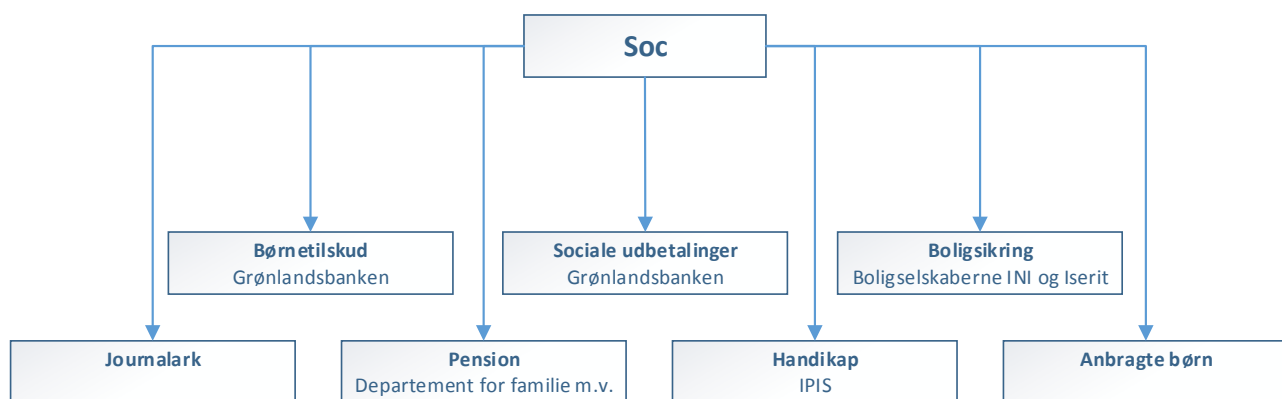
Figur 4 Diagram over interne afhængigheder, farvekodet og udlagt i forhold til primær afhængighed (System administration udeladt)

En analyse af de enkelte moduler og siderne under dem, ville kunne afgøre 'størrelsen' af afhængigheden. En sådan analyse giver kun god mening i forbindelse med et konkret behov for at specificere snitfladerne for modulet. Derfor er der i afsnit 5.2.1 forsøgt at give et delvist eksempel på en sådan analyse.

5.2.1 Analyse eksempel af et modul – Social - Pension

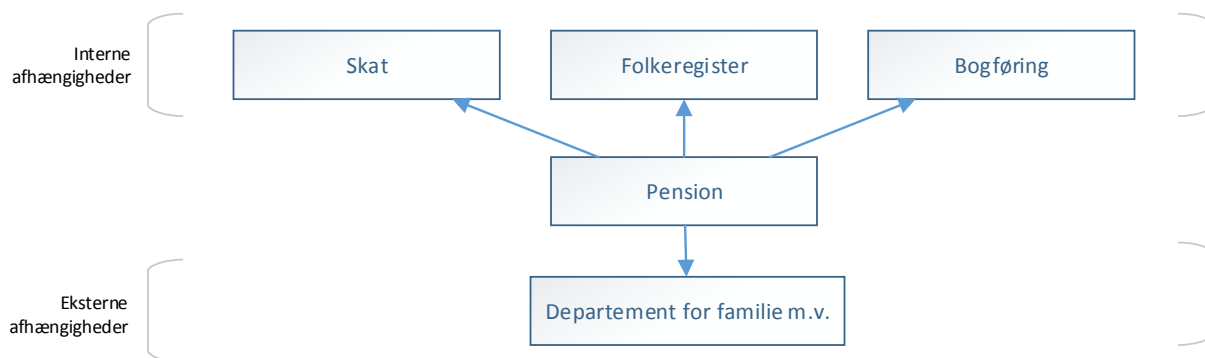
Del-systemet Social er et af de systemer, der har ganske mange moduler, og disse moduler har ligeledes mange integrationer. For interne integrationer er det primært folkeregister og bogføring der refereres til. For eksterne er der flere forskellige referencer - både til økonomiske og sags relaterede systemer.

Se Figur 5 for eksterne integrationer og afsnit 5.3 for yderligere detaljer.



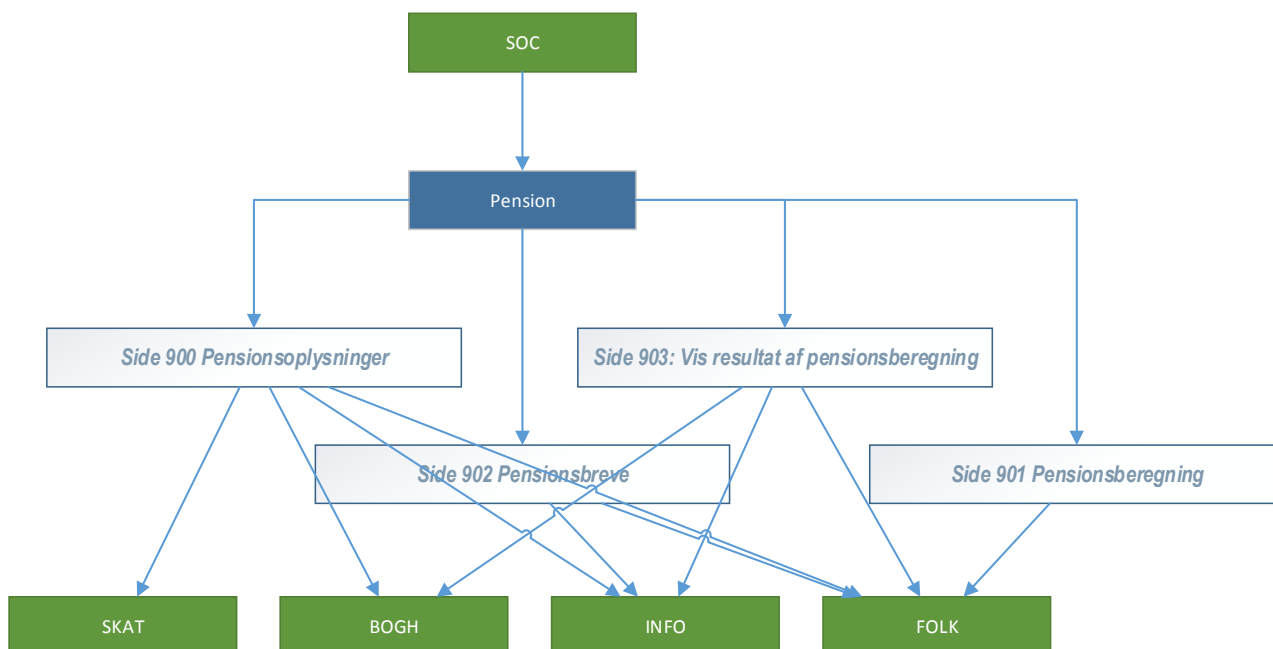
Figur 5 Social del-systemets moduler og disses eksterne afhængigheder

I Figur 6 ses såvel interne som eksterne afhængigheder for modulet Pension i del-system Social. Administrationsmodulet er udeladt.



Figur 6 Pension modulets afhængigheder

I de efterfølgende afsnit gennemgås modulets enkelte sider med de afhængigheder de har, og hvori afhængigheden består. Figur 7 viser modulets enkelte siders afhængigheder af del-systemer.



Figur 7 Pensions interne afhængigheder på side-niveau

5.2.1.1 Side 900 Pensionsoplysninger

Delsystem Folk:

Navn, adresse, fødselsdato, ægtefælle, civilstand.

Delsystem Info:

Via folkeregister-adressen findes en lokalitet (by/bygd). Og fra lokaliteten findes et kontaktpunkt (kaldet Område på side 900). Kontaktpunkter er defineret i delsystem Info.

Delsystem Skat:

Oplysninger om skattepligtig indkomst (år minus 2) for pensionisten og evt. ægtefælle (hentes fra Skattestyrelsen til Delsystem Skat en gang om måneden).

Delsystem Bogh:

Oplysninger om udbetalt pension (år minus 2) for pensionisten og evt. ægtefælle (hentes til pensionssystemet ved batch-kørsel en gang om året).

Se endvidere Beregningsrutinen nedenfor.

5.2.1.2 Side 901 Pensionsberegning

Se Beregningsrutinen nedenfor.

5.2.1.3 Side 902: Pensionsbreve

Bestillingsside samt rapport udviklet med Crystal Reports.

Delsystem Info:

Kontaktpunkter: Navn, adresse, telefon m.m.

Liste af printere, der kan udskrives til.

Delsystem Folk:

Pensionisters navne + adresser.

5.2.1.4 Side 903: Vis resultat af pensionsberegning

Delsystem Folk:

Pensionisters navne + lokaliteter.

Delsystem Info:

Kontaktpunkter.

Delsystem Bogh:

Oplysning om faktisk bogført pension – til kontrol sammenlignet med beregnet pension.

5.2.1.5 Beregningsrutinen

Stored procedure findes i pakken pension under schema soc.

Anvender satser, der modtages fra Familiedepartementet i form af regneark og indlægges i pensionssystemet (tabel: PENS_RATE) ved ændringer af satserne.

Delsystem Folk:

Oplysning om fødselsdato, civilstand og ægtefælle.

5.3 Integration og eksterne afhængigheder

Winformatik har en række afhængigheder til eksterne systemer. Typisk tilgås sådanne systemer ved, at der indhentes data fra dem eller aktivt leveres data til dem. Denne integration kan foregå på mange måder, lige fra manuelle udtræk fra databasen til integration via en webservice.

Nedenfor i Figur 8 ses et overblik over, hvilke moduler der anvender integration til hvilke eksterne snitflader, og hvordan integrationen til snitfladen foregår.

Delsystem	Modul	Eksterne	Integration	Retning *	Form	Frekvens	Udløst af
Bogh	Bogføring, kontoplan m.v.	Grønlandsbanken	Bankoverførsler	➡	Upload af fil efter login på eksterens hjemmeside	Efter behov	Operatør
Bogh	Bogføring, kontoplan m.v.	Grønlandsbanken	FI-indbetalinger (Fælles Indbetalingskort)	⬅	Download af fil efter login på eksterens hjemmeside	Efter behov	Operatør
Bogh	Rekvitioner						
Bosc	Familiebudget						
Budg	Budget						
Dagi	Daginstitution						
Folk	Folkeregister	Det Centrale Personregister (CPR)	Ændringer i CPR-abonnementer	➡	Send filer med Secure FTP	Efter hver hverdag	Scheduled task
Folk	Folkeregister	Det Centrale Personregister (CPR)	Daglig ajourføring	⬅	Hent filer med Secure FTP	Efter hver hverdag	Scheduled task
Info	Systemadministration						
Inka	Betalingservice	NETS (PBS)	Hent aftaler og betalinger	⬅	Download af fil efter login på eksterens hjemmeside	Pr. måned	Operatør
Inka	Betalingservice	NETS (PBS)	Send opkrævninger	➡	Upload af fil efter login på eksterens hjemmeside	Pr. måned	Operatør
Inka	Bidragssager						
Inka	Debitor						
Inka	Inddrivelse	Inddrivelsesmyndigheden	Send fordringer og nedskrivninger	➡	Upload af fil efter login på eksterens hjemmeside	Fordringer: ca pr. måned. Nedskrivninger: ca pr. uge	Operatør
Inka	Inddrivelse	Inddrivelsesmyndigheden	Modtage afregninger	⬅	Fil sendt med email fra eksteren	Pr. måned	Ekstern
Inka	Inddrivelse	Inddrivelsesmyndigheden	Hent afviste fordringer samt afstemninger	⬅	Download af fil efter login på eksterens hjemmeside	Pr. måned	Kimik IT
Krdi	Kreditor	Grønlandsbanken	Bankoverførsler	➡	Upload af fil efter login på eksterens hjemmeside	Efter behov	Operatør
Krdi	Kreditor	Inddrivelsesmyndigheden	Modregning i offentlige udbetalinger	➡ ⬅	Kommunikation med eksterens webservice. https://aka.nanoq.gl/etaxOIO	Efter behov	Operatør
Led	Ledighed	Grønlands Statistik	Ledighedsdata		Winformatik sender email	Pr. måned	Operatør
Led	Ledighed	Piarsersarfiit	Ledighedsdata		Winformatik sender email	Pr. måned	Kimik IT

Lon	Grænseflade til lønsystem	Det centrale lønsystem	Bogføring af lønkørsel	←	Databaselink	Efter behov	Operatør
Lon	Grænseflade til lønsystem	Det centrale lønsystem	Udtræk af løndata til regneark	←	Databaselink	Efter behov	Operatør
Skat	Skat	Skattestyrelsen	Skattepligtige social ydelser via fil	→	Upload af fil efter login på eksterns hjemmeside (sulinal)	Pr. år	Operatør
Skat	Skat		Skattepligtige indkomster, manuel hentet fil (af Kimik iT) fra eSkat, efterfølgende upload i 4 * Winformatik	←	Download af filer efter login på eksterns hjemmeside, efterfulgt af upload i 4 * Winformatik	Pr. måned	Kimiik iT
Soc	Anbragte børn						
Soc	Boligsikring	Boligselskaberne INI og Iserit	Oplysning om lejere, husleje-størrelse m.m.	←	Email med vedhæftet fil	Pr. måned	Ekstern
Soc	Boligsikring	Boligselskaberne INI og Iserit	Oplysning om tildelt boligsikring	→	Winformatik sender email	Pr. måned	Operatør
Soc	Børnetilskud	Grønlandsbanken	Udbetalinger	→	Upload af fil efter login på eksterns hjemmeside	Pr. måned	Operatør
Soc	Handicap	IPIS	Handicap-oplysning	→	Download af fil	Efter behov	Ekstern logger på Winformatik via https
Soc	Journalark						
Soc	Pension	Departement for familie m.v.	Pensionssatser	←	Email med vedhæftet fil (regneark)	Pr. år (cirka)	Ekstern
Soc	Sociale udbetalinger	Grønlandsbanken	Udbetalinger	→	Upload af fil efter login på eksterns hjemmeside	Efter behov	Operatør
Tekn	Bygningsregister						
Tekn	Renovation	Affaldsvægte (ScanVægt) i Sermersooq	Oplysning om afleveret affald - dannelsen af regninger	←	Fil anbragt i aftalt fil-mappe	Efter behov	Ekstern
Udd	Uddannelsetilskud (pt ikke i drift)	Grønlandsbanken	Udbetalinger				

Figur 8 Overblik over eksterne afhængigheder

Retning*:

→ Kommune til ekstern (der sendes data ud af systemet)

← Ekstern til kommune (der hentes data fra ekstern kilde)

Winformatik har ingen eksterne snitflader, der eksponerer data for andre systemer (med undtagelse af Handikap, hvor IPIS henter data på kommunernes servere). Der er udviklet en webservice til ét enkelt modul; Tids- og faktureringsystem der dog udelukkende har været anvendt internt i Kimik iT. Borgerselvbetjeningsserveren, der er implementeret i én kommune, er påtænkt som værende dér hvor evt. fremtidige webservices implementeres.

5.4 Design og arkitektur

Winformatik er lavet som en webapplikation, hvilket vil sige at klienten er brugerens web-browser der via en netværksforbindelse (internetforbindelse) kommunikerer med en web-server. Web-serveren persisterer data i en database på en database-server der er forbundet til web-serveren. Dette bliver normalt kaldt en client-server arkitektur og er en meget udbredt arkitektur for web applikationer.

5.4.1 Arkitektur

Når man taler om en 1, 2 eller n-tier (3 eller derover) model, kan man anvende begrebet i forskellige sammenhænge. Begrebet dækker over adskillelse af de forskellige dele af systemet – hvor dele kan være platforme, moduler eller lign. Eksempelvis kan Microsoft Word ansues som en 1-tier applikation, da både brugergrænseflade, forretningslogik og data-håndtering foregår på samme platform – brugerens pc. Typisk siges det, at fordelene ved et n-tier system er at det ofte forbedrer ydeevne og skalerbarhed. Ved at systemet er opdelt i adskilte enheder, lader disse sig lettere udskifte, opgradere eller erstatte.

