

**University of Twente**  
*The Netherlands*

***Master Thesis Business Information Technology***

---

# **Public Services Integration & Interoperability**

---

*the use of an Enterprise Service Bus to facilitate Service-Oriented Architecture in a  
public environment*

**Master Thesis Business Information Technology**

|                          |   |
|--------------------------|---|
| author:                  | <b>M. Holtkamp</b>  |
| institute:               | University of Twente, Enschede, the Netherlands   |
| date:                    | <b>June 14, 2006</b>  |
| version:                 | <b>Final</b>  |
| graduation<br>committee: | <b>dr. L. Ferreira Pires</b><br><i>Faculty of Electrical Engineering, Mathematics and Computer Science</i><br><b>dr. P.M. Wognum</b><br><i>Faculty of Business, Public Administration and Technology</i><br><b>dr. ir. J.P. Verhoosel</b><br><i>TNO Information and Communication Technology</i><br><b>ir. M.A. van Bekkum</b><br><i>TNO Information and Communication Technology</i> |







---

# Title page

---

title: **Public Services Integration  
& Interoperability**  
*the use of an Enterprise Service Bus to facilitate Service-Oriented  
Architecture in a public environment*

author: **M. Holtkamp**

student number: **s0002232**

date: **June 14, 2006**

version: **Final**

MSc programme: **Business Information Technology**

track: **Networked Business**

institute: **University of Twente, Enschede, the  
Netherlands**

graduation company: **TNO Information and Communication  
Technology**  
*Colosseum 27  
7521 PV Enschede  
the Netherlands  
<http://www.tno.nl/ict/>*

graduation  
committee:

**dr. L. Ferreira Pires**  
primary supervisor  
*faculty of Electrical Engineering,  
Mathematics and Computer Science*

**dr. P.M. Wognum**  
secondary supervisor  
*faculty of Business, Public Administration  
and Technology*

**dr. ir. J.P. Verhoosel**  
manager  
*TNO Information and Communication  
Technology*

**ir. M.A. van Bekkum**  
supervisor  
*TNO Information and Communication  
Technology*









---

# Management summary

---

The Dutch government has defined an agenda to enforce interoperability in public services by means of Information and Communication Technology (ICT). Three dimensions of interoperability can be distinguished: the organizational, semantic and technical one. These dimensions pose specific requirements on candidate solutions. Additionally, integration projects are accompanied by a number of common requirements which should be addressed by a potential solution as well. Dutch municipalities are slowly evolving towards a mature level of Electronic Service Delivery (ESD). Unfortunately, a vast collection of *differing* systems, policies, adopted standards, and social aspects are typically involved. This disparate situation within and between these municipalities slows the process towards sophisticated electronic government down.

Enterprise architecture can be used to provide insight in the structures, relations, information flows and products in an organization. We have adopted an enterprise architecture framework which covers a municipality and the required interoperability by means of seven domains. By using these domains in combination with views to focus on relevant information, this information can be targeted at specific stakeholders. The business, application, and technology views have been used in this research. These views correspond with the layers of abstraction used in the adopted enterprise architecture framework. The techniques discussed in this research have been positioned in the enterprise architecture framework by relating it to a certain domain. We use the technologies' description and position in the framework to predict their influence on surrounding domains.

The Service-Oriented Architecture (SOA) is an application architecture which follows the Service-Oriented Computing (SOC) paradigm and can be positioned in the application domain of the enterprise architecture framework. This architecture is expected to provide the interoperability as required by the Dutch government. It describes a design, development and deployment approach for information systems, by defining concepts and general techniques for designing, encapsulating, and invoking reusable business functions through loosely bound service interactions. SOA can operate platform, time and location independently. Therefore this approach enables the development of highly distributed cross-organizational systems.

SOA consists of five key elements: application front-ends, services, a service repository, a service choreographer and a service bus. The application front-end is the part of the architecture through which an end-user communicates with the underlying applications or data-sources. A service is a unit of functionality provided by a service-provider to a requesting service-consumer. We recognize three service types which are in line with the adopted enterprise architectural views: (1.) business services, (2.) application services and (3.) technology services. A service repository contains the description of all services which are available for invocation by service consumers. The sequence in which they are invoked, eventually resulting in the representation of a business process, is determined by the service choreographer. Services communicate with each other through a service bus which is the infrastructural component of SOA. We position SOA in the application domain but stress that it seriously influences the information, process, organization, and infrastructure domains as well. Realizing this is essential in implementing an enterprise-wide SOA.

An Enterprise Service Bus (ESB) is an implementation of the service bus and provides a collection of infrastructural capabilities, implemented by middleware

technology. In essence, it is a set of combined hubs that acts as the backbone of SOA that can be *centrally* monitored and managed. The ESB allows an enterprise to connect, mediate, and control the interaction between services across highly distributed and heterogeneous environments. Interoperability is ensured because the ESB supports international standards, adopts the use of technology specific adapters, and provides content-based processing and a high level of scalability.

Since a vast collection of ESB-capabilities is available, a justified selection has to be made to gather a suitable ESB-based solution. For this selection approach we suggest three steps that involve seven Enterprise Application Integration-scenarios (EAI-scenarios), six ESB solution patterns and fourteen architecture decision issues. First, an EAI-scenario which resembles the municipal situation to a large extent should be identified. Next, an ESB solution pattern which suits the particular municipal needs can be selected. Thirdly, the advised ESB capabilities are distinguished by addressing a number of architecture decision issues which are related to the EAI-scenario and the ESB solution pattern. After these issues have been addressed, a collection of ESB capabilities that cover the posed (technical) requirements can be derived.

In combination with the BEST framework we applied this approach on four Dutch municipalities: Enschede, Voorst, Almelo and Groningen. The BEST framework can be used to assess an organization's maturity concerning enterprise-wide implementation projects and to identify human and organizational risks involved in such projects. After some interviews concerning the municipal and ICT organization, we identified the involved departments, teams, processes and systems for two selected scenarios. With these characteristics and the gathered situation sketch in mind we applied the selection approach and derived a possible ESB-based solution. The BEST framework was used to analyze this solution and the involved municipal situation. This analysis enabled us to predict the impact of the *fictive* implementation project to a certain extent.

The flexible nature of the ESB provides municipalities the possibility to select the capabilities which are required for their specific situation. The ESB is able to supply a significant level of technical interoperability for municipalities. Therefore, technology is not the main problem to achieve a higher level of interoperability. For the analyzed administrations, the main problems are expected at the organizational level of interoperability. Top-management support at the level of the municipal council and a centralized ICT department with sufficient mandate to make decisions are important aspects of achieving sufficient harmony within the organization concerning agreements, adopted standards, and technologies.

We conclude that ESB and SOA are suitable technologies to achieve a higher level of interoperability. However, technology is not the only factor of reaching the envisioned situation. Organizational issues have a significant influence on this process and should be acknowledged and tackled before an enterprise-wide implementation project is initiated. After initiation, decent project management should guide the implementation project. Since long term implementation project management is not common for municipalities, external skills might be required to guide this. A centralized ICT management organization should be considered to maintain and update the system after implementation has been completed.

It is recommended to embed ICT related activities in the organization and the (political) agenda. An ICT-organization with sufficient authority and mandate which covers *or* coordinates all ICT activities should be considered. Top-level commitment can be provided by assigning a strong natural leader to manage this ICT-organization and to take full responsibility for the achieved results. This encourages line managers and employees to cooperate, or at least decrease resistance.

We suggest a roadmap to ESB and SOA adoption that helps decision makers and advisors to maintain overview of the situation and the remaining steps which have to be taken. This roadmap is presented in chapter 6.

---

# Management samenvatting

---

De Nederlandse overheid heeft een programma opgesteld ter verbetering van samenwerking en efficiency binnen de onderliggende overheden met behulp van Informatie- en Communicatie Technologie (ICT). De combinatie van samenwerking en efficiency wordt samengevat in de term interoperabiliteit. Drie dimensies van interoperabiliteit worden hierbij van belang geacht: de organisatorische, semantische en technische dimensie. Deze dimensies stellen specifieke eisen aan een eventuele oplossing. Hier komt bij dat integratie trajecten aanvullende eisen aan een dergelijke oplossing stellen. Momenteel boeken de Nederlandse gemeenten vooruitgang op het gebied van Electronic Service Delivery (ESD). De grote variëteit die tussen en binnen (de procedures en systemen van) deze publieke instanties bestaan remt deze voortgang echter af.

Een bedrijfsarchitectuur kan gebruikt worden om inzicht te bieden in de bestaande structuren, relaties, informatie stromen en geleverde producten binnen een organisatie. In dit onderzoek wordt gebruik gemaakt van een bedrijfsarchitectuur die aan de hand van zeven domeinen een gemeentelijke organisatie en de erkende interoperabiliteitsdimensies beslaat. Het gebruik van deze domeinen in combinatie met zogenaamde “views” zorgt ervoor dat er geconcentreerd kan worden op informatie die voor een specifieke groep betrokkenen relevant is. In dit onderzoek is gebruik gemaakt van de bedrijfs-, applicatie-, en technologie view. Deze indeling naar abstractieniveau komt overeen met degene die gebruikt wordt in de gehanteerde bedrijfsarchitectuur. De technieken die gedurende dit onderzoek aan de orde komen, worden stap voor stap in deze bedrijfsarchitectuur geplaatst door ze aan een bepaald domein te relateren. We gebruiken de technologiebeschrijvingen en de posities in de bedrijfsarchitectuur om de consequenties van het gebruik van dergelijke technieken op een bedrijf te kunnen voorspellen.

Een Service-Oriented Architecture (SOA) is een applicatiearchitectuur dat invulling geeft aan het paradigma van Service-Oriented Computing (SOC). Men verwacht dat deze architectuur de mate van interoperabiliteit kan bieden zoals die binnen de Nederlandse overheid vereist is. De architectuur beschrijft een benadering voor het ontwerp en de ontwikkeling van informatiesystemen. Deze methode bevat een collectie van concepten en technieken voor het ontwerp, inkapseling en uitvoering van herbruikbare bedrijfsfuncties die aan de hand van service interacties met elkaar communiceren. SOA kan onafhankelijk van het systeemplatform, tijd en locatie functioneren, wat de ontwikkeling van organisatieoverschrijdende gedistribueerde systemen mogelijk maakt.

SOA bestaat uit vijf essentiële elementen: application front-ends, services, een service repository, een service choreographer, en een service bus. De application front-end is dat deel van de architectuur waarmee een gebruiker interactie voert. Een service is een stukje functionaliteit die door een service provider wordt aangeboden aan de verschillende service consumers. Drie verschillende service typen worden erkend: (1.) bedrijfs-, (2.) applicatie, en (3.) technologieservices. Deze indeling komt weer overeen met de indeling die binnen de bedrijfsarchitectuur gehanteerd wordt. De service repository bevat de omschrijvingen van alle services die binnen de architectuur beschikbaar zijn. De volgorde waarin deze services aangeroepen worden bepaalt welk bedrijfsproces er uitgevoerd wordt en daarmee de uiteindelijke functionaliteit. Het aanroepen van services in een specifieke volgorde is de verantwoordelijkheid van de service choreographer. De communicatie tussen de verschillende services verloopt via de service bus, het infrastructurele deel van SOA.

Wij positioneren SOA binnen het applicatiedomein maar benadrukken dat deze technologie ook het informatie-, proces-, organisatorisch-, en infrastructurele domein beïnvloedt. Een organisatie moet zich dit goed realiseren alvorens een bedrijfsbrede SOA geïmplementeerd wordt.

Een Enterprise Service Bus (ESB) is een implementatie van het service bus principe. Dit product combineert een aantal bekende middleware technieken wat resulteert in een breed aanbod van infrastructurele functies. In feite is het een combinatie van gekoppelde messaging hubs die samen de infrastructuur van een SOA vormen en tegelijkertijd als eenheid *centraal* beheerd kunnen worden. Een ESB biedt de mogelijkheid om services binnen een gedistribueerde en heterogene omgeving met elkaar te verbinden en te beheren. Interoperabiliteit wordt geboden door algemeen aanvaarde standaarden te ondersteunen, evenals transformaties tussen de verschillende standaarden, content gebaseerde verwerking van berichten en een hoge mate van schaalbaarheid.

Aangezien de ESB een flinke hoeveelheid mogelijkheden biedt, moet er een selectie van mogelijkheden plaatsvinden worden om tot een geschikte ESB oplossing te komen. Wij stellen drie stappen voor die gebruik maken van zeven Enterprise Application Integration-scenario's (EAI-scenario's) zes ESB solution patterns en veertien architecture decision issues. Ten eerste zal een EAI-scenario geïdentificeerd moeten worden die zoveel mogelijk overeenkomt met de situatie zoals die zich op dat moment bij de gemeente in kwestie voordoet. Ten tweede kan er een ESB solution pattern geselecteerd worden die aansluit bij de gemeentelijke behoefte. Als derde dienen er een aantal architecture decision issues behandeld te worden die aan het erkende EAI-scenario gerelateerd zijn.

In combinatie met het BEST raamwerk hebben we deze selectiemethode toegepast op een viertal Nederlandse gemeenten, te weten: Enschede, Voorst, Almelo en Groningen. Het BEST raamwerk kan gebruikt worden om een de volwassenheid van een organisatie aangaande bedrijfsbrede implementatie projecten te bepalen en om de menselijke en organisatorische risico's die betrokken zijn bij dit soort trajecten te identificeren. Na een aantal interview aangaande de gemeentelijke en ICT organisatie, hebben we de betrokken departementen, teams, processen en systemen geïdentificeerd voor een tweetal realistische scenario's. Met deze karakteristieken en de verkregen situatieschets in het achterhoofd hebben we de hierboven voorgestelde selectiemethode toegepast. Op deze manier hebben we voor elke gemeente een mogelijke oplossing verkregen waarbij de ESB een centrale rol speelt. Vervolgens hebben we het BEST raamwerk gebruikt om de verkregen oplossing in de gemeentelijke context te analyseren. Aan de hand van deze analyse kunnen we de impact van de voorgestelde *fictieve* implementatie tot op zekere hoogte voorspellen.

De flexibiliteit zoals die door een ESB geboden wordt stelt gemeenten in staat om juist die eigenschappen te selecteren die voor hen vereist zijn. Op deze manier biedt de ESB een significant niveau van interoperabiliteit voor gemeenten. De technische factor hoeft dus niet het probleem te zijn om interoperabiliteit binnen gemeenten naar een hoger niveau te brengen. Voor de bestudeerde gemeenten bleek dat de voorspelde problemen zich voornamelijk voordoen op organisatorisch vlak. Betrokkenheid van het top-management op het niveau van de gemeenteraad en een gecentraliseerde ICT-afdeling met voldoende autoriteit en mandaat om organisatiebrede beslissingen te nemen zijn belangrijke aspecten teneinde voldoende harmonie binnen een gemeente te brengen.

Wij concluderen dat ESB en SOA geschikte technologieën zijn om tot een hoger niveau van interoperabiliteit te komen binnen een gemeenschappelijke context. Toch blijkt wederom dat de technologie niet de enige factor is om de gewenste situatie te bereiken. Organisatorische aspecten hebben significante invloed op dit proces en zullen uitgebreid behandeld moeten worden alvorens over te gaan op een organisatiebreed implementatietraject. Nadat een implementatie project opgestart is zal gedegen project management ervoor moeten zorgen dat het project naar behoren verloopt. Aangezien gemeenten niet dagelijks omgaan met complexe bedrijfsbrede implementatietrajecten zal deze deskundigheid wellicht aangetrokken of ingehuurd moeten worden. Een gecentraliseerde ICT beheersorganisatie zou overwogen moeten worden om het beoogde systeem te beheren.

Het wordt aanbevolen om ICT-gerelateerde activiteiten te verankeren in de organisatie en de (politieke) agenda. Een ICT-organisatie met voldoende autoriteit en mandaat die alle ICT-activiteiten coördineert zou overwogen moeten worden. Betrokkenheid op het hoogste niveau kan gerealiseerd worden door een sterke natuurlijke leider aan te wijzen als eindverantwoordelijke van deze ICT-afdeling. Dit moedigt lijnmanagers en personeel aan om mee te werken, of om op zijn minst de weerstand te verminderen.

We stellen een stappenplan richting ESB en SOA adoptie voor dat adviseurs en beleidsmakers kan helpen het overzicht in het probleemdomein en de te nemen stappen te behouden. Dit stappenplan wordt in hoofdstuk 6 gepresenteerd.



---

# Preface

---

## Expectations

After initiation of this research, it was temporarily unclear which way to go or which approach to adopt. Luckily the support was great during this phase and gradually my research model grew into a stable guide for my research. Now, seven months of work, various interviews, workshops and conferences later, I think my expectations have been exceeded. Broad knowledge of promising technologies, experience with a large professional company, interview-techniques and insight in the Dutch public sector are the skills which I have gained during this graduation project.

## Use of the results

This research is supported by TNO ICT and the University of Twente. The published results can be freely used when properly referred to.

## Acknowledgements

This thesis is the final product of my master's programme Business Information Technology. In the year 2000, when I started this master at the University of Twente, Enschede, the Netherlands, a concluding thesis was way too far to think about. First I had to get familiar with a great city, new friends, new colleagues and especially a new style of life. Nearly six years of my life I've spent at students house "MorgenMisschien", house of the fit. This house provided the solid foundation for great parties, new experiences and friendships. Therefore I would like to thank the current and former residents of this special house. Op Morgen!

Then, after 4 year, the university provided me with a great opportunity: a study tour to the country of happiness: Brazil. One year later I returned to this intriguing country for an internship and fell in love with the city of Salvador da Bahia. This city showed me its beauty, the passionate inhabitants and way of living for nearly six months. The gained friendships and bonds with Brazil will be a great part of my future personal (and professional?) life. Again I would like to thank the irmãs Rodrigues for their hospitality; you're always welcome in the Netherlands!

Finally, the process of graduation started, TNO ICT provided me with a hospitable environment to focus on the subject, discuss the traditional gap between the business and technology, drink coffee, and play "Rozenkoning" with King of Roses Frederik Bonte, I still have to beat your topscore of 26<sup>2</sup>! Specially I would like to thank Michael van Bekkum and Jack Verhoosel for their comments and feedback.

From the university the support came from dr. Luís Ferreira Pires and dr. Nel Wognum for whom I am thankful for their constructive comments and provided support. Nel, enjoy your retirement!

During the graduation process my family and my Deventer friends always supported me, although I think the hardest aspect in this was to remember the name of my course: "Yes, something with computers". Great mental support from this city was also provided by a new energy in my life, the most special, linda and beautiful girl I have ever met, Chantal thanks for the great moments I have had, and will have with you!

And last, but not least, this research would not have existed without the support of a number of helpful people, expression of gratitude goes out to Robin Grimmelhuyse, Holger Peters, Wout Brinkhuis, Ale Berends, Gert Hylkema, Cor Top, Henk Wobbles, Erik Legtenberg, Raymond Honings, and Sonic Software.

Enschede, June 14, 2006

Menno Holtkamp





*“Technology will be the solution, as soon as it stops being the problem.”*

- Charles Poirier, 2004



---

# Table of contents

---

|           |  |            |
|-----------|--|------------|
| <b>1.</b> | <b>INTRODUCTION .....</b>                            | <b>1</b>   |
| 1.1.      | BACKGROUND .....                                     | 1          |
| 1.2.      | MOTIVATION .....                                     | 2          |
| 1.3.      | CONCEPTUAL RESEARCH DESIGN .....                     | 2          |
| 1.4.      | TECHNICAL RESEARCH DESIGN .....                      | 4          |
| 1.5.      | THESIS OUTLINE .....                                 | 5          |
| <b>2.</b> | <b>STATE OF AFFAIRS .....</b>                        | <b>6</b>   |
| 2.1.      | DUTCH GOVERNMENT ORGANIZATION .....                  | 6          |
| 2.2.      | DUTCH ELECTRONIC GOVERNMENT .....                    | 7          |
| 2.3.      | EGOVERNMENT ACTORS: ROLES AND RESPONSIBILITIES ..... | 10         |
| 2.4.      | REQUIRED INTEROPERABILITY .....                      | 11         |
| 2.5.      | CURRENT MUNICIPAL USE OF ICT .....                   | 15         |
| 2.6.      | IN SUMMARY .....                                     | 16         |
| <b>3.</b> | <b>STATE OF THE ART .....</b>                        | <b>17</b>  |
| 3.1.      | ENTERPRISE ARCHITECTURE FRAMEWORK .....              | 17         |
| 3.2.      | SERVICE-ORIENTED ARCHITECTURE .....                  | 19         |
| 3.3.      | THE ENTERPRISE SERVICE BUS .....                     | 28         |
| 3.4.      | IN SUMMARY .....                                     | 34         |
| <b>4.</b> | <b>SELECTING A SUITABLE ESB SOLUTION .....</b>       | <b>35</b>  |
| 4.1.      | SELECTION APPROACH .....                             | 35         |
| 4.2.      | EAI-SCENARIOS INVOLVING AN ESB .....                 | 38         |
| 4.3.      | ESB SOLUTION PATTERNS .....                          | 39         |
| 4.4.      | ARCHITECTURE DECISION ISSUES .....                   | 41         |
| 4.5.      | IN SUMMARY .....                                     | 47         |
| <b>5.</b> | <b>CONSEQUENCES OF ESB-FACILITATED SOA .....</b>     | <b>48</b>  |
| 5.1.      | THE BEST REFERENCE FRAMEWORK .....                   | 48         |
| 5.2.      | ENSCHDEDE .....                                      | 53         |
| 5.3.      | VOORST .....   | 62         |
| 5.4.      | ALMELO .....   | 69         |
| 5.5.      | GRONINGEN .....                                      | 75         |
| 5.6.      | IN SUMMARY .....                                     | 81         |
| <b>6.</b> | <b>CROSS-CASE ANALYSIS .....</b>                     | <b>83</b>  |
| 6.1.      | CASE COMPARISON .....                                | 83         |
| 6.2.      | DISCUSSION .....                                     | 84         |
| 6.3.      | REFLECTION .....                                     | 86         |
| <b>7.</b> | <b>CONCLUSIONS AND RECOMMENDATIONS .....</b>         | <b>87</b>  |
| 7.1.      | CONCLUSIONS .....                                    | 87         |
| 7.2.      | RECOMMENDATIONS .....                                | 91         |
| 7.3.      | FUTURE RESEARCH .....                                | 94         |
|           | <b>REFERENCES .....</b>                              | <b>95</b>  |
|           | <b>APPENDIX I. GLOSSARY .....</b>                    | <b>100</b> |
|           | <b>APPENDIX II. LIST OF FIGURES .....</b>            | <b>104</b> |
|           | <b>APPENDIX III. LIST OF TABLES .....</b>            | <b>105</b> |
|           | <b>APPENDIX IV. GRAPHICAL NOTATION .....</b>         | <b>106</b> |
|           | <b>APPENDIX V. MODEL-DRIVEN ARCHITECTURE .....</b>   | <b>107</b> |
|           | <b>APPENDIX VI. INTERVIEW QUESTIONS .....</b>        | <b>111</b> |



---

# 1. Introduction

---

This chapter introduces our research by discussing the context, main objectives, scope and delimitation, approach and structure. After describing the background and motivation for this research, we divide our research design in a conceptual and technical part as suggested by the research design theory of Verschuren & Doorewaard [VER00].

## **1.1. Background**

It is becoming more common for public services and information to be organized around the customers' perspective of events, e.g. 'getting married', 'having a child', 'buying a new house', et cetera. This requires a personalized, uniform and consistent view on the available information and services. Typical government interactions, however, require a number of transactions across multiple departments or even agencies. This results in a situation where government agencies are forced to cross traditional boundaries in order to share information and services.

For the Dutch government, the ideal situation is an environment where agencies work together by sharing information and business processes. This enables the provision of services that can be accessed jointly by those agencies. To reach service delivery across public organizations, the tools currently used in government agencies to perform business should be compatible. Since these tools are typically based on information and communications technology (ICT) a government-wide homogeneous ICT-approach is necessary.

In the Netherlands, the process towards interoperable government is formalized in a national agenda. This agenda states that in the year 2009, 65 percent of the public services should be available electronically. In addition, public organizations should be accessible by means of electronic identification, authentication and exchange of information. This has a great impact on municipalities because: 1) the majority of interaction between citizens or companies and the government occurs at municipal level and 2) their ICT-systems are currently not suited for this task. The scheduled goals concerning electronic government (eGovernment) combined with the information system's functionality currently available at municipalities, results in the following challenge: the definition and implementation of public services' electronic interoperability. In this context, interoperability not only involves technical issues concerned with connecting application interfaces, but is also an essential requirement to share and reuse knowledge between networks, and reorganize administrative processes to better support the services themselves.

Problems might occur for instance when a municipality wants to offer certain services by means of an electronic desk. Citizen-related information should be available at the front-office to support this. Typically, this information is stored in disparate back-end systems that are not coupled with the front-office. A single connection between an application in the front- and back-office can be achieved relatively easy, but the integration of a number of services, both technical and organizational, can result in an unmanageable and tightly coupled business architecture. Such an architecture is expensive to maintain and difficult to open up to other government agencies. The latter is especially important when agencies are expected to work together extensively, which is exactly what the Dutch government aims at.

## **1.2. Motivation**

Currently, governmental bodies in the Netherlands are taking the concept of Service-Oriented Architecture (SOA) facilitated by an Enterprise Service Bus (ESB) into consideration. By applying a service-oriented approach as proposed in this architecture, government agencies can integrate current systems, enabling them to open up their services in a controlled manner. SOA is an approach to designing, implementing, and deploying information systems. The system is based on components implementing discrete business functions. These components, called "services", can be geographically and organizationally distributed. Existing services can be reused in new business processes when needed. An ESB provides a set of infrastructure capabilities that enable coupling and co-operation of services in SOA.

Government agencies are aware that too few specifications and best practices are available concerning the integration and implementation of services and the impact they have on organizational structures and processes. TNO ICT has taken up the challenge of collecting the available knowledge in this area and to combine this in a set of recommendations. Our research is part of that challenge.

## **1.3. Conceptual research design**

Following Verschuren's research design theory, the objectives of our research should be covered by a central research question. This question is addressed by answering several underlying research questions while staying within the pre-defined scope. During this process, a so-called research model supports the research team in approaching the defined subjects using the proper context.

### **1.3.1. Objectives**

The aim of our research is to gain experience concerning the application and implementation of an ESB in a governmental environment. We focus on the use of the ESB to support the concept of an interoperable government and the related impact on the organization. Our research particularly applies to the local governmental layer in the Netherlands, namely municipalities. The contribution in knowledge has been made by clarifying the concepts of SOA, ESB, and their mutual relations. In addition, the features provided by an ESB have been compared to the requirements as posed by the need for an interoperable government.

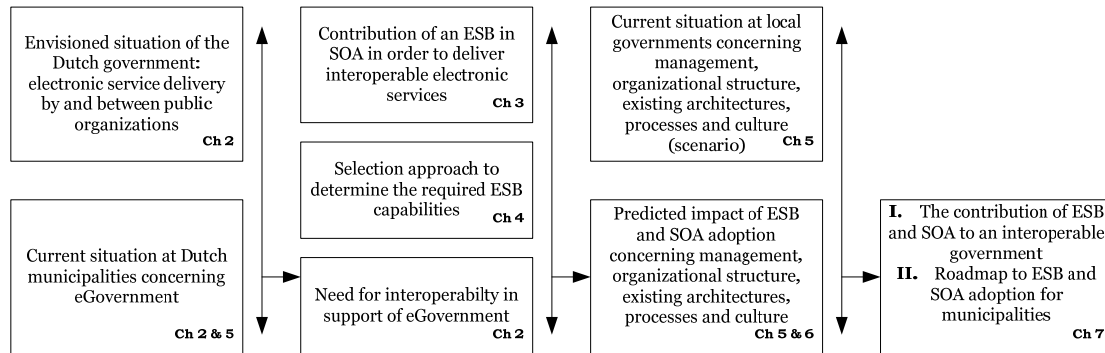
### **1.3.2. Central research question**

The central question should cover both the principles of SOA and ESB. Combined with a focus on the integration of public services within the typical structure of a municipality, the following question is posed:

*"Which contribution to the integration of public services and realization of an interoperable government in the context of municipalities can an Enterprise Service Bus provide in Service-Oriented Architecture and what should be done to make such implementation successful?"*

### **1.3.3. Research model**

The central research question as stated in the previous section can be visualized schematically by means of a research model. This model contains the relations between the distinguished subjects of research, their context and the order in which the research should be performed. Vertical arrows represent a "confrontation" between two or more components and the horizontal ones represent: "concluding from this".



**Figure 1: research model**

The research model shows that the difference between the envisioned and current situation is the basis of our research. The use of an ESB to realize SOA might be an approach to provide the required interoperability. We have tried to predict the impact of the use of these concepts. This prediction is compared with the results of a realistic scenario, from which conclusions are drawn, resulting in a set of guidelines which should be addressed when adopting SOA facilitated by an ESB.

### 1.3.4. Research questions

To answer our central research question some knowledge concerning specific subjects has to be gathered. With the suggested research model in mind, the following research questions can be defined to acquire this knowledge:

- What needs for interoperable public services can be recognized?
- What role can Service-Oriented Architecture have in the context of public services interoperability?
- What is the role of an Enterprise Service Bus in Service-Oriented Architecture?
- Which organizational aspects are affected by embedding an Enterprise Service Bus in a public organization?

By answering the enumerated questions, insight was provided on how an ESB can contribute to the envisioned integration between, and within, governmental institutions.

### 1.3.5. Scope

Due to the growing need for interoperable public services, the Dutch government initiated the Modernizing Government agenda. Roughly, this agenda contains five components [AOV05]:

1. improve provided services;
2. decrease bureaucracy;
3. improve the governmental organization;
4. improve co-operation between (local) authorities;
5. increase attention for citizen's opinions.

To achieve (part of) these goals, the government promotes the use of ICT in all underlying governmental bodies. This is clustered within the concept of eGovernment, which manifests itself in four ICT areas as depicted in Figure 3. Our research mainly focused on services integration issues at municipalities. The remaining public organizations and areas within eGovernment have not been addressed in detail. The focus of our research is depicted in Figure 2 and Figure 3.

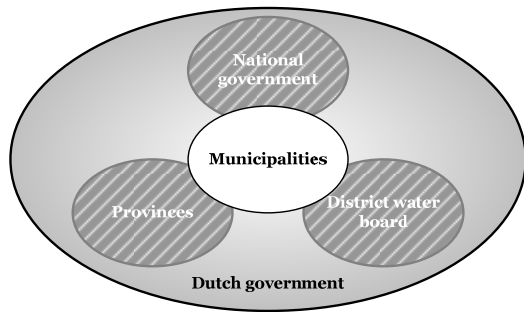


Figure 2: our focus is concentrated on municipalities

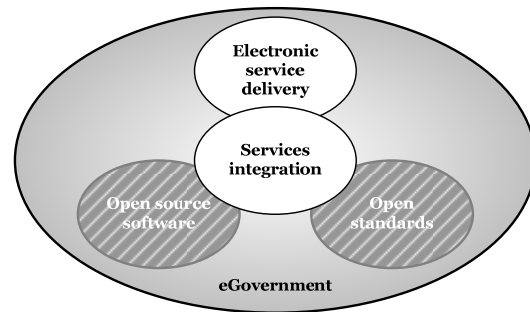


Figure 3: services integration in the context of eGovernment

SOA can be regarded as an architectural approach to building composite applications using reusable services. Identified business functions should be bundled in stand-alone services, which can be invoked through their interface(s). The ESB is the ICT-backbone that facilitates the infrastructure of SOA. Developed services can be deployed on this bus, after which they become available for all other services connected [PAT04].

For this research we particularly focus on the role of ESB within SOA and the capabilities this technology provides. Since no industry-standard for ESB has been defined yet, we aim for an inventory and categorization of capabilities which can be used throughout our research.

### 1.3.6. Definition

This research uses some concepts and phrases which can be interpreted in different ways. To introduce the reader to our research, we present some *brief* definitions of used concepts which will be considered in-depth later on in this thesis:

- *Disparate*, composed of or including markedly differing elements. A disparate situation between and within municipalities refers to the collection of *differing* systems, policies, adopted standards, and social aspects;
- *Interoperability*, the ability to transfer and use information in a uniform and efficient manner across multiple organizations and ICT systems [AGI05];
- *Service*, a unit of functionality that a service provider makes available to its environment. 'Service' in a municipal organizational context refers to an organizational component that typically has departments, which, on their turn, are divided in teams [JON04].

## 1.4. Technical research design

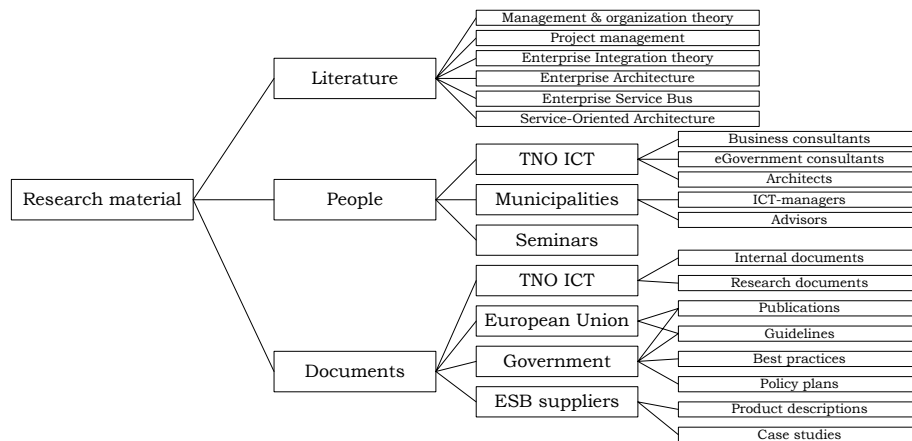
Now we have described the concept of our research in the conceptual research design, we describe the chosen approach on how to actually perform this research in the technical research design. We discuss which research material and sources are used to answer our defined research questions and which strategy we pursued in gathering this knowledge and approaching the selected sources.

### 1.4.1. Research material

A collection of research material is used to address our defined research questions. This collection consists of scientific literature, conversations or interviews with various people and published documents. The latter are written from a more practical point of view opposed to scientific literature. An overview of the used material is depicted in Figure 4.

Preliminary research concerning the Dutch government, ESB and SOA has been performed using scientific literature and publications. Since the subject of interoperability is currently studied by the European Union, these sources have been used as well. Conversations with consultants and architects within TNO ICT resulted in a bigger understanding of typical issues which occur within municipalities. Interviews with municipal public servants confirmed most of these opinions.





**Figure 4: Overview of the used research material**

We were able to address our first research question by combining literature concerning management & organization theory with documents as published by the Dutch government and the European Union. The second research question was answered by studying literature concerning SOA, discussions with consultants, best practices of the Dutch government and recent experiences of industry ESB suppliers. The third and fourth research questions were approached more practically. Hands-on experience with ESB and SOA was gained during a one-week course and one month experimenting in TNO's laboratory. Combined with available literature, we were able to determine the role of ESB within SOA. Enterprise integration theory and a scenario-based analysis among four municipalities involving various interviews provided us the information to predict the impact of ESB and SOA on municipalities, covering the fourth research question.

### 1.4.2. Research strategy

Since our research has a significant empirical component, the adopted strategy for this research has to align with this characteristic. Therefore we have chosen to combine literature research with so-called *comparing case studies* [VER00]. To verify the assumptions and conclusions made during our literature research, we have performed a scenario-based analysis at four Dutch municipalities: Enschede, Voorst, Groningen and Almelo. We choose to carry out multiple cases studies because Dutch municipalities show great variety concerning organization structures, available knowledge, resources and culture.

The suggested analysis involved interviews with various municipal information managers and strategic advisors. We have studied how two simple but realistic scenarios are currently carried out at those municipalities and which technology could be used to increase municipal interoperability. The way how various organizational aspects are affected by such technologies was studied next.

Based on the four case studies, we compare the various situations and highlight interesting similarities, or differences, between the studied municipalities.

## 1.5. Thesis outline

The remainder of this document is structured as follows: chapter two discusses the current state of affairs regarding electronic government in the Netherlands, the growing need for an interoperable government, and the requirements this poses to a potential solution. Chapter three investigates the opportunities of emerging ICT-technologies to facilitate the envisioned interoperability. Chapter four proposes an approach which makes it possible to select the appropriate set of required technologies in a certain situation. Chapter five analyses the proposed solution by means of a scenario-based impact-of-change analysis performed at four municipalities. A cross-case analysis in chapter six highlights interesting similarities and differences between the studied cases. Based on this analysis a conclusion and a set of recommendations are constructed in chapter seven.

---

## 2. State of affairs

---

This chapter discusses the efforts performed in the Dutch public sector to achieve a more transparent, efficient, interoperable, and customer-oriented government, facilitated by electronic government (eGovernment). First we take a glance at the Dutch governmental structure, available experience with eGovernment, and the scheduled efforts to further extend this knowledge. This section is followed by a description of roles and responsibilities involved with eGovernment. Three dimensions of interoperability and the related requirements are defined thirdly. Finally, some common requirements as posed by typical integration projects are discussed.

### 2.1. *Dutch government organization*

The Netherlands is a decentralized, unitary state [JON99]. This indicates that the Dutch constitution recognizes several other governmental layers next to the national government. These layers are the provincial and municipal layer. The latter consists of about 460 municipalities with their own constitutional responsibilities in policy development and execution. Next to local service responsibility, municipalities partly account for the administration of national services and products. For instance, the administration of passports is coordinated centrally, but municipalities facilitate the collection of required information and delivery of the passport. Essentially, the municipal layer accounts for over 70 percent of governmental interaction with citizen and companies [LEE05]. The national government stimulates the use of ICT to support this broad administration performed at the municipal level.

Typically, Dutch municipalities are organized using a number of task-specific services. These services have departments, which, on their turn, are divided in teams. For the interaction with service consumers, use of a front-, and back-office is common. The front-office covers both the physical (service-counter) and semi- or non-physical (post, fax, telephone, email, webpage, Web Service) channels through which service consumers communicate or perform transactions with a municipality. Therefore, the front-office covers the processes, information and applications responsible for the decoupling between citizen and (back-end) municipality. An important task of the front-office is to query and coordinate the activities performed in the back-office properly.

The back-office refers to the part of the organization which registers and processes requests which are received through the front-office. The back-office contains the processes, information and applications which are responsible for the municipal primary processes. The eventual product (i.e. the public service) is delivered to the service-consumer directly or through the front-office.

Some authors tend to refer to the area between the front- and back-office as “mid-office”. The term “office” implies a certain level of formal organizational regulation. Though, this coordinating office does not exist as a *formal* department within municipalities. It is a term or concept used by specialists and information architects to use in discussions and solution development. Therefore we choose to adopt the term mid-ware instead. Mid-ware covers information systems and technological infrastructure used in the area between front- and back-office. This part is able to lookup, bundle and coordinate back-office processes to present this as one product to the service consumer through the front-office. Mid-ware should not be confused with middleware which covers standardized infrastructural (software) components to realize coupling between applications and operating systems. This choice is supported by a research by the Dutch government concerning the formal status of the so-called mid-office [GEU02].

## **2.2. Dutch electronic government**

The kingdom of the Netherlands has a long history concerning effective and concise administration. The punctuality of Dutch administrators has contributed to major improvements and cost-decreasing in various areas of trade. The electronic era provided people with new tools to further increase efficiency. The Dutch government started using computers in the late 1950's. Their processing power was used primarily for calculations concerning the Deltaworks initiated after the infamous Flood disaster in 1953 [NIJ94]. After that, several departments started experimenting with automation of administrative processes. Research was performed concerning further application of computing. At that time, every ministry had its own ICT-department and no central coordination was needed yet. Catalyzed by the emerging concept of eGovernment the call for interoperability emerged in the early 1990's. At this time the Dutch government recognized and announced that [LEE05]:

- ICT can be applied to enhance the internal functioning of government as a whole;
- The internal application of ICT requires some planning and coordination between various government bodies and different levels of government.

Based on the first statement, the following definition of eGovernment is adopted for this research:

*"The utilization of electronic technology to streamline or otherwise improve the business of government and to stimulate citizen participation in democratic institutions and processes."*

To cover the second aspect, the Dutch ICT-agenda Electronic Highways was introduced by the ministry of Finance in 1994. This project restricted itself to the adjustment of laws, required for eGovernment adoption, and electronic highway publicity. Additionally the Ministry of Interior and Kingdom Relations introduced The Public Counter Project (OL2000) in 1996. This project can be regarded as the first official stimulant for eGovernment in the Netherlands. The project aimed at a one-stop desk for interaction between citizens and the government. OL2000 acknowledged the need to connect front- and back-offices at government agencies. Combined with a customer-oriented approach based on life events (e.g. birth, marriage, requesting a certain license, change of address), this should facilitate the delivery of governmental services directly to Dutch citizens [ELO05].

The third major political effort to stimulate the use of ICT is known as the Electronic Government Action Programme ("Elektronische Overheid", ELO) and dates back to 1998. This project contained the following themes [MIN98]:

- a proper level of electronic accessibility of the government;
- an improved provision of service (25 percent of the provided public services should be electronically available);
- an improved level of quality concerning governmental functioning.

Recently, the government suggested the Modernizing Government agenda which covers the period 2003-2009 and involves improved co-operation of the central government with provinces and municipalities. This co-operation aims at creating a more effective and efficient interoperable government. In order to reach this situation, five themes have been defined. The deadlines related to these themes vary from 2006 to 2009. The main objectives, relevant underlying objectives and published deadlines for this agenda are [AOV05][GRA04]:

1. Improve quality of provided services:
  - 1.1. Electronic government:
    - 1.1.1. 65 percent of public services should be electronically available (2007);
    - 1.1.2. Electronic access to the government: open up information via the Internet;
    - 1.1.3. Electronic authentication: check citizens' and organizations' identity electronically;
    - 1.1.4. Unambiguous number for persons: Citizen Service Number (2006);
    - 1.1.5. Base registries: central repositories containing base information concerning citizens and organizations (2006);
    - 1.1.6. Electronic means of identification: the Dutch Identity card based on chip card technology;
    - 1.1.7. Electronic exchange of information: standardization to exchange information between citizens, organizations and government agencies electronically;
    - 1.1.8. Broadband connections between governmental bodies, to facilitate the exchange of services and information (2009).
  - 1.2. Quality of service.
2. Decrease bureaucracy:
  - 2.1. Decrease regulations: lower level agencies receive more responsibility;
3. Organizational improvements:
  - 3.1. Task analysis / redesign government;
  - 3.2. Operational management;
  - 3.3. Supervision.
4. Stimulate co-operation between governmental bodies;
5. Listen to citizens.

The sub-targets of project 1.1 are the projects concerning eGovernment. Every citizen should have access to a significant part of the services as provided by the government. Data will be stored once and used by several departments as facilitated by the base registries and broadband communication between all governmental bodies. The defined objectives require a high level of (electronic) co-operation between the various public organizations, as is stressed by project 4.

To gain more knowledge and experience on how to develop, deploy electronic services and to share the acquired knowledge at municipal level, the Super Pilots project was initiated. This project started in 2001 and refers to a covenant between the minister of Urban Policy and Integration and the municipalities of Enschede, The Hague, and Eindhoven and Helmond. The aim of this project was to supply these municipalities with the resources needed to develop an approach towards Electronic Service Delivery (ESD). Eventually, the developed approaches were envisioned to be adopted by other municipalities as well. The project ended in 2004 with the presentation of three approaches. These approaches differed at various aspects, as were the achieved results [ZOU05].

