

**UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA**

DIPLOMSKO DELO

**STORITVENO USMERJENA ARHITEKTURA -
PROGRAMSKE PLATFORME**

Ljubljana, oktober 2010

JERNEJ ŠINKO

IZJAVA

Študent **Jernej Šinko** izjavljam, da sem avtor tega diplomskega dela, ki sem ga napisal pod mentorstvom **dr. Tomaža Turka**, in da dovolim njegovo objavo na fakultetnih spletnih straneh.

V Ljubljani, dne 4.10.2010

Podpis: _____

KAZALO

UVOD	1
1 STORITVENO USMERJENA ARHITEKTURA	2
1.1 KAJ JE SOA?.....	2
1.2 KATERE PREDNOSTI IMA SOA?.....	4
2 METODOLOGIJA IZVEDBE	5
2.1 REFERENČNA ARHITEKTURA SOA	7
2.2 RAVNI STORITVENO USMERJENE ARHITEKTURE	8
2.3 POSLOVNA ARHITEKTURA	12
2.4 POSLOVNI PROCESI.....	13
2.5 STORITVENO PODJETNIŠKO VODILO (ESB).....	14
2.6 UPRAVLJANJE.....	16
2.7 SODELUJOČI.....	17
2.8 REGISTER.....	18
2.9 REPOZITORIJ	19
2.10 POSREDNIK	20
2.11 VARNOST	21
3 STORITVE	22
3.1 VMESNIK.....	23
3.2 ZNAČILNOSTI STORITEV	23
3.3 VRSTE STORITEV	25
3.4 PROTOKOLI	27
3.4.1 XML jezik.....	27
3.4.2 SOAP.....	29
3.4.3 WDSL	30
4 PROGRAMSKE PLATFORME	30
4.1 IBM	30
4.2 Oracle	36
4.3 Microsoft	42
SKLEP.....	45
LITERATURA IN VIRI.....	47

KAZALO SLIK

<i>Slika 1: Elementi SOA</i>	3
<i>Slika 2: Ravni SOA</i>	9
<i>Slika 3: Povezanost SOA in BPM</i>	13
<i>Slika 4: Storitveno podjetniško vodilo</i>	15
<i>Slika 5: Upravljanje, register in repozitorij</i>	19
<i>Slika 6: Ločitev vmesnika in izvedbe storitve</i>	22
<i>Slika 7: Poslovne storitve</i>	26
<i>Slika 8: Nabor protokolov storitev</i>	28
<i>Slika 9: IBM referenčni model SOA</i>	31
<i>Slika 10: Skupni stroški lastništva strežnika aplikacij</i>	34
<i>Slika 11: WebSphere Portal</i>	35
<i>Slika 12: Oracle referenčni model SOA</i>	38

KAZALO TABEL

<i>Tabela 1: Primerjalna tabela programske opreme</i>	44
---	----

UVOD

V času globalnega trga morajo biti podjetja hitra in prilagodljiva na nastale spremembe. Preživijo le tista, ki so se sposobna najhitreje odzvati na spremembe. Fleksibilnost informatike je ključen dejavnik za mnoga podjetja. Informatika je postala pomemben dejavnik dodajanja poslovne vrednosti. Hkrati so postali procesi in informacijski sistemi vedno bolj kompleksni. Avtomatizacija ni več del enega samega sistema. Bližamo se trenutku, ko bodo vsi individualni sistemi postali en sam porazdeljen sistem. Ukrepi, ki so se uporabljali v preteklosti za reševanje problemov razširljivosti porazdeljenih sistemov v prihodnosti ne bodo zadostovali. Vse težje je zagotavljati harmonizacijo in nadzor nad sistemi. Centraliziran nadzor je dosegel svoje meje razširljivosti. Potreben je nov pristop. Pristop, ki sprejema heterogenost in vodi v decentralizacijo.

Naslednje desetletje bo polno zmagovalcev in tistih, ki bodo za njimi zaostajali. Uspešna podjetja se bodo naučila uporabljati informatiko kot strateški diferenciator, ki jim bo omogočal vzdržno konkurenčno prednost. Zgrajena bo na temeljih storitveno usmerjene arhitekture (angl. *service oriented architecture*) ali SOA, ki omogoča uporabo ključnih poslovnih zmogljivosti in informacij kot fleksibilne in ponovno uporabne storitve. SOA je kombinacija poslovnega modela, strategijo informatike, arhitekturnega pristopa in izvedbe, ki temelji na konceptu storitev. SOA omogoča podjetjem rast, spremembo, partnerstva in preobrazbo. Storitve bodo podpirale raven poslovnih procesov, ki se jih bo lahko zlahka spremenilo za zagotavljanje novih produktov in storitev. Le tako bodo podjetja lahko vzdrževala prednost pred konkurenco.

Lahko si je predstavljati, da bo v prihodnosti večina programske opreme dostavljena in uporabna v obliki storitev. Danes je dostopnost orodij omogočila hitro gradnjo storitev. Izziv je zgraditi dobro storitev. V naslednjem koraku je potrebno storitev vmestiti v krovno arhitekturo, ki omogoča združevanje storitev v večje poslovne procese znotraj podjetij. Arhitektura in proces načrtovanja morata določiti kontekst, ki vpliva na izvedbo storitev.

Namen diplomskega dela je predstaviti koncept storitveno usmerjene arhitekture. Prikazal bom povezanost poslovne strategije, poslovne arhitekture, storitveno usmerjene arhitekture in programskih platform. Sestavljena je iz štirih glavnih delov. V prvem delu bom izpostavil prednosti storitveno usmerjene arhitekture, ki omogočajo podjetjem konkurenčne prednosti. V drugem delu bom predstavil poglede na primere metodologij vpeljave SOA. V nadaljevanju bom povezal SOA, poslovno arhitekturo in poslovne procese. V tretjem delu bom razložil kaj so storitve in kako omogočajo konkurenčne prednosti. Sledil jim bo opis storitev. V četrtem - zadnjem delu bom predstavil najbolj razširjene programske platforme za izvajanje storitveno usmerjene arhitekture. V sklepu bom strnil vso pridobljeno znanje.

Pri preučevanju obravnavane tematike sem uporabil predvsem tujo strokovno literaturo,

članke iz različnih publikacij, spletne vire ter interne vire podjetij, kot so primeri uporab in raziskave. Največ literature sem našel predvsem na spletu. Pri pridobivanju gradiva sem si pomagal z javno dostopnimi in naročniškimi viri.

1 STORITVENO USMERJENA ARHITEKTURA

1.1 KAJ JE SOA?

Storitveno usmerjena arhitektura je ogrodje za integracijo poslovnih procesov in podporo informacijski tehnologiji v obliki šibko sklopljenih (ang. *loosely coupled*), varnih in standardiziranih komponentah – storitvah, ki se jih da ponovno uporabiti in združevati z namenom obravnavanja spreminjajočih se poslovnih prioritet. SOA združuje poslovanje in zahteve informacijske tehnologije z namenom pridobitve večje tržne prednosti preko gibkih poslovnih operacij (Biberstein et al., 2008).

V zadnjih letih je bila ločitev uporabniškega vmesnika od izvedbe aplikacij na programski ravni uvrščena na arhitekturno raven s šibko sklopljenostjo vmesnika storitve, ki jo uporablja uporabnik storitve in izvedbo storitve s strani ponudnika storitve. Ravno tako se je izvedba storitve ločila od programske opreme. Kar pomeni, da izvajanje aplikacije ni več vezana na specifično programsko opremo. Ta slog arhitekture je dobil ime storitveno usmerjena arhitektura. V novi arhitekturi so objavljene le storitve, zato so uporabniki izolirani od podrobnosti izvedbe storitve pri ponudniku (Open Group, 2009).

SOA zagotavlja metode za sistemski razvoj in integracijo, kjer se sistemi funkcionalnosti izrazijo kot interoperacijske storitve. SOA infrastruktura omogoča komunikacijo med različnimi aplikacijami. Storitvena usmerjenost cilja na šibko sklopljenost storitev z operacijskimi sistemi, programskimi jeziki in ostalimi tehnologijami, ki so osnova aplikacijam. Vendar pa ustvarjanje storitev ni glavni izziv SOA. SOA ločuje funkcije v značilne enote oz. storitve, ki jih razvijalci naredijo dostopne preko mreže, tako da jih uporabniki lahko združujejo in ponovno uporabijo v izdelovanju aplikacij. Cilj je, da uporabnikom zagotavlja združevanje delov funkcionalnosti v procesu izdelovanja ad hoc aplikacij, ki so skoraj v celoti zgrajene iz obstoječih programskih storitev. Večji kot je del funkcionalnosti, manj vmesniških točk je potrebnih za izvedbo posamezne funkcionalnosti. Granularnost, velikih kosov funkcionalnosti je premajhna za lahkotnost ponovne uporabe.

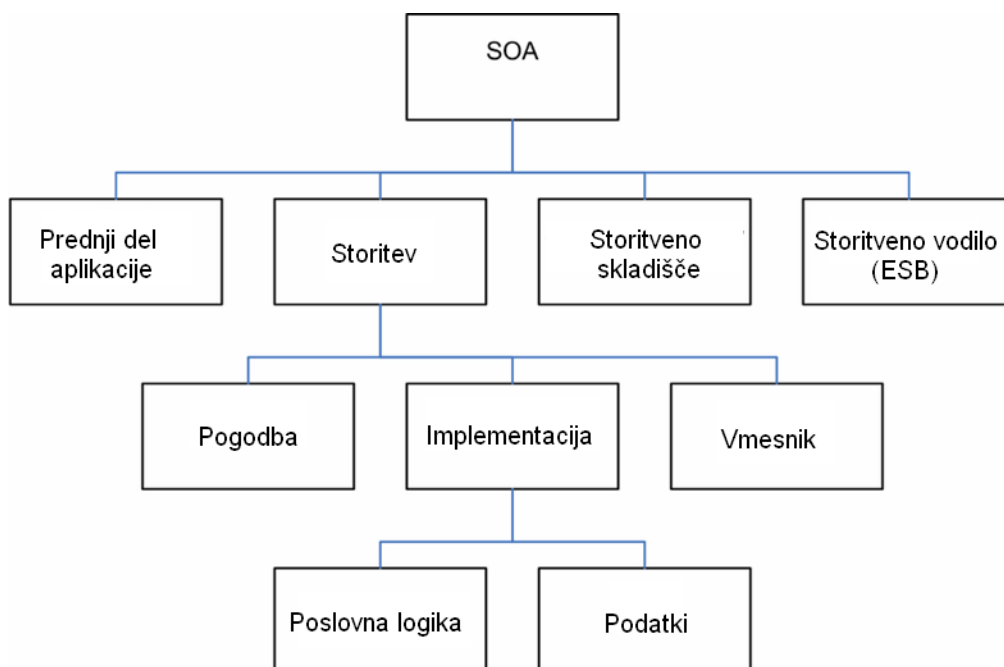
Privlačnost SOA je v tem, da bi lahko stroški izdelave aplikacij bili nizki zaradi obstoja programske opreme, ki zadovoljuje stvaritev zahtev ostalih aplikacij. V idealnem stanju bi bilo potrebno le orkestriranje za izdelavo nove aplikacije. To stanje bi bilo lažje doseči, če bi vse storitve ustvarjalo ena organizacija oz. podjetje. Vendar pa dejansko temu ni tako. Zato je del SOA arhitekture zadolžene za ustvarjanje okolja, ki je potrebno za izdelavo in uporabo komponentnih storitev v organizaciji. Arhitektura omogoča različnim organizacijam

neodvisno izvedbo storitev, ki služijo zadovoljevanju njihovih trenutnih potreb, a se lahko hkrati združujejo v poslovne procese višjih stopenj in organizacijske rešitve.

Glavni elementi SOA so:

- **Procesi** – predvsem poslovni procesi, ki pogosto združujejo veliko aplikacij oz. LOB (ang. *line-of-business*) Aplikacije LOB so nabori programskih aplikacij, ki so nujne za poslovanje npr. računovodstvo, ravnanje z logistiko, planiranje razpoložljivosti sredstev. Aplikacije LOB so večinoma veliki programi, ki vsebujejo veliko število integriranih zmogljivosti in so povezani z bazami podatkov in sistemi za ravnanje z bazami podatkov. Vedno pogostejše se aplikacije LOB povezujejo z mrežnimi aplikacijami preko uporabniških vmesnikov na internetu in osebnimi aplikacijami kot je elektronska pošta.
- **Storitve** — Modularne enote poslovne funkcionalnosti.
- **Integracija**— Povezanost in odkritost obstoječih aplikacij in podatkov.
- **Obstoječi sistemi** — Sistemi, ki jih poseduje podjetje v danem trenutku, komercialne aplikacije in podatki, ki jih podjetje hoče izkoristiti.
- **Dokumenti**—Enote poslovne informacije visoke vrednosti npr. dokument nabave.
- **Semantika**— Pomen informacij, ki se izmenjajo.
- **Transformacija**— Konverzija informacij ali semantike iz enega formata v drugega.
- **Komunikacija**— Sposobnost komuniciranja med storitvami.

Slika 1: Elementi SOA



Vir: Wikipedia, SOA.

1.2 KATERE PREDNOSTI IMA SOA?

SOA omogoča podjetjem združevanje vodenja poslovanja in informatike preko pogodbe. Določa nabor poslovno orientiranih storitev informatike, ki skupno podpirajo poslovne procese in cilje. Zagotavlja ne le nesklopljeno funkcionalnost, ki je ponovno uporabljiva, ampak tudi mehanizme za zunanji nadzor storitve. Kot fleksibilen in širok arhitekturni okvir ima SOA naslednje značilne zmogljivosti (Open Group, 2009):

- **Zmanjševanje stroškov** – Manjše stroške dosežemo z združevanjem funkcionalnosti in ločevanje izvajanja aplikacij na ozkem naboru programske opreme. Vzporedno povečuje uspešnost obstoječih orodij.
- **Gibkost** - Strukturira poslovne rešitve temelječe na naboru poslovnih storitev in storitev informatike na tak način, da prične s hitro rekonstrukcijo in rekonfiguracijo rešitev poslovnih procesov.
- **Povečevanje konkurenčnih prednosti** – Zagotavlja priložnost za vstop na nove trge in povečevanje obstoječih poslovnih zmogljivosti na nov in inovativen način z uporabo šibko sklopjenih storitev.
- **Hitrost dostopa na trg** – Dostavlja poslovne rešitve hitreje, kar omogoča poslovodstvu odločanje o ključnih rešitvah in prepušča informacijski tehnologiji podporo in izvedbo poslovnih rešitev.
- **Združevanje** – Integrira rešitve preko poslovno-informacijskih silosov in podjetij, zmanjšuje število sistemov in omogoča združevanje. Iz obstoječega spleta odvisnosti naredi bolj organiziran in integriran nabor sistemov.

Bieberstein (Bieberstein, et al., 2008, poglavje 2.1.1) navaja šest trendov, ki vplivajo na SOA:

- **Globalna integracija** – Podjetja se morajo hitro odzivati na globalne priložnosti. Kar pomeni, da morajo zaposleni uporabljati najboljši intelektualni kapital, ki je na voljo znotraj in zunaj podjetja in odkrivati nove ključne sposobnosti. Znanje vključuje izdelavo, odkrivanje in ponovno uporabljivost poslovnih storitev, ki omogočajo hiter dostop do trga.
- **Participativen internet** – Potrebno je zajeti interakcije strank na internetu in razumeti želje kupcev. Takšne informacije poganjajo razvrščevanje strani na Google iskalniku ali eBay razvrščanju prodajalcev. Tehnika omogoča obširen vir informacij in hitro graditev vrednot in zvestobe. Podjetje bo moralo imeti enoten pogled na kupce in sposobnost analiziranja interakcij z njimi.
- **Demografika delovne sile** – V razvitem svetu se podjetja srečujejo s prvim velikim valom upokojevanja s čimer bodo zgubile veliko pridobljenega tematskega znanja. Novi uslužbenci morajo dobiti pregleden in dostopen vir podatkov do svojih kolegov. Podjetje mora zagotoviti, da so podatki na voljo vsem preko storitvenih vmesnikov.
- **Programska oprema kot storitev** – Programska oprema kot storitev preoblikuje industrijo paketne programske opreme. Gostujoče aplikacije in podatki so cenejši, lažji za vzdrževanje in izboljšavo, preprosti za integracijo in podporo in dostopni po svetu. Programska oprema, ki jo prodajajo prodajalci je večinoma dražja od tiste, ki se jo ponuja

pod filozofijo programska oprema kot storitev.

- **Virtualizacija podatkov in naprav** – Aplikacije in podatki se bodo prenesli na mrežno dostavo in hrambo, kar bo omogočalo katerikoli računalniški platformi z mrežnim brskalnikom dostop do podatkov, ki jih potrebujemo. Odločilno je zagotavljanje relativnih informacij podjetniškim interesnim skupinam.
- **Preprostost** – Svet postaja vedno bolj zapleten a le tisti, ki zagotavljajo preproste metode bodo preživeli. Za zagotavljanje preprostosti potrebujemo načrt in disciplino. Stroški se nam bodo povrnil, ker vedno več podjetij in potrošnikov išče preprostost

Arhitekt SOA naj ima vedno v mislih potrebe svojega podjetja in ceno različnih potencialnih rešitev. S tem znanjem naj se odloči katere od rešitev naj se uporabijo in kako naj se uporabijo pri izdelavi SOA v tem podjetju. Arhitekt naj vedno pregleduje dane informacije o zahtevah in rešitvah, da pridobi stopnjo razumevanja, ki gre globlje od tržnih izrazov. Npr. pogosto je rečeno, da SOA omogoča gibčnost podjetja. Vprašati se je treba kaj gibčnost pomeni za vaše podjetje. Ali je to sposobnost hitre rekombinacije obstoječih funkcionalnosti za doseganje spreminjajočih zahtev strank? Ali je to sposobnost hitrega razvoja nove funkcionalnosti? Ali je to sposobnost povečanja obsega poslovanja? Vse te rešitve se lahko vgradi v SOA, a vsaka zahteva drugačno in posebno podporno infrastrukturo. Potrebno je izbrati rešitev, ki ustreza zahtevam.

2 METODOLOGIJA IZVEDBE

Uspešna izvedba storitveno usmerjenih rešitev je zapleten podvig, ki mora upoštevati številne vidike vplivanja na SOA. Uresničevanje izvedbe zahteva sodelovanje med velikim številom skupin v podjetju, upravo, vodji poslovanja, arhitekta, razvojne skupine, operacij itd. Na ravni podjetja vse to ne bi bilo mogoče brez dobro začrtane metodologije, ki bi opisovala večje korake in delovne produkte ter vloge in zadolženosti vsake sodelujoče skupine. Metodologija naj bi vsebovala (Rosen, 2008, str. 80):

- **Referenčna arhitektura SOA** – Definira pomembne dele SOA, še posebej kaj so storitve in kakšno je njihovo medsebojno razmerje, koncepte načrtovanja, izvedbe, procesov in razmerij do ostalih arhitektur in komunikacije.
- **Definicija poslovne arhitekture** – V prvem koraku določimo poslovno arhitekturo podjetja. Naše definicije bodo vplivale na procese, storitve, informacije in zgrajene rešitve.
- **Identifikacija storitev** – Določimo nabor storitev znotraj konteksta podjetja, ki podpira poslovno arhitekturo. Nabor storitev sestavlja inventar storitev.
- **Semantična definicija informacijskega modela** – Ustvarimo podjetniški informacijski model, ki zajema semantiko procesov in storitev. Aktivnost poteka vzporedno z identifikacijo storitev. Semantični model je pod vplivom poslovne arhitekture in informacijske arhitekture.
- **Določitev storitev** – Ustvarimo storitvene pogodbe, ki se lahko v času načrtovanja

uporabijo za izbiro relevantnih storitev, ki sodijo v rešitve. Specifikacije storitev vsebujejo storitveni vmesnik kot tudi ostale uporabniške informacije.

