IT-stöd för säljande servicetekniker

Integration genom tjänster

Innehållsförteckning

[Inledning 3](#_Toc346821175)

[Bakgrund 3](#_Toc346821176)

[Problemet 4](#_Toc346821177)

[Syfte 4](#_Toc346821178)

[Målgrupp 5](#_Toc346821179)

[Avgränsning 5](#_Toc346821180)

[Metod 5](#_Toc346821181)

[Problembeskrivning 5](#_Toc346821182)

[Arkitekturansats 6](#_Toc346821183)

[Säljprocessen 6](#_Toc346821184)

[Begreppsmodell 7](#_Toc346821185)

[Systemförteckning 8](#_Toc346821186)

[Problembeskrivning 9](#_Toc346821187)

[Om enheters identitet 9](#_Toc346821188)

[Krav på IT-stödet 9](#_Toc346821189)

[Om verksamhetsobjekt och beroenden 10](#_Toc346821190)

[Problemområden kondenserade 10](#_Toc346821191)

[Arkitekturansats 11](#_Toc346821192)

[Användningsfall - övergripande 11](#_Toc346821193)

[Användningsfall ERP-fasad : Registrera Arbetsorder 13](#_Toc346821194)

[Användningsfall Process : Registrera Larmkonfiguration 14](#_Toc346821195)

[Användningsfall Process : Testa larminstallation 14](#_Toc346821196)

[Användningsfall Process : Lagra säljunderlag 15](#_Toc346821197)

[Logisk vy 16](#_Toc346821198)

[Klassmodell – Beroenden 16](#_Toc346821199)

[17](#_Toc346821200)

[Slutsatser 18](#_Toc346821201)

[Diskussion 18](#_Toc346821202)

[Referenser 20](#_Toc346821203)

# Inledning

## Bakgrund

Det har sedan en tid blivit vanligare att företag och organisationer gör satsningar för att röra sig bort från komponentbaserade lösningar för att fokusera mer på att ett tjänstebaserat synsätt på sin IT-flora. De tidigare komponentbaserade lösningarna har ofta haft hög detaljupplösning för att tillgodose nichade behov medan de tjänstebaserade gränssnitten har blivit vidare och mer generaliserade för att kunna tillgodose ett större antal klienter i mer heterogena sammanhang. Då dessa tjänster ofta utgör vissa kärnprocesser i organisationen är det inte ovanligt att de bygger på ett flertal underliggande komponenter och tjänster som arbetar tillsammans för att leverera värde. Detta samarbete ställer höga krav på detaljinkapsling och tydliga avgränsningar mellan komponenterna de implementerande komponenterna.

[LARM AB] är ett internationellt bolag inom säkerhetsbranschen och erbjuder vakter och säkerhetslösningar för både företag och privatpersoner. De har gjort en stor satsning på den privata marknaden genom hemförsäljning där larminstallatörer besöker privatpersoner i deras hem och utför både själva installationen av larmutrustningen samt genomför den slutgiltiga affären.

Tidigare har säljprocessen till stor del skett manuellt genom att installatörerna fört bok över de komponenter som installerats hos den nyblivna kunden samt de tjänstetillval denna gjort vid installationstillfället. Detta underlag har sedan lämnats över till den administrativa personalen på [LARM AB] som slutfört affären formellt. I den administrativa behandlingen av säljet var det inte ovanligt att personalen upptäckte att enskilda installatörers lagersaldon skiljde sig mot bokförda volymer och att vissa installationer/sälj avvek från normen med avseende på vilka komponenter som blivit sålda/inkluderade i installationen. Företagets verksamhetsutvecklare kunde efter en genomlysning av sälj- och lagerprocesserna slå fast att enskilda installatörer ofta ”lånade” komponenter från varandras lager när deras egna lagernivåer blev låga utan att rapportera tillbaka till administrationen, men också att det ibland helt saknades komponenter i lagrena, varpå dessa aldrig ens erbjöds kunderna.

[LARM AB] kom fram till att ett tjänstebaserat IT-stöd för hela säljprocessen skulle implementeras och genom detta skulle larminstallatörerna få bättre möjligheter att sköta den egna lagerhållningen; vilket både skulle underlätta det administrativa uppföljningsarbetet och dessutom uppmuntra installatörerna att arbeta tillsammans med varandra med en gemensam lagerhållning och på så vis sälja större installationer till kunderna.

