



Fakultet
(Ämne/Avdelning)

Linus Svensson & Anders Nguyen

Migrering av en applikation till datormolnet Windows Azure

Migration of the an application to the cloud Windows
Azure

Examensarbete 15 hp

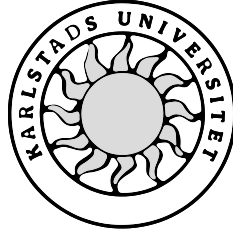
Program

Datum/Termin: 2010-06-08

Handledare: Katarina Asplund

Examinator: Martin Blom

Ev. löpnummer: C2010:04



Karlstad universitet

Migrering av en applikation till datormolnet Windows Azure

Linus Svensson

Anders Nguyen

Denna rapport är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

Linus Svensson

Anders Nguyen

Godkänd, 2010-06-08, Karlstad universitet

Handledare: Katarina Asplund

Examinator: Martin Blom

Sammanfattning

Den här uppsatsen beskriver arbetet bakom migreringen av en .NET webapplikation skriven i C#, kallad AMM, till en molnapplikation som sedan lagts upp på Windows Azure datormoln.

Arbetet är gjort på uppdrag av Sogeti Karlstad som är ett konsultföretag för IT-relaterade lösningar. Målet bakom migreringen var att utreda om Windows Azures molntjänst lämpade sig till att användas för mindre applikationer såsom denna. Själva uppgiften var att få applikationen att fungera i Windows Azures datormoln samt att förbättra befintliga funktioner inom applikationen. Resultatet blev en applikation som drivs helt och hållet av Microsoft Windows Azure, och den har funktionen att kunna skapa nya databaser beroende på hur många kunder som använder sig av tjänsten.

Migration of an application to the cloud

Windows Azure

Abstract

This dissertation describes the work behind the migration of a .NET web application written in C#, called AMM, to a cloud application, which later have been uploaded on the Windows Azure cloud.

The work has been done on behalf of Sogeti Karlstad, which is a consulting company for IT-related solutions. The goal behind the migration was to investigate if the Windows Azure cloud services were suitable for smaller applications such as this. The actual task was therefore to make the application work with Windows Azure as well as improving existing features within the application. The result is an application that is run entirely by Microsoft Windows Azure, and that has the option to create new databases, depending on how many customers that use the service.

Innehållsförteckning

Figurförteckning.....	vii
1 Inledning	1
1.1 Disposition.....	1
2 Bakgrund	3
2.1 Applikationen - AMM	3
2.2 Datormolnet	5
2.3 Windows Azure	7
2.3.1 Windows Azure operativsystem.....	7
2.3.2 Windows Azure-plattformen AppFabric.....	7
2.3.3 SQL Azure	7
2.4 Compute Storage och Fabric	8
2.4.1 Windows Azure Compute Service	8
2.4.2 Windows Azure Storage Service (Storage).....	9
2.4.3 AppFabric	12
2.4.4 Betalningsalternativ för Windows Azure	13
2.5 Summering.....	14
3 Migrering och utveckling av applikationen.....	15
3.1 Krav	15
Förbättringar.....	16
3.2 Utvecklingsmiljö.....	16
3.3 Molnprojekt för Windows Azure.....	17
3.4 Förbättringar av AMM-applikationen	18
3.4.1 Inget defaultsvar ikryssat i testet	18
3.4.2 Kunna släcka ner ”vet ej” och ”delvis” alternativen i testet	18
3.4.3 Kunna släcka ner dropdownlistor i testet	18
3.4.4 Inloggning per användare.....	19
3.4.5 Flytta användarinformation från rapport till användare	19
3.4.6 Krypterat lösenord för användare och administratören	20
3.4.7 Mindre förändringar	20
3.5 Migreringen av AMM till molnet	20
3.5.1 Steg 1: Byta databas	21
3.5.2 Steg 2: Få SQL Server-databasen att fungera med applikationen	21
3.5.3 Steg 3: Föra över SQL Server - databasen till SQL Azure.....	21
3.5.4 Steg 4: Lägga till applikationen i ett molnprojekt med en webbroll.	22
3.5.5 Steg 5: Publicera Applikationen i molnet.....	22
3.6 Skapa flera versioner av applikationen	22
3.6.1 Steg 1: Lägga till/ta bort ett företag.	24
3.6.2 Steg 2: Logga in som ett företag	24
3.7 Summering.....	26

4	Applikationsöversikt	27
5	Resultat och utvärdering	39
5.1	Resultatet	39
5.2	Utvärdering av datormolnet	40
5.3	Problem.....	42
5.3.1	Migreringen (tillvägagångssättet)	42
5.3.2	Användarvänlighet i applikationen	42
5.3.3	Vad molnet inte stödjer (DLL-filer).....	42
5.4	Framtida arbete	43
5.4.1	Enklare skriptande av databas	43
5.5	Summering.....	44
6	Slutsats	45
	Referenser	46
	Bilaga A	49
	Ordlista.....	49

Figurförteckning

FIGUR 2-1 MOGNADSMATRIS FÖR AMM.....	3
FIGUR 2-2 EXEMPEL PÅ FRÅGOR FRÅN TESTET	4
FIGUR 2-3 MOLNET	6
FIGUR 2-4 ÖVERSIKTSBILD FÖR ANVÄNDNING AV BLOBBLAGRING	10
FIGUR 2-5 ÖVERSIKTSBILD FÖR ANVÄNDNING AV KÖHANTERING	11
FIGUR 2-6 ÖVERSIKTSBILD FÖR ANVÄNDNING AV TABELLAGRING.....	11
FIGUR 4-1 COMPANY LOGIN	27
FIGUR 4-2 COMPANY ADMIN LOGIN	27
FIGUR 4-3 COMPANY ADMIN-GRÄNSSNITTET.....	28
FIGUR 4-4 AMM STARTSIDAN	28
FIGUR 4-5 ADMIN LOGIN.....	29
FIGUR 4-6 AMM-TESTET.....	29
FIGUR 4-7 RESULTAT	30
FIGUR 4-8: ADMINGRÄNSSNITTET: LÄGGA TILL ANVÄNDARE	31
FIGUR 4-9 ADMINGRÄNSSNITTET: ÄNDRA ANVÄNDARINFORMATION	32
FIGUR 4-10 ADMINGRÄNSSNITTET: DROPDOWN HANTERING	33
FIGUR 4-11 ADMINGRÄNSSNITTET: HANTERA LEDTEXTER	34
FIGUR 4-12 ADMINGRÄNSSNITTET: LÄGGA TILL RAPPORTPERIOD	35
FIGUR 4-13 ADMINGRÄNSSNITTET: HANTERA FRÅGESTÄLLNINGAR	36
FIGUR 4-14 ADMINGRÄNSSNITTET: UPPDATERA RAPPORTPERIOD	37
FIGUR 4-15 ADMINGRÄNSSNITTET: ÄNDRA LÖSENORD	38

1 Inledning

Detta är ett examensarbete på C-nivå inom datavetenskap vid Karlstad Universitet. Vi fick i uppdrag av Sogeti Karlstad att förbättra och migrera applikationen AMM till Microsofts datormoln Windows Azure, vilket är en teknik där stora skalbara resurser, exempelvis processorkraft, lagring och funktioner, tillhandahålls som tjänster på Internet. Syftet med vårt uppdrag var att undersöka om datormolnstekniken är en användbar tillhandahållningsmiljö för AMM-applikationen och undersöka vilka sorts applikationer tekniken kan lämpa sig för. Vårt examensarbete utreder också fördelar respektive nackdelar med ett datormoln och vad det innebär att arbeta med dessa. Resultatet blev en applikation som drivs helt och hållet av Microsoft Windows Azure, och den har funktionen att kunna skapa nya databaser beroende på hur många kunder som använder sig av tjänsten. När som helst kan man minska eller öka AMM-applikationens resurser. Skulle applikationen bli väldigt populär kan vi enkelt skala upp och skulle applikationen användas mindre kan vi enkelt skala ner på resurser eller till och med ta bort applikationen vid de tidpunkter den inte alls används.

1.1 Disposition

Vi har valt att dela upp vårt examensarbete i 5 delar: Bakgrunden, Migrering och Utveckling, Applikationsöversikt, Resultat och Utvärdering och sist Slutsats. För att förstå rapportens innehåll berättar vi i kapitel 2 om bakgrunden vilket beskriver principerna bakom AMM-applikationen, datormolnet och Windows Azure. Sedan i kapitel 3 går vi igenom migreringen och utveckling av applikationen som är de praktiska delarna i vårt examensarbete. Detta kapitel innehåller de krav som ställdes på oss av vår betsgivare, hur vi implementerade kraven och hur applikationen förändrades av kraven. I kapitel 4 visar vi hur applikationen ser ut efter att den blev klar, samt beskriver dess funktioner. I kapitel 5 beskriver vi vad vi fick för resultat av examensarbetet, de problem vi hade och framtida förbättringar av applikationen. Vi skriver också en utvärdering av datormolnet. Slutligen i kapitel 6 beskriver vi vilken slutsats vi drog utifrån vårt arbete.

2 Bakgrund

Detta kapitel beskriver applikationen AMM och dess användningsområde samt uppbyggnad i avsnitt 2.1. I avsnitt 2.2 ger vi bakgrundsinformation om vad datormolnet är och vad det används till. Sist men inte minst kommer vi att beskriva och förklara hur Microsoft datormolnsverktyg Windows Azures fungerar i avsnitt 2.3.

2.1 Applikationen - AMM

DYA-AMM (Dynamic Architecture - Architecture Maturity Matrix)[1,2] är ett instrument skapat av Sogeti för att bedöma mognadsnivån hos ett företags IT-arkitektur. DYA är Sogetis vision av hur man ska arbeta med en arkitektur som enligt dem måste vara dynamisk i sin natur. AMM - applikationen ger organisationen en bild över vilka områden de behöver jobba med samt på vilka områden de ligger bra på. Applikationen är ett test där företagets anställda får svara på en mängd frågor om IT-arkitekturen hos deras företag. Utifrån svaren på dessa frågor skapas sedan en matris (se figur 2-1) som visar vilken mognadsgrad företaget har inom olika IT-arkitekturella områden. Via matrisen kan man sedan göra en bedömning av hur organisationen ska arbeta för att förbättra sin IT-arkitektur för att få en flexiblare och effektivare verksamhet. För exempel på frågor från testet se figur 2-2.

