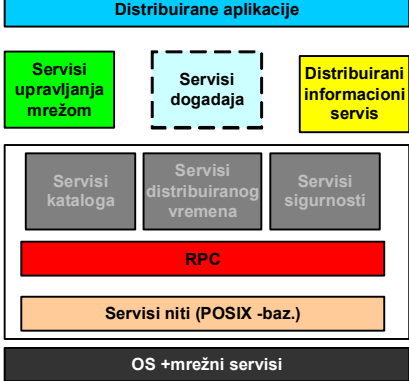
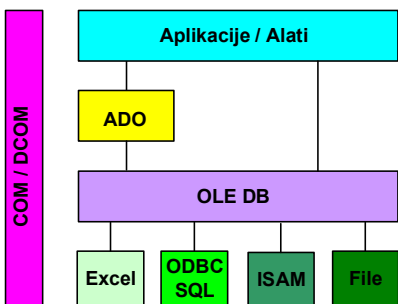
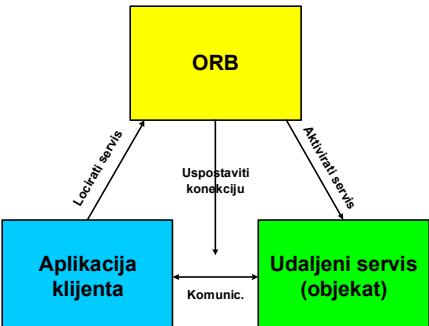


<p><b>124. Transakciiono orijentisan MW (transakcijski monitori)</b>Postoji u dve osnovne kategorije:-Transakcioni monitori (TPM)(Transaction Processing Monitor)-Aplikacioni serveri (AS) (Application server)Koje podržavaju apstrakciju transakcije.-Obavljaju posao na koordinaciji transfera informacija i deljenja metoda između brojnih različitih resursa.-Ova paradigma obezbeđuje odličan mehanizam deljenja metoda.-Transakcioni MW tipično kreira čvrsto povezano EAI rešenje.-Tr MW obrađuje transakcije u ime klijenta i obezbeđuje:-Multipleksiranje BP-Balansiranje opterećenja-Imunost na greške (fault tolerance) -Komunikaciju-Klijentska apl komunicira direktno sa TPM putem TRPC, ponekad RMI, IIOP.-Standardi koji definišu kako radi transakcioni MW su DTP (Distributed Transaction Process) specifikacije donete od strane ISO i X/Open.-X/Open 1991. kreirao TP referentni model, a 1994. Referentni model distribuiranih transakcija.-Jedan od rezultata: XA interfejs koji definiše kako komuniciraju transakc. menadžer i menadžer resursa (DB).Transakcioni MW: TP monitori-Su u stvari transakcioni MW produkti i aplikacioni serveri prve generacije.-Obezbeđuju "lokaciju" za aplikativnu logiku osim mehanizma koji omogućava komunikaciju između dve ili više aplikacija.-Bazirani su na premisi transakcije, i obezbeđuju konektore niskog nivoa koji omogućavaju da se priključe na resurse (BP, druge aplikacije, que-ovi) .-Nepr evazideni su kada se zahteva podrška mnogim klijentima i visoko opterećenje transakcione obrade.-Koriste mehanizme: redova ulaznih bafera, kao zaštite od špirova opterećenja, planiranje prioriteta kako bi se dao prioritet porukama, podrška serverskim nitima kao i mehanizme balansiranja opterećenja.-TPM obezbeđuju svojstva: queing-a, rutiranja i slanja poruka, čime se omogućava zaoblazajenje TPM-a u slučaju potrebe.-Važno svojstvo balansiranja opterećenja omogućava da se prihvati nalet transakcija, npr. kod periodičnih obrada.-Primeri:-IBM: CICS (via COBOL) familija-Transarc/IBM: Encina ( TXSeries )-BEA Systems: Tuxedo -BEA/NCR: Top End-Microsoft: MTS (MS Transaction Services) ⇨OTM</p>	<p><b>119. OSF DCE arhitektura</b></p> 	<p><b>116. Srednji sloj (middleware) C/S sistema, namenam, položaji, servisi</b>  -MW je klasa s/w tehnologija koja ima zadatak da pomogne u savladavanju kompleksnosti i heterogenosti koji su inherentni distribuiranim sistemima.-Definiše se kao sloj s/w iznad OS a ispod aplikativnog programa, sloj koji obezbeđuje programsku apstrakciju zajedničku za ceo distribuirani sistem.-U žargonu, MW\ vodovod (plumbing)-Suštinski, MW je s/w koji čini distribuirani sistem programabilnim.-MW uvek maskira heterogenost mreža i hardvera.-Većina MW okvira takođe maskira heterogenost u pogledu OS i programskih jezika.-Neki maskiraju heterogenost između različitih proizvođačkih implementacija istog MW standarda (npr. CORBA, ili ODBC, JDBC) .-Aplikaciju je moguće povezati i direktno sa OS putem odgovarajućeg API-a (npr. putem socket-a) ali je to zahtevno i podložno greškama.-Network socket programiranje nije trivijalno.Namena MW, funkcije, položaj-MW je ključni element distribuiranih, heterogenih sistema.