**Table 1: the superpilot cities, their targets and results**

| Superpilot            | Targets and approach  | Result  |
|-----------------------|---|---|
| Enschede              | <p>Extension of the already initiated OLE21 project ("Overheidsloket Enschede van de 21e eeuw"), which is the successor of OL2000 ("Overheidsloket 2000"). This manifested itself in <i>technology-based coupling</i> of front- and back-office. The following areas were defined:</p> <ul style="list-style-type: none"> <li>■ The digital service counter. The web-site through which citizens and companies can access the services provided by the municipality;</li> <li>■ Unity in communication. Independent of the communication channel, the same information has to be provided. OLE21 acts as a central source of information for all available services;</li> <li>■ Front- to back-office <i>co-operation</i> (not integration). The responsibility of each service is assigned to either front-office, back-office, or machine. Based on this division of labor and responsibility a redesign of workflows where ICT has a central role is required;</li> <li>■ Management of production processes. A high level of service quality, combined with transparency is targeted by adopting auditing processes, performance and quality measures.</li> </ul> | <p>During this pilot, a lot of knowledge in setting up ESD was gained. Opening up services via Internet is much more an organizational change project opposed to a technical one. Automation of public service delivery affects the business processes providing this service and therefore, subjects such as tasks and responsibilities, power structure and private and public agenda's are affected as well. For this project, the slogan: "Everything what <i>can</i> be done digitally, will be done digitally" was used. The "<i>can</i>" largely depends on organizational barriers such as the requirement to have a physical signature on specific forms.</p> <p>Enschede approached this pilot from the information and technology perspective. The digital service counter is only used for intake purposes and is barely coupled with the back-office using a standard methodology. It does enable batch-processing of received request, improving efficiency and accuracy.</p> <p>After this first step has been realized, Enschede looks at further interoperability between front- and back-office.</p> <p>In the beginning of 2005, information concerning 497 services was available for consulting and 110 for a certain level of electronic transaction. The level of sophistication varied from providing information only, to full electronic transactions, minimizing physical contact moments.</p> |
| The Hague             | <p>Service delivery is defined as a chain of processes and events. From initial moment of contact until eventual delivery of the requested service.</p> <ul style="list-style-type: none"> <li>■ Extend the concept of municipal architecture to support this chain.</li> <li>■ Define relations, processes and data required to deliver services electronically.</li> <li>■ The architecture should cover and bind <i>all</i> delivered public services.</li> </ul>  | <p>The Hague approached the project mainly from a technological perspective, including detailed specifications at component level. Organizational and process-based changes remained unclear though. A generic process model was defined, including citizen question patterns, transaction modules and a data warehouse. To make these components work, organizational changes are required as well.</p>  |
| Eindhoven and Helmond | <p>Develop a complete comprehensive system for ESD and explore new ways to make municipal service delivery more effective and efficient. Extend the current services already exploited electronically. The following areas were defined:</p> <ul style="list-style-type: none"> <li>■ Broadening projects: further development of services;</li> <li>■ Deepening projects: extending existing services by adding functionality;</li> <li>■ Explorative projects: explore new possibilities of ESD.</li> </ul>   | <p>Eindhoven approached the project from a technological and information perspective and particularly focused at the application layer. A clear distinction was made between front-, mid-, and back-office. The mid-office was assigned as coupling domain between front- and back-office. This makes it possible to leave the back-office unchanged to a certain extent.</p> <p>The intentions of this project were big; the results were slightly disappointing though. This was mainly caused by the big amount of defined sub-projects. In the end it was difficult to combine the results of these smaller projects. Additionally, the capability-gaps concerning ICT were still not defined, therefore, the search for an integrated solution was difficult to complete successfully.</p>   |

The evaluation of the Superpilot project indicated that the objective to accelerate the development of ESD for all Dutch municipalities was not accomplished. Every municipality used a specific approach, which are difficult to combine in an integrated set of techniques or approach methodologies [ZOU05].

### **2.3. eGovernment actors: roles and responsibilities**

In the Netherlands, the political and administrative coordination is bundled in the eGovernment ministers' consultation. The following ministries are represented in this group:

- Ministry of Interior and Kingdom Relations;
- Ministry of Economic Affairs;
- Ministry of Finance;
- Ministry of Social Affairs and Employment.

Harmonization between state and provinces and state and municipalities is ensured by means of the "eProvinces steering group ICT" and the "government coordination group", respectively. To support the 12 provinces and nearly 460 municipalities in their task to realize their part in eGovernment, two support programmes have been initiated: eProvinces (Electronic Provinces) and EGEM (Electronic Municipalities).

To ensure interoperability between targeted and realized services, the Ministry of the Interior and Kingdom Relations monitors the compatibility of the results. This ministry is also responsible for promotion of developed services towards organizations and guides provinces and municipalities in scheduled automation projects. Additionally, provinces and municipalities receive information from the eGovernment Knowledge Centre. This centre provides information concerning the use of ICT to enable eGovernment in a structured way.

As stated before, 70 percent of the public services are realized by interactions at municipal level. These interactions can be divided in the following types [LEE04][LEE05]:

- *Truly local services.* Refers to services which are based on local policy and autonomy concerning the management of municipalities' own affairs. Street and community care and safety, local taxes, sports recreation and culture can be regarded as such services;
- *Joint governance services.* Refers to services which are rooted in national legislation, but which are administered by municipalities while having their own policy responsibilities and discretionary powers. Municipal social assistance, based on the Act Employment & Assistance ("Wet Werk en Bijstand", formerly "Algemene Bijstands Wet", WWB), can be regarded as such a service;
- *Municipal delivery of national services.* Refers to the administration of national policy by municipalities, where the policy is defined at national level. The only task of the municipalities is to deliver this service to the citizen. Issuing passports and driver's licenses can be regarded as such services.

It seems normal to have the government agency from which a service (legally) originates have this service developed and maintained as well. In practice, this responsibility is often handed over to the municipal level. The municipalities can be influenced by the national and provincial government agencies to a certain extent, but usually this is only by means of financial adjustments and guidelines. With other words, municipalities have a high level of independence. This is the main reason why a heterogeneous environment exists throughout municipalities. This heterogeneous situation also exists in the deployed information systems. The current environment lacks transparency and makes it difficult to compare municipalities based on provided services and efficiency.

## **2.4. Required interoperability**

The targeted situation is a government where agencies work together, sharing information and business processes. This makes it possible to provide services which can be accessed jointly by those agencies. Essentially, an interoperable government is required. Interoperability encapsulates the co-operation of systems, processes and people in order to deliver seamless and customer-oriented services. We define interoperability as [AGI05]:

*“The ability to transfer and use information in a uniform and efficient manner across multiple organizations and ICT systems. It underpins the growing level of benefits for enterprises, government and the wider economy through e-commerce.”*

To realize interoperable government, several dimensions of interoperability should be addressed. The European Union’s programme “Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens” (IDABC) has distinguished three dimensions of interoperability which we have adopted in this research [FIN03]:

- organizational interoperability;
- semantic interoperability;
- technical interoperability.

Regarding the scope of this research, the latter dimension is discussed in detail in this chapter. The remainder of this chapter pays extra attention to the technical dimension of interoperability as well.

### **2.4.1. Organizational interoperability**

This interoperability dimension covers the definition of business goals, modeling business processes, helping these processes to co-operate and aligning information architectures with organizational goals. It aims at addressing the requirements as posed by the end-user (i.e. the citizen) by making services available, easily identifiable accessible and user-oriented.

This dimension of interoperability uses so-called life-events and business episodes for citizens (G2C) and organizations (G2B) respectively. This way, customer-oriented development of services is enforced without focusing on the specific functional organization of the public sector [EIF04]. A life-event or business episode refers to either:

- a single business process located at a single department;
- several business processes scattered around several departments or agencies.

An example of such a life-event is when a couple wants to get married. When the bridal couple origin from different municipalities and no central base registry is in use yet, both municipalities should update their information records after the ceremony. The municipalities’ system where the marriage is registered should be able to cross the organizational border and trigger a comparable process at the other municipality.

### **2.4.2. Semantic interoperability**

The semantic level of interoperability is important to ensure unambiguous interpretation, combination with other resources and meaningful processing of information. Therefore, guidelines concerning the following aspects should be addressed:

- *Context.* Refers to the context in which information is represented. This involves meaning of information, granularity of detail, urgency, authority, monetary and measurement units. For Dutch municipal use the StUF XML standard is suggested (Standard Exchange Format, “Standaard Uitwisselings Formaat”). This standard is still subject to discussion and not embraced by all governmental parties and suppliers however. To overcome differences in context, mediation between and translation of data might be required. This functionality should be available in the technical dimension of interoperability;
- *Representation.* Refers to the format in which information should be represented. Standard formats such as (X)HTML, PDF, SVG and XML should be available while future additional formats are anticipated for.

These guidelines in combination with the required technical functionality enable disparate systems to communicate in an *automated* way. It is important to stress the term “automated”. The aim is not only to be able to link information resources statically, but to enable systems to discover and share information dynamically without human interaction. The intended guidelines should be defined in a semantically rich language, e.g. the Web Ontology Language (OWL, note the irregular abbreviation).

The situation can occur where the municipalities’ registers differ in structure. This contrasting situation should be recognized by comparing the related schemas. Next the information should be adapted to conform to the structure in use by the targeted administration.

### 2.4.3. Technical interoperability

The technical dimension of interoperability is essential to ensure communication within an agency and with related agency. The level of required technical interoperability is related to the level of envisioned ESD complexity. In order to structure the level of technical interoperability in the context of eGovernment, four of the following stages of sophistication have been originally defined by the European Union in order to track the progress of the various union members [CGE05]. We have extended these stages with aspects such as vertical and horizontal integration and a fifth stage which covers cross-organizational interoperability:

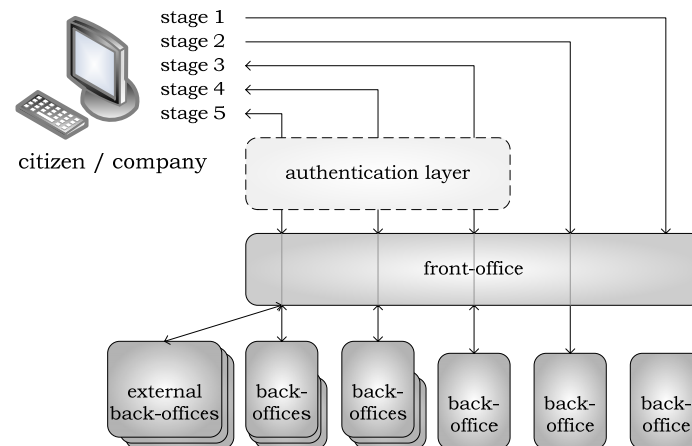
- *Stage 0: traditional.* Total absence of any publicly accessible website managed by the service provider;
- *Stage 1: information.* The information required to initiate the procedure of requesting a product / public service is available on-line;
- *Stage 2: one-way interaction.* The publicly accessible website features paper forms to initiate formal procedures. An electronic form to order a non-electronic form is also considered to be stage 2;
- *Stage 3: two-way interaction / vertical integration.* The publicly accessible website offers the possibility of electronic intake. A form of authentication of the person (physical or juridical) requesting the service is required to reach this stage. An incoming request is registered directly in a back-office system after which it can be processed. For various products / public services, various forms have to be filled though. Since manual procedures are skipped, a significant increase in efficiency can be achieved;
- *Stage 4: full electronic case handling / horizontal integration.* The publicly accessible website offers the possibility to completely process the public service via the website and underlying systems, including decision and delivery. No other (formal) procedure for the applicant is necessary via “paperwork”. The business processes are in line with the user-perception and the way requests are posed. The processes between the front- and back-office are automated and orchestrated. The sequence in which back-office services



are invoked is arranged automatically. This enables the provision of multiple products / public services by submitting only one request.

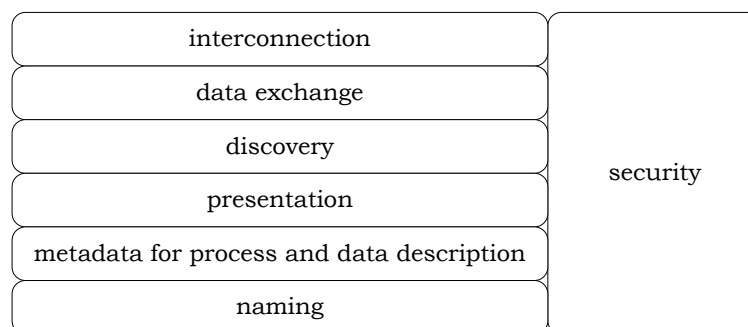
- *Stage 5: cross-organizational interoperability / chain integration.* Certain requests require cross-organizational process management and the use of centrally stored or synchronized information. Requests do not necessarily have to be processed locally. Regional, provincial or national service centers are able to communicate with lower level organizations and administration which enables these centers to act as a centralized point of ESD.

The enumerated stages imply an increasing level of complexity, caused by the growing amount of systems involved. For stage 1, online information has to be provided. Stage 4, in contrast, requires identification, authentication, exchange of information and connections between several administrations within one organization. These administrations are likely to be heterogeneous and spread throughout one or more departments. When systems are barely connected with each other and information is stored in disparate systems as the situation nowadays, complex operations are hard to carry out [LUT04]. In case an organization is required to communicate with external partners, the involved level of complexity further increases since additional features are required, for example extensive security functionality. The difference in complexity among the five stages is depicted in Figure 5.



**Figure 5: overview of the five stages of technical interoperability, or ESD-sophistication**

This research particularly addresses the challenge posed by achieving a high level of technical interoperability. Providing a framework for integrating and synthesizing information in a secure and timely way across disparate and neo-merged agencies or departments is the challenge municipalities are facing [SEB05]. A suggested model for this framework, the technical interoperability framework, contains seven main elements as depicted in Figure 6.



**Figure 6: conceptual model of a technical interoperability framework [AGI05]**

Each element has its task in achieving technical interoperability to facilitate the semantic and organizational dimensions of interoperability:

- *Interconnection.* Addresses standards and technologies used to communicate with a collection of systems;
- *Data exchange.* Addresses the standards and technologies used to describe and encode data for exchange;
- *Discovery.* Addresses the mechanism uses to discover resources. Is likely to adopt a thesaurus in order to recognize comparable descriptions of resources;
- *Presentation.* Addresses standards to present information to the end-user. The used standard depends on both the end-user's application and channel of delivery;
- *Metadata for process and data description.* Addresses the orchestration of processes and data and makes sure dependencies are respected;
- *Naming.* Contains the standards as defined in the semantic dimension of interoperability to ensure consistent naming of entities and processes. For the Netherlands this standard is likely to be the GFO;
- *Security.* Addresses standards and technologies to ensure secure operation of the entire framework.

The described framework describes the requirements which are posed to achieve technical interoperability. This is not a fixed set of requirements though. The exact collection of requirements should be determined individually for every situation. It largely depends on the business strategy, the envisioned stage of ESD, and the existing municipal environment. The suggested steps to recognize these requirements are discussed in chapter 4 and applied in chapter 5.

#### **2.4.4. Common integration issues**

Requirements concerning organizational, semantic and technological interoperability have been addressed in the previous sections. Additionally, integration issues in the following areas can be recognized as well [CHA04]:

- *Complexity.* Due to former choices, current budget constraints and future requirements, integration projects can be regarded as rather complex. Outdated legacy systems are forced to be reused because there is no budget, time or knowledge to re-develop the system with the required functionality. Current governmental services are often developed using point-to-point or supplier-specific solutions. The future situation requires these and new systems to be able to communicate with each other [OSS05].
- *Redundant and nonreusable programming.* Parts of the systems currently in use at Dutch municipalities are redundant or contain useful functions that cannot be invoked by other systems [LUT05]. This heterogeneous situation is mainly caused by the lack of centralized and properly coordinated ICT-development during the last decades. Future efforts should address guidelines concerning the development of reusable software components.
- *Multiple interfaces.* Due to the diversity of developed systems, the same diversity of interfaces is currently in operation. When these interfaces have to be interconnected,  $n(n-1)/2$  connections are required, where  $n$  is the amount of applications in use. Eventually, this results in a so-called spaghetti-architecture. Examples of this situation are depicted in Figure 7 and Figure 8. Since governmental bodies intend to operate transparently, flexibly and efficiently, so many different one-to-one connections which are hard to realize and expensive to maintain, should be avoided. Due to the large amount of interfaces in governmental architectures, ease of coupling is essential.

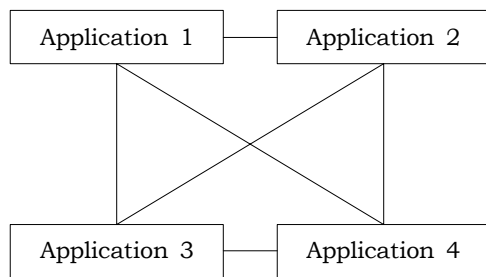


Figure 7:  $n(n-1)/2$  integration problem

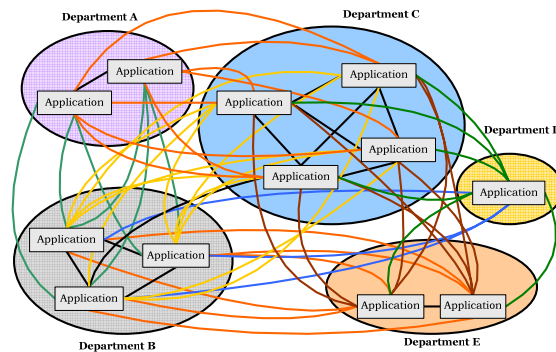


Figure 8: spaghetti architecture [WAL05]

To acknowledge the problem areas discussed priory, the following requirements related to integration projects can be defined for the envisioned solution [CHA04]:

- *Leverage existing assets.* Typically, the existing systems are (still) of great value for the organization. Strategically all systems should conform to the new architecture to reach the optimal situation, but usually this situation cannot be reached at once. Therefore, existing systems should be leveraged and integrated according to a migration plan. They can be gradually migrated when necessary.
- *Support all required types of integration.* This covers:
  - user interaction: a single interactive user experience;
  - application connectivity: a connecting (middleware) communications layer which connects the applications interfaces;
  - process integration: application and services choreography;
  - information integration: store once, use many times;
  - build to integrate: build and deploy new applications and services without disturbing already deployed services.
- *Support incremental implementations and migration of assets.* It should be able to approach integration projects in several phases.
- *Build around a standard component framework.* This promotes reuse of models and systems, migration of legacy systems and adoption of new technologies.
- *Allow implementation of new computing models.* Future computing models might exploit concepts like grid-computing or computing on demand.

## 2.5. Current municipal use of ICT

Due to the efforts of the national government to promote eGovernment, the use of ICT in public service delivery at a municipal level is extensive nowadays. ICT is part of daily routine; employees are connected to both the municipal network and the closed network of the Association of Local Authorities: GEMNET. An important aspect of the adoption of ICT at this level is the use of the Municipal Public Records Database (“Gemeentelijke Basis Administratie”, GBA). The various GBA’s are installed at municipal level, updates are synchronized with related organizations (up to 300). This semi-centralized administration has a key role in various citizen-related processes which depend on accurate information [LEE05].

Toward citizens and companies municipalities are extending there ICT-activities as well. Municipal Internet presence is already 100% and the use of electronic voting in local, provincial, national and European elections is emerging. Though, the quality and maturity of the delivered electronic services vary. Roughly, small municipalities are lagging behind the larger ones. This is mainly caused by lack of knowledge and (financial) resources. This can be seen in recent researches initiated by both municipal (Webdam Monitor) and national (Overheid.nl Monitor) ranking monitors. Recent results are depicted in Figure 9 and Figure 10. The “dips” in both

figures are caused by the municipality of Utrecht. In spite of the significant number of inhabitants, the municipality's quality of ESD is disappointing, according both monitors.

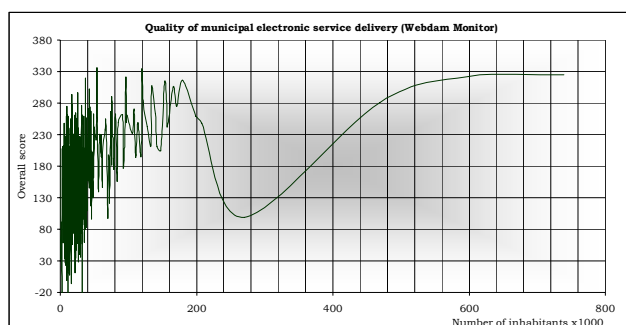


Figure 9: quality of municipal ESD [WEB05][CBS05]

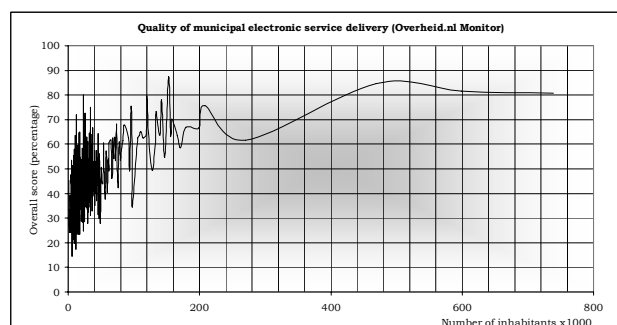


Figure 10: quality of municipal ESD [ADV05][CBS05]

Concerning the technical sophistication, or maturity level, the Netherlands does not score well in a European context. 32 percent of services delivered electronically has reached the stage 4 of eGovernment sophistication (full electronic case handling / horizontal integration, as introduced in section 2.4.3). The main problem for a lot of municipalities is the disparate situation in their administration. Legacy back-end systems which are in use in municipal back-offices are not designed to be opened up via the Internet. Therefore, the majority of the available services is classified as a stage 1 or 2 service. Some municipalities are experimenting with stage 3 services. The amount of stage 4 services which indicate full integration with the underlying administrations lags behind compared to other European countries, as presented in Table 2. Integration with chain partners, indicating a stage 5 of sophistication has not been analyzed yet, therefore data concerning this stage is not available.

Table 2: technical sophistication of ESD in European countries [CAP05]

| Stage 4 electronic service delivery in Europe |          |          |          |          |
|---|----------|----------|----------|----------|
|   | Oct 2004 | Oct 2003 | Oct 2002 | Oct 2001 |
| Sweden  | 74%      | 67%      | 67%      | 28%      |
| Austria                                       | 72%      | 68%      | 20%      | 15%      |
| Finland                                       | 67%      | 61%      | 50%      | 33%      |
| United Kingdom                                | 59%      | 50%      | 33%      | 24%      |
| Denmark                                       | 58%      | 72%      | 61%      | 32%      |
| Norway  | 56%      | 47%      | 35%      | 35%      |
| Spain   | 55%      | 50%      | 40%      | 30%      |
| Italy   | 53%      | 45%      | 35%      | 15%      |
| France  | 50%      | 45%      | 35%      | 25%      |
| Iceland                                       | 50%      | 28%      | 28%      | 11%      |
| Ireland                                       | 50%      | 56%      | 50%      | 22%      |
| Germany                                       | 47%      | 40%      | 35%      | 20%      |
| Portugal                                      | 40%      | 37%      | 32%      | 32%      |
| Belgium                                       | 35%      | 35%      | 25%      | 0%       |
| Greece  | 32%      | 32%      | 32%      | 11%      |
| Netherlands                                   | 32%      | 26%      | 21%      | 5%       |
| Luxembourg                                    | 20%      | 15%      | 5%       | 5%       |
| Czech Republic                                | 6%       | 0%       | 0%       | -        |

## 2.6. In summary

The Dutch government has defined an agenda to enforce interoperability in public services. Three dimensions of interoperability can be distinguished: organizational, semantic and technical. These dimensions pose some specific requirements on candidate solutions. Additionally, integration projects are accompanied by common integration issues which should be addressed by the potential solution as well. Currently, Dutch municipalities already have a 100% Internet presence and are slowly evolving towards a mature level of ESD; the disparate situation at these public administrations slows this process down though.

---

## 3. State of the art

---

This chapter introduces the state of the art concerning potential enablers of interoperable government: SOA and ESB. To determine the position of these technologies in organizations, we use an enterprise architecture framework that covers an organization by means of seven architectural domains. After literature-based analysis of SOA and ESB, we tend to position them in one of the distinguished domains and to predict the influence on surrounding domains to a certain extent.

### 3.1. *Enterprise architecture framework*

Enterprise architecture can support an organization in managing daily operations. Additionally, it provides a means to enhance or change business structures and processes that connect these structures. In this sense, it supports an impact-of-change analysis. More specifically, enterprise architecture is defined by the Dutch Telematica Instituut as [LAN04]:

*“a coherent whole of principles, methods and models that are used in the design and realization of an enterprise’s organizational structure, business processes, information systems, and infrastructures.”*

To support (local) government agencies in achieving a higher level of interoperability, enterprise architecture is useful to gain insight in the current processes, related structures and involved systems. By relating these elements with each other, enterprise architecture can be used to identify problem areas and to communicate them with various parties throughout the organization. We choose to use a framework suggested by Jonkers et al. since it combines simplicity with an extended coverage of an organization by means of seven architectural domains [JON03]. The framework consists of a selection of elements of The Open Group Architecture Framework (TOGAF), the tool for Architecture of Integrated Information Systems (ARIS) and slightly resembles Zachman’s framework for enterprise architectures [ZAC87].

We assume that the used domains of the selected framework cover a municipal organization sufficiently for the purpose of this research. The following domains are used in Jonkers’ framework [JON03]:

- the product domain: describes the offered public services;
- the organization domain: describes the actors and fulfilled roles, working in processes to deliver services;
- the process domain: describes business processes or functions that offer services;
- the information domain: describes information that is relevant from a business perspective;
- the data domain: describes the information suitable for automated processing;
- the application domain: describes business process or function supporting applications;
- the technical infrastructure domain: describes used hardware platforms, middleware and infrastructure used for communication.

As a next step, a “system”, e.g. an organization or an information system, is assumed to primarily consist of a set of actors or active components. Jonkers argues that these components at least have the following interesting aspects:

- *Information.* An actor creates, modifies and exchanges information with other actors in the “system”;
- *Structure.* An actor can be composed of other actors. For instance, a municipality consists of department, which consists of teams, which employs various public servants. The structure aspect describes the static properties of the actor;
- *Behaviour.* An actor has a behaviour, or dynamics. For instance a process which has to be completed, a certain action which is triggered by an event or a scheduled task which repeats itself over time.

The enumerated aspects can be identified at a business, application and technology layer. The aspects, combined with these layers, form the framework in which the seven domains are positioned, as depicted in Figure 11.

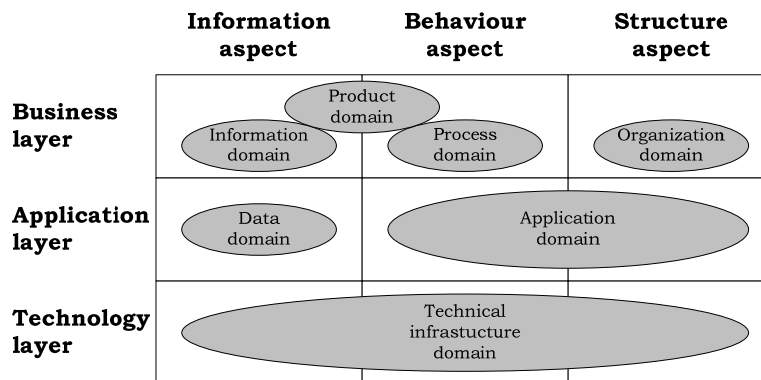


Figure 11: enterprise architectural framework suggested by Jonkers et al. [JON03]

By studying the principles of SOA and ESB, we are able to determine the position of these technologies in the adopted framework and the influence they are expected to have on surrounding domains. This knowledge is used during our impact-of-change analysis. Section 3.2 and 3.3 discuss SOA and ESB respectively.

Since enterprise architecture covers complete organizations, involving various backgrounds and interests, views are used to focus on relevant concerns and abstract from unnecessary information. A view is specified by a viewpoint, formalized in the IEEE-1471 standard. A viewpoint offers a perspective on an architecture, while a view is the visible part of the architecture from that viewpoint. Therefore, a viewpoint specifies what should and what should not be visible for a certain stakeholder. Using a view, the architect is able to explain an architecture to a stakeholder who observes the system from a specific viewpoint. This interaction between stakeholders, viewpoints, views, and architects is depicted in Figure 12 and is used throughout this research.

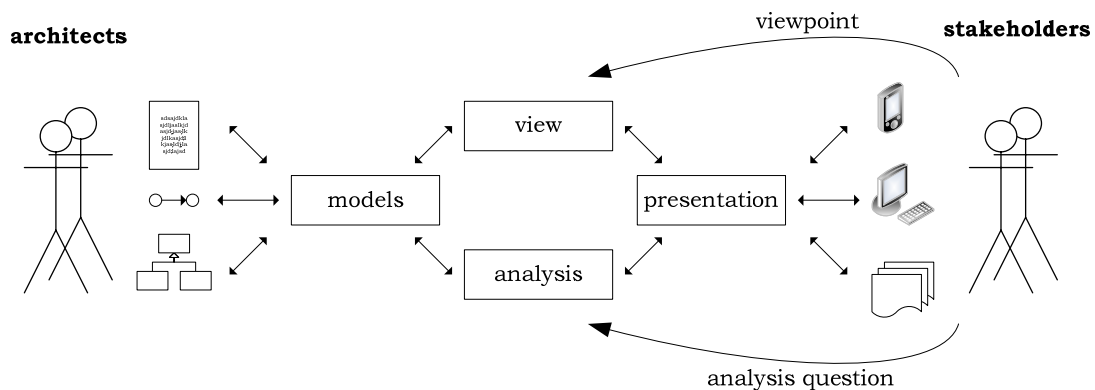


Figure 12: communication between architects and stakeholders using views and analysis [LAN05]

For this research we have distinguished three views, which correspond with the three layers in our adopted enterprise architecture framework. As stated, a view involves a number of models. For this research we have used an architecture for each of the domains as introduced previously. This is presented in Table 3.

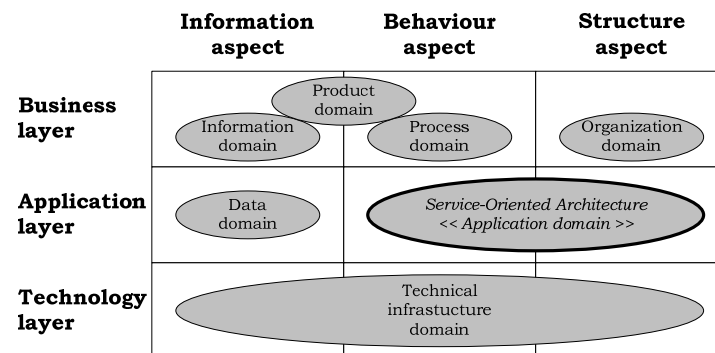
**Table 3: the distinguished views and related architectural models**

| View             | Description  | Involved architectural models   |
|------------------|--|---|
| Business view    | Covers system's users' scope and describes business information flows between people and business processes. | <ul style="list-style-type: none"> <li>Information architecture</li> <li>Product architecture</li> <li>Process architecture</li> <li>Organization architecture</li> </ul> |
| Application view | Covers information systems, individual applications and data sources within an organization.                 | <ul style="list-style-type: none"> <li>Application architecture</li> <li>Data architecture</li> </ul>   |
| Technology view  | Covers integration, distribution and underlying infrastructure components.                                   | <ul style="list-style-type: none"> <li>Technical infrastructure architecture</li> </ul>   |

The following sections describe the principles of SOA and ESB after which these technologies are positioned in the adopted framework. By doing so, we have determined the influence of SOA and ESB on surrounding domains.

### 3.2. Service-Oriented Architecture

SOA is introduced as a candidate solution that is able to provide a high level of interoperability, as required by (local) government agencies. But how can it be defined and how does it fit in the enterprise architecture framework as described in the previous section? The principles of SOA can be used to describe an *application architecture* [PAP03][NAT03]. This implies a position of SOA in the *application domain* of the enterprise architecture framework, as depicted in Figure 13.



**Figure 13: SOA's position in the application domain of the enterprise architecture framework**

To study SOA's influence on surrounding domains, we provide a definition and detailed description of this architecture.

SOA defines the use of services to support the requirements of software users, rather than supplying data or (remote) procedures. Based on definitions of IBM, the World Wide Web Consortium (W3C), and Krafzig et al. we define SOA as [CHA04][W3C04][KRA05]:

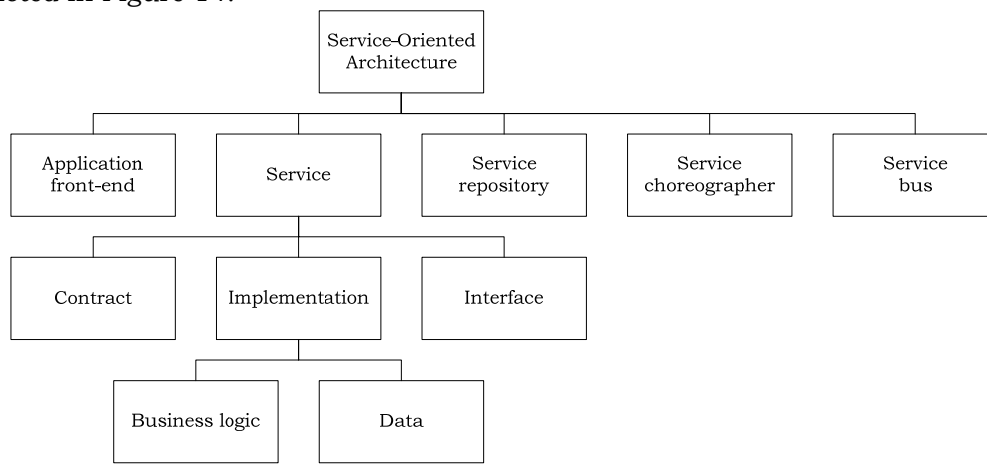
*“An application architecture within which all functions are defined and published as independent services with well-defined interfaces, which can all be invoked through a service bus in defined sequences to form business processes accessible via an application front-end.”*

Breaking down this definition we obtain the following components:

- all functions are defined as services. This includes business functions, business transactions composed of lower-level functions and system service functions;

- all services are independent. External components neither know nor care how individual services achieve their functionality, merely that they return the pre-defined result;
- all interfaces are invokable. At an architectural abstraction level, this implies that it is irrelevant:
  - where the services are located;
  - which protocol or connection scheme is used to invoke the interface;
- all services are connected to a service-bus;
- combining the services in defined sequences results in automated business processes which are accessible via an application front-end.

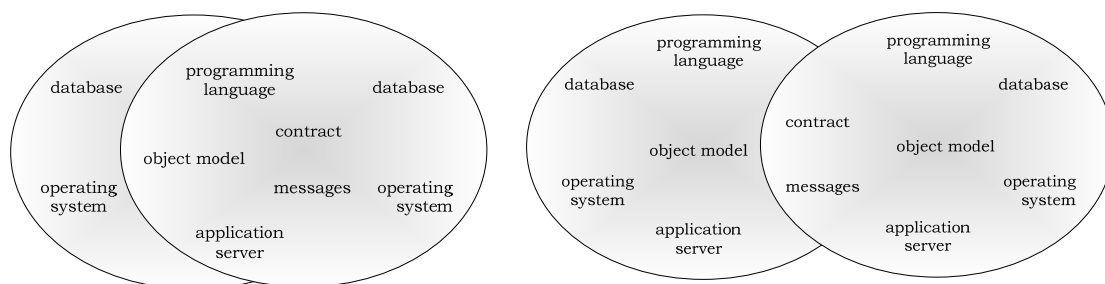
The rationale behind this definition is to realize a *loosely-coupled* architecture that is able to support flexible composite applications by utilizing a selection of reusable and (optionally) distributed services [JAA05]; an overview of the key elements is depicted in Figure 14.



**Figure 14: the key elements of Service-Oriented Architecture**

Before addressing the recognized key elements we stress the importance of a loosely coupled architecture and the benefits a service-oriented approach provides in this context. The Dutch professor Edsger W. Dijkstra used the phrase “Minimal coupling, maximal cohesion” to stimulate the development of high-quality computer applications [BAA05]. Traditionally, applications and systems contain too much rigidity; once built, they are hard to adapt. For a small system, this is problematic. For large-scale systems this can be catastrophic. To avoid rigidity, the amount of *dependencies* should be minimized.

A major benefit of the service-oriented approach is the facilitation of development, deployment and operation of loosely coupled software modules. These modules, or services, are able to operate distributed and independently. This means they can communicate and co-operate with each other, but pose minimal requirements concerning programming language, knowledge of object models, operating systems and database structures. The services are both independent and interoperable. The distinction between tightly- and loosely-coupled architectures is depicted in Figure 15 and Figure 16.





**Figure 15: tight coupling [BIJ05]**

**Figure 16: loose coupling [BIJ05]**

A loosely-coupled architecture enables the following features which are useful for municipalities [WIL04]:

- robustness;
- maintainability;
- changeability;
- change-resilience;
- scalability;
- platform independence;
- time, location and invocation independence (asynchronous messaging).

Additional information concerning the key elements of SOA and their contribution towards a loosely-coupled architecture is outlined in the following sections.

### 3.2.1. Application front-end

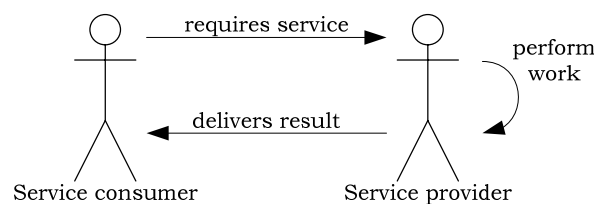
The application front-end is the part of the architecture an end-user can communicate with. For instance, this can be a graphical user interface (GUI) through which a user has interaction with the underlying system. Since a application front-end represents a certain (part of a) business process, the responsibility for this process also lies with this element. Typically the application front-end delegates much of this responsibility to one or more services. In turn, these services perform their task and return their findings to the application front-end. Whereas application front-ends act at the upper layer of traditionally stacked multi-layer applications, services are grouped according to functionality. This is discussed in the following section.

### 3.2.2. Services

A “service” can invoke conflicting reactions as it generally applies to all of the following [LUT04]:

- a part of a governmental organization, e.g. the secret service;
- the act of public worship following prescribed rules, e.g. the Sunday service;
- a branch of the armed forces, the military service;
- a function provided to someone or something.

The latter should be regarded as the service referred to when talking about SOA. This introduces two entities, the service consumer and the service provider. The relationship between them is depicted in Figure 17.

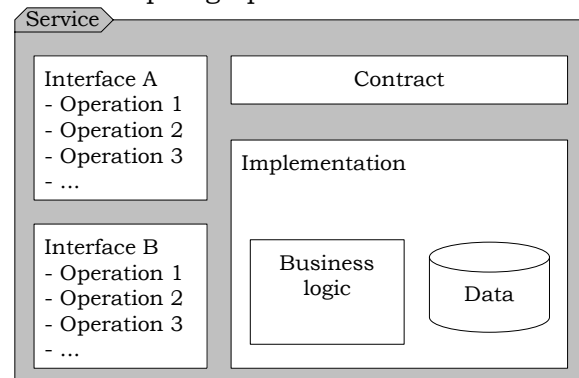


**Figure 17: our interpretation of a service: a provider provides a certain function to a consumer**

Combining these entities with the definition of a service by Jonker et al. [JON04], we have formalized the extent of a service in the following definition:

*“A service is a unit of functionality that a service provider (e.g. organization, department, or system) makes available to its environment through an interface, and which has some value for service consumers in the environment of the service.”*

As depicted in the overview of SOA, a service contains a number of sub elements. These elements and their position within a service are depicted in Figure 18 and explained in detail in the next paragraphs.



**Figure 18: an overview of a service with its key elements**

- *Contract.* The contract contains a specification of the purpose, functionality, constraints and use of the service. At a business level this contract can be compared to a Service Level Agreement (SLA), which defines the expectations between service consumer and provider. Typically, the contract also contains the interface definition. A formal language such as the Interface Definition Language (IDL) or the Web Services Description Language (WSDL) is recommended to define such interfaces. This makes it possible to abstract from technology, resulting in a more loosely coupled architecture. It is possible though, that a service is developed to encapsulate a legacy system function which can not communicate in a formal interface language. In this case it is important to provide detailed information on the functionality and parameters, such as a detailed technical description of the expected network protocol.
- *Interface.* This element is essential for the service to function. The functionality provided by a service is delivered to the service consumer through this interface. The interface *definition*, or *description* contains information concerning expected parameters, provided functionality and possibly quality-of-service. This description *can be* incorporated by the contract which is described above. The interface receives incoming requests and delegates them to the designated business logic for processing.
- *Implementation.* The actual functionality of the service resides in this part. It is the technical realization that fulfills the contract and consists of both business logic and data:
  - *Business logic.* This part represents the actual programmed logic which a service contains. This logic is made available to other services through the service interface.
  - *Data.* Services can contain data as well; this is the purpose of data-centric services which are capable of handling persistent data. This includes storage and retrieval of data, locking mechanisms and transaction management. Typical data-centric services are able to access databases, file systems or tape libraries.

SOA breaks down the overall structure of an architecture in smaller components: the services. These services are the flexible building blocks of the architecture. To organize these building blocks, a certain classification has to be used. This classification results in a number of service types. Nowadays, various research teams suggest varying levels of service abstraction, or granularity. For this research we have used three main service types: (1) business services, (2) application services, and (3) technology services. These types conform to the business, application and technology layers of the enterprise architecture, as adopted previously. The following section discusses these service types in more detail.

## Service types

At the highest architectural level, an organization provides its customers and partners with certain services which result from the interaction performed between organizational parts at a lower level. These services represent the actual functionality offered to the customers and are referred to as *business services*. At a lower level, the services which are offered by an application to the organization reside. These services are the intermediate products which have a role somewhere in the value chain. These services are referred to as *application services* and are typically used by one or more business processes. Eventually, the lower level services can be recognized. These services are referred to as *technology services* and provide the required technical functionality to carry out the higher level services.

For all service types, a distinction is made between *in-* and *external* services. External services are directly invocable by higher level entities. An internal service has a supporting character and can only be invoked by services which are positioned at the same layer. This can be useful to share data models at a specific layer. In a typical municipality, various departments may have adopted naming standards for specific purposes, i.e. jargon is used. At a higher level, similar names can lead to differences in semantics. Internal services belonging to a department can communicate using this jargon, but when the department services are opened up to other parts of the organization using higher level services, a translation has to be made to conform to the organization wide *common data model*.

A service is able to invoke other external services at *all* lower levels. For example, an external technology service can be invoked *directly* by an internal business service, external business service, or service consumer. The position of the addressed service types in the three layers is depicted in Figure 19.

