

Implementacija SOA koncepata u kompanijama upotrebom Java EE 5 platforme

Aleksandar Aladžić¹⁾

Sadržaj — Ovaj rad ima svrhu da pokaže mogućnosti implementacije SOA koncepata primjenom Java EE 5 razvojnog okruženja u realizaciji softverskih sistema baziranih na SOA arhitekturi u poslovnom okruženju

Ključne reči — Java EE, platforma, servis, SOA

I. UVOD

U svijetu se, poslednjih godina, javila tendencija ka arhitekturnom pomaku u IT kompanija. Sa razvojem novih IT tehnologija, koncepata i pravila poslovanja, u kompanijama se nameće potreba za implementacijom IT arhitektura, koje bi bile dovoljno robustne i fleksibilne da mogu da adekvatno odgovore na sadašnje i buduće zahtjeve poslovanja. Jedna takva arhitektura je i Servisno Orjentisana Arhitektura (u daljem tekstu SOA). Pored odabira određene arhitekture, koja će da se koristi u kompaniji, nameće se i pitanje njene implementacije, jer je arhitektura nezavisna od implementacije. Java EE 5 je jedna od tehnologija koja u velikoj mjeri zadovoljava potrebe za implementacijom SOA. Tokom svog postojanja razvila se u zrelu tehnologiju za implementaciju IT sistema kompanija.

II. GRADIVNI ELEMENTI SOA

SOA predstavlja stil u arhitekturi u kome se softverske tvorevine predstavljaju pomoću jasno definisanih servisa, kojima se može pristupiti na jasno definisani način i koji takođe, sami mogu da pristupe drugim servisima.

Ona nameće kao cilj konvergenciju poslovnih i informatičkih servisa.

Elemente SOA-e možemo predstaviti sledećom slikom:



SL.1. Arhitekturni okvir SOA

Pošto se nivoi SOA-e ne mapiraju direktno na Java EE nivoe (prezentacijski, poslovni, EIS), kako strukturno,

1) Aleksandar Aladžić, Telekom Srpske, Kneza Lazara 4, 74000 Doboj, (telefon:+387 53 202 203, email: a.aladzic@telekomsrpske.com)

tako i funkcionalno, to pravilna implementacija SOA-e predstavlja veći izazov.

Pored toga, težnja SOA-e je da postigne nezavisnost poslovnih servisa od IT infrastrukture, omogućavajući slabu spregu između poslovnih koncepata i njihovih implementacija, fleksibilnost rješenja i mogućnost brzih promjena IT infrastrukture u skladu sa poslovnim potrebama. Java EE može efikasno da odgovori zahtjevima servisne orjentisanosti na više načina, ali se vremenom nametnulo rješenje upotrebom web servisa. Razlog za to je jasna razdvojenost interfejsa servisa od njegove implementacije, kao i baziranost na široko prihvaćenim standardima.

A. Servisna magistrala

Pred servisnu magistralu se nameću ozbiljni zahtjevi u zadovoljenju potreba za komunikacijom servisa. Ona mora da podržava što veći broj tipova komunikacije. Naime, pošto SOA servisi imaju manju funkcionalnu rezoluciju od enterprise rješenja baziranih na komponentnom modelu Java EE tehnologije, komponente unutar SOA servisa mogu da komuniciraju na bilo koji način da bi ostvarile potrebnu funkcionalnost.

Kod komunikacije SOA servisa, koji su implementirani upotrebom web servisa, komunikacija se odvija preko SOAP poruka, sinhrono ili asinhrono, što Java EE podržava.

Pored toga, Java EE, u svojoj JSR 208 specifikaciji opisuje JBI, kao standardizovanu integracionu platformu za poslovne aplikacije, preko koje može da se odvija komunikacija između različitih servisa. Ona je omogućena upotrebom odgovarajuće JBI komponente, a čije su mogućnosti opisane u WSDL fajlu.

B. Registar servisa

Registri predstavljaju mjesta gdje se čuvaju relevantne informacije o servisima. Informacije se klasifikuju u skladu sa nekim standardom, koji treba da poštuje i provajder i konzument servisa. Java EE ima svoj registar resursa kome se pristupa pomoću Java API-ja. To je JNDI API koji omogućava Java aplikaciji da locira resurse (korisnike, mašine, mreže, objekte, servise, štampače,...).