- **Realizacija storitev** – Razvoj in izvedba storitev.
- **Izvedba storitveno orientiranih rešitev** – Zgradimo podjetniške rešitve iz storitev. Paziti je potrebno, da na storitvene rešitve vpliva aplikacijska arhitektura. Opozoriti je potrebno, da postopek izvedbe ni zaporeden proces. Proces je postopen in iterativen. Začne se s stvaritvijo poslovne arhitekture podjetja in nabora storitev. Sledi izvedba prvih storitev, ki podpirajo specifičen poslovni cilj. Z učenjem lahko nadgrajujemo arhitekturo SOA, poslovno arhitekturo, storitveno arhitekturo, nabor storitev, standarde in upravljanje.

Vendar lahko situacija v podjetju zahteva drugačne pristope za izvedbo SOA. Najti je potrebno primerno rešitev za specifične razmere. Pristopi se lahko tudi dopolnjujejo med seboj. Sledi nekaj primerov, ki so lahko v pomoč (Josuttis, 2007, str. 271):

“Štirje P-ji” - Pulier in Taylor

Priporočata najboljšo prakso štirih P-jev (angl. *people, pilot, plan, proceed*) ljudje, pilotni projekt, načrt in izvršitev:

- Zaposleni naj se izučijo o principih SOA in z njimi povezanimi tehnologijami.
- Pilotni projekt koristi pri dokazovanju pravilnosti koncepta.
- Fleksibilen več fazni načrt projekta služi integraciji in potrditvi SOA strategij v vsakdanjem poslovnem okolju.
- Glede na prejšnje korake izvršimo SOA.

Arhitektni načrt - Krafzig, Banke in Slama

Avtorji predstavijo arhitekturni načrt kot tri faze širjenja zmogljivosti:

- Osnovna SOA samo z osnovnimi storitvami.
- Mrežna SOA s kompozitnimi in orkestriranimi storitvami.
- Procesno usmerjena SOA s procesnimi storitvami.

Od zgoraj navdol in od spodaj navzgor - Krafzig in Slama

Pristop od spodaj se začne z majhno skupino navdušencev SOA, se razvija v napredno bolj uspešne projekte in se konča z izvedbo preko celotnega podjetja. Pristop od zgoraj se idealno začne na pobudo zgornjega ravnateljstva. Cilji se določijo, razišče se poslovni primer in pridobi financiranje. Sledi izvedba SOA strategije, ki vključuje vzpostavitev arhitekture, organizacijskih struktur in infrastrukture. Nato se začne s SOA projekti. Vendar je ta pristop skoraj nemogoče izvršiti zaradi decentralizacije in različnih lastnikov velikih porazdeljenih sistemov. Potrebno je imeti skupna sredstva in podporo vrhnjega ravnateljstva, recimo za organizacijsko reorganizacijo.

Pristop sredine

Pristop od zgoraj navzdol generira veliko dokumentacije vendar je pogosto prekinjen, zaradi neustreznosti, ker vodstvo pogosto nima vseh informacij o potrebnih storitvah. Pristop od spodaj navzgor zgradi napačne storitve, ki ne ustrezajo poslovnim strategijam in procesom. Pristop sredine (angl. *middle out*) spodbuja nastanek arhitekture, ki jo določi majhna, lahko tudi virtualna skupina ljudi. Skupina naj določi abstraktno kombinacijo minimalno potrebnih standardov identifikatorjev, formatov in protokolov (angl. *Identifier standard, Format standard and Protocol standard, IFaP*), ki so dovolj ohlapni, da omogočajo naknadne spremembe pri načrtovanju (Malik, 2007). Poimenovala bo storitve, ki jih podjetje potrebuje in opisala kako naj delujejo. Rezultat je kratek dokument na katerega se lahko navezujejo tako razvijalci kot vodstvo. Naloga skupine je s tem končana. Razvoj bo prepustila inženirjem in arhitektom. Sledi pristop razvoja od spodaj navzgor. Določitev minimalnega nabora standardov s katerimi se strinjajo razvijalci in vodstvo, omogoča sodelovanje med skupinama. Hkrati so skupine odgovorne za svoje področje, kar odstrani potrebo za stalno potrjevanje poteka dela s strani vodstva. Pristop je primeren v podjetjih, kjer vodstvo ni proaktivno pri določanju informacijske arhitekture

2.1 REFERENČNA ARHITEKTURA SOA

Uporaba referenčne arhitekture SOA je ključni del za doseg prednosti, ki jih omogoča SOA. Referenčna arhitektura predstavlja navodila za odločitve o vrsti arhitekture, izgledu in izvedbi. Z njo si pomagamo za izdelavo in ocenjevanje arhitekture. Ponuja vzorce in vpogleda za integracijo sestavnih delov SOA. Referenčna arhitektura je v pomoč pri odgovarjanju na vprašanja s katerimi se soočajo arhitekti (Opengroup, 2009):

- Katere lastnosti SOA so izražene v obliki ravni in jih je potrebno vključiti v razvoj rešitev, ki temeljijo na principih storitvene usmerjenosti?
- Katere gradnike je potrebno vključiti v vsako raven rešitve SOA?
- Katere so ključne arhitekturne odločitve, ki jih je potrebno sprejeti pri načrtovanju?
- Katere vloge v projektu bodo imele koristi od teh principov in vodil?

Referenčna arhitektura SOA se uporabi kot načrt in vsebuje predloge in vodila za arhitekta. Ti koraki vodijo k olajšanju uporabe avtomatizacije in racionalizaciji procesa modeliranja in dokumentiranja arhitekturnih ravni, arhitekturnih gradnikov na ravneh, opcije za ravni in gradnike, iskanje produktov za gradnike, arhitekturne in načrtovalne odločitve, ki prispevajo k izvedbi SOA. Namenjena je podpori podjetjem, ki uvajajo SOA, prodajalcem SOA infrastrukturnih komponent, integratorjem in telesom, ki se ukvarjajo z razvojem standardov in specifikacij SOA. Ostaja še veliko težav in izzivov za gradnjo rešitev SOA. Pravilen načrt, izbira elementnih gradnikov in njihovo kombiniranje, modeliranje storitev, upravljanje, interoperativnot in sposobnost razločanja različnih komponent so ključna za uspešno

načrtovanje, uporabo in evolucijo SOA. S tehničnega stališča se mora arhitek vprašati naslednje:

- Kateri so kriteriji za izdelavo rešitev SOA?
- Kako je lahko rešitev SOA organizirana kot arhitekturni okvir s povezanimi arhitekturnimi in transformacijskimi zmogljivostmi?
- Kako lahko razvijemo rešitev, ki bo maksimizirala ponovno uporabljivost sredstva?
- Kako lahko orodja avtomatizacije omogočijo boljše informiranost potrditve pravilnosti arhitekture in načrtovanja?

Za naslavljanje naslednjih vprašanj mora biti referenčna arhitektura definirana z naslednjimi principi:

- Referenčna arhitektura naj bo generična rešitev, ki ni vezana na posameznega ponudnika.
- Temelji naj na modelu skladnosti s standardi.
- Uporabi se naj jo za izdelavo posrednih poslovnih arhitektur in specifičnih arhitekturnih rešitev.

Naslovi naj večino zahtev interesnih skupin:

- Za podjetja mora biti referenčna arhitektura dovolj generična za neodvisnost od rešitev ponudnikov. Opisuje na gradnike, arhitekturne odločitve in ostale lastnosti za izdelavo okvirja razumevanja, ki je zadosten za ocenjevanje skladnosti.
- Za ponudnike storitev mora referenčna arhitektura omogočiti nabor standardov in dovolj specifičnosti za vodenje ocenjevanja skladnosti z danimi standardi.
- Za integratorje naj zagotavlja model za definiranje specifičnih omejitev in smeri za izvedbo SOA.
- Za organizacije standardov naj zagotavlja referenco za razširitev standardov in ponudbo vodil ter bolj detajlirane ravni specifičnosti.
- Način za izdelavo primera naj bo določen s strani uporabnika referenčne arhitekture. Vključuje naj najmanjše število ravni za prikaz možnih kombinacij in elementov znotraj SOA.

2.2 RAVNI STORITVENO USMERJENE ARHITEKTURE

Raven operativnih sistemov

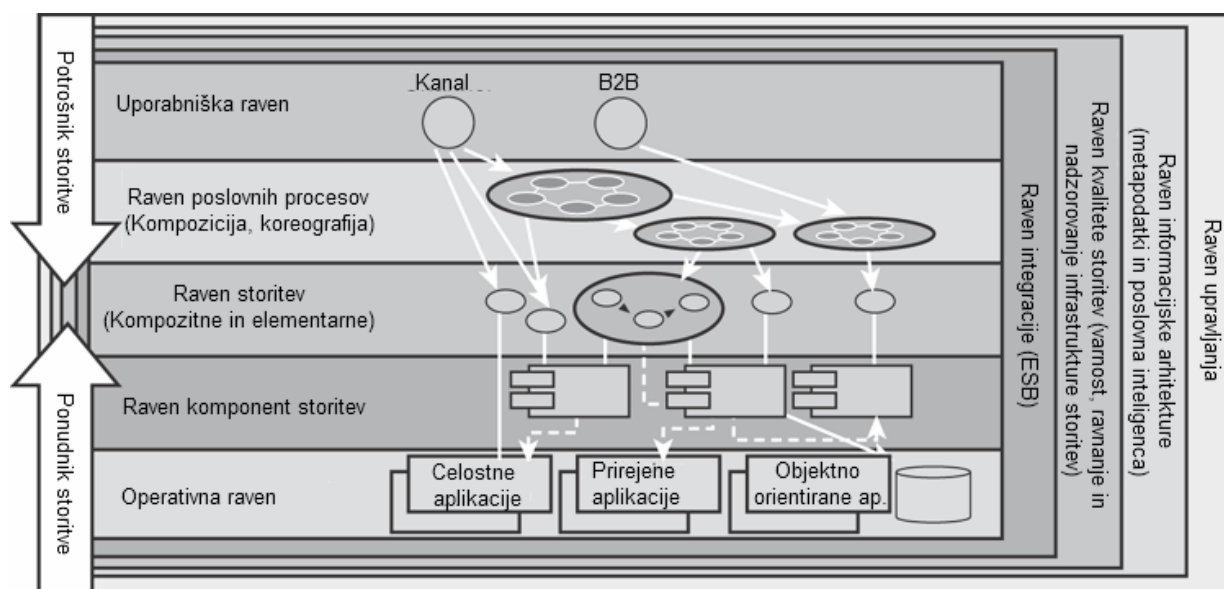
Na tej ravni se določi nova in obstoječa infrastruktura podjetja. Tukaj se nahajajo operativni sistemi, ki v danem trenutku podjetju pomagajo podpirati poslovne aktivnosti. Operativni sistemi vključujejo vse prirejene aplikacije, dane sisteme, transakcijsko-procesne sisteme in podatkovne baze, obstoječe prirejene cele aplikacije kot so J2EE (java) in .Net aplikacije, obstoječe pakirane aplikacije in storitve kot so ERP (angl. *enterprise resource planning*) in CRM (angl. *customer relationship management*) paketi ter akterje, ki podpirajo izvajanje storitev v SOA, kar neposredno vpliva na skupne stroške izvedbe storitev SOA znotraj podjetja.

Na tej ravni storitve izvršujejo svoje funkcionalnosti. Za primer lahko vzamemo storitev za posodabljanje pogojev pri odobritvi kredita strankam. Storitve se izvede na različnih komponentah, ki tečejo na aplikacijah, te pa gostujejo na ravni operativnih sistemov. Mnogo obstoječih programskih sistemov je del ravni. Zmožljivosti, ki jih podpira raven, vsebuje integracijo infrastrukturnih storitev in SOA, ponovno uporabljivost obstoječih sestavljenih sredstev na ravni in infrastrukturnih elementov potrebnih za izvedbo SOA.

Raven komponent storitev

Komponente na tej ravni so skladne s pogodbami storitev v ravni storitev. Komponenta omogoča izvedbo ene ali več storitev. Komponenta lahko izvede poslovno funkcionalnost iz številnih pogosto različnih operativnih sistemov, med tem ko prikrije integracijo in zapletenost dostopa storitve, ki je v uporabi.

Slika 2: Ravni SOA



Vir: Arsanjani, A. et al., 2007.

Uporabnik se ne zaveda zahtevnosti implementacije komponente storitve. Prednost le tega je, da lahko zlahka zamenjamo operativni sistem ne da bi spremenili definicijo storitve. Komponenta storitve zagotavlja točko uveljavljanja realizacije storitve in zagotavlja kvaliteto storitve (angl. *quality of service* ali QoS) in skladnost s SLA (angl. *service level agreement*). Omogoča poslovno fleksibilnost s podporo funkcionalne izvedbe fleksibilnih storitev informatike, njihovo sestavljenost in ravni. Zagotavlja fleksibilnost informatike s krepljenjem šibke sklopljenosti v sistemu. Razklapljanje je doseženo s prikrivanjem podrobnosti izvedbe pred strankami.

Raven storitev – Raven vključuje vse storitve, ki so definirane v portfelju storitev podjetja. Semantične in sintaksične informacije o storitvi so zapisane na tej ravni. Informacije o sintaksi so razvidne iz operacij na posamezni storitvi, vhodna in izhodna sporočila in

definicije o prelomnicah storitev, informacije o semantiki so izpeljane iz smernic storitev, odločitvah o ravnanju s storitvami, potrebah o dostopu do storitev itd. Storitve so definirane tako, da so dostopne in na dosegu preko kanalov in uporabnikov, neodvisno od izvedb in raznih transportnih protokolov. Storitve so pripravljene na odkritje s strani iskalnikov in na izvajanje ali pa na sestavljanje kompozitnih storitev.

Poslovni procesi – Poslovni procesi prikazujejo delovanje poslovanja podjetja. Poslovni proces je informacijski prikaz različnih aktivnosti koordiniranih in sodelujočih v podjetju za doseganje določene funkcije višje stopne v podjetju. Ta raven predstavlja procese kot orkestracijo ali kompozicijo storitev šibke sklopljenosti, ki podpirajo storitve predstavljene na ravni storitev. Raven je zadolžena tudi za ravnanje s celotnim življenjskim ciklom procesov njihove orkestracije in koreografije. Tokovi podatkov in informacij med dvema korakoma znotraj vsakega procesa so prikazani na tej ravni. Predstavljeni procesi so povezava med poslovnimi zahtevami in manifestacijo njihovih rešitev informatike. Uporabniki, kanali in partnerski sistemi B2B (angl. *business to business*) na ravni uporabnikov uporabljajo poslovne procese kot eden izmed načinov za sprožanje funkcionalnosti aplikacij. Raven vsebuje informacije o izmenjavi podatkov med uporabniki, tako posamezniki kot podjetji, sredstvi, procesi v številnih obrazcih za doseganje poslovnih ciljev. Izmenjane informacije vsebujejo tudi nestrukturirana in netransakcijska sporočila. Za oblikovanje tokov storitev se uporablja poslovna logika, kot so poslovna pravila, smernic in ostale poslovne zahteve. Poslovni kontekst in metapodatki se uporabljajo za podporo zbiranja storitev znotraj podjetij. Uporabimo jih lahko za oblikovanje poslovnih procesov v podjetju ali preko podjetij. Gradnji procesnih blokov na zahtevo sledi zmanjšanje stroškov. Danes imajo poslovni procesi osrednjo vlogo povezovanja poslovanja in informatike.

Uporabniška raven – Na tej ravni so prikazani različni kanali skozi katere so dostavljene funkcije informatike. Kanali so lahko v obliki raznih uporabniških tipov (notranji in zunanji uporabniki, ki dostopajo do funkcionalnosti aplikacij skozi dostopne mehanizme kot so portali, sistemi B2B itd.) Cilj te ravni je standardiziranje dostopnih protokolov in podatkovnih formatov za omogočanje hitrih stvaritev dostopnih mask poslovnih procesov in storitev. Spoštovanje pravil standardnih mehanizmov za razvoj komponent predstavitvenih ravni za poslovne procese in storitve pomaga pri ponudbi osnovnih podlag v obliki standardnih arhitekturnih vzorcev. S tem pa lahko razvijalna skupnost posvoji skupne dostopne maske za uporabo storitev. S prevzemom uveljavljenih dostopnih mehanizmov kot so portali in odprti standardi lahko zmanjšamo čas za razvoj in izvedbo z uporabo številnih v naprej zgrajenih, preverjenih in ponovno uporabljivih gradnikov dostopnih mask. Z njihovo uporabo zmanjšamo kompleksnost in stroške vzdrževanja. Tak način spodbuja združen enoten pogled na predstavitev znanja, enotno dostopno točko podprtim poslovnim procesom in aplikacijam.

Raven integracije – Raven omogoča zmogljivosti uporabnikom storitev za lokacijo ponudnikov storitev in izvedbe storitev. Preko osnovnih zmogljivosti mediacije, usmerjanja in podatkovnih ter protokolnih transformacij se na tej ravni izvršujejo komunikacije med

storitvami, medtem ko so del poslovnega procesa. Ključne nefunkcijske zahteve kot so varnost in kvaliteta storitve med sosednjimi ravni v arhitekturi so izvedene preko gradnikov arhitekture na tej ravni. Funkcije te ravni se vse pogosteje definirane kot podjetniško storitveno vodilo oz. ESB. (angl. *enterprise service bus*) Izvedbo osnovnih zmogljivosti in omogočajo plast posrednosti med uporabniki storitev in ponudnikom storitev, s tem ko odstrejo storitve le preko ESB-ja. Produkti ESB v večini primerov vsebujejo specializirane vsebine, ki ponujajo diferencirane zmogljivosti na trgu. Integracijska raven ponuja osnovna poslovna pravila, ki jih uporablja ESB in ostale ravni na referenčni arhitekturi SOA. Poslovna pravila je potrebno stalno izvrševati na številnih ravneh v SOA ali pa se čez čas pokažejo razlikovanja med pravili in izguba konsistenčnosti. Raven vsebuje podporo virtualizaciji pogonskim storitvam preko registra. Virtualizacija storitve odklopi lokacijo storitve uporabnikom storitve. Storitve so izpostavljene uporabnikom v registru, vendar jim njihova lokacija ni znana, zaradi potreb upravljanja, posodabljanja, spremembe lokacije storitve brez vpliva na uporabnika.

Raven kvalitete storitev QoS (angl. *quality of service*) – Tukaj se odvija izvedba in ravnanje z nefunkcijskimi zahtevami, ki ji mora izvesti storitev. Čeprav SOA prinaša veliko prednosti preko novega arhitekturnega sloga, prinaša programski model, ki podpira izgradnjo prvovrstnega sistema SOA, veliko izzivov. Izzivi nastanejo, ko se spopademo z osnovami SOA: abstrakcijo, odprtimi standardi in protokoli, porazdeljenemu procesiranju, heterogenosti procesirne infrastrukture itd. Spoštovanje skladnosti teh zahtev povzročijo težjo izvedbo nefunkcijskih zahtev. Zato raven zagotavlja infrastrukturo za realizacijo nefunkcijskih zahtev. Zajame podatkovne elemente, ki ponujajo informacije o nespoštovanju nefunkcijskih zahtev na vsaki od SOA ravni. Signalizira neskladnost z nefunkcijskimi zahtevami, ki se navezujejo na značilnosti storitev in smernic za vsako raven SOA. Podatki o storitvah se zajamejo in povežejo z njihovimi storitvami, da lahko uporabniki ocenjujejo uspešnost. Raven deluje kot opazovalec ostalih ravni in ustvarja dogodke, ko se pojavijo neskladnosti ali še raje, ko predvidimo pogoje neskladnosti. Standardne nefunkcijske zahteve, ki jih nadzoruje so varnost, razpoložljivost, dostopnost, razširljivost in zanesljivost. Raven povečuje poslovno vrednost SOA z nadzorovanjem poslovnih procesov preko ključnih indikatorjev uspeha. Še posebej se na ravni kvalitete storitev realizira varnost SOA.