För att realisera detta IT-stöd bestämde sig [LARM AB] för att utveckla en mobilapplikation för installatörernas sälj- och installationsarbete, samt en samordnande tjänst för att automatisera och koordinera de tjänster och verksamhetssystem som implementerar de administrativa funktionerna som är förknippade med försäljningen. Det högst prioriterade kravet på lösningen var att inget sälj skulle gå förlorat och att hela säljet så långt som möjligt skulle genomföras vid ett besökstillfälle, utan kompletterande besök och installationer. Dessutom fanns höga krav på att de mobila applikationerna skulle vara feltoleranta och ha korta svarstider för att göra försäljningsarbetet så smidigt som möjligt.

Alla [LARM AB]s underliggande IT-system erbjöd integrationsmöjligheter i viss mån. Dock visade det sig att företagets ERP-system rotat sig så pass djupt i organisationen att dess tekniska egenskaper starkt influerat det sätt säljverksamheten bedrevs på. Ett faktum som påverkade integrationsmöjligheterna med övriga system och i förlängningen även möjligheterna för det nya tjänstebaserade IT-stödet att till fullo stödja det säljförfarande som organisationen önskade sig.

## Problemet

Hur bör en samordnande tjänst som realiserar IT-stödet för den komplexa säljprocessen implementeras övergripande för att säkerställa att [LARM AB] inte tappar ett sälj, hur bör denna tjänst verka för att isolera användarna från potentiella avbrott i underliggande system och tjänster; och hur bör dessa implementeras för att isoleras ifrån läckande implementationsdetaljer från andra verksamhetskomponenter?

## Syfte

Syftet med denna uppsats är att belysa vilka problem det innebär att integrera ett antal system, med varierande grad av direkt verksamheteninverkan, för att realisera ett IT-stöd för en hel verksamhetsprocess; samt presentera ett antal arkitektoniska lösningsförslag som överkommer dessa problem i allmänhet samt kapslar in och motverkar [LARM AB]s ERP-systems bieffekter i synnerhet.

## Målgrupp

Uppsatsen skrivs som en del av examinationen för Dataföreningens kurs ”Certifierad IT-arkitekt” omgång 45. Den tänkta målgruppen är aspirerande IT-arkitekter i allmänhet och integratörer i synnerhet.

För att tillgodogöra sig denna uppsats på ett lämpligt sätt bör läsaren ha grundläggande förståelse för UML, detta är dock inget krav då de flesta modellerna även beskrivs med text.

En viss kännedom om mönster för integration och systemutveckling i distribuerade miljöer kan vara till hjälp för att snabbt sätta sig in i resonemang. Mönstrena kommer även i förekommande fall redovisas i referenslistan för eventuell inläsning.

## Avgränsning

Uppsatsens fokus ligger på konceptuell och logisk nivå och ämnar endast beskriva lösningar på dessa nivåer. Teorier som presenteras i denna uppsats kommer baseras på vedertagna mönster och resonemang, dessa teorier kommer inte nödvändigtvis verifieras mot en teknisk implementation.

Då målet med uppsatsen är att belysa arkitektoniska detaljer kring integration, kommer kringliggande applikationer och system endast beröras i den mån det tillför något till uppsatsens resonemang. Uppsatsen kommer inte heller föreslå några förändringar i, eller byten av, system eller applikationer även om det kan verka lämpligt för att uppnå uppsatsens syfte.

## Metod

Uppsatsen skrivs på svenska men innehåller en del engelska ord och uttryck där direkta översättningar inte är möjliga eller skapar förvirring. Dessa uttryck beskrivas utförligare i en ordlista.

Jag har valt att dela upp uppsatsen i två huvudsakliga avsnitt, där det första är en mer detaljerad problembeskrivning för att formulera de krav och förutsättningar som ligger till grund för den tänkta arkitekturen och knyta an dessa mot uppsatsens problemformulering. I den andra delen presenteras ett arktitekturförslag som en potentiell lösning anpassad efter problembeskrivningen.