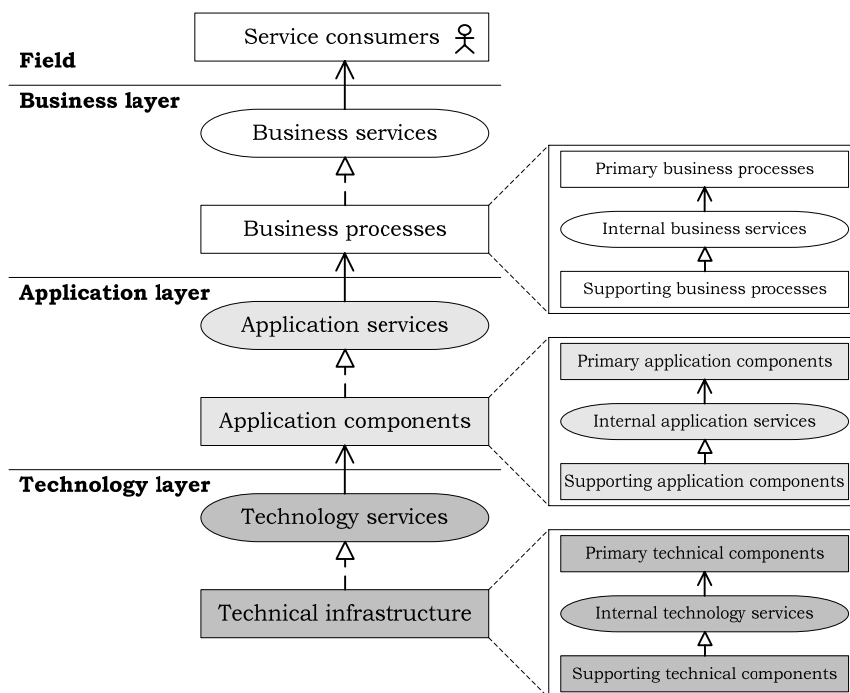


Figure 19: the addressed service types as linking pins between architectural layers [LAN05]

For the position of SOA in the enterprise architecture framework, this means that SOA's influence or impact extends beyond the application domain. Essentially, SOA affects information, behavioral, and structural aspects of the domains located at the business layer: the information, process and organization domain. This influence is depicted in Figure 20, an updated version of the enterprise architecture framework.

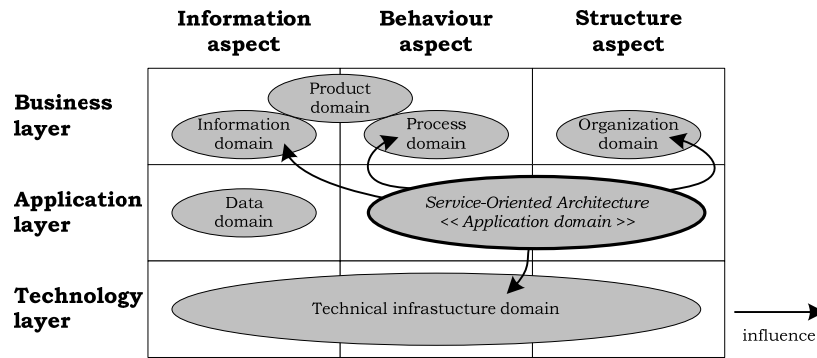


Figure 20: the position and influence of SOA in the revised enterprise architecture framework

By appointing a business function to a service, the responsibility of the function also shifts towards the service. Therefore, service-oriented computing regards a system as a network of functional responsibilities, as opposed to a means for “processing information” in other paradigms. The aforementioned responsibilities are not just technical responsibilities, but business responsibilities, essential for an organization. This is why SOA affects the domains located at the business layer. Documentation concerning business processes is required to model and formalize the envisioned services. Related to the influence on the process domain is the influence on the information domain. Information models within this domain are required for semantic correct communication between service consumers and providers. Additionally, to make SOA successful, the process and organization domain should embrace the philosophy of service-orientation. Municipalities are organizations which offer certain services to their environment and are in that way service-oriented. Their internal organization is often not that service-oriented though. This can cause significant troubles when introducing SOA in municipalities. Further analysis concerning this essential issue takes place in the next chapter.

SOA’s influence on the technology layer is significant as well. Messaging models and underlying capabilities concerning service storage, choreography, management and security are required to properly deploy services throughout an organization. This is where the SOA components: service repository, choreographer, and bus come in. These elements have a more technical character and are described in the following sections.

### 3.2.3. Service repository

The service repository is used to store (references to) services within SOA. Developed services can be *published* in a repository after which they become available to all other services deployed in the architecture. A service consumer is able to query the repository by means of a *lookup*-action. In an advanced SOA, this request for functionality can also be performed by the service choreographer that coordinates the sequence in which services are invoked. The service choreographer is discussed in the next section. This section globally describes the interaction which takes place when a service consumer tends to invoke a certain service. An overview of this interaction between service consumer, provider and repository is provided in Figure 21.

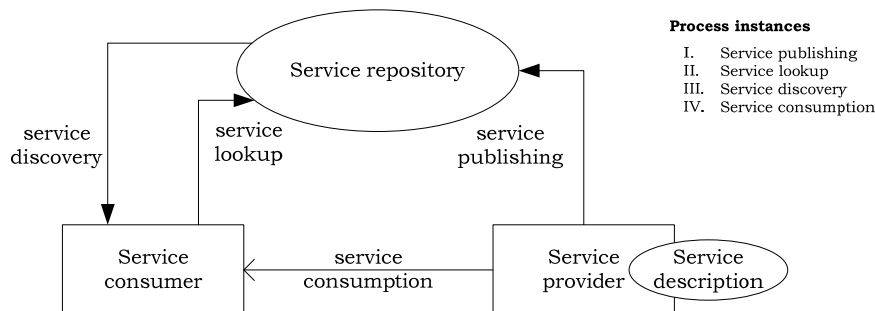


Figure 21: global overview of service interaction using the service repository [MC104]

The following features are pivotal in the interaction process of publishing and looking up services. This set of features enables the realization of a loosely-coupled architecture [MCI04]:

- *messaging*, a protocol implementation which allows communication between the distributed services;
- *service repository*, the repository where published services are stored;
- *service lookup*, the mechanism which enables service consumer to query the service repository;
- *service publishing*, this aspect refers to the process by which a service provider presents the related service description to potential consumers. This does not imply direct communication with the consumers, but publishing a description in the repository, which is used by consumers whenever they try to lookup a certain service. In the context of the service bus publishing refers to the service consumer publishing a service *request*, instead of a service offer;
- *service discovery*, this refers to the process by which the consumer becomes aware of an available service, its functionality and related requirements;
- *service interface description*, this feature defines a published contract or specification of the service. This contract provides information concerning the service functionality;
- *service interface introspection*, which extends the service interface description by providing more detailed information concerning available methods, attributes and return values of the service. Using this introspection in combination with the contract described above, the service consumer can select the proper service and access it in the right way;

### **3.2.4. Service choreographer**

As introduced in Figure 21, the basic form of service interaction contains service providing entities which make their services available for (public) use. In turn, these services are consumed by service consumers. The service repository facilitates the process of looking up services and invoking them. Since services are (usually) not aware of the state in which neighboring services reside. A higher level entity which manages and coordinates the sequence in which services are invoked is a useful solution to this problem. This part of SOA is referred to as the service choreographer.

The role of the choreographer is to define and execute business processes whose configuration and flows are determined by business logic. As a conductor that guides an orchestra to achieve cooperation between musicians, this component guides services' invocation to achieve proper business functionality. The choreographer enables service consumers to dynamically find and interact with various service providers in a sequence which is defined in the process model. When a service is able to operate dynamically, a loosely-coupled architecture is enforced, which was previously stressed as an important aspect.

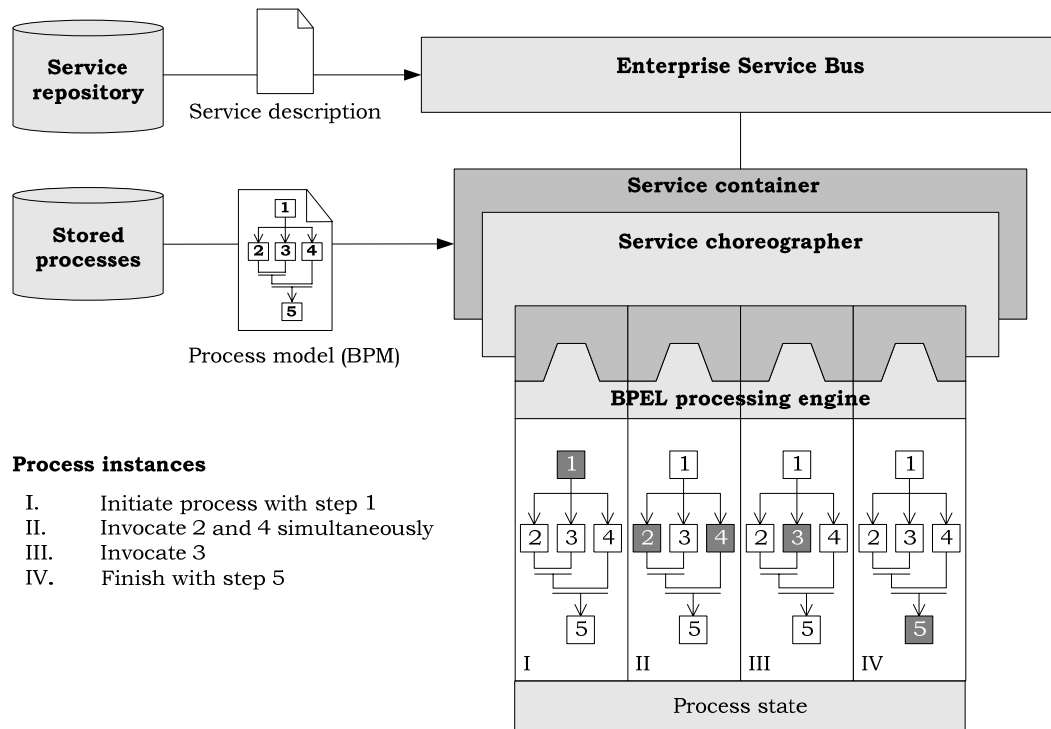


Figure 22: overview of the service choreography service, based on [CHA04b]

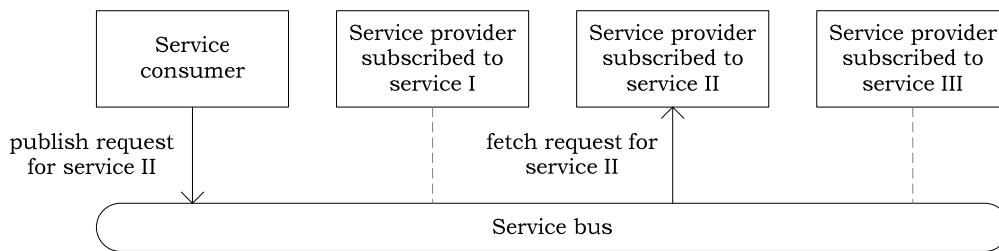
As depicted in Figure 22, the service choreographer typically uses Business Process Modeling (BPM) techniques in combination with the Business Process Execution Language (BPEL). Using these standards, which are not discussed in detail here, small changes to existing business processes can be implemented at a high level. In case the proper services have already been defined, only the sequence in which they are invoked have to be altered. Reuse of code (services), is enforced in this way. The ideal situation is the one where adaptation or introduction of a business process can be achieved by selecting a number of services and solely define the sequence of invocation.

Note that the service choreographer is deployed on the ESB as a *service* as well.

### 3.2.5. Service bus

Loosely coupled architectures can be achieved by encouraging direct or brokered interactions between service consumer and provider. Time is an important aspect when dealing with such interactions. Whenever a service consumer intends to invoke a certain service, the provider should be online and available for processing. Probably this service provider relies on other providers, which on their turn should be available as well. These dependencies can be alleviated, by using a service bus.

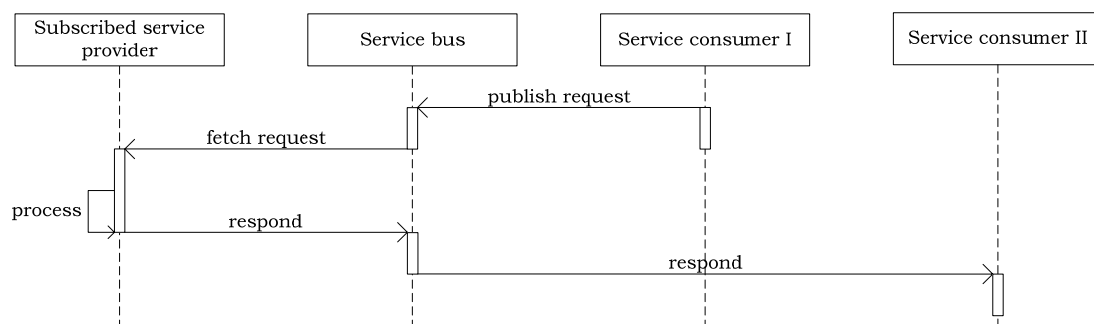
The service bus acts as the infrastructural foundation of SOA. It provides the capabilities which are required to deploy and interconnect services. The service bus typically uses a *publish and subscribe* (pub/sub) mechanism as presented in Figure 23. Other messaging mechanisms such as *request and response* can be supported by service bus implementations as well. Since pub/sub enables service orientation when *no service registry and choreographer are available*, we have specifically highlighted this mechanism. Using a dedicated service registry and choreographer introduces an increased level of complexity to the solution. Smaller and/or initiating municipalities are likely to avoid such situations. Therefore we respect the possible solution which lacks a service registry and choreographer. This can be useful to gain preliminary knowledge on the principles and advantages of service-orientation.



**Figure 23: the pub/sub mechanism as used in the service bus pattern**

The pub/sub mechanism expects *consumers* to publish their *requests* on the service bus. Subscribed service providers are able to fetch the request from the bus, process it and post their replies on the bus again. The consumer does not have to know where the provider is located, which encourages loosely-coupled and service-oriented interaction between the service consumers and providers.

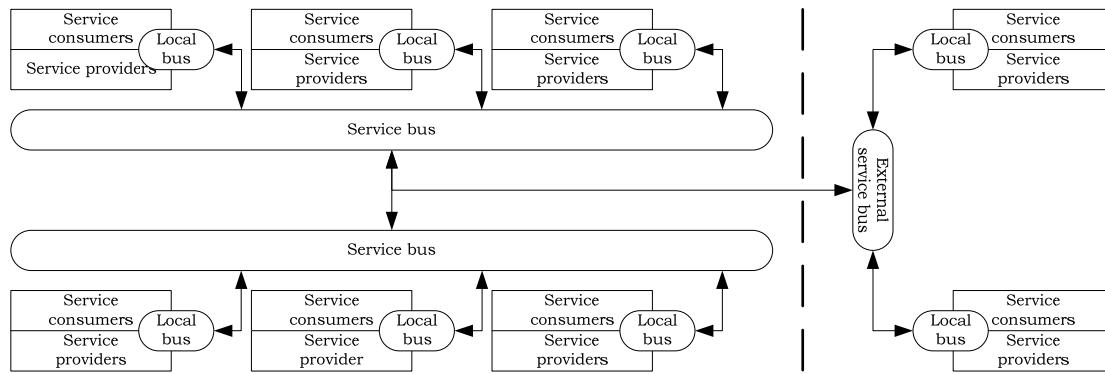
It is also possible to (temporarily) store a message in a queue in case no provider is available to process the request. As soon as a suitable provider becomes available, the bus forwards the request and processing can be dealt with. This principle is known as *store and forward* and enables (long term) asynchronous parallel processing. Eventually, the bus transfers the results to the appropriate consumer. In a complex interaction across multiple processes and services, the response may not even go to the original requestor. Each response may actually be a new message being sent to a forwarding address. This also contributes to loose coupling and is the basis of what is known as *itinerary-based routing* in a service bus, also depicted in Figure 24.



**Figure 24: UML Sequence diagram of itinerary-based routing using a service bus**

Service consumers and providers do not have to worry about the transportation of their messages in the pub/sub messaging mechanism, which is a major benefit. Since the service bus is able to keep track of all messages which are published on the bus, a situation is provided in which suppliers of messages can *fire and forget*. By means of the service contracts which contain agreements concerning quality of service, the service bus is able to properly prioritize the flow of messages.

The distributed service bus provides an even higher level loose coupling. As depicted in Figure 25, local buses are applied to coordinate the interaction with co-located services. Whenever a request can not be completed locally, the request is forwarded to the central bus which takes care of it. When required, external buses can be adopted co-operate. Eventually the collection of busses results in a highly distributed and loosely-coupled architecture [WIL04].



**Figure 25: the distributed service bus**

Section 3.3 provides a detailed description of the ESB, which implements the concept of a service bus.

### 3.2.6. Provided interoperability and independency

Services using XML-based standards are able to provide true *technical* interoperability [STE02]. But SOA can stimulate *organizational* and *semantic* interoperability as well by minimizing the requirements for shared understanding. A service description and a collaboration and negotiation protocol referring to the proper data model are the only requirements for semantically correct communication between a service provider and a service user [LAN05]. For the organizational dimension this means that other (third) parties can connect to provided services or provide services to the organization as long as the used data model remains consistent. Therefore SOA, when deployed properly, can provide interoperability in the organizational, semantic and technical dimension.

In an interoperable government, a high level of independency is important. Governmental agencies should be able to work together and share information. They should, however, not become *technically* dependent of each other. Whenever a province gathers information provided by a collection of municipalities to facilitate a public service on provincial level, this service depends on the provided municipal *information*. Provinces should not become dependent on the underlying applications or infrastructure.

### 3.3. The Enterprise Service Bus

As described in the previous section, SOA defines concepts and general techniques for designing, encapsulating, and invoking reusable business functions through loosely coupled service interactions. Since the SOA is an architectural approach, a technology is needed to implement this architecture.

The distributed service bus was introduced as the infrastructural SOA pattern which provided the ultimate in loose-coupling. A product which follows this pattern is the ESB. In essence, an ESB provides a combination of infrastructural capabilities, implemented by middleware technology, which enables an organization to integrate services using SOA. It facilitates a manageable wide-spread ICT-backbone supporting a range of integration technologies. As it is common with new technologies, a uniform definition of an ESB has not been agreed upon yet. Industry-players tend to focus more on the general concept of SOA when describing an ESB as the basis of integration [POL05]. The ESB though, solely provides the *software infrastructure* for SOA, therefore we use the following definition [GIL04]:

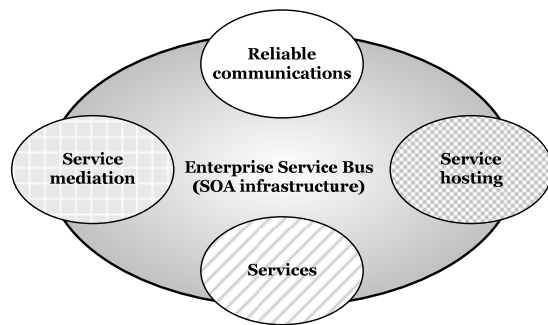
*“An Enterprise Service Bus is software infrastructure that enables Service-Oriented Architecture by acting as an intermediary layer of middleware through which a set of reusable services are made widely available.”*



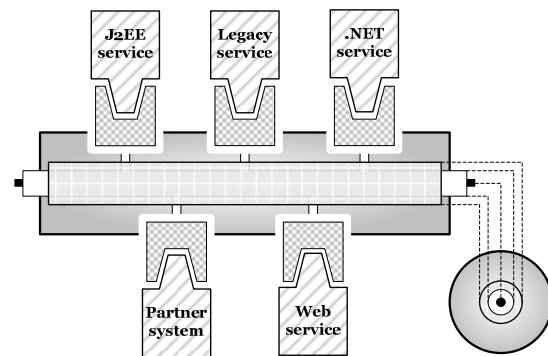
Breaking down this definition we obtain the following components:

- an ESB is software infrastructure;
- an ESB acts as intermediary;
- an ESB is middleware;
- an ESB enables SOA by making services widely available.

The combination of infrastructural capabilities and the application of these capabilities is depicted in Figure 26 and Figure 27.



**Figure 26: the ESB connects a collection of middleware capabilities**



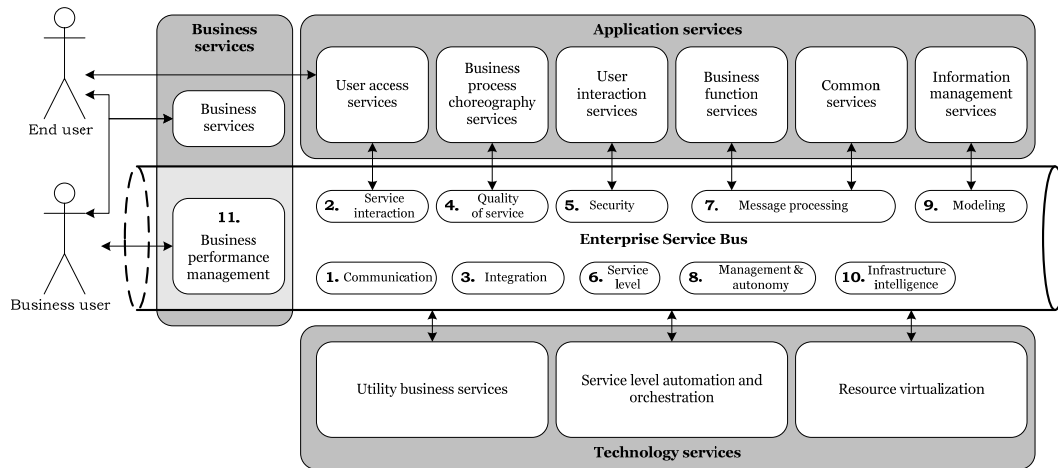
**Figure 27: the combination of available capabilities results in the "bus"**

The figures above contain the following concepts:

- *Services*. As already introduced; refers to software components that can be invoked by means of an interface and *can* be large enough to enable complete business process representation;
- *Service hosting*. Represents the bridge between the service and the ESB, also known as adapter. Because various adapters can be used, services using various technologies can be deployed on the ESB;
- *Reliable communications*. This layer encapsulates the actual transportation layer, or network. It refers to the provided quality of transport. Messages are transported over a dedicated pipeline which interconnects applications, possibly spanning multiple systems. For the services a "fire and forget" approach is adopted, a message is deployed on the service bus after which this component takes care of routing and transportation. To ensure quality of service, reliable communications are essential;
- *Service mediation*. This layer encapsulates communication activities in the bus. It allows any service on the bus to communicate with other services by means of namespaces, regardless their location or type.

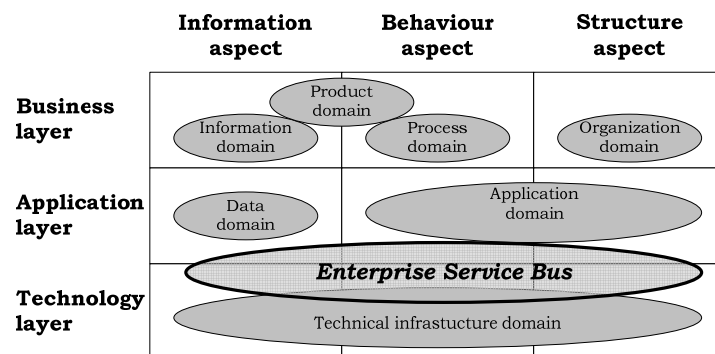
From a service perspective, the ESB acts as a single point of management and communication for both service consumers and providers. The service bus is an infrastructure component and as such does not process business logic; this is typically done by application or web servers [NOT04]. Essentially, an ESB centralizes control and communication and facilitates distribution of the actual processing.

To be able to position the ESB in the enterprise architecture framework properly, more details concerning specific components and provided capabilities are required. A detailed visualization of the ESB and its environment is depicted in Figure 28.



**Figure 28: the ESB and its environment, Table 4 contains detailed capability information [IBM04]**

Since the ESB acts as enabling ICT-backbone of SOA, the three defined service types interact with each other through the ESB. Using our definition, the derived components, and the relation with various service types, we state that the ESB affects the data, application and technical infrastructure domains. Because the ESB is software infrastructure implemented by middleware technology and does not directly involve operating systems or physical infrastructure aspects such as cables, servers and routers, it is modeled as the linking pin between the application and technology layer; this is depicted in Figure 29, our updated version of the enterprise architecture framework.



**Figure 29: position of the ESB in the enterprise architecture framework**

Eleven main ESB components have been introduced in Figure 28. To structure the vast collection of underlying middleware infrastructure technologies, a categorization of provided capabilities is useful. This enumeration of categorized features, as presented in Table 4, is used in this research to match requirements with available features. The numbers preceding the name of an ESB component do not imply a level of importance or necessity; they are used to refer to this component later in this research.

**Table 4: the ESB capability model, based on [IBM04] (extended)**

|   |  |
|---|--|
| <b>1. Communication</b>   | <b>2. Service interaction</b>  |
| <ul style="list-style-type: none"> <li>▪ Routing;</li> <li>▪ Addressing;</li> <li>▪ Protocols and standards (e.g. HTTP, HTTPS);</li> <li>▪ Messaging mechanisms: <ul style="list-style-type: none"> <li>○ Publish / subscribe (pub/sub);</li> <li>○ Request / response;</li> <li>○ Fire &amp; forget, events;</li> </ul> </li> <li>▪ Synchronous and asynchronous messaging.</li> </ul>   | <ul style="list-style-type: none"> <li>▪ Service interface definition (e.g. WSDL);</li> <li>▪ Substitution of service implementation;</li> <li>▪ Service messaging models required for communication and integration (SOAP, XML, or proprietary Enterprise Application Integration models);</li> <li>▪ Service routing directory and discovery.</li> </ul> |
| <b>3. Integration</b>   | <b>4. Quality of service</b>   |
| <ul style="list-style-type: none"> <li>▪ Database;</li> <li>▪ Legacy and application adapters;</li> <li>▪ Connectivity to enterprise application integration middleware;</li> <li>▪ Service mapping;</li> <li>▪ Protocol transformation;</li> <li>▪ Data enrichment;</li> <li>▪ Application server environments (e.g. J2EE and .Net);</li> <li>▪ Language interfaces / adapter for service invocation (e.g. Java, C/C++/C#).</li> </ul> | <ul style="list-style-type: none"> <li>▪ Transactions (atomic transactions, compensation, WS-Transaction);</li> <li>▪ Various assured delivery paradigms (e.g. WS-Reliable Messaging or support for Enterprise Application Integration middleware).</li> </ul>   |
| <b>5. Security</b>  | <b>6. Service level</b>  |
| <ul style="list-style-type: none"> <li>▪ Authentication;</li> <li>▪ Authorization;</li> <li>▪ Non-repudiation;</li> <li>▪ Confidentiality;</li> <li>▪ Security standards (e.g. Kerberos, WS-Security).</li> </ul>   | <ul style="list-style-type: none"> <li>▪ Performance;</li> <li>▪ Throughput;</li> <li>▪ Availability;</li> <li>▪ Other continuous measures that might form the basis of contracts or agreements.</li> </ul>  |
| <b>7. Message processing</b>  | <b>8. Management and autonomy</b>  |
| <ul style="list-style-type: none"> <li>▪ Encoded logic;</li> <li>▪ Content-based logic;</li> <li>▪ Message and data transformations;</li> <li>▪ Message / service aggregation and correlation;</li> <li>▪ Validation;</li> <li>▪ Intermediaries;</li> <li>▪ Object identity mapping;</li> <li>▪ Service / message aggregation;</li> <li>▪ Store and forward.</li> </ul>   | <ul style="list-style-type: none"> <li>▪ Administration capability;</li> <li>▪ Service provisioning and registration;</li> <li>▪ Logging;</li> <li>▪ Metering;</li> <li>▪ Monitoring;</li> <li>▪ Integration to systems management and administration tooling;</li> <li>▪ Self-monitoring and self-management.</li> </ul>                                  |
| <b>9. Modeling</b>  | <b>10. Infrastructure intelligence</b>   |
| <ul style="list-style-type: none"> <li>▪ Data modeling;</li> <li>▪ Common data models;</li> <li>▪ Data format libraries;</li> <li>▪ Public versus private models for business-to-business integration;</li> <li>▪ Development and deployment tooling.</li> </ul>  | <ul style="list-style-type: none"> <li>▪ Business rules;</li> <li>▪ Policy-driven behavior, particularly for service level, security and quality of service capabilities (e.g. WS-Policy);</li> <li>▪ Pattern recognition.</li> </ul>  |
| <b>11. Business performance management</b>  |  |
| <ul style="list-style-type: none"> <li>▪ Management information;</li> <li>▪ SLA alerts;</li> <li>▪ Data mining.</li> </ul>  |  |

The components as described in the capability model are able to work together but not all featured capabilities are necessary for every scenario. For example, in the situation where an ESB is used in a secure environment, the category containing security components might be superfluous. Since every situation requires a specific set of capabilities, it is difficult to define a minimal set of components which facilitates a “true” ESB. To do so, we have used the following (extended) aspects of both the ESB and SOA definition:

- the ESB is an architectural component that provides an *integration* infrastructure consistent with the principles of SOA;
- SOA principles require the use of *implementation-independent interfaces* and *communication protocols* that stress *location transparency* and *interoperability*;
- the ESB may be implemented as a *distributed heterogeneous* infrastructure;
- the ESB provides the means to *manage* the service infrastructure.

From those aspects, the minimal set of ESB capabilities to enable SOA are derived and listed in Table 5.

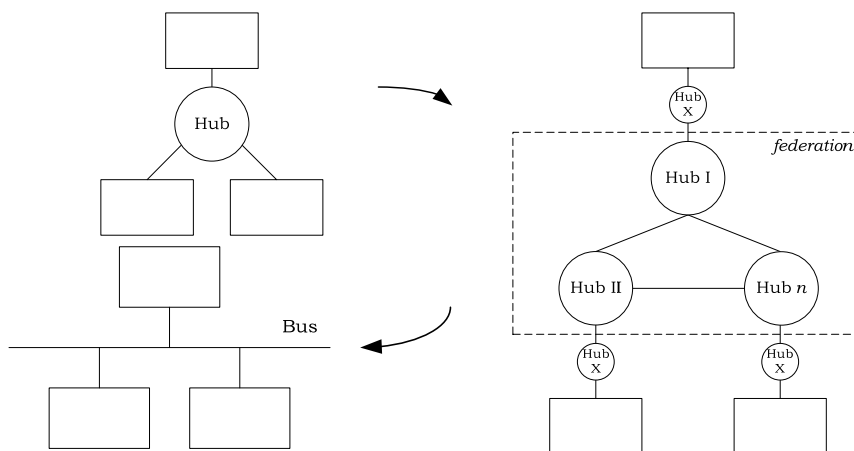
**Table 5: the minimum set of ESB capabilities [IBM04]**

| Category                   | Capability  | Reason  |
|----------------------------|---|---|
| 1. Communication           | <ul style="list-style-type: none"> <li>▪ Routing;</li> <li>▪ Addressing;</li> <li>▪ Pub / sub, <i>or</i> response / request;</li> <li>▪ At least one transport protocol;</li> </ul> | <ul style="list-style-type: none"> <li>▪ Provide location transparency;</li> <li>▪ Enable a distributed environment;</li> </ul>   |
| 2. Service interaction     | <ul style="list-style-type: none"> <li>▪ Service interface definition;</li> <li>▪ Service messaging models;</li> <li>▪ Substitution of service;</li> </ul>                          | <ul style="list-style-type: none"> <li>▪ Support integration in heterogeneous environments;</li> <li>▪ Support service substitution to increase service independency;</li> </ul>                                |
| 3. Integration             | <ul style="list-style-type: none"> <li>▪ Several integration styles or adapters;</li> <li>▪ Protocol transformation;</li> </ul>   | <ul style="list-style-type: none"> <li>▪ Support SOA principles, separating application code from specific service protocols and implementations;</li> <li>▪ Support a heterogeneous infrastructure;</li> </ul> |
| 8. Management and autonomy | <ul style="list-style-type: none"> <li>▪ Administration capability.</li> </ul>  | <ul style="list-style-type: none"> <li>▪ A point of control over service;</li> <li>▪ Manage addressing and naming of services.</li> </ul>   |

The enumerated capabilities form the set of minimal ESB capabilities. We state that a solution that provides the functionality as described above can be referred to as an ESB.

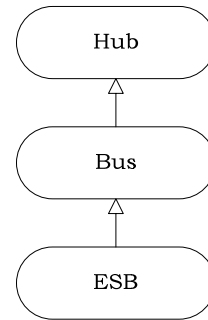
### 3.3.1. The architectural concept of an ESB

The ESB pattern can be misinterpreted as “just a combination of distributed hubs”. In this context, hubs are referred to as the centralized messaging components in an architecture. To clarify the concept of ESB from an architectural point of view, this pattern is compared with hubs and buses. A hub can be deployed as a component which acts as a middleman. In this role the component intermediates between service consumers and providers. When a hub is physically distributed as a set of federated hubs, the term bus is used. This is depicted in Figure 30 [IBM04].



**Figure 30: a set of federated hubs is referred to as a bus [IBM04]**

In case the bus, as a federation of hubs, also supports a centralized configuration by means of a virtual namespace manageable as a single entity, it can be referred to as an ESB. Concluding, an ESB is classified as a type of bus, which in turn is classified as type of hub. This relation is depicted in Figure 31.



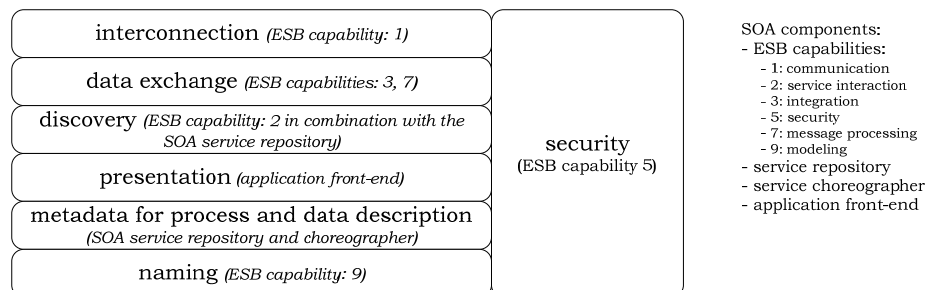
**Figure 31: the relation between the concepts of hub, bus and ESB [IBM04]**

### 3.3.2. Provided technical interoperability

The technical layer of required interoperability, involving hardware, operating systems, application platforms, and software can be fulfilled by adopting a loosely-coupled architectural pattern which enables SOA. Since the ESB is the product which provides such pattern, this is a potential candidate to achieve SOA. By creating a layer of technical interoperability, an environment for organizational and semantic interoperability can be realized. The following aspects can be regarded as pivotal characteristics of the ESB which contribute to a significant level of technical interoperability:

- the use of standards, possibly in combination with technology specific adapters enables technical compatibility, and therefore technical flexibility;
- content-based processing capabilities enables information sharing in the proper format and routing to the proper services;
- various security standards can be adopted and applied on specific services;
- scalability can be provided by deploying an ESB on clustered machines;
- extendibility can be achieved by connecting multiple ESB's;
- centralized monitoring and management of services enables a well-organized view of the services in operation, also in a widely distributed setting.

The way in which these characteristics can be used to provide municipalities with the proper level of technical interoperability, is illustrated by means of the technical interoperability framework which was introduced in section 2.4.3. Figure 32 depicts which SOA components are required to realize the proposed framework for use in a municipal context. Since the ESB implements the service bus component of SOA, the specific capabilities of this technology are used instead of solely referring to the service bus.



**Figure 32: SOA components and their capabilities in the technical interoperability framework**

As depicted in the figure above, all framework elements can be covered when combining ESB capabilities and the recognized SOA components. Again, we stress that every municipal situation is unique and an analysis has to be made to determine which features are useful for that specific situation. A suggestion on how to do this is outlined in chapter 4.

### 3.4. In Summary

Enterprise architecture can be used to provide insight in the structures, relations, information flows and products in an organization. We have adopted an enterprise architecture framework as proposed by Jonkers et al [JON03] which covers a municipality and the required interoperability by means of seven domains. By using these domains in combination with views to focus on relevant information, this information can be targeted at specific stakeholders. The business, application, and technology views have been used in this research. These views correspond with the layers of abstraction used in the adopted enterprise architecture framework. The techniques discussed in this research have been positioned in the enterprise architecture framework by relating it to a certain domain. We use the technologies' description and position in the framework to predict their influence on surrounding domains.

The Service-Oriented Architecture (SOA) is an application architecture which follows the Service-Oriented Computing (SOC) paradigm and can be positioned in the application domain of the enterprise architecture framework. This architecture is expected to provide the interoperability as required by the Dutch government. It describes a design, development and deployment approach for information systems, by defining concepts and general techniques for designing, encapsulating, and invoking reusable business functions through loosely bound service interactions. SOA can operate platform, time and location independently. Therefore this approach enables the development of highly distributed cross-organizational systems.

SOA consists of five key elements: application front-ends, services, a service repository, a service choreographer and a service bus. The application front-end is the part of the architecture through which an end-user communicates with the underlying applications or data-sources. A service is a unit of functionality provided by a service-provider to a requesting service-consumer. We recognize three service types which are in line with the adopted enterprise architectural views: (1.) business services, (2.) application services and (3.) technology services. A service repository contains the description of all services which are available for invocation by service consumers. The sequence in which they are invoked, eventually resulting in the representation of a business process, is determined by the service choreographer. Services communicate with each other through a service bus which is the infrastructural component of SOA. We position SOA in the application domain but stress that it seriously influences the information, process, organization, and infrastructure domains as well. Realizing this is essential in implementing an enterprise-wide SOA.

An Enterprise Service Bus (ESB) is an implementation of the service bus and provides a collection of infrastructural capabilities, implemented by middleware technology. In essence, it is a set of combined hubs that acts as the backbone of SOA that can be *centrally* monitored and managed. The ESB allows an enterprise to connect, mediate, and control the interaction between services across highly distributed and heterogeneous environments. Interoperability is ensured because the ESB supports international standards, adopts the use of technology specific adapters, and provides content-based processing and a high level of scalability.

The three recognized service types of SOA communicate with each other through the ESB. In that sense it is infrastructure. Because the ESB does not directly model physical infrastructure aspects such as cables and servers, it is referred to as software infrastructure. Therefore the ESB is positioned as the linking pin between the application and the technology layer in the enterprise architecture framework.

---

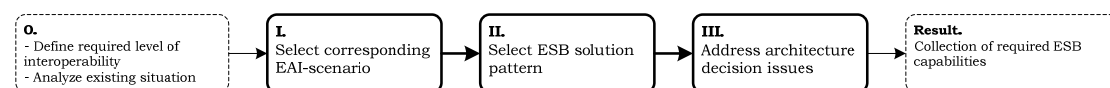
# 4. Selecting a suitable ESB solution

---

Every integration project poses specific requirements to a potential solution. Now we have investigated the requirements for interoperable government and the capabilities of available technologies, we need a capability selection approach to determine a suitable solution. This chapter describes three steps to determine an ESB-based solution which conforms to a certain municipal situation. For seven Enterprise Application Integration (EAI) scenarios, a number of solutions patterns are available in which the ESB has a key role. The required ESB capabilities are identified by addressing a number of architecture decision issues which are related to the EAI-scenarios and the suggested ESB solutions. The EAI-scenarios, ESB solution patterns, and architecture decision issues are suggested by IBM in [IBM04]. We adopt these elements and structure the selection approach by describing the required steps in detail. This clarifies the relation between the three involved components.

## 4.1. Selection approach

IBM describes seven EAI-scenarios, six ESB solution patterns and fourteen architecture decision issues to discuss ESB's role in SOA in [IBM04]. Since the component descriptions are quite technical and no overall selection approach has been suggested, an overview that outlines the existing relations is useful. We adopt the suggested elements and relate them to each other by distinguishing three selection steps, which are rather straightforward. These steps, as depicted in Figure 33, prevent the reader from getting entangled in the vast amount of provided information and possibilities. We relate the involved components by means of the summarizing Table 5, the remainder of this chapter discusses these components in more detail.



**Figure 33: the three steps of our suggested selection approach**

After assessment of the current and envisioned situation of a municipality where integration of applications is required, a coarse match can be made with a typical *EAI-scenario*. This corresponds with the first step: *Select corresponding EAI-scenario*. For this research we distinguish seven scenarios, which are described in detail in section 4.2. For these scenarios a number of *ESB solution patterns* are available to provide the required capabilities. Which pattern is suitable depends on the current municipal situation and the adopted eGovernment strategy, therefore these aspects have to be studied properly prior to this second step. To exactly identify the required ESB capabilities (see Table 4 for a summary of the available capabilities), a number of *architecture decision issues* should be addressed, forming the third selection step. The mentioned ESB solution patterns and architecture decision issues are described in detail in section 4.3. and 4.4. respectively. The result of this selection is a number of ESB capabilities which are considered to be required for the analyzed situation.

We stress that the suggested solution patterns are *base patterns*, the foundation of a solution which might be subject to extension in certain situations. The

suggested combination of scenarios, candidate solutions and related architecture decision issues is presented in Table 6.



**Table 6: the distinguished EAI-scenarios, their ESB solution patterns and related issues [IBM04]**

| <b>EAI-scenario (section 4.2)</b> |   | <b>Candidate ESB solution patterns (section 4.3)</b>   | <b>Architecture decision issues (section 4.4) *</b> |
|-----------------------------------|---|--|---|
| <b>1.</b>                         | Basic integration of two systems;   | <ul style="list-style-type: none"> <li>▪ Implement basic integration using wrappers or adapters (<i>Basic adaptors</i>);</li> <li>▪ Or, with future expansion in mind, either:               <ul style="list-style-type: none"> <li>○ Add a controlling <i>Service gateway</i>;</li> <li>○ Implement a sophisticated infrastructure (<i>Web Services compliant broker</i>, or <i>EAI-infrastructure for SOA</i>).</li> </ul> </li> </ul>                                     | 1,3,4,6,10,13                                       |
| <b>2.</b>                         | Facilitate wider connectivity to one or more applications;  | <ul style="list-style-type: none"> <li>▪ Implement basic integration using wrappers or adapters (<i>Basic adaptors</i>);</li> <li>▪ Or add a controlling <i>Service gateway</i>;</li> <li>▪ If more sophisticated capabilities are required, implement a <i>Web Services compliant broker</i>, or an <i>EAI infrastructure for SOA</i>;</li> <li>▪ If process choreography is required, implement <i>Service choreographer</i> or <i>Full SOA infrastructure</i>.</li> </ul> | 1,2,3,4,6,8,9,10,11,12,13,14                        |
| <b>3.</b>                         | Facilitate wider connectivity to legacy systems   | <ul style="list-style-type: none"> <li>▪ Implement basic integration using wrappers or adapters (<i>Basic adaptors</i>);</li> <li>▪ Or, with future expansion in mind, either:               <ul style="list-style-type: none"> <li>○ Add a controlling <i>Service gateway</i>;</li> <li>○ Implement a sophisticated infrastructure (<i>Web Services compliant broker</i>, or <i>EAI-infrastructure for SOA</i>).</li> </ul> </li> </ul>                                     | 1,2,3,4,7,8,9,10,11,13,14                           |
| <b>4.</b>                         | Facilitate wider connectivity to an enterprise application integration infrastructure                         | <ul style="list-style-type: none"> <li>▪ Extend the EAI-infrastructure using open data formats (<i>EAI-infrastructure for SOA</i>);</li> <li>▪ Add a <i>Service gateway</i>;</li> <li>▪ Or add open standard support to the infrastructure (<i>Web Services compliant broker</i>).</li> </ul>  | 3,4,5,8,9,11,13,14                                  |
| <b>5.</b>                         | Implement controlled integration of services or systems between company divisions or partnering organizations | <ul style="list-style-type: none"> <li>▪ Add a <i>Service gateway</i>;</li> <li>▪ If more sophisticated capabilities are required implement <i>Web Services compliant broker</i>.</li> </ul>   | 1,2,3,4,6,8,9,10,11,13,14                           |
| <b>6.</b>                         | Automate processes by choreographing services   | <ul style="list-style-type: none"> <li>▪ If the direct connection of services is possible implement a <i>Service choreographer</i>;</li> <li>▪ If more sophisticated integration or control is required, implement a <i>Full SOA infrastructure</i>.</li> </ul>  | 1,2,3,4,6,10,11,12,13,14                            |
| <b>7.</b>                         | Implement SOA infrastructure with high quality of service and Web Services support                            | <ul style="list-style-type: none"> <li>▪ Implement a <i>Full SOA infrastructure</i>.</li> </ul>  | 1 to 14   |

**\* used architecture decision issues**

- |   |  |
|---|--|
| <ol style="list-style-type: none"> <li>1. Function and data interfaces;</li> <li>2. Common data model;</li> <li>3. Technologies for interoperability;</li> <li>4. Advanced interaction characteristics;</li> <li>5. Adoption of standards at the edge or in the heart of infrastructure;</li> <li>6. Technology support in existing systems;</li> </ol> | <ol style="list-style-type: none"> <li>7. Legacy XML support and processing;</li> <li>8. Availability of services in EAI-infrastructure;</li> <li>9. Service provider protection;</li> <li>10. Consistent and controlled service enablement;</li> <li>11. External access to services;</li> <li>12. Business service choreography;</li> <li>13. Service level requirements;</li> <li>14. Security requirements.</li> </ol> |
|---|--|

Now we have described the three selection steps to determine the required ESB capabilities, we discuss the EAI-scenarios, the ESB solution patterns, and the architecture decision issues in the following sections.

