

RBS and LafaData use Microsoft's SOA vision for the next generation of retail system

1. Introduction

IT architecture always means compromises: compromises on flexibility, costs, speed or other aspects. To make these compromises in an analytically and correct way is what makes the difference between bad and good architecture. In this customer reference is explained how RBS, in collaboration with LafaData Application Architecture, thought when the company decided to build a new service-oriented architecture, SOA, for the next generation of retail systems.

2. Conventional retail systems

RBS builds retail systems. A retail system is a number of applications used to administrate articles and prices, handle payments at cash desks and also to store sales information, among other things. It consists of both hardware components such as computers, bar code readers together with software for operating the system. Point-of-sale systems (cash desks) must be fault tolerant, which means that even if external services (for example the stock system) are not accessible, the cash desk system should still be able to register sales and at least handle cash payments so that operations do not come to a standstill.

Fig. 1 shows how a conventional retail system works. The start point is the receipt. SP (StorePos) is the cash desk or cash desks, SO (Store Office) is a local BackOffice placed in the store, ERP is the overlying business system.

There are six paths to be considered, calculations on at least three points and report outputs in three steps which continuously escalate in extent.

Modules and subsystems are tightly connected to one another which involve risks in changes since other modules/systems are also affected.

This system was modern a couple of years ago, the customers are satisfied and thanks to the retail system RBS has become market leading in the convenience goods trade in Sweden.

QUESTION: *Interesting; how many transactions does that mean?*

ANSWER: *It's difficult to give an exact answer, but around 100 billion SEK (ca 15 billion USD) per year pass through this architecture.*

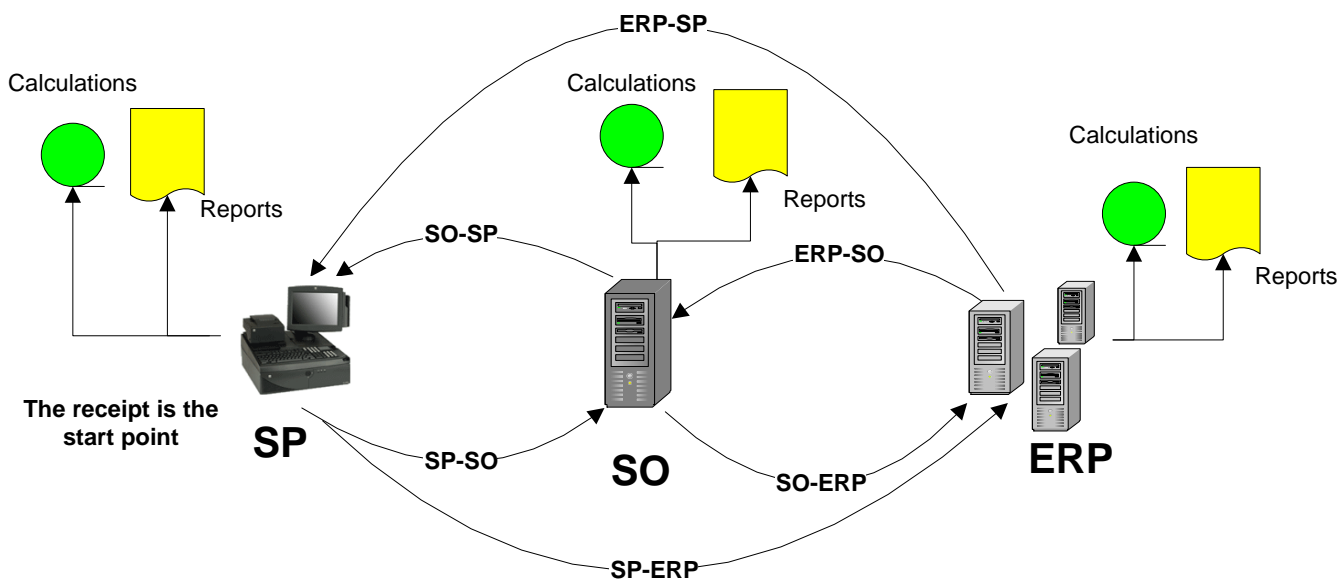


Fig. 1. A conventional retail system, showing the paths in a conventional retail system solution.

3. New demands

The retail trades, which RBS have specialised in, have the aim of treating articles and customers in a similar way. The systems are directed towards the customer being dealt with correctly and above all quickly. The queues in a conventional supermarket for foodstuffs are an example of the need for quickness.