### Problembeskrivning

Problembeskrivningen utgörs dels av de funktionella och icke-funktionellla krav som [LARM AB] ställt på IT-stöden men även till stor del av en genomlysning av organisationens befintliga IT-stöd som är av intresse för uppsatsen. Detta avsnitt kommer alltså utöver de uttalade kraven även fokusera på att lyfta fram de egenskaper i den befintliga miljön som är arkitekturdrivande

### Arkitekturansats

Uppsatsen utgår ifrån Kuchtens 4+1 modell för att beskriva arkitekturen för den lösning som föreslås. Modellen bygger på dessa fem vyer

* **Scenarier** – i denna vy beskrivs interaktionen mellan användare/system, mellan processer och komponenter för att belysa de arkitekturdrivande användningsfallen.
* **Logisk vy** – I denna vy ligger fokus på att beskriva systemets funktioner ur slutanvändarens perspektiv
* **Utvecklingsvy** – beskrivningar anpassade för utvecklare och programmerare. Fokus ligger på beskriva strukturer som klasser och komponenter
* En **processvy** där fokus ligger på systemets beteende under exekvering. Exempelvis för att beskriva parallellitet och hur trådar och processer kommunicerar med varandra
* **Fysisk vy** – beskriver hur systemet installeras/distribueras på olika fysiska noder och miljöer. Beskriver också hur olika delar i systemet är sammanlänkade fysiskt

Varje vy kan sedan brytas ned med finare upplösning i nivåerna *övergripande, logisk* och *fysisk.*

Eftersom det inte är uppsatsens syfte att vara en komplett arkitekturbeskrivning så kommer endast de vyer och de nivåer som anses tillföra något att användas. I vyerna kommer arkitekturen beskrivas på en avslappnad form av UML så långt detta är möjligt och relevant.

# Säljprocessen

[LARM AB] bedriver traditionellt försäljarbete för att bygga upp en bas av potentiella kunder. En stor del av företagets kunder har tillkommit genom säljarbete på säkerhetsmässor, reklam i media men även genom att kunden själv tagit kontakt med bolaget. Efter att en kontakt med en potentiell kund upprättats, lagras dennes kontaktuppgifter i [LARM AB]s CRM-system och ett nytt säljteam tar vid för att försöka boka in möten med kunden och en av företagets installationstekniker för att diskutera en skräddarsydd säkerhetslösning. Mötet äger rum i kundens hem (eller fastighet/lokaler i de då kunden inte är en privatperson) och målsättningen är att teknikern ska installera larmsystemet och slutföra försäljningen på plats.

Det faktiska säljet som [LARM AB] avser att implementera IT-stöd för inleds med att teknikern åker ut till kunden för att genomföra installation och konfiguration av säkerhetslösningen. Teknikerns bil fungerar som lager för det material som behövs för att utföra installationen och i de fall vissa komponenter saknas kan teknikern komplettera sitt lager genom ett lageruttag från en kollegas bil.

Väl på plats installerar teknikern en central larmenhet och till denna kopplas ett antal periferienheter (rörelsedetektorer, kameror, röksensorer) som tillsammans utgör själva larminstallationen. Den centrala enheten förses även ett så kallat PROM-nummer för att kunna identifieras unikt, och ett GSM-chip för att den ska kunna kommunicera med [LARM AB]s centrala larmövervakningssystem.

När enheten är installerad rapporterar teknikern in kund- och enhetuppgifter till [LARM AB] och larmutrustningen sätts sedan i ett testläge, där installation och konfiguration verifieras mot det centrala övervakningssystemet.

Efter att testet genomförts sammanställer teknikern materialåtgång och eventuella lageruttag i en arbetsorder som registreras hos [LARM AB]. Därefter erbjuds kunden teckna diverse tilläggstjänster via abonnemang och får slutligen signera ett kontrakt. Teknikern registrerar slutligen kontraktet tillsammans med en kundorder hos [LARM AB] och säljet ansers i och med detta vara genomfört.

## Begreppsmodell

Ur processbeskrivningen ovan kan ett antal begrepp och informationsflöden utläsas, *figur1* ordnar in dessa begrepp i en begreppsmodell för att beskriva hur de hänger samman och hur de relaterar till varandra. Slutligen ges en övergripande beskrivning över de system som är involverade i processen och vilka begrepp som berörs av dem.



