


# RBS använder Microsofts SOA-vision för nästa generations butikssystem

## 1. Introduktion

IT-arkitektur innebär alltid kompromisser. Kompromisser kring flexibilitet, kostnader, snabbhet eller andra aspekter. Att göra dessa kompromisser på ett analytiskt och korrekt sätt är det som skiljer en bra arkitektur från en dålig. I den

här kundreferens förklaras hur RBS i samarbete med LafaData tänkte när företaget beslöt sig för att bygga en ny tjänstebaserad arkitektur eller service oriented architecture, SOA, för dess nästa generations butikssystem. 

## 2. Traditionellt butikssystem

RBS bygger butikssystem. Ett butikssystem är ett antal applikationer som används för att administrera artiklar och priser, hantera betalningar i kassor samt att lagra försäljningsinformation, bland annat. Det består både av hårdvarukomponenter som datorer, streckkodsläsare samt mjukvara för att köra systemet. Point-of-sale-system (Kassa) måste vara feltoleranta, vilket innebär att även om externa tjänster (till exempel lagersystemet) inte är tillgängliga, måste kassasystemet ändå kunna fånga försäljningar och hantera åtminstone kontant betalning så att verksamheten inte stannar.


**Bild 1** visar hur ett traditionellt butikssystem fungerar. Utgångspunkt är kvittot. SP (StorePos) är kassan eller kassor, so (Store Office) är ett lokalt BackOffice som står i butiken, ERP är det överliggande affärssystemet.

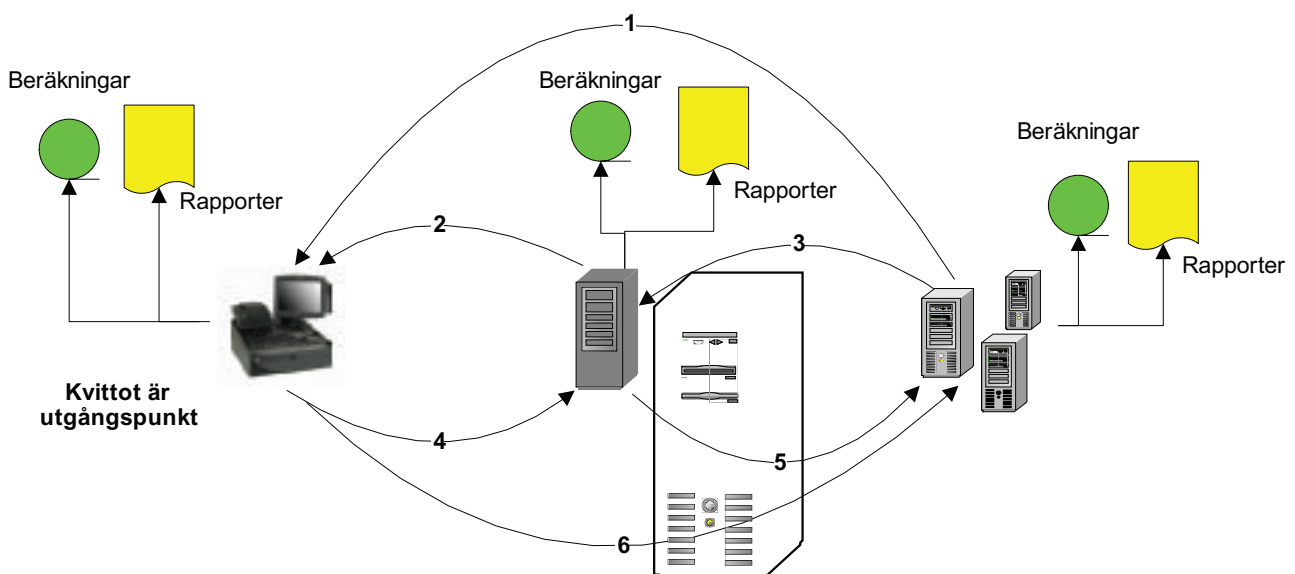
Det finns sex flöden att ta hänsyn till, beräkningar på minst tre ställen och rapportuttag i tre steg som hela tiden eskalerar i omfattning.

