

# FORORD

Denne oppgaven er et resultat av min mastergradsstudie ved Institutt for Informatikk ved Universitetet i Oslo, studieretning informasjonsdesign.

Først og fremst vil jeg takke min veileder Ingvil Hovig for veiledning i forbindelse med valg av problemstilling, og for å presentere meg for personale i Skattedirektoratet (SKD) som har bidratt med informasjon. Hun har vært veldig tålmodig og har med stor innsikt ledet meg gjennom arbeidet med oppgaven.

Deretter vil jeg gjerne takke Knut Fiskerud som har gitt meg problemstilling til oppgaven, og som gjennom mange samtaler har gitt meg mulighet til å nøste opp i denne.

Dernest vil jeg takke IT-avdelingen i Skattedirektorat for å ha stilt arbeidsplass til rådighet for meg under arbeidet med oppgaven. Takk for verdifull hjelp og positiv innstilling fra ansatte i IT-avdelingen.

Jeg vil rette en stor takk til Lars Holter for hans tålmodighet gjennom lesing og oppretting av språket i min oppgave.

Sist vil jeg også spesielt takke min familie og mine venner som har støttet meg gjennom hele mitt studium.

Oslo den 29.04.2006

Huynh, Thi Phi Thuyen



## **SAMMENDRAG**

Mange norske bedrifter har gjort store investeringer for å forbedre både programvare og maskinvare, men oppnår ikke den ønskede gevinst fra investeringen. Oppgaven beskriver problemer ved administrasjon av IT-ressurser og foreslår hvordan situasjonen kan forbedres.

Skattedirektoratet (SKD) er en stor organisasjon som tilbyr flere tjenester til blant annet innbyggere, ligningskontor, folkeregister og forskjellige skattekontor. SKD er i stadig utvikling, med et stort antall brukere av tjenestene de tilbyr. For å tilby bedre tjenester på en effektiv og profesjonell måte, vil SKD ha bedre oversikt over sin IT-infrastruktur. Blant mange såkalte rammeverk for å administrere IT-ressurser, har IT-avdelingen i SKD valgt å bruke ITIL.

I tillegg til å presentere SKD fra et IT-tekniske perspektiv, presenterer oppgaven også ITIL-rammeverkets teori generelt og går detaljert inn i prosessene i ”Service Support” modulen som brukes i oppgaven.

Denne rapporten beskriver prosessstilpasning i SKD i tråd med ITIL rammeverket. Den beskriver hvilke prosesser som er valgt, beskriver tilpasningen av de valgte prosessene til arbeidet i organisasjonen og dennes arbeidsrutiner. Prosessene forutsetter en konfigurasjonsdatabase ("Configuration Management Database", CMDB) med identifiserte og klassifiserte konfigurasjonsenheter, og denne skisseres med et gitt eksempel på en relasjonsdatabase ved hjelp av NIAM. Det følger også med grupperte tabeller for aktuelle innholdselementer med attributter, primærnøkler og fremmednøkler. Noen eksempler på SQL-spørringer er også inkludert.

Denne oppgaven kan hjelpe Skattedirektoratet ved innføring av ITIL-prosessene i sin organisasjon. De kan på bakgrunn av rapporten vurdere om de skal bygge eller kjøpe en CMDB.

<b>FORORD</b> .....	<b>1</b>
<b>SAMMENDRAG</b> .....	<b>3</b>
<b>1 INNLEDNING</b> .....	<b>8</b>
1.1 ITIL.....	8
1.2 SKATTEDIREKTORATET (SKD) OG ITIL .....	8
1.3 BESKRIVELSE AV PROBLEMSTILLING .....	8
1.3.1 Utdyping av problemstilling.....	9
1.3.2 Hva skal oppgaven bidra til for SKD? .....	10
1.3.3 Avgrensinger.....	10
1.3.4 Framgangsmåte.....	10
1.4 SPRÅK, BEGREPER OG TEORI .....	12
1.5 OPPGAVENS STRUKTUR .....	12
1.6 OPPSUMMERING.....	13
<b>2 BEDRE RESSURSUTNYTTELSE GJENNOM BEDRE PROSESSER OG BEDRE BRUK AV IT VERKTØY</b> .....	<b>14</b>
2.1 HVORFOR ER ADMINISTRASJON AV IKT ET AKTUELT TEMA? .....	14
2.1.1 Bedriften investerer mye i sin IKT-infrastruktur (maskinvare, programvare etc.), men benytter lite av de ressursene som faktisk er tilgjengelige. ....	14
2.1.2 Mangelfull dokumentasjon .....	15
2.1.3 Dårlig kommunikasjon med sluttbrukere .....	15
2.1.4 Mye "workarounds" og brannslukking, lite proaktiv forretningsutvikling.....	15
2.1.5 Ressursene er lite samordnet, blir dårlig utnyttet. ....	15
2.1.6 Dårlig ansvarsfordeling .....	16
2.1.7 Mangelfull kontroll på operasjonell risiko.....	16
2.2 BEDRE ADMINISTRASJON GJENNOM PROSESSFORBEDRING VIA STANDARDISERING.....	17
2.2.1 Rammeverkt for administrasjon av IT-ressurser .....	17
2.2.2 Generelt om ITIL- IT Infrastructure library [9].....	18
2.2.2.1 Hva er ITIL?.....	18
2.2.2.2 Innholdet i ITIL.....	19
2.2.2.3 "Planning to Implement Service Management" .....	19
2.2.2.4 "The Business" - Forretningsperspektivet .....	20
2.2.2.5 "ICT Infrastructure Management (ICTIM)".....	21
2.2.2.6 "Security Management" - sikkerhetsadministrasjon.....	21
2.2.2.7 "Application Management" – administrasjon av applikasjoner .....	21
2.2.2.8 "Service Delivery" .....	22
2.2.2.9 "Service Support".....	22
2.2.2.10 Prosesser i samspill .....	23
2.2.2.11 Noen fordeler som kan oppnås ved bruk av ITIL.....	24
2.3 BEDRE ADMINISTRASJON GJENNOM IT VERKTØY.....	25
2.3.1 Overvåkingsløsning i generelt [11, 12].....	25
2.3.2 Systemovervåking.....	26
2.3.3 Nettverksovervåking.....	26
2.3.3.1 Driftsfilosofi [13] .....	26
2.3.4 Populære overvåkningsverktøy .....	27
2.3.4.1 Kategorisering av verktøy [14].....	27
2.3.4.2 IBM Tivoli monitoring[15] .....	28
2.3.4.3 HP Openview [12, 14].....	29
2.3.4.4 CA Unicenter [14].....	30
2.3.4.5 Andre produkter som finnes og utviklingstrender [14] .....	31
2.4 OPPSUMMERING.....	32
<b>3 SERVICE SUPPORT</b> .....	<b>33</b>
3.1 "INCIDENT MANAGEMENT" (HENDELSESADMINISTRASJON) .....	33
3.1.1 Introduksjon .....	33
3.1.1.1 Formål .....	33
3.1.2 Prosessen: .....	33

3.1.2.1	Aktiviteter for hendelsesadministrasjon .....	33
3.1.2.2	Relasjon med andre prosesser .....	35
3.1.3	<i>Prosessadministrasjon</i> .....	35
3.1.3.1	Rapporter: .....	35
3.1.3.2	Viktige faktorer .....	36
3.1.3.3	Parametre .....	36
3.1.3.4	Funksjoner og roller .....	36
3.1.4	<i>Kostnader og mulige problem</i> .....	36
3.1.4.1	Kostnader .....	36
3.1.4.2	Mulige problemer .....	36
3.1.5	<i>Fordeler</i> .....	37
3.2	"PROBLEM MANAGEMENT" (PROBLEMADMINISTRASJON) .....	37
3.2.1	<i>Introduksjon</i> .....	37
3.2.1.1	Formål .....	37
3.2.1.2	Definisjon .....	37
3.2.2	<i>Prosess</i> .....	38
3.2.2.1	Aktiviteter som ligger under problemadministrasjon: .....	38
3.2.2.2	Relasjon med andre prosesser .....	39
3.2.3	<i>Prosesskontroll</i> .....	40
3.2.3.1	Rapporter: .....	40
3.2.3.2	Viktige faktorer: .....	40
3.2.3.3	Funksjoner og roller: .....	40
3.2.4	<i>Kostnader og mulige problemer</i> .....	40
3.2.4.1	Kostnader .....	40
3.2.4.2	Mulige problem .....	40
3.2.5	<i>Fordeler</i> .....	41
3.3	"CHANGE MANAGEMENT" (ENDRINGSADMINISTRASJON) .....	41
3.3.1	<i>Introduksjon</i> .....	41
3.3.2	<i>Formål</i> .....	41
3.3.3	<i>Prosess</i> .....	41
3.3.3.1	Endringsadministrasjonens aktiviteter .....	41
3.3.3.2	Relasjon til andre prosesser .....	42
3.3.4	<i>Prosesskontroll</i> .....	43
3.3.5	<i>Kostnader og mulige problem</i> .....	43
3.3.6	<i>Fordeler</i> .....	43
3.4	"CONFIGURATION MANAGEMENT" (KONFIGURASJONSADMINISTRASJON) .....	43
3.4.1	<i>Introduksjon</i> .....	43
3.4.2	<i>Prosess</i> .....	43
3.4.2.1	Konfigurasjonsadministrasjonens aktiviteter .....	43
3.4.2.2	Relasjon med andre prosesser .....	44
3.4.3	<i>Prosesskontroll</i> .....	44
3.4.4	<i>Kostnader og mulige problem</i> .....	45
3.4.4.1	Kostnader .....	45
3.4.4.2	Mulige problem .....	45
3.4.5	<i>Fordeler</i> .....	45
3.5	DETTE ER EN CMDB .....	45
3.6	OPPSUMMERING .....	47
<b>4</b>	<b>LITT OM SKATTEDIREKTORATET (SKD) .....</b>	<b>48</b>
4.1	SKATTEDIREKTORATET I TALL. [21] .....	48
4.2	IT-AVDELINGEN I SKD .....	49
4.3	TEKNISK BAKGRUNN .....	49
4.3.1	<i>Datautstyr i generell</i> .....	49
4.3.2	<i>Driftsverktøy</i> .....	50
4.4	BRUKERSTØTTE .....	53
4.4.1	<i>Brukerkontakten på Helsfyr</i> .....	53
4.4.2	<i>Brukerstøtte Service (BSS) avdelingen i Grimstad:</i> .....	55
4.5	OM MIT-PROSJEKTET I SKD .....	56
4.5.1	<i>Formål</i> .....	56
4.5.2	<i>Hvilke ITIL deler skal implementeres i SKD?</i> .....	57

4.5.3	<i>Hvorfor velger SKD å implementere disse komponentene?</i> .....	57
4.6	OPPSUMMERING.....	58
<b>5</b>	<b>ETABLERING AV CMDB I SKD.....</b>	<b>59</b>
5.1	BESKRIVELSE AV AKTUELLE ITIL-PROSESSER SOM SKAL IMPLEMENTERES I SKD .....	59
5.1.1	<i>Generelt</i> .....	59
5.1.2	<i>Hendelsesadministrasjon</i> .....	62
5.1.3	<i>Problemadministrasjon</i> .....	62
5.1.4	<i>Endringsadministrasjon</i> .....	63
5.2	OVERORDNET BESKRIVELSE AV FREMGANGSMÅTE VED BYGGING AV CMDB .....	64
5.2.1	<i>Prosesstilpasning - Etablere aktiviteter og attributter</i> .....	64
5.2.2	<i>Bygge opp CMDB</i> .....	65
5.3	ETABLERING AV AKTIVITETSATTRIBUTTER OG RELASJONER FOR PROSESSER .....	65
5.3.1	<i>Kartlegge formål, oppgave og aktivitet</i> .....	65
5.3.2	<i>Kartlegge rutiner og skissere arbeidsinstruksjoner</i> .....	65
5.3.3	<i>Bestemme prosess attributter</i> .....	66
5.4	BYGGE OPP CMDB.....	66
5.4.1	<i>Identifisere konfigurasjonsenheter</i> .....	66
5.4.2	<i>Skille mellom logiske og fysiske konfigurasjonsenheter</i> .....	67
5.4.3	<i>Tegne CI klassifikasjonsstruktur</i> .....	67
5.4.4	<i>Spesifisere CI attributter for email tjeneste</i> .....	70
5.4.5	<i>Spesifisere CI relasjoner i NIAM modell for Email tjeneste</i> .....	73
5.4.5.1	<i>Beskrivelse av NIAM-diagrammet [33]</i> .....	73
5.4.5.2	<i>Antagelser:</i> .....	74
5.4.5.3	<i>Niam diagram (Figur 25)</i> .....	77
5.4.6	<i>Gruppering av tabeller</i> .....	78
5.4.7	<i>Eksempler på SQL-spørringer</i> .....	81
5.5	OPPSUMMERING.....	82
<b>6</b>	<b>DISKUSJON OG KONKLUSJON .....</b>	<b>83</b>
6.1	DISKUSJON RUNDT SKD SIN PROSESSTILPASNING .....	83
6.2	DISKUSJON RUNDT BYGGING AV CMDB .....	84
6.2.1	<i>Valg av attributter</i> .....	85
6.2.2	<i>Logisk og fysisk database</i> .....	85
6.3	HVA ER OPPNÅDD .....	85
6.3.1	<i>For meg personlig</i> .....	85
6.3.2	<i>For SKD</i> .....	86
6.4	VANSKELIGHETER UNDERVEIS I ARBEIDET .....	86
6.5	UTFORDRINGER FREMOVER .....	86
6.5.1	<i>Avgrensning av informasjon som lagres i databasen</i> .....	86
6.5.2	<i>Datakvalitet</i> .....	87
6.5.3	<i>Databasens struktur</i> .....	87
6.5.4	<i>Integrasjon av verktøy</i> .....	87
6.6	VIDERE ARBEID .....	87
6.6.1	<i>Utvidelse av CMDB</i> .....	87
6.6.2	<i>Utvidelse i bruk av CMDB</i> .....	88
6.7	OPPSUMMERING.....	88
<b>7</b>	<b>ORDLISTE .....</b>	<b>89</b>
<b>8</b>	<b>FIGURER.....</b>	<b>91</b>
<b>9</b>	<b>VEDLEGG .....</b>	<b>93</b>
9.1	PROSESSADMINISTRASJONS-OVERSIKT .....	93
9.1.1	<i>Oversikt over hendelsesadministrasjon</i> .....	93
9.1.2	<i>Oversikt over problemadministrasjon</i> .....	95
9.1.3	<i>Oversikt over endringsadministrasjonen</i> .....	97
9.1.4	<i>Oversikt over konfigurasjonsadministrasjonen</i> .....	99

9.2	TABELL FOR PROSESS ADMINISTRASJON .....	101
9.2.1	<i>Tabell for hendelsesadministrasjon</i> .....	101
9.2.2	<i>Tabell for problemadministrasjon</i> .....	108
9.2.3	<i>Tabell for endringsadministrasjon</i> .....	113
9.3	PROSESS ADMINISTRASJON ATTRIBUTTER.....	117
9.3.1	<i>Attributter for hendelsesadministrasjon</i> .....	117
9.3.2	<i>Attributter for problemadministrasjon</i> .....	120
9.3.3	<i>Attributter for endringsadministrasjon</i> .....	122
9.4	ANNEX 7C: SUGGESTED CI ATTRIBUTES .....	125
<b>10</b>	<b>REFERANSE</b> .....	<b>128</b>

# 1 Innledning

*Kapitel 1 er en introduksjon til oppgavens tema, og en spesifisering av masteroppgavens problemstilling.*

## 1.1 ITIL

System- og nettverksadministrasjon er et stort område som de fleste bedrifter må arbeide aktivt med, for at ulike datasystem og tjenester skal kunne utnyttes på en god måte. Systemvedlikehold kan være å installere ny programvare, oppdatere maskinvare, administrere brukere, sikkerhetskopiering og mest av alt er å rette opp feil som dukker opp under brukers arbeidsprosess. Ved en stor bedrift, er drift en krevende jobb, antall hendelser som oppstår er mange, forandringer i en eller annen enhet i infrastrukturen kan komme til å berøre mange andre enheter, eventuelt en stor andel brukere.

For å gjøre driften lettere, har det blitt utviklet et rammeverk som heter ITIL ("IT Infrastructure Library"). Det er en samling av beste praksis innen administrasjon av IT i bedrifter. ITIL er utviklet med bidrag fra hele verden, over lang tid. ITIL er nyttig for store og mellomstore bedrifter og det er gratis å bruke. Hensikten bak ITIL er bedre tjenester med et fornuftig budsjett, gjennom å holde orden i IT prosesser. ITIL rammeverket tilbyr beskrivelse av prosesser og aktiviteter, fra en forretningsynsvinkel til en teknologisyndsvinkel. Det finnes blant annet applikasjonsadministrasjon, planlegging for å implementere tjenesteadministrasjon, sikkerhetsadministrasjon, tjenestelevering, tjenestestøtte, IKT infrastruktuadministrasjon.

I system og nettverksadministrasjon er det modulene for tjenesteadministrasjon ("Service Support") og tjenestelevering ("Service Delivery") som er mest brukt. Tjenestelevering gir organisasjonene oversikt over hvilke tjenester og hvilke nivå de må holde, for å gi kundene det de er lovet. Mens tjenestestøtte gir rammeverk for prosesser som hendelsesadministrasjon, problemadministrasjon, endringsadministrasjon, konfigurasjonsadministrasjon og til og med hva som bør skje i "Service Desk", samt hvordan "Service Desk" organiseres.

## 1.2 Skattedirektoratet (SKD) og ITIL

Skattedirektoratet videreutvikles kontinuerlig ved å tilby stadig flere elektroniske tjenester. Dagens løsninger er basert på komplekse IT- løsninger, som er en samling av flere del-løsninger. De er blitt implementert på ulike tidspunkt. Driftløsninger er i tillegg fragmentert med forskjellige produkter og brukergrensesnitt.

Skattedirektorat (SKD) har besluttet å bruke ITIL rammeverket og implementere dette gjennom "Modernisering av IT- drift i Skatteetaten (MIT)" prosjektet.

## 1.3 Beskrivelse av problemstilling

En sentral komponent i ITIL er den såkalte CMDB ("Configuration Management Database"). Dette er en database som blant annet gir en samlet oversikt over teknisk informasjon om infrastrukturkomponenter i organisasjonen. Oppgavens målsetting er å



beskrive hvordan man kan bruke ITIL rammeverket som grunnlag, for å designe en database som benyttes i driften.

Følgende vil da være spørsmål som denne oppgaven beskjeftiger seg med:

Hvordan benytte ITIL rammeverket for å designe et forslag til en "Configuration Management Database"(CMDB) for SKD sin drift.

Gjennom arbeidet vil det danne seg et bilde av hvor SKD står i dag, det vil si å skape en oversikt over hvilke tjenester som tilbys, hvem kundene er, og hvordan SKD sin infrastruktur er bygget opp. I tillegg vil jeg få en oversikt over driftsavdelingens rutiner, og hvordan disse kan forbedres gjennom en tilpasning til ITIL-rammeverket.

Det er også nødvendig å forstå hva man vil oppnå ved å bruke ITIL i SKD. Hvilke prosesser fra ITIL ønsker man å inkludere, og hvilken rekkefølge bør velges for implementasjon. Dette innebærer også å se på hvilke prosessaktiviteter hos SKD som kan ha nytte av en CMDB. Hvordan kan en CMDB for SKD se ut? Hvilke data/metadataber bør man registrere, og hvilke avhengigheter bør de ha?

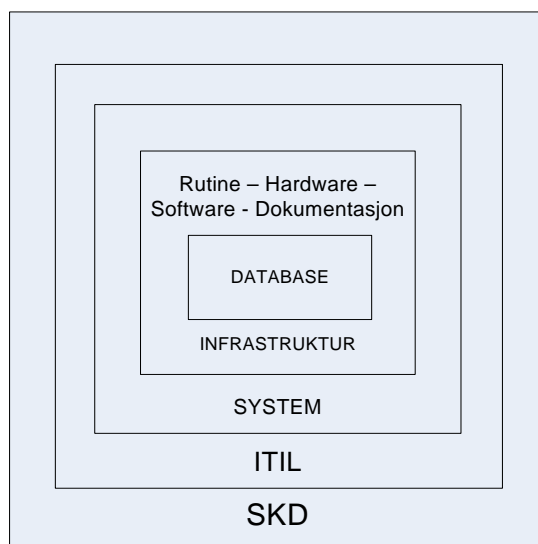
Input til en CMDB er informasjon om SKD sin IT-infrastruktur, i tillegg til data fra kanaler som ulike overvåkningsverktøy, brukere, brukerstøtte, driftspersonell m.m., som prosessene trenger for å fungere.

### **1.3.1 Utdyping av problemstilling**

Oppgaven skal kartlegge SKD sin infrastruktur, som består av alt av data og teleutstyr. Både maskin- og programvare som involveres i design, utvikling, testing og drift av IT-systemer. IT-systemer i denne sammenheng inkluderer bl.a. terminalutstyr, PC-er, skrivere, skannere, nettverk (LAN, WAN), telefonsentraler, servere og datalagringsutstyr. Etatens fagsystemer inngår ikke i begrepet IT infrastruktur, og inngår derfor ikke i oppgaven. [1]

ITIL benytter en CMDB som hjelpemiddel ved utførelse av driftstjenester. Arbeidet med etablering av en CMDB innebærer å trekke frem ulike parametre for å betegne infrastrukturen, trekke veksler på eksisterende data, etablere metadataber for infrastrukturen og relasjoner mellom disse metadatabene. For å utføre ITIL prosessene på en effektiv måte, trengs det støtte i CMDB. Hensikten med en modell for CMDB, er å skissere en oversikt over funksjonalitet for alle prosesser i driftsarbeidet samt gi et enkelt eksempel av CMDB i relasjon database modellering.

Figuren nedenfor illustrerer hvordan oppgaven kan forstås og bygges opp. Det kan oppsummeres slik: oppgaven skal foreslå en skisse av CMDB for den komplekse infrastrukturen av systemer som brukes i SKD, alt i henhold til rammeverket som tilbys via ITIL. (Figur 1)



Figur 1: Spesifikasjon for oppgaven

### 1.3.2 Hva skal oppgaven bidra til for SKD?

Formålet med oppgaven er primært å designe en overordnet konfigurasjonsdatabase. Dette resultatet skal brukes som grunnlag for å vurdere tilbud fra flere leverandører, ettersom SKD skal kjøpe inn ulike ITIL-tjenester.

Kunnskapen oppgaven bidrar med skal også brukes til å vurdere innkjøpt overvåkingsverktøy, samt integrere dette med eksisterende verktøy.

### 1.3.3 Avgrensinger

ITIL er veldig omfattende og kan brukes på mange måter. For å avgrense oppgavens omfang, vil denne oppgaven begrense seg til kun fem av ITIL-prosessene for "Service Management (SM)" (tjenesteadministrasjon), nemlig "Service Level Management (SLM)" (administrasjon av tjenestenivå), "Incident Management (IM)" (hendelsesadministrasjon), "Problem Management (PM)" (problemadministrasjon), "Change Management (CM)" (endringsadministrasjon) og Configuration Management (Con.M)" (konfigurasjonsadministrasjon). Det vil imidlertid bli nødvendig å holde et lite øye til andre prosesser underveis.

Implementeringen av ITIL i den daglige driften er under arbeid i SKD, derfor kan det være en fare for oppgaven at den vil kunne preges av teori og lite praktisk erfaring.

Denne oppgaven tar ikke høyde for eventuelle organisasjonsendringer.

Arbeidet innebærer ikke å se på hvordan prosesser implementeres.

Det skal bare skapes et rammeverk for nettverksadministrasjon og drift, uten å ta hensyn til ekstraordinære driftsavbrudd som kan oppstå ved branntilløp, vannskader etc.

### 1.3.4 Framgangsmåte

For å utføre oppgaven, må man først danne seg et begrep om hva ITIL er og hvilke egenskaper i dette verktøyet som vil fungere best for SKD sine systemer. Man må forstå hva en "Configuration database (CMDB)" er, og hvordan man bygger den? Det er en nyttig ting

å også kjenne til SKD sine gjøremål mer generelt, og etablere en god oversikt over SKD sine systemer og infrastruktur forøvrig.

Først må de ovennevnte felter forstås hver for seg, men det viktigste er hvordan de samspiller. Det skal følge med prosessbeskrivelse i ITIL-rammeverket, steg for steg for hver prosess for å plukke ut aktuelle forslag til metadata i rammeverket. Deretter skal det tilpasses SKD sin kultur, kravspesifikasjonen og ikke minst dagens framgangsmåte, for å finne ut hvilke metadata som er aktuelle, samt relasjoner mellom disse. Metadataene brukes til å utføre prosessene i "Service Support" modulen.

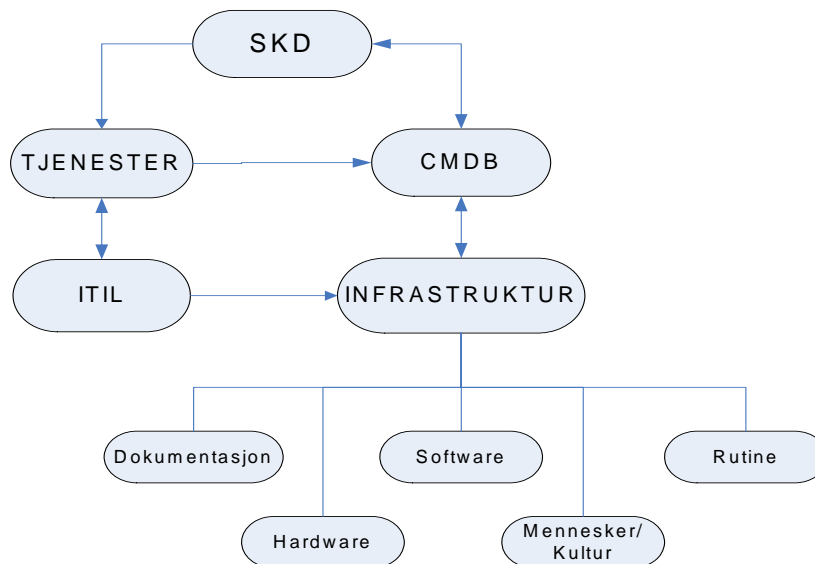
Vi trenger også andre data om organisasjonens infrastruktur. Det være seg innen programvare, maskinvare, dokumentasjon, nettverk, brukere, hvilke tjenester de tilbyr, og hvordan disse ulike elementene knytter seg til hverandre. F.eks. hvem som har tilgang til hva, hvilke typer hardware som brukes, hvor, og sammen med hva (programvare, tjenester, dokumentasjon etc.). Disse dataene skal ligge i CMDB ("Configuration Management Database") sammen med metadata til prosessene.

Dataene for infrastrukturen har flere nivå. F.eks. kan en datamaskin eksistere i kun ett nivå hvor hver enkelt datamaskin er en likeverdig enhet uansett funksjon, eller den kan deles inn i underkategorier som server, printserver, arbeidsstasjon, bærbar etc. Den kan evt. også brytes ned i mindre deler som harddisk, ram, tastatur etc. Den samme problemstillingen gjelder for programvare, tjenester, dokumentasjon etc. Hvor detaljnivået skal gå bestemmes i stor grad av bedriftens/organisasjonens kravspesifikasjon, og må stemme overens med data man registrerer i prosessadministrasjonen.

Denne masteroppgaven vil i tillegg til å reise problemstillingene, diskutere disse, og forsøke å bringe forslag til gode løsninger.

Jeg vil benytte intervjuer med ansatte i SKD i forbindelse med oppgaven for å få innblikk i hvordan situasjonen er i SKD i dag, samt få et innblikk i ulike ønsker og behov i forbindelse med innføring av et nytt system/rammeverk fra forskjellige synsvinkler. I en sån situasjon er intervju en god metode å bruke, selvfølgelig sammenholdt med annen relevant informasjon.

Bildet nedenfor illustrerer relasjon mellom de forskjellige områder i oppgaven. (Figur 2)



Figur 2: Sammenheng mellom ulike felter i oppgaven

## 1.4 Språk, begreper og teori

Oppgaven bruker en del engelske ord siden engelsk er ”hovedspråket” i informatikk. I tillegg er hovedteorien i denne oppgaven (ITIL-rammeverket) full av engelske ord og begreper. Noen ord er imidlertid oversatt, men jeg velger å bruke de engelskspråklige forkortelsene slik at det ikke blir for mye gjentakelse av lange ord. Grunnen til at jeg bruker engelskspråklige i forkortelsene, er at disse figurerer i figurer, tabeller o.l. og det gjør det lettere for leseren å orientere seg. Alle ord og begreper som blir gjenstand for forkortelse vil imidlertid først bli presentert i sin fulle lengde, fulgt av forkortelsen i parentes. I tillegg vil vedlegg til oppgaven inneholde en oversikt over alle forkortelsene for å understøtte leserens mulighet for rask referanse.

## 1.5 Oppgavens struktur

Resten av oppgaven skal gå dypere i teorien som er fundamentet for ITIL-rammeverket, og beskrive hvordan oppgaven utføres.

Kapittel 2 beskriver dagens situasjon i forbindelse med drift av IKT i vanlige bedrifter, og skisserer mulige forbedringsalternativer. I dette kapittelet beskrives to mulige løsninger som kan lede til forbedringer, nemlig gjennom prosesser eller verktøy.

Kapittel 3 beskriver ”Service Support” mer detaljert, ettersom dette er oppgavens hoveddel. Det omtaler blant annet prosessintroduksjon, formål, aktiviteter, hvordan prosessene administreres og fordeler ved prosessene som tas i bruk, til og med kostnader og mulige problemer som kan oppstå. Kapittelet gir også et inntrykk av hva en konfigurasjonsadministrasjons-database (”Configuration Management Database”, CMDB) er.

I kapittel 4 beskriver jeg Skattedirektorat utfra et teknisk perspektiv. Det beskrives blant annet hva som er formålet til organisasjonen, hvor mange ansatte den har, og hvilke typer datautstyr de bruker. Her får vi vite om brukerkontakten, hvordan denne organiseres, og hva

som er dennes ansvar. Deretter gir kapittelet innsikt i hvorfor SKD velger å modernisere drift av IT etter ITIL-rammeverket.

Kapittel 5. Her blir selve arbeidet i oppgaven forklart, helt fra valg av prosesser til databaseforslag. Kapittelet beskriver prosessenes arbeidsforløp, hvordan de passer til organisasjonen, hvordan CMDB bygges opp, og gir et eksempel på en relasjonsdatabase for en enkelt tjeneste.

Kapittel 6 diskuterer hvordan oppgaven er utført med tanke på prosessstilpasning, etablering av CMDB, vanskeligheter, utfordringer, samt resultater som er oppnådd med oppgaven. Kapittelet foreslår også videre arbeid for SKD når prosessene er godt i gang.

## **1.6 Oppsummering**

Kapittelet gir en beskrivelse av oppgavens problemstilling, og enkelte valg som er gjort ved fremstilling av rapporten.

## 2 Bedre ressursutnyttelse gjennom bedre prosesser og bedre bruk av IT verktøy

*Dette kapitlet diskuterer hvordan bedrifter kan styre sine IT ressurser bedre og hva som hindrer dem. Hvilke problemer kan identifiseres ut fra dagens situasjon. Mulige løsninger på noen av utfordringene er bedre administrasjon gjennom bedre prosesser. Her kan ITIL rammeverket være til hjelp. Det gis en overordnet beskrivelse av hva ITIL er, hva rammeverket inneholder, samt fordeler og ulemper ved det i bruk. Prosessen som benevnes "Service support" som er utvalgt for SKD og denne oppgaven omtales i eget kapittel. For å bedre prosessene benyttes verktøy for system og nettverksadministrasjon. Noen av de mest brukte produkter på markedet omtales: IBM Tivoli, HP Openview og CA Unicenter. IBM Tivoli Monitoring, som er det verktøyet Skattedirektoratet bruker i deres drift, skal gåes nærmere etter i sømmene.*

### 2.1 Hvorfor er administrasjon av IKT et aktuelt tema?

Ledelsen i norske bedrifter har fått et økende fokus på administrasjon av IT-ressurser. Dette har skjedd i takt med at IKT-situasjonen har endret seg betraktelig i løpet av de siste årene. Et seminar [2] arrangert av Bekk Consulting tok for seg flere aspekter ved dette temaet og følgende momenter ble trukket frem.

Bedrifter ønsker å levere bedre tjenester, ønsker fornøyde kunder, og ønsker å være konkurransedyktige innen IKT. Derfor investerer de store summer i ny teknologi, både innen programvare og maskinvare. De vil tilby gode løsninger for kundene, bedre driftsløsninger og effektivisere arbeid og arbeidsprosesser, for derigjennom å kutte kostnader.

Dagens situasjon er imidlertid slik at mange bedrifter har det til felles at de ikke klarer å hente ut gevinst på IKT-investeringen. Det være seg økonomisk, arbeidsmessig, eller på andre måter. Og det er mange grunner til dette:

#### **2.1.1 Bedriften investerer mye i sin IKT-infrastruktur (maskinvare, programvare etc.), men benytter lite av de ressursene som faktisk er tilgjengelige.**

Ettersom bruken av IKT økte i omfang, startet de fleste bedrifter å investere i IKT for å følge med i samfunnsutviklingen og å være konkurransedyktige. Dette ble sett på som en naturlig oppgave for bedriftene. IKT er og har vært, en investering som man ikke alltid ser fordelene med umiddelbart. Det tar tid å forandre arbeidsmåter. I tillegg tar det tid og ressurser å finne ut hvordan man skal arkivere elektroniske dokumenter. På toppen av dette kommer opplæring av ansatte i bruken av ulike IKT-løsninger. Ettersom tiden gikk, kunne det hende at man mistet målet med investeringene sine; nemlig at den skulle føre til en arbeidsmessig og økonomisk gevinst.

En annen grunn kan være at bedrifter investerer i IKT-varer og -tjenester for å overleve i en konkurransesituasjon, hvor de fleste andre også brukte IKT i sitt virke. I den grad dette var mer et slags signal utad om at man var framtidsrettet, hadde investeringen ikke de store positive konsekvensene for arbeidet innad. Mangel på kunnskap gjorde at man ikke så hvilket potensial som lå i disse investeringene.

Et annet problem er at mange av bedriftene ikke har en helhetlig plan for investeringene sine innen IKT.

### **2.1.2 Mangelfull dokumentasjon**

Et annet moment ved IKT-administrasjon er mangelfull dokumentasjon. Det kan være at dokumenter flyter rundt og at man ikke har oversikt over dem og hva de inneholder. Men det som kanskje er verre, er at de som har ansvar ikke tar oppdatering av dokumentasjonen alvorlig ved forandringer i IKT-infrastrukturen. Dette er et tilbakevendende problem som er spesielt knyttet til oppdateringer/nyinvesteringer innen programvare. Et eksempel i denne sammenheng er endringer i brukergrensesnitt som ikke følges opp av endringer i brukerhåndbøker/ manualer. Selv en liten endring, som for noen kan virke ubetydelig, kan skape problemer for andre. Et lignende eksempel er ved tilfeller hvor man introduserer helt ny programvare uten å følge det opp på dokumentasjonssiden.

Slike tilfeller kan være med på å øke avhengigheten til enkelt superbrukere i organisasjonen, og en overbelastning av brukerstøtteavdelingen, og kan derigjennom skape problemer i forbindelse med omorganisering av bedriften.

### **2.1.3 Dårlig kommunikasjon med sluttbrukere**

I mange tilfeller produseres nye systemløsninger med lite eller ingen dialog med sluttbrukerne. Firmaer som holder på med utvikling av slike løsninger ser på bedriften som en kunde, mens det er bedriftens ansatte som er de egentlige kundene. Det er disse som skal bruke systemet i det daglige arbeidet.

Kravspesifikasjonen som designerne forholder seg til er ofte satt sammen av folk som ikke har daglig befatning med systemet som skal designes. Ofte er det bare i siste fase, da man skal teste det nye systemet, hvor dialogen med sluttbrukerne begynner. Det betyr at systemløsningen i mange tilfeller kan være langt fra hva sluttbrukerne ønsker og trenger, og dette igjen skaper behov for økte ressurser i forbindelse med opplæring av personellet.

### **2.1.4 Mye ”workarounds” og brannslukking, lite proaktiv forretningsutvikling.**

Mange bedrifter bruker mye tid på ”workarounds” og brannslukking innen IKT, noe som går utover ressurser som kunne vært brukt til å etablere en proaktiv forretningsutvikling.

Som nevnt over, er det ikke alltid en klar målsetning med investeringer i IKT. Og ofte er det personer som ikke har den nødvendige kompetansen på feltet som foretar avgjørelsene på området. Det er ofte ledelsen som tar de viktige avgjørelsene vedrørende investeringer innen IKT. Hvis da avstanden mellom ledelsen og brukerne av systemene er stor, kan det bli konflikt mellom hva ledelsen ser på som formålstjenlig, og hva brukerne faktisk trenger. Som nevnt i forrige avsnitt har de som utvikler systemene ofte liten innsikt i hvem brukerne er, og hvordan de bruker IT. Disse to momentene kan skape grunnlag for konflikter i brukernes arbeidssituasjon, som igjen kan føre til lite hensiktsmessige arbeidsrutiner, minelagt med ”workarounds” og brannslukning.

I tillegg eksisterer det ingen klar målsetning med hva man ønsker å oppnå forretningsmessig med introduksjon av ny teknologi. Investeringene kan dermed bli en sovepute, istedenfor å være et hjelpemiddel for å utvikle selskapets konkurransedyktighet.

### **2.1.5 Ressursene er lite samordnet, blir dårlig utnyttet.**

Et annet moment er hvordan IKT-ressursene, og bruken av dem, kartlegges. For å øke utnyttelsesgraden, samt minske kostnadene ved IKT-investeringer, er det viktig å hele tiden kartlegge systemene. Med spesielt fokus på:

- Unngå overlappende funksjonalitet. Dette kan sørge for at man ikke nødvendigvis investerer i det som er ”hot”, eller ”kjekt å ha”, men det man faktisk trenger. Innkjøp av duplikatlisenser er et kjent problem i så måte.
- Bruksmønstre. Ved å identifisere og isolere rutiner, kan man implementere tidsbesparende funksjonalitet, samt fjerne unødvendig funksjonalitet ved senere oppgraderinger av programvare.
- Identifisering av flaskehals/kapasitetsproblemer. Dette kan gjelde både maskin- og programvare. På maskinvaresiden kan et eksempel være at man ved hjelp av kartlegging av ressursene ser at man kan klare seg med oppgradering av enkelte komponenter (RAM, prosessorer, etc.) fremfor å kjøpe inn nye datamaskiner.
- Identifisering av andre problemer. Kartlegging av hendelser kan bidra til å identifisere hvorfor og med hvilken frekvens de ulike hendelsene opptrer. Dette tar man siden høyde for i forbindelse med drift eller oppgraderinger.

Fra ledelsens synspunkt er det god forretningsdrift å investere i ny teknologi, men det er ikke alltid en klar retning forbundet med investeringene. Dermed er det fare for at ressursene skaffet til veie gjennom disse investeringene er lite samordnet, og dette igjen kan føre til dårlig utnyttelse. En kartlegging vil i så måte legge grunnlaget for en bedre organisering av eksisterende ressurser, samt identifisere nødvendige nyinvesteringer.

