

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Anže Krpič

**Integracija ERP Microsoft Dynamics AX z mobilnimi
napravami**

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM
RAČUNALNIŠTVO IN INFORMATIKA

Mentor: doc. dr. Damjan Vavpotič

Ljubljana, 2014

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Anže Krpič,

z vpisno številko 63060429,

sem avtor diplomskega dela z naslovom:

Integracija ERP Microsoft Dynamics AX z mobilnimi napravami

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom
doc. dr. Damjana Vavpotiča
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 17. 3. 2014

Podpis avtorja:

Rezultat diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil MS Word



Št. naloge: 01990 / 2014
Datum: 7.3.2014

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ANŽE KRPIČ**

Naslov: **INTEGRACIJA ERP MICROSOFT DYNAMICS AX Z MOBILNIMI
NAPRAVAMI
INTEGRATION OF MICROSOFT DYNAMICS AX ERP WITH MOBILE
DEVICES**

Vrsta naloge: DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJA

Tematika naloge:

V okviru diplomske naloge predstavite možnosti za integracijo sistema ERP Microsoft Dynamics AX z mobilnimi napravami. V nalogi najprej predstavite omenjeni sistem ERP in možnosti, ki jih sistem ponuja za integracijo z zunanjimi storitvami. Nato se osredotočite na povezovanje z mobilnimi napravami. Predstavite mehanizme, ki jih v ta namen ponuja omenjeni ERP ter za eno od mobilnih platform izdelajte manjši prototip, ki bo predstavil delovanje omenjenih mehanizmov. Na kratko komentirajte tudi možnosti povezovanja z različnimi mobilnimi platformami.

Mentor:

doc. dr. Damjan Vavpotič

Dekan:

prof. dr. Nikolaj Zimic



Zahvaljujem se svojemu mentorju doc. dr. Damjanu Vavpotiču za vso pomoč in strokovno svetovanje pri pisanju diplomskega dela. Hvala tudi podjetju Adacta d.o.o., Mateju Skubicu, Boštjanu Golobu in Iztoku Juvanu, ki so mi omogočili izdelavo diplomske naloge. Iskrena hvala staršem, zaročenki Marini Zibelnik in prijateljem za vso podporo tekom študija.

Kazalo

Povzetek

Abstract

1	Uvod.....	1
2	Pregled ERP sistemov in primerjava Microsoft Dynamics AX s konkurenco	3
2.1	Opis konkurenčnih ERP sistemov	3
2.2	ERP sistemi in mobilne platforme	5
2.2.1	Razlogi za vzpostavitev podpore za mobilne platforme ERP aplikacij.....	5
2.2.2	SAP in mobilne aplikacije	6
2.2.3	Oracle in mobilne aplikacije.....	7
2.2.4	Sage in mobilne aplikacije.....	8
2.2.5	Infor in mobilne aplikacije	9
3	Microsoft Dynamics AX	11
3.1	Zgodovina	11
3.2	Arhitektura	12
3.2.1	Razvojno okolje.....	14
3.2.2	Modelna arhitektura.....	17
3.2.3	Programski jezik X++.....	18
3.2.4	Dostop do podatkov	20
4	Integracija Microsoft Dynamics AX z zunanjimi storitvami	23
4.1	Možnosti, ki jih ponuja Microsoft Dynamics AX	23
4.2	SOA – Storitveno-orientirana arhitektura.....	24
4.3	WCF – Windows Communication Foundation	24
4.4	Storitve v Microsoft Dynamics AX	25
4.4.1	»Document services«.....	25
4.4.2	Storitve po meri	26
4.4.3	Sistemske storitve	27
4.4.4	Omogočanje storitev	27
5	Mobilna aplikacija za Microsoft Dynamics AX	29
5.1	Koncept in opredelitev problema.....	29
5.2	Razvoj aplikacije.....	31

5.2.1	UML diagram postavitve	31
5.2.2	AIF storitev	32
5.2.2.1	Podatkovna pogodba (Data contract).....	32
5.2.2.2	Storitev.....	33
5.2.2.3	Izpostavitev storitve.....	34
5.2.2.4	Avtentikacija na strani Dynamics AX	35
5.2.3	Windows Azure Service bus	35
5.2.4	Mobilna aplikacija »Windows Phone App«	36
5.2.4.1	Grafični vmesnik	42
5.2.5	Integracija z iOS, Android in ostale platforme	46
6	Zaključek	47
7	Literatura.....	49

Kazalo slik

Slika 1 Postopek komunikacije Mobile Webdynpro Java [12]	6
Slika 2 Komunikacijske poti pri SUP [12]	7
Slika 3 Arhitektura integracije Oracle aplikacij z mobilnimi napravami [13]	8
Slika 4 Komunikacijske poti pri uporabi Sage Enterprise Webtop [16].....	9
Slika 5 Primer zaslonske maske klienta Dynamics AX [20].....	12
Slika 6 Postopek komunikacije SQL server, AOS in klienti.....	13
Slika 7 Povezave med Dynamics AX in možnimi integracijami ter klient [20]	13
Slika 8 AOT [21]	14
Slika 9 X++ urejevalnik [21]	16
Slika 10 Prikaz razdelitve po nivojih in modelih [20].....	17
Slika 11 Sintaksa kode [21]	19
Slika 12 Postopek prevajanja [32]	19
Slika 13 Sintaksa poizvedbe [21]	20
Slika 14 »Query« v AOT [21]	21
Slika 15 Komunikacija pri integracijah [34]	23
Slika 16 Postopek komunikacije pri uporabi Windows Azure [26]	29
Slika 17 UML diagram postavitve.....	31
Slika 18 Storitve v AOT	34
Slika 19 Zaslonska maska nastavitvev porta	34
Slika 20 Nastavitve Windows Azure v Dynamics AX.....	35
Slika 21 Status storitve na Windows Azure Service bus.....	36
Slika 22 Zaslonska maska prve strani.....	37
Slika 23 Zaslonska maska prikaza vseh projektov	43
Slika 24 Zaslonska maska podrobnosti projekta	44
Slika 25 Pogled transakcije in podrobnosti na projektu	45

Povzetek

V diplomski nalogi so predstavljene možnosti integracije integriranega poslovnega sistema (ERP) z mobilnimi napravami, osrednji integriran poslovni sistem pa je Microsoftov Dynamics AX 2012 R2.

V prvem delu je prikazana konkurenca in stanje ERP sistemov na trgu, na kratko so predstavljeni nekateri največji ERP sistemi, ki predstavljajo konkurenco Dynamics AX-u, kakšne možnosti razvoja mobilnih aplikacij ponujajo in možnosti integracije z njimi.

V drugem delu diplomske naloge je predstavljen ERP sistem Microsoft Dynamics AX, kratka zgodovina in razvoj sistema ter pregled verzij. Nato je predstavljena arhitektura in pregled področij, ki jih Dynamics AX pokriva, in kaj je z njim podprto. Prav tako je opisano delovanje ter razvoj v okolju Dynamics AX. Predstavljeni so programski jezik, v katerem se razvija, vse strukture, ki jih podpira, in postopek razvoja.

Tretji del je posvečen možnostim integracije sistema Dynamics AX z zunanjimi aplikacijami in storitvami. Predstavljen je Application Integration Framework (AIF). Opisane so vse tehnologije, ki so zajete in potrebne za njegovo delovanje.

V zadnjem delu je predstavljen primer implementacije mobilne aplikacije in vseh ostalih aplikacij ter storitev, ki so potrebne za integracijo s storitvami, ter aplikacijami, ki dostopajo do ERP sistema iz sistemov izven organizacije. Predstavljena je zasnova in funkcionalnost aplikacije za pregled projektov in knjiženje delovnih ur nanje.

Ključne besede:

- Microsoft Dynamics AX
- ERP
- Windows Azure
- Storitve
- Integracija
- Mobilne aplikacije

Abstract

The diploma thesis describes integration options of enterprise resource planning (ERP) system with mobile devices, where Microsoft Dynamics AX 2012 R2 is the main enterprise resource planning system.

The first part shows the competition and the positioning of ERP systems on the market, as well as a brief introduction to some of the largest ERP systems that pose competition to Dynamics AX and what possibilities they offer with integration and mobile development.

In the second part of the thesis the ERP system Microsoft Dynamics AX is described. Brief overview of the history, development and the versions of the system are shown. It presents an overview of the fields covered by Dynamics AX and what functionalities are supported. Operation of Dynamics AX and development environment are also described. There is a description of programming language used for development, structures it uses and process of development.

The third part is devoted to the integration capabilities of the system Dynamics AX with external applications and services. Application Integration Framework (AIF) is presented and all the necessary technologies are described.

The last part describes the actual implementation of mobile applications and all other applications and services needed for the integration with ERP system and applications and services from systems outside the organization. The design and functionalities of application for viewing projects and posting work hours is also shown.

Keywords:

- Microsoft Dynamics AX
- ERP
- Windows Azure
- Services
- Integration
- Mobile applications

Poglavje 1

1 Uvod

V današnjem poslovnem svetu je vedno več konkurence in pritiskov z vseh strani k doseganju vedno višjih ciljev. Tako se vedno več podjetij srečuje s problemom, kako povečati učinkovitost svojega poslovanja in zmanjšati stroške. Eden ključnih dejavnikov je informatizacija poslovanja, kar omogoča dostop do pravočasnih in kar se da točnih informacij. To pa najučinkoviteje dosežemo z vpeljavo integriranega poslovnega sistema (Enterprise Resource Planning - ERP).

Vedno več pa je dejavnikov, ki na nek način silijo podjetja k omogočanju poslovanja na mobilnih platformah. Tako se je pojavila potreba k povezavi integriranega poslovnega sistema z mobilnimi napravami. Prek teh storitev se uporabnikom omogoča dostop do podatkov in funkcionalnosti sistema prek spleta ter mobilnih aplikacij. Slednje je dobrodošlo predvsem zato, ker so pametne mobilne naprave zelo razširjene in kot take sposobne poganjati zahtevnejše aplikacije, ki so povezane z ERP. Z možnostjo interakcije z osnovnim ERP sistemom prek oddaljenih storitev in aplikacij lahko dosežemo lažje poslovanje in hitrejši prenos informacij, kar še izboljšuje učinkovitost poslovanja.

Podatki v ERP sistemu predstavljajo pomemben del poslovanja, zato je ključno da so dostopi omejeni in varovani. Pri integraciji je zato pomemben dejavnik prav varnost. Preden lahko začnemo z razvojem mobilne aplikacije, je potrebno razmisliti, kako zagotoviti varen dostop do ERP sistema izven omrežja podjetja in na kakšen način bomo to izvedli.

V diplomskem delu bi radi ugotovili, kakšne možnosti obstajajo za integracijo z Microsoft Dynamics AX-om in nato razvili mobilno aplikacijo, ki bi dostopala do sistema, ter ugotovili in analizirali, kakšni so problemi pri razvoju integracije. Poskušali bomo ugotoviti, kaj je pri dostopu do Microsoft Dynamics AX potrebno upoštevati in na kakšen način se da to izvesti v primeru mobilnih aplikacij. Ugotovitve bomo poskušali implementirati in pokazati koncept integracije mobilne aplikacije na primeru projektov.

Poleg osnovnih ciljev bi radi ugotovili, kakšne možnosti za integracijo ponuja konkurenca in kako se z njimi primerja Microsoft Dynamics AX. Pregledali bomo tudi

zgodovinski razvoj Dynamics AX-a in verzije ter katera področja poslovanja so zajeta. Dotaknili se bomo tudi arhitekture in razvojnega okolja.

Poglavje 2

2 Pregled ERP sistemov in primerjava Microsoft Dynamics AX s konkurenco

2.1 Opis konkurenčnih ERP sistemov

Na trgu je veliko ponudnikov ERP sistemov. Nekateri med njimi so bolj znani, nekateri manj. Eden največjih ponudnikov na trgu integriranih poslovnih sistemov je podjetje SAP. Ustanovljeno je bilo leta 1972 v Nemčiji, naslednje leto pa se je na trgu pojavil njihov prvi produkt. Prva verzija je bila namenjena financam in računovodstvu. Zadnja njihova verzija pa podpira področja računovodstva, upravljanja oskrbovalne verige, kontrolinga, upravljanja materialov, prodaje, logistike, načrtovanja proizvodnje, upravljanja kakovosti, vzdrževanja proizvodnje, sisteme projektov in upravljanje s človeškimi viri. [1]

SAP je v marsičem podoben Microsoft Dynamics AX-u. Oba sistema sta zasnovana na trinivojski arhitekturi (odjemalec – strežnik – podatkovna baza) [6]. V obeh primerih gre za celoviti rešitvi za podporo poslovnih sistemov. Tako SAP kot Dynamics AX podpirata podjetja, ki poslujejo v multinacionalnih okoljih in z različnimi valutami. Moduli v SAP so zelo podobni tistim v sistemu Dynamics AX [7]. Podjetje SAP cilja na velika podjetja in kar 75 odstotkov podjetij iz seznama Fortune Global uporablja SAP [5].