-Predstavljaju tehnologiju, razvijenu početkom '90-ih sa ciljem obezbeđenja interoperabilnosti u heterogenom okruženju.-Implementira se kao s/w koji se nalazi u sredini između aplikacije, osnovnog sistemskog s/w sistema (OS, DBMS, mrežne funkcije) .-Početkom '90-ih to je bio uglavnom DB MW za povezivanje klijenata sa BP.-Sredinom '90-ih su se pojavile i brojne druge kategorije (OO, poruke) .-Slični koncepti su se tada nazivali -mrežni operativni sistem (NOS)-distribuirani operativni sistem (DOS)-distribuirano računarsko okruženje (DCE)-Upotreba zajedničkog MW API-a "maskira" kompleksnost, kako entiteta koji se povezuju, tako i platformi na kojima se oni nalaze.-MW je inicijalno bio tip s/w koji omogućava komunikaciju između dva ili više sistema.-MW je sistemski s/w koji se nalazi između aplikacija i OS, mrežnih protokol stackova i hardvera, koji se nalaze ispod njih.-Zbog intenzivnih promena MW je teško kategorizovati.-Čini ga skup servisa koji ima zadatak da aplikativnog programera izoluje od detalja nižeg nivoa u sferi: OS, DBMS, komunikacionih protokola,...-MW obezbeđuje standardne interfejsje koji povećavaju interoperabilnost i portabilnost aplikacija.-Aplikativni programeri pristupaju ovim servisima putem MW API-a.</p>
<p><b>140. MS ADO.NET</b>  -ActivX Data Objekti (ADO) je objektni sloj preko OLE DB koji obezbeđuje pristup pod jezicima bez pointera (Visual Basic) i skript jezicima tipa Java Script i VB Script.-Interfejs podataka višeg nivoa ADO, ActivX Data Objects (DAO ranije) obezbeđuje pristup podacima za jezike i alate bez pointera (VB, VBscript, Java script)  <b>141. Arhitektura ADO.NET</b></p> 	<p><b>132. Srednji sloj za povezivanje sa BP (namenam, položaji i funkcije)</b>  -U osnovi to je MW orijentisan na BP, koji povezuje klijente sa severom BP.-Aplikacioni programi koriste DB MW kao mehanizam za unos/iznos podataka iz lokalnih ili udaljenih BP.-U ranim fazama razvoja c/s sistema sa DB serverom, navedeni MW se nazivao DBA (Data Base Access) sloj.Funkcije DB MW- Aplikacioni interfejs predstavlja interfejs prema aplikaciji.-Konverzija zahteva, jezik apl. se prevodi u nešto razumljivo (SQL) od strane ciljne BP.-Slanje zahteva, sposobnost da se preko mreže pošalje upit prema BP.-Obrada zahteva, sposobnost da se inicira obrada upita na ciljnoj bazi.-Přebacivanje rezultata, sposobnost da se rezultantni skup (rezultat upita) vrati nazad do aplikacije.-Konverzija rezultata, sposobnost da se rezultantni skup konvertuje u format razumljiv od strane aplikacije, klijenta, koja je postavila upit.<b>133. Načini povezivanja klijenta i servera</b>Tipovi DB MW:-Nativni (proprietary) DB MW-DB MW tipa zajedničkog interfejsa-DB MW tipa zajedničkog "gateway"-a-DB MW tipa zajedničkog protokola<b>134. Native/proizvođački DB MW</b>-Kreiran je za određeni tip BP (DBMS)(npr. Oracle, Sybase, DB/2, Informix)-Obezbeđuje najbolje performanse i pristup specifičnim (nativnim) svojstvima (npr. store procedure, trigger) .-Tipično su to API-i koje obezbeđuje proizvođač RDBMS-a, u obliku C i/ili C++ biblioteka.-DB MW u homogenom okruženju-Korišćeno DB MW rešenje ima više slojeva.-Interfejs klijenta-Protokol -Interfejs servera-Jedinstveni protokol: SQL FAP (Format and Protocol) <b>137. Implementacije tipa zajedničkog interfejsa</b>-Prvi korak, standardizacija zajedničkog SQL interfejsa (SQL API) koji će koristiti sve aplikacije.