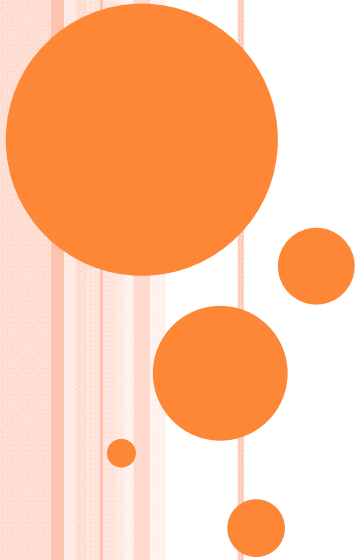


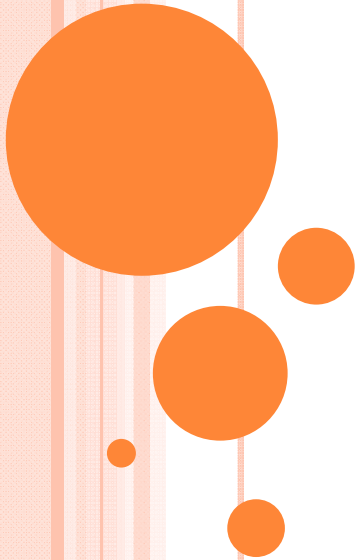
INTEGRAREA APLICATIILOR

Cursul 8

Agenda

I. Arhitecturi utilizate în integrarea aplicațiilor
   1. Arhitectura orientată pe servicii SOA
   2. Oracle Application Integration Architecture
   3. Enterprise Services Architecture

II. Tehnologii informatice de integrare a aplicațiilor - tehnologia middleware



I. Arhitecturi utilizate în integrarea aplicațiilor

I.1. SOA

- Valoarea reala SOA – cand servicii reutilizabile sunt combinate pentru a crea procese de afaceri agile si flexibile
- Acest lucru presupune ca serviciile:
  - Au dimensiuni, forma, functie si alte caracteristici similare
  - Sunt conforme cu standardele intreprinderii
  - Comunica la nivel tehnic
  - Comunica la nivel semantic
  - Nu au lacune si suprapuneri in responsabilitati