## 4.2. EAI-scenarios involving an ESB

This section describes seven EAI-scenarios which are likely to occur within an organization at a certain point of time. These scenarios are used in our analysis to recognize typical situations in municipal environments where integration is required [IBM04]. Each scenario is accompanied with a schematic representation of the particular pattern to help the user interpret the description correctly.

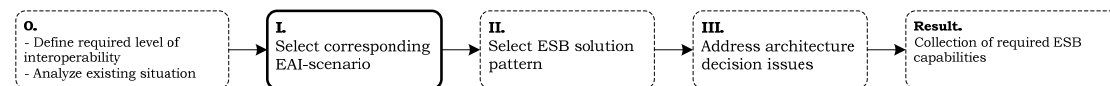
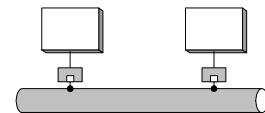


Figure 34: the first selection step: select a corresponding EAI-scenario

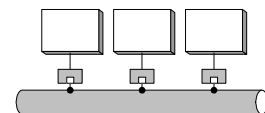
### 4.2.1. Basic integration of two systems

Two systems, using different technologies, need to be connected using basic integration. The Simple Access Object Protocol (SOAP), which can be interpreted as a Web Services standard or messaging middleware, is a potential technology to facilitate the envisioned level of connectivity between the systems. To some extent, the chosen technology has to be supported by both system environments. An important issue in this scenario is the question whether the solution has to provide future integration with *additional* systems. The adoption of an *extensible* technology can facilitate this need. However, this need has to be validated against the initial simplicity of connecting two systems.



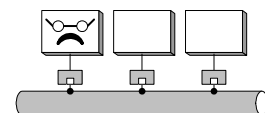
### 4.2.2. Facilitate wider connectivity to one or more applications

Commercial Off-The-Shelf (COTS) applications such as CRM and ERP packages, or custom-made applications developed in J2EE or other application server environments, contain functions which have to be reused in other applications. These functions can be opened up by exposing them as services to enable communication between applications or to provide new channels with this functionality. The adoption of interoperable or open standards as communication or service protocol is advised to increase the potential scope of the application in context.



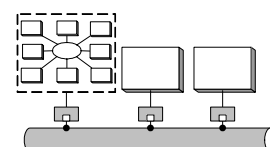
### 4.2.3. Facilitate wider connectivity to legacy systems

One or more legacy systems which support an organizations' core business have to be opened up to various other systems. The targeted functionality remains in both business transactions and data access, for example transactions that query account balances, create orders, schedule or track documents. Because of past investments and the key role in the company, replacement of the legacy system is not an option. Interoperable or open standard service-based access to this functionality provides significant added value for the organization since key functionality can be reused throughout the organization.



### 4.2.4. Facilitate wider connectivity to an enterprise application integration infrastructure

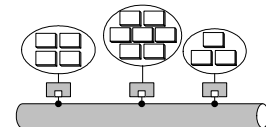
Organizations may have partially adopted an enterprise application integration infrastructure such as IBM's



WebSphere, Oracle's Fusion, or Microsoft's BizTalk already. This infrastructure has to be opened up to other applications using interoperable or open standards-based access. The exposure of service interfaces defined in terms of XML business data through an interoperable protocol such as HTTP can provide a sufficient level of interoperability. However, support for WSDL and SOAP Web Services standards might be required if a proprietary extension of the current integration solution does not provide the envisioned interoperability.

#### **4.2.5. Implement controlled integration of services or systems between company divisions or partnering organizations**

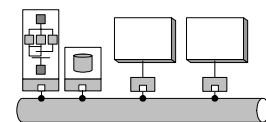
This scenario tends to open up one or more systems deployed in the organization to customers, divisions which operate rather autonomously and partnering agencies. These external entities should have the possibility to directly connect to the (legacy) applications or functions deployed in various parts of the organization. This situation requires a manageable secure access for the external parties which intend to connect to the organization. Additionally, open standards in service and data protocols are required to achieve this level of interoperability.



When department specific data models are operational, these models have to be interconnected by means of an organization-wide common data model.

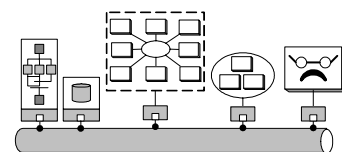
#### **4.2.6. Automate processes by choreographing services**

COTS applications such as CRM and ERP packages, or custom-made applications developed in J2EE or other application server environments contain functions which have to be reused in other applications. These functions can be opened up by exposing them as services and combine these services to form business processes at some level. Orchestration of the services to shape and execute such processes should be done using appropriate (and possibly open standards-based) modeling and process execution technology such as BPEL.



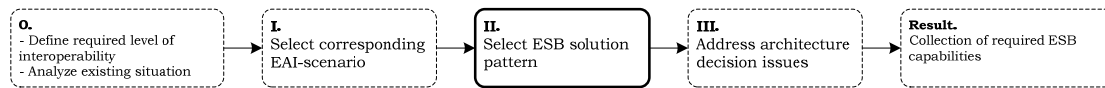
#### **4.2.7. Implement SOA infrastructure with high quality of service and Web Services support**

When all previously outlined scenarios are combined, this scenario is obtained. Widespread internal access to services provided by multiple applications, legacy or otherwise, is required. Security, aggregation, transformation, routing and service choreography capabilities are required to fulfill this implementation. This scenario is typically initiated by an ICT-organization which has to facilitate widespread and flexible integration between systems to support business activities operating in a dynamic environment.



### **4.3. ESB solution patterns**

A number of coupling patterns that involve extensive use of an ESB are discussed in this section. These ESB solution patterns are used to coarsely determine the set of requirements. Specific ESB capabilities may be required to meet certain needs. A set of *architecture decision issues*, as discussed in section 4.4, can be used to identify these capabilities [IBM04]. The suggested solutions have an increasing level of complexity.



**Figure 35: the second selection step: determine suitable ESB solution pattern**

### 4.3.1. Basic adapters

This solution suggests the use of basic adapter, or wrapper technologies to achieve point-to-point integration opposed to using a complete ESB package. The adapters are used to translate one interface into another. The interface can be defined using WSDL after which SOAP or another interoperable technology is used to enable communication between both interfaces. This solution is extendable when accepted standards are used. In case the interfaces are described using WSDL, which is already adopted by a large number of integration packages, future integration projects are supported as well.

The suggested adapters do not have to be developed in-house necessarily. Some companies have specialized themselves in providing technology specific adapters which are able to communicate with certain applications, protocols or file-systems such as .NET, J2EE, CICS, or packages from SAP, Oracle, Centric, IBM, Getronics PinkRoccade and Siebel.

Additional benefits can be found in the minimal requirements which are posed to the existing infrastructure. Since this solution uses recognized standards, it usually can be developed and deployed on the existing (non-legacy) platform. More advanced features such as security and management capabilities are not provided by the adapters and should be available in the existing applications when this is required.

When the provided features do not suffice, or near-future extension is planned, it might be a good idea to consider the service gateway or an even more sophisticated solution pattern.

### 4.3.2. Service gateway

This pattern suggests a basic ESB implementation where the ESB acts as gateway between the deployed services or applications. It provides the client connectivity using SOAP/HTTP, interface binding using various adapters and more advanced features such as service routing, protocol transformation and basic security. The latter is useful when external partners require access to operational systems. The transformation of protocol can come in handy in such situations as well, for example when connected partners have adopted varying protocols.

Implications for the infrastructure are minimal, some gateway technology should be available or introduced, but the use of standards and the availability of adapters simplifies infrastructure requirements. This pattern is a good starting point to introduce an ESB into an organization. From this point, the solution can be further extended when the situation requires so.

### 4.3.3. Web Services compliant broker

This pattern extends the service gateway with all capabilities required to provide a sophisticated ESB-solution. The situation specifies precisely which capabilities are required. All Web Services standards (WS-\*) can be adopted, but the exact set of required capabilities depends on the situation.

Some of the Web Services standards have not been completely matured yet (e.g. WS-Policy and WS-Transaction). The other, matured, standards provide a full range integration platform. Since this pattern has a large scope, the implications for the organization grow as well. Specific knowledge of the adopted standards is required and should be embedded in the ICT-management organization.

### 4.3.4. EAI-infrastructure for SOA

Whenever the Web Services standard does not fit in an organization, but the principles of SOA can provide significant benefits. This can be achieved using

proprietary, bespoke or alternative open standards and technologies. Enterprise application integration middleware provided by commercial vendors can provide such an infrastructure. The main benefits can be found in the maturity of EAI-products and with that, a vast amount of ESB capabilities. During the years the products have been extended and stabilized, a user-group is founded and a large number of standards is supported by default. When an organization already has a line of products of a vendor, this pattern is advised when direct connectivity between a large number of applications is required. Nowadays, a lot of large vendors are transforming (or rebranding) their EAI-products as service-oriented. Whether they truly are should be analyzed for each package individually when service-orientation is specifically required. Disadvantages can be found in the costs of such EAI-products. Next to the licensing costs, future costs can be expected when proprietary standards are to be migrated to new, currently evolving standards. Whenever an EAI-product is based on such open standards, these costs may decrease.

### 4.3.5. Service choreographer

A service choreographer is responsible for the sequence in which various services are invoked. As discussed in section 3.2.4, the choreographer is one of the five main components of SOA and is deployed on the service bus as a service as well. Since the use of this technology requires specific ESB-capabilities, we have adopted it explicitly as an ESB solution pattern.

By means of process modeling, business processes can be modeled as a sequence of services. The choreographer is responsible for invoking the proper service, as part of the process, at the right time. It is highly advised to adopt the Web Services standards for service implementation and communication. Emerging standards for process modeling support these services as well, for instance the Business Process Execution Language for Web Services (BPEL4WS). When proprietary standards are adopted for service delivery within the municipality, significant aggregation and correlation functionality is required when the deployed services ought to be opened up by means of Web Services.

### 4.3.6. Full SOA infrastructure

This pattern suggests a full SOA infrastructure as discussed in section 3.2. It combines a service bus implementation with the service choreographer. An important issue is whether the service bus implementation has to be based on proprietary standards (using EAI-infrastructure for SOA), or open standards such as the Web Services (using a Web Services compliant broker).

## 4.4. Architecture decision issues

This section covers the fourteen architecture decision issues which an organization should address after a suitable solution pattern has been selected. By studying these issues in the context of a certain situation, the set of *relevant* ESB capabilities can be selected. Since these issues are aimed at specialists, this section contains technical abbreviations which are explained in Appendix I [IBM04].

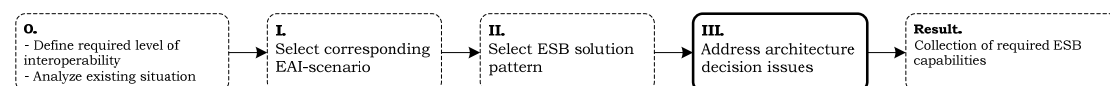


Figure 36: the third selection step: address relevant architecture decision issues

### 4.4.1. Function and data interfaces

Do the existing functions and their data interfaces align with the envisioned services, or can appropriate modification or aggregation be performed in the applications to realize this alignment?

If not, the following capabilities are required either in adapters, the ESB infrastructure, or by the service consumers themselves:

- 3. Integration:
  - service / message aggregation.
- 7. Message processing:
  - message and data transformations;
  - message /service aggregation and correlation.

#### **4.4.2. Common data model**

Should the services be exposed using an adopted common data model? If so, do the systems that implement those services already support that model, or can they be made to do so?

If not, or when various business models are used throughout the organization, the following capabilities are required either in adapters or the ESB infrastructure:

- 3. Integration:
  - service / message aggregation.
- 7. Message processing:
  - message and data transformations.

#### **4.4.3. Technologies for interoperability**

Are open standards required, or can appropriate interoperability be achieved through EAI middleware? If open standards are required, which ones conform to the current and future requirements?

- Although the use of open standards is one way to achieve interoperability, proprietary EAI-middleware can be highly interoperable as well, at least when compatible systems are involved. Organizations might already have an extensive existing infrastructures that, in *some* scenarios, can minimize the benefits of open standards. This alternative does decrease the level of *real* interoperability and introduces the risk of vendor lock-in. With other words, proprietary software can provide interoperability, but one should be aware of the related risks;
- In scenarios where open standards are preferred or required, Web Services are perhaps the most obvious choice in the municipal context. However, JMS, JDBC, basic XML, or several other technologies such as EDI or industry XML formats can also be applied;
- In practice, interoperability between different implementations of the same standards cannot always be assumed, particularly if the standards are recent or emerging. In the case of Web Services, the Web Services Interoperability Organization has published the Basic Profile for interoperability [WSI04]. Until such profiles are comprehensive, established, and widely supported by products, the use of open standards does not *guarantee*, and may not always facilitate, interoperability;

Relevant capabilities which should be considered:

- 1. Communications:
  - routing;
  - addressing;
  - protocols and standards;
  - an appropriate messaging mechanism.
- 2. Service interaction:
  - service interface definition;
  - service messaging models.
- 3. Integration (all capabilities);
- 4. Quality of service (all capabilities);
- 6. Service level (all capabilities);
- 7. Message processing:
  - message and data transformations.

#### **4.4.4. Advanced interaction characteristics**

Is support for basic communication protocols and standards such as SOAP and WSDL, or more sophisticated capabilities such as WS-Security and WS-Transaction required? Requirements to support more sophisticated standards pose significant constraints on the options for implementation technologies and may imply the use of less mature technologies. The following capabilities are relevant for this issue:

- 3. Integration:
  - protocol transformation;
- 4. Quality of service (all capabilities);
- 5. Security (all capabilities).

#### **4.4.5. Adoption of standards at the edge or in the center**

Where changes to used message formats and protocols are considered, including the adoption of open standards, do the changes apply to the entire infrastructure, or can they be applied at the edges? If EAI-technology is in use or considered, does that have its own internal format, or can it process open standards as an internal format?

- Any use of open standards is likely to be driven by needs to extend access, so it is usually more important that they are available at the interfaces to existing infrastructure than that they are used internally.
- If internal use of specific formats, technologies or standards is required, this places constraints on the choice of implementation technology and involves the risk of vendor lock-in.

The following capabilities should be addressed to cover this issue:

- 1. Communications:
  - routing;
  - addressing;
  - protocols and standards;
  - an appropriate messaging mechanism.
- 2. Service interaction:
  - service interface definition;
  - service messaging models.
- 3. Integration (all capabilities);
- 4. Quality of service (all capabilities);
- 6. Service level (all capabilities);
- 7. Message processing:
  - message and data transformations.

#### **4.4.6. Technology support in existing systems**

Do the systems that implement functions which should be exposed as a collection of services support the required technologies or open standards such as SOAP, JMS, or XML?

If not, either the ESB infrastructure or adapters need the capability to transform between the required open standards and the formats that are supported by the service providers:

- 1. Communications:
  - routing;
  - addressing;
  - protocols and standards;
  - an appropriate messaging mechanism.
- 2. Service interaction:
  - service interface definition;
  - service messaging models.
- 3. Integration (all capabilities)
- 7. Message processing:
  - message and data transformations.

#### **4.4.7. Legacy XML support and processing**

Where access to legacy systems is required using more recent XML-based technologies, is direct support available, or is a separate adapter required? Does the legacy platform support XML processing and is such processing a sensible use of the platform capabilities?

If for any of these reasons a required SOAP or XML capability is not made available on a legacy platform, appropriate transformation capability is required either in adapters, in an integration tier, or in the ESB infrastructure:

- 3. Integration:
  - legacy and application adapters;
  - protocol transformation.
- 7. Message processing:
  - message and data transformations.

#### **4.4.8. Availability of services in an EAI-infrastructure**

Whenever an EAI-infrastructure is already available, does it implement services as message flows with appropriate function and interface granularity, or can it be made to do so? What connectivity protocols does it support (e.g., JCA, SOAP, WebSphere MQ, RMI)?

If existing message flows do not provide the required services, then additional flows are needed to perform transformations. If the EAI-technology does not directly support the required standards, a gateway component can be added:

- 1. Communications:
  - routing;
  - addressing;
  - protocols and standards;
  - an appropriate messaging mechanism.
- 2. Service interaction:
  - service interface definition;
  - service messaging model.
- 3. Integration:
  - connectivity to EAI-middleware;
  - service mapping;
  - protocol transformation.
- 7. Message processing:
  - message and data transformations;
  - message / service aggregation and correlation.

#### **4.4.9. Service provider protection**

Which protection measures should be applied to service consuming channels in the form of workload buffering, security, logging, etc. to 'protect' the service provider from overloading or hack attempts?

Buffering is often a role of the ESB infrastructure and defines some of the capabilities it requires. If specific service provider systems (such as legacy transactional systems) have additional needs for protection, a dedicated integration tier with specific adapters should be considered:

- 1. Communications:
  - messaging mechanism: synchronous and asynchronous messaging.
- 3. Integration:
  - legacy and application adapters;
  - protocol transformation.
- 5. Security (all capabilities)
- 6. Service level (all capabilities)
- 7. Message processing:
  - store and forward.
- 8. Management and autonomy:
  - logging;



- metering;
- monitoring.

#### **4.4.10. Consistent and controlled service enablement**

How many services should be enabled? What aspects of enablement should be consistent across the services, and how can consistency be enforced, perhaps across multiple platforms and applications?

If few services are involved, a point-to-point integration model may be appropriate. If more services are involved, or are likely to become so over time, the addition of a control point such as provided by an ESB is beneficial:

- 1. Communications:
  - routing;
  - addressing;
- 2. Service interaction:
  - substitution of service implementation;
  - service directory and discovery.
- 3. Integration:
  - service mapping;
  - protocol transformation.
- 5. Security (all capabilities)
- 8. Management and autonomy:
  - service provisioning and registration;
  - logging;
  - metering;
  - monitoring.

#### **4.4.11. External access to services**

Are the service interactions contained within the organization or will they be accessed by third parties? As the requirements for security and service routing typically differ for services made available externally, a gateway component is suggested to provide additional control for these services:

- 1. Communications:
  - routing;
  - addressing.
- 2. Service interaction:
  - substitution of service implementation;
  - service directory and discovery.
- 3. Integration:
  - service mapping;
  - protocol transformation.
- 5. Security (all capabilities)
- 8. Management and autonomy:
  - service provisioning and registration;
  - logging;
  - metering;
  - monitoring.
- 9. Modeling:
  - public versus private models for business-to-business integration.

#### **4.4.12. Business service choreography**

Is business service choreography required, and does this involve short- or long-lived (stateful) processes, or both? Do these processes involve activities which are performed by employees?

- Where the defined requirements involve *business* functions, choreography should be implemented in the service choreographer component, as discussed in section 3.2.4, *separate* from the ESB. The requirement to

support long-lived stateful processes or manual activities places constraints on the choice of implementation technology;

- Long-lived services or event models may require message processing: message / service aggregation and correlation capabilities;

The following capabilities should be considered when a service choreographer is required for business service choreography:

- 1. Communications:
  - messaging mechanism: asynchronous messaging.
- 4. Quality of service (all capabilities);
- 6. Service level (all capabilities);
- 7. Message processing:
  - message and data transformations;
  - message / service aggregation and correlation.

#### **4.4.13. Service level requirements**

What service level requirements, such as service response time, throughput, and availability, should be supported by the infrastructure? Future plans should be anticipated for in terms of scalability.

- Some of the candidate technologies for ESB implementation are relatively new and may only have been tested against limited service levels. Similarly, because the relevant open standards are either recent or emerging, support for them in more established products and technologies is also new;
- For the foreseeable future, critical architectural decisions are concerned with balancing the benefits of specific open standards supported by emerging or mature product technologies against service level requirements. These point-in-time decisions have to recognize that some standards, and product support for them, are relatively mature (such as XML and SOAP), some (such as WS-Security) are newer, and some (such as WS-Transaction) are still emerging;
- The trade-off between the benefits of standards and proven service-level characteristics often drive a mixed approach that combines standards-compliant technologies with proprietary or customized technologies in an ESB and SOA architecture.

Relevant capabilities which should be considered:

- 1. Communication:
  - routing;
  - addressing
  - protocols and standards;
  - an appropriate messaging mechanism;
- 3. Integration:
  - connectivity to EAI-middleware;
  - protocol transformation;
- 4. Quality of service (all capabilities);
- 5. Security (all capabilities);
- 6. Service level (all capabilities);
- 7. Message processing:
  - message and data transformations;

#### **4.4.14. Security requirements**

Is a point-to-point or end-to-end security model required? With other words, should the ESB simply authorize service requests, or should it pass the requestor identity or other credentials through to the service provider? Is there a need to integrate the service security model with application or legacy security systems?

If point-to-point security is acceptable, several existing solutions (such as SSL, J2EE security for database access, and adapter security models) can be applied. If

end-to-end security is required, the WS-Security standard is a possibility if it is supported by all of the involved systems. Alternatively, a customized model using custom message headers or passing security information as application data could be used.

Relevant capabilities are:

- Security (all capabilities).

## **4.5. *In Summary***

IBM defines a number of EAI-scenarios, ESB solution patterns and architecture decision issues. We suggest an approach which clarifies existing relations between these three components and determines the ESB capabilities that meet the requirements as posed by a municipality. First, an EAI-scenario which resembles the municipal situation to a large extent should be identified. Next, an ESB solution pattern can be selected. Thirdly, the advised ESB capabilities are distinguished by addressing a number of architecture decision issues which are related to the EAI-scenario and the ESB solution pattern. After these issues have been addressed, a collection of relevant ESB capabilities can be derived.

Since the suggested solution is a base pattern, the foundation of a solution, adjustments might be required to align better with the situation in context.

---

# 5. Consequences of ESB-facilitated SOA

---

To bridge the gap between the current situation and the envisioned interoperable environment at municipalities, some changes can be expected and have to be anticipated for. The majority of these changes typically occur at the organizational, information, and technical dimension of interoperability. This chapter investigates the consequences for municipalities in these dimensions when they adopt SOA, facilitated by an ESB. This investigation was performed by means of an impact-of-change analysis at four Dutch municipalities using the BEST reference framework. This framework can be used to assess enterprise-wide implementation projects in an organization.

During our analysis, the situation concerning a simple but realistic scenario was compared with the envisioned (service-oriented) situation, as facilitated by an ESB. The selected municipalities were all *initiating* relevant activities concerning interoperability between front- and back-office, and in the future, external partners. Based on the analysis, conclusions at the three distinguished dimensions of interoperability can be drawn concerning the consequences of (1.) applying an ESB and (2.) SOA implementation.

## 5.1. *The BEST reference framework*

The Better Enterprise SysTem (BEST) implementation project studied a significant number of enterprise-wide implementation projects and related implementation process dynamics. The goal was to specify an assessment approach to determine the maturity of an organization that is about to initiate an enterprise-wide system implementation project and to identify human and organizational risks involved in such a project. Using the gathered information, a process-based reference framework was introduced which can support an organization in preparing an implementation project. We have adopted this framework to assess the potential implementation of an ESB in a municipal environment. We assumed such a project aims at: (1.) provide an interoperable integration infrastructure consistent with the principles of SOA and (2.) gradually adopt these principles to further extend and improve ESD within municipalities and with external service consumers.

The BEST framework consists of the following three dimensions which represent the main processes existing in an enterprise system implementation project [WOG04]:

- *Permanent business processes (BP)*. These are the business processes for which (part of) the system is implemented. In the context of this research, the delivery of a public service and all underlying activities can be regarded as the permanent business processes. We have highlighted two simple but realistic scenarios to act as these business processes: “requesting a felling license” and “getting married”.
- *Enterprise system design processes (ES)*. These processes cover all activities which are required to adapt or tune the envisioned solution and align it with the current organizational situation. For this dimension, seven possible enterprise application integration scenarios in which an ESB has a central role have been introduced previously. Based on these scenarios, a candidate solution can be selected. This solution will be regarded as the enterprise system.

- *Project management processes (PM)*. All activities required to plan and monitor the implementation process are covered by these project management processes. This includes selection, realization and evaluation of the implementation strategy, implementation organization, project team establishment and management of project documentation. The project team acts as a boundary spanner between the development and operational organization, and is especially important in large organizations.

For each of these three dimensions, six elements have been identified [WOG04]:

- *Strategy and goals*. This covers the short-, mid- and long-term plans to achieve the goals which are set, or which have to be set, for each of the three dimensions. The project's goals and strategy should preferably be in line with the common business goals and strategy;
- *Management*. This refers to management-related activities, such as setting priorities, assigning resources, and planning and monitoring processes which are required to achieve the goals as described by the "Strategy and goals" element;
- *Structure*. All relations between organizational system elements such as processes, employees and means are covered by this aspect. Team, process and system structures, related tasks, authorities and responsibilities belong to this aspect as well;
- *Process*. This aspect refers to the set of required changes or adaptations in each of the three main processes (BP, ES, and PM) to reach the envisioned situation;
- *Knowledge and skills*. This covers specific knowledge and skills in various disciplines that are required to carry out an implementation project. Knowledge and skills provided by tools, methods and technology are covered as well. This aspect also includes the number of resources available for performing the activities.
- *Social dynamics*. The behavior of employees, their norms, rituals, commitment, awareness, attitude, and political behavior are important aspects for enterprise-wide projects, but difficult to analyze as well.

The combination of the enumerated aspects with the three dimensions results in the process-based reference framework as proposed by the BEST-project and depicted in Figure 37.

|                      | Permanent business | Enterprise system | Project management |
|----------------------|--------------------|-------------------|--------------------|
| Strategy and goals   |                    |                   |                    |
| Management           |                    |                   |                    |
| Structure            |                    |                   |                    |
| Process              |                    |                   |                    |
| Knowledge and skills |                    |                   |                    |
| Social dynamics      |                    |                   |                    |

**Figure 37: the BEST reference framework for enterprise system implementation processes**

The derived framework is used to categorize a set of questions which can be found in Appendix VI. Using these questions, the readiness of an organization to initiate and fulfill an enterprise-wide implementation project can be analyzed. We have interviewed a number of key figures within the selected municipalities to be able to answer these questions. Since the municipalities all have their own characteristics and interesting aspects, we highlight a specific point of interest for each one.

The research performed by Wognum et al. involved a large number of companies. The following issues were recognized as potential problem areas which require special attention throughout the whole implementation project:

- lack of management involvement;
- people resistance and fear for change;
- adaptation to the system;
- lack of an explicit business goal and strategy for implementing the system;
- management of the implementation project.

These issues were kept in mind while performing the interviews and analysis at the selected municipalities. Based on the results, a number of problem areas were highlighted and positioned in the framework.

### **5.1.1. Permanent business processes: two scenarios**

Since a complete analysis of an enterprise-wide system implementation goes beyond the scope of this research, we have highlighted two permanent business processes which are put into practice by means of two scenarios. These scenarios cover the organization sufficiently to address the interoperability problems subject to this research. First we give a common description of each scenario. In the actual analysis, we provide specific information on how each studied municipality deals with such a scenario.

#### **Scenario I “Requesting a felling license”**

This scenario covers the request of a felling license within the built-up area posed by citizen Mr. Joe Schmoe. We assume Joe Schmoe owns the ground where the tree is located on. No nation-wide regulations have been defined concerning a felling license. Commonly, a license is required for felling, when:

- The circumference of the tree is larger than 100 centimeters, measured at a height of 1,30 meters;
- The tree is fifty years or older;
- The tree is located in a designated (protected) area.

Additionally, a felling license request is usually published in a municipal magazine or regional newspaper to inform neighboring parties. Any objection by such parties against the license request should be received by the municipality within three weeks after publication. In case a license is granted, a notice of dishonor (“bezwaarschrift”) can be issued up to six weeks after the granting date.

#### **Scenario II “Getting married”**

This scenario describes an important live-event of citizens: getting married. This occurrence involves several municipal administrations. Before our happy couple, Mr. Joe Schmoe and Miss Jane Doe, can get married, their intention to get married has to be registered at the civil office by applying for an “issue of intended marriage for public inspection” (“ondertrouw”). To qualify for this procedure, some personal documents are required. This scenario describes the simplest situation possible where both Joe and Jane have the Dutch nationality, have not been married before, are both adults, and no exceptions apply. The following documents are required in this case:

- birth certificate;
- proof of identity (e.g., a passport);
- excerpt of the municipal register (GBA);
- information concerning the witnesses (passport copies).

After the public inspection concerning marriage has been fulfilled, the ceremony can take place officially, resulting in a marriage certificate. Completion of this event poses additional changes for the information related to the actors: registered personal information and records should be updated in the various administrations. Possibly, the couple decides to live together and a change of address process has to be triggered as well.

### 5.1.2. Enterprise system design

For the two scenarios introduced, we made an inventory of involved processes, employees and facilitating services in four relevant Dutch municipalities. We tried to match the acquired municipal situation and the envisioned integration goals with one of the EAI-scenarios as introduced in chapter 4. Next, an ESB solution pattern with a collection of ESB capabilities is suggested. The proposed solution is used as the enterprise system dimension in the BEST framework.

Some schematic designs of the envisioned solution are used to clarify the use of services in our solution. The adopted notation is described in Appendix IV.

### 5.1.3. Project management

The project management dimension of the BEST framework covers all activities required to plan and monitor the intended implementation process. Since not all involved municipalities have actually *initiated* an enterprise-wide implementation project yet, it is difficult to perform an extensive and complete analysis on of this dimension for each municipality. For example, relations with third-party suppliers of ESB and SOA technology are hard or even impossible to analyze since they do not exist (yet). However, our analysis mainly aims at the *impact* of ESB and SOA implementation in a municipal environment, rather than the success of such a project and the role project management processes have in this. Therefore the lack of information in this dimension did not affect the outcome of our research.

The potential problems presented to those responsible for the implementation relate to the project's characteristics [BOD05]. Two dimensions can be distinguished which have significant influence on the importance of proper project management. The first dimension questions how the envisioned project is related to the organization's primary task. This dimension ranges from core (the system highly affects the core of an organization), to marginal (the system barely affects the organization). The second dimension scales the novelty of the system for the organization, ranging from novel (no experience with similar systems is available) to familiar (significant experience with similar systems is available). Using these dimensions as X, and Y-axis, four quadrants are derived, as depicted in Figure 38.

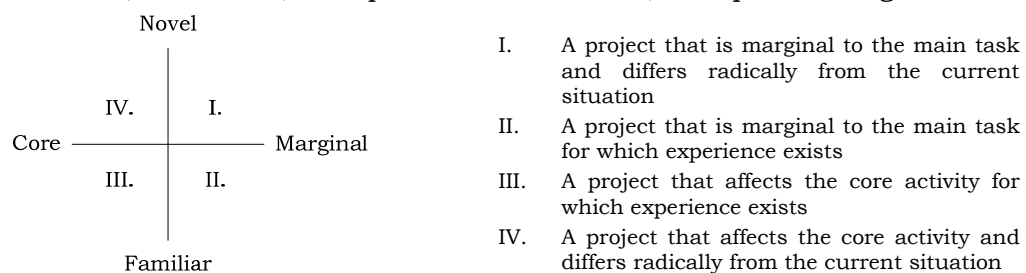


Figure 38: critical dimensions of ES-projects [BOD05]

The derived categories are quite subjective; persons with a differing background are likely to interpret the derived categories differently. But a small indicator to keep in mind what kind of project we are dealing with can be useful to maintain involved parties focused. Keeping this in mind we state that project management *always* is an important issue. However, inexperienced organizations that plan to implement a system that affects the primary process should pay a lot more attention to project management activities opposed to experienced organizations that envision a system that barely influences the organization's heart. To which extent an enterprise-wide system as analyzed in this research affects the primary task of an organization

depends on the defined objectives. But ESB and SOA are EAI technologies that cover the organization to a *significant* extent. Combined with the fact that municipalities are typically *inexperienced* with big implementation projects, we position these projects in the fourth quadrant. For municipalities this means that external knowledge concerning the implementation project, both for project management and implementation tasks, is useful and should be seriously considered.

Wijnen, Renes and Storm state that projects which are initiated by governmental agencies have a variation of possible project-structures [WIJ99]. These structures typically lack a clear definition of their own role, or define a conflicting role; for example when a municipality is a solution's end-user, developer and owner at the same time [WIJ99]. During our analysis we keep this in mind and study how the municipality's role has been determined and whether conflicts are likely to arise.

To set the context for this part of our research, we have provided an extensive description of the ICT-organization structure of each municipality, their formal status, related authorities and available skills. We keep the following recognized aspects concerning enterprise-wide implementation project management in mind during our analysis at the selected municipalities [MUN01][WIJ99]:

- Clear definition of project objectives;
- Clear definition of the project scope;
- Clear definition of the stakeholders roles and responsibilities;
- Existence of a work plan;
- Existence of a resource plan;
- Careful tracking of project progress.



## 5.2. Enschede

The municipality of Enschede is a large municipality in the Netherlands with about 153.000 citizens. More specific, based on population Enschede is the 12<sup>th</sup> municipality of the Netherlands and employs about 1600 public servants [CBS05].

Enschede's pursued strategy aims at offering public services directly targeted at the citizen. The possibilities as provided by ICT are embraced to further fulfill this strategy. Since 1996 the use of ICT has been used publicly to provide electronic service delivery. The Superpilot project, as described in Table 1, catalyzed the development of a digital service counter by providing sufficient resources and the opportunity to present themselves to other Dutch municipalities as a frontrunner.

We interviewed R. Grimmelikhuyse, team manager development [GRI05].

### 5.2.1. Municipal and ICT organization

The organizational structure of the municipality is based on five pillaring services, which in this context refer to parts of a governmental organization. Each service has a number of departments, containing various teams.

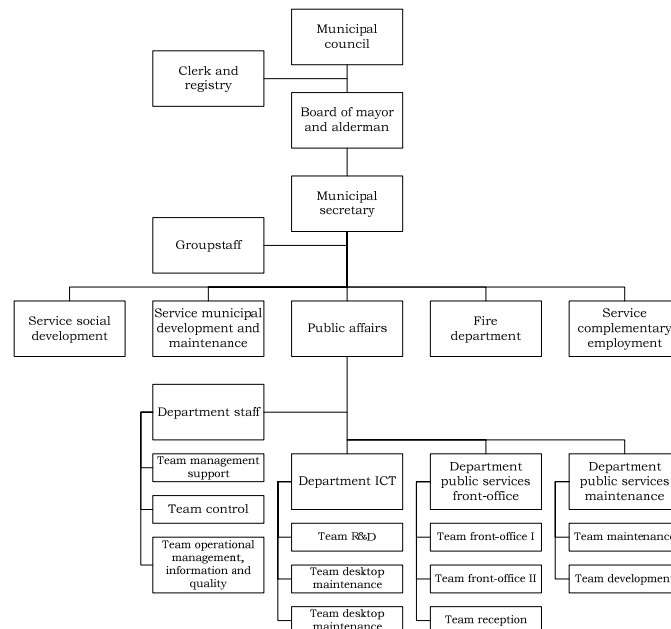


Figure 39: organization chart of the municipality of Enschede

The Department ICT belongs to the service Public affairs and contains three teams. The department Public services front-office is responsible for the front-office of the organization; back-office related activities are spread throughout the remainder of the organization. The department of public services maintenance carries out ICT-related maintenance tasks and minor updates.

Enschede has adopted the use of ICT since the early 90's. During the years, the ICT steering group's formal status has matured. It transformed from a small project team to the Department ICT, an embedded part of the organization which acts as ICT authority within the municipality. This department also acts as (stand-alone) project organization of implementation projects. During these projects, designated project hours are assigned to the employees. This enables them to concentrate on the project without being disturbed by operational activities. For enterprise-wide implementation projects, this is an essential success factor. Another interesting issue is the increasing adoption of SLA's in the organization to formalize quality of service between the consuming and providing parties, which is in line with service-orientation.

A top-level sponsor who actively promotes the adoption and extensive use of ICT in the organization and remains committed throughout the project from a top

position does not exist however. With “top” we refer to actors which carry final responsibility of the involved processes, essentially a member of the municipal council and *not* a manager. Since this is a success factor in convincing the whole organization of the purpose of a project, problems with enterprise-wide adoption of new ICT technologies and procedures can be expected.

This year, Enschede has set up a plan in which they will develop a new solution for the digital service counter in cooperation with some third-party suppliers and four partnering municipalities, also known as “Dimpact” [MOM06]. Enschede’s role in this project shifts gradually from motivator, developer and user to a user role only. The maintenance of the system will be carried out by a foundation, financed by the participating users (municipalities). Municipalities other than the initiating ones can join this foundation and benefit from the developed services, knowledge and experience for approximately 1 euro per citizen a year [GRI05].

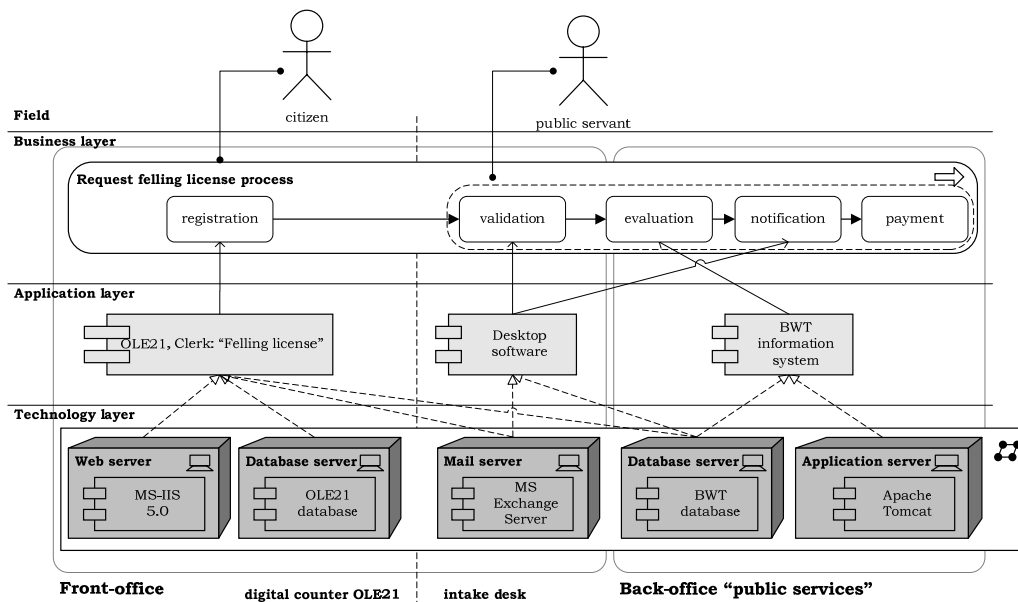
The described plans have been *partially* used to perform our analysis of a candidate solution, particularly for the enterprise system, and project management dimension of the analysis. For example, these plans provide useful information on how Enschede addresses certain activities related to the implementation of an enterprise-wide system.

The aspect we have particularly focused on in Enschede is the already existing digital counter, the interaction between the citizen and the municipality (G2C), and later on between municipalities (G2G).

### **5.2.2. Business process: scenario I**

In Enschede, the process of requesting a felling license can already be performed via the municipal website. Our requesting citizen, Joe Schmoe, is expected to accept an agreement with the municipality which enables *digital signing* of PDF documents. When a citizen refuses to accept such an agreement, he has to request the license traditionally: offline. The use digitally signed documents circumvents the need for a *physical* signature and therefore enables complete electronic service delivery. This indicates a stage 3 of ESD-sophistication: *two-way interaction / vertical integration*. However, the link between front- and back-office can be regarded as tightly-coupled. After filling in a web-form, the request is registered *directly* by means of a point-to-point, and therefore tightly-coupled, connection with the “BWT” database (“Bouw-, Woning, en Toezicht”, Construction, Housing, and Supervision). This database is located in the back-office and owned by the department Public services maintenance. After validation of the request, the BWT system generates a digitally signed PDF document which represents a formal request made by Joe Schmoe. A back-office employee processes this request. Part of this process is an evaluation of the tree and the situation on site, which is performed by a public servant. After evaluation, this servant decides whether to grant or decline the license request. This decision is processed in the BWT back-office systems, after which the requesting citizen is notified of this decision. Eventually the license has to be paid by means of a money transfer.

The involved processes are not formalized in a document which can be accessed by all involved employees. The procedures are publicly known though.



**Figure 40: high level overview of involved processes and systems of scenario I in Enschede**

The involved systems and processes are depicted in Figure 40. An interesting part of this situation is the point-to-point connection between the applications at the front-office and the back-office BWT database. No intermediary layer is used for this connection, which indicates a tightly-coupled situation and does not encourage reuse of functionality. Deployment of a *new* OLE21 clerk requires specific coding to correctly communicate with the BWT-database or databases which are relevant for the clerk in context.

### 5.2.3. Business process: scenario II

Information concerning this public service can be gathered using Enschede's digital counter. An appointment to assess the personal documents for an issue of intended marriage can be arranged by phone or using the digital counter. The formal documents which are required during this appointment (birth certificate, proof of identity and an excerpt of the municipal register), have to be handed over during this appointment. In case a person is currently not registered in Enschede, these documents have to be acquired in the person's municipality prior to the public inspection. After approval of the documents, a new appointment has to be scheduled to formally complete the issue of intended marriage. This can only be done on a Wednesday.

After the public inspection has been carried out, the delivered information is processed in the municipal public records database, which in this case is Centrics PIV4all (Person Information Facility, "PersoonsInformatieVoorziening") application. Now the formal ceremony can be arranged by selecting a date, time and location. After the ceremony is officially completed, the records in PIV4all are updated.

An interesting aspect of this scenario is the requirement that the involved citizens have to arrange the required documents themselves, while these documents are stored in the municipal public records database. Two aspects would be ideal: (1.) the public servant who is responsible for the issue of intended marriage is able to access the stored documents in Enschede, and (2.) in other municipalities as well.

### 5.2.4. Suggested ESB solution

Currently, the digital service counter, also referred to as OLE21, acts as the connecting channel between the front-office (including the Internet) and some back-office systems. Through this channel, a citizen can consult municipal information, request licenses or consume other public services. The services' stages of sophistication vary from stage 1 (providing information), to stage 3 (two-way interaction / vertical integration).