Informacijska arhitektura – Na ravni se zagotavlja primerno reprezentacijo podatkov in informacij, ki so potrebni v SOA. Predstavitev podatkovne in informacijske arhitekture na vsaki od horizontalnih ravni je naloga te ravni. Poslovni protokoli, ki se uporabljajo za izmenjavo poslovnih podatkov so nameščeni na tej ravni. Shranjuje tudi metapodatke potrebne za podatkovno rudarjenje in poslovno inteligenco.

Upravljanje – Na tej ravni se zagotavlja primerno ravnanje celotnega življenjskega cikla storitev. Zadolžena je za prioritizacijo katere visoko-vredne storitve so na vrsti za izvedbo na vsaki od ravni v arhitekturi in za omogočanje odločanja kako storitve zadovoljujejo poslovni ali cilj informatike podjetja. Uveljavljanje smernic načrtovanja in delovanja, katere morajo

storitve izvesti in se jim pokoriti je ena izmed ključnih zadolženosti ter ravni. Raven se lahko uporablja z vsemi ostalimi ravni v referenčni arhitekturi SOA. Povezana je z ravno kvalitete storitve glede kvalitete in uspešnosti, z ravno storitev pa zaradi življenjskega cikla storitev itd. Poskrbi za upravljanje naslednjih sestavnih delov SOA:

- Dogovorov o ravni storitve ali SLA (angl. *service level agreement*), ki temelji na ključnih indikatorjev uspeha in ravni kvalitete storitve.
- Zmogljivosti in uspešnost ravnateljskih smernic.
- Omogočanje varnosti.
- Zbira informacije o vodilih za upravljanje SOA, življenjski cikel in ravnanje s portfeljem storitev, najboljše prakse, varnostne smernice, standardi, dokumentacijo o skladnosti, oprostivni in komunikaciji.

Drugače povedano, raven upravljanja ponuja okvir učinkovitega nadzora nad načrtovanjem in izvedbo storitev, tako da so skladne z različnimi poslovnimi in regulatornimi smernicami informatike in zahtevami.

2.3 POSLOVNA ARHITEKTURA

Poslovna arhitektura predstavlja funkcije in procese, ki podpira poslovanje, organizacijo in kraj kjer se izvaja poslovanje. Povedano drugače, poslovna arhitektura naslavlja izvajanje operativno kritičnih funkcij podjetja. Predstavlja, kako podjetje dejansko dosega svojo nalogo in ne kako je podjetje organizirano za doseganje naloge. Obsega tudi strateško smer, ki jo podjetje skuša doseči. Pomembni vplivi na poslovno arhitekturo so zakoni in regulacije, zunanje in notranje smernice, organizacijska struktura, organizacijska kultura, poslovne spremembe, ljudje, proračuni in tehnološki dejavniki. Ta raven ne upošteva fizičnih omejitev in ne vsebuje elementov systemskega načrta. (Rosen, 2008)

Open Group opisuje poslovno arhitekturo kot upoštevanje skrbi uporabnikov in obravnava (Open Group, 2009):

- **Ljudje** – človeška sredstva kot del sistema. Preučuje človeške akterje vključene v sistem.
- **Procese** – ukvarja se z uporabniškimi procesi vključene v sistem.
- **Funkcije** – ukvarja se s funkcijami potrebnimi za podporo sistemu.
- **Poslovne informacije** – obravnava informacije potrebne za podporo sistema.
- **Uporabnost** – obravnava element uporabnosti sistema in njegovega okolja.
- **Uspešnost** – obravnava element uspešnosti sistema in njegovega okolja..

Kaj je naloga poslovne arhitekture? Poslovna arhitektura prevede poslovno strategijo v akcijske procese. Omogoča organizaciji, da si predstavlja in izraža bistvo svoje organizacije, vzporedno z uskladitvijo poslovne arhitekture z informacijsko arhitekturo. Podjetju omogoča

vizualizacijo, analizo, redefiniranje in reinžiniranje načina notranje komunikacije ter komunikacije s poslovnimi partnerji in informatiko. Usklajuje poslovno strategijo informatike z (Rosen, 2008, str. 122):

- Identifikacijo ciljev in strategije
- Identifikacijo organizacijske strukture in upravljanjem ter njun vpliv na strategijo
- Uporablja strategijo preko celotnega podjetja
- Usklajevanjem z zunanjimi entitetami
- Identificira kvantitativne rezultate za merjenje uspešnosti strategije
- Identificira procese, pravila in informacije potrebne za podporo rezultatov
- Ravnanje in sinhronizacijo modela procesov, modela poslovnih pravil in informacijskega modela preko uporabe vsebine poslovne arhitekture na posamezne projekte in delovanja kot premostitve med podjetjem in zahtevami posameznih projektov.

Na ravni poslovne arhitekture moramo zagotoviti njeno interno konsistenčnost. Strategijo, cilje in organizacijo mora izraziti kot procese in informacije. Nato sledi ugotavljanje konsistenčnosti poslovne arhitekture in modela SOA. Potrebno je zagotoviti, da vodi dolgoročno vizijo in da imamo stalno sledenje od strategije do procesov, nalog in izvedbe v obliki poslovne storitve. Poslovni model SOA ima nalogo ravnanja s storitvami in informacij preko procesov.

2.4 POSLOVNI PROCESI

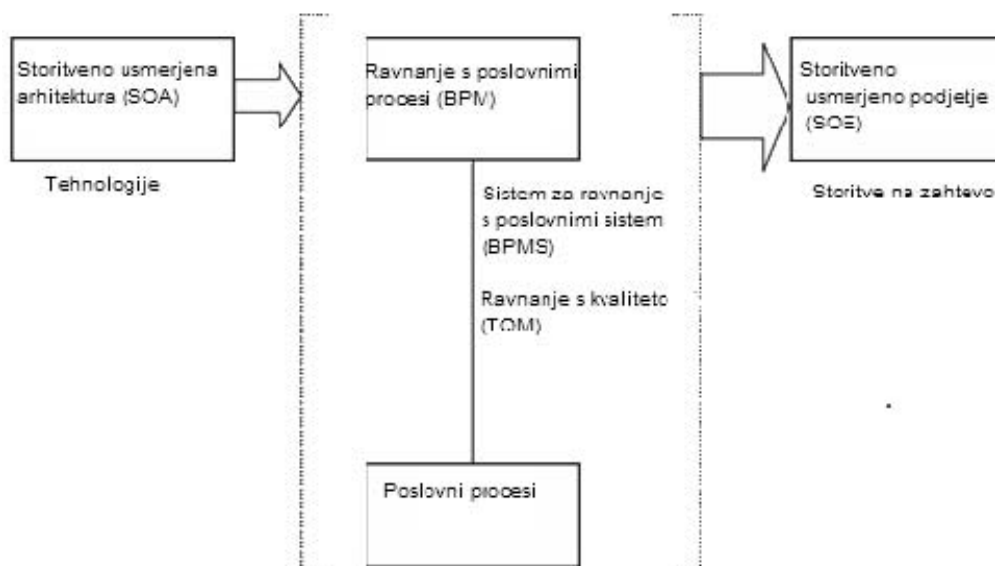
SOA se združuje z ravnanjem poslovnih procesov BPM (angl. *business process management*) v podjetjih, kjer se zavedajo pomena poslovnih procesov. BPM je pristop za doseganje poslovnih ciljev, koordiniranje procesov v podjetjih, vzpostavljanje najboljših praks in izdelovanje programskih rešitev primer. BPMS (ang. *business process management system*) opisovanje, analiziranje in izboljševanje učinkovitosti procesov nasproti poslovnim ciljem. BPMS pomaga pri grafičnem modeliranju procesov z jezikom WSBPEL (ang. *Web Services Business Process Execution Language*), tako da lahko informacijski oddelek izvede ali izboljša funkcije procesov (Bieberstein, 2004, str. 204).

Poslovni procesi se morajo spremeniti relativno pogosto, a hkrati temeljijo na stabilnih zmogljivostih. Fleksibilnost za spremembe izhaja iz zmožnosti hitre izdelave novih poslovnih procesov iz poslovnih storitev, ki so relativno stabilne. Poslovni procesi morajo:

- Biti določeni z modeli poslovnih procesov in izvršeni v BPMS.
- Biti sestavljeni iz aktivnosti, ki so uvedene s poslovnimi storitvami s pomočjo SOA.
- Prenašati informacije v, iz in znotraj procesov v obliki dokumentov.

Poslovni procesi so namenjeni izboljšavam in fleksibilnosti konkurenčnih procesov, omogočeno s strani informacijske tehnologije. (Chang, 2006, str. 30, 33.)

Slika 3: Povezanost SOA in BPM



Vir: P. Lawler J., et al., *Service-oriented architecture: SOA strategy, methodology and technology*, 2008, str. 6.

Združevanje SOA in BPM dodaja koristi z združevanjem storitev kot procesov. Tako pospešuje spremembe »na zahtevo« procesov kot storitev. SOA z BPM lahko konkurenčno diferencira zmogljivosti podjetja s stalno strategijo izboljšav ali strategijo diferenciacije z večanjem produktivnosti in reševanjem poslovnih problemov. (Chang, J.F. 2006 str. 21). Koristi izboljšav poslovnih procesov, izboljšave izkušenj strank, omogočanje boljših partnerskih povezav med podjetji, določanje časa potrebnega za pošiljanje izdelka na trg so kriteriji, ki jih podjetja primerjajo pri odločanju za vpeljevanje SOA. Iz logično povezanih nalog, načrtovanih za doseganje določenih poslovnih izidov, kot so finance, človeški viri, proizvodnja, trženje, prodaja, storitve in dobava.

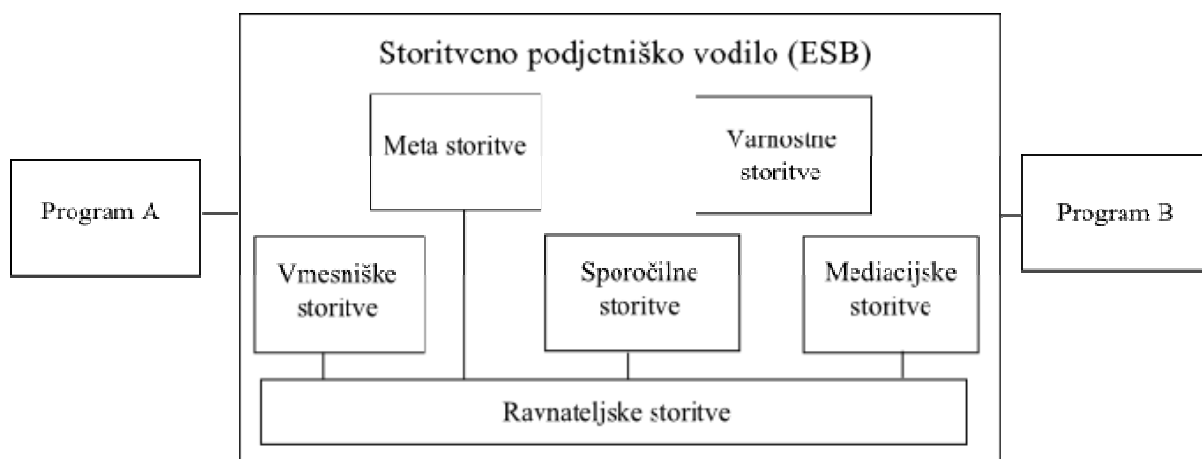
2.5 STORITVENO PODJETNIŠKO VODILO (ESB)

Storitveno podjetniško vodilo (angl. *Enterprise Service Bus, ESB*) je glavno komunikacijsko središče za storitve v SOA. ESB je načrtovan, da deluje kot posrednik med komponentami SOA, infrastrukturnimi storitvami in poslovnimi procesi. ESB-ji niso bili načrtovani kot posredniki za SOA. Ker pa SOA potrebuje dodeljenega posrednika za povečevanje števila uporabnikov, so postali nepogrešljiv del. ESB-ji so načrtovani vsestransko. Povežejo se lahko z različnimi tipi srednjih ravni (angl. *middleware*) (Pinus, 2004, str.1) skladišči metapodatkov, registri in vmesniki aplikacij. ESB-je lahko razumemo kot črne skrinjice. Ni potrebno, da razumemo kako delujejo, samo kaj počnejo. Poslovne storitve so povezane z vodilom. Za vodilo ni potrebno, da ve kaj počne storitev, dokler je možno storitev pravilno konfigurirati. ESB izvaja veliko infrastrukturnih nalog, ki bi tradicionalno morale biti zapisane v programski kodi aplikacij.

ESB sestavljajo naslednje komponente:

- **sporočilne storitve** – Podpirajo različne vrste sporočil, omogočajo vsebinsko usmerjanje in zagotavljajo dostavo sporočil. Lahko tudi delijo in združujejo sporočila.
- **storitve ravnanja** – Lahko nadzorujejo lastno delovanje in s tem pomagajo pri uveljavitvi SLA. Lahko tudi izvedejo prioritete sporočil in določajo poslovna pravila vsem aplikacijam in komponentam s katerimi pridejo v stik.
- **storitve vmesnikov** – Lahko potrjujejo sporočila glede na definicije shem. Podpirajo standarde spletnih storitev in ponujajo adapterje aplikacijam za nekatere ne-spletne storitve in vmesnike.
- **mediacijske storitve** – Transformirajo sporočila med formati pošiljatelja in prejemnika.
- **metapodatkovne storitve** – Transformirajo podatke z definicijami metapodatkov.
- **varnostne storitve** – Šifrirajo sporočila kjer je potrebno in vključujejo standardni varnostni model za avtorizacijo, avtentikacijo in revizijo vseh ESB aktivnosti.

Slika 4: Storitveno podjetniško vodilo



Vir: J. Hurwitz, Service Oriented Architecture For Dummies, 2007, str. 110.

Podjetja lahko začnejo z enim ESB-jem, ki omogoča eno nalogo. Kasneje lahko dodajamo ESB-je glede na potrebe rasti SOA. Vsak ESB ima svoj nabor pravil in smernice, vendar so združeni pod upravljanjem SOA. Upravljanje SOA je nabor pravil, ki obsega vse oddelke v podjetju. ESB deluje kot okvir, ki omogoča šibko sklopljenost programskih komponent in storitev. Deluje kot platforma za delovanje in povezovanje storitev. ESB ponuja tudi ponovno uporabljivost ključnih sredstev, tako da omogoča njihovo povezljivost na način lažji od prejšnjega. (Acharya&Keen, et al., 2004, str. 82). Šibka sklopljenost prida ESB-ju fleksibilnost, s tem pa dovoljuje podporo skorajda neomejenemu številu uporabniških vmesnikov. Od mobilnih telefonov in RFID (angl. *radio frequency identification*) naprav do mainframe sistemov. Podpira napredne in modularne nabore programskih komponent, storitev in zdajšnjih aplikacij. Omogoča jim skupno delovanje v mrežnem okolju.

2.6 UPRAVLJANJE

Upravljanje SOA je ustvarjanje, komunikacijo, uveljevanje in prilagajevanje smernic, ki se uporabljajo za usmerjanje in kontrolo izdelave ter izvedbe življenjskega cikla storitev.

Jean Jacques Dubrey v svojem članku »*Establishing a Service Governance organization*« poda dober cilj upravljanja storitev: Upravljanje uveljavlja skladnost z arhitekturo ter semantiko in olajša ravnanje razvoja, uporabo in evolucijo storitev na ravni podjetja. Upravljanje sestoji iz nabora smernic, nabore pravil za izvedbo storitev in nabore procesov, za pravilno izvedbo smernic, ki jih ponudniki storitev in uporabniki morajo sprejeti. Obstajati mora organizirana struktura, ki definira in izvede upravljalne smernice in pogosto tudi skladišča za avtomatizacijo in uveljavljanje.

Upravljanje SOA naj bi vsebovalo:

- Smernice (angl. *policy*) – Regulirajo definicije storitev in izboljšave lastništva, vlog, kriterij, preverjanje vodil itd.
- Identifikacijo vlog, odgovornosti in lastništva.
- Izvajanje smernic.
- Vodila, primeri, ki omogočajo lažje prilagajanje zahtevam upravljanja.
- Pregledi definicij storitvenih vmesnikov. Pregledi zagotavljajo, da so definicije v skladu s standardi, poslovanjem in informacijami.

Poslovanje in informatika govorita dva različna jezika. Mnogo ljudi se znajde v enem ali drugem v obeh hkrati pa večinoma ne. Za skupno delovanje ljudi v SOA je potrebno imeti v mislih upravljanje, ki bo poskrbelo za poravnanje poslovanje in informacijske tehnologije. Cilje se uporablja kot vodilo strategije SOA in ravnanje s portfeljem podjetja.

Življenjski cikel upravljanja SOA se deli v štiri časovne faze: načrtovanje, izvedba, uvajanje in sprememba. (Rosen, 2008, str. 451)

- Čas načrtovanja - Upravljanje zagotavlja definiranje in nadzor nad izdelavo smernic ter storitev podjetja. Smernice bodo skrbale za usmerjanje in nadzor nad izvajanjem življenjskega cikla storitev podjetja. Poleg smernic za zagotavljanje izvedbe moramo izdelati tudi smernice za čas, ko storitve tečejo na sistemu, ki določajo natančne omejitve in zmogljivosti specifičnih storitev. V tej fazi je arhitekt zadolžen za definicijo in avtorizacijo smernic za skladnost standardov, zahteve zasebnosti, nadzora dostopa, zanesljivost, delovanje, komunikacija in razvoj SLA.
- Čas uvajanja - Upravljanje v tem obdobju vključuje procese testiranja in kontroliranja skladnosti s smernicami podjetja z namenom uvajanja storitev v SOA. Vključuje opcije uvajanja, saj naj bi spoštovanje smernic določalo ali naj se storitev uporablja na mreži.
- Čas delovanja - Tukaj se upravljanje nanaša na proces uveljavljanja spoštovanja smernic

storitev v času delovanja. Poleg tega naj bi stalno nadzorovali skladnost, izvajali revizijo in ukrepe za zbiranje statističnih podatkov o delovanju.

- Čas spremembe - Upravljanje vključuje delo s storitvami skozi celoten cikel spremembe. V življenjskem ciklu storitve se vmesniki, smernice storitev in dogovori lahko velikokrat spremenijo. Upravljanje v času sprememb se osredotoča na staranje, verzioniranje in prilagajanje smernic za čas delovanja.

