



DEGREE PROJECT IN INFORMATION TECHNOLOGY, SECOND CYCLE
STOCKHOLM, SWEDEN 2016

Dynamics AX in the Cloud

Possibilities and Shortcomings

DANIEL CARLSSON

Dynamics AX in the Cloud

Possibilities and Shortcomings

Daniel Carlsson

2016-10-03

Master's Thesis

Examiner
Gerald Q. Maguire Jr.

Academic adviser
Anders Västberg

Abstract

The usage of the cloud is rapidly increasing and is something that is of large interest to everyone involved in technology. The purpose of this thesis is to examine the benefits and possible shortcomings of using Microsoft Dynamics AX in the cloud, specifically Microsoft Azure, instead of using local datacenters. This thesis project has been done at Scania IT using their implementation of Dynamics AX.

This thesis project consists of an extensive literature study regarding both ERP Systems as well as other systems in regards to the cloud. It was decided early on to focus on the new version of Dynamics AX, which currently is only available in the cloud and compare this implementation to the two versions that the majority are using today, AX 2009 and AX 2012. The benefits of AX and Azure both being Microsoft products are clear with the well-designed integrations and support all the way through the clients to the servers regarding backups and load balancing. It is shown how the developers have to work differently in regards to integrations with outside systems, especially in regards to AX 2009 with the frameworks having changed. The addition of Data Entities mean that the developers can save a lot of time by only needing a reference to the location of the object in the database instead of having to keep track of all the tables themselves.

The analysis focuses on the differences in four different areas, performance & accessibility, scalability, cost savings as well as security & privacy. The background knowledge that is being used for the analysis primarily comes from the literature study as well as knowledge gained by studying the implementation at Scania today.

The result shows that there are clear advantages regarding performance, cost savings and especially accessibility, however it is also clear that laws in a lot of countries still have not caught up with the fact that it is possible to use the cloud for data storage these days. Which in turn means that the best move in the near future for the majority of ERP users would be either a hybrid or private cloud within the borders of the same country.

Keywords

Dynamics AX, Microsoft, ERP, Enterprise resource planning, Azure, Cloud

Sammanfattning

Användningen av molnet är snabbt expanderande och är något som är relevant för alla inblandade inom teknologin. Meningen med det här projektet är att undersöka fördelarna och de möjliga problem som kan uppstå genom användning av Microsoft Dynamics AX i molnet, specifikt Microsoft Azure, istället för lokala datacenter. Det här projektet har utförts hos Scania IT med hjälp av deras nuvarande implementation av Dynamics AX.

Arbetet innehåller en omfattande litteraturstudie angående både ERP system och andra system från varierande områden med fokus på molnet. Det beslutades tidigt att fokusera på den nya versionen av Dynamics AX, som för tillfället bara är tillgänglig i molnet, och jämföra denna implementationen med de två versioner som huvudsakligen används idag, AX 2009 och AX 2012. Fördelarna med både AX och Azure som Microsoft produkter är tydliga med välintegrerade hjälpmedel hela vägen från klienterna till serverna med ett särskilt fokus på säkerhetskopiering och lastbalansering. Det visas hur utvecklare behöver ändra sitt arbetssätt i avseende på integrationer med andra system, särskilt för AX 2009 då ramverken har ändrats. Tillägget av Data Entities betyder att utvecklare kan spara mycket tid på att bara behöva ha koll på en referens till platsen för ett objekt istället för att behöva veta exakt i vilken tabell i databasen objektet befinner sig.

Analysen fokuserar på skillnaderna inom fyra olika områden, prestanda & tillgänglighet, skalbarhet, kostnadsbesparingar samt säkerhet & integritet. Kunskapen för analysen kommer framförallt ifrån litteraturstudien samt den kunskap som har intagits från implementationen samt medarbetarna vid Scania idag.

Resultatet visar att det finns tydliga fördelar när det kommer till prestanda, kostnadsbesparingar och framför allt, tillgänglighet. Dock är det även tydligt att lagar i många länder ännu inte har hunnit ikapp det faktum att molnet är en av de bättre möjligheterna att spara data i idag. Detta betyder i sin tur betyder att det bästa nästa steget för majoriteten av ERP användarna idag är ett hybrid- eller privatmoln inom landsgränserna.

Nyckelord

Dynamics AX, Microsoft, ERP, Enterprise Resource Planning, Azure, Molnet

Acknowledgments

I would like to thank everyone at Scania for providing me with the opportunity for this thesis project and in particular Mikael Stenroth, for everything administrative as well as being a large help with setting up interviews with other employees at Scania, and Fredrik Håkansson for insight within Dynamics AX at Scania. I would also like to thank my examiner Gerald Q. Maguire Jr. for his work in examining me as well as providing some great feedback throughout the project.

Stockholm, September 2016
Daniel Carlsson

Table of contents

Abstract	i
Keywords.....	i
Sammanfattning	iii
Nyckelord.....	iii
Acknowledgments	v
Table of contents	vii
List of Figures	ix
List of Tables	xi
List of acronyms and abbreviations	xiii
1 Introduction	1
1.1 Background.....	1
1.2 Problem definition	1
1.3 Purpose	2
1.4 Goals	2
1.5 Research Methodology	2
1.6 Delimitations	2
1.7 Structure of the thesis	3
2 Background	5
2.1 ERP systems and the basics of Dynamics AX.....	5
2.1.1 ERP Systems.....	5
2.1.2 Cloud ERP	6
2.1.3 Dynamics AX History	7
2.1.4 Application Object Layers	9
2.2 Cloud computing basics and Microsoft Azure.....	10
2.2.1 Cloud computing.....	10
2.2.2 Microsoft Azure	12
2.3 Related work	12
2.3.1 ERP Systems.....	13
2.3.2 ERP Systems and the Cloud	13
2.3.3 Modular Architecture.....	14
2.3.4 Summary of related work	15
2.4 Summary	15
3 Methodology	17
3.1 Literature Study	17
3.2 Case Study	18
3.3 Quality Assurance	18
4 Dynamics AX Architecture Comparison	19
4.1 Microsoft Dynamics AX 2009.....	19
4.2 Microsoft Dynamics AX 2012.....	20
4.3 New Dynamics AX 7	21

5	Case Study	23
5.1	Deployment	23
5.2	Client	25
5.3	Authentication	25
5.4	Integrations	26
5.5	Customizations and Data Management.....	26
5.6	Summary	27
6	Analysis	29
6.1	Performance and Accessibility	29
6.2	Scalability.....	30
6.3	Cost Savings.....	30
6.4	Security and Privacy	31
7	Conclusions and Future work	33
7.1	Conclusions	33
7.2	Limitations	33
7.3	Future work	34
	References	35

List of Figures

Figure 2-1: Dynamics AX layered model.....	10
Figure 4-1: Simplified Architecture of Dynamics AX 2009 and Dynamics AX 2012.....	20
Figure 5-1: AX 7 Deployment on Azure	24
Figure 5-2: AOS and Batch Server Architecture	25
Figure 5-3: Data is managed with data entities in AX 7 (right side of figure) in comparison with multiple tables in Dynamics AX 2012 (left side of figure).....	27

List of Tables

Table 6-1: Dynamics AX 7 Pricing [53]..... 30

List of acronyms and abbreviations

AAD	Azure Active Directory
AIF	Application Integration Framework
AOS	Application Object Server
CRM	Customer Relationship Management
DIXF	Data Import/Export Framework
ERP	Enterprise Resource Planning
IIS	Internet Information Services
ISV	Independent Software Vendor
IT	Information Technology
LAN	Local Area Network
MRP	Material Requirements Planning
RPC	Remote Procedure Calls
SSAS	SQL Server Analysis Services
SSRS	SQL Server Reporting Services
VAR	Value Added Reseller
VM	Virtual Machine
WAN	Wide Area Network

1 Introduction

This chapter describes the area that this thesis focuses on, the actual problem, and the goal of the thesis project. The chapter concludes with an outline of the complete thesis.

The focus of this thesis is Microsoft Dynamics AX[1], an Enterprise Resource Planning (ERP) [2] system, and the advantages and possible shortcomings of moving this ERP system into the cloud instead of running it in a local datacenter. With more and more systems being moved into the cloud every day[3], it is important to know what the possible shortcomings of this might be and if the advantages exceed the shortcomings that exist. The main focus will be on interactions between Dynamics AX and integrated applications that are still running in the local data centers, while there will be a smaller focus on possible legal issues that may be relevant, with the primary focus being privacy and security issues, and a comparison of the costs for the implementation for (1) a completely local version of Dynamics AX 2009 or Dynamics AX 2012 and (2) the cloud based new Dynamics AX.

This thesis project has been done at Scania IT with the base for this work being their current implementations of Dynamics AX 2009 and Dynamics AX 2012.

1.1 Background

Microsoft Dynamics AX has been a Microsoft product since 2002 when it was called Axapta [4]. Microsoft starting integrated this product with the rest of the Microsoft product family after acquiring it and this integration is finally becoming reality [5]. The new version of Dynamics AX, called Microsoft Dynamics AX or Dynamics AX 7, was released on March 9th 2016 [6]. This new release is a complete redesign of the earlier Dynamics AX 2012 and is fully integrated with Microsoft Azure cloud infrastructure.

Cloud computing itself is becoming more important every day. However, this completely new version of Dynamics AX means that there is little experience with it, hence there is no way of knowing what problems may arise when upgrading to this new version from the older local versions of the system. This thesis will focus primarily on integration between Dynamics AX with other local applications and how this changes when Dynamics AX is running completely in the cloud.

1.2 Problem definition

Microsoft Dynamics AX is moving towards a becoming a complete service. However, little analysis has been done on cloud services for business solutions, hence many companies feel that by adopting cloud solutions they are moving into the unknown. The advantages of moving to the cloud are generally advertised by solution providers (in this case Microsoft). However, within Scania there are several different implementations of Dynamics still in use, specifically Dynamics AX 2009 and Dynamics AX 2012. It is currently unknown how easy it will be to take advantage of the advertised advantages and what work will be necessary when upgrading the existing implementations. Moreover, there is no research available about the disadvantages of running Dynamics AX in the cloud. While it should be obvious that Microsoft will not advertise these disadvantages, this means that companies thinking about using Dynamics AX will have to do their own research. The goal of this thesis project is to do this research and analysis.

1.3 Purpose

The purpose of this thesis project is to provide a company with an analysis of the advantages and disadvantages of running Dynamics AX in the cloud in comparison with a local version. The thesis project will look at the potential problems that may arise. The focus will be on the integration between Dynamics AX with other local applications (systems) that the company is currently using. Anyone who is currently using Dynamics AX together with a lot of locally integrated systems will also benefit from this thesis. The thesis should provide information that could be used when making a decision of whether to upgrade to a cloud based version of Dynamics AX or remain with a local version. In addition, if the decision is to migrate to the cloud, the thesis will enumerate what steps need to be taken to make the transition to the cloud as smooth as possible.

1.4 Goals

The goal of this thesis project is to provide an analysis of the benefits and possible shortcomings of using Microsoft Azure for Dynamics AX with a focus on integration with local applications. This goal has been divided into the following three sub-goals:

1. Identify the relevant differences between the architectures of the different Dynamics AX versions.
2. Identify the differences between integration with Dynamics AX running in the cloud and locally with regards to communication cost & complexity, and authentication & security.
3. Identify relevant legal and security issues that may arise when data is placed in the cloud rather than locally, specifically with regards to regional laws and privacy.

1.5 Research Methodology

The methodology that will be used in this work will be qualitative research with the main work being an extensive literature study followed by a case study of the implementation at Scania. This implementation will be analyzed qualitatively in order to understand what needs to be done in order to move to the cloud as well as the path to solve those problems that will arise. This research will use an inductive approach. A qualitative method was chosen because there are no easy ways to quantitatively measure a Dynamics AX implementation without setting up a realistic implementation yourself.

1.6 Delimitations

This project will mainly focus on theoretical implications of migrating from locally running Dynamics (in either version that is currently in use) to Dynamics AX running in the Microsoft Azure cloud. It will be a general look at these implications and will use Scania's current Dynamics implementations as a base to predict what problems could arise when migrating from a version of Dynamics AX running locally to a version running completely in the cloud. The emphasis on Scania's current implementation as the base could mean that for other Dynamics AX users this thesis will not be as relevant as their problems and issues differ due to the specifics of their implementations of local applications and their locally running Dynamics AX.

This thesis project will *not* implement a version of the new Dynamics AX and directly compare its performance to the performance of older versions of Dynamics AX as this was considered to be too costly. This decision was made external to this project.