### **2.1.6 Dårlig ansvarsfordeling**

Dårlig definert eller skjev ansvarsfordeling, kan bidra til at en liten gruppe blir stående igjen med en stor del av oppgavemengden. Årsakene til disse svakhetene kan være flere:

- Man ønsker ikke å gi fra seg ansvar, beskytter sin egen spisskompetanse.
- Noen brukere kan ha rykte på seg for å være såkalte ”guruer”. Dette kan høyne terskelen for andre å høyne sine egne kvalifikasjoner, ettersom det er lettere å gi oppgaven til noen som helt sikkert kan det.
- Arbeidstakere vet ikke hva de har ansvar for.

Alle disse momentene kan bidra til at arbeidet blir utført med mer tid og ressurser enn nødvendig fordi det blir samlet i ”klumper” her og der, heller enn å bli spredd utover.

### **2.1.7 Mangelfull kontroll på operasjonell risiko**

I en artikkel av Hvidkjær fra 2004 om ”Risk Management i praksis”, defineres operasjonell risiko slik (kopi av risk manager Bent Juls (Nordeas Electronic Banking) definisjon):

*”the risk of direct or indirect loss, or the damaged reputation due to deficiencies or errors in the Group’s internal operations which may be attributable to employees, the organisation, control routines, processes or technology, or due to external events and relations.”[3]*

Om man har oversikt over risiko av operasjonell karakter, kan man forberede hva man skal



takle i hvilke situasjoner, og hvordan. Ved en god risikohåndtering kan bedrifter vite hvilken sannsynlighet det er for at en operasjonell risiko kan inntreffe, hva konsekvensen er, og hvordan man kan takle evt. redusere risiki. [4]

Mange organisasjoner har i dag ikke kontroll på slike risiki. Hver gang de møter slike utfordringer, har det en tendens til å bli kaos. «Brannslukkingen» som må foretas i slike tilfeller koster både i form av ressurser, tid, og økonomi.

## **2.2 Bedre administrasjon gjennom Prosessforbedring via Standardisering**

For å søke å komme problemene identifisert i forrige avsnitt til livs, er en innfallsvinkel å se på bedre utnyttelse gjennom bedre prosesser. Hva kan bedriftene gjøre for å endre situasjonen, slik at utgifter til investeringer kan snus til inntekter? For å øke kvaliteten på IT-tjenester og styre bruken av IT for å skape forretningsverdi, og en konkurransemessig fordel i forhold til konkurrentene, er det formålstjenlig å ha en administrasjon av IT-ressursene, særlig administrasjon innen "IT Service Management" (Service Management forklares inngående i kapittel 3).

### **2.2.1 Rammeverkt for administrasjon av IT-ressurser**

Det finnes flere ulike rammeverk som passer til dette formålet som COBIT, ITIL, CMM og ISO 17799.

- **COBIT (Control Objectives for Information and related Technology):** Cobit brukes for å implementere IT-administrasjon. Man bruker rammeverket sammen med andre ressurser, og skreddersyr dette slik at det passer inn i organisasjonen sitt eget miljø. Forskjellen mellom Cobit og andre rammeverk, er at Cobit har et sett med såkalte kontrollmål, som tilrettelegger for ulike typer loggføringer og målinger. Det som er viktig i dette rammeverket, er hva som skal gjøres, ikke nødvendigvis hvordan. [5]
- **ISO 17799:** Dette er en internasjonal standard for sikkerhet i IT-systemer, som gir retningslinjer og krav til sikkerhetsledelse. Organisasjoner som har ansvar for sikkerhet i ulike henseende må følge denne standarden for å bli sertifisert i henhold til den. Standardens mål er at man skal stimulere til økt fokus på informasjonssikkerhet. [6]
- **The IT Service Capability Maturity Model (The IT Service CMM):** er en modell som hjelper tilbydere av IT tjenester til å få en effektiv kontroll og administrasjon på ulike IT-komponenter, som maskinvare, programvare (også vedlikehold) og drift. [7]
- **ITIL – Information Technology Infrastructure Library:** Fokuserer på såkalte "best practices" ved IT-administrasjon. ITIL rammeverket foreslår hvordan hver prosess bør utføres på beste mulige måte. [8] Det hjelper til å:
  - Unngå at man må finne opp hjulet på nytt, for hver gang man utfører arbeid.
  - Fokus på fordeler for kunder, ved å levere akkurat det kunden ønsker. Ved å ha klare begreper som det er enighet om hos bedrifter og deres kunder, sikrer man god kommunikasjon mellom dem. Da de to partene forstår hverandre

godt, kan bedriftene produsere de riktige produktene/tjenestene. Med andre ord, får bedriften muligheter for kostnadsreduksjoner i tillegg til fornøyde kunder.

- Profesjonalitet i prosessforbedring med klare regler, prosedyrer og arbeidsflyt i hver prosess.
- Det finnes også i ITIL en klart definert grense for hver prosess og hvordan prosessen skal måles. Slike ting gir både bedrifter og kunder et godt beslutningsgrunnlag.

## **2.2.2 Generelt om ITIL- IT Infrastructure library [9]**

En undersøkelse utført av Mercury og Economist viste at mer enn 60 % av IT-sjefene i Norge har planer om å investere i såkalt IT-Governance (IT-administrasjon) i 2005 [2]. Mange av disse ønsker å bruke ITIL-rammeverket til dette formålet.

Denne masteroppgaven behandler ITIL inngående, siden det er dette verktøyet SKD har valgt å benytte i sine bestrebelser på å hente ut gevinster. IT personell i SKD med ansvarsmyndighet har sett på de fordeler de kan oppnå med ITIL, og har derfor besluttet å implementere rammeverket.

### **2.2.2.1 Hva er ITIL?**

Alle bedrifter ønsker å tilby sine kunder bedre tjenester, med mindre ressursbruk. Mange organisasjoner sliter, som sagt, med en unødvendig grad av brannslukning i forbindelse med IT-systemer. De har systemer som er fragmentert, både når det gjelder programvare så vel som databaseløsninger. ITIL er et "best practice" rammeverk som man benytter for å utføre og levere best mulig IT-tjenester til en rimelig kostnad ved å foreslå et sett av prosesser og rutiner for å hjelpe bedrifter til å oppnå sine formål.

ITIL ble utviklet i 1980 av Office og Government Commerce (OGC), og beskriver arbeidsprosessen steg for steg fra begynnelse med planlegging til de siste stegene i prosessen som inkluderer applikasjoner og sikkerhet. Rammeverket tar hensyn til både forretningsdelen så vel som systemets tekniske attributter.

Den definerer prosessmål, fordeler, kostnader, hvor relevant redskap finnes og gir oss vanlige rutiner, ansvarsfordeling, metrikker (måleparametere), tidsfrister, innhold i dokumenter, mulige problem som kan oppstå og hvordan man kan håndtere disse i hele prosessen. Et viktig mål ved ITIL er å få alle til å snakke samme "språk" for å sikre at prosessen blir vellykket. I administrasjonsprosessen, beskrives det hva man bør gjøre trinn for trinn ved ulike input-parametre, hvilke aktiviteter som bør utføres, og hva man kan forvente å få i resultater.

ITIL beskriver nøyaktig prosessen man bør gå igjennom for å oppnå målet, og prosessene har prosedyrer som beskriver hvem som gjør hva, til og med arbeidsinstruksjoner som steg for steg beskriver hvordan ulike arbeidsoppgaver skal utføres.

ITIL er utformet for større organisasjoner, men det er mulig å skalere det ned, slik at det også kan brukes for små- og mellomstore organisasjoner. Imidlertid, fordi det trenger mange støtteredskaper og opplæring for ansatte, så egner det seg best til store bedrifter.

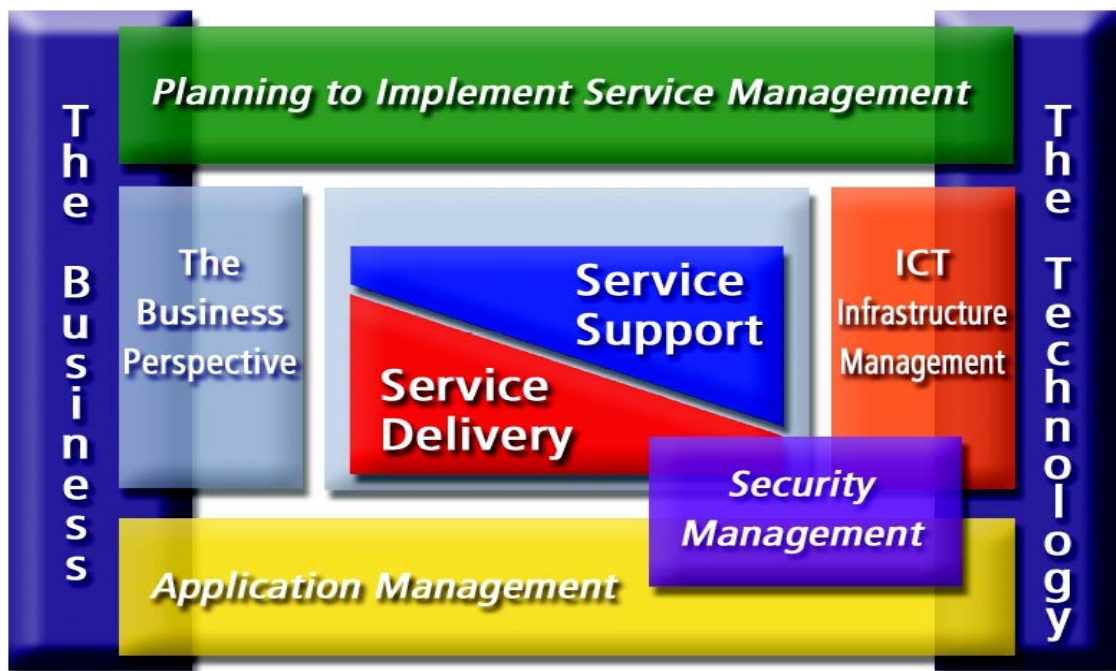
Ved å benytte ITIL kan bedrifter snu fra et "kaotisk" nivå til et proaktivt nivå, hvor man kan

være i forkant av utvikling og trender som kan virke negativt inn på IT-infrastrukturen.

Selv om det er kjent og brukt av hele verden som "the world-wide de facto standard", er ikke ITIL en standard. Det er bare en samling av erfaringer gjennom lang praksis. Hver bedrift som bruker ITIL må vurdere de ulike komponentene som rammeverket tilbyr og velge å forme dem slik at de tilpasses til egen organisasjon.

### 2.2.2.2 Innholdet i ITIL

ITIL tilbyr et rammeverk for alle prosesser, fra planlegging til applikasjon og sikkerhet. Den deles typisk inn i 7 moduler (se Figur 3)



Figur 3: The ITIL Framework (kilde [9])

Figur 3 viser i sin helhet alle modulene i ITIL-strukturen. Her kan vi se forholdet mellom de forskjellige ITIL-modulene, og deres forhold til omgivelsene. Vi kan se her at modulen "The Business Perspective" knytter seg til forretningsdelen, "ICTIM" er koblet til den teknologiske delen, mens "Service Support" og "Service Delivery" er rammeverkets "hjerne". Figuren er laget slik at man kan se hvilke komponenter som er "avhengige" av hverandre ved å se på hvordan de overlapper hverandre i den grafiske representasjonen av rammeverket. "Planning to Implement Service Management" og "Application Management" er nødvendig for alle de andre komponentene i rammeverket, mens "Security Management" er bare nødvendig for "ICTIM", "Service Delivery" og "Application Management".

### 2.2.2.3 "Planning to Implement Service Management"

Denne modulen planlegger hvordan en tjenesteadministrasjon innenfor en organisasjon skal iverksettes eller forbedres. Den vurderer hvor og når oppgaven skal startes, hvordan man skal forandre organisasjon og kultur, og planlegger program og prosjekt eller definerer

prosess.

Det første man gjør, er å etablere noe som kan kalles en visjon for hvordan "IT Service Management" skal fungere i samspillet mellom forretning og IT. Visjonen beskriver mål og formål med "Service Management's" "Continuous Service Improvement Programme (CSIP)".

For å benytte seg av denne modulen, må ledelsen tenke gjennom, og svare på et sett av spørsmål. De kan kalles hovedspørsmålene for prosessgjennomføringen.

"Hva er vårt formål?", er det første spørsmålet ledelsen bør stille seg. Når ledelsen har kommet til et tilfredsstillende svar på dette spørsmålet, kommer neste.

"Hvor er vi nå?". Svaret på dette spørsmålet bestemmer den nåværende tilstanden til IT-delen av bedriftsorganisasjonen når det gjelder: visjon og strategi, administrasjon, prosesser, mennesker, teknologi, og (bedrifts)kultur.

Det er viktig at forretnings- og IT-delen av organisasjonen på bakgrunn av en tilstandsrapport etablert på bakgrunn av spørsmålet ovenfor kommer til enighet om hvor man ønsker å ta organisasjonen. "Hvor vil vi gå?".

Det neste steget, er å utarbeide en plan for et CSIP-prosjekt for hvordan man vil komme seg fra nåværende tilstand, til en fremtidig ønsketilstand. "Hvordan kommer vi dit vi ønsker å komme?". I denne forbindelse er det formålstjenlig å tenke på hvordan man skal få til konstruktive endringer. I denne prosessen, velger man et sted å starte, og hvilke elementer å ta hensyn til innen CSIP.

Gjennom det man har bestemt seg for med disse spørsmålene som "loddessnorer", har man dermed dannet et godt grunnlag for forløpet til CSIP-prosjektet. Når man så er i gang med dette, er det viktig å holde orden på milepæler, slik at man ikke mister framdriften i prosjektet. For å sikre at aktivitetene har fordeler for forretningen, bør det hele tiden fokuseres på endringer som innebærer fordeler for organisasjonen.

#### **2.2.2.4 "The Business" - Forretningsperspektivet.**

Forretningsperspektivet assisterer forretningsdelen av organisasjonen til å holde sine mål ved å gi veiledning og råd. Formålet er maksimering av utnyttelse og tilpasning av tjenester.

Forretningsperspektivets tilnærming i forbindelse med levering av IT-tjenester, er fokus på prinsipper og retningslinjer for forretningsorganisasjonen og dennes virksomhet. Spesielt viktig her er hvordan man kan integrere bruken av IT innen alle områder av Service Management. Slik sørger man for effektive koblinger, gode grensesnitt, og gode vilkår for forretningsdelen. Dette gjør organisasjonen i stand til å dra nytte av IT i forretningsøyemed.

Forretningsperspektivet mål kan oppsummeres slik:

- Sørge for at IT-personell får viten om hvordan de kan bidra i forretningsøyemed.
- Maksimere utnyttelsen av IT.
- Legge forholdene til rette for en god forretningskultur, samt gi gode vilkår for smidighet ved endringer i faktorer som har innvirkning på forretningen.

- Integrering av IT som en del av forretningsdelen.

### **2.2.2.5 "ICT Infrastructure Management (ICTIM)"**

Denne modulen har ansvar for infrastrukturen når det gjelder IKT, og dekker ledelse og administrasjon, design og planlegging, teknisk støtte, utvikling og drift. Den forvalter også produktene, forhold med personalet og andre parter. Et begrep er nært relatert til ICTIM, nemlig "The four P's". Det er "People", "Partners", "Products" og "Processes".

ICTIM knytter seg til tjenester i hele livssyklusen. Helt fra kravspesifikasjon, design, utvikling, test, til drift. Den sørger for at alle operasjoner fungerer som de skal.

Ledelse og administrasjon må passe på at infrastrukturen er vedlikeholdt slik at tjenester oppnår de leveringskrav som bedriften avtalefester med kunder, både nåværende og fremtidige.

Ved hjelp av teknisk support holder man tjenester tilgjengelige og oppdaterte, i tillegg til at det her identifiseres muligheter for utvikling av nye teknologiske løsninger.

### **2.2.2.6 "Security Management" - sikkerhetsadministrasjon**

Sikkerhetsadministrasjon er prosessen som administrerer definert sikkerhetsnivå for informasjon, spesielt innen IT-service og -infrastruktur.

Sikkerhetsadministrasjon har ansvaret for iverksetting av forandringer som kan virke inn på forretningen, tjenestebehov for IT, hele eller deler av arkitekturen, ulike trusler etc.

Sikkerhetsadministrasjon sørger for at:

- Sikkerhetskontroll er implementert og vedlikeholdt etter bedriftens behov. Det gjelder både i IT-arkitektur, IT-tjenester og trusler fra omgivelsene.
- Man har kontroll på ulike sikkerhetshendelser.
- Resultatkontroll viser sammenhengen mellom sikkerhetsadministrasjon og iverksatte tiltak.
- Rapporter sørger for å vise status for informasjonssikkerheten.

### **2.2.2.7 "Application Management" – administrasjon av applikasjoner**

Administrasjon av applikasjoner er viktig med tanke på å utvikle eller ta i bruk applikasjoner (programvare) som tilfredsstillende ulike behov i organisasjonen. I tillegg er det nødvendig å følge applikasjoner gjennom deres livsløp, slik at man kan registrere behov for endringer. På bakgrunn av dette er det naturlig å gå hånd i hånd med prosessene "ICT Infrastructure Management" og "Service Management". I en utvidelses-/utviklingsprosess bør arbeidet utføres sammen med "Service Management" helt fra designfasen, slik at applikasjoner kan utarbeides i henhold til krav og spesifikasjoner. Disse kravene kan innebære at applikasjonene utvikles med tanke på ulike forretningskrav og tjenesteavtaler, i tillegg til de mer teknisk rettede metrikkene som går på: operasjonell stabilitet, tilgjengelighet, pålitelighet, vedlikeholdbarhet, og ulike prestasjonsmetriker.

En god kontroll på applikasjonene bidrar til å gi en god oversikt over hvordan de fungerer i de ulike stadier i deres livssyklus. Dermed er det lett å identifisere endringer som er

nødvendige, enten for bedriften, både i form av nye funksjoner, såvel som tilpasning til nye miljøer. Og ikke minst for kunder, f.eks. i form av utvidede tjensteavtaler/SLA's.

### **2.2.2.8 "Service Delivery"**

"Service delivery" er en modul som har prosesser som beskriver hva som trengs for å kunne levere ulike IT- tjenester til kunder.

Prosesser i denne modulen er henholdsvis "Capacity Management" (kapasitetsadministrasjon), "Financial Management for IT Service (FITS)" (finansiell administrasjon), "Availability Management (AM)" (tilgjengelighetsadministrasjon), "Service Level Management (SLM)" (administrasjon av tjenstenivå), og "IT Service Continuity Management (ITSCM)" (kontinuitetsadministrasjon).

Alle tjenester leveranse prosessene samarbeider for å sikre, eventuelt forbedre, bedriftstjenestetilbudet i fremtiden. Med god administrasjon i "Service Delivery", kan bedriften styre både sine ressurser, sin kapasitet og sine produkter slik at bedriftens mål blir oppnådd.

Hovedansvaret til SLM prosessen er å bygge "Service Level Agreement (SLA)" (avtaler om tjenstenivå) med kunder. I SLA skal tjenester defineres, man skal bli enige om hvordan de skal leveres og støttes, til og med hvilke nivåer i bedriften som skal støtte det.

Finansiell administrasjon gir grunnlaget for en oversikt over de kostnader bedriften investerer i IKT. Dette gir bakgrunn for god budsjettering ved framtidige IKT-investeringer.

Kapasitetsadministrasjonen sørger for at adekvat kapasitet hele tiden er tilgjengelig slik at forretningskravene møtes. Man lager kapasitetsplaner som er nært knyttet opp mot forretningsstrategien. Disse planene blir gjennomgått og reprodusert jevnlig. Planene inkluderer fokus på tre hovedområder; forretning ("Business Capacity Management"), tjenester ("Service Capacity Management"), og ressurser ("Resource Capacity Management"). Disse tre områdene omfatter aktiviteter som er nødvendige for å sikre at IT kapasiteten og kapasitetsplanen(e) er på samme banehalvdel som forretningsbehovet.

I tillegg produseres det planer for andre utfall som kan virke inn på evnen til tjensteleveransen, som ulike forstyrrelser i driften, eller planer som kan hjelpe til med å identifisere mulige endringer i forretningsmiljøet, som nye forretningsmuligheter.

Tilgjengelighetsadministrasjon, foreslår et sett av målinger, rapporter, overvåkning, og testing av tjenester. Ut fra resultater fra disse tiltakene, kan bedriftene få identifisert mulige ønskede tjenester. Både nåværende, og framtidige.

### **2.2.2.9 "Service Support"**

Denne oppgaven fokuserer på denne prosessen. Det er i første runde denne som er knyttet til design av CMDB. Denne modulen er derfor omtalt nærmere i eget kapittel 3.

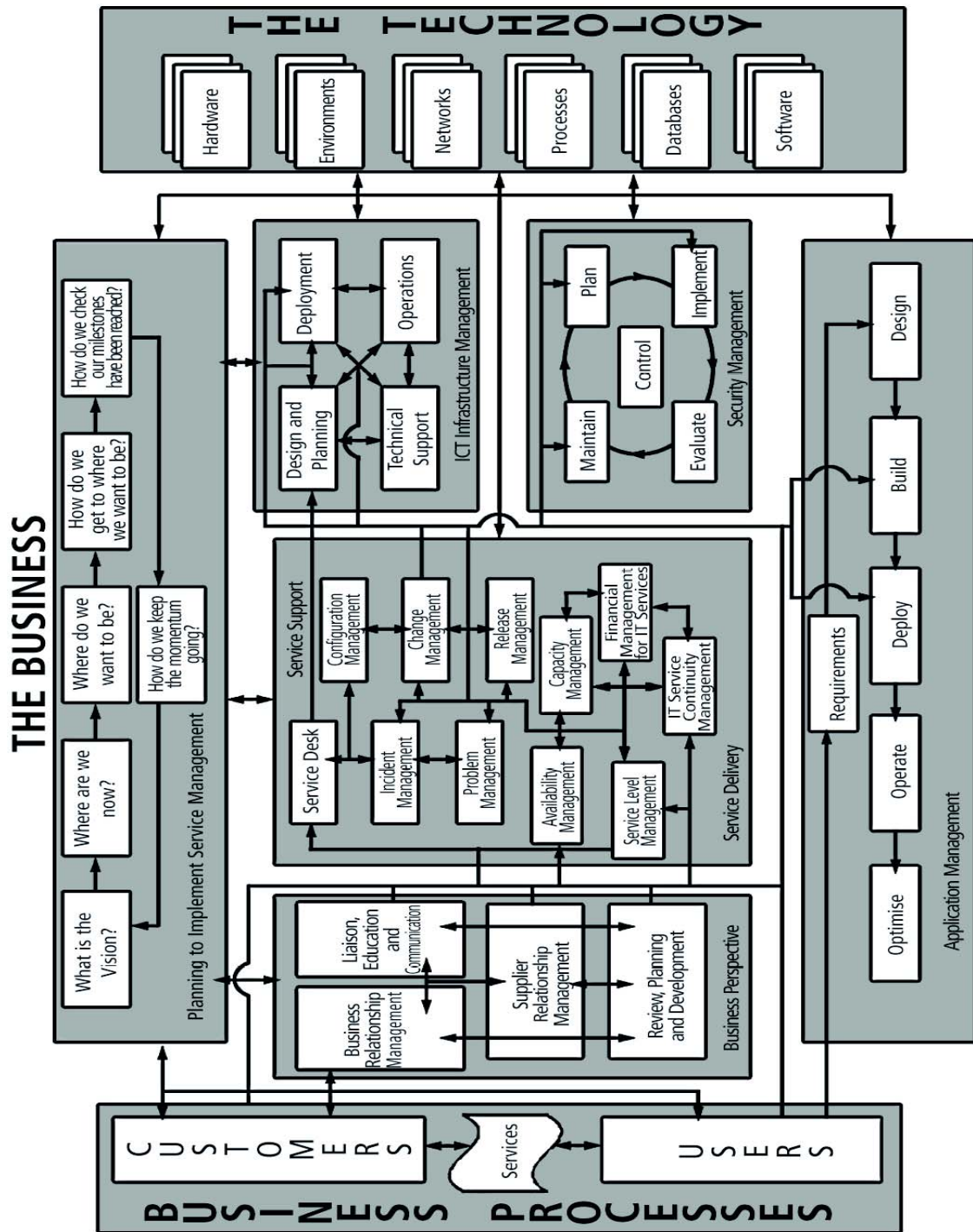
Prosesen fokuserer på at kunder skal ha tilgang til IT-tjenester og at de kan få støtte når de opplever problemer med datasystemet. Det tar for seg daglig support og driftsrutine.

Service Support består av prosessene "Service Desk", "Incident Management", "Problem Management", "Configuration Management", "Change Management" og "Release

Management (RM)” (versjonsadministrasjon).

#### ***2.2.2.10 Prosesser i samspill***

Alle prosessene i ITIL er knyttet til hverandre og samspiller. Men organisasjonen behøver ikke å implementere alle modulene for å benytte ITIL. Det kan velges ut etter behov, hvilke moduler som skal benyttes, og hvilke prosesser i hver modul som er nyttig for organisasjonen. Bildet nedenfor illustrerer hvordan prosessene i hver modul samspiller og relasjoner mellom de forskjellige modulene. (Figur 4)



Figur 4: ITIL prosessene og deres samspill (kilde[9])

### 2.2.2.11 Noen fordeler som kan oppnås ved bruk av ITIL

- Forbedret kvalitet på tjenester og mer pålitelig forretningsstøtte.
- Bedre kommunikasjon mellom bedrift og kunde, og internt i bedriften.



- Bedre informasjon om nåværende tjenester til kunde og ledelse.
- Bedre oversikt over nåværende IT- kapasitet.
- Større fleksibilitet for forretningen gjennom øke forståelse av IT support.
- Økt tilfredshet hos Kunder, da man leverer det kunden ønsker.
- Økt fleksibilitet og tilpasningsevner innenfor tjenester.
- Forbedre syklustid for forandring og større suksessrate.
- Bedre informasjonsgrunnlag for strategiske beslutninger.
- Mindre brannslukking.
- Lettere hverdag for ansatte.

## **2.3 Bedre administrasjon gjennom IT verktøy**

System- og nettverksadministrasjon er et stort område som bedrifter må ta spesielt hensyn til for at datasystem og tjenester skal kunne fungere optimalt. Systemadministrasjon består av et sett med verktøy, prosesser og prosedyrer.

Med gode systemadministrasjonsverktøy kan bedriftene redusere driftsutgifter, samtidig som tjenestenivået i systemet kan økes. [10]

Datasystemer er knyttet sammen i nettverk, det kan være lokale nett for datautveksling innad i bedriften eller internett for utveksling av data med andre. I tillegg blir en del tjenester tilbudt over internett i form av bl.a. ulike web-baserte applikasjoner. Ettersom båndbredden har økt, har bruk av internett, antall brukere, datatyper, enheter og tjenester økt dramatisk. I kjølvannet av dette har vi sett en klar økning i behovet for nettverksadministrasjon. Det er et stort område i fortsatt sterk vekst.

### **2.3.1 Overvåkingsløsning i generelt [11, 12]**

I et driftsmiljø trenger man tjenesteovervåking for kontinuerlig kontroll. Et eksempel er behov for å kontrollere oppetid og minimere risiko for nedetid. Det er viktig å vurdere i hvilken grad man kan benytte denne type overvåkingsløsninger til å gi en automatisert oppdatering av en CMDB. En CMDB inneholder primært informasjon om eksisterende infrastruktur og mindre om status for hvert av infrastruktur elementene, men likevel vil en overvåkingsløsning (eller samlingen av overvåkingsløsninger) nødvendigvis kjenne til de enkelte infrastrukturelementer som skal overvåkes.

*”Tjenesteovervåking kan benyttes for å sikre at et driftsmiljø fungerer tilfredsstillende. Enheten som overvåkes loggfører sin tilstand, resultatet fra en slik overvåking brukes blant annet til å se historiske trender, kapasitet, øyeblikkstilstander til systemer og statistikk som oppetid, reparasjonstid og lignende. En god overvåkingsløsning kan detektere tilstander som over tid med sikkerhet vil føre til driftsavbrudd. En tekniker kan reagere raskere dersom en feil skulle oppstå, og på denne måten redusere konsekvensene av feilen.” [11]*

Systemet kan overvåkes på tre måter: historiske data, sanntidsdata og fjernstyring.

- **Historiske data:** det er en samling data som beskriver systembruk over et lengre tidsrom. Dataene analyseres ved behov i ettertid for å finne feil, klassifisere feil til en enhet, eller lokalisere feilens kilde. Resultatene fra slike analyser kan brukes til å finne ut om hvilke problemområder som bør prioriteres videre, og hvilke som kan nedprioriteres.
- **Sanntidsdata:** sendes ut fra en sanntidsapplikasjon på en tidsriktig og forutsigbar måte. Applikasjonen må kunne sende ut feilmelding uten å måtte spørre om systemtilstand. Applikasjonen består av to deler. Den første overvåker et system for å finne ut om tjenester er feil og den andre varsler aktuelle personer/enheter når feil oppdages.
- **Fjernstyring:** man kan aktivt påvirke tilstanden til overvåket utstyr ved å endre driftsstatus eller konfigurere utstyret. Man kan via fjernstyring stenge en server hvis man får en sanntidsmelding om at noen prøver å bryte seg inn i system.

### 2.3.2 Systemovervåking

Systemovervåking brukes for å ha en oversikt over ulike systemer og systemressurser i nettet. Det kan hjelpe til med å oppdage problemer så vel lokalt, som systemer som er utenfor den umiddelbare lokale sfære. Verktøy som brukes ved systemovervåking kan brukes til å overvåke alt fra komponenter som CPU, minne, harddisker, nettverksadaptere, til ulike tjenester og programvare, til brukere av systemene. Man kan sette det opp til å logge begivenheter/hendelser og å lage rapporter på bakgrunn av logget bruk til databaser som kan plasseres hvor som helst i nettverket. Disse dataene kan brukes til å lage en grovskisse for tilstanden til ulike systemkomponenter. Man kan også bruke dataene for å beregne statistikk slik at man kan få et mer forståelig og målrettet bilde av tingenes tilstand.

### 2.3.3 Nettverksovervåking

Nettverksovervåking sørger for oversikt over nettverksressursene, som rutere, broer, hub'er, switcher etc., og infrastrukturen som disse understøtter. Overvåkingen er ofte basert på en såkalt klient/server modell, der konsollen fungerer som en klient hvorfra administratoren sørger for drift av systemet, mens informasjonen lagres på nettservere.

#### 2.3.3.1 Driftsfilosofi [13]

Man kan grovt sett dele driftsfilosofien inn i to ulike metoder: drift etter brannslukkingsprinsippet, og forebyggende drift. Ingen av disse metodene er rennasede, og i praksis forekommer en blanding. Hvilket prinsipp som velges som retningsgivende er ofte avhengig av bedriftens behov.

- **Drift etter brannslukkingsprinsippet:** Det betyr at det oppstår en feil i nettverket. Det kan variere fra en liten feil til en eller flere alvorlig feil. Systemansvarlig reagerer når feilen blir oppdaget, ikke før. Feilen må rettes opp så fort som mulig for å redusere tap. Når et nettverk er "nede", må man finne ut hva som er feil, og hvor feilen befinner seg. Det kan være en vanskelig jobb. Det kan hjelpe å kategorisere feilen først. Om det er en brukerfeil, programvarefeil, maskinvarefeil eller feil på kabling. Så analyser man informasjonen og stiller diagnose for å isolere feilen. Når den er isolert må den repareres og dokumenteres.
- **Forebyggende drift:** Det vil si å gjøre noe før et problem oppstår, og for å unngå at

det oppstår. Man bruker to metoder i forbindelse med forebyggende drift: Kapasitetsplanlegging og System-tuning.

- *Kapasitetsplanlegging*, er å tenke fremover for å forutse problemer som kan oppstå og forebygge disse. Det er to typer verktøy som benyttes ved denne metoden, nemlig simulatormodeller og lastgeneratorer.
- *System-tuning*, er å samle inn data om drift av nettverket og analysere disse for å lage en oversikt over potensielle problemområder. Deretter finner man tiltak for å unngå at disse materialiserer seg som faktiske problemområder.

### 2.3.4 Populære overvåkningsverktøy

ITIL virker best når man benytter gode verktøy, m.a.o. mest mulig automatisering, f.eks. automatisk overvåking. Ved automatisering kan man satse på proaktiv systemadministrasjon.

#### 2.3.4.1 Kategorisering av verktøy [14]

Verktøyene innenfor system- og nettverksovervåking kan deles inn i 6 kategorier slik:

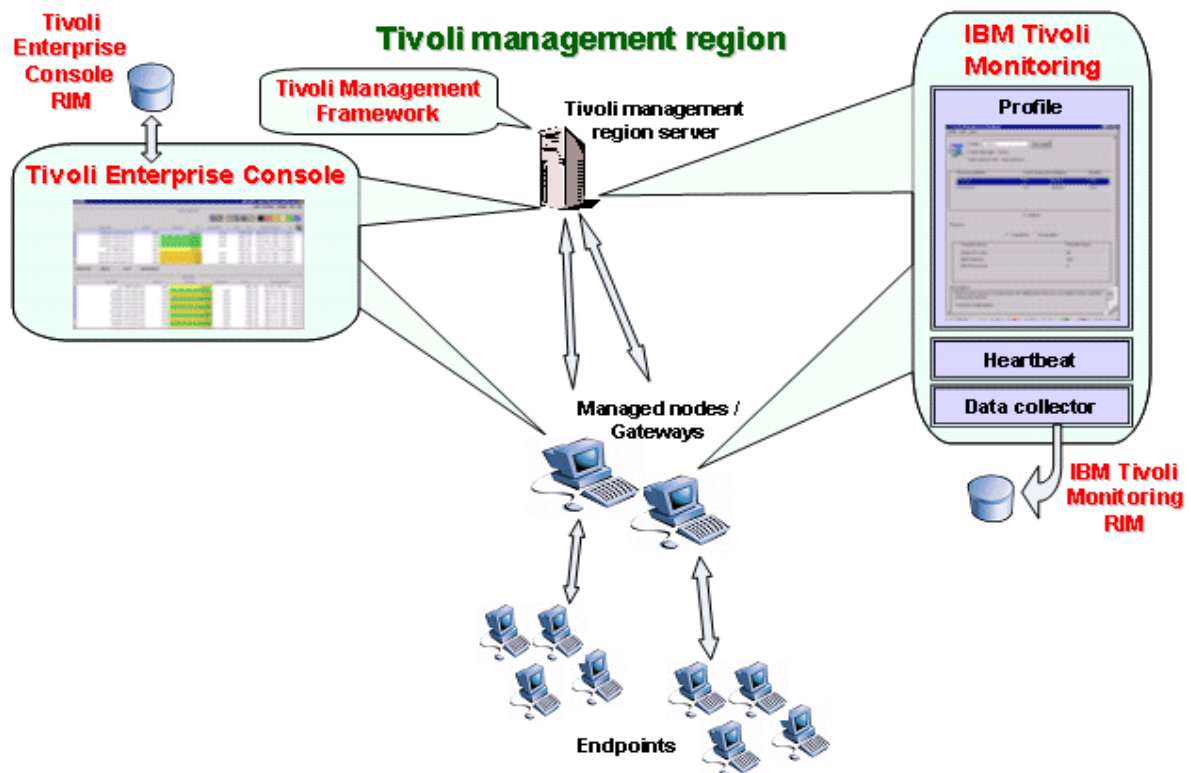
- **Generelle produktivitetsverktøy:** er tradisjonelle SNMP-baserte verktøy. Noen av produktene på markedet inkluderer CA Unicenter, SunNet Manager, HP Openview, Cabletron Spectrum, IBM Tivoli, IBM Netview og Bull OpenMaster.
- **Verktøy for å styre egenproduserte komponenter:** Bedrifter kan selv lage programmer til administrasjon av egne nettkomponenter. Disse kan igjen styres via programvare som er spesielt tilpasset til å ta seg av egenkomponerte programmer. Et typisk produkt i denne kategorien er Cisco Works. Det finnes i tillegg andre produkter som kan tilpasses mer generelle verktøy som CA Unicenter eller HP Openview.
- **Lan- administrasjon:** egnet godt til enkeltstående LAN, men har ikke kapasitet eller funksjoner for å takle nettverk i større målestokk. Blant produsenter av disse applikasjonene, kan nevnes bl.a. Microsoft, Novell, Intel og Symantec.
- **Funksjonsorienterte verktøy:** Produsenter i denne kategorien retter seg mot spesielle funksjoner i nett som e-handel, transaksjoner over internett, administrasjon av Web-applikasjoner etc. Som eksempler kan nevnes; Candle som benyttes av UPC for å styre deres CATV-nett, Remedy som håndterer feilmeldingsadministrasjon med web-grensesnitt og egen database, Compuware. Lotus Notes er et produkt med automatisk generering av feilrapporter.
- **Ekspert- system:** Man kan lage slike system med egen database og sannhetstabeller ved hjelp av for eksempel Excel.
- **Offentlige nett:** Telekommunikasjonsleverandørene Alcatel, Siemens og Ericsson er leverandører av noen av produktene i denne kategorien. Disse produktene er under kontinuerlig utvikling.

I det følgende omtales populære produkter i nettverksovervåking, med vekt på "IBM Tivoli monitoring", siden dette er det produktet SKD vil bruke i sin utvikling.

### 2.3.4.2 IBM Tivoli monitoring[15]

IBM Tivoli Monitoring er programvare som automatiserer overvåkningen av nøkkelsystemressurser, identifiserer systemsvikt, sjekker om system er nede, finner flaskehalsar og andre mulige problemer, og som automatisk løser kritiske situasjoner.

I Figur 5 ser vi hva et såkalt Tivoli administrasjonsområde kan omfatte, og interaksjon innenfor dette området. Vi ser at IBM Tivoli Monitoring er en viktig bestanddel. Den kan overvåke både hovedserver ("region server") og gateway-node.



Figur 5: IBM Tivoli Monitoring (kilde [15])

IBM Tivoli Monitoring kan utføre følgende oppgaver:

- Overvåke operativsystemstatus.
- Sende notifikasjon om system- eller prestasjonsproblemer og foreta automatiserte, korrigerende tiltak.
- Samle og lagre systemdata.

Et felles informasjonsmodell-grensesnitt (CIM-grensesnitt, etter Common Information Model) gir alle data til IBM Tivolis overvåkningsmekanisme.

Tivoli overvåker installasjoner på Tivoli-ledelsesregionens tjener. Programvaren gir tre

viktige overvåkende elementer:

- **Profiler:** er samlede sett av informasjon som rapporteres til anvendelsesressursene i Tivoli. For eksempel inneholder profilen ressursmodeller som definerer overvåkende konfigurasjoner som overvåker spesifikke kriterier i deres miljø.
- **Hjerteslag:** Den såkalte hjerteslag-funksjonen overvåker regulære sluttpunkter for å bekrefte om de er tilgjengelig og at de kjører riktig. Denne funksjonen gir notifikasjon om maskinen på et sluttpunkt svikter. Hjerteslag-funksjonen kan utføres på gateway'en til de overvåkede sluttpunktene. Hjerteslag som overvåker funksjon sender hendelser til IBM Tivoli Enterprise Console og IBM Tivoli Business Systems Manager, (hvis IBM Tivoli Business Systems Manager komponenten er installert på gateway'en) og sender meldinger til IBM Tivoli Monitoring meldingsgruppen.
- **Datasamler:** Datasamleren er en komponent som henter data fra sluttpunktene til RIM databasen for datainnsamling. Denne RIM databasen er den samme databasen som brukes for IBM Tivoli Enterprise Console eller en separat database.