The new demands that have emerged and that already exist in the specialised retail trade are that articles and customers are able to be treated in a unique way but with continued demands on quick handling.

4. The first sketches

The first sketches showed that RBS had reached a cul-de-sac: it was not possible to develop systems in the same way as before.

– We needed to think along new lines to effectively meet the market's new and continuously changing demands on a new retail system, says Stefan Melander, business architect at RBS.

To continue sending down all the information required to deal individually with article and customer at the cash desk would create big problems. How will the system be able to keep a check on what price and stock information are correct at the time?

5. The demands suit SOA

RBS and LafaData concluded that the new demands fitted in to the SOA way of thinking.

– We need to develop systems that from the very start are built for changes. We want to go from building functions to supporting processes and building loosely connected systems to reduce complexity and dependence, says Lajos Farkas, solution architect at LafaData

SOA is fundamentally about supporting business processes, reducing the complexity and acquiring flexible systems, and we saw several possibilities of working towards these goals.

Supporting business processes

Since RBS had focused for many years on business processes in the retail trade, the company had a unique possibility of working top down. This means starting from the shop, asking what it needs and identifying IT services and functionality which support the main processes. By RBS having an understanding of the whole process, the company can also avoid suboptimization in individual parts of the processes. This also has the advantage of knowing the systems in detail, which gives it the possibility of working bottom up and utilising the functionality that already exists.

6. New architecture

There exist a number of frameworks for architecture, e.g. the Zachman Framework, but RBS in cooperation with LafaData decided to use the framework with four views and five aspects, developed by Microsoft's Regional Directors in Sweden – 2xSundblad Architectural Framework gave the company a map and assisted it by communicating the ideas to architects, customers and the management.

Articles should be receipted against stock immediately at the time of sale and can have their own identity.

Customers often have their own agreed price lists and special offers via the CRM systems.

In the follow-up, demands are made that article and customer types are able to be reported separately, i.e. overlying systems must be given information about what was applicable at the time of sale.

All these new demands we wanted to include in a cost-effective and flexible way.

Can the cash desk rely on the local information being correct, and what happens if the updating with central systems shows dissimilarities and errors regarding local as against central information?

QUESTION: *OK, could you please briefly summarise what you regard as a 'cul-de-sac'?*

ANSWER: *It's very difficult to build distributed systems in which the same data appears in several places. It creates a complexity which increases with new functions and which eventually becomes obstructive.*

Reducing complexity

By using independent services LafaData and RBS wanted to reduce the dependence between different parts in the system. The principle of focusing on exchange of information between services, so called 'contract first', would be able to give us the possibility of focusing on individual services and of working with them independently without needing to worry so much about other parts of the system.

Flexible systems

The ideas in SOA of independent services means to us that it can upgrade, add and remove functionality in a much easier way. We can also integrate with other systems in an easier way if it has well-defined interfaces which build only on open standards.

QUESTION: *At the same time did you see any disadvantages with using SOA?*

ANSWER: *Yes, the initial costs can be higher the first time since it is a partly different way of thinking. . However, these are costs which one gets back, taking the long view.*

We have made a thorough analysis of the retail system to break down the system into autonomous services. It was a question of finding the right level of the breakdown. It eventually transpired that it was sufficient having a handful of services such as user- and concern administration, and also cash services.

Then we would easily be able to increase these with new services such as credit note handling and gift voucher handling.

To implement these services LafaData and RBS used much of the material that Microsoft had uploaded on to the internet to support SOA developers. We have used design patterns, guides and application blocks from Microsoft patterns & practices, adopted the ideas regarding fiefdoms and emissaries, the developers have looked at webcasts and have been frequent visitors at Microsoft's Road Shows and other Microsoft seminars.

LafaData and RBS build services as multilayered servers in accordance with Microsoft's reference architecture, and use its Web Services and Transaction Services design patterns. As a data bearer we use typed datasets all the way along from the database (Microsoft SQL Server) out to the clients who are implemented as rich clients in Window's Forms. Typed datasets suit us perfectly since they can be consumed by the clients, and it is easy to transform them into XML documents which external clients and applications can consume.

However, it is not in all situations best if one has a very mixed environment.