1.7 Structure of the thesis

Chapter 2 presents relevant background information about Dynamics AX and the history of the product as well as relevant information regarding ERP systems and cloud in a more general sense. Chapter 3 presents the methodology and method used to analyze the available information and solve the actual problem. Chapter 4 presents the Dynamics architecture and compares the different versions of Dynamics AX. Chapter 5 presents what an implementation of Dynamics AX 7 in the cloud would look like. Chapter 6 presents the evaluation of the information gathered throughout the project. Chapter 7 presents the conclusions, and what future work remains to be done.

2 Background

This chapter provides basic background information about cloud computing and what an ERP system can do. The focus will be on Microsoft Azure as the cloud and Microsoft Dynamics AX as the ERP system.

2.1 ERP systems and the basics of Dynamics AX

This section gives some general background about ERP systems and how they have evolved to what they are today. This is followed by a summary of the basic differences between on-premise ERP and Cloud(-based) ERP systems. This is followed by a brief summary of the history of Dynamics AX itself and the layering architecture that makes Dynamics AX unique compared to all other ERP system. This section ends with a discussion of the application object layers that form Dynamics AX.

2.1.1 ERP Systems

The origins of ERP systems can be traced back to the 1940s with large calculating machines [7]. Actual systems that resemble the ERP systems used today were introduced in the 1960s when J.I. Case, a tractor manufacturer, and IBM developed one of the first Material Requirements Planning (MRP) systems[7]. This system was used to plan and schedule materials for the manufacturing of complex products. MRP systems in the early 1970s were large and clumsy and required a lot of maintenance, which meant a large staff just to oversee the system; but they became the best way to control production and by the late 1970s MRP was the most widely used method for companies to handle production management[8].

In the early 1980s MRP evolved into MRP-II, an extended version of the MRP system. While the original MRP systems ended when materials arrived for manufacturing, this new system kept track of production from manufacturing to the shipping dock [9].

The first ERP systems came in the early 1990s when MRP-II evolved to cover other areas, such as finance and human resources. The goal of an ERP system is to handle and integrate all the different areas of the company and consolidate all information in a single system. A properly set up ERP system should facilitate communication between the different areas and make it easier to monitor what is actually happening in the company [9].

The adoption of ERP started to become widespread in the late 1990s and early 2000s because of the millennium bug and the introduction of the euro as IT companies saw this as an opportunity to replace their older systems with an ERP system rather than having to do a lot of work to fix their older systems [10].

The first ERP systems focused on internal company problems that did not affect customers or the public. However, by the early 2000s, the second generation of ERP (ERP-II) included web-based software which made it possible for both employees and partners to access the ERP system. This led to the introduction of the Customer Relationship Management (CRM) as part of ERP[10].

ERP has continued to evolve. Today, the main focus of ERP is integrating ERP systems with as many devices as possible, especially mobile devices as well as integrating itself into Industry 4.0[11]

According to Microsoft, today's ERP system focus on six different areas [2]:

Financial Management	Everything financial in the company, for example, cash flow and accounting.
Supply Chain and Operations Management	Purchasing, manufacturing, and everything related to them should be as streamlined as possible.
CRM	Handles everything related to customer relations.
Project Management	Ensure that everything related to internal projects goes as smoothly as possible and results are delivered on time.
Human Resources Management	Tools to help with everything related to Human Resources.
Business Intelligence	Make it as easy as possible to report and analyze data.

There are two main ERP system architectures: two-tier and three-tier. These are each described below based on descriptions from eresource Infotech Pvt. Ltd. [12]:

Two-tier Architecture In a two-tier architecture the system is separated into servers and clients. Servers handle both the database and the application., while clients handle everything related to presenting the data, showing it to the user, and accepting user inputs, and sending this input to the server. There is theoretically no limit to the number of servers and clients that could be part of one system. The traffic in such systems are carried via wide area networks (WANs) and/or local area networks (LANs) and the processing responsibilities remain the same.

Three-tier Architecture A three-tier architecture separates the database from the application server. This is the most common architecture for larger ERP systems and is the way Dynamics AX is built. Communication is usually handled in these systems by the client establishing a connection to the application server followed by the application server establishing a connection to the database.

2.1.2 Cloud ERP

Cloud ERP provides the same functionality as on-premise solutions do; however, there are some general differences that apply no matter which ERP solution is used.

One basic difference is that an on-premise solution is installed locally on the company's own hardware and managed by the company's IT staff. In contrast in almost all cases, cloud ERP is provided as a service. One ERP system that is a bit different is Dynamics AX, as Microsoft provides both the software and the cloud infrastructure for AX, although the customer can implement their own modifications and integration of this ERP system with other systems that the company utilizes.

According to NetSuite Inc., an ERP provider, the four other main differences between cloud-based and on-premise ERP are [13]:

Upfront and Maintenance Costs	With an on-premise system, all the hardware and facilities are owned by the company. In contrast, in a cloud-based ERP all of these capital costs are paid by the ERP cloud service provider. For an on-premise system the company's staff maintain all the hardware and software, while it is up to the ERP provider to ensure that the system is always up and running and that all of the necessary security measures are in place. Furthermore, the user can easily scale the ERP system up and down depending on what they actually need to use at a given time, while always only paying for the resources that they actually use; unlike the case of on-premise solutions where they need to pay for all of the resources whether they are used or not. Sanjeev Aggarwal and Laurie McCabe have shown in their work that over time all of these differences add up, thus a company with 100 users can save 50% of their costs over a four year period by adopting a cloud-based solution [14].
System Upgrades	As the actual hardware and software implementation is managed by the ERP cloud service provider, deploying any update is a much simpler task and is not something that the customer has to worry about or even consider.
System Performance	Cloud ERP systems are usually designed to take advantage of maximum available network performance. As they are based in the cloud they can also easily be accessed from many different locations.
Deployment Speed	Since there are no actual hardware requirements and implementations to consider with a cloud ERP, a company can save a lot of time by deploy a cloud ERP solution in comparison with an on-premise solution. For example, NetSuite says that a cloud solution usually takes 3-6 months to deploy from scratch, while an on-premise deployment typically takes at least 12 months to deploy. This difference represents a lot of time being saved, which in turn saves a lot of money. More importantly, it enables the company to exploit the benefits of the ERP solution earlier.

2.1.3 Dynamics AX History

Erik and Preben Damgaard developed their own accounting system in 1983 called DANMAX [15]. This was the start of a long road to what is today called Dynamics AX. In the mid-1990s IBM bought 50% of what was then called Damgaard Software Company.

2.1.3.1 Axapta 1.0

The first version of the Damgaard brothers' ERP system was released in March 1998 in Denmark and USA and was called Axapta. At this point the company had evolved into a publicly traded company and was now called Damgaard A/S [15]. Some notable features of this first version were financial, inventory management, trade, and production functions [16].

2.1.3.2 *Axapta 1.5*

The first real upgrade to Axapta came out six months later as Axapta 1.5. This version was available in most of Europe and the USA. This version added Service Pack and support for Microsoft SQL Server 7.0 [17].

2.1.3.3 *Axapta 2.0*

The third major version was released in July 1999 and some of its notable features were the Project Accounting Module, Warehouse Management, ActiveX support, and an early version of the Axapta Object Server. The Axapta Object Server made it possible to offload some operations from the clients to a separate server in order to improve performance [16].

2.1.3.4 *Axapta 2.1*

In January 2000, a fourth and final version was released by Damgaard A/S as Axapta 2.1. The major new feature of this version was the Customer Self-Service Web tool, a precursor to the Enterprise Portal used today in Dynamics AX [18]. A later service pack update to Axapta 2.1 introduced the full version of the Axapta Object Server. This in turn made Axapta the first completely three-tier ERP system.

2.1.3.5 *Axapta 2.5*

In 2000, Damgaard merged with Navision. The new company was called Navision-Damgaard. Navision was another Danish company with an ERP system. The new company released their first Axapta version in December of 2000 and the main focus of this update was an environment for Web Application development. It also included functionality to handle banking and a project module as well as XML support (this XML support is used a lot in Dynamics AX to this day) [16].

A Market package for Axapta 2.5 was released 10 months later in France & Italy and this new application layer contained the CRM module and a Product Builder for both client and server sides [16].

2.1.3.6 *Microsoft Acquisition*

The company was renamed Navision A/S before it was acquired by Microsoft during July of 2002 [4]. Axapta was initially renamed as Microsoft Business Solutions Axapta. Microsoft separated the Navision and Damgaard ERP systems into one product (Axapta) focused on large companies and another product (Navision) focused more on medium sized and smaller companies. This is still the case today with Dynamics AX and Dynamics NAV being two completely different ERP systems, although both are fully supported by Microsoft [19].

2.1.3.7 *Axapta 3.0*

In October 2002 a major update was the introduction of Microsoft Axapta Enterprise Portal. This update featured a completely rebuilt user security system and system configuration. As mentioned earlier, Enterprise Portal is still used and is mainly a portal to web sites that give access to data stored within the system. This portal can be configured to display Role Centers, thus it can show the currently active user information relevant for their specific task(s). In Dynamics AX 2012 the portal has over 25 predefined Role Centers. The portal not only gives users access to pages that show available data, but also enables users to participate in predefined business processes using web-based forms [18].

2.1.3.8 Microsoft Dynamics AX 4.0

When it was released in March 2006, Microsoft Dynamics AX 4.0 brought a new updated look and feel, similar to a modern Microsoft program. As the first version that Microsoft was involved in from the start it also had much tighter integration with other Microsoft products and services, including a .Net business connector, a completely new framework to handle XML exchanges with other programs (the Application Integration Framework - AIF), and it made the Axapta Object Server a true Windows service [17].

2.1.3.9 Microsoft Dynamics AX 2009

When this version was released in June 2008 it featured an improved User Interface (UI), add role-based concepts for both the clients and the Enterprise Portal, and introduced a way to develop Enterprise Portal projects in Visual Studio [15]. Microsoft Dynamics AX 2009 is still widely used today. For example, it is used in multiple projects at Scania.

2.1.3.10 Microsoft Dynamics AX 2012

In August 2011, Microsoft Dynamics AX 2012 was released. Today it is the main version used. It has over 1000 improvements over Dynamics AX 2009, with some of the more notable improvements being tie-ins to Microsoft Office products (thus Word and Excel can be used to build templates) and it has the ability to create much more complex workflows that can be customized and some workflows can even be created automatically [20].

2.1.4 Application Object Layers

Dynamics AX is built with eight layers (shown in Figure 2-1) [21]:

USR	The USR layer is where the customer puts modifications that are specific for their installation or user.
CUS	The CUS layer is where a company or customer puts their own customer wide modifications.
VAR	The Value Added Reseller (VAR) layer is where a partner stores their modifications to their solutions that are in the ISV layer.
ISV (BUS)	The ISV (Independent Software Vendor) layer, or BUS layer as it was previously called, is where solutions created by partners for many of their customers are located.
SLN	The SLN layer is used by distributors of Dynamics AX to implement solutions from vertical partners.
FPK (HFX)	The FPK layer, previously called the HFX layer, is a Microsoft reserved layer for hot fixes and future patches. By using this layer developers do not have to risk compatibility issues whenever a small hot fix is deployed. When a full update is subsequently released this layer is emptied [22].
GLS	The GLS layer is combined with the SYS layer in Dynamics AX 2012. In this layer modifications are made to match specific legal demands for countries or regions.
SYS	The SYS layer is the standard and also the lowest layer. This is where Microsoft implements their standard code and the content can never be deleted.

The way all these layers work together is that the code in all these layers is available to every developer working in the system. However, when a developer signs into the CUS layer and makes a modification to the SYS code, the base Microsoft code remains in the

SYS layer - while the modified code is saved in the CUS layer. Whenever a user activates a method it will first look through all of the layers for this method, starting from the USR layer and working its way inwards. The first implementation of this method is the one that will be used, thus if no modifications have been made to this method it will use the code inside the SYS layer.

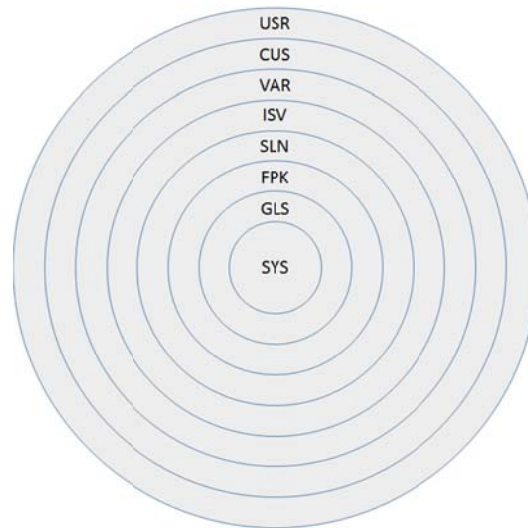


Figure 2-1: Dynamics AX layered model

Layer access is handled via keys supplied directly by Microsoft. The USR layer is the only layer that can be customizable without any key.