Figur 1

**Test**Avser det test som verifierar larminstallationen och utgör kvittens för en Arbetsorders uppfyllnad.

**Enhet**En Enhet är den centrala enheten i en larminstallation som [LARM AB] förhåller sig till med avseende på övervakning. Enheten identifieras dels genom det inprogrammerade PROM-numret, men också genom ett enhetsnummer. Enhetsnumret kan vara aktivt eller inaktivt, till exempel om kundens avtalsperiod gått ut. Enheten associeras också med ett GSM-nummer för kommunikation med larmcentralen.

**Kund**Kunden är den primära kontakten i händelse av larm och även den person eller organisation som anses vara betalningsansvarig för både abonnemang och den GSM-trafik som säkerhetslösningen genererar. Kunden identifieras genom ett referensnummer eller ett kundnummer

**Kontrakt**Ett Kontrakt reglerar de avtal och abonnemang kunden tecknat med [LARM AB].

**Kundorder**Kundordern specificerar de tilläggstjänster en kund har valt för en viss larminstallation och utgör underlag till den faktura kunden sänds för teknikerns arbete under installationstillfället.

**Arbetsorder**Arbetsordern specificerar de olika periferienheter utöver Enhet som ingår i en installation. Denna utgör en del av slutfaktureringen till kunden efter installationstillfället samt är underlag för teknikerns lön.

**Lageruttag**Avser det uttag en tekniker gör direkt från [LARM AB]s eller kollegas lager.

**Periferienhet**Den extrautrustning som utgör en del av hela säkerhetslösningen

Dessa begrepp har vissa inbördes beroenden som illustreras i *figur*

**

Figur 2 Beroenden

Från figuren kan vi utläsa att

* En **Enhet** är beroende av en Kund
* Ett **Test** är beroende av en Enhet och en Kund
* **Kontraktet** är beroende av ett Test, en Enhet, en Kundorder och en Kund
* **Kundorder** är beroende av en Kund
* Ett **Lageruttag** är beroende av en Arbetsorder
* **Arbetsordern** är beroende av en Enhet och ett Test

### Systemförteckning

Följande är en förteckning över de system som ingår i processen samt en beskrivning av de begrepp från modellen som systemen berör

**SAP**SAP utgör [LARM AB]s ERP-lösning och hanterar information om samtliga begrepp i modellen. SAP är det centrala systemet som det dagliga administrativa arbetet utgår ifrån i organisationen.

**Canonix**Larmövervakningssystemet är utvecklat internt och heter Canonix. Detta operativa system håller ordning på larmenheter och kontaktpersoner (typiskt Kund) i händelse av larm. Canonix är också det system som används för att testa och verifiera larmkonfigurationer. Systemet är en vital del i företagets leveranser och har därför nära noll neretid.

**Superoffice**[LARM AB]s CRM-lösning levereras genom Superoffice och det är genom detta system som tekniker erhåller information om potentiella kunder. Superoffice berörs inte direkt av tjänstelagret som ska utvecklas för säljprocessen, då det bara har en initial roll tills teknikern påbörjat arbetet ute hos kund. Däremot hänvisar referensnumret i Kundobjektet till motsvarande kundinformation i Superoffice.

**Handyman**Handyman är den mobilapplikation som används av tekniker för sälj- och installationsarbetet ute hos kunden. Det är denna applikation som kommer kommunicera med det tjänstelager som behandlas i denna uppsats. Handyman hanterar av förklarliga skäl information om samtliga begrepp i domänen.

# Problembeskrivning

I detta avsnitt redovisas de krav som organisationen ställt på IT-stödet samt de tekniska förutsättningar som har inflytande på arkitekturens förmåga att tillgodose dessa krav. Då uppsatsens fokus ligger på tjänstelagret mellan klienten och [LARM AB]s linjesystem kommer endast de krav med bäring på tjänstelagret att redovisas. Slutligen ställs kraven mot de tekniska förutsättningarna för att identifiera några problemområden som arkitekturen bör adressera i synnerhet.

## Om enheters identitet

Definitionen av det PROM-nummer som programmeras in i en Enhet ägs av [LARM AB], som själva utvecklat algoritmen för att generera det unika identifikationsnumret. Trots detta har Enhetsnummer blivit en de facto standard för identifiering av en larminstallation och det är detta nummer organisationen refererar till i kontrakt, kund- och arbetsorder och fakturor. Detta beror på att verksamhetens dagliga arbete sker i ERP-systemet, och det är detta system som äger denna definition.