Moduler och delsystem är hårdkopplade till varandra vilket medför risker vid förändringar eftersom även andra moduler/system påverkas.

Det här systemet var modernt för ett par år sedan, kunderna är nöjda och RBS har tack vare butikssystemet blivit marknadsledande inom dagligvaruhandeln.

**FRÅGA:** Intressant, hur många transaktioner rör det sig om?

**SVAR:** Det är svårt att svara på exakt, men ungefär 100 miljarder kronor per år passerar igenom denna arkitektur. 



**Bild 1.** Ett traditionellt butikssystem. Visar flöden i en traditionell butikssystemlösning.

## 3. Nya krav


Detaljhandel som RBS varit specialiserad på har en inriktning på att behandla artiklar och kunder på ett likartat sätt. Systemen är inriktade på att kunden ska hanteras korrekt och framförallt snabbt. Köerna i en traditionell stormarknad för matvaror är ett exempel på behovet av snabbhet.

De nya krav som växt fram och som redan finns inom den specialiserade fackhandeln är att artiklar och kunder ska kunna hanteras unikt men med fortsatta krav på snabb hantering.

Artiklar ska kvitteras mot lager redan vid försäljningstillfället och kan ha sin egen identitet.

Kunder har ofta sina egna överenskomna prislistor och speciella erbjudanden via CRM-systemen.

I uppföljningen ställs krav på att artikel- och kundtyp ska kunna särredovisas, d v s överliggande system måste ges information om vad som var aktuellt vid försäljningstillfället.

Alla dessa nya krav ville vi bygga in på ett kostnadseffektivt och flexibelt sätt. 


## 4. Första skisserna


De första skisserna visade att RBS nått en återvändsgränd. Det gick inte utveckla system på samma sätt som tidigare.

– Vi behövde tänka i nya banor för att effektivt möta marknadens nya och ständigt förändrade krav på ett modernt butikssystem, säger Stefan Melander, arkitekt på RBS.

Att fortsätta sända ner all information som krävs för att hantera artikel och kund unikt i kassan skulle skapa stora problem. Hur ska systemet kunna hålla kontroll vilket pris, lageruppgift som är korrekt vid tillfället? Kan kassan lita på

att de lokala uppgifterna är korrekta och vad händer om uppdateringen mot centrala system visar på olikheter, felaktigheter gällande lokal information kontra den centrala?

 **FRÅGA:** OK, skulle ni kunna sammanfatta kort vad som ni betraktar som "återvändsgränden"?

**SVAR:** Det är mycket svårt att bygga distribuerade system där samma data finns på flera ställen. Det skapar en komplexitet som ökar vid nya funktioner och till slut blir hindrande. 

## 5. Kraven passar SOA

RBS kom fram till att de nya kraven passade in i SOA-tänket.

– Vi behöver utveckla system som redan från början är byggda för förändringar. Vi vill gå från att bygga funktioner till att stödja processer och bygga löst kopplade system för att minska komplexitet och beroende, säger Stefan Melander.

SOA handlar i botten om att stödja affärsprocesser, reducera komplexitet och få flexibla system och RBS såg flera möjligheter att arbeta mot dessa mål.

### Stödja affärsprocesser


Då RBS under många år fokuserat på affärsprocesser inom detaljhandeln hade företaget en unik möjlighet att arbeta top-down. Detta innebär att utgå från affären, fråga vad den behöver och identifiera IT-tjänster och funktionalitet som stöder huvudprocesserna. Genom att RBS har en förståelse för hela processen, kan företaget också undvika suboptimering i enskilda delar av processer. Det har också fördelen att känna till systemen i detalj vilket ger det en möjlighet att arbeta bottom-up och utnyttja funktionalitet som redan finns.


### Reducera komplexitet

Genom att använda självständiga tjänster ville RBS reducera beroendet mellan olika delar i systemet. Principen att fokusera på informationsutbyte mellan tjänster, så kallad "contract-first", skulle kunna ge företaget möjlighet att fokusera på enskilda tjänster och arbeta med dem självständigt utan att behöva bry sig så mycket om andra delar av systemet.