2.7 SODELUJOČI

Kdo pa so nekateri sodelujoči, ki sodelujejo v upravljanju SOA skozi življenjski cikel? Skupine za upravljanje SOA v podjetju določajo procese, ki se pojavljajo skozi življenjski cikel kot tudi vloge in odgovornosti sodelujočih. Ena oseba lahko zapolnjuje več vlog. Nekateri med njimi so (Rosen, 2008, str. 460):

- **Vodja za reševanje problemov** - V večini primerov ima poslovni arhitekt nalogo reševanja specifičnega poslovnega problema. Naloge zaposlenega vključujejo predvsem prevajanje poslovnih zahtev v predloge storitev.
- **Arhitekt funkcij** - Arhitekt mora vzdrževati in izboljševati funkcijski model podjetja. Vzdrževati mora slovar sporočil (semantična sporočilnost), ki skrbi za interoperativnost med storitvami podjetja. Skrbi pa tudi za poravnavanje storitev z funkcijskim modelom.
- **Arhitek portfelja** – Arhitekt, ki ravna s portfeljem aplikacij. Zaposleni skrbi za poravnanost storitev z določenimi usmeritvami danih portfeljev. Zagotavlja poslovne odnose med poslovnimi enotami, kot z ostalimi portfelji. Končno pa služi kot strokovnjak za trenutno funkcionalnost aplikacij znotraj portfelja.
- **Arhitekt podjetja** (angl. *enterprise architect*) - Igra ključno vlogo v identifikaciji storitev in načrtovanju. Zadolžen je, da se vsaka storitev sklada s kontekstom podjetja. Med drugim je tudi zadolžen za pristop integracije in načrt adapterjev. Arhitekt izbira primerne platforme za izvedbo storitev in je zadolžen za nefunkcijske zahteve storitev in njihove izvedbe.
- **Arhitekt poslovnih procesov** – Arhitekt, ki ravna z poslovnimi procesi podjetja in njihovo evolucijo. Sodeluje z arhitektom funkcij in s tem zagotavlja naravnanost storitev s trenutnimi in prihodnjimi poslovnimi procesi.
- **Knjižničar storitev** – Oseba zadolžena za vzdrževanje registra storitev; hrbenice upravljanja storitev. Zagotavlja, da so vse informacije o storitvah v standardizirani obliki,

točne, na tekočem in pravilno klasificirane.

- **Arhitekt vzdrževanja** – Arhitekt zadolžen za vzdrževanje in izboljšave delovanja storitev. Vsebuje naloge kot so standardizacija programskega vmesnika aplikacij storitev, podpora v času delovanja, nadzorovanje.
- **Razvijalec aplikacij** – Oseba, ki gradi in vzdržuje poslovno funkcionalnost za osnovne storitve.
- **Sestavljalec storitev** – Razvijalec, ki gradi in vzdržuje poslovno funkcionalnost za kompozitne storitve.
- **Preizkuševalec storitev** – Predstavnik za zagotavljanje kvalitete zadolžen za testiranje storitev.
- **Specialist storitvene arhitekture** – Infrastrukturni inženir zadolžen za izdelavo diagramov in topologij uvajanja storitev in vzdrževanje registrov storitev in rešitev ravnanja s storitvami. Nadzoruje pa tudi uporabo storitev.

CIO (angl. *Chief Informations Officer*) - Vodja informacijske službe je lahko najpomembnejša oseba v poslovnih procesih in tehnologiji SOA, ki jih organizacijsko sponzorira direktor ali organizacijski direktor. Kritičnost vloge leži v izboljševanju poslovnih procesov podjetja s poslovno strategijo SOA in tehnologijo SOA. Za ta položaj je primerna oseba, ki ima široke izkušnje iz poslovanja in tehnologije. S tem lahko zagotovi SOA kot strategijo podjetja.

Upravljanje storitev podjetja, ki temelji na strateškem načrtu v sodelovanju s poslovnimi enotami podjetja zagotavlja ponovno uporabnost storitev. Vodja informacijske službe se mora pozicionirati kot nekdo, ki prispeva k poslovni strategiji z izobraževanjem in proaktivnim vključevanjem sponzorjev v upravi podjetja in poslovnih enotah o pomembnosti SOA in novih tehnologij (Lawler & James, et al., 2008, str. 230).

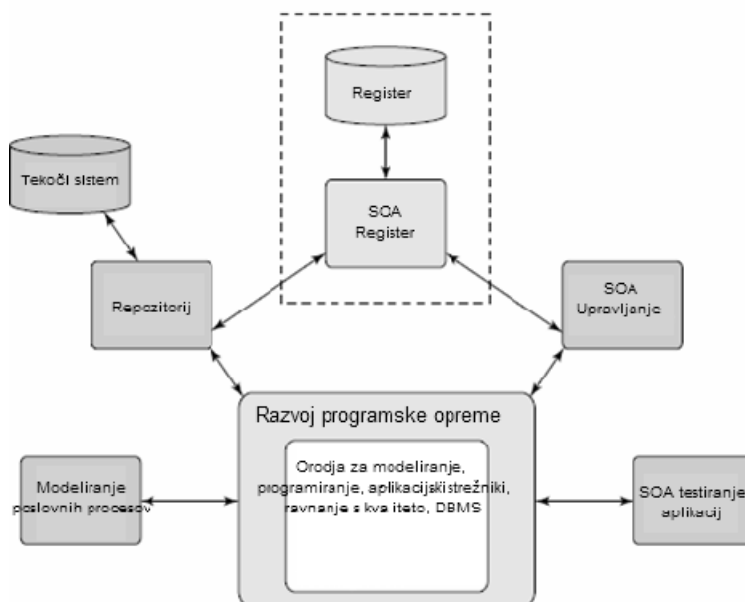
2.8 REGISTER

SOA register je osrednja referenčna enota namenjena iskanju poslovnih storitev in zagotavljanju opisov teh storitev. Register shranjuje informacije o vseh poslovnih storitvah, ki so bile odobrene za uporabo in so prestale sito pravil informatike upravljanja. Register ravna z metapodatki poslovnih storitev. Vključuje informacije o zgodovini posamezne storitve (kdo jo je ustvaril, spremenil, namen uporabe, dovoljenje dostopa do informacij). Ko je storitev odobrena in je zagotovljeno, da je njena uporaba varna v celotnem podjetju, se jo zabeleži v

registru. Register je nujno posodabljeni sproti v realnem času v koraku s poslovnimi pravili (Hurwitz, 2007, str. 97).

Register SOA zagotavlja poslovnim uporabnikom vpogled v zbirko poslovnih storitev. Storitve lahko uporabljajo tudi zunaj organizacije npr. stranke in poslovni partnerji. V registru se shranjujejo ne le definicije programskih komponent za razvijalce in poslovne analitike, vendar je tudi kraj kjer se objavljajo poslovne storitve v korist poslovnim partnerjem in strankam, ki se hočejo neposredno povezati na te storitve. Ker hočemo hitro delovanje se morajo poslovne storitve dinamično povezovati med seboj v realnem času. Po drugi strani pa repozitorij SOA omogoča razvijalcem osrednjo lokacijo za gradnjo in spreminjanje storitev. (Biberstein, 2008.) Dobro organiziran register je hrbtenica za upravljalne operacije in podpira učinkovit razvoj aplikacij.

Slika 5: Upravljanje, register in repozitorij



Vir: J. Hurwitz, Service Oriented Architecture For Dummies, 2007, str. 190.

Dobro definirano in namenoma oskrbovan repozitorij omogoča zaposlenim takojšnjo povezanost in sodelovanje v reševanju zahtev kupca v najkrajšem možnem času in na najvišji kvalitetni ravni. Oboje zagotavlja zaželeno poslovno gibljivost podjetja.

2.9 REPOZITORIJ

Kot sem že omenil je repozitorij (angl. *repository*) ključnega pomena za ravnanje s procesom razvoja programske opreme. Upravljanje SOA se zanaša na repozitorij SOA za shranjevanje komponent in ostalih gradnikov aplikacij. Repozitorij vsebuje in zagotavlja dostop do izvirne

programske kode aplikacije. Razvojna skladišča, ki shranjujejo in vzdržujejo celoten zapis vseh stopenj razvoja niso nova. Vendar jih uporablja le malo mrežnih strani kot sistem beleženja vseh aplikacij, ki tečejo na strani.

Kaj je razlika med registrom SOA in repozitorijem SOA? Mnogi ju bodo poskušali združiti v enoten pojem, kar pa ne smemo. Razlika je v statičnosti skladišča in dinamičnosti registra. Repozitorij je statičen, saj se ne spreminja iz trenutka v trenutek. Razvijalci storitev te odložijo v repozitorij, skupaj z informacijami o zgodovini in verzijah. Vendar te spremembe ne nastajajo trenutno, zato je repozitorij dokaj stabilen. Repozitorij podpira SOA, vendar ne na dinamičen način. Register po drugi strani jemlje storitve iz skladišča. Ko je storitev pod nadzorom registra, se jo lahko uporablja. Lokacija storitve se lahko spreminja. Register sledi tem spremembam in se tako dinamično odziva na zahteve operacij SOA.

2.10 POSREDNIK

Posrednik (angl. *broker*) povezuje storitve. Informacije, ki jih potrebuje poišče v registru. Register in posrednik storitev morata delovati usklajeno. Posrednik mora poznati pravila o delovanju storitev, ki so shranjena v registru, ko se izvede storitev. Register vsebuje pravila o prevajanju, storitvenih ravneh, varnosti in raznih poslovnih storitvah. Delovanje posrednika sledi naslednjim korakom (Hurwitz, et al., 2007, str. 104):

1. Uporabnik se prijavi in zahteva npr. aplikacijo o obdelavi in izpolnitvi naročila. Trenutno ta aplikacija ni v uporabi, zato se aktivira posrednik.
2. Posrednik poišče v registru navodila za izvedbo aplikacije. Hkrati pogleda v pravila storitvenih ravni, če je dovoljena uporaba aplikacije.
3. Posrednik preveri, če delujejo ostale komponente potrebne za izvedbo, recimo skladiščni sistem. Če ne delujejo, jih posrednik zažene.
4. Sledi iskanje informacij o vmesnikih med posameznimi komponentami. Ker so bile v tem primeru komponente že povezane v preteklosti, imajo komponente ustrezne vmesniške informacije v svojih adapterjih. Posrednik ugotovi katere informacije potrebuje. Komponente se nato povežejo neposredno.
5. Posrednik storitev sproti ugotavlja, da je pri vsaki povezavi med komponentami potrebno izvajanje dodatnih pravil. Predpostavimo, da je kupec ustvaril naročilo preko 5000€. Poslovna pravila zapovedujejo, da je potrebno za naročila v tako visokih zneskih izvesti pregled kupčevega kredita. Posrednik se poveže z zunanjo storitvijo pregleda kredita, a še eno pravilo pravi, da to stori samo po potrebi. V tem primeru se posrednik ne poveže s storitvijo.
6. Posrednik storitve zagotovi glavnim komponentam storitve vse informacije za povezavo z generatorjem pravil za izvedbo pravil, ki so tam shranjene. Komponente se nato povežejo z generatorjem pravil in celotni poslovni proces se izvrši. Posrednik storitev je organiziral delovanje in se deaktivira.

2.11 VARNOST

Izvedba SOA obstaja v skoraj vseh industrijah. Razvijalci hvalijo izjemno interoperabilnost med aplikacijami zgrajenimi s SOA. Vendar včasih razvijalci ne poskrbijo za zaščito SOA storitev in arhitekture. Združeno s protokoli, ki imajo dostop skozi požarne zidove preko standardnega port-a 80 (angl. *firewall*) kot je npr. SOAP, ki je pogost pri mrežnih storitvah, je SOA lahko varnostna bomba. Nenatančni prodajalci izvedb SOA, problemi konfiguracij in napake v kodiranju lahko v praksi vodijo do izkoriščanja slabosti v mrežnih storitvah. Razvijalci in sistemski administratorji morajo razumeti tveganje, ki ji porajajo slabosti in imeti v mislih izogibanje težavam pred uvedbo storitev. (NSA, *SOA Security Vulnerabilities Web Services*)

Pri področju varnosti je potrebno paziti na več vidikov. Lahko pa se jih lotimo na različne načine. Razdelimo jih v nekaj širših kategorij (Josuttis, 2007, str. 173):

- Avtentikacija; S preverjanjem pristnosti ugotavljamo identiteto. Identiteta je lahko uporabnik, fizična naprava ali zunanji vložnik storitve. V primeru uporabe SOA to pomeni kdo je kontaktiral storitev.
- Avtorizacija; Avtorizacija ugotavlja katere pravice ima identita. Ali ima identiteta pravico uporabiti storitev in videti rezultate?
- Zaupnost; Podatke lahko ostanejo zaupni v prehodu ali v hrambi. Samo uporabnik lahko vidi podatke storitve, ko se jih prenaša med ponudnikom storitve in uporabnikom.
- Integriteta; Ključno pri integriteti je preprečiti manipulacijo ali ponarejanje podatkov. S tem se predvsem poskuša preprečiti ponareditev dokumentov avtentikacije in avtorizacije za nepooblaščen dostop do podatkov.
- Dostopnost; Možno je napasti »preplaviti« sistem, tako da postane neoperativen. Tipičen primer je napad onemogočitve storitve (angl. *denial of service*, DoS).
- Računovodstvo; Pri tem vidiku moramo slediti porabi sredstev. Slediti moramo recimo stvarjem, ki so relevantne za managerje, načrtovanje, finance in ostale namene.
- Revizija; Z revizijo si pomagamo pri ocenjevanju varnostnih konceptov in njenih aplikacij z namenom izboljševanja zanesljivosti delovanja. Revizija lahko vsebuje snemanje vse za varnost relevantnih podatkov, tako da lahko zaznamo in analiziramo luknje v varnosti in napade. Vsebuje tudi spremljanje, logiranje in sledenje vseh relevantnim tokom podatkov. Dodatno lahko revizija deluje tudi kot funkcionalna komponenta. Ko storitve manipulirajo s podatki, je to manipulacijo potrebno revidirati.

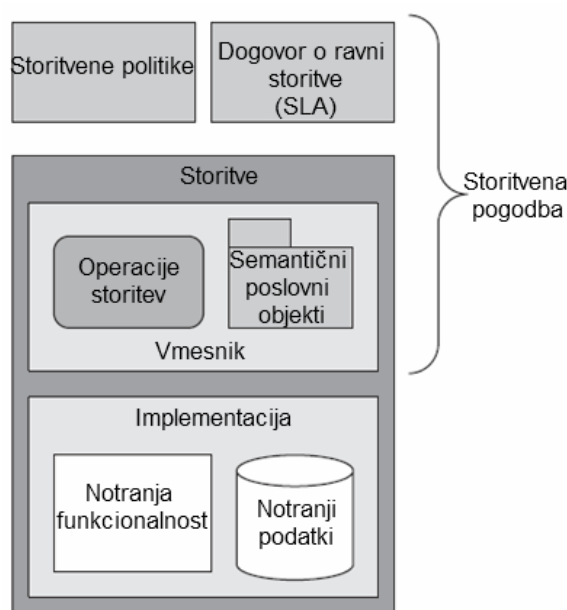
Vendar je v praksi slika varnosti SOA drugačna. Pogosto se jo zanemarja zaradi naporov, ki so potrebni, nezmožnosti zagotoviti popolno varnost, razen izklopa sistema, domneve, da so običajni varnostni mehanizmi zadostno sredstvo preprečevanja vdorov. Predvsem požarni zidovi in posebni protokoli kot je SSL. V težave nas lahko spravijo tudi domneve o dovolj varni infrastrukturi SOA in nedoločene odgovornosti zaposlenih za varnost. V SOA se

osredotoča na izgradnjo boljših požarnih zidov, virtualnih privatnih mrež (angl. *Virtual Private Network; VPN*), sistemov zaznavanja vdorov (angl. *Intrusion Detection Systems; IDSes*) itd.

3 STORITVE

Storitve so v osnovi vmesniki za mnoga sporočila, ki vračajo informacije in/ali spreminjajo stanje povezane entitete. Ali kot je povedal Yefim V. Natis, analitik pri analitični hiši Gartner, eden izmed tistih, ki je skoval pojem SOA: V osnovi je SOA programska arhitektura, ki se začne z definicijo vmesnika in gradi celotno topologijo aplikacij kot topologijo vmesnikov, izvedbo vmesnikov in vmesniških klicev. SOA bi bilo glede na to bolje klicati vmesniško usmerjena arhitektura. Po teh besedah je težko narediti definicijo storitve bolj specifično. Vse kar se lahko uporabi kot vmesnik in predstavlja samovsebujočo poslovno funkcionalnost je lahko storitev. Kot smo že rekli so storitve diskretne enote poslovne funkcionalnosti, ki so na voljo na podlagi storitvene pogodbe. Storitvena pogodba določa vse interakcije med stranko storitve in ponudnikom storitve. Vsebuje naslednje elemente: vmesnik storitve, dokumente vmesnika, storitve smernic, kvaliteto storitve (ang. *quality of service; QoS*), zmogljivost.

Slika 6: Ločitev vmesnika in izvedbe storitve



Vir: M. Rosen, *Service-Oriented Architecture and Design Strategies*, 2008, str. 51.

Ena izmed glavnih razlik med storitvijo in ostalo programsko opremo kot so komponente ali objekti je, da storitev eksplicitno upravljamo. Kvaliteta storitve in zmogljivost je določena preko dogovora o ravni storitve (ang. *service level agreement*). Celotni življenjski cikel je nadzorovan, od načrtovanja do izvedbe in od izboljšav do vzdrževanja. Storitev ima pomembo značilnost razvidno iz spodnje slike. Storitev ločuje vmesnik od izvedbe.

3.1 VMESNIK

Vmesnik določa operacije storitve, torej kaj storitev počne. Storitve vsebuje različne, a povezane operacije. Izvedba storitve določa kako storitev omogoča zmogljivosti vmesnika. Izvedba lahko temelji na obstoječih aplikacijah, na orkestriranju drugih storitev, da kombinirajo svoje zmogljivosti, na programski kodi napisani posebej za storitev ali pa na vse skupaj. Pomembno je, da uporabniki storitve vidijo le kaj storitev omogoča in ne kako deluje. Izdelovalec storitve lahko prosto spreminja izvedbo storitve, dokler s svojimi dejanji ne spremeni vmesnika ali delovanja storitve. Nova storitev lahko temelji na obstoječi funkcionalnosti že obstoječe aplikacije. Ko je pogodba za vmesnik sprejeta, lahko stranke začnejo uporabljati storitev. Med tem lahko proizvajalec ustvari novo in moderno izvedbo in umakne staro aplikacijo iz prometa. Uporabniki storitve ne zaznajo spremembe v delovanju, če se le obnašanje in pogodba ne spremenita.

Vmesnik storitve je točka, kjer uporabniki storitve sodelujejo s ponudnikom storitve. Vmesnik določa slog in podrobnosti sodelovanja. Izvedba določa kako posamezni ponudnik ponuja svoje sposobnosti. Koncept povezovalne točke (angl. *connection point*) omogoča lažje vključitev sposobnosti razklopitve (angl. *decoupling*) v načrt rešitve. Obstajata pa tudi dve različni lastnosti vmesnika in izvedbe. To so funkcije, ki se izvajajo in informacije, na katerih se izvajajo. Storitve je potem takem kombinacija nabora operacij in pripadajočih virtualnih poslovnih podaktov, ki prehajajo skozi operacije. Virtualni poslovni podatki so abstrakcije poslovnih enot, ki so neodvisne od podatkovnih shramb in izvedbe.

3.2 ZNAČILNOSTI STORITEV

Poleg značilne strukture imajo storitve tudi svoje značilnosti.

- **Modularnost in granularnost:** Pri SOA so poslovni procesi razdeljeni v modularne samostojne storitve. Storitve same so lahko sestavljene iz drugih modularnih storitev, tako da po potrebi ustvarijo novo kompozitno storitev. Granularnost pa je lastnost funkcionalne bogatosti storitve. Bolj, ko je storitev »grobozrnata«, večja in bolj bogata je funkcionalnost, ki jo ponuja storitev. Grobo-zrnate storitve omogočajo večjo raven funkcionalnosti znotraj posamezne storitvene operacije. S tem se zmanjša kompleksnost, saj imamo tako manj potrebnih korakov za zagotavljanje dane poslovne aktivnosti. Pogosto to dosežemo z sestavljanjem manjših nalog v eno samo grobo-zrnato operacijo. Finozrnate storitvene operacije omogočajo izmenjavo majhnih količin informacij za izvedbo specifične naloge. Primer fino-zrnate storitve je recimo vrnitev podatkov iskanja številu nakupov po računih kupcev glede na naslov kupca.
- **Enkapsulacija** – Storitev izraža strogo ločitev vmesnika storitve (kaj storitev počne) od

izvedbe storitve (kako to stori). Enkapsulacija skriva interne detajle izvedbe storitve in podatkovne strukture pred objavljenimi vmesniškimi operacijami in semantičnim modelom.