A recent research by TNO ICT determined that the current situation is far from loosely-coupled. The various transaction and information modules which are used to electronically provide services to a citizen, connect with back-end databases *directly*. Little functionality stored in back-end systems is reused by the digital service counter. No intermediate layer between front- and back-office is used and no overall architecture of the current situation is available. Since about 120 transaction and 350 information modules are currently in use, without proper documentation or monitoring, the overall situation has become tightly-coupled and complex to manage. The ESB can act as an intermediate layer between the front- and back-office, enabling a loosely-coupled situation.

In Enschede, multiple applications deployed in a heterogeneous environment have to be opened up to the organization and stakeholders (including citizens), possibly using new channels. Therefore, an integration scenario which conforms mostly to this situation is EAI-scenario 2: *Facilitate wider connectivity to one or more applications*, described in section 4.2.2. A candidate ESB solution pattern for this situation is the *Service gateway*, which can be extended to a *Web Services compliant broker* later on. These solution patterns are described in section 4.3.2 and 4.3.3 respectively. Since Enschede strives to implement a widely accessible solution, we choose the Web Services compliant broker as the ESB solution pattern for this analysis. With this pattern in mind we studied the architecture decision issues 1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 13 and 14. Based on this study we have selected the capabilities which are required to cover the requirements as posed by the municipality of Enschede. These requirements are presented in Table 7.

**Table 7: suggested ESB-capabilities for the municipality of Enschede**

| Category               | Capability   | Reason  |
|------------------------|--|---|
| 1. Communication       | <ul style="list-style-type: none"> <li>▪ All</li> </ul>  | <ul style="list-style-type: none"> <li>▪ Extensive communication is required;</li> </ul>  |
| 2. Service interaction | <ul style="list-style-type: none"> <li>▪ Service interface definition;</li> <li>▪ Service messaging models;</li> <li>▪ Service directory and discovery.</li> </ul>           | <ul style="list-style-type: none"> <li>▪ Services need to be invocable throughout a significant part of the organization;</li> <li>▪ An open standard messaging model such as SOAP is advised to encourage interoperability;</li> <li>▪ Loose coupling is required and has to be encouraged;</li> </ul>   |
| 3. Integration         | <ul style="list-style-type: none"> <li>▪ Service / message aggregation;</li> <li>▪ Legacy and application adapters;</li> <li>▪ Database.</li> </ul>                          | <ul style="list-style-type: none"> <li>▪ Functionality embedded in applications needs to be opened up using services;</li> <li>▪ No common data model is defined yet;</li> <li>▪ Possibly, department data models have to be aggregated to other (common) data models.</li> </ul>   |
| 5. Security            | <ul style="list-style-type: none"> <li>▪ Authentication;</li> <li>▪ Authorization;</li> <li>▪ Confidentiality.</li> </ul>  | <ul style="list-style-type: none"> <li>▪ Opening up functionality and information to mobile users or third parties requires authentication and authorization;</li> <li>▪ Providing specific functionality based on a user, or group profile requires (process-level) authorization;</li> <li>▪ Transaction and exchange of formal and privacy sensitive issues and documents requires confidentiality;</li> <li>▪ Point-to-point security suffices in some situations, for instance within a department;</li> <li>▪ End-to-end security is advised however, but poses serious requirements on operational systems.</li> </ul> |
| 6. Service level       | <ul style="list-style-type: none"> <li>▪ Availability;</li> <li>▪ Performance.</li> </ul>  | <ul style="list-style-type: none"> <li>▪ Required when departments are becoming dependent of each other. To be combined with service contracts and SLA's.</li> </ul>  |
| 7. Message processing  | <ul style="list-style-type: none"> <li>▪ Message and data transformations;</li> <li>▪ Message /service aggregation and correlation;</li> <li>▪ Store and forward.</li> </ul> | <ul style="list-style-type: none"> <li>▪ Messages have to be transformed or merged based on various data models when crossing department/service borders;</li> <li>▪ Functionality embedded in applications needs to be opened up using services;</li> <li>▪ No common data model is defined yet;</li> <li>▪ Possibly, open standards are not supported by default and transformations are required.</li> </ul>   |

|                            |   |  |
|----------------------------|---|--|
| 8. Management and autonomy | <ul style="list-style-type: none"> <li>Administration capability;</li> <li>Service provisioning and registration;</li> <li>Logging;</li> <li>Monitoring.</li> </ul> | <ul style="list-style-type: none"> <li>One point of control over services;</li> <li>Manage service addressing and naming;</li> <li>Service interfaces should be registered, concentrated, and discoverable, for instance using UDDI;</li> <li>Logging is required for debugging, security, and performance analysis purposes;</li> <li>Monitoring is required when SLA's have been agreed upon and to detect potential security breaches in time.</li> </ul> |
| 9. Modeling                | <ul style="list-style-type: none"> <li>Common data models.</li> </ul>   | <ul style="list-style-type: none"> <li>A unified view on the processed data is required to promote semantic interoperability.</li> </ul>   |

Adoption of the suggested ESB solution pattern enables the involved systems to communicate with each other by means of the three services types, as depicted in the layered overview in Figure 41. This form of system integration is not bound to internal systems only. Using the loosely-coupled architecture, connections with third-party service suppliers can be set up without radical changes to the architecture. In our suggested solution we have adopted the Dutch DigiD and Bank Dutch Municipalities, (“Bank Nederlandse Gemeenten”, BNG) business services for authentication and payment functionality respectively. Since we are mainly interested in the business functionality of these services, the lower level services and application of these parties are not adopted in this figure, decreasing the figure’s complexity. This is where the added value of a loosely-coupled SOA can be exploited as well.

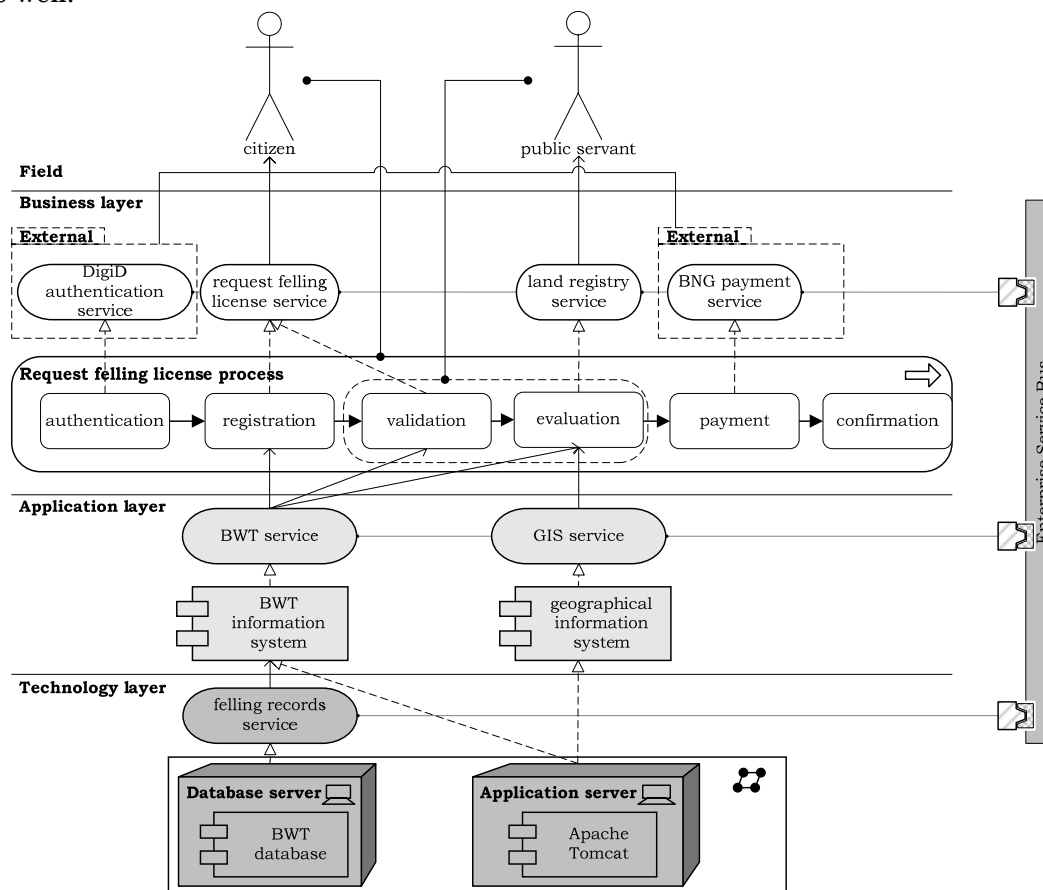


Figure 41: layered model of the envisioned solution in Enschede

The business services contain the business rules and process flows of a felling license request by a citizen and the evaluation of this request by a public servant. The application services open up the functionality of the traditional BWT information system and the Geographical Information System (GIS) to all connected services. By doing so, methods for geographical purposes and the registration, fetching and evaluation of felling license requests are available for other services as well. This

promotes reuse of functionality, as can be seen in the invocation of the BWT application service by two business services (the request felling license service and the land registry service).

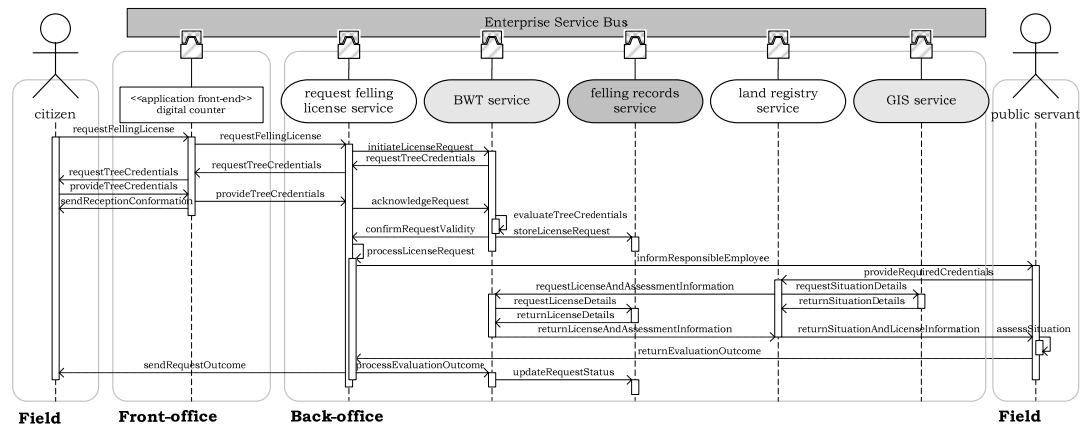


Figure 42: interaction diagram connecting the various services by means of an ESB

Figure 42 shows the interaction between the various entities and services. Note that the used services match the ones as used in Figure 41. We have used this notation to stress the idea of *cooperating* services. Since the services act at different levels of abstraction: the business, application, and technology layer respectively, they actually do not interact *directly*. Higher level services are realized by the lower level ones, as explained in Figure 19. Therefore Figure 42 is not accurate from an architectural viewpoint where entities of differing levels of abstraction are not allowed to communicate with each other. This note also accounts for Figure 43, introduced next. We solely choose this notation to outline the cooperating character of services.

The process of authentication and payment has not been modeled in this diagram to maintain a certain degree of simplicity. A number of available options are presented to the citizen through the digital service counter. After selecting the tree felling license, the procedure for this license is initiated. The business service knows which underlying application services need to be invoked and in what sequence. This has to be defined either manually or using a service choreographer. In case of a valid request, a public servant receives an order to evaluate the request on site. After evaluation, the updated status of the request is stored in the BWT database and the requesting citizen is informed. How the citizen is informed, does not matter that much (either by letter, or, when email is accepted as *formal* communication channel, by email). An email application service can be connected to the ESB to realize this. Since all services which are connected with the ESB can communicate with each other, this introduced email service can be invoked by other services as well.

We abstract from the channel through which a public servant is informed. In the current situation, a public servant receives a message with the request details. After acquiring some geographical information, the employee heads to the location of the license request. But suppose all public servants who perform field tasks on a regular basis are equipped with a PDA, or smartphone. The business process channel can be altered in such way that a public servant receives new instructions through the new mobile channel and is not bound to his desktop anymore. A new application service for such purpose can be connected to the ESB, and after a modification in the business service this new channel can be put in operation.

Since the second scenario is more interesting from an interaction point of view, we solely discuss this aspect of our solution. The model of involved systems and their position in the layered model resembles the one as depicted in Figure 41.

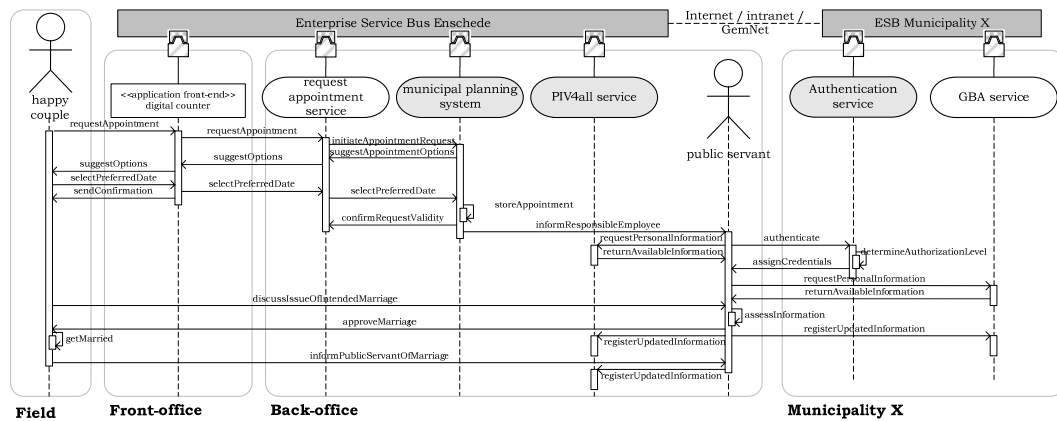


Figure 43: interaction diagram using external services describing scenario II

Figure 43 models the interaction where a public servant examines the issue for intended marriage and has *direct access* to the municipal personal records database of Enschede. Whenever municipality X adopts a similar approach an environment can be created where the public servant has access to the personal records of municipality X as well. This can be extended to the situation where the two ESB's are coupled, this conforms to the distributed service bus pattern as discussed in section 3.2.5. This enables the operational systems deployed in Enschede to communicate with remote systems, deployed at other municipalities. Technically the sketched situation is possible. To realize this however, municipal agreements concerning organizational (contracts, service dependencies, security issues) and semantic interoperability (used standards) should be made. On the short-term, the mentioned nation-wide agreements are not expected.

### 5.2.5. Roundup of the analysis

Now we have described the three dimensions of the BEST framework (project management, permanent business and enterprise system), we can use this framework to identify potential problem areas concerning the implementation of the suggested ESB solution. This is done by describing the six elements for each dimension in Table 8 and discussing the derived situation sketch afterwards.

Table 8: scenario-based analysis of the envisioned EAI-project at the municipality of Enschede

|                    | Municipal situation /<br>Scenario 1 & 2<br><<permanent business>>  | Service gateway<br><<enterprise system>>  | ICT department and<br>partners<br><<project management>>  |
|--------------------|--|---|---|
| Strategy and goals | <ul style="list-style-type: none"> <li>Offer public services directly targeted at the citizen utilizing ICT;</li> <li>ICT business goals and strategy have matured and adjusted to comply with the growing need for interoperability.</li> </ul> | <ul style="list-style-type: none"> <li>Increased interoperability at various organizational levels;</li> <li>Optimized processes;</li> <li>Re-use of functionality by means of modules and loose-coupling.</li> </ul>   | <ul style="list-style-type: none"> <li>Facilitate the organization with the ICT-means required to achieve the defined business goals;</li> <li>As part of integral management, clear goals are defined and resources are assigned.</li> </ul>                             |
| Management         | <ul style="list-style-type: none"> <li>Enschede has adopted integral management for all departments;</li> <li>A top-level sponsor who actively promotes the project and takes full responsibility does not exist.</li> </ul>                     | <ul style="list-style-type: none"> <li>A dedicated foundation ("Dimpact") is set up to carry out development, implementation and further exploitation;</li> <li>New municipalities can join the foundation;</li> <li>The ownership remains at the foundation;</li> <li>Enschede's role gradually shifts to an end-user role.</li> </ul> | <ul style="list-style-type: none"> <li>Sufficient authority and mandate has been acquired to carry out the task properly;</li> <li>The department manager is enthusiastic concerning current developments and has sufficient means to lead the ICT-department.</li> </ul> |

|                      |   |  |   |
|----------------------|---|--|---|
| Structure            | <ul style="list-style-type: none"> <li>• The system is designed to fit the existing organization structure;</li> <li>• No formal process description, including process owners and related responsibilities is available throughout the organization;</li> <li>• Public service / front-office is responsible for intake activities;</li> <li>• No business intelligence or monitoring tools are currently used.</li> </ul>   | <ul style="list-style-type: none"> <li>• The foundation is responsible for the system;</li> <li>• Technology should be tuned to the business structure;</li> <li>• The fit of the envisioned system with the organization has not been analyzed in detail;</li> <li>• In the first stage of the project, a number of selected services will be developed and deployed on the ESB;</li> <li>• After evaluation, additional services will be developed.</li> </ul> | <ul style="list-style-type: none"> <li>• Enschede acts as both end-user, developer and owner;</li> <li>• The team responsible for the implementation consists of both members of the ICT department and specialized third-parties;</li> <li>• Senior management and end-users are not formally represented in this team but can be consulted when necessary.</li> </ul> |
| Process              | <ul style="list-style-type: none"> <li>• The possible need to adapt business processes to optimally exploit system functionality is recognized by involved departments;</li> <li>• Direct communication with mobile employees requires process changes;</li> <li>• Citizens have to request various documents which are available within the municipality themselves. A public servant should have access to the digital versions of these documents when required to.</li> </ul>   | <ul style="list-style-type: none"> <li>• No formal mechanisms for change control (data transformation, integration with existing systems, configuration) have been identified yet;</li> <li>• No formal testing approach is in operation yet;</li> <li>• No formal ICT-management approach is in operation yet;</li> <li>• Intermunicipal communication with other municipalities requires authentication and authorization processes.</li> </ul>                | <ul style="list-style-type: none"> <li>• No formal implementation approach has been defined (yet);</li> <li>• Not all process owners are <i>actively</i> involved during the project;</li> <li>• Initially, five municipalities are involved in the project;</li> <li>• Sharing of knowledge between the foundation's members is stimulated.</li> </ul>                 |
| Knowledge and skills | <ul style="list-style-type: none"> <li>• The team related with the selected business processes already has some experience with automation and integration of business processes;</li> <li>• Some employees fail to see the overall goal and tend to focus too much on "their" sub-process;</li> <li>• The "protection" of processes by employees does not affect the progress of an automation project and the eventual process efficiency positively;</li> <li>• Some experience with service contracts is available within the organization, this is stimulated by the use of integral management.</li> </ul>  | <ul style="list-style-type: none"> <li>• The technology supplier has experience with service bus technology and adaptors and messaging mechanisms;</li> <li>• Experience with operational versions of this product is available.</li> </ul>  | <ul style="list-style-type: none"> <li>• The involved parties have some basic experience with such projects; this specific situation is new for all parties though;</li> <li>• Sufficient resources have been reserved for this project.</li> </ul>   |
| Social dynamics      | <ul style="list-style-type: none"> <li>• Management attitude towards change varies, some persons are open for innovation while others are more conservative;</li> <li>• Some line-managers are conservative and reluctant to change and increased use of ICT;</li> <li>• Past experiences with automation projects are both positive and negative;</li> <li>• Some employees are afraid to lose their authority concerning a specific process and resist to cooperate by making "their" process as important as possible;</li> <li>• Other employees are able to appreciate the process efficiency and correctness of data and recognize that ICT can facilitate more or different work instead of less.</li> </ul> | <ul style="list-style-type: none"> <li>• The involved technology supplier has sufficient experience in dealing with municipalities and related social dynamics;</li> <li>• Since the project generates important knowledge for the technology supplier, they are expected to act rather flexibly.</li> </ul>   | <ul style="list-style-type: none"> <li>• All involved parties have sufficient experience in dealing with municipalities and related social dynamics.</li> </ul>   |

In the context of Enschede, an ESB can provide added value in *loosely* connecting multiple systems and opening up functionality stored in back-end systems to



citizens and other parts of the organization. Throughout the years, the ICT-organization has gained sufficient mandate to guide the introduction of this new technique. A top-level sponsor (i.e. a member of the municipal council) who actively promotes this project and takes responsibility does not exist however. This can cause problems in enterprise-wide adoption of the suggested solution, particularly at departments which are reluctant to change and (extended) use of ICT [GRI05].

Pitfalls can be found mainly in the lack of architectural insight, process descriptions, process owners and related responsibilities, unclear implications of process changes, and the culture of current employees. Some departments are positive towards further optimization of processes and actually *serving the citizen effectively and efficiently* supported by ICT, instead of trying to maintain or increase authority by increasing the importance of “their” process [GRI05]. Additionally, the risk of conflicting roles exists since Enschede has multiple interests in this project. It acts as end-user, developer and owner. Each role is related with certain responsibilities and interests which might become entangled.

Since no enterprise-wide projects with such impact have been carried out yet and experience is moderate, it is advisable to set up project planning to achieve quick results. This way, involved employees can experiment with the new services to convince them of the added value. Based on these results and experience, motivated teams can head on to the next milestone and gradually provide the organization with a growing amount of functionality. Changes to processes can be performed gradually using a migration plan based on this approach [GRI05].

### 5.3. Voorst

The municipality of Voorst is with over 23.000 citizens an average municipality in the Netherlands, more specific, based on population Voorst is the 204<sup>th</sup> municipality of the Netherlands, contains twelve small towns, and employs about 220 public servants [CBS05].

The pursued strategy in the context of this research aims at providing the community with a proper collection of services for a justified price and to stimulate pro-active and structured communication with the local community. An important role for ICT has been assigned to achieve these goals [VOO05].

We interviewed H. Peters, information advisor of the municipality of Voorst and contributor to EGEM and J. Uiterweerd, coordinator Public Affairs [PET06][UIT06].

#### 5.3.1. Municipal and ICT organization

As of January 2006, Voorst has adopted an organization structure based on five departments which are organized like a “flower”. In the center of this model, the municipal council resides. This council consists of the board of mayor and alderman, and the municipal secretary.

During the end of the previous millennium an internal research performed by the municipality itself, described the municipality as a highly internally oriented, defensive, risk- and cooperation avoiding organization with an excessive amount of rules, bureaucracy and a lack of consistency in their information systems. Based on these results, a plan was initiated to transform Voorst into an efficiently cooperating municipality to provide their citizens with the required services at well-justified costs. The use of information improvement and management processes had a central role in this project. Change of structure, culture and processes were recognized as critical success factors. The mentioned changes included decreased management, better management, a decreased amount of rules and hierarchy, an increase of personal responsibility and interoperability.

A cross-organizational and financial independent project team involving members from various parts of the organization was assembled to lead this project. This structure, combined with well-organized communication towards the remainder of the organization caused broad understanding and acceptance for the change process. The project results were embedded in the organization on a structural basis, which further increased the visibility of the project results for the whole organization. Eventually the improvements in structure resulted in a new interactive model of the organization structure which was proposed by the municipality itself. Rather than a static collection of departments, the model as depicted in Figure 44 was adopted.

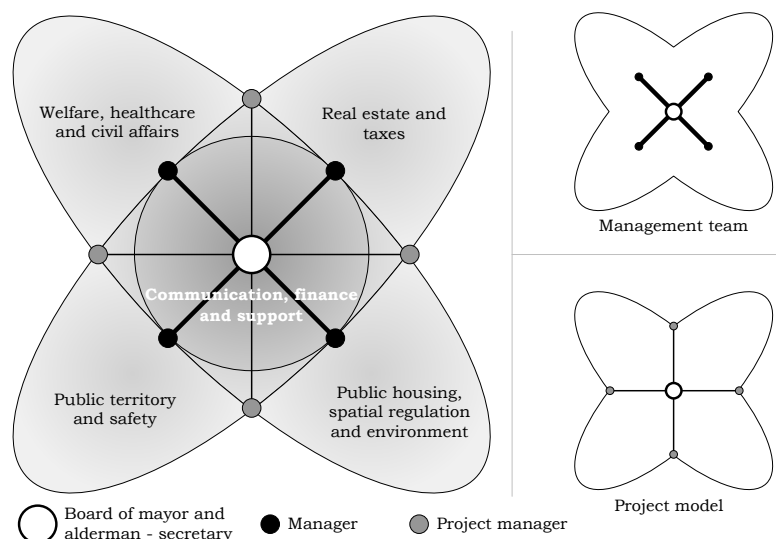
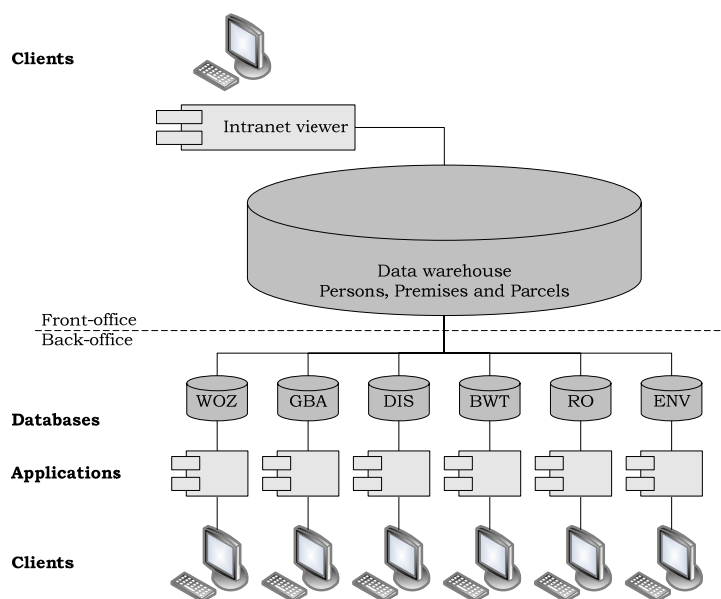


Figure 44: model of the interactive organization adopted in Voorst, also referred to as the flower model

The model shows an overlap between the four departments which focus on the municipal environment and the supporting department of “Communication, finance and support”, which is located in the middle. The management team consists of all department managers, the board of mayor and alderman, and the municipal secretary. Projects are carried out using a designated project model. Following this model, employees from various departments can be assigned as project manager. These project managers operate on the intersection of two departments and report *directly* to the municipal secretary. This has some interesting benefits. Employees from various departments can be adopted in the project team and communication between the project managers and the secretary do not have to go *through* a department manager, circumventing department managers and their interests, which might conflict with the projects targets. This way, multi-disciplinary project teams which span the entire organization can be formed, lines of communication are kept short, and undesired involvement of department managers is avoided.

Voorst has an interesting ICT strategy as well. Since it is a “small” municipality, the use of ICT and further automation and integration between the front- and back-office does not necessarily imply an interesting business case. Therefore, the municipality mainly focused on quality of information, digitizing the existing archives and recognizing relationships between Persons, Premises, and Parcels (PPP, “Personen, Panden en Percelen”). Nowadays, a data warehouse containing these PPP relations is operational. This Oracle database contains the content of various underlying databases. The data in this data warehouse is updated periodically, depending on the update frequency of the original data. For instance, mutations concerning BWT data are forwarded to the data warehouse daily, while data concerning topographical information is only synchronized with the data warehouse once every six years. This results in a disparate situation where data is stored twice, once in the task specific database (and possibly parts of the operational applications), and once again in the covering database. This situation is depicted in Figure 45.



**Figure 45: high-level overview of the data warehouse's central position in Voorst [PET06]**

For Voorst, this environment works well though. The data warehouse is opened up to employees using an intranet viewer. Based on the derived information, employees can assess a certain case and perform the required tasks in one of the 200 back-end applications. These applications, used protocols and data models, are maintained by two application administrators and still perform properly.

Concerning ESD, Voorst has an interesting vision as well. Currently, some forms are available for download on the municipal website, indicating a stage 2 of ESD-sophistication. Increasing this stage does not have priority though:

*“On a maximum, we receive four change of address notifications per day. These notifications, received either by personal contact at the counter, telephone, or paper form, are forwarded to the responsible employee who processes it in the system. Full electronic handling by means of web-forms and processing it directly in our back-end administration is (financially) not very interesting for us. We tend to focus on original ideas.”*[PET06].

The mentioned ideas mainly concentrate on opening up municipal information to citizens, which is currently stored in the data warehouse. An example is a detailed map of Voorst, which can be imported in the popular Google Earth application [GOO05]. Using this application in combination with the provided map and a developed Web Service, which partially opens up the central data warehouse, location specific information concerning construction license requests can be provided.

As stated, the use of ICT in the back-office remained fairly unchanged during the years. The attitude of employees towards ICT has improved during the past five years. The intranet viewer is used extensively, and employees were, and are involved in various ICT projects. Still *“...the majority of the employees do not have the foggiest idea of the future ICT-based changes in service delivery, as formalized by the Dutch national government...”* [PET06].

The aspect we have focused on particularly in Voorst is the relative small size of the municipality and the interesting organizational structure with its project model.

### **5.3.2. Scenario I**

Our requester, Mr. Schmoe decides to use the request form which is published on the municipal website. After he has printed and filled the form, he sends it to the municipality. There it is manually processed in the proper application; no scan is made to store the form digitally. A number of employees are responsible for processing a felling license requests. The way how incoming requests are processed is not formalized, the requirements of a felling license are applied which results in a positive or negative outcome. The request and related outcome is stored in the proper database, after which this (change in) information can be forwarded to the central data warehouse and a notification is sent to the requestor.

This scenario is a good example of the use of task-specific applications, databases and possibly ambiguous storage of data (not all data is stored twice). The registered requests are currently published in the municipal magazine. When these requests are coupled with relevant geo-information and opened up to the Internet, they can be published on the municipal website, or emailed to citizens living in a specific area as well.

### **5.3.3. Scenario II**

Currently, our bride and groom can look up some information concerning getting married, or download an electronic brochure on the municipal website. For an appointment with a public servant to issue the intended marriage for public inspection, they have to contact the municipality by phone. The procedure of making an appointment online was subject of discussion within the municipality. Due to lack of time and resources this project has been canceled.

The formal documents which are required during this appointment (birth certificate, proof of identity and an excerpt of the municipal register), have to be handed over during this appointment. In case a person is currently not registered in Voorst, these documents have to be acquired in the person's municipality prior to the public inspection. The credentials of the couple are registered in the municipal public records database, which in this case is Centrics PIV4all application. The entered data is stored in an Oracle database. The municipal messenger (“bode”) receives a message when the official ceremony takes place, after which he arranges the necessary arrangements. After the ceremony has been fulfilled, the data stored in the PIV4all is updated [UIT06].

This scenario conforms to the situation as depicted in Figure 45. A specific application with related database is used. The functionality, or specific parts of it, which is located in the application is not available for other applications.

### 5.3.4. Suggested ESB solution

The past few years, Voorst focused on the development and maintenance of high-quality information. This has been achieved by digitizing existing archives and combining the data as stored in task-specific databases in an overall data warehouse. An intranet viewer which connects with the resulting database opens up the required information and their relations to the entire organization. Task-specific (legacy) applications are used to enter new information, or update information in the task-specific databases. No connection between these applications and the data-warehouse has been realized yet. Voorst stresses that this does not have priority since all applications operate properly and potential integration projects are expected to have great difficulties in becoming profitable. To outline the added value provided by an ESB and the concepts of SOA, we abstract from this argument and envision a candidate solution which enables the use of data as stored in the data-warehouse by task-specific applications.

The task-specific databases remain in operation but can be gradually dismantled during future integration projects. Eventually a situation can be achieved, where task-specific functionality, which currently remains in the mentioned applications, is opened up to the organization and later on to chain partners or citizens. This can be achieved by means of application services which encapsulate the applications while technology services act as the intermediate layer between the data warehouse and these application services.

The envisioned solution resembles the third integration scenario: *Facilitate wider connectivity to legacy systems*. A candidate ESB solution pattern for this scenario is the adoption of *Basic adapters*, or a *Service gateway*. These solution patterns are described in section 4.3.1 and 4.3.2 respectively. For Voorst, it seems useful to start experimenting with the basic adapters pattern, during which some adapters are implemented. This has minimal implications on the existing infrastructure and sufficient experience can be gained using these adapters. Gradually the municipality can decide to (partly) open up functionality located in the back-office systems to specific chain partners or citizens. Reliable authentication of chain-partners makes it possible to exchange formal documents with other municipalities. Since an authorization service such as the national DigiD initiative is already in use by Voorst and security-issues can possibly be covered by hardware means (by means of a firewall including a DMZ and possibly crypto-boxes), advanced security features are not required.

Scenario II benefits from this situation when either bride or groom is currently not registered in Voorst and documents such as birth certificates can be acquired instantaneously by the involved public servant.

The sketched situation (opening up low volume functionality to the organization and a subset of this functionality to chain partners) and the use of the Basic adapters solution pattern was kept in mind when we applied the ESB capability selection approach as described in chapter 4. Therefore we studied the architecture decision issues 1, 2, 3, 4, 7, 8, 9, 10, 11, 13 and 14. Based on this study we have selected the capabilities which are required to cover the requirements as posed by the *fictive* situation in Voorst. These requirements are presented in Table 9.

**Table 9: suggested ESB-capabilities for the municipality of Voorst**

| Category               | Capability   | Reason  |
|------------------------|--|---|
| 1. Communication       | <ul style="list-style-type: none"> <li>Protocols and standards;</li> <li>A messaging mechanism.</li> </ul>         | <ul style="list-style-type: none"> <li>Basic communication is required;</li> <li>Routing and addressing can be handled by existing gateways which contain routing and DNS functionality;</li> </ul>           |
| 2. Service interaction | <ul style="list-style-type: none"> <li>Service interface definition;</li> <li>Service messaging models.</li> </ul> | <ul style="list-style-type: none"> <li>Services need to be invocable throughout the organization;</li> <li>An open standard messaging model such as SOAP is advised to encourage interoperability.</li> </ul> |

|                            |   |  |
|----------------------------|---|--|
| 3. Integration             | <ul style="list-style-type: none"> <li>Service / message aggregation;</li> <li>Legacy and application adapters;</li> </ul>  | <ul style="list-style-type: none"> <li>Functionality embedded in applications needs to be opened up using services;</li> <li>No common data model is defined yet: aggregation is required;</li> <li>Communication with legacy systems is required.</li> </ul>  |
| 7. Message processing      | <ul style="list-style-type: none"> <li>Message and data transformations;</li> <li>Message / service aggregation and correlation;</li> <li>Store and forward.</li> </ul> | <ul style="list-style-type: none"> <li>Functionality embedded in applications needs to be opened up using services;</li> <li>No common data model is defined yet;</li> <li>Possibly, open standards (XML / SOAP) are not supported by default;</li> <li>Periodic data synchronization between various data sources is required.</li> </ul>   |
| 8. Management and autonomy | <ul style="list-style-type: none"> <li>Administration capability;</li> <li>Service provisioning and registration;</li> <li>Logging;</li> <li>Monitoring.</li> </ul>     | <ul style="list-style-type: none"> <li>One point of control over services;</li> <li>Manage service addressing and naming;</li> <li>Service interfaces should be registered, concentrated, and discoverable, for instance using UDDI;</li> <li>Logging is required for debugging, security, and performance analysis purposes;</li> <li>Monitoring is required when SLA's have been agreed upon and to detect potential security breaches in time.</li> </ul> |
| 9. Modeling                | <ul style="list-style-type: none"> <li>Common data models.</li> </ul>   | <ul style="list-style-type: none"> <li>A unified view on the processed data is required to promote semantic interoperability.</li> </ul>   |

Exploiting the features of the selected ESB capabilities, we have the situation in mind for Voorst as depicted in Figure 46. This figure sketches an architecture where application functionality is opened up to citizens and the remainder of the organization using an external PPP & viewer application service. Also note that the task-specific applications are able to access the PPP data warehouse using an technology service. This way, required data does not have to be looked up using the intranet viewer by an employee after which the acquired data is typed over in the application in context. Eventually, all data can be derived from the central data warehouse. Whether this is desirable is subject to further research. A situation where all essential data is stored on one location can impose privacy and security threats to the municipality.

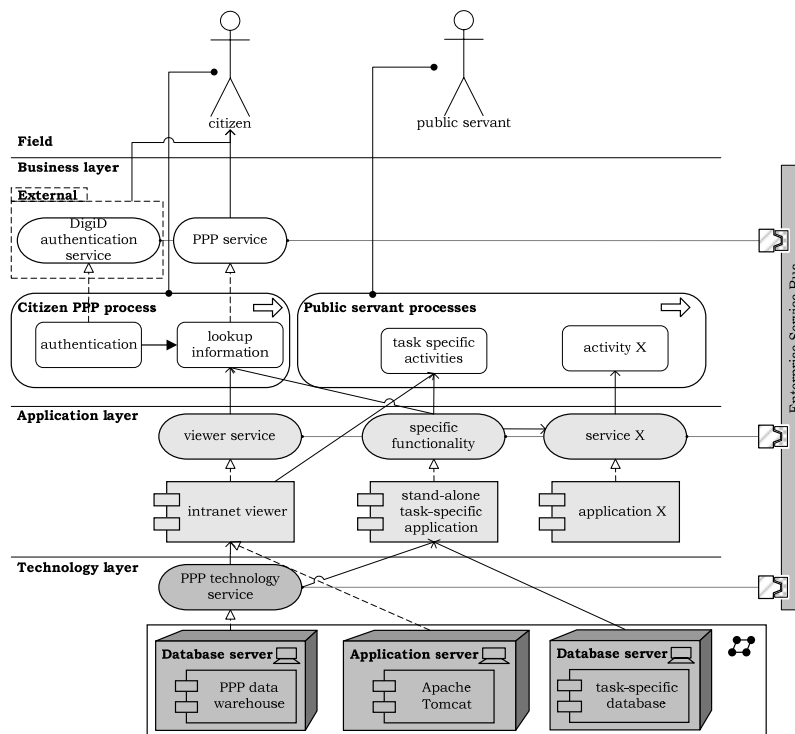


Figure 46: layered model of the envisioned solution in Voorst

The envisioned solution communicates via the local network in Voorst. Since this LAN is quite small, no extensive security measurements for local communication are required.

### 5.3.5. Roundup of the analysis

After we have described the three dimensions of the BEST framework for the situation in Voorst, the six elements for each dimension are analyzed in Table 10. After this analysis wrap-up, the impact of our suggested solution can be distilled.

**Table 10: scenario-based analysis of the envisioned EAI-project at the municipality of Voorst**

|                    | <b>Municipal situation /<br/>Scenario 1 &amp; 2<br/>&lt;&lt;permanent business&gt;&gt;</b>   | <b>Service gateway<br/>&lt;&lt;enterprise system&gt;&gt;</b>  | <b>Project team:<br/>ICT department and partners<br/>&lt;&lt;project management&gt;&gt;</b>   |
|--------------------|--|---|---|
| Strategy and goals | <ul style="list-style-type: none"> <li>• Provide citizens with the required services at well-justified costs;</li> <li>• Achieve and maintain high quality of information;</li> <li>• Offer added value to citizens by focusing on original ideas.</li> <li>• Ignore financially uninteresting business cases;</li> <li>• Increase efficiency and interoperability.</li> </ul>   | <ul style="list-style-type: none"> <li>• Open up functionality embedded in task-specific applications to the organization via the intranet;</li> <li>• Open up functionality embedded in task-specific applications to citizens via the Internet;</li> <li>• Open up data warehouse to task-specific applications;</li> <li>• Reduce disparate storage of data.</li> </ul>  | <ul style="list-style-type: none"> <li>• Facilitate the organization with the ICT-means required to achieve the business goals;</li> <li>• Focus on original business cases and keep the limited resource in mind.</li> </ul>   |
| Management         | <ul style="list-style-type: none"> <li>• The municipal secretary is expected to spread ICT-awareness throughout the organization.</li> </ul>   | <ul style="list-style-type: none"> <li>• Required maintenance should be aligned with the tasks of the current application administrators;</li> <li>• Authorization and authentication management is required when opening up functionality and data to the organization and citizens.</li> </ul>  | <ul style="list-style-type: none"> <li>• A project team involving employees from all departments should be formed.</li> </ul>   |
| Structure          | <ul style="list-style-type: none"> <li>• No, or little up-to-date documentation concerning the current situation is available;</li> <li>• No immediate need to open up embedded functionality is recognized. Mainly due to the low volume of incoming service requests.</li> </ul>   | <ul style="list-style-type: none"> <li>• The fit of the envisioned system with the organization has not been analyzed in detail yet;</li> <li>• In the first stage of the project, a number of selected services will be developed and deployed on the ESB;</li> <li>• After evaluation, additional services will be developed;</li> <li>• The technology should be tuned to the business structure;</li> <li>• Gradually, existing task-specific databases can be dismantled.</li> </ul> | <ul style="list-style-type: none"> <li>• The project team consists of users spread throughout the organization, enabling short communication lines;</li> <li>• The municipal secretary leads the project team, enabling top management support and encouraging user involvement;</li> <li>• Tight cooperation with technology suppliers is required.</li> </ul> |
| Process            | <ul style="list-style-type: none"> <li>• No formal process-description including process owners and related responsibilities is available;</li> <li>• The envisioned system enables the applications involved in scenario 1 to check provided information in the data warehouse and display location-specific geo-information;</li> <li>• Formal documents from other municipalities might be required in scenario 2, exchange should be realized electronically.</li> </ul> | <ul style="list-style-type: none"> <li>• No formal mechanisms for change control (data transformation, integration with existing systems, configuration) have been identified yet;</li> <li>• No formal testing approach is in operation yet;</li> <li>• No formal ICT-management approach is in operation yet.</li> </ul>  | <ul style="list-style-type: none"> <li>• Little experience with enterprise-wide implementation projects is available;</li> <li>• No experience with formal implementation methods is available yet.</li> </ul>  |

|                      |  |  |   |
|----------------------|--|--|---|
| Knowledge and skills | <ul style="list-style-type: none"> <li>• The employees responsible for the selected business processes have little experience with automation and integration of business processes.</li> </ul>  | <ul style="list-style-type: none"> <li>• No experience with ESB techniques is available;</li> <li>• No resources are available to train or employ employees specifically for this purpose;</li> <li>• Technology suppliers are required to provide long-term maintenance and support.</li> </ul> | <ul style="list-style-type: none"> <li>• No experience with formal implementation methods is available yet.</li> </ul>              |
| Social dynamics      | <ul style="list-style-type: none"> <li>• Employees were tightly involved during previous projects;</li> <li>• Common acceptance among employees is growing;</li> <li>• Ignorance of future developments and ICT-schedules are not known commonly.</li> </ul> | <ul style="list-style-type: none"> <li>• Little implications for end-users are involved;</li> <li>• Involved users are open for new techniques.</li> </ul>   | <ul style="list-style-type: none"> <li>• Employees are eager to learn;</li> <li>• Employees are open for new techniques.</li> </ul> |

Since Voorst is a small municipality, the added value of an ESB is limited to interconnecting a number of task-specific applications and opening up stored information to citizens via the Internet. Voorst prefers to innovate with creative solutions instead of automating traditional procedures which do not occur often anyway. Long-term enterprise wide integration and automation projects are not likely to be initiated in this municipality for the near future. Therefore, the lack of knowledge in managing such projects does not necessarily imply problems for Voorst. Additionally, the small ICT-organization does not require highly formalized and structured ICT-maintenance methodologies to manage operational systems, databases and processes. However, a higher degree of process formalization and documentation is advised to increase transparency and prepare the organization for long-term automation projects. Such projects are not expected for the near future though; they are part of the never ending quest to increase efficiency and transparency in governmental organizations.