Kopplingen mellan PROM-nummer och enhetsnummer upprätthålls i SAP och detta, tillsammans med status om huruvida enhetsnumret är aktivt eller inte, synkroniseras mot Canonix genom nattliga batchkörningar. På det viset kan Canonix koppla ett specifikt PROM-nummer mot ett motsvarande enhetsnummer samt inaktivera övervakning av enheter med inaktiva enhetsnummer.

SAP allokerar ett antal enhetsnummer i en pool och när en ny Enhet registreras reserveras ett enhetsnummer från denna pool. Om en kunds kontrakt av någon anledning avslutas, raderas dennes enhet från systemet och enhetsnumret som den associerats med återgår till SAPs interna pool igen. Det är dock inte ovanligt att enheten har inaktiverats innan kontraktet löper ut och när enhetsnumret återförs till poolen följer även aktiveringstatusen med. Detta medför att det finns en risk att nya enheter blir tilldelade ett inaktivt enhetsnummer vilket riskerar att inaktivera övervakningen av enheten om denna status hinner bli synkad med Canonix.

För att komma till bukt med detta har [LARM AB] ordnat så att SAP returnerar ett felmeddelande om en ny enhet lagras med ett inaktivt enhetsnummer. Enheten kommer kunna lagras, men inga andra entiteter (exempelvis Kund- och Arbetsorder) kommer kunna kopplas mot den så länge des enhetsnummer är inaktivt. I och med att det är SAP som tilldelar enhetsnumret kommer detta fel inte gå att upptäcka innan det sker, någon måste försöka lagra enhetsinformationen innan felet uppdagas. SAP tillhandahåller inte heller något API för att aktivera enhetsnumret externt, utan personal måste istället manuellt aktivera enhetsnumret och lagra enheten igen.

## Krav på IT-stödet

I huvudsak ska tjänstelagret tillgodose klientens behov av följande funktioner:

* Kunna rapportera in larmkonfiguration i form av kund- och enhetsuppgifter för registrering i företagets ERP-lösning samt det centrala larmövervakningssystemet
* Kunna initiera ett larmtest
* Kunna rapportera in säljunderlag i form avkontraktsinformation och materialåtgång till företagets ERP-system.

Problematiken kring återanvändning av enhetsnummer behöver också hanteras eftersom det har bäring på hela säljprocessen. I det fall en inrapporterad enhet blir tilldelad ett inaktivt enhetsnummer måste en administratör notifieras så att felet kan åtgärdas omgående.

Följande egenskapskrav har ställts på IT-stödet

* Återanvändbarhet  
  Funktionerna för att registrera kontrakt och materialåtgång måste gå att återanvända eftersom en del eftersälj och tekniskt underhåll av larmutrustningen kan komma att bli aktuellt efter att larmsystemet installerats.
* Möjlighet att utföra hela säljet med så få server-anrop som möjligt

Då mobilklienten är beroende av 3G-anslutning är det önskvärt att hålla nede datakommunikationen av både stabilitets- och kostnadsskäl

* Minimal klientpåverkan vid avbrott i linjesystem  
  Under normala kontorstider ska det vara möjligt att rapportera in säljet trots avbrott i ERP-system.

## Problemområden kondenserade

Genom problemanalysen kan vi härleda följande konkreta problem

* **Tjänsterna kräver tillstånd** för att koordinera mellan de tre operationerna
* **Generaliserade tjänster** krävs för att möjliggöra återanvändning. Tjänsterna för inrapportering av kontrakt/kundorder och arbetsorder/lageruttag måste hållas rena från processlogik relaterad till säljprocessen. Koordination måste läggas ovanpå dessa tjänster
* **Garanterat ”tjänsteavbrott”** i samband med återanvändning av enhetsnummer, något som Handyman måste skyddas ifrån.

Det är dessa utmaningar utöver de uttalade kraven som uppsatsen avser behandla i det kommande arkitekturavsnittet.