I en deployment sammenhæng tænkes der her på fysiske enheder/platforme, og i den forstand kan Winformatik ses som en 3-tier model; brugergrænseflade (webbrowser), forretningslogik (webserver), data (databaseserver).

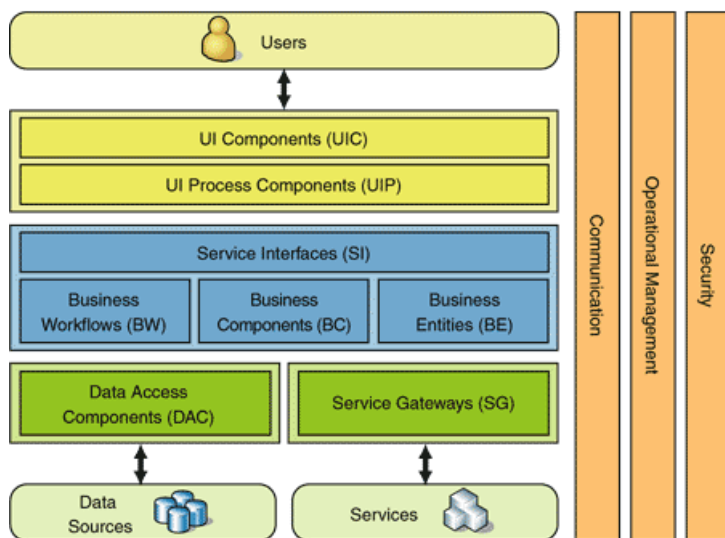
Et, på det seneste, ofte anvendt begreb inden for enterprise arkitektur er Service Oriented Architecture (SOA). SOA er en software arkitektur der baserer sig på løst koblede software enheder, såkaldte services, der hver især kun kender og løser én specifik opgave i den pågældende enterprise. Dette gøres ved at udstille en servicekontrakt, således de forskellige klient applikationer kan tilgå disse services uanset hvilken teknologi og platform de måtte anvende.

Service Oriented Architecture (SOA) var ikke en arkitektur, der blev prioriteret da Winformatik projektet startede. Selvom principperne bag SOA går tilbage til 60'erne, var det først i 1996 (af teknologi analyseinstituttet Gartner) der blev sat navn på SOA, og på daværende tidspunkt var arkitekturen for Winformatik fastlagt. Winformatiks arkitektur er mere rettet mod at løfte en konkret opgave frem for at være åben og modulariseret. I kraft af den valgte platform og teknologi (.NET, MS Server 2003, Oracle) vil systemet dog være velegnet til en fremtidig implementering af en SOA arkitektur, såfremt kunden som systemejer måtte ønske det.

5.4.2 Design

Fra en designmæssig synsvinkel, hvor der ses på de enkelte software elementer, systemet er opbygget af, samt hvorledes de er sammensat, vil man ofte opbygge systemet af forskellige lag (der udmøntes i forskellige assemblies). Disse lag kan i princippet betragtes som tiers i en n-tier model, som beskrevet ovenfor. Når der tales om lag, er én tier blot sammenlignelig med et lag. Et lag er ikke (som en tier beskrevet under Arkitektur) en fysisk enhed/platform men en adskilt softwarekomponent. Formålet er, som ovenfor, at adskille ansvar, opgaver og dermed viden om andre enheder/lag for at forbedre/sikre flexibility, maintainability, reusability og scalability. Ofte vil man desuden adskille de forskellige lag ved at definere interfaces, der er en slags kontrakt/skabelon mellem den, der udstiller en funktion, og den der anvender den.

Der findes best-practices på området. Der er dog mange nuancer inden for disse. For en webapplikation som Winformatik, vil det nærmeste, man kan komme på en sådan anbefaling, være et mønster der baseres på en tre-lags arkitektur [4] som den, der er vist i Figur 9. Tre lags service applikationen, som figuren viser, består af 3 lag; presentation (gul), business (blå) og data (grøn).



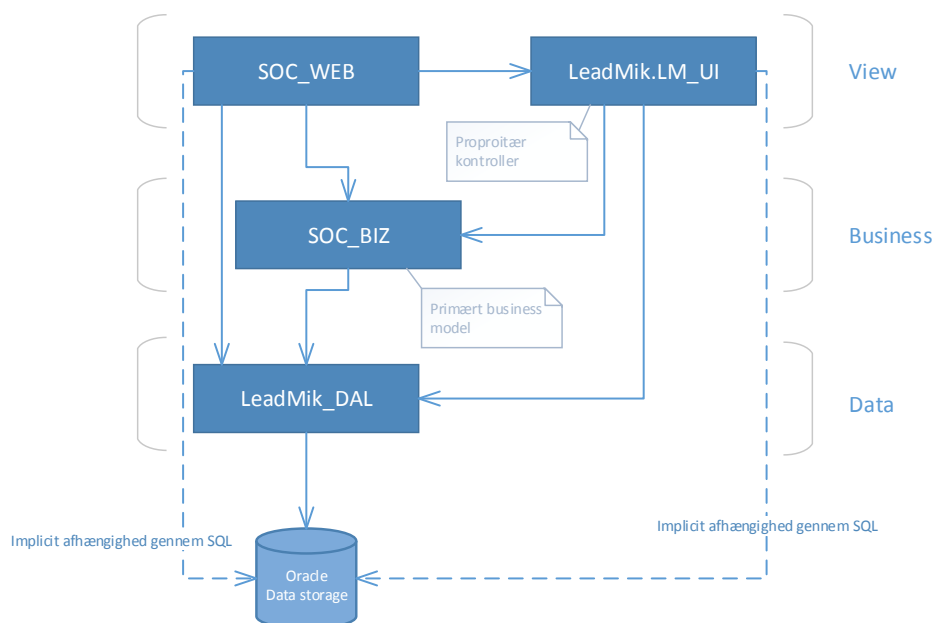
Figur 9 Tre lags service applikation [4]

Som udgangspunkt har det ikke været et erklæret mål fra Kimik IT's side at følge ovenstående mønster. Den nævnes og anvendes her udelukkende som en reference til best-practice på området. I de situationer hvor den anvendte praksis afviger fra ovenstående mønster, er det derfor ikke en fejl eller en forglemmelse.

Årsagen til afvigelsen kan meget vel være, at man gik fra Oracle Forms til en web løsning, og at nogle af de principper der anvendtes i Oracle Forms er arvet af den udviklede web løsning. Da transformationen blev foretaget, var man i dialog med Teknologisk Institut for at få deres input til, hvordan den nye løsning skulle designes.

I nogle af de seneste tilføjelser til systemet, er Object Relational Mapping (ORM) teknologi blevet anvendt til at tilgå databasen. Dette erstatter den implicitte afhængighed til Oracle SQL med en logisk model, der abstrahere dette bort.

Figur 10 nedenfor illustrerer et forsimplet diagram af Winformatiks normale lagdeling/opbygning (navnene er respektive namespaces).



Figur 10 Forsimpleret lagdelingsdiagram med social del-modul som eksempel

Generelt kan det siges, at der normalt kun skal være afhængigheder ned til næste lag, og dermed ingen afhængigheder der 'springer over' lag. Ej heller bør der være afhængigheder op til de øvre lag. Det eneste assembly, der er "rent" i den forstand, at det ingen viden har om andre lag/assemblies, er Data access laget - naturligt nok fordi det ligger nederst. I den anden ende af "skalaen" er det i særdeleshed UI komponenterne, der har viden og dermed implicit afhængighed helt ned til data access laget og sågar anvender Structured Query Language (SQL). Denne afhængighed er primært af sproget SQL, og dermed relationelle databaser, og kun i nogen grad af at det er en Oracle database. Pt. understøtter data access laget integration til Oracle og Microsoft SQL Server. Understøttelse af andre database leverandører er forholdsvis overkommeligt at implementere.

Et eksempel på et potentielt problem med uhensigtsmæssige afhængigheder kunne være udskiftning af den eksisterende database server med en helt anden data storage teknologi. I så fald ville UI komponenterne skulle laves om, da de anvender SQL statements, når de oprettes og tilgår (via data access laget) databasen for at hente deres indhold. Denne afhængighed skulle principielt kun være tilstede i data access laget. Sandsynligheden for, at det nogensinde skulle komme på tale at udskifte storage-systemet med en "ikke relationel" database er dog forsvindende lille. Det er derimod umiddelbart muligt at supplere med andre storage-typer til nye datatyper (f.eks. multimedia, "big data" el.lign.), som måtte blive relevante i fremtidig udvikling.

Typisk vil al forretningslogik ligge i business laget. Således vil det være muligt at udskifte (eller tilføje) et nyt presentation lag med en anden teknologi (f.eks. en webservice) uden at miste forretningslogik.

I Winformatiks tilfælde vil dette også i nogen udstrækning kunne lade sig gøre, da en del af forretningslogikken ligger i stored procedures i databasen. Dette er ikke nødvendigvis en ulempe, i det der kan være mange gode grunde til, at det ligger der eksempelvis bedre performance. Den forretningslogik der ikke ligger i databasen, ligger overvejende i presentation laget, hvorfor denne del af forretningslogikken ville gå tabt hvis man, som ovenfor nævnt, skulle erstatte presentation laget.

5.4.2.1 Komplexitet

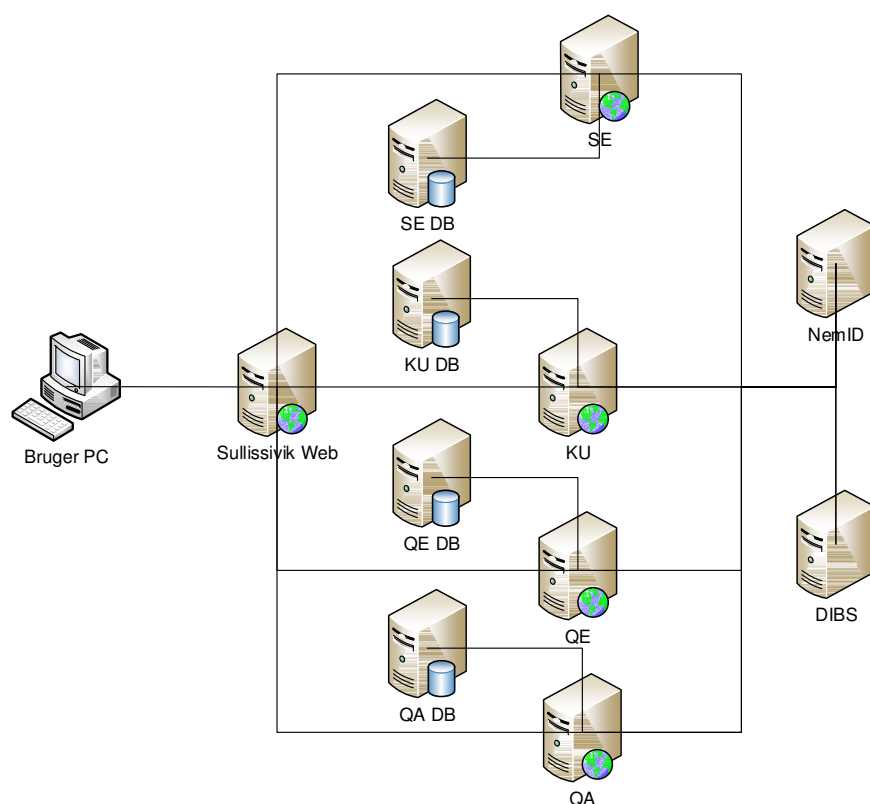
Der er forskellige metoder til at vurdere en softwares kompleksitet. Når man vælger at foretage en sådan analyse, er det ofte for at identificere kode, der er særligt kompleks og dermed sårbar over for fejl. Man kan ikke sige, at kompleksiteten skal være under et givent niveau for at foretage yderligere analyse. Oftest bruges resultatet som en hjælp til at spotte de

potentielt svage punkter, for så efterfølgende at vurdere, om der faktisk er tale om en reel risiko for fejl. En metode er at måle kodens såkaldte cyklomatiske kompleksitet (Engelsk cyclomatic complexity (CC)).

Når man foretager en simpel analyse af et del-system i Winformatik i forhold til CC med alle tilhørende og nødvendige assemblies, kan man få en CC score for hver fil, der indeholder kode – hvilket giver rigtig mange resultater. Sætter man en maksimumværdi på 10, og dermed siger at CC på over 10 udgør en moderat risiko, der er værd at undersøge nærmere, vil man nå frem til, at blot ca. 5% af alle sider i det konkrete eksempel ligger over dette niveau. Da en CC værdi over 10 ikke i sig selv udgør et problem, må dette siges at indikere, at der ikke er tale om voldsom kompliceret kode, samt at koden er designet på en måde der holder kompleksiteten nede.

5.5 Borgerservice

Der er pt. udviklet en borgerservice (køb af bopælsattest). Denne integrerer med NemID-login via sullissivik.gl og tilbyder mulighed for kreditkortbetaling via DIBS/Teller-Nets.



Figur 11 Deployment af borgerservice

Servicesen er udviklet i ASP.NET og danner basis for, hvad kommunerne fremover måtte kunne ønske sig af borgerrettede selvbetjeningsløsninger.

Servicesen afvikles på en Windows 2008 R2 server placeret uden for kommunens firewall, der kan tilgå Winformatiks database igennem firewall'en.

På længere sigt vil serveren også kunne anvendes til f.eks. at tilbyde dataudveksling med andre offentlige myndigheder via webservices.

På nuværende tidspunkt er løsningen i drift i Kommune Kujalleq og Qaasuitsup Kommuniar, løsningen er desuden implementeret i Qeqqata Kommuniar hvor man blot afventer aftaler om Internetbetaling (Dibs og Nets).

5.6 Folkeregister

Det Centrale Personregister (CPR) indeholder persondata vedrørende personnummer, navn, adresse, fødselsregistrering, statsborgerskab, folkekirkeforhold, slægtskab, civilstand samt statusoplysninger for den enkelte personregistrering.

Desuden registreres der eventuelt oplysninger i CPR vedrørende databeskyttelsesforhold, stilling, kommunale forhold, folkeregisternotater, abonnementsforhold, valgret, værgemål/umyndiggørelse og opholdsgrundlag afhængig af den enkelte persons forhold eller ønsker. Endelig registreres historiske data i CPR, når noget slettes eller ændres.

Alle borgere i Grønland er oprettet i CPR og ansvaret for disse data ligger udelukkende hos CPR. At dataansvaret ligger hos CPR betyder, at selvom alle borgere hentes fra CPR og oprettes i delmodulet folkeregister (FOLK) i Winformatik, vil det kun fungere som en passiv kopi, der vender tilbage (hvis man havde foretaget ændringer på et eksisterende CPR nummers relaterede data) til den tilstand, den er i hos CPR næste gang integrationen kører. Denne metode, der i øvrigt er et krav fra CPR's side i forhold til deres dataansvar, sikrer, at man altid vil kunne indhente nye passive kopier og ajourføre de eksisterende passive kopier.

Konkret er der i ét tilfælde (ved kommunesammenlægningen) foretaget en fuld kopi til de lokale grønlandske instanser af folkeregisteret med borgere der har eller har haft bopæl i Grønland. Efterfølgende er der på alle ugens hverdage foretaget en ændringsopdatering, der indeholder alle ændrede borgere med forbindelse til Grønland. Disse borgeres oplysninger bliver så slettet fra de lokale grønlandske instanser og erstattet med dem, der kommer fra CPR. Den daglige integration med CPR sker i form af en sikker FTP forbindelse hvorfra der hentes dokumenter, der indeholder dagens ændringsopdatering.

Opdateringer i CPR foretages udelukkende af de personer, der har tilsvarende adgangsrettigheder hos CPR. Opdateringen foregår via en tjeneste udviklet af CPR; CPR Ajour og opslag i hele CPR kan foretages af de brugere, der har adgang til den CPR udviklede tjeneste; CPR Online.