Pored ovog servisa za lociranje resursa, Java EE podržava i koncept UDDI-ja, baziranog na XML formatu, koji se koristi za smiještanje i traženje resursa, kompletniji je u smislu opisa servisa i omogućava slabu spregu između servisa.

C. Servisi

Servisi predstavljaju ključni koncept na kome se zasniva SOA. On bi se mogao definisati kao nezavisan softverski entitet čija je svrha obavljanje neke konkretne funkcije –

aktivnosti. Pošto SOA, po definiciji, podrazumijeva funkcionalnu dekompoziciju sistema, tako i servisi u SOA-i treba da imaju jasno određenu funkciju. Pored toga, servisi bi trebali da zadovolje većinu ovih karakteristika:

- oni treba da budu nezavisni, što omogućava njihovu efikasnu izmjenu (što ne treba da utiče na ostatak sistema) i ponovnu upotrebu u nekom drugom kontekstu,
- trebali bi imati mogućnost objavljivanja servisnog ugovora, koji služi za opis servisa,
- enkapsulaciju implementacione logike,
- mogućnost grupisanja u veće servise, da bi se omogućila funkcionalna dekompozicija manje granularnosti, ili ponovna upotrebljivost na drugom nivou apstrakcije,
- servisi ne bi trebalo da upravljaju stanjima u konverzaciji sa drugim servisima, jer to negativno utiče na njihovu nezavisnost, a ako je to neophodno, tada bi taj dio funkcionalnosti trebali delegirati nekom servisu, specijalno uvedenom za tu svrhu,
- trebali bi omogućiti da se, na osnovu servisnog ugovora, servis može pozvati i iskoristiti njegova funkcionalnost.

S obzirom na zahtjeve, servise bi mogli, sa funkcionalnog aspekta, podijeliti na:

1. servisi čije su funkcionalnosti u domenu poslovne logike,
2. servisi čije su funkcionalnosti u domenu manipulacije podacima,
3. servisi čije su funkcionalnosti u domenu prilagođavanja "neservisnih" sistema servisnim osobinama,
4. servisi čije su funkcionalnosti u domenu definisanja poslovnih procesa,
5. servisi čije su funkcionalnosti u domenu povezivanja sa IT sistemima drugih kompanija,
6. servisi čije su funkcionalnosti u domenu komunikacije sa korisnicima.

III. NEFUNKCIONALNI SERVISI

Nefunkcionalni servisi su uvijek, u enterprise okruženju, bili jednako bitni kao i funkcionalni. Ako bi sistem imao manjkavosti u zadovoljenju nefunkcionalnih zahtjeva, to bi prilično uticalo na njegov životni vijek, jer bi ga zahtjevi poslovanja vremenom prevazišli i ne bi mogao adekvatno odgovoriti na zahtjeve. Takva situacija vodi kompaniju ka skupim rješenjima, čija implementacija može da traje vrlo dugo.

Efikasan odgovor na nefunkcionalne zahtjeve garantuje trajnost rješenja, kako sa arhitekturnog, tako i sa tehnološkog stanovišta.

Neki od bitnih nefunkcionalnih zahtjeva koji se moraju uzeti u obzir su: performanse sistema, skalabilnost, pouzdanost, raspoloživost, proširivost, upravljivost i sigurnost.

Jedna od trenutno najpopularnijih implementacija zahtjeva je pomoću web servisa. Ovdje razlikujemo web servise prve generacije (UDDI, WSDL, SOAP) i web

servise druge generacije (WS – *). Upravo se standardi vezani za WS – * odnose na definisanje nefunkcionalnih zahtjeva.

IV. JAVA EE 5 RAZVOJ SOA SERVISIA

Prilikom implementacije servisa po SOA konceptima, trebamo imati na umu da servis u SOA-i predstavlja jedan poslovni koncept ili proces, što je u direktnoj vezi sa manjom funkcionalnom granularnošću SOA servisa. Takav koncept omogućava razvoj SOA servisa koji mogu inkorporirati više layer-a i tier-a Java EE 5 implementacionog steka, što može drastično da poveća kompleksnost implementacije i održavanja takvog servisa, a takvi servisi mogu da budu raspoređeni na više računara u distribuiranom i heterogenom okruženju.