- **Šibka sklopljenost** - Sklopljenost opisuje število odvisnosti med ponudnikom storitve in uporabniki. Šibko sklopljene storitve imajo malo dobro poznanih odvisnosti. Močno sklopljene storitve pa imajo mnogo znanih in neznanih odvisnosti. Stopnja sklopljenosti neposredno vpliva na fleksibilnost in doseg sistema.
- **Izolacija odgovornosti** – Storitve so odgovorne za diskretne naloge ali ravnanje z določenimi sredstvi. Ključna lastnost načrta storitev je izolacija odgovornosti za specifične funkcije ali informacij v eno storitev. To omogoča samo en kraj za izvajanje vsake posamezne funkcije. Kar doprinese h konsistenčnosti in zmanjšanju redundnosti.
- **Avtonomija** - Avtonomija je značilnost, ki omogoča storitvam izvajanje, modificiranje in vzdrževanje neodvisno druga od druge in rešitev, ki jih uporabljajo. Življenjski cikel avtonomne storitve je neodvisen od ostalih storitev.
- **Ponovna uporabljivost** – Modularnost, enkapsulacija, šibka sklopljenost, izolacija odgovornosti in avtonomija omogočajo storitvam, da se združujejo v mnoge poslovne procese ali da do njih dostopa veliko število uporabnikov iz različnih lokacij in v mnogih kontekstih. Storitve so deljene in ponovno uporabljive kot gradbeni deli v procesu izdelave procesov in kompozitnih storitev. SOA se uvaja iz različnih razlogov zagotovo pa je ena izmed bolj pomembnih pridobitev koristi ponovne uporabe sredstev. Ponovna uporaba je pomemben del povrnitve investicije.
- **Dinamično odkrivanje in vezava** – Storitve se lahko odkrije med časom, ko so načrtovane preko časovne shrambe storitve zdaj in ne med delovanjem. Uporabniki se lahko dinamično vežejo na ponudnika v času ko storitev deluje. Dinamična vezava povečuje šibko sklopljenost in omogoča dodatne zmogljivosti kot je mediacija.
- **Storitve so brez stanja** – To pomeni, da se storitve ne spomnijo kaj so obravnavale nazadnje in jim je vseeno kaj bodo obravnavale v prihodnje. Pri načrtovanju težimo k temu konceptu, vendar pa to vedno ni mogoče ali praktično. Storitve niso odvisne od konteksta ali stanja ostalih storitev, samo na njihovo funkcionalnost. Storitve brez stanja omogočajo boljšo fleksibilnost, razširitev in zanesljivost.
- **Samoopisna** - Pogodba storitve podaja celoten opis vmesnika storitve, njenih operacij, vhodnih in izhodnih parametrov in sheme. Pogodba lahko vsebuje tudi pred in post-pogoje ter omejitve glede operacij.

- **Kompozitnost** - Storitve lahko sestavljamo v nove storitve ali poslovne procese.
- **Urejeno s smernicami** - Odnose med uporabnikom storitve in ponudnikom ter med storitvijo in storitveno domeno so urejene z dogovori ravni storitve. Smernice opisujejo, kako smejo različni uporabniki uporabljati storitve.
- **Neodvisne od lokacije, jezika in protokola** – Storitve so načrtovane, tako da je njihova lokacija znana in protokoli ter platforme neodvisni. Dostopne so vsakemu avtoriziranemu uporabniku na vsaki platformi iz katerikoli lokaciji.

3.3 VRSTE STORITEV

Ko definiramo storitveno usmerjene rešitve je pomembno, da se spomnimo na referenčno arhitekturo. Arhitekturo, ki določa gradnike SOA: storitve, dele storitev in tokove, ki podpirajo poslovne procese podjetja in poslovne cilje. Referenčna arhitektura ponuja lastnosti in definicije za vsako raven in odnose med njimi ter pomaga pri postavitvi gradnikov znotraj vsake ravni. Način gradnje v plasteh pa zahteva izdelavo arhitekturnih načrtov v SOA. S tem se lažje doseže tudi ponovna uporabljivost rešitev in sredstev.

- **Podjetniški poslovni procesi**- Poslovni procesi obsegajo celotno podjetje in rahlo uporabljajo podporne storitve.
- **Poslovne storitve** – Poslovne storitve so najbolj grobo zrnate storitve. Odkrivajo kompozitne poslovne funkcije visoke ravni podjetju. Funkcije in informacije se ujemajo s semantiko in sintakso potrebne za poslovne storitve.

Poslovne storitve nato sestavljajo naslednje storitve.

- **Storitve LOB** (angl. *Line of business*) – Določene funkcionalnosti LOB, ki so vidne ostalemu podjetju.
- **Navadne poslovne storitve** – Omogočajo aplikacijam deljenje osnovne poslovne funkcionalnosti.
- **Storitve domen** – Storitve domen so srednje zrnate. Omogočajo poslovne storitve, ki so specifične za določeno poslovno domeno in jih uporablja veliko različnih poslovnih storitev v tej domeni. Ne sme se jih razkriti zunaj domene. Primer: potrditev članstva.
- **Storitve uporabnosti** – Storitve uporabnosti so najbolj fino zrnate. Omogočajo storitve na nižjih ravneh, ki ponujajo skupno funkcionalnost preko vsega podjetja.
- **Storitve integracije** – Te storitve razkrivajo obstoječe aplikacije kot storitve namenjene uporabi vsega podjetja. Omogočajo tudi dostop do podatkov podjetja, ki so razpršeni na različnih podatkovnih virih. Granularnost je odvisna od aplikacij, ki jih odkrivajo. Storitve integracije vpeljujejo transformacijo med modelom podjetja in aplikacijskim modelom, tako na ravni funkcionalnosti kot informacijski ravni.

- **Zunanje storitve** – Omogočajo dostop do sistemov in aplikacij ponudnikov in dobaviteljev zunaj podjetja. Primer: potrditev veljavnosti kreditne kartice in sledenje poštnim pošiljkam. Granularnost bo odvisna od posameznega ponudnika storitve. Večinoma pa so fino-zrnate, ker zunanji ponudniki izdelujejo veliko vrst storitev na mnogih področjih.
- **Temeljne storitve** – Omogočajo fino-zrnate storitve, ki se jih uporablja v izdelavi storitev višjih ravni neodvisnih od poslovnih domen. Primer: varnost, logiranje in orkestracija.

Poslovne storitve

- **Storitve nalog** – Storitve, ki izvedejo poslovno funkcijo kot so npr. storitve, ki računajo zavarovalniške kvote ali potrdijo formate nekega naslova. Storitve namenjene nalogam se nahajajo v različnih velikostih. Obsegajo diskretne uporabnostne storitve do velikih poslovnih storitev. Manjše storitve so bolj splošno namenske in ponujajo višji potencial za ponovno uporabo. Poslovne storitve so večinoma veliki kompoziti manjših storitev in lahko podpirajo enega ali več specifičnih procesov. Kot take imajo manj potenciala ponovne uporabe v procesih.
- **Entitetne storitve** – So storitve, ki večinoma ravnaajo dostop do poslovnih entitet. Primeri poslovnih entitet so stranke, smernice, terjatve itd. Pripadajo velikim poslovno-informacijskim konceptom. Entitete so srednje do velike velikosti. Entitete se držijo ločene od poslovnega procesa in so zato del mnogih različnih procesov. Entitetne storitve ponujajo potencialno visoko stopnjo ponovne uporabe. Storitve nalog so aktivne in dodajajo vrednost. Entitetne storitve podpirajo storitve nalog s prilagajanjem informacij potrebnih za izvedbo storitve.
- **Storitve odločitev** – Storitve, ki izvršujejo poslovna pravila ponujajo poslovne odločitve. Primer takšne storitve bi bil Odobritev kreditne zmožnost stranke. Storitve odločitev večinoma ponujajo odgovore z ja ali ne na zahtevna vprašanja ali podpirajo pogoste spremembe zunanjih pravil kot so davčne regulacije. Storitve odločitev so večinoma dodane ostalim storitvam. So majhne do srednje velike.

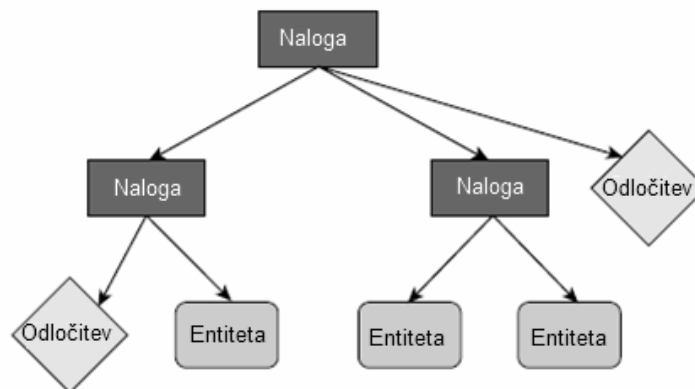
Povezano s temo storitev je vprašanje na katerega je potrebno odgovoriti. Kdo bo plačal za izvedbo novih storitev? Težava je v tem, da je poslovni SOA primer odvisen od nekaterih konceptov, ki bodo povrnili investicijo šele ob razširljivosti. Na primer, integrirane in ponovno uporabljive storitve se ne bodo izplačale za komunikacijo med enim ponudnikom in uporabnikom, ampak šele, ko se bodo pojavili številni ponudniki in kupci. Šibka sklopljenost se bo izplačala šele, ko bomo imeli veliko število rešitev, ki tečejo v realnem času in so razširljive. Kdo naj torej plača za izvedbo novih storitev? Naštejmo nekaj modelov:

- Prvi uporabnik, ki potrebuje storitev za rešitev.
- Vsi uporabniki nove storitve morajo plačati za njen razvoj.
- Ponudba storitve je investicija ponudnika je izdelan cenik, tako da je vsaka zahteva po

storitvi povezana s stroški.

- Obstaja bazen sredstev za financiranje izdelave novih storitev.

Slika 7: Poslovne storitve



Vir: lastna izdelava

V praksi se uporabljata le prvi in zadnji model. Ostale možnosti vodijo do nevzdržnih finančnih in cenovnih modelov. Vendar je možno, da se bodo razvojem SOA prakse financiranja spremenile. Potrebno je biti pozoren na to, da modeli vodijo v sodelovanje v nasprotju z orientacijo v dobičkovna mesta. Lahko se lotimo izvajanja tudi med dobičkovna mesta, vendar je delo ocenjevanja vrednosti vsake rešitve in storitve močno otežena. Poskrbeti je potrebno, da uporabniki plačajo za storitev pravično ceno.

3.4 PROTOKOLI

3.4.1 XML jezik

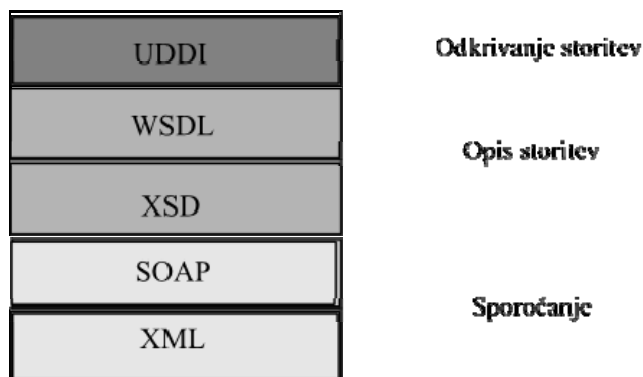
Če hočemo doseči nemoteno spontano izmenjavo informacij, moramo imeti uveljavljen standard za metapodatke. Metapodatki so podatki o podatkih. Metapodatki nam povedo lastnosti hranjenih podatkov. Prav metapodatki so potrebni, če hočemo poenotiti in pohitrili poizvedbe. Z uvedbo metapodatkov nam je omogočeno poizvedovanje po točno zahtevanih podatkih, vendar to ni dovolj, da določimo željene podatke kakor hočemo. Če nas zanima vrednost poizvoda, ki ima metapodatek cena in nam brskalnik pokaže številko 100, ne vemo v kateri valuti je cena. To pa nam oteži razumevanje dobljenih podatkov.

Zato se uveljavlja deljena zveza (angl. *shared context*), ki narekuje način kako moramo metapodatke oblikovati. S tem dosežemo, da prejemnik podatkov razume sporočilo na enak način, kot je to razumel pošiljatelj. Deljeno zvezo zato uporabljajo podjetja pri poslovanju. Tudi ljudje lahko razvijajo deljene zveze za svoje lastne potrebe in jih ponujajo drugim na

spletu. Osnovna XML enota je element, ki je od avtorja določen skupek informacij. Element vključuje ime elementa in vsebino elementa. Od primera do primera moramo paziti, ko določamo imena elementov in oblikujemo vsebino elementov. Začetni in končni priviski določajo ločnice med elementi in vsebujejo ime elementa. Vsebina elementa je sestavljena iz podatkov o drugih elementih. V XML je lahko le en glavni element ki ga imenujemo osnovni element ali podatkovni element. Ta element je kot deblo strukturnega drevesa iz katerega izhajajo drugi elementi. Obstajajo štiri tipi vsebine elementov: podatkovna vsebina (vsebina v kateri so samo podatki), vsebina elementa (predstavlja le druge elemente, prazen element (v njem ni elementov in ne podatkov), mešana vsebina (vključuje podatke in druge elemente). Vsi elementi imajo začetno in končni privisek razen praznega elementa.

UDDI (angl. *Universal Description, Discovery and Integration*) je od platforme neodvisen, na XML temelječ register, ki podjetjem po celem svetu omogoča objavo na internet. UDDI je industrijska iniciativa, ki jo sponzorira organizacija OASIS. Podjetjem omogoča objavo seznamov storitev, odkrivanje ostalih podjetij in določanje kako se storitve in aplikacije povezujejo preko interneta. Poslovna prijava UDDI sestavlja tri komponente; bele strani, ki vsebujejo naslov, kontakt in znane identifikatorje; rumene strani, ki imajo navedene industrijske kategorizacije določene v standardni taksonomiji; zelene strani, ki vsebujejo tehnične informacije o odkritih storitvah.

Slika 8 : Nabor protokolov storitev



Vir: C. Peiris et al, Pro WCF: Practical Microsoft SOA Implementation, 2007, str. 12.

Dobro oblikovan dokument XML ima en osnovni element, vsi neprazni elementi imajo začetne in končne zavijke, vsi prazni elementi imajo pravilno strukturo in elementi so strogo gnezdeni, torej ni prekrivajočih se elementov. V pravem dokumentu XML tudi izvemo v kateri različici XML je dokument napisan. Bistvena prednost XML-ja je, da lahko tako ljudje kot programi uporabljajo dokumente v istem formatu. Ponavadi so potrebe iz naslednjih zahtev: Avtomatizacija toka obdelave podatkov. S poslovnimi dokumenti XML lahko vsi sistemi, ki obdelujejo podatke delujejo medsebojno, medtem ko si poslovneži prosto

ogledujejo dokumente, sistem obdeluje temeljne podatke. Sintetizacija dokumentov iz poslovnih programskih sistemov. Ker poslovni sistemi predstavijo dokumente v obliki njihovih temeljnih podatkov, kar ni primerno za poslovneže, je potrebna sintetizacija poslovnega dokumenta s pomočjo avtomatiziranega orodja v obliko bolj primerno človeškim uporabnikom. Avtomatizirana izmenjava podatkov.

Za uspešno uveljavljanje standarda XML pa je potrebna tudi ustrezna programska oprema. Potrebna je podpora standarda XML v razvijalskih programih, podpora programov, ki producirajo kakršnekoli dokumente, podpora spletne infrastrukture in podpora prevajalskih programov, ki vse dosedanje standarde prevedejo v XML in tako omogočijo hitrejši prehod na standard XML.

3.4.2 SOAP

Programerji obsežno uporabljajo XML v SOA za strukturiranje podatkov, ki jih nato zavijejo v praktično neizčrpen opisni zabojnik. Analogno, WSDL (angl. *Web Services Description Language*) tipično opisujejo storitve same, medtem ko SOAP protokol opiše komunikacijski protokol. SOAP (angl. *Simple Object Access Protocol*) je pomemben standard mrežnih storitev, ki opisuje sporočilne strukture, ko v izvajanju kličejo mrežne storitve (Mattern & Woods, 2006).

SOAP zagotavlja preprost in lahek mehanizem za izmenjavo strukturiranih in natipkanih informacij med uporabniki v decentraliziranem, porazdeljenem okolju z uporabo XML. SOAP sam ne določa semantike aplikacij kot so programski modeli ali semantiko specifičnih izvedb. Namesto tega definira preprost mehanizem za izražanje semantike aplikacij z zagotavljanjem modularnega pakirnega modela in kodirne mehanizme za kodiranje podatkov znotraj modulov. To omogoča, da se SOAP uporablja v velikem številu sistemov.

Sporočilo SOAP je dokument zapisan v jeziku XML, ki opisuje katere operacije se morajo izvesti in kateri parametri se prenesejo k aplikaciji. Po želji lahko SOAP sporočilo vsebuje tudi ostale informacije, ki opisujejo kako naj prejemnik procesira SOAP sporočilo. Sporočilo SOAP zavije klic za storitev v zabojnik SOAP. Zabojnik lahko vključuje glavo za dodatne procesne informacije kot so kvaliteta storitve ali omejitve procesiranja. Vključuje tudi predpisano sekcijo, ki opisuje klicane funkcije in parametre, ki jih prenese storitvi. Prednost SOAP kot kodne ovojnice je, da lahko tekstovna datoteka, ki se jo ponavadi prenaša s HTTP (angl. *Hypertext Transfer Protocol*) ali HTTPS (angl. *Hypertext Transfer Protocol*), prehaja skozi korporativne požarne zidove. Drugi interoperabilni standardi kot so CORBA, DCOM in RPC ne morejo prehajati skozi požarne zidove in so zato neprimerni za mrežne storitve (Wikipedia, SOAP).

3.4.3 WDSL

Dokument WDSL definira storitev kot zbirko mrežnih končnih točk ali vhodov. V WDSL je abstraktna definicija končnih točk in sporočil ločena od mrežnih uporab ali vezanosti na podatkovne formate. (Mattern & Woods, 2006) S tem lahko ponovno uporabimo abstraktne definicije: sporočila, ki so opisi izmenjanih podatkov in tipe končnih točk, ki so abstraktne zbirke operacij. Specifičen protokol in specifikacije za posamezno končno točko sestavljajo ponovno uporabljivo vezo. Končna točka je definirana z združevanjem mrežnega naslova z zvezo in zbirko končnih točk, ki definirajo storitev. Vsebuje naslednje elemente (W3C, Web Services Description Language (WSDL 1.1)):

- Vrste (angl. *types*) – zabojnik za definicijo tipov podatkov, ki uporablja recimo tipni sistem XSD.
- Sporočila – abstraktna, definicija tipov komuniciranih podatkov
- Operacija – abstraktni opis akcije, ki jo podpira storitev
- Vrsta vhoda (angl. *port type*) – je abstrakten nabor operacij, ki jih podpira ena ali več končnih točk
- Zveza – specifičen protokol in specifikacija podatkovnega formata za posamezno vrsto vhoda
- Vhod – posamezna končna točka, ki je določena s kombinacijo zveze in mrežnega naslova
- Storitve – zbirka povezanih končnih točk

WDSL se uporablja za opisovanje klicane mrežne storitve. WDSL je besedni zaklad XML za opisovanje mrežnih storitev, kje se nahajajo in kako se jih lahko pokliče. Z uporabo WDSL opisa lahko tudi avtomatiziramo del kode, ki je potrebna, z uporabo kodirnih generatorjev v kodirnem klientu. Kako vemo katero storitev lahko pokličemo? Operacija klicanja opiše klicano funkcionalnost, vhod, izhod in napačne tipe sporočila povezane z njo. Datoteka WDSL opisuje kako mrežna storitev formatira sporočilo, ki bo izmenjana, tako da izvede storitev in mrežno lokacijo za dostop do storitve.