Der findes pt. 7 kopier af folkeregisteret i Grønland. De fire kommuner, Økonomi & Personalestyrelsen, Skattestyrelsen og Sundhedsvæsenet har af performance og generelle driftstekniske hensyn hver sin regionale kopi af folkeregisteret, som holdes løbende opdateret fra CPR.

CPR stiller en række services, kaldet "CPR Services", til rådighed for offentlige og private. Grunden til, at man ikke direkte anvender disse er, at man ville undgå en permanent kobling til CPR, da det ville betyde, at opslag var afhængige af forbindelsen til en server i Danmark. Fungerede denne ikke, ville det ikke være muligt at lave opslag i det grønlandske folkeregister. Desuden ville man ikke kunne afvikle batchkørsler med fornuftig performance, hvis man ønskede at anvende det til opgaver som f.eks. ajourføring af data.

5.7 Brugerstatistik

Winformatik anvendes i alle fire grønlandske kommuner og i hver kommune anvendes Winformatik på flere forskellige lokationer. Anvendelsesgraden af Winformatik varierer fra kommune til kommune. Nogle moduler er kun brugt enkelte steder mens andre anvendes overalt. Derudover er enkelte del-systemer mere anvendte end andre, når man ser på antallet af brugere. Således kan anvendelsen af enkelte moduler være ganske begrænset både hvad angår geografisk udbredelse og antallet af brugere.

At indsamle anvendelsesstatistik kan enten foregå ved at analysere IIS logfilerne, hvor antallet af kald til de enkelte sider kan findes, og ved at analysere hvor mange records der er blevet oprettet/ændret i en given periode. Alternativt kunne såkaldte web analytics software løsninger anvendes f.eks. Google Analytics.

Der ligger ikke data umiddelbart tilgængeligt for en sådan analyse, hvorfor det ville indebære et arbejde der ligger uden for denne rapports scope. Desuden skulle en sådan analyse inkludere data fra alle fire kommuner. Det er dog ganske muligt at foretage denne analyse, og den ville naturligvis være nødvendig i forbindelse med en nøjere gennemgang af hvad der evt. skal ske med de enkelte moduler og hvorledes arbejdet skulle prioriteres.

Et eksempel på et resultat af en 'læsning' af IIS loggen (én side: 900) kan ses herunder:

IIS LogReader

Generel Benyttelsesstatistik

Fradato:130401

Tildato:130430

Der læses i Logbibliotek:C:\WINDOWS\system32\LogFiles\W3SVC1

efter:/soc/900.aspx

Fil:ex130401.log indeholder:0 kald

Fil:ex130402.log indeholder:5 kald

Fil:ex130403.log indeholder:3 kald

Fil:ex130404.log indeholder:9 kald

Fil:ex130405.log indeholder:11 kald

Fil:ex130406.log indeholder:0 kald

Fil:ex130407.log indeholder:0 kald

Fil:ex130408.log indeholder:14 kald

Fil:ex130409.log indeholder:0 kald

Fil:ex130410.log indeholder:29 kald

Fil:ex130411.log indeholder:11 kald

Fil:ex130412.log indeholder:5 kald

Fil:ex130413.log indeholder:0 kald

Fil:ex130414.log indeholder:0 kald

Fil:ex130415.log indeholder:4 kald

Fil:ex130416.log indeholder:5 kald

Fil:ex130417.log indeholder:8 kald

Fil:ex130418.log indeholder:5 kald

Fil:ex130419.log indeholder:3 kald

Fil:ex130420.log indeholder:0 kald

Fil:ex130421.log indeholder:0 kald

Fil:ex130422.log indeholder:5 kald

Fil:ex130423.log indeholder:4 kald

Fil:ex130424.log indeholder:9 kald

Fil:ex130425.log indeholder:2 kald

Fil:ex130426.log indeholder:0 kald

Fil:ex130427.log indeholder:0 kald

Fil:ex130428.log indeholder:0 kald

Fil:ex130429.log indeholder:5 kald

Fil:ex130430.log indeholder:3 kald

Der var 140 kald af /soc/900.aspx i intervallet

6 Dokumentation

Udarbejdelse af teknisk dokumentation af Winformatik forefindes kun i det omfang, det er fundet nødvendigt for leverandøren selv. Dette skyldes, at dokumentation ikke er blevet prioriteret i produkt backloggen – et bevidst fravalg af kunden allerede ved opstarten på grund af manglende ressourcer. Derfor er den tilgængelige dokumentation udviklet med leverandøren selv som målgruppe. Dokumentation er primært udviklet på engelsk og sekundært på dansk.

Ifølge Bestemmelser om bevaring og kassation af arkivalier hos de grønlandske kommuner [5] betragtes dokumentation for IT systemer som værende bevaringsværdige arkivalier.

6.1 Kravspecifikation

Med kravspecifikation menes der en fuldstændig beskrivelse af adfærd for det system, der skal udvikles. Specifikationen kan omfatte en beskrivelse af kravene i form af use cases eller user stories, som beskriver de interaktioner brugerne vil have med softwaren. Desuden indeholder den ikke-funktionelle krav. Ikke-funktionelle krav kan beskrive krav til udformningen eller gennemførelsen, såsom ydeevne, kvalitetsstandarder, konstruktionskrav eller design begrænsninger.

Som tidligere nævnt er Winformatik blevet udviklet i tæt samarbejde og dialog med de brugere, der skulle anvende systemet, systemet er således udviklet med netop denne målgruppe for øje og kun denne målgruppe. Formålet med kravspecifikationen har således været at beskrive problemdomæne og funktionalitet fremfor at være en sammenfatning af kundens krav og forventninger til leverandør. Kravene er specificerede i det omfang, det er fundet nødvendigt for at opnå en fælles forståelse af, hvad en given side/modul skulle kunne. Således kan omfanget af kravspecifikation variere betydeligt afhængigt af problemdomænet.

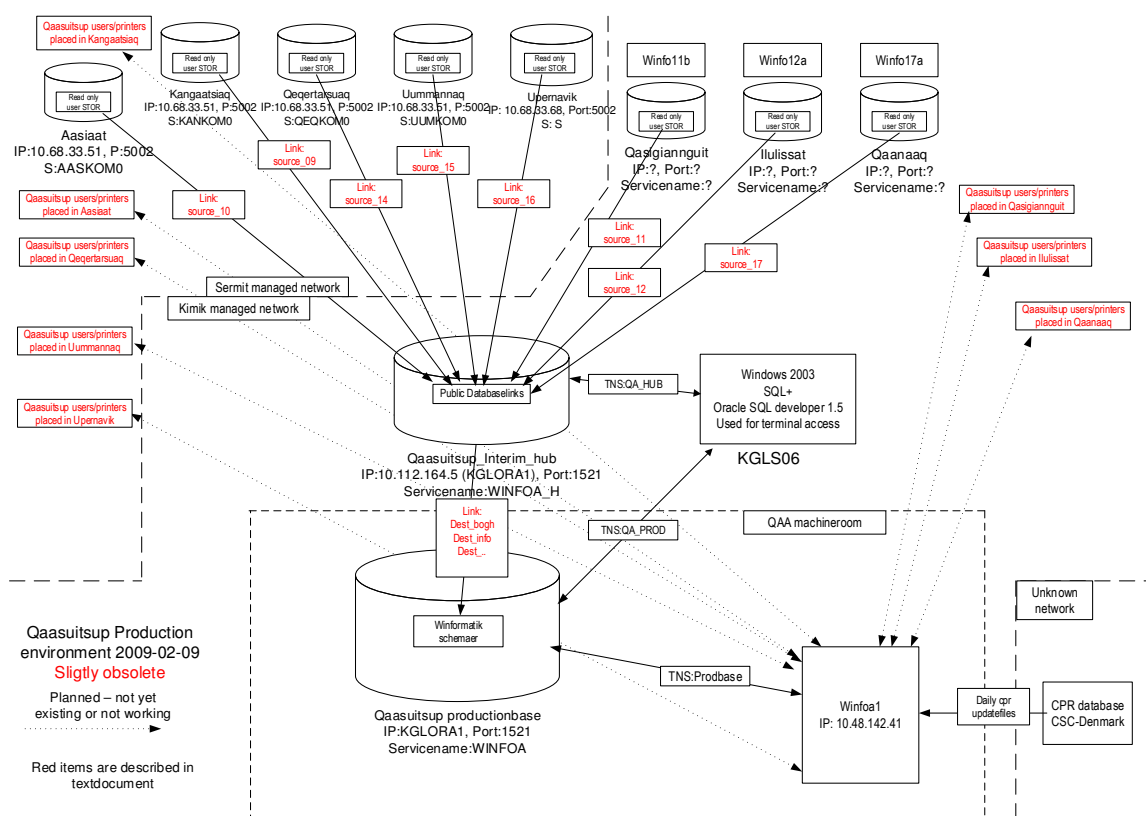
Konkrete kravspecifikationer for moduler kan bedst beskrives som bestående af de såkaldte designdokumenter, der findes på supportsitet under f.eks. Udvikling/Design dokumenter/Social system/Boligsikring samt i nogle tilfælde e-mail korrespondance mellem leverandør og kunde. Designdokumenterne fungerer også i en vis udstrækning som planlægning og prioritering af modulets udvikling.

På baggrund af den kravspecifikation der findes, kan man således ikke sige, at der findes en entydig og fuldstændig beskrivelse af, hvilke krav der stilles til de enkelte moduler, sider og felter. Det vil være mere korrekt at karakterisere det som en overordnet beskrivelse af, hvad modulet og i visse tilfælde de enkelte sider skal kunne. Ligeledes forefindes der ikke dokumentation for ikke-funktionelle krav.

6.2 Design og arkitektur dokumentation

Arkitektur skal her forstås som en overordnet beskrivelse af de software og hardware elementer og deres sammenhæng, som systemet er opbygget af. Desuden omfatter begrebet en overordnet beskrivelse af principper, krav og begrænsninger samt en begrundelse for, hvorledes arkitekturen opfylder disse.

Der er ikke udarbejdet en decideret arkitektur dokumentation for systemet. Der findes dog information, der beskriver systemets arkitektur i andre dokumenter. Herunder kan nævnes "Winformatik strateginotat om teknologi" [6], der primært omhandler systemets arkitektur set ud fra en teknologi synsvinkel. Dokumentet fungerer samtidig som et teknologi road-map for Winformatik. Derudover findes der forskellige deployment-diagrammer, der giver et overblik over platforme og sammenhænge.



Figur 12 Deployment diagram for Qaasuitsup fra 2009

Design dokumentation dækker i denne sammenhæng over detaljerede beskrivelser af systemets enkelte dele, hvordan disse er opbygget og i nogle sammenhænge en beskrivelse af den logiske domænemodel. Design dokumentationen er således mere konkret og detaljeret end arkitektur dokumentationen og fungerer dermed som et arbejdsredskab for udviklerne. Ud fra denne definition er det blot en meget lille del, af det der på supportsitet kaldes "Udvikling/Design dokumenter", der kan kaldes for designdokumentation.

Der findes dog enkelte dokumenter, der kan henføres under designdokumentation. Af disse kan f.eks. nævnes dokumentation af borgerservice.

6.3 Teknisk dokumentation og kode dokumentation

Teknisk dokumentation forstås her som dokumentation, der beskriver, hvordan et givent system, teknisk set, er udviklet ud fra en software udviklers synsvinkel. Dokumentationen kunne f.eks. indeholde beskrivelser af: anvendte frameworks og teknologier, design mønstre, datamodel, klasse-diagrammer, klasser, metoder, systemopsætning og opsætning af udviklingsmiljø.

6.3.1 Teknisk dokumentation

Der forefindes teknisk dokumentation på enkelte delsystemer af Winformatik- enten fordi de anses som særligt komplekse eller fordi man af anden årsag har erkendt et behov for at udarbejde dokumentation. Som eksempel kan nævnes:

- Office integration [7]
- Technical code standard for aspx-WebPages in Winformatik
- GUI design guidance
- Database conventions

- Håndtering af ændringsopdatering fra CPR
- Webservice communication in Winformatik – XAL Sulinal

Der er desuden en række dokumenter, der går under fællesbetegnelsen: Webpage descriptions. Disse dokumenter er på én gang tekniske design dokumenter med krav til de enkelte sider. Det er dog ikke alle sider, hvortil der er oprettet et sådant dokument.

6.3.2 Kode dokumentation

Kode dokumentation skrives direkte i sourcekoden. Derfor er det også udelukkende rettet mod udvikleren af systemet. Kvalificeret kode dokumentation vil gøre det lettere for en anden udvikler eller leverandør at overtage og forstå koden.

Kvaliteten af kode dokumentation i Winformatik varierer. Den umiddelbare vurdering er, at i systemets 'hjælpe' assemblies (dvs. afgrænsede dele af applikationen der ved kompilering bygges til én fil – assembly) er kvaliteten af dokumentation af metoder relativ høj og konsistent. Et konkret eksempel er DAL laget, det har ansvar for kommunikation med databasen og anvendes på tværs af del-systemer (og i Kimik iT på tværs af produkter). Det gør sig til gengæld ikke gældende i de konkrete del-systemer, hvor siderne har dokumentation af meget varierende karakter og kvalitet. Der er ikke konsekvent udarbejdet dokumentation af metoder som i de generelle projekter - dog er der sider hvor kvalificeret dokumentation forefindes i selve koden.

Generelt er der udarbejdet konsekvent metode dokumentation for de projekter, der anvendes i flere sammenhænge i modsætning til de øvrige projekter, der svinger fra ingen form for dokumentation til at have gode kommentarer indarbejdet i koden.

6.4 Brugervejledning

Der er udviklet en omfattende brugervejledning, dog kun på dansk og som nok burde oversættes, i form af beskrivelser af systemets enkelte sider. Beskrivelserne er offentligt tilgængelige på websitet: support.kimik-it.gl [8].

Hver side har en hjælpetekst, der beskriver anvendelsen af siden. Fra en del af disse hjælpetekster er der link til de relevante brugervejledninger. Der er således ikke direkte links fra alle sider til den relevante del af brugervejledningen, men et link til forsiden af brugervejledningssamlingen.

Der er desuden udviklet video vejledninger (~35 stk.) til systemets vigtigste procedurer. Disse forefindes, som de tekstbaserede vejledninger, kun på dansk. Dog er det desværre således at flere brugere fra deres arbejdsstation ikke har adgang til at se videoer, i det det er aktivt blevet blokeret fra system administrators side.

Man er fra leverandørs side p.t. i gang med af lave fjernundervisningsmateriale i form af et Webinar, der optages og efterfølgende kan anvendes af nye brugere. Undervisningsmaterialets formål er at undervise brugere i den daglige anvendelse af systemet – dette gælder både eksisterende funktionalitet samt nye funktioner/moduler der måtte komme. Udviklingen af et Webinar er en del af product backloggen og skal dermed prioriteres på fredags-møderne. Se mere om dette i afsnit 8.1 nedenfor.

6.5 Vurdering af systemets dokumentation

Overordnet set foreligger der ganske meget dokumentation på Winformatik, men det er ikke udarbejdet konsekvent og mangler konsistens.

Teknisk, design og arkitektur dokumentation er ikke blevet opdateret med samme frekvens som produktet. Sandsynligvis fordi der er en relativ stor udgift forbundet med løbende at vedligeholde dokumentation der til enhver tid skal være opdateret. For Winformatiks vedkommende fremgår det tydeligt, at dokumentationen er udarbejdet på baggrund af et

konkret identificeret behov. Umiddelbart lader det til at være tilstrækkeligt til at opfylde formålet, dog synes systemet at være sårbart over for udskiftning af nøglemedarbejdere hos Kimik iT, et problem man dog endnu ikke er stødt ind i.