### Flexibla system

Idéerna i SOA om självständiga tjänster innebär för RBS att det kan uppgradera, addera och ta bort funktionalitet på ett mycket lättare sätt. Företaget kan också integrera mot andra system på ett lättare sätt om det har väldefinierade gränssnitt som bara bygger på öppna standarder.

 **FRÅGA:** Såg ni samtidigt några nackdelar med att använda SOA?

**SVAR:** Ja, de initiala kostnaderna kan vara högre första gången då det är ett delvis annat sätt att tänka. Dock är det kostnader som man får igen på ett lite längre perspektiv. 

## 6. Ny arkitektur

Det finns ett antal olika ramverk för arkitektur, t ex Zachman Framework, men RBS beslöt sig för att använda det ramverk med fyra views och fem aspekter som tagits fram av Microsofts Regional Directors i Sverige – 2xSundblad. Ramverket gav företaget en karta och hjälpte det med att kommunicera

idéerna till arkitekter, kunder samt företagsledning.

RBS har gjort en ordentlig analys på butikssystemet för att bryta ned systemet till autonoma tjänster. Det gällde att hitta rätt nivå på nedbrytningen. Till slut kom det fram till att det räckte med att ha en handfull tjänster som användar- och

koncernadministration samt kassatjänster. Sedan kunde företaget lätt utöka med nya tjänster som tillgodokvittohantering och presentkortshantering.

För att implementera dessa tjänster utnyttjade RBS mycket av det material som Microsoft har lagt ut på nätet för att stödja SOA-utvecklare. Företaget har använt designmönster, guider och applikationsblock från Microsoft patterns & practices, anammat idéerna kring fielfdoms och emissaries, utvecklarna har tittat på webcasts och har varit flitiga besökare på Microsofts Road Shows och andra Microsoftseminarier.

RBS bygger tjänsterna som flerskiktade servrar enligt Microsofts referensarkitektur och använder dess Web Services- och Transaction Services- designmönster. Som databärande använder företaget typade dataset rakt igenom från databasen (Microsoft SQL Server) ut till klienterna som är implementerade som rika klienter i Windows Forms. Typade dataset passar RBS perfekt eftersom de kan konsumeras av klienterna och det är lätt att omvandla dem till XML-dokument som externa kli-

enter och applikationer kan konsumera. Det är dock inte i alla situationer bäst om man har en väldigt blandad miljö.

Användargränssnitt byggs efter Microsoft P&A gruppens rekommendationer där användargränssnitt betraktas som en presentationstjänst. Även presentationstjänsterna är skiktade och består av User Interface- och User Process komponenter. Tack vare denna uppdelning gick det att kundanpassa dialogerna mycket lättare än förr.

Problem som upptäckts under resans gång var att hitta tjänster som verkligen är autonoma och varken är för stora eller för små. Andra problem var att det vid utvecklingens början inte fanns något annat att återanvända än verksamhetskunskap och erfarenhet från det gamla butikssystemet. Företaget ville inte trassla in sig i den gamla koden, så allt byggdes upp på nytt från början och det tog tid innan det gick att visa upp någonting som fungerade. Allt eftersom nya utvecklare tillkom i projektet fick de lära sig dem och det har också påverkat den initiala tiden negativt. ■