Eden večjih ponudnikov ERP sistemov je podjetje Oracle. Podjetje ponuja lastne produkte pod skupnim imenom Oracle E-Business Suite, ki podpirajo področja odnosov s strankami, življenjski cikel sredstev, računovodstva, upravljanja s človeškimi viri, upravljanja projektov, nabave, življenjskih ciklov produktov, proizvodnje ter upravljanja oskrbovalne verige. Podjetje Oracle pa ponuja še dva ERP produkta pod imenom JD Edwards EnterpriseOne in JD Edwards World. [2]

Podjetje Oracle ne ponuja celovite rešitve ERP, kot je Dynamics AX, vendar ponuja različne produkte, ki vsak pokriva svoje področje. [2] To je posledica različnih prevzemov, ki jih je izvajalo podjetje tekom svoje zgodovine. Povezovanje teh modulov je lahko precej zahtevno, saj imajo različno zgodovino in razvoj. Podjetje cilja na velika podjetja in predstavlja konkurenco predvsem produktu SAP [8].

Večji konkurenčni podjetji na področju ERP sistemov sta še Sage in Infor. Podjetje Sage ponuja produkt Sage ERP X3, ki podpira proizvodnjo, nabavo, upravljanje z zalogo ter skladišči, prodajo, odnose s strankami ter računovodstvo. [3] Infor pa je ameriško podjetje s produktom Infor ERP in še mnogimi drugimi za pokrivanje potreb podjetij. [4]

Tako kot Dynamics AX je Sage ERP X3 [3] celovita rešitev in prav tako pokriva podobna področja. Glavni cilj podjetja Sage je pokriti srednje velika podjetja in podružnice velikih podjetij, ki poslujejo na mednarodnem trgu. Tako kot Dynamics tudi Sage podpira večjezično in večvalutno poslovanje. [15]

ERP sistem podjetja Infor predstavlja celovito rešitev, vendar je specializiran za distribucijo in proizvodnjo. Sistem omogoča hitro prilagajanje industriji, predvsem spremembam v proizvodnji. Glavni cilji so pokriti podjetja, ki se ukvarjajo s proizvodnjo in distribucijo. [4]

2.2 ERP sistemi in mobilne platforme

2.2.1 Razlogi za vzpostavitev podpore za mobilne platforme ERP aplikacij

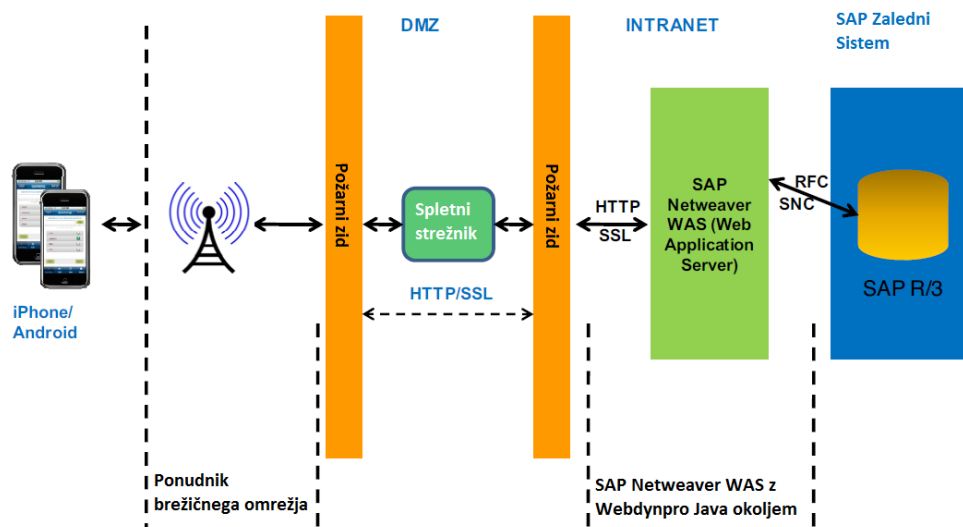
Potrebe po razvoju in uporabi povezljivosti mobilnih aplikacij z ERP sistemi so v današnjem času vedno večje, saj s tem omogočajo večjo fleksibilnost in produktivnost poslovanja. Vedno več podjetij spoznava, da je integracija njihovega poslovanja z mobilnimi napravami zelo pomembna, saj so tako lahko bolj konkurenčni. Vedno več poslovanja se dogaja zunaj poslovnih prostorov podjetja, tako da je dostop do informacij prek interneta zelo pomemben. Mobilne aplikacije, ki dostopajo do podatkov ERP sistema podjetja, omogočajo, da uporabniki lahko dostopajo do informacij od koderkoli in kadarkoli.

Za razvoj mobilnih aplikacij poznamo dva pristopa:

- **Non-Middleware pristop** je pristop, kjer obstajata dve vrsti aplikacij. Lahki odjemalci (thin client) in težki odjemalci (thick client). Pri lahkih odjemalcih gre za prikazovanje aplikacije kar prek brskalnika, pri težkih odjemalcih pa je razvita aplikacija, ki omogoča boljše uporabniško izkušnjo in se povezuje na ERP sistem prek spletnih storitev (SOAP web services) [11],
- **Middleware pristop** temelji na aplikaciji, ki ni del operacijskega sistema in prav tako ni del osnovne programske opreme. Aplikacija je nekakšno »lepilo«. Razvijalcem omogoča lažje izvajanje komunikacije in I/O operacije. Uporablja se predvsem za kompleksne in porazdeljene aplikacije [10]. Aplikacije so težki odjemalci, zgrajeni na podlagi MEAP (Mobile Enterprise Application Platform) [9]. Te aplikacije omogočajo »real-time« delovanje in delovanje brez povezave. Prednost takega pristopa je predvsem hiter razvoj na različnih napravah in konstantnost podatkovnega modela na različnih mobilnih tipih. [12]

2.2.2 SAP in mobilne aplikacije

SAP omogoča dva načina integracije s spletnimi aplikacijami. Prvi način predstavlja lahkega odjemalca z uporabo SAP-ove tehnologije Mobile Webdynpro Java. Slika 1 prikazuje postopek komunikacije med klientom in strežnikom. Tak pristop je primeren za aplikacije, ki temeljijo na zasnovi brskalnika in ne potrebujejo zahtevnega uporabniškega vmesnika ali potrebe po uporabi aplikacije brez povezave.

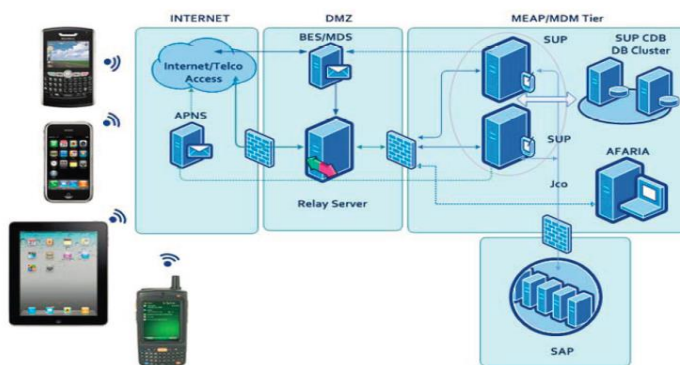


Slika 1 Postopek komunikacije Mobile Webdynpro Java [12]

Drugi način pa je razvoj z uporabo Sybase Unwired Platform (SUP). SUP je middleware MEAP aplikacija, ki omogoča povezovanje med različnimi viri podatkov, kot so relacijska baza ali ERP aplikacija, in mobilno napravo, ki mora dostopati do podatkov. SUP vsebuje razvojna orodja, ki omogočajo razvoj poslovne logike aplikacije, ki jo potem avtomatično ustrezno prevede za uporabo na različnih mobilnih napravah in operacijskih sistemih.

SUP je sestavljen iz štirih glavnih komponent. Sybase Unwired Server omogoča uporabnikom nalaganje aplikacij in upravlja s podatkovnimi viri podjetja ter komunikacijo z mobilnimi napravami, varnostjo in procesiranjem transakcij. Sybase Control Center je spletna konzola, namenjena konfiguraciji in upravljanju Unwired Serverja in njegovih komponent. Razvojno okolje za razvoj mobilnih aplikacij in njihovo testiranje se imenuje Sybase Unwired

Workspace. Zadnja komponenta pa se imenuje Sybase Afaria, ki omogoča upravljanje in varnost različnih mobilnih naprav preko celotnega podjetja v katerem koli omrežju ne glede na velikost. Prednost tega načina je v tem, da podpira večnapravno strategijo, možnost delovanja aplikacije s povezavo ali brez in da so stroški vzdrževanja ter podpore nizki. Problem pa predstavljajo visoki stroški začetne naložbe. Slika 2 prikazuje način interakcije sistemov. [12]



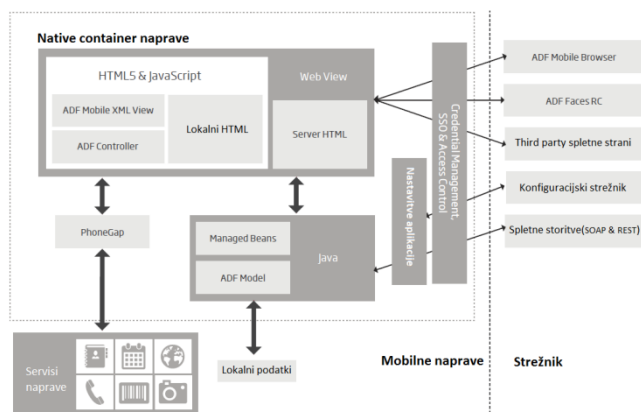
Slika 2 Komunikacijske poti pri SUP [12]

2.2.3 Oracle in mobilne aplikacije

Način razvoja mobilnih aplikacij je pri Oraclu podoben tistemu pri SAP. Poznamo dve vrsti in obe bazirata na platformi Oracle Application Development Framework (ADF) Mobile. Prvi je Oracle ADF Mobile browser, razvoj spletnih aplikacij do katerih dostopamo prek brskalnika na mobilnem telefonu. Uporabniški vmesnik je implementiran s komponentami Apache Trinidad JavaServer Faces (JSF), do katerih dostopa brskalnik. Vmesniki so optimizirani za vsak brskalnik posebej. [13]

Drugi način je razvoj aplikacij z ADF Mobile. Aplikacija, razvita z ADF Mobile, je inštalirana na mobilni napravi. Uporabniški vmesnik je predstavljen s pomočjo HTML, preko aplikacije pa je omogočen dostop do storitev naprave. To omogoča enotnost in zmožnost prenosa aplikacije na različne naprave. Ker se take aplikacije poganja lokalno na napravi sami, internetna povezava nima vpliva na izvajanje. Za razvoj uporabniškega vmesnika se lahko

uporabijo trije načini. Eden od načinov je uporaba lokalnega HTML5 in JavaScript. Prikaz poteka s pomočjo ADF Mobile XML (AMX). Prek Java virtual machine (VM) se lahko dostopa do spletnih storitev SOAP/REST in JavaBeans, kar omogoča integracijo z ERP. Za lokalno shrambo podatkov in dostop do podatkov, ko aplikacija deluje brez povezave, se lahko uporablja SQLite. Za prikaz se lahko uporabi tudi direktno HTML5 in JavaScript, brez AMX. Drugi način je prikaz HTML5 in JavaScript strani, do katerih dostopa aplikacija. ADF pa omogoča tudi prikaz uporabniškega vmesnika in funkcionalnosti, razvitih v kodi naprave same. ADF omogoča varnost aplikacij z enkripcijo, avtentikacijo in nadzorom dostopa. Povezave potekajo prek protokola HTTPS, avtentikacija je podprta prek centralnega servisa, kot je Oracle Identity Management. Slika 3 prikazuje kako je arhitekturno izvedena povezava med sistemom in mobilnimi napravami. [14]

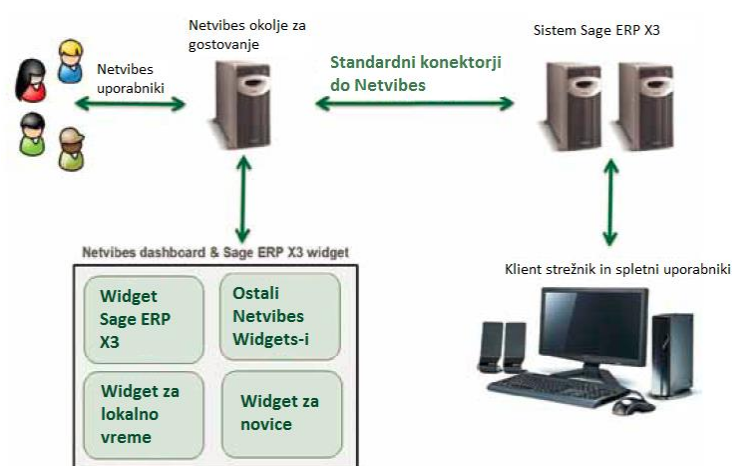


Slika 3 Arhitektura integracije Oracle aplikacij z mobilnimi napravami [13]

2.2.4 Sage in mobilne aplikacije

Sage ima za dostop do ERP sistema nekoliko drugačen pristop. Do sistema se dostopa s pomočjo tehnologije, ki se imenuje Sage Enterprise Webtop. Za dostop se uporablja Netvibes Widgets, ki je personalizirana platforma za objavljanje na spletu. Uporabnikom omogoča, da prek različnih pripomočkov objavljajo različne stvari iz svojega življenja [20]. Preko teh pripomočkov se lahko tudi varno dostopa do Sage ERP od koderkoli in z različnimi napravami. Na sliki 4 vidimo po kakšnih poteh poteka komunikacija. Pozitivne točke tega so, da se prek