IBM Tivoli overvåkning kan benyttes til overvåkning på mange områder. Det finnes overvåkingsverktøy for bl.a. forretningsintegrasjon, databaser, Microsoft.NET, og nettverk (inkludert for infrastruktur).

#### 2.3.4.3 HP Openview [12, 14]

Hewlett Packard's Openview er et plattformssystem. Det tilbyr faste grensesnitt slik at andre nettkomponentleverandører kan knytte sine produkter til Openview. Applikasjonen kan brukes for både Unix, Linux, og Windows. Hovedprodukter er Network Node Manager (NNM).

*"Applikasjon gir en god oversikt over nettverkstopologien, og går ned i dybden, helt ned til lag 2-nivå i nettverket med mac-adresser til NIC (Network Interface Card)(...). NNM finner noder i nettverkene og tilbyr et intuitivt grafisk format. NNM finner noder i nettverkene og gir et kart som illustrerer nøyaktig hvordan nettverket ser ut - hvordan de forskjellige komponentene er koplet sammen. Topologikartet, som er hierarkisk bygget opp, viser ved hjelp av fargekoder status på nettverksutstyr slik at man raskt kan identifisere og diagnostisere feil og initiere korrigerende tiltak. Ved hjelp av NNM's SNMP-baserte polling-funksjonalitet kan man også samle inn sanntids ytelsesinformasjon fra nettverket som hjelper en til å realisere en proaktiv driftsmodell hvor man fokuserer mer på å forebygge feil."* [14]

Hovedtrekkene til Network Node Manager:

- **Kartlegging:** grafisk presentasjon av reell nettverkskonfigurasjon. Man kan styre hvilke deler av nettet som trenger å overvåkes. De andre delene blir bare markert med farger.
- **Bruk av farger:** Det brukes 6 ulike farger for å vise tilstander i nettet og 4 andre farger for å vise at noden ikke overvåkes, eller er under testing, etc.
- **Bruk av symboler:** For å skille de forskjellige delene i nettet, for eksempel sirkel symboliserer nett, firkant er en maskin, etc.
- **Alarmer:** Både standardalarmer og muligheter for egne alarminnstilinger finnes.

Siden mange kan ha fått alarmen når det oppstår feil, så for å hindre at alle jobber med problemløsning samtidig, er det bare den som har kvitteringsfelt som kan jobbe med saken.

- **Meldinger/varsler:** Man kan få varsler om bestemte problem både via pop-up bokser på skjermen, e-mail, eller personsøker. Det eksisterer muligheter for varslinger pr. SMS eller via telefonoppringning, ved hjelp av en robot som leser opp feilmeldingen og får respons via telefon.
- **Filtrering:** Det er mulig å bruke avanserte lesingsprogram for å filtrere meldinger/varsler, slik at de som har sammen mening skal representeres som én. Dette har stor betydning i store nettverk.
- **Grafisk trafikkpresentasjon:** Man kan følge med trafikken i nettet for både enkeltknoter eller en del av nettet.
- **Diagnostic tools (Overvåkning):** Med denne funksjonen kan man finne rutingtabellen, MAC-adressen, IP-adressen og hvilken nettverkstjeneste som hører til hver enkelt node.
- **Nettverktøy:** Er funksjoner som synkroniserer klokke, ping, traceroute, look-up, etc.
- **Fjernovervåkning:** Tilbyr muligheten for å overvåke et nett med web-browser.
- **Java/Web-basert brukergrensesnitt:** NNM har et web-basert grensesnitt, og stor bruk av Java slik at det gir tilgang til det meste av funksjonalitet, samt øket fleksibilitet. Med et web-basert grensesnitt kan man kontakte systemet hvor som helt i nettet via http.
- **Sikkerhet:** NNM støtter blant annet Name Address Translation (NAT), portfiltrering, og SNMPv3 som muliggjør sikker administrasjon av nettverk. Men man må ha en klar policy om man vil ha høyest mulig sikkerhet, eller fleksibel tilgang til sikkerhets- funksjonalitet når man implementerer verktøyet.

#### 2.3.4.4 CA Unicenter [14]

CA står for Computer Associates, og er et programvarehus som har som hovedprodukt et system og nettverksadministrasjonsverktøy, i tillegg til flere andre produkter som applikasjonsutvikling, programvare for produksjon og økonomi, database og systemadministrasjon. Unicenter legger vekt på distribuerte løsninger. I motsetning til andre leverandører, som selger maskinvare for å drive administrasjon, har Unicenter en uavhengig programvareløsning.

Unicenter markedsfører ca. 500 programvareprodukter. Produktene egner seg til bedrifter som har fra 50 til 100 brukere og oppover.

Det brukes Microsoft SQL-server for databaseslagring av driftsinformasjon. En egen objektorientert database, kalt Jasmin, er under utarbeidelse.

Unicenter er et rammeverk som består av mange andre objekter med ulik funksjonalitet. Unicenter legger vekt på totalløsninger. De tilbyr total systemadministrasjon av tjenestemaskiner i et datasenter med jobbkøer, beredskap osv. De har programvare for kun

én pålogging, System Alert for PC-overvåking, Agentworks for UNIX-rapportering. New Technology Generation (NTG), har tatt i bruk tredimensjonal spillteknikk for overvåking av nettverket. NTG har muligheten til å vise feil i spekteret fra fugleperspektiv til detaljnivå. F. eks. en feil som oppstår et eller annet sted i New York vil kunne følges helt opp til et system som har en databasefeil i en tjenestemaskin.

#### 2.3.4.5 Andre produkter som finnes og utviklingstrender [14]

Der er mange andre produkter på markedet som retter seg mot små- og mellomstore bedrifter. De fleste er blitt designet for å kunne fungere sammen med de store produktene (f. eks de tre overnevnte). I tillegg kan de fungere som enkeltstående produkter. Det de prøver å tilby er tilleggsfunksjonalitet samt bedre applikasjoner, eller en spesiell design for et spesielt system, mål eller spesielle brukere.

- **Sun Management Center 3.0**, er et produkt som er designet for Sun sine operativsystemer. Det tilbyr en enhet og nettverksressurser som gir en totalløsning for datasenteret. Det kan overvåke over tusen maskiner fra et enkelt konsoll, og automatiserer mange oppgaver som tidligere måtte gjøres manuelt.
- **Dell OpenManager** er et verktøy som egner seg til Dell-systemer. Det tilbyr grafiske brukergrensesnitt samt varsling til e-post og personsøkere.
- **Cisco** er et sentralt produkt i UNINETT, som overvåker sine egne ”bokser”, blant annet tilbyr Cisco konfigurasjonsverktøy, RMON-applikasjoner, produktspesifikk status overvåking, etc. De legger ikke vekt på å generere Nettverk og Administrasjonsverktøy (NAV) plattform, men tilbyr imidlertid integrasjon mot NAV plattformer som IBM, HP, og Sun.
- **Novells Netware** “... er det mest utbredte nettoperativsystemet på markedet og det eneste av de store operativsystemene som er spesielt utviklet for å styre nett.” Deres produkt kan benyttes både i Ethernet- og Token Ring-nettverk, men brukes mest i Ethernet-nettverk.
- **Cabletron Spectrum** Er et omfattende og intelligent NAV som er bygget etter objektorientert, klient-server arkitektur. Deres distribuerte NAV kan bl.a. tilby statusmonitor, datainnsamler, plotter, rapportgenerator, terskelmonitor, og hendelseshåndtering.[12]
- **Network Toolbox** Tilbyr en rekke nettverksdiagnoseprogrammer som kan finne brukerinformasjon om en spesifikk person på en spesifikk datamaskin (finger), kan søke på spesielle maskiner med IP-adresse (IP-adresse søk), eller skanne en maskin for nettverkstjenester (port skanner).
- **LightWatch** Dette produktet tilbyr fysisk oversikt over nettverksressurser. For eksempel kan man klikke på en valgt hub på kartet for å få oversikt over koblinger mot denne hub'en. Programvaren gir muligheten for å sette alarmer og utarbeide statistikker. LightWatch kan operere med HP OpenView, derfor kan alle OpenView og LightWatch funksjoner sammen integreres i en totalpakke.
- **EtherVisio** Er et verktøy som er designet for Ethernet LAN. Det kan benyttes for både 10 Mbps og 100 Mbps for å overvåke trafikken i et Ethernet-nettverk.

EtherVisio kan varsle om nettverkets tilstand og generere ulike typer rapporter. Det fungerer for alle Ethernet medietyper fordi det er et DOS-basert program.

Det bør påpekes at produktene er i konstant utvikling. Felles for all programvare, er at det stilles store krav. Programvaren skal ikke bare overvåke et system, men også kunne beskytte systemet mot fare. ”Vi er nå midt inne i et generasjonsskifte, hvor tradisjonelle IDS (*Intrusion Detection System*) produkter skiftes ut til fordel for nykommeren IPS (*Intrusion Protection System*).” [16]

## **2.4 Oppsummering**

Kapitlet gir en oversikt over generell situasjon av IT-administrasjon i norske bedrifter og forslag til løsninger.

SKD har valgt å bruke ITIL. Derfor gis en generell presentasjon av ITIL rammeverket. Dette er et valg som medarbeidere i SKD har gjort, derfor vil dette bli fokus i oppgaven videre.

Kapitlet omtaler også verktøy som benyttes i SKD for System- og Nettverksadministrasjon. Dette verktøyet som primært brukes ved overvåking kan man og vurdere som kilder for å automatisere oppdatering av CMDB.



## 3 Service support

*Dette kapitlet skal gi en dyp forståelse av modulen i ITIL kalt "Service Support". Jeg velger å beskrive bare de prosessene som benyttes videre i oppgaven, det vil si "Incident Management", "Problem Management", "Change Management" og "Configuration Management". Delen er skrevet med utgangspunkt i boken "IT Service Management an Introduction based on ITIL"[17]*

### 3.1 "Incident Management" (Hendelsesadministrasjon)

#### 3.1.1 Introduksjon

Hendelsesadministrasjon er en reaktiv aktivitet som gjenoppretter normale tjenester, eventuelt minimaliserer problemer i forbindelse med et tjenesteavbrudd.

En hendelse (incident) defineres slik i ITIL:

*"Any event which is not part of the standard operation of a service and which causes, or may cause, an interruption to, or a reduction in the quality of that service".[17]*

Hendelser inkluderer ikke bare maskin- og programvarefeil, men også tjenesteforespørsler, og får prioritet etter hvor stor effekt de kan ha, eller hvor viktige de er å ta tak i med. Hendelsesadministrasjonen har mekanismer som har evne til å redusere hendelsers effekt på infrastrukturen, eller hendelsers grad av viktighet i form av at det er behov for en mer eller mindre rask løsning. Eksempler i denne sammenheng kan være å bytte maskinvare eller opprette en ny printerkø. Effekt og viktighet kan også endres av andre faktorer, som økt trafikk i perioder, hvor hendelsen har effekt på flere brukere, eller i kritiske perioder, som for eksempel når en avisredaksjon har "deadline"

Hvis en hendelse ikke kan løses i førstelinje support, eskaleres den. Enten funksjonelt, eller hierarkisk. Funksjonell eskalering innebærer å trekke inn mer spesialisert kunnskap, og kan strekke seg fra første-, andre-, til n-linje support. Desto mindre organisasjon, desto færre eskaleringsnivåer. Hierarkisk eskalering innebærer å flytte hendelsen til et høyere organisatorisk nivå.

##### 3.1.1.1 Formål

Formål med hendelsesadministrasjon, er å gjenopprette tjenestetilbudet så fort som mulig, ved forstyrrelser forbundet med IKT-tjenester, samtidig med at effektene av dette blir så små som mulig for organisasjonens forretningsaktiviteter.

For at dette skal fungere optimalt, er det nødvendig med tett oppfølging av hendelser i statistisk øyemed. Slik kan man måle og forbedre prosessen, gi nødvendig informasjon til berørte parter, og kontrollere hva man lykkes med.

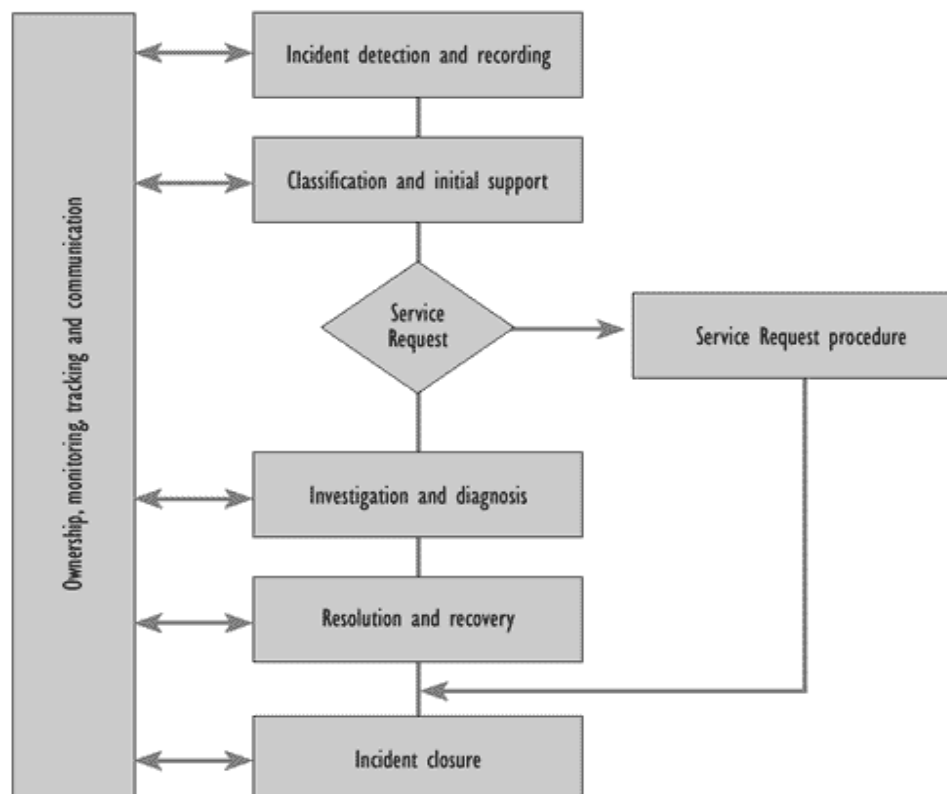
#### 3.1.2 Prosessen:

##### 3.1.2.1 Aktiviteter for hendelsesadministrasjon

- **Detektering og registrering:** En hendelse blir oppdaget, så registrert, og det lages en rapport for hendelsen. Hendelser kan oppdages av så vel brukere, maskin-/programvare, som serviceskranken ("Helpdesk"). Serviceskranken tar seg av registreringen av hendelser.

- **Klassifisering og ad hoc løsningsforslag:** Hendelsen klassifiseres etter type, status, effekt, viktighet, prioritet, SLA, etc. Det gis mulige forslag for hvordan løse eller arbeide seg rundt problemet.
- **Tjenesteforespørsel:** Hvis en slik forespørsel foreligger, initieres relevant prosedyre. Hvis ikke, går vi til neste punkt, som er såkalt matching (likhetssjekk).
- **Likhetssjekk:** Det kontrolleres om hendelsen er av kjent natur, og om den er relatert til en allerede eksisterende hendelse, et problem, eller en kjent feil, og om det allerede finnes en løsning eller såkalt workaround for denne typen hendelser.
- **Undersøke og diagnostisere:** Hvis det ikke finnes noen umiddelbar løsning, trer brukerstøtte til for å finne en løsning.
- **Beslutningstagen og gjenvinning:** Man løser problemet, og arkiverer løsningen for evt. senere bruk. Det kan være nødvendig med en RFC. Hvis ingen løsning finnes, vil hendelsen fortsatt være ”åpen”.
- **Lukke hendelse:** Når en løsning på et problem har blitt iverksatt, sender brukerstøtten en tilbakemelding til service skranken, som igjen sjekker med vedkommende som rapporterte hendelsen om løsningen har fungert. Hvis ting nå fungerer tilfredsstillende, lukkes hendelsen. I motsatt fall starter man prosessen med hendelsesadministrasjon på nytt i relevant stadium.
- **Eierskap, overvåkning, forfølge og kommunisere med brukere:** Service skranken, som har eierskap for alle hendelser, følger hendelsesforløpet til hendelser, og informerer brukere ved behov.

Figur 6 nedenfor viser prosessens ulike stadier.



Figur 6: Hendelsesadministrasjon aktiviteter (kilde[18])

### 3.1.2.2 *Relasjon med andre prosesser*

- **Konfigurasjonsadministrasjon ("Configuration Management"):** CMDB er sentral i alle "Service support" prosesser. Her finner man hva eller hvem som er berørt og konsekvensen en hendelse kan ha på organisasjonen, i form av for eksempel brukere, maskin-/programvare, tjenester etc. Konfigurasjonsadministrasjonen har oversikt over hvem som er ansvarlige for infrastruktur komponenter, slik at hendelser kan rutes til riktig adresse. Viktig med tanke på "workaround" alternativer, samt ved registrering.
- **Problemadministrasjon ("Problem Management"):** har kriterier for hvordan hendelser skal rapporteres, slik at evt. underliggende problemer kan identifiseres. Assisterer i tillegg hendelsesadministrasjonen ved å tilby informasjon om problemer, kjente feil, "workaround" alternativer, og midlertidige løsninger.
- **Endringsadministrasjon ("Change Management"):** Det kan hende at en forandring i infrastrukturen et eller annen sted kan være årsak til hendelsen. Hendelsesadministrasjonen gir informasjon til forandringsadministrasjonen om disse hendelsene. I motsatt fall kan man behandle hendelsen ved å gjøre forandringer i infrastrukturen. Forandringsadministrasjonen gir informasjon til hendelsesadministrasjonen ved å opplyse om endringer og disses status.

En annen forbindelse er at eksakt samme hendelse gjentar seg, denne hendelsen skaper da en forespørsel om endring ("Request for Change" (RFC)), og RFC er da midlertidig løsningen.

- **Tjenestenivå administrasjon ("Service Level Management (SLM))":** SLM har ansvaret for avtaler med kunder med hensyn på support-tjenester. Hendelsesadministrasjonen må ha oversikt over tjenesteavtalene, eller "Service Level Agreements (SLA)", slik at denne informasjonen kan benyttes i kommunikasjon med brukere. Man kan lage rapporter på bakgrunn av hendelsesarkivene, og ved hjelp av disse avgjøre om riktig tjenestenivå tilbys. For eksempel, hvis det således er mange duplikathendelser eller det er alvorlige problem som oppstår, må man sjekke om det er innenfor eller utenfor rammene gitt av SLA. Hvis det er utenfor SLA så er det ikke akseptabelt. Hendelse må bli registrert som et problem og må løses.
- **Kapasitetsadministrasjon ("Capacity Management"):** Tar for seg hendelser som har oppstått ved kapasitetsendringer, for eksempel ved for lang responstid eller manglende disk kapasitet.
- **Tilgjengelighetsadministrasjon ("Availability Management"):** Availability bruker hendelsesregisteret og status i Configuration Management for å overvåke tjenester og tildele status til tjenester. Denne informasjon er nyttig for å bestemme aktuelle tilgjengelig tjenester og responstid på tjenester.

## 3.1.3 **Prosessadministrasjon**

### 3.1.3.1 *Rapporter:*

Prosser kontrolleres basert på rapporter til ulike grupper. Rapportene utvikles med tanke på følgende funksjoner:

- Hendelsesadministrasjon: Bruker rapportene til å identifisere trender, konflikt med SLA, prøve å finne mangler ved prosessen, samt å følge opp hele prosessen.
- IT-avdeling: Henter rapporter fra ulike supportgrupper. De trenger informasjon om hvordan prosessen med å løse hendelser går, i tillegg til hendelsesfrekvensen.
- Tjenestenivå administrasjon: Handler om kvalitet på tjenester. Rapporter til kundene inneholder informasjon om tjenestenivået holdes i forhold til inngåtte avtaler.
- Prosessdirektører for andre prosesser i tjenesteadministrasjonen mottar informative rapporter som viser ulike hendelsesattributter, som kategori, prioritet, status, løste/uløste hendelser etc.

### **3.1.3.2 Viktige faktorer**

For at prosessen skal være vellykket, er det viktig at man har en god verktøykasse. Denne inkluderer bl.a. en CMDB som hele tiden er gyldig i forhold til gjeldende konfigurasjon og infrastruktur, altså at kartet viser terrenget. Videre er det nødvendig med en kunnskapsbase, kanskje i form av en database som viser kjente problem, og deres løsning. Andre faktorer er automatisering av deler av prosessen rundt hendelser, samt en tett forbindelse til SLM.

### **3.1.3.3 Parametre**

For å kunne følge hvordan prosessen fungerer, er det essensielt å ha parametre som sier noe om dette. De må kunne være målbare, og med klart definerte mål som bakgrunn. Disse målingene gir utgangspunkt for historiske data, som kan være med på å gi en retning med tanke på fokus.

### **3.1.3.4 Funksjoner og roller**

Ettersom prosessene er tilstedt i alle lag av organisasjonen, er det formålstjenlig å identifisere ansvarsområder for ulike aktiviteter. Tildeling av roller/funksjonsområder kan sørge for fleksibilitet i så måte. I små organisasjoner, eller for å redusere kostnader, kan roller og funksjonsområder kombineres.

## **3.1.4 Kostnader og mulige problem**

### **3.1.4.1 Kostnader**

Kostnader forbundet med hendelsesadministrasjon, innebærer bl.a. oppretting av prosedyrer, opplæring av personal, valg av verktøy for å understøtte prosessen, implementeringen og bruken av disse. Kostnadene avhenger i stor grad av bl.a. hendelsesadministrasjonens struktur, skala, og ansvarsområder.

### **3.1.4.2 Mulige problemer**

Som ved tilsvarende tilfeller av introduksjon av nye systemer/arbeidsmetoder, kan også hendelsesadministrasjon innebære problemer.

- Brukere og IT-personale dropper prosedyrene som er essensielle for at hendelsesadministrasjonen skal fungere. Eksempler er at informasjon om hendelser ikke blir registrert. Problemene blir løst der og da, eller det kan bli etablert en "workaround". Det fører til at andre prosesser involvert i hendelsesadministrasjonen ikke får full informasjon om tingenes tilstand. I tillegg får ikke IT-avdelingen god nok informasjon om tjenestenivåer og antall feil. Dette igjen leder til rapporter som

ikke beskriver situasjonen til fulle.

- Hvis det kommer altfor mange hendelser, kan det føre til at det ikke blir nok tid til å registrere hendelser ordentlig. Resultat er at andre involverte prosesser får ukorrekt informasjon, og det kan bli vanskelig å finne en god løsning.
- Eskalering: Hendelser blir eskalert hvis de ikke har en umiddelbar løsning. Men hvis et problem fører til altfor mye eskalering, kan det få for dårlig vurdering i tillegg til at det kan hindre spesialistene i deres daglige virke på andre områder.
- Mangel på klare definisjoner i tjenestekatalogen, SLA, eller OLA, eller mangel på enighet, kan gjøre det vanskelig for hendelsesadministrasjonen å møte brukernes og kundenes forventninger.
- Endring av arbeidsmetoder krever at grunnarbeidet, i form av informasjon og motivasjon, er gjort. I så måte er samarbeid på kryss og tvers av organisasjonen viktig.

### **3.1.5 Fordeler**

Hendelsesadministrasjon innebærer at hendelser kan bli løst med minimale konsekvenser for driften av organisasjonen. I tillegg ser man muligheter for økt produktivitet. Med fokus på kunder ved overvåkning av hendelser, er informasjon i forbindelse med SLA tilgjengelig.

Når det gjelder IT-delen av organisasjonen, ser man klare fordeler i forhold til hvordan hendelser påvirker SLA'er. Fordeler inkluderer i tillegg bedre personellutnyttelse, at alle hendelser blir registrert og registrert korrekt, som igjen sørger for en god oppdatering av CMDB. Summen av dette gir fornøyde brukere og kunder.

## **3.2 "Problem Management" (Problemadministrasjon)**

### **3.2.1 Introduksjon**

Hendelsesadministrasjon har ansvar for å gjenopprette tjenester så fort som mulig. Når målet er nådd, er hendelsesadministrasjonen ferdig med sin jobb. Det kan hende at feilen ikke ble tatt ved roten og at den kan oppstå igjen. Problemadministrasjonen har ansvar for å unngå at feil gjentar seg.

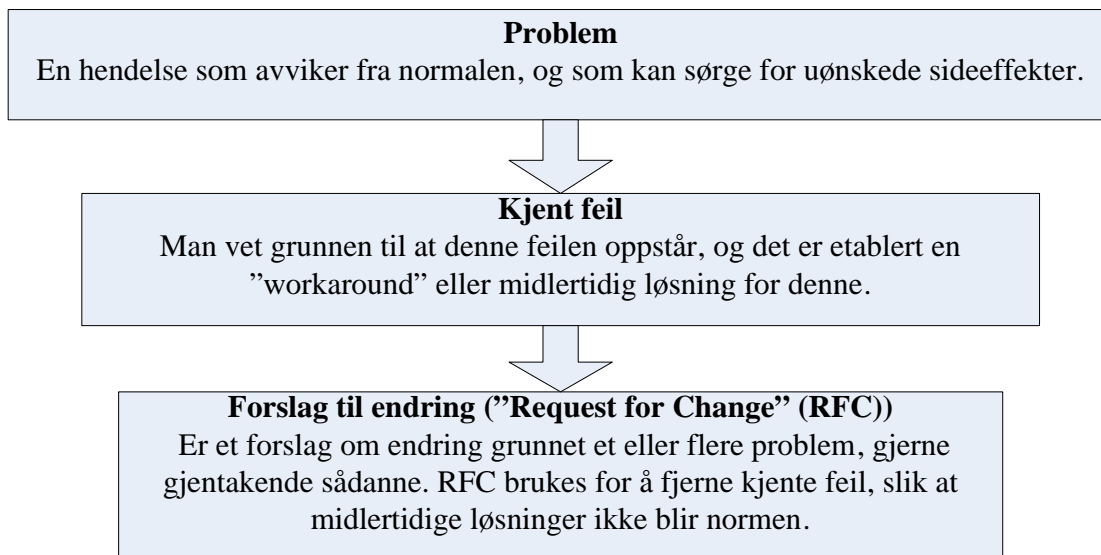
Problemadministrasjon samler all informasjon om infrastruktur og hendelser for å finne ut skjulte årsaker til eventuelt potensielle feil i tjenester.

#### **3.2.1.1 Formål**

Formålet med problemadministrasjon, er å finne ut årsaken til at hendelsene oppstår. I tillegg er ansvaret å lage såkalte "workarounds" som gir en umiddelbar omvei rundt problemet, og å lage midlertidige løsninger. I denne forbindelse som en støtte til hendelsesadministrasjon. Forøvrig skal problemadministrasjon produsere endringsforslag for kjente problemer, eller iverksette mekanismer for å begrense effekten av problemet.

#### **3.2.1.2 Definisjon**

Prosessen bruker følgende begreper: (Figur 7)



Figur 7: Relasjon mellom problem, kjent feil og RFC (kilde[17])

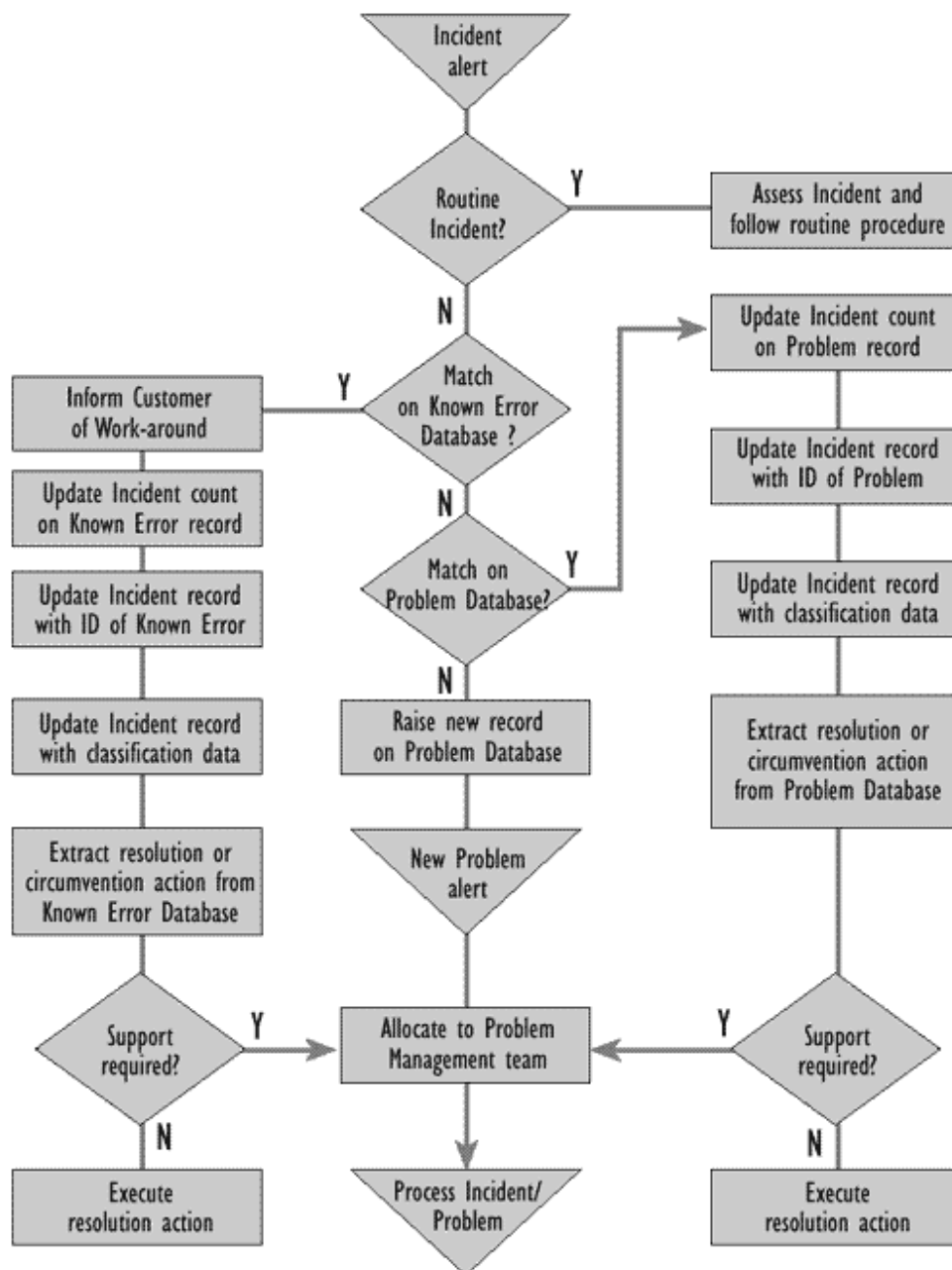
## 3.2.2 Prosessen

### 3.2.2.1 Aktiviteter som ligger under problemadministrasjon:

- **Problemkontroll:** Denne aktiviteten identifiserer problemet og undersøker hva som er årsaken, for å prøve å identifisere problemet som en kjent feil.
- **Feilkontroll:** Innebærer å overvåke kjente feil samtidig som nødvendigheten av en RFC vurderes. Disse tiltakene danner grunnlaget for å løse opp i kjente problemer via "Change Management" (endringsadministrasjon).
- **Proaktiv problemadministrasjon:** Generelt, så er ansvaret for denne aktiviteten å holde kvaliteten på infrastrukturen på et akseptabelt nivå. Proaktiv er det operative ordet her. Aktiviteten går altså ut på å analysere med tanke på å finne potensielle problemer, og forhindre at de skjer, bl.a. gjennom RFC.
- **Utarbeidelse og distribusjon av informasjon:** Aktiviteten sørger for å gi informasjon om "workarounds" eller midlertidig løsninger til hendelsesadministrasjonen og informere berørte brukere. CMDB og SLA'er brukes for å finne ut hva slags informasjon som skal distribueres, og til hvem.

Produkter av problemadministrasjon er en database over kjente feil, RFC, oppdatert problem-registrering med kjente feil, løsning eller workaround, samt registre over problemer som er lukket.

Problemadministrasjons prosedyre illustreres i Figur 8



Figur 8: Problemadministrasjons arbeidsflyt (kilde[18])

### 3.2.2.2 Relasjon med andre prosesser

- **Hendelsesadministrasjon:** Fører problemadministrasjonen med nødvendig informasjon, slik at problemer kan identifiseres og løses. PM på sin side hjelper hendelsesadministrasjonen med å tilby ”workarounds” og andre temporære løsninger, før den til slutt finner en egnet permanent løsning.
- **Endringsadministrasjon:** Sørger for å sette i verk foreslåtte endringer, også RFC-kall foreslått av problemadministrasjonen. Informerer PM om hvordan arbeidet med endringene skrider frem. Sammen evaluerer PM og CM endringene, og dette resulterer i en såkalt Post-Implementation Review (PIR).

- **Konfigurasjonsadministrasjon:** Står for informasjonen om infrastrukturen, og relasjoner mellom ulike komponenter. Dette hjelper PM til å danne et fullstendig bilde av infrastrukturen, slik at de riktige beslutninger kan taes med tanke på å løse opp i problemer, og ikke skape nye underveis.
- **Tilgjengelighetsadministrasjon:** Gir informasjon til PM om tilgjengelighetsnivåer. PM på sin side støtter AM ved å identifisere hva som forårsaker brudd i tilgjengelighet, og bøter på problemer. AM forsøker å holde infrastrukturens arkitektur designet slik at problemer og hendelser i minst mulig grad opptrer. PM og AM arbeider sammen i forbindelse med analyse av tjenestebrudd.
- **Kapasitetsadministrasjon:** Sørger for optimert utnyttelse av IT-ressursene. Kapasitetsadministrasjonen gir informasjon til PM som er nyttig i forbindelse med problemdefinering, og PM støtter CM i å identifisere og løse opp i kapasitetsproblemer.
- **Tjenestenivå administrasjon:** Igjen får PM hjelp til å definere problemer. Når det gjelder relasjon den andre veien, bør prosedyrene i PM understøtte gjeldende tjenestekvalitetsnivåer.

### 3.2.3 Prosesskontrollen

#### 3.2.3.1 *Rapporter:*

Det kan produseres en lang rekke rapporter som kan si noe om hvordan prosessen fungerer. Det kan handle om tid, effektivitet, statusrapporter, forslag til forbedringer etc. Rapportene bør ha en direkte sammenheng med hvilket nivå man legger seg på når det gjelder problemadministrasjon.

#### 3.2.3.2 *Viktige faktorer:*

For at problemadministrasjonen skal fungere optimalt, er det flere ting som må være på plass. Det er viktig at rammeverket rundt prosessen er klart definert og at man har etablert et klart mål med arbeidet. I tillegg må nødvendige ressurser være tilgjengelig. Godt dokumenterte prosedyrer, hendelseskontrolldata som beskriver konsist, samt et fruktbart samarbeid mellom PM og IM er andre faktorer som virker inn på hvordan problemadministrasjonen vil fungere.

#### 3.2.3.3 *Funksjoner og roller:*

Funksjoner er ikke definert gjennom hierarki, men heller gjennom kunnskap og ansvar. Således kan det være ønskelig med en løsning basert på roller, eller en kombinasjon av roller.

### 3.2.4 Kostnader og mulige problemer

#### 3.2.4.1 *Kostnader*

Kostnader ved problemadministrasjon er bl.a. til diagnoseverktøy, brukerstøtte, og personell. Personalkostnader kan være både materielle (eksempelvis leie av eksterne konsulenter), som immaterielle (eksempelvis tid).

#### 3.2.4.2 *Mulige problem*

Mulige problem kan oppstå i løpet av problemadministrasjon prosessen er:



- Dårlig forbindelse og samarbeid mellom hendelsesadministrasjon og problemadministrasjon.
- Dårlig kommunikasjon om kjente feil fra utviklingsmiljøet til de som er de daglige brukerne.
- Mangel på entusiasme. Overgangen fra en situasjon med ”ad hoc”-løsninger på problemer, til en mer rigid modell basert på klart definerte prosedyrer og prosesser, bør foregå på en måte som holder nøkkelpersonell oppdatert informasjonsmessig. Dette vil hindre at folk mister tråden og etablerer uformelle tilnærminger til problemer.

### 3.2.5 Fordeler

Problemadministrasjon sikrer at feil i tjenester og infrastruktur blir identifisert og løst på en mannskaps-, tids- og kostnadseffektiv måte ved riktig implementering av arbeidsmetoden. Prosessen sørger for at symptomer identifiseres og diagnoser stilles. Midlertidige og permanente løsninger etableres, og alt dokumenteres. Rapportene fra prosessene gir viktig informasjon om kvaliteten på IT-infrastrukturen. Feil og problemer løses og forhindres.

## 3.3 ”Change Management” (Endringsadministrasjon)

### 3.3.1 Introduksjon

Når en organisasjon vil gjøre endringer i IT-infrastrukturen, er det behov for en viss grad av administrasjon for å understøtte prosessen. Prosessen skal sørge for at forandringer utføres på en så effektiv måte som mulig, med et øye til faktorer som risiko og forretningskontinuitet, konsekvenser og ressursbehov.

### 3.3.2 Formål

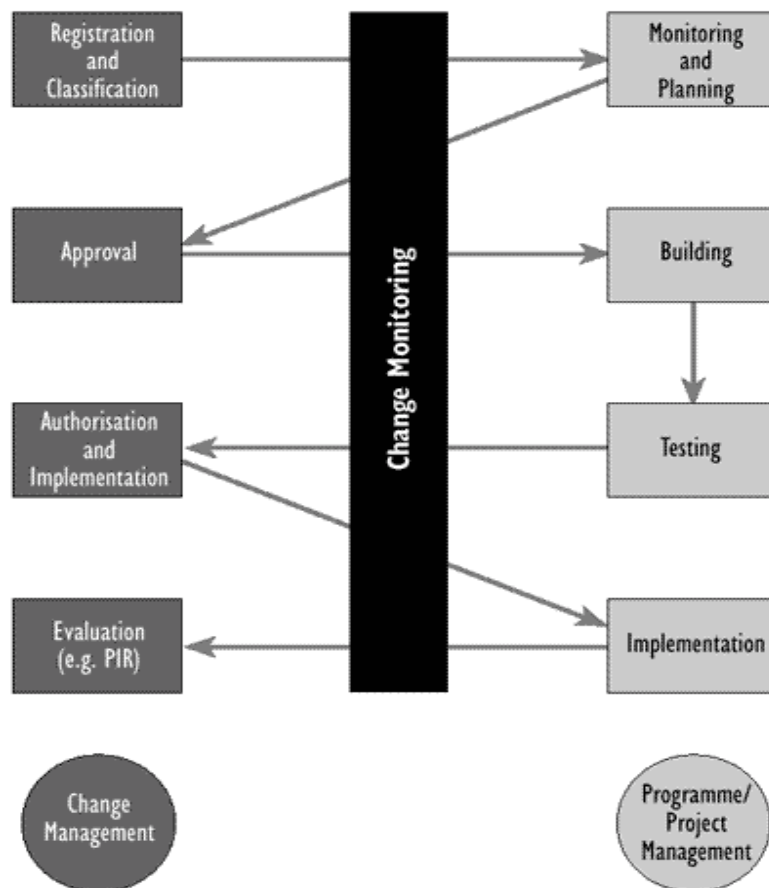
Sørge for en strukturert og definert endringsprosess, ved å bruke prosedyrer, redskaper og se til at forhold legges til rette for en god planlegging av endringer på en effektiv og kostnadseffektiv måte. Noe av det viktigste er at endringer ikke forstyrrer eksisterende IT-konfigurasjonen, slik at forretningsdriften kan foregå på tilnærmet normal måte under de pågående endringene, og at overgangen blir smidig.