Brugervejledning, og det materiale der i øvrigt er udarbejdet hertil, er i stor udstrækning dækkende og af en rimelig konsistent kvalitet. En forbedring på dette område ville kræve ekstra ressourcer og en engageret indsats fra kundens side.

Manglen på kravspecifikation er afgjort det mest påfaldende, når man kigger på graden af dokumentation for systemet. Det bærer tydeligt præg af at man har arbejdet med rammeaftaler, hvor prioritering og fokus har været på mængden af funktionalitet fremfor dokumentation, detaljerede tids- og handlingsplaner, implementering og kontraktmæssige forhold. Skulle samarbejdet fremadrettet have en anden struktur og indhold, hvor funktionalitet eksempelvis skulle leveres til en fast pris, ville det medføre, at der skulle udarbejdes en tydelig kravspecifikation og kontrakt for at kunne afgøre om den lovede funktionalitet var opfyldt. Desuden ville det i den sammenhæng være anbefalelsesværdigt at se mere på processen og samarbejdet. En anden samarbejdsform, som den der er beskrevet, ville kræve en større indsats fra kundens side og et ændret fokus fra leverandørs side, dette ville naturligvis have andre konsekvenser så som en dårligere ROI, i forhold til funktionalitet, uden nødvendigvis samtidig at føre til et bedre produkt. Overordnet set kan manglen på konkret kravspecifikation være et problem/en udfordring i forbindelse med at skulle genimplementere dele af systemet i en anden sammenhæng.

7 Sikkerhed

7.1 Sikkerhedsmodel

Winformatik anvender Windows Authentication til at verificere, at brugeren er medlem af domænets Active Directory (AD). Hvis brugeren er verificeret, kan dennes brugernavn slås op i Winformatik for at se, hvilke sider brugeren har adgang til. Det vil sige, at en systemadministrator først skal oprette brugeren i Winformatik med de ønskede rettigheder, før denne i realiteten kan få adgang til Winformatik.

I forbindelse med at verificere en brugers adgang til en given side, benytter Winformatik sig af en to-delt sikkerhedsmodel. På den ene side verificeres brugeren vha. Windows Authentication og på den anden verificeres det at brugeren er ved dennes arbejdsstation (terminal). Det sidste krav knytter sig dog kun til visse sider.

Når brugerne er logget på med deres domænelogin, kan de få adgang til Winformatik. Det kræver dog, at en administrator opretter dem i Winformatik med brugerens domænelogin og giver brugeren rettigheder til de sider, han/hun skal kunne tilgå. Derudover skal hver bruger, der har behov for at kunne bogføre, have oprettet en såkaldt terminal, der identificerer brugerens PC (Se <http://support.kimik-it.gl/winfosupport/brugervejledning/Systemadministration/terminaler.aspx>).

Enkelte sider kræver verifikation af både bruger og terminal. Således kan en given bruger, der har rettigheder til en af disse sider, ikke blot gå over til kollegaens PC og tilgå disse sider – med mindre dennes PC også er oprettet som terminal. Dette forhindrer personer i at kunne tilgå siderne fra deres PC, såfremt de på uærlig vis har erhvervet sig et login, der giver adgang til siderne. Desuden registreres det, fra hvilken terminal transaktionen har fundet sted. Uafhængig verifikation af både bruger og PC (terminalen) er et krav fra revisionen og er typisk gældende på økonomiske transaktioner, der foretages i Winformatik.

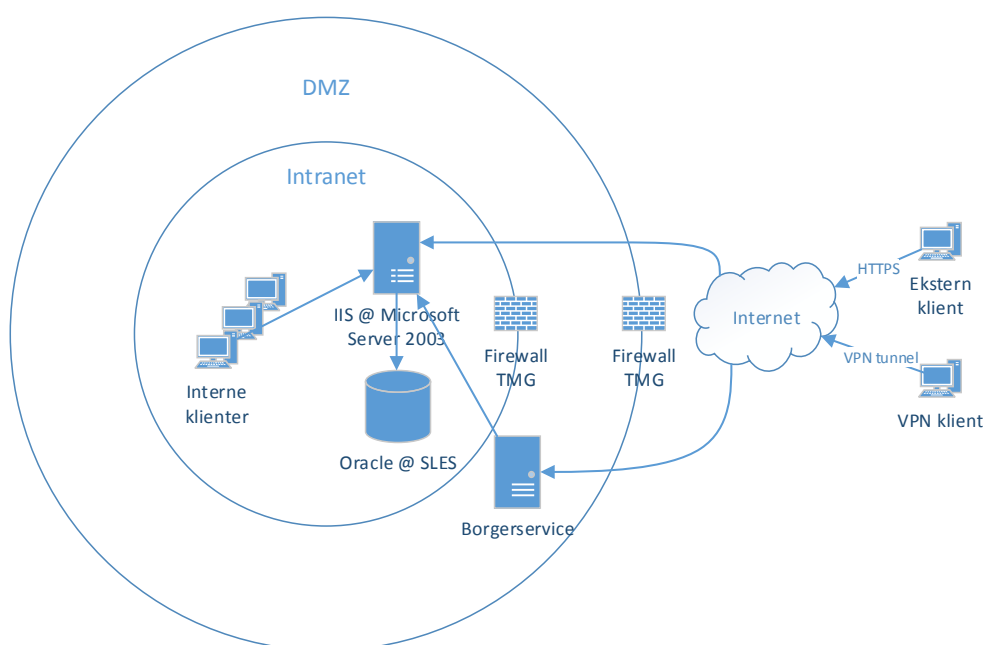
Winformatik anvender det, der kaldes Role-based security. Således findes der 9 niveauer/roller som systemadministratoren kan tildele side-rettigheder til. De enkelte brugere kan så få tildelt ét eller flere af disse niveauer, og dermed arve disses rettigheder. Derudover kan rettigheder til specifikke kategorier tildeles.

Kategoribegrebet er oprindeligt designet til at indikere en anvendelses- og brugeradgangsmæssig sammenhæng mellem grupper af skærbilleder i Winformatik. Der kan defineres op til 99 kategorier inden for hvert delsystem. Brugere gives derved ikke rettigheder til specifikke enkeltsider, men tildeles i stedet rettigheder til de konkrete kategorier inden for hvert delsystem, som denne er autoriseret til at benytte. (Se evt. <http://support.kimik-it.gl/winfosupport/brugervejledning/Systemadministration/Default.aspx#Brugerrettigheder>).

7.2 Teknisk sikkerhed

For at sikre et så homogent miljø som muligt, hvor de enkelte elementer er bygget til at fungere sammen, har man valgt at anvende Microsoft produkter i så stor udstrækning som muligt. Desuden sikres anvendelse og udnyttelse af licenser, man i forvejen har.

I Figur 13 ses den sikkerhedsmæssige opbygning af systemet. Både når systemet tilgås fra intranettet og når det tilgås fra internettet, anvendes Windows Authentication til at verificere brugeren. Dette er betegnet som best-practice fra Microsofts side, når det gælder applikationer som Winformatik. Der anvendes naturligvis Hypertext Transfer Protocol Secure (HTTPS) på en Secure Socket Layer (SSL) forbindelse, når systemet tilgås fra internettet.



Figur 13 Sikkerhedsmæssig deployment af Winformatik

HTTPS giver brugeren en sikkerhed for, at webstedet og den tilhørende web-server er dem, de udgiver sig for, og derigennem beskytter mod Man-in-the-middle-angreb. Derudover giver det en tovejs kryptering af kommunikation mellem en klient og server, som beskytter mod aflytning og manipulation med og/eller forfalskning af kommunikationens indhold.

Brugere kan ligeledes via en VPN forbindelse få adgang som var de i intranettet. Som nævnt tidligere anvendes Windows Authentication til at verificere brugeren.

Således anvendes følgende produkter i forskellige sammenhænge i relation til sikkerhed:

- Database server: SUSE Linux Enterprise Server (SLES)

- Webserver: Microsoft Server 2003
- Firewall: Microsoft Forefront Threat Management Gateway (Forefront TMG)
- VPN: Microsoft VPN Client
- Virtualisering: Microsoft's Remote Desktop Session Host services (RDSH)

For yderligere information om teknik se afsnit 10 nedenfor.

7.3 Fysisk sikkerhed

Da det er kommunerne selv, der står for den fysiske placering og de mekaniske processer, der er forbundet med driften af en server, er det ligeledes kommunen, der vælger hvordan serveren fysisk sikres.

Se afsnit 11 for yderligere information om teknik og IT drift.

7.4 Backuprutiner

For såvel den fysiske sikkerhed som backup og restore gælder det, at den mekaniske del af backup rutinerne bliver foretaget af det IT personale kommunerne har ansat. Kimik iT foretager overvågning og oplyser om manglende backup i de tilfælde, hvor fjern-drift af Winformatik serverne er en ydelse, der er tilkøbt af kommunerne.

I de kommuner hvor Kimik iT anvendes som IT leverandør, anvendes Symantec Backup Exec som backupløsning.

Se i øvrigt afsnit 11 nedenfor for yderligere information om teknik og IT drift.

7.5 Vurdering af sikkerhedsmodellen og fysisk sikkerhed

Systemets tekniske sikkerhed synes at være tilstrækkelig og passende til de forhold, der arbejdes under. Således er bindingen til AD'et med til at simplificere adgangen for brugeren og for administratoren. I forhold til hvordan man praktisk anvender rollebegrebet, kunne en mere fleksibel løsning der i højere grad var integreret med AD'et være at foretrække. Men overordnet set synes behovet for store forandringer ikke til at være til stede.

Den fysiske sikkerhed, administration af systemet og backuprutiner er alt sammen noget, der i princippet ikke er en integreret del af systemet. En mere specifik liste over krav til deployment af systemet kunne måske medvirke til at minimere eventuelle problemer for kunder, der vælger selv at stå for administration og deployment.

8 Udviklingsmetoder og værktøjer

8.1 Udviklingsprocesser

Softwareudviklingsprocesser er den metode og struktur, der er valgt at følge en given udvikling. Processen beskriver oftest, hvilke aktiviteter der skal udføres i løbet af processen, samt hvornår og i hvilken rækkefølge disse skal laves.

Oftest ser man, at store som små udviklingshuse er meget pragmatiske i forhold til i hvor høj grad én bestemt udviklingsproces vælges og hvor konsekvent den følges. Som oftest lader man sig inspirere af flere af de mest udbredte udviklingsprocesser og ender med en metode der er tilpasset virksomhedens særlige behov/situation.

Hvor veldokumenteret den valgte metode er, hvor konsekvent den følges og hvor bevidst man er i valget af metode kan variere meget. Det er her, differentieringen på en god og en dårlig udviklingsproces skal findes, og ikke i hvor høj grad én anerkendt metode følges slavisk.

Udviklingen af Winformatik har, som tidligere nævnt, frastarten foregået i nær dialog med de mennesker der skulle bruge systemet og løse problemerne. Udviklingsmetoden er derfor i stor udstrækning baseret på dette samarbejde, hvor en

række forskellige personer inden for forskellige faggrupper er blevet udpeget til at være domæne eksperter inden for et givent del-system.

I regi af det fælleskommunale IT-udvalg, KAQA, afholdes der hver fredag et sprint planning (SCRUM terminologi) møde hvor produkt back-loggen vedligeholdes og prioritering af drift og udviklingsopgaver aftales. Det er således repræsentanter fra KAQA, der er product owner (SCRUM) af Winformatik, og SCRUM begreber anvendes i forbindelse med systemets overordnede planlægning og prioritering.

På møderne deltager der mellemledere og/eller superbrugere fra 2-3 kommuner. Der er pt. ikke deltagelse fra topledelse.

Efterfølgende ser udviklingsprocessen ud som følger:

1. Design (kravspecificering) af løsningen i samarbejde med faggruppe-medlemmer for det aktuelle del-system, udvalgt af KAQA
2. Udvikling af løsningen
 - a. Ofte vil udvikling af større moduler o.l. blive udviklet i iterationer (faser) for at sikre, at løsningen er på rette vej.
3. Intern test af løsningen
4. Pilot-drift i udvalgt kommune
5. Endelig og fuld idriftsættelse

Udviklingsprocessen lader umiddelbart til primært at være baseret på en vandfaldsmodel, men i praksis er der tale om en mere iterativ proces end som så. Under udvikling, design og test kan krav udvikle sig, og man er i kontinuerlig kontakt med faggruppen, som det også ses i pkt. 2.a.

8.2 Systemudrulning

Når et system skal udrulles, indebærer det altid en række forskellige processer, der hver især skal være på plads for at sikre en smertefri ibrugtagning og anvendelse af systemet [9].

Som afsnit 8.1 ovenfor beskriver, er der naturligvis system-implementeringen. Denne del dækker over selve udviklingen af systemet. For Winformatiks vedkommende er dette fortrinsvist noget, der varetages af udviklere hos leverandøren, naturligvis deltager faggrupper undervejs, men ansvaret ligger udelukkende hos Kimik iT.

Næste skridt er teknisk implementering herunder indkøb, installation og opsætning af den fysiske hardware og systemet. Dette samt fjerndrift af serverne varetages i dag af Kimik iT for tre ud af de fire storkommuner. Kommuneqarfik Sermersooq står som den eneste kommune selv for denne del af udrulningen. Dog bistår Kimik iT med rådgivning og hjælp til installation af systemet – denne ydelse ligger som udgangspunkt ud over det kontrakten dækker.

Sidste skridt er organisatorisk implementering, altså konkret ibrugtagning af systemet dvs. klargøring af organisationen til at anvende systemet, herunder evt. procesomlægninger og i særdeleshed brugeruddannelse. I forhold til hvilken brugeruddannelse slutbrugerne skulle tilbydes, besluttede man (kunde og leverandør) - primært på baggrund af begrænsede ressourcer og ønsket om at anvende disse på funktionalitet - at faggruppemedlemmer, der havde været involveret i udviklingen af de forskellige moduler, skulle forestå undervisningen af de medarbejdere der anvendte disse, således at faggruppemedlemmerne blev en slags superbrugere med ansvar for undervisning på de enkelte moduler. I realiteten viste det sig dog at være svært for disse superbrugere at få tid til og mulighed for konsekvent og struktureret at gennemføre et uddannelsesforløb. Således må brugeruddannelse i dag anses for at være sporadisk og noget der foregår som sidemandsoplæring.

Kimik iT oplever, at der i nogle sammenhænge er enkelte moduler/funktionalitet, som ikke anvendes af alle fire kommuner f.eks. elektronisk bilagshåndtering. Noget der til dels ses som et tegn på manglende brugeruddannelse.

Også fra leverandørs side vil brugeruddannelse altid være en meget vigtig del af udrulningen, idet systemets succes på overfladen (ud fra slutbrugerens synsvinkel) er afhængig af, hvor let det er at anvende set fra brugerens side. Brugeruddannelse er i Winformatik sammenhænge noget, som kunden selv er ansvarlig for. Dog har man fra leverandørs side forsøgt med en support webside, video vejledninger og som det sidste nye; Webniar der efterfølgende lægges tilgængeligt for alle brugere. Se mere om dette i afsnit 7.4 ovenfor. Disse tiltag prioriteres i product backloggen og trækker således p.t. på drift og udviklingsressourcerne.

8.3 Udviklingsværktøjer

Med anvendelse af ASP.NET til klient og Oracle som database, er en del af de anvendte værktøjer stort set givet på forhånd.