Sage Enterprise Webtop-a odprejo možnosti dostopa do ERP sistema ne samo uporabnikom, ampak tudi poslovnim partnerjem (stranke, dobavitelji, ...) [15]. Razvijalci pa lahko razvijejo nove pripomočke za razširitev funkcionalnosti sistema. Sage ERP X3 ponuja tri vrste pripomočkov: »List Widget« omogoča izpis podatkov v obliki seznama iz ERP sistema, »Detail Widget« ponuja podrobnosti določenega podatka iz seznama in »Purches Request Widget«, ki uporabnikom omogoča ustvarjanje nabavnih zahtevkov. Sage Enterprise Webtop je dostopen vsem, ki imajo internetno povezavo in pravice za dostop do pripomočkov Sage ERP X3. Tako je Sage ERP X3 dostopen na vseh napravah. [16]



Slika 4 Komunikacijske poti pri uporabi Sage Enterprise Webtop [16]

2.2.5 Infor in mobilne aplikacije

Za dostop do Infor ERP se uporablja tehnologijo Infor Motion, ki omogoča hiter razvoj, s pomočjo tehnologije Intelligent Open Network (ION) pa je integracija mobilnih aplikacij z ERP neposredna in preprosta. Lahko se določa pravice mobilnega dostopa, tako da so na voljo le informacije, ki so za to namenjene. Obstaja že kar nekaj razvitih mobilnih aplikacij, ki so standardno prisotne pri Infor ERP. [17]

Poglavje 3

3 Microsoft Dynamics AX

3.1 Zgodovina

Začetek Dynamics AX-a sega v leto 1983, ko sta brata Erik in Preben Damgaard iz Kopenhagna na Danskem razvila začetno aplikacijo imenovano DANMAX. Marca leta 1998 je prišla na danski in ameriški trg aplikacija Axapta 1.0. Že istega leta, meseca novembra, je izšla različica Axapta 1.5. Na voljo je bila na Norveškem, Švedskem, Nizozemskem, v Veliki Britaniji, Nemčiji, Avstriji, Švici, Belgiji, Španiji in drugod v Evropski uniji. Leta 2001 se je podjetje Damgaard A/S združila z Danskim podjetjem Navision. Tako sta se naslednji dve verziji imenovali Navision Damgaard Axapta 2.5 in 3.0. Sredi leta 2002 je Microsoft prevzel podjetje Navision Damgaard in produkt sprva preimenoval v Microsoft Business Solutions Axapta, kasneje pa v Microsoft Dynamics AX. Sledili sta dve večji izdaji, Axapta 3.0 oktobra 2002, in 4.0 marca 2006. Junija 2008 je na trg prišla različica, sprva poimenovana Axapta 4.1, Dynamics AX 2009 in avgusta 2011 Dynamics AX 2012. Trenutna verzija na trgu je Dynamics AX 2012 R2, ki je bila izdana decembra 2012. [18]

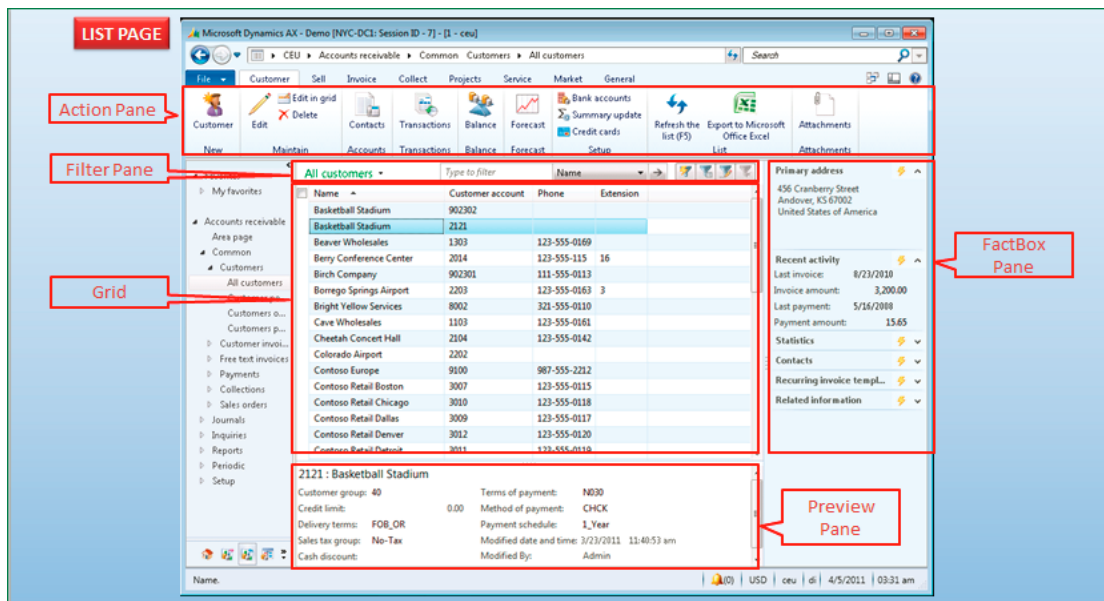
Prva verzija (Axapta 1.0) je podpirala podatkovno bazo SQL Server in Oracle. Področja, ki so bil pokrita, so bila finance, trgovanje, skladiščenje, logistika in proizvodnja. Naslednja verzija (Axapta 2.0) je prinesla novost zunanjega OLAP (Online analytical processing) in začetno verzijo Axapta Object Server, ki je omogočala izvajanje določenih operacij na strežniku. V Axapta 2.1 se pojavi Customer Self-Service (CSS), ki je bil predhodnik današnjemu Enterprise Portal-u. Prav tako pa je s servisnim paketom za to verzijo (Service Pack 3) prišel AOS (Axapta Object Server), s katerim je postala Axapta prvi povsem trinivojski ERP sistem na trgu. Z verzijo 2.5 postane OLAP del sistema. V verziji 3.0 pod okriljem Microsofta vpeljejo Enterprise portal. Verzija 4.0 je prva verzija, ki je bila v celoti razvita pod Microsoftom. AOS postane Windows servis. Prav tako se v tej verziji pojavi Application Integration Framework-a (AIF). Naslednji dve verziji (AX 2009 in 2012) prineseta spremembe pri varnosti in posodobitev grafičnega vmesnika. [19]

3.2 Arhitektura

Microsoft Dynamics AX ima ločeno podatkovno bazo, strežnik in odjemalca. Takšna arhitektura je tipična za trinivojske aplikacije. Podatkovna baza, ki jo uporablja Dynamics AX, je SQL Server. Podatkovna baza je tipa Online Transactional Processing in shranjuje vse podatke v SQL tabelah. Prav tako pa se v »Model Store« shranjujejo metapodatki in programska koda.

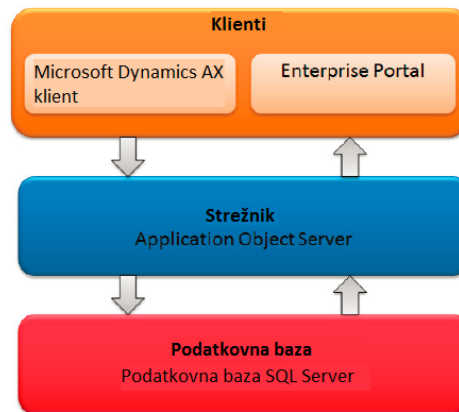
AOT (Application Object Server) predstavlja servis, ki konstantno teče na strežniku in skrbi za izvajanje večine poslovne logike. AOT izvaja številna opravila, kot so poganjanje X++ kode, ki predstavlja poslovno logiko in zagotavlja varnost s tem, da upravlja dostope do podatkov in funkcionalnosti sistema.

Dynamics AX ima več možnosti dostopanja uporabnika do strežnika. Večinoma se uporablja namiznega klienta ali dostopanje preko Enterprise portala. Namizni klient se imenuje Microsoft Dynamics AX Rich Client. Primer okna takšnega klienta prikazuje slika 5.



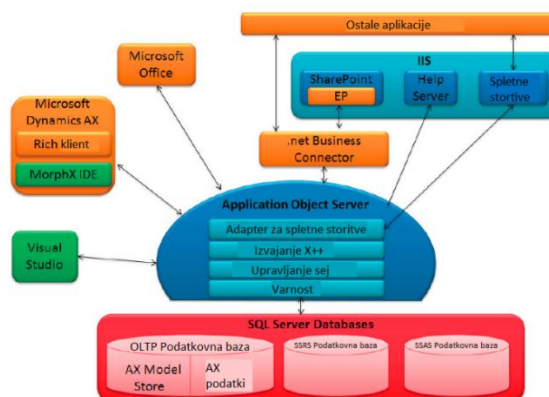
Slika 5 Primer zaslonske maske klienta Dynamics AX [20]

Do podatkov sistema je omogočen dostop tudi s pomočjo Microsoft Worda ali Microsoft Excela. Ena od možnosti dostopa je tudi .NET Business Connector, ki ga lahko uporablja tako uporabnik kot zunanja aplikacija. Na sliki 6 je prikazana komunikacijska pot pri poizvedbah klienta.



Slika 6 Postopek komunikacije SQL server, AOS in klienti

Poleg osnovnih komponent ponuja Dynamics AX možnost uporabe različnih Microsoftovih tehnologij v povezavi s sistemom. Za poročila se uporablja komponenta SQL Serverja, imenovana SQL Reporting Services (SSRS). SQL Server vsebuje tudi komponento imenovano SQL Analysis Services (SSAS), ki sestavlja Online Analytical Processing (OLAP) kocke, katere uporabljajo »Key Performance Indicators«, ki so komponenta Business Intelligence-a (BI). Za gostovanje Enterprise Portal-a se uporablja SharePoint. Ostale tehnologije je potrebno integrirati drugače, na primer preko spletnih storitev. Slika 7 prikazuje diagram povezav pri integraciji s sistemom.



Slika 7 Povezave med Dynamics AX in možnimi integracijami ter klient [20]

Za delovanje spletnih storitev Help Serverja in Enterprise Portala pa skrbi spletni strežnik Internet Information Server (IIS). [20]

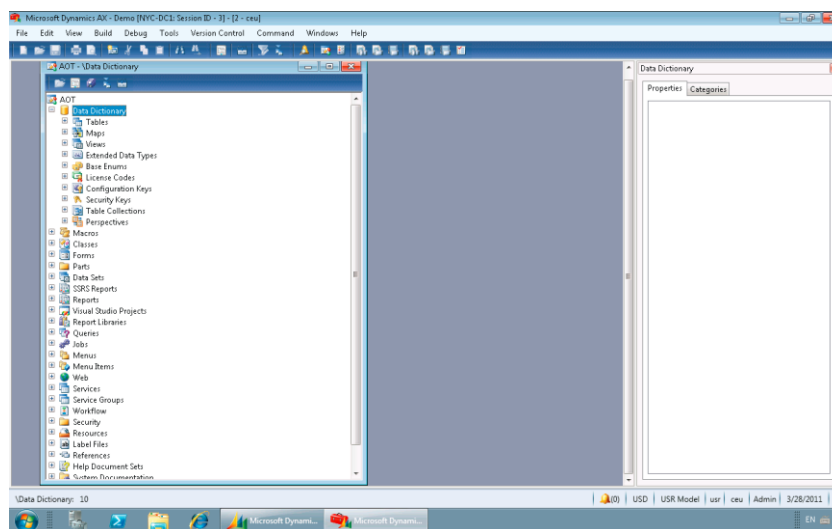
3.2.1 Razvojno okolje

Za razvoj aplikacij Microsoft Dynamics AX 2012 se lahko uporablja Microsoft Visual Studio ali Microsoft Dynamics AX Integrated Development Environment (IDE), ki je znan tudi kot MorphX IDE. Do MorphX okolja je možno dostopati prek Microsoft Dynamics AX klienta.

Preko teh razvojnih okolij se lahko popravlja ali dodaja elemente, iz katerih je sestavljena aplikacija Dynamics AX. Elementi so shranjeni v »model store« bazi SQL, delijo pa se na tri večje skupine:

- na polja in tabele, ki predstavljajo podatkovne strukture,
- na menije in forme, ki omogočajo interakcijo uporabnika z aplikacijo
- in na razrede ter metode, ki vsebujejo kodo poslovne logike.

Hierarhija elementov je predstavljena v AOT (Application Object Tree), ki omogoča tudi premikanje po strukturi aplikacije (slika 8). V Dynamics AX aplikaciji nastopa več tipov elementov.



Slika 8 AOT [21]

V skupino »Data Dictionary« spadajo:

- tabele, ki predstavljajo skupek polj različnih tipov (AX pozna deset tipov polj: »Integer«, »String«, »Real«, »Date«, »Time«, »UtcDateTime«, »Enum«, »Container«, »Guid« in »Int64«). Posamezna polja podedujejo lastnosti pripadajočih razširjenih podatkovnih tipov (Extended data types),
- razširjeni podatkovni tipi (EDT), ki definirajo podatkovni tip in ga razširijo s poljubnimi lastnostmi. Definirajo lahko dolžino, oznake (»labele«), obveznost polja, lastnosti prikaza na formah ...,
- enumeracije (Base Enums), naštevanja tekstovnih podatkov, ki so v bazi predstavljena kot celoštevilski zapis, kateremu pripada omejeno število možnosti. Tako ima na primer enumeracija z imenom NoYes dve možnosti: 0 (No) in 1 (Yes),
- pogledi (Views), mapiranja (Maps), licenčne kode, konfiguracijski ključi ...