### 3.3.3 Prosessen

#### 3.3.3.1 Endringsadministrasjonens aktiviteter

- **Registrere og klassifisere** ønske om/behov for endring(er).
- **Sette opp et endringsstyre** (CAB: Change advisory board), og holde endringsstyremøter, hvor et bredt utvalg av IT-ressurser møtes for å diskutere og vedta endringer.
- **Koordinere endringsprosessen.**
- **Evaluere prosessen** for å se på nytteeffekten av endringer, samt lokalisere forbedringspotensiale i arbeidet med endringer.

Figur 9 nedenfor viser grense mellom endringsadministrasjon og prosjektadministrasjon



Figur 9: Grense mellom endringsadministrasjon og prosjektadministrasjon (kilde[18])

### 3.3.3.2 Relasjon til andre prosesser

- **Hendelsesadministrasjon:** Nesten alle RFC er opprinnelig fra hendelsesadministrasjonen, det kan komme en situasjon der flere hendelser har bare workaround som løsning, det vil si at det fremdeles kommer duplikathendelser. Disse kan kun løses ordentlig når årsaken blir funnet. Når forandring er gjennomført, skal hendelsesadministrasjonen underrettes.
- **Konfigurasjonsadministrasjon:** Endringsadministrasjon er nært beslektet med konfigurasjonsadministrasjon, så disse kan ofte sees under ett. Hovedsaken her er at konfigurasjonsadministrasjonen trenger å vite hvilke endringer som er gjort og hvilke konfigurasjonsenheter ("Configuration Item" (CI)) som influeres.
- **Problemadministrasjon:** Problemadministrasjonen finner ut hva som bør endres, og endringsadministrasjonen tar seg av dette. Imidlertid må endringene som foretas ikke skape ny problemer.
- **Tjenestenivå administrasjon:** Gir informasjon om hvilke konsekvenser endringer kan få.
- **Tilgjengelighetsadministrasjon:** Gjennom endringsstyret ("Change advisory board" (CAB)) foreslås det endringer som kan ha positive ringvirkninger for

tilgjengeligheten til tjenester. I motsatt fall kan tilgjengeligheten lide under endringer, derfor er det viktig med dialog for å unngå uheldige effekter som følge av endringer.

- **Kapasitetsadministrasjon:** Endringer kan forårsake behov for øket kapasitet i IT-infrastrukturen i form av f.eks. lagringskapasitet, nettverkskapasitet etc.
- **Kontinuitetsadministrasjon:** Det er nødvendig at kontinuiteten holdes ved like når det gjelder tjenester. Ettersom endringer kan forårsake at tiltak som skal iverksettes ved forstyrrelser i tjenesteleveringen ikke virker til fulle, må kommunikasjonen holdes ved like slik at kontinuitetsadministrasjonen har klart for seg hvilke forandringer som gjøres, og, om mulig, hvilke potensielle konsekvenser disse kan ha.
- **Versjonsadministrasjon:** Endringer i program-/maskinvare er versjonsadministrasjonens ansvar. Endringsadministrasjonen har ansvaret for å sende ut nye utgivelser til bruk i infrastrukturen.

### 3.3.4 Prosesskontroll

Proessen kontrolleres med rapporter som sier noe om IT-infrastrukturen i organisasjonen med fokus på endringer. Disse rapportene kan bl.a. si noe om hvor mange endringer som foretas, når, hvor, hvorfor og hvordan de foretas. Man kan i tillegg måle faktorer som går på effektivitet.

### 3.3.5 Kostnader og mulige problem

Kostnader forbundet med endringsadministrasjon, er hovedsakelig på personell- og verktøysiden.

Mulige problem inkluderer bl.a. papir-basert dokumentasjon, motstand mot overvåkning av infrastrukturen, og implementering av endringer uten å følge ”tjenesteveien”.

### 3.3.6 Fordeler

For organisasjonen innebærer det muligheten til å oppnå en infrastruktur som er mer robust mot følger som forandringer kan føre med seg. I tillegg kan man oppnå bedre kostnadseffektivitet i forhold til endringer. For den enkelte arbeidstaker er det en fordel å ha et stabilt system, hvor endringer ikke hindrer dem i å utføre sine arbeidsoppgaver.

## 3.4 ”Configuration Management” (konfigurasjonsadministrasjon)

### 3.4.1 Introduksjon

Ansvar til konfigurasjonsadministrasjonen er å hele tiden ha en oversikt over gjeldende IT-infrastruktur, både med tanke på enkelte konfigurasjonseenheter, og relasjonene mellom dem. I tillegg skal informasjon og dokumentasjon være tilgjengelig for andre administrasjonsprosesser.

### 3.4.2 Prosessen

#### 3.4.2.1 Konfigurasjonsadministrasjonens aktiviteter

Disse inkluderer planlegging av arbeidet med tanke på strategi, retningslinjer, mål, samt identifisering og kartlegging av komponenter for innføring i databasen (CMDB), kontroll av databasen, holde oversikt over nåværende og historisk informasjon over

konfigurasjonskomponenter, verifisere databasen, og rapportering til andre prosesser ved behov.

- **Planlegging:** dette trinnet definerer formålet med administrasjonen, dens rekkevidde, rutiner, konvensjoner, og prosedyrer, og hvordan det hele skal organiseres.
- **Identifikasjon:** her skal konfigurasjonsstrukturer velges, og identifisering for alle konfigurasjons komponenter, deres relasjoner og dokumentasjon.
- **Kontrollere CI:** sikrer at alle forandringer av konfigurasjonskomponenter er oppdatert og dokumentert i databasen.
- **Statuskontroll:** innebærer å produsere rapporter som sier noe om utviklingen og statusendringer for konfigurasjonsenheter, som kan forkomme på grunn av endringer i infrastrukturen.
- **Bekreftelse og revisjon:** innebærer å sikre at kartet stemmer med terrenget i den forstand at databasen beskriver alle CI riktig, og på en slik måte at det ikke bryter med noen av prinsippene for hvordan konfigurasjonsenheter skal registreres og følges opp.

#### 3.4.2.2 *Relasjon med andre prosesser*

- **Hendelsesadministrasjon:** Konfigurasjonsadministrasjonen stiller til rådighet informasjon om infrastrukturen slik at hendelsesadministrasjonen kan få en oversikt over hvilke brukere, kunder, tjenester som er berørt av en hendelse.
- **Problemadministrasjon:** Bruker databasen (CMDB) som bakgrunn ved analysing av hendelser og problemer.
- **Endringsadministrasjon:** Bruker databasen aktivt i arbeidet med endringer, både til å hente ut, og å legge inn informasjon.
- **Versjonsadministrasjon:** Stiller med informasjon om status og versjoner og endringer som er gjennomført. Henter ut informasjon fra databasen ved endringer i IT-infrastrukturen.
- **Administrasjon av tjenestenivå:** Er ute etter informasjon om tjenester og relasjoner mellom tjenester og underliggende infrastruktur.
- **Finansiell administrasjon:** Ser på bruk av tjenester og konfigurasjonskomponenter. Ulike data hentet ut fra databasen er med på å legge grunnlag for prising av tjenester.
- **Tilgjengelighetsadministrasjon:** Bruker CMDB'en for å se på forhold mellom konfigurasjonsenheter (CI) og tjenester, for å identifisere eventuelle svakheter som kan gå ut over tilgjengeligheten til tjenester.
- **Kontinuitetsadministrasjon:** Bruk av CMDB'en for å sette vilkår for systemgjennoppretting.
- **Kapasitetsadministrasjon:** Bruker CMDB'en i forbindelse med administrasjon av kapasiteten.

#### 3.4.3 **Prosesskontroll**

Konfigurasjonsadministrasjonen bør hele tiden holde et øye på effektivitet, slik at prosesser

leder fram til en oversikt som gir et dagsaktuelt og tilgjengelig bilde over IT-infrastrukturen, uten å belaste organisasjon med unødvendig ekstra byråkrati.

### **3.4.4 Kostnader og mulige problem**

#### **3.4.4.1 Kostnader**

Kostnader for prosessen vil innebære økte lønnsutgifter i og med at det sannsynligvis vil være behov for økt personell. I tillegg vil det være behov for økte investeringer i ulike typer verktøy for å understøtte prosessen, foruten kostnader forbundet med tilpasning og opplæring av medarbeidere.

#### **3.4.4.2 Mulige problem**

Uansett i hvilket omfang man etablerer konfigurasjonsadministrasjon, og hvilket detaljnivå man legger seg på, kreves både mannskapsressurser såvel som maskin-/programvareressurser for å holde kontroll på infrastrukturens skiftende konfigurasjon. Balansen finnes i krysningspunktet mellom behov for informasjon og tilgjengelighet av ressurser. Andre faktorer som spiller inn inkluderer bl.a. bruk av dataverktøy kontra papirbaserte løsninger, når endringer skal iverksettes, og hvordan, og faren for at folk omgår systemet. Overærgjerrig taktiske planer og definering av CI på galt nivå er mulige fallgruber for organisasjoner.

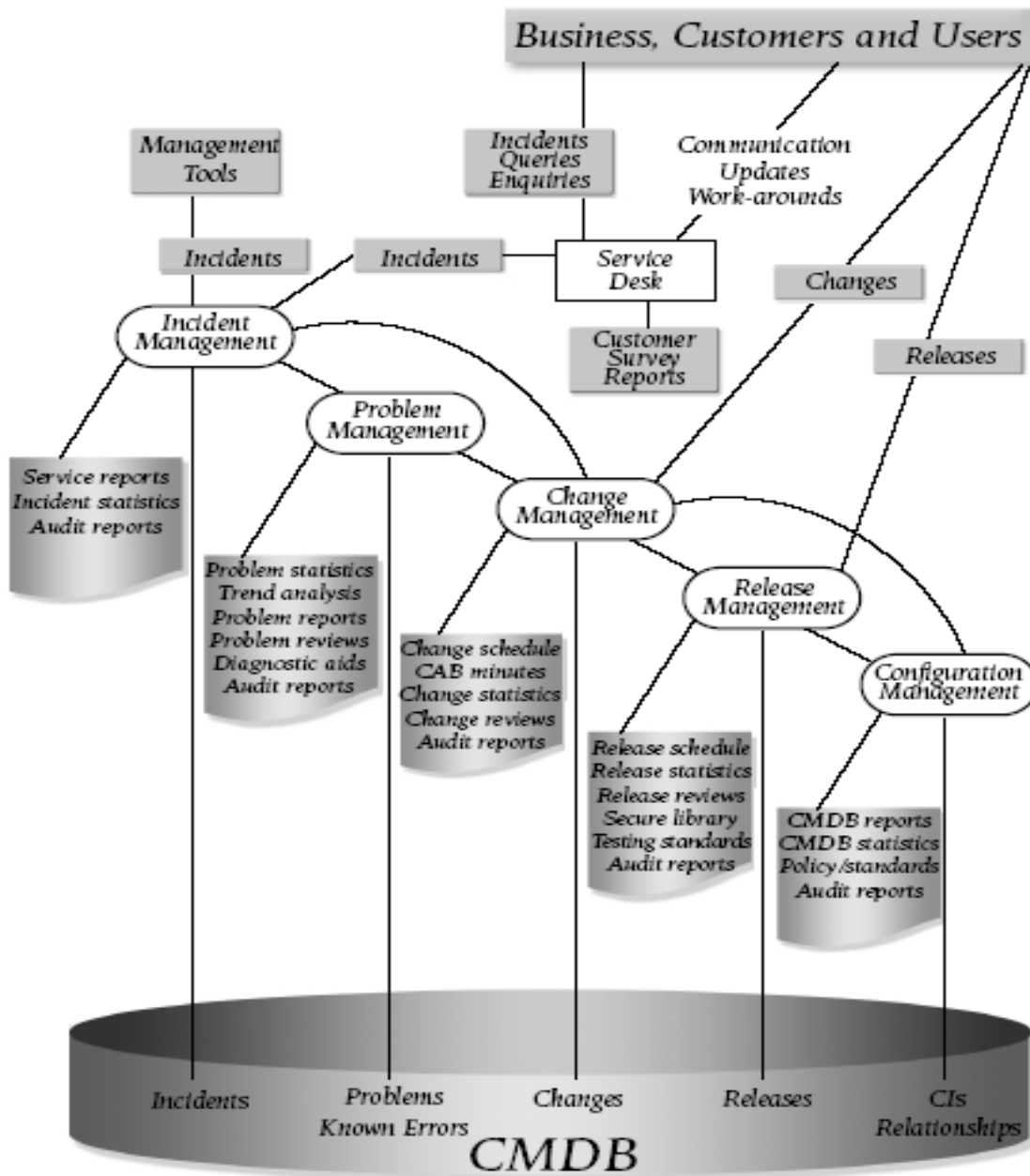
### **3.4.5 Fordeler**

Det å holde en oppdatert oversikt over IT-infrastrukturen, dets dokumentasjon og relasjoner mellom ulike konfigurasjonsenheter, gir verdifull informasjon til andre enheter innen organisasjonen, i tillegg til at man bl.a. kan oppnå bedre kvalitet på tjenester, løse problemer mer effektivt, forbedre sikkerheten, få bedre kontroll over budsjettering av vedlikehold, kontrakter og lisenser. Andre fordeler er støtte til og forbedret versjonsadministrasjon, og støtte til proaktive aktiviteter i problemadministrasjonen ved å bruke overvåkningsverktøy for å finne potensielle problemer.

## **3.5 Dette er en CMDB**

Formålet med CMDB'en er at den skal gi en oversikt over hvilke konfigurasjonsenheter som knyttes til hvilke tjenester. Den kan vise attributter eller nivåer ved SQL-spørringer. F.eks. når en tjeneste er nede, kan driftsansatte i førstelinje support finne ut hvor mange brukere som er berørt, for å tildele prioritet til hendelsen. Andre- og tredjelinje support har interesse for hvilke CI som er involvert i denne hendelsen.

CMDB inneholder informasjon om (navn på figurene i parentes): hendelsesadministrasjon ("Incident Management"), problemadministrasjon ("Problem Management"), endringsadministrasjon ("Change Management"), versjonsadministrasjon ("Release Management") og konfigurasjonsenheter og deres relasjon ("CIs Relationship"). Figur 10 illustrerer innholdet i en CMDB.



Figur 10: Detaljene og sammenhengene i "Service Support" (kilde[9])

Å bygge opp en CMDB kan deles inn i 5 aktiviteter: [19]

1. Identifisere konfigurasjonsenheter
2. Skille mellom logiske og fysiske konfigurasjonsenheter
3. Tegne CI-klassifikasjonsstruktur
4. Spesifisere CI-attributter
5. Spesifisere CI-relasjoner.

Et av hovedprinsippene i et CMDB-design er at attributter må stemme overens med prosessene som er registrert.

### **3.6 Oppsummering**

Dette kapitlet gir et innblikk i de prosessene som skal brukes i oppgaven i praksisdelen. Gjennom kapitlet kan leseren skaffe seg en helhetlig prosesskunnskap som er beskrevet i ITIL-rammeverket. Denne forståelsen pluss informasjon om organisasjonen i kapittel 4, er grunnlaget for å skissere prosessaktiviteter for organisasjonen. Det gis også et lite innblikk i hva en CMDB er, og hvordan det bygges.

## 4 Litt om Skattedirektoratet (SKD)

For å skape et bilde av hvordan situasjonen er i dag, er det formålstjenlig med en presentasjon av skatteetatens organisasjon både rent generelt, men samtidig med hensyn på etatens IT- løsninger. Dette avsnittet henter sin informasjon bl.a. fra dokumentet ”Skatteetatens organisering og IT-løsninger” [20], SKD sitt intranett, intervju med ansatte, og samtaler med medarbeider fra MIT-prosjektet.

### 4.1 Skattedirektoratet i tall. [21]

Skatteetatens er underlagt Finansdepartementet. Det er en stor organisasjon som består blant annet av Skattedirektoratet, Sentralskattekontoret, Oljeskattekontoret, Fylkesskattekontor, Skattefogdekontor, Ligningskontor (LK) og Folkeregisteret (FR). Det omfatter en stor mengde ansatte. Tabellene nedenfor forteller mer om Skatteetaten og Skattedirektoratet.

Skatteetaten består av:

Skattedirektoratet (SKD)
19 Fylkesskattekontor
18 Skattefogdkontor
Sentralskattekontoret for utenlandssaker
Sentralskattekontoret for storbedrifter
Oljeskattekontoret
99 Ligningskontor (LK) og Folkeregisteret (FR)

(Kilde [22])

Ansatt i skatteetaten

Ansatt i hele skatteetaten cirka	6 757
Skattedirektoratet	593
Fylkesskattekontorene	1 209
Skattefogdkontorene	600
Sentralskattekontorene	168
Ligningskontorene og folkeregistrene	4 186

(Kilde [22])



## 4.2 IT-avdelingen i SKD

IT- avdelingen har ansvar for utvikling og drift av data- og informasjonssystemene for hele etaten. Bruk av informasjonsteknologi er et viktig verktøy i virksomhetsutviklingen og i utviklingen av nye produkter og tjenester.

IT- avdelingen er delt inn i fire seksjoner:

- Systemseksjonen har ansvar for utvikling og vedlikehold av etatens skatterelaterte applikasjoner
- Driftsseksjonen med totalansvar for all drift både sentralt og ute på all kontorer i ytre etat
- Teknikkseksjonen som har ansvar for infrastruktur og dimensjonering av anskaffelser og implementering av nye løsninger.
- Prosjektseksjonen har ansvar for oppfølging, ledelse og gjennomføring av etatens IT-relaterte prosjekter.

I tillegg har IT- avdelingen en egen gruppe som jobber på tvers av alle disse seksjonene med sikkerhetsløsninger for etaten.” [23]

## 4.3 Teknisk bakgrunn

”Skatteetatens kontorer deles i to hovedkategorier hvor direktoratets sentrale avdelinger på Helsefyrt og i Grimstad utgjør den ene kategorien som huser alle sentrale datasystemer og alle interne funksjoner og tjenester. Den andre delen omtales som ytre etat og består av alle kontorer som har direkte kontakt med skatteyttere”. [24]

### 4.3.1 Datautstyr i generell

Tabellen nedenfor viser ulike typer datautstyr skatteetaten benytter.

Utstyr:	Totalt antall:
PC-er	7200 stk
Printere	1000 stk
Servere (OS390, AIX og NT)	400 stk
Lagringsnett (SAN)	110 porter
Brukerdata på disk	10 TB
Brukerdata på tape	115 TB
Telefonsentraler (MD110)	150 stk
Telefonapparater	6000

Utstyr:	Totalt antall:
Switcher	850
HUB	160
Rutere	400

*Kilde: Oversikt over Skatteetatens infrastruktur [24]*

Det finnes altså 99 likningskontor/enheter som består av distriktskontor med eventuelt etatskontor(er) under seg. Alle Distriktskontorene har kommunikasjon mot Helsfyr og Grimstad hvor de sentrale database- og driftsløsninger er plassert.

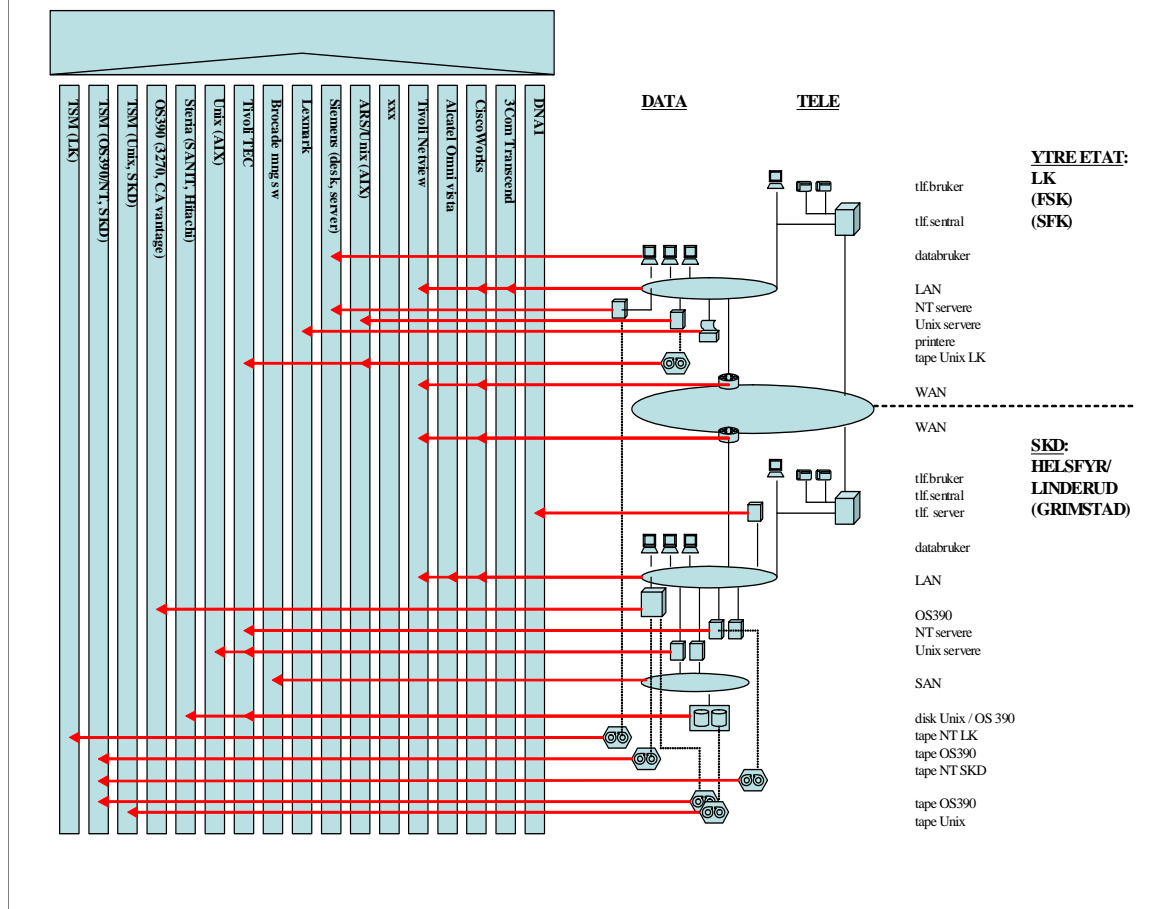
Tjenestene som skatteetaten leverer, er mange og omfattende. Noen av hovedoppgavene er:

- Folkeregistrering
- Utstedelse av skattekort og skatteberegning
- Selvangivelser, næringsoppgaver etc.
- Skattefastsettelse på formue og inntekter, beregning og innkreving av avgifter (mva, investeringsavgift, etc.)
- Faglig ansvar for kommunale avdelinger

### **4.3.2 Driftsverktøy**

Figur 11 illustrerer verktøy brukes for drift i Skatteetaten.

SW for drift av Skatteetatens tele- og datanett.



Figur 11: Software for drift av Skatteetatens tele- og datanett (kilde [25])

Tabell over programvare som benyttes i drift:

Sted:	Produkt:	Til drift av:	Bruksområde:
LK/FR SKD	SMIT (System Management Interface Tool)	AIX: Advanced Interactive Executive (IBM's version of Unix.)	Sikkerhetsadministrasjon Brukeradministrasjon Menyer Backup Logging Printeradministrasjon Prosesshåndtering
LK/FR SKD	Tivoli 3.6.1 med følgende: T/EC (Tivoli Enterprise Consol) Tivoli User admin Tivoli Distributed Monitoring	NT servere og PC-er	Overvåkning Brukeradministrasjon Logging Utstyrsoversikt Sikkerhetsadministrasjon

Sted:	Produkt:	Til drift av:	Bruksområde:
	Tivoli inventory Tivoli security manager Tivoli Remote Control Tivoli software distribution Tivoli Storage Manager/ ADSM (ADSTAR Distributed Storage Manager)		Fjernkontroll Programvare distribusjon Backup
LK/FR SKD	PUCS	AIX	Programvare-distribusjon
LK/FR SKD FSK	Cisco Works  MRTG (“The Multi Router Traffic Grapher”)  nGenius Performance Manager 1.4  Tivoli Netview  D.N.A. 5.0 (Dynamic Network administration)	Cisco rutere og switcher  WAN  Netscout prober  LAN og WAN  MD110 PBX: <i>private branch exchange</i> , en privat telefon nettverk brukes i virk- somhet	Konfigurasjonsverktøy  Kapasitetsovervåking  Kapasitetsovervåking og rapportering  Nettverksovervåking  Konfigurasjonsverktøy, ekspedient- og katalogtjenester
SKD	OmniVista  Concord’s eHealth Suite	Alcatel switcher	Konfigurasjonsverktøy og overvåking  Kapasitetsovervåking og rapportering
FSK	Tivoli 3.6.2 med følgende: T/EC (Tivoli Enterprise Consol)	NT servere og PC-er	Overvåkning

Sted:	Produkt:	Til drift av:	Bruksområde:
	Tivoli User admin Tivoli Distributed Monitoring Tivoli inventory Tivoli security manager Tivoli Remote Control Tivoli software distribution  Tivoli Storage Manager	AIX	Brukeradministrasjon Logging Utstyrsoversikt Sikkerhetsadministrasjon Fjernkontroll Programvare distribusjon  Backup

Kilde: Utdrag fra VEDL- A- Skatteetatens Organisering og IT løsning [25]

## 4.4 Brukerstøtte

”Brukerstøtten er et sentralt kontaktpunkt i en organisasjon, hvor ansatte kan få hjelp til å finne løsninger på sine IT-problemer, eller få besvart tekniske spørsmål.” [19]

I en stor organisasjon, kan man med en god brukerstøtte håndtere mange hendelser og løse problemer på en systematisk måte. I tillegg kan brukere veiledes slik at de kan undersøke og rette opp i enkelte problemer selv.

SKD har for tiden to avdelinger som driver med brukerkontakt. Den ene avdelingen er brukerkontakt for interne brukerne. Dvs. de som jobber lokalt på Helsfyr. Her har SKD 593 ansatte. Den andre avdelingen er service avdelingen i Grimstad (BSS). Denne avdelingen har ansvar for alle Ligningskontorene, Fylkesskattekontorene, Skattefogdkontorene og Sentralskattekontorene som hører til Skatteetaten over hele landet.

### 4.4.1 Brukerkontakten på Helsfyr

Avdelingen på Helsfyr er Skattedirektoratets egen IT supporttjeneste. De skal ta imot problemer/hendelser, registrere disse, og sørge for at de løses på riktig måte. Henvendelsene til brukerkontakten, er bredspektret i den forstand at de kan innebære fra relativt enkle problemer knyttet til innlogging, internett og programvare, til mer komplekse hendelser som bl.a. feil i databaser og enkelte maskinvareproblemer.

Remedy, Candle, Notes er noen av de verktøy som kan benyttes i brukerstøtte. Brukere kan selv kjøre brukerstøtteverktøyet for å lese FAQ, finne ut feil, sjekke status på sine endringsforespørsler. Verktøyet har innebygget en påminnelsesfunksjon som automatisk kan gi beskjed om kommende problemer. Når problemet er løst, kan det gis automatisk beskjed til brukere og automatisk endringer i konfigurasjon.

Brukerkontakten på Helsfyr benytter Remedy, Action Request System (ARS) (Figur 12) for å registrere hendelsene. Alle hendelser meldes inn, men de enkleste hendelsene, som ikke har den helt store effekten på driften, blir ikke registrert. De som derimot må registreres, er hendelser som bruker tid og ressurser. Disse blir meldt inn i systemet via enten telefon eller email fra brukere, eller ved automatisk registrering via ulike overvåkningsverktøy.

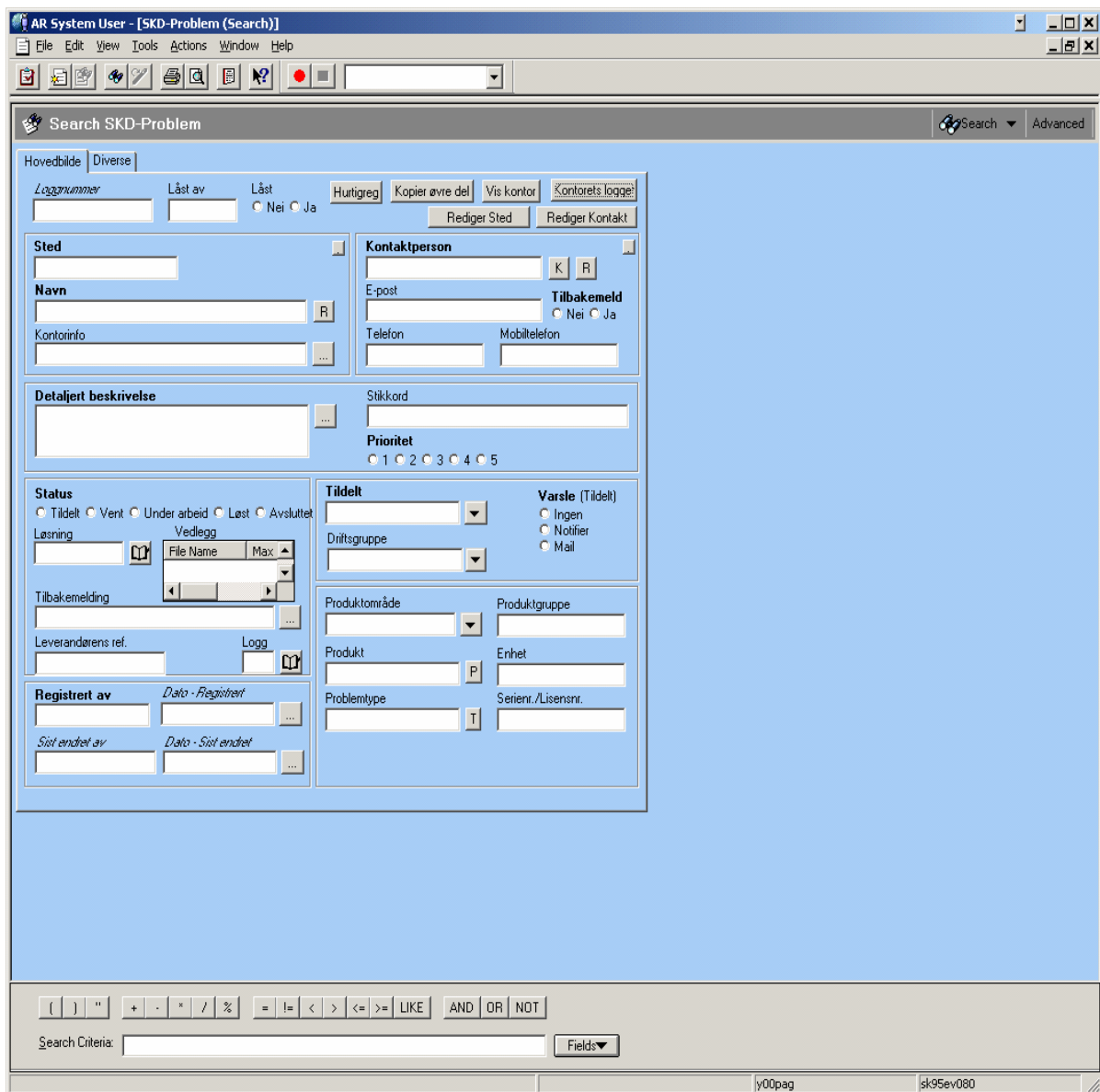
ARS tilbyr muligheten til å registrere hendelser med mange attributter. I applikasjonen kan man tildele loggnummer, angi prioritet for hendelse, samt registrere hendelseskilde, kontaktperson med navn, e-post, telefonnummer, detaljert beskrivelse av hendelse og mottakers navn, dato etc. I dette programmet kan man også finne data om hendelser, så som hendelsesstatus, når en hendelse sist ble endret, om det finnes vedlegg, leverandørsreferanse, og om det finnes en løsning. I tillegg vises ”matchende” problemer fra tidligere, og det søkes å finne løsning på bakgrunn av hvordan disse ble løst.

Utfra ARS kan man utføre mange statistikkoppgaver, som hvor mange hendelser som mottas, hvor mange er løst, hvor ofte en hendelse oppstår, om det bør defineres som et (tilbakevendende) problem etc. Denne statistikken er til hjelp for overvåkning av hvordan arbeidet utføres, og kan bidra til en bedre administrasjon av arbeidsressursene.

Avdelingen er også i gang med å bruke en applikasjon som heter **InfoChannel**. InfoChannel er et enkelt program som beskriver hvordan man kan løse enkelte problemer, skrevet av Hanna Meland [26]. Hun jobber i brukerteknisk gruppe i teknisk seksjon i SKD. Da Meland begynte i 2001, hadde brukekontakten ingen formell rutine i forbindelse med selve jobbutførelsen. Hvilke rutiner og kunnskaper som ble brukt, var opp til hver enkelt arbeidstaker. Og løsninger på problemer var ofte ofre for tilfeldighetenes spill.

Hanna Meland laget et lite program som hun kalte InfoChannel. Det var først og fremst utviklet med tanke på hennes eget bruk. Etter hvert ble imidlertid dette programmet brukt som et støtteverktøy for hele brukerkontakt avdelingen. Programmet er et enkelt system som beskriver hvordan man kan løse enkelte problemer.

InfoChannel systemet har altså ingen oversikt over relasjoner mellom infrastrukturkomponentene, men fokuserer på rene rutinebeskrivelser, og tilbyr enkel informasjon om infrastrukturen i en hierarkisk form.[27]



Figur 12:Hovedbildet i Remedy ARS

#### 4.4.2 Brukerstøtte Service (BSS) avdelingen i Grimstad:

BSS har ansvaret for brukerstøtte for alle ligningskontor (LK) og fylkeskattekontorer (FSK). Ansatte grupperes i forskjellige faglige grupper. Således har LK-gruppen ansvaret for hendelser fra ligningskontor, og FSK-gruppen skal løse hendelser fra fylkeskattekontorene.

I likhet med avdelingen på Helsfyr, har de ikke prosessorientert arbeidsmåte, og hver enkelt ansatt bruker sine kunnskaper og sin erfaring for å finne løsninger på hendelser som måtte oppstå, gjennom en teknisk orientert arbeidsmåte. Etter hvert som problemer oppstår, søkes det å finne tekniske løsninger for disse. Uten en overordnet aktivitet, som en prosessorientering vil være.

Ved en prosessorientert arbeidsmåte, har man en hendelsesadministrasjon som tilrettelegger en fast rutine for alle problemer. Man følger altså en fastlagt prosedyre steg for steg. Dette sikrer at man ikke glemmer deler av prosedyren, og man unngår dermed et eventuelt kaos som kan oppstå som følge av at man gjør et eller annet feil i løsningsprosessen.

ITIL kan tilby en god atmosfære for proaktiv hendelsesadministrasjon i og med at det er et rammeverk som er tuftet på prosessorientering. En stor del av personalet i BSS har gått gjennom grunnleggende kurs om ITIL. Personalet ved brukerkontakten på Helsfyr skal også få tilbud om ITIL-kurs i nær fremtid, slik at de kan lære seg, og benytte en ny arbeidsmåte i sitt virke i Skattedirektoratet.

## **4.5 Om MIT-prosjektet i SKD**

### **4.5.1 Formål**

Skattedirektoratet har foretatt en kartlegging av driftsprosesser, og konkludert med at disse må forbedres. De vil ha en tilpasning av prosess-støtte verktøy til sine behov, og ha dem integrert i en samlet løsning. SKD har startet et prosjekt som skal etablere en samlet drifts- og overvåkningsløsning, og de vil at dette prosjektet skal omfatte prosessforbedring med ITIL som utgangspunkt[28].

De forskjellige seksjonene og gruppene i IT-avdelingen ønsker å ha en oversikt over infrastrukturen, både med tanke på program- og maskinvare. Bakgrunnen for dette, er at en slik oversikt gir et godt utgangspunkt for å bestemme hvor og hvordan man eventuelt skal oppgradere. Med en slik oversikt vil man i tillegg lettere kunne identifisere flaskehalsen i infrastrukturen. Grupper og seksjoner i avdelingen som har oppgaver knyttet til drift og problemløsning kan også ha stor nytte av denne oversikten. De ønsker også å forbedre service i tråd med internasjonale standarder. Ved bruk av ITIL kan dette profesjonaliseres, slik at de på tross av økende kompleksitet, kan ha høye krav til driftsstabilitet, og gi gode servicegarantier.

Nåværende prosesser i IT-driftsseksjonen anses for ikke å være profesjonelle nok. Dokumentasjon av etatens IT infrastruktur består av mange løse papirer og filer her og der, og oversikt over program- og maskinvare kan i stor grad betraktes som fragmentert.

Fordi løsningen ikke er helhetlig, har man ikke god nok oversikt over elementer som programvare og dennes "release", versjonsnummer, oppdateringstidspunkter, etc. Som det ble beskrevet i avsnitt 2.1. om hvorfor IKT-administrasjon er viktig, ser man at SKD opplever de samme problemene som mange andre bedrifter som er avhengige av IT-systemer for å kunne fungere. For å komme fra en situasjon med liten eller ingen oversikt, til en situasjon hvor man har mye bedre kontroll med IT-ressursene, er det essensielt å skape et miljø for god kontroll av infrastruktur.

I ITIL kan man gjøre dette ved å bygge opp en "Configuration Management Data Base" (CMDB). Dermed kan man få tilgang til nødvendige data om IT-infrastrukturen ved å gjøre enkle spørringer mot en database. Oppdateringen av databasen må følges opp hver gang det foretas endringer i infrastrukturen, det være seg innen programvare, maskinvare eller annet.

I forbindelse med MIT prosjektet, prosjekt "Samlet drifts- og overvåkningsløsning", vil man gjennom å evaluere eksisterende prosesser etter ITIL rammeverket for Service Management, kunne identifisere mulige forbedringer til eksisterende arbeidsmåter, prosesser, dokumentasjon, roller og arbeidsfordeling. Dette vil i sin tur kunne brukes til å utvikle og endre prosesser og rutiner i en retning som tilfredsstiller nye krav til disse.

Ved evaluering av eksisterende verktøyportefølje, søker man å sette fokus på hvordan verktøyene skal integreres og tilpasses, samt identifisere nødvendige nyanskaffelser.



Rutiner skal utvikles og tilpasses for hensiktsmessig forvaltning av prosesser og verktøy. I forbindelse med de grepene man tar, vil man sørge for at nøkkelpersonell får nødvendig opplæring, slik at de prosjektene blir gjennomført i en atmosfære av informasjon, motivasjon, og kommunikasjon. Dette skal lede fram til at man, ifølge ITIL sitt modenhetsrammeverk, skal nå et høyere modenhetsnivå.

#### **4.5.2 Hvilke ITIL deler skal implementeres i SKD?**

Service Support og Service Delivery utgjør kjernen (eller hjertet) i ITIL rammeverket, og de fleste bedrifter velger å implementere disse komponentene når de bestemmer seg for å benytte dette rammeverket. MIT prosjektet har valgt å fokusere på "Availability Management" og "Service Level Management" fra "Service Delivery", og hendelsesadministrasjon, problemadministrasjon, endringsadministrasjon og konfigurasjonsadministrasjon fra "Service Support".

"Service Level Management" brukes tidlig i MIT prosjektet for å definere tjenestekvalitet. Bedrifter bruker "Service Level Agreement (SLA)" som en avtale med sine kunder om hva inneholdte i en tjeneste er, hvordan og på hvilket nivå tjenesten blir støttet.