COMPONENTE ARHITECTURALE SOA

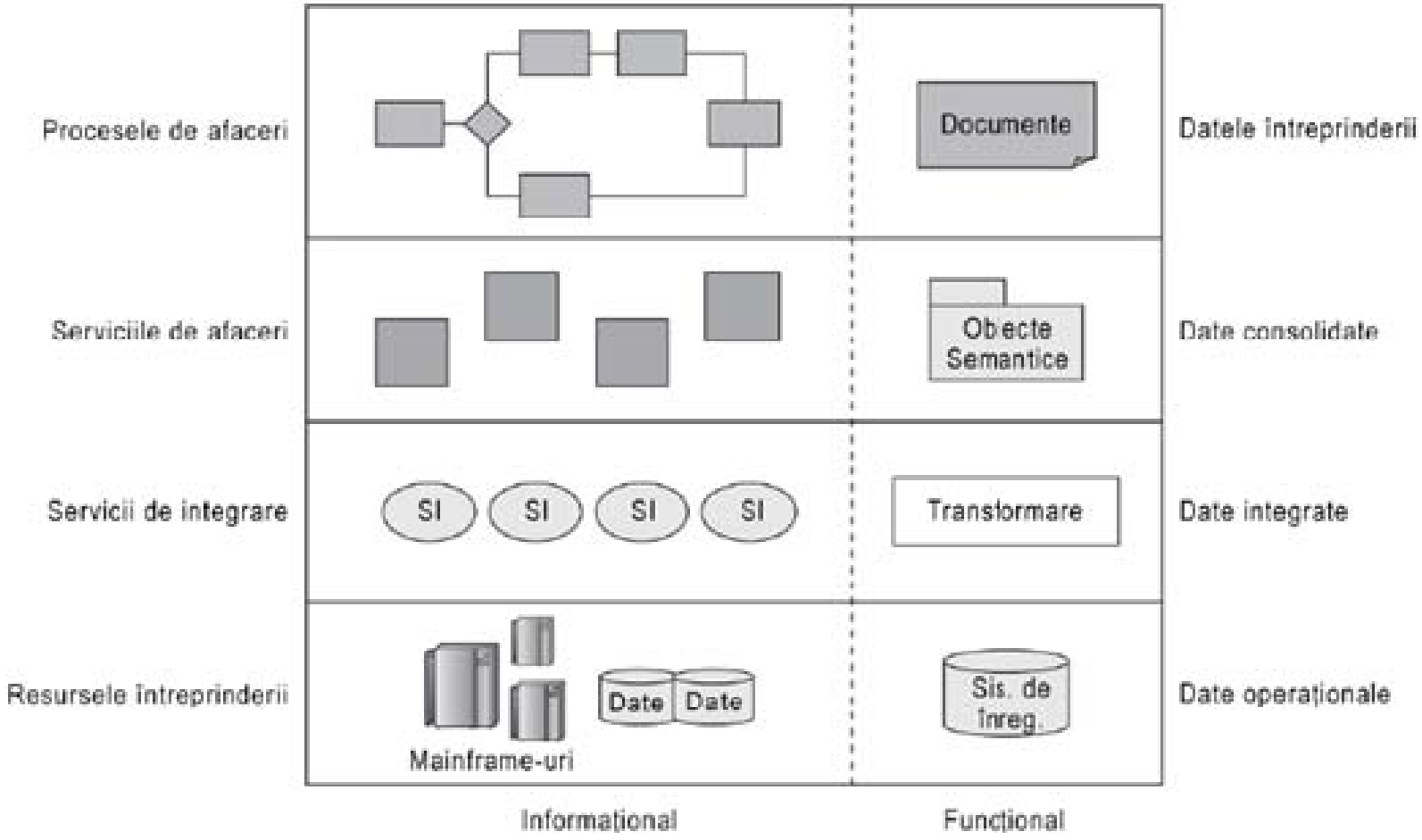


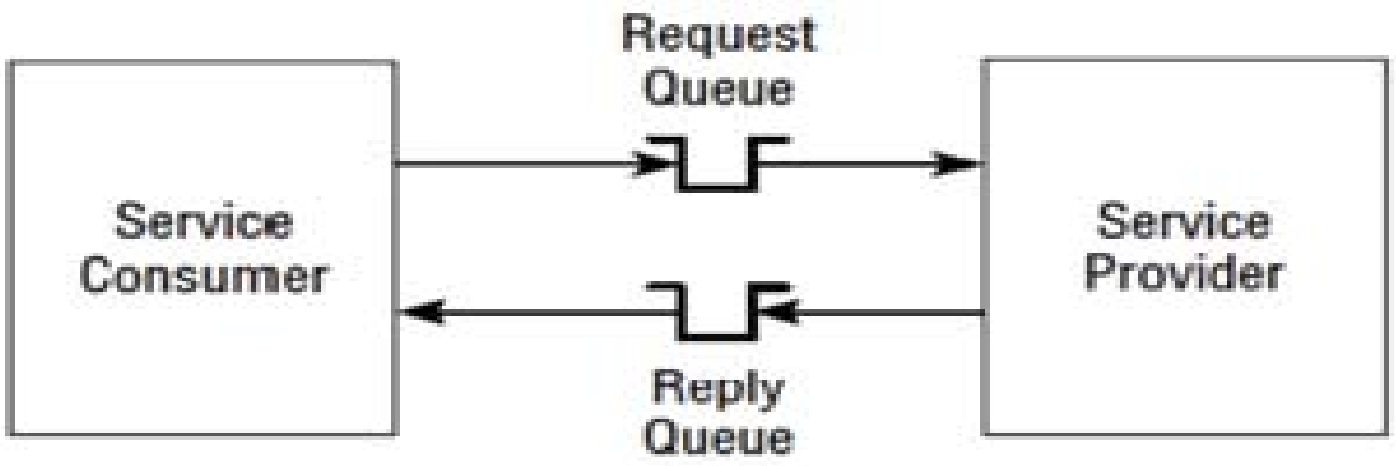
Figura 1.1. Componentele arhitecturale SOA

SOA

- poate fi implementat în multe moduri, folosind o varietate de instrumente şi tehnologii pentru integrarea aplicatiilor
  A. Infrastructura de mesagerie
  B. Broker de mesaje
  C. Servicii Web
  D. Adaptori JCA/J2C pentru integrare
  E. Impachetare prin servicii Web
  F. Acces direct la baze de date
  G. Enterprise Service Bus