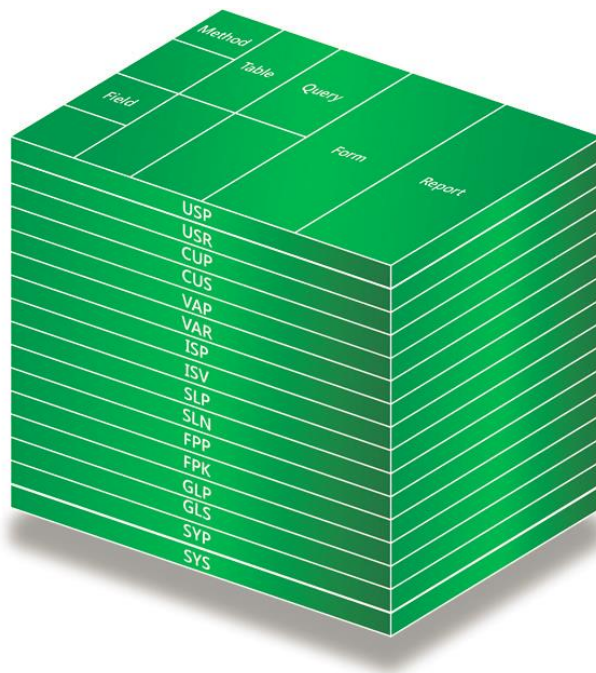
Pomembnejše skupine so še »Classes«, kjer so vsi razredi uporabljeni v aplikaciji, skupina »Forms« vsebuje vse forme aplikacije, »Menus« definira meni in poročila aplikacije, elementi menija pa spadajo pod skupino »MenuItems«, ki pa se delijo na skupine »Display«, »Action« in »Output«.

Pri ERP sistemih, ki podpirajo multinacionalno delovanje podjetij, je pomembno zagotavljanje možnosti večjezičnega programa. Microsoft Dynamics AX tako pozna poimenovanje s pomočjo oznak. Vsaka oznaka ima svojo kodo (oblike npr. @SYS5241), kateri pripada prevod za jezik. Za vsak podprt jezik obstaja datoteka, ki vsebuje prevode oznak. Pri razvoju se lahko oznake takoj prevede v več jezikov s pomočjo pripomočka »Label editor«.

Za razvoj aplikacij Dynamics AX se najpogosteje uporablja MorphX IDE, ki omogoča dostop do AOT. Omogoča preprosto urejanje elementov brez kodiranja. Če pa je potrebno razviti kompleksnejšo poslovno logiko, se uporablja programski jezik X++. MorphX okolje vsebuje orodja, ki pripomorejo k lažjemu razvoju. X++ urejevalnik (Code Editor) je urejevalnik besedila s podporo prikaza objektov in vizualizacije kode. Slika 9 prikazuje razred v urejevalniku X++.

3.2.2 Modelna arhitektura

Microsoft Dynamics AX je zasnovan tako, da je razdeljen na nivoje (»layer«) in modele (slika 10), ki ločujejo modifikacije aplikacije med seboj. To omogoča nadgradnjo novih verzij kljub spremembam na aplikaciji.



Slika 10 Prikaz razdelitve po nivojih in modelih [20]

Obstaja osem nivojev, ki vsebujejo vse elemente aplikacije. Od teh osmih ima še pripadajoč »patch layer«, ki služi shranjevanju konfliktov pri nadgradnji kode. Ti nivoji omogočajo popraviljanje kode nižjega nivoja, ne da bi ga spremenil na obstoječem nivoju. Vsaka sprememba na višjem nivoju prekrije tiste na nižjem. Pri prevajanju kode se na koncu uporabijo le elementi z najvišjega nivoja.

Nivoji so grupirani v 3 glavne skupine:

- Spodnji trije nivoji so namenjeni razvoju aplikacije s strani Microsofta. »System layer« (SYS) vsebuje standardno aplikacijo, ki je lokalizirana za večino podprtih držav. »Globalization layer« (GLS) vsebuje dodelave za nekatere države, ki še niso

bile predstavljene na nivo SYS. »Feature Pack layer« vsebuje dodelave, za katere skrbi Microsoft.

- Naslednji trije nivoji so na voljo Microsoftovim partnerjem, na katerih lahko razvijajo dodelave. »Solution layer« (SLN) vsebuje rešitve, za katere zagotavlja podporo Microsoft, »Independent Software Vendor layer« (ISV) vsebuje rešitve, ki so jih razvila neodvisna podjetja za razvoj Dynamics AX, »Value Added Reseller layer« (VAR) pa vsebuje dodelave za različne stranke.
- Dva nivoja pa sta na voljo partnerjem in strankam z licenco za dostop do kode Dynamics AX. »Customer layer« vsebuje funkcionalnosti, ki so specifične posamezni stranki, in »User layer« za posebne specifike in dodelave.

Prednost take arhitekture je v tem, da je originalna koda neodvisna od dodelav in nespremenjena. Tako se lahko vsaka posebna dodelava naredi na višjem nivoju brez spremembe kode na nižjem nivoju.

Modeli predstavljajo logično grupiranje elementov znotraj nivoja. Vsak element je lahko samo v enem modelu na nivoju. Modele se lahko izvozi in instalira v poljubni »model store«, kar predstavlja preprost prenos kode med instancami. [20]

3.2.3 Programski jezik X++

Programski jezik X++ je podoben drugim programskim jezikom, kot sta C# in Java. Uporablja se kot jezik razvojnega okolja MorphX. Je objektni programski jezik z integriranimi ukazi za poizvedbe SQL. Za različne funkcionalnosti, kot so vhod/izhod operacije, XML, kontroliranje obnašanja uporabniškega vmesnika, se lahko uporablja obstoječe Dynamics AX razrede ali pa se jih razširi.

Za razvoj poslovne logike se uporablja urejevalnik X++ Editor. Urejevalnik omogoča lažji in hitrejši razvoj s pomočjo obarvanih besed in samodejnega izpolnjevanja kode. Tako na primer obarva rezervirane besede z modro, komentarje z zeleno, števila z rdečo, nize s temno rdečo, oznake z vijolično in ostalo kodo s črno barvo.

Programski jezik X++ je zelo podoben ostalim množično uporabljenim programskim jezikom. Za razvoj lahko uporabljamo enake kontrolne izraze kot drugod, med njih spadajo »for« zanke, »while« zanke, »switch« izrazi in »if-else« pogoji. Poleg osnovnih podatkovnih tipov podpira tudi sestavljene, kot so »array«, »container«, »map« ... Pozna aritmetične in logične operatorje. Za deklariranje metod poznamo tri vrste dostopa do njih: »public«, »protected«, »private«.

X++ je objektni jezik in podpira razvoj prek objektov, predstavljeni so kot razredi, ki vsebujejo statične in dinamične metode ter globalne in lokalne spremenljivke.

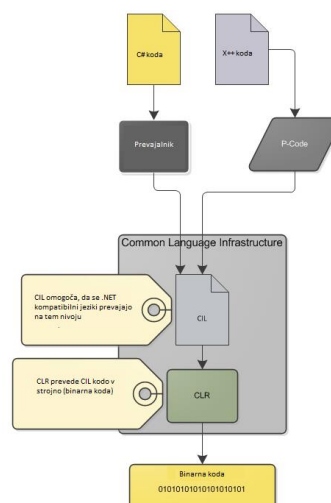
```
static void Job1 (Args _args)
{
}

```

Slika 11 Sintaksa kode [21]

X++ podpira samostojen sestav kode, imenovan »Job«, ki se ga lahko izvede iz urejevalnika. Slika 11 prikazuje primer sintakse. Največkrat se uporablja za testiranje kode, preden se kodo prenese v metode, tabele in razrede. Po sestavi je podobna metoda z razliko možnosti samostojnega izvajanja. [21]

Za razhroščevanje kode se uporablja integriran razhroščevalnik, ki podpira postavljanje kontrolnih točk, spremljanje stanja spremenljivk, ročno nastavljanje programskega števca ...



Slika 12 Postopek prevajanja [32]

Microsoft Dynamics AX lahko celotno kodo pretvori v CIL (Common Intermediate Language), ki se izvaja v .NET CLR (Common Language Runtime) (slika 12)[32].

CIL je v osnovi objektno orientiran zbirni jezik in predstavlja skupek ukazov, ki so neodvisni od platforme in CPE, kar omogoča izvajanje na različnih platformah, ki so znotraj CLI (Common Language Infrastructure) specifikacij. [22]

Prednost uporabe CLI je v tem, da je izvajanje hitrejše, kot v X++.

3.2.4 Dostop do podatkov

Microsoft Dynamics AX omogoča dostop do podatkov na dva načina. Prvi način je dostop do podatkovne baze prek integriranih ukazov. Za shranjevanje določenega zapisa iz baze služi »table buffer«. Je neke vrste objekt s podatki tabele iz baze. Za pridobivanje podatkov iz baze se uporablja »select« izraz, ki je podoben SQL izrazu.

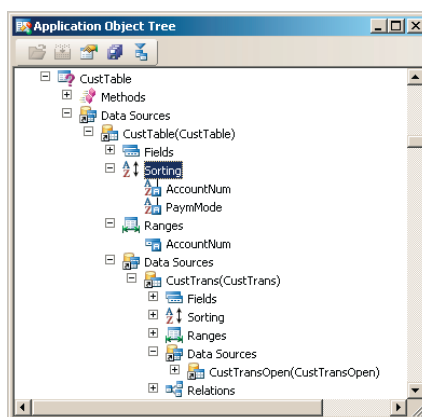
Vnos podatkov in posodobitev se vrši s pomočjo »table buffer-ja«, tako, da se mu nastavi želene vrednosti in se nato na njem pokliče metodo »insert« ali »update«. Lahko pa se uporabi masovno vnašanje ali posodabljanje preko izraza »insert_recordset« in »update_recordset«. Za pridobivanje več podatkov iz baze pa je namenjena poizvedbena zanka »while select«, kjer se zaporedno vrši poizvedovanje zapisov iz baze.

```

select <Find Options>
<TableBuffer or {FieldList from [TableBuffer]}>
<Aggregate (field identifier)>
<Sorting Options field identifier[Direction]>
<Index clause (index)>
where <selection criteria>
<Join Clause join>

```

Slika 13 Sintaksa poizvedbe [21]



Slika 14 »Query« v AOT [21]

Drug način poizvedovanja pa je prek elementa »Query«. »Query« lahko zgradimo na dva načina, v kodi ali preko uporabniškega vmesnika. Določimo mu podatkovne vire in relacije med njimi, ter grupiranja, sortiranja in omejitve. »Query« se uporablja za pridobivanje podatkov na obrazcih, poročilih ... Slika 13 prikazuje drevo »Query-ja«, slika 14 pa sintakso poizvedbe. [21]

Poglavje 4

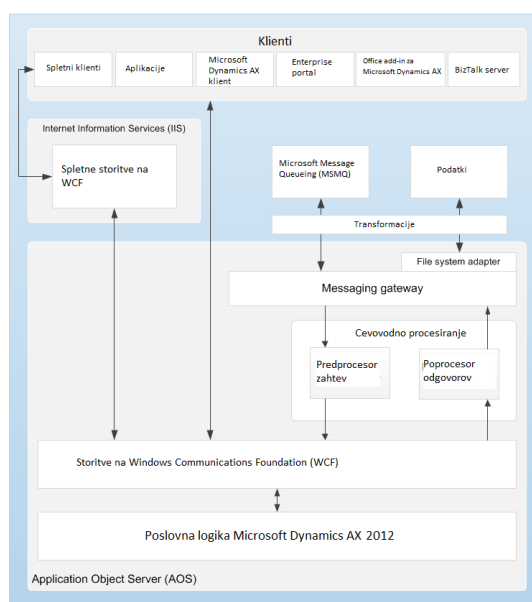
4 Integracija Microsoft Dynamics AX z zunanjimi storitvami

4.1 Možnosti, ki jih ponuja Microsoft Dynamics AX

Microsoft Dynamics AX uporablja za integracijo s storitvami znotraj in zunaj podjetja »Application Integration Framework« (AIF). Integracija je izvedena s pomočjo izmenjave podatkov v XML dokumentih. AIF omogoča podporo številnim sinhronim in asinhronim komunikacijam za izmenjavo XML dokumentov z zunanjimi sistemi prek spletnih storitev. Slika 15 prikazuje komunikacijo pri različnih integracijskih možnostih [34].

Obstajata dva načina za izmenjavo podatkov z AIF:

- spletne storitve omogočajo uporabo v zunanjih sistemih,
- adapterji (»Adapters«) omogočajo komunikacijo in izmenjavo podatkov Microsoft Dynamics AX z zunanjimi sistemi, kot so podatkovni sistem (»file system«), vrste sporočil (»MSMQ«) ali BizTalkServer.