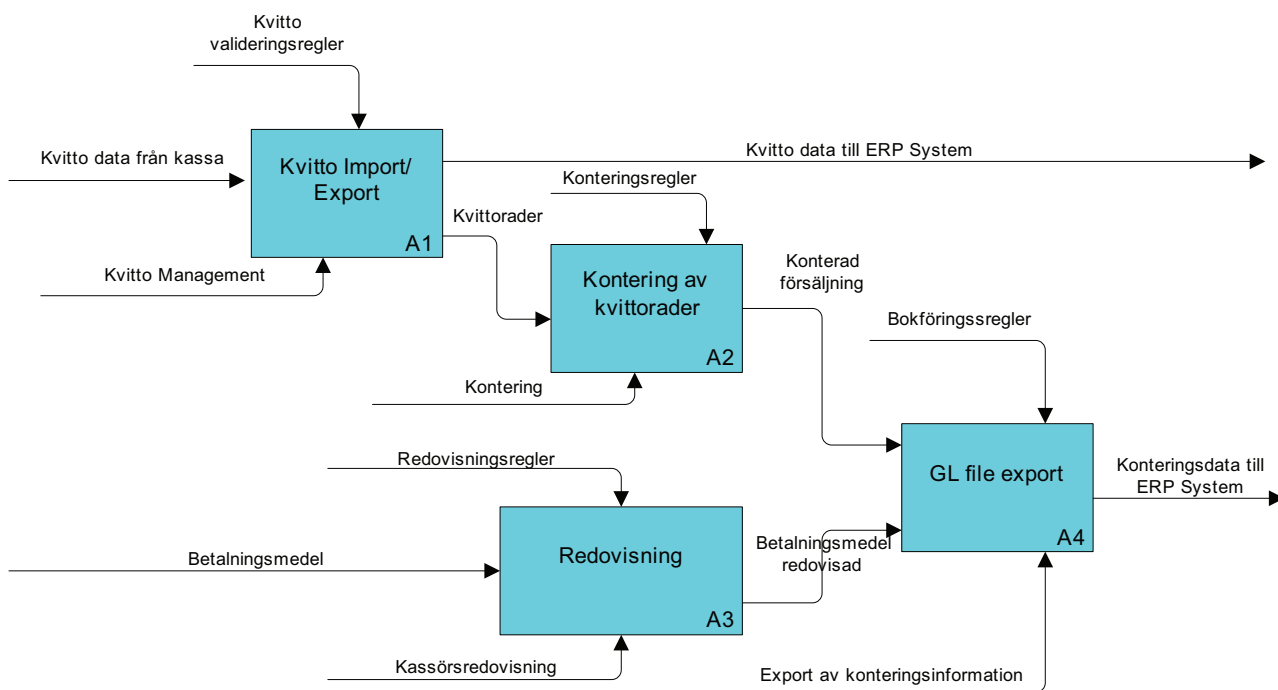


Bild 2. Processdiagram. Processhantering av kvittoinformation

## 7. Slutlig lösning

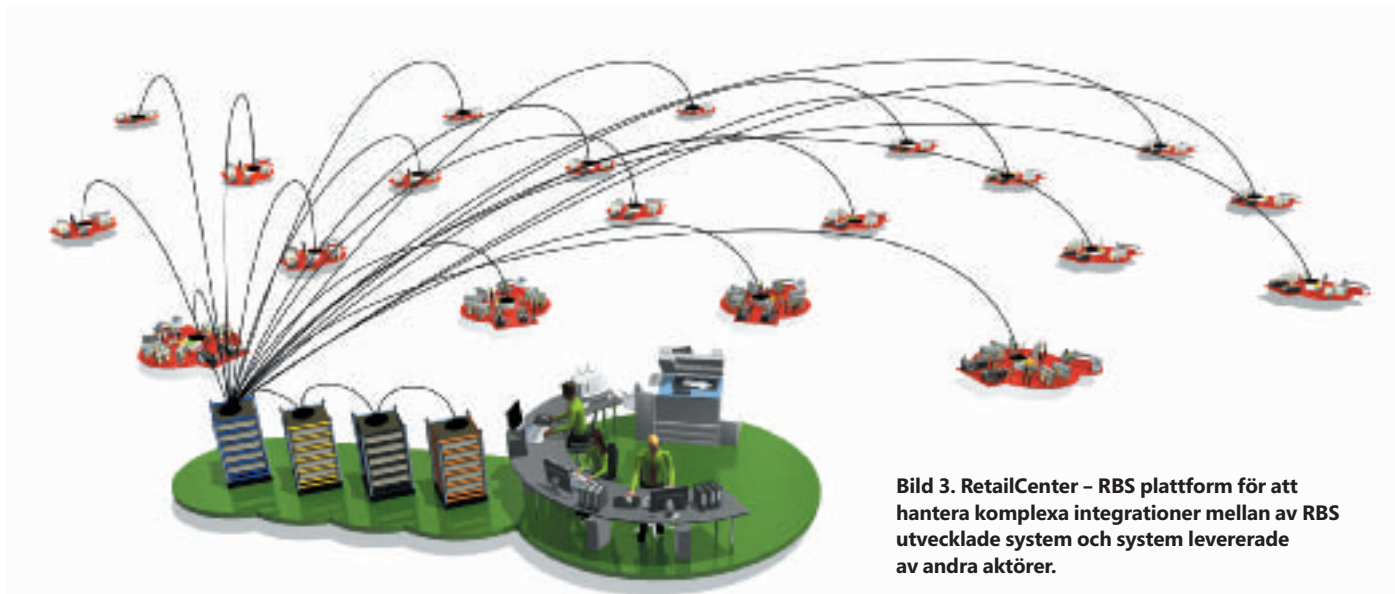
RBS har lärt sig mycket under resans gång och har skaffat nya nöjda kunder. Förutom att företaget nu har en mycket flexibel lösning lyckades det också med att integrera butikssystemet med sina kunders ERP-system online, exempelvis Oracle, PeopleSoft och SAP.