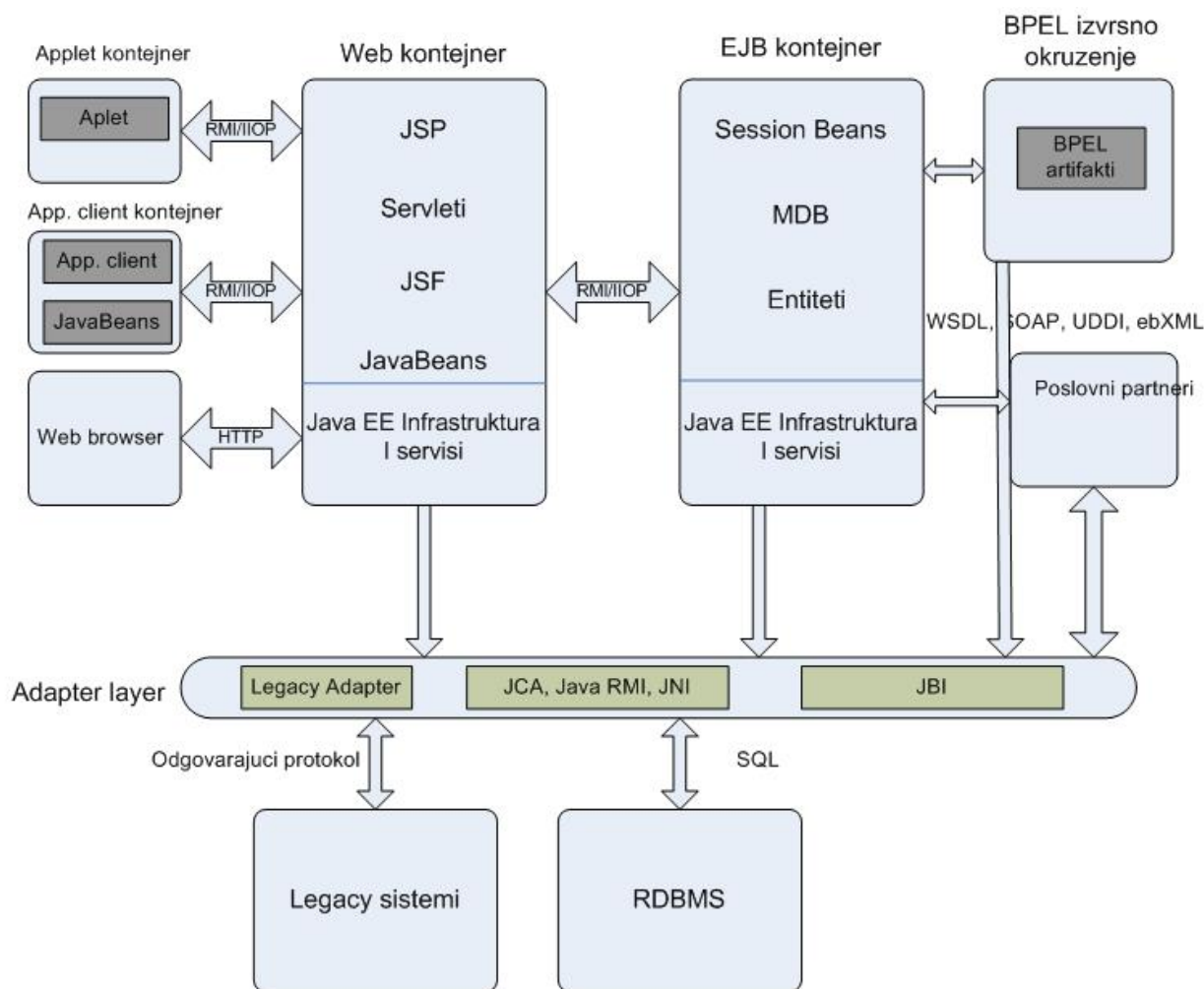
Kako je Java EE 5 zasnovana na komponentnom i distribuiranom modelu, ona predstavlja solidnu osnovu za implementaciju SOA koncepata. Komponentni model omogućava razvoj softverskih tvorevina koje su nezavisne i samostalne, ali su prisnije povezane nego što je to poželjno kod SOA servisa, dok joj distribuirani model daje mogućnost udaljenog poziva funkcionalnosti, kao da se semantički radi o lokalnom pozivu. Pored toga, imamo i prednosti samog programskog jezika Java, kao npr.:

- eksplicitnu odvojenost interfejsa i implementacije (interface i class),
- sigurnost i bezbjednost u izvršavanju,
- portabilnost (JVM).