This method of storing code modifications has been a part of Dynamics AX since the early versions of Axapta and was one of the major reasons for Microsoft deciding to buy Axapta. One of the huge advantages of such a system is that there is very little risk to the system as developers can only edit the layer or layers that they have access to, while all lower levels remain unmodified [23].

All of these eight layers have their own patch layer. The reason for this is to make it easy and safe to update the system (as described above concerning the FPK layer).

Microsoft also provides a tool to compare the different layers. This tool is called "Compare Layers". It can be used to compare any two of the application object layers. This tool works by taking in a source layer and a reference layer, it then looks through the layers for objects that are either changed or only exists in the reference layer [24].

2.2 Cloud computing basics and Microsoft Azure

This section will introduce the concept of cloud computing, including some definitions of the different service models. The section ends with a look into what makes Azure different from all the other cloud providers that exist today.

2.2.1 Cloud computing

Cloud computing is a way of enabling access to a shared on-demand pool of configurable computing resources that can almost instantly be provisioned and released with minimal effort and interaction with the service provider [25].

According to the U.S. National Institute of Standards and Technology (NIST), the main characteristics of the cloud are [26]:

On-demand Services	A consumer can use as much computing power as required without human interaction with the cloud provider(s).
Broad Network Access	The cloud's resources are available over the network and can be accessed from a variety of platforms, such as mobile phones, tablets, or computers.
Resource Pooling	The computing resources of the cloud provider are dynamically assigned and reassigned to users according to the current demand. The cloud provider generally has no control over how exactly the resources are used, while the user has no direct control over the assignment of specific resources.
Rapid Elasticity	The cloud resources allocated to a given user can rapidly, sometimes automatically, be scaled upwards and downwards depending on the current demand. Therefore, for a given consumer, the resources (in most cases) appear to be unlimited.
Measured Service	Cloud systems automatically control and optimize the usage of resources by keeping track of the relevant resource usage (e.g., storage, processing, and bandwidth) by each service. Transparency is provided by monitoring and reporting resource usage to both the consumer and cloud provider.

Three service models are included in the NIST definition. These service models are:

Software as a Service (SaaS)	The consumer can use applications that are provided and running on a cloud infrastructure. These applications are accessible through either a program interface or a client interface, such as a web browser. The only settings the consumer may have access to are limited to user specific settings inside the application.
Platform as a Service (PaaS)	The platform provides a way for the consumer to deploy created or acquired applications that are created using the programming languages, libraries, services, and tools supported by the provider. The consumer has full control of the deployed applications, but cannot manage or control the actual cloud infrastructure (e.g. operating system, storage, or network).
Infrastructure as a Service (IaaS)	The consumer is provided with a platform to deploy and run arbitrary software on. The consumer has control over what operating system to run, how to use storage, and full control over the deployed applications. In some cases, the consumer may also have control of networking components, for example a firewall.

2.2.2 Microsoft Azure

Microsoft Azure is Microsoft's cloud platform. It consists of over 50 different services [27]. The more notable ones are the storage, computing, data management, identification, and business analytics.

Azure is available in over 140 countries with support for 10 different languages and 24 different currencies[28]. There are currently 26 different data centers all across the world with 8 more to come soon. The majority of services are not available across all data centers however but are in almost all cases available in another location of a reasonable distance so that the communication times are barely affected.

With Azure being a Microsoft product and Microsoft themselves moving toward cloud computing and subscription models within their larger products such as Microsoft Office, which had 1.2 billion active users in March 2016[29]. Being able to integrate those users seamlessly to other Microsoft products using their own cloud service is something no other cloud provider can say and is the largest advantages Azure has over any other cloud service.

2.3 Related work

There is a very limited amount of academic work done when it comes to Dynamics AX, especially recently with only 742 hits on Google Scholar since 2012 on the phrase "Dynamics AX". The majority of these hits are also not any deeper work into the system itself but tutorials on how to implement it or books that have the same goal as Microsoft's own TechNet documentation, teaching someone how to use the system.

The one academic work for Dynamics AX that I found to be useful was a bachelor thesis by Henrik Karlsson and Mattias Olsson [30] from Linköping University in 2013 looks at Dynamics AX 2012 and how it communicates with the Commerce Runtime [31] tool that can be integrated with Dynamics AX to setup a separate database for a store. Using this tool avoids the issue of having to communicate with the entire Dynamics AX application every time information about the stock is updated, either by a customer buying something or a delivery coming in. While this is not related to this thesis on a deeper level, it was helpful to read about how other people have tried to evaluate Dynamics AX and although their analysis is not very deep and it is hard to quantify the results. This thesis gave a base for how an AX analysis can be done.

With the Dynamics AX work being so limited a wider net has to be thrown which means ERP systems in general, applications being moved to the cloud as well as modular systems. Modular systems being relevant as the new AX 7 is using a modular architecture.

The search about ERP systems had to be narrowed down as just searching for "ERP systems" in google scholar returns over 13000 hits from 2016 alone. Through different searches and reading of abstracts a relevant reading list of four papers was decided on. The information retrieved from these papers will also be used in the analysis which will be done in Chapter 5.

When it comes to the cloud it was quickly realized from searches that it also had to be narrowed down to ERP systems in the cloud as well as migrating other systems to the cloud. There are still thousands of hits in Google Scholar on these subjects from this year alone and it was quickly found that this contained enough information that a broader search on cloud was not deemed necessary. The five papers selected were selected by narrowing down the search even more based on the information gathered from the results.

It was quite difficult to find papers that focused on a modular software architecture and what this actually means for a system. However, after multiple searches of many different phrases a reading list was selected. Each of these papers is summarized below with the title of the paper as the subsection's heading.

2.3.1 ERP Systems

Li Fang and Sylvia Patrecia in their paper "Critical Success Factors in ERP Implementation"[32] provide a good base to start from even though their paper is just over 10 years old. As the concept of ERP systems has not been rebuilt from the ground up, but instead has evolved with the addition of more functionality over the years. As a result, the basic concept has remained the same, therefore the information in this paper was considered to still be valid for this thesis. The paper looks at the strategic and tactical factors relevant when an ERP system is implemented. Additionally, from an empirical study they decide what the most important factors are for a successful implementation.

Stefan Koch and Kurt Mitteregger in "Linking customization of ERP systems to support effort: an empirical study"[33] focus on the work required to maintain a heavily customized ERP system and what this means for the company and their implementation. Considering the number of customizations currently in Scania's implementation of Dynamics AX this was considered an important factor to keep in mind. Their work shows that the actual development takes up about 80% of a developer's time, and customer specific adaptations, which would mean development that most likely cannot be reused, require about one eighth of the full effort. They show that the more customizations that are implemented the greater the support costs grow, as the number of users in the system does not matter nearly as much as the number of modules that are being used.

In "Enterprise Systems Upgrade Driver: A Technological, Organisational and Environmental Perspective"[34] Gerald Feldman, Hanifa Shah, Craig Chapman, and Ardavan Amini try to pinpoint the exact factors that makes an ERP user want to upgrade and also focuses on making it easier for a user to comprehend those factors that influences the decision to want to upgrade. This paper was relevant as it helped me understand the thought process behind the problem this thesis focuses on. They use a qualitative survey to gather data from 23 organizations, with most respondents having more than 6 years of experience. They show that different parts of a company have different agendas in regards to upgrades as well as the decision to upgrade being a very complex question that has no single right or wrong answer.

Lori Hogan and Greenbrier Companies are the authors of "Critical Success Factors for ERP Implementations"[35], this paper which has the same title as the paper described in Section 2.3.1. was selected as it is almost 10 years newer than the earlier paper with the same name and I thought that comparing the differences would be interesting, both as a way to see what has changed, but also as a way to understand how things might change in the future. What can be seen is that not much has changed and the most critical factors are still good management and communication between the different sections and employees.

2.3.2 ERP Systems and the Cloud

"A Comparative Analysis On Cloud and ERP"[36] by Amit Shukla and Sumit Verma is quite basic, but very relevant to this thesis. According to their research cloud ERP is the best solution for an implementation of ERP. However, as the paper is very basic and they

do no research in regards to the issues and appear to take most things written about cloud ERP at face value.

Dhiraj Jain Sr. and Yuvraj Shama in a recent paper called “Cloud Computing with ERP – A Push Business Towards Higher Efficiency”[37] present some results regarding the advantages of cloud ERP. These results are very relevant to this thesis. They showed that the adoption of Cloud ERP has had a positive impact on business as well as improved the efficiency of the ERP systems themselves.

The paper “Cloud Infrastructure for Higher Education: The Sullivan Experience”[38] by Emmanuel Udoh, Mohammad Khan, Michael Grosse, and Drew Arnette was selected since it looks into the advantages of the cloud in a more general sense for a larger institution as well as examining the challenges that occur when moving to the cloud. Their findings can probably be applied to an ERP implementation. They show that for their university the improvements by moving to the cloud were very large and this change would save the university a lot of money in the future.

The paper “ERP Evaluation in Cloud Computing Environment”[39] by Valdir Morales, Oduvaldo Vendrametto, Samuel Dereste dos Santos, Vanessa Santos Lessa, and Edivaldo Antonio Sartor was taken from the book *Advances in Production Management Systems: Innovative Production Management Towards Sustainable Growth*. This paper looks into the advantages and disadvantages of ERP in the cloud by using a survey to gather data from various sources. Their findings can most be applied to Dynamics AX. Their data show that the majority of companies either plan on moving into the cloud in the near future or they have no plans at all (with 17% of responding companies having no idea and the remaining 27% being somewhere in between). They show that a company should look into why they want to move to the cloud and see if they actually benefit from such as change.

Tong-Ming Lim, Angela Siew-Hoong Lee, and Mun-Keong Ya in their paper “User Acceptance of SaaS ERP Considering Perceived Risk, System Performance and Cost”[40] focus on the user and what is required by a Cloud ERP implementation in order for the users to accept it as an improvement over an on-premises implementation. They present six open ended questions regarding moving to the cloud that need to be investigated in order to get a reliable answer as to whether moving to the cloud is a benefit or not. The major issues they identify are risks, hidden costs, such as training, and system performance.

2.3.3 Modular Architecture

The book *Rich Client Programming: Plugging into the NetBeans Platform* [41] written by Geertjan Wielenga, Jaroslav Tulach, and Tim Boudreau for Sun Microsystems has two chapters about modular programming and architecture. They go through modular programming on a basic level and describe the general advantages of this kind of architecture over other architectures. One of the main reasons modular architectures have become such a large concept these days is that most systems are built by a lot of people, sometimes from all across the world, and being able to take all these separately built parts and combine them into a system is a huge advantage over the entire codebase being lumped together because of easier bug fixing and less dependencies on other people working with the same code.

Manoj Nambiar, Ajay Kattepur, Gopal Bhaskaran, Rekha Singhal, and Subhasri Duttagupta in “Model Driven Software Performance Engineering”[42] focused on why performance modeling is not used as much within the software industry as within other industries. This is relevant as being able to model and measure performance improvements

and weaknesses is an important part of deciding upon upgrading a system. They analyze current limitations and shortcomings and provide multiple potential solutions for how to improve performance modeling in the future.

The paper “Modularity’s impact on the quality and productivity of embedded software development”[43] by Hongyi Sun, Waileung Ha, Min Xie and Jianglin Huang was selected as it was important to get some real world data. They collect data from 30 different projects inside a company with varying degrees of modularity. Their results show that quality, productivity, and cost performance are all significantly better with higher modularity, while time to market was unchanged. Their paper shows that updates in Microsoft in AX 7 should be of higher quality, but possibly not arrive quicker than in AX 2012.

Mohammad Mahdi Hassan, Wasif Afzal, Birgitta Lindström, Syed Muhammad Ali Shah, Sten F. Andler, and Martin Blom in “Testability and software performance”[44] focused on the importance of being able to test a system. This paper was selected as a modular architecture should generally make this testing easier.

M.S. Essers and T.H.J. Vaneker in “Design of a decentralized modular architecture for flexible and extensible production systems”[45] focused on the manufacturing industry, but the paper was selected as a way to compare a modular approach within a software system as well as within a large scale production system consisting of multiple physical objects. In the end their findings were not that relevant to this thesis, although their paper was a very interesting paper to read.

2.3.4 Summary of related work

As previously mentioned the information gathered from these books and articles will be used for the analysis in Chapter 5 together with all of the other information gathered.

There was a lot of information gathered from Microsoft themselves regarding AX and AX implementations. This information was taken from the Dynamics AX pages on Microsoft TechNet, as well as the multiple books regarding Dynamics AX found in the KTH Library’s databases.

There were other papers that appeared to be interesting or relevant, but were unavailable for reading for various reasons. However, the above selected papers together provided a base that was sufficient for this thesis.

2.4 Summary

This chapter provided the background and specific concepts of Dynamics AX, including its history and how it became the product it is today. The chapter also described what the cloud actually is. The third section of this chapter summarized some of the papers read as part of my literature study.