"Availability Management" involveres i design, implementasjon og måling av tjenester. Den skal se om tjenestelevering stemmer overens med avtalen. Hvis en feil oppstår, skal man gjennom "Availability Management" forstå grunnen og hvor lang tid det vil ta for å gjenopprette tjenesten. Målinger og rapporter fra "Availability Management" forsikrer aktørene om at tjenesten tilfredsstillende kravene nedfelt i SLA.

"Service Support" velger de fleste organisasjonene å implementere av hovedsakelig to grunner:

- Prosessene i ITIL styrer jobben hver dag ved å ha oversikt over organisasjonens infrastruktur og hvordan de forskjellige konfigurasjonselementer (CI) knytter seg til hverandre i CMDB'en. Med "Service Support" kan problemer og forandringer i infrastrukturen systematisk håndteres. Alt i henhold til "Service Support" rammeverket.
- CMDB kan gi også informasjon om hva slag tjenester organisasjonen tilbyr sine kunder.

#### **4.5.3 Hvorfor velger SKD å implementere disse komponentene?**

Siden "Modernisering av IT-drift" (MIT) er et prosjekt som har som mål å modernisere etatens infrastruktur, og Service Desk funksjonen er blitt valgt, er det naturlig å benytte "Service Support".

Service Support har 5 prosesser, men etter samtaler med IT-teknisk avdeling, har SKD valgt å fokusere arbeidet på hendelsesadministrasjon, problemadministrasjon, endringsadministrasjon og konfigurasjonsadministrasjon. Disse prosessene skal støtte Service Desk-funksjonen.

Hvis vi tar utgangspunkt i en hendelse som blir registrert i hendelsesadministrasjonen trenger vi problemadministrasjonen for å sjekke om det er et gammelt problem, om det er registrert noen løsning for det, eller om det er en duplikathendelse.

Hvis det siste er tilfellet, er det nødvendig å varsle problemadministrasjonsprosessen, slik at

dette problemet kan få en skikkelig løsning. Det er ikke nok med en ”workaround” pga. at mange duplikathendelser kan ha oppstått igjen.

problemadministrasjon og endringsadministrasjon kommuniserer med hverandre for å utveksle informasjon om hvilke problem som ”Change Management” skal håndtere.

Alle prosessene er i kontakt med konfigurasjonsadministrasjon for å finne ut hvilke konfigurasjonsenheter (CI) som er involvert i hver enkelt hendelse, hvert problem, eller hver forandring. En slik CI kan være f.eks. brukere, maskinvare eller programvare.

I første omgang skal SKD se bort fra ”Release Management” prosessen for enkelhets skyld. Siden SKD ikke har brukt ITIL rammeverket før, har driftsarbeid stor sett vært utført manuelt. Driftspersonal bruker Remedy ARS systemet for å registrere og forfølge hendelser. De har imidlertid ikke benyttet noe som likner på en CMDB før, så hvis man benytter også ”Release Management”, dvs. at man må bryte ned en CI til flere versjoner, må mer registrering og flere relasjoner mellom enheter håndteres. Dette blir vanskeligere å både implementere og bruke i første omgang, derfor velger altså SKD å se bort fra dette.

SKD har ikke implementert ”Service Level Management”(SLM) før, men vil nå bygge opp Service Level Agreement (SLA) slik at de kan tilby gode tjenester til kunder/brukere. SLA er veldefinert tjenester med hva og hvordan tjenestene skal leveres, til hvem, når, på hvilken måte skal den oppfølges, rapporteres. [29]

De vil også bruke SLM i kundeperspektivet slik at de kan forhandle bedre med ulike leverandører når de skal kjøpe inn tjenester.

Siden Skatteetaten er en stor organisasjon blir prosjektets omfang også stort. SKD velger å bruke også ”Availability Management”, for å sikre at dette prosjektet har mulighet til å bli gjennomført i henhold til SLA.

## **4.6 Oppsummering**

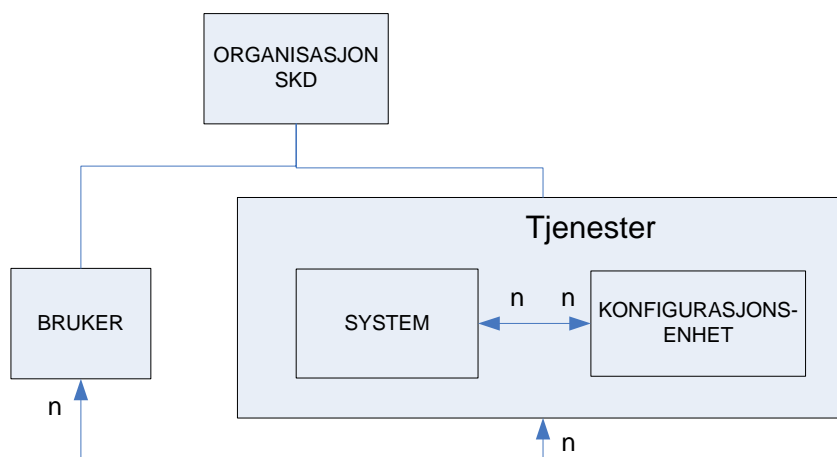
Kapitelet har skapt et bilde av Skatteetaten. Hva er etatens funksjon, deres tekniske bakgrunn samt hvordan driftsavdeling fungerer og deres arbeidsmåte. Driften skal forbedres for å bli profesjonelle mot prosessorientering, gjennom prosjekt ”Modernisering av IT-drift i Skatteetaten”. Vi har gjennom kapitelet sett på prosjektets formål.

## 5 Etablering av CMDB i SKD

Her beskrives valg av prosesser, tilpasning til organisasjon(SKD), valg av attributter til prosessene, konfigurasjonsenheter og deres nivå bestemmes, helt fram til et forslag til en relasjonsdatabase for deler av CMDB.

### 5.1 Beskrivelse av aktuelle ITIL-prosesser som skal implementeres i SKD

Vi kan skissere relasjonen mellom brukere og tjenester i Skattedirektorat slik: En komplisert tjeneste kan bestå av flere enkeltsystemer, for eksempel så består merverdiavgiftstjenesten av systemer som behandler regnskap, kunderegistrering, og kommunisering mellom klient og server for å transportere data. Hvert enkelt system trenger flere konfigurasjonsenheter. Dette forholdet har imidlertid to sider, ettersom en konfigurasjonsenhet kan brukes av flere tjenester. Det er samme relasjon mellom brukere og tjenester. Figur 13 nedenfor illustrerer dette forholdet:



Figur 13: Relasjon mellom tjenester og brukere

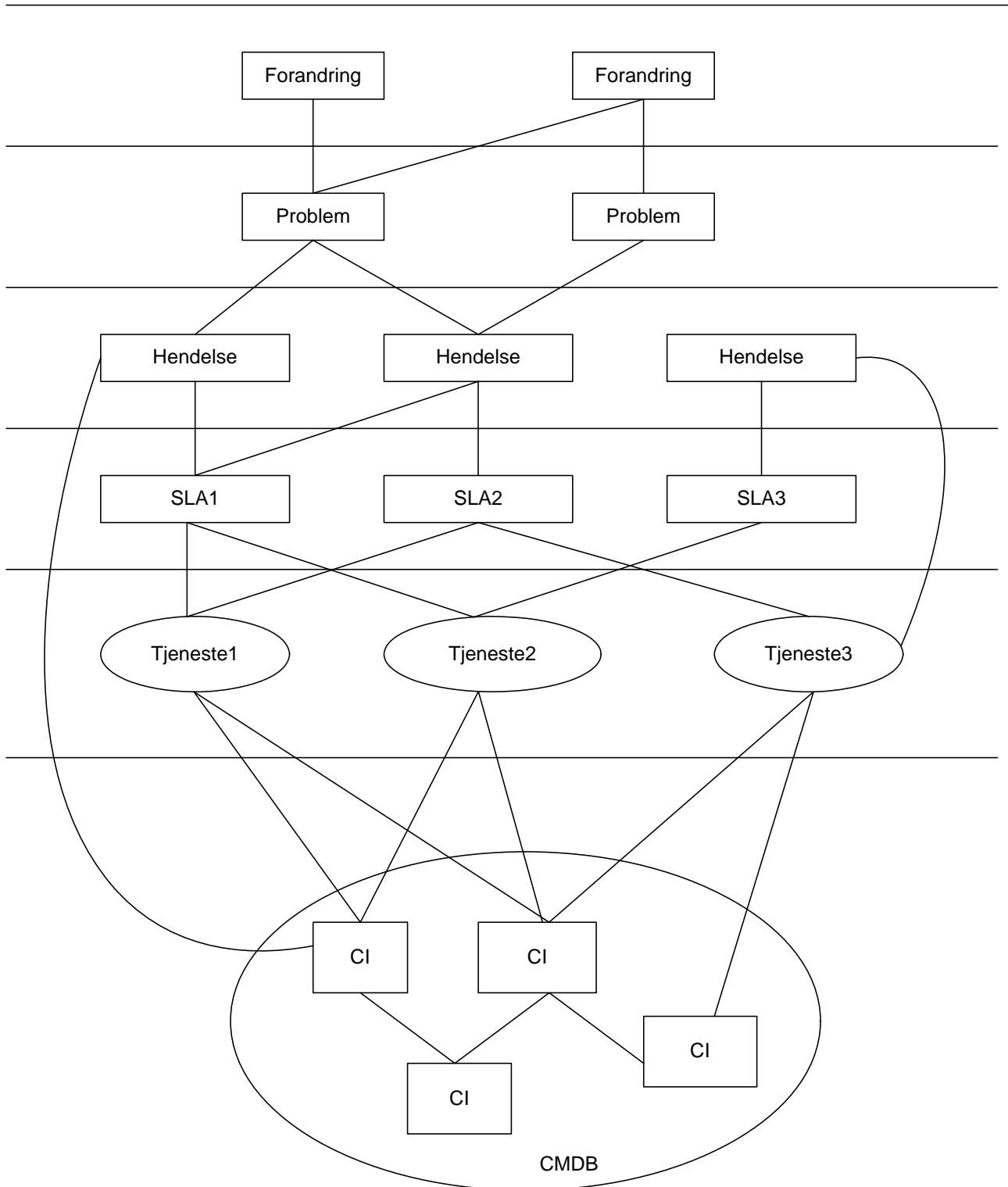
Når en hendelse oppstår, har driftsenheten behov for å bl.a. vite hvilke tjenester og tilhørende komponenter som er involvert, hvor mange og hvilke brukere som er berørt etc. for å kunne behandle hendelsen på en effektiv måte.

#### 5.1.1 Generelt

Som vi har sett av forrige figur (Figur 13), kan en tjeneste bruke flere konfigurasjonsenheter, og en enhet kan brukes av flere tjenester.

Generelt, kan tjenesten ha flere SLA, og omvendt. Organisasjonen kan også definere det slik at en tjeneste skal ha en eller flere SLA, men en SLA kan bare knyttes til én tjeneste. En hendelse kan imidlertid opprinne fra flere tjenester og SLA'er, i tillegg til andre konfigurasjonsenheter (CI), f.eks. maskin- eller programvare. Når det gjelder forholdet mellom hendelse og problem, kan én hendelse knyttes til flere problem og vice versa. I

tillegg, som vi også ser av figuren nedenfor, kan én forandring være resultat av ett eller flere problemer. (Figur 14)



Figur 14: Logisk relasjon mellom prosessene i Service Support

Følgende avsnitt beskriver hvordan en hendelse blir behandlet gjennom ulike prosesser.

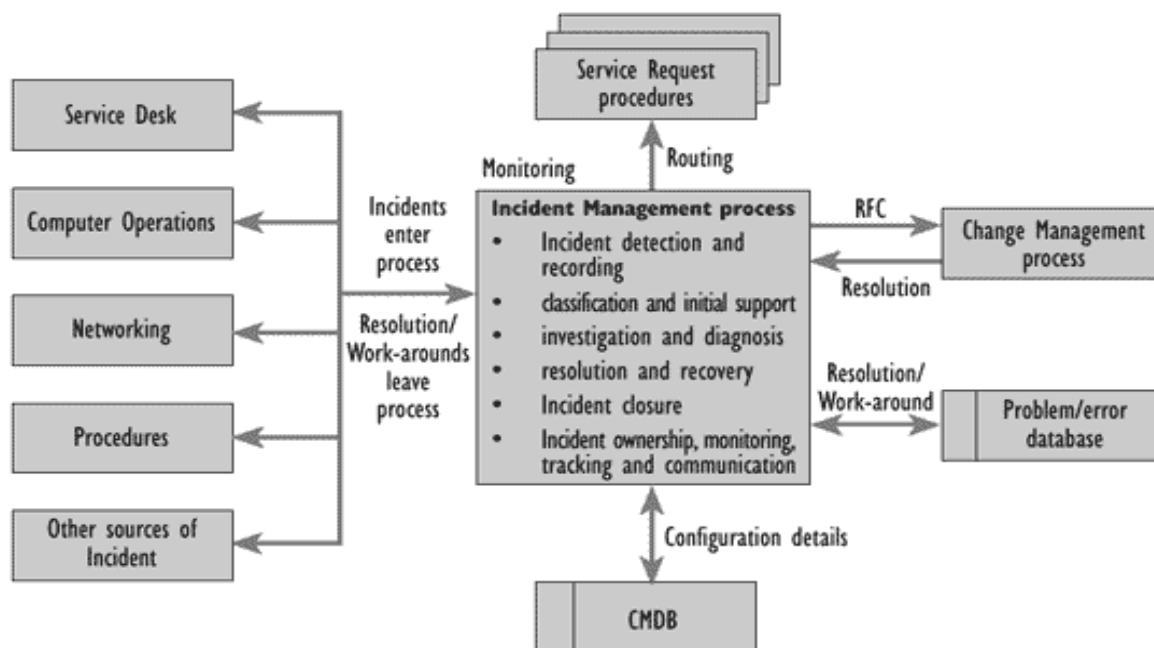
Når en bruker melder en hendelse og denne blir registrert i prosessen hendelsesadministrasjon, skal de nødvendige attributter registreres. Hvis det er en enkel hendelse å håndtere, løses den med en gang i førstelinje-støtte. Hvis hendelsen er av mer komplisert art, skal den løses etter en bestemt prosessinstruksjon for å gjenopprette tjenester. Hendelsen blir fulgt helt til den er håndtert eller såkalt lukket.

Hvis samme eller lignende hendelser oppstår om og om igjen, kan det bety at årsaken til problemet ikke har blitt funnet. Fordi hendelsesadministrasjonens mål er å gjenopprette tjenester så fort som mulig, kan en midlertidig løsning være formålstjenlig. For eksempel i form av en "workaround", slik at man unngår at hendelsen oppstår. I det korte bildet er det viktig å gjenopprette normalt tjenestenivå. For å hindre at samme hendelse oppstår igjen, overlates ansvaret til prosessen problemadministrasjon. Her skal saken undersøkes for å finne underliggende årsak. Problemadministrasjonen sørger for å løse problemet om mulig, eller fremmer en forespørsel om endring (RFC), og informerer om dette til hendelsesadministrasjonen, aktuell ledelse og brukere.

Når endringsadministrasjonen mottar en RFC, skal en endringskontrollgruppe ("Change Advisory Board", CAB) med aktuelle personer utnevnes. De skal vurdere om en RFC virkelig er nødvendig utfra organisasjonens situasjon og behov, i tillegg til i ettetid å kontrollere om endringen som er gjort har taklet problemet. Prosessen har ansvar for også å etablere en overordnet målsetning for endringsfunksjoner.

Alle ovennevnte prosesser trenger en CMDB for å utføre rutinen sin, dvs. finne ut informasjon om berørte konfigurasjonseenheter og -brukere, eventuelt oppdatere informasjonen i CMDB.

Figur 15 illustrerer denne arbeidsflyten for en hendelse mellom prosessene.



Figur 15: Incident Management Prosess (kilde: [18])

## 5.1.2 Hendelsesadministrasjon

Brukerkontakt ("Service Desk"), eller Brukerstøtte ("Help Desk") tar imot hendelse fra kunde(r)/ bruker(e) gjennom email eller telefon, eller fra overvåkningsverktøy og registrerer hendelsens attributter.

Hvis det er en relativt enkel hendelse som dokumentasjonsforespørsel eller problem med passord, skal personen som tar imot beskjeden løse problemet umiddelbart.

Hvis det er større problemet skal det klassifiseres etter definerte kategorier. I denne fasen skal man kontakte problemadministrasjonen for å sjekke om det er et gammelt problem, om det er registrert noen løsning for det, eller om det er en duplikathendelse. Hvis det er et gammelt problem og det finnes en løsning på det, kan løsningen hentes og brukes. Hvis det er en duplikathendelse og denne har oppstått et bestemt antall ganger, er det nødvendig å varsle problemadministrasjonen, slik at problemet kan få en skikkelig løsning. En endringsanmodning (RFC) kan være en slags midlertidig løsning for hendelsen.

Ellers skal det undersøkes mer om konsekvenser av hendelsen ved hjelp av CMDB. For eksempel hvor mange brukere som er berørt, hvilken maskinvare og/eller programvare som er involvert. Dette blir igjen brukt for å tildele prioritet.

Deretter diagnostiseres hendelsen, her kommer CMDB til benyttelse igjen. CMDB skal vise hvilke konfigurasjonenheter som er involverte i hendelsen, det besluttes hvilken aksjon som skal iverksettes for at systemet skal kunne gjenopprettes. Når berørte tjenester fungerer igjen, kan hendelsen lukkes ved å registrere hvor lang tid det tok å løse problemet (løsnings\_tid), når hendelse er løst, og når den er lukket (lukket\_tid), samt registrere hvor lenge hendelsen okkuperte systemet (okkupert\_tid).

Så blir eierskap bestemt og brukerstøtte skal kommunisere og følge opp brukere.

## 5.1.3 Problemadministrasjon

Når hendelsesadministrasjonen har definert problemet og gitt beskjed til problemadministrasjon, skal det startes en undersøkelse for å finne mulig skjult årsak. Problemadministrasjon skal analysere hendelsen, bruke CMDB og dataene fra problemregistrert for å identifisere problemet til fulle.

Det er forskjellig behandlingsmåte for reaktive og proaktive problemer. Hvis et problem allerede har oppstått, er det en reaktiv handling å ta itu med det. Da skal man først identifisere med tanke på hvilke tjenester og komponenter som er involverte, hvilke som er berørt, og registrere dette. Deretter klassifiseres det etter definerte kategorier (f.eks. maskinvare, programvare, dokumentasjon/opplæring etc.) for å finne ut hvilke mulige konsekvenser det kan medføre for organisasjonen, om det haster å reparere, og hvilken prioritet det bør ha. Nå kan man utføre en undersøkelse og diagnostisere underliggende årsak. Hvis diagnosen viser at årsaken ligger i en prosedyre, for eksempel registrere feil informasjon, må man passe på å oppdatere kjent\_feil i databasen manuelt, hvis det er feil i konfigurasjonsenhet, skal problemets status endres til kjent\_feil, og herfra tar avdelingen som har med feilkontroll/-administrasjon over ansvaret.

Feiladministrasjon dekker prosessene som er involvert i korleksjon av kjente feil. Målet er å unngå at kjente feil har en mer eller mindre jevn frekvens. Endringer foretas i IT

infrastrukturen slik at man fjerner den kjente feilen for godt, samtidig som man prøver å forutse om nye hendelser kan komme til å oppstå som følge av endringene. Feiladministrasjonen skal gjennomgå disse aktiviteter: identifikasjon og registrering for å finne defekte CI. Når problemet er identifisert, skal det gjennomføres vurdering av løsningsmuligheter. Hvis det er nødvendig, etableres en RFC. Når man har funnet en passelig løsning for den kjente feilen, skal denne sendes til endringsadministrasjonen. Selve forandringen i konfigurasjonen, er endringsadministrasjonens ansvar. Det kan være aktuelt å rapportere til leverandør hvis det er en tredjepart sitt produkt. Når forandringen er vellykket, skal problemadministrasjonen endre status for den kjente feilen (eventuelt slette den fra databasen). Når en feil er løst skal det registreres og hendelsen lukkes.

Proaktiv problemadministrasjon er til for at man skal søke å unngå at problemer oppstår. Trendanalyser er en viktig del av bakgrunnsinformasjonen. Analysene etableres ved å gå igjennom rapporter fra hendelses- og problemadministrasjonen. Hvis man på bakgrunn av disse analysene ser at det er svakheter i infrastrukturen, setter man i gang forebyggende tiltak som har som mål å unngå at trender går over i problemer. Et forebyggende tiltak er proaktivt i den forstand at det settes i kraft for å unngå at potensielle hendelser/problemer materialiserer seg.

Ved administrasjon av store problemer, er det formålstjenlig å foreta en vurdering i etterkant av hva som er vellykket, hva som kan forbedres, og hva som bør unngås.

#### **5.1.4 Endringsadministrasjon**

Endringsadministrasjonen bestemmer seg for en måte å implementere prosedyrene på, som passer med organisasjonens størrelse og struktur. I noen tilfeller, litt avhengig av organisasjonens størrelse, kan det, som nevnt tidligere, være formålstjenlig å kombinere endrings- og konfigurasjonsadministrasjonen, ettersom disse er nært knyttet opp til hverandre.

Det er flere ulike metrikker som skal følge endringsanmodninger (RFC). De første er konsekvens og viktighet. Man evaluerer hvilke konsekvenser problemet som RFC'en skal løse har for organisasjonen, og om disse konsekvensene vil materialisere seg i nær eller fjern fremtid. På bakgrunn av dette angir man prioritet til RFC'en. Det neste steget er å kategorisere hendelsen. Denne kategoriseringen tar utgangspunkt i hvor mye ressurser som trengs for å gjennomføre endringen. Det er nødvendig med en logging av endringsanmodningene, slik at man kan følge prosessen, og evaluere den, samt se hvilke problemer som er knyttet til hvilke anmodninger.

Hvem er det så som kan sende inn endringsanmodninger? I utgangspunktet bør enhver bruker i organisasjonen oppmuntres til å komme med en anmodning om endring. Denne kan være basert på f.eks. et tilfelle av dårlig brukervennlighet i et eller annet program. På denne måten sikrer man deltakelse fra hele organisasjonen, og kan dermed oppnå løsninger som kan føre til ressurs- og kostnadsbesparelser på bred front. Imidlertid kan det være nødvendig med en viss filtrering av henvendelsene, slik at man ikke blir opptatt med alle mulige små problemer som kan virke negativt inn på forholdet mellom kost og nytte. Denne filtreringen kan ordnes slik at det er noen som har hovedansvaret for å godkjenne at anmodningen blir sendt. Filtrering vil også forekomme underveis i livsløpet til en endringsanmodning.

Organisasjoner oppmuntres til å sette opp et endringsstyre (CAB). Dette styret skal bl.a. gjøre tilgjengelig nye anmodninger for styrets medlemmer, vurdere anmodninger som styrets

medlemmer har gått igjennom, foretatte endringer, endringer som ikke er gjennomført, og endringer som har blitt foretatt gjennom hendelses-, problem- og endringsadministrasjon uten at CAB har vært involvert.

Etter hvert blir en endringsanmodning akseptert, denne aksepteringen kan gjøres enten gjennom endringsadministrasjonen, tjenesteadministrasjonen (service manager), eller den blir godkjent av en gruppe eller en person som har den nødvendige avgjørelsesmyndigheten. Når dette er skjedd, bestemmes tidspunkt for når endringen skal foretas. Det å foreta selve endringen, er ikke alltid like enkelt. Det kan selvfølgelig være så enkelt som f.eks. å bytte ut en harddisk i et eller annet RAID-system, men oftest er det mer komplisert. Under endringer kan essensielle tjenester bli berørt på en slik måte at det skader organisasjonen økonomisk og/eller ressursmessig. Derfor er det ofte nødvendig å foreta endringene på et referansemiljø. Referansemiljøet er et miljø som ligner på det miljøet man skal foreta endringen på, men har mindre størrelse. Deretter kjøres nødvendige tester, med metrikker som bl.a. sikkerhet, funksjonalitet, tilgjengelighet, og pålitelighet. Når systemet basert på den nye endringen fungerer til fulle, blir den implementert i en operativ sfære.

Etter at infrastrukturen har operert med endringene en stund, gjøres det en evaluering av endringene. Nå kan man se om endringene har vært vellykkede, i forhold til interne og/eller eksterne brukere/kunder, og i forhold til budsjettering ressurs-, tids-, og kostnadsmessig. Det kan hende at det gjennom denne prosessen identifiseres "bivirkninger" av endringen(e) i forhold til noen av testmetrikkene nevnt ovenfor, eller andre, eller at det fremkommer behov for ytterligere endringer.

## **5.2 Overordnet beskrivelse av fremgangsmåte ved bygging av CMDB**

ITIL er et åpent rammeverk, så det må tilpasses organisasjonen etter behov. I dette avsnittet viser jeg hvordan rammeverket tilpasses prosjektet "Modernisering av IT-infrastruktur" i SKD.

Først prøver jeg å tilpasse prosessene til SKD sin kultur og arbeidsmåte, deretter foreslår jeg etablering av en konfigurasjonsdatabase (CMDB) som prosessene bruker.

### **5.2.1 Prosesstilpasning - Etablere aktiviteter og attributter**

1. Kartlegge formål, oppgaver og aktivitet: For å forstå prosessene i SKD drift og hvordan disse kan beskrives i henhold til rammeverket, må de aktuelle ITIL prosessene gjennomgås (hendelsesadministrasjon, problemadministrasjon, endringsadministrasjon). For å gi en oversikt over hver av disse prosessene, må blant annet hensikt, formål, oppgave, input, output, relasjon til andre prosesser og prosessaktiviteter få en generell beskrivelse.
2. Kartlegge rutiner, prosesser og skissere arbeidsinstruksjoner: Her har jeg fulgt rammeverket ganske slavisk for å skissere en prosedyre. Ut fra oversikten over arbeidsmåten i SKD, lager jeg en overordnet tabell med instruksjoner for hver prosess.
3. Bestemme attributter for prosessene: ut fra informasjon prosesser trengs, kan attributter til prosessen bestemmes.



## 5.2.2 Bygge opp CMDB

De konfigurasjonsenheter jeg har tegnet er logiske enheter, men det er de fysiske enhetene som er mest interessante når man vil undersøke hvilke konsekvenser hendelser og problemer kan føre til. Med andre ord, logiske enheter er metadata mens fysiske enheter er dataene i database. Disse fysiske enhetene skal registreres i databasen. Hver enhet skal ha et unikt identifikasjonsnummer (CI\_id). Ellers skal de ha attributter som er naturlige for hver enkelt enhet.

Som følge av ønsker fra prosjektmedarbeider, og masteroppgavens tids- og omfangsramme, skal jeg modellere en relasjonsdatabase av en delmengde av CMDB'en, nemlig E-mail tjenesten. Resultatet/produktet skal presenteres i form av en eller flere NIAM-modeller som viser relasjoner mellom begreper/tabeller. Det skal også produseres grupperte tabeller med tilhørende attributter, primærnøkler og fremmednøkler, i tillegg til eksempler på utvalgte SQL-spørringer.

## 5.3 Etablering av aktivitetsattributter og relasjoner for prosesser

Hendelsesadministrasjonen er prosessen jeg begynner med, videre følger problemadministrasjon og endringsadministrasjon, i denne rekkefølge. Jeg velger å se bort fra versjonsadministrasjon ("Release Management") i første omgang, for enkelhets skyld. Alle prosessene har et forhold til konfigurasjonsadministrasjonen, og denne kan defineres parallelt med disse, fordi konfigurasjonsadministrasjonens CMDB er nødvendig for alle prosessene.

Prosessene bør knyttes sterkt til hverandre, for at de skal fungere optimalt. Attributter skal registreres i samme nivå i hele CMDB, både i konfigurasjonsenheter og i prosessene. For eksempel bør maskinvare registrert i CMDB ved maskinnavn, også bli registrert med maskinnavn i hendelsesadministrasjonen, men man behøver ikke spesifisere flere detaljer. Melding fra overvåkningsverktøy ved hendelser, behøver altså ikke inneholde flere detaljer enn maskinnavn.

### 5.3.1 Kartlegge formål, oppgave og aktivitet

Gjennom samtaler med personalet i SKD, og studier av referanser, har jeg dannet meg et grunnleggende bilde av hvilken informasjon SKD er ute etter i prosessene. Dette er formål, oppgave, input, output, hvordan prosessene knytter seg til hverandre og ikke minst prosess-aktiviteter. Disse er i detalj beskrevet i prosessoversikten, og finnes vedlagt (Vedlegg "Oversikt over hendelsesadministrasjon", "Oversikt over problemadministrasjon", "Oversikt over endringsadministrasjon", "Oversikt over konfigurasjonsadministrasjon").

### 5.3.2 Kartlegge rutiner og skissere arbeidsinstruksjoner

I tillegg til å beskrive rent tekstlig, som jeg har fortalt om i avsnittet ovenfor, plasserer jeg nødvendig informasjon i en tabell, med kolonner som viser: overordnet aktivitet, detaljert aktivitet, relasjon med andre prosesser, input, og output, i kronologisk rekkefølge. Tabellen fokuserer på attributtregistrering for hver detaljert aktivitet, fordi målet med oppgaven min er å skissere en CMDB. Siden SKD bruker Remedy for å registrere hendelser, bruker jeg også Remedy grensesnittet for å bestemme prosessattributter (Vedlegg "Tabell for hendelsesadministrasjon", "Tabell for problemadministrasjon", "Tabell for

endringsadministrasjon”).

### **5.3.3 Bestemme prosess attributter**

Databasedesignet må ta hensyn til at det skal kunne gå an å gjøre spørringer som kan produsere en tabell over hvilke konfigurasjonsenheter som relateres til de enkelte tjenestene, og omvendt. For eksempel hvis man får en feil i tjenesten for merverdiavgift slik at dokumenter ikke lenger kan lagres, da må man kunne undersøke hvilke applikasjoner og maskinvare som denne tjenesten bruker. Derfra kan man hjelpe brukere. Dette er å finne konfigurasjonsenheter via tjenester. I motsatt fall kan man ha behov for å finne ut hvilke tjenester som blir berørt hvis en enhet endres eller byttes ut. For eksempel når en server er nede, hvilke tjenester er berørt, og hvor?

All informasjon som er nødvendig i prosessen samles i en tabell, her følger også kolonner som forklarer hva som er hensikten med hvert enkelt attributt. (Vedlegg ”Attributter for hendelsesadministrasjon”, ”Attributter for problemadministrasjon”, ”Attributter for endringsadministrasjon”).

## **5.4 Bygge opp CMDB**

For å utføre tjenesteadministrasjons-prosessene etter ITIL’s rammeverk i praksis, er det først og fremst viktig å ha en konfigurasjonsdatabase (CMDB) som er oppdatert og gyldig.

### **5.4.1 Identifisere konfigurasjonsenheter**

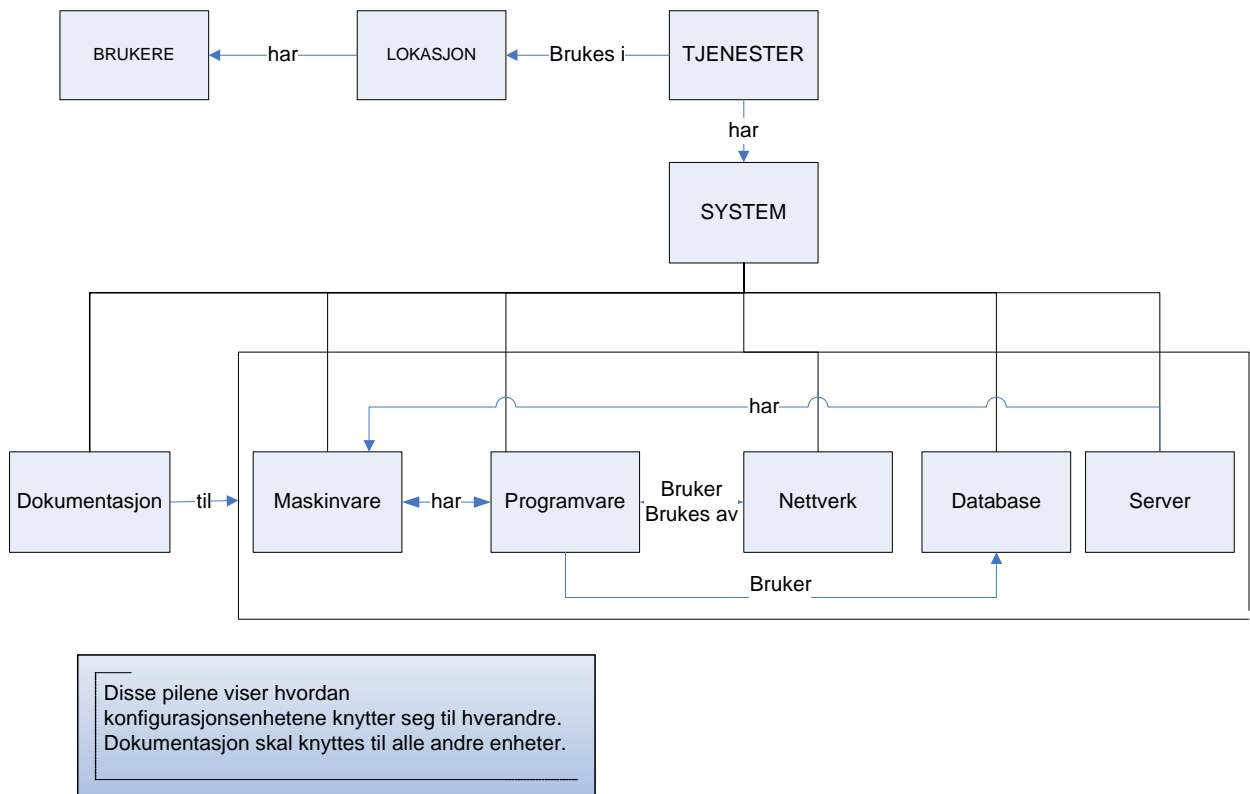
ITIL er et omfattende rammeverk, denne oppgaven konsentrerer seg som forklart hovedsakelig om å finne metadata og relasjoner for prosessene i ”Service Support”.

Konfigurasjonsenheter kan defineres etter det nivået man ønsker. En CI i SKD kan kategoriseres slik: En tjeneste kan installeres i forskjellige lokasjoner, og i hver lokasjon kan tjenester brukes av mange brukere. En komplisert tjeneste kan settes inn i et system, som trenger disse enheter for å fungere: maskinvare, programvare, nettverk, database, server og dokumentasjon (se Figur 16).

Et eksempel vi kan se på. Merverdiavgiftstjenesten er en stor tjeneste som beregner merverdiavgift (mva). Vi kan tenke oss at i tillegg til maskinvare og programvare, og omgivelser nødvendige for at tjenester skal fungere, trenger mva-tjenesten en database for å lagre nødvendig informasjon om kunder og avgifter. Den trenger et nettverk for å distribuere informasjonen. I tillegg trenger tjenesten en eller flere servere, og disse kan være database-servere og servere for nettverkskommunikasjon. Denne tjenesten krever et høyt sikkerhetsnivå for å sikre informasjonen om avgiftene. Dokumentasjon kan være SLA, brukerveiledning, eller annen dokumentasjon som følger med maskinvare og/eller programvare.

Ikke alle tjenestene i SKD trenger alle elementene omtalt over, noen tjenester trenger bare noen av disse elementene. For eksempel trenger email-tjenester ikke nødvendigvis en database for lagring av kundeopplysninger.

Figur 16 illustrerer enhetsidentifisering (identifisering av CI) og hvordan relasjonen dem imellom kan se ut.



Figur 16: System struktur

### 5.4.2 Skille mellom logiske og fysiske konfigurasjonsenheter

Driftspersonale trenger informasjon om representasjon av fysiske enheter for å håndtere hendelser i prosessene. De vil gjerne vite nøyaktig hvilke datamaskiner som er involvert i hendelsen, hvem som er berørt, og hvilket sted disse befinner seg på. De trenger altså ikke bare informasjon om at det er feil ved en PC eller server, men også hvilken, og hvor.

Til dette formålet tar vi i bruk logisk representasjon. Logiske enheter er begreper som danner utgangspunkt for tabeller i databasen, mens fysiske enheter er forekomster i tabellene.

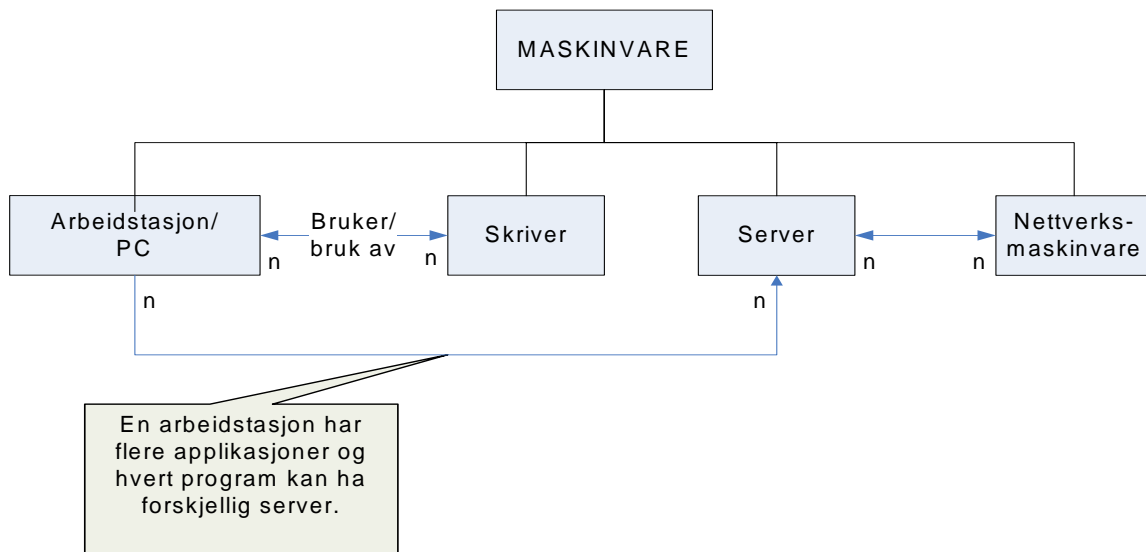
Jeg har kartlagt klassifikasjonsstruktur, med andre ord kartlagt på hvilket nivå konfigurasjonsenheter skal spesifiseres for hele CMDB. I denne oppgaven tillegges imidlertid et fullstendig forslag til relasjoner og attributter bare for enheter som er aktuelle i email-tjenesten. Denne tjenesten modelleres i NIAM med nødvendige diagrammer og gruppering av tabeller.

### 5.4.3 Tegne CI klassifikasjonsstruktur

Gjennom diskusjon med medarbeidere [30],[31], [32] i teknisk avdeling i SKD, kommer jeg til at konfigurasjonsenheter skal beskrives på følgende nivå (se figuren nedenfor). Figuren viser også relasjoner mellom enheter som trenger å modelleres.