Abstracting from the limited use of an ESB, the sketched solution *can* provide a solid foundation to open up an increasing number of services via the Internet. User authentication and authorization functionality can be used to open up the specific information to inform a selected audience, for instance citizens of Voorst or other municipalities.

The organizational structure is quite interesting. The project model might be useful for other Dutch municipalities because communication lines are kept short and responsibilities are assigned to the right employees.



## 5.4. Almelo

The municipality of Almelo is an above-average municipality in the Netherlands with about 73.000 citizens. More specific, based on population Almelo is the 45<sup>th</sup> municipality of the Netherlands, contains a number of neighboring villages and employs about 720 public servants [CBS05].

The pursued strategy in the context of this research aims at providing high-quality urban facilities, stimulating pro-active and open communication with citizens and providing services focused on problem solving and self-service [ALM02].

We interviewed E. Legtenberg, information advisor and manager [LEG06].

### 5.4.1. Municipal and ICT organization

The organizational structure of the municipality of Enschede is based on five pillaring services, each service has a number of departments, containing various teams. Three staff services support the municipality in their daily tasks. Almelo has recognized the need for proper information facilities about ten years ago. To ensure this, the staff service Computerization (“Informatisering”) and Automation (I&A) was established. Reorganization in 2002 separated this service in two individual staff services. After an evaluation of this reorganization the decision was made to merge these services in the future again. Therefore we refer to these services as the service I&A, as is depicted in Figure 47.

Application and database administrators were previously employed in the department where assigned applications or databases were operational. Additionally, some services had an information coordinator. Nowadays these administrators and coordinators are grouped in the service I&A, concentrating ICT-skills and knowledge and shortening the communication lines between them. The requirement for administrators to keep up with the state-of-affairs of the department where they were previously employed is acknowledged and respected.

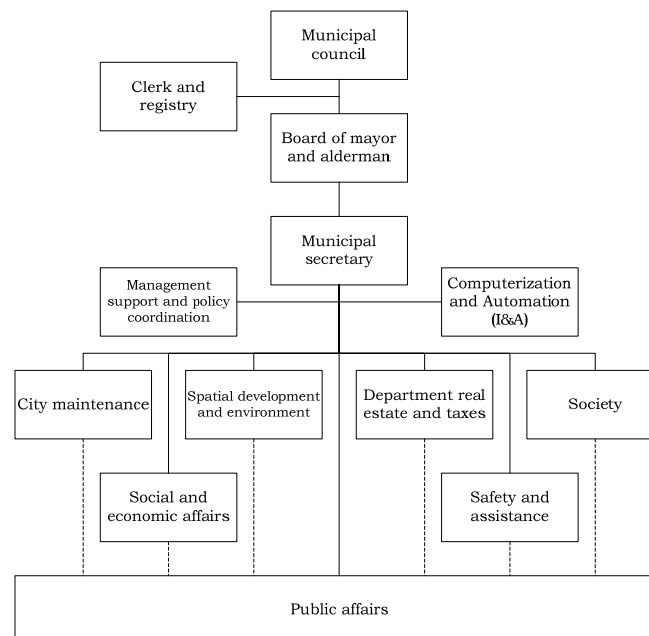


Figure 47: organization chart of the municipality of Almelo

Figure 47 depicts the service Public affairs as a broad, detached service. This service takes care of all received requests, documents, phone calls and emails. Therefore, Public affairs, can be regarded as the municipal front-office. After a request is received which can not be processed directly by Public affairs, it is forwarded to the responsible service, or department. Currently, requests are forwarded physically, but in the near-future, every form is scheduled to be scanned and forwarded digitally.

As of the year 2003, an *information policy plan* is in use. This plan is aligned with the municipal agenda and contains long-term strategic goals. Main goals of this plan are to:

- Increase (business process) transparency;
- Improve provision of services;
- Improve operational management.

This plan is commonly based on the principles of increased office automation, increased quality of information, increased use of geographical information, and increased cooperation and coordination. A so-called I&A credit fund was set up to cover the financial aspects of reaching the targets as defined in the information policy plan. Every year this fund receives a predetermined amount of money which can be used for maintenance and replacement of systems and devices. New investments have to be approved by the municipal council. Based on the information policy plan and an inventory of service-specific requests, an annual *project plan* is formulated. This plan is the guideline for computerization and automation projects of the municipality of Almelo for that year.

During the years the staff service I&A has gained respect and authority. Project proposals which lack information concerning ICT management, maintenance and impact for current systems or processes are likely to be declined by the municipal council. The service I&A strives to reach as much as possible by informal means such as close cooperation with departments, and involvement of these departments and end-users in new projects. This has resulted in increasingly business-driven projects opposed to the technology-driven projects which were common previously. Some departments are more experienced and eager to cooperate in such projects than others however.

The cooperation with the various departments has resulted in a rather good insight of operational applications, databases, connections and related responsibilities as well. Another result of increased cooperation and experience within the I&A service is the adoption of SLA's throughout the municipality. These agreements are mainly in use between the supporting staff services and the line services and are not in effect between the various line services yet. A number of business process descriptions are made as well; this has been done using Protos, a product of Pallas Athena. These descriptions represent the formal status of a process, are publicly available and can be used for optimization purposes or used when automation projects have to be prepared. For ICT management the ITIL methodology has been adopted in combination with Marval tooling. Currently, little management information is generated and actively used. This is expected to change with the adoption of the web based Cognos 8 business intelligence management tool.

Similar to Voorst, Almelo has adopted the use of a data warehouse which combines data from various task-specific databases as well. Since we have already discussed the data warehouse in section 0, we have particularly focused on the formalized and concentrated intake of forms and request via Public affairs. Interesting issue is a pilot project that aims at scanning all incoming post and storing them in a document information system.

#### **5.4.2. Scenario I**

Our requester, Mr. Schmoe uses the request form which is published on the municipal website. After he has printed and filled the form, he delivers it at the Public Affairs counter. This counter accepts the request and forwards it to "Spatial development and environment". Once received by this service, the responsible public servant processes it in Centrics BWT4all application. This application stores the request in an underlying Oracle-based BWT database.

#### **5.4.3. Scenario II**

To prepare themselves, our happy couple looked up some information concerning getting married on the municipal website. For an appointment with a public servant

to issue the intended marriage for public inspection, they have to make an appointment which should take place on a Monday; this can not be handled via the website. The required documents have to be collected prior to this appointment. The public servant processes the details concerning the wedding in the Cipers GBA application provided by PinkRoccade. The central data warehouse receives relevant changes in the records of Joe Schmoe and Jane Doe as well.

#### 5.2.4. Suggested ESB solution

Almelo already has a concentrated point of intake which is formally embedded in the organizational structure. The service Public Affairs takes care of all intake activities. They process and forward the requests to the proper back-office departments or teams. Since an increasing amount of requests is received via Internet and email, digital forwarding of these requests is embraced. As soon as physical forms are digitized as well, paperless communication between the various departments can be considered. To realize a connection between the front- and back-office (i.e. Public Affairs and the other services and departments), a number of applications have to be coupled. Therefore the second EAI-scenario: *Facilitate wider connectivity to one or more applications*. Section 4.2.2 describes this scenario in detail.

Since the emergency services and some public servants in the field are already experimenting with using PDA's to access municipal data through the Internet, the *Web Services compliant broker* as described in section 4.3.3 seems the ideal ESB solution pattern to start with. From ICT-perspective, we qualify the use of PDA's or other mobile devices in the field as a *high-quality urban facility*, as defined in Almelo's business goals. The fire-department uses it to register dangerous areas, or in case of emergency, to lookup information concerning the environment and relevant data sources.

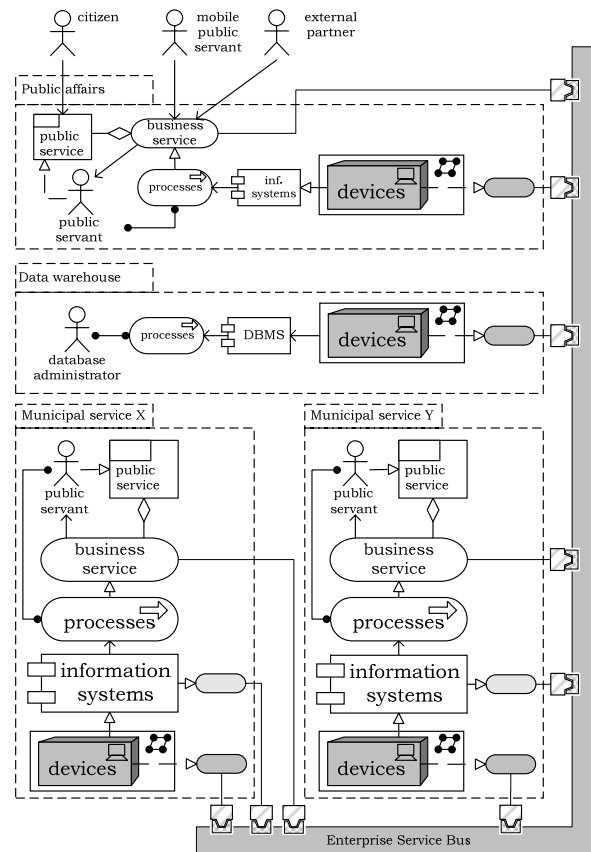
Keeping the selected pattern, experience with the data ware house, the centralized intake and the aim to increase the sophistication of ESD in mind, we have studied architecture decision issues 1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 13, and 14.

**Table 11: suggested ESB-capabilities for the municipality of Almelo**

| Category               | Capability  | Reason  |
|------------------------|---|---|
| 1. Communication       | <ul style="list-style-type: none"> <li>Protocols and standards;</li> <li>A messaging mechanism.</li> </ul>  | <ul style="list-style-type: none"> <li>Basic communication is required;</li> <li>Routing and addressing can be handled by existing gateways which contain routing and DNS functionality;</li> <li>Various communication channels will be used.</li> </ul>   |
| 2. Service interaction | <ul style="list-style-type: none"> <li>Service interface definition;</li> <li>Service messaging models.</li> </ul>  | <ul style="list-style-type: none"> <li>Services need to be invocable throughout the organization;</li> <li>An open standard messaging model such as SOAP is advised to encourage interoperability.</li> </ul>   |
| 3. Integration         | <ul style="list-style-type: none"> <li>Service / message aggregation;</li> <li>Legacy and application adapters;</li> </ul>  | <ul style="list-style-type: none"> <li>Various communication channels will be used;</li> <li>Applications need to communicate with each other;</li> <li>No common data model is defined yet: aggregation is required;</li> <li>Communication with specific systems is required.</li> </ul>  |
| 5. Security            | <ul style="list-style-type: none"> <li>Authentication;</li> <li>Authorization;</li> </ul>   | <ul style="list-style-type: none"> <li>Opening up functionality and information to mobile users or third parties requires authentication and authorization;</li> </ul>  |
| 6. Service level       | <ul style="list-style-type: none"> <li>Availability;</li> <li>Performance.</li> </ul>   | <ul style="list-style-type: none"> <li>Formalized, concentrated intake involves functionality to measure availability;</li> <li>Formalized, concentrated intake involves functionality to measure performance;</li> </ul>   |
| 7. Message processing  | <ul style="list-style-type: none"> <li>Message and data transformations;</li> <li>Message / service aggregation and correlation;</li> <li>Store and forward.</li> </ul> | <ul style="list-style-type: none"> <li>Functionality embedded in applications needs to be opened up using services;</li> <li>No common data model is defined yet;</li> <li>Possibly, open standards (XML / SOAP) are not supported by default;</li> <li>Periodic data synchronization between various data sources is required.</li> <li>Mobile (instable) environments should be supported;</li> </ul> |

|                            |   |  |
|----------------------------|---|--|
| 8. Management and autonomy | <ul style="list-style-type: none"> <li>Administration capability;</li> <li>Service provisioning and registration;</li> <li>Logging;</li> <li>Monitoring.</li> </ul> | <ul style="list-style-type: none"> <li>One point of control over services;</li> <li>Manage service addressing and naming;</li> <li>Service interfaces should be registered, concentrated, and discoverable, for instance using UDDI;</li> <li>Logging is required for debugging, security, and performance analysis purposes;</li> <li>Monitoring is required when SLA's have been agreed upon and to detect potential security breaches in time.</li> </ul> |
| 9. Modeling                | <ul style="list-style-type: none"> <li>Common data models.</li> </ul>   | <ul style="list-style-type: none"> <li>A unified view on the processed data is required to promote semantic interoperability.</li> </ul>   |

Adoption of the suggested ESB solution pattern enables the municipal back-end systems to communicate with the centralized intake desk as is depicted in Figure 48. This form of system integration is not bound to internal systems only. Public servants in the field are able to communicate with functionality and information as stored in back-end systems or the data warehouse to carry out their tasks. Concentrated, digital or digitized intake of formal requests poses requirements on the SLA's between involved parties. Security and privilege issues become increasingly important when municipal systems are opened up to remote, or external users.



**Figure 48: the service Public affairs acts as front-office serving various channels**

The depicted situation positions the ESB as the connecting entity between the service Public affairs, underlying municipal services and departments, and the data warehouse. Through business services as maintained by the municipal front-office Public affairs, public servants in the field and external partners can lookup information or use functionality which is stored in back-end municipal systems. The data warehouse opens up its vast data collection using a technology service. This service is used to query the Oracle database. Manual maintenance processes are carried out by a database administrator using a DataBase Management System (DBMS).

### 5.2.5. Roundup of the analysis

Now we have described the three dimensions of the BEST-framework for the situation in Almelo, the six elements of these dimensions are analyzed in Table 12. After this analysis wrap-up, the impact of our suggested solution can be distilled.

**Table 12: scenario-based analysis of the envisioned EAI-project at the municipality of Almelo**

|                      | <b>Municipal situation /<br/>Scenario 1 &amp; 2<br/>&lt;&lt;permanent business&gt;&gt;</b>   | <b>Web Services compliant<br/>broker &lt;&lt;enterprise system&gt;&gt;</b>  | <b>ICT department “I&amp;A”<br/>&lt;&lt;project management&gt;&gt;</b>  |
|----------------------|--|---|---|
| Strategy and goals   | <ul style="list-style-type: none"> <li>• Provide high-quality urban facilities;</li> <li>• Offer public services directly targeted at the citizen to facilitate problem solving and self-service;</li> <li>• Increase (business process) transparency;</li> <li>• Improve provision of services;</li> <li>• Improve operational management.</li> </ul>   | <ul style="list-style-type: none"> <li>• Open up back-end information and functionality to mobile users and external partners;</li> <li>• Use Public affairs as centralized intake point for digital and digitized requests;</li> <li>• Extend the use of service contracts by means of setting up between the involved parties SLA's.</li> </ul>   | <ul style="list-style-type: none"> <li>• Facilitate the organization with the ICT-means required to achieve the defined business goals;</li> <li>• A clear goal and budget has been defined in the information policy plan and the annual project plan.</li> </ul>  |
| Management           | <ul style="list-style-type: none"> <li>• A top-level sponsor who actively promotes the project and takes full responsibility does not exist;</li> <li>• Importance of consistency and future maintenance of new ICT projects is acknowledged;</li> <li>• Monitor progress of the information policy plan;</li> <li>• Monitor progress of the annual project plan;</li> <li>• Assess I&amp;A credit fund requests.</li> </ul> | <ul style="list-style-type: none"> <li>• Public affairs acts as front-office and is responsible for all incoming service requests;</li> <li>• Security management should be addressed when the organization is opened up using the Internet;</li> <li>• ITIL is adopted as formal ICT-management methodology;</li> </ul>  | <ul style="list-style-type: none"> <li>• Realize goals of the information policy plan (long-term);</li> <li>• Realize goals of the annual project plan (short-term);</li> <li>• Maintain good relationship with other departments.</li> </ul>   |
| Structure            | <ul style="list-style-type: none"> <li>• Public affairs acts as front-office, they are formally responsible for the intake process;</li> <li>• Process descriptions are available and can be analyzed and adapted using Protos;</li> <li>• Currently Public affairs uses a national set of process descriptions;</li> <li>• Cognos 8 is used for Business Intelligence purposes creating insight and awareness;</li> </ul>   | <ul style="list-style-type: none"> <li>• Technology should be tuned to the business structure;</li> <li>• Public affairs will be responsible for the communication with external users (from a business point of view);</li> <li>• I&amp;A is responsible for all ICT-related aspects;</li> <li>• A number of SLA's already exist, especially between staff and line services.</li> </ul> | <ul style="list-style-type: none"> <li>• After a short flirt with separated departments, the computerization and automation departments will be merged into one I&amp;A department again;</li> <li>• Relations with other departments are good;</li> <li>• ITIL is used for ICT-management</li> </ul>                 |
| Process              | <ul style="list-style-type: none"> <li>• Public affairs gains a lot of new responsibility, processes have to be reviewed, updated and made publicly available;</li> </ul>  | <ul style="list-style-type: none"> <li>• ITIL documentation should be updated to align with the new situation;</li> <li>• SLA's between Public affairs, I&amp;A, and other departments have to be revised and formalized;</li> </ul>  | <ul style="list-style-type: none"> <li>• ITIL can be used for change control as well (ITIL set 2);</li> </ul>   |
| Knowledge and skills | <ul style="list-style-type: none"> <li>• The employees responsible for the selected business processes have little experience with automation and integration of business processes;</li> <li>• Employees at Public affairs already have some experience with handling digital requests and SLA's.</li> </ul>  | <ul style="list-style-type: none"> <li>• The I&amp;A department has matured and gained experience throughout the years;</li> <li>• Common office automation tasks are handled well.</li> </ul>  | <ul style="list-style-type: none"> <li>• Step towards service-orientation is considered an evolutionary and logical one;</li> <li>• The I&amp;A department has matured and gained experience throughout the years;</li> <li>• Little experience with enterprise-wide implementation projects is available.</li> </ul> |

|                 |  |  |  |
|-----------------|--|--|--|
| Social dynamics | <ul style="list-style-type: none"> <li>• Management attitude towards change varies;</li> <li>• Public servants show interest and pose questions opposed to resistance, resulting in various business-driven projects;</li> <li>• Various employees are enthusiastic, some show less commitment;</li> <li>• The need for 'control of the process' is still common, automated update or insert queries on a database are considered 'hostile';</li> <li>• Changes in a business process can result in resistance.</li> </ul> | <ul style="list-style-type: none"> <li>• Public affairs is respected for their centralized intake activities;</li> <li>• Regular contact between Public affairs and connected departments encourages a good relationship;</li> <li>• I&amp;A has gained respect and authority throughout the years;</li> <li>• ICT-related employees from various departments have been placed in I&amp;A, resulting in a heterogeneous and open environment.</li> </ul> | <ul style="list-style-type: none"> <li>• I&amp;A has gained respect and authority during the years;</li> <li>• Informal cooperation is embraced;</li> <li>• Contacts with 'former' departments are kept alive by I&amp;A employees.</li> </ul> |
|-----------------|--|--|--|

Table 12 shows a municipality prepared for a small step towards service-orientation. Various departments, Public affairs and I&A included, are familiar with SLA's, ITIL and business intelligence. Additionally, projects are increasingly business-driven which stimulates end-user involvement. Processes are documented properly and easy accessible for involved employees. This provides public servants with an overview of the "big picture" of their task or sub-process. Relationships are maintained both by formal and informal measures and I&A has gained a good status within the municipality. The long-term information policy plan and the annual project plan ensure alignment between operational activities and strategic goals.

Direct and responsible involvement of top-management can be improved by appointing a strong person as ICT-representative of the municipality. This encourages involvement throughout the organization and creates acceptance for unpopular decisions which possibly has to be made by I&A. Serious attention has to be paid to (formal) project management and monitoring activities. The use of ITIL should be extended to ICT-maintenance tasks.

The ESB can provide an infrastructure foundation for further exploring the principles of service-orientation. Stored data and functionality can be reused throughout the organization and in the field to provide a higher level of service quality to the citizens of Almelo.

## 5.5. Groningen

The municipality of Groningen is a large municipality in the north-east of the Netherlands, with nearly 180.000 citizens. More specific, based on population Groningen is municipality the 7<sup>th</sup> municipality of the Netherlands and employs about 3300 public servants [CBS05].

The pursued strategy in the context of this research aims at the improvement of communication between the municipal council and citizens, and the reduction of operational costs by means of optimized use of ICT functionality [GRO02].

We interviewed G. Hylkema, strategic information manager, C. Top, information manager department Environment and H. Wobbes, information and automation advisor of the department of spatial development and economic affairs [HYL06][WOB06].

### 5.5.1. Municipal and ICT organization

The organizational structure of the municipality of Groningen is based on eight pillaring services. Each service has a number of departments, containing various teams. Every service has assigned a head of ICT who is responsible for all ICT related issues in the particular service.

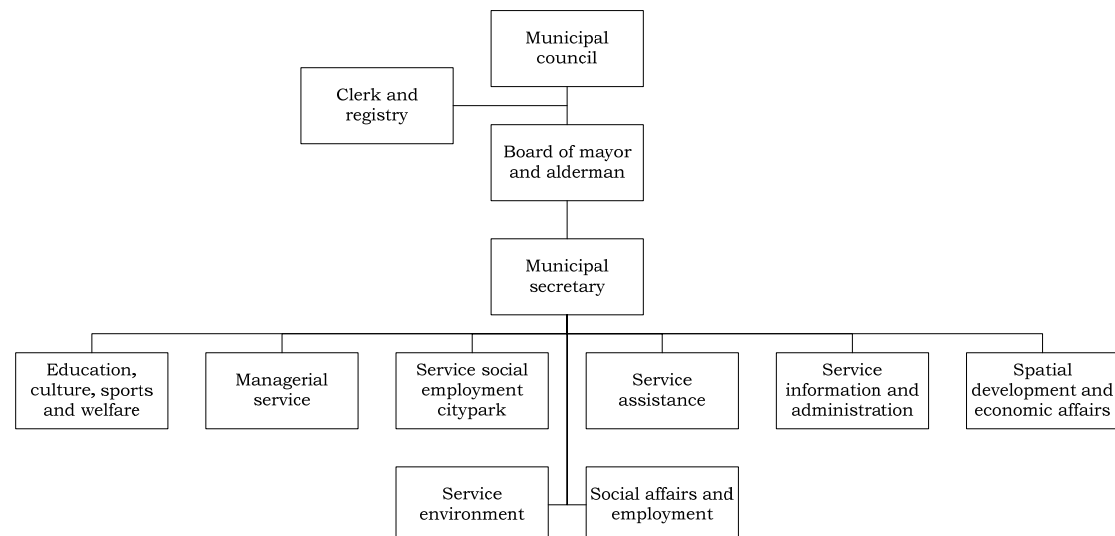


Figure 49: organization chart of the municipality of Groningen

The eight services are ordered decentralized and can be characterized as highly autonomous. Each service has their own agenda, culture, set of methodologies, collection of operational information systems and commonly used suppliers.

The Managerial service is responsible for policy-development, including the municipal ICT-policy. This service advises the board of mayor and alderman and the municipal council on this subject. The management team consists of the eight service directors and the municipal secretary. To coordinate municipal ICT-activities, a so-called iPlatform was installed. This platform seats the heads of ICT of all services and is chaired by the managerial service. The platform-members meet twice a month during which operational policy is discussed and coordinated. The representative of the managerial service stated that [HYL06]:

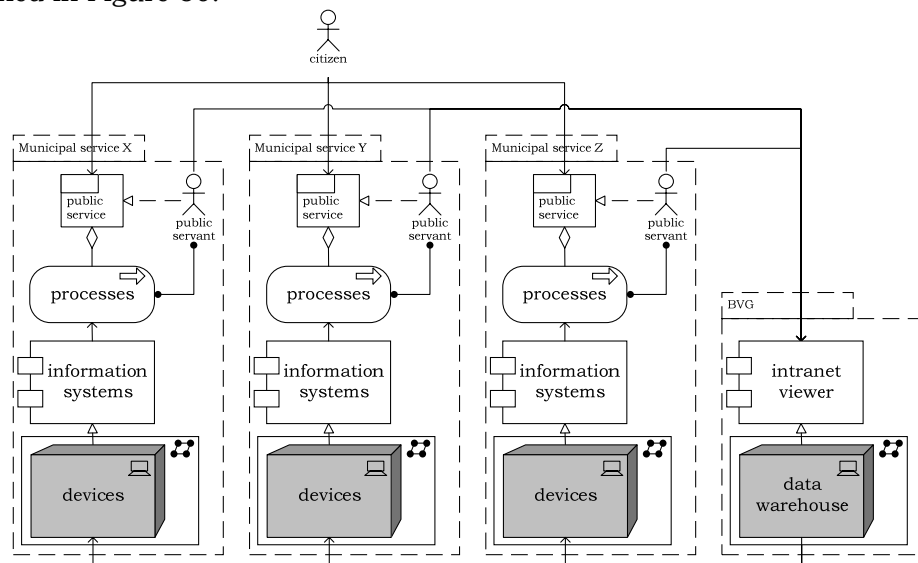
*“The iPlatform should have a policy developing task with increased authority, rather than the coordinating task and advising character it has nowadays.”*

The iPlatform initiated some working groups, including the architecture working group. This group is gradually gaining awareness of the purpose of, and need for a clear and enterprise-wide architecture to meet the increasing requirements as posed to municipalities by citizens and the national government.

Since the line services have a significant level of autonomy, they are free to determine the priority of such an architecture in their organization and agenda. No service or department has the mandate to enforce certain standards or methodologies throughout the organization. This slows down the process of integration between the services and encourages the current disparate situation to exist. To illustrate this situation, 800 applications and related databases are in use by the various back-offices, including three different packages which carry the title “mid-office” and multiple ESB packages. Knowledge (process, application- or technique-specific) is scattered throughout the organization, and in some cases lost when employees quit their job.

In 2004 the Basis Voorziening Gegevens (BVG) project was completed. This project aimed at concentrating data which is often used in one single data ware house, similar to the situation in Voorst and Almelo. Essential information concerning persons, premises and parcels are now stored in an intermediate database, which is synchronized with underlying application weekly. An increasing number processes need this information to be synchronized real-time however; for example when a citizen reports a change of address, other data such as information concerning parking licenses should be updated immediately. An internet viewer opens up the information to authorized public servants. Functionality as stored in the various applications is not opened up by this solution though. Therefore, interoperability in the context of business processes has not been realized yet, these processes are only provided with a higher quality of data.

For Groningen, we have focused on the described *distributed* situation, which is sketched in Figure 50.



**Figure 50: sketch of the disconnected situation in Groningen**

The managerial service experiences difficulties in convincing the municipal council and the board of mayor and alderman of (1.) the increasing need for a clear architecture and (2.) the benefits this architecture can provide. The council and board expect investments to *directly* support the primary process. The envisioned architecture, on its turn, is expected to provide insight in the current situation and possibilities for business process optimization, re-use of functionality, information and knowledge. This does not directly support the primary process, therefore no need for such architecture is commonly acknowledged at a high organizational level.

Currently some experience is gained in applying ESB techniques in a project which automates the process of requesting a parking license to some extent. The project leader regards this project as a pilot to gain knowledge concerning the required techniques and highlight organizational limitations [WOB06]. Due to a shortage of time, this project has not been analyzed in detail.



### 5.5.2. Scenario I

Our requester, Mr. Schmoe decides to use the request form which is published on the digital counter of the municipal website. After he has printed and filled the form, he sends it directly to the service *Spatial development and economic affairs*. There it is manually processed in Centrics BWT4all application; no scan is made to store the form digitally. A number of employees are responsible for processing a felling license requests. The way how incoming requests are processed is not formalized in a publicly available document; the distinguished steps are stored in Centrics workflow module WFM4all application however. The requirements of a felling license are applied which results in granting or declining a license request. The request and related outcome is stored in the Oracle BWT database.

Notices of objection are processed by another department; a new file is generated based on the information as stored in the BWT database. These notices are processed by (another instance of) the BWT4all application as well.

### 5.5.3. Scenario II

Information concerning getting married can be accessed via the digital counter on the municipal website. An appointment can be made using this counter as well. The required documents have to be gathered by our happy couple themselves.

During the assessment of the intended marriage, all required information is stored in the Cipers GBA application, supplied by PinkRoccade. This application leads the public servant through the whole process. After the marriage has taken place officially, all involved information is updated. The information stored in the Oracle database is synchronized periodically with the municipal BVG system to ensure that important information can be reused by other municipal departments.

### 5.5.4. Suggested ESB solution

Groningen is a large municipality which organizational structure is based on eight highly independent services. Together these services employ about 3300 public servants and serve the 180.000 citizens of Groningen, plus neighboring regions. Besides the municipal council, no service or department has sufficient authority to make enterprise-wide decisions. Because the managerial service experiences difficulties in convincing the municipal council and the board of mayor and alderman of the emerging need for an enterprise-wide approach, this situation is expected to keep existing on the short-, and mid-term. To share information, knowledge, functionality and resources throughout the municipality, the gap between the various services has to be crossed. Since these services can be regarded as small, *independent* organizations, this situation conforms to the fifth EAI-scenario: *implement controlled integration of services or systems between company divisions or partnering organizations*. Section 4.2.5 describes this scenario in detail.

A solution has to be found where all services maintain their current applications, but the embedded functionality or stored data can be opened up to authorized users, or applications throughout the organization. Security and semantic interoperability issues have to be addressed thoroughly prior to setting the technical requirements. When departments are becoming dependent of (business) services provided by other parts of the municipality, organizational agreements have to be made as well, for instance by means of SLA's.

The *Service gateway* pattern, as described in section 4.3.2, can be adopted to set up a preliminary environment in which data and functionality is securely shared between various departments. This pattern also enables the division to open up specific functionality to citizens through the Internet. This requires increased adoption of Web Services techniques. Basic Web Services technology is supported by the Service gateway, but when sophisticated Web Services functionality is required, the adopted pattern can be extended to a *Web Services compliant broker*, as described in 4.3.3. We expect that the Service gateway provides the municipality with sufficient integration functionality to set up an initial integration project. Therefore this pattern was used to perform our scenario-based analysis. To select

the ESB capabilities which are suggested in the municipality of Groningen, we have studied architecture decision issues 1, 2, 3, 4, 6, 8, 9, 10, 11, 13, and 14 while keeping the disconnected situation of Groningen in mind. Based on this study, we propose the ESB-capabilities as summarized in Table 13.

**Table 13: suggested ESB-capabilities for the municipality of Groningen**

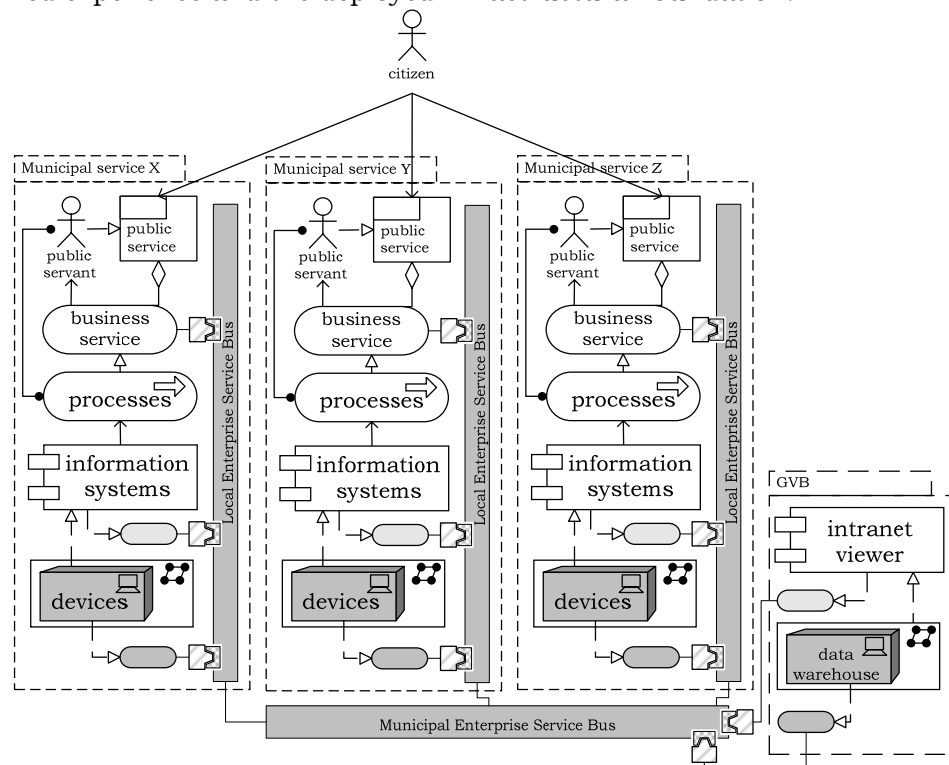
| Category                   | Capability   | Reason  |
|----------------------------|--|---|
| 1. Communication           | <ul style="list-style-type: none"> <li>▪ Routing;</li> <li>▪ Addressing;</li> <li>▪ Protocols and standards;</li> <li>▪ Select appropriate messaging mechanism;</li> </ul>                     | <ul style="list-style-type: none"> <li>▪ Basic communication is required;</li> <li>▪ Asynchronous messaging might be required when various services spread throughout the organization are invoked and no reply can be provided, or expected, instantaneously.</li> </ul>   |
| 2. Service interaction     | <ul style="list-style-type: none"> <li>▪ Service interface definition;</li> <li>▪ Service messaging models</li> </ul>  | <ul style="list-style-type: none"> <li>▪ Services need to be invocable throughout the organization;</li> <li>▪ An open standard messaging model such as SOAP is advised to encourage interoperability.</li> </ul>   |
| 3. Integration             | <ul style="list-style-type: none"> <li>▪ Service / message aggregation;</li> <li>▪ Service mapping;</li> <li>▪ Legacy and application adapters;</li> <li>▪ Protocol transformation.</li> </ul> | <ul style="list-style-type: none"> <li>▪ Department data models have to be (partly) combined based on a common data model;</li> <li>▪ Functionality embedded in applications needs to be opened up using services;</li> <li>▪ Within a department, task-specific applications using specific protocols may have to be opened up;</li> </ul>   |
| 4. Quality of service      | <ul style="list-style-type: none"> <li>▪ Transactions.</li> </ul>  | <ul style="list-style-type: none"> <li>▪ Support of atomic transactions is required when mutations are to be performed by partnering departments.</li> </ul>  |
| 5. Security                | <ul style="list-style-type: none"> <li>▪ Authentication;</li> <li>▪ Authorization;</li> <li>▪ Confidentiality.</li> </ul>  | <ul style="list-style-type: none"> <li>▪ Opening up functionality and information to mobile users or third parties requires authentication and authorization;</li> <li>▪ Providing specific functionality based on a user, or group profile requires (process-level) authorization;</li> <li>▪ Transaction and exchange of formal and privacy sensitive issues and documents requires confidentiality;</li> <li>▪ Point-to-point security suffices in some situations, for instance within a department;</li> <li>▪ End-to-end security is advised however, but poses serious requirements on operational systems.</li> </ul> |
| 6. Service level           | <ul style="list-style-type: none"> <li>▪ Availability;</li> <li>▪ Performance.</li> </ul>  | <ul style="list-style-type: none"> <li>▪ Required when departments are becoming dependent of each other. To be combined with service contracts and SLA's.</li> </ul>  |
| 7. Message processing      | <ul style="list-style-type: none"> <li>▪ Message and data transformations;</li> <li>▪ Message /service aggregation and correlation.</li> </ul>   | <ul style="list-style-type: none"> <li>▪ Messages have to be transformed or merged based on various data models when crossing department/service borders;</li> <li>▪ Functionality embedded in applications needs to be opened up using services;</li> <li>▪ No common data model is defined yet;</li> <li>▪ Possibly, open standards are not supported by default and transformations are required.</li> </ul>   |
| 8. Management and autonomy | <ul style="list-style-type: none"> <li>▪ Administration capability;</li> <li>▪ Service provisioning and registration;</li> <li>▪ Logging;</li> <li>▪ Monitoring.</li> </ul>                    | <ul style="list-style-type: none"> <li>▪ One point of control over services;</li> <li>▪ Manage service addressing and naming;</li> <li>▪ Service interfaces should be registered, concentrated, and discoverable, for instance using UDDI;</li> <li>▪ Logging is required for debugging, security, and performance analysis purposes;</li> <li>▪ Monitoring is required when SLA's have been agreed upon and to detect potential security breaches in time.</li> </ul>  |
| 9. Modeling                | <ul style="list-style-type: none"> <li>▪ Common data models;</li> <li>▪ Public versus private models for business-to-business integration.</li> </ul>  | <ul style="list-style-type: none"> <li>▪ A unified view on the processed data is required to promote semantic interoperability;</li> <li>▪ The 'private' models are for internal use, while the public ones are used for integration purposes with partnering departments.</li> </ul>   |

As can be seen in the set of suggested capabilities, we stress the importance of data models. Such models have to be clearly defined and made publicly available. For the

autonomous municipal services to cooperate, features such as service and message aggregation and correlation are essential. Adoption of municipal-wide SLA's can enforce departments to deliver services which can be consumed by partnering departments or opened up to citizens.

The situation within the municipal services can remain unchanged. We stress that this disparate situation is not desired since it discourages transparency, effectiveness and efficiency. The chances for change of this situation in the near future are small though, therefore these constraints were kept in mind during our analysis. The use of internal services, as described in section 3.2.2, makes it possible to use various data models within an organization. Departments which have already defined or adopted specific data models can keep on using these models internally. When the department services are opened up to other parts of the organization using higher level services or external services, correlation and aggregation of data models enables the involved services to communicate syntactically and semantically correct. This can be realized at the entry-point of the ESB. An overview of the ESB as linking pin between the municipal services is depicted in Figure 51.

The envisioned situation where departments share information, data and functionality enables the municipality to open up selected services to citizens through the Internet as well. The digital service-counter can be connected to the ESB and invoke already deployed services. This service-counter is not explicitly depicted in the figure below, but can be any municipal service. Functionality which resides in the intranet viewer can be reused by citizens as well. Specific authentication and authorization rules in combination with hardware-based security should make this technically possible. But as stated before, the organizational and semantic requirements should be addressed in detail *prior* to opening up functionality to citizens through the Internet. Provided services, interfaces, related SLA's and adopted data models need to be defined and used by all participating parties. This way, service-enabled interoperability between the municipal services can be embedded in the organization from the beginning. Future projects can rely on the gained experience and the deployed infrastructural foundation.



**Figure 51: local ESB's and a municipal ESB as linking pin between the autonomous municipal services**

### 5.5.5. Analysis roundup

After we have described the three dimensions of the BEST framework (project management, permanent business and enterprise system), we can use this framework to identify potential problem areas concerning the implementation of the suggested solution. This is done by describing the six elements for each dimension in Table 14 and discussing the derived situation sketch afterwards.

**Table 14: scenario-based analysis of the envisioned EAI-project at the municipality of Groningen**

|                    | <b>Municipal situation /<br/>Scenario 1 &amp; 2<br/>&lt;&lt;permanent business&gt;&gt;</b>  | <b>Service gateway<br/>&lt;&lt;enterprise system&gt;&gt;</b>  | <b>iPlatform and<br/>service-specific ICT-teams<br/>&lt;&lt;project management&gt;&gt;</b>  |
|--------------------|---|---|---|
| Strategy and goals | <ul style="list-style-type: none"> <li>• Provide citizens with the required services at well-justified costs;</li> <li>• In common, new investments should <i>directly</i> support the primary process;</li> <li>• No overall need for integration or interoperability is formally acknowledged.</li> </ul>   | <ul style="list-style-type: none"> <li>• Achieve technical interoperability between the highly autonomous municipal services;</li> <li>• Open up the collection of spread functionality to other, demanding municipal services;</li> <li>• Realize a municipality wide foundation for future interoperability and service deployment.</li> </ul>                      | <ul style="list-style-type: none"> <li>• Develop and maintain ICT policy throughout the municipality (currently it acts as a coordinator of eight service-specific ICT policies);</li> <li>• Facilitate the organization/service with the ICT-means required to achieve the business goals.</li> </ul>  |
| Management         | <ul style="list-style-type: none"> <li>• The municipal council has assigned the responsibility for innovation to the iPlatform;</li> <li>• The iPlatform requires more authority and should be formally embedded in the organization;</li> <li>• Political conviction of the municipal board does not have a positive effect on new ICT ideas.</li> </ul>   | <ul style="list-style-type: none"> <li>• Every municipal service becomes responsible for their own ESB;</li> <li>• The iPlatform should form a team to maintain the municipal ESB;</li> <li>• Regular meetings between all maintenance team, coordinated by the iPlatform should enforce organizational and semantic interoperability.</li> </ul>                     | <ul style="list-style-type: none"> <li>• The iPlatform consists of the eight service ICT-directors;</li> <li>• The ICT-directors lead the service-specific ICT-teams;</li> <li>• No organizational entity with sufficient authority has been assigned to enforce enterprise-wide decisions. The iPlatform is a candidate but requires more autonomy and authority to be able to initiate and fulfill municipality-wide integration projects.</li> </ul> |
| Structure          | <ul style="list-style-type: none"> <li>• Some departments are making a basic inventory of operational systems and relations, no municipality-wide approach is adopted;</li> <li>• The municipal services have a high level of autonomy and their own (ICT) agenda;</li> <li>• No, or little up-to-date documentation concerning the current situation is available;</li> <li>• No formal process-description including process owners and related responsibilities is available;</li> <li>• The process-steps are embedded in a workflow-module.</li> </ul> | <ul style="list-style-type: none"> <li>• Each municipal service keeps their autonomy and responsibilities;</li> <li>• The iPlatform should coordinate the municipal services with sufficient authority to enforce (semantic) standards when services access the <i>municipal</i> ESB;</li> <li>• The technology should be tuned to the business structure.</li> </ul> | <ul style="list-style-type: none"> <li>• The iPlatform covers the eight services;</li> <li>• The services have their own ICT-organization, this can be maintained.</li> <li>• The iPlatform has limited authority among the municipal services;</li> <li>• Each ICT-service remains responsible for internal activities;</li> </ul>   |
| Process            | <ul style="list-style-type: none"> <li>• Awareness of the need for a municipality-wide architecture should be created;</li> <li>• The iPlatform should have a centralized policy developing task with increased authority, rather than a coordinating task;</li> <li>• Gained knowledge should be spread throughout the municipality.</li> </ul>  | <ul style="list-style-type: none"> <li>• No formal mechanisms for change control (data transformation, integration with existing systems, configuration) have been identified yet;</li> <li>• No formal testing approach is in operation yet;</li> <li>• No formal ICT-management approach is in operation yet.</li> </ul>  | <ul style="list-style-type: none"> <li>• No experience with municipal-wide projects is available;</li> <li>• No experience with formal implementation methods is available yet;</li> <li>• The ICT-teams should be monitored and coordinated by the iPlatform;</li> <li>• Frequent consultation between the ICT-teams to share experiences and knowledge is advised.</li> </ul>   |

|                      |  |   |   |
|----------------------|--|---|---|
| Knowledge and skills | <ul style="list-style-type: none"> <li>• Awareness of the growing need for interoperability is growing, the role of enterprise-wide architecture is ignored however;</li> <li>• Sufficient experience with the current system is available for operational activities;</li> <li>• The knowledge how to further exploit deployed systems is missing, or scattered throughout the organization, for example the use of ESB, and messaging brokers is available, but not reused.</li> </ul>                         | <ul style="list-style-type: none"> <li>• Some experience with ESB techniques is already available;</li> <li>• Experience should be shared using a municipality- wide knowledgebase or forum;</li> <li>• Little experience with department crossing systems is available.</li> </ul>           | <ul style="list-style-type: none"> <li>• Awareness of the need and use of a clear architecture is growing;</li> <li>• Some persons have experience in dealing with integration projects and ESB techniques;</li> <li>• Available ICT knowledge and skills are scattered throughout the organization.</li> </ul> |
| Social dynamics      | <ul style="list-style-type: none"> <li>• The managerial layer, , both line and top layer, is difficult to convince of the need for interoperability and architectural consistency, partly caused by personal careers and political conviction;</li> <li>• Employees value ICT as “a given fact”, not as a flexible means which can be used to optimize processes;</li> <li>• Employees are inclined to focus on “their” process and lose touch with the overall process: the provided public service.</li> </ul> | <ul style="list-style-type: none"> <li>• The managerial layer should be more involved and encouraged by the municipal council to cooperate by means of ICT;</li> <li>• Adopting SLA's can cause serious implications, proper information and involvement of end-users is required.</li> </ul> | <ul style="list-style-type: none"> <li>• Autonomy aspects, insufficient authority, and conflicting interests can cause problems;</li> <li>• A strong leader should chair the iPlatform to coordinate the seated members (originating from various municipal services).</li> </ul>                               |

The situation in Groningen has evolved to an interesting situation where service-orientation can be exploited. The autonomous, decentralized and vertical oriented municipal silos are difficult to adapt or convince of the need for cooperation. As long as their tasks are carried out properly, why bother to invest in new techniques? When the currently provided services are formally defined and documented, they can be used as service contracts within and between the various departments. The iPlatform is assigned as centralized coordinating agency but lacks mandate and requires more authority to set up a municipality-wide ICT strategy and actually embed this in the organization. Convincing the top management of this need is hard due to lack of insight in the costs of current problems and the potential benefits which can be provided by an integrated municipal ICT approach. The size of Groningen suggests distributed workgroups to work together. These groups should be coordinated, or led, by the iPlatform.