# Arkitekturansats

För att angripa problemställningen och beskriva en arkitektur som tillgodoser de krav [LARM AB] ställer på tjänstelagret har jag valt en ”utifrån och in” approach (top-down) där jag kommer utgå ifrån Handyman-användarens interaktion med tjänsterna. Arkitekturförslagen kommer utgå från en samling användningsfall som kommer vara utgångspunkt för ytterligare avsnitt där varje användningsfall/tjänst beskrivs mer detaljerat. I dessa avsnitt kommer även arkitekturvalen motiveras och stämmas av mot kravbilden som avhandlats i tidigare kapitel.

## Användningsfall – övergripande

Utifrån de funktionella kraven har tre övergripande användningsfall identifierats:

* Registrera Larmkonfiguration
* Testa Larminstallation
* Registrera Säljunderlag

Kravet på återanvändbarhet kräver dock att användningsfallet Registrera Säljunderlag bryts ned ytterligare för att exponera funktionaliteten för att registrera Kundorder och Arbetsorder. Figur 2 illustrerar alla identifierade användningsfall samt de beroenden som finns mellan dem. Efterföljande avsnitt beskriver användningsfallen i mer detalj.



Figur 3 – Övergipande användningsfall

### Användningsfall: Registrera Larmkonfiguration

Användningsfallet inleds med att inkommande data valideras, vid valideringsfel returneras en felkod och flödet avslutas, annars fortsätter flödet med att PROM-nummer och kontaktuppgifter registreras i Canonix. Därefter lagras Kund och Enhet i SAP som returnerar ett allokerat Enhetsnummer och en statuskod som anger om enhetsnumret är aktivt eller inte. Denna information lagras och flödet avslutas. Hela flödet illustreras i *Figur 6* nedan.

**

Figur 4 - Användningsfall Lagra Larmkonfiguration

### Användningsfall: Testa larminstallation

Flödet inleds med en valideringskontroll av inkommande data, om kontrollen misslyckas returneras ett felmeddelande och flödet avslutas. Annars sänds larmkonfigurationen till Canonix för verifiering och denna returnerar status för operationen samt ett verifikationsnummer, verifikationsnumret returneras till Handyman-klienten och flödet avslutas. Detta illustreras i *Figur 7*.



Figur 5 - Användningsfall Testa Larminstallation

### Användningsfall: Lagra säljunderlag

*Figur 8* illustrerar användningsfallet Lagra säljunderlag, som inleds med att inkommande data valideras och eventuellt avslutas med felkod om något valideringsfel inträffar. Om data är giltigt säkerställs att aktuellt enhetsnummer är aktivt och sedan registreras Arbetsorder och Kundorder innan flödet avslutas



Figur 6 - Andvändningsfall Lagra Säljunderlag

### Användningsfall: Registrera Arbetsorder

*Figur 5*  illustrerar flödet för att Registrera en Arbetsorder. Flödet startar genom att en Arbetsorder läses in och valideras. Vid valideringsfel returneras en felkod och flödet avslutas, om inga fel påträffas lagras Arbetsordern. Därefter extraheras Lageruttag som slutligen lagras och flödet avslutas.



Figur 7 - Användningsfall Registrera Arbetsorder

Användningsfallet Registrera Kundorder har utelämnats avsiktligt då det i princip är identiskt med flödet i Registrera Arbetsorder, bortsett från den information som användningsfallet hanterar.

## Logisk vy

Figur bla visar en konceptuell modell över tjänstelagret stödjer [LARM AB]s interna och externa Handyman-användare. Handyman-användaren kommunicerar med en orkestrerande tjänst som delegerar arbete vidare till en ERP-fasad som exponerar funktionalitet som är gemensam för de interna och externa användarna.



Figur 8 - Tjänstelagret i logisk vy, övergripande nivå



Figur 9 - Logisk vy, logisk nivå

### Tjänsten ERP-fasad

### Tjänsten Säljprocess

# Slutsatser

Uppsatsens syfte var att identifiera antal problemområden inom tjänsteorienterad systemintegration och relatera dessa till de krav som ställts på IT-stödet i en säljprocess. Vidare skulle jag presentera en arkitektur som överbryggar dessa problem för att kunna tillgodose kraven i största möjliga mån.