Area	Level	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Defining architecture			A			B			C						
Use of architecture				A			B				C				
Involvement of business			A				B				C				
Involvement of development process				A				B		C					
Involvement of operations						A			B			C			
Relation to current situation						A				B					
Roles and responsibilities					A		B					C			
Coordination of activities								A			B				
Controlling					A		B		C		D				
Quality management									A		B			C	
Maintenance architecture process								A		B		C			
Maintenance architecture deliverables						A			B					C	
Commitment and motivation			A					B		C					
Architecture roles and training					A		B			C			D		
Use of architecture method					A						B				C
Consultation				A		B				C					
Architecture tools								A				B			C
Estimation and planning						A						B		C	

Figur 2-1 Mognadsmatris för AMM

Id Question

- 1 Utvecklas arkitektur enbart om det finns en sponsor?
- 2 Är projektplanen för arkitekturprojekt granskad innan arbetet startas?
- 3 Är utveckling av arkitektur betraktat som en kontinuerlig process?
- 4 Uppdateras arkitekturen regelbundet?
- 5 Finns det någon form av versions- och konfigurationshantering när det gäller arkitekturprodukter?
- 6 Har ett sponsorskap etablerats innan arkitekturprodukter utvecklas?
- 7 Är det fastställt vem som tjänar på resultatet innan utveckling av arkitektur startas?
- 8 Är det andra än arkitekter involverade i utveckling av arkitektur?
- 9 Finns det något som ledningen betraktar som arkitektur?
- 10 Visar arkitekturen tydligt vad verksamheten behöver? Innehåller den delar som är verksamhetsutvecklingsfokuserade?
- 11 Är arkitekturen tillgänglig för alla anställda som behöver den?
- 12 Används arkitekturen för att pro-aktivt styra och vägleda verksamhets- och IT-utveckling?
- 13 Är projekten medvetna om alla delar av arkitekturen som berör dess arbete?
- 14 Tillhandahåller arkitekturen konkreta riktlinjer som är till nytta för projekten?

Figur 2-2 Exempel på frågor från testet

2.2 Datormolnet

Ur ett fysiskt perspektiv består datormolnet[3,4,5,6] av nätverk med servrar som kan vara spridda över ett stort område men ändå fungerar ihop och som tillsammans erbjuder en stor processorkraft. Detta datormoln av servrar används för att sköta internetbaserade applikationer och för att lagra stora datamängder. Slut användaren har dock ingen aning om var i datormolnet deras data finns. Ur ett tekniskt perspektiv består datormolnet av två delar. Den första delen är applikationer som levereras som tjänster över internet, vilket man kallar ”Software as a Service(SaaS)”[7]. Den andra delen är Utility Computing[34], vilket är upplevt oändliga resurser och betalning efter resursförbrukning. Dessa två delar beskrivs nedan.

Software as a Service

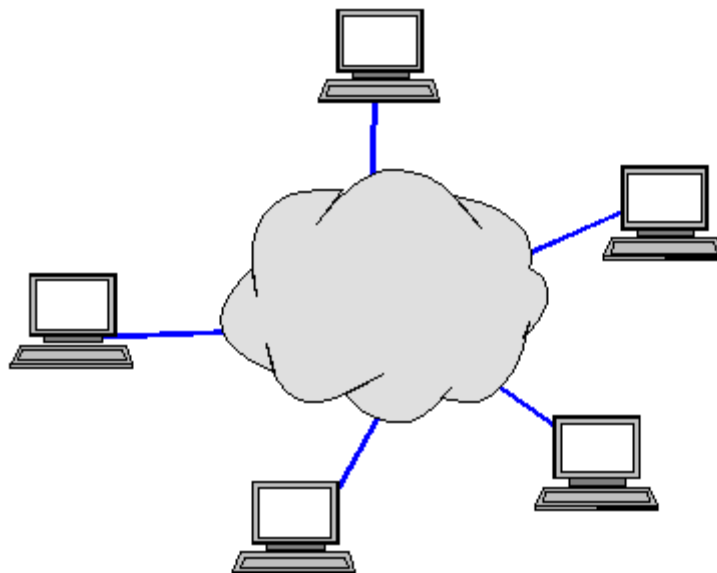
Software as a Service är en modell för programdistribution via Internet. Med SaaS licensierar en leverantör en applikation till kunder för att användas som en tjänst på begäran, antingen genom ett tidsbestämt abonnemang eller genom betalning efter resursförbrukning. SaaS-modellen tillåter leverantörer att utveckla, organisera och driva applikationer för kundens räkning. Istället för att köpa hårdvara och programvara behöver kunderna endast en dator eller en server för att hämta programmet och givetvis tillgång till internet för att köra programvaran. Programvaran kan licensieras för en enskild användare eller för en grupp av användare.

Utility Computing

Utility Computing[34] innebär att resurser och tjänster karaktäriseras som oändliga resurser och betalning efter resursförbrukning. Detta leder till en effektivare användning av resurser då möjligheten finns för företag att endast investera i så mycket datorkraft som de behöver i en viss situation. När en internetbaserad tjänst ökar i popularitet kan företaget då kostnadseffektivt skala upp systemet. Utility Computing kan jämföras med resursförbrukning av t.ex. el och vatten. Du betalar för det vatten och den el du använder och du kan använda så mycket el och vatten som du behöver efter behov.

Datormolnets fördelar och nackdelar

Fördelen med datormoln är att skalbara resurser såsom exempelvis processorkraft, lagring och funktioner tillhandahålls som tjänster på internet. Med denna teknik kan användaren använda och lagra resurser på internet istället för att använda datorns egen processorkraft, minne och hårddisk. Man överläter istället detta arbete till datormolnet (se bild 2.3). Din dator fungerar istället som en slags terminal eller monitor i vilken man anropar datormolnet. Med datormolnet får man också fördelar som t ex. att alltid ha tillgång till datorkraft och endast betala för det man använder. Med denna teknik slipper man också bekymra sig om uppdateringar, antivirusprogram och andra säkerhetsåtgärder. Man har alltid tillgång till samma resurser oavsett vilken dator man använder. Förhoppningsvis blir det billigare att använda sig av datormolns-tekniken än att själv tillhandahålla de resurser som behövs för att utföra arbetet. De nackdelar man har märkt är att datormoln kan vara relativt svårhanterliga jämfört med att använda egna resurser. För att det ska gå smidigt så förutsätts det också att man har en snabb internetuppkoppling. Du måste också kunna lita på att säkerheten, stabiliteten och övriga driftstörningar hanteras korrekt. Hittills har det även varit svårt att byta leverantör utan krånglig överföring då olika datormolntjänster inte stödjer samma tjänster.



Figur 2-3 Molnet

2.3 Windows Azure

Windows Azure[23] är ett internetbaserat skalbart datormoln och en tjänsteplattform som tillhandahålls av Microsofts datacenter. Windows Azure innehåller även ett antal utvecklingstjänster vilka kan användas separat eller tillsammans. Nedan beskriver vi de utvecklingstjänster som Windows Azure består av.

2.3.1 Windows Azure operativsystem

Windows Azure[23] är datormolnstjänstens operativsystem som hanterar utveckling, tillhandahållandet av tjänster och tjänstehanteringsmiljön för Windows Azure plattformen. Windows Azures operativsystem ser till att du kan beräkna, lagra, skala och hantera webbapplikationer och tjänster via internet på Microsofts datacenter.

2.3.2 Windows Azure-plattformen AppFabric

Windows Azure plattformen AppFabric[21] är grundlaget till hela Windows Azure. Plattformen består av en uppsättning av Microsofts tillhandahållna, skalbara och utvecklingsorienterade tjänster som förser användaren med de byggstenar som krävs till molnbaserade applikationer. AppFabric ger applikationer en gemensam infrastruktur för att kunna namnge, upptäcka, exponera, säkra och ordna webbtjänster. Den ser också till att utvecklaren kan fokusera på sina applikationer istället för att själv behöva skapa sina egna miljöer för applikationerna. När kunden flyttar sin applikation till Windows Azure plattformen så kallar man det för att applikationen befinner sig off-premises då den inte längre befinner sig eller hanteras lokalt. Windows Azure plattformen ser till att alla datorberäkningar görs på Microsofts servrar och detta gör att plattformen använder sig av datormolnstekniken.

2.3.3 SQL Azure

Microsoft SQL Azure[23] utökar SQL Servers funktionalitet i molnet. SQL Azure erbjuder en relationsdatabastjänst som kallas Microsoft SQL Azure Database. Genom att använda SQL Azure kan du enkelt skapa och distribuera relationsdatabaslösningar. Fördelarna är bland annat hanterbarhet, hög tillgänglighet, skalbarhet, utveckling och relationshantering. Olikt andra molnbaserade databaser så tillåter SQL Azure relationsfrågor att ställas mot lagrad data, vilken antingen kan vara strukturerad eller semistrukturerad eller helt ostrukturerad

dokument. SQL Azure erbjuder även möjligheten att ställa frågor mot data, söka data, analysera data och synkronisera data. SQL Azure använder Microsoft SQL Server som grund men använder sig bara av en delmängd av de datatyper som finns. SQL Azure använder sig av ett XML-baserat format för dataöverföring. Likt MS SQL Server så använder sig SQL Azure av T-SQL¹ som sitt frågespråk och TDS (Tabular Data Stream) som protokollet för att få tillgång till tjänster över internet.

2.4 Compute Storage och Fabric

Windows Azure i sin helhet består av tre nyckeldelar; Compute, Storage och AppFabric. Nedan går vi in i på detalj om dem och deras uppbyggnad.

2.4.1 Windows Azure Compute Service

Windows Azure Compute Service[9] är värdmiljön som tar hand om exekvering och beräkningar för datormolnsapplikationer som implementerar en eller flera roller och tjänsten kan köra flera instanser av var och en av dessa roller. En roll är ett projekt som är del av datormolntjänsten som en användare kan skapa i Visual Studio och den utför olika sorters uppgifter beroende på vad den är definierad som.

Roller i Windows Azure

Det finns två typer av roller för ett molnprojekt. De två rollerna som kan definieras är Web och Worker – roller. Rollerna bestäms av programmeraren när man skapar en Cloud Service i Visual Studio. En Cloud Service är en speciell typ av projekt man kan skapa med hjälp av Windows Azure SDK(se avsnitt 3.1) i Visual Studio. Cloud Service används för att göra ett vanligt webbprojekt gjort i Visual Studio till ett molnprojekt.