Følgende værktøjer og udviklingsmiljøer anvendes:

- Microsoft Visual Studio 2010 (VS) til udvikling af ASP.NET siderne.
- Oracle v. 11.2 som database
- Oracle SQL developer 3.2
- SAP Crystal Reports v. 12 (CR) som rapport genereringsværktøj. CR var en integreret del af VS indtil 2008, hvorefter Microsoft gik over til SQL Server Reporting Services (SSRS)

Ud over disse anvendes følgende tredjeparts komponenter i forskellige sammenhænge:

- wodFtpDLX ActiveX component - FTP klient fra WeOnlyDo (<http://www.weonlydo.com/>)
- ZBit Zip Unzip - Zip værktøj fra ZBITZ (<http://www.zbitinc.com/>)
- Aspose.Words for .NET til konvertering af Word-dokumenter til PDF (I forbindelse med bopælsattest i borgerselvbetjening) fra ASPOSE (<http://www.aspose.com/>)
- FCK editor – tekst editor fra Drupal (<http://drupal.org/>)

Microsoft har i mange år haft stor succes med deres udviklingsværktøjer, som i sig selv kan tage æren for en del succesfulde Microsoft teknologier. Visual Studio er Microsofts integrated development environment (IDE) og betragtes som de facto standard, når det gælder udvikling af alle typer Microsoft løsninger. Seneste stabile udgave af Visual Studio er i dag 'Visual Studio 2012 Update 1' (Nov. 2012). Man forventer at kunne overgå til denne omkring årsskiftet 2013/14.

Kimik iT's udviklingsmiljø er baseret på virtuelle arbejdsstationer (Microsoft Hyper-V), der er hostet i Danmark.

8.4 Test

Som nævnt i afsnit 9.1 ovenfor, gennemføres der en intern test af nye færdigudviklede funktioner. Testen udføres af egentlige testere og udføres som exploratory testing. Ved fejl oprettes der en testrapport, som følger fejlen, indtil den er løst.

Der foretages ingen automatiseret test i form af unittest, integrationstest, system tests eller lignende. Da systemet ikke som udgangspunkt er forberedt på automatiserede tests, må det forventes, at introduktion af automatiserede tests, vil omfatte betydelige omskrivninger af den eksisterende kode.

Automatiseret test af brugergrænseflader er i stor udstrækning afhængig af gode værktøjer og opsætning/anvendelse af disse. Man har tidligere analyseret muligheden for at lave automatiseret tests af brugergrænsefladen i Winformatik, men

afviste det på baggrund af for store omkostninger i forbindelse med indførelsen og erkendelse af, at de relativ få fejl, der blev fundet, kunne findes og rettes med det aktuelle setup.

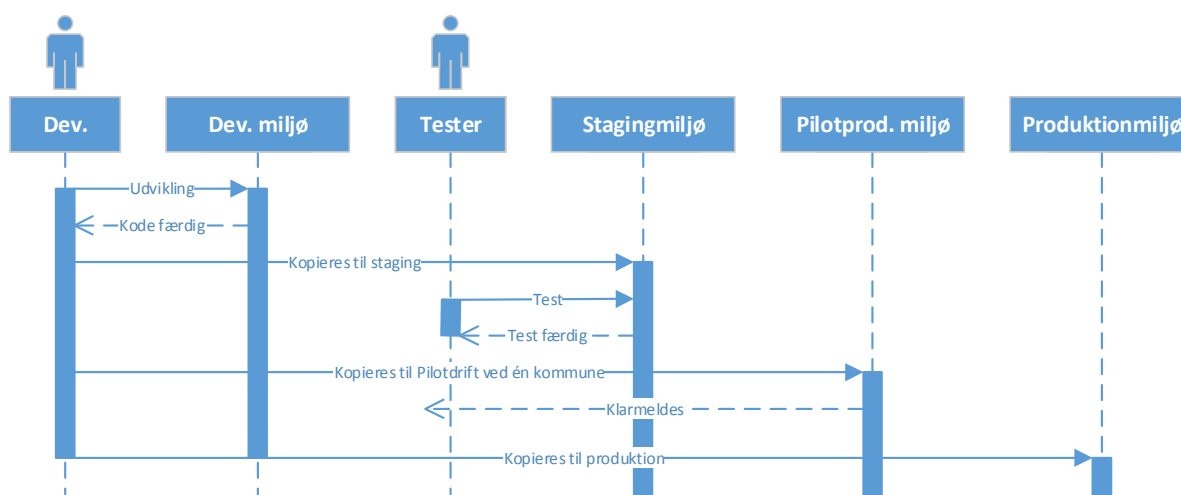
Test af brugervenlighed og systemets indfrielse af krav foregår primært under den planlagte pilot-drift.

Taget i betragtning af den manglende automatiserede test samt manglen på scriptede tests, forekommer det sandsynligt, at der vil opstå nye fejl ved ændringer og nyudvikling. Om disse fejl bliver fundet ved den planlagte interne test er muligt, der er blot ingen form for regressionstests, der fanger disse. Denne mangel er ikke nødvendigvis et problem, hvis den øvrige testaktivitet fanger disse fejl, og man i øvrigt anser det for værende en mere kosteffektiv metode. Der findes desværre ikke et egentlig bug/defect-tracking system, der f.eks. viser, hvor mange åbne fejl der p.t. ligger. Tilbage meldingen er, at der generelt kun er få åbne fejl herunder ingen alvorlige.

8.4.1 Udviklings-, test- og produktionsmiljø

Hver af de fire storkommuner har et produktionsmiljø. Derudover findes der et test miljø, hvor test af færdigudviklet funktionalitet foretages og dertil en række udviklingsmiljøer.

Nedenfor i Figur 14 ses et forsimplet diagram, der viser, hvorledes ny funktionalitet 'flytter' sig fra det ene miljø til det næste. Der er her anvendt et sekvensdiagram til at illustrere processen. De horisontale liv-linjer repræsenterer aktørerne eller miljøerne. For at se et mere detaljeret diagram over et udviklingsmiljø og testmiljø samt et eksempel på en storkommunes produktionsmiljø henvises til afsnit 11.



Figur 14 Udviklingssekvens, fra kode til produktion

8.5 Vurdering af udviklingsmetoder og værktøjer

Leverandør forsøger kontinuerligt at holde produktet opdateret i forhold til den teknologi, der anvendes. Dette er en klar fordel for produktet i forhold til at kunne videreudvikle på det. Desuden anvender man udbredte og standardiserede udviklingsværktøjer, der bl.a. gør det lettere at få kompetent personale.

På trods af at man, i lighed med stort set alle øvrige softwarehuse, anvender offshore udvikling, findes der kun ét udviklingsspor for produktet. Er der ønsker eller planer om sideløbende udviklingsforløb med flere udviklere involveret, burde flere spor overvejes fremadrettet. Det, at der kun er ét udviklingsspor, simplificerer til gengæld deployment af ny kode, hvilket er en fordel, så længe det er muligt.

9 Logning

9.1 Teknisk og performance logning

Generelt kan software logning opdeles i to overordnede kategorier:

- Eventlogning der primært anvendes af en systemadministrator og som generelt har en lidt mere overordnet karakter.
- Tracelogning der typisk bruges af udvikleren til at identificere og finde fejl.

Alt logning af fejlmeddelelser, advarsler og øvrige informationer fra Winformatik sker til Windows Eventlogs. IIS-loggen (internet Information Services) opsamler generelle informationer omkring de enkelte brugersessioner, som tilgår Winformatik. Således findes både eventlogging og tracelogging under 'et tag'. Hver dag gemmes dagens logfil - i princippet uendeligt.

De informationer, der logges, er primært af teknisk karakter og vil oftest anvendes som sådan. Som udgangspunkt er det ikke meningen, at loggen kan eller skal anvendes som historik, men den kan i visse tilfælde være anvendelig i revisionssammenhænge (hvilket er sket i et konkret tilfælde).

Det er ikke fundet nødvendigt at lave nogen form for specifik performancelogning, da de vanskeligheder, brugerne som oftest står med i forhold til responstider, skyldes forbindelsen mellem klient og server fremfor manglende performance fra systemets side.

9.2 Brugerhandlinger der logges

Generelt kan man sige, at stort set alle borger/leverandør/økonomi relaterede data har påstemplet en bruger, en dato og et tidsstempel. Således er det for denne type data (som udgør den overvejende del af systemet) muligt at se, hvem der har rettet/oprettet data og hvornår. Desuden vil denne information ligeledes ofte være tilgængelig for brugeren på selve siden. Det er derimod ikke muligt at se historikken for et enkelt felt i databasen dvs. på et vilkårligt tidspunkt at se hvem der har rettet sidst og til hvad.

Eksempelvis vil én af tabellerne, der danner grundlag for siden Social/Boligsikring/Boligsikringsansøgninger (570) have følgende felter, hvor logning udgør de sidste to kolonner (Tabelnavn: BOLIGSIKRING_LEJEMAAL):

- 1 Kunde-nummer på udlejer
- 2 Nummer på lejemål hos udlejer. Svarer til feltet LEJER_NR i tabellen BOLIGSIKRING
- 3 Dato fra hvilken oplysninger gælder
- 4 Kommune-nummer 01..18
- 5 Bygningsnummer
- 6 Lejlighedsnummer
- 7 Lejers CPR-nummer
- 8 Lejers navn jvf oplysninger fra udlejer
- 9 Lejers adresse jvf oplysninger fra udlejer
- 10 -
- 11 Månedlig husleje
- 12 Areal i kvadratmeter
- 13 Antal rum i boligen
- 14 Indflytningsdato
- 15 Udflytningsdato
- 16 -

- 17 -
- 18 Hvis udlejer er INI, så er dette muligvis fremlejers navn og adresse
- 19 -
- 20 01..19: kommunalt selskab, 30+kom-nr: hj.st.selskab (f.eks. 36 Hj.st.selskab i Nuuk
- 21 Afdelingsnummer
- 22 Last date record was updated
- 23 FK to info.Im_user. Person who did last update

9.3 Historisk data

Felters historik er som beskrevet i det foregående afsnit som udgangspunkt ikke gemt i databasen. Der er dog et par særlige undtagelser på baggrund af ønsker fra kundens side:

- Søgning og visning af detaljer i folkeregisteret, konkret logges følgende (for en øget kontrol med hvem der udsøger hvem):
 - Tidspunkt for logning
 - Initialer på operatør der udfører hændelsen.
 - Er hændelsen gået godt, skidt eller er det en adm. hændelse. (J | N | A).
 - Webside nummer som har logget hændelsen.
 - CPR nummer på person som logningen vedrører.
 - Beskrivelse af hvad loghændelsen bestod af. Søgekriterium eller beskrivelse.
 - Ip-adressen på den bruger det drejer sig om
 - Audit typen - login, log ud og diverse
 - PID på den bruger der er logget på Citizen
 - SessionsID på brugeren
- Bankkonto (for at kunne se f.eks. om kontonr. er ændret lige før udbetaling og tilbage igen lige efter)
 - Modtager af beløb
 - Type (1:soc, 2: kreditor, 3:overskydende skat)
 - Reg. Nr.
 - Kontonr.
 - Journal-nummer. Bruges som kundenummer ved soc.udb.
 - Init
 - Reg. Dato
 - Operation (I: indsæt, U: opdater, D: slet)

Historiske data kan dog ikke tilgås via brugergrænsefladen. Det er derfor nødvendigt at kontakte Kimik iT for at få disse oplysninger.

9.4 Lovmæssige krav

9.4.1 Lov om offentlige myndigheders registre

Den generelle persondataloven for offentlige myndigheder i Grønland udgøres i skrivende stund af lov om offentlige myndigheders registre, som blev ikraftsat i 1978 [10]. De efterfølgende ændringer af loven er dog ikke blevet sat i kraft i Grønland.

Persondataloven blev indført i Danmark i maj 2000 [11], som erstatning for registerloven. Persondataloven er heller ikke ikraftsat i Grønland.

Følgende fremgår af persondatalovens § 83

§ 83. Loven gælder ikke for Færøerne, men kan ved kongelig anordning sættes i kraft for rigsmyndighedernes behandling af oplysninger med de afvigelser, som de særlige færøske forhold tilsiger. Loven gælder heller ikke for Grønland, men kan ved kongelig anordning sættes i kraft med de afvigelser, som de særlige grønlandske forhold tilsiger.

Det skal understreges, at det alene er loven der kan sættes i kraft. Eventuelle ændringer skal være nødvendige af hensyn til grønlandske forhold, f.eks. forudsætter en ikraftsættelse, at man fjerner persondatalovens regler om TV-overvågning, fordi disse regler indgår i et samspil med den danske lov om TV-overvågning (som ikke gælder i Grønland).

9.4.2 Sikkerhedsbekendtgørelsen og sikkerhedsvejledningen

Det fremgår af § 41, stk. 5 i den danske persondatalov, at der kan fastsættes nærmere bestemmelser angående organisatorisk og teknisk sikkerhed. Denne hjemmel er blevet udnyttet med bekendtgørelse om sikkerhedsforanstaltninger til beskyttelse af personoplysninger, som behandles for den offentlige forvaltning. Desuden har Datatilsynet udstedt en Vejledning til bekendtgørelse nr. 528 af 15. juni 2000 om sikkerhedsforanstaltninger til beskyttelse af personoplysninger, som behandles for den offentlige forvaltning.

Disse retskilder er ikke gældende ret i Grønland; men retskilderne tjener samme formål, som tilgodeses i 1978-registerlovens mere summariske bestemmelser om sikkerhed. Grønlandske offentlige myndigheder er dermed ikke bundet af disse retskilder; men på den anden side er der intet til hinder for at grønlandske offentlige myndigheder søger inspiration i disse retskilder.

9.4.3 CPR

Den danske lov om Det Centrale Personregister [CPR] er ikraftsat i Grønland med enkelte tilpasninger. Det er værd at bemærke, at loven bl.a. er tilpasset således, at videregivelse fra de grønlandske folkeregistre til andre grønlandske offentlige myndigheder skal respektere den danske persondatalov selvom denne ikke er ikraftsat i Grønland. Dette betyder bl.a. at opslag i CPR på tværs af de Grønlandske kommuner og sågar i danske kommuner er inden for gældende lovgivning.

9.4.4 Arkivloven

For alle grønlandske offentlige myndigheder gælder Landstingslov nr. 22 af 30. oktober 1998 om arkivvæsen [12]. Denne lov indeholder regler om hvorledes offentlige myndigheder skal forholde sig til arkivalier.

Den danske arkivlov kan alene sættes i kraft i Grønland for så vidt angår rigsmyndighedernes sager, jf. arkivlovens § 56, hvoraf følgende fremgår:

§ 56. Loven gælder ikke for Færøerne og Grønland. Loven kan med de afvigelser, som de særlige færøske og grønlandske forhold tilsiger, ved kongelig anordning sættes i kraft for sager, der er eller har været under behandling af rigsmyndighederne.

9.4.5 Overholdelse af gældende lovkrav

I forhold til om Winformatik overholder alle gældende love og bekendtgørelser om IT systemer og registre vil en mere tilbundsående undersøgelse være nødvendig.

Umiddelbart vurderes det at Winformatik overholder registerloven, i det at registerloven omhandler data (og videregivelse af data) i registre og at der i forbindelse med opbevaring og videregivelse af data ofte anvendes et nødvendigheds princip i forhold til hvilken type data der må opbevares. Som udgangspunkt er data i Winformatik nødvendige for udførelsen af den givne opgave. Desuden er det også organisationens behandling/anvendelse af data der skal leve op til lovgivningen.

I det omfang Winformatik automatisk, gennem snitflader til andre offentlige som private systemer, leverer data, bør det vurderet hvorvidt registerlovens videregivelsesregler er overholdt. Som nævnt ovenfor anvendes der et nødvendigheds princip der gør at tt forsigtigt skøn ville være at Winformatik også her overholder lovgivningen. Et eksempel på nødvendighedsprincippet ses herunder i §17 fra Lov om offentlige myndigheders registre:

§ 17. Oplysninger om bestemte personer kan videregives til private virksomheder m.v., ved løsning af vedkommende myndigheds opgave. Der må dog kun ske videregivelse af oplysninger, der er nødvendige i den modtagende virksomheds arbejde for det offentlige. Oplysningerne må ikke benyttes til andre formål end fastsat af den registeransvarlige myndighed.