4 PROGRAMSKE PLATFORME

4.1 IBM

International Business Machines, krajše IBM, je multinacionalno podjetje, ki se ukvarja z računalniškim, tehnološkim in svetovanjem o informatiki s sedežem v Armonk, North Castle, NY, ZDA. Podjetje je eno redkih podjetij s področja informacijske tehnologije, katere korenine segajo v 19.stol. V 1980 je revolucioniralo računalništvo za množice z razvojem enega prvih komercialno usmerjenih osebnih računalnikov. Čeprav je IBM poleg računalniške opreme dolgo investiralo v razvoj programske opreme, se je v drugi polovici 90-ih let 20.stol in v prvem desetletju 21.stol vedno večji poudarek dal na programsko opremo in storitve.

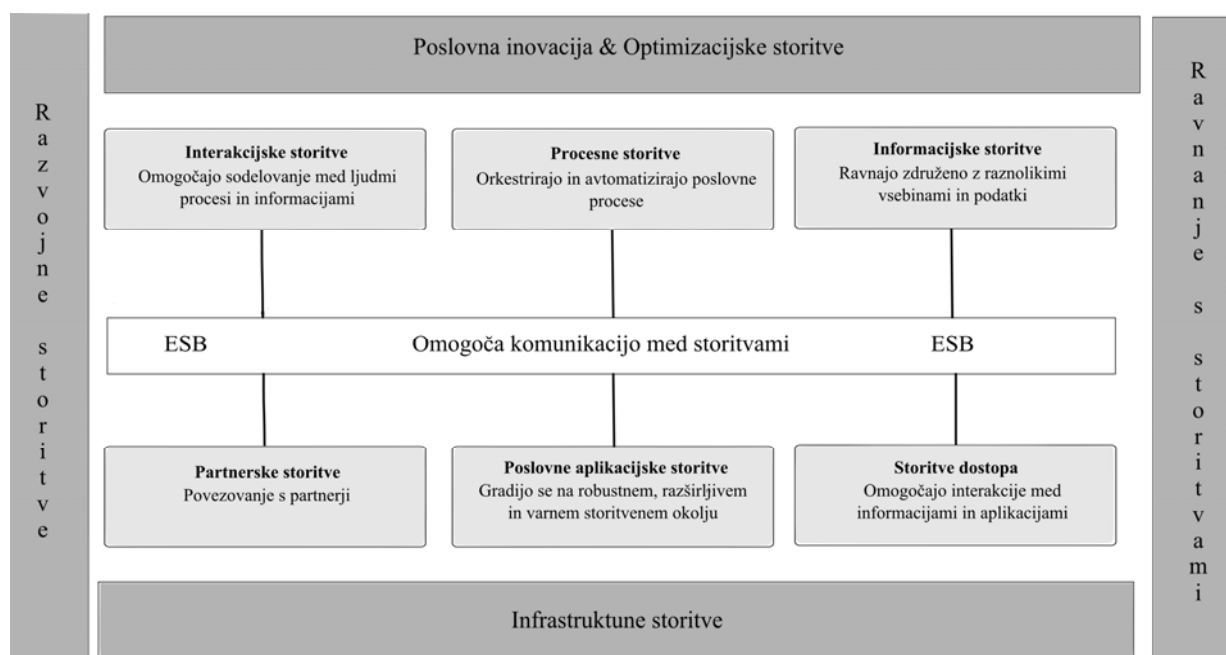
IBM strategija je še posebej široka in se osredotoča na infrastrukturne programske rešitve in storitve. Infrastrukturna programska oprema vključuje programske rešitve, ki zagotavljajo načrtovanje, razvoj, testiranje, ravnanje, varnost in povezljivost programske opreme.

Najpomembnejši IBM diviziji za vsebino diplomske naloge sta IBM Software in IBM Global Services. Prva se ukvarja s produkti in partnerstvi na področju podjetniške programske opreme (angl. *enterprise business software*). Druga pa zagotavlja poslovno in tehnološko svetovanje. Skupaj ustvarjata osnovo SOA pri IBM. Pri podjetju so se odločili, da postavijo SOA za skupni imenovalac svoje strategije razvoja programske opreme. SOA omogoča podjetju integracijo novih produktov z že obstoječo programsko opremo svojih uporabnikov. Razvili in kodificirali so najboljše prakse za hitro izvedbo SOA. V večini primerov se uporabniki odločijo za IBM-ovo programsko opremo ali opremo IBM partnerjev. (Hurwitz, et al., 2007, str. 226)

IBM WEBSHERE

Ime IBM WebSphere se nanaša na blagovno znamko produktov programske opreme. Pogovorno se izraz uporablja za specifičen izdelek: IBM WebSphere strežnik aplikacij (angl. *IBM WebSphere Application Server* ali krajše WAS). WebSphere je načrtovan za vzpostavitev, operiranje in integracijo elektronskih poslovnih aplikacij preko številnih računalniških platform z uporabo mrežnih tehnologij, ki temeljijo na programskem jeziku Java. Vsebuje komponente za operiranje in orodja za razvoj aplikacij, ki tečejo na WAS. (Wikipedia, WebSphere)

Slika 9: IBM referenčni model SOA



Vir: J. Hurwitz et al., *Service Oriented Architecture For Dummies*, 2007, str. 228.

Proizvajalec predstavi Websphere takole: IBM WebSphere je zgrajen, da omogoči gradnjo poslovno kritičnih mrežnih aplikacij. Websphere vključuje širok nabor izdelkov, ki pomagajo strankam razvoj in ponudbo mrežnih aplikacij. Aplikacije so načrtovane za lažjo gradnjo, razpored in bolj produktivno ravnanje z dinamičnimi mrežnimi stranmi.

Sam nabor programskih rešitev, ki jih vključuje Websphere je огromen, zato naštevam primere na vsakem delovnem področju, ki ga pokriva Websphere, da si lahko predstavljamo obseg nabora programske opreme:

- **Strežnik Aplikacij** - IBM Websphere Application Server
- **Poslovna integracija** - Integracija in povezanost aplikacij zmanjšuje kompleksnost poveljivosti aplikacij; adapterji WebSphere in adapterji poslovne integracije WebSphere Podjetniško storitveno vodilo (ESB); IBM Websphere ESB, IBM Websphere Sporočilni broker, IBM Websphere Storitveni Register in Repozitorij
- **Trgovina** – Platforme za e-trgovino, trženje, prodajo, stranke in ostale funkcionalnosti; IBM Websphere Trgovina
- **Mobilni, glasovni in podjetniški dostop** – Programska oprema v mobilnih napravah, ki se lahko poveže s programsko opremo na strežnikih; Everyplace Custom Environment, Everyplace Access, Voice Server, Translation Server, Radio-frequency identification (RFID) Premises Server.
- **Mreže** – Omogoča večprotokolni dostop do gostiteljevih aplikacij; WebSphere Host Integration Solution.
- **Produktivnost podjetij** – Portali na katere lahko namestimo razne storitve, vire podatkov; WebSphere Portal.
- **Razvoj programske opreme** – orodja za razvoj programske opreme; IBM Rational Application Developer, WebSphere Developer, Application Performance Analyzer forSystem.
- **Ravnanje s sistemi** – Določanje, merjenje in ravnanje storitvenih ravni preko heterogenih delovnih okolij; Studio Application Monitor.
- **Ravnanje s podjetniško vsebino**; IBM WebSphere Information Integration.

Za delovanje vse te programske opreme je bilo potrebno določiti standarde. Najnovejši standard je v tem trenutku Websphere 7.0. Podporo se omogoča še standardom Websphere 6.0 in sicer do 30.9.2010 in 6.1 do v prihodnje.

V nadaljevanju naloge se bom osredotočil na tiste rešitve, ki so ključnega pomena za izvajanje strategije SOA v podjetjih in sicer:

- Strežnik aplikacij WebSphere Application Server
- Strežnik procesov Websphere Process Server&Integration Developer
- Register in Repozitorij Websphere Registry and Repository
- Strežnik portala Websphere Portal Server
- Modeliranje procesov WebSphere Modeler

- Nadzor poslovnih procesov WebSphere Business Monitor

STREŽNIK APLIKACIJ

V sedanjem času se v informacijski tehnologiji uporablja več ravni strežnikov. Delijo se na tako imenovano tri ravninsko arhitekturo: podatkovna raven, raven aplikacij in raven uporabnikov. Strežniki aplikacij so namenjeni, čemu že samo ime namiguje, poganjanju aplikacij, ki so kot bazen sredstev med uporabnikom in podatkovno ravno. Z rastjo popularnosti arhitekture z več ravnmi se je povečala tudi ponudba funkcionalnosti strežnikov aplikacij. Dodajalo se je vedno več komponent, ki so omogočila večji razmah predzgrajenim področjem funkcionalnosti. (Greenwald et.al, 2007, str. 333)

Strežnik aplikacij je programski okvir namenjen učinkovitemu izvrševanju procedur (aplikacij, rutin in skript) za podporo izgradnje aplikacij. Pojem se je originalno uporabljal za diferenciacijo med strežniki na katerih so tekle storitve v programskem jeziku SQL in middleware storitve v nasprotju z datotečnimi strežniki. V sedanjem času omogočajo strežniki aplikacij veliko več kot le generiranje mrežnih strani. Izvedejo storitve grozdenja, uravnoteženje pretoka podatkov, preklop na rezervni strežnik. S tem omogočijo razvijalcem osredotočenje na izvedbo poslovne logike – različnih algoritmov, ki vodijo izmenjavo podatkov med uporabniki in bazami podatkov.

IBM WebSphere Application Server ali na kratko WAS je ena najpogostejše predlaganih popularnih platform za poganjanje aplikacij SOA Java. Storitve so lahko zgrajene po želji in se jih poganja na WAS. Izvršilni poslovni procesi so izvedeni z organiziranjem atomskih in kompozitnih storitev, ki so podrejene specifikacijam BPEL. (Bieberstein et al., 2008)

Za IBM Websphere Application Server navajajo pri IBM-u naslednje podatke:

- stabilna arhitektura
- integriteta transakcij
- neprekinjene operacije
- nizki stroški lastništva ali TCO (angl. *total cost of ownership*)
- omogoča računalništvo v oblakih – Temelji na principu internetnega računalništva, kjer so skupna sredstva, aplikacije in informacije na voljo računalnikom in ostalim napravam, ko jih potrebujemo.

Poglejmo si nekaj primerov uporabe Websphere aplikativnega strežnika. Internetna prodajalna eBay.com teče na Websphere in vsak dan premore eno milijardo ogledov strani. Med teniškim turnirjem Wimbledon so uporabniki milijonkrat na uro dostopali do njihove internetne strani z več kot 30000 sočasnih dostopov. Pri investicijski družbi Schwab vsak dan obdelajo 16.5 milijona transakcij.

Slika 10: Skupni stroški lastništva strežnika aplikacij



Vir: *Making business better*, 2010.

Vendar pa WebSphere strežnik aplikacij ni namenjen le največjim podjetjem. Obstaja več različic, ki so namenjene podjetjem različnih velikosti: WebSphere Application Server Community Edition za majhne skupnosti. Ta verzija je napisana v odprtokodnem standardu in njena uporaba je brezplačna. WebSphere Application Server Express, ki je namenjen majhnim podjetjem. WebSphere Application Server namenjen majhnim in srednjim podjetjem. WebSphere Application Server Network Deployment namenjen srednjim in velikim podjetjem. Omogoča visoko raven dostopa tudi v najbolj prometnih urah. WebSphere Application Server for zOS (zOS je eden izmed IBM-ovih operacijskih sistemov) namenjen največjim podjetjem in najvišji ravni dostopnosti (Huelsenbeck, 2006).

WebSphere Process Server

Na WebSphere strežniku procesov tečejo aplikacije, ki so nastale na poslovno orientiranem razvojnem procesu. Strežnik procesov je nameščen na strežniku aplikacij in s tem razširi poslovno storitveno vodilo. Za razvoj aplikacij se na strežniku uporablja WebSphere Integration developer. Zgrajen je na odprtokodnih standardih in izvršuje procese, ki integrirajo poslovne procese, informacije, sisteme znotraj infrastrukture SOA ali ki niso SOA. Strežnik podpira dve zmogljivosti, ki so potrebne za učinkovito SOA: univerzalni klicni model, izveden kot SCA (angl. *service composition architecture*) in univerzalno podatkovno reprezentacijo izvedeno kot poslovni objekti SDO (angl. *service data object*). Poslovni objekti so tip entitet na poslovni ravni v objektno orientirani aplikaciji. Za primer lahko vzamemo poslovni objekt »ravnatelj« z lastnostmi kot so ime, priimek, starost, država med tem ko ima ena proti mnogo povezavo s svojimi zaposlenimi (instancami zaposlenih.)

Razvojni integraciji WebSphere omogoča sestavni urednik, kjer razvojne skupine sestavljajo komponente v module in ti moduli določajo kateri storitveni vmesniki so vidni uporabnikom.

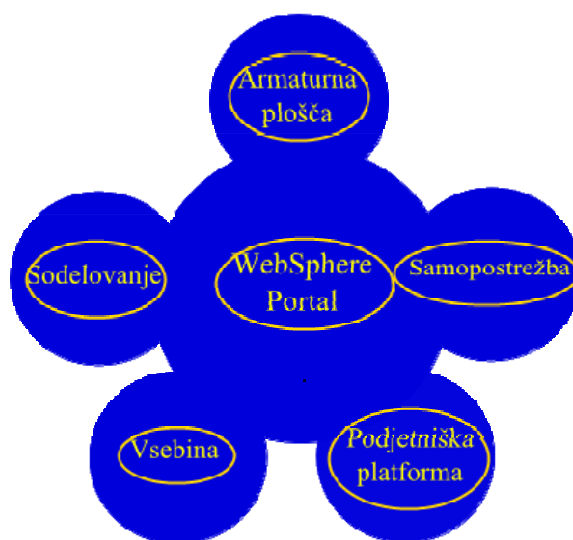
Storitve, ki so na voljo, vključujejo komponente iz mrežnih storitev in storitvenih komponent, ki so na strežniku procesov. (Kulhanek, 2005)

Projekti integracij so zahtevni. Obstaja veliko načinov, kako predstaviti in delati s podatki. Predstavil bom način, ki se osredotoča na tri glavne elemente: podatke, kompozicijo in priklic storitve. Na podatkovni ravni orodje WebSphere Process Integration izvede poslovne objekte, s pomočjo abstraktnih storitvenih objektov in podaljškov kot je podpora za metapodatke, informacije o kontekstu in zgodovini sprememb. Pojem poslovni objekt omogoča način opisa in dostopa do podatkov. Na ravni priklica strežnik ponuja storitveno komponentno arhitekturo. SCA opisuje vse integracije kot storitvene komponente z natanko določenimi vmesniki. SCA vpelje koncept modulov, ki vežejo skupaj storitvene komponente in zagotavljajo nadaljnjo specifikacijo in enkapsulacijo storitev. Tretja raven pri projektu integracije je kompozicija. Tukaj se je potrebno vprašati kaj je potrebno storiti za povezavo storitvenih komponent v tok. Lahko povežemo že narejene module v celotno rešitev ali pa spremenimo le posamezno komponento v danem modulu. (Van de Putte, G. et al., 2005)

IBM WebSphere Portal

Portal je podjetniški portal z naborom storitev, ki so potrebne za ustvarjanje skupne stične točke personaliziranih interakcij z aplikacijami, vsebinami, poslovnimi procesi in ljudmi za enotno uporabniško izkušnjo. Portal povečuje produktivnost in zadovoljstvo strank, povečano operativno učinkovitost, produktivnost in pospešene aplikacije ter uvajanje vsebin. Na spodnji sliki lahko vidimo graf funkcij, ki jih ponuja WebSphere-ov portal.

Slika 11: WebSphere Portal



Vir: P. Monson & F. Feng, et.al., IBM WebSphere Portal V6 Self Help Guide, 2008, str. 3.

IBM WebSphere Business Modeler&Monitor

Procesa modeliranja se lahko lotimo na že znani način z uporabo ključnih konceptov:

- Kronološko orientiranost modelov. Natančni modeli morajo biti orientirani na časovni črti.

Večinoma od leve proti desni v zaporedju.

- Procesi se začnejo s sprožilnimi dogodki in se nadaljujejo proti poslovnim rezultatom.
- Vsem nalogam in aktivnostim so pripisane vloge, ki jih opravljajo zaposleni v podjetju. Potrebno je zajeti tudi vloge, ki se nahajajo zunaj podjetja.
- Modeliranje tokov mora biti oblikovano na razviden način, da lahko vidimo kako se objekti in podatki prenašajo in kam.

Za spreminjanje poslovnih procesov potrebujemo orodje za razvoj in simulacijo procesov. Aplikacijo WebSphere za modeliranje poslovnih procesov. Orodje vsebuje vleci in spusti modeliranje procesov, ponuja strukturirano okolje za lažje sodelovanje pri razvoju procesov. Za realistično simuliranje in analiziranje lahko določimo omejitve s katerimi potem pridobimo vpogled v pomembne poslovne zmogljivosti. Modeliranje in simuliranje pripomoreta k pridobitvi pogleda na operativno raven poslovanja. S tem pa lahko z orodji ustvarimo bolj učinkovite poslovne procese. Za testiranje je omogočen »peskovnik« v katerem lahko preizkusimo nove modele procesov pred izvedbo. Model optimizacije poslovnih procesov ima naslednje prednosti:

- Poravnost preko celotnega podjetja vključno z oddelkom informatike pri vpogledu v procese in operacije za zmanjševanje tveganja in izvedbo sprememb.
- Rešitve, ki nastanejo z informiranim odločanjem. Procesi so integrirani od začetka do konca. Preprosta ponovna uporaba sredstev.
- Fleksibilnost za prilagajanje na dinamične spremembe in učinkovitost pri odražanju rešitev s poslovnimi zahtevami.
- Omogoči storitveno orientirano ravnanje s procesi v podjetju temelječe na SOA.

WebSphere Monitor poslovanja omogoča razumevanje kje nastajajo ozka grla v izvajanju procesov. Z orodjem lahko sledimo trenutne in zgodovinske ravni zmogljivosti pri nadzoru ključnih indikatorjev. Pomagamo si s preglednimi armaturnimi ploščami. Z njimi lahko sledimo specifičnim delovnim predmetom in poskrbimo za njihovo ustreznost. Z nadzorovanjem zmogljivosti poslovnih procesov pripomoremo k najboljšim možnim storitvam za stranke in partnerje.

4.2 Oracle

Oracle je multinacionalno računalniško podjetje, ki je specializirano na razvoj in trženje podjetniških računalniških aplikacij, predvsem sisteme za ravnanje z bazami podatkov. Sedež podjetja je v Redwood City, CA, ZDA. Oracle zaposluje več kot 100.000 ljudi po celem svetu. Podjetje je bilo ustanovljeno leta 1976 in je tri leta kasneje pustilo velik vtis z uvedbo enega prvih produktov relacijskih baz podatkov. V prvem desetletju se je podjetje osredotočalo predvsem na trg podatkovnih baz. Nato se je začelo širiti tudi na razvoj aplikacij, Internet, programske srednje ravni (angl. *middleware*), celostnih aplikacij in svetovanje. Širitev je delno pomenila iskanje primernih podjetij za prevzem za krepitev svojih ključnih sposobnosti

in širjenje trga. Svoj tržni delež na področju programske opreme so povečali z organsko rastjo in velikim številom prevzemov podjetij. V letu 2007 je imel Oracle tretji največji prihodek na področju programske opreme, za Microsoftom in IBM-om. Najbolj znan produkt podjetja je Oracle-ova baza podatkov Oracle Database. Podjetje gradi tudi orodja za razvoj podatkovnih baz in programskih sistemov srednjih ravni za podjetniško ravnanje s sredstvi ERP (angl. *enterprise resource planning*), ravnanje s strankami (CRM, angl. *customer relationship management*) in ravnanje z dobavno verigo (SCM, angl. *supply chain management*).