3 Methodology

Since there has not been much recent academic work done on Dynamics AX as already mentioned in Chapter 2 and even less when it comes to how it works in the cloud this is a great subject for this time considering the quick evolution when it comes to cloud services. This thesis is based on qualitative research which means that no experiments or tests were done, instead there has been an extensive literature study for Chapter 2 and Chapter 4. The main source of direct knowledge was a case study at Scania (see Chapter 5). The thesis uses the literature studies together with the case study and the knowledge gained from these two processes to evaluate the current environment at Scania for Dynamics AX and come up with suggestions to both Scania and in a more general sense on how a company should proceed with regards to using the cloud for their ERP system, with the focus being on Dynamics AX.

The work started with gaining a basic understanding of what an ERP system is and what is unique about Dynamics AX. This was followed by gaining some understanding of cloud computing and the general advantages and disadvantages of moving to the cloud. An extensive literature study was done on the architectural differences between the Dynamics AX versions. In parallel to this a case study at Scania was done with the main focus being on how much work, specifically what work, would have to be done in order to move their current Dynamics AX implementation to the cloud.

Based upon the literature studies and case study an analysis of the information gathered was performed. These results and an evaluation of these results, were used to draw the conclusions of this thesis (see Chapter 6).

3.1 Literature Study

The literature study was done by initially using the KTH Library and Google to figure out the best way to gain a basic understanding of Dynamics AX. The lack of academic research done on this topic was quickly noticed, this led in turn to collect a lot of data from Microsoft's own documentation and books about Dynamics AX. However, these books were mainly written by Microsoft employees so it is important to keep in mind what sections of the books are facts and what are the opinions of the author as those can be affected by the bias of working for the company.

With almost all the data on the product being collected from the developer itself and because this a product that is for sale, it was extremely important to filter out opinions and marketing/sales language to reveal the actual facts. Microsoft, as well as every other company, will of course write their documents in such a way that their product sound as good as possible, while trying (as much as possible) to avoid mentioning the downsides and problems. For an academic thesis one cannot take everything that was said by the vendor (and the vendor's employees) for granted, but one must actually think about the material critically and understand it on a deeper level.

All of the research was done in an empirical way with an inductive approach[26][27]. Considering the majority of the work was finding and evaluating information this approach seemed to be the best. The work was done without any previous theories and using all the collected data served as the basis for the evaluation. This collected data is used as the empirical evidence for the proposed goals in chapter 1.4.

3.2 Case Study

The case study was done together with Daniel Snellman, who was doing his thesis at Scania IT in Södertälje, during the same time as this work was being done. His thesis project being called “Difference in Cloud ERP Systems – A comparison”.

The main goal of the case study was to understand the specific problems an upgrade from a local ERP to a cloud based ERP could produce in the context of Scania’s ERP systems. The study was split into technical and infrastructural problems. The infrastructural problems were the main focus of this thesis. The technical problems were to be addressed in Daniel’s thesis. What was quickly realized however was that it made much more sense to do the case study together and then use the data relevant for your specific thesis. The result of this case study will be presented in Chapter 5. The case study itself consisted of multiple steps:

Step 1 investigated the current implementation of Dynamics AX, what modules are used, and if there is a large number of customizations currently used.

Step 2 combined this information, excluding the customizations, with the information learned during the literature study about the different architectures in order to present a theoretically optimal path on how and when to move to the cloud.

Step 3 looked at current customizations in order to know what issues are likely to arise and to see if there is a way to solve these issues and how to solve them.

Step 4 uses the information gathered in steps 2 and 3 to propose a realistic path for moving from a local to a cloud based ERP system - based on the current implementation and known information about expected future updates of both types of ERP systems.

The information was mainly gathered in the cases study by conducting interviews with employees who either directly work with Dynamics AX or work with cloud computing in more general terms and then comparing this information with the data collected from the literature study. Additionally, some was data gathered by visiting Microsoft where an interview was conducted with Pepijn Richter, who at the time was Director of Product Marketing for Dynamics AX at Microsoft. This interview was mainly used to confirm some assumptions that were gathered from the literature study and interviews. As Pepijn Richter worked in marketing it was important to consider the reliability and bias in this information in the same way as was presented for the information from the literature study (see the previous section).

3.3 Quality Assurance

Considering the sources of the information that this thesis is based on it is important to note that the documentation and information about future plans are only valid at the time this data was collected for this thesis. There is a high probability that if this same work were repeated in three years that the results and analysis would be quite different.

Another notable aspect is the use of a qualitative analysis based on what I learned during the project - there is a high probability, probably even certainty, that if someone else had done the work that they would propose a different path to reach the same goal.

4 Dynamics AX Architecture Comparison

This chapter compares the architectural of the three different versions of Dynamics AX that are commonly used today. The two older versions (Dynamics AX 2009 and Dynamics AX 2012) were previously described. The new Dynamics AX 7 architecture is similar to Dynamics AX 2012 in terms of its underlying architecture, thus it is very easy to upgrade to Dynamics AX 7 as long as there few customized implementations of Dynamics AX 2012. The four base components that have to be installed in order to get AX running is an Application Object Server (AOS, previously called Axapta Object Server), application files, a database, either a Microsoft SQL database or an Oracle database, and at least one client. This changes a bit with AX 7 as it is cloud based and the client is HTML5 based in a web browser.

Dynamics AX 2009 is not receiving any more updates, but it will have extended support from Microsoft until December 10th 2021. This means that a lot of companies and people using AX 2009 today will see no reason to upgrade for the foreseeable future. This is why this earlier version is included in this part of the report. However, the main focus for the thesis following Section 4.1 will be on AX 2012 and AX 7.

4.1 Microsoft Dynamics AX 2009

Figure 4-1 shows the underlying architecture of a Dynamics AX 2009 implementation. In the top right corner is the client that communicates with the AOS using remote procedure calls (RPC). AOS handles all the requests to the database. It is also possible to use a browser to connect to the client using the Enterprise Portal; however, this is limited in the sense that no actual development can be done. In AX 2009, AIF handles external systems as shown on the left side of figure 4-1. The middle section of the figure shows that Microsoft's Internet Information Services (IIS) can also be configured to work with AX 2009.

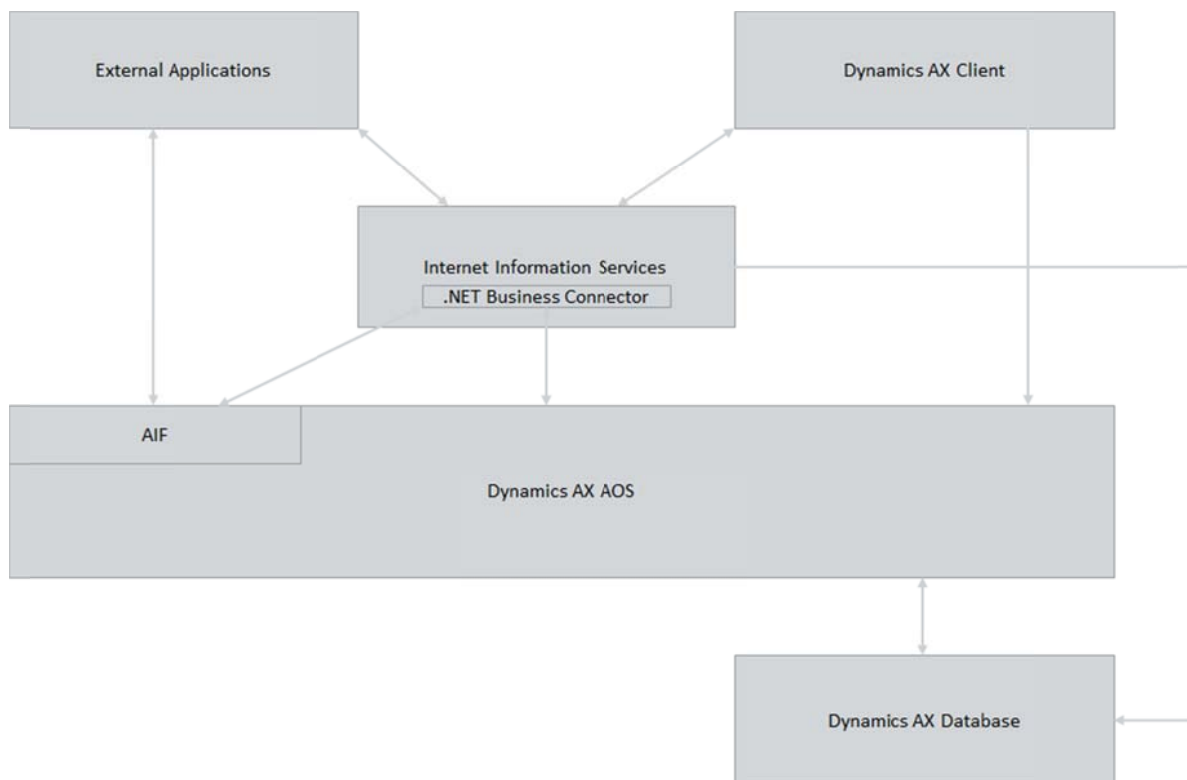


Figure 4-1: Simplified Architecture of Dynamics AX 2009 and Dynamics AX 2012

4.2 Microsoft Dynamics AX 2012

The basic architecture of AX 2012 is roughly the same as for AX 2009, in that the client still communicates with AOS and AOS handles the communication with the database server. A large change that could affect people running AX 2009 is that from 2012 onwards the only compatible database server is an SQL server. Additionally, Visual Studio can be used as a development tool in AX 2012. Another new feature is that it is possible to isolate data from parts of a company. Consider a larger company with 4 different subsidiaries that all use the same implementation of AX, then it might occur that some data should only be visible to some of the subsidiaries, in this case data partitioning can be used to restrict the data's visibility. This is a new feature that was unavailable in AX 2009, because this feature possibly affects a lot of data and database tables it is important that this data partitioning is carefully planned as it would be very complicated to change this partitioning later.

The most important new feature relevant to this work is the introduction of a Data Import/Export Framework as a replacement for AIF. What this means for AX 7 and the future will be described in Chapter 5.4 about Integrations.

4.3 New Dynamics AX 7

A simple way to describe the AX 7 architecture would be “Dynamics AX 2012 remade for the cloud”. While this is true, for system administrators and companies thinking about upgrading to Dynamics AX 7 in reality it is a bit more complicated – as Figure 4- shows that there are differences at every level of the AX implementation.

At the lowest layer, identification is done using Azure’s Active Directory over the regular Active Directory. What this means for the end user and system administrators is mentioned in Chapter 5.

At the data layer, all the data is stored in an Azure SQL server in the cloud rather than an on-premise SQL Server. Additionally, there is Power BI - an analytics tool. Power BI is an interactive visualization tool for business analytics that can be said to be an extension of SQL Server Analysis Services (SSAS). In both architectures there is SQL Server Reporting Services (SSRS), but in AX7 financial SSRS reports are replaced by Management Reporter capabilities and default reports –simplifying configuration. A new feature in AX 7 is the possibility to pin Power BI reports into a workspace and then launched and run within the client just as an SSRS report[46].

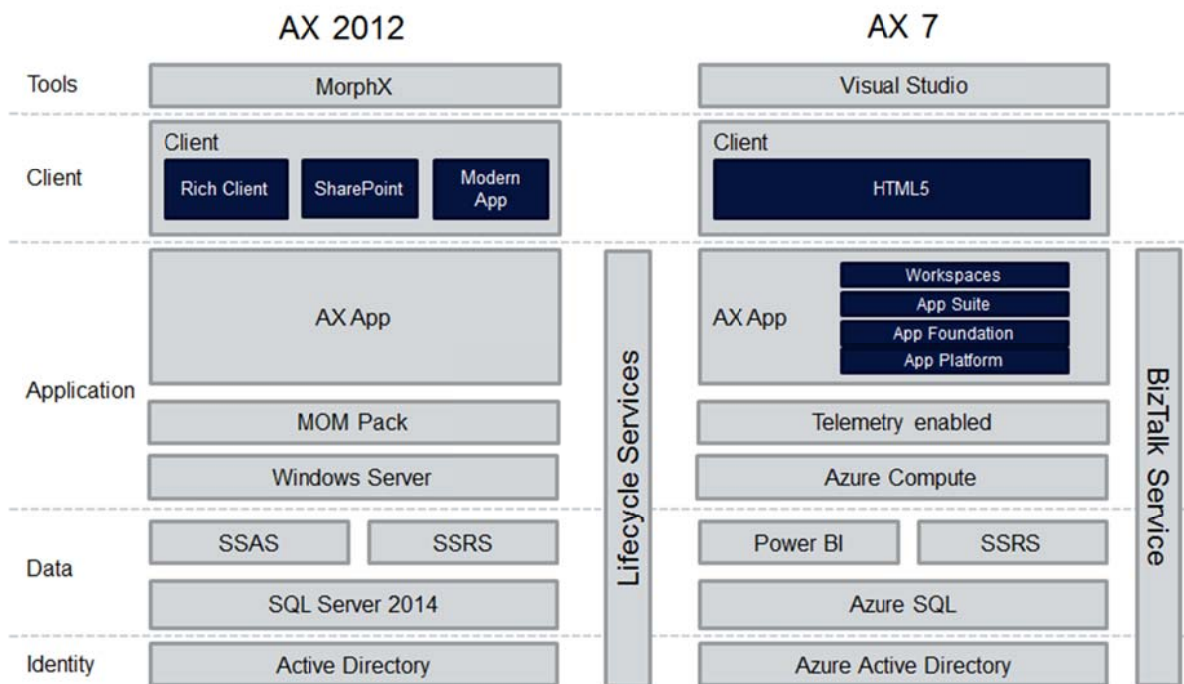


Figure 4-2: AX 2012 and AX 7 architecture comparisons

The application itself is where the majority of the major changes have taken place. Azure Compute replaced Windows Server as the computing platform, with both Azure Compute and Azure SQL considered as two separate parts of Azure. This could be a problem for systems that require a 100% uptime since it is twice as likely that one of them will have a problem causing it to be unavailable for a short in comparison with the probability if only one were used. It is also worth mentioning that Azure as a cloud service providers had the most downtime (almost 40 hours) of all the large cloud service providers in 2014 [28]. However, Azure availability improved a lot in 2015 and the total downtime was only at approximately 11 hours [29]. For most implementations 11 hours per year (~0.13% - equivalent to an availability of 99.87%) is acceptable and not something that needs to be considered as a problem for most companies.