RBS har gått från att bygga funktioner till att stödja processer. Jämför gärna bilden om det traditionella butikssystemet med bild 2 ovan som är ett processdiagram över kvittohantering i RBS kommunikationslösning, RetailCenter (RC).

I och med att det nu finns ett väldefinierat gränssnitt mel-

lan kassa och RC i form av XML-scheman och Web Services blev lösningen transparent och andra leverantörers kassor kan nu kopplas till RBS RC. Det i sin tur kan öppna nya marknader för RBS där företaget kan erbjuda mervärde till nya kunder i form av processer och funktionalitet som deras nuvarande kassor och ERP-system inte klarar av.

RBS har byggt upp en ny utvecklingsprocess med hjälp av Microsoft Solution Framework och kan stödja produkten effektivt under hela dess livscykel både i förvaltning och i vidareutveckling.



**Bild 3. RetailCenter – RBS plattform för att hantera komplexa integrationer mellan av RBS utvecklade system och system levererade av andra aktörer.**

Tack vare att företaget har infört den nya SOA-arkitekturen och nya utvecklingsprocesser lyckades det öka produktiviteten avsevärt. Det tog mer tid att utveckla de första lösningarna jämfört med att jobba på det gamla sättet men den investeringen återbetalade sig ganska snabbt i förvaltning och vidareutveckling. RBS kan återanvända mycket kod och komponenter från den ena tjänsten till den andra och nu bygger företaget sina nya lösningar cirka 50 procent snabbare än tidigare. Det

blev allt lättare att sätta in nya resurser i projekten i och med att det finns lösningar som kan användas som referenser och utbildningsmaterial till nya medarbetare.

– För oss är det här en succé och vi är mycket glada att vi valde en service oriented architecture och Microsofts rekommendationer som plattform till en ny generations butikssystem, säger Stefan Melander. ☒

## Referenspersoner

### HAR DU FRÅGOR OM RBS SOA-INRIKTADE BUTIKSLÖSNINGAR, KONTAKTA:

#### **Stefan Melander**

Business arkitekt  
RBS – Retail Business System AB  
Mobile: +46 70 318 15 21  
Mail: stefan.melander@rbs.se  
[www.rbs.se](http://www.rbs.se)

### HAR DU FRÅGOR OM SOA OCH LÖSNINGSARKITEKTUR, KONTAKTA:

#### **Lajos Farkas**

Certifierad Microsoft .NET-arkitekt  
LafaData Applikationsarkitektur  
Mobile: + 46 70 728 01 47  
Mail: lajos.farkas@lafadata.se  
[www.lafadata.se](http://www.lafadata.se)

### HAR DU FRÅGOR KRING OMRÅDET "ERP MÖTER SOA", KONTAKTA:

#### **Annika Granebring**

Lic, Ph D Candidate at the National research school of Management and Information Technology  
School of Business  
Mälardalen University  
Office: + 46 21 103 175  
Mail: annika.granebring@mdh.se  
[www.mdh.se](http://www.mdh.se)

### FÖR ÖVRIGA FRÅGOR OM KUNDREFERENSEN, KONTAKTA:

#### **Daniel Akenine**

Architect Evangelist  
Developer Platform & Evangelism  
Microsoft Sweden  
Office: +46 8 752 3136  
Mobile: +46 73 408 3136  
Mail: daniel.akenine@microsoft.com  
[www.microsoft.se](http://www.microsoft.se)