Slika 15 Komunikacija pri integracijah [34]

4.2 SOA – Storitveno-orientirana arhitektura

SOA je oblika programa in programske arhitekture, kjer je funkcionalnost podana v obliki storitve. Storitev je zaključen del funkcionalnosti, ki jo program zagotavlja. Namen SOA je omogočiti sodelovanje različnih računalnikov med seboj, ki so povezani prek omrežja. Storitev običajno implementira eno akcijo, ki jo je možno storiti v aplikaciji. Za izmenjavo podatkov se večinoma uporabljata XML in protokol SOAP. Za opis storitev se uporablja WSDL (Web Service Description Language). Prednost uporabe SOA je v tem, da se lahko izmenjuje podatke med različnimi sistemi različnih proizvajalcev brez potrebe po razširjenem programiranju in spreminjanju kode. [23]

4.3 WCF – Windows Communication Foundation

WCF je ogrodje za razvoj SOA aplikacij, ki združuje že poznane tehnologije in koncepte Microsofta (Web Services Enhancements, MSMQ ...). Osnovna ideja WCF je bila postaviti model za razvoj, konfiguracijo in nalaganje storitev na omrežje. S pomočjo WCF-ja se lahko pošilja asinhrona sporočila od storitve do storitve. Sporočila so lahko v obliki XML ali kompleksnih binarnih tokov. Vsaka storitev v WCF-ju ima »endpoint«, preko katerih je možna komunikacija. Vsaka takšna točka ima:

- svoj URI naslov, kjer je dostopna,
- »binding«, ki pove kako je možno komunicirati, preko katerih protokolov, varnost in enkripcija, ki jo zahteva storitev,
- »contract«, ki pove kaj omogoča storitev. Te pogodbe se delijo na tri vrste. Prva je »service contract«, ki določa funkcionalnost storitve, ki je izpostavljena, druga »operation contract«, ki pove katere operacije zagotavlja storitev, ter »data contract«, ki določa obliko podatkov, ki so potrebni za izmenjavo informacij. [24]

4.4 *Storitve v Microsoft Dynamics AX*

Microsoft Dynamics AX deli storitve v tri skupine.

4.4.1 »Document services«

Vsi podatki v Microsoft Dynamics AX se nahajajo v tabelah. Te tabele so lahko med seboj povezane in med njimi obstajajo relacije. Prav tako pa je lahko na vsaki tabeli tudi poslovna logika, ki skrbi za poslovna pravila in validacijo. Za ta namen obstaja v Dynamics AX-u čarovnik za ustvarjanje storitev, ki omogoča izpostavitve teh tabel in pripadajoče logike zunanjim odjemalcem. Pri teh storitvah so pomembni štirje elementi. Najprej je potrebna poizvedba (»query«), ki določa povezave entitet in je osnova za čarovnik, ki ustvari storitev. Vsaka storitev ima tako za osnovo poizvedbo, iz katere nastane XML shema, ki zajema strukturo vsebovanih tabel in služi za serializacijo XML.

Nato so potrebni razredi »dokumenti«, ki predstavljajo entitete in poslovno logiko zanje. Zajemajo vso logiko uporabljenih tabel in jo skrijejo pred uporabnikom, tako da je le-ta neopazna. Prav tako ta razred skrbi za generiranje XSD sheme, ki določa strukturo in pravila. V XSD je zapisana tudi logika za serializacijo razredov v XML in njihovo deserializacijo iz XML. Glavne naloge tega razreda so generiranje XSD sheme, serializacija in deserializacija razredov in XML, zagotavljanje upoštevanja pravil poslovne logike in logike, povezane z vsemi tabelami, in upravljanje z napakami ter posredovanje le-teh odjemalcu.

Razredi »AxBC« so ovoji okoli tabel, ki skrbijo za manipulacijo z njimi. Skrbijo za dostop do podatkov in vsebujejo poslovno logiko, ki se drugače nahaja na obrazcih. Prav tako lahko specificirajo privzete vrednosti polj. »AxBC« razredi so zadolženi za validacijo. Poskrbijo tako za izvajanje poslovne logike kot tudi za integriteto podatkov. Prav tako določajo vrstni red procesiranja polj, kar je koristno v primeru, da so polja med seboj odvisna.

Storitveni razredi pa vsebujejo operacije, ki so uporabljene na integracijskih vratih za to storitev. Operacije, ki so na voljo, so »create«, ki prejme razred dokumenta in ustvari zapis, »delete«, ki izbriše zapis, »find«, ki vrne iskani zapis v obliki razreda, »findkeys«, ki vrne samo ključne iskanih zapisov, »read«, ki s pomočjo rezultatov funkcije »find« vrne vse zapise, ki so vsebovani, »update«, ki posodobi obstoječ zapis, »getKeys«, ki vrne ključne zapisov, ki so

filtrirani glede na kriterij, ter »getChangedKeys«, ki vrne vse ključe spremenjenih zapisov po določenem datumu. [22]

4.4.2 Storitve po meri

Poleg dokumentnih storitev Dynamics AX omogoča tudi razvoj storitev po meri. Razvoj takšnih storitev v Dynamics AX-u je zelo podoben razvoju WCF storitev v drugih programskih jezikih. Z atributi za ustvarjanje podatkovnih pogodb (»data contracts«) in storitvenih pogodb (»service contracts«) je razvoj olajšan, saj razvijalcu ni potrebno skrbeti za serializacijo in deserializacijo, za kar skrbi WCF. Glavne komponente storitev po meri so atributi podatkovne in storitvene pogodbe.

Atributi so razredi, ki so enaki ostalim razredom, le da je njihov namen drugačen. Vsebujejo metapodatke, ki opisujejo cilje. Cilji so lahko različni elementi, kot razredi, vmesniki, metode ali dogodki (»events«) ... Obstajata dva tipa atributov: notranji, ki so del CLR, in so vsebovani v .NET ogrodju, in atributi po meri, ki jih lahko ustvarja razvijalec. Največkrat uporabljeni atributi, ki se uporabljajo pri razvoju storitev po meri, so:

- SysEntryPointAttribute, ki je obvezen za metode, ki so izpostavljene kot operacije storitve,
- DataContractAttribute pove, da je razred uporabljen kot podatkovna pogodba za storitev,
- DataMemberAttribute pove, da je metoda, ki je parameter, uporabljena kot del podatkovne pogodbe,
- AIFCollectionTypeAttribute določa tip, ki je uporabljen v podatkovni strukturi za določen parameter.

»Data contracts« ali podatkovne pogodbe so določila, ki določajo tip podatkov, ki se bodo izmenjali med storitvijo in odjemalcem. S temi pogodbami serializira podatke, ki si jih izmenjujeta obe strani. Storitve uporabljajo podatkovne pogodbe za določanje parametrov in oblike odgovorov za svoje operacije. Poleg tako določenih tipov se za komunikacijo lahko

uporablja tudi primitivne tipe, razširjene podatkovne tipe ..., ki se lahko serializirajo brez podatkovnih pogodb.

Storitvene pogodbe ali »service contracts« v WCF opisujejo funkcionalnosti in operacije storitve. V Dynamics AX-u so te pogodbe razredi, ki tečejo na strežniku. V teh razredih je potrebno ustvariti metode z atributi SysEntryPointAttribute, ki povedo, da je ta metoda operacija storitve. [22]

4.4.3 Sistemske storitve

Sistemske storitve so bile v Dynamics AX-u dodane v zadnji verziji. So storitve, katerih ni moč spreminjati, in se nahajajo na AOS, na naslovih, definiranih v nastavitvah strežnika. Obstajajo trije tipi sistemskih storitev. »Query service« omogoča dostop in izvajanje poizvedb, definiranih v AOT. Za izvajanje poizvedb so potrebne pravice za dostop do podatkov. S pomočjo storitev »metadata service« lahko odjemalec dostopa do informacij o nekaterih ključnih metapodatkih. Dostopa se lahko do podatkov, katere storitve in poizvedbe so na voljo in podatkov o instancah strežnika. »User session service« pa omogoča pridobivanje podatkov odjemalca o uporabniku, ki želi dostopati do storitve. Te podatki so na primer podatki o podjetju uporabnika, časovni coni, potrebni lokalizaciji ... [22]

4.4.4 Omogočanje storitev

WCF storitve so se v prejšnjih verzijah izvajale prek adapterjev za BizTalk, MSMQ ali pa kot storitve WCF 3.5 prek IIS. Nove verzije pa omogočajo storitve WCF 4.0 direktno v AOS Windows storitvi. Kot vrata za dostopanje do storitev služijo WCF adapterji. Poleg standardnih adapterjev je možno razviti tudi svoje prek vmesnika »AifIntegrationAdapter«.

Adapterji, ki jih ponuja Microsoft Dynamics AX so:

- NetTcp adapter, ki je privzet adapter pri ustvarjanju integracijskih vrat. Omogoča sinhrono komunikacijo prek protokola TCP (»Transmission Control Protocol«),

- adapter za podatkovni sistem, ki uporablja asinhrono izmenjavo XML sporočil, shranjenih na podatkovnem disku,
- MSMQ adapter za podporo vrst. Izmenjava sporočil je asinhrona in se uporablja skupaj z MSMQ,
- HTTP adapter podpira sinhrono izmenjavo sporočil prek protokolov HTTP in HTTPS.

Integracijska vrata v Dynamics AX-u predstavljajo način administracije storitev. Vsaka vrata izpostavlja eno ali več storitev, ki se nahajajo na določenem URI naslovu. Vsaka vrata spadajo v eno od dveh skupin, glede na smer. Med »inbound ports« spadajo vrata, ki sprejemajo sporočila, ki izvirajo zunaj sistema Dynamics AX-a. Ta vrata so namenjena sprejemanju in procesiranju zahtev različnih odjemalcev. »Outbound ports« pa omogočajo komunikacijo oziroma pošiljanje sporočil sistema Dynamics AX zunanjim storitvam. Ta vrata lahko uporabljajo asinhrono komunikacijo prek adapterjev podatkovnega sistema ali MSMQ.

Za integracijska vrata obstajata dve kategoriji:

- Osnovna vrata so povezana na storitvene skupine v AOT in jih lahko ustvari le razvijalec. V kategorijo osnovnih vrat spadajo le »inbound« vrata in uporabljajo NetTcp adapter. Nastavljivi so WCF parametri in le nekatere druge stvari. Tako je njihova funkcionalnost nekoliko omejena.
- Izboljšana vrata pa omogočajo naprednejše možnosti nastavljanja integracijskih vrat. Vsa »outbound« vrata spadajo pod to kategorijo, prav tako pa lahko sem spadajo tudi »inbound« vrata. Tem vratom je možno nastaviti, da se nahajajo na AOS ali na IIS (Internet Information Services), omogočajo različne protokole (HTTP, NetTcp, MSMQ), omogočajo pred in po procesiranje zahtev in odgovorov ter napredne možnosti nastavljanja varnosti. [22]

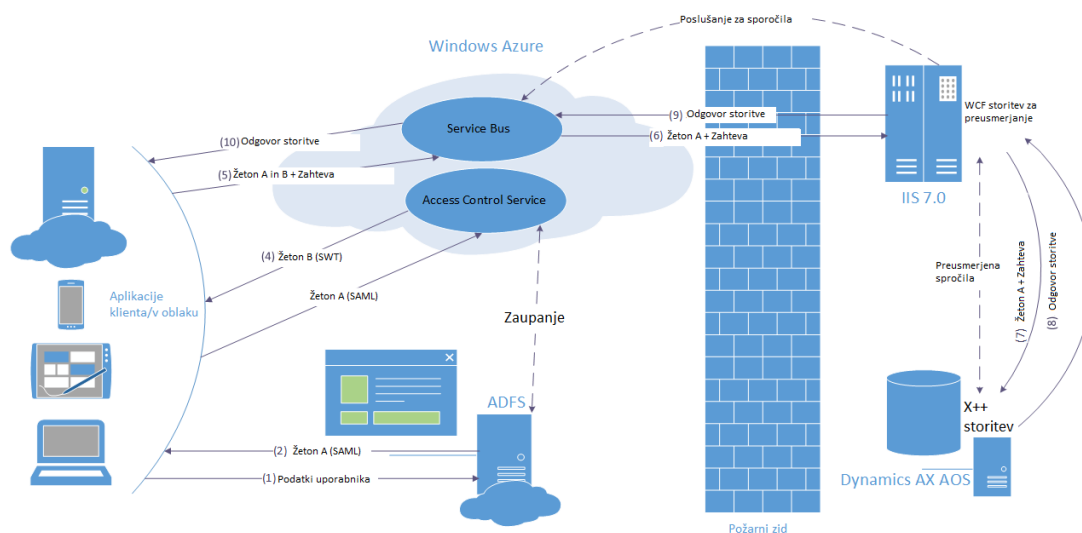
Poglavje 5

5 Mobilna aplikacija za Microsoft Dynamics AX

5.1 Koncept in opredelitev problema

Če želimo povezati mobilno aplikacijo z Dynamics AX-om, moramo poskrbeti za varno komunikacijo. Dynamics AX je postavljen za požarni zid in tako direktno ločen od spleta. Pri komunikaciji znotraj požarnega zidu ni tako kritično zagotavljanje varnosti, saj so vse aplikacije v omrežju podjetja. Pri izpostavitvi sistema navzven pa je kritično prav to, da je vsak dostop do sistema avtoriziran. Tako pravilo velja za vsako komunikacijo zunanjega sveta z Dynamics AX-om, tako da je pri razvoju mobilne aplikacije primarno potrebno poskrbeti za varno komunikacijo med aplikacijami.