Selvom sikkerhedsbekendgørelsen ikke er ikraftsat i Grønland vil det som tidligere nævnt være fornuftigt at forholde sig til de konkrete retningslinjer der er nævnt i denne. Der er i det efterfølgende gennemgået nogle af de krav/problematiske der nævnes i særligt § 18 og § 19 i sikkerhedsbekendgørelsen.

Kontrol med afviste adgangsforsøg

§ 18. Der skal foretages registrering af alle afviste adgangsforsøg. Hvis der inden for en fastsat periode er registreret et nærmere fastsat antal på hinanden følgende afviste adgangsforsøg fra samme arbejdsstation eller med samme brugeridentifikation, skal der blokeres for yderligere forsøg. Der skal løbende ske opfølgning i myndigheden.

I forhold til § 18 der skal brugeren først være logget på Windows for at kunne tilgå Winformatik. Det betyder at Winformatik i relation til overordnet adgangskontrol benytter Windows sædvanlige mekanismer, der indebærer logning af ethvert afvist pålogningsforsøg i Windows Security eventlog samt spærring af brugerkontoen i X tid efter Y forsøg inden for Z tid.

Hvis en (Windows-akkrediteret) bruger forsøger at tilgå Winformatik uden at være oprettet som bruger i Winformatik eller uden at have ret til nogen sider i Winformatik, logges det anvendte brugernavn i Eventloggen. (Arbejdsstationens IP-adresse vil kunne findes via IIS-loggen).

Idet Winformatik kun kan tilgås af validerede windows-brugere, fokuseres i Winformatik på kontrollen med og begrænsning af disse autoriserede brugeres adgang til konkrete skærbilleder og funktionalitet. Der sker ikke nogen automatiseret blokering af brugeren i Winformatik.

Eksempel på logning i Eventloggen af forsøg på til at tilgå Winformatik uden rettigheder:

=====

<i>Event Type:</i>	<i>Error</i>
<i>Event Source:</i>	<i>Info</i>
<i>Event Category:</i>	<i>None</i>
<i>Event ID:</i>	<i>1100</i>
<i>Date:</i>	<i>02-05-2013</i>
<i>Time:</i>	<i>21:41:02</i>
<i>User:</i>	<i>N/A</i>
<i>Computer:</i>	<i>WINFOQA1</i>
<i>Description:</i>	

Initialer NUK er ikke oprettet i Info.Operator-tabel eller side 12 i Winformatik.

For more information, see Help and Support Center at <http://go.microsoft.com/fwlink/events.asp>.

=====

Der foretages logning af alle opslag via søgesider i folkeregister. Loggen bliver ikke slettet automatisk. Loggen opbevares i en database-tabel uden tilgang fra nogen side i Winformatik. Indholdet af loggen kan læses af Kimik, og oplyses til kommunen på anfordring.

Der foretages ikke pt. logning fra andre sider i Winformatik, som anvender personoplysninger. F.eks. viser siden til pensionsberegningen følgende oplysninger: CPR-nr, navn, civilstand, CPR-nr og navn for evt. (tidligere) ægtefælle.

Logning

§ 19. Der skal foretages maskinel registrering (logning) af alle anvendelser af personoplysninger. Registreringen skal mindst indeholde oplysning om tidspunkt, bruger, type af anvendelse og angivelse af den person, de anvendte oplysninger vedrørte, eller det anvendte søgekriterium. Loggen skal opbevares i 6 måneder, hvorefter den skal slettes. Myndigheder med et særligt behov kan opbevare loggen i op til 5 år.

I forhold til § 19 foretages der logning af alle opslag via søgesider i folkeregister. Loggen bliver ikke slettet automatisk. Loggen opbevares i en database-tabel uden tilgang fra nogen side i Winformatik. Indholdet af loggen kan læses af Kimik, og oplyses til kommunen på anfordring.

Der foretages ikke pt. logning fra andre sider i Winformatik, som anvender personoplysninger. F.eks. viser siden til pensionsberegningen følgende oplysninger: CPR-nr, navn, civilstand, CPR-nr og navn for evt. (tidligere) ægtefælle.

9.4.5.1 Arkivloven

Det er påfaldende at der ikke er taget skridt til at lave aftale med Grønlands Nationalarkiv i forhold til at foretage systematisk arkivering og videregivelse til arkivvæsenet. Systemet har været i drift længere end loven har eksisteret og selvom systemet i starten overvejene var en økonomi system, er der kommet flere og flere sags relaterede data i systemet med tiden. Det må forventes at man vil vurdere noget af den data der ligger i Winformatik er bevaringsværdig. Derfor vil der i givet fald sandsynligvis være behov for en datakonvertering i forhold til aflevering af data til arkivvæsenet.

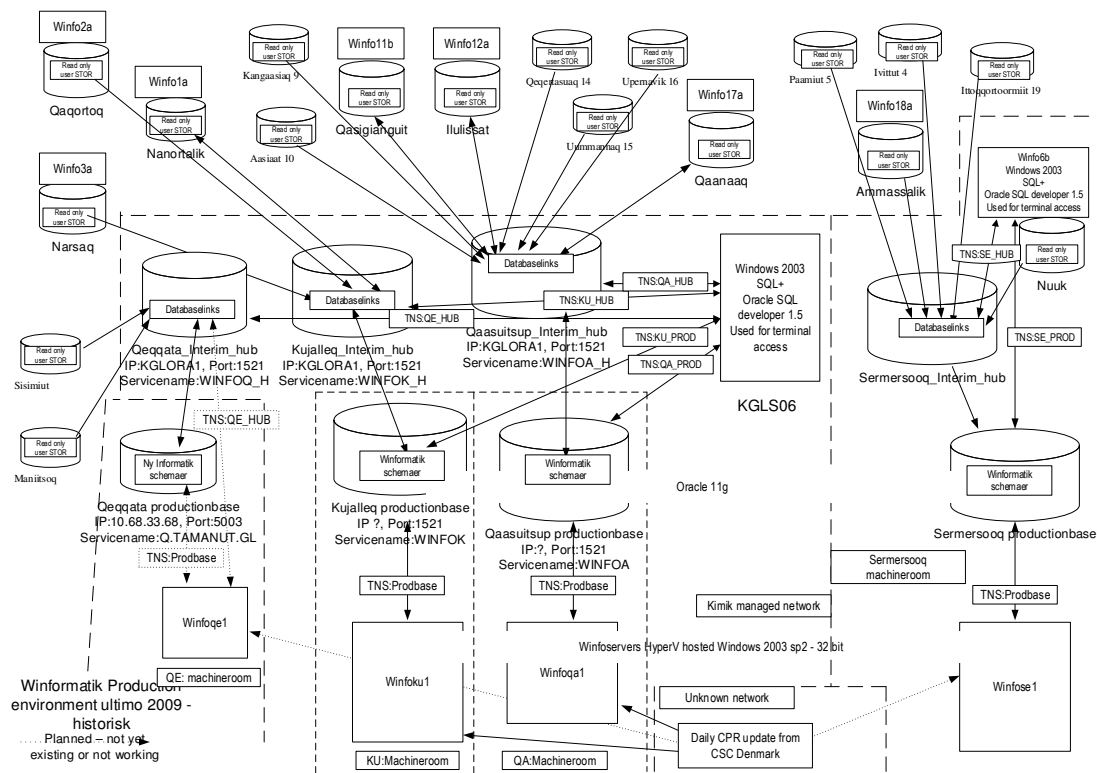
Der ligger så vidt Kimik iT er bekendte ingen aftaler om arkivering mellem Grønlands Nationalarkiv og de enkelte kommuner.

9.5 Vurdering af systemets logning

Historik og ændringslog har i systemet en udbredelse, der nøje svarer til de behov, kunden har givet udtryk for. Havde man fra leverandørs side haft en mere principiel holdning til, hvordan man ønskede at dette skulle løses, havde man i dag været i stand til at tilbyde en konsistent og ensartet historik samt ændringslog på alt i systemet – forudsat naturligvis at det var noget kommunerne som systemejer ønskede at finansiere. En højere grad af historik og ændringslog indebærer ikke nødvendigvis voldsomme ændringer, men er et tilvalg der i givet fald skulle foretages af kommunerne.

10 Teknik og infrastruktur

Ideologien bag Winformatiks deployment strategi er grundlagt, da IT infrastrukturen i Grønland var ganske ny. Det betyder, at man havde placeret maskinerne dér, hvor de skulle bruges, for således ikke at være afhængige af data forbindelser der ikke altid var tilgængelige. Muligheden for at lave distribuerede systemer har været og er i nogen grad stadig ganske begrænset. Begrænsningen ligger ikke kun i ustabile dataforbindelser (som i dag er blevet en del bedre men fortsat er ramt af nedbrud og langsomme satellitforbindelser) men også i manglende ressourcer til at udvikle serverreplikering o. lign.



Figur 15 Deployment diagram for produktionsmiljø fra 2009

Systemet er designet til at fungere i den grønlandske IT infrastruktur primært igennem den tilvalgte deployment strategi. Selve løsningen er desuden lavet som letvægts (uden meget grafik/billede) web applikation, hvilket gør at selve klienten er meget 'tynd' – en browser - og ikke kræver særlig teknisk support hos klienterne. Derudover giver det også mulighed for at tilgå applikationen via HTTPS.

I forbindelse med kommunesammenlægningen i 2009 hvor 18 kommuner blev sammenlagt til 4, blev det nødvendigt for administrationen i en række byer at anvende en Winformatik webserver, der stod i en anden by, og dermed en dataforbindelse, der var dårligere end da man havde serveren stående i det samme LAN. Dette betød samtidig, at man mange steder oplevede, at brugeroplevelsen blev ringere, da systemet nu skulle fungere via en dataforbindelse med en noget anden båndbredde og latency, end det lokale LAN kunne præstere, læs mere herom i afsnit 11.2 nedenfor.

10.1 Virtualisering

I forhold til anvendelsen af virtualisering af både klienter og servere har man på baggrund af den historisk dårlige IT infrastruktur i Grønland valgt at anvende fysiske servere/klienter. I den udstrækning hvor disse er blevet opsat korrekt, har det fungeret uden problemer, på trods af at man er afhængig af en velfungerende IT afdeling/administration (i nogle tilfælde har dette blot været én udpeget person uden egentlig IT baggrund) dér, hvor maskinerne var placeret.

Kimik IT har primært anvendt Remote Desktop Protocol (RDP) i forbindelse med opsætning og overvågning af servere.

Selve IT infrastrukturen i Grønland er i de senere år blevet betydelig bedre end den har været tidligere. Dette betyder, at der nu er åbnet op for flere muligheder, i forhold til hvordan man deployer et system som Winformatik, end det var tilfældet for blot få år tilbage. På baggrund af det og forskellige tekniske behov, har én kommune i dag opsat og anvendt klient virtualisering (Citrix).

10.2 Latency

Hastigheden på en dataforbindelse er afhængig af båndbredde (høj er godt) og latency (lav er godt). Båndbredde angiver den egentlige hastighed og latency angiver forsinkelsen. Latency kan desuden i nogle tilfælde have indflydelse på båndbredden.

Høj latency kan have mange årsager, da data bliver transmitteret over mange forskellige netværk, inden den når frem, men ofte er det den fysiske transport af signalet, der udgør den primære statiske latency.

En typisk DSL internetforbindelse bør have en latency under 100 ms., og helst omkring 25 ms. for at være god. En satellitforbindelse har ofte en latency 500-1000 ms. og i nogle tilfælde endnu mere.

Nedenfor er nævnt en række emner, der relaterer sig til dataforbindelsen og dermed latency.

10.2.1 Postback strategi

Postback er et teknisk ASP.NET begreb, der beskriver det, der sker, når der fra klientens side foretages en ændring, der returneres til serveren til behandling. Et postback er ikke nødvendigvis ensbetydende med et egentligt sideskift set fra brugerens side men kan være det.

Enhver kontrol på en ASP.NET side kan være opsat således, at der ved en ændring foretages et postback. Det kan der være mange grunde til herunder løbende validering. Dette er en helt normalt måde at udvikle sider på i ASP.NET, og under normale omstændigheder giver dette flere fordele end ulemper for brugeroplevelsen.

Hvis man forestiller sig en formular med f.eks. 20 felter, hvor måske blot halvdelen resulterer i et postback, vil man i løbet af udfyldelse af formularen have kontaktet serveren 10 gange, inden formularen rent faktisk bliver 'gemt', hvilket samtidig betyder, at siden bliver genereret fra serveren og sendt til klienten 10 gange. Som nævnt er dette normalt ikke en stort problem, da selve siden ikke nødvendigvis repræsenterer voldsom megen data. Men under normale omstændigheder er båndbredden god og latency lav (~20ms.), og selvom båndbredden kan være fin på en satellitforbindelse er latency normalt høj (~1000 ms.). Dette medfører, at brugeren oplever 10 pauser á godt ét sekund, hvilket sædvanligvis opleves som meget for en bruger, hvorfor brugeren har en oplevelse af en meget lidt responsiv brugergrænseflade.

Årsagen til, at dette kan være et problem for Winformatik i dag, er, at man før kommunesammenlægningen havde en server stående dér, hvor man anvendte systemet i modsætning til i dag, hvor systemet anvendes på distancen i de sammenlagte kommuner. Et eksempel kunne være, at man på østkysten i Tasiilaq anvender Winformatik, men serveren står på vestkysten i Nuuk. Dette er ikke et ukendt problem for Kimik iT, men i forbindelse med kommuneomlægningen var der ikke tilstrækkelige ressourcer til at løse alle de kendte problemer, der var blevet identificeret. Leverandøren har efterfølgende forsøgt at tilgå problemet ved konkret at ændre de sider, der primært anvendes på distancen, når der blev indmeldt problemer herom. Ligeledes vil anvendelsen af RDS (se mere i afsnit 11.3 nedenfor) også give tilsvarende problemer.

10.2.2 Virtualisering

Desktop Virtualisering kan i nogle tilfælde være en fornuftig løsning i distribuerede miljøer, hvor teknisk support ikke kan være fysisk tilstede, og hvor høj latency giver problemer for konventionelle client-server løsninger. Dog vil man ofte sige, at f.eks. en Citrix session normalt mindst skal have en båndbredde på 20 kbit og en latency under 150 ms. for at køre flydende – hvilket en satellitforbindelse ikke kan garantere. Dette diskvalificerer ikke i sig selv denne løsning, men vil som oftest afhænge af en test af de konkrete opgaver og den virtualiserings software, man vælger at anvende (de forskellige produkter håndterer høj latency forskelligt). I Kommuneqarfik Sermersooq har man valgt at anvende Desktop Virtualisering (Citrix). Den konkrete implementering er ikke helt problemfri f.eks. i Tasiilaq (problemet ligger sandsynligvis

ikke udelukkende i anvendelsen af Citrix men i lige så høj grad i opsætningen), hvorfor man må konkludere, at det ikke nødvendigvis er en god løsning på alle problemer, og at de normale anbefalinger ift. båndbredde og latency for virtualisering er reelle nok.

10.3 Road-map for teknik og kode

Winformatik er under konstant udvikling. Det er primært ny funktionalitet, der fokuseres på under de ugentlige fredagsmøder. Kimik iT ønsker ligeledes, at produktet og de moduler, det er opbygget af, udvikles ud fra et teknisk synspunkt. Dette er ofte ikke noget, kunden umiddelbart vil kunne se vigtigheden af, men det er en udvikling der ikke desto mindre er vigtig for, at produktet fortsat kan udvikle sig. Nogle af disse tiltag udgør en del af den prioriterede backlog.

10.3.1 RDS (Remote Data Services)

RDS er en funktion, der gør det muligt at sende data fra serveren til klienten, manipulere data på klienten og returnere til serveren i én handling. RDS teknologien er under afvikling og samtidig den primære årsag til, at serveren skal være Microsoft Server 2003.

RDS, der resulterer i hyppigere kald til serveren, er medårsag til de problemer, der kan vise sig, når klienten bruges på en forbindelse med høj latency. Læs mere herom i afsnit 11.2. Selvom der er alternativer til RDS såsom WCF Data Service eller AJAX, vil disse have samme udfordringer i relation til høj latency.