Danes je podjetje osredotočeno na dve področji: podatkovne baze in celostne aplikacije. V zadnjih letih je Oracle-u uspelo postati pomemben igralec na trgu celostnih aplikacij z nakupi znanih podjetij, ki so se ukvarjala z ERP kot so PeopleSoft, Seibel in JD Edwards. Po drugi strani so prevzeli podjetja, ki so bila na trgu celostnih aplikacij. Oracle-ova vizija SOA je ponuditi strankam celoten nabor SOA produktov. Hoteli so, da stranke dobijo vso integrirano programsko opremo za izvedbo rešitev SOA na enem mestu. Strategija vključuje tudi prevzeme podjetij s produkti SOA in razvojnimi oddelki programske opreme. (Hurwitz, et al., 2007, str. 271)

Oracle Fusion Middleware

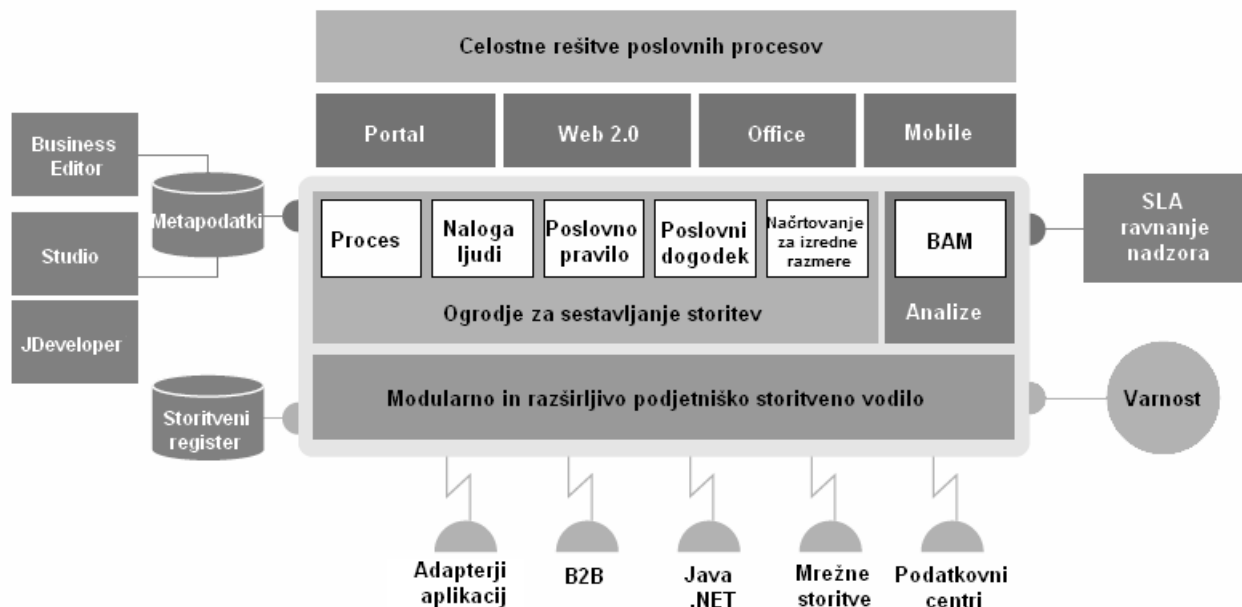
Oracle Fusion Middleware sestavlja število programskih produktov s katerimi se razvija, izvede in ravna s SOA. Oracle je združil svoj strežnik aplikacij in ravnatelj procesov pod skupno znamko Oracle Fusion Middleware (OFM). (Rudie, E.) Poleg so dodali še produkte Java. Produkti omogočajo strankam integriran način za izdelavo in upravljanje s poslovnimi storitvami, integracijo podatkov in storitev ter poganjanje aplikacij in mrežnih storitev. Čeprav OFM vključuje novo kodo temelječo na standardih SOA kot sta XML in BPEL, mnogo tehnologij OFM izvira iz obstoječe Oracle-ove tehnološke ponudbe. Veliko število produktov, ki so vključeni pod blagovno znamko OFM niso klasificirani kot produkti srednje programske ravni. »Fusion Middleware« predstavlja preimenovanje blagovnih znamk podjetja izven osrednje ponudbe programske opreme za podatkovne baze in aplikacije. OFM vključuje mnoge storitve kot so Java2EE in razvojna orodja, integracijske storitve, poslovna inteligenca, sodelovanje in ravnanje z vsebinami.

Oracle Fusion Middleware vključuje komponente iz številnih področij. Naštevam nekaj primerov iz vsakega področja:

- Podjetniški strežnik aplikacij – Oracle Application Server
- Integracija in ravnanje s procesi – Oracle ESB, Oracle BPEL Process Manager
- Razvojna orodja - **Oracle SOA Suite**
- Poslovna inteligenca - Oracle Business Activity Monitoring, Oracle Reports services
- Ravnanje s sistemi - Oracle Enterprise Manager, Oracle Web services manager
- Interakcija uporabnikov - Oracle Beehive, Oracle Portal
- Ravnanje z vsebinami - Oracle Imaging and Process Management

- Ravnanje z identiteto - Oracle Adaptive Access Manager
- Mrežna infrastruktura - Services registry, application-server security

Slika 12: Oracle referenčni model SOA



Vir: J. Hurwitz, et al., *SOA for Dummies*, 2007, str. 274.

V nadaljnjem besedilu se bom osredotočil na opis strežnika aplikacij in celovite ter integrirane Oracle rešitve SOA, Oracle SOA Suite, ki omogoča podjetjem uresničevati svojo SOA arhitekturno rešitev.

Oracle Application Server

Oracle-ov strežnik aplikacij je izdelan v štirih verzijah: Java, Standard, Standard One, Enterprise (Greenwald, 2007, str. 333). Vsaka izdaja ima kumulativno zmogljivosti prejšnje. Vsebujejo določene produkte iz nabora Oracle aplikacij. Možno je tudi dokupovati produkte in jih integrirati v obstoječo arhitekturo. Oracle se osredotoča na razvoj aplikacij z enotno programsko kodo za vse svoje produkte. Aplikacijski strežnik je dostopen na operacijskih sistemih Windows, Linux RedHat in SuSe, Unix kot so HP Compaq True64, HP/UX, IBM AIX in Sun solaris. Ker so pogoji za izvedbo strežnika aplikacij enaki na vseh platformah, lahko podjetja lažje prehajajo med različnimi ponudniki strojne opreme in operacijskih sistemov. Oracleova razvojna strategija omogoča izvedbo novih lastnosti le enkrat za celotno število produktov, namesto posamičnega dodajanja funkcionalnosti.

Oracle SOA Suite

Oracle SOA Suite (OSS), najnovejša različica 11g, vsebuje vse zmogljivosti potrebne za

robustno, gibčno in zanesljivo rešitev SOA. Z njo lahko nadziramo celoten življenjski proces programskih rešitev, kot so aplikacije, storitve in tudi strojno opremo, od razvoja do izvedbe in skrbništva. OSS poenostavi razvoj SOA in ravnanje z izvedbo naslednjih standardov, strategij, tehnologij in orodij:

- Storitveno komponentna arhitektura
- Integrirano razvojno orodje
- Združeno okolje pogona
- Osrednje ravnanje in nadziranje

OSS poskrbi za tehnično infrastrukturo (ravnanje s smernicami, ravnanje s procesi, sporočilnost, ravnanje s transakcijami itd.) uporabniki pa se lahko osredotočijo na razvoj arhitekture in ne detajle izvedeb. Paket vsebuje integrirane komponente, ki se povezujejo s SOA standardi. Sledi opis produktov, ki sodijo v OSS.

JDeveloper

Aplikacija omogoča integrirano razvojno orodje za razvijalce programske opreme in arhitekta. Ponuja preprost, skupen in integriran nabor orodij za vsak produkt v OSS. Eno orodje omogoča uporabnikom standard za razvoj podjetniških aplikacij. Novejša zmogljivost SOA Suite 11g je ustvarjanje SCA kompozitov oz. povezanost heterogenih sistemov v logično, na standardih temelječ sistem. Prednosti orodja so:

- Vsestranskost – Omogoča vizualna in deklarativna orodja za vrsto produktov v Middleware Fusion.
- Nižji stroški – JDeveloper je brezplačno orodje za integrirano razvojno okolje. Znižuje stroške nabave za razvijalce.
- Podpora – Orodje je podprto s proizvajalčevo tehnično podporo 24 ur na dan, 7 dni v tednu.

Oracle BPEL Process Manager

Orodje je namenjeno razvoju in poganjanju poslovnih procesov. Omogoča celostno, s standardi podprto rešitev za ustvarjanje, izvedbo in ravnanje s procesi preko različnih aplikacij. S tem postane zaželeno orodje ustvarjanja integriranih procesov, ki so prenosljivi. Povezuje se lahko z zunanjimi sistemi in procesi in je primeren za izvedbo procesne poslovne logike. Prednosti:

- Lahkota uporabe – Celostno in z lahkoto uporabno orodje.
- Napredna razširljivost
- »Hot-Pluggable« - Izraz, ki ga Oracle uporablja za poudarek povezljivosti orodja z aplikacijami drugih podjetij kot sta IBM in Jboss.

Oracle Service Bus

Oracle je pridobil svoj podjetniško vodilo z nakupom podjetja BEA, ki je bilo leta 2005 eno večjih akterjev na področju SOA. Vodilo se je imenovalo AquaLogic Service Bus, leta 2008 so ga preimenovali v Oracle Service Bus (OSB). OSB teče na operacijskih sistemih Windows,

Linux in Solaris. Zagotavlja vse zmogljivosti, ki se jih pričakuje v podjetjih. (Davies et.al, 2008) Za delovanje SOA je potrebno prosto komuniciranje storitev. Komunikacija je otežena z velikim številom protokolov in s potrebo po povezovanju opisov storitev v jeziku WDSL znotraj arhitekture (Greenwald et al, 2007, str. 347).

OSB je razširljivo, zanesljivo podjetniško storitveno vodilo. Namenjeno je za povezavo, mediacijo in ravnanje interakcij med heterogenimi storitvami, starejšimi sistemi, ERP-ji in ostalimi komponentami v podjetju. Orodje ponuja tudi standardne vmesnike v celem podjetju in ne samo za specifične aplikacije. OSB lahko prevede in transformira sporočilne formate in varnostne sheme, tako da omogoči šibko sklopljenost med sistemi. Združuje elemente SOA in se s tem približa idealu gibčnosti SOA. Prednosti:

- Neprekinjeno delovanje – Izvede stroge pogodbe storitev z inteligentnim uravnoteženjem obremenitve uporabe in dostopnostjo.
- Optimizacija sredstev – Zmanjša stroške uporabe z uporabo obstoječih strežnikov v podjetju, tudi med časom največje obremenitve.
- Lahkota ravnanja – Omogoča takojšnjo, časovno obnovitev nastavitve in dostopnost kjerkoli v mrežnem okolju.

Oracle Web Services Manager

Orodje za ravnanje procesov omogoča centraliziran nadzor nad smernicami za interakcije SOA. Večina podjetij si ob izvedbi SOA rešitev želi poslovno gibčnost s ponovno uporabljivostjo storitev, ki si jih delijo uporabniki znotraj podjetja. Sčasoma je potrebno recikliranje storitev upravljati, da ne zgubimo nadzora nad SOA sredstvi. Oracle-ovo orodje upravlja interakcije med deljenimi sredstvi z ravnanjem in izvajanjem varnostne ter operacijske smernice. Orodje je vgrajeno v Oracle SOA Suite od verzije 11g dalje. Prednosti:

- Varnost – Grajenje varnostnih in operacijskih smernice, ki so lahko uporabljena za nove in obstoječe aplikacije in storitve.
- Avtomatizacija ravnanja s strankami – Avtomatizira izvajanje pogodb med strankami in ponudniki.
- Zmanjševanje tveganosti skladnosti – Združuje ustvarjanje in ravnanje s smernicami.

Oracle Business Activity Monitoring (Oracle BAM)

Orodje za nadzor aktivnosti je celovita rešitev za gradnjo interaktivnih, realno-časovnih nadzornih plošč in proaktivnih alarmov za nadzor poslovnih procesov in storitev. Orodje podaja poslovnim ravnateljem informacije, ki so potrebne za boljše odločitve in spremembe smeri v poslovanju, če to zahteva okolje. Prednosti:

- Racionalizacija operacij – Pridobitev pregleda nad kritičnimi poslovnimi procesi z nadzornimi ploščami in alarmi. Pogled je posodobljen v trenutku dostopa do podatkov.
- Izboljšana produktivnost – Gradnja nadzornih plošč, opozoril in poročil po naročilu za poslovne in IT uporabnike brez programskega kodiranja.
- Fleksibilna integracija – Omogoča vpogled v obstoječe SOA, BPM in dogodkovno gnane

arhitekturne investicije, mrežne storitve, Java sporočilne storitve. Ponuja vgrajeno integracijo s Oracle SOA Suite, Oracle BPM Suite itd.

Oracle Complex Event Processing

Orodje za procesiranje kompleksnih dogodkov omogoča uporabniku okolje za razvoj aplikacij dogodkovnega procesiranja. S tem, ko procesira več tokov dogodkov odkriva vzorce in trende v realnem času, kar preko orodja za monitoring omogoča večjo vidljivost poslovanja. Prepozna lahko trende preko tisočev dogodkov znotraj določenega časovnega intervala ali odkriva manjkajoče dogodke. Prednosti:

- Realno-časovno primerjanje vzorcev – Določi in identificira vzorce za zgodnje odkrivanje poslovnih trendov
- Razširljivost – Pregledovanje tisočev dogodkov na sekundo.
- »Hot-plugging« - Analizira dogodke preko heterogenih sistemov in usmerja izhodne podatke za vizualizacijo in avtomatiziran odziv aplikacijam Oracle.

Bussiness Rules

Bussiness Rules so izvedba poslovne logike ločene od aplikacij, ki jih uporabljajo. Omogočajo fleksibilno recikliranje poslovne logike in konsistentno izvedbo te logike. V primeru spremembe zakonodaje je pomembno, da podjetje dokaže pravilno izvajanje poslovnih odločitev. Namestitev poslovne logike na ločeni sistem, lahko omogočimo konsistentno izvedbo poslovne logike preko celotnega podjetja. Za potrebe revizije lahko pokažemo na specifične primere izvajanja poslovne logike. Orodje je del aplikacijskega strežnika. Skupaj z avtorskim orodjem pravil, ki omogoča uporabnikom ustvarjanje pravil z uporabo angleške sintakse, pogonom pravil (angl. *rules engine*), ki izvede pravila in SDK pravilnik (angl. *system development tool-kit*), ki omogoča dostop do pravil v skladišču pravil. Orodje poslovnih pravil pripomore k fleksibilnosti procesov in aplikacij. Poslovni analitiki in uporabniki lahko zlahka določijo in modificirajo poslovno logiko brez programiranja. Z uporabo razvijalne platforme JDeveloper in postavitev poslovnih pravil na samostojnem strežniku lahko lažje in hitreje spreminjamo pravila s čimer zmanjšamo stroške naknadnih postavitev. Prednosti:

- Visoka zmogljivost – Hitro ocenjevanje pravil z zmogljivim pogonom pravil
- Optimiziran za SOA – Močna integracija z Javo, XML in Oracle SOA Suite
- Razširljiv – Odprta arhitektura, ki omogoča orodjem spremembo pravil na katerem koli vmesniku.

Oracle Repository/Registry

Orodje ni del SOA Suite je pa nepogrešljivo pri upravljanju SOA. Narejeno je za brezhibno integracijo s SOA Suite. Deluje kot edini vir skozi življenjski cikel SOA za zagotavljanje pravih informacij o aplikacijah SOA in odvisnih aplikacijah. Kombinacija skladišča in registra ponuja skupni komunikacijski kanal za avtomatizirano izmenjavo metapodatkov in informacij o storitvah med strankami, proizvajalci, ponudniki in drugimi orodji upravljanja.

Omogoča vidljivost, povratne informacije, nadzor in analizo. Osredotočenje na avtomatizacijo olajša postavitev rešitev SOA in racionalizira upravljanje preko življenjskega cikla. Prednosti:

- Celotna vidljivost – Pregled nad SOA, načrtovanjem in razvojem.
- Gibljiv metamodel – Uporaba zunanjih definicij in primerov ali pa lahka prirejenost potrebam podjetja.
- Avtomatizacija – Avtomatizira delovanje in zmanjšuje tveganje in povečuje sprejetje.

4.3 Microsoft

Microsoft je mednarodno tehnološko podjetje, ki razvija, izdeluje, licencira in podpira širok nabor programske opreme za računalniške naprave. Sedež podjetja je v Redmond, WA, ZDA. Največ uspeha so imeli s prodajo operacijskega sistema Microsoft Windows in Microsoft Office. Podjetje je zasedlo vodilni položaj na področju operacijskih sistemov na domačih računalnikih z MS-DOS v sredini 1980-ih, sledil je operacijski sistem Windows.

Microsoft je bil na področju SOA označen kot zamudnik. SOA so se lotili na svoj način in jo vezali na svoje jedrne sposobnosti prodaje operacijskih sistemov in z njimi povezanimi aplikacijami. Izdelali so lastno referenčno arhitekturo in uporabljajo svoje standarde, kar otežuje povezljivost s produkti drugih ponudnikov rešitev SOA. Gartner in ostali opazovalci industrije so kritizirali Microsoftov poslovni model, da zahteva uporabo oz. nakup ostale Microsoftove programske opreme; operacijske sisteme, aplikacije, razvojna orodja in okolja preden se lahko posvetijo razvoju SOA, ki je prirejen za njihove potrebe. (Scannell & Kanaracus, 2007). Menijo tudi, da je Microsoft SOA bolj stvar priročnosti, v primeru premika trga stran od jedrnih zmogljivosti podjetja. Podjetje naj bi uporabilo SOA za interoperabilnost sistemov Windows in ne široke integracije z ostalimi sredstvi informatike. (McKendrick, 2005)

Microsoft priporoča srednjo pot pri razvoju rešitev SOA, ki združuje »top-down« in »bottom-up« metodologijo. Pri tem pristopu so prizadevanja usmerjena s strateško vizijo in poslovnimi potrebami in združena s postopnimi, iterativnimi projekti SOA, ki postopoma izpolnjujejo poslovne cilje, eno poslovno potrebo za drugo. Microsoft uporablja to tehniko za pomoč uporabnikom SOA rešitev že od izdaje prvega .NET okvira leta 1999.

Podjetje verjame, da obstajajo tri abstraktne ravni znotraj SOA: (Linthicum, 2010)

Odkrivanje storitev

Pomemben del razvoja rešitev SOA je omogočanje obstoječih sredstev informatike. Integriranje pakiranih poslovnih aplikacij, prirejenih rešitev, obstoječe mrežne aplikacije predstavljajo izzive. Pridobitev lastniških protokolov, metapodatkovnih pogledov in transakcijski management. Vsaka spodaj navedena rešitev je zgrajena na skupnih temeljih okvirja .NET, ki omogočajo odkrivanje obstoječih sistemov kot dostopnih na standardih temelječih storitvah.