Web Role

Web Role[9] är en roll som är specialiserad för webbapplikationsprogrammering. Webrollernas applikationer lyssnar och svarar på webbförfrågningar via HTTP eller HTTPS-ändpunkter. Ett exempel på en webbroll skulle kunna vara gränssnittet för en gästbok och logiken för gränssnittets funktioner.

¹ T-SQL[33] är en förlängning av frågespråket SQL, specifikt utformat för att hantera kommunikationen mellan en applikation och en SQL Server databas.

Worker Role

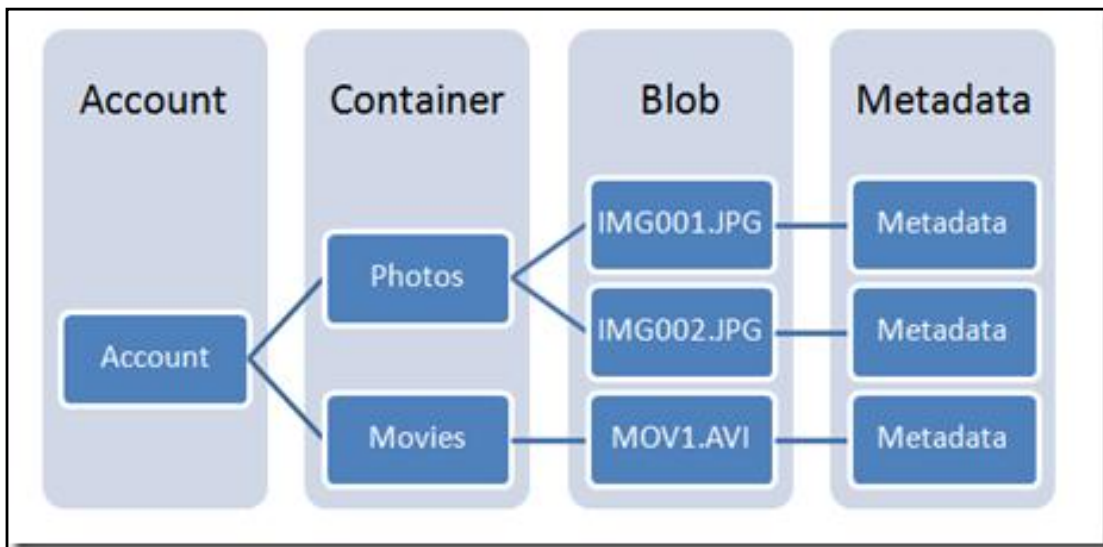
Worker Role [9] är en roll som är användbar för generell utveckling. Workerroll-applikationer körs som bakgrundsprocesser, oftast åt en webroll. En workerroll exponerar inga HTTP eller HTTPS - ändpunkter och kan därför inte nås via internet. Om vi fortsätter vårt exempel med gästboken, så kunde en workerroll vara logiken för databasklassen till gästboken som tar indata från webrollen och lagrar det i en databas.

2.4.2 Windows Azure Storage Service (Storage)

Windows Azure Storage Service[9] erbjuder ett lagringskonto. Man kan skapa ett sådant lagringskonto via Windows Azure Web Portal Interface. Användaren kommer då att mottaga en säkerhetsnyckel som används för att autentisera sig till lagringssystemet. Lagringskontot erbjuder tre olika typer av lagring, vilka vi beskriver nedan.

Blob Storage

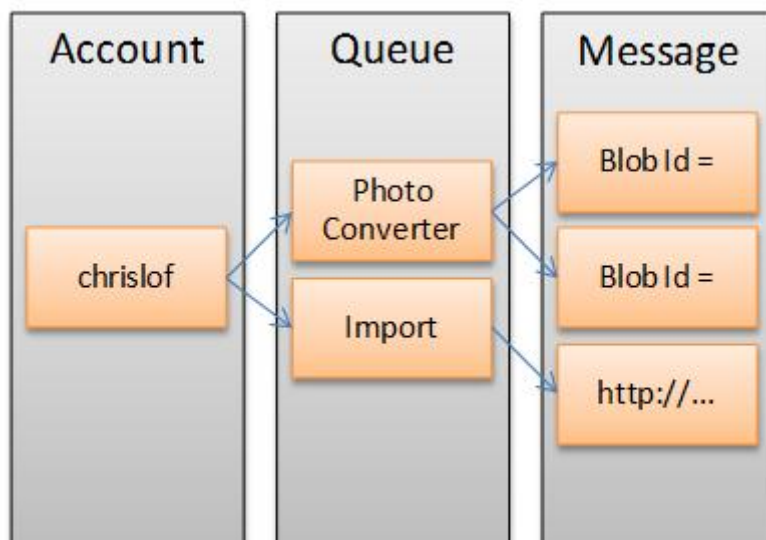
Ordet Blob i Blob Storage[10,11,12] står för Binary Large Object och är en samling av binär data lagrad som en enda enhet. Oftast så innehåller blobbar bilder, ljud eller andra objekt av media-typ, även om de ibland innehåller binär kod. Windows Azure lagrar dessa blobbar i en tjänst som kallas för Blob storage. Varje lagringskonto har tillgång till ett blobblagringskonto(se figur 2-4). För varje blobkonto kan det existera ett oändligt antal containers. Containers är behållare för blobbar och de kan vara antingen publika eller privata. I publika containers så kan URL:en till blobben nås via internet medan i en privat container så kan endast ägaren av kontot få tillgång till blobbens innehåll. För varje blobb kan man även lägga till metadata i form av ett namngivet värde.



Figur 2-4 Översiktsbild för användning av blobblagring

Queues

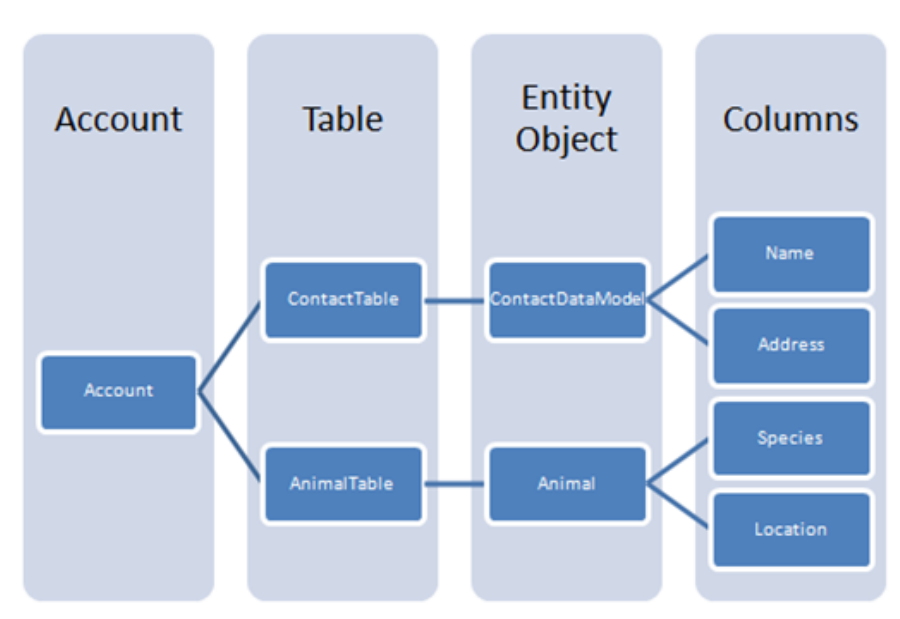
Windows Azure Queues[17,18] är ett datalagringskoncept inom Windows Azure som handlar om köhantering (illustrerat i figur 2-5). Köer inom detta koncept används som en pålitlig meddelandeutdelningsmekanism. Kömeddelanden innehåller instruktioner som skickas till en lyssnare som utför dessa instruktioner. Denna lyssnare kallas för en Worker process. Meddelanden kan vara max 8 kB stora, vilket gör att man inte kan skicka större filer eller data än så. Detta gör att den tyngre delen av beräkningarna och uppdateringarna kommer att ligga hos Worker processerna. Som ett exempel på användandet av köer så kan en identifierare(adress) till en bild som ligger på ett lagringskonto skickas med i meddelandet, och Worker processen kan sedan läsa bilden från identifieraren på kontot. Köer kan användas för att kontrollera skalbarheten genom att mäta hur stor beräkningskapacitet som din applikation har för tillfället. Om det är för många meddelanden i din kö så betyder det att du inte har tillräckligt med Worker processer som arbetar av dem. Ifall det är för få meddelanden i kön då kan du istället skala ner antalet Worker processer. Köer tillhandahåller också en buffert för att kunna absorbera stora trafikutbrott som kan uppstå till din applikation. Detta gör att en applikation som använder köerna kommer vara skyddad mot dataförluster och fortfarande vara tillgänglig under utbrottet.



Figur 2-5 Översiktsbild för användning av köhantering

Table Storage

Table Storage[14,15](se figur 2-6) innehåller en mängd av entiteter. Man kan skapa flera olika tabeller på sitt lagringskonto. Inom varje tabell kan du skapa entiteter och varje entitet har en mängd olika egenskaper du kan välja mellan. Varje egenskap representeras som en kolumn i en tabell. Tabellagring är ett sorts primitivare sätt att lagra data jämfört med att lagra det i en databas.



Figur 2-6 Översiktsbild för användning av tabellagring

2.4.3 AppFabric

Som nämnades i avsnitt 2.3.2 är Windows Azure plattformen AppFabric värdmiljön i Windows Azure. AppFabric gör det möjligt för användaren att på ett enkelt sätt bygga och hantera webbapplikationer både lokalt och i datormolnet. Vi går nedan igenom AppFabric's olika beståndsdelar mer i detalj; Windows Azure Fabric Controller (avsnitt 2.3.11) som tar hand om själv värdmiljön i Windows Azure samt AppFabric Service Bus(avsnitt 2.3.12) som är kopplingen mellan applikationer och resurser och AppFabric Access Control(avsnitt 2.3.13) som autentiserar uppkopplingen.

Windows Azure Fabric Controller

Fabric controller är en tjänst som har hand om själva värdmiljön i Windows Azure. Den övervakar, uppehåller och tillhandahåller maskinresurser till applikationer som utvecklaren skapar och lagrar i Microsofts datormoln. Fabric controllern har hand om hanteringen av alla noder och länkar i Azure Fabric. Denna är i grund och botten servrar, switchar och routrar vilka tillsammans skapar ett nätverk. Detta nätverk kan automatiskt hantera resurser, lastbalansering, geo-replication(hur data är lagrad geografiskt i nätverket) och applikationens livslängd. Skalbarhet och pålitlighet kontrolleras av Windows Azure Fabric Controller så att tjänsterna och miljön inte kraschar om en av serverna på Microsoft datacenter skulle göra det. Den tillhandahåller också resurser för användarens applikation, såsom primärminne.