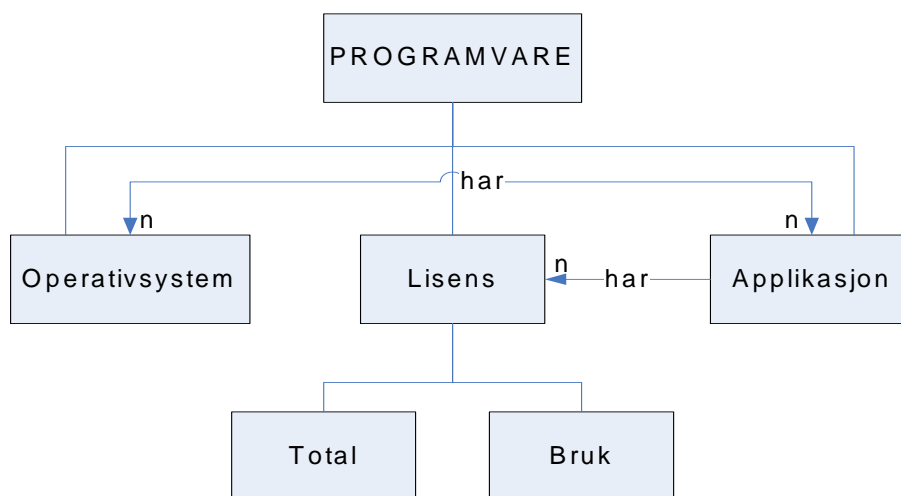
- Maskinvare skal kategoriseres følgende kategorier: Arbeidsstasjon/PC, Skrivere, Server og Nettverk. (Figur 17)
- PC'er skal ikke spesifiseres ned til RAM, hardisk, CPU etc., ettersom driftspersonale selv

kan undersøke nærmere hva som er feil i en PC.



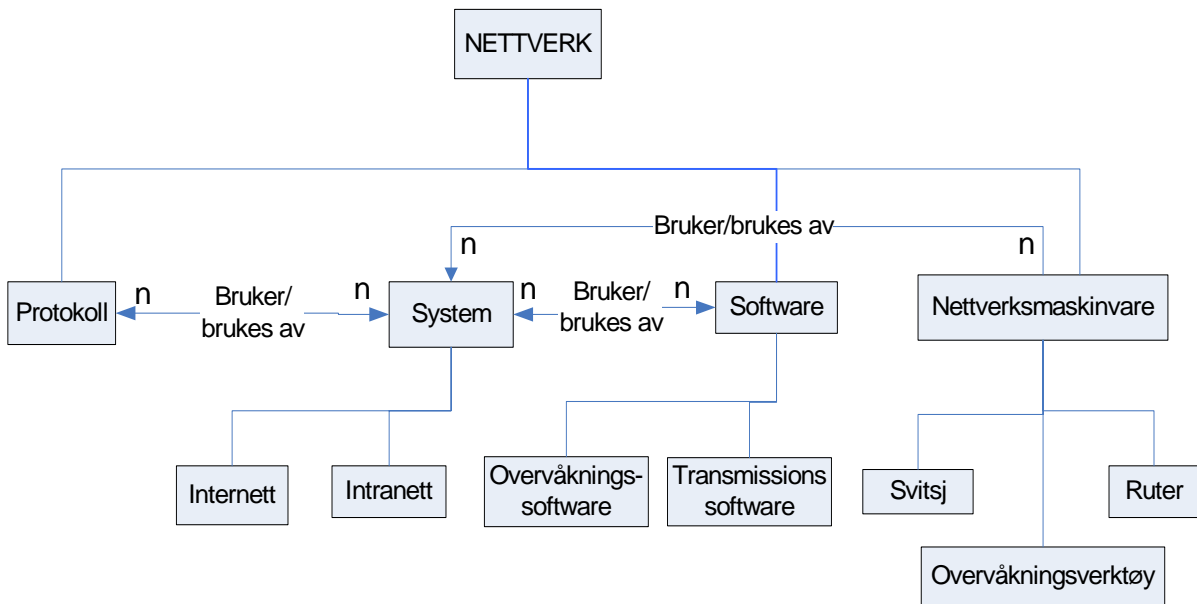
Figur 17: Maskinvarestruktur

- Programvare kan klassifiseres til operativsystem og applikasjon. Det er behov for å vite om antall lisenser som tilhører ulik programvare og hvor mange av dem som blir benyttet. (Figur 18)



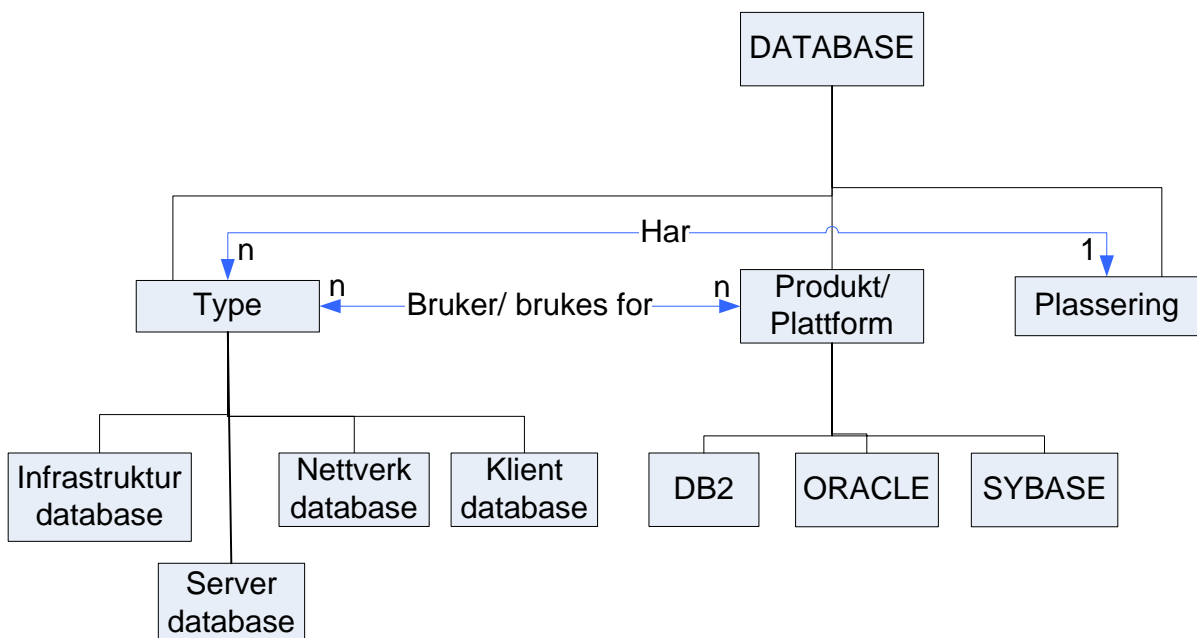
Figur 18: Programvarestruktur

- Nettverksystemer er ganske komplisert og kan spesifiseres til nivå vist i Figur 19. Gjennom sql-spørringer, kan driftspersonale få en foreløpig viten om hvor feilen(e) kan ligge. På bakgrunn av denne informasjonen, kan de bestemme hvor de skal sette inn ressurser.



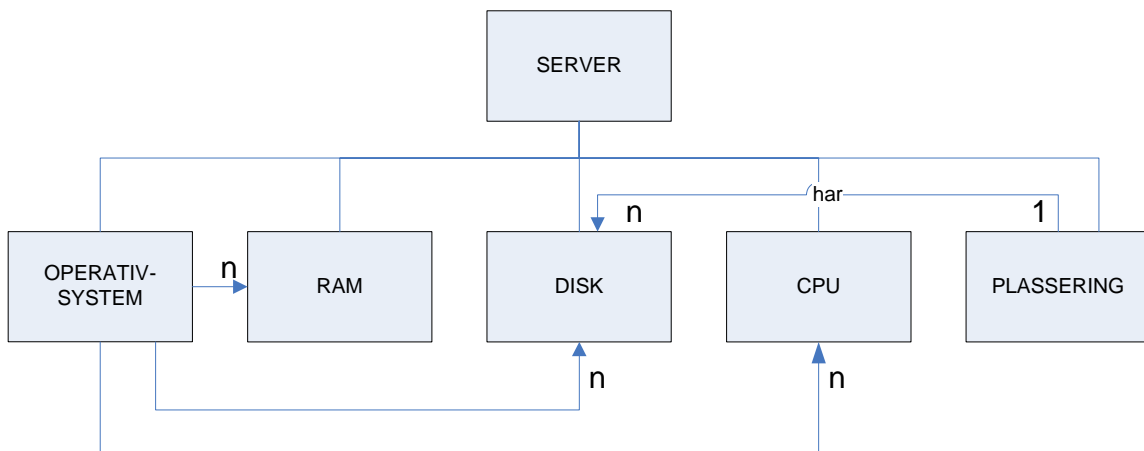
Figur 19: Nettverksstruktur

- Siden SKD har flere avdelinger på forskjellige steder, skal plassering registrere hvilke filialer de fysiske databasene hører til. (Figur 20)
- Attributtet "lokasjon" skal registrere hvilken etasje og rom databasemaskinvaren finnes i i hver filial.



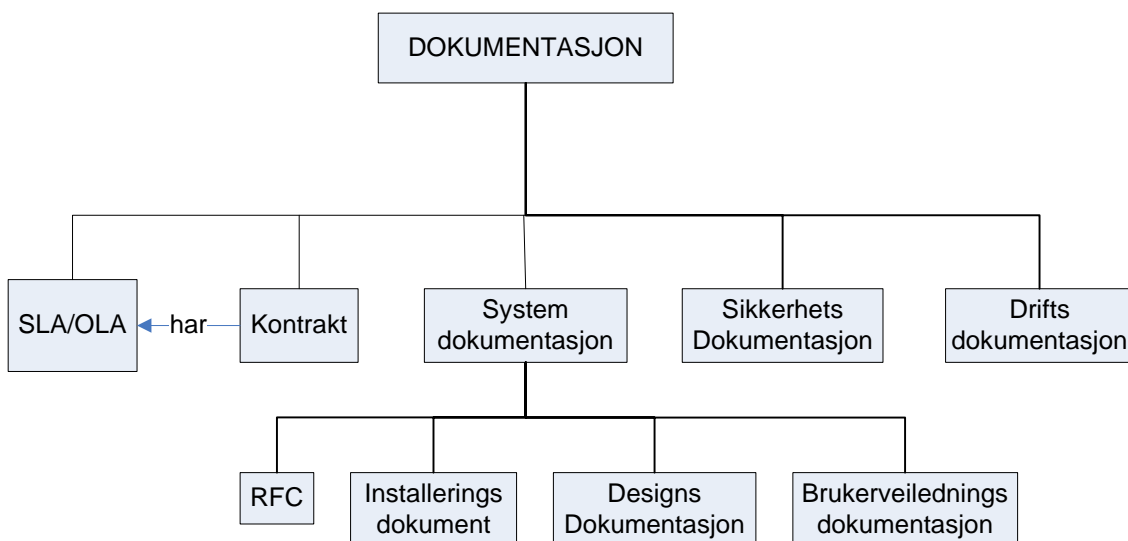
Figur 20: Databasesstruktur

- En server er en stor maskin, derfor, i motsetning til PC'er, trenger servere en oversikt over minne, harddisk, prosessor, og operativsystem. Det er i tillegg behov for å vite hvor den står plassert. (Figur 21)



Figur 21: Serverstruktur

- Dokumentasjonen kan deles opp etter kategorier som er vist i Figur 22. Det er behov for en relasjon mellom kontrakt og SLA/OLA. Ellers holder det å registrere de som ulike typer dokumentasjon i databasen.



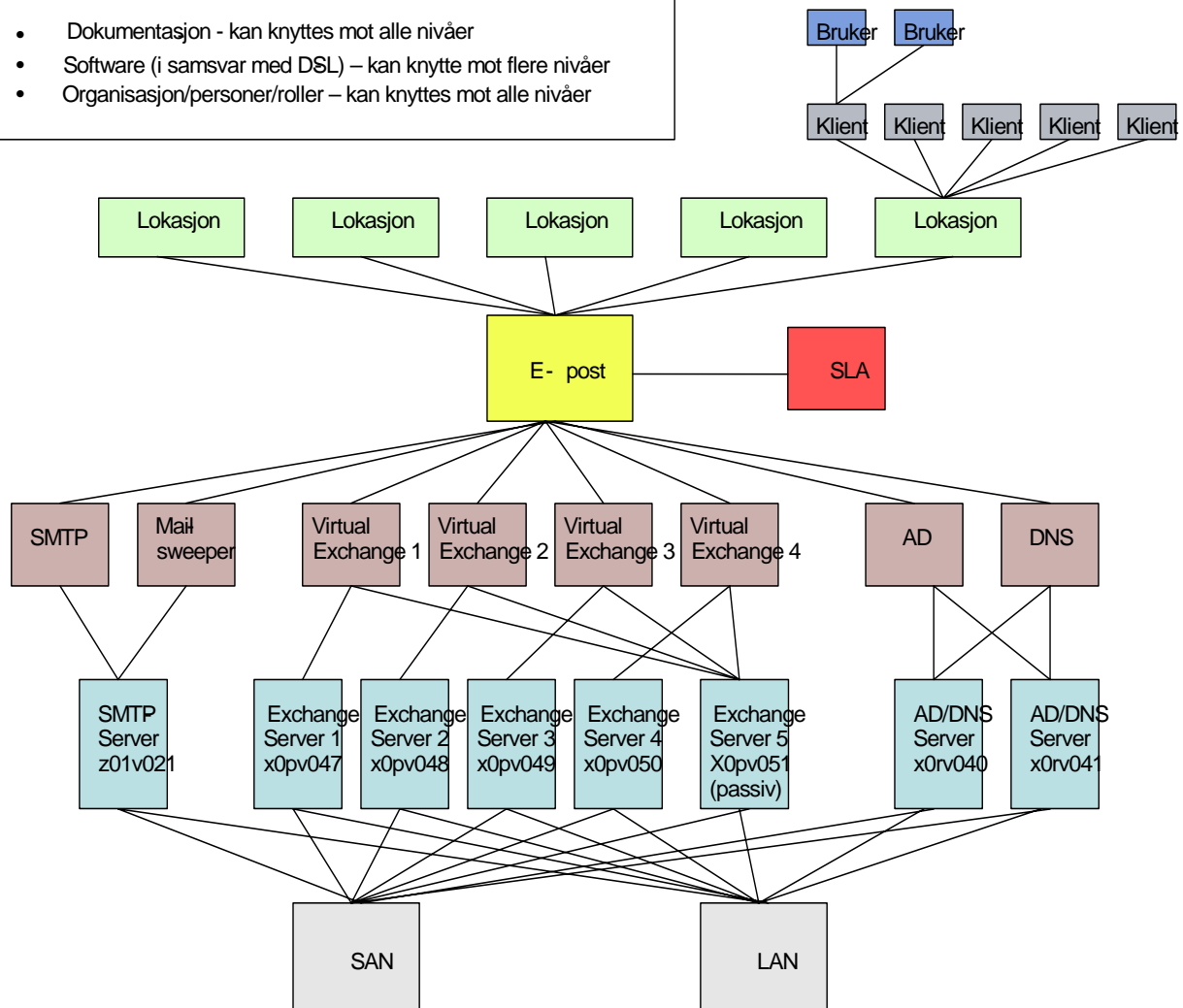
Figur 22: Dokumentasjonsstruktur

#### 5.4.4 Spesifisere CI attributter for email tjeneste

Siden hele CMDB for SKD altså er for stor i forhold til denne oppgavens omfang, har jeg valgt å skissere en database for email-tjenesten (se Figur 23). Vi skal se hvordan hendelsesadministrasjon, problemadministrasjon, og endringsadministrasjon bruker konfigurasjonsenheter for å utføre sitt arbeid.

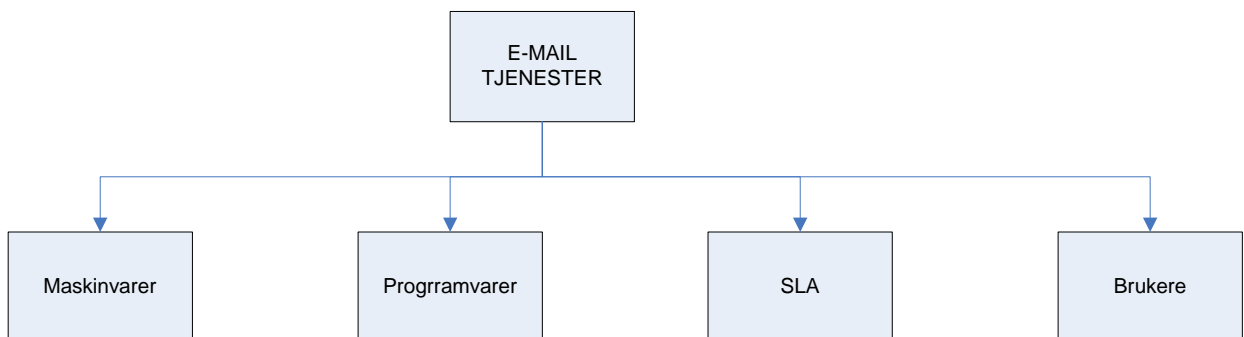
**Disse Cls er del av modellen men ikke tegnet inn:**

- Dokumentasjon - kan knyttes mot alle nivåer
- Software (i samsvar med DSL) – kan knytte mot flere nivåer
- Organisasjon/personer/roller – kan knyttes mot alle nivåer



Figur 23: Email tjenesten

Ut fra dette kan vi klassifisere email-tjenesten slik: (Figur 24)



Figur 24: Email tjeneste i klassifisering til konfigurasjon enheter

Ut fra klassifikasjonsstrukturen for konfigurasjonsenheter (punkt 5.4.3), prosessenes attributter (punkt 5.3.3) og databasemodellering, kommer jeg frem til disse begrepene med deres respektive egenskaper i forbindelse med en NIAM-modellering. Etter ”Think big – start small”-prinsippet [19], så skal ikke alle av prosessenes attributter brukes i databasen. Det er bare de attributtene som er nødvendige som brukes her, andre kan utvikles etter behov senere.

I tabellene nedenfor blir en del av attributtene forklart rett etter tabellen, foruten dette, så gjelder disse reglene:

- Fet skrift brukes for å angi en tabells primærnøkkel. Hver tabell har en identifikator som er markert med postfix ”id”, for noen av dem utgjør CI\_id og CI\_type sammen primærnøkkelen.
- Fordi andre begreper kan ha fremmednøkkel til forskjellige begreper, trenger man både CI\_id og CI\_type for å vite hvilket begrep den er ment for.
- ”Navn” er begrepsnavn, for eksempel navn i tjenestetabell betyr at man skal registrere tjenestenavn.
- ”For\_[ ]” er fremmednøkkel til [begrepet]. For eksempel ”for\_tjeneste” i SLA-tabellen er fremmednøkkel til tjeneste tabellen.
- ”Reg\_tid” er registreringstidspunktet til begrepet.

#### Tjeneste

<b>CI_id</b>	<b>CI_type</b>	navn	lokasjons_id	ansvar	status
--------------	----------------	------	--------------	--------	--------

- ”ansvar” er hovedansvar eller kontaktperson for tjenesten.

#### SLA

<b>SLA_id</b>	reg_tid	leverandør	kunde	kontaktperson	beskrivelse	for_tjeneste
---------------	---------	------------	-------	---------------	-------------	--------------

#### Person

<b>Ansattnr</b>	navn	funksjon	avdeling	rom	email	tlf	adresse
-----------------	------	----------	----------	-----	-------	-----	---------

- ”funksjon” er en persons rolle(r) i organisasjonen.

#### Lokasjon

<b>lokasjon_id</b>	lokasjons_navn	adresse	tlf	lokasjons_sjef
--------------------	----------------	---------	-----	----------------

#### Lisens

<b>lisens_id</b>	totalt	antall_brukt	for_CI
------------------	--------	--------------	--------

#### Maskinvare

<b>serialnr</b>	modellnr	type	kilde	status	leverings_dato	eier	garantiutløps_dato
-----------------	----------	------	-------	--------	----------------	------	--------------------

- ”kilde”: hvor maskinvare og programvare kjøpes fra

#### Programvare

<b>CI_id</b>	<b>CI_type</b>	modellnr	type	kilde	status	leverings_dato	eier	garantiutløps_dato
--------------	----------------	----------	------	-------	--------	----------------	------	--------------------



## Hendelse

CI_id	CI_type	mottaker	kilde	avsender	type	reg_tid	lukket_tid	prioritet	status	beskrivelse	label
-------	---------	----------	-------	----------	------	---------	------------	-----------	--------	-------------	-------

- ”label”: kort beskrivelse av hendelse. Dette skal stemme overens med label i problem-tabellen. Dette attributtet brukes for å finne ut om en ny hendelse tilhører et registrert problem, og derfra finne kjent feil og eventuelt løsningen.
- ”type”: Hvilken hendelsestype det er, altså hendelser som enten er forårsaket av programvare, maskinvare eller dokumentforespørsel.

## Status

Status	forklaring
--------	------------

## Problem

CI_id	CI_type	kilde	type	antall_hendelser	beskrivelse	reg_tid	lukket_tid	prioritet	status	label	konsekvenser	viktighet
-------	---------	-------	------	------------------	-------------	---------	------------	-----------	--------	-------	--------------	-----------

- ”type”: Se forklaring av type for relasjonen ”hendelse”, det samme gjelder for problem-relasjonen.
- ”antall\_hendelser”: Viser hvor mange hendelser som er knyttet til dette problemet.

## Løsning

løsning_id	for_CI	for_CI_type	reg_tid	type	beskrivelse
------------	--------	-------------	---------	------	-------------

- ”type” for løsning kan være endelig, RFC, eller workaround.

## Kjent feil

kjent_feil_id	reg_tid	beskrivelse	for_løsning
---------------	---------	-------------	-------------

## Forandring

CI_id	CI_type	prioritet	rapport	kostnad	konsekvenser
-------	---------	-----------	---------	---------	--------------

## RFC

RFC_id	reg_tid	kilde	beskrivelse	for_forandring
--------	---------	-------	-------------	----------------

### 5.4.5 Spesifisere CI relasjoner i NIAM modell for Email tjeneste

#### 5.4.5.1 Beskrivelse av NIAM-diagrammet [33]

Jeg velger NIAM (”Natural language Information Analysis Methodology”) modellen for å beskrive relasjonsdatabasen her, fordi NIAM er basert på naturlig språk, og den bruker grafikk for å fremstille vanlige setninger. NIAM diagrammet skal leses slik:

**Begrep:** de gjenstander/egenskaper/hendelser som tilhører samme gruppe, danner et begrep. Begreper tegnes som heltrukne sirkler/ellipser med et navn inni. Begrepet skal omformes til tabeller i databasen.

**Begrepsdannelse:** to eller flere begreper kan danne et dannelsesbegrep. F. eks ”maskinvare\_for\_tjeneste” dannet fra ”tjeneste”- og ”maskinvare”-begrepene.

Dannelsesbegreper tegnes ved å bruke heltrukne sirkler rundt rollene, og har sin primærnøkkel som en kombinasjon av primærnøkklene til de begreper den basert på.

**Underbegrep:** ikke alle begreper i heltrukne sirkel blir til tabeller, hvis begreper har bare ett attributt, og den har uendelig verdi, kan den bli en subtype av begrepet den tilhører, for eksempel begreper som "beløp" og "registrerings\_tidspunkt" er typisk slike underbegreper.

**Roller:** sammenhenger mellom begreper beskrives ved roller. For eksempel tjeneste har lisens, og lisens for tjeneste, dvs. tjeneste spiller rollen "har" overfor lisens mens lisens spiller rollen "for" for tjeneste.

**Representasjon:** hvordan begrepet representeres beskrives med et substantiv i parentes under begrepsnavnet, medfølgende et "#" tegn. Dette blir primærnøkkel til tabellen senere.

**Entydighetsskranke:** Pilen som står over en rolle viser entydighet. Dvs. verdien som står under pilen kan ikke gjentas i tabellen. F.eks. én SLA kan bare tilhøre én tjeneste, og derfor kan en SLA-verdi kun dukke opp én gang i tabellen SLA\_for\_tjeneste.

**Påkrevd:** Påkrevd symbol sier at "alle forekomster av begrepet finnes i denne rollen"

**Påkrevdkombinasjon:** i oppgaven har jeg brukt en påkrevd rollekombinasjon. Det vil si at begreper har enten den ene rollen eller den andre rollen eller begge.

**Faktatype:** begreper, roller, påkrevd, entydighetsskanker, etc. tegnes etter faktatyper, dvs. faktatyper er regler for å oversette vanlige setninger til NIAM-diagrammer. For eksempel: "én SLA må tilhøre minst én tjeneste, men én tjeneste kan ha 0 eller flere SLA" er en faktatype.

Det er mange begreper og roller, så jeg har ikke tegnet representasjoner i figuren, men henviser isteden til punkt 5.4.4- "Spesifisering av CI-attributter".

#### 5.4.5.2 *Antagelser:*

- Kategori (type) defineres av driftspersonell. I tabellene som er vedlagt, er det kun gitt eksempler på dette for å gjøre det mer lesbart/forståelig. Disse kategoriene er bare domener for attributter. Det er metadata som er mest interessant her.
- Krav til problemvarsling er definert av driftspersonalet, dvs. hvor mange ganger hver enkelt hendelse dukker opp, eller hva slag hendelse som skal håndteres i hendelsesadministrasjonen før det defineres som et problem og blir sendt til problemadministrasjonen. Krav til problemvarsling kan bestemmes av driftspersonale i arbeidsrutiner eller defineres på forhånd i SLA.
- I endringsprosessen, forenkler jeg attributter ved å registrere avgjørelsesgruppe, utførelsesgruppe, avgjørelses- og utførelsesdokument ved å registrere dem i rapport istedenfor egne attributter for dem som er forslaget fra ITIL-rammeverket.
- Se bort fra dokumentasjon for enkelhets skyld.
- Én SLA må knyttes til kun en tjeneste og hver tjeneste kan ha null til én SLA (det

kan tenkes at alle avtaler for tjenesten er samlet i et dokument).

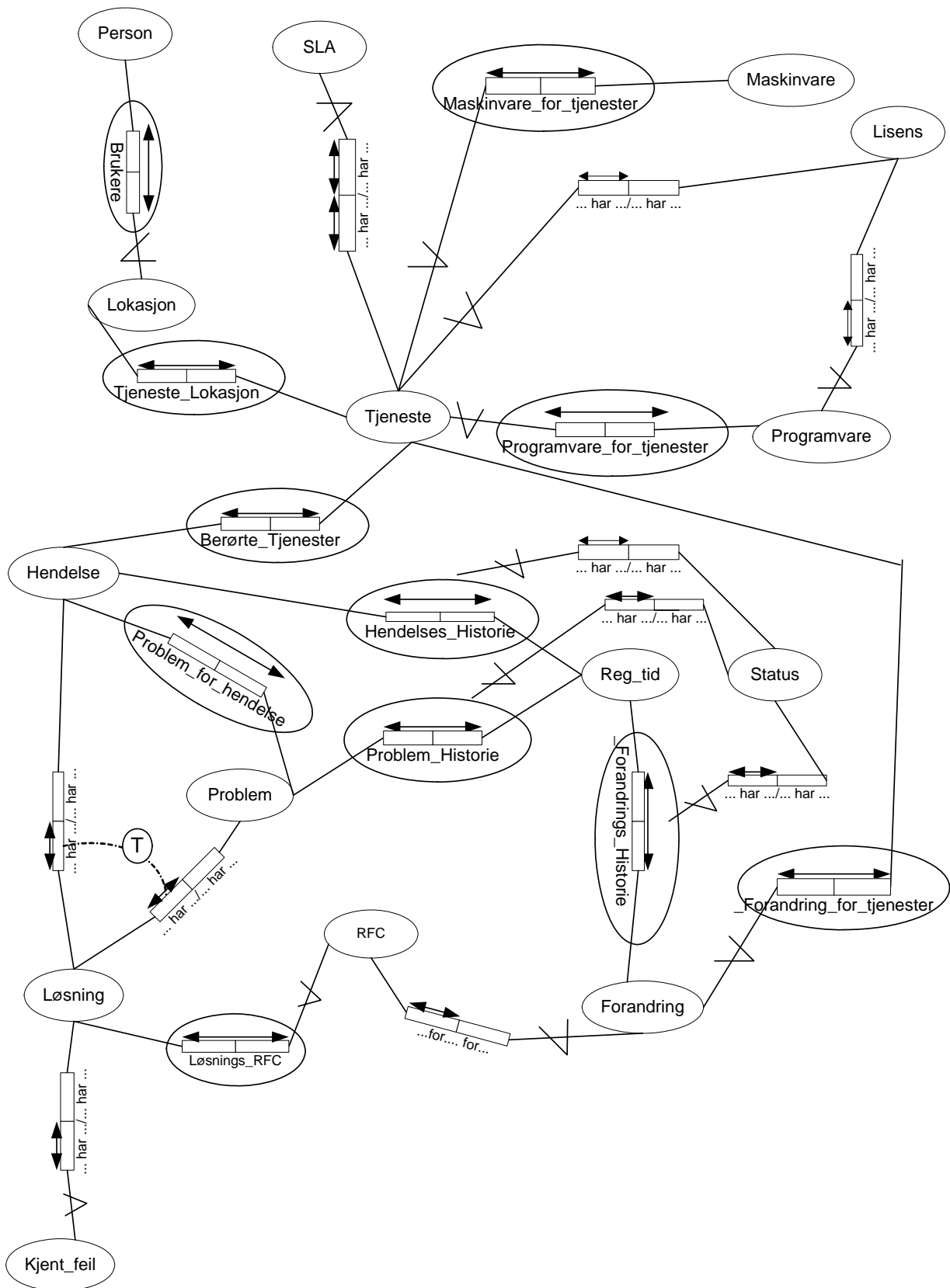
- Tjenester og programvare må ha lisens(er) og én tjeneste/programvare knyttes til én lisens.
- Flere hendelser kan knyttes til ett problem og omvendt.
- En hendelse har alltid en løsning, enten via problemadministrasjonen eller en "quick fix".
- Hvert problem kan innebære en eller flere forandringer.
- Hvert problem kan være årsak til flere RFC og én RFC kan knyttes til flere problemer.
- Én forandring kan ha flere RFC, men én RFC tilhører kun én forandring.
- Hver konfigurasjonsenhet kan brukes av flere tjenester, og én tjeneste bruker flere konfigurasjonsenheter.
- Mottaker skal stå registrert som enkeltperson i SKD, ikke som avdeling. Denne personen skal ha ansvar til saken er løst.
- Avsender er en eller flere av etatens ansatte, og all informasjon om avsender/bruker ligger i etatens Ansatt Database (AD). Hvis man har et ansattnr, kan all annen informasjon finnes i AD gjennom spørringer, og derfor skal avsender stå registrert med ansattnr i hendelsesadministrasjonen.
- Konfigurasjonsenheter (CI) inkluderer programvare, maskinvare, tjenester, problemer, hendelser og endringer.
- Problemer og hendelser kan eskaleres flere ganger til ulike avdelinger, ikke til enkeltpersoner. Eskalering kan modelleres likt som historie til hendelse i relasjonsdatabasen (NIAM-modellen), men ettersom diagrammet er for komplisert allerede, velger jeg å ikke tegne inn eskalering i både NIAM og grupperingen av tabellene.
- Når forandring er vellykket, så fjernes eventuelt korresponderende kjent feil fra databasen.
- Én løsning kan ha flere kjente feil, men én kjent feil kan knyttes til kun én løsning.
- Én hendelse/problem kan ha flere løsninger og én løsning kan knyttes til én hendelse/problem eller begge på en gang, men ikke til flere hendelser/problemer.
- Begrepsdannelse er brukt for å identifisere begrepet i en mange-til-mange relasjon. For eksempel, har tjeneste et mange-til-mange forhold til programvare og omvendt. Så begrepsdannelse "Programvarer\_for\_tjeneste" viser oss hvilke programvarer som brukes for den aktuelle tjenesten.

Dette gjelder også for følgende begreper: "Berørte\_tjenester", "Problem\_for\_hendelse", "Programvare\_for\_tjeneste", "Hendelse\_historie",

”Problem\_historie”, ”Forandrings\_historie”. De ”historiske” tabellene gir oss informasjon om en hendelses, et problem, eller en forandrings ”livssyklus”.

Siden ansatte i SKD kan bytte arbeidsplass når de jobber i prosjekter (f. eks i IT-avdelingen), kan vi tenke at lokasjon har flere brukere og at én bruker kan jobbe på flere steder (lokasjoner). Begrepsdannelsen ”Brukere” gir oss informasjon om hvem som jobber hvor.

5.4.5.3 Niam diagram (Figur 25)



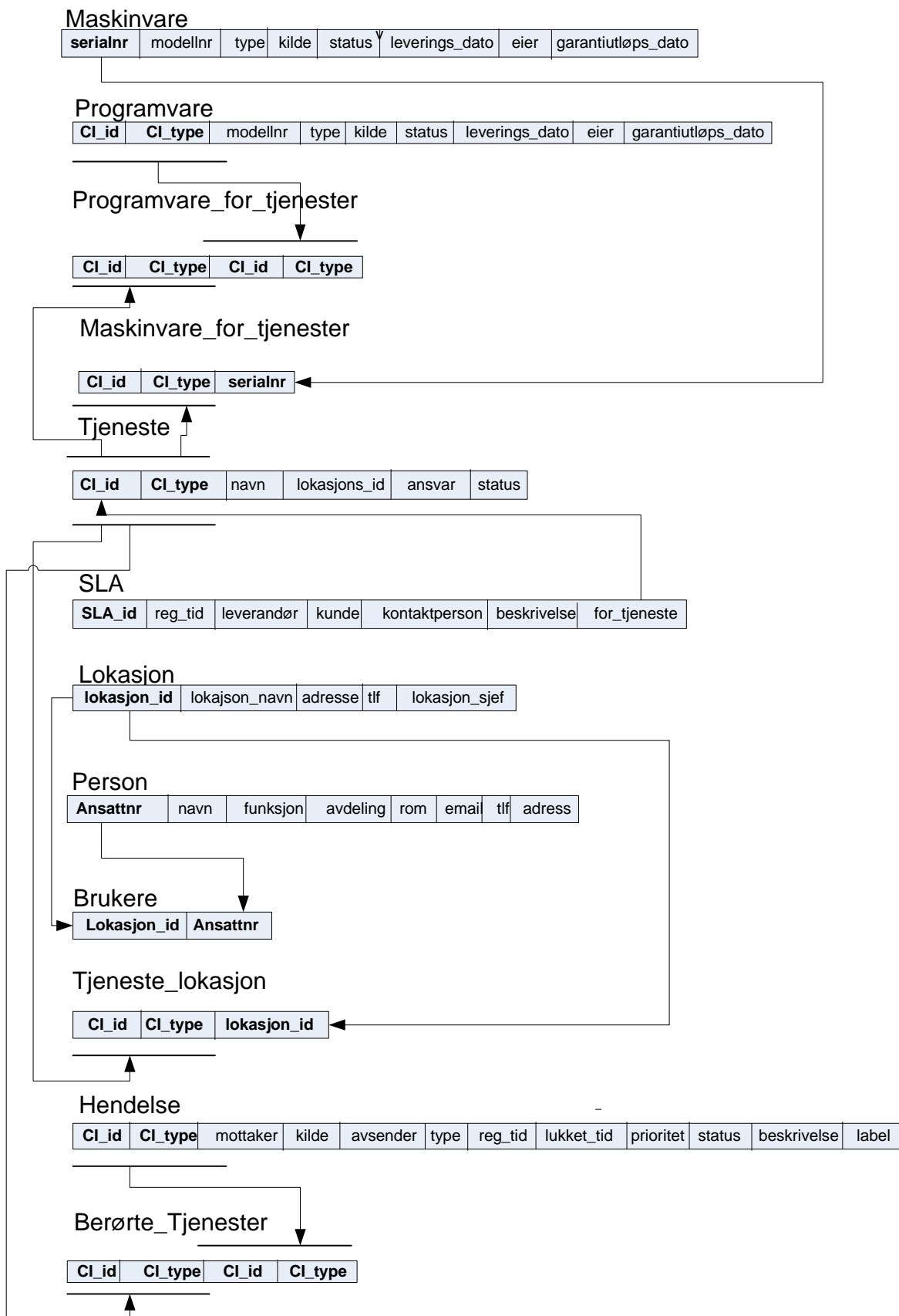
Figur 25: NIAM diagram for email tjenesten

### 5.4.6 Gruppering av tabeller

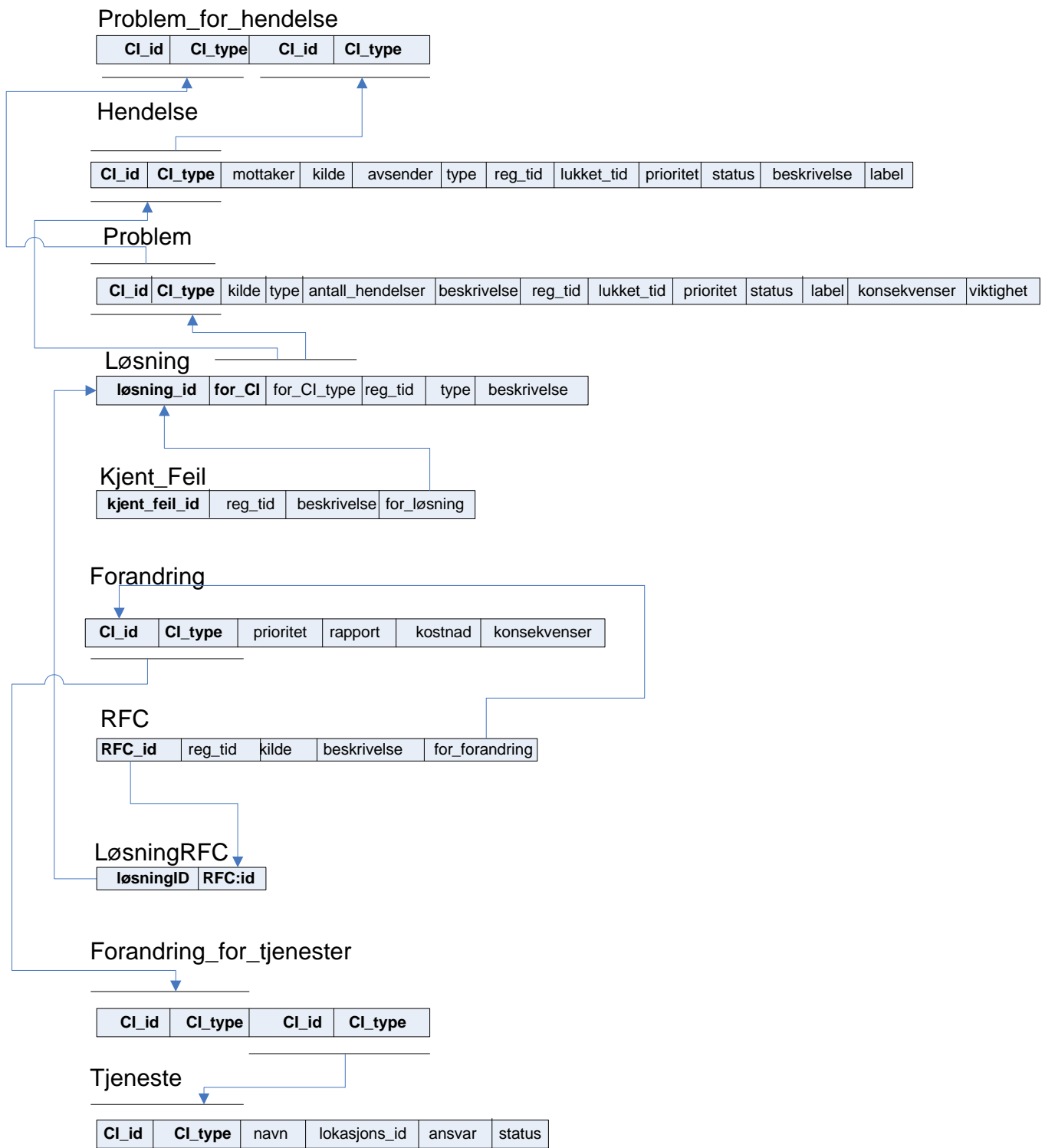
Ut fra NIAM diagram kan vi gruppere tabeller. Gruppering av tabeller gir oss et klart bilde av relasjonen mellom tabellene av databasen. Hver tabell representeres med navn og tilhørende attributter. Primærnøkler er uthevet i oversikten. Pilene som går fra en tabell til en annen, viser at tabellen pilen kommer fra, har en fremmednøkkel til tabellen pilen går til. Denne fremmednøkkel er sammensatt av en eller flere attributter, og den har samme antall attributter, med tilsvarende domene(r), i tabellen den peker på.