There are also a lot of different telemetry tools implemented in AX 7. These tools will show an administrator how, from where, and when users are using the system. This is important to be able to decide on a roadmap for dimensioning the system for the future.

The greatest change (other than the move to the cloud) is that the application itself is now split into different modules. The core of the application is included in the App Platform, while the App Foundation is where the different frameworks and tools that are not necessary to run the actual system - but are still very useful for providing functionality.

App Suite can be considered as the SYS layer in AX, as it is the functionality that is visible and possible to change by the end user.

AX 7 also has functionality for modifying workspaces. These workspaces are all stored in the workspaces module which is at the very top layer of the application layer.

Both the new version of Lifecycle Services and the BizTalk services are integrated across everything mentioned so far (as shown in the figure). BizTalk is an integration platform from Microsoft that helps companies automate processes within their business [30]. Lifecycle Services has been completely redone for AX 7 and now include many quality of life services for both system administrators and developers for an implementation. These services are all focused on making the code and implementation faster and better.

The client layer itself is now HTML5 based and can be used from any device with an HTML5 capable web browser. Details on the new client layer will be given in Chapter 5.

All development has moving away from MorphX inside the AX 2012 client to Visual Studio. While all of the old functionality is still there it is now completely integrated with all of the different features that Visual Studio provides for development.

5 Case Study

This chapter analyses data from the case study that was done based on the implementation of Dynamics AX at Scania IT. It mainly focuses on how an implementation of Dynamics AX 7 looks in the cloud and the differences from one or more of the previous on-premise versions of Dynamic AX. Based on the information gathered in the literature study and the architectural comparisons presented in the previous chapter, it was decided that this analysis of the case study should focus on what should be done in order to upgrade from the current Dynamics AX 2012 version to the new Dynamics AX 7. Note that Dynamics AX 7 is the only (fully supported) cloud based version of Dynamics AX (at this point in time) and the cloud deployments available for AX 2012 are much less detailed and according to Microsoft offer much fewer possibilities.

It was quickly realized, when comparing the implementation at Scania with everything learned in the literature study, that their implementation is very similar to the majority of implementations of Dynamics AX 2012 that are in use today. This because the core of AX comprises the majority of the code, while customizations are only implemented when required. In the case of Scania, the majority of these customizations are because of integrations with other systems in use at Scania. This means that the result of this thesis should be relevant for many companies (other than Scania) because the same thoughts will apply in a majority of the Dynamics AX deployments.

The main usage of Dynamics AX at Scania (as a vehicle manufacturer and support organization*) is to support financial transactions, invoices, and orders. This requires a large amount of integration with other enterprise systems, such as billing systems and order systems, both internal and external to Scania. As previously mentioned the main focus of this thesis project is what needs to be done to upgrade to the cloud, specifically Dynamics AX 7 in as smooth a manner as possible. Therefore, we will focus on the deployment differences and the client differences between the on-premise Dynamics AX 2012 and a fully cloud based Dynamics AX 7.

5.1 Deployment

Figure 5-1 shows how a typical deployment of AX 7 in the cloud. Every virtual machine that has a version of AX running on it also has its own AOS, the advantages of this as well as the changes to the AOS will be described later in this section.

There are a lot of advantages to an on-premise deployment that are only possible to achieve in this cloud deployment because Microsoft is the developer of both Dynamics AX and the owner (and operator) of Azure. As a result, Microsoft can ensure that the system and the cloud are integrated in many ways that would be much harder or even impossible if a third party cloud service provider was being used.

* Note that Scania is a very customer focused business with a desire to create tailor made solutions by using modular subsystems. Scania is also increasingly focused on how to exploit connectivity with their vehicles to provide high performance and high efficiency, while providing their customers with full-service.

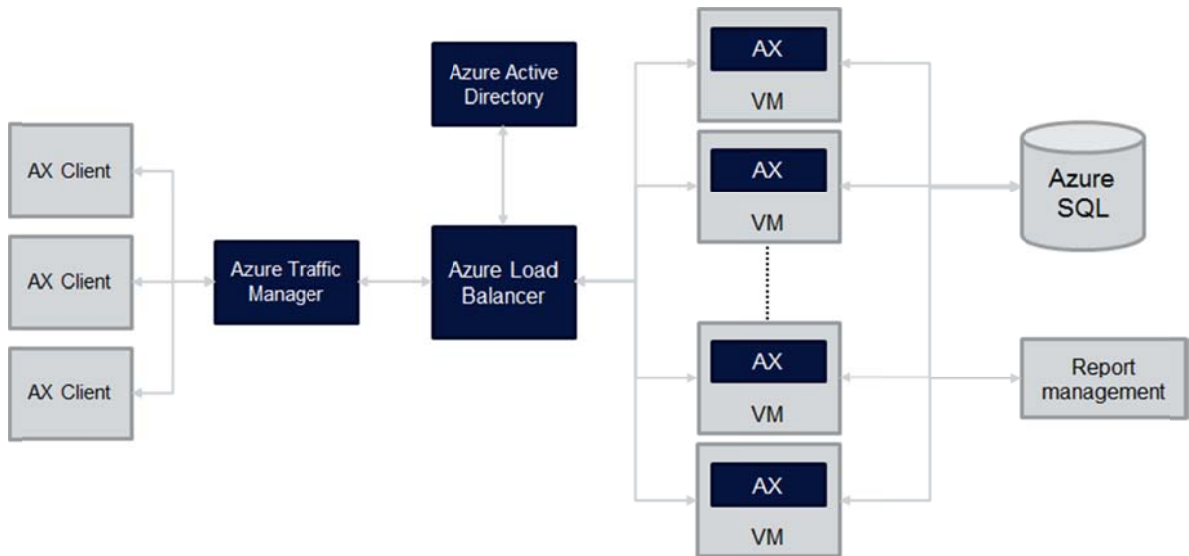


Figure 5-1: AX 7 Deployment on Azure

As can be seen in the figure, the clients (any device with a web browser) connect via a traffic manager to the main datacenter that is running Dynamics AX. However, to provide increased reliability and availability, the traffic manager also connects to a recovery datacenter to which it can route traffic if the main datacenter is down or unavailable for some reason. The disaster recovery system of Azure is very helpful here and this is a huge advantage of running Dynamics AX 7 in a cloud. All of the VMs as well as the database itself are backed up in the recovery datacenter. The database backup uses active geo replication[47], which Microsoft guarantees will only lose *at most* 5 seconds of data in the event of a disaster. Snapshots are taken of the VMs every 30 minutes. These snapshots can be restored on request. The reason for this is that in most cases, you would just spin up a new VM, rather than restore the old one backed up; as a result, it would typically be a waste of money to keep live backups of the VMs. This choice trades off some loss of data and re-computation against the increased cost for live backups of the running VMs.

Exploiting a load balancer is another major advantage of the cloud. This load balancer directs requests from the clients to the different VMs. The load balancer keeps track of which forms are used on which VMs, thus it can send all the requests for a given form to the same VM – thus saving a lot of time since the form will be cached in this VM, hence requiring much less delay to display the form or to perform computations on the cached data.

In AX 7, AOS is also greatly changed as it has separated batch jobs, which run at a set interval, to a separate part of the VM. Moreover, every AOS now includes both a regular AOS and a batch server. This ensures that the regular AOS is not overloaded by many batch jobs and it ensures that the batch jobs are always able to run at their set time intervals. Figure 5-2 shows the architecture of this server. The figure also shows how this batch server is nearly an AOS in itself. Moreover, those parts of AOS that are irrelevant to batch jobs have been removed.

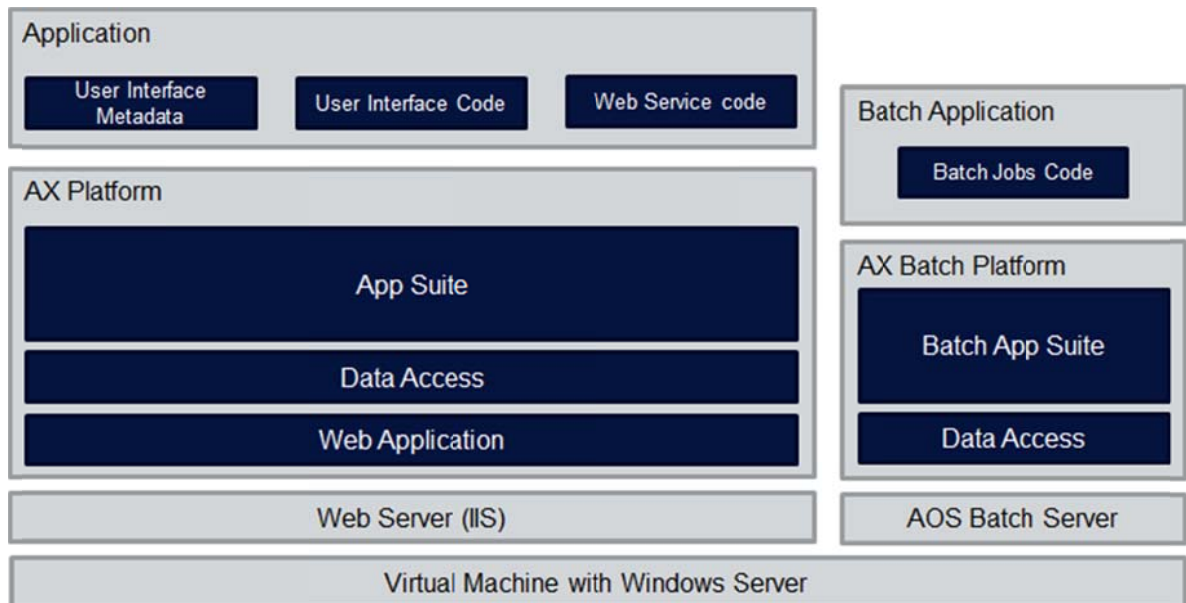


Figure 5-2: AOS and Batch Server Architecture

5.2 Client

In AX 7, the client has also changed a lot. It is now a completely HTML5 web based client that can be accessed from any device at any network location using a HTML5 capable web browser. This is a huge change since the previous client did many of the calculations itself, thus requiring it to communicate quite a lot with the database and Application Object Server. This communication lead to longer response times, which in turn lead to a poor user experience as a lot of time is spent waiting for the computation to complete. Whenever a user requested to view a form of any kind the previous client would send a message to the Application Object Server, which in turn queried the database and returns the data. This had to be done for any information included in the form, leading to many redundant calls and unnecessarily increased the response time. In AX 7, all computations occur on the same hardware as the Application Object Server, thus when a request from the client arrives, the complete computation is performed and all the relevant information in the form is sent back at once. For implementations with many different forms and a lot of large forms this change saves a huge amount of time and is a major benefit of AX 7.

5.3 Authentication

An essential function of any ERP system is to ensure that users can only see what they are supposed to see. This means that users might authenticate themselves when they sign-in and all of the communication must use appropriate security measures (especially when communicating over any network outside of the company's physical premises). AX 7 uses the Azure Active Directory (AAD) as its authentication tool, unlike previous versions of AX which used the regular Active Directory. This is unlikely to be a major change or even a problem for anyone looking to upgrade to AX 7, but it does require regular synchronizations between the local Active Directory and AAD to ensure the valid users exist in both directories. There is a tool in AAD that does this, but it may require some modifications to get it working if one is using a local active directory configuration that is non-standard[27]. Scania is using a standard configuration and therefore this would not be needed but it is a good thing to keep in mind.