Service Bus

Servicebussen[27] tar hand om kopplingen mellan användarens applikation och resurserna som applikationen använder i molnet, oavsett vilken plattform som används. Servicebussen är en kommunikationsinfrastruktur som innebär att utvecklaren slipper skapa den komplexa kod som behövs för att kunna skapa en perfekt sammanhängande och logisk koppling mellan nätverken. Den tillåter dig också att kunna exponera en tjänst på internet, även om du sitter bakom en brandvägg eller NAT-router. Den har även stöd för peer-to-peer kommunikation åt båda hållen, publicering samt att prenumerera, lagra och vidarebefordra meddelanden.

Access control

Access Control-tjänsten hanterar vem som får koppla upp sig till användarens applikation. Den gör det oavsett vilket sorts identifieringssystem som används av den som försöker koppla upp sig. Genom Access Control så kan Azure Storage Service använda sig av offentlig åtkomstkontroll. Denna åtkomstkontroll tillåter allmänna läsrättigheter, till exempelvis en

container och till blobben som finns i den, eller allmän läsrättighet till endast blobben och inte själva containern. Det senare skall förhindra icke autentiserat listande av alla blobbar i containern. Azure Storage Service stödjer också delade tillträdessignaturer som kan användas för att ge en tidsbegränsad token som tillåter ickeautentiserade användare en temporär möjlighet att kunna komma åt containern eller blobbarna. Access Control[27] gör att man kan samla tillträdeslogik i hanterbara samlingar av regler och automatisera föreningen mellan olika identifieringsleverantörer.

2.4.4 Betalningsalternativ för Windows Azure

Det finns två typer av erbjudande när man väljer att använda Windows Azure. Den första typen kräver inget åtagande utan man betalar endast för det man använder och den andra typen är en tidsbestämd prenumeration där man betalar en månadsavgift. Man får då till ett fast pris ett visst begränsad utrymme och trafik. Skulle man använda mer än denna begränsning betalar man enligt standardförbrukningsmodellen.

2.5 Summering

I detta kapitel har vi beskrivit AMM-applikationen som är ett IT-arkitekturellt mognadstest som görs via webben. Vi har även förklarat begreppet datormoln som består av två delar, Software As a Service och Utility Computing. Software As a Service är applikationer som levereras som tjänster över internet. Utility Computing betyder att resurser förbrukas efter behov. Efter det beskrev vi Windows Azure och dess olika beståndsdelar. Windows Azures nyckeldelar är Compute Service, Storage och Fabric. Compute Service tar hand om exekveringen av koden i applikationen. Storage lagrar data som blobbar, köer och tabeller och Fabric övervakar, uppehåller och tillhandahåller maskinresurser. Vi gick även igenom Windows Azures operativsystem, vilket består av en samling av tjänster som underlättar utvecklingen av molnapplikationer i Microsoft Azures datormoln. Operativsystemet utgör länken mellan Windows Azures maskinvara och de tillämpade program som användaren vill köra i molnet. Vi gick även igenom SQL Azure som är Windows Azures databashanteringstjänst.

3 Migrering och utveckling av applikationen

I det här kapitlet berättar vi om de praktiska delarna av vårt examensarbete. Vi beskriver i avsnitt 3.1 de krav vi fick av arbetsgivaren, sedan i avsnitt 3.2 berättar vi om arbetsmiljön som vi byggde upp för att kunna utföra arbetet. I avsnitt 3.3 tar vi upp molnprojekt i Visual Studio. I avsnitt 3.4 beskriver vi hur vi implementerade de förbättringar som önskades på AMM-applikationen och hur det såg ut innan respektive efter våra förändringar, d vs. hur våra förändringar påverkade applikationen. Sedan i avsnitt 3.5 går vi igenom överföringen av applikationen till datormolnet med hjälp av Windows Azure, vad som krävdes och hur vi utförde det. Sist tar vi upp hur vi skapade flera versioner av applikationen för olika företag i avsnitt 3.6.

3.1 Krav

De krav vi fick från Sogeti var att lägga upp applikationen AMM på Windows Azures datormoln, hur vi gjorde det var upp till oss. Vi skulle också migrera den databas applikationen hade tidigare från SQLite till SQL Server för att få den att fungera i molnet. Vi fick också kravet att kunna skapa flera versioner av applikationen genom att trycka på en knapp. Hur vi löste detta tas upp i kap 3.6. Som del av arbetet fick vi några krav på förbättringar som skulle göras för applikationen, dessa listas nedan.

Förbättringar

- Inget defaultsvar ikryssat i testet (se 3.4.1).
- Kunna släcka ner ”vet ej” och ”delvis” alternativen i testet (se 3.4.2)´.
- Kunna släcka ner dropdownlistor i testet (se 3.4.3).
- Inloggning per användare (se 3.4.4).
- Flytta användarinformation från rapport till användare (se 3.4.5)
- Krypterat lösenord för användare och administratören´(se 3.4.6)
- Mindre förändringar (se 3.4.7)

3.2 Utvecklingsmiljö

För att utföra projektet behövde vi först bygga upp en utvecklingsmiljö. De olika delarna som ingick i miljön förklaras nedan.

Windows 7

Windows 7 eller Windows Vista krävdes för att köra Windows Azure. Vi valde att använda oss av det förstnämnda alternativet.

MSDN-Konto

Ett MSDN-konto med licens behövdes för att få tillgång till Microsoftprogram.

Visual Studio 2010

Visual Studio 2010 är utvecklingsmiljön som vi använde för att utveckla applikationen skriven i Visual C#. Vi har också använt Visual Studio 2008 men uppdaterade senare till Visual Studio 2010, eftersom vissa funktioner som behövdes inte fanns i version 2008.

.Net Framework 3.5 Service Pack 1

För att kunna skapa datormolntjänster i Visual Studio behöver man också .Net Framework[28]. .Net Framework är en systemkomponent för Windows vilken består av en samling mindre komponenter. Dessa komponenter hanterar exekvering av program som är skrivna för speciella ramverk. .Net Framework består också av ett stort klassbibliotek

innehållande lösningar på vanliga programmeringsuppgifter. För att kunna skapa molntjänster krävdes därför version 3.5 och SP1 .

Azure SDK for Visual Studio

Azure SDK[29] för Visual Studio är en uppsättning utvecklingsverktyg som gör det möjligt för mjukvaruutvecklare att kunna utveckla datormoln-applikationer i Visual Studio.

SQL Server Express 2009 IIS 7.0 R2

SQL Server[35] är ett databashanteringsprogram. Denna version krävs för att kunna lägga upp och modifiera databaser på SQL Azure.

Konto på Windows Azure

Vi behövde ett konto på Windows Azure för att kunna logga in på Windows Azures hemsida för att få tillgång till Windows Azure plattformen och operativsystemet.

SQL Azure Migration Wizard

SQL Azure Migration Wizard[31] hjälper dig att migrera dina lokala SQL Server 2005/2008 databaser till SQL Azure. Programmet leder användaren genom val av SQL objekt och skapar SQL-skript som lämpar sig för SQL Azure och migrerar data.

Cloudberry Explorer for Azure Blob Storage

CloudBerry Explorer[30] gör hantering av filer på blobblagringskonton lättare. Genom att erbjuda ett gränssnitt till mot lagringskonton och filer gör detta program att du kan hantera dina filer på datormolnet precis som du gör på din egen dator.

3.3 Molnprojekt för Windows Azure

När man arbetar med molnprojekt för Windows Azure så använder man sig av Windows Azure SDK för Visual Studio som vi nämnt om i utvecklingsmiljön avsnitt 3.2. I Visual Studio kan man då skapa Cloud Service projekt. I detta projekt kan man antingen skriva koden som en Web- eller Workerroll eller så associerar man en roll med ett befintligt projekt. Detta kallas för ett molnprojekt.

Ett molnprojekt är konfigurerat med två XML-filer: Service Definition file och Service Configuration file. Dessa filer skapas när man publicerar molnprojektet i Visual Studio. Detta måste man göra innan man kan lägga upp applikationen i molnet. Tjänstdefinitionsfilen

definierar rollernas tillgång till en tjänst, anger tjänstens ingångspunkter, och fastställer konfigurationsparametrar för tjänsten. Tjänstens definitionsinställningar kan inte ändras när en tjänst är i bruk. Tjänstkonfigurationsfilen anger värden för konfigurationsinställningar för en eller flera roll-instanser för tjänsten. Du kan dynamiskt ändra i tjänstkonfigurationsfilens inställningar utan att behöva lägga upp hela tjänsten igen.

3.4 Förbättringar av AMM-applikationen

Vi går i detta avsnitt igenom de förbättringar vi gjorde i applikationen. För varje förbättring ges först en beskrivning av hur applikationen fungerade innan förändringen och sedan en beskrivning hur applikationen fungerade efter vår förändring.

3.4.1 Inget defaultsvar ikryssat i testet

Innan: När alla frågor för AMM-testet listades fanns det ett defaultsvar ikryssat för varje fråga.

Efter: Inga frågor har ett defaultalternativ ikryssat. Vi har lagt till så att när man trycker på någon av knapparna 'spara utkast' eller 'skicka rapport' och man missat att svara på någon av frågorna kommer det att uppenbara sig röd färg runt de frågor man inte svarat på. När man svarat på alla frågor kan man skicka rapporten.

3.4.2 Kunna släcka ner ”vet ej” och ”delvis” alternativen i testet.

Innan: När man skapade ett mognadstest hade man alltid svarsalternativen ”Vet Ej”, ”Delvis/ beror på”, ”Nej” och ”Ja”.

Efter: När man skapar och ändrar en rapportperiod ska man kunna lägga till eller ta bort svarsalternativen ”Vet Ej” och ”Delvis beror på”. Vårans uppdragivare ville att kunna bestämma om dessa svarsalternativ ska vara med i AMM-applikationen eller inte.

3.4.3 Kunna släcka ner dropdownlistor i testet

Innan: Förut fanns alla dropdownlist-alternativ med i testet även om de behövdes för användaren eller inte.

Efter: Nu kan man välja att släcka eller öppna de tre olika dropdownlistorna.