På grund af bindingen til en specifik server version og det faktum at teknologien er under afvikling, er det Kimik iT's ønske og plan at have afviklet RDS i Winformatik ved udgangen af 2013.

Dokumentet Webpage_name_status.xls [13] er et arbejdsdokument, der beskriver de enkelte sider og hvilke af dem der anvender RDS.

10.3.2 Klassisk ASP

Forløberen til ASP.NET teknologien er klassisk ASP. Begge teknologier kan for så vidt godt eksistere under samme tag, og der er ikke pt. noget, der på serversiden forhindrer fortsat at afvikle ASP sider. Dog er der en lang række begrænsninger forbundet med at anvende klassisk ASP fremfor ASP.NET.

I forhold til begrænsninger vil det i forbindelse med f.eks. en fuld sprogversionering blive nødvendigt med en fuld omlægningen fra klassisk ASP til ASP.NET.

Winformatik er oprindeligt lavet i klassisk ASP, men nyere moduler og omskrivninger har medført, at det i dag kun er ca. 20% af siderne, der stadig er i klassisk ASP. En omskrivning af de resterende sider er i gang, men det er lavt prioriteret i backloggen, så det skridt kun langsomt frem.

Dokumentet Webpage_name_status.xls [13] er et arbejdsdokument, der beskriver de enkelte sider og hvilken teknologi de er lavet i. Dertil er det en indikation af, hvor kompleks en omskrivning ville være og hvorvidt en omskrivning er i gang.

Al datatilgang i klassisk ASP sker via et tyndt data access lag som tilvejebringes af en række COM+ komponenter udviklet i Visual Basic version 6. Visual Basic udvikles ikke længere, og en 64 bits version af disse komponenter er ikke tilgængelig. Også af denne årsag er afvikling af klassisk ASP en opgave, der må prioriteres.

Det kan bemærkes, at de klassiske ASP sider i gennemsnit har nået/når en praktisk levetid på ca. 10 år, hvilket må betragtes som værende ganske tilfredsstillende.

Der arbejdes således pt. med at genudvikle alle sider, der benytter data access, i ASP.NET. På nuværende tidspunkt er alle del-systemer, undtagen bogh, inka og soc genudviklet (planen er at færdiggøre dette arbejde ved udgangen af 2013). Omskrivningen foregår under iagttagelse af kodeteknik, der forbereder Winformatik til brugen af ASP.NET masterpages samt forøger sidernes portabilitet til andre sammenhænge.

Summarisk liste over sites med klassisk ASP og deres status ses underneden. Et komplet overblik kan fås i dokumentet Webpage_name_status.xlsx [13]:

- /Bogh: Ca 22 sider heraf ca. 11 sider med RDS resterer
- /Inka: Ca 16 sider, heraf ca. 7 sider med RDS resterer
- /Soc: Ca 15 sider, heraf ca. 9 sider med RDS resterer

10.4 Krav til platform

Krav til platform kan opdeles i to; hardwarekrav og softwarekrav. I Winformatiks tilfælde er der ingen formelle krav til hardware, dog anbefales der et setup der anvendes i de sammenhænge, hvor Kimik iT i øvrigt er generel IT leverandør. Setupet afspejler et forsøg på at holde hele IT platformen så homogen som muligt. De anvendte leverandører er:

- Servere: IBM
- Arbejdsstationer: Lenovo
- Printere: Hewlett Packard

Der foreligger ingen konkrete krav til størrelsen - i forhold til f.eks. CPU ydelse - af de valgte servere. Her er det primært kommunens økonomiske formåen på området, der afgør, hvad der indkøbes. Dog anbefaler Kimik iT gerne systemer baseret på det konkrete behov. Da systemet ikke skal servicere tusindevis af samtidige brugere og ikke indeholder flere millioner borgere, er serverens performance typisk af mindre betydning i forhold til andre faktorer, der har indflydelse på performance.

I forhold til softwarekrav ligger der pt. et krav til, at applikationsserveren er en Microsoft Windows 2003. Bindingen til 2003 versionen er noget, man fra Kimik iT's side er ved af fjerne, således at en nyere version kan anvendes. Database-serveren skal blot kunne afvikle en Oracle database. Her er SUSE Linux Enterprise Server (SLES) anvendt. I princippet kunne andre platforme og teknologier anvendes, men disse er anbefalet og anvendes i de kommuner, hvor Kimik iT i øvrigt er IT leverandør. Principielt set er eneste formelle krav, at applikationsserveren kan køre en IIS, og databaseserveren kan køre en Oracle database.

10.5 Vurdering af teknik og infrastruktur

Tanken bag systemet er efter kommunesammenlægningen blevet udfordret, i det både infrastruktur, arkitektur og design er underlagt forhold, der ikke fandtes før. Planlægningen af og arbejdet med kommunesammenlægningen havde haft gavn af flere midler og mere fokus. Det er ikke voldsomme ændringer, systemet generelt skulle undergå for i højere grad at kunne fungere i det nye miljø. I stedet for har man valgt at foretage ændringerne løbende og kun hvor strengt nødvendigt, hvilket naturligvis betyder, at en tæt dialog mellem leverandør og kunde er helt afgørende. Det teknologiske road-map for Winformatik som Kimik iT har udarbejdet, er både fornuftig og nødvendig. Implementeringen af den er dog med til at sætte pres på videreudviklingen af produktet (ny funktionalitet).

På grund af begrænsede midler og en lang historik, er der store forskelle på design, kode og anvendte teknologier på tværs af systemets moduler (og i nogle sammenhænge sider). Derfor vil det sandsynligvis indebære en del arbejde at "løfte" hele produktet til en ny teknologi eller platform. Samtidig kan man derfor konstatere, at produktet ikke fremstår homogent, ud fra en rent teknisk synsvinkel.

11 Brugervenlighed

At give en objektiv vurdering af et systems brugervenlighed er ikke nogen simpel opgave. Oftest vil dette involvere at identificere vigtige elementer for brugervenligheden, så som; usability, funktionalitet og indhold. For at kunne måle og kvantificere disse tilsyneladende uhåndgribelige elementer, kunne man f.eks. lade eksperter og/eller almindelige brugere svare på udsagn, der sagde noget om hver af disse elementer. Således ville man få en empirisk analyse, der ikke blot var baseret på gisninger og personlig smag.

I denne sammenhæng er en sådan analyse uden for scope. Derfor er det følgende forslag/konstatering primært en observation af, hvordan funktionaliteten er udført, og i mindre grad hvordan og i hvor brugervenligt den er udført. Dog forsøges det at vurdere, hvordan forholdet mellem funktionalitet og indhold hænger sammen.

11.1 Idéen bag brugergrænsefladen

Overordnet set deles hele systemet op i del-systemer. Disse vælges ved hjælp af en valgboks. Derefter vælges den ønskede side. Disse sider er nummererede (der findes en funktion der direkte kan gå til et givent side-nummer) og logisk set opdelt i moduler - dog vises alle sider i ét del-system samlet i én valgboks og er således på samme niveau.

Når del-system og side er valgt, vil brugeren typisk skulle udsøge f.eks. en borger ved hjælp af nogle søgekriterier, der resulterer i en liste over matchende borgere (records). Herefter vælges der, hvilken record man ønsker at se detaljer for. Denne sidste del kaldes ofte for search-list-detail, altså man søger, får en liste resultater og vælger at se detaljer for én record. Dette er et mønster, der ofte ses hos databasebaserede webapplikationer, dels fordi mønsteret passer godt i forhold til udsøgning og visning af den type data, men også fordi der ofte er en ganske veludbygget værktøjsunderstøttelse af netop dette mønster.

Typisk er den afgørende forskel på brugergrænseflader forankret i hvilken context data vises i. I Winformatik sammenhæng er context; siden, altså funktionaliteten (funktions context). Således bliver det let at se data for forskellige borgere/records inden for samme emne/side, da man allerede har valgt funktionen og blot skal søge og vælge (to handlinger) en ny borger/record. Modsat er det mere omstændigt at få en anden type data om samme borger, da man her skal ud og vælge del-system, side, søge borger/record og vælge (fire handlinger).

En anden context kunne f.eks. være en borger (borger context). Således kunne man fremsøge og vælge en borger (to handlinger), hvorefter forskellige data for borgeren i teorien kun var ét klik væk (én handling).

Forskellene på context kan synes teoretisk, men den er afgørende for, hvordan systemet opleves af brugeren. For Winformatiks vedkommende er den valgte context valgt ud fra en betragtning om, at én given bruger primært arbejder med én side/funktion, og altså ikke har brug for at se data for en borger på tværs af sider/funktioner og dermed en række unødvendig information.

Muligheden for at give brugeren et større overblik (borger context) over borgeren på socialområdet har dog været drøftet men er endnu ikke blevet udviklet.

Brugergrænsefladen vil sandsynligvis af mange blive anset for værende en smule gammeldags i forhold til, hvad man normalt oplever på nettet generelt. Dette er Kimik iT klar over, og begrundet det med, at kunden ikke har ønsket at anvende billeder og grafik, der ville gøre anvendelsen langsommere. For en webapplikation af denne art, er data transmissionen bl.a. er afgørende for applikationens responstid.

11.2 Sprogversionering

I forhold til systemets mulighed for sprogversionering sondres der imellem sproget på selve brugergrænsefladen og sproget i det materiale, der produceres af systemet, f.eks. breve, rapporter og udskrifter.

For det produceredes materiales vedkommende er alt materiale dobbeltsproget (grønlandsk og dansk), det vil sige at man kan vælge, hvilket sprog det er på, eller man kan få materialet på begge sprog.

Når det gælder selve brugergrænsefladen, er den primært på dansk. De nyeste moduler, der er blevet udviklet, er dog blevet udviklet som flersprogede moduler, hvilket betyder, at brugeren kan vælge sit primære sprog centralt, hvorefter alle moduler der er flersprogede så vil optræde på dette sprog.

Cirka 5% af brugergrænsefladen er p.t. dobbeltsproget.

Muligheden for at gøre flere moduler dobbeltsprogede er umiddelbart tilgængelig. Det kræver naturligvis en oversættelse af alt på brugergrænsefladen og derudover en, for de fleste siders vedkommende, begrænset udviklingsmæssig indsats. Dette gælder som nævnt kun for de sider, der er udviklet i ASP.NET, da systemet teknisk set er en blanding af ASP.NET og klassiske ASP sider, hvor kun ASP.NET sider umiddelbart understøtter den valgte sprogversionering.

Cirka 20% af alle sider er stadig udviklet i klassisk ASP, hvorfor en sprogversionering af disse er noget der må forventes at tage ekstra ressourcer.

Skulle klassiske ASP sider sprogversioneres ville det indebære en omskrivning af disse sider. Der er p.t. en omskrivning af de klassiske ASP sider i gang, men denne proces er langsom og kræver ressourcer.

11.3 Vurdering af brugergrænsefladen

Siderne i systemet er som sagt nummererede. Numrene bruges i mange sammenhænge og for at opnå en effektiv anvendelse af systemet, forudsættes det næsten, at man husker de sidenumre man primært arbejder med. Dette forhold gør, at systemet ikke er umiddelbart let tilgængeligt for nye brugere. Dog kan det begrænsede antal sider, en bruger typisk arbejder med, hurtigt læres, og det er netop brugerens begrænsede berøringsflade af systemet, der gør at dette er muligt, og dermed i realiteten gør løsningen ganske anvendelig for de fleste brugere af systemet.

I forhold til hvordan funktionalitet og indhold hænger sammen, ville systemet sandsynligvis have gavn af en tættere sammenhæng mellem sider inden for de enkelte moduler. Således kunne man stadig imødesee behovet for, at den enkelte bruger primært arbejder inden for et ganske snævert fagområde. På denne måde kunne man anvende borger context på minimum modul niveau. Denne betragtning er dog udelukkende af teoretisk karakter og ville naturligvis skulle underbygges af en dialog med slutbrugerne. Dette er også blevet drøftet i forbindelse med at kunne give brugeren et større overblik i socialmodulet.

12 Winformatik i et åbent marked

Winformatik er udviklet til ét marked og for ét marked – det grønlandske. Det betyder, at produktet aldrig har været tænkt som en generel platform (bortset fra nogle af de komponenter Kimik iT har opbygget det af) men en konkret løsning til et meget konkret problem. Det er netop dette forhold, der er afgørende for en lang række af de beslutninger og løsninger, der er beskrevet i nærværende dokument. Det er således ikke hensynet til at udvikle et system til en bredere målgruppe eller vælge arkitektur og design ud fra hensyn til åbenhed eller integration, der har været vægtet højt. Det, der er blevet højt prioriteret er at kunne levere funktionalitet til en pris, de grønlandske kommuner ville/kunne betale og efterfølgende levere en konsistent, kompetent og vedvarende service.

Det er svært at forestille sig en ny leverandør overtage drift og udvikling af Winformatik. Alene den årsag, at produktet har en historie på over 20 år gør det umiddelbart urealistisk. En leverandør ville i givet fald skulle investere så mange ressourcer i en overtagelse (dette til trods for, at systemet er designet og kodet på en sådan måde, at det ikke umiddelbart fremstår komplekst, se mere i afsnit 6.4.2.1 ovenfor), at det vil være vanskeligt at se forretningsmodellen

hænge sammen for en sådan leverandør. Hvis en konkurrenceudsættelse af Winformatik overhovedet skulle kunne være mulig, ville den sandsynligvis skulle ledsages af en kontraktlig garanti for, at man kunne regne med at drifte og udvikle det en del år ud i fremtiden.

På den anden side er der spørgsmålet om en konkurrenceudsættelse ville være i kundernes interesse. Det er ikke vurderingen, at det ville være en beslutning, der havde en positiv indvirkning på produktets udvikling og drift i dets nuværende form.

12.1 Winformatik som ERP-system

For at vurdere Winformatik i forhold til hvad et ERP system skal kunne, er det nødvendigt at definere hvad reference systemet kan. En definition på begrebet ERP er ikke tilstrækkeligt her, da et ERP system kan anvendes i mange sammenhænge og derfor have mange forskellige afskygninger.

Der er blevet udarbejdet et delprojekt; "1.9.0 Økonomistyring og økonomisk ledelsesinformation i kommunerne", hvor slutproduktet indeholder et servicekatalog, der beskriver områder, som ville kunne blive en del af et fremtidigt ERP-system. En gennemgang af Winformatik i forhold til disse områder vil, på nuværende tidspunkt, være en for omfangsrig opgave.

I de efterfølgende afsnit er en række relevante områder undersøgt.

12.1.1 Økonomistyring

Winformatik har både kreditor, debitor, kontoplan, bogføring osv. Om det dermed er på højde med et decideret ERP system er nok for meget at sige. Systemet har de helt basale økonomiske elementer til at løfte kommunernes behov, men er sandsynligvis ikke tilstrækkeligt, når kompleksiteten stiger, hvis der skal arbejdes mere med effektmåling, aktivitetsbaseret budgettering og økonomistyring baseret på omkostningsprincipper samt mål- og rammestyring i fremtiden.

12.1.2 Forretningsprocesser

Det er her, hvor Winformatik har sin styrke. Systemet er skræddersyet til de behov, der er i de grønlandske kommuner, og det er vurderingen, at der ikke er mange steder i systemet, hvor eksisterende funktionalitet ikke dækker de enkelte behov. Disse forretningsprocesser, og en række andre der udføres direkte på papir eller i Excel, ville skulle genimplementeres i et nyt ERP system - tilretninger der nok i særdeleshed ville være det, der blev dyrt.

12.1.3 Ledelsesinformation

Winformatik har indbygget mulighed for at generere rapporter (Crystal Reports). Svagheden er dog, at disse skal laves af Kimik iT på opfordring af kunden. Moderne ERP systemer har ofte en Business Intelligence (BI) løsning integreret hvor det er muligt, med de rette evner (kræver bla. indgående kendskab til en rapport designer), at definere standardrapporter samt at udtrække vilkårlige rapporter og lister.