Siden det er mange tabeller og mange relasjoner mellom dem, blir det vanskelig å skaffe seg en klart oversikt hvis alle tabellene og sine relasjoner tegnes opp. Derfor har jeg valgt å dele det opp i to grupper som vises nedenfor.

Figur 26 illustrerer forhold mellom tjenester og brukere, maskinvare, programvare, lokasjon og SLA, mens Figur 27 viser forhold mellom tjenester og service support prosessene.



Figur 26: Relasjon mellom Tjenester og Brukere, Maskinvare, Programvare, Lokasjon og SLA



Figur 27: Relasjoner mellom Hendelse, Problem, Forandring, RFC, Løsning, Kjent\_Feil



## 5.4.7 Eksempler på SQL-spørringer

### 1. Finne berørte brukere av en hendelse

```
SELECT count (H.ansattnr), T.navn, L.navn, H.CI_id
FROM Berørte_Tjenester BT, Tjeneste_Lokasjon TL, Brukere B, Hendelse H,
     Lokasjon L, Tjeneste T
WHERE BT.ci_id = TL.ci_id
     AND TL.lokasjon.id= BT.lokasjon_id
     AND BT. ci_id = T.ci_id
     AND TL.lokasjons_id = L.lokasjons_id
     AND H.ci_id = BT.ci_id
GROUP BY lokasjons_id;
```

### 2. Finne berørt maskinvare og programvare til en hendelse

```
SELECT BT.ci_id, M.serialnr, P.ci_id
FROM Berørte_tjenester BT, Maskinvare_for_tjenester M, Programvare_for_tjenester P
WHERE BT.ci_id = M.ci_id
     AND BT.ci_id = P.ci_id;
```

### 3. Finne ut om en ny hendelse har et eller flere eksisterende problem

```
SELECT H.ci_id, P.ci_id
FROM Hendelse H, Problem P
WHERE H.label= P.label;
```

(Her må man sørge for at label er fra den nye hendelsen)

### 4. Finn ut om en hendelse har eksisterende kjent feil

```
SELECT H.ci_id, K.Kjent_feil_id
FROM Hendelse H, Løsning L, Kjent_feil K
WHERE H.ci_id = L.for_ci
     AND L.for_ci_type = hendelse
     AND L.løsnings_id = K.for_løsning;
```

### 5. Finn berørte tjenester av en endring

```
SELECT T.navn, FT.forandrings_id
FROM Tjeneste T, Forandring_for_tjenester FT
WHERE T.ci_id = FT.ci_id
     AND T.type = "tjeneste "
     AND FT.type = "tjeneste ";
```

### 6. Finne ut hvilke RFC som knytter seg til en bestemt løsning

```
SELECT L.løsnings_id, LR.rfc_id
FROM Løsning L, Løsnings_RFC LR
WHERE L.løsnings_id = LR.løsning_id
     AND L.løsnings_id = 'L1234';
(Antar at løsningen har løsnings_id = L1234)
```

## **5.5 Oppsummering**

Dette kapitlet er praksisdelen av oppgaven. Her har jeg beskrevet hvordan jeg bruker teori for å oppnå målet mitt, nemlig skissere deler av CMDB'en.

Kapitlet viser en mulig framgangsmåte, som spenner fra å bruke ITIL-teori samt organisasjonskultur og -struktur, for å kartlegge prosessene, finne attributter for prosessene, bestemme nivå på konfigurasjonsenheter, til å modellere et eksempel på en relasjonsdatabase for en enkelt tjeneste. Eksempelet blir presentert i NIAM-modellen og i gruppering av tabeller. I tillegg følger det med eksempler på noen SQL-spørringer.

Dette inneholder et eksempel på hvordan alle relasjoner mellom hendelsesadministrasjon, problemadministrasjon og endringsadministrasjon, de tre hovedprosessene i Service Support, kan modelleres.

## 6 Diskusjon og konklusjon

*Dette kapitlet tar opp mine tanker om oppgavens utførelse. Her diskuterer jeg blant annet ITIL's prosesser og hvordan disse kan tilpasses til Skattedirektoratet. Fordeler og ulemper ved prosessene, hvordan CMDB bygges opp, utfordringer for organisasjonen og hvordan SKD kan arbeide videre. Jeg tar også opp hva som er oppgavens resultat, og hva som har vært vanskelig underveis.*

### 6.1 Diskusjon rundt SKD sin prosesstilpasning

For en stor bedrift som tilbyr IT-tjenester og som har mange avdelinger og ansatte, er det en stor utfordring å styre/administrere avdelinger, ansatte, budsjetter og andre deler av organisasjonen. I en slik situasjon er det derfor behov for en god og oppdatert oversikt over infrastrukturen. Dette behovet er også med på å forme prosessene som ITIL-rammeverket tilbyr, som igjen er med på å legge forholdene til rette for gode, varierte og kostnadseffektive tjenestetilbud til kunder.

ITIL-rammeverket legger et grunnlag for, og faktisk tildels oppmuntrer til, at organisasjoner kan treffe sine egne organisasjonsspesifikke beslutninger om hvilke rutiner og prosesser som er formålstjenlige. IT-avdelingen i Skattedirektoratet søker å forbedre sine prosesser etter dette rammeverket.

Arbeidsrutinene i *driftsavdelingen* i dag er teknisk orientert. Det vil si man tenker bare på hvilken teknikk man skal benytte for å løse et problem. En slik rutine fører til at et problem kan løses på forskjellige måter, avhengig av kunnskap og erfaringen til den som løser det. Man må finne en ny løsning hver gang et problem oppstår, uansett om det er et gammelt eller nytt problem. Løsningen kan være forskjellig fra gang til gang, og forskjellig fra person til person. Denne måten å løse problemer på er ikke god nok, når organisasjonen er stor. Den viktigste grunnen til dette, er at man ikke bygger opp kunnskap om hyppigheten til enkelte problemer, og dermed får begrenset evne til å identifisere mulige svakheter i infrastrukturen. En slik rutine sløser altså bort tid og arbeid, på å gjøre mange av de samme tingene om og om igjen, og gir mangel på erfaringsoppbygging på det som faktisk utrettes.

Denne arbeidsmåten endres nå til å bli prosessorientert, dvs. at arbeid med hendelser og problemer skal følge en fast rutine. Prosessen skal støttes av en konfigurasjonsdatabase. All informasjon om prosesser fra hva som bør gjøres, kunder, hvem som løser hvilke hendelser/problem, og hvordan etc., kan registreres i databasen. Hovedformålet med databasen er imidlertid å registrere informasjon om IT-infrastrukturen, og hvordan de enkelte komponentene er satt sammen i ulike konfigurasjoner.

*Brukerkontakten* i SKD har til nå arbeidet mest med hendelsesadministrasjon, mindre når det gjelder problem- og endringsadministrasjon. Hendelsesadministrasjon er daglig arbeid i brukerkontakten og de er godt i gang. Prosessens arbeid er støttet av et verktøy av typen "Action Request System (ARS)", som heter Remedy. Remedy brukes blant annet til å registrere hendelser, kategorisere, tildele til riktig person, følge opp hendelser etter status og føre statistikk.

Hvis man begynner å implementere ITIL's hendelsesadministrasjon, så kan ansatte gi tilbakemelding om hvilke forandringer som er gode nok og hvilke som fremdeles bør justeres eller som rett og slett ikke passer inn i organisasjonen. Parallelt med fordelene ved å begynne med prosesser er det en del ulemper som også kan oppstå. En ulempe man kan tenke seg er at ansatte må forandre sine arbeidsrutiner samtidig med at de må utføre selve jobben. Dette kan i begynnelsen lede til noe kaos og frustrasjon.

Utfordringen er hvordan man kan redusere de initielle negative følgene ved overgang til nye rutiner og prosedyrer i organisasjonen. Mine forslag er i tråd med forslag som også tildels fremkommer i [34]:

- Administrasjon av hendelser skal fremdeles mottas og eskaleres som før. Den største forskjellen etter implementasjon av ITIL-prosessene, er mer registrering og oppdatering. Selve registreringen er også annerledes, for eksempel bør problemer registreres ved hjelp av kategorier, istedenfor symptomer. Ved hektiske situasjoner, blir oppdatering lett glemt bort og CMDB kan få dårligere kvalitet. Ettersom CMDB sterkt påvirker problemadministrasjon og endringsadministrasjon, kan dette føre til at mange andre prosesser som bruker CMDB blir påvirket.
- Ved hjelp av det internettbaserte E-hjelp-systemet kan man avhjelpe arbeidsmengden sentralt ved at superbrukere og lokale driftsavdelinger selv registrerer hendelser. Her kan også infrastrukturkritisk informasjon bringes ut til brukerne slik at man unngår mange unødvendige henvendelser.
- Integrere e-hjelp med CMDB, samt sørge for opplæring av brukere, slik at de velger e-hjelp istedenfor telefon.

Siden andre prosesser som problemadministrasjon, endringsadministrasjon og konfigurasjonsadministrasjon i dag ikke har noen faste rutiner/prosedyrer basert på prosessorientering, kan man bruke ITIL-rammeverket til dette. Arbeidsinstruksjoner og valg av attributter fra ITIL's forslag er blitt tilpasset organisasjonen (se vedlegg i punkt 5.3). På disse områdene får man kun fordeler (se kapittel 3) ved å innføre prosessorientering.

## **6.2 Diskusjon rundt bygging av CMDB**

Databaseforslaget som er laget med utgangspunkt i e-mail tjenesten, er en "empirisk" skisse som kan brukes som et hjelpemiddel for prosjektgruppen ved vurderinger om SKD skal bygge en CMDB og implementere ITIL-prosessene selv eller kjøpe dem inn, eventuelt for å forhandle om pris og tegne SLA ved innkjøpet.

Databasen er modellert etter min forståelse av teorigrunnet, og observasjoner av arbeidsrutiner i brukekontakten i SKD. Det har ikke blitt utført en detaljert undersøkelse, derfor kan det være noen regler som er uaktuelle, eller det kan mangle viktige regler i modellen.

Det er potensielt en stor database, med mange relasjoner å ta hensyn til. Noen relasjoner er i tillegg såkalt mange-til-mange. Det er mange forhold man må vurdere, f.eks. om databasen skal tilfredsstillende 4. eller 5. normal form, skal man bruke subklasser, er det noen "eop" (Equivalence Of Path) etc. Det dukker også opp en del praktiske spørsmål som f.eks. om en tjeneste skal ha en eller flere SLA'er, eller om er behov for å vite livsløpet til en

tjeneste. Fordi det som her presenteres bare er en skisse, vil verken jeg eller personalet i SKD forløpig bruke tid på å utforske disse praktiske spørsmålene. Som databasedesigner, kan jeg sette opp en del antagelser, men ikke forutsette alle.

De ovennevnte momenter er med på å gjøre det vanskelig å oppnå en god løsning basert på et relativt tynt grunnlag.

### **6.2.1 Valg av attributter**

Detaljer for konfigurasjonsenheter er valgt etter ønske og kompetanse til personalet i SKD. Det de vil ha hjelp til fra en CMDB er å vite hvilke tjenester som er berørt ved en hendelse, og eventuelt hvor mange brukere som er berørt. På bakgrunn av dette kan de så bestemme hvor viktig denne hendelsen er for å tildele prioritet. Deretter vil de vite hvilke fysiske enheter som er involvert og som trenger reparasjon/utskiftning. Dette er for å unngå å bruke tid på å lete etter involverte enheter. Etter samtaler med personalet i IT-avdelingen, har jeg besluttet å identifisere og klassifisere konfigurasjonsenheter som beskrevet i punkt 5.4.

Hovedprinsippet i databasen er konsistent dataregistrering. Derfor vil attributtene til prosessene være valgt etter følgende kriterier: hvilken informasjon prosessene trenger for å fungere, forslag i ITIL-rammeverket, hvilket nivå man velger å legge seg på i forhold til konfigurasjonen (hvordan CI skal representeres i CMDB'en), og hvordan attributter til hendelser er registrert i dag i Remedy-applikasjonen. I relasjonsdatabaseeksempelet, har jeg foreslått en enkel database med bare de mest nødvendige attributter, derfor er ikke alle attributter som står i attributt-tabellene (vedlegg 10.3) tatt med.

### **6.2.2 Logisk og fysisk database.**

Denne rapporten skisserer en logisk database, og vi ser at den kan komme til å bli både stor og komplisert. For å bestemme den fysiske utforming av databasen, er det en del spørsmål man må tenke gjennom:

- Hvordan skal transaksjoner og spørringer utføres?
- Resultater fra analyser om forventet frekvens av spørringer, transaksjoner og oppdateringer.
- Resultater fra analyser av krav om unikheter i attributter.
- Hvordan skal man indeksere attributter og hvilke attributter skal man indeksere fra?
- Skal databasen være sentralisert eller distribuert?

## **6.3 Hva er oppnådd**

### **6.3.1 For meg personlig**

Jeg forstår ITIL-prosesser og rammeverket, hvordan det hele tas i bruk, særlig i databaseperspektivet. Arbeidet med oppgaven har gitt meg kjennskap til hvordan driftsavdelinger i organisasjoner administrerer hendelser, problemer etc. Jeg har fått økt forståelse av hvordan man kan sørge for, i størst mulig grad, riktig omfang og pris på IT-investeringer.

Gjennom denne masteroppgaven har jeg fått øvelse i praktisk databasedesign. Det har gitt meg erfaring med vurdering og valg av metoder, attributter, skranker og diverse annen databaseteori (bl.a. ER, ORM, relasjonsalgebra, SQL), samt å sette opp regler for dataregistrering, osv.

### **6.3.2 For SKD**

Resultater for SKD, inkluderer bl.a. en kartlegging av instruksjoner for hendelsesadministrasjon, problemadministrasjon, endringsadministrasjon og konfigurasjonsadministrasjon. De har fått en skisse til et forslag til CMDB med kartlagte konfigurasjonsenheter, både for prosessene og komponenter i infrastrukturen, samt et forslag til mulig detaljnivå å legge seg på i forbindelse med registrering av ulike komponenter. Dette inkluderer attributter som kan være nyttige for prosessene, så vel som for relasjonsdatabasen.

I tillegg har de fått et eksempel på en mulig grafisk representasjon av en database i form av en NIAM-modell. De har fått en modellert relasjonsdatabase med fullstendige relasjoner når det gjelder prosessene, og enkelte komponenter i email-tjenesten (som består av kun program- og maskinvare). Dette eksempelet gir en modellering av alle relasjoner for hendelsesadministrasjon, problemadministrasjon etc.

Opgaven tar også hensyn til tjenestenivå-administrasjon (SLM) i forbindelse med databasen. SLA representeres som en tabell i CMDB'en.

## **6.4 Vanskeligheter underveis i arbeidet**

Opgaven er utført i en periode hvor SKD forløpig ikke har tatt i bruk ITIL-rammeverket. Personalet i SKD har heller ikke tidligere arbeidet på en prosessorientert måte, og jeg hadde ikke erfaring i praksis med hvordan driftsavdelinger håndterer hendelser.

Arbeidsinstruksjoner for organisasjonen er skissert på bakgrunn av studier av relevant teori og litteratur, samtaler, samt observasjon av personalet ved brukerkontakten. Jeg har imidlertid liten praktisk erfaring med hvordan spesielt hendelser håndteres, og mitt forslag til hvordan en CMDB kan se ut, forblir derfor kun en skisse. Det behøves flere diskusjoner med profesjonelle på det aktuelle fagområdet for å bygge en CMDB som reelt kan taes i bruk.

Ved siden av ulempene jeg nevnte overfor, finnes det fordeler ved at SKD ikke har tatt i bruk ITIL-prosessen også. Det vil si at jeg fritt kan bruke mitt grunnlag sammen med teori for å tilpasse ITIL på en slik måte at det legger til rette for at organisasjonen skal nå de mål som er ønskelige.

## **6.5 Utfordringer fremover**

### **6.5.1 Avgrensning av informasjon som lagres i databasen**

Hvis en CMDB består av all informasjon som gjelder IT-infrastrukturen, blir den et godt verktøy. Hvis derimot for mye informasjon som ikke er direkte nødvendig tas med, kan dette virke negativt inn på databasens nytteverdi. Sagt på en annen måte: IT-avdelingen i SKD ønsker å bruke CMDB for å styre tjenestene som Skatteetaten tilbyr. Dette kan imidlertid medføre en uforholdsmessig stor, og uhandterlig database. Selv om hastighet og dataprosesseringskraft stadig øker, er det viktig å tenke på å oppdatere en database som kan gi raske svar på de aller fleste SQL-spørringer.

## 6.5.2 Datakvalitet

For at prosessene skal være vellykket, må datakvaliteten være god. Dette betyr at oppdatering skal skje ved alle endringer. Det kreves gode vaner for oppdatering av databasen, fordi man bruker infrastruktur-informasjonen fra CMDB for å håndtere både hendelses-, problem- og endringsprosessene. I begynnelsen er det en stor utfordring å samle all informasjon, registrere det i en stor database, og å oppdatere data som er ”utløpt på dato”. Etablering av automatisert oppdatering byr på utfordringer.

## 6.5.3 Databasens struktur

Når man bygger opp en database, må man sette opp en del regler/antakelser, slik at databasen fungerer. Men hva skjer hvis en tilstand går utover en gjeldende antakelse, eller det skjer omorganiseringer i organisasjonen som bryter med regler i databasen. Skal man da bygge en ny database, eller forandre datastrukturen?

## 6.5.4 Integrasjon av verktøy

ITIL-rammeverket gir best utnyttelse ved automatisering, og det behøves verktøy for å automatisere.

Det kreves en sikkerlig undersøkelse av hvilke verktøy organisasjonen benytter, hvilke som er med i ulike pakker, men som ennå ikke er i bruk. Om verktøyene har den funksjonaliteten man trenger, og om det er mulig å integrere med eksisterende infrastruktur. Det største problemet her er å integrere verktøyene. For eksempel hvordan de forskjellige overvåkningsverktøyene kan knyttes direkte til Remedy-applikasjonen, og gi varsling på samme nivå som er spesifisert i databasen.

## 6.6 Videre arbeid

En komplett database trenger tid på å utvikles, så gjør også nye rutiner. Dette er en fordel for organisasjon fordi mens database og rutiner utvikler seg, og blir til, har personalet tid på å omstille seg.

### 6.6.1 Utvidelse av CMDB

Ettersom databasen modelleres etter prinsippet ”start small”, består den forløpig av bare de mest nødvendige attributtene. Det er flere rom for utvidelse etter hvert.

- For eksempel kan tabellene organiseres etter objektorientering istedenfor relasjonsdatabase som nå.
- Spesifisere CI mer detaljert, attributtens domener kan endres. For eksempel kategori for hendelser kan registreres etter tjeneste først, og deretter underkategoriseres etter tastevalg når brukere henvender seg via telefon i MV3-tjenester.
- Registrere flere attributter. Det kan lede til at relasjonene blir mer komplisert, men det kan også gi bedre informasjon.
- Attributter knyttes til relatert dokumentasjon som driftspersonale kan få rask tilgang til om det trengs.

- Neste prosess som kan inkluderes, er versjonsadministrasjon ("Release Management"). Siden Skatteetaten er under en kontinuerlig utvikling, og tjenester stadig endres og forbedres, er det behov for å vite hvilken versjon som brukes/prøves hvor. Dette er versjonsadministrasjonens ansvar.
- Integrere InfoChannel system (se punkt 4.4.2) til hendelseadministrasjonsverktøy, eller registrere problemer og løsninger i databasen.
- Tilknytning av ulike overvåkningsverktøy, slik at oppdatering av CMDB kan automatiseres i større grad.
- En annen vei til databaseutvikling er å knytte til flere av ITIL's moduler.

### **6.6.2 Utvidelse i bruk av CMDB**

I tillegg til å bruke CMDB i "Service Support" modulen, kan den brukes i mange andre situasjoner. For eksempel i forbindelse med applikasjonsadministrasjon. Denne løsningen kan gi ledelsen mulighet til å investere mer målrettet, og budsjettere mer effektivt, ved å ha kunnskap om hvilke tjenester som bruker flest ressurser, og hvilke tjenester man tjener mest på.

## **6.7 Oppsummering**

Kapittelet har diskutert hvordan ITIL prosessene kan tilpasses SKD. CMDB er bygd opp etter ulike kriterier. Jeg skisserer også hvordan CMDB'en kan utvikles videre, og trakk frem en del utfordringer som IT-avdelingen må tenke gjennom før utvikling.

Jeg oppsummerer også hvilke resultater som er oppnådd både for meg og for SKD, og noen vanskeligheter jeg har støtt på under arbeidet.



## 7 Ordliste

Liste over forkortelser.

AD: SKD's Ansatte Database  
AIX: Advanced Interactive Executive - IBM's version of Unix.  
AM: Availability Management  
ARS: Action Request System  
BSS: Brukerstøtte Service  
CA: Computer Associates  
CAB: Change Advisory Board  
CI: Configuration Item  
CIM: Common Information Model  
CM: Change Management  
CMDB: Configuration Database  
CMM: Capability Maturity Model  
COBIT: Control Objectives for Information and related Technology  
Con.M: Configuration Management  
CSIP: Continuous Service Improvement Programme  
DIS: Intrusion Protection System  
FITS: Financial Management for IT Service  
FR: Folkeregistrere  
FSC: Forward Schedule Change  
FSK:Fylkeskatekontoret  
ICTIM: ICT Infrastructure Management  
IDS: Intrusion Detection System  
IM: Incident Management  
IPS: Intrusion Protection System  
ITIL : IT Infrastructure Library  
ITSCM: IT Service Continuity Management  
LK: Ligningskontor  
MIB: Management Information Base  
MIT prosjekt: Modernisering av IT-drift i Skatteetaten prosjekt i SKD  
NA: Nettverk-administrasjon  
NAV = NMS : NAV står for Nettadministrasjonsverktøy, NMS står for Network Management System  
NIAM: Natural language Information Analysis Method  
NNM: Network Node Manager  
NTG: New Technology Generation  
OGC: Office and Government Commerce  
OLA: Operating Level Agreement  
PBX: private branch exchange, en privat telefon netverk brukes i virksomhet  
PIR: Post-Implementation Review

PM: Problem Management  
RFC: Request for Change  
RIM database: RDBMS Interface Module database  
RM: Release Management  
RMON: Remote Monitoring  
SKD: Skattedirektorat  
SLA: Service Level Agreement  
SLM: Service Level Management  
SM: Service Management  
SMI: Structure of Management Information  
SNMP: Simple Network Management Protocol  
The IT Service CMM: The Service Capability Maturity Model

## 8 Figurer

Figur 1: Spesifikasjon for oppgaven.....	10
Figur 2: Sammenheng mellom ulike felter i oppgaven .....	12
Figur 3:The ITIL Framework (kilde [9]).....	19
Figur 4: ITIL prosessene og deres samspill (kilde[9]) .....	24
Figur 5: IBM Tivoli Monitoring (kilde [15]).....	28
Figur 6: Hendelsesadministrasjon aktiviteter (kilde[18]).....	34
Figur 7: Relasjon mellom problem, kjent feil og RFC (kilde[17]).....	38
Figur 8: Problemadministrasjons arbeidsflyt (kilde[18]) .....	39
Figur 9: Grense mellom endringsadministrasjon og prosjektadministrasjon (kilde[18]).....	42
Figur 10: Detaljene og sammenhengene i "Service Support"(kilde[9]).....	46
Figur 11: Software for drift av Skatteetatens tele- og datanett (kilde[25]) .....	51
Figur 12: Hovedbildet i Remedy ARS .....	55
Figur 13: Relasjon mellom tjenester og brukere .....	59
Figur 14: Logisk relasjon mellom prosessene i Service Support .....	60
Figur 15:Incident Management Proses (kilde: [18]).....	61
Figur 16: System struktur .....	67
Figur 17: Maskinvarestruktur .....	68
Figur 18: Programvarestruktur .....	68
Figur 19:Nettverksstruktur .....	69
Figur 20:Databasesstruktur.....	69
Figur 21: Serverstruktur.....	70
Figur 22:Dokumentasjonsstruktur .....	70
Figur 23: Email tjenesten.....	71
Figur 24: Email tjeneste i klassifisering til konfigurasjon enheter.....	71
Figur 25: NIAM diagram for email tjenesten.....	77
Figur 26: Relasjon mellom Tjenester og Brukere, Maskinvare, Programvare, Lokasjon og SLA.....	79

Figur 27:Relasjoner mellom Hendelse, Problem, Forandring, RFC, Løsning, Kjent\_Feil.... 80

## 9 Vedlegg

### 9.1 Prosessadministrasjons-oversikt

#### 9.1.1 Oversikt over hendelsesadministrasjon

##### Hensikt/Målsetting

Å gjenopprette normale tjenester, eventuelt minimalisere problemer i forbindelse med et tjenesteavbrudd.

##### Oppgaver

Ta imot, klassifisere og følge opp innmeldte problemer. Hjelp brukere. Eskalere vanskelige problemer. Detektere mønstre og identifisere når det er underliggende problemer.

##### Rekkevidde

Enhver begivenhet (event) som bryter med en tjenestes normale drift, og som forårsaker en forstyrrelse eller reduksjon i tjenestekvalitet er innen rekkevidde av hendelsesadministrasjon.

##### Input

- Hendelsesbeskrivelse fra Service Desk, nettverk eller overvåkningsverktøy.
- Detaljert konfigurasjon fra CMDB'en.
- Resultat fra hendelse-"matcher" mot problem og kjent feil.
- Detaljert beslutning.
- Eksisterende løsning for hendelsen.

##### Output

- Oppdatere hendelsesregistrering inkludert beslutning og/eller "workaround".
- RFC for hendelsesbeslutning.
- Beslutte og lukke hendelse.
- Kommunisere med brukere/kunder.
- Oppdatere informasjon om hendelse i database.

##### Relasjon med andre prosesser

- **Problemadministrasjon:** hendelsesadministrasjonen skal kontakte problemadministrasjonen for å informere om nye problem, evt. oppdatere gamle problem. Hvis det er en kjent feil, kan hendelsesadministrasjonsprosessen hente løsning herfra.
- **Endringsadministrasjon:** Det kan hende at forandring til nye versjoner et eller annen sted kan være årsak til hendelser. Ved å vite om forandringsstatus for aktuell maskinvare kan dette hjelpe til med å løse problemet. En annen forbindelse er at eksakt samme hendelse gjentar seg, denne hendelsen skaper da en anmodning om endring (RFC), og RFC er da midlertidig løsningen.
- **Konfigurasjonsadministrasjon:** CMDB'en er sentral i alle service support prosesser. Her finner man hva og eventuelt hvem som er berørt av hendelsen,

altså typisk program-/maskinvare, brukere, tjenester, og hendelsens konsekvenser for infrastrukturen, inkludert dens brukere og tjenester.

- **SLA i “Service Management”**: Et eksempel. Hvis det er mange duplikathendelser eller det er et alvorlig problem, må man sjekke at tjenestenivået ikke faller utenfor rammene nedfelt i SLA. Hvis det siste er tilfellet, må leverandøren kontaktes.

### Overordnede aktiviteter

- Detektere og registrere hendelser.
- Klassifisere og initiere support.
- Undersøke og diagnostisere.
- Beslutningstagen og gjenvinning.
- Lukke hendelse.
- Eierskap, overvåkning, forfølge og kommunisere med brukere.

### Detaljert oppgavehåndtering

1. Ta imot beskjed om hendelse:

- Registrere grunnleggende hendelses-attributter (kilde, ankomme tid, beskrivelse, vedlegg).
- Registrere attributter om avsendere, herunder brukere (ansatt\_id/navn) og overvåkningsverktøy (hvilket verktøy, for hvilket system).
- Automatisk generere hendelses\_id (logg\_nr).
- Tildele status (dette skjer automatisk hvis overvåkingsverktøy er kilden, ved andre kilder gjøres dette manuelt).
- Hvis det er enkelthendelser som kan løses med en gang, løses problemet, og man går videre til punkt 6 (lukking av hendelse).

2. Klassifisere etter definerte kategorier:

- Klassifisere hendelse.
- Se etter om det er en kjent feil. Hvis dette er tilfellet, hentes en ”quick fix” eller ”workaround”, og man går til punkt 6.
- Informere problemadministrasjon hvis det er et nytt problem eller et problem som ikke har kjent feil.
- Undersøke konsekvenser (avhengighet mellom maskinvare og brukere, maskin-/programvare, brukere og brukertjenester), hente informasjon fra SLA/OLA og definere prioritet. Et illustrerende eksempel i denne sammenheng: Hvis en tjeneste går ned, må det kartlegges:
  - hvor mange brukere som er berørt
  - hvilke typer maskinvare som affekteres av denne tjenesten
  - hvordan dette influerer på annen programvare, og
  - hvilke konsekvenser dette har for driften.For å angi prioritet må i tillegg SLA konsulteres.
- Eskalere til høyere supportlinje, om nødvendig.

OBS! I følge hendelsesadministrasjon så er det her vi avgjør relatert konfigurasjonsavhengighet, men etter min mening hører dette punkt til i undersøke og diagnostisere.

3. Undersøke og diagnostisere:

- Vurdere hendelsen nøye.

- b. Samle all relatert informasjon, samt analysere denne.
- c. Vurdere en såkalt "workaround" eller om hendelsen skal sendes til en høyere supportlinje.
- d. Registrere kjent feil om nødvendig.
- e. Sjekk om det finnes en relatert RFC og/eller om det har skjedd endringer i maskinvare som følge av hendelsen.
- f. Registrere "workaround" hvis det i punkt c ble benyttet en slik.

Undersøkelse innebærer kontakt med konfigurasjonsadministrasjon for å få oversikt over hvilke brukere/tjenester/programvare/maskinvare som er berørt av hendelsen.

Diagnostiseringen innebærer å:

- Innhente informasjon fra endringsadministrasjon for vurdering om det er en RFC (detaljert oppgave håndteres i endringsadministrasjon).
- Innhente informasjon fra problemadministrasjonen. Hvis det er en kjent feil, hentes riktig løsning. Hvis ikke, benyttes "workaround".
- Hvis det er behov for "workaround" eller det er et nytt problem, registreres support (internt), eventuelt så velges en høyere support (internt/eksternt).
- Deretter be problemadministrasjonen registrere løsningen.

4. Beslutte og gjenvinne:

- a. Registrere nye løsninger eller "quick fix/ workaround".
- b. Informere problemadministrasjonen om nye problemer eller duplikathendelser.
- c. Send RFC til endringsadministrasjonen, om nødvendig (mest sannsynligvis for hendelse fra overvåkningsverktøy).

5. Hjelpe/kommunisere med brukere:

- a. Innhente informasjon om brukere/kunder fra personaldatabasen.
- b. Send svar til brukere eller oppsøk brukere for å løse problemet/ene.

6. Lukke hendelse:

- a. Oppdateres hendelse og generere rapport.
- b. Registrere kundebekreftelse.

7. Eierskap, overvåke, forfølge og kommunisere.

### **9.1.2 Oversikt over problemadministrasjon**

#### **Hensikt/Målsetting**

Formalisert innsats for å komme til hendelsens hovedårsak. Dette hjelper til med å redusere hendelsens konsekvenser og forhindre at den gjentar seg, og gjør dermed infrastrukturen mer robust.

Både proaktiv og reaktiv fremgangsmåte er nødvendig for å redusere effekten og hyppigheten av hendelser som reduserer tjenestenes tilgjengelighet og kvalitet.

#### **Oppgaver:**

Finne underliggende årsaker til ukjente problemer. Lage "workaround" eller endringsforslag for kjente problemer, eventuelt iverksette mekanismer for å begrense effekten av problemer/feil.

#### **Rekkevidde**

Formell definisjon av problemer som underliggende årsak til en eller flere hendelser. En kjent feil er et problem som har blitt vellykket diagnostisert.

Administrasjon av problemer og feil, og en proaktiv tilnærming til problemer, er alle komponenter i problemadministrasjonen.

### **Input**

- Detaljert hendelse fra hendelsesadministrasjonen
- Informasjon/varsler fra overvåkningsverktøy
- Detaljert konfigurasjon fra CMDB
- Alle definerte "workaround" fra hendelsesadministrasjonen
- Eksisterende løsning på problemet/ene.

### **Output**

- Kjente feil
- Anmodning om endring (RFC)
- Oppdatert problemregistrering (inkludere løsning og/eller tilgjengelig "workaround")
- Hvis problemet er løst, registrere lukking
- Svar fra hendelsesadministrasjonen om det passer til problem og kjent feil
- Annen relevant informasjon fra problemadministrasjonen.

### **Relasjon med andre prosesser**

- **Hendelsesadministrasjon:** Detaljert informasjon fra hendelsesregistrering som kan gi grunnlaget for både prioritering av problemet og løsning. I tillegg kan man få en såkalt "match" på kjent feil, og dermed ha en mulig midlertidig løsning i bakhånd.
- **Konfigurasjonsadministrasjon:** Er sentral i alle prosesser i "Service support". Ved å identifisere hvilke CI som er involvert eller berørt, kan man vurdere hvordan problemet skal løses.
- **Endringsadministrasjon:** Løsningen på et problem kan være å gjøre endringer i IT-infrastrukturen. Endringsadministrasjonen har ansvaret for alle disse. I tillegg kan en kommunikasjon mellom disse to prosessene avklare om et problem har oppstått fra en nylig foretatt forandring i infrastrukturen.

### **Hovedaktivitet**

- Problemkontroll
- Feilkontroll
- Proaktiv problemforebyggelse
- Identifisere tendens
- Få informasjon fra administrasjonsdata
- Finne hovedproblemet, og ta itu med dette.

### **Rekkefølge handling**

1. Problemkontroll: Problemadministrasjonen mottar hendelsesbeskrivelse fra hendelsesadministrasjon.

- a. Identifisere problemet og registrer det
- b. Klassifisere problemet (kategori, mulig konsekvens, viktighet, prioritet)
- c. Supportgruppe analyserer og diagnostiserer problemet for å finne årsaken.

2. Feilkontroll.



Hvis det er kjent feil (system):

- a. Identifisere som kjent feil, registrere
- b. Vurdering
- c. Registrere løsning (RFC),
- d. Lukking
- e. Overvåke hvordan gitt løsning fungerer
- f. Finn mulige løsninger og registrer
- g. Send løsningen til hendelsesadministrasjonen.

Hvis det ikke er kjent feil:

- a. Lag endelig løsning eller "workaround"
- b. Når det gjelder å etablere en "workaround", kan en eller flere RFC være nødvendig.

### 3. Proaktiv forebygging av problemer

- a. Ved hjelp av trendanalyse kan man identifisere mulig svakheter i infrastrukturen, og iverksette eventuelle forebyggende tiltak for å demme opp for hendelser som oppstår på grunn av disse svakhetene.
- b. Utarbeide og distribuere nødvendig informasjon til organisasjonen/ bedriften.

### 4. Send endringsanmodning (RFC) til endringsadministrasjonen

Hvis problemadministrasjonen får godkjent endringsanmodning sendt fra endringsadministrasjonen

- a. Iverksette endelig løsning eller "workaround"
- b. Beskriv løsning og lukk problemet  
ellers beskriv "workaround" og definér som kjent feil

### 5. Informasjon til hendelsesadministrasjonen om endelig løsning eller "workaround"

## 9.1.3 Oversikt over endringsadministrasjonen

### Hensikt/Målsetting:

Sørge for en strukturert og definert endringsprosess, prosedyrer, redskaper og andre komponenter som er nødvendige for å planlegge endringer på en effektiv og kostnadseffektiv måte.

Så langt det er mulig, skal man søke å unngå at endringen ikke forstyrr eksisterende IT-konfigurasjon.

### Oppgave

Ta imot, administrere behandling av endringssøknad, prioritere, godkjenne og koordinere endringer ved å opprette et endringsstyre ("Change Advisory Board", eller CAB) hvor det beskrives utviklingsovervåkning, testing og implementering og -evaluering, dokumentering og informering.

Håndtere endringslogg og historikk.

### Rekkevidde

- Maskinvare
- Programvare
- Forretningsapplikasjoner
- Kommunikasjonsutstyr og programvare

- Administrasjon og prosedyrer angjeldende IT infrastruktur

### **Input**

- Detaljert RFC
- Detaljert konfigurasjon
- Registrering av forrige modus av forandring ("Forward Schedule Change", FSC)
- Registrering av involverte hendelser og/eller problem
- Detaljert RFC vurdering

### **Output**

- RFC rapport (om det er godkjent, utført og lukket eller forandret til ny(e) RFC)
- CMDB oppdatert
- FSC oppdatert
- Rapporter og aktiviteter som endringsstyret er involvert i.
- Endrings-prosesserapport

### **Relasjon til andre prosesser**

- **Hendelsesadministrasjon:** Nesten alle RFC opprinner fra hendelsesadministrasjonen. Det kan oppstå en situasjon der flere hendelser har bare "workaround" som løsning, det vil si at det fremdeles kommer duplikathendelser. Denne situasjonen kan bare løses ordentlig når hovedårsaken blir funnet (forhåpentligvis). I tillegg, når en forandring har vært vellykket, skal det gis beskjed om hvilken hendelse eller hvilke hendelser som denne berører, til hendelsesadministrasjonen.
- **Problemadministrasjon:** Endringsadministrasjon foregår etter RFC fra problemadministrasjonen og skal gi denne tilbakemelding for å oppdatere problemløsninger og tabell over kjente feil.

Her kan man fjerne både tilsvarende problemløsninger og kjente feil siden hovedårsaken er kjent, eller registrere at disse problemene er løst.

Jeg foretrekker løsning nummer to, nemlig registrere at hovedårsaken til problemet/ene er løst. Ikke fjerne problemene for godt fra databasen. De gir oss nemlig oversikt over problemer som har oppstått, noe som vil hjelpe til med å forstå og løse problemer i fremtiden.

- **Konfigurasjonsadministrasjon:** Informasjon om gjeldende maskin- og programvarekonfigurasjon etter foretatte endringer

### **Hovedaktivitet**

- Planlegging
- Sette opp et endringsstyre, og holde endringsstyremøter, hvor et bredt utvalg av IT-ressurser møtes for å diskutere og vedta endringer.
- Koordinere endringsprosessen.
- Evaluere prosessen for å se på nytteeffekten av endringer, samt lokalisere forbedringspotensiale i arbeidet med endringer.

### **Rekkefølge handling**

#### 1. Planlegging

- a. Registrere og klassifisere ønske om/behov for endring(er).
- b. Vurdere konsekvenser, omkostninger, fordeler og risiki ved å gjennomføre

- endringen, og tildele prioritet.
  - c. Godkjenne/ avslå endringsanmodninger (RFC).
  - d. Hvis den blir godkjent, blir det aksept for å utføre endringen i det omfang og tidspunkt som legges til grunn, og prioritet tildeles.
- 2. Sette opp et endringsstyre, og holde endringsstyremøter, hvor et bredt utvalg av IT-ressurser møtes for å diskutere og vedta endringer.
  - a. Forvalte produksjonssettingsplanen.
  - b. Håndtere og koordinere utviklingen og produksjonssettingen av endringen.
  - c. Informere berørte parter.
- 3. Koordinere endringsprosessen.
  - a. Koordinere utviklingsarbeidet og produksjonssettingen.
  - b. Måling og rapportering.
  - c. Fakturere for medgåtte ressurser.
- 4. Evaluere prosessen for å se på nytteeffekten av endringer, samt lokalisere forbedringspotensiale i arbeidet med endringer.
  - a. Vurdere resultatet av endringen.