3.4.4 Inloggning per användare

Innan: Administratören var förut den enda med behörighet att lägga till användare. När en användare lades till angav man för- och efternamn för att identifiera en användare. Användare som skulle utföra mognadstestet placerades sedan i en rapportperiod av admin. För att logga in som användare användes ett allmänt lösenord som var samma för alla användare.

Efter: Administratören anger information om användaren som för- och efternamn, användarnamn och lösenord. När användaren ska logga in anger han sitt användarnamn och det tilldelade lösenordet. Det allmänna lösenordet har byts ut mot ett lösenord bundet till en användare.

3.4.5 Flytta användarinformation från rapport till användare

Innan: Via användargränssnittet kan administratören lägga till användare. När man lägger till nya användare anger man användarnamn, förnamn, efternamn och lösenord. I användartabellen i databasen fanns endast användarinformation som användarnamn, förnamn, efternamn, lösenord och en boolesk variabel som anger om användaren blivit borttagen eller inte. När en användare utförde AMM-testet fick användaren ange annan användarinformation som år inom arkitektur, år inom organisationen, division, roll och annan dropdownlist-information. Den här informationen tillsammans med informationen från användartabellen lagrades sedan i rapporttabellen efter att man rapporterat som användare. Med andra ord fanns det användarinformation i rapporttabellen som inte fanns bunden till en användare i användartabellen.

Efter: Applikationen strukturerades om så att all användarinformation sparades hos användarens användartabell i databasen. Den information som användaren angav när han gjorde testet flyttades och utfördes istället av administratören. Administratören fick då lägga till all användarinformation i användargränssnittet (se figur 4-9) som sedan sparas i användartabellen i databasen. Skillnaden är nu att administratören lägger till all användarinformation istället för att användaren ska ange viss information när användaren ska rapportera. All användarinformation som finns i rapporttabellen finns bunden till en användare i användartabellen.

När man lägger till en ny användare anger man förutom användarnamn, förnamn, efternamn och lösenord även dropdownlist-information, division, roll, år inom organisationen och år

inom arkitekturen. Vi skapade även ett GUI för att ändra på befintlig användarinformation. I GUI't kan man ändra all information i användartabellen.

3.4.6 Krypterat lösenord för användare och administratören

Innan: Förut sparades alla lösenord i klartext. Adminlösenordet var sparat i webconfig².

Efter: Alla lösenord sparas krypterade med hjälp av krypteringsalgoritmen MD5. Lösenord för användaren sparas tillsammans med all annan användarinformation i användartabellen i databasen. Adminlösenordet sparas nu i en egen tabell som heter AdminSecurity.

3.4.7 Mindre förändringar

- Vi har lagt till "gå bakåt"-knappar.
- Vi har sett till att rapporterade personer försvinner om man går bakåt från resultatformuläret till testformuläret.
- Vi har lagt in labels som visar vilka företag som är inloggade.
- I Testformuläret och i resultatformuläret har vi lagt till blockering av texttrutor där man förut kunde ändra information. Istället hämtas nu rätt information direkt från databasen.
- När man är inloggad som användare kan man välja rapportperiod. Förut kunde man endast visa en rapportperiod åt gången.
- Vi har lagt till utloggningssknappar för sessionerna. Är man inloggad som en administratör eller användare ska man kunna logga ut.

3.5 Migreringen av AMM till molnet

I det här avsnittet går vi igenom steg för steg hur vi migrerade applikationen till molnet. Vi börjar med hur vi bytte databas, därefter förklarar vi applikationens integration med databasen och sedan hur vi överförde SQL server-databasen till SQL Azure. Efter det förklarar vi hur vi gjorde applikationen till ett molnprojekt och slutligen förklarar vi publiceringen av applikationen i molnet.

² Webconfig är en XML-fil som alla C# webprojekt får i Visual Studio, vilket innehåller uppkopplingssträngar och annan information som applikationen använder på nätet.

3.5.1 Steg 1: Byta databas

AMM-applikationen använde en SQLite-databas från början. SQLite-databasen behövde bytas till en SQL Server 2008 databas för att kunna fungera på SQL Azure. För att göra detta behövde vi föra över schemat från SQLite-databasen till en ny SQL Server-databas, samt föra över data från tabellerna.

Genomförande: Vi använde oss av programmet SQLite Data Wizard[34] för att kopiera över schemat från databasen. Eftersom det inte var ett gratisprogram bytte denna demoversion ut första bokstaven på alla poster till bokstaven T, vilket gjorde att vi bara kunde kopiera över schemat och inte data. För att kopiera över alla poster från SQLite-databasen till SQL Server-databasen importerade vi alla poster till datatabeller med hjälp av SQL-förfrågan i Visual Studio och sedan exporterade vi informationen ifrån datatabellerna till motsvarande tabell i SQL Server-databasen.

3.5.2 Steg 2: Få SQL Server-databasen att fungera med applikationen.

Eftersom SQLite och SQL Server fungerar på olika sätt behövdes ändringar göras i koden för att få SQL Server-databasen att fungera med applikationen. Utöver detta ingick testning av applikationens funktionalitet för att se att allting fungerade som förut med den nya databasen.

Genomförande : Vi bytte ut all SQLite-relaterad kod till SQL-Server-relaterad kod i applikationen. Sedan testade vi för att se att allting fungerade.

3.5.3 Steg 3: Föra över SQL Server - databasen till SQL Azure.

Vi hade nu bytt ut en SQLite-databas till en SQL Server-databas och testat så att den nya databasen fungerade med applikationen. Nu skulle vi testa om applikationen fungerade med en databas i molnet. För att testa detta så behövde vi föra över SQL Server databasen till SQL Azure.

Genomförande: Vi använde oss av SQL Azure Migration Wizard för att lägga upp en kopia av databasen på SQL Azure. Via Windows Azure webbportal kunde vi se att en kopia av databasen hade lagts upp i datormolnet. Via webbportalen kunde vi sedan ta fram uppkopplingssträngen till den nya SQL Azure-databasen. Vi bytte sedan ut den lokala

uppkopplingssträngen mot uppkopplingssträngen vi fått specificerad av SQL Azure. Vi testade detta och applikationen fungerade nu med en databas placerad i datormolnet.

3.5.4 Steg 4: Lägga till applikationen i ett molnprojekt med en webbroll.

För att kunna publicera applikationen i molnet behöver applikationen som är gjord i Visual Studio vara en molntjänst med en WebRole kopplad till AMM-projektet.

Genomförande: Vi skapade en molntjänst med en WebRole. Vi kopplade sedan Webrollen till huvudprojektet i AMM-applikationen. För att lägga upp applikationen på Windows Azure behövde vi göra en publicering av applikationen. Publicering är en funktionalitet som gör att hela projektet packas ner i en fil, vilken man sedan kan ladda upp på Windows Azure. När man publicerar kommer även en konfigurationsfil att skapas. Dessa två filer behövs sedan för att kunna lägga upp applikationen på Windows Azure.

3.5.5 Steg 5: Publicera Applikationen i molnet.

Med hjälp av Windows Azure ska applikationen publiceras i ett datormoln. Sedan skulle applikationen testas i datormolnet för att fastställa att allting fungerar som det skulle.

Genomförande: Genom Windows Azures webbportal publicerade vi paketfilen och konfigurationsfilen. Applikationen får då en tilldelad URL med angivet namn som man kan anropa via sin webbläsare. Efteråt testade vi applikationen genom att ange den tilldelade adressen i en webbläsare. Av testet kunde vi fastställa att applikationen fungerade i datormolnet.

3.6 Skapa flera versioner av applikationen

Varje gång ett företag ville använda sig av AMM-applikationen kopierades förut en lokal version av och lades sedan upp på Sogetis egna servrar med en unik URL-adress.

Vi skulle på ett enkelt sätt genom exempelvis en knapptryckning kunna skapa en egen version av webb-applikationen bundet till ett företag. Tex om företag A och B vill använda sig av applikationen så ska båda kunna använda sig av egna versioner av applikationen och versionerna ska inte kunna påverka varandra.

Detta utförde vi genom att skapa en ny databas för varje företag som skulle använda sig av applikationen och koppla den till det valda företagets databas vid inloggning. När ett annat företag vill använda sig av applikationen anger företaget sig själv och sedan kopplas AMM-applikationen mot det angivna företagets databas.

3.6.1 Steg 1: Lägga till/ta bort ett företag.

Vi skapade ett GUI, i vilket man kan lägga till ett företag som ska använda sig av applikationen och ta bort ett företag som inte behöver den längre. När man lägger till ett nytt företag till applikationen genereras en ny databas till AMM-applikationen som ska vara unikt för det företaget. För denna funktion skapade vi ett GUI där man med lösenord kunde logga in som en administratör (endast administratören har behörighet att lägga till företag och ta bort företag). I detta GUI listades alla företag man hade lagt till och även databaskopplingssträngen till det specifika företaget. Gränssnittet erbjuder även funktionalitet för att modifiera databaskopplingssträngen för ett valt företag.

Genomförande: Vi skapade en defaultdatabas till applikationen, vilken endast fick innehålla generell information som var likadan för alla företagsdatabaser. Databasen fick endast innehålla information som applikationstexter och frågor, annan information som användare och rapportperioder togs bort. Defaultdatabasen användes för att kopieras och kopplas till ett företagsnamn.

För att göra en kopia av defaultdatabasen krävdes följande:

- Generera ett T-SQL-skript för defaultdatabasens schema.
- Skapa ett ny tom databas utifrån de genererade skriptet under ett nytt valt företagsnamn.
- Kopiera över all data från defaultdatabasen till den nya tomma databasen.
- Spara databaskopplingssträngen till databasen i en XML-fil på blobblagringskontot.

3.6.2 Steg 2: Logga in som ett företag

Vi kunde nu skapa nya databaser under valt företagsnamn. Applikationen behövde nu kunna identifiera vilket företag som för tillfället ska använda sig av applikationen och koppla applikationen mot företagets databas. Utöver detta behövde vi kunna logga in som företag och koppla applikationen till motsvarande databas.

Genomförande: Vi skapade ett GUI för inloggning av företag. I GUI:t finns även en länk som tar dig till admingränssnittet där du kan lägga till/ta bort företag. I det grafiska gränssnittet finns en dropdownlist som listar alla tillgängliga företag. När man väljer vilket företag man ska logga in så hämtar man databaskopplingssträngen som finns lagrad på ett

blobblagringskonto och lagrar den som en session. Databaskopplingssträngen i sessionen används sedan för att veta vilken databas man ska koppla upp sig mot vid inloggning.