Kompozicija storitev

Sestavljanje storitev omogoča uporabo novih funkcionalnih procesov, ki podjetju omogočajo vpeljevanje novih poslovnih procesov ali izboljševanje kvalitete storitev za uporabnike in partnerje. Microsoftova infrastruktura SOA ponuja podjetjem fleksibilnost z dodajanjem koordinacije, preglednosti in ravnanje s storitvami. Dosega interoperabilnost z odkritjem storitev, mediacijo in virtualizacijo. S tem pristopom lahko podjetja storitve razvijajo hitreje, izvajajo spremembe, ko se zgodijo in maksimizirajo ponovno uporabo obstoječih sistemov.

Poraba storitev

Veliko pozornosti je namenjeno izvedbi storitev, a je potrebno poudariti, da storitve nimajo vrednosti, če jih ne morejo porabiti ostale storitve ali končni uporabnik. Povečuje se zahteva končnih uporabnikov po dostopu do storitev na več komunikacijskih kanalih kot so računalniška namizja, spletne strani in mobilne naprave. Sestavljene storitve se lahko uporabi za hitro izdajo aplikacij, ki pripomorejo do novih poslovnih zmogljivosti in izboljšane produktivnosti. Izdaje se lahko uporabi tudi za merjenje ROI (angl. *return on investment*) v SOA.

Platforma Microsoft SOA omogoča pet zmogljivosti (Linthicum, 2010):

- Sporočanje in storitve – Osredotočajo se kako sporočila potujejo med pošiljateljem in prejemnikom.
- Delovni tok in procesi – Prilagodljivost modeliranja poslovnih procesov, nadzor in analiza delovnega toka.
- Podatki – Ravnanje s podatki je ključnega pomena za razumljiv prikaz podatkov iz več različnih podatkovnih virov.
- Izkušnje uporabnikov – Integrira porabo storitev na uporabniku prijazen način preko spletnih aplikacij, mobilnih naprav in aplikacijskih klientov.
- Identifikacija in dostop – Olajšuje dostop uporabnikov do zahtevanih aplikacij na različnih ravneh z uporabo centralizirano ravnanega dostopa.

Microsoft BizTalk Server

Produkt popularno imenovan kar BizTalk je podjetniško storitveno vodilo. S pomočjo adapterjev, ki so prirejeni za komuniciranje z različno programsko opremo, omogoča podjetjem avtomatizacijo poslovnih procesov. Omogoča naslednje funkcije: integracijo aplikacij na ravni podjetja, avtomatizacijo poslovnih procesov, B2B komunikacijo in sporočilni broker. Standardna uporaba strežnika omogoča podjetjem integracijo in ravnanje s poslovnimi procesi z izmenjavo dokumentov kot so naročila in računi med različnimi aplikacijami znotraj ali zunaj organizacijskih meja. Jedro strežnika je strežniški pogon BizTalk, ki ima dva dela. Prvi je del, ki omogoča komunikacijo z vrsto druge programske opreme. Zanaša se na adapterje, s katerimi lahko podpira različne protokole in podatkovne formate, med drugim tudi spletne storitve. Drugi del je podpora za grafično definirane procese, ki orkestrirajo izvedbo poslovne logike (Introducing BizTalk Server, 2010).

Tabela 1: Primerjalna tabela programske opreme

Orodje	Ponudnik	Verzija izdaje	Datum izida	Cena	Odprtost	Licenca
WebSphere Application Server	IBM	7.0.0.0	Sep.2008	ok. 2000\$ - 24000\$ odvisno glede na izdajo	Ne	Lastniška
WebSphere AS Community Edition	IBM	2.1.1.4	Apr.2010	Brezplačno	Ja	Brezplačna
WebSphere Enterprise Service Bus	IBM	6.1.2.0	Dec.2007	17500\$ na procesor + 14000\$ vzdrževanje na leto	Ne	Lastniška
WebSphere Process Server	IBM	6.1.2.0	Dec.2007	850\$ na procesorsko enoto	Ne	Lastniška
WebSphere Message Broker	IBM	6.1.0.2	Dec.2007	15000\$-85000\$	Ne	Lastniška
Oracle WebLogic Server	Oracle	10	Mar.2008	25000\$ na procesor	Ne	Lastniška
Oracle BPEL Process Manager	Oracle	10.1.2.0.2	Jan.2006	50000\$ na procesor	Ne	Lastniška
Oracle Enterprise Service Bus	Oracle	10.1.3.1	2006	10550\$ na procesor	Ne	Lastniška
JBoss Enterprise SOA Platform	Jboss Inc.	4.6	Jul.2009	5062\$ naročnina na procesor	Ja	Brezplačna
JBossESB	JBoss Inc.	4.6	Jul.2009	Brezplačno	Ja	Brezplačna
BizTalk Server	Microsoft	2006	2006	34999\$ na procesor	Ne	Lastniška

Vir: lastna izdelava

SKLEP

V diplomski nalogi sem obravnaval storitveno usmerjeno arhitekture in njeno povezanost s poslovanjem podjetij od ravni poslovne strategije do ravni razvijalnih orodij. Poudaril sem, da je potrebno pred sprejemom poslovne odločitve poznati lastno podjetje in način poslovanja. Podjetja morajo biti fleksibilna in se hitro prilagoditi spremembam, če hočejo ohraniti konkurenčno prednost. Preživijo le tista, ki so se sposobna najhitreje odzvati na spremembe. Poslovni procesi se morajo spremeniti relativno pogosto, a hkrati temeljijo na stabilnih zmogljivostih. Fleksibilnost za spremembe izhaja iz zmožnosti hitre izdelave novih poslovnih procesov iz poslovnih storitev. SOA se združuje z ravnanjem poslovnih procesov (BPM) v podjetjih, kjer se zavedajo pomena poslovnih procesov. BPM je pristop za doseganje poslovnih ciljev, koordiniranje procesov v podjetjih, vzpostavljanje najboljših praks in izdelovanje programskih rešitev. SOA združuje poslovanje in zahteve informacijske tehnologije z namenom pridobitve večje tržne prednosti preko gibkih poslovnih operacij. SOA omogoča podjetjem združevanje vodenja poslovanja in informacijske tehnologije preko pogodbe. Določa nabor poslovno orientiranih storitev informatike, ki skupno podpirajo podjetniške poslovne procese in cilje. SOA zagotavlja metode za sistemski razvoj in integracijo, kjer se sistemi funkcionalnosti izrazijo kot interoperacijske storitve. SOA infrastruktura omogoča komunikacijo med različnimi aplikacijami. Poslovna arhitektura prevede poslovno strategijo v akcijske procese. Omogoča organizaciji, da si predstavlja in izraža bistvo svoje organizacije, vzporedno z uskladitvijo poslovne arhitekture z informacijsko arhitekturo. Podjetju omogoča vizualizacijo, analizo, redefiniranje in reinžiniranje načina notranje komunikacije ter komunikacije s poslovnimi partnerji in informatiko. Usklajuje poslovno strategijo z izvedbo.

Uvajanje vsake nove informacijske tehnologije je zahtevna naloga, ki se jo je potrebno rešiti premišljeno. Če je podjetje v prvem koraku ugotovilo primernost uvajanja SOA, glede na svojo razvojno strategijo, zdaj sledi sama izvedba z vsemi svojimi pastmi. Pomembna je primerna metodologija izvedbe. Potrebno je izdelati referenčno arhitekturo SOA, ki se uporabi kot načrt. Vsebuje naj industrijske standarde, naj ne bo vezana na specifične programske platforme in naj naslovi zahteve sodelujočih. SOA projekt mora imeti ves čas podporo posloводства. Vedeti je potrebno tudi kdo bo plačal za razvoj. Določiti je potrebno kako bo podjetje upravljalo z življenjskim ciklom SOA. Življenjski cikel upravljanja SOA se deli v štiri časovne faze: načrtovanje, izvedba, uvajanje in sprememba. Pridobiti je potrebno pravi nabor znanj preko celotnega življenjskega cikla, ki jih imajo projektni vodje, arhitekti, razvijalci aplikacij, administratorji sistema, nadzor kvalitete. Najbolje je imeti osebo, ki bo imela pregled nad celotnim uvajanjem SOA v podjetju. Vodja informacijske službe je lahko najpomembnejša oseba v poslovnih procesih in SOA tehnologiji, ki jih organizacijsko sponzorira direktor. Za ta položaj je primerna oseba, ki ima široke izkušnje iz poslovanja in tehnologije. S tem lahko zagotovi SOA kot strategijo podjetja.

Na koncu sledi še izbira programskih platform na katerih bo podjetje izvajalo svoje poslovne procese. Smo v fazi konsolidacije ponudnikov SOA rešitev. Večino trga zapolnjujejo tri podjetja in njihove programske platforme. IBM z WebSphere, Oracle z Oracle Fusion Middleware, ki vključuje Oracle SOA Suite in Microsoft s svojim Biztalk Server. Večina orodij, ki jih ponujajo niso odprtokodna. Potreben je nakup licence. Katerim so ljubše odprtokodne rešitve se lahko pozanimajo pri podjetju Jboss o njihovi ponudbi. Nakupne cene programske opreme so varovana skrivnost. Zato mora stranka neposredno stopiti v stik s ponudnikom. Cene se lahko nanašajo na število računalnikov, procesorjev, procesorskih enot, vzdrževanje ali letno naročnino. Oracle ponuja enotni paket za izdelavo SOA rešitev, a je hkrati najdražji. Prihodke poskušajo dobiti predvsem iz naslova visokih cen letnih vzdrževanj orodij. Microsoft po mnenju analitikov, poskuša utrditi predvsem svoj položaj ponudnika operacijskih sistemov. Podjetja morajo uporabljati Microsoft operacijske sisteme. IBM omogoča lažje kombiniranje WebSphere produktov. Paziti je potrebno, da so vsa orodja dobro integrirana. Ponudniki imajo programska orodja na voljo za vsako raven organizacij. Od najmanjših z enim zaposlenim do globalnih multinacionalk z milijonom zaposlenih in tisoči strank. Orodja imajo podporo za povezovanje z orodji konkurentov, tako da se lahko povežejo tudi z orodji partnerjev, če ti uporabljajo različne programske sisteme. Podjetje naj se odloči za tisto platformo, ki najbolje ustreza zastavljeni referenčni arhitekturi in nudi vzdržno rast tudi v prihodnosti. Vsako podjetje zase pravi, da ima najboljšo ponudbo, zato je potrebno previdno primerjati prednosti in slabosti posameznih rešitev.

Med nastajanjem diplomskega dela sem predstavil koncept storitveno usmerjene arhitekture. Spoznal sem povezanost poslovne strategije, poslovne arhitekture, storitveno usmerjene arhitekture in programskih platform. S tem sem dosegel cilj, ki sem si ga zastavil v uvodu.

LITERATURA IN VIRI

1. Acharya, A. & Keen, M., et al. (2004). *Patterns: Implementing an SOA Using an Enterprise Service Bus*. IBM. Najdeno 15. maja 2010 na spletnem naslovu <http://www.redbooks.ibm.com/redbooks/pdfs/sg246346.pdf>
2. Arsanjani, A. et al. (2007). *Design an SOA solution using a reference architecture*. Najdeno 3. aprila 2010 na spletnem naslovu <http://www.ibm.com/developerworks/library/ar-archtemp/>
3. *Best Practices for Using WebSphere Business Modeler and Monitor*. (2006) Redbooks Paper, ZDA. Najdeno 10. avgusta 2010 na spletnem naslovu <http://www.redbooks.ibm.com/redpapers/pdfs/redp4159.pdf>
4. Bieberstein, N. et al., 2008. *Executing SOA: A Practical Guide for the Service-Oriented Architect*. Etext edition, Indianapolis, IBM Press
5. Chang, J.F., 2006. *Business Process Management Systems: Strategy and Implementation*. Auerbach Publications
6. Chris, P. & Mulder, D. (2007). *Pro WCF: Practical Microsoft SOA Implementation*, ZDA
7. Davies, J., Schorow, D., Samrat, R. & Rieber, D. (2008) *The Definitive Guide to SOA: Oracle Service Bus. Second Edition*, Apress, ZDA, CA, Berkeley.
8. Davies, J. et al. (2009). *Getting started with Oracle SOA Suite*. CA, ZDA Najdeno 13. maja 2010 na spletnem naslovu <http://www.ayokait.eu/images/pdf/5/getting-started-soa-suite-whitepaperjuly2009.pdf>
9. Dubray, J.J. (2007). *Establishing a Service Governance Organization*. InfoQ. Najdeno 12. avgusta 2010 na spletnem naslovu <http://www.infoq.com/articles/soa-governance-organization>.
10. Fischer-Wasels, B. *SOA Governance – Oracle Enterprise Repository Take the Pain Out of SOA Governance: Streamlining AIA Development Lifecycle*. Najdeno 13. aprila 2010 na spletnem naslovu http://www.fischer-wasels.de/oracle/soa/SOA_GOVERNANCE_MIT_AIA_FW_4_Pro-IT-Forum.pdf
11. Greenwald, R., Stackowiak, R., in Stern, J. (2007). *Oracle Essentials, Oracle Database 11g, FOURTH EDITION*. O'Reilly Media, Sebastopol, ZDA.
12. Huelsenbeck, J. (2006). *IBM WebSphere Discussion with Novell* Najdeno 10. aprila 2010 na spletnem naslovu <http://www.novell.com/partners/ibm/club/doc/websphere.pdf>
13. Hurwitz, J., Bloor, R., Carol Baroudi, C., Kaufman, M. (2007) *Service Oriented Architecture For Dummies*. O'Reilly, Hoboken, NJ, ZDA
14. *IBM Websphere*. Najdeno 7. aprila 2010 spletnem naslovu http://en.wikipedia.org/wiki/IBM_WebSphere

15. *Introducing BizTalk Server 2010*. Najdeno 15. maja 2010 na spletnem naslovu [http://msdn.microsoft.com/en-us/library/aa547058\(v=BTS.70\).aspx](http://msdn.microsoft.com/en-us/library/aa547058(v=BTS.70).aspx)
16. Kulhanek, W., Serna, C, 2005. *WebSphere Process Server: IBM's new foundation for SOA*. Najdeno 15. avgusta 2010 na spletnem naslovu http://www.ibm.com/developerworks/websphere/library/techarticles/0509_kulhanek/0509_kulhanek.html
17. Lawler, James. P. et al. (2008). *Service-oriented architecture : SOA strategy, methodology and technology*. ZDA, FL, Taylor&Francis Group
18. Linthicum, D. (2010), *Service-Oriented Architecture (SOA)*. Najdeno 15. maja 2010 na spletnem naslovu [http://msdn.microsoft.com/en-us/library/bb833022\(v=MSDN.10\).aspx](http://msdn.microsoft.com/en-us/library/bb833022(v=MSDN.10).aspx)
19. M. Josuttis, N. (2007) *SOA in Practice*. O'Reilly. Sebastopol, CA, ZDA
20. *Making business better: Business process management with SOA, Facilitating innovation through improved productivity*. (2010) Najdeno 15. maja 2010 na spletnem naslovu http://download.boulder.ibm.com/ibmdl/pub/software/au/websphere/products/businessint/2007WSB11257-USEN-00_BPM_brochure_0921.pdf
21. Malik, N. (2007) *IFaP : Middle Out Architecture*. Najdeno 16. septembra 2010 na spletnem naslovu <http://blogs.msdn.com/b/nickmalik/archive/2007/02/08/ifap-middle-out-architecture.aspx>
22. Marks, Eric A. (2008). *Service-Oriented Architecture Governance for the Services Driven Enterprise*. John Wiley & Sons, Inc., Hoboken, New Jersey, ZDA
23. Mattern, T., Woods., D. (2006) *Enterprise SOA: Designing IT for Business Innovation*. O'Rilley.
24. McKendrick, J. (2005). *Microsoft's SOA strategy: wait and see?* Najdeno 15. maja 2010 na spletnem naslovu <http://www.zdnet.com/blog/service-oriented/microsofts-soa-strategy-wait-and-see/423>
25. *Microsoft SOA and BPM*. Najdeno 15. maja 2010 na spletnem naslovu <http://www.microsoft.com/biztalk/en/us/soa.aspx>
26. Monson, P., Feng, F. et.al. (2008). *IBM WebSphere Portal V6 Self Help Guide*. Najdeno 11. aprila 2010 na spletnem naslovu <http://www.redbooks.ibm.com/redpapers/pdfs/redp4339.pdf>
27. Mulder, D., Peiris, C. (2007). *Pro WCF: Practical Microsoft SOA Implementation*, Apress, ZDA
28. Open Group. (2009) *SOA Reference Architecture*. Najdeno 10. aprila 2010 na spletnem naslovu <https://www.opengroup.org/projects/soa-ref-arch/uploads/40/19713/soa-ra-public-050609.pdf>
29. Pinus, H. (2004). *Middleware: Past and Present a Comparison* Najdeno 9. aprila 2010 na spletnem naslovu <http://userpages.umbc.edu/~dgorin1/451/middleware>

/middleware.pdf

30. *RATIONAL / WEBSphere: The Real Scoop on Open Source*. IT University, Ottawa, 2010. Najdeno 15. maja 2010 na spletnem naslovu [https://www-950.ibm.com/events/wwe/ca/canada.nsf/vLookupPDFs/1D_-_RATIONAL_WEBSphere_-_The_Real_Scoop_on_Open_Source/\\$file/1D%20-%20RATIONAL%20WEBSphere%20-%20The%20Real%20Scoop%20on%20Open%20Source.pdf](https://www-950.ibm.com/events/wwe/ca/canada.nsf/vLookupPDFs/1D_-_RATIONAL_WEBSphere_-_The_Real_Scoop_on_Open_Source/$file/1D%20-%20RATIONAL%20WEBSphere%20-%20The%20Real%20Scoop%20on%20Open%20Source.pdf)
31. Rosen, M. et al., (2008) *Service-Oriented Architecture and Design Strategies*. Wiley Publishing Inc., IN, ZDA
32. Rudie, E., *SOA Architecture and Oracle Fusion Middleware*. Najdeno 15. aprila 2010 na spletnem naslovu <http://www.afcea-infotech.org/2007/media/presentations/SOA%20Architecture%20and%20Oracle%20Fusion%20Middleware.pdf>
33. Scannell, E. in Kanaracus, C. (2007). *Microsoft Does Have a SOA Strategy*. Najdeno 15. maja 2010 na spletnem naslovu <http://redmondmag.com/articles/2007/08/01/microsoft-does-have-a-soa-strategy.aspx>
34. *Service Oriented Architecture Security Vulnerabilities – Web Services*. Najdeno 12. aprila 2010 na spletnem naslovu http://www.nsa.gov/ia/_files/factsheets/SOA_security_vulnerabilities_web.pdf
35. *Simple Object Access Protocol (SOAP) 1.1*. (2000). Najdeno 15. maja 2010 na spletnem naslovu <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
36. *SOA Source Book*. Najdeno 10. aprila 2010 na spletnem naslovu <http://www.opengroup.org/projects/soa-book/>
37. SOAP Najdeno 17. maja 2010 na spletnem naslovu [http://en.wikipedia.org/wiki/SOAP_\(protocol\)](http://en.wikipedia.org/wiki/SOAP_(protocol)).
38. van de Putte, G. & Gavin, L. (2005). *Technical Overview of WebSphere Process Server and WebSphere Integration Developer*. IBM, ZDA. Najdeno 10. avgusta 2010 na spletnem naslovu <http://www.redbooks.ibm.com/redpapers/pdfs/redp4041.pdf>
39. *Websphere Smarter Software For A Smarter Planet*. Najdeno 15. maja 2010 na spletnem naslovu <http://public.dhe.ibm.com/software/dw/techbriefings/presentations/webspheresoa/appserver.pdf>