Implementaciju servisa čije su mogućnosti u domenu manipulacije podacima možemo uraditi pomoću EIS Java adaptera (jednostavnih ili distribuiranih), gdje je uloga adaptera da mapira EIS konkretni API u standardni API, upotrebom različitih nivoa apstrakcije u zavisnosti od kompleksnosti zahtjeva za povezivanje. Druga mogućnost je upotrebom JCA 1.5, koja predstavlja standardnu arhitekturu Java EE platforme za konekciju ka heterogenim EIS-ima. Ona definiše mehanizme za skalabilnost, sigurnost, transakcije, životni ciklus adaptera resursa, ugovor o upravljanju poslom, ugovor o protoku transakcija i ugovor o protoku poruka, što omogućava povezivanje Java EE platforme sa EIS-om na sistemskom nivou. JCA takođe definiše i CCI (Common Client Interface), koji predstavlja klijentski API za rad sa EIS-ima.

Ova tehnologija može da se iskoristi i za implementaciju servisa čije su funkcionalnosti u domenu povezivanja sa IT sistemima drugih kompanija.

Implementacija servisa čije su funkcionalnosti u domenu poslovne logike može se uraditi pomoću EJB tehnologije. Oni predstavljaju komponente na serverskoj strani i služe za implementaciju poslovne logike. Sa njima imamo mogućnost da pamtimo konverzaciona stanja u komunikaciji između klijenta i EJB-a, ili da ih ne pamtimo, u zavisnosti od poslovnih potreba. Takođe, oni nam omogućavaju korištenje sinhronne ili asinhronne komunikacije na nivou poslovne logike, kao i mogućnosti manipulacije EIS-ima preko entiteta. Oni se izvršavaju u EJB kontejneru, gdje, pored logike koju sami implementiraju, mogu koristiti i servise koje im obezbjeđuje kontejner. Oni se implementiraju po POJO



Sl. 2. Java EE 5 infrastruktura primjenjiva u implementaciji SOA

modelu, a, pored mogućnosti upotrebe deployment deskriptora, postoji mogućnost upotrebe anotacija. Mogu se pozivati udaljeno ili lokalno, a moguće ih je pronaći JNDI lookup-om, injection-om, ili anotacijama.

Jedna od velikih prednosti EJB-a je da se stateless session bean-i mogu jednostavno, pomoću anotacija predstaviti kao web servisi, a postoji i mogućnost implementacije klijenta web servisa. Web servisi, zahvaljujući komponentnom modelu Java EE, mogu biti proizvoljne kompleksnosti i inkorporirati komponente iz više tier-a Java EE implementacionog steka (uključujući i čitave baze podataka).

Oni se takođe mogu iskoristiti i za implementaciju servisa čije su funkcionalnosti u domenu povezivanja sa IT sistemima drugih kompanija.

Kod implementacije servisa čije su odgovornosti u domenu definisanja poslovnih procesa, možemo da koristimo BPEL – XML vokabular, specijalno dizajniran za tu svrhu. Određena Java EE 5 razvojna okruženja podržavaju razvoj ovog tipa artifakta (npr. NetBeans 6.8), koji omogućava kreiranje tokova poslovnih procesa i orkestraciju servisa u složenije poslovne procese, komunicirajući sa njima preko korendentnih .wsdl fajlova i zaokružujući razvojni ciklus SOA servisa u Java EE 5 okruženju.

A. Java EE 5 nefunkcionalni servisi

Java EE 5 predstavlja robustnu platformu kada je riječ o zadovoljenju nefunkcionalnih zahtjeva za enterprise rješenjem. Kako je, od svog nastanka, predviđena za takvu vrstu rješenja, o zadovoljenju nefunkcionalnih zahtjeva se od početka vodilo računa. Ona podržava load balancing na svim nivoima (prezentacijski, aplikativni i EIS), što omogućava uticaj na performanse tačno u onom dijelu sistema u kome one opadnu.