Man kan också logga in som ett företag via en URL. Detta gör man genom att ange en URL med en frågesträng. Frågesträngen definierar vilket företag du är inloggad som. Genom att ange vilket företag man vill logga in som i frågesträngen hämtas databaskopplingssträngen från det angivna företaget. Strängen lagras i en session och sessionen används sedan för att få applikationen att jobba mot rätt databas.

3.7 Summering


I detta kapitel gick vi först igenom de krav som ställdes på oss av vår uppdragsgivare. Sedan gick vi igenom den arbetsmiljö vi byggde upp och program vi behövde för att utföra arbetet. Sedan gick vi igenom de förbättringar som vi gjorde i AMM-applikationen. För att få applikationen att fungera i molnet gick vi igenom migreringen av databasen från SQLite till SQL Server och slutligen migreringen till SQL Azure. Applikationen måste även konverteras från en vanlig webbapplikation till ett molnprojekt. Väl uppe i molnet utförde vi molnrelaterade förbättringar för att göra applikationen användarvänlig på det sätt som vår uppdragsgivare önskat sig. Applikationen skulle på ett enkelt sätt kunna ha olika versioner beroende på vilket företag det är som ska använda sig av den .

4 Applikationsöversikt

I detta kapitel beskrivs hur applikationen ser ut idag.

Första sidan är företagsinloggningssidan (se figur 4-1). Här kan man välja vilket företag som man ska logga in som. Genom länken Admin Login kommer man till inloggningssidan för företagsadmingränssnittet.

DYA® ARCHITECTURE MATURITY MATRIX

Area	Level	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Defining architecture			A			B			C						
Use of architecture				A			B				C				
Involvement of business			A				B				C				
Involvement of development process				A				B		C					
Involvement of operations					A			B			C				
Relation to current situation					A				B						
Roles and responsibilities				A		B					C				
Coordination of activities							A			B					
Controlling				A		B			C		D				
Quality management								A	A	B		C			
Maintenance architecture process								A	B		C				
Maintenance architecture deliverables						A			B				C		
Commitment and motivation				A				B		C					
Architecture roles and training					A		B			C			D		
Use of architecture method					A					B					C
						A			A				B		C

Choose Company:

[Admin Login](#)

Figur 4-1 Company login

Här (se figur 4-2) loggar företagsadmin in för att ändra uppkopplingssträngar och lägga till och ta bort företag.

Password :

Figur 4-2 Company Admin login

I företagsadmingränssnittet: Här är gränssnittet för att lägga till företag, ta bort företag och modifiera databasuppkopplingssträngen (se figur 4-3).

CompanyName

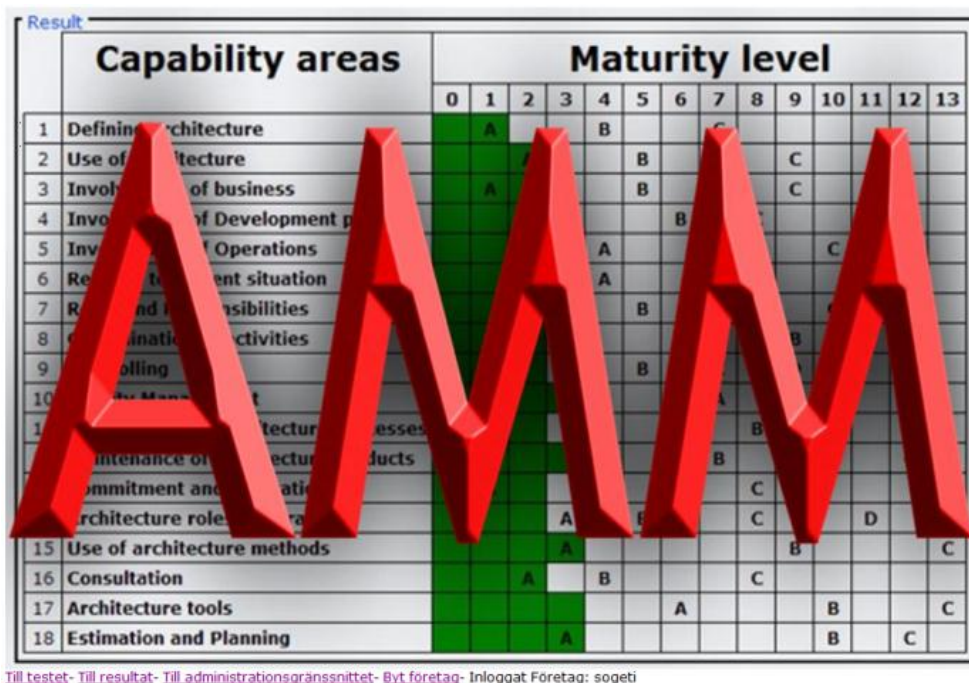
Change/Delete Company

DBPath

[Back](#)

Figur 4-3 Company Admin-gränssnittet

När man loggat in som ett företag tas man till AMM-applikationens första sida (se figur 4-4) som nu är bunden till det inloggade företags databas. Här finns länkar till testet, resultat, admin-gränssnitt och byt företag som tar dig tillbaka till företagsinloggningssidan. Applikationen visar även vilket företag som du är inloggad som



Figur 4-4 AMM startsidan

Klickar man in sig till testet på föregående sida tas man till en inloggningssida (se figur 4-5) där användaren som vill utföra AMM-testet måste logga in med användarnamn och lösenord. Användaren måste finnas tillagd av admin för att kunna logga in

Figur 4-5 Admin login

Inom testet: När man loggat in som en användare kommer man till denna sida där man svarar på alla frågor som rör AMM-testet (se figur 4-6). Användaren kan skriva kommentarer för varje fråga och ladda/spara utkast av sin inmatning.

Id	Question	Answer	Comment
1	Utvecklas arkitektur enbart om det finns en sponsor?	<input type="radio"/> Nej <input type="radio"/> Ja	
2	Är projektplanen för arkitekturprojekt granskad innan arbetet startas?	<input type="radio"/> Nej <input type="radio"/> Ja	
3	Är utveckling av arkitektur betraktat som en kontinuerlig process?	<input type="radio"/> Nej <input type="radio"/> Ja	
4	Uppdateras arkitekturen regelbundet?	<input type="radio"/> Nej <input type="radio"/> Ja	
5	Finns det någon form av version- och konfigurationshantering när det gäller arkitekturprodukter?	<input type="radio"/> Nej <input type="radio"/> Ja	
6	När ett sponsorskap etablerats innan arkitekturprodukter utvecklas?	<input type="radio"/> Nej <input type="radio"/> Ja	
7	Är det fastställt vem som tjänar på resultatet innan utveckling av arkitektur startas?	<input type="radio"/> Nej <input type="radio"/> Ja	
8	Är det andra än arkitekter involverade i utveckling av arkitektur	<input type="radio"/> Nej <input type="radio"/> Ja	

Figur 4-6 AMM-Testet

Resultatet av testet: Efter att man utfört testet tas man till resultatdelen (se figur 4-7) vilken visar en mognadsmatrix av dina svar. Resultatdelen erbjuder filtreringsalternativ.

ARCHITECTURE MATURITY MODEL

-- Rapport --

Filtrering

Rapportperiod:

Namn:

År inom organisationen:

År inom arkitektur:

Division:

Roll:

...:

Välj vy (Min, Medel, Max):

Resultat

	Fokusområden	Mognadsnivå													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	Framtagande av arkitekturprodukter	A			B			C							
2	Användning av arkitektur		A			B				C					
3	Verksamhetens medverkan		A			B				C					
4	Integration med utvecklingsprocessen		A				B		C						
5	Medverkan från drift och förvaltning				A			B			C				
6	Relation till aktuell situation				A				B						
7	Roller och ansvar			A		B					C				
8	Koordinering av aktiviteter						A			B					
9	Arkitekturkontroll				A		B		C		T				
10	Arkitekturkvalitet							A		B				C	
11	Underhåll av arkitekturprocessen						A		B		C				
12	Underhåll av arkitekturleverabler				A			B						C	
13	Motivation, förpliktigande och mandat		A					B		C					
14	Arkitekturroller och utbildning				A		B			C			T		
15	Användning av arkitekturmetod				A					B					C
16	Arkitekturrådgivning			A		B				C					
17	Arkitekturverktyg						A				B				C
18	Planering av arkitekturarbetet				A						B			C	

Figur 4-7 Resultat

Inom administratörssidan som man kommer till genom länken på förstasidan, finns följande funktioner.

I admingränssnittet: Lägga till användare (se figur 4-8). Informationen om användaren kommer att användas i testet.

ARCHITECTURE MATURITY MODEL Administrati

[< - Lägg till namn](#) - [Ändra på användarinformation](#) - [Hantera DropDownList1](#) - [Hantera DropDow](#)
[rapportperiod](#) - [Hantera frågeställningar](#) - [Ändra lösenord](#)

Lägg till namn

Användarnamn	<input type="text"/>
Lösenord	<input type="text"/>
Lösenordet igen	<input type="text"/>
Förnamn	<input type="text"/>
Efternamn	<input type="text"/>
År inom organisationen	0 ▾
År inom arkitektur	0 ▾
Division	Verksamhetsstöd ▾
Roll	Verksamhetsutvecklare ▾
...	T ▾
	<input type="button" value="Lägg till"/>

Namn i databasen:

Johansson, Erik ▲

Östlund, Jonas ▼

Figur 4-8: Admingränssnittet: Lägga till användare

I **admingränssnittet**: Ändra användarinformation (se figur 4-9).

ARCHITECTURE MATURITY MODEL Adminis
[< - Lägg till namn](#) - [Ändra på användarinformation](#) - [Hantera DropDownList1](#) - [Hantera rapportperiod](#) - [Hantera frågeställningar](#) - [Ändra lösenord](#)

Välj en användare ▾

Användarnamn

Förnamn

Efternamn

År inom organisationen ▾

År inom arkitektur ▾

DropDown1 ▾

DropDown2 ▾

DropDown3 ▾

Figur 4-9 Admingränssnittet: Ändra användarinformation

I **admingränssnittet**: Lägg till information för dropdownlist (se figur 4-10). Samma sorts gränssnitt för dropdownlist 1, 2 och 3.