The user interface is built according to Microsoft's PAG group's recommendations in which the user interface is regarded as a presentation service. Even the presentation services are layered and consist of User Interface- and User Process components. Thanks to this division it was possible to customise the dialogues much more easily than before.

Problems that were discovered on the way were finding services that really are autonomous and that are neither too large nor too small. Other problems were that at the outset of the development there was nothing else to reuse than operational know-how and experience of the retail system. We did not wish to enmesh itself in the old code, so everything was rebuilt from the beginning and it took time before it was possible to show something that worked. As new developers were added to the project they had to learn these things and it has also influenced the initial time in a negative way.

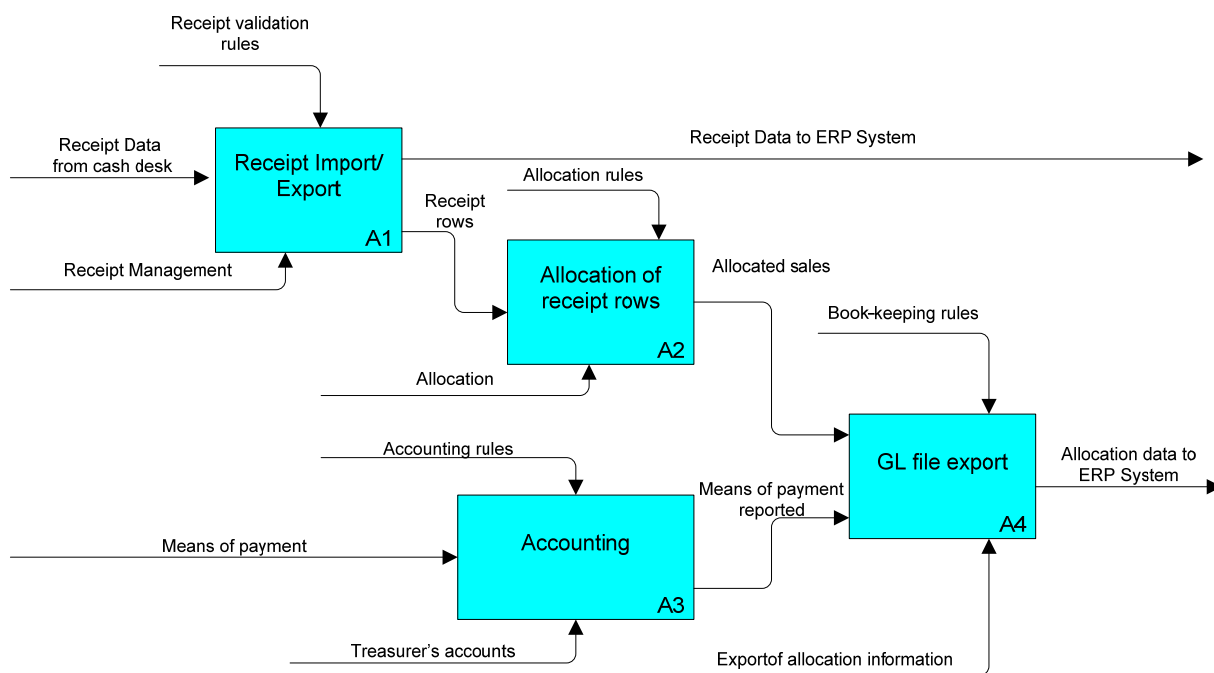


Fig. 2. Process diagram. Process handling of receipt information

7. Final Solution

RSB and LafaData have learnt a lot during the process and have acquired new satisfied customers. Besides us now having a very flexible solution, it also succeeded in integrating the retail system with our customers' ERP system online, for example Oracle, PeopleSoft and SAP.

We have gone from building functions to supporting processes. Please compare the picture of the conventional retail system with Fig. 2 above, which is a process diagram of receipt handling in RBS' communication solution, RetailCenter (RC).

In that there now exists a well-defined interface between cash desk and RC in the form of XML schemas and Web Services, the solution became transparent and other suppliers' cash desks can now be linked to RC. This can in turn open up new markets for RBS in which the company can offer greater value to new customers in the form of processes and functionality that their present cash desks and ERP systems cannot manage.

LafaData and RBS has built up a new development process with the aid of Microsoft Solution Framework and it can support the product efficiently during the whole of its life cycle, both in maintenance and further development.