Vurdering af nytteværdien for slutbrugere af at kunne tilgå "rå data" direkte har hidtil konkluderet i langt de fleste tilfælde at ville stille uforholdsmæssigt store krav til de som skulle benytte en sådan adgang. I det fælleskommunale KOMLIS-projekt (Kommunal Ledelsesinformation) er der derfor truffet det valg i stedet at basere sig på en række aggregerede talmaterialer, som KAQA udsender flere gange hver måned sammen med konkrete vejledninger og forslag til mulige analyser og evt. prognostisk anvendelse.

12.1.4 Organisationsunderstøttelse

Da kommunerne altid har benyttet fælles offentlige løsninger til Løn (pt. Akissarsiat) og HR-funktionalitet (pt. Webtas / MitPerspektiv), har det ikke været aktuelt at udvikle dette i Winformatik.

Der findes funktioner i Winformatik, der kan henholdsvis kontrollere og bogføre lønkørsler. Desuden findes et udtræk, der henter data i lønsystemet og præsenterer disse i Winformatik-kontekst (f.eks. som input til budgetlægning/opfølgning). Desuden har Winformatik et tidsregistreringsmodul der dog pt. ikke anvendes i kommunerne.

12.1.5 Kimik iT's vurdering

Winformatik er udviklet af de grønlandske kommuner på grundlag af de konkrete ønsker, prioriteringer og økonomiske rammer, som kommunerne har valgt at fastsætte for deres fælles administrative løsninger henover de sidste 25 år.

Fokus har lige siden starten været lagt på målrettethed i forhold til opgaven og organisationens praktiske formåen samt på stabilitet og troværdighed over tid til sikring af størst mulig return of investment (ROI). Løsningernes delelementer er dermed blevet udviklet i takt med, at kompleksiteten har ændret sig.

Med afslutningen af den igangværende teknologiske opgradering til seneste .NET framework og Microsoft serverplatform taler al sandsynlighed for, at denne udvikling vil kunne fortsætte, og at Winformatik også i fremtiden vil kunne bygge bro mellem opgavernes kompleksitet og det organisatoriske kompetenceniveau inden for det offentlige i Grønland.

Det væsentligste i den sammenhæng er dog nok, at man (red. kommunerne) gennem sit ejerskab til løsningerne har sikret sig adgangen til suverænt at fastsætte udviklingstakten – og dermed kontrollen over heraf udløste krav til kompetenceniveau på medarbejderside. Hertil kommer de enkelte kommuners autonome adgang til at beslutte hvilke dele af systemerne, der skal idriftsættes lokalt og hvornår.

For så vidt en vurdering af Winformatik's anvendelighed uden for kommunerne kan det være svært at få øje på konkrete områder, hvor kompleksiteten i den eksisterende kommunale administration ikke er absolut på højde med eller overstiger, hvad der er gældende for de tilsvarende områder i f.eks. Selvstyret.

Det kan derfor være tilsvarende svært at få øje på objektive årsager til at Winformatik – baseret på absolut nyeste teknologier – ikke skulle kunne være en reel kandidat til en fællesoffentlig administrativ platform for de øvrige myndigheder i Grønland.

Der vil utvivlsomt skulle ske en væsentlig udbygning af systemet ligesom helt nye områder skulle opdyrkes, men omkostningerne hertil vil alt andet lige næppe blive meget større end ved skulle tilpasse et vilkårligt andet standardsystem. Tværtom må man nok i en vis udstrækning kunne forestille sig, at et system, der allerede i sin grundform er dedikeret til offentlig administration i Grønland, og samtidig kendt af og i daglig drift hos en stor del af brugerorganisationen, hurtigere og med færre ressourcer vil kunne bringes til at fungere på tværs af den konkrete kontekst end en ny og for brugerne ukendt og mere generisk løsning med afsæt i en ikke-grønlandsk kontekst.

13 Bidragsydere

Herunder findes alle de personer der i et eller andet omfang har bidraget med information og viden. De listes med; navn, titel, firma, bidrag og e-mail adresse:

Anders Djursaa (AD), sektionschef for udviklingssektionen i Kimik iT

Indledende kontakt, review

ad@kimik-it.gl

Peter Trier (PT), seniorkonsulent (særligt for Winformatik) i Kimik iT

Overblik, dokumentation, teknik, database, review o.m.a.

pt@kimik-it.gl

Henrik Jacobsen (HJ), systemudvikler i KIMIK iT

Database, kode o.l.

hj@kimik-it.gl

Ian N. Wennefeldt (INW), administrerende direktør i Kimik iT

Overblik, historik, teknik, integration og sikkerhed, review

inw@kimik-it.gl

Anders Hoffgaard (AHOF), projektleder på ERP Pisariillisaaneq-projektet.

Arbejdspakke, review.

ahof@nanoq.gl

Martin Ib Bruun (MIB), cand. jur. Departement for Finanser og Indenrigsanliggender

Juridiske overvejelser ift. registerloven, persondataloven, arkivloven og lignede.

mbru@nanoq.gl

Lene Gøhlmann Madsen (LGM), IT-chef, Kommuneqarfik Sermersooq

Arbejdspakke, review.

legm@sermersooq.gl

Ole Peter Kristensen (OPK), specialkonsulent (IT), KANUKOKA

Arbejdspakke, review.

opkr@kanukoka.gl

14 Bibliografi

- [1] Oracle, "Oracle Database 11g Release 2," [Online]. Available: <http://www.oracle.com/technetwork/database/enterprise-edition/overview/index.html>.
- [2] Kimik iT, »Database_conventions.doc,« 2010.
- [3] Kimik iT, »Winformatik grants.xlsx,« 2013.
- [4] Microsoft, »Three-Layered Services Application,« [Online]. Available: <http://msdn.microsoft.com/en-us/library/ff648105.aspx>. [Senest hentet eller vist den april 2013].
- [5] Grønlands Nationalmuseum og Arkiv, »Bestemmelser om bevaring og kassation af arkivalier hos de grønlandske kommuner,« 1991. [Online]. Available: http://www.arkiv.gl/Portals/0/Dokumenter/Bevaringkassation_komm_dk.pdf.
- [6] Kimik iT, »Winformatik strateginotat om teknologi - version marts 2013.doc,« 2013.
- [7] Kimik iT, »Description of interface between winformatik and word.doc,« 2012.
- [8] Kimik iT, »WINFORMATIK SUPPORT,« Kimik iT, [Online]. Available: <http://support.kimik-it.gl/winfosupport/>.
- [9] T. I. I.-U. Chefkonsulent Stephen Biering-Sørensen, »Udrulning af it-projekter,« Teknologisk Institut, 2005. [Online]. Available: http://www.teknologisk.dk/_root/media/29338_Udrulning.pdf.
- [10] Datatilsynet, »Lov om offentlige myndigheders registre (Registerloven) - Lov nr. 294 af 8. juni 1978,« 8 Juni 1978. [Online]. Available: <http://www.datatilsynet.dk/internationalt/groenland/lov-om-offentlige-myndigheders-registre/>.
- [11] Datatilsynet, »Lov om behandling af personoplysninger (Persondataloven), Lov nr. 429 af 31. Maj 2000,« 31 Maj 2000. [Online]. Available: <https://www.retsinformation.dk/Forms/r0710.aspx?id=828>.
- [12] Grønlands Hjemmestyre, »Landstingslov nr. 22 af 30. oktober 1998 om arkivvæsen.,« Oktober 1998. [Online]. Available: <http://arkiv.lovgivning.gl/gh.gl-love/dk/1998/22-98.htm>.
- [13] Kimik iT, »Webpage_name_status.xls«.
- [14] Statens arkiver, »Bekendtgørelse af arkivloven (Arkivloven) - Lov nr. 1050 af 17. december 2002 med de ændringer, der følger af § 1 i lov nr. 563 af 24. juni 2005 og lov nr. 532 af 6. juni 2007.,« 4 September 2007. [Online]. Available: <https://www.retsinformation.dk/Forms/r0710.aspx?id=12066>.
- [15] Kimik iT, »Design Dokument - borgerservice.docx,« 2012.
- [16] ERP Pisariillisaaneq, »Arbejdspakke 1 9 1 3 2 Snitflader Winformatik til Kimik-iT.doc,« 2013.
- [17] Det Fælleskommunale IT-samarbejde i Grønland, »Projektorganisation winformatik.doc,« 2011.

15 Begreber

Assembly

Egentlig et engelsk ord der beskriver et kompileret software komponent. Som udgangspunkt kan der ikke ændres i assemblies, men uden egentlig sikkerhedsforanstaltninger kan de inspiceres med særlige værktøjer.

Cyklomatisk kompleksitet

Cyclomatic complexity er et begreb, der beskriver kompleksiteten af den sourcekode, der ligger bag en applikation. (http://en.wikipedia.org/wiki/Cyclomatic_complexity)

Deploymentdiagram

Et deploymentdiagram viser en model af den fysiske udlægning af noder i et givent system. Det kunne f.eks. være webservere, applikationsservere og databaseservere, samt hvilken software der kører på dem og hvordan de internt er forbundet.

DMZ

DMZ er et begreb der dækker over en særlig del af netværket der indeholder organisationens eksternt vendte services ([http://en.wikipedia.org/wiki/DMZ_\(computing\)](http://en.wikipedia.org/wiki/DMZ_(computing)))

Exploratory testing

Exploratory testing er en test metode, der lader testeren frit udforske systemet. Der kræves dermed en særlig indsats fra testers side. Metoden kræver dygtige testere der kender problem domænet. Metoden anses ofte som det modsatte af scripted testing, hvor udvalgte forløb bliver scriptede for efterfølgende at kunne blive 'kørt' ensartet gang på gang (http://en.wikipedia.org/wiki/Exploratory_testing).

Flexibility

"The ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed." (IEEE Standard Computer Dictionary)

Internet Information Services (IIS)

IIS er Microsofts web server software som kan afvikles på Microsoft Windows.

Kunde

Ved kunde menes der, hvis ikke andet er specificeret, de grønlandske kommuner.

Leverandør

Ved leverandør menes der Kimik iT der har udviklet systemet sammen med kommunerne fra bunden.

Maintainability

"The ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment." (IEEE Standard Computer Dictionary)

Namespace

Inden for programmering angiver et namespace en gruppering af logisk sammenhængene objekter/symboler.

Problemdomæne

Et problemdomæne kan opfattes som et velafgrænset fagområde. Det omfatter kun emner der er af interesse for at løse et givent problem.

Product backlog

Product backlog er et begreb, der stammer fra softwareudviklingsprocessen SCRUM. Begrebet dækker over en prioriteret liste af krav/funktionaliteter til den software der udvikles. Listen vedligeholdes i et samarbejde mellem udvikler og product owner (kunde) men prioriteres af product owner.

Regressionstests

Regressionstests er tests der søger at afdække nye fejl i eksisterende funktionalitet på baggrund af andre nye funktionaliteter, konfigurationsændringer eller blot ændringer i den eksisterende kode.

Reusability

"The degree to which a software module or other work product can be used in more than one computing program or software system." (IEEE Standard Computer Dictionary)

Scalability

"The ease with which a system or component can adjust to changing load." (IEEE Standard Computer Dictionary)

SCRUM

SCRUM er en iterativ og inkrementel agil softwareudviklingsproces (<http://en.wikipedia.org/wiki/Scrum>)

Sourcekode (source code)

Den kode der skrives af en softwareudvikler og efterfølgende compiles (bygges) til færdigt software.

Tier

1-tier, 2-tier og n-tier er angivelser af software arkitektur der er opdelt i tiers/lag, således at hvert lag har hvert sit ansvarsområde. Begrebet kan anvendes i forskellige perspektiver. (http://en.wikipedia.org/wiki/Multitier_architecture her specifikt om client-server arkitektur)

Use cases og user stories

Begreber fra udbredte software udviklings processer (unified process og agile development) der beskriver brugeres anvendelse/brugsmønstret for en given funktionalitet i et system.

Vandfaldsmodel

En sekventiel softwareudviklingsprocess der blev udviklet i halvfjerdserne. Processen har i de seneste år fået hård konkurrence fra mere agile udviklingsprocesser og man er, fra flere sider, direkte blevet anbefalet ikke at anvende den til store og komplicerede projekter (http://en.wikipedia.org/wiki/Waterfall_process).

Web Analytics

Web Analytics Software indsamler og præsenterer data om et websites besøgende. (http://en.wikipedia.org/wiki/Web_analytics)

Webinar

Et webinar er en møde/konference form, der foregår over potentielt store geografiske afstande ved hjælp af en internetforbindelse.

16 Figuroversigt

Figur 1 Generel opbygning af Winformatik.....	6
Figur 2 Et overblik over modulers afhængigheder	8
Figur 3 Modulernes afhængigheder	9
Figur 4 Diagram over interne afhængigheder, farvekodet og udlagt i forhold til primær afhængighed (System administration udeladt)	10
Figur 5 Social del-systemets moduler og disses eksterne afhængigheder	10
Figur 6 Pension modulets afhængigheder	11
Figur 7 Pensions interne afhængigheder på side-niveau	11
Figur 8 Overblik over eksterne afhængigheder	14
Figur 9 Tre lags service applikation [4].....	16
Figur 10 Forsimplet lagdelingsdiagram med social del-modul som eksempel	17
Figur 11 Deployment af borgerservice	18
Figur 12 Deployment diagram for Qaasuitsup fra 2009	22
Figur 13 Sikkerhedsmæssig deployment af Winformatik	25
Figur 14 Udviklingssekvens, fra kode til produktion	29
Figur 15 Deployment diagram for produktionsmiljø fra 2009.....	35

17 Index

- Arkitektur; 14; 15; 21; 22; 23; 38; 40
- ASP.NET; 6; 18; 28; 36; 37; 38; 40
- Assemblies; 15; 17; 18; 23
- Authentication; 24; 25
- Best-practice; 16; 25
- Bogføring; 8; 10; 41
- Borgerservice; 18; 22
- Brugergrænsefladen; 6; 28; 31; 39; 40
- Context; 39; 40
- CPR; 13; 19; 23; 30; 31; 32
- Debitor; 8; 41
- Deployment; 15; 21; 25; 26; 29; 34; 35
- Design; 14; 15; 21; 22; 27
- Dokumentation; 5; 20; 21; 22; 23; 24; 43
- Firewall; 18
- Hardware; 21; 27; 38
- Ikke-funktionelle krav; 21
- Integration; 12; 17; 19; 22; 40; 43
- Kommunesammenlægningen; 19; 35; 36; 38
- Krav; 4; 19; 21; 23; 24; 26; 27; 29; 31; 38; 41; 42
- Kravspecifikation; 21
- Lag; 15; 17; 37
- Latency; 36
- Logning; 30
- Lov; 31; 32; 33; 34
- Microsoft; 6; 15; 17; 25; 26; 28; 37; 38; 42
- Object Relational Mapping; 16
- Oracle; 6; 7; 16; 17; 28; 38
- Oracle Forms; 6
- Product backlog; 23; 28
- Rapporter; 39; 41
- Road-map; 21; 37
- Sagsmoduler; 9
- Schema; 7
- Scope; 20; 39
- SCRUM; 27
- Sikkerhed; 25; 26; 32; 43
- Snitflader; 12; 14; 33
- SOA; 15
- SQL; 7; 16; 17; 28
- Stored procedures; 7; 17
- Test; 28; 29
- Usability; 39
- Use cases; 21
- User stories; 21
- Virtualisering; 26; 35; 36
- Vurdering; 23; 26; 29; 34; 38; 40; 41
- Webapplikation; 14; 15; 39
- Webinar; 23
- Webservice; 12; 13; 14; 17; 18
- Windows; 18; 24; 25; 30; 38
- Økonomimoduler; 9