ARCHITECTURE MATURITY MODE

[< - Lägg till namn](#) - [Ändra på användarinformation](#) - [Hantera DropI](#)
[rapportperiod](#) - [Hantera frågeställningar](#) - [Ändra lösenord](#)

Hantera DropDownList1

Text

Texter i databasen

- Verksamhetsstöd
- IT-stab
- TI-Systemutveckling
- IT-support
- IT-drift
- Konsult
- Annan
- IT-förvaltning
- DIVISION I
- DIVISION F
- DIVISION K
- DIVISION D

Figur 4-10 Admingränssnittet: DropDown hantering

I admingränssnittet: Hantera ledtexter (se figur 4-11).

ARCHITECTURE MATURITY MODEL Ad

[< - Lägg till namn](#) - [Ändra på användarinformation](#) - [Hantera DropDownList rapportperiod](#) - [Hantera frågeställningar](#) - [Ändra lösenord](#)

Hantera ledtexter

ReportPeriod	<input type="text" value="Rapportperiod"/>	<input type="button" value="Uppdatera"/>
Name	<input type="text" value="Namn"/>	<input type="button" value="Uppdatera"/>
YearsInOrganisation	<input type="text" value="År inom organisationen"/>	<input type="button" value="Uppdatera"/>
YearsInArchitecture	<input type="text" value="År inom arkitektur"/>	<input type="button" value="Uppdatera"/>
DropDown1Title	<input type="text" value="Division"/>	<input type="button" value="Uppdatera"/>
DropDown2Title	<input type="text" value="Roll"/>	<input type="button" value="Uppdatera"/>
DropDown3Title	<input type="text" value="..."/>	<input type="button" value="Uppdatera"/>
ReportData	<input type="text" value="Rapportdata"/>	<input type="button" value="Uppdatera"/>
Questions	<input type="text" value="Frågor"/>	<input type="button" value="Uppdatera"/>
Submit	<input type="text" value="Skicka rapport"/>	<input type="button" value="Uppdatera"/>
ShowReport	<input type="text" value="Visa rapport"/>	<input type="button" value="Uppdatera"/>

Figur 4-11 Admingränssnittet: Hantera ledtexter

I admingränssnittet: Lägga till rapportperiod. Här läggs också till vilka personer som ska kunna rapportera för den perioden samt vilka alternativ det testet ska kunna visa för sina frågor(se figur 4-12). Man kan också välja vilka dropdownlistor som ska vara relevanta för denna rapportperiod.

ARCHITECTURE MATURITY MODEL Ad

[< - Lägg till namn](#) - [Ändra på användarinformation](#) - [Hantera DropDownList: rapportperiod](#) - [Hantera frågeställningar](#) - [Ändra lösenord](#)

Lägg till rapportperiod

Namn

Startdatum

Slutdatum

Ska dessa svarsalternativ visas i frågorna?

Delvis/Beror på

Vet inte

DD1

DD2

DD3

Figur 4-12 Admingränssnittet: Lägga till rapportperiod

I admingränssnittet: Hantera frågeställningar (se figur 4-13).

ARCHITECTURE MATURITY MODEL Administration					
< - Lägg till namn - Ändra på användarinformation - Hantera DropDownList1 - Hantera DropDownList2 - Hantera DropDownList3 - Hantera ledtexter - Lägg till rapportperiod - rapportperiod - Hantera frågeställningar - Ändra lösenord					
Hantera frågeställningar					
	QuestionId	QuestionGroupId	Level	SortOrder	Text_sv-SE
Edit	1	1	1	0	Utvecklas arkitektur enbart om det finns en sponsor?
Edit	2	1	1	0	Är projektplanen för arkitekturprojekt granskad innan arbetet startas?
Edit	3	1	2	0	Är utveckling av arkitektur betraktat som en kontinuerlig process?
Edit	4	1	2	0	Uppdateras arkitekturen regelbundet?
Edit	5	1	2	0	Finns det någon form av versions- och konfigurationshantering när det gäller arkitekturprodukter?
Edit	6	1	3	0	Har ett sponsorskap etablerats innan arkitekturprodukter utvecklas?
Edit	7	1	3	0	Är det fastställt vem som tjänar på resultatet innan utveckling av arkitektur startas?
Edit	8	1	3	0	Är det andra än arkitekter involverade i utveckling av arkitektur
Edit	9	2	1	0	Finns det något som ledningen betraktar som arkitektur?
Edit	10	2	1	0	Visar arkitekturen tydligt vad verksamheten behöver? Innehåller den delar som är verksamhetsutvecklingsfokuserade?
Edit	11	2	1	0	Är arkitekturen tillgänglig för alla anställda som behöver den?
Edit	12	2	2	0	Används arkitekturen för att pro-aktivt styra och vägleda verksamhets- och IT-utveckling?
Edit	13	2	2	0	Är projekten medvetna om alla delar av arkitekturen som berör dess arbete?
Edit	14	2	2	0	Tillhandahåller arkitekturen konkreta riktlinjer som är till nytta för projekten?
Edit	15	2	3	0	Används arkitektur i organisationens beslutsprocesser?
Edit	16	2	3	0	Är arkitekturarbetet en naturlig del av organisationens kontinuerliga planerings- och uppföljningsarbete?
Edit	17	2	3	0	Är arkitekturens underliggande koncept baserade på högsta ledningens vision?
Edit	18	3	1	0	Finns det tydliga kopplingar/samband mellan arkitekturen och organisationens verksamhetsmål?
Edit	19	3	1	0	Utvärderas arkitekturen gentemot verksamhetsmål?

Figur 4-13 Admingränssnittet: Hantera frågeställningar

I admingränssnittet: Uppdatera en rapportperiod (se figur 4-14).

ARCHITECTURE MATURITY MODEL Admin

[<](#) - [Lägg till namn](#) - [Ändra på användarinformation](#) - [Hantera DropDownList: rapportperiod](#) - [Hantera frågeställningar](#) - [Ändra lösenord](#)

Uppdatera rapportperiod

Vårtest ▼

Namn
Vårtest

Startdatum
1/1/2010

Slutdatum
12/12/2010

	>>	Johansson, Erik Östlund, Jonas
	<<	

Ska dessa svarsalternativ visas i frågorna?

Partly

Do Not Know

DD1

DD2

DD3

Figur 4-14 Admingränssnittet: Uppdatera rapportperiod

I **admingränssnittet**: Ändra lösenord för en användare/admin (se figur 4-15).

ARCHITECTURE MATURITY MODEL A

[<](#) - [Lägg till namn](#) - [Ändra på användarinformation](#) - [Hantera DropDown rapportperiod](#) - [Hantera frågeställningar](#) - [Ändra lösenord](#)

Ändra på lösenord för en användare

Jonas Östlund ▾

Nytt lösenord till frågeformuläret:

Bekräfta lösenordet till frågeformuläret:

Ändra på lösenord för admin

Nytt lösenord till admingränssnittet:

Bekräfta lösenord till admingränssnittet:

Figur 4-15 Admingränssnittet: Ändra lösenord

5 Resultat och utvärdering

I detta kapitel går vi igenom resultatet av vårt examensarbete, vår utvärdering om datormolnet, vilka problem som uppstod under projektets gång och vilket framtida arbete som kan göras.

5.1 Resultatet

Vad vi har skapat är en applikation som ligger i Windows Azure datormoln. Applikationen har egenskapen att kunna skapa flera databaser i molnet för varje kund som ska använda sig av den, samt att kunna koppla upp sig till databasen genom en unik uppkopplingssträng. Denna databasinstansiering gör att varje kund har en egen instans av applikationen, vilket är ett effektivare arbetssätt än att kopiera en lokal version av applikationen, tillhandahålla resurser och publicera den på en egen adress. Med vår metod behöver du bara ha en applikation och en defaultdatabas liggande på datormolnet och sedan görs databasinstansieringen automatiskt när du trycker på en knapp. Vårt arbete har också gjort att applikationen fungerar smidigare för uppladdning av information och har gjort den mer användarvänlig, med tex. en inloggning för användare som använder lagrad information istället för att man måste mata in ny information varje gång.

5.2 Utvärdering av datormolnet

Datormolnet är idag ett omdiskuterat ämne och många tror att datormolnet är framtiden medan andra är skeptiska. Vi ska nu diskutera fördelar och nackdelar om datormolnet.

En fördel är att det är lätt att komma igång med datormolnet. Windows Azure erbjuder träningspaket gratis och det finns mycket information att hitta på internet. Datormolnet erbjuder en användarvänlig utvecklingsmiljö för att publicera dina molnapplikationer och komma åt data. För företag är datormolnet ur ett ekonomiskt perspektiv lönsammare då du endast betalar för den mängd av data och trafik du använder. Med denna teknik överlåter vi driften till ett externt företag som i teorin erbjuder oändliga resurser. Detta innebär att molnet skapar en effektivare användning av IT både affärsmässigt och resursmässigt. När en tjänst, applikation eller ett system blir populärt kan man kostnadseffektivt skala upp respektive skala ner systemet. På grund av att vi överlåter driften till en annan kommer det att leda till att företagens egna serverhallar försvinner till molnet och den kommer istället att ersättas av enorma serverfarmar från stora leverantörer som Google, IBM, Amazon och Microsoft. Istället för drift kan då företag fokusera sina resurser på utveckling. Dessutom kommer driften att bli effektivare då enskilda företag inte kan erbjuda samma resurser som IT-jättarna. Genom att företag kan lägga mer energi på utveckling kommer detta att bidra till en starkare utveckling inom IT-världen. Internet är en växande kraft och applikationer kräver mer och mer idag en internetanslutning. Tanken med molnet är att använda internet till sin fördel. Utvecklingsmiljöer ska inte längre vara platsberoende utan överallt kunna komma åt via din internetuppkoppling och din webbläsare. Men en oro kring tekniken existerar. Många företag känner sig exempelvis inte trygga med att känslig information hanteras av ett externt företag. I många fall går detta helt emot företagets policy, vilket gör att lokala serverlösningar fortfarande kommer vara viktiga i framtiden. En annan nackdel är att man är beroende av en internetanslutning som ständigt måste fungera. Driftstörningar hos leverantören kan påverka den egna verksamheten vilket kan leda till stora ekonomiska förluster. Fortfarande är detta ett problem med molnet som användarna måste vara medvetna om. Vad kan vi lägga i molnet? Har vi råd att få ett driftstopp som vi själv inte kan påverka? Molnet kräver även en snabb uppkoppling för att det ska fungera effektivt. Trots dessa nackdelar är molnet fortfarande ett begrepp under utveckling. Fördelarna hos molnet väger starkare än nackdelarna och vi är övertygade om att molnet är framtiden. Den ständigt förbättrande uppkopplingshastigheten

samt ett ökande behov av att ständigt vara online är till molnets fördel. Molnet är enligt oss en framtida förutsättning för en effektivare användning av resurser.