There are two different kinds of authentication used in AX 7, one for users and another for applications. In AX 7, applications can sign-in on behalf of a user, such an application should only have access to the data that this user would have access to. This requires a more complicated process than the regular user sign in. As the details on authentications are not really relevant to this thesis we will simply mentioned that the regular sign-in uses SAML 2.0, while application sign-in uses OAuth 2.0[48][49]. Both of these standards are common and well known to be reliable.

5.4 Integrations

The most important part of the case study is the importance of integrations, as these are used to send/receive data to/from the other systems used at Scania. Moreover, these integrations account for the majority of customizations of Dynamics AX at Scania. At Scania, integrations are used in many different ways, but the main uses are starting some sort of order upon receiving an order request from another system. This processing is handled by running a batch job. If this job finds that a new order has been sent it then initiates a larger chain of events within Dynamics AX. Integrations can be both inbound and outbound, but the concept is similar in both directions.

In Dynamics AX 2012 there is a built in framework called the Application Integration Framework (AIF) that handles all incoming and outgoing files. However, this framework has been deprecated in AX 7 because Microsoft could not get bulk importing to scale well in the cloud[50]. The new way to handle integrations, which also is possible in AX 2012 and is used in newer projects, is to use the Data Import/Export Framework (DIXF). One of the major problems with the AIF is that it has no way of handling multiple files. This problem occurs because there is no queuing system of any kind, this means that there has to be some sort of system connected to the AIF that accepts all of the files, transforms the contents of the XML file (if required) and then forwards the file to the AIF. DIXF can do all of this by itself which could lead to a cheaper implementation. Furthermore, if this feature is used the result is a simpler implementation with fewer parts that can break or fail.

AIF works by taking in a file and decoding the XML, followed by forwarding the information to a separate framework consisting of staging table. Staging tables in Dynamics AX are temporary tables used when taking in data in order to validate all the data before moving storing it to the database. DIXF includes these staging tables directly in the framework and DIXF has the ability to validate the XML file and its data. This means that when using DIXF rather than AIF, two steps in the process can be removed, since the staging tables have moved into the DIXF itself along with the queue, validation, and transformation system that DIXF completely replaces.

AX7 has also made recurring integrations much more efficient. In AX 2012 this was mainly done using the AIF but with the new version it has been simplified using the data entities and DIXF together with REST APIs[51].

5.5 Customizations and Data Management

In AX 7 there are two ways of handling customizations: Overlaying (similar to the old system used in AX 2012) and Extensions.

When a developer wants to modify an existing model in a lower layer, they overlay the new code by adding to the same package as the old model but in a higher layer. AX understands that it should only run the new code when it is available, while it uses the old model if nothing has been modified. The problem with this approach is that whenever an

update comes for the full model, then the customizations all need to be manually reviewed in order to decide on whether they should stay unchanged, be eliminated completely or partly changed.

Extensions were introduced in AX 7 and work on the same principle as the application itself, i.e., using models. Extensions can either add completely new code to the old model or replace parts of the older code. The reason that extensions do not turn into a problem when there are updates is that the extensions are compiled separately from the rest of the code.

AX 7 also changed the way data is stored inside the database. In Figure 5-3 we can see that in AX 2012 the customer has 4 smaller tables, hence the developer has to keep track of all the data and all the relations - but this can become unnecessarily complex. In contrast, Dynamics AX 7 offers data entities allowing an abstract customer entity to be created as seen on the right side of figure 5-3. This abstract customer entity includes the metadata for all the fields and all of the information about where in the database this information is stored. This means that the developer no longer needs to keep track of this information, thus greatly simplifying their work.

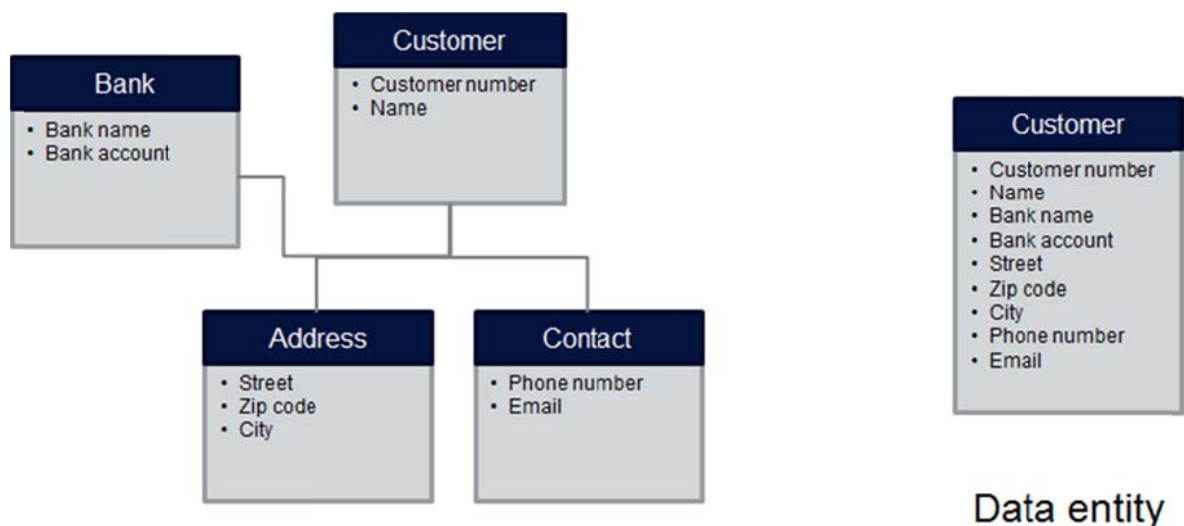


Figure 5-3: Data is managed with data entities in AX 7 (right side of figure) in comparison with multiple tables in Dynamics AX 2012 (left side of figure)

5.6 Summary

This chapter described many advantages of Dynamics AX 7 as well as moving to the cloud. Because Scania has offices and warehouses all over the world, including parts of the world with low speed internet connections, moving to an implementation with a client that simply acts as an interface and communicates much less with the server is a huge advantage. Today it is quite common that a lot of time is spent waiting on the client to do a computation. In contrast, Dynamics AX 7 can deal with large forms by sending and receiving one message with all the information, thus saving a lot of roundtrips as compared with AX 2012 where each field was sent one by one.

The disadvantage is that a lot of work needs to be done to convert older projects to use DIXF over AIF. Moreover, those projects that still use AX 2009 will have a much tougher upgrade path than projects already using AX 2012. As a result, I feel that moving to AX 7 must be explored on a project by project basis. My recommendation is that a good first step is to move all the AX 2009 projects to AX 2012, as this architecture is similar to AX 7.

Additionally, because AX 2009 and AX 2012 are both mature software there are many tools to help support such an upgrade. To facilitate reaching the final goal (running everything on Dynamics AX 7) it is important to use as much best practice as possible in all new developments and to only use DIXF for integrations. I also recommend keeping up to date on what Microsoft is doing in order to follow the AX 7 standards whenever it is possible to do so. There should be an investigation into the amount of work required to change all the current AIF implementations to use DIXF. Given this information one could come up with a plan to achieve this change in a reasonable timeframe.

Overall, it is clear that AX 7 is an improvement over the earlier two versions with regard to performance and usability. Moreover, it is clear that this has been the main focus of Microsoft, as there is almost no new actual functionality in AX 7. However, AX 7 has only recently been released, hence is important to learn what its users are saying about their experience – rather than blindly listening to Microsoft. It is also important to note that as Scania is an international company based out of Sweden* and at the time of this thesis AX 7 has not yet been localized for Swedish users, Microsoft has said that this will happen in the large update of AX 7 sometime in 2016, they have however not specified it more than that. While it is possible to use AX7 in Sweden, Microsoft does not guarantee support for local policies and laws, one example of a Swedish localization in AX 2012 would be how to handle SEPA/ISO20022 credit transfers according to Swedish Bankers Association standards. It also needs to be taken into account that considering so many projects at Scania are international, they would have to comply with both Swedish laws and policies as well as the local laws and policies of the other countries involved in the projects.

* Although Volkswagen AG owns the majority of the shares and the next largest holder of shares are owned by MAN – both of which are German companies.

6 Analysis

This chapter combines data from both the literature study and the case study. This data is analyzed and from this analysis a conclusion will be drawn as to what the important steps are for a company that wants to move their version of Dynamics AX to the cloud as well as when and why this should be done. This analysis will focus on both general cloud benefits/issues as well as specific benefits of AX 7 as this specific system was the focus of this thesis project.

6.1 Performance and Accessibility

The one thing other than functionality that usually gets the largest improvements from upgrades in systems would be performance which sometimes is enough reason to want to upgrade on its own. The interest for this thesis however regarding performance is all about the improvements that can be gathered from a cloud solution. As AX 7 currently only is available for the cloud, hence it is impossible to do a comparison of performance on the same hardware as AX 2012 is currently running on. As a result, the analysis will be mostly theoretical. None of the analysis on AX 2012 on-premise compared to AX 2012 in the cloud focuses on performance as such, but rather the focus is on the general advantages that can be gained by this upgrade. To get to the bottom of this subject one first needs to realize what parts of the ERP system can benefit from the cloud with respect to their performance. Based on the case study it is clear that having a load balancer that can ensure that caching is used optimally will be a great advantage. Additionally, while in theory it would be possible to implement such load balancing for the on-premises ERP, it would be both costly and require specific maintenance and support that such an implementation is not worth it. As previously mentioned in Section 5.1 the fact that the batch server and the regular AOS are separated introduces sever bottlenecks. This on its own is also not a specific cloud feature, but it is a new feature in AX 7. How much this benefits a specific implementation has to be calculated and analyzed on a case-by-case basis, as it depends entirely on how the system is used. Shukla and Verma[36] mention another huge benefit: information can be retrieved from anywhere as the cloud is accessible as long as you have an internet connection. The data entities used in AX 7 are also a large benefit here because while the computation times and the data lookup in the backend will not change, the fact that only one entity has to be looked at rather than having to go into many different tables at the time will save a lot of development time. However, how much this will change for a general implementation is impossible to measure without knowing on average how many entities each system will use.

Hassan et al.[44] relate testability of a piece of software to the performance of said software and show that if a system can easily be tested for functionality, then it most likely is also easy to test it for performance. It has been known for a long time that writing tests using a modular approach is a great way of doing writing tests [52]. A conclusion can be drawn that AX 7 will be easier to test for both functionality and performance than AX 2012 which in turn will mean that it is much easier to identify where the performance issues are and in turn solve those problems. This is expected to lead to a more balanced and better ERP system performance.

6.2 Scalability

Scalability is one of the largest advantages of the cloud as mentioned by Shukla and Verma[36] as well as Jain and Sharma[37]. As Jain and Sharma[37] also mentions it appears that people more experienced with Cloud and ERP solutions see the scalability and accessibility of the cloud as the two large advantages that people without this experience can easily miss. One can assume that people with more experience will have a better understanding of the advantages that are available; therefore, a conclusion is that scalability is a great advantage of the cloud. In the end, this is both an advantage in cost savings and an environmental advantage. The next section describes this cost savings. An environmental advantage also occurs as instead of having idle servers running as backup in case they are needed - one only runs the required number of servers as it is easy to scale both upwards and downwards. Therefore, a cloud-based solution is much better for the environment as less energy and electricity will be used to run the required servers.

6.3 Cost Savings

One of the huge benefits of the cloud is the fact that there is no need to run any local servers. Moreover, as described above, there will never be any idle servers running just in case a quick scale up is necessary. Everything about cloud contributes to either convenience or cost savings for most companies. In most cases, the two go hand in hand. A pure cost savings analysis in regards to moving Dynamics AX to the cloud is nearly impossible to do without having all the information about the current implementation at hand (as one needs to know the number of users and current pricing). Dynamics AX 2012 works on a flexible pricing plan and therefore varies for each implementation. However, estimates can be made based upon public information. For AX 7, public pricing is available (shown in Table 6-1) when the license is bought directly from Microsoft, rather than through one of their partners.

Table 6-1: Dynamics AX 7 Pricing [53]

Enterprise User	20 User minimum: \$185 per user/month
Task User	Approve time/expense/ invoices, add sales & purchase invoices: \$30 per user/month
Self-Service User	Record time & expense, create requisitions & budget activities, view rights into AX – \$8 per user/month

The other tool that can be used for an AX cost calculation would be the LCS tool provided by Microsoft called “How to Run Dynamics AX on Azure” [54]. This tool shows estimated savings for development, test, and production environments. According to this tool the development environment will save 16% of cost (as compared to running AX locally) as well as being 88% faster. The test environment will save 14% in costs and be 88% faster. The gains in these two environments will also not change with the number of developers. Production environment cost savings, varying from 15% to 6%, will decrease with increasing number of users and the daily transaction volume. However, the time saved will remain at 89%. Based on this information, I conclude that if you have a very

large system even less money will be saved, but the time that is saved may still result in larger profits.