-Razlike u serverskoj strani će tretirati drajveri za različite BP.-Problemi: -koji SQL API uzeti za standard?-Da li ESQl ili CLI API?-I dalje trebaju višestruki drajveri-Potrebne višestruke ws za upravljanje (DBA), višestruki FAP-ovi-ESQl standardizovan (ISO SQL-92), dopuna za Javu (JSQl) deo SQL-3 standarda.-SQL CLI omogućava kreiranje i izvršavanje SQL instrukcija u run-time-u, u nekoliko varijanata:-SAG CLI-X/Open SQL. CLI-MS ODBC</p>	<p><b>127. Middleware tipa OMG CORBA</b>  -OMG konzorijum osnovan 1989 sa ciljem promocije oo tehnologije, više od 800 članova.-Kreira standarde, specifikacije (ali ne i s/w) koji omogućavaju interoperabilnost i portabilnost distribuiranih oo aplikacija-OMA: arhitektura upravljanja objektima, sadrži opšti okvir u obliku referentnog modela.  <b>128. Centralni deo CORBA arhitekture</b></p> 
<p>87. Fajl server  88. DB server  89. Transakcioni server  90. Objektni aplikacioni server  91. Grupni server  92. Web aplikacioni server  93. Komponente distribuiranih IS  94. Dvoslojna C/S arhitektura  95. Troslojna C/S arhitektura  96. Komparacija 2-slojne i 3-slojne arhitekture  97. Osnovna struktura monolitne aplikacije i alokacija funkcija na elemente C/S sistema  98. Moguća distribuirana rešenja partitioniranja aplikacije  99. Osnove C/S tehnologije  100. Tehnološki elementi C/S sistema  101. Generalizacija s/w arhitekture klijent-server sistema  102. Tehnologije servera (arhitektura s/w servera, h/w platforma)  103. OS servera  104. Bazni servisi OS  105. Prošireni servisi OS  106. Tehnologije klijenta (arhitektura sw klijenta, h/w platforma)  107. OS klijenta  108. Korisnički interfejs klijenta  109. Alati za razvoj C/S aplikacija  110. Struktura i elementi savremenog razvojnog alata  111. Izbor alata za razvoj C/S aplikacija  112. Integrisana razvojna okruženja (IDE)  113. Primeri proizvođačkih IDE-ova  114. Primeri otvorenih IDE-ova  115. MS .NET okvir (Framework)</p>	<p>29. Modeli podataka  30. SQL i programski interfejsi  31. Programski interfejsi BP  32. Ugnježdjeni API (ESQl API), funkcije, obrada  33. Obrada ugnježdenog SQL API-ja  34. DB interfejsi na nivou poziva (CLI API)  35. Komparacija ESQl API i CLI API  36. Tipovi (kategorije) IS, aplikacija i korisnika  37. Front-end alati baze podataka  38. Osnovi upravljanja procesima  39. Upravljanje transakcijama  40. Svojestvo ACID  41. Kratke i duge transakcije, definicija i komparacija  42. Upravljanje konkurentnim pristupom  43. Problemi konkurentnosti (transakcija)  44. Pesimistički pristup upravljanju konkurentnim radom  45. Nivoi zaključavanja objekata BP  46. Upravljanje transakcijama primenom TPM  47. Pojava i razrešavanje dead-lock-ova  48. Upravljanje sigurnošću baze podataka  49. Problemi sa transparentnošću procesa  50. Distribuirani sistemi (definicije, prednosti, nedostaci)  51. Distribuirana obrada, distribuirana baza i distribuirani DBMS  52. Tehnike distribucije podataka  53. Horizontalno i vertikalno partitioniranje podataka iz BP  54. Keširanje podataka  55. Ekstrakti i replika  56. Replika kod MySQL-a  57. Korišćenje replikacionih servera</p>	<p><b>143. Povezivanje putem zajedničkog protokola</b></p> 