AIF storitve v primeru mobilnih aplikacij same ne zadostujejo, saj ne omogočajo dostopa zunaj požarnega zidu. Tako je potrebno najti način, kako take storitve izpostaviti zunaj le-tega. Dynamics AX za ta namen uporablja »Azure Service Bus« releja. Slika 16 prikazuje postopek komunikacije odjemalca in Dynamics AX prek interneta. [26]



Slika 16 Postopek komunikacije pri uporabi Windows Azure [26]

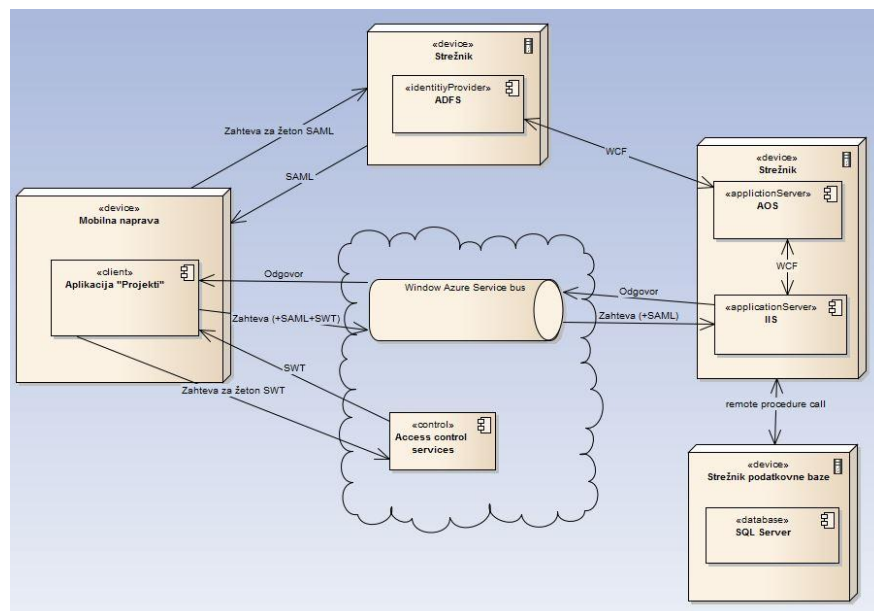
Za vzpostavitev komunikacije potrebujemo tri tehnologije. Poleg spletnega strežnika IIS še Windows Azure Service Bus in še način, kako avtenticirati uporabnika. Za ta namen se lahko uporabi ADFS.

- IIS (Internet Information Services) je spletni strežnik, ki ga je razvil Microsoft. Podpira protokole HTTP, HTTPS, FTP, FTPS, SMTP in NNTP in je del Windows NT verzije NT 4.0. [27]
- Windows Azure je platforma in infrastruktura v oblaku, ki jo je razvil Microsoft, in je bila izdana 1. februarja 2010. Omogoča tako PaaS (Platforma kot storitev), kot tudi IaaS (Infrastruktura kot storitev) in podpira številne programske jezike (Microsoftove in »third-party«) [28]. Z Azure je mogoče razvijanje, nalaganje in upravljanje z aplikacijami ter storitvami. Med možnostmi, ki jih ponuja, so spletne strani, virtualna okolja, mobilne storitve, storitve v oblaku, SQL podatkovne baze, hramba podatkov (»storage«), podatkovna skladišča (»HDInsight service«), multimedijske storitve, »service bus«, Visual Studio Online, BizTalk storitve, »recovery« storitve, mreže, upravljanje prometa, »manegment« storitve in »Active directory« [29].
- ADFS (Active Directory Federation Services) je programska komponenta, ki jo je možno namestiti na operacijske sisteme Windows Server. Predstavlja točko, kjer se lahko uporabniki identificirajo za uporabo aplikacij in sistemov prek celotne organizacije. Uporablja dostopno avtorizacijo glede na zahteve. Za dostop se uporablja žetone, ki vsebujejo podatke o identiteti uporabnika. [30]

5.2 Razvoj aplikacije

5.2.1 UML diagram postavitve

Za implementacijo aplikacije so potrebni poleg AOS in pripadajoče podatkovne baze tudi strežnik z ADFS, ki služi za potrditev uporabnikovih podatkov, Windows Azure Service bus, ki preusmerja zahteve na AOS, in Access Control Services, ki omogočajo varno komunikacijo z Azure. Poleg teh komponent pa je potrebna tudi mobilna naprava, ki gostuje mobilno aplikacijo. Slika 17 prikazuje postavitev takšnega sistema.



Slika 17 UML diagram postavitve

5.2.2 AIF storitev

V prvem koraku je potrebno razviti storitev, ki bo predstavljala poslovno logiko na strani Dynamics AX. V našem primeru bomo delovali v okvirju modula projektov. Za primer bomo razvili storitev z dvema funkcijama. Prva bo vračala seznam vseh projektov, ki jih imamo vnesene v sistem, druga pa bo na določen projekt knjižila delovne ure uporabnika. V tem delu potrebujemo tri razrede. Dva razreda bosta vsebovala podatkovno pogodbo, tretji pa dejansko storitev.

5.2.2.1 Podatkovna pogodba (Data contract)

Najprej je potrebno definirati parametre, ki jih bo pogodba vsebovala. Za prenos podatkov o projektih bomo potrebovali identifikacijsko številko projekta, naziv, status in tip projekta. V deklaraciji razreda navedemo spremenljivke in definiramo parametrične metode. Da razred predstavlja podatkovno pogodbo, nad deklaracijo razreda navedemo podatek »DataContractAttribute« (Koda 1), nad definicijo parametra pa »DataMemberAttribute« (Koda 2).

```
[DataContractAttribute]
public class Dipl_ProjDC
{
    ProjId projId;
    ProjName projName;
    ProjType projType;
    ProjStatus projStatus;
}
```

Koda 1 Definicija podatkovne pogodbe

```
[DataMemberAttribute]
public ProjId parmProjId(ProjId _projId = projId)
{
    projId = _projId;
    return projId;
}
```

Koda 2 Definicija parametra

Postopek ponovimo za drugo pogodbo. Za vnos ur delavca potrebujemo podatke: delavec, število ur, cena ure, datum transakcije, id projekta, na katerega se naj knjiži, in dodaten opis.

5.2.2.2 Storitve

Prva funkcija storitve je pridobitev seznama vseh projektov. Tako je potrebno implementirati metodo, ki bo iz baze pridobila potrebne podatke (Koda 3). Da metoda razreda predstavlja funkcijo storitve, je definirano z atributom »SysEntryPointAttribute«, ki je nastavljen na »true«. S tem povemo, da ta metoda predstavlja vstopno točko. Ker metoda vrača seznam razredov, je potrebno to navesti z »AifCollectionTypeAttribute('return', Types::Class, classStr(Dipl_ProjDC))«. Prvi atribut nam pove, kje se bo pojavil seznam, drugi pove, katerega tipa so elementi, tretji pa, za kateri razred gre. V metodi je izveden klic na SQL, kjer se prebere v »while« zanki vse projekte in podatke zapisuje v instanco objekta. Nato se vsak objekt doda v seznam, ki predstavlja rezultat. Podobno storimo za funkcijo vnosa ur.

```
[AifCollectionTypeAttribute('return', Types::Class, classStr(Dipl_ProjDC)),
 SysEntryPointAttribute(true)]
public List getAllProjects()
{
    List projDCList;
    ProjTable proj;
    Dipl_ProjDC projRet;
    projDCList = new List(Types::Class);

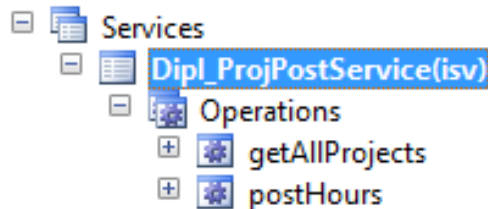
    while select * from proj
    {
        projRet = new Dipl_ProjDC();
        projRet.parmProjId(proj.ProjId);
        projRet.parmProjName(proj.Name);
        projRet.parmProjStatus(proj.Status);
        projRet.parmProjType(proj.Type);
        projDCList.addEnd(projRet);
    }

    return projDCList;
}
```

Koda 3 Funkcija storitve (»Entry point«)

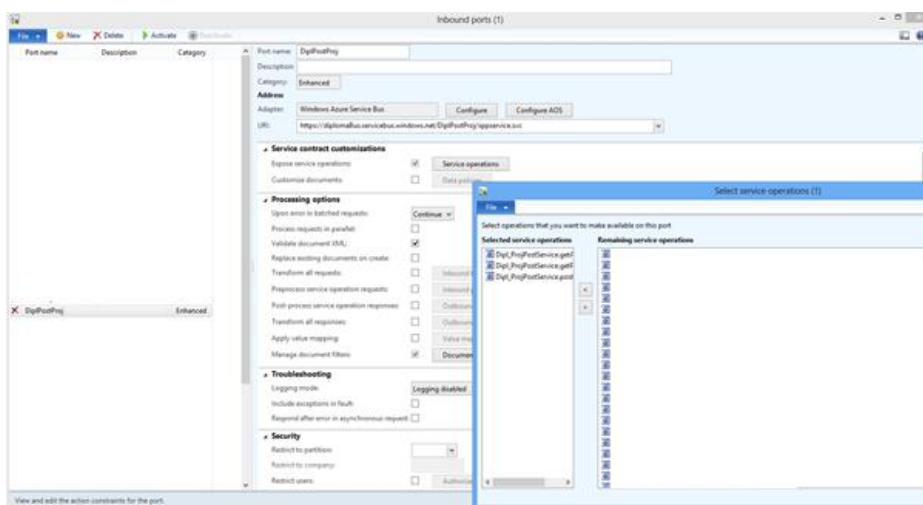
5.2.2.3 Izpostavitve storitve

Da lahko storitev izpostavimo, je potrebno definirati operacije. Definiramo storitev, ki ji določimo, da je storitev razred, ki smo ga razvili, in dodamo operacije (Slika 18).



Slika 18 Storitve v AOT

Nato je potrebno kreirati port, kjer bo storitev dosegljiva. Za naše namene bomo uporabili »inbound port« kategorije »enhanced«. Slika 19 prikazuje zaslonsko masko portov in izbiro operacij, ki bodo dosegljive. Kot adapter pa je izbran »Windows Azure Service Bus«, ki je bil predhodno konfiguriran. Za konfiguracijo je potrebno implementirati tudi »management« komponente, ki skrbijo za avtentikacijo zahteve. V tem primeru je uporabljena koda pridobljena na <http://blogs.msdn.com/b/aif/archive/2013/04/29/aif-windows-azure-service-bus-adapter.aspx>, ki zadostuje za osnovne potrebe dostopanja do Dynamics AX-a preko »service busa«.



Slika 19 Zaslonska maska nastavitve porta

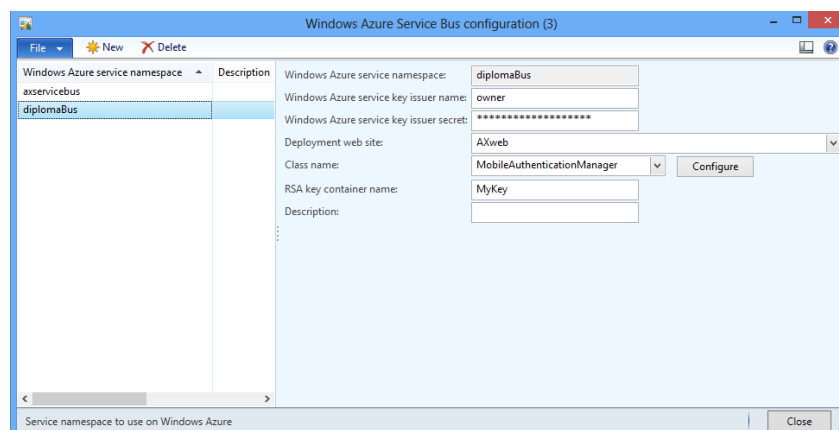
5.2.2.4 Avtentikacija na strani Dynamics AX

Avtentikacijska komponenta za AX je sestavljena iz dveh delov. Nastavitve in parametri se nahajajo v AX tabeli, ki se vnesejo preko obrazca v AX. Ko AX želi izvesti avtentikacijo, dobi parametre s pomočjo X++ razreda. Avtentikacija zahteve pa se izvede v kodi C#. Ko pride zahteva po storitvi, je potrebno iz nje izluščiti žetone in jih validirati. Preveri se, ali je žeton izdan za pravilen imenski prostor. Nato se iz žetona prebere identifikacijske podatke, ki so posredovani AX. AX nato preveri identiteto in omogoči klic storitve.

5.2.3 Windows Azure Service bus

Praden lahko uporabljamo storitev, ki nam omogoča dostop do Dynamics AX-a, moramo nastaviti »service bus«. Na spletnem upravljalcu Windows Azure je potrebno kreirati nov »bus« in nastaviti »Access control service«, ki bo skrbel za način avtentikacije. V tem koraku se vzpostavi zaupanje med »service busom« in ADFS. ADFS služi kot potrjevalec identitete klicatelja storitve. Avtentikacija v primeru AX poteka preko Windows identitete. Tako je potrebno ob dostopu do storitve podati domeno, uporabniško ime in geslo. Zaradi načina avtentikacije so podani podatki vidni le uporabniku in ADFS-ju, ki na osnovi podatkov potrdi identiteto.