The gained knowledge concerning ESB techniques should be carefully consolidated and shared using the iPlatform. The adoption of a data model on both municipal-wide and department level enables the distributed islands of functionality and data to cooperate. The systems which do not require perfectly up to date data can consult the data warehouse. The increasing need for real-time data suggests direct communication with several databases using the ESB.

## 5.6. *In summary*

We have interviewed four municipalities in the Netherlands concerning their municipal and ICT organization. For two scenarios, we have identified the involved departments, teams, processes and systems. Every municipality has its own interesting characteristics. Based on these characteristics and the gathered situation sketch, we have applied our suggested selection approach and derived a possible solution where the ESB has a central role. Based on this solution and the

specific municipal situation, we were able to predict the impact of the fictive implementation project to a certain extent using the BEST framework.

The flexible nature of the ESB provides municipalities the possibility to select the capabilities which are required for their specific situation. The ESB is able to supply a significant level of technical interoperability for municipalities. Therefore, technology is not the main problem to achieve a higher level of interoperability. For the analyzed administrations, the main problems can be expected at the organizational level of interoperability. Top-level support at the level of the municipal council and a centralized ICT department with sufficient mandate to make decisions are important elements of achieving sufficient harmony within the organization concerning agreements, adopted standards, and technologies.

---

# 6. Cross-case analysis

---

This chapter compares the studied municipal cases, highlights interesting similarities and differences, and provides a reflection on our analysis. Additionally we discuss some aspects in more detail to outline the problems that might occur during the selection and implementation process of ESB to facilitate SOA.

## 6.1. Case comparison

The studied municipalities have some interesting similarities. Enschede, Almelo and Groningen clearly lack top-level involvement concerning ICT-related issues. With top-level we refer to the level of the municipal council. Without support from this council, enterprise-wide changes are hard to define and implement successfully. During the years Enschede's and Almelo's ICT-organization has gained sufficient authority and mandate to operate properly, but a top-level sponsor who acts as ICT representative and takes full responsibility is not available. Such a sponsor should be able to span possible islands of authority and initiate necessary discussions at a high organizational level to span this boundary. Furthermore he is able to motivate public servants to embrace the advantages of ICT and focus on working together as one municipal entity [MUN01].

All municipalities lack a clear and formal description of involved business processes. Knowledge concerning such processes is located in either involved systems or employees. Since SOA heavily relies on dependencies between clearly defined services, such descriptions are required to exist and to be available throughout the organization. Prior to using a service, a potential service consumer should be aware of the required input, provided functionality and the output which can be expected. Enschede and Almelo already have some experience with the use of service contracts which formalizes cooperation and service quality to a certain extent. This knowledge can be used to set up contracts concerning automated cooperation as well.

Furthermore, a certain level experience with enterprise-wide implementation projects is available within all municipalities. Whether this knowledge is concentrated or scattered throughout the organization determines the effectiveness of such knowledge [LEG06]. The distributed character of Groningen resulted in scattered knowledge and experience and a heterogeneous architecture which is difficult to interpret and manage. Voorst on the opposite, has a clear view on its architecture. Additionally, the informal and centralized relationships between the employees in Voorst resulted in a high level of flexibility and reuse of knowledge. Experienced employees and responsible process owners should be combined in the project teams. This stimulates inter-disciplinary discussions which helps the development of solutions that are both organizationally and technically feasible. External partners should be involved for specialized knowledge; during this process, the risk for vendor-lock in should be acknowledged and avoided to achieve and maintain a sufficient level of interoperability.

A number of characterizing differences should be noted as well. Groningen struggles with its size and distributed character to install a municipality-wide ICT-organization. Currently the iPlatform acts as advising body for all ICT-organizations while a policy developing entity with sufficient authority is required to structure and coordinate the use of ICT for all departments. Again, top-level sponsorship at the

level of the municipal council might be able to connect the existing departmental islands and enforce a unified ICT strategy.

Voorst clearly profits from its relatively small size and, in this research, unique organizational structure. The adopted flower model with its distinction between the operational and project organization provides an effective instrument to avoid islands of authority that affect the progress of a project. Project management activities are positioned in the middle of the organizational flower to avoid conflicting interests and realize project teams that actually span the entire municipality. Since the municipal secretary is the chairman of initiated projects by default, communication with the municipal council can be arranged easily. This way, top-level involvement is increased, which eases the process of making municipal-wide decisions concerning ICT and to actually implement them as well.

## **6.2. Discussion**

The lack of formal process descriptions, involved process owners and related responsibilities can result in problems when services and service contracts have to be defined. It should be clear which department, team, and possibly even public servant is responsible for which part of a process, what input is required and what output can be expected. Based on this information, services can be defined, formalized, and published throughout the organization. This organizational level of interoperability should be clear for all involved parties before the underlying levels, semantic and technical, are addressed. Gained experience and new insights at these levels can cause changes in already discussed subjects or decisions made.

Semantic interoperability should be anticipated for to ensure proper communication between varying parties. A municipality-wide data model would be ideal, but probably impossible to realize. In this case, domain, or department, specific models should be considered. Message aggregation and correlation functionality which resides in the ESB can be used to connect various models, and therefore domains or departments. One should keep in mind that adopting a multiplicity of models results in a higher level of complexity caused by scattered maintenance tasks, stakeholders and involved parties.

Each municipal situation shows some unique characteristics where a potential integration solution should anticipate for. The flexible nature of ESB-based solutions provides municipalities the choice to select a suitable combination of ESB capabilities. Table 15 shows a broad variety in suggested capabilities, where the bigger municipalities such as Enschede and Groningen apply for 25 and 23 capabilities respectively, opposed to 14 and 18 capabilities for the municipalities Voorst and Almelo. This number of capabilities implies a certain level of required experience, knowledge and skills to develop, implement, and maintain the ESB-facilitated SOA. Only Groningen has some hands-on experience with the use of these technologies, the use of external expertise is expected to be beneficial for all municipalities. Long-term issues such as maintenance and further development should be covered at the organizational level of interoperability. The involved stakeholders and related responsibilities should be clear for all parties.

Table 15 bundles these and other interesting issues of the studied cases.



**Table 15: summarizing table highlighting interesting issues**

| Aspect                        | Point of interest                       | Enschede   | Voorst  | Almelo   | Groningen   |
|-------------------------------|---|--|---|--|---|
| <i>Organization</i>           | <i>Citizens</i>                         | • 153.000  | • 23.000  | • 73.000   | • 180.000   |
|                               | <i>Rank (of 460)</i>                    | • 12 <sup>th</sup>   | • 204 <sup>th</sup>   | • 45 <sup>th</sup>   | • 7 <sup>th</sup>   |
|                               | <i>Public servants</i>                  | • 1.600  | • 220   | • 720  | • 3.300   |
|                               | <i>Structure</i>                        | • Hierarchical;<br>• 5 departments.  | • Flower-model;<br>• 4 departments;<br>• 1 staff department.  | • Hierarchical;<br>• 7 departments;<br>• Centralized intake.   | • Hierarchical;<br>• 6 departments.   |
|                               | <i>ICT-organization</i>                 | • Embedded in the department Public Affairs;<br>• Municipal-wide ICT-authority;<br>• Sufficient mandate has been gained.   | • Small but recognized;<br>• Spread throughout the organization.  | • Embedded in the department Computerization and Automation;<br>• Municipal-wide ICT-authority;<br>• Sufficient mandate has been gained;<br>• An information policy plan guides the ICT-organization.  | • Department-specific ICT teams;<br>• ICT knowledge is Scattered around the organization;<br>• The iPlatform connects all ICT-managers;<br>• iPlatform lacks authority and mandate;   |
| <i>ICT Strategy and goals</i> | <i>Status</i>                           | • Clearly defined.   | • Unclear.  | • Clearly defined.   | • Unclear.  |
|                               | <i>Goals</i>                            | • Provide services directly targeted at the citizen;<br>• Increase level of electronic service delivery sophistication.  | • Provide a proper collection of services;<br>• Keep costs justified;<br>• Stimulate communication with local community.  | • Increase transparency;<br>• Improve provision of services;<br>• Improve operational management.  | • Improvement of communication with citizens;<br>• Reduction of operational costs;  |
|                               | <i>Strategy / plans</i>                 | • Setup foundation to develop and exploit a municipal services framework: the Dimpact-project.   | • Focus on creative and practical ideas.  | • Information policy plan / annual plan.   | • Analyze architectural problems and requirements (Architecture working group).   |
| <i>Management</i>             | <i>Top-level involvement</i>            | • Mediocre.  | • Sufficient.   | • Mediocre.  | • Limited.  |
|                               | <i>Top-level sponsor</i>                | • Not available.   | • Not available.  | • Not available.   | • Not available.  |
|                               | <i>ICT-management</i>                   | • Experienced;<br>• Concentrated in organization.  | • Experienced;<br>• Concentrated in organization.   | • Experienced;<br>• Concentrated in organization.  | • Experienced;<br>• Scattered throughout organization.  |
| <i>Process</i>                | <i>Formalization</i>                    | • Little formal documentation is available;<br>• Some experience with contracts is available.  | • Little formal documentation is available.   | • An increasing amount of documentation and insight is available;<br>• Some experience with contracts is available.  | • Little formal documentation is available.   |
|                               | <i>Development &amp; implementation</i> | • Some experience with enterprise-wide projects.   | • Little experience with enterprise-wide projects.  | • Little experience with enterprise-wide projects.   | • Some experience with department-wide projects.  |
|                               | <i>Maintenance</i>                      | • Increasingly formalized.   | • Little formalized.  | • Increasingly formalized.   | • Little formalized.  |
| <i>Knowledge and skills</i>   |   | • Experienced;<br>• No explicit experience with ESB & SOA;<br>• Concentrated in organization.  | • Little experience available;<br>• No explicit experience with ESB & SOA;<br>• Scattered throughout organization.  | • Experienced;<br>• Some experience with ESB & SOA;<br>• Concentrated in organization.   | • Experienced;<br>• Explicit, but hidden experience with ESB & SOA;<br>• Scattered throughout organization.   |
| <i>Social dynamics</i>        | <i>Internal relations</i>               | • Formalized.  | • Informal.   | • Close relations with other departments;<br>• Tend to arrange as much as possible informally.   | • Limited communication with other departments.   |
|                               | <i>Attitude towards ICT</i>             | • Increasingly involved.   | • Neutral.  | • Neutral, some departments are more involved.   | • Mainly reluctant, some departments are more involved.   |
| <i>Suggested solution</i>     | <i>EAI Scenario</i>                     | • 2: Facilitate wider connectivity to one or more applications.  | • 3: Facilitate wider connectivity to legacy systems.   | • 2: Facilitate wider connectivity to one or more applications.  | • 5: Implement controlled integration of services or systems between company divisions or partnering organizations.   |
|                               | <i>ESB Solution pattern</i>             | • Start with a Service Gateway;<br>• Consider future extension to a Web Services compliant broker.   | • Start with Basic Adapters;<br>• Consider future extension to a Service Gateway  | • Web Services compliant broker.   | • Start with a Service gateway;<br>• Consider future extension to Web Services compliant broker.  |
|                               | <i>Suggested capabilities</i>           | • 1. Communication (5/5);<br>• 2. Service interaction (3/4);<br>• 3. Integration (3/8);<br>• 5. Security (3/5);<br>• 6. Service level (2/4);<br>• 7. Message processing (3/9);<br>• 8. Management and autonomy (4/7);<br>• 9. Modeling (1/5);<br>• Total: 25 capabilities. | • 1. Communication (2/5);<br>• 2. Service interaction (2/4);<br>• 3. Integration (2/8);<br>• 7. Message processing (3/9);<br>• 8. Management and autonomy (4/7);<br>• 9. Modeling (1/5);<br>• Total: 14 capabilities. | • 1. Communication (2/5);<br>• 2. Service interaction (2/4);<br>• 3. Integration (2/8);<br>• 5. Security (2/5);<br>• 6. Service level (2/4);<br>• 7. Message processing (3/9);<br>• 8. Management and autonomy (4/7);<br>• 9. Modeling (1/5).<br>• Total: 18 capabilities. | • 1. Communication (4/5);<br>• 2. Service interaction (2/4);<br>• 3. Integration (4/8);<br>• 4. Quality of service (1/2)<br>• 5. Security (3/5);<br>• 6. Service level (2/4);<br>• 7. Message processing (2/9);<br>• 8. Management and autonomy (4/7);<br>• 9. Modeling (2/5);<br>• Total: 23 capabilities. |

### **6.3.     *Reflection***

For each municipality we have analyzed the same scenarios: requesting a felling license and getting married. Afterwards, we can say that the outcome of the analysis concerning these scenarios were slightly disappointing. All selected municipalities use a certain software package which guides the whole process involved in the analyzed scenarios. We expected more differences in handling the two scenarios. A municipality-specific selection of business process might have resulted in more interesting information. On the other hand, this situation outlines a certain degree of resemblance between the analyzed municipalities which increases the significance of adopting a standard for all municipal activities, making a nation-wide solution possible. However, the independent character of Dutch municipalities which is not likely to change in the near future prevents this interesting solution.

Despite the disappointing results concerning the analyzed scenarios, we think our analysis still provides a realistic view on the disparate situation within municipalities. For each municipality we have selected an interesting characteristic to focus on. By combining the four municipal situations and gained results while staying in the pre-defined scope, we assert that our findings can still be used to draw valid conclusions concerning the impact of ESB-facilitated SOA in municipalities.

---

# 7. Conclusions and recommendations

---

We have researched the level of increased interoperability which can be provided to Dutch municipalities by promising technologies such as ESB and SOA. Related to the potential introduction of these technologies we have analyzed the impact of a fictive enterprise-wide implementation project on the organization. This chapter presents our main findings and a recommended roadmap to ESB, and next, SOA adoption for municipalities. A reflection on our research and some suggestions for future research concludes this chapter.

## 7.1. Conclusions

Our research was based on a set of predefined questions which were used to answer the central research question. We present our conclusions by repeating and discussing the mentioned questions:

- *What needs for interoperable public services can be recognized?*  
The Dutch government and municipalities are becoming increasingly aware of the need for interoperability within and between governmental agencies. Nowadays, citizens require their government to provide accessible and high-quality services quickly and for a justified price. Since municipalities are responsible for the delivery of about 70 percent of these services, they are highly affected by the increased demand for quality. To meet these demands, municipalities should address three dimensions of interoperability:
  - *Organizational interoperability.* Municipalities are typically structured by means of municipal services (in an organizational sense) with a certain level of autonomy. In large municipalities, these services have gained a significant amount of autonomy, which can result in a disparate situation. To open up (a combined set of distributed) processes to other parts of the organization, citizens, or chain partners, interoperability between the municipal services, departments and teams is required. To achieve a higher level of organizational interoperability, insight and formalization of provided services, related processes, quality levels, and process owners is essential.
  - *Semantic interoperability.* Processing and sharing data that is distributed throughout an organization requires a data model which is adopted by all involved parties: the common data model. During the years, municipalities have adopted various task-specific, closed, or legacy data models. Therefore, standardization is required. Municipality-wide adoption of a nation-wide open data model for municipal issues (for instance StUF XML) would be ideal, but due to the vast variety of operational “standards” and guidelines within departments, this is not realistic for the near future. Segmentation of this envisioned data model based on specific tasks makes a consistent approach feasible. Combining these task-specific data models, which typically contain technology-bound models and jargon, to achieve semantic interoperability can be achieved by means of aggregation and correlation between the models at the crossing of department borders. Summarizing, within departments or certain

teams, the operational standards will be maintained. For information exchange between these parties, a standard should be used. Gradually, for instance when applications have to be replaced, legacy or task-specific data models can be sidetracked by selecting compatible standards.

- *Technical interoperability.* Typically, a vast collection of applications, databases, protocols and standards is in operation to support public servants in their daily tasks. To interconnect these concentrations of functionality and data, and to support the organizational and semantic dimensions of interoperability, an intelligent infrastructural foundation is required. This foundation should be based on widely accepted standards and technologies. The exact set of used interoperability standards and techniques should be in line with the environment it is planned to operate in. For the situation within Dutch municipalities, the technical interoperability framework can be used to structure the requirements of technical interoperability. This framework contains seven elements concerning standards and interoperability techniques. These elements should be analyzed concisely to determine the standards and techniques which are required for the situation in context.

- *What role can Service-Oriented Architecture have in the context of public services interoperability?*

The principles of SOA are in line with the service-delivering task of municipalities. The architecture aligns with the perception of administrators and public servants to deliver a service targeted at a service consumer, whether this is a citizen, partnering municipal department, or even a partnering municipality. Since service-delivery exists at various levels in the organization, we distinguished three types of services: business, application and technology services. Using three views which conform to these levels of service abstraction, more insight can be provided concerning the tasks and processes at various layers of the organization and their relation with each other. SOA can operate independent of platform, time, and location. This approach enables the development of highly distributed cross-organizational systems which can facilitate public services interoperability, both within as between municipalities. We conclude that the principles of SOA meet the requirements concerning interoperability as posed by municipalities and discussed in the previous research question.

However, adoption of complete service-orientation where all business processes are represented by a collection of services that interact with each other requires significant change concerning both technology and business. Employees need to realize that they need to cooperate to realize a coherent operating organization. With SOA this cooperation crosses the borders of traditional teams or departments. Certain activities that have been carried out might become superfluous when the organization realizes that another department or team has been doing the same thing for years. Therefore an inventory of formalized business process descriptions, involved parties and responsibilities is required to initiate the change process towards SOA. Additionally, service contracts need to be adopted to formalize the expectations between service consumer and provider. For municipalities these changes might be too big to start with, especially for smaller municipalities that lack sufficient resources and knowledge to exploit the benefits of service-orientation optimally. For municipalities which can benefit from SOA, a small introducing project to gain knowledge concerning the advantages and pitfalls of service-orientation can be useful.

- *What is the role of an Enterprise Service Bus in Service-Oriented Architecture?*  
The ESB is a collection of infrastructural middleware capabilities and implements the service bus pattern, one of the five key components of SOA.

The service bus is the infrastructural foundation of SOA and contains the capabilities which are required to deploy and interconnect services. The ESB provides additional capabilities such as message processing, integration, security and management functionality. The exact set of required capabilities depends on the current situation and envisioned (interoperable) situation. A scenario-based capability selection approach is proposed in section 4 and applied on four Dutch municipalities in section 5. We state that the ESB can act as a scalable infrastructural foundation of SOA which can be adopted in a preliminary stage of service-orientation. As soon as the level and sophistication of service-orientation matures, an increasing amount of requirements as posed to the operational infrastructure can be expected. The additional infrastructural middleware capabilities can be deployed on the ESB to align with the new situation.

- *Which organizational aspects are affected by embedding an Enterprise Service Bus in a public organization?*

The organizational impact caused by ESB adoption largely depends on the exact purpose for which the service bus is planned to be used in and what level of service-orientation is aimed for. The specific municipal situation and the defined objectives determine the collection of capabilities which are required to achieve the envisioned situation. An increasing amount of deployed capabilities comes with an increasing amount of required knowledge and skills concerning development, maintenance and monitoring tasks. When an ESB is implemented to achieve integration between two applications, little workload has to be prepared for once the system is up and running and minor impact on the organization should be anticipated for. On contrary, full SOA significantly influences the process and organizational domain. Processes are divided in services which are orchestrated by the service orchestration component. Service Level Agreements should be adopted to formalize quality and delivery of services. From an organizational view, departments and teams throughout an organization have to cooperate to deliver a product to the service-consumer. This consumer can be a citizen, a partnering municipality or an internal user. This can result in a changed and unexpected set of responsibilities that might conflict with other responsibilities of budgetary issues. For example, in Almelo the fire-department became responsible of making an inventory of all dangerous products in the region. Previously they did this as well, but in their own time using their own systems. Suddenly, the entire municipal organization became dependent on their input, increasing their responsibility.

Enterprise-wide SOA requires broad knowledge of involved systems, deployed services and their dependencies. From an organizational point of view, this requires a professional ICT-organization with sufficient authority to make and implement decisions. Organizational structures, in which certain departments have a high level of autonomy and particular interests, tend to complicate this requirement. A centralized enterprise-wide ICT-organization which has top-level support and a respected sponsor as chairman seems more plausible. A position as a staff services seems logical for the envisioned ICT-organization.

Based on the discussed research questions, our central research question can be answered.

|  |
|--|
| <i>“Which contribution to the integration of public services and realization of an interoperable government in the context of municipalities can an Enterprise Service Bus provide in Service-Oriented Architecture and what should be done to make such implementation successful?”</i> |
|--|

The advent of the Service-Oriented Architecture in public organizations is a logical step towards an effective way of administration. This type of architecture approaches architecture as a service facilitating platform rather than a data facilitating service.

Therefore this architecture is in line with the service providing task of municipalities and the way citizens expect services to be provided. Commitment and active support from the managerial top of a municipality is crucial in an enterprise-wide implementation. This enables the adoption and implementation of broad organizational regulations and standards which are essential in enterprise-wide integration projects. But even when top-level commitment is ensured and organizational and semantic issues have been addressed properly, successful adoption of service-orientation is not necessarily assured.

Since a service-oriented way of working requires a change of working mentality, commitment at lower levels is essential as well. Public servants in large municipalities tend to have difficulties with reminding the overall task of public service delivery. In small municipalities employees are typically responsible for the processing of a whole process. They know what, when, and how to do in order to complete the process. When this process is divided in smaller parts and spread over task-specific departments, the overview can get lost. This underlines the need for an up-to-date enterprise architecture, containing all relevant elements of a municipality to function properly. Proper process alignment and process coordination, cooperation, and formalized interoperability between employees, teams, departments and municipal services are essential elements of the envisioned interoperable government.

The described changes in working mentality, needs for formalized process descriptions and clear responsibilities of involved stakeholder has a great organizational impact on municipalities. Therefore the step towards (complete) service-orientation might be too big. A small project that introduces the principles, benefits, and pitfalls of SOA can be useful for municipalities to start with and to gain insight in the requirements at the level of organizational, semantic and technical interoperability. As the ESB acts as a scalable backbone of SOA, it can be used to implement a preliminary municipal-wide infrastructural foundation. On this foundation, knowledge can be gained during integration projects. The complexity of such projects depends on the municipal strategy, goals and managements willingness to invest in new technologies.

In the context of requirements as set by government agencies, the combination SOA and ESB cover all required aspects of the technical interoperability framework. Existing systems can be connected with the ESB, opening up functionality to other parts of the organization and possibly to citizens using the Internet. Various service types can be deployed on the bus, opening up technical, application or business functionality to potential service consumers. The use of a municipal common data model is essential to achieve syntactically and semantically correct communication between the providing and consuming parties. When involved parties require, or insist to use different models, aggregation and correlation between the models is required. The ESB can act as an intermediary layer and facilitate the required translations.

## **7.2. Recommendations**

Based on the conclusions, we suggest a set of recommendations which can guide municipalities in adopting emerging technologies such as ESB and SOA to achieve interoperability. First we introduce a set of common recommendations, followed by a so-called *roadmap to ESB and SOA adoption* and concluded with some recommendations specifically for TNO ICT.

### **7.2.1. Common recommendations**

These recommendations have a common nature, but are based on our analysis activities as performed in a municipal context. Therefore, they particularly apply to municipalities.

- Top-level commitment at the level of the municipal council is essential;
- The ICT-organization should span the entire organization and have sufficient authority and mandate; a central position as facilitating staff service should be considered;
- A top-level sponsor should be the chairman of the ICT organization, a strong municipal secretary should be considered for this role;
- A flower-based project model, as adopted by the municipality of Voorst should be considered for long-term projects;
- An overall up-to-date enterprise architecture should be available and maintained;
- Formalized business process descriptions and service level agreements should be available for involved parties. Based on these descriptions and agreements, services and related responsibilities, deliverables and quality can be determined;
- The selection of the required ESB capabilities should not be technology-driven, but business-driven. Based on the envisioned interoperability, the current situation and the business strategy, a proper set of capabilities should be selected;
- Service-orientation is not synonymous with interoperability. Currently, Web Services technologies are quite immature. Other, more mature and proven, approaches can be used to achieve interoperability as well. Service-orientation is a young approach which can have a significant impact on the organizational and process domain when adopted completely. This impact is mainly caused by the high level of introduced dependencies and information sharing. These issues should be acknowledged and realized by the initiating municipality.

The enumerated recommendations mainly cover our findings concerning organizational related problems. Interoperability projects should be business-driven, not technology-driven. But to realize such projects, the organization should be prepared to cope with involved changes, both expected and unexpected. We stress that involvement of employees and line management is essential; this involvement can be triggered by a top-level sponsor, preferably familiar with ICT, who encourages the organization and takes responsibility to complete the project successfully. Unfortunately, these persons are hard to find in governmental institutes.

### **7.2.2. Roadmap to ESB and SOA adoption**

This roadmap is based on the results of this research. It can be regarded as a summary of our derived conclusions that addresses the three recognized levels of interoperability, our suggested ESB capability selection approach and the adopted BEST framework. The roadmap supports decision makers to maintain overview of the situation and determine which step has to be taken at a particular stage of an enterprise-wide implementation project. When using this roadmap, it should be

realized that enterprise-wide integration projects are complex and dynamic projects and no uniform solution exists.

1. Define the integration project's goals and strategy in an (annual) project plan;
  - a. These should be in line with the business goals and strategy, which are preferably already defined in a long-term information policy plan;
  - b. The highest level of electronic service delivery may not be the holy grail, determine which goals are relevant for the organization.
2. Define and specify the required organizational interoperability:
  - a. Determine the role of departments and formalize them:
    - i. One centralized entity which is responsible for all intake activities is advised;
    - ii. Use service level agreements or service contracts to formalize the expectations between service consumer and provider;
  - b. Determine and formalize the role and responsibilities of process owners and actively involve them;
  - c. Involve potential external partners in an early stage;
  - d. Involve the ICT management organization at an early stage, education concerning ESB and SOA is plausible.
3. Define and specify the required semantic interoperability:
  - a. Define, or adopt a common data model:
    - i. When specific parties require task specific data models, (complex) aggregation and correlation procedures have to be defined;
    - ii. Try to maintain one model for future use within the architecture;
  - b. Define and adopt the level of granularity and naming standards of the services.
4. Define and specify the required technical interoperability:
  - a. Select the appropriate enterprise application integration-scenario;
  - b. Determine the preliminary level of infrastructure complexity:
    - i. A rich infrastructure combined with some pilot functions demonstrating the possibilities can have catalyzing effects on developments and employee commitment;
    - ii. A basic infrastructure is easy to maintain and rich functions can be deployed quickly, showcasing the added value to involved parts of the organization;
  - c. Select the ESB solution pattern which should be able to provide the required technical interoperability;
  - d. Select the required ESB capabilities by addressing the architecture decision issues related to the ESB solution pattern.
5. Assess the municipal situation and try to predict the impact on the organization, this can be done by analyzing the six elements of the BEST reference framework for each of the following processes:
  - a. Permanent business processes:
    - i. Should be properly defined, formalized, documented and made available for involved employees;
    - ii. An overview and ultimate goal of a public service reminds employees of their task in the whole process.
  - b. Enterprise system design processes:
    - i. Automation of business processes should be performed carefully; a bad business process does not get better just by means of automation;
    - ii. Review process responsibilities and service level agreements to conform to the envisioned situation.
  - c. Project management processes:
    - i. Determine the project's level of complexity for the organization and involved parties;
    - ii. Define the role and responsibilities of the involved parties, conflicting roles and related risks should be clearly described and acknowledged;
    - iii. Assign a supporting top level sponsor who encourages user involvement and creates acceptance for unpopular, but required decisions;
    - iv. Recognize and respect the lack of long term project management skills and consider adopting methodologies or attracting external experience to compensate these inabilities;
    - v. Regularly check whether the envisioned goals are still in line with performed activities.

### **7.2.3. Recommendations for TNO ICT**

TNO ICT suggested this research to clarify the principles of service-orientation, the role for ESB in the envisioned SOA, and the use of both techniques in a municipal context. To further increase and spread this knowledge we recommend to formalize and examine the suggested roadmap in practice. A methodology to derive specifications of service-based solutions for organizational issues would be ideal for future use. This way, a clear relation between organizational and supporting technical parts can be identified and used to introduce service orientation in an organization.

At the organizational level it is important to be able to explain and justify involved issues to high level entities in a municipality, i.e. the municipal council. A clear description of ESB and SOA, the added value and an indication of the costs are useful during initial discussions. When understanding can be created at a high organizational level, the potential municipality-wide adoption of ICT in common and



ESB and SOA in particular can be coordinated centrally which is beneficial for all related parties, including TNO ICT.

From a technical viewpoint, increased hands-on experience with an ESB-facilitated SOA is required to demonstrate the benefits of service-orientation and the use of ESB in this approach. Using the suggested roadmap, a fictive municipality can be set-up to showcase the possibilities and highlight important, possibly yet unknown, issues.

#### **7.2.4. Reflection**

This research has been performed after an emerging popularity of ESB and SOA was noticed. This popularity can be attributed to the ICT and consultancy sector's emerging trend of creating a hype every few years. It cannot be ignored that large solution suppliers such as Microsoft, IBM, SAP, Sun, and Oracle are re-branding their traditional products to service-oriented products. As discussed with senior architects who have been developing information systems since the late '70s, services are not a new concept at all. The combination of new network technologies such as the Internet, widely accepted standards such as XML and modular software development methodologies have paved the way for SOA. This 'new' approach of setting up information systems in a loosely coupled distributed manner poses a collection of requirements to the underlying infrastructure. Candidate solutions which meet these requirements such as message processing, addressing, naming, and security features have been available for some time already. The ICT industry has chosen the term 'ESB' to relate to this collection of infrastructural capabilities. The combination of SOA and ESB is regarded to be the new 'way' of connecting collections of functionality and data over widely spread networks. This does not mean that old techniques can not be used to achieve a similar solution; the adoption of accepted standards and techniques, a characteristic of the ESB, *does* prevent the creation of another legacy solution within an organization or industry sector.

On their turn, municipalities have been hit by an explosive 'need' for online service delivery by their consumers. The sudden growth of Internet connectivity, need for transparency and the required flexibility to meet those needs is not something a typical governmental agency is designed for. The problems these organizations have with aligning their organizational structures and procedures can be attributed to the fact that administrations are historically statically organized for one specific purpose or task only. The introduction of new products in such organizations is usually related with some resistance and problems. But the introduction of a service-oriented way of handling requests, not only towards citizens but towards and between other municipal departments requires a significant amount of effort and acceptance by involved employees to realize.

The potential adoption of ESB and SOA by the public sector, municipalities in particular, was the subject of this research. The empirical part is based on four different municipalities. Based on size, this number is not a realistic reflection of the 460 Dutch municipalities. For this research, Voorst can be regarded as a small municipality, but compared to all Dutch municipalities, it is an average one at position 204. The Netherlands has about 60 municipalities with over 50.000 habitants; the remaining 400 municipalities have fewer inhabitants. Therefore this research particularly applies to above-average municipalities. The average and under-average municipalities are expected to have less problems with island-automation and highly autonomous parts of the organization, but they typically have fewer resources to solve these problems as well. For these 'smaller' municipalities it would be interesting to study the feasibility of electronic cooperation based on the application service provider pattern. An ESB-facilitated SOA may be a candidate to share common functionality and involved costs in this 'coalition' of connected municipalities.

During our interviews at the four selected municipalities, we interviewed a number of key-persons from each municipality. The background of these employees varied from a high level *strategic* advisor of the municipal board in Groningen, to a part-time information advisor with a highly *technical* approach in Voorst. These

differences in backgrounds, ways of thinking about ICT in municipalities and even involved political convictions influenced our research. The questions which were discussed during the interviews were defined and structured using the BEST framework and not always properly aligned with the interviewed persons. Additionally, the project management part of the BEST framework was assumed to be less relevant for this research. Therefore, the questions concerning this subject were prepared insufficiently to define clear and consistent interview questions. A more thorough knowledge of project management theory and familiar issues could have increased the quality of the acquired information and structure of our analysis.

### **7.3. *Future research***

The transition to the envisioned interoperable customer-oriented municipality can be initiated by some simple integration project, acting as a pilot. This is where the flexible character of the ESB comes in handy. Gradually the collection of provided infrastructural middleware capabilities can be extended. This way the infrastructural foundation of municipal SOA can be deployed in a municipality. But still some interesting aspects have to be analyzed. The level of service abstraction, also referred to as service granularity is very important when services are defined at various abstraction layers, which in practice always happens. We have distinguished three service types, which are assumed to function either internal or external. Whether this classification of services suffices, is subject to future research.

The Model-Driven Architecture (MDA) envisions an approach where system functionality is modeled in both a computation and platform-independent way, after which these models are transformed in platform-specific models. Based on these models, (skeletons of) code can be generated which can be further developed and eventually deployed as software. This is expected to decrease the gap between architectural models and the actual systems. For service orientation this would be interesting since a system is build up from small modules of functionality which can be easily reused. Alignment between business and the developed services can possibly be coordinated by means of this approach. Future research on this subject should be performed how the mapping from business to functionality can be performed in a structured and semi-automated way. An introduction to the MDA is provided in Appendix VI.

From the viewpoint of ICT-management, it is interesting to study alternative ways to manage services properly. When these services are operational in a highly distributed environment, the management and maintenance is expected to differ from traditional systems that can be managed centrally.

---

# References

---

## Literature

| Reference | Description  |
|-----------|--|
| [ADV05]   | Advies Overheid.nl (2005), <i>Overheid.nl Monitor</i> , <a href="http://www.advies.overheid.nl/monitor/">http://www.advies.overheid.nl/monitor/</a>  |
| [AGI05]   | The Australian Government Information Management Office (Agimo)(2005), <i>Australian Government Technical Interoperability Framework (AGTIF)</i> , Chief Information Officer Committee (CIOC), Australia   |
| [ALH03]   | Alhir, S. S. (2003), <i>Understanding the Model-Driven Architecture (MDA)</i> , Methods & Tools, October 2003<br><a href="http://home.comcast.net/~salhir/UnderstandingTheMDA.pdf">http://home.comcast.net/~salhir/UnderstandingTheMDA.pdf</a>   |
| [ALM00]   | Al-Mashiri, M., Zairi, M. (2000), <i>The effective application of SAP R/3: A proposed model of best practice</i> , Logistics Information Management 13 (3)m, pages 156-166, 2000   |
| [ALM02]   | Municipality of Almelo (2002), <i>Collegeprogramma 2002-2006</i> , Almelo, the Netherlands<br><a href="http://www.almelo.nl">http://www.almelo.nl</a>  |
| [AOV05]   | Dutch government (2005), <i>Andere overheid – het programma</i> , the Hague, the Netherlands, 2005<br><a href="http://www.andereoverheid.nl/AndereOverheid/Web/Het+Programma/">http://www.andereoverheid.nl/AndereOverheid/Web/Het+Programma/</a>  |
| [BAA05]   | Baars, C. (2005), <i>SOA/ESB The Business-IT</i> , Cibit  Serc, presentation held at the SOA seminar, Utrecht, the Netherlands, October 2005   |
| [BAC05]   | Bacham, J. (2005) <i>Movin' SOA On Up: Introducing a new Service-Oriented Maturity Model</i> , SOA Insights, Sonic Software, September 2005,<br><a href="http://www.sonicsoftware.com/solutions/learning_center/soa_insights/movin_soa_on_up/index.ssp">http://www.sonicsoftware.com/solutions/learning_center/soa_insights/movin_soa_on_up/index.ssp</a>            |
| [BEN76]   | Bennis, W.G., Benne, K.D. (1976), <i>The planning of change</i> , Holt, Rinehart and Winston, New York, New York, U.S.A<br>ISBN: 0-03-089518-9   |
| [BIJ05]   | Bijl, D. (2005), <i>Service Oriëntatie en ICT</i> , Microsoft BV, 2005   |
| [BOD05]   | Boddy, D., Boonstra, A., Kennedy, G. (2005), <i>Managing Information Systems: an organizational perspective</i> , Prentice Hall, Pearson Education Inc, Edinburg Gate, Harlow, United Kingdom<br>ISBN: 0-273-686356  |
| [CAP05]   | Cap Gemini (2005), <i>Online availability of public services: how is Europe progressing?</i> , Cap Gemini, March 2005<br><a href="http://europa.eu.int/information_society/soccul/egov/egov_benchmarking_2005.pdf">http://europa.eu.int/information_society/soccul/egov/egov_benchmarking_2005.pdf</a>   |
| [CBS05]   | Statistics Netherlands (CBS) (2005), <i>Population per region</i> , Statistics Netherlands, Voorburg, The Netherlands, 2005<br><a href="http://statline.cbs.nl/">http://statline.cbs.nl/</a>   |
| [CHA04a]  | Channabasavaiah, K., Holley, K., Tuggle, E. M. (2004) , <i>Migrating to a service-oriented architecture</i> , International Business Machines Corporation, Castle Drive Armonk, New York, U.S.A, April 2004  |
| [CHA04b]  | Chappell, D. (2004), <i>Enterprise Service Bus</i> , O' Reilly Media, Sebastopol, California, U.S.A, ISBN: 0-596-00675-6   |
| [CRA03]   | Craggs, S. (2003), <i>Best-of-Breed ESBs: Identifying best-of-breed characteristics in Enterprise Services Busses</i> , Saint Consulting Limited, Hampshire, United Kingdom, June 2003,<br><a href="http://www.sonicsoftware.com/products/whitepapers/docs/best_of_breed_esbs.pdf">http://www.sonicsoftware.com/products/whitepapers/docs/best_of_breed_esbs.pdf</a> |

- [EIF04] Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens (IDABC, formerly IDA)(2004) *European Interoperability Framework for pan-European eGovernment services*, IDABC, European Communities, November 2004, ISBN: 92-894-8389-X  
<http://europa.eu.int/idabc/servlets/Doc?id=19529>
- [ELO05] Kenniscentrum Electronische Overheid (2005), *e-Government in Nederland: een beknopte geschiedenis*, Kenniscentrum ELO, The Hague, The Netherlands, January 2005, [http://www.elo.nl/elo/Images/Geschiedenis-egov-nl\\_tcm70-54316.pdf](http://www.elo.nl/elo/Images/Geschiedenis-egov-nl_tcm70-54316.pdf)
- [ENS02] Municipality of Enschede (2002), *E-Mail organisatie: Gemeente Enschede, Digitaal Loket*, [http://www.enschede.nl/nieuw/Topframe/Mail\\_VROM\\_Gemeente\\_Enschede\\_Gereedschap-def.pdf](http://www.enschede.nl/nieuw/Topframe/Mail_VROM_Gemeente_Enschede_Gereedschap-def.pdf)
- [ERR05] Erradi, A, Maheshwadi, P. (2005), *wsBus: QoS-Aware Middleware for Reliable Web Services Interactions*, 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05), 2005, pages 634-639
- [ES05] Es, van R., Gerwen, N., Graave, J., Ligthart, A., Rooij, van R. (2005), *Service-Oriented Architecture – een praktische leidraad voor invoering: Socrates™*, Academic Service, Ordina System Integration & Development BV en Sdu Uitgevers BV, The Hague, The Netherlands, 2005. ISSN 1566-3922, ISBN 90-395-2431-9, NUR 980
- [FIN03] Finetti, M. (2003), *Interoperability matters*, Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens (IDABC, formerly IDA), European Communities, 2003  
<http://www.publicservice.co.uk/pdf/europe/autumn2003/manuelaf.pdf>
- [FIN05] Fingar, P. (2005), *Government's BPM Imperative*, Chief Information Officer Australia, September 2005  
<http://www.cio.com.au/index.php/id;882797746>
- [GEU02] Geurts, P.J.M, Hulst, van der, J.J.M. (2002), *Het mid-office bestaat niet*, Cap Gemini Ernst & Young, Utrecht, The Netherlands, May 2002
- [GIL04] Gilpin, M., Vollmer, K. Peyret, H., Dowling, K.Q. (2004), *What Is An Enterprise Service Bus?*, Forrester research, August 2004, <http://www.forrester.com/Research/Document/Excerpt/0,7211,35193,00.html>
- [GRA04] de Graaf, T.C. (2004), *Voortgangsrapportage 2004 - Programma Andere Overheid*, the Hague, the Netherlands, 2004  
<http://www.andereoverheid.nl/NR/rdonlyres/9073A6F7-0B54-4339-BC15-228C0BDA8001/0/Voortgangsrapportage.pdf>
- [GRO02] Municipality of Groningen, *Collegeprogramma 2002-2006*, Groningen, the Netherlands, <http://www.groningen.nl>
- [GOO05] Google Earth, Launched in June 2005:  
<http://earth.google.com>
- [HEU04] van den Heuvel, W.J.A.M. (2003) *Call for papers: International Conference on Service Oriented Computing(IC-SOC)*, Trento, Italy, 2003
- [HUL76] Hulshof, A.H. , (1976), *College Organisatiekunde*, University of Twente, Enschede, The Netherlands, 1976
- [IBM04] International Business Machines Corporation (2004), *Patterns: Implementing an SOA Using an Enterprise Service Bus*, North Castle Drive Armonk, New York, U.S.A, 2004, ISBN:0738490008
- [JAA05] Jaakkola, L. (2005), *Masters' thesis: Applying Service-Oriented Architecture to Geographically Distributed Industrial Information Systems*, Helsinki University of Technology, Helsinki, Finland, April 2005, <http://users.tkk.fi/~ljaakkol/thesis/jaakkola-lauri-050411.pdf>
- [JON03] Jonkers, H., Buuren, van, R., Arbab, F., Boer, de, F., Bonsangue, M., Bosma, H., Doest, ter, H., Groenewegen, L., Scholten, J.G., Hoppenbrouwers, S., Iacob, M.E., Janssen, W., Lankhorst, M., Leeuwen, van, D, Proper, E., Stam, A., Torre, van der, L., Zanten, van, G.V. (2003), *Towards a Language for Coherent Enterprise Architecture Descriptions*, Enterprise Distributed Object Computing Conference, Proceedings of the 7th International Conference on Enterprise Distributed Object Computing, September 2003, pages:28 - 37  
ISBN:0-7695-1994-6  
[http://www.cs.ru.nl/~veldhvz/papers/edoc2003\\_jonk.pdf](http://www.cs.ru.nl/~veldhvz/papers/edoc2003_jonk.pdf)