5.3 Problem

I detta avsnitt beskriver vi några av de problem som uppstod under vårt arbete.

5.3.1 Migreringen (tillvägagångssättet)

När vi skulle migrera databasen från SQLite till SQL Server fick vi göra mycket sökande på internet innan vi kom fram till hur vi skulle göra. Det enda program vi hittade som kunde göra en migrering av SQLite var inte gratis och vi använde oss då av dess demoversion som hade en stor nackdel; den bytte nämligen ut första bokstaven i alla poster som fanns i databasen till T. Detta gjorde att vi bara kunde kopiera över schemat och inte data. För att kopiera över alla poster från SQLite-databasen till SQL Server-databasen importerade vi alla poster till datatabeller med hjälp av SQL-Queries i Visual Studio och sedan exporterade vi informationen ifrån datatabellerna till motsvarande tabell i SQL Server-databasen

5.3.2 Användarvänlighet i applikationen

När vi implementerade inloggning för användare så skapade detta problemet att vi behövde omstrukturera nästan hela applikationens sätt att hantera användardata.

Innan så lagrade man data för användaren med varje rapport. Nu var man tvungen att lagra den både i användartabellen och i rapportens tabell för att kunna bevara tidsaspekten (man vill t ex veta vad användaren hade för information vid det tillfället man gjorde testet, och inte vilken information den personen hade i dagens läge).

5.3.3 Vad molnet inte stödjer (DLL-filer)

SMO (Server Management Objects) är ett bibliotek för överföringen av data mellan databaser. Tanken var att vi skulle använda detta bibliotek för att kunna kopiera databaser, men delar av SMO hade inte stöd av Windows Azure. Detta gjorde att vi inte kunde använda oss av SMO för att kopiera databaser, så vi fick arbeta runt problemet och använda oss av en blandning av ADO-biblioteket³, bulk-kopiering⁴ och ett skript för databasens struktur.

³ Microsofts ActiveX Data Objects (ADO) är en samling Component Object Model (COM) objekt för att få tillgång datakällor.

⁴ SqlBulkCopy-klassen kan användas för att skriva data till SQLServer-tabeller.

5.4 Framtida arbete

Här beskriver vi framtida arbete som skulle kunna göras efter vårt examensarbete.

5.4.1 Enklare skriptande av databas

Eftersom SMO inte fungerade med Azure så var vi tvungna att använda ett förskrivet skript för att kopiera databaser. Ett framtida arbete skulle då vara att kunna skapa skript som kopierade strukturen för en databas och sedan lagra det i skriptfilen som ligger på storage i molnet, eller implementera SMO om det en dag får full support av Windows Azure.

5.5 Summering

I detta kapitel har vi beskrivit resultatet av vårt examensarbete. Sedan utvärderade vi datormolnet. Vi har också beskrivit de problem som har uppstått inom arbetet som tex. migrerings- och kompatibilitetsproblem och har också diskuterat vilken framtida förbättring som kan göras.

6 Slutsats

Med AMM-applikationen har Sogeti innan varit tvugna att tillhandahålla lagringsplats och underhåll, men nu sköts detta istället av Microsoft. Detta gör tillhandahållandet av applikationen smidigare då Microsoft kan erbjuda resurser som inte Sogeti kan. Nackdelen är dock att Sogeti inte har kontroll över tillgängligheten över applikationen ifall Microsofts Servrar skulle vara nere. Men i detta fall skulle inte det innebära stora förluster eller problem då AMM-applikationen inte är beroende av ständigt tillgänglighet. Efter våra förbättringar blev det mycket enklare att leverera applikationen till kund, eftersom arbetet som krävs för att leverera applikationen har minskat avsevärt. Genom en knapptryckning kan man enkelt få en helt ny databas bunden till en kund och därmed kan flera parallella versioner existera med olika databasuppkopplingar under samma applikation. Med applikationen i molnet kan vi när som helst bestämma att minska eller öka applikationens resurser. Skulle applikationen bli väldigt populär kan vi enkelt skala upp och skulle applikationen användas mindre kan vi enkelt skala ner på resurser eller till och med ta bort applikationen vid de tidpunkter den inte alls används.

Referenser

- [1] DYA - Homepage, <http://eng.dya.info/Home/>, 2010-05-04
- [2] DYA - What is DYA, http://eng.dya.info/Home/dya/what_is_dya/index.jsp, 2010-05-04
- [3] Definition för Cloud Computing klar !, <http://blog.akenine.net/2009/11/03/definition-fr-cloud-computing-klar/>, 2010-05-04
- [4] Datormoln, <http://sv.wikipedia.org/wiki/Datormoln> , 2010-05-13
- [5] Cloud Computing, http://en.wikipedia.org/wiki/Cloud_computing,2010-05-13
- [6] Datormolnet(Cloud Computing), <http://merteknik.se/datormoln-cloud-computing/>,2010-05-13
- [7] Software As A Service, http://en.wikipedia.org/wiki/Software_as_a_service, 2010-05-13
- [8]Serverhallar sugns upp i datormolnet, <http://www.idg.se/2.1085/1.153588>, 2010-05-13
- [9]Windows Azure Platform, http://en.wikipedia.org/wiki/Azure_Services_Platform , 2010-04-30
- [10] Blob data, [http://sv.wikipedia.org/wiki/Blob_\(data\)](http://sv.wikipedia.org/wiki/Blob_(data)), 2010-05-04
- [11]Blob Service API, [http://msdn.microsoft.com/en-us/library/dd135733\(v=MSDN.10\).aspx](http://msdn.microsoft.com/en-us/library/dd135733(v=MSDN.10).aspx) ,2010-05-04
- [12]Cloudy In Seattle, <http://blogs.msdn.com/jnak/archive/2008/10/29/walkthrough-simple-blob-storage-sample.aspx>, 2010-05-04
- [13]SQL Azure, (http://en.wikipedia.org/wiki/SQL_Services), 2010-05-04
- [14]Windows Azure Tables,<http://download.microsoft.com/download/3/B/1/3B170FF4-2354-4B2D-B4DC-8FED5F838F6A/Windows%20Azure%20Table%20-%20Dec%202008.docx>
- [15]Windows Azure Walkthrough, Table storage,<http://blogs.msdn.com/jnak/archive/2008/10/28/walkthrough-simple-table-storage.aspx>, 2010-05-04
- [16] Representational State Transfer, http://sv.wikipedia.org/wiki/Representational_State_Transfer, 2010-05-05
- [17] Windows Azure datalagring Queues, <http://blogs.msdn.com/chrislof/archive/2009/05/11/windows-azure-datalagring-queues.aspx>, 2010-05-05

- [18] Windows Azure Queues, <http://download.microsoft.com/download/5/2/D/52D36345-BB08-4518-A024-0AA24D47BD12/Windows%20Azure%20Queue%20-%20Dec%202008.docx> , 2010-05-05
- [19] Azure Services Platform, http://en.wikipedia.org/wiki/Azure_Services_Platform, 2010-05-05
- [20] Live Services, http://en.wikipedia.org/wiki/Live_Services , 2010-05-05
- [21] Windows Azure Platform AppFabric, <http://www.microsoft.com/windowsazure/appfabric/>, 2010-05-05
- [22] AppFabric Programming Guide, <http://msdn.microsoft.com/en-us/library/dd582779%28v=MSDN.10%29.aspx>, 2010-05-05
- [23] Windows Azure Platform, <http://msdn.microsoft.com/en-us/library/dd163896%28v=MSDN.10%29.aspx>, 2010.05-10
- [24] Windows SharePoint Services, http://en.wikipedia.org/wiki/Windows_SharePoint_Services, 2010.05-10
- [25] Windows SharePoint Services, http://en.wikipedia.org/wiki/Microsoft_Dynamics_CRM, 2010.05-10
- [27] NetServices Video, http://www.microsoft.com/windowsazure/videoPlayerPopup.aspx?w=720&h=480&vid=NetServices_720x480_FINAL_101609.wmv, 2010-05-10
- [28].Net Framework, http://sv.wikipedia.org/wiki/.NET_Framework, 2010-05-10
- [29] Software Development Kit, http://sv.wikipedia.org/wiki/Software_Development_Kit, 2010-05-10
- [30] CloudBerry Explorer for Azure Blob Storage, <http://cloudberrylab.com/default.aspx?page=explorer-azure>, 2010-05-10
- [31] SQL Azure Migration Wizard v3.2.1, <http://sqlazuremw.codeplex.com/>, 2010-05-10
- [32] Orkestrering, http://en.wikipedia.org/wiki/Orchestration_%28computers%29, 2010-05-12
- [33] T-SQL, <http://en.wikipedia.org/wiki/TSQL>, 2010-05-21
- [34] SQLite Data Wizard, <http://www.sqlmaestro.com/products/sqlite/datawizard/>, 2010-05-21

[34] Utility Computing,http://en.wikipedia.org/wiki/Utility_computing,2010-05-21

[35]Microsoft SQL Server,http://sv.wikipedia.org/wiki/Microsoft_SQL_Server,2010-05-21

Bilaga A

Ordlista

AMM: Architecture Maturity Model

Blob : Filer innehållande binär data

Container: En filbehållare , innehåller blobbar i detta fall

Datormoln: Datormoln[4] även kallat *molnet*, är en teknik baserad på användning av datorer över Internet. Det är en teknik där stora skalbara resurser, exempelvis processorkraft, lagring och funktioner, tillhandahålls som tjänster på Internet till användare som inte behöver ha den tekniska kunskapen eller kontrollen över infrastrukturen.

DYA: Dynamic Architecture

Off-premise : inte lokalt (utanför företagets gränser, i detta fall på molnet)

On-premise : lokal (inom företagets gränser)

Orkestrering:Beskriver det automatiska arrangemanget, samordning och hantering av komplexa datorsystem, middleware och tjänster.

Storage: en tjänst i Windows Azure för lagring

SQL Azure : databastjänsten som ligger i Windows Azure

Windows Azure: själva molntjänsten

Web Role: webbprojekt

Worker Role: klassbibliotek eller andra projekt som inte är webbprojekt