Arkitekturförslaget bemöter problemen med väletablerade mönster och svarar väl upp mot de krav som ställts på den, men gör det genom att tillföra komplexitet. Å ena sidan erbjuder arkitekturen

* Näst intill totalt löst kopplade klienter
* Näst intill totalt isolerade klienter med avseende på driftstörningar
* Ett opakt tjänstelager som totalt kapslar in komplexiteten i de bakomliggande systemen och döljer denna för konsumenter
* Nya, generaliserade tjänstegränssnitt som går att återanvända i andra scenarior än de som legat till grund för uppsatsen

Å andra sidan kommer dessa möjligheter till kostnader i form av

* Ökad komplexitet med ett flertal abstraktionslager som gör lösningen svårare att förstå
* Ett ökat underhållsarbete och ett mer ansträngt utrullningsscenario som en följd av distribution

Även om arkitekturen föreslår distribuering av tjänsterna i Process- och ERP-lagret finns det inget tekniskt hinder att köra dessa in-process, om än endast för säljprocessen. För att göra de generella ERP-tjänsterna tillgängliga utanför säljprocessen krävs givetvis en viss distribution, men genom att inte distribuera dem i normalfallet skulle kommunikationen i Process-tjänsterna förenklas avsevärt och även underlätta utrullning och underhåll. Det skulle också ge den goda effekten att de båda tjänstelagrena tillåts evolvera oberoende av varandra (något som i och för sig skapar ett större behov av governance).

Arkitekturförslaget har legat till grund för den tekniska implementation som idag körs av [LARM AB]. Lösningen har baserats på Biztalk 2009 och Windows Communication Foundation som ryggrad och i och med detta teknikval har en del koncept som diskuterats i uppsatsen helt levererats out-of-the-box genom dessa produkter. Produkterna tillför självklart andra former av komplexitet och nya kostnader, men med en stor del av den tekniska plattformen redan utvecklad kunde [LARM AB] ta systemet i drift fort och börja göra de kostandsbesparingar man hoppats på.

# Diskussion

Det blir mer och mer tydligt att SOA handlar mycket mer om Process och mycket mindre om Teknik än vad jag som tekniker tidigare föreställt mig. Det är svårt att undgå att komplexiteten i den process som beskrivits i uppsatsen är artificiell och en direkt följd av ett verksamhetssystem som tagit för stor plats i det dagliga arbetet i organisationen. Som leverantör av skräddarsydda lösningar är det här en intressant observation eftersom detta tyder på att det fortfarande finns ett visst motstånd till att göra en genomlysning av sina egna processer, och börja förändringsarbetet där istället för att blint följa de arbetsformer som den befintliga IT:n påtvingar oss. Detta trots att SOA nuförtiden börjar ses som något moget och nästan alldagligt; och trots den aktuella organisationens egna uttalade önskemål att bli mer tjänsteorienterade.

Viss självkritik är också på sin plats. I retrospekt kan man konstatera att en organisation som saknar väl avgränsade och väldefinierade tjänsteområden och verksamhetsobjekt inte är mogen för composite services. De svårigheter implementationsteamet hade med att definiera tjänstegränssnitten och de arkitektoniska beslut som krävdes för att leverera stödet utgjorde sammantaget en ganska god grund att ifrågasätta dessa önskemål från organisationen.

Å andra sidan levererades ett IT-stöd som löste de problem man ansåg sig ha, på ett av beställaren godtagbart sätt. Så ett akademiskt rättfärdigande för arkitekturen är irrelevant, sett ur arkitektens vypunkt.

# Referenser

1. **Kruchten, Philippe.** *Architectural Blueprints - The 4+1 view model of software architecture.* u.o. : IEE Software, 1995. IEE Software 6(1995)
2. **Buschmann, Frank, Henney, Kevlin och Schmidt, Douglas C.** Facade. *Pattern-Oriented Software Architecture Vol. 5, On patterns and pattern languages.* Chichester : John Wiley, 2007.
3. **Hohpe, Gregor, Woolf, Bobby.** Guaranteed delivery. *Enterprise Integration Patterns* [online] Hohpe, Woolf  
   http://www.eaipatterns.com/GuaranteedMessaging.html
4. **-.** Canonical Data Model. *Enterprise Integration Patterns* [online] Hohpe, Woolf  
   http://www.eaipatterns.com/CanonicalDataModel.html