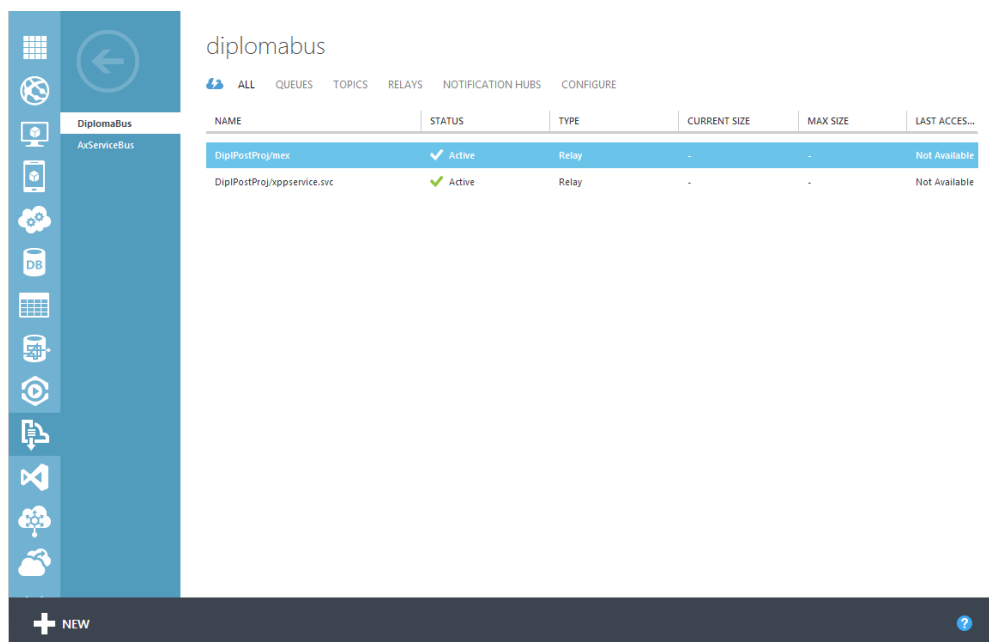
Za uporabo »service busa« je potrebno v AX nastaviti podatke o imenskem prostoru, ki definira kateri »service bus« se bo uporabljal za izpostavitve storitve na spletu.



Slika 20 Nastavitve Windows Azure v Dynamics AX

Na sliki 20 so prikazane nastavitve za »service bus«. Podatki o izdajalcu ključa so podani v nastavitvah v upravljavcu Windows Azur. Spletna stran, kjer se nahaja storitev, je nastavljena v Dynamics AX-u. Potrebno je določiti tudi razred, ki skrbi za avtentikacijo na strani AX. Prav tako pa je potrebno izdelati in definirati RSA ključ za potrebe komunikacije med strežnikom, na katerem se nahaja AOS in Window Azure service bus.

Ko so vse potrebne nastavitve definirane, je storitev pripravljena in ob aktivaciji porta tudi vidna na strani Azurja. Slika 21 prikazuje stanje storitve na upravljavcu Azure.



The screenshot shows the Windows Azure Service Bus portal for a namespace named 'diplomabus'. The interface includes a left-hand navigation pane with various service bus components and a main content area displaying a table of relays. The table has columns for NAME, STATUS, TYPE, CURRENT SIZE, MAX SIZE, and LAST ACCESS... Two relays are listed, both with a status of 'Active' and a type of 'Relay'. The current and maximum sizes are both indicated as '-' (empty), and the last access time is 'Not Available'.

NAME	STATUS	TYPE	CURRENT SIZE	MAX SIZE	LAST ACCESS...
DiplPostProj/mex	Active	Relay	-	-	Not Available
DiplPostProj/xppservice.svc	Active	Relay	-	-	Not Available

Slika 21 Status storitve na Windows Azure Service bus

5.2.4 Mobilna aplikacija »Windows Phone App«

Ko je storitev na strani AX pripravljena in dostopna prek »service busa«, jo je potrebno s strani mobilne aplikacije pravilno poklicati. V aplikacije je potrebno dodati referenco na storitev in nastaviti »binding« ter nastavitve za klienta, ki bodo omogočale povezavo s storitvijo.

Pri klicu storitve je potrebno izvesti avtentikacijo in izmenjavo žetonov. Ogrodje .NET 4.5 Mobile omogoča asinhron klic storitve. Preden lahko storitev pokličemo, je potrebno

pridobiti SAML in SWT žetone. SAML žeton je potreben za avtentikacijo na AOS, SWT pa za komunikacijo z Azure.

V aplikaciji bi radi prikazali seznam vseh projektov, ki so vneseni v AX. Na osnovni strani aplikacije je gumb, ki nas preusmeri na seznam le-teh (slika 22).



Slika 22 Zaslonska maska prve strani

Ob inicializaciji strani se pokliče storitev, ki vrne seznam objektov vseh projektov. Preden pa se lahko izvede klic, je potrebno izpeljati pridobitev žetonov. Za ta namen obstaja koda, ki poskrbi za pravilne formate in obdelave podatkov med komunikacijo.

```
public void getProjsList()
{
    authenticator = new AuthenticationProvider(@"domena\UporabnikoIme", @"Geslo",
"serviceBusImenskiProstor", "ADFSServiceUrl");
    authenticator.AuthenticationCompleted += authenticator_AuthenticationCompleted;
    authenticator.IssueToken();
}
```

Koda 4 Inicializacija in klic avtentikacije

Funkcija »getProjsList« (koda 4) inicializira avtentikacijski razred. Kot parametre sprejme domeno in uporabniško ime uporabnika, geslo, imenski prostor »service busa« ter url naslov storitve ADFS. Prav tako se registrira »event handler« za zaključek avtentikacije. Na koncu se pokliče metodo za pridobitev žetonov.

```
private void SendRequestSecurityToken()
{
    DateTime stsAuthStartTime = DateTime.UtcNow;
    requestSecurityTokenPayload = String.Format(this.StsUrlPayload, this.stsEndpoint,
    stsAuthStartTime.ToString("s") + ".000", stsAuthStartTime.AddMinutes(5).ToString("s") + ".000",
    credentials.Alias, credentials.Password, this.AppliesTo);
    try
    {
        HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(stsEndpoint);
        httpWebRequest.ContentType = "application/soap+xml; charset=utf-8";
        httpWebRequest.Method = "POST";
        httpWebRequest.BeginGetRequestStream(new
    AsyncCallback(RequestSecurityTokenStreamCallback), httpWebRequest);
    }
    OnAuthenticationCompleted(new AuthenticationMessageArgs(new
    AuthenticationMessage(e.Message, false)));
    }
}
```

Koda 5 Pošiljanje zahteve za SAML žeton

Pri zahtevi za SAML žeton je potrebno pripraviti zahtevek v določenem formatu, ki ga bo ADFS obdelal (koda 5).

```
private void RequestSecurityTokenStreamCallback(IAsyncResult asynchronousResult)
{
    HttpRequest httpRequest = (HttpRequest)asynchronousResult.AsyncState;
    Stream requestStream = null;

    try
    {
        using (requestStream = httpRequest.EndGetRequestStream(asynchronousResult))
        {
            if (requestStream != null)
            {
                byte[] bytes = Encoding.UTF8.GetBytes(requestSecurityTokenPayload);
                requestStream.Write(bytes, 0, bytes.Length);
            }

            httpRequest.BeginGetResponse(new
                AsyncCallback(RequestSecurityTokenResponseStreamCallback),
                httpRequest);
        }
        catch (Exception e)
        {
            OnAuthenticationCompleted(new AuthenticationMessageArgs(
                new AuthenticationMessage(e.Message, false)));
        }
    }
}

private void RequestSecurityTokenResponseStreamCallback(IAsyncResult asynchronousResult)
{
    HttpRequest httpRequest = (HttpRequest)asynchronousResult.AsyncState;

    try
    {
        HttpResponse httpResponse =
            (HttpResponse)httpRequest.EndGetResponse(asynchronousResult);

        requestSecurityTokenResponse = Utilities.GetResponseStringFromResponse(httpResponse);

        SendTokenToAcs();
    }
    catch (WebException e)
    {
        string responseString = Utilities.GetResponseStringFromResponse(
            (HttpResponse)e.Response);

        string errorStr = Utilities.GetErrorFromStsResponseString(responseString);

        if (errorStr.Contains("FailedAuthentication"))
        {
            errorStr = String.Format("ERROR");
        }

        OnAuthenticationCompleted(new AuthenticationMessageArgs(new
            AuthenticationMessage(errorStr, false)));
    }
}
}
```

Koda 6 Obdelava odgovora ADFS

Ob prejemu odgovora s strani ADFS je potrebno obdelati povratne informacije. Tako ugotovimo ali je avtentikacija uspela. Koda 6 prikazuje dve metodi. Prva se proži ob prejemu

odgovora in obdela tok podatkov. Druga pa iz toka podatkov razbere informacije, ki jih je posredoval ADFS. Če je avtentikacija uspešna, smo pripravljeni, da začnemo s komunikacijo prek Azure »service bus«.

```
private void SendTokenToAcs()
{
    const string acsPayloadFormat = @"wrap_scope={0}&wrap_assertion={1}&wrap_assertion_format=SAML";
    SamlAssertionToken = Utilities.GetSamlTokenFromResponseString(requestSecurityTokenResponse);

    if (!String.IsNullOrEmpty(SamlAssertionToken))
    {
        acsPayload = String.Format(acsPayloadFormat, Uri.EscapeDataString(RelyingParty),
            Uri.EscapeDataString(SamlAssertionToken));

        try
        {
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(AcsEndPoint);
            request.ContentType = @"application/x-www-form-urlencoded";
            request.Method = "POST";
            request.BeginGetRequestStream(new AsyncCallback(RequestAcsTokenStreamCallback),
                request);
        }
        catch (WebException e)
        {
            OnAuthenticationCompleted(new AuthenticationMessageArgs(
                new AuthenticationMessage(e.Message, false)));
        }
    }
    else
    {
        OnAuthenticationCompleted(new AuthenticationMessageArgs(
            new AuthenticationMessage("SAML Token from STS is empty", false));
    }
}

public static string GetSamlTokenFromResponseString(string responseString)
{
    string samlToken = string.Empty;
    StringReader reader = null;
    try
    {
        reader = new StringReader(responseString);
        using (XmlReader xmlparser = XmlReader.Create(reader))
        {
            reader = null;
            while (xmlparser.Read())
            {
                switch (xmlparser.NodeType)
                {
                    case XmlNodeType.Element:
                        string name = xmlparser.Name;
                        if (String.Equals(xmlparser.Name, "saml:Assertion"))
                        {
                            samlToken = xmlparser.ReadOuterXml();
                        }
                        break;
                }
            }
        }
    }
    finally
    {
        if (reader != null)
        {
            reader.Dispose();
        }
    }
    return samlToken;
}
```

Preden pošljemo zahtevek na »service bus« je potrebno v zahtevo vstaviti SAML žeton. Do žetona je potrebno priti z obdelavo ADFS XML odgovora. Ko razberemo žeton, ga je potrebno vstaviti v zahtevo, ki jo bomo poslali na »service bus«. (Koda 7)

»Service bus« obdela našo zahtevo in nam vrne odgovor, ki vsebuje SWT žeton, ki ga bomo potrebovali za komunikacijo s »service busom«. Postopek prejemanja odgovora je podoben, kot pri ADFS klicu. Najprej je potrebno prejeti podatkovni tok in ga nato obdelati. Koda 8 prikazuje obdelavo povratnih informacij in pridobitve SWT žetona. Če je celotna avtentikacija uspešna, se lahko začne klic storitve.

```
private void RequestAcsTokenResponseCallback(IAsyncResult asynchronousResult)
{
    HttpRequest request = (HttpRequest)asynchronousResult.AsyncState;

    try
    {
        HttpResponseMessage response =
        (HttpResponse)request.EndGetResponse(asynchronousResult);
        string rawAcsResponse = Utilities.GetResponseStringFromResponse(response);

        SwtAcsToken = Uri.UnescapeDataString(rawAcsResponse.Split('&').Single(value =>
        value.StartsWith("wrap_access_token=", StringComparison.OrdinalIgnoreCase)).Split('=')[1]);

        if (!String.IsNullOrEmpty(this.SwtAcsToken))
        {
            OnAuthenticationCompleted(new AuthenticationMessageArgs(new
            AuthenticationMessage("Security Token from ACS issued successfully", true)));
        }
    }
    catch (WebException e)
    {
        string responseString =
        Utilities.GetResponseStringFromResponse((HttpResponse)e.Response);
        OnAuthenticationCompleted(new AuthenticationMessageArgs(new
        AuthenticationMessage(responseString, false)));
    }
    catch (Exception e)
    {
        OnAuthenticationCompleted(new AuthenticationMessageArgs(new
        AuthenticationMessage(e.Message, false)));
    }
}
```

Koda 8 Obdelava odgovora Window Azure Servic bus

Ob zaključku uspešne avtentikacije se dvigne »event handler«, ki izvede dejanski klic potrebne storitve.