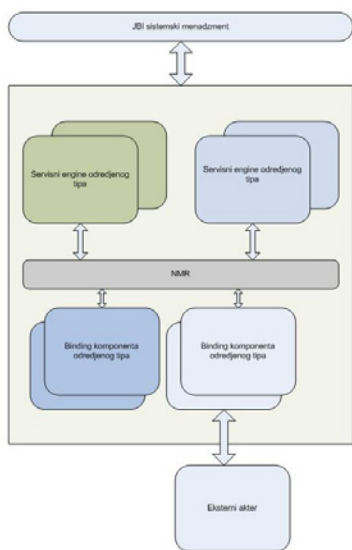
Sa stanovišta sigurnosti, Java EE 5 obezbjeđuje autentifikaciju, zaštitne domene, enkriptovanu komunikaciju, SSL, podršku za CSIV2, autorizacijuna više načina, što daje jedno kompletno rješenje sigurnosti, primjenjivo kod implementacije SOA rješenja.

Java EE 5 platforma podržava klastering, kako na nivou mašine, tako i na nivou kontejnera. EIS-i takođe, mogu da imaju svoja rješenja za klastering, ali su i na ovom nivou ona moguća. Platforma takođe podržava i pooling resursa,, što direktno utiče na kontrolisano i optimalno korištenje računarskih resursa, omogućavajući da više korisnika koristi postojeće računarske resurse.

B. Java EE 5 magistrala

Kompanija Sun Microsystems, za ovaj tip rješenja predlaže upotrebu Java Business Integration (JBI) platforme, koja je opisana u specifikaciji JSR 208. Ona se sastoji od sledećih komponenti:

1. Servisni engine (SE) - JBI komponente, koje omogućavaju uključenje poslovne logike,
2. Binding komponente (BC) – JBI komponente, koje omogućavaju uključenje eksternih konekcija,
3. Ruter za normalizovane poruke – usmjerava normalizovane poruke od ishodišta ka odredištu u skladu sa odgovarajućim polisama i
4. JBI izvršno okruženje – obuhvata JBI komponente i NMR.



Sl. 3. Arhitektura JBI

Ovo rješenje se često koristi u praksi, jer, preko normalizovanih poruka, obezbjeđuje komunikaciju različitih komponenti, što omogućava da se takve, raznorodne komponente, mogu integrisati u jedno rješenje, kako na nivou jednog SOA servisa, tako i na nivou više poslovnih procesa.

Svakako, ovo nije jedino rješenje problema komunikacije na nivou Java EE 5 platforme. Komponente koje čine rješenje na nivou Java EE 5, mogu da komuniciraju i na standardne načine (ne samo preko normalizovanih poruka), ali je prednost JBI rješenja u tome što daje jedno kompozitno rješenje problema komunikacije u enterprise okruženju.

IV ZAKLJUČAK

Ovaj rad je imao za cilj da prikaže moguće načine implementacije koncepata SOA, koristeći pri tom Java platformu Enterprise Edition, koja predstavlja, u svijetu, jednu od vodećih platformi za implementaciju enterprise informacionih sistema. Poseban značaj joj daje i činjenica da na njenim standardima radi, kako akademska zajednica, tako i vodeće svjetske kompanije u području informacionih tehnologija, što na neki način garantuje kompanijama koje je koriste, dugotrajnost njihovih poslovnih sistema, kako u smislu koncepata, tako i u smislu korištenih tehnologija, a vjerovatno će im i olakšati eventualnu migraciju na neke nove sisteme, jer je ova platforma u širokoj upotrebi.

LITERATURA

- [1] Sun Certified Enterprise Architect for Java Study Guide, Second Edition, Prentice Hall, 2010.
- [2] Enterprise SOA: Service-Oriented Architecture Best Practices, Dirk Krafcig, Karl Banke, Dirk Slama, Prentice Hall PTR 2004,
- [3] Service-Oriented Architecture: Concepts, Technology, and Design, Thomas Erl, Prentice Hall PTR, 2005
- [4] Designing Enterprise Applications with Java 2 Platform Enterprise Edition, Nicholass Kassem and Enterprise Team, Sun Microsystems, 2000.

ABSTRACT

This paper has the objective to show implementation capabilities of SOA concepts, using Java EE 5 platform on SOA based systems in enterprise environment.

IMPLEMENTATION OF SOA CONCEPTS USING JAVA EE 5 PLATFORM IN ENTERPRISES

Aleksandar Aladzic