### 9.1.4 Oversikt over konfigurasjonsadministrasjonen

#### Oppgave/Målsetning:

Konfigurasjonsadministrasjonens oppgave og målsetning, er å holde en hele tiden oppdatert oversikt over komponentene som utgjør IT-infrastrukturen. Dette er nødvendig for at andre prosesser som f.eks. hendelses-, problem-, og endringsadministrasjon skal kunne fungere til fulle. Administrasjonen innebærer at enhetene identifiseres og registreres i en database (CMDB) over konfigurasjonsenheter (CI), slik at denne til enhver tid reflekterer tingenes tilstand i organisasjonens IT-infrastruktur. Utfra denne CMDB kan man f.eks. finne ut hvilke konfigurasjonsenheter som knytter seg til de enkelte tjenestene organisasjonen tilbyr, ved hjelp av SQL-spørringer. Når en tjeneste er nede, kan således driftsansatte i førstelinje-support finne ut hvor mange brukere som er berørt, for å tildele prioritet til en hendelse. Mens de i andre- eller tredjelinje-support har interesser i hvilke CI som er involvert i en hendelse.

#### Rekkevidde

- Maskinvare
- Programvare
- Tjenester
- Brukere
- Relatert dokumentasjon
- Informasjon fra andre prosesser i ITIL.

#### Input

All nødvendig informasjon om konfigurasjonsenheter. Eksempler herunder kan være alt fra versjoner, produsent og lisenser for programvare, SLA for tjenester, til hvilke typer maskinvare det er snakk om.

#### Output

En oversikt over IT-komponenter i relasjonsdatabasen (CMDB). CMDB'en inneholder også informasjon om bl.a. hendelsesadministrasjon, problemadministrasjon,

endringsadministrasjon, kjente feil, versjonsadministrasjon samt arbeidstakere, leverandør, plassering og forretningsenheter.

### **Relasjon til andre prosesser**

Konfigurasjon er en prosess som de fleste andre prosessene drar veksler på. Både hendelses-, problem-, og endringsadministrasjon er avhengig av informasjonen som konfigurasjonsadministrasjonen tilbyr. Dette gjelder ved tildeling av såvel prioritet som viktighetsgrad for hendelser, problemer og endringer, og ikke minst med å finne ut hvilke komponenter i infrastrukturen som er berørt av en eller annen hendelse, og hvilke tjenester og brukere som lider under dette. Forholdet er toveis i og med at resultater fra andre prosesser kan innebære endringer i konfigurasjonen.

### **Hovedaktivitet**

- Planlegging
- Identifikasjon
- Kontroll
- Statuskontroll
- Bekreftelse og revisjon

### **Rekkefølge handlinger**

#### 1. Planlegging:

Innebærer å definere formålet med administrasjonen, dens rekkevidde, rutiner, konvensjoner, og prosedyrer, og hvordan det hele skal organiseres.

#### 2. Identifikasjon:

Bestemme hvordan konfigurasjonsenheter skal representeres i databasen. Med tanke på bl.a.:

- a. Type (program-, maskinvare, tjenester).
- b. Relasjoner med andre enheter.
- c. Dokumentasjon.

#### 3. Kontroll:

- a. Registrere nye CI.
- b. Oppdatering av CMDB ved endringer.
- c. Lisenskontroll.
- d. Beskytte konfigurasjonens integritet.

#### 4. Statuskontroll:

- a. Produsere rapporter som sier noe om utviklingen og statusendringer for konfigurasjonsenheter, som kan forkomme på grunn av endringer i infrastrukturen.

#### 5. Bekreftelse og revisjon

- a. Sjekke om nye CI eller konfigurasjoner passer til gjeldende konfigurasjon og/eller omvendt. Sikre at kartet stemmer med terrenget i den forstand at databasen beskriver alle CI riktig, og på en slik måte at det ikke bryter med noen av prinsippene for hvordan konfigurasjonsenheter skal registreres og følges opp.

## 9.2 Tabell for prosess administrasjon

### 9.2.1 Tabell for hendelsesadministrasjon

Overordnet oppgave (hoved-aktivitet)	Input	Relasjon med andre prosesser	Detaljert oppgave	Attributter	Hva skal disse attributter brukes til (hvilke behov dekkes av informasjonen)	Output	Merknad
Ta imot hendelse	Hendelse fra Service Desk	CMDB	1) Registrere hendelse med grunnleggende hendelsesattributter (kilde, avsender, mottaker, ankommet tid, beskrivelse, vedlegg)	- Innkommende kilde (fra e-mail, tlf, eller fra overvåkningsverktøy)	- Skille problem/ feil type fra overvåkningsverktøy og fra andre kanaler. Hendelser fra overvåkningsverktøy er typisk tekniske problemer og/eller maskinvare, mens hendelser fra andre kanaler er mest typisk problemer med programvare eller i forbindelse med veiledning.	- Registrert hendelse  - Gjenkjenne feil i CMDB	Vi skal bare registrere ansattnr til avsender, annen info kan hentes fra AD
				- Avsender (ansattnr eller progamvare_id til overvåkningsvertøy)	- Gi tilbakemelding til avsender hvis henvendelse er brukere - Automatisk klassifisering og automatisk prioritet hvis hendelse kommer fra overvåkningsverktøy.		

				- Mottaker	- Tildele ansvar for hendelse. Brukere kan få navn på saksbehandler, som de kan henvende seg til		
				- Ankommet tid (reg_tid)	- Oversikt over hvor mange hendelser i en gitt tidsperiode		
				- Beskrivelse	- Tekstlig beskrivelse av hendelsen		
				- Vedlegg	- Lenke til vedlegg		
			2) Tildele id	- Loggnr (etter AR System User)	- Bruke som referanse nr	- Informere til avsender om hendelses-  håndtering eller bare rett og slett bekreftelse om hendelse(r) er mottatt og gi den/dem et loggnr som referanse nummer.	
			3) Tildele status	- Status (ny)	- For å følge med og styre hendelse - For å styre ansatt kapasitet (tildele oppgaver til de som har ansvar)		Forslag til domene for status: ny/ gjenoppretting/  videresendt til høyere supportlinje/ løst/lukket

			4) Grovt kategorisere problem	- Type	- Kategorisere hendelsen etter definert type.		Forslag til kategori-domene: feil, dokumentforespørsel, tjenesteforespørsel
Klassifisere og initiere support	<ul style="list-style-type: none"> <li>- Lagret detaljert hendelse</li> <li>- Detaljert konfigurasjon fra CMDB</li> <li>- Svar fra hendelse-matcher mot problem-adminis-trasjon og kjent feil</li> </ul>	<ul style="list-style-type: none"> <li>- CMDB (ved inn registrert)</li> <li>- Problem-admin.</li> <li>- SLA om nødvendig (f.eks. problem med oppetid skal sjekkes mot SLA om det er innenfor rammer.)</li> </ul>	1) Klassifisere hendelse	<ul style="list-style-type: none"> <li>- Kategori</li> </ul> <p>Forslag i ITIL (program-/maskinvre/ tjenesteforespørsel/for spørsel om informasjon, dokumenter, råd)</p>	<ul style="list-style-type: none"> <li>- Hjelp til å fordele problemer til riktig person/avdeling</li> <li>- Hjelp til å tildele prioritet</li> <li>- Utarbeide statistikk</li> </ul>	<ul style="list-style-type: none"> <li>- RFC for å ta itu med hendelse</li> <li>- Oppdatert hendelse</li> <li>- Workaround eller send hendelse til høyere support</li> </ul>	Vi kan kategorisere etter tjeneste. Dette er forslaget fra SKDs IT avdeling.

		2) Finn SLA for tjeneste som har hendelsen	- Involverte SLA	- For å forhandle med leverandør hvis hendelse er utenfor avtale.		
		3) Matcher kjent feil og problem-administrasjon (hvis ja, hent løsning og gå til "lukke hendelse")				
		Informere problemadministrasjon hvis det er et nytt problem, duplikathendelse, eller et problem som ikke har noen "match" (det kan være samme beskrivelse, men annerledes grunn, eller feillen kan komme som følge av en annen årsak)	- Definere nytt problem - Registrere duplikate hendelser			



			4) Undersøke virkning og definere prioritet	- Berørte_tjeneste - Berørte_brukere	- Hente info fra Service katalog (CMDB) om tjenester (hvilke tjenester som er berørt) og brukere (hvilke, og hvor mange brukere som er berørt)  - Ved å vite relasjon mellom tjenester og brukere, kan vi vite hvor mange som er berørt av en hendelse. Da kan man gi riktig prioritet		
			6) Tildele prioritet	- Prioritet - Tilgjengelighets -krav	- Slik at man kan bruke ressursene på de alvorligste problemene først.		- Oppdatert detaljert hendelse i CMDB  - Oppdagelse av feil i CMDB
Undersøke og Diagnostisere	- Oppdatert detaljert hendelse  -Detaljert konfigurasjon fra CMDB	- Konfigurasjon detaljert fra CMDB (både hente info og registrere inn)  - Endringsadministrasjon.  - Problemasministrasjon.	1) Vurdere detaljert hendelse  - samle og analysere av alle relatert informasjon og beslutning  - finne workaroud eller send til høyere support linje	- Involverte brukere/tjeneste/programvare/ maskinvare		- RFC  - Hendelsesløsning , inkludere detaljert hendelse	Konfigurasjonen må vise oversikt over relasjon mellom maskinvare og brukere, maskin- og programvare, brukere og tjenester.
			2) Bygge en RFC	- RFC-id			

			3) Matcher kjent feil				
			4) Er det behov for workaroud?				
Beslutte og gjenvinne	-Oppdatert hendelse  - Respons på RFC fra Endrings-admin.  - Avgjøre om workaroud eller utforme løsning	Endringsadmi n.	1) Hente løsning (hvis kjent problem) eller henvende brukere til løsningskilde			- Oppdatert detaljert hendelse  - Lukket hendelse registrering  - Rapportering	
			2) Eller sende til høyere support linje hvis nye problem	- Eskalering (tid, avdeling, ansvarlig)	- Hvis hendelsen eskaleres til høyere support linje, må eskalering registrert med tidspunkt, avdeling og person som har ansvar for hendelsen		
			3) Registrere ny løsning eller work-around, eller sende en RFC til endringsadministrasjonen	- Løsning (kategorier, konsulent som står for løsning, dato for løsning, problembeskrivelse, løsningsbeskrivelse, RFC(ja,nei), RFCs beskrivelse, løsningsstid , kostnad )			

Hjelpe/ Kommunisere med brukere			1) Innhente info om avsenderen  2) Sende svar til brukere eller komme og løse problem				
Lukke hendelse	- Oppdatert detaljert hendelse  - Løst hendelse	CMDB (registrere inn)	1) Be om bekreftelse fra kunder/ brukere om at problem er løst  2) Registreres dato for avslutning	- Oppdatere hendelses- status = lukket  - Oppdatere detaljert hendelse			
Eierskap, overvåke, forfølge og kommunisere	Hendelse registrering		1) Overvåke hendelse 2) Eskalere hendelse 3) Informere brukere	Hente info om hvem som har hvilket ansvar for å eskalere videre (navn, roller, ansvar, problem)		- Skrive ut rapport om hendelsens framgang  - Kommunisere med brukere og rapportere	

## 9.2.2 Tabell for problemadministrasjon

Overordnet oppgave (hoved-aktivitet)	Input	Relasjon med andre prosesser	Detaljert oppgave	Attributter	Hva skal disse attributter brukes til (hvilke behov dekkes av informasjonen)	Output	Merknad
Problem kontroll	<ul style="list-style-type: none"> <li>- Detaljert hendelse fra hendelses-administrasjonen</li> <li>- Informasjon/ varsler fra overvåknings-verktøy</li> <li>- Detaljert konfigurasjon fra CMDB</li> <li>- Alle definerte workaround fra hendelses-administrasjonen</li> <li>- Eksisterende løsning på problemet(ene)</li> </ul>	<ul style="list-style-type: none"> <li>- Hendelses-administrasjon</li> <li>- Konfigurasjons-administrasjon</li> <li>- CMDB</li> <li>- SLM</li> </ul>	- Identifisere problemet og registrer det	Problem_id	- Identifikator for problemet i databasen	<ul style="list-style-type: none"> <li>- Oppdatert problemregistrering (inkludere løsning og/eller tilgjengelig workaround)</li> <li>- Svar fra hendelses-administrasjonen om det passer til problem og kjent feil</li> </ul>	- Vi skal bare registrere ansattnr til avsender, andre info kan hentes fra AD
				Beskrivelse	- Utfyllende beskrivelse av problemets art og omfang.		
				Relatert hendelser og antall	- Opplyse om hvor mange, og hvilke hendelser som er årsaken(e) til problemet.		
				Reg_tid	- Tidspunkt for registrering av problem		
				Kilde	- Hvor problemet oppstod		
				lokasjon	- Hvilke avdelinger som er berørt av problemet		
				- Klassifisere problem	Type		
			Prioritet				
			Status		- Eksempler på status: åpen, i_arbeid, lukket		

				Label			
				Hastighet	- Hvor lenge situasjonen kan vedvare før problemet må være løst. Brukes som del av bakgrunnsinformasjon for å tildele prioritet.		
			- Analyse og diagnose	Kostnad		- Undersøke dokumenter. - Oppdatere beskrivelse (beskrivelsesattributtet) med konsekvenser. - Kjente feil - Anmodning om endring (RFC)	
Feil-kontroll	- Eksisterende feil-registrering  - Konfigurasjonse nhet	- Endrings- administrasjon  - SLM	- Hvis kjent feil (system):  1) Identifisere som kjent feil, registrere	Kjent_feil_id		- Kjent feil oppdatert  - Løsning  - Oppdatert prioritet for RFC  - Konsekvensanalyse	
			2) Vurdering				
			3) Registrere RFC som løsning	RFC_id	Unik identifikator		
				Reg_tid	Tidspunkt for registrering av RFC		
Kilde	Hvem som anmoder om endring						

			Beskrivelse	Tekstlig beskrivelse av RFC		
		5) Overvåke hvordan gitt løsnings fungerer				
		6) Finn mulige løsninger og registrer	Løsnings_id	Unik identifikator		
	Reg_tid		Tidspunkt for registrering av løsning			
	Type		Type løsning (workaround, RFC, kjent feil)			
	Beskrivelse		Tekstlig beskrivelse			
		7) Send løsningen til hendelsesadministrasjonen				
		- Hvis det ikke er kjent feil:  1) Lag endelig løsning eller workaround	Kjent_feil_id	Unik identifikator		
	Reg_tid		Tidspunkt for registrering av kjent feil			
	Beskrivelse		Tekstlig beskrivelse av kjent feil			
		2) Når det gjelder å etablere en workaround, kan en eller flere RFC være nødvendig.	RFC_id	Unik identifikator		
	Reg_tid		Tidspunkt for registrering av RFC			
	Kilde		Hvem som anmoder om endring			

				Beskrivelse	Tekstlig beskrivelse av RFC		
Proaktiv problemforebygging	- Data fra overvåkningsverktøy, både direkte varsler, og når ulike terskler nåes (antall brukere, CPU, minne, grafikk)	- SLM	1) Trendanalyse			- Registrering av potensielle problemer/hendelser  - Tilrettelagt dokumentasjon	
			2) Mål				
			3) Utarbeide og distribuere nødvendig informasjon til organisasjonen/bedriften				
Send endringsanmodning (RFC) til endringsadministrasjonen		- Endringsadministrasjon	1) Iverksette endelig løsning			Oppdatere løsningen/workaround /kjent feil, eventuelt slette kjent feil fra databasen  Oppdatere status til problemet  Registrere lukket tid	- Hvis Problemadministrasjonen får godkjent endringsanmodning sendt fra endringsadministrasjon, skal punkt 1 og 2 under kolumnen ”detaljert oppgave” iverksettes.
			2) Beskriv løsning og lukk problemet	Lukket_tid			

			3) beskriv "workaround" og definér som kjent feil				Ellers skal punkt 3) gjelde.
Informasjon til hendelses-							
Ad. om endelig løsning eller "work- around"							



### 9.2.3 Tabell for endringsadministrasjon

Overordnet oppgave (hoved-aktivitet)	Input	Relasjon med andre prosesser	Detaljert oppgave	Attributter	Hva skal disse attributter brukes til (hvilke behov dekkes av informasjonen)	Output	Merknad
Planlegging	<ul style="list-style-type: none"> <li>- Detaljert RFC</li> <li>- Detaljert gjeldende konfigurasjon</li> <li>- Foreløpig endringsoppsett (Forward Schedule of Change, FSC)</li> <li>- Involvert hendelse og/eller problem</li> <li>- Detaljert RFC-vurdering</li> </ul>	<ul style="list-style-type: none"> <li>- Hendelses administrasjon</li> <li>- Problem administrasjon</li> <li>- Konfigurasjons administrasjon</li> <li>- CMDB</li> </ul>	1) Registrere og klassifisere ønske om/behov for endring(er)	For_RFC	Fremmednøkkel til RFC-tabellen. Den RFC som forandringen gjelder for.	Godkjenningsrapport eller oppdatert RFC	Dette er hoved-RFC for forandringen
				Linket_RFC	Eventuelle andre RFC som endringen gjelder for.		Er det noen andre RFC som har relasjon (forelder/barn) til "hoved-forandringen"
			2) Vurdere konsekvenser, omkostninger, fordeler og risiko ved å gjennomføre endringen, og tildele prioritet				
			3) Godkjenne/avslå endringsanmodninger	Avgjørelse (Lenket til avgjørelses-dokumenter)	Rapport som beskriver vurderingen av hoved-RFC, basert på konsekvensanalyser.		Avgjørelse kan være godkjenning, avlyst/suspendert eller forandret RFC
			Avgjørelsesgrupper	Skal registrere navn på den/de som har avgjørelsesmyndighet i de aktuelle gruppene.			

			4) Tildel prioritert hvis RFC godkjennes	Prioritet	Endringens tildelte prioritert etter analyse av konsekvenser.		
Logg av endringer og filtrering	- Endringsanmodninger fra organisasjonen  - Filtrering		Loggføring	Forandrings-id	For å skille mellom forandringer som blir foretatt flere ganger eller ulike forandringsmodus av en RFC.	-Produksjonssettingsplan rapport  - Endringsutførelsesrapport	
				Beskrivelse	Tekstlig beskrivelse av forandringen		
			Filtrering	Linket_hendelse	For å finne ut mer informasjon om konsekvenser av forandringen		
				Linket_problem	For å oppdatere i problemadm. om permanent løsning og lukke hovedårsak.		
			Kategorisering	Type	Ved å dele forandring i kategori, kan man velge riktig arbeidsplan, kostnader og ressurs.		
			Konsekvensutredning	Hovedforandrings_CI	Denne kan man finne frem til ved hjelp av sql-spørringer.		
				Involverte_CI	For å vite hvilke andre konfigurasjons-enheter som er involvert i forbindelse med forandringen.		

				Andre konsekvenser	Andre konsekvenser av forandring, f.eks arbeidsrutiner som må endres, behov for opplæring, personal-endringer, og/eller konsekvenser i sin helhet som må registreres som informasjon til berørt enhet, slik at planleggingen kan bli foretatt på riktig grunnlag.	
			Ressurvaluering og -allokering	Utførelsesgrupper (leder, navn)	Navn på alle personer i gruppene og hovedansvarlig i gruppene	
				Antatt tid nødvendig for forandrings-utførelsen	Foreløpig planlagt tidsangivelse. Gjenstand for endringer underveis.	
		Endrings-styremøter	Ressurs		Beskrivelse av hvilke ressurser som går med til selve utførelsen.	Attributter til utførelse
				Arbeidsplan/ arbeidsprosedyre	Arbeidsplan for utførelsen.	
				Kostnader	Kostnader i forbindelse med forandringen, i form av såvel økonomi, tid, og andre ressurser.	
		Godkjenning av endringer	Dokumentasjon		Dokumenter nødvendig for utførelsen av endringen(e).	All relatert dokumentasjon, både Service Desk, teknisk dokumentasjon, programvare-

							dokumentasjon etc.
			Lage et endrings-oppsett	Start_tid	Start for utførelse av endringen.		
			Gjennomføre endringer	Test	Registrering av tid og sted for testen, samt produksjon av en rapport for testen.		
				Versjon	Registrere ny versjon hvis man bruker versjonshåndtering.		
				Avsluttet_tid	Tidspunktet for ferdig utført endring.		
				Faktisk tidsforbruk	Hva resultatet ble, når det gjelder tidsforbruk.		
			Evaluere endringer som er foretatt	Rapport	Tekstlig rapport om hele forandringens livsløp.		
Evaluering av endringsprosessen			Dokumentering			Evaluerings rapport	
			Informering				

## 9.3 Prosess administrasjon attributter

Disse tabellene viser hvilke data brukes i prosessen, og hvordan. Ut fra denne tabellen kan man finne ut attributter, eventuell tabeller og relasjoner mellom dem i prosessen. Her bruker jeg begrepet attributt for alle dataene. Imidlertid vil det ikke være naturlig å registrere alle som attributter, noen av dem kan finnes ut fra andre, for eksempel *okkupert\_tid* kan regnes ut fra *start\_tid* og *slutt\_tid*.

Det er også et tenkelig bruksformål til "nesten" hver enkelt attributt. Det som ligger i "merknad" felt forklarer hvilken mening attributtet har.

### 9.3.1 Attributter for hendelsesadministrasjon

#### Antakelse:

- "Bruker"-attributter er ferdig definert og registrert i AD katalogen. Her finner man all informasjon man trenger om brukere som adresse, telefon, avdeling, rom, type...
- Ettersom SKD er en statlig etat, og alle tjenester er gratis distribuerte til brukerne, nemlig ligningskontorer og fylkeskattekontorer, kan vi slå sammen "kunder"-begrepet med "bruker"-begrepet. Her foretrekker jeg "**BRUKERE**" for alle som henvender seg til "Service Desk".

RELASJONS-NAVN	ATTRIBUTT	HVORDAN ATTRIBUTTER BRUKES	MERKNAD
Hendelse (Incident)	Kilde	Skille hendelse fra overvåkingsverktøy og fra andre kanaler. Hendelser fra overvåkningsverktøy er typisk tekniske problemer og/eller maskinvare, mens hendelser fra andre kanaler er mest typisk problemer med programvare eller i forbindelse med veiledning.	Registrere om hendelsen kommer fra email, tlf, personlig henvendelse eller overvåkingsverktøy
	Avsender	For kommunisering	Person_id til avsender
	Mottaker	Mottaker skal ha ansvar til den er løst.	Person_id til mottaker.
	Reg_tid	Utføre statistisk, ha oversikt over når ulike hendelser opptrer. Hjelp til å planlegge arbeidskapasitet.	Tidspunktet hendelsen ankom på
	Beskrivelse		

	Loggnr	For å identifisere hendelser	Dette er hendelsens unike id
	Status	For å følge med hvordan hendelser bli håndtert.	Forslag domene (Ny, lukket, under behandling, løst,...)
	Type	Hendelsen blir registrert etter definerte kategorier.	Forslag domene: (Maskinvare, programvare, dokumentforespørsel)
	Involverte_SLA		
	Berørte_tjenester	For å få gitt prioritet. Hvis det er mange tjenester som blir berørt, eller det er en viktig tjeneste som er ned, må hendelsen ha høy prioritet.	
	Prioritet		
	Problem	Finn ut om hendelsen tilhører et definert problem.	
	RFC	Finn ut hvilke(n) RFC som eventuelt skapte følgen(e) av hendelsen	
	Involverte_CI	Her skal alle relaterte CI plukkes ut, men i brukergrensesnittet kan man vise de aktuelle CI man behøver å vite om, eller eventuelt vise i flere trinn, ikke alle på en gang.	Liste av alle involverte Konfigurasjonsenheter (CI) fra CMDB, både maskinvare, programvare, dokumentasjon, brukere etc.
	Okkupert_tid		Tiden tjenester blir okkupert pga. hendelsen
	Løsnings_tid		Tid som brukes for å løse hendelsen
	Lukket_tid		Tid( dato/klokkeslett )
Eskalering	Eskalerings_tid		Registreres hver gang eskalering skjer

	Avdelings_id		Hvilken avdeling eskaleringen har kommet til
	Ansvarlig	Person med ansvar for hendelsen i gitt avdeling	registrere person_id til ansvarlig
Brukere (fra AD database; AD: Ansatt Database)	Kontornr		
	Tlf		
	Reg_tid	Forandringstid	Ved å vite når status er forandret, kan man følge saken bedre.
	Status		
	ansvarlig (Person_id)	Hvis hendelsen blir eskalert til høyerelinje support, så personen som forandrer status er vanligvis den som behandler hendelsen etter eskalering.	
Status	Kommentar		Beskrive kort om status til hendelsen.
	Loggnr		
	Grupper	Navn til konsulent firma	
	Navn	Navn til den/de som har ansvar for hendelsen, eller kontaktperson	
	Kontakt (tlf/email/mobil)		
Konsulent (For ekstern konsulent-support om nødvendig)	Start_tid		
	Slutt_tid		
	Tidsforbruk		Her registreres tid som er brukt for å løse hendelsen
	Løsnings_id		
	Beskrivelse		

	RFC_lenket		
Løsning	Løsnings_tid	Registrere total tid som er brukt for å løse problem	

### 9.3.2 Attributter for problemadministrasjon

Attributter	Sub-attributter	Hvordan attributt brukes	Merknad
Problem	Problemnr	Problem identifisering	
	Regdato	Følge opp hvor lenge et problem kan tåles uten ordentlig løsning, eller for diskusjon med leverandør hvis tjenester er innkjøpt	Første gang problem definert
	Kilde		Reaktiv: problme definert etter henvendelse fra brukere  - Proaktiv:problem definert etter informasjon fra overvåkningsverktøy.
	Beskrivelse		
	Årsak type	Årsak skal kategorisere som hardware, software...	
	Prioritet		
	CI_id	Finn ut alle berørte konfigurasjon komponent	Liste id til alle berørte CI
	Incident_id	Forbundet incident	Knytte til alle hendelse som involverte i problem
	Status	Nåværende status	
	RFC_id	Registrert alle RFC som er involvert	Et problem kan ha flere RFC



	Arbeidslogg	Med arbeidslogg kan man har en rask oversikt over problem og hvordan det har blitt behandlet.	Registrere kort hva man gjøres med problem hver gang saken åpnes.
	Undersøkelse dokument		
Work-around	Instruksjon	Beskriver hvordan problem behandles	
	Referenser	Link til referanse dokumenter	
	Ressurs	Service support person/grupper som jobber med saken	
Status historie	Tid ( dato, time)	tiden siste status endres	
	Person_id	Registrere person som forandre status	
	Kommentar		Status beskrivelse
Ansvarsgruppe	Person_id	Navn til alle medlemmer i gruppen	
	Person_id	Problems styring skal registrert	
	Start_tid	Registrering start og slutt tid gruppe brukes til problem.  Et problem kan ha flere ansvarsgrupper jobber parallell eller etter hverandre	
	Slutt_tid		
Løsning	Løsning_id		
	Beskrivelse		
	RFC lenket		

	Løsningstid	Registrere total tid brukes for å løse problem	
--	-------------	--	--

### 9.3.3 Attributter for endringsadministrasjon

ATTRIBUTTE R	SUB-ATTRIBUTTER	HVORDAN ATTRIBUTTET BRUKES	MERKNAD
Forandring	Forandrings_id	For å skille mellom forandringer som blir foretatt flere ganger eller ulike forandringsmodus av en RFC.	
	For_RFC	Fremmednøkkel til RFC-tabellen. Den RFC som forandringen gjelder for.	Dette er hoved-RFC for forandringen
	Linket_RFC	Eventuelle andre RFC som endringen gjelder for.	Er det noen andre RFC som har relasjon (forelder/barn) til "hovedforandringen"
	Type	Ved å dele forandring i kategori, kan man velge riktig arbeidsplan, kostnader og ressurs.	
	Beskrivelse	Tekstlig beskrivelse av forandringen	
	Linket_hendelse	For å finne ut mer informasjon om konsekvenser av forandringen	
	Linket_problem	For å oppdatere i problemadm. om permanent løsning og lukke hovedårsak.	
	Hovedforandrings_CI	Denne kan man finne frem til ved hjelp av sql-spørringer.	Registrert hoved konfigurasjonsenh et som er gjenstand for forandring

	Involverte_CI	For å vite hvilke andre konfigurasjons-enheter som er involvert i forbindelse med forandringen.	
	Andre konsekvenser	Andre konsekvenser av forandring, f.eks arbeidsrutiner som må endres, behov for opplæring, personal-endringer, og/eller konsekvenser i sin helhet som må registreres som informasjon til berørt enhet, slik at planleggingen kan bli foretatt på riktig grunnlag.	
	Prioritet	Endringens tildelte prioritet etter analyse av konsekvenser.	
	Antatt tid nødvendig for forandringsutførelsen	Foreløpig planlagt tidsangivelse. Gjenstand for endringer underveis.	
	Faktisk tidsforbruk	Hva resultatet ble, når det gjelder tidsforbruk.	
	Rapport	Tekstlig rapport om hele forandringens livsløp.	
	Kostnader	Kostnader i forbindelse med forandringen, i form av såvel økonomi, tid, og andre ressurser.	
Avgjørelsesgrupper	Navn	Skal registrere navn på den/de som har avgjørelsesmyndighet i de aktuelle gruppene.	
Avgjørelse	Lenket til avgjørelsesdokumenter	Rapport som beskriver vurderingen av hoved-RFC, basert på konsekvensanalyser.	Avgjørelse kan være godkjenning, avlyst/suspendert eller forandret RFC
Utførelsesgrupper	Navn	Navn på alle personer i gruppene.	
	Leder	Navn på hovedansvarlig i gruppene	
Utførelse	Ressurs	Beskrivelse av hvilke ressurser som går med til selve utførelsen.	

	Dokumentasjon	Dokumenter nødvendig for utførelsen av endringen(e).	All relatert dokumentasjon, både Service Desk, teknisk dokumentasjon, programvare-dokumentasjon etc.
	Test	Registrering av tid og sted for testen, samt produksjon av en rapport for testen.	
	Versjon	Registrere ny versjon hvis man bruker versjonshåndtering.	Hvordan man håndterer versjon i forandringsprosessen.
	Start_tid	Start for utførelse av endringen.	
	Avsluttet_tid	Tidspunktet for ferdig utført endring.	
	Arbeidsplan/ arbeidsprosedyre	Arbeidsplan for utførelsen.	

## 9.4 Annex 7C: Suggested CI attributes

The following attributes are examples that could be used in the [CMDB](#). Note that hardware [CI](#) types will have different attributes from software [CI](#) types. <<<

Attribute	Description
<a href="#">CI</a> Name	The unique name by which this type of <a href="#">CI</a> is known.
Copy or Serial Number	The number that uniquely identifies the particular instances of this <a href="#">CI</a> - for example, for software the copy number, for hardware the serial number.
<a href="#">Category</a>	<a href="#">Classification</a> of a <a href="#">CI</a> (e.g. hardware, software, documentation etc).
Type	Description of <a href="#">CI</a> type, amplifying 'category' information (e.g. hardware configuration, software package, hardware device or program module).
Model Number (hardware)	Model of <a href="#">CI</a> (corresponding, for example, to supplier's model number e.g. Dell model xxx, <a href="#">PC</a> /aa model yyy).
Warranty expiry date	Date when the supplier's warranty expires for the <a href="#">CI</a> .
<a href="#">Version</a> Number	The version number of the <a href="#">CI</a> .
Location	The location of the <a href="#">CI</a> , e.g. the library or media where the software <a href="#">CIs</a> reside, the site/room where a service is located.
OwnerResponsible	The name and/or designation of the owner responsible for the <a href="#">CI</a> .

Responsibility Date	Date the above owner became responsible for the <a href="#">CI</a> .
Source/supplier	The source of the <a href="#">CI</a> , e.g. developed in-house, bought in from companyxxxxx etc.
Licence	Licence number or reference to licence agreement.
Supply Date	Date when the <a href="#">CI</a> was supplied to the organisation.
Accepted Date	Date when the <a href="#">CI</a> was accepted by the organisation as satisfactorily tested.
Status (current)	The current status of the <a href="#">CI</a> ; e.g. under 'test', 'live', 'archived'.
Status (scheduled)	The next scheduled status of the <a href="#">CI</a> (with the date or indication of the event that will trigger the status change).
Parent <a href="#">CI</a> (s) relationships	The unique <a href="#">CI</a> identifier(s) - name/copy/number/model/number/ of the 'parent(s)' of this <a href="#">CI</a> .
Child <a href="#">CI</a> (s) relationships	The unique <a href="#">CI</a> identifier(s) of all 'children' of this <a href="#">CI</a> .
Relationships	The relationship of the <a href="#">CI</a> with all <a href="#">CI</a> s other than 'parent' and 'child' (e.g. this <a href="#">CI</a> 'uses' another <a href="#">CI</a> , this <a href="#">CI</a> 'is connected to' another <a href="#">CI</a> , this <a href="#">CI</a> is 'resident on' another <a href="#">CI</a> , this <a href="#">CI</a> 'can access' another <a href="#">CI</a> ).
<a href="#">RFC</a> Numbers	The identification number of all <a href="#">RFC</a> s affecting this <a href="#">CI</a> .

Change Numbers The identification number of all [Change records](#) affecting this [CI](#).

---

[Problem](#)  
Numbers The identification number of all [Problem](#) records affecting this [CI](#).

---

[Hendelse](#)  
Numbers The identification number of all [Hendelse](#) records affecting this [CI](#).

---

Comment A comment field to be used for textual narrative; for example, to provide a description of how this version of the [CI](#) is different from the previous version.

---

For [RFCs](#), [Change records](#), package [Release](#) records, etc, the names, copy numbers, model numbers and version numbers of [CIs](#) affected by the Change, and how they are affected, should be recorded in the [CMDB](#). A reversion path, and the consequences of reversion, should also be recorded.

For RFCs, forandringsprotokoller emballer Utløserprotokoller, osv, navnene, kopiantall, modellerer antall og versjonsantall CIs berørt ved Forandringen, og hvordan de berører, lagrer i CMDB. En reversion sti, og konsekvensene av reversion, lagrer også.

## 10 Referanse

1. Skattedirektoratet; *SKD Mandat for "Dokumentasjon - Infra"*, SKD\_intranett,
2. Bekk\_Consulent\_AS, *Hvordan få maksimalt ut av IKT ressursene?*, in *Frokost seminar: Vippetangen*.
3. Ole Hvidkjær (2004) *Risk Management i praksis*. DataZ Norway Chapter, 4
4. Arnesen Jørn (2004) *Svaret er risikohåndtering*. DataZ Norway Chapter,
5. Philip L. Campell (2005) *A Cobit Primer*.
6. Jean-Pierre Garbani, Laura Koetzle, and Thomas Powell. *The Management Process Alphabet Soup*. Analyst Corner [cited 21.11.05]; Available from: <http://www2.cio.com/analyst/report3888.html>.
7. *Introduction to the Service CMM* The IT Service Capability Maturity Model [cited 23.11.2005]; Available from: <http://www.itservicecmm.org/>.
8. SYSCOM\_AS, *ITIL i praksis*, in *Frokost seminar*: Ernst & Young.
9. Colin Rudd. *A high-level overview of the IT Infrastructure Library*. [cited 2.2005].
10. Richard Jarrell Françoise Tourniaire, *The Art of software support. Design & Operation of support centers and help desks*. 1997: Prentice Hall PTR.
11. Tendo\_Tech\_AS. *Hva er tjenesterovervåking*. [cited 1.2005]; Available from: <http://www.tendotech.no/index.php?id=66>.
12. Vidar Faltinsen; *Nettadministrasjonsverktøy*, Markedoversikt og Anbefaling 2,
13. Arne B. Mikalsen & Per Borgensen, *Drift av lokalnettverk Design og sikkerhet*. 5.utgave.
14. Thor E. Hasle, *Nettverks-administrasjon*. 2.utgave.
15. IBM\_Corp. *Components of a Tivoli monitoring environment*. IBM Software 2003 [cited 11.1.2005]; Available from: <http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=/com.ibm.tivoli.wiwuser.doc/wasugmst21.htm>.
16. Roy Allan Hansen. *Fra overvåking til beskyttelse*. 16.11.2004 [cited 2.2005]; Available from: <http://nettverk.idg.no/index.cfm?fuseaction=artikkel&id=411E17E7-BFA0-F1ED-6B9CFA863964A2D9>.
17. Jan van og al. Editor:Bon, *IT Service Management, an introduction based on ITIL*, ed. second edition First impression. 2004: Van Haren Publishing.
18. Brian Johnson of Pink Elephant, ed. *CD ROM Best practice Service Support*. 2002, 2002.
19. Zaventem. *Configuration Management*. An Econocom Group Company 2003 [cited 20/10/2005]; Available from: [http://www.itsmf.be/itsmfnewsite/events/2003/RTM19062003/Configuration%20Management\\_ITSMf%20190603%20d1.1.ppt](http://www.itsmf.be/itsmfnewsite/events/2003/RTM19062003/Configuration%20Management_ITSMf%20190603%20d1.1.ppt).
20. Skattedirektoratet; *Skatteetatens organisering og IT-løsninger*, SKD\_intranett,
21. SKD\_internett. *Skatteetaten* 2006 [cited 1/2006]; Available from:



- <http://bedriftsprofil.monster.no/skatteetatenno/>.
22. Skattedirektoratet; *Skatteetaten* SKD\_intranett,
  23. SKDs\_Intranett, *Software for drift av Skatteetatens tele- og datanett*, Software for drift av Skatteetatens tele- og datanett.vsd, Editor.
  24. Skattedirektoratet; *Oversikt over Skatteetatens infrastruktur*, SKD\_intranett, 1/2006
  25. Skattedirektoratet; *Utdrag fra VEDL- A- Skatteetatens Organisering og IT løsning*, SKD\_intranett,
  26. Hanna Meland, *i Skattedirektorat, om InfoChanell grensesnitt*.
  27. Skattedirektoratet. *Info Chanell*. SKD\_intranett [cited; Available from: <http://x00info/>].
  28. Skattedirektoratet; *Ligningsfaglig forandring av Modernisering av IT-Drift* SKD\_intranett,
  29. Den Norske Dataforening. *Om SLA*. [cited 13.10.2005 ]; Available from: <http://dataforeningen.no/?module=Articles;action=ArticleFolder.publicOpenFolder;ID=1029>.
  30. Knut Fiskerud, *IT teknisk avdeling i Skattedirektoratet*.
  31. Per Åge Gabrielsen, *IT teknisk avdeling i Skattedirektoratet*.
  32. Karen Marie Reinsnes, *Brukerkontakt avdeling i Skattedirektoratet*.
  33. Ragnar Normann. *Dataorientert modellering*. 2006 [cited 31.3.2006]; Available from: <http://www.uio.no/studier/emner/matnat/ifi/INF2120/v05/undervisningsmateriale/INF2120-NIAM.pdf>.
  34. Marina Angell (2005) *Informasjon fra sluttbruker- analyse av Skatteetatens brukerstøtte, og utnyttelse av brukerstøtteverktøy*