How does this compare to regular cloud saving estimates. The experiment done at Sullivan University showed that migration to the cloud resulted in cost savings of \$405,034 for 700 clients[38], corresponding to savings of \$578.62 per client. Considering that the cost of a client in the cloud for them is \$265.20 this savings is almost 70%. Based on this information, I concluded that the cost saving for Dynamics AX in the cloud will be far from the theoretically optimal cost savings for cloud computing. However, considering a company where 50 users have Dynamics AX licenses, this corresponds to almost \$10,000 in licensing fees alone per month or \$120,000 a year. To this one would have to add server costs and computing costs which according to the LCS tool are a minimum of \$100,000 a year in the cloud. Being able to save 15% of the cost as well as being able to save time adds up to quite a lot of money.

6.4 Security and Privacy

One of the larger things to consider before moving to the cloud are the security issues as well as the privacy issues that may come up. The importance of these issues will vary between implementations, but the core concept which will be analyzed here will always be the same.

The first question to consider is if the data will be more or less secure in the cloud instead of in an on-premise server room. As currently everything concerning an AX 7 deployment will be stored in Azure, this is the cloud that will be analyzed. As previously mentioned in Section 4.3, Azure has an uptime of almost 100%. While this means that the data is always accessible anywhere, it also means that the system is available as a target for hackers almost all of the time. Given the large number of services and customers that use Azure it will be a prime target for hackers. This is why Microsoft is using an “Assume Breach” process in regards to their security measures. This means that they never consider the data within the system secure and have a team working on security measures all the time. This is something that most companies cannot afford to have, so every company thinking about moving to the cloud will need to do an analysis of whether the benefits of always cloud computing outweighs the downsides of having a team working on security measures at all times and being much more likely to be targeted of an attack. Considering the resources that are required to keep a system secure and how quickly technology evolves, the conclusion that I draw from this is that unless the data is very unimportant, possibly even public information, from a security standpoint one should always aim to keep it in the most secure location - even if that location might be more prone to be targeted.

Privacy issues actually support a decision to move the data to the cloud. Morales et al.[39] showed that the cloud being less secure than on-premise is a common myth that is one of, if not the largest, reason for people being skeptical about moving information to the cloud. The reality is that it is all up to the service provider to keep their system secure, and as previously mentioned Microsoft is one of the largest companies within the IT world and spends \$1 billion each year on IT security[55]. Considering that Microsoft themselves are using Azure for as much as possible, it would be foolish to assume that they would not keep it as secure as possible.

Unfortunately, the full spectrum of privacy issues is much more complicated, especially for countries other than the USA and even more so for countries that do not have a data center inside their borders. A Forbes article from 2012 by Patrick Baillie [56], who is the

CEO and co-founder of CloudSigma, describes the compatibility issues between the U.S. Patriot Act [57] and the EU laws regarding data protection [58]. As Scania's main base of operations is in Sweden and the EU and the majority of the data at the company goes through Sweden and the EU - this will be the focus of this section. Regarding Dynamics AX in Azure, the company has full control over what data center to use, so we will assume that this data is stored somewhere within the EU, which today would mean either Ireland or The Netherlands.

As Microsoft mentions on their website, their policy regarding releasing the data in their cloud to a government and/or law enforcement agency is: "When a government wants customer data – including for national security purposes – it must follow the applicable legal process, serving us with a court order for content or a subpoena for account information. If compelled to disclose customer data, Microsoft will promptly notify you and provide a copy of the demand, unless legally prohibited from doing so. We do not provide any government with direct or unfettered access to customer data, except as directed by you or where required by law." This statement is as close to a non-statement as it could possibly be and is exactly what Baillie described in his article. If the US government asks for data to be released it can be assumed that they would also ask for this request not to be conveyed to the targeted company, which in turn means that a company might never know that the data was released. This is the reason for Azure creating a new data center in Germany as the data will always be stored *only* in Germany and will be controlled by T-Systems which is a Deutsche Telekom company. T-Systems will act as the trustee for the data.

However, if the Irish or Dutch data center are used, there is a risk that the being stored in Azure could be released by Microsoft if asked for it *without* the owner of the data ever knowing. Which Baillie mentions is in complete contradiction to EU laws which can lead to Microsoft being held liable if any of the data is ever released. This information is also includes relevant financial data used to calculate taxes. The UK in particular has very specific laws on what is and what is not allowed to be store in the cloud when it comes to financial services. They have recently opened up the possibility of storing some data in the cloud, but it is still quite limited and can cause huge problems as parts of the regulations are quite unclear as to what is and what is not allowed [59].

In summary, privacy and data protection are an extremely complicated and need to be analyzed in great detail at each company considering moving to AX 7 at this point in time. Moreover, AX7 is only available as a cloud service. As soon as there is a possibility for a company to use AX 7 either locally or within a private or hybrid cloud these issues will be reduced, but at this point in time it requires a case by case analysis of whether moving to the cloud is worthwhile for a company.

7 Conclusions and Future work

This chapter will describe the arrived at conclusion as well as the limitations of the study and the future work that needs to be done in order to get a better understanding of the subject.

7.1 Conclusions

Any company considering a move to Cloud ERP and specifically Dynamics AX 7 will need to do a thorough analysis of its own current implementation and look at what the changes would actually mean. It is clear that from a performance, accessibility, and cost perspective there would be benefits of varying degrees in the end. However, what is unclear is if those benefits outweigh the possible downsides when it comes to security and privacy. The security issues will not matter for the majority of implementations and in the case of Azure the data will most likely be even more secure in the cloud than it is in an on-premises deployment. This is especially true when one considers the amount of time and money Microsoft is spending on security versus what a single company can afford to spend.

Privacy is where the main problem area today for any company that is not in the same country as the cloud provider. The larger ERP systems and the larger cloud providers are all American companies. Baillie mentions in his Forbes article that American companies owned more than 90% of the cloud solutions in use in Europe in 2012. As a result, my suggestion regarding Dynamics AX for any American company would be to jump onboard as soon as possible - in order to stay at the front edge of technology. However, for European companies and especially countries that are part of the EU, I would recommend a wait and see approach. Microsoft has said that they will unveil information regarding AX 7 in a hybrid or private cloud later this year; hence I would recommend waiting until then to take the plunge. Such alternative deployments would ensure that the company is in the clear legally.

What can be done meanwhile is to ensure that all analysis regarding the on-premise implementation is done, including cost analysis, minimizing customizations, and using Microsoft best practice as much as possible will help make the transition much smoother and quicker in the future. If running AX 2009, a recommendation would also be to try to transition to AX 2012, as this would be a large step towards a cloud implementation and AX 7.

The modularity of AX 7 means that there will be smaller, easier, and more effective updates than before. Microsoft has said that they plan small updates each month and one larger update every 3 months, rather than only two large updates across the entire life of the product as was the with AX 2012. The modularity of the architecture means that these smaller upgrades will be much more beneficial than they would have been in AX 2012 [43].

7.2 Limitations

The fact that there was no actual implementation of AX 7 done is a clear limitation as it means that there is no quantitative data. For the purposes of this thesis project, it would have been preferable to actually have an implementation, but not doing so was a necessary decision due to the large costs in setting up such an implementation.

7.3 Future work

The obvious task for the future would be to collect some quantitative data regarding a direct upgrade from AX 2012 to AX 7 running on the same underlying hardware. This is very unrealistic to expect, unless there is a business who has already decided to upgrade. However, even in this case they would still need to keep AX 2012 running in parallel with the upgrade in order to perform the required comparisons. Unless Microsoft themselves decide to release such a direct comparison I would not be surprised if such a comparison never happens. Another thing to look at would be to identify additional specific differences between AIF and the DIXF and to see if these can be measured in a similar way.

What can be done however is to look into possibilities regarding the performance that can be gained from batch jobs. Can some sort of batch scheduler be implemented which in turn can prioritize the jobs in a way that minimizes the time lost by waiting for another job to finish.

Another task would be to look into the possibilities of using encrypted data in the cloud. Is it possible to implement some sort of encryption while using AX on Azure, will it only be possible to do in a hybrid or private cloud or can it not be done at all? Depending on the answer to this question some of the possible legal issues mentioned in section 6.4. might be irrelevant.

References

- [1] Microsoft Corporation, “Next generation of business applications,” *Microsoft Dynamics*. [Online]. Available: <https://www.microsoft.com/en-us/dynamics/erp-ax-overview.aspx>. [Accessed: 18-May-2016].
- [2] Microsoft Corporation, “What is ERP? Enterprise Resource Planning,” *Microsoft Dynamics*, Feb-2013. [Online]. Available: <https://www.microsoft.com/en-us/dynamics/what-is-erp.aspx>. [Accessed: 18-May-2016].
- [3] Archana Venkatraman, “UK cloud adoption swells by 61% in four years,” *ComputerWeekly.com*, 04-Jul-2014. [Online]. Available: <http://www.computerweekly.com/news/2240223928/UK-cloud-adoption-swells-by-61-in-four-years>. [Accessed: 18-May-2016].
- [4] Microsoft Corporation, “Microsoft Acquires Navision,” *News Center*, 11-Jul-2002. [Online]. Available: <http://news.microsoft.com/2002/07/11/microsoft-acquires-navision/#sm.00008q0o1u7uvd1xryu1j9z4qqg2z>. [Accessed: 18-May-2016].
- [5] Andreas Luszczak, *Using Microsoft Dynamics AX 2012*. Wiesbaden: Springer Fachmedien Wiesbaden, 2015.
- [6] Jordan Novet, “Microsoft launches new Dynamics AX service out of public preview,” *VentureBeat*, 09-Mar-2016. [Online]. Available: <http://venturebeat.com/2016/03/09/microsoft-launches-new-dynamics-ax-service-out-of-public-preview/>. [Accessed: 18-May-2016].
- [7] “About ERP History,” *ERPandmore.com*. [Online]. Available: <http://www.erpandmore.com/erp-reference/erp-history/>. [Accessed: 18-May-2016].
- [8] John Harhen, “MRP/MRP II,” in *Computer-Aided Production Management*, Springer Berlin Heidelberg, 1988, pp. 23–35.
- [9] Elliott Taylor, “MRP Vs. MRPII,” *Chron.com*. [Online]. Available: <http://smallbusiness.chron.com/mrp-vs-mrpii-15365.html>. [Accessed: 18-May-2016].
- [10] E. Mullaney, “The Origin of Enterprise Resource Planning (ERP),” *Cornerstone Consulting*, 08-Jun-2012. .
- [11] Ruth Raistrick, “Why ERP has never been more alive during Industry 4.0,” *LinkedIn*, 25-Nov-2015. [Online]. Available: <https://www.linkedin.com/pulse/why-erp-has-never-been-more-alive-during-industry-40-ruth-raistrick>. [Accessed: 17-Aug-2016].
- [12] erezource Infotech Pvt. Ltd., “ERP Architecture.” [Online]. Available: <http://www.erezourceerp.com/erp-architecture.html>. [Accessed: 18-May-2016].
- [13] NetSuite, “Cloud ERP vs. On-Premise ERP.” [Online]. Available: <http://www.netsuite.com/portal/resource/articles/on-premise-cloud-erp.shtml>. [Accessed: 18-May-2016].
- [14] Sanjeev Aggarwal and Laurie McCabe, “The Compelling TCO Case for Cloud Computing in SMB and Mid-Market Enterprises: A 4-year total cost of ownership (TCO) perspective comparing cloud and on-premise business application deployment.” Hurwitz & Associates, 2009.
- [15] Sycor GmbH, “Axapta 1.0 to Microsoft Dynamics AX 2012.” [Online]. Available: <http://en.sycor-group.com/dynamics-ax/solutions/dynamics-ax/history/#.VzyNHr4VZ81>. [Accessed: 18-May-2016].
- [16] Microsoft Corporation, “Dynamics AX Milestones,” *TechNet Articles*. [Online]. Available: <http://social.technet.microsoft.com/wiki/contents/articles/5812.dynamics-ax-milestones.aspx>. [Accessed: 18-May-2016].