Ker bi radi pridobili podatke o projektih, je potrebno poklicati storitev »getAllProjects«. Potrebno je definirati »proxy«, ki bo klical storitev in naslov, kjer se storitev nahaja. Ker je klic storitve asinhron, je potrebno povezati metodo, ki se bo prožila ob prejemu odgovora.

```

private void GetProjs()
{
    DiplomaAzure.DiplomaServiceClient client = new DiplomaAzure.DiplomaServiceClient();
    client.Endpoint.Address = new EndpointAddress(new Uri(
        "serviceAddress"));

    client.GetAllProjectsCompleted += new EventHandler
        <DiplomaAzure.GetAllProjectsCompletedEventArgs>
        (onGetAllProjCompleted);

    using (new OperationContextScope(client.InnerChannel))
    {
        byte[] binToken = Encoding.UTF8.GetBytes(authenticator.SamlAssertionToken);
        var customSecurityHeader = MessageHeader.CreateHeader("PassthroughBinarySecurityToken",
            "",
            Convert.ToBase64String(binToken));
        OperationContext.Current.OutgoingMessageHeaders.Add(customSecurityHeader);
        OperationContext.Current.OutgoingMessageHeaders.Add(new AcsHeader(authenticator.SwtAcsToken));

        client.GetAllProjectsAsync();
    }
}

```

Koda 9 Klic storitve skupaj z avtentikacijskimi podatki

Prav tako pa je potrebno v glavi zahtevka podati avtentikacijske podatke, ki smo jih pridobili. Sedaj lahko izvedemo klic storitve. (Koda 9)

5.2.4.1 Grafični vmesnik

Za prikaz rezultatov je potrebno pripraviti grafični vmesnik. Windows mobilne aplikacije imajo grafični vmesnik predstavljen z deklarativnim jezikom XAML (Extensible Application Markup Language). Z njim inicializiramo objekte in jim nastavimo lastnosti. Prav tako je z njim predstavljena hierarhična struktura objektov. Koda je shranjena v XML datoteki s končnico »xaml«. XAML datoteka je primarni format Windows mobilne aplikacije in predstavlja eno stran aplikacije. [31]

Da bi lahko prikazali seznam vseh projektov, je potrebno uporabiti element, ki bo omogočal povezavo na podatkovno strukturo, ki jo bomo prejeli od storitve. V našem primeru je to »ListBox«. V kontroli sta dve polji, ki sta povezani na podatke. (Koda 10)

```

<ListBox Name="listtest"
  Margin="0,0,0,71"
  SelectionChanged="listtest_SelectionChanged"
  Width="auto">
  <ListBox.ItemTemplate>
    <DataTemplate>
      <StackPanel Orientation="Horizontal" Margin="10" Width="auto">
        <TextBlock Text="{Binding parmProjId}" Margin="2,2,2,2"></TextBlock>
        <TextBlock Text="{Binding parmProjName}" Margin="2,2,2,2"></TextBlock>
      </StackPanel>
    </DataTemplate>
  </ListBox.ItemTemplate>
</ListBox>

```

Koda 10 Grafična kontrola »ListBox« in povezava polj na seznam rezultatov

Ko klic storitve prejme odgovor, lahko podatke povežemo na kontrolo. To storimo tako, da dobljen seznam projektov priredimo lastnosti »ItemsSource« naše kontrole. (Koda 11)

```

public void updateList(List<Dipl_ProjDC> list)
{
  Deployment.Current.Dispatcher.BeginInvoke(()=>
  {
    this.EnableUIAfterLoad();
    listtest.ItemsSource = list;
  });
}

```

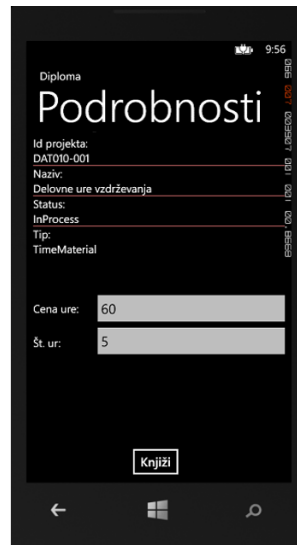
Koda 11 Povezovanje kontrole s seznamom

Slika 23 prikazuje rezultat izvedenega klica storitve. Prikazan je seznam vseh projektov z identifikacijo projekta in nazivom.



Slika 23 Zaslonska maska prikaza vseh projektov

Če želimo pridobiti podrobnejši pregled določenega projekta, ga s klikom lahko odpremo. Ker je klicana storitev vrnila seznam objektov, imamo vse potrebne podatke že na strani klienta in ponoven klic storitve ni potreben. Izbira projekta je izvedena preko akcije »selectionChange« na kontroli, ki je vsebovana v kontroli »ListBox«, »StackPanel«. Na strani podrobnosti projekta so navedeni podatki projekta (identifikacija, naziv, stanje in tip). Poleg osnovnih podatkov je omogočeno zapisovanje opravljenih ur na projekt. (Slika 24)



Slika 24 Zaslonska maska podrobnosti projekta

Za knjiženje je potrebno na zaslonski maski vnesti ceno opravljene ure in število ur ter klikniti na gumb knjiži. Knjiženje predstavlja klic storitve, ki prejme podatke, ki jih želimo knjižiti, in jih poknjiži v Dynamics AX-u. Za klic je potrebno imeti pripravljen žeton, ki smo ga pridobili ob avtentikaciji. Če ga ni ali pa je neveljaven, ga je potrebno ponovno pridobiti.

V kodi 12 je postopek priprave podatkov, ki jih prejme storitev, in klic storitve. Postopek za pripravo žetonov je enak kot pri prvem klicu storitve. Nato pa je potrebno pripraviti podatke za klic. Storitve za knjiženje prejme seznam objektov tipa »Dipl_HourPostDC«. V našem primeru bo seznam vseboval en element. Definirati je potrebno ceno opravljene ure in število ur. Ta podatek preberemo iz kontrol na zaslonski maski. Poleg tega je potrebno podati datum transakcije, ki je v našem primeru kar trenutni datum, id projekta, ki ga pridobimo iz argumenta, ki smo ga podali pri prehodu na trenutno stran, ter številko delavca. V našem primeru je številka delavca konstanta. Nato se objekt doda v seznam, seznam pa poda kot argument pri klicu storitve.

5.2.5 Integracija z iOS, Android in ostale platforme

Primer, ki je bil predstavljen v diplomski nalogi je bil implementiran s pomočjo .NET Mobile 4.5 za Windows Phone. Če bi želeli podobno aplikacijo razviti za operacijski sistem Android ali iOS, bi bil postopek zelo podoben. Pri obeh je potrebno klicati storitev, pri čemer je potrebno »na roke« kreirati SOAP zahtevo. Rezultat, ki je v obliki XML, pa je potrebno s »parserjem« obdelati in iz njega razbrati rezultat storitve. Za Android pa poleg takšnega pristopa obstaja tudi knjižnica (KSOAP), ki nam je v pomoč pri kreiranju zahtev in pri obdelavi odgovorov na osnovnem nivoju [33].

Prav tako se lahko tak pristop uporabi za razvoj aplikacije na katerikoli platformi, ki je zmožna klicati SOAP storitve in obdelati odgovore. Torej bi lahko takšen pristop uporabili tudi za razvoj spletne aplikacije, ki bi bila dostopna preko spletnega brskalnika.

Poglavje 6

6 Zaključek

V diplomski nalogi je bil predstavljen ERP sistem Microsoft Dynamics AX 2012 R2 in integracija le-tega z mobilno aplikacijo.

Najprej je bila predstavljena konkurenca izdelkov na trgu ERP sistemov in opisane možnosti, ki jih ponujajo pri integraciji z mobilnimi platformami. Zanimivo je to, da so načeloma vsi sistemi zelo podobni in tako je tudi pri integracijah. Pristop pri implementaciji integracijskih rešitev, med katere spada tudi razvoj mobilne aplikacije, je pri vseh zelo podoben in na tak ali drugačen način izkorišča pristop s storitvami.

V diplomski nalogi je bil podrobneje predstavljen sistem Dynamics AX in njegovo razvojno okolje. Predstavljene so bile integracijske možnosti, ki jih sistem ponuja, in opisana tehnologija, ki jih omogoča. Pri podrobnejšem pregledu rešitev za Microsoft Dynamics AX 2012 sem ugotovil, da je osnovna ideja tega sistema pri integraciji prav komunikacija preko storitve. Sistem sam po sebi ponuja veliko fleksibilnost pri implementaciji storitev v splošnem. Problem pri integraciji z mobilnimi napravami se pojavi predvsem pri zagotavljanju varnosti pri dostopih, saj je potrebno zagotoviti dostop iz omrežji, ki se nahajajo zunaj zasebnega omrežja podjetja.

V zadnjem delu je predstavljena rešitev, ki omogoča klicanje storitev, ki so bile pripravljene za integracijo. Ugotovil sem, da so za integracijo potrebne različne tehnologije, ki omogočajo varno komunikacijo preko spleta. Pri Microsoftu so združili različne tehnologije in storitve ter omogočili, da je razvoj takih aplikacij lažji. Poleg storitev, ki jih izpostavi Dynamics AX, je preferenčna uporaba Windows Azur Service busa, ki omogoča posredovanje zahtevkov do AX na varen način.

Pri delu diplomske naloge sem spoznal, na kakšne načine je možno izvesti implementacijo mobilnih aplikacij za Dynamics AX in kakšne ovire se lahko pojavijo pri tem. Spoznal sem način delovanja Windows Azur Service busa in ACS-a ter različnih protokolov, ki so povezani z overjanjem uporabnikov.

Literatura

7 Literatura

- [1] SAP history, pridobljeno decembra 2013 na: <http://global.sap.com/corporate-en/our-company/history/1972-1981.epx>
- [2] Oracle E-Business Suite overview, pridobljeno decembra 2013 na:
<http://www.oracle.com/us/products/applications/ebusiness/overview/index.html>
- [3] Sage ERP X3, pridobljeno decembra 2013 na: <http://na.sage.com/us/erp/sage-erp-x3>
- [4] Infor ERP, pridobljeno decembra 2013 na: <http://www.infor.com/solutions/erp>
- [5] L. Columbus, 2013 ERP Market Share Update: SAP Solidifies Market Leadership, 12.5.2013, pridobljeno decembra 2013 na:
<http://www.forbes.com/sites/louiscolumbus/2013/05/12/2013-erp-market-share-update-sap-solidifies-market-leadership/>
- [6] R/3 Architecture, pridobljeno decembra 2013 na:
http://help.sap.com/saphelp_46c/helpdata/en/ea/2941373c1ede6fe10000009b38f936/content.htm
- [7] SAP ERP solutions overview, pridobljeno decembra 2013 na:
<http://www.sap.com/pc/bp/erp/software/overview.html>
- [8] Top 5 ERP Client/Server ERP software applications, pridobljeno decembra 2013 na:
<http://www.erpsoftware360.com/erp-software.htm>
- [9] Mobile enterprise application platform, pridobljeno decembra 2013 na:
http://en.wikipedia.org/wiki/Mobile_enterprise_application_platform
- [10] Middleware, pridobljeno decembra 2013 na: <http://en.wikipedia.org/wiki/Middleware>
- [11] Thick client, pridobljeno decembra 2013 na:
<http://www.techterms.com/definition/thickclient>
- [12] S. Santhana, A. Kumar, *Mobilizing SAP Enterprise Applications, Building Tomorrow's Enterprise*. Bangalore: Infosys Limited, maj 2011.

- [13] *Mobilizing Oracle Applications ERP (An Approach for Building Scalable Mobility Solutions)*, white paper, RapidValue, marec 2013.
- [14] *Develop Mobile Applications With Oracle ADF Mobile*, Oracle, 2011.
- [15] *Sage ERP X3*, Sage Software, Inc, 2010.
- [16] *Sage Enterprise Webtop*, Sage Software, Inc, 2011.
- [17] *Infor Motion*, Infor, 2013.
- [18] Microsoft Dynamics AX, pridobljeno decembra 2013 na:
http://en.wikipedia.org/wiki/Microsoft_Dynamics_AX
- [19] Dynamics AX Milestones, pridobljeno decembra 2013 na:
<https://social.technet.microsoft.com/wiki/contents/articles/5812.dynamics-ax-milestones.aspx>
- [20] *Development 1 in Microsoft Dynamics AX 2012*, Microsoft, 2011.
- [21] *Development 2 in Microsoft Dynamics AX 2012*, Microsoft, 2011.
- [22] K. Deforche, K. Saelen, *Microsoft Dynamics AX 2012 Services*. Birmingham: Packt Publishing Ltd, december 2012
- [23] SOA, pridobljeno decembra 2013 na: http://en.wikipedia.org/wiki/Service-oriented_architecture
- [24] WCF, pridobljeno decembra 2013 na: [http://msdn.microsoft.com/en-us/library/dd456779\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/dd456779(v=vs.110).aspx)
- [25] J. Pandya, R. Drollinger, *Developing secure mobile apps for Microsoft Dynamics AX 2012*, white paper, maj 2013, pridobljeno decembra 2013 na:
<http://www.microsoft.com/enus/download/details.aspx?id=38413>
- [26] Microsoft Dynamics AX Services and Windows Azure Service Bus, pridobljeno decembra 2013 na:
<http://blogs.msdn.com/b/aif/archive/2013/04/29/aif-windows-azure-service-bus-adapter.aspx>
- [27] IIS, pridobljeno decembra 2013 na:
http://en.wikipedia.org/wiki/Internet_Information_Services

- [28] Windows Azure, pridobljeno decembra 2013 na:
http://en.wikipedia.org/wiki/Windows_Azure
- [29] Windows Azure documentation, pridobljeno decembra 2013 na:
<http://www.windowsazure.com/en-us/documentation/>
- [30] ADFS, pridobljeno decembra 2013 na:
http://en.wikipedia.org/wiki/Active_Directory_Federation_Services
- [31] XAML for Windows Phone, pridobljeno decembra 2013 na:
[http://msdn.microsoft.com/en-us/library/windowsphone/develop/cc189036\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/cc189036(v=vs.105).aspx)
- [32] AX 2012 CIL - How does it work?, pridobljeno decembra 2013 na:
<http://axwonders.blogspot.com/2013/04/ax-2012-cil-how-does-it-work.html>
- [33] Invoke Web Services from Android, pridobljeno decembra 2013 na:
<http://java.dzone.com/articles/invoke-webservices-android>
- [34] Services and AIF architecture [AX 2012], pridobljeno decembra 2013 na:
<http://technet.microsoft.com/EN-US/library/dd309705.aspx>