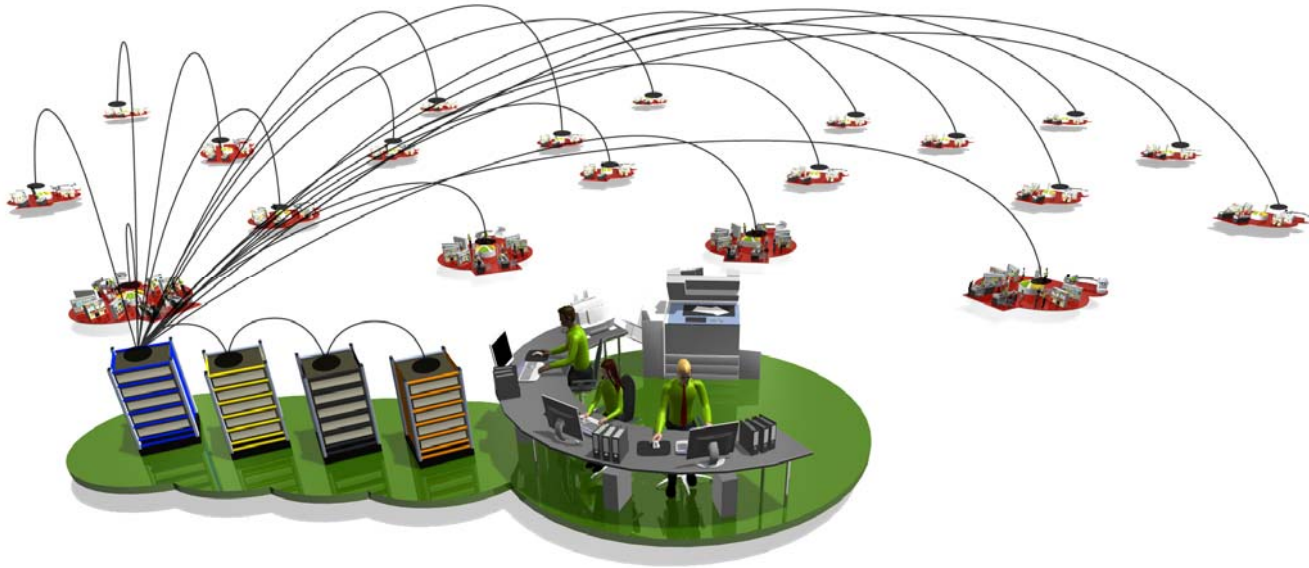


Fig. 3. RetailCenter – RBS’ platform for handling complex integrations between systems developed by RBS and LafaData and systems developed by other operators.

Thanks to us having introduced the new SOA architecture and new development processes, we succeeded in increasing productivity considerably. It took more time to develop the first solutions compared with working in the old way but that investment repaid itself quite quickly in maintenance and further development. LafaData and RBS can reuse a lot of coding and components from the one service to the other and now we are building its solutions about 50 per cent faster than previously.

It became increasingly easier to put new resources into the project in that there were solutions that could be used as references and educational material for new coworkers.

– For us this is a success and we are very pleased that we chose service-oriented architecture and Microsoft’s recommendations as a platform for a new generation of retail systems, says Stefan Melander and Lajos Farkas.

Reference persons

IF YOU HAVE QUESTIONS ABOUT RBS’ SOA-ORIENTED RETAIL SOLUTIONS, PLEASE CONTACT:

Stefan Melander

Business Architect
RBS – Retail Business System AB
Mobile: +46 70 318 15 21
Email: stefan.melander@rbs.se

www.rbs.se

IF YOU HAVE QUESTIONS ABOUT THE AREA “ERP MEETS SOA”, PLEASE CONTACT:

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
Email: annika.granebring@mdh.se

www.mdh.se

IF YOU HAVE QUESTIONS ABOUT SOA AND SOLUTION ARCHITECTURE, PLEASE CONTACT:

Lajos Farkas

Certified Microsoft .NET Architect
LafaData Application Architecture
Mobile: + 46 70 728 01 47
Email: lajos.farkas@lafadata.se

www.lafadata.se

FOR OTHER QUESTIONS ABOUT CUSTOMER REFERENCE, PLEASE CONTACT:

Daniel Akenine

Architect Evangelist
Developer Platform & Evangelism
Microsoft Sweden
Office: +46 8 752 3136
Mobile: +46 73 408 3136
Email: daniel.akenine@microsoft.com

www.microsoft.se