- [17] Erik P. Ernst and Ernesto Muñiz, “The history of Dynamics AX / Axapta - Dynamics AX Wikipedia - AX User Group,” *Dynamics User Group*, 11-Oct-2010. [Online]. Available: <http://dynamicsuser.net/ax/w/axapta/the-history-of-dynamics-ax-axapta>. [Accessed: 18-May-2016].
- [18] Microsoft Corporation, “Overview of Enterprise Portal for Microsoft Dynamics AX [AX 2012],” *TechNet*, 17-Apr-2013. [Online]. Available: <https://technet.microsoft.com/en-us/library/gg731806.aspx>. [Accessed: 18-May-2016].
- [19] Microsoft Corporation, “ERP Business Systems and Solutions,” *Microsoft Dynamics*. [Online]. Available: <https://www.microsoft.com/en-us/dynamics/erp.aspx>. [Accessed: 18-May-2016].
- [20] SBS Group, “Top 10 Enhancements in Dynamics AX 2012,” *ERP Software Blog*, 14-Jul-2011. [Online]. Available: <http://www.erpsoftwareblog.com/2011/07/top-10-enhancements-in-dynamics-ax-2012/>. [Accessed: 18-May-2016].
- [21] Microsoft Corporation, “Layers [AX 2012],” *Microsoft Developer Network*, 16-Sep-2014. [Online]. Available: <https://msdn.microsoft.com/en-us/library/aa851164.aspx>. [Accessed: 18-May-2016].
- [22] Microsoft Corporation, “Best Practices for Application Layers,” *Microsoft Developer Network*. [Online]. Available: [https://msdn.microsoft.com/en-us/library/aa868050\(v=ax.50\).aspx](https://msdn.microsoft.com/en-us/library/aa868050(v=ax.50).aspx). [Accessed: 18-May-2016].
- [23] Tridea Partners, “Understanding Layers in Microsoft Dynamics AX 2012,” *ERP Software Blog*, 25-Jun-2014. [Online]. Available: <http://www.erpsoftwareblog.com/2014/06/understanding-layers-in-microsoft-dynamics-ax-2012/>. [Accessed: 18-May-2016].
- [24] Microsoft Corporation, “How to: Compare Layers [AX 2012],” *Microsoft Developer Network*, 15-Sep-2011. [Online]. Available: <https://msdn.microsoft.com/en-us/library/aa890665.aspx>. [Accessed: 18-May-2016].
- [25] IBM, “What is cloud computing?” [Online]. Available: <https://www.ibm.com/cloud-computing/what-is-cloud-computing>. [Accessed: 18-May-2016].
- [26] Peter Mell and Timothy Grance, “The NIST Definition of Cloud Computing,” U. S Chamber of Commerce, National Institute of Standards and Technology, Gaithersburg, MD, USA, Special Publication 800-145, Sep. 2011.
- [27] Microsoft Corporation, “Directory of Azure Cloud Services,” *Microsoft Azure*. [Online]. Available: <https://azure.microsoft.com/en-us/services/>. [Accessed: 18-May-2016].
- [28] Microsoft, “Azure Regions,” *Azure Regions*. [Online]. Available: <https://azure.microsoft.com/en-us/regions/>. [Accessed: 19-Aug-2016].
- [29] “There are now 1.2 billion Office users and 60 million Office 365 commercial customers,” *Windows Central*, 31-Mar-2016. [Online]. Available: <http://www.windowscentral.com/there-are-now-12-billion-office-users-60-million-office-365-commercial-customers>. [Accessed: 17-Aug-2016].
- [30] M. O. Henrik Karlsson, “I Microsoft Dynamics AX – databasaccess, kommunikation och tjänster för Commerce Runtime,” Linköping University, 2013.
- [31] Microsoft Corporation, “Commerce Runtime [AX 2012],” *TechNet*, 29-Apr-2014. [Online]. Available: <https://technet.microsoft.com/sv-se/library/jj916620.aspx>. [Accessed: 23-Aug-2016].
- [32] S. P. Li Fang, “Critical Success Factors in ERP Implementation,” Jönköping University, 2005.

- [33] K. M. Stefan Koch, "Linking customisation of ERP systems to support effort: an empirical study," 22-May-2014. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/17517575.2014.917705>. [Accessed: 18-Aug-2016].
- [34] H. S. Gerald Feldman and A. A. Craig Chapman, "Enterprise Systems Upgrade Drivers: A Technological, Organisational and Environmental Perspective," *PACIS 2015 Proceedings.*, 2015.
- [35] Lori Hogan, "Critical Success Factors for ERP Implementations," Dec-2014. [Online]. Available: <https://scholarsbank.uoregon.edu/xmlui/bitstream/handle/1794/19637/Hogan2014.pdf>. [Accessed: 23-Aug-2016].
- [36] S. V. Amit Shukla, "A Comparative Analysis On Cloud And ERP," *International Journal Of Research In Computer Engineering And Electronics.*, vol. 4, no. 3.
- [37] Y. S. Dhiraj Jain Sr., "Cloud Computing with ERP - A Push Business Towards Higher Efficiency," *Annul Research Journal of SCMS Pune*, vol. 4, pp. 140–155, Mar. 2016.
- [38] M. K. Emmanuel Udoh and D. A. Michael Grosse, "Cloud Infrastructure for Higher Education: The Sullivan Experience," *Information and Computer Technology (GOICT), 2014 Annual Global Online Conference on*, Jun. 2015.
- [39] O. V. Valdir Morales, V. S. L. Samuel Dereste dos Santos, and Edivaldo Antonio Sartor, "ERP Evaluation in Cloud Computing Environment," in *Advances in Production Management Systems: Innovative Production Management Towards Sustainable Growth*, vol. 459, 2015, pp. 78–84.
- [40] A. S.-H. L. Tong-Ming Lim and Mun-Keong Yap, "User Acceptance of SaaS ERP Considering Perceived Risk, System Performance and Cost," in *Information Science and Applications (ICISA) 2016*, vol. 376, 2016, pp. 53–63.
- [41] J. T. Tim Boudreau and Geertjan Wielenga, *Rich Client Programming: Plugging into the NetBeans Platform.* .
- [42] A. K. Manoj Nambiar, R. S. Gopal Bhaskaran, and Subhasri Duttagupta, "Model Driven Software Performance Engineering: Current Challenges and Way Ahead," *ACM SIGMETRICS Performance Evaluation Review*, vol. 43, no. 4, pp. 53–62, Mar. 2016.
- [43] W. H. Hongyi Sun and J. H. Min Xie, "Modularity's impact on the quality and productivity of embedded software development: a case study in a Hong Kong company," *Journal Total Quality Management & Business Excellence*, vol. 26, no. 11–12, pp. 1188–1201, Jun. 2014.
- [44] W. A. Mohammad Mahdi Hassan, S. M. A. S. Birgitta Lindström, and M. B. Sten F. Andler, "Testability and software performance: a systematic mapping study," *SAC '16 Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pp. 1566–1569.
- [45] T. H. J. V. M.S. Essers, "Design of a decentralized modular architecture for flexible and extensible production systems," *Mechatronics*, vol. 34, pp. 160–169, Mar. 2016.
- [46] Milinda Vitharana, "Pinning Power BI reports to Dynamics AX client," *Microsoft Developer*, 06-Jul-2016. [Online]. Available: <https://blogs.msdn.microsoft.com/dynamicsaxbi/2016/07/06/pinning-power-bi-reports-to-dynamics-ax-client/>. [Accessed: 26-Aug-2016].
- [47] Steve Stein, "Overview: SQL Database Active Geo-Replication," *Overview: SQL Database Active Geo-Replication*, 14-Jul-2016. [Online]. Available:

- <https://azure.microsoft.com/en-us/documentation/articles/sql-database-geo-replication-overview/>. [Accessed: 08-Aug-2016].
- [48] Committee Draft 02, “Security Assertion Markup Language (SAML) V2.0 Technical Overview,” *Security Assertion Markup Language (SAML) V2.0 Technical Overview*, 25-Mar-2008. [Online]. Available: <http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html>. [Accessed: 08-Aug-2016].
- [49] “OAuth 2.0,” *OAuth 2.0*. [Online]. Available: <http://oauth.net/2/>. [Accessed: 08-Aug-2016].
- [50] Sally Erickson, “Deprecated features,” *Deprecated features*, 13-Jul-2016. [Online]. Available: <https://ax.help.dynamics.com/en/wiki/deprecated-features-in-dynamics-ax-7/#aif-axd-and-axbc-integrations>. [Accessed: 08-Aug-2016].
- [51] Kuntal Mehta, “Recurring integrations,” *Dynamics AX Help Wiki*, 07-Mar-2016. [Online]. Available: <https://ax.help.dynamics.com/en/wiki/recurring-integrations/#authorization-for-the-integration-rest-api>. [Accessed: 31-Aug-2016].
- [52] “Software testing for complex, integrated applications? Go modular!,” *TechBeacon*, 21-Aug-2015. [Online]. Available: <http://techbeacon.com/software-testing-complex-integrated-applications-go-modular>. [Accessed: 30-Aug-2016].
- [53] RoseAsp Dynamics Hosting, “Microsoft Dynamics AX Pricing and Licensing,” *Dynamics Community*, 16-Mar-2016. [Online]. Available: <https://community.dynamics.com/gp/b/roseaspdynamicsgphostingnews/archive/2016/03/16/microsoft-dynamics-ax-pricing-and-licensing>. [Accessed: 30-Aug-2016].
- [54] Microsoft Corporation, “The Cloud Makes You a Business Hero.” [Online]. Available: <https://lcs.dynamics.com/AxOnAzure/axOnAzure.html>. [Accessed: 29-Aug-2016].
- [55] Julie Bort, “By spending \$1 billion a year on security, Microsoft went from ‘worst’ to ‘best,’” *Business Insider*, 17-Nov-2015. [Online]. Available: <http://www.businessinsider.com/microsoft-security-from-worst-to-best-2015-11?r=US&IR=T&IR=T>. [Accessed: 30-Aug-2016].
- [56] Patrick Baillie, “Can European Firms Legally Use U.S. Clouds To Store Data?,” *Forbes*, 02-Jan-2012. [Online]. Available: <http://www.forbes.com/sites/ciocentral/2012/01/02/can-european-firms-legally-use-u-s-clouds-to-store-data/#21604aa27ede>. [Accessed: 29-Aug-2016].
- [57] “USA PATRIOT Act (H.R. 3162),” *Epic.org*, 24-Oct-2001. [Online]. Available: <https://epic.org/privacy/terrorism/hr3162.html>. [Accessed: 31-Aug-2016].
- [58] “Protection of personal data,” *Building a European Area of Justice*, 02-Aug-2016. [Online]. Available: <http://ec.europa.eu/justice/data-protection/>. [Accessed: 31-Aug-2016].
- [59] “FCA gives green light to use of cloud computing in UK financial services,” *Out-law.com*, 08-Jul-2016. [Online]. Available: <http://www.out-law.com/en/articles/2016/july/fca-gives-green-light-to-use-of-cloud-computing-in-uk-financial-services/>. [Accessed: 31-Aug-2016].
- [60] K. D. Hall and H. W. Wimmer, “Integrating ERP and CRM into the University Curriculum with Microsoft Dynamics AX and CRM,” in *Celebrating America’s Pastimes: Baseball, Hot Dogs, Apple Pie and Marketing?*, K. K. Kim, Ed. Cham: Springer International Publishing, 2016, pp. 89–98.
- [61] Explorable.com, “Empirical Research,” *Empirical Research*. [Online]. Available: <https://explorable.com/empirical-research>. [Accessed: 02-Aug-2016].

- [62] John Dudovskiy, “Inductive Approach,” *Inductive Approach (Inductive Reasoning)*. [Online]. Available: <http://research-methodology.net/research-methodology/research-approach/inductive-approach-2/>. [Accessed: 02-Aug-2016].
- [63] Brandon Butler, “And the cloud provider with the best uptime in 2015 is...,” *And the cloud provider with the best uptime in 2015 is...*, 07-Jan-2016. [Online]. Available: <http://www.networkworld.com/article/3020235/cloud-computing/and-the-cloud-provider-with-the-best-uptime-in-2015-is.html>. [Accessed: 08-Aug-2016].
- [64] Brandon Butler, “Which cloud providers had the best uptime last year?,” *Which cloud providers had the best uptime last year?*, 12-Jan-2015. [Online]. Available: <http://www.networkworld.com/article/2866950/cloud-computing/which-cloud-providers-had-the-best-uptime-last-year.html>. [Accessed: 08-Aug-2016].
- [65] “Microsoft BizTalk Server,” *Microsoft BizTalk Server*. [Online]. Available: <https://www.microsoft.com/en-us/cloud-platform/biztalk>. [Accessed: 08-Aug-2016].
- [66] Andreas Kjellman, “Azure AD Connect sync: Understand and customize synchronization,” *Azure AD Connect sync: Understand and customize synchronization*, 27-Jun-2016. [Online]. Available: <https://azure.microsoft.com/en-us/documentation/articles/active-directory-aadconnectsync-what-is/>. [Accessed: 08-Aug-2016].
- [67] J. Edwards, *3-tier client/server at work*, Rev. ed. New York: John Wiley, 1999.

TRITA-ICT:EX-2016:145