- [JON99] de Jong, H.M., Schuszler, P.A. (1999), *Staatkunde Nederland in drievoud*, Bussum, the Netherlands: Coutinho, 1999
- [KRA05] Krafzig, D., Banke, K., Slama, D. (2005), *Enterprise SOA: Service-Oriented Architecture best practices*, Prentice Hall, Pearson Education Inc, Upper Saddle River, New York, U.S.A, 2005 ISBN: 0-13-146575-9
- [LAN05] Lankhorst, M.M. (2005), *Enterprise Architecture at Work - Modelling, Communication, and Analysis*, Springer-Verlag, Berlin, 2005. ISBN 3-540-24371-2
- [LEE04] Leenes, R., Svensson, J. (2004), *Electronic Services in a Decentralized State*, R. Traunmüller, (Ed.), *Electronic Government, Third International Conference, EGOV 2004 (Lecture Notes in Computer Science, LNCS3183)* (pp. 496-502). Berlin Heidelberg: Springer-Verlag.
- [LEE05] Leenes, R.E., Svensson, S.J. (2005), *Local eGovernment in the Netherlands*, Tilburg, the Netherlands: Tilburg University, *Local Electronic Government, an international comparison*, Routledge, 2005, pp 199-155  
<http://arno.uvt.nl/show.cgi?fid=14433>
- [LUT04] Oude Luttighuis, P. (2004), *Over diensten en bussen - Geen verhaal over busdiensten (of toch?)*, TNO, Programme OSSOS, Enschede, the Netherlands, 2004
- [MCI04] McIntosh, R.L. (2004), *Open-Source Tools for Distributed Device Control Within a Service-Oriented Architecture*, *Journal of the Association for Laboratory Automation*, Volume 9, Issue 6, December 2004, pages 404-410
- [MIN98] Ministry of internal affairs (1998), *Electronic Government Action Programme: A more efficient and effective government on the electronic highway*, Dutch Government, 1998  
<http://www.minbzk.nl/contents/pages/7597/electronicgovernmentactionprogramme.pdf>
- [MOM06] Mom, P. (2006), *Vijf gemeenten sluiten pact voor aanbesteding*, *Automatiseringsgids*, issue 11, March 17th 2006  
[http://automatiseringgids.sdu.nl/ag/nieuws/nieuws/toon\\_nieuwsbericht.jsp?di=181667](http://automatiseringgids.sdu.nl/ag/nieuws/nieuws/toon_nieuwsbericht.jsp?di=181667)
- [MUN01] Muntslag, D.R. (2001), *De kunst van het implementeren*, inaugural lecture, University of Twente, Enschede, The Netherlands
- [NAT03] Natis, Y.V. (2003), *Service-Oriented Architecture Scenario*, Gartner Incorporated, April 2003, <http://www.gartner.com/resources/114300/114358/114358.pdf>
- [NIJ94] Nijholt, N., Ende, van den, J. (1994), *Geschiedenis van de rekenkunst, van kerfstok tot computer*, Academic Service, Schoonhoven, the Netherlands, 1994, ISBN: 90-395-0048-7
- [NOT04] C. Nott (2004), *Patterns: Using Business Service Choreography In Conjunction With An Enterprise Service Bus*, *International Business Machines Corporation*, North Castle Drive Armonk, New York, U.S.A, October 2004
- [OMG03] The Object Management Group (2003), *MDA Guide*, June 2003, <http://www.omg.org/docs/omg/03-06-01.pdf>
- [OSS05] Pogramma Open Standaarden en Open Source Software voor de overheid, *Waarom OSSOS?*, *The Hague, The Netherlands*, 2005  
<http://www.ossos.nl/index.jsp?alias=waaromososs>
- [PAP03] Papazoglu, M.P. (2003), *Service-Oriented Computing: Concepts, Characteristics and Directions*, INFOLAB, department of Information Systems and Management, Tilburg University, Tilburg, The Netherlands, October 2003  
<http://infolab.uvt.nl/pub/papazogloup-2003-51.pdf>
- [PAP05] Papazoglou, M.P., Heuvel, van den, W. (2005), *Service-oriented Architectures: Approaches, Technologies and Research Issues*, INFOLAB, department of Information Systems and Management, Tilburg University, Tilburg, The Netherlands, June 2005, <http://infolab.uvt.nl/pub/papazogloup-2005-81.pdf>
- [PAT04] Patrick, P. (2004), *Impact of SOA on Enterprise Information Architectures*, BEA Systems Incorporated, Burlington, Massachusetts, U.S.A. , 2004
- [POL05] Polarlake, *Understanding the ESB - What it is, why it matters, and how to choose one*, Polarlake, Dublin, Ireland  
<http://www.polarlake.com/en/assets/whitepapers/esb.pdf>

- [SEB05] See Beyond – Sun Microsystems, 2005, *Governments*, Global Corporate Headquarters, Monrovia, California, U.S.A, 2005  
<http://www.seebeyond.com/industries/government.asp>
- [STA04] Stantchev, T, Scherz, M. (2004), *Challenges of Service-oriented Architectures*, Proceedings of IPSI2004, Venice, Italy, 2004  
[http://www.informatik.hu-berlin.de/~vstantch/pubs/stantchev\\_scherz\\_soa\\_1104.pdf](http://www.informatik.hu-berlin.de/~vstantch/pubs/stantchev_scherz_soa_1104.pdf)
- [STE02] Stevens, M. (2002), *Service-Oriented Architecture introduction, part 1*, JupiterMedia, 2002,  
<http://www.developer.com/design/article.php/1010451>
- [STE04] Steen, M.W.A., Strating, P., Lankhorst, M.M., ter Doest, H., Iacob, M.E. (2004), *Service-Oriented Enterprise Architecture*, Telematica Instituut, Enschede, the Netherlands, December 2004,  
[https://doc.telin.nl/dscgi/ds.py/Get/File-47205/SOEA\\_section\\_v14.pdf](https://doc.telin.nl/dscgi/ds.py/Get/File-47205/SOEA_section_v14.pdf)
- [TOO04a] O'Toole, A. (2004), *Service-Oriented Architecture: The Right Side Up*, Cape Clear Software, 2004  
[http://www.capeclear.com/technology/clearthinking/SOA\\_The%20Right\\_Side\\_Up.pdf](http://www.capeclear.com/technology/clearthinking/SOA_The%20Right_Side_Up.pdf)
- [TOO04b] O'Toole, A. (2004), *Business Services - The Business Application of Web Services*, Cape Clear Software, 2004  
[http://www.capeclear.com/technology/clearthinking/Business\\_Services.pdf](http://www.capeclear.com/technology/clearthinking/Business_Services.pdf)
- [VER00] Verschuren, P., Doorewaard, H. (2005), *Het ontwerpen van een onderzoek*, Utrecht, the Netherlands: Lemma BV, 2005, 3rd edition, 7th circulation, ISBN: 90-5189-886-X
- [VOO05] Municipality of Voorst (2005), *Organisatieplan 2005*,  
[http://www.voorst.nl/Internet/Organisatie/Documenten/Organisatieplan2005\\_def.pdf](http://www.voorst.nl/Internet/Organisatie/Documenten/Organisatieplan2005_def.pdf)
- [W3C04] W3C Web Services Architecture Working Group (2004), *Web Services Architecture Working Group Note*, W3C, February 2004  
<http://www.w3.org/TR/ws-arch/>
- [WAG01] Wagter, R., Berg, M. van den, Luijpers, J., Steenbergen, M. van (2001), *DYA® - Snelheid en samenhang in business- en ICT-architectuur*, Sogeti Nederland B.V., Tutein Nolthenius, Den Bosch, 3rd Edition, August 2002, ISBN 90-72194-62-4.
- [WAL05] Gateway to the Walloon Region (2005), *Valeur ajoutée*, 2005  
[http://lamp.wallonie.be/ume/article.php3?id\\_article=40](http://lamp.wallonie.be/ume/article.php3?id_article=40)
- [WEB05] Webdam (2005), *Online Monitor*, Gemeenteweb, <http://www.webdam.nl>
- [WEL99] Welte, N., de Schepper, A. (1999), *Successful SAP R/3 Implementation: Practical Management of ERP Projects*, Massachusetts, U.S.A: Addison-Wesley Longman Publishing Company Incorporated, Boston, 1999
- [WIJ99] Wijnen, G., Renes, W., Storm, P.M., *Projectmatig werken*, Het spectrum, Utrecht, The Netherlands  
ISBN: 90-274-3328-3
- [WIL04] Wilkes, S. (2004), *Loosen up*, Dev2Dev, April 2004  
[http://dev2dev.bea.com/pub/a/2004/11/bizlogic\\_wilkes.htm-1](http://dev2dev.bea.com/pub/a/2004/11/bizlogic_wilkes.htm-1)
- [WOG04] Wognum, P.M., Krabbendam, J.J., Buhl, H., Ma, X., Kenett, R. (2004), *Improving enterprise system support – a case based approach*, Advanced Engineering Informatics 18 (2004), pages 241-253  
[http://www.kpa.co.il/english/publications/papers/JAEI\\_Wognum\\_%20Ma\\_Buhl\\_Kenett\\_2005.PDF](http://www.kpa.co.il/english/publications/papers/JAEI_Wognum_%20Ma_Buhl_Kenett_2005.PDF)
- [WSI04] Web Services interoperability organization (2004), *Basic profile version 1.1*,  
<http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html>
- [ZAC87] Zachman, J.A. (1987), *A Framework for Information Systems Architecture*, IBM Systems Journal, 26 (3), 1987, pages: 276 – 292  
<http://www.research.ibm.com/journal/sj/263/ibmsj2603E.pdf>
- [ZAL77] Zaltman, G., Duncan, R.B. (1977), *Strategies for planned change*, Wiley, New York, New York, U.S.A.  
ISBN: 0-471-98131-1
- [ZOU05] Zouridis, S., Thaens, M., Vrijling, I. (2005), *Vliegwiél voor de wil: Superpilots en elektronische dienstverlening in gemeenten*, Tilburg, the Netherlands: Tilburg University  
<http://www.elo.nl/elo/actueel/nieuwsarchief/2005/05May/20/vliegwiél.jsp>

## Interviews

- [GRI05] Grimmelikhuyse, R. (2005), *Interview concerning eGovernment at the municipality of Enschede*, team manager development, municipality of Enschede, Enschede, The Netherlands, December 2005
- [HYL06] Hylkema, G., Top, C. (2006), *Interview concerning eGovernment at the municipality of Groningen*, strategic information advisor and information manager department of Environment respectively, municipality of Groningen, Groningen, January 2006
- [LEG06] Legtenberg, E. (2006), *Interview concerning eGovernment at the municipality of Almelo*, information advisor, municipality of Almelo, Almelo, January 2006
- [LUT05] Oude-Luttighuis, P. (2005), *Interview concerning eGovernment in the Netherlands*, consultant public sector, TNO ICT, Enschede, August 2005
- [PET06] Peters, H. (2006), *Interview concerning eGovernment at the municipality of Voorst*, information advisor, municipality of Voorst, Voorst, The Netherlands, January 2006
- [UIT06] Uiterweerd, J. (2006), *Interview by telephone concerning the procedure of getting married at the municipality of Voorst*, coordinator Public Affairs, municipality of Voorst, Voorst, The Netherlands, January 2006
- [WOB06] Wobbes, H. (2006), *Interview concerning the implementation of an ESB within the service Spatial development and social affairs*, Advisor ICT / Automation municipality of Groningen, service Spatial development and social affairs, Groningen, The Netherlands, April 2006

---

# Appendix I.

## Glossary

---

| Term                    | Description   |
|-------------------------|---|
| API                     | Application Program Interface.  |
| Back-office             | The part of the organization which registers and processes requests which are received through the front-office. The back-office contains the processes, information and applications which are responsible for the municipal primary processes. The eventual product / public service is delivered to the service-consumer directly or through the front-office. Direct communication with a service-consumer is unusual, but can occur in exceptional situations. |
| BEST Framework          | Better Enterprise SysTEm implementation framework, a framework based on significant number of enterprise-wide implementation projects and related implementation process dynamics. This framework enables a consultant to determine the maturity of an organization that is about to initiate an enterprise-wide system implementation project and to identify human and organizational risks involved in such a project.   |
| BNG                     | Bank Dutch Municipalities, ("Bank Nederlandse Gemeenten")   |
| BPEL                    | Business Process Execution Language.  |
| BPM                     | Business Process Modeling.  |
| CICS                    | Customer Information Control System, a transaction server that is nowadays considered a legacy system.  |
| COM                     | Component Object Model.   |
| COTS                    | Commercial Off-The-Shelf, systems which are manufactured commercially, and then may be tailored for specific uses.  |
| DBMS                    | DataBase Management System  |
| EAI                     | Enterprise Application Integration.   |
| EDI                     | Electronic Data Interchange   |
| EGEM                    | Electronic Municipalities ("Electronische Gemeenten").  |
| eGovernment             | The utilization of electronic technology to streamline or otherwise improve the business of government and to stimulate citizen participation in democratic institutions and processes.   |
| End-to-end security     | Security is realized between the consuming and the providing entity, even when there are intermediates involved.  |
| Enterprise architecture | A coherent whole of principles, methods and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructures.   |
| ESB                     | Enterprise Service Bus: software infrastructure that enables a Service-Oriented Architecture by acting as an intermediary layer of middleware through which a set of reusable services are made widely available.   |



|                         |   |
|-------------------------|---|
| Front-office            | Both the physical (service-counter) and semi- or non-physical (post, fax, telephone, email, webpage) channels through which a service consumer communicates or performs transactions with an organization, in the context of this research: a municipality. Therefore, the front-office covers the processes, information and applications responsible for the decoupling between citizen and (back-end) municipality. An important task of the front-office is to query and coordinate/orchestrate the activities performed in the back-office properly. |
| FTP                     | File Transfer Protocol.   |
| GBA                     | Municipal Public Records Database ("Gemeentelijke Basis Administratie").  |
| HTTP                    | Hypertext Transfer Protocol.  |
| HUB                     | Centralized messaging components in an architecture.  |
| ICT                     | Information and Communication Technology.   |
| IDL                     | Interface Definition Language.  |
| Interoperability        | The ability to transfer and use information in a uniform and efficient manner across multiple organizations and ICT systems. It underpins the growing level of benefits for enterprises, government and the wider economy through e-commerce.   |
| Itinerary-based routing | A response is not sent to the original requester, but is sent as a new message to a forwarding address.   |
| J2EE                    | Java 2 Platform, Enterprise Edition.  |
| JCA                     | Java Connector Architecture.  |
| JDBC                    | Java DataBase Connectivity, Java API for database connectivity in SQL   |
| JMS                     | Java Messaging Service, J2EE API for messaging purposes.  |
| MDA                     | Model-Driven Architecture.  |
| MOF                     | Meta-Object Facility.   |
| OLE21                   | 21 <sup>st</sup> century governmental counter Enschede ("Overheidsloket Enschede van de 21e eeuw").   |
| OMG                     | Object Management Group.  |
| OWL                     | Ontology Web Language.  |
| PIM                     | Platform Independent Model.   |
| PIV4all                 | Person Information Facility, "PersoonsInformatieVoorziening"  |
| Point-to-point security | The client and the intermediate server need to authenticate themselves, after which a secure connection is set up to communicate through. This type of security operates between end-points, not between the consumer and provider.   |
| PPP                     | Persons, Premises, and Parcels ("Personen, Panden en Percelen").  |
| PSM                     | Platform Specific Model.  |
| Pub/sub                 | Publish / subscribe mechanism.  |
| RMI                     | Remote Method Invocation, communication protocol developed by Sun to provide connectivity between Java objects located on various machines.   |
| Service                 | A unit of functionality that a service provider (e.g. organization, department, or system) makes available to its environment through an interface, and which has some value for service consumers in the environment of the service.   |
| SLA                     | Service Level Agreement, defines the expectations between service consumer and provider. It helps define and clarify the relationship between the involved parties.   |
| SOA                     | Service-Oriented Architecture: an application architecture within which all functions are defined and published as independent services with well-defined interfaces, which can all be invoked through a service bus in defined sequences to form business processes accessible via an application front-end.   |
| SOAP                    | Simple Object Access Protocol.  |
| SQL                     | Standard Query Language.  |

|           |  |
|-----------|--|
| StUF      | Standard Exchange Format (Standaard Uitwisselings Formaat)                                   |
| UDDI      | Universal Description, Discovery and Integration.  |
| W3C       | World Wide Web Consortium  |
| WSDL      | Web Services Description Language.   |
| WWB / AWB | Act Employment and Assistance (“Wet Werk en Bijstand”, previously “Algemene Bijstands Wet”). |
| XML       | eXtensible Markup Language.  |

Throughout this research a number of governmental and municipal related phrases have been used. Since no formal translation rules exist for such phrases, literal translation can cause confusion, therefore a list of translations is provided to guide the (Dutch) reader.

| <b>Government-related phrase</b>              | <b>Dutch translation</b>                                 |
|---|--|
| Board of mayor and alderman                   | College van Burgemeester en wethouder(s)                 |
| Clerk and registry                            | Griffier en griffie                                      |
| Common administrative affairs                 | Algemeen bestuurlijke zaken                              |
| Department maintenance of public services     | Afdeling publieksdienstenbeheer                          |
| Environment and waste                         | Milieu en afval  |
| Financial and internal services               | Financiële en interne dienstverlening                    |
| Ground affairs                                | Grondzaken   |
| Internal affairs                              | Interne zaken  |
| Management support and policy coordination    | Bestuursondersteuning en beleidscoördinatie              |
| Managerial service                            | Bestuursdienst   |
| Ministry of Economic Affairs                  | Ministerie van Economische Zaken                         |
| Ministry of Finance                           | Ministerie van Financiën                                 |
| Ministry of Interior and Kingdom Relations    | Ministerie van Binnenlandse zaken en Koninkrijksrelaties |
| Ministry of Social Affairs and Employment     | Ministerie van Sociale Zaken en Werkgelegenheid          |
| Municipal council                             | Gemeenteraad   |
| Operations support                            | Bedrijfsondersteuning                                    |
| Personnel, organization and information       | Personeel, organisatie en informatie                     |
| Public affairs                                | Burger- en algemene zaken                                |
| Public housing                                | Volkshuisvesting   |
| Public Works                                  | Openbare werken  |
| Roads, sewerage and vegetation                | Wegen, rioleringen en groen                              |
| Secretary                                     | (Gemeente)secretaris                                     |
| Service complementary employment              | Dienst complementaire werkgelegenheid                    |
| Service municipal development and maintenance | Dienst stedelijke ontwikkeling en beheer                 |
| Service social development                    | Dienst maatschappelijke ontwikkeling                     |
| Service social employment                     | Dienst sociale werkvoorziening                           |
| Space and general administration              | Ruimte en algemeen bestuur                               |

|  |   |
|--|---|
| Space, construction and accommodation                | Ruimte, bouwen en wonen                       |
| Spatial development                                  | Ruimtelijke ontwikkeling                      |
| Spatial regulation                                   | Ruimtelijke ordening                          |
| Team operational management, information and quality | Team bedrijfsvoering, informatie en kwaliteit |

---

# Appendix II.

## List of figures

---

|   |     |
|---|-----|
| Figure 1: research model.....   | 3   |
| Figure 2: our focus is concentrated on municipalities .....   | 4   |
| Figure 3: services integration in the context of eGovernment.....   | 4   |
| Figure 4: Overview of the used research material .....  | 5   |
| Figure 5: overview of the five stages of technical interoperability, or ESD-sophistication .....              | 13  |
| Figure 6: conceptual model of a technical interoperability framework [AGI05].....                             | 13  |
| Figure 7: $n(n-1)/2$ integration problem .....  | 15  |
| Figure 8: spaghetti architecture [WAL05] .....  | 15  |
| Figure 9: quality of municipal ESD [WEB05][CBS05] .....   | 16  |
| Figure 10: quality of municipal ESD [ADV05][CBS05] .....  | 16  |
| Figure 11: enterprise architectural framework suggested by Jonkers et al. [JON03].....                        | 18  |
| Figure 12: communication between architects and stakeholders using views and analysis [LAN05].....            | 18  |
| Figure 13: SOA's position in the application domain of the enterprise architecture framework.....             | 19  |
| Figure 14: the key elements of Service-Oriented Architecture .....  | 20  |
| Figure 15: tight coupling [BIJ05] .....   | 21  |
| Figure 16: loose coupling [BIJ05].....  | 21  |
| Figure 17: our interpretation of a service: a provider provides a certain function to a consumer.....         | 21  |
| Figure 18: an overview of a service with its key elements.....  | 22  |
| Figure 19: the addressed service types as linking pins between architectural layers [LAN05] .....             | 23  |
| Figure 20: the position and influence of SOA in the revised enterprise architecture framework.....            | 24  |
| Figure 21: global overview of service interaction using the service repository [MCI04] .....                  | 24  |
| Figure 22: overview of the service choreography <i>service</i> , based on [CHA04b] .....                      | 26  |
| Figure 23: the pub/sub mechanism as used in the service bus pattern.....                                      | 27  |
| Figure 24: UML Sequence diagram of itinerary-based routing using a service bus .....                          | 27  |
| Figure 25: the distributed service bus.....   | 28  |
| Figure 26: the ESB connects a collection of middleware capabilities .....                                     | 29  |
| Figure 27: the combination of available capabilities results in the "bus" .....                               | 29  |
| Figure 28: the ESB and its environment, Table 4 contains detailed capability information [IBM04] .....        | 30  |
| Figure 29: position of the ESB in the enterprise architecture framework .....                                 | 30  |
| Figure 30: a set of federated hubs is referred to as a bus [IBM04].....                                       | 32  |
| Figure 31: the relation between the concepts of hub, bus and ESB [IBM04].....                                 | 33  |
| Figure 32: SOA components and their capabilities in the technical interoperability framework .....            | 33  |
| Figure 33: the three steps of our suggested selection approach .....  | 35  |
| Figure 34: the first selection step: select a corresponding EAI-scenario .....                                | 38  |
| Figure 35: the second selection step: determine suitable ESB solution pattern .....                           | 40  |
| Figure 36: the third selection step: address relevant architecture decision issues .....                      | 41  |
| Figure 37: the BEST reference framework for enterprise system implementation processes.....                   | 49  |
| Figure 38: critical dimensions of ES-projects [BOD05] .....   | 51  |
| Figure 39: organization chart of the municipality of Enschede.....  | 53  |
| Figure 40: high level overview of involved processes and systems of scenario I in Enschede.....               | 55  |
| Figure 41: layered model of the envisioned solution in Enschede.....  | 57  |
| Figure 42: interaction diagram connecting the various services by means of an ESB .....                       | 58  |
| Figure 43: interaction diagram using external services describing scenario II .....                           | 59  |
| Figure 44: model of the interactive organization adopted in Voorst, also referred to as the flower model..... | 62  |
| Figure 45: high-level overview of the data warehouse's central position in Voorst [PET06].....                | 63  |
| Figure 46: layered model of the envisioned solution in Voorst.....  | 66  |
| Figure 47: organization chart of the municipality of Almelo.....  | 69  |
| Figure 48: the service Public affairs acts as front-office serving various channels .....                     | 72  |
| Figure 49: organization chart of the municipality of Groningen .....  | 75  |
| Figure 50: sketch of the disconnected situation in Groningen .....  | 76  |
| Figure 51: local ESB's and a municipal ESB as linking pin between the autonomous municipal services.....      | 79  |
| Figure 52: from need to potential solution .....  | 107 |
| Figure 53: position of the ESB related to the SOA and MDA .....   | 107 |
| Figure 54: overview of the MDA transformation process .....   | 108 |
| Figure 55: the architect as linking entity between MDA and SOA [ES05] .....                                   | 109 |
| Figure 56: MDA transformations in SOA context and the resulting traceability [ES05].....                      | 109 |
| Figure 57: interpretation of the application of MDA in a municipal context, based on [ALH03] .....            | 110 |

---

# Appendix III.

## List of tables

---







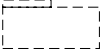
|  |    |
|--|----|
| Table 1: the superpilot cities, their targets and results .....  | 9  |
| Table 2: technical sophistication of ESD in European countries [CAP05] .....                           | 16 |
| Table 3: the distinguished views and related architectural models.....                                 | 19 |
| Table 4: the ESB capability model, based on [IBM04] (extended) .....                                   | 31 |
| Table 5: the minimum set of ESB capabilities [IBM04] .....   | 32 |
| Table 6: the distinguished EAI-scenarios, their ESB solution patterns and related issues [IBM04] ..... | 37 |
| Table 7: suggested ESB-capabilities for the municipality of Enschede .....                             | 56 |
| Table 8: scenario-based analysis of the envisioned EAI-project at the municipality of Enschede.....    | 59 |
| Table 9: suggested ESB-capabilities for the municipality of Voorst .....                               | 65 |
| Table 10: scenario-based analysis of the envisioned EAI-project at the municipality of Voorst.....     | 67 |
| Table 11: suggested ESB-capabilities for the municipality of Almelo .....                              | 71 |
| Table 12: scenario-based analysis of the envisioned EAI-project at the municipality of Almelo .....    | 73 |
| Table 13: suggested ESB-capabilities for the municipality of Groningen .....                           | 78 |
| Table 14: scenario-based analysis of the envisioned EAI-project at the municipality of Groningen ..... | 80 |
| Table 15: summarizing table highlighting interesting issues.....                                       | 85 |

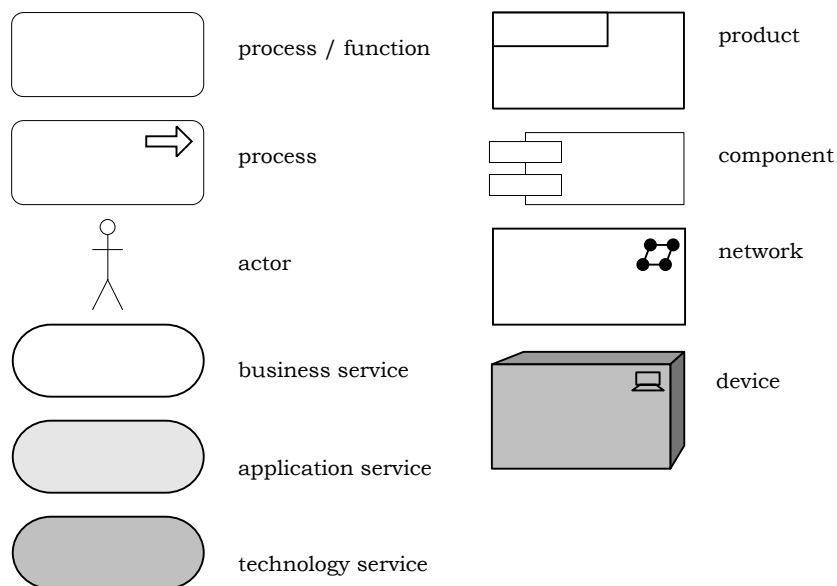
---

# Appendix IV.

## Graphical notation

---

|             |   |   |
|-------------|---|---|
| Association |    | Association models a relation between objects that is not covered by another, more specific relationship.   |
| Triggering  |    | The triggering relation describes the temporal or causal relations between behavioural elements, interactions, and events.  |
| Used by     |    | The used by relation models the use of services by processes, functions, or interactions and the access to interfaces by roles, components or collaborations.                                 |
| Realization |    | The realization relation links a logical entity with a more concrete entity that realizes it.   |
| Assignment  |    | The assignment relation links units of behavior with active elements (e.g. roles, components) that perform them, roles with actors that fulfill them or artifacts that are deployed on nodes. |
| Aggregation |  | The aggregation relation indicates that an object groups a number of other objects.   |
| Grouping    |  | The grouping relation indicates that objects belong together based on some common characteristic.   |



---

# Appendix V. Model-Driven Architecture

---

Model-driven software development, as adopted in the Model-Driven Architecture (MDA), provides a mechanism to (partially) transform business and system models into software. This enables the transformation of modeled business processes to services, followed by semi-automatic deployment on the ESB. When the number of deployed services grows, MDA is expected to support the maintenance and control of deployed services and the way they interact in the architecture.

The development process encapsulates the transition from problem to a solution provided by technology. Since MDA provides the design and (automated) generation of reusable components-based on business process models, this approach enables a high level of traceability and maintainability of the resulting technology. Figure 52 and Figure 53 depict the relation between SOA, MDA and ESB.

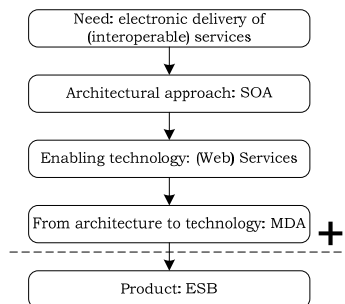


Figure 52: from need to potential solution

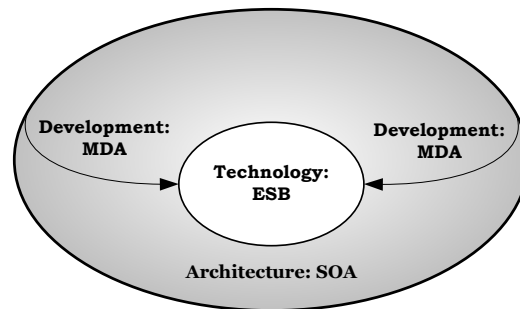


Figure 53: position of the ESB related to the SOA and MDA

Model-Driven Architecture is an evolutionary step in the approach towards software development. Ever since computers were available, they had to be programmed. Computer language is hard to interpret for humans, therefore specialized languages were developed to model computer functionality. This way the vision concerning software development abstracted more and more from the actual computer language at the level of bits and bytes. MDA envisions an approach where real-life requirements and events can be modeled in both computational and platform independent ways.

The abstraction from these aspects enables developers to think in real-life scenarios and solve problems in a way that is not affected by system specific limitations or requirements. Eventually the idea of MDA is to allow the definition of machine-readable applications and data models which allow long-term flexibility of [OMG03]:

- *implementation*, new implementation infrastructures can be integrated or targeted while using existing designs;
- *integration*, data integration bridges and new integration infrastructure connections can be generated based on the designs;
- *maintenance*, because the design is machine-readable, developers can directly adapt design specifications enabling maintenance flexibility;
- *testing and simulation*, since the developed models can be used to generate code, they can be used to (automatically) test and simulate designs as well.

The envisioned flexibility regarding implementation, integration, maintenance and testing should eventually enable portability, interoperability and reusability by

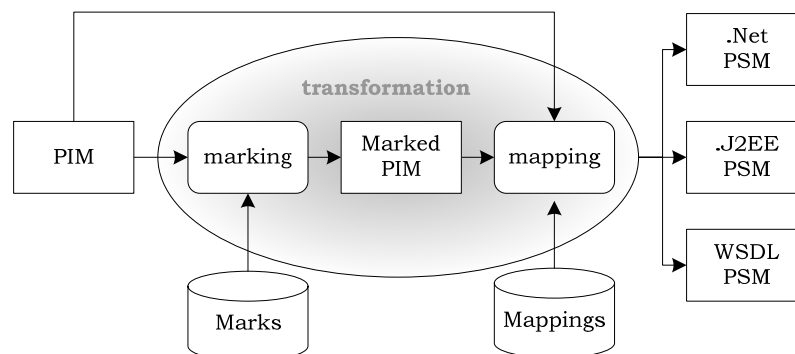
means of architectural separation of concerns [OMG03]. To realize this, MDA supports modeling based on the OMG Meta-Object Facility standard (MOF). This standard provides the abstract language for defining different kinds of metadata and enables an approach for:

- specifying a system independently of the platform that supports it;
- specifying platforms;
- choosing a particular platform for the system;
- transforming the platform-independent system specification into a platform specific one.

This approach to system development observes the specifications from three different viewpoints as introduced in section 3.1. Each viewpoint involves a perspective which focuses on specific aspects and abstracts from details and therefore results in a *simplified view*. The recognized views are represented by the following models:

- the Computation Independent Model (CIM) contains domain and requirements descriptions on the informational level. This information is acquired using the business viewpoint. Therefore these models do not contain any business logic or platform-specific data.
- the Platform Independent Model (PIM), contains the systems functionality specified in a platform independent manner. This model contains information concerning the system's data and functionality. It describes how the system operates by means of (business) process modeling. This model contains the part of the systems that does not change from one platform to another;
- the Platform Specific Model (PSM), contains the system functionality for a specific platform which is used for the actual implementation.

In order to convert the platform-neutral PIMs to platform-specific PSMs, a transformation process is required. This process is performed by means of so-called mappings and markings.



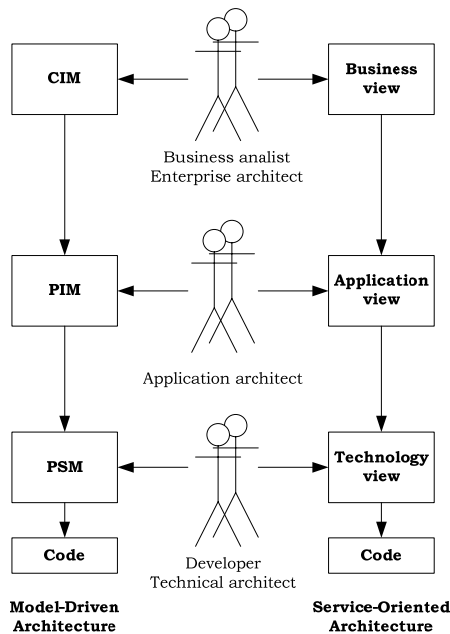
**Figure 54: overview of the MDA transformation process**

The mapping process is supported by a collection of mappings; these mappings are defined at metamodel level. This process transforms a (marked) PIM to a PSM. The used mapping determines which PSM is the result of the transformation process. The provided marks contain additional information on how certain PIM elements should be mapped.

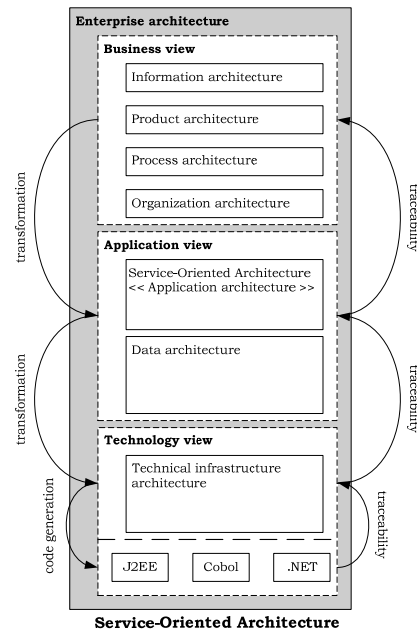
MDA uses three viewpoints. This makes it possible for system analysts and developers to focus on the implementation of the specified requirements at the right level of abstraction. This separation of (business) processes and technology can be related to the three views which are used to classify the models used in SOA: the business, application and technology view. The connecting entity in this relation is the architect. The architect adopts a view when he analyzes a problem area and develops a potential solution. The position of the architect between MDA and SOA, and MDA procedures in SOA-context are depicted in Figure 55 and Figure 56



respectively [ES05]. Figure 56 introduces the way how MDA enables traceability between the various levels of abstraction. This can be used to trace back applications or code to the business processes they belong to, providing insight in the actual use of applications and the business processes they account for. Eventually this enables an increase in maintainability, prevents complexity and can come in handy when an impact of change analysis is required. We have modeled SOA as part of the application architecture, as stated before, the *influence* of SOA goes beyond solely this architecture and affects both the process and organization domain as well.



**Figure 55: the architect as linking entity between MDA and SOA [ES05]**

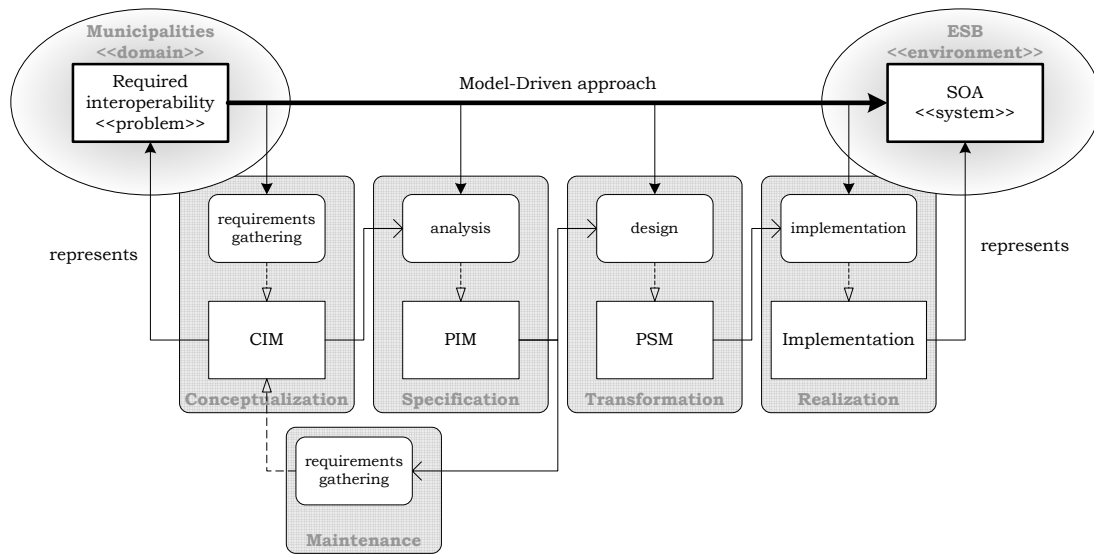


**Figure 56: MDA transformations in SOA context and the resulting traceability [ES05]**

When BPM and BPEL (as used by the service choreographer) is adopted in combination with MDA and SOA using an ESB, *theoretically* a situation can be achieved where:

1. problem domain and requirements are specified in the CIM by business analysts;
2. business processes and functionality are modeled (PIM) by application architects in cooperation with end-users;
3. the resulting models are transformed to platform-specific models (PSM) which are completed by developers, assisted by technical architects;
4. the resulting services are deployed on the ESB;
5. the deployed services are maintained by adjusting the models introduced at the first step. Underlying transformation and development tasks should be addressed as well.

Since eGovernment aims at a customer-oriented electronic service delivery, the focus shifts from “data processing” (providing information which should be transported to another desk or agency) to “process processing” (orchestrate the various sub-processes so that the customer has immediate result). This makes it possible to actually deliver services instead of forms and other static information electronically, whether this is G2C, G2B or G2G [FIN05]. Therefore the combination of BPM, BPEL, MDA and deployment of the resulting services on an ESB can be regarded as an effective approach to achieve SOA in terms of loosely-coupled software development and maintainability. This *interpretation* of how MDA can be applied is depicted in Figure 57.



**Figure 57: interpretation of the application of MDA in a municipal context, based on [ALH03]**

---

# Appendix VI.

## Interview questions

---

These questions were partially used during our interview sessions with representatives of the municipalities. Prior to the interviews, we studied published documents and information concerning the specific municipal situation. Based on this information we adapted the questions to suit the municipality in context. Additionally, during the interviews interesting discussions started. Therefore, not all discussed subjects are covered by the questions as listed in this Appendix.

### **Goals & strategy**

What are the objectives on the mid- and long-term concerning the improvement of delivery of service, possibly supported by ICT. Are these objectives defined and formalized in operational activities, or are they regarded as 'extra' projects.

### **Management**

- How does the top-level management, i.e. the municipal council, supports the concept of automation and further develop the eGovernment strategy? Is this an established issue in policy development which gains frequent attention?
- Is the mid- and lower management supported by top-level management or are they restricted in some way in performing their tasks to reach the defined targets.
- How are priorities determined, resources assigned, scheduled defined and monitoring procedures set up?
- To what extent does (structural) consultation with other departments take place?

### **Structure**

- Following which formal structure are the determined objectives supported and carried out? Is this solely through the ICT department, or is a 'task-force' used, involving the representatives of various departments?
- Is their a recent inventory or description of the entire architecture, both infrastructural, technical, information technical and organizational?
- Can you tell about the organizational component concerning the interaction between the front- and back-office? Are service level agreements or another formalized form of a service-contract used between involved parties?

### **Sociale issues**

- To what extent does the main workforce supports possible improvements as provided by IT?
- Can you describe how this situation was in the past? Which role has the municipal council in motivation public servants? Is their a top-level sponsor who motivates the organization and takes full responsibility?
- What is the common expectation of public servants for the future? Do they regard automation as a threat (citizens are able to communicate directly with back-end systems through the Internet), or to they expect IT to improve the quality of services towards the citizen. For example, automating the registration of changing a citizen's address en Enschede resulted in an increase of incoming registrations. Therefore, automation of this process did not result in a decrease in work, but an increased exactness of the register. Another example is the use of email as official communication channel, making it easier for citizens to contact the government with an increase of work as result.
- What activities are initiated to provide employees with proper information concerning the expected developments for the near-future? Is a centralized news-channel, for instance using the intranet, in operation to increase employee involvement in this area.

### **Scenario-specific questions**

Some information concerning the scenarios is published on the municipal website. These questions are used for each scenario to gain more insight in the required procedures and involved parties.

### **Structure**

#### *Organizational*

- Which departments or department-specific teams are involved in the scenario. In what way is support provided by IT, how does the front-office communicate with the back-office?
  - How is the intake of documents arranged?
  - Which team has end-responsibility of processing incoming requests?
- Is an incoming document scanned and processed digitally in the remainder of the process?

#### *Information technical / technical*

- Which IT-systems are involved in this scenario (databases, application servers, workflow management systems, planning systems, etc). Can you describe how these systems are connected with each other, the way these connections are secured, and the flexibility of these connections?
- How long are the involved systems already in use and do they function as required? Is communication with Web Services required or wished? Can they be adapted easily for this purpose? Can these systems be regarded as legacy systems?
- Is there a common understanding concerning the 'impact-of-change' of all systems. Suppose a change has to be made in a database table, can you point out which applications are affected by this change, on which term this is and what the consequences are?

### **Processes**

#### *Organizational*

- Are the required process-steps and responsibilities documented in process-descriptions?
- What is the status of formality of these descriptions?
- Can these descriptions be publicly accessed?
- Who is the end-responsible person?

#### *Information technical / technical*

- Is a workflow management system in use?
- Is the progress of a request traceable via the Internet?
- Can the progress be monitored by monitoring systems?
- Is (real-time) management information generated concerning the state of affairs?

### **Knowledge & skills**

#### *Organizational*

- Is specialized (human) knowledge required to process a request, or can a decision table be considered?
- What experience do you have with external parties such as consultants or (software) suppliers.
- Are sufficient (human) resources available to carry out future projects?

#### *Information technical / technical*

- How are existing applications and portals documented? How formal is the use of this documentation and are they available for involved employees?
- How is code reused? Is there a code-base which can be accessed by development and maintenance teams?
- What about the documentation of older (legacy?) systems? Is the knowledge required to operate and maintain these systems still available within the organization? Is this knowledge easy to communicate to other employees?