A. Infrastructura de mesagerie

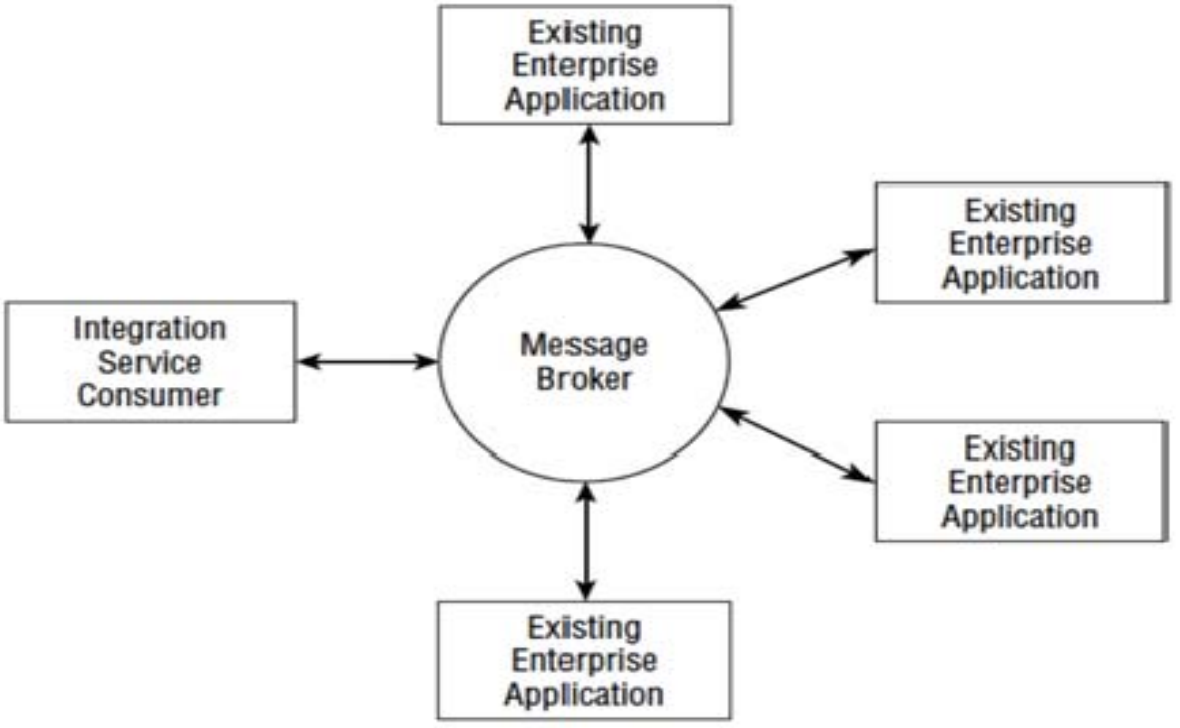
- Dc majoritatea aplicatiilor intreprinderii sunt conectate la o **infrastructura de mesagerie** (Java Message Service, WebSphere MQ, Rendezvous TIBCO), acest nivel middleware poate fi folosit pentru integrarea accesului.
- **Componenta de business (consumer)** va trimite o cerere la **Request Queue**, coada la care asculta **aplicatia (provider)**, iar raspunsul este trimis la **Reply Queue**, la care asculta componenta de business



Integrare folosind middleware orientat pe mesaje (MOM)

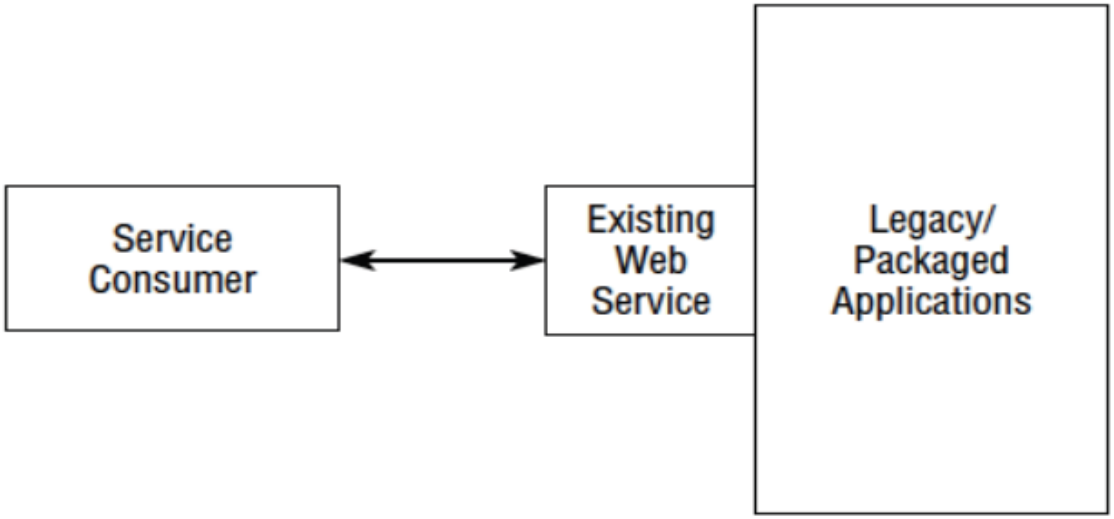
B. Broker de mesaje

- **Hub/centru de mesaje** – un **strat suplimentar** peste structura de mesagerie, care controleaza **centralizat** prelucrarea fluxurilor de mesaje
- Reduce impactul schimbarilor pe ambele sisteme, reducand cuplarea si costurile de integrare
- **Adaptoare** pentru conectarea aplicatiilor la broker
- Transformarea mesajului (EDI, FIXML)
- Mesaje rutate pe baza de continut



C. Servicii Web existente, oferite de aplicatiile companiei

- Sunt sustinute de toate platformele de executie
- Exista instrumente care ofera posibilitate de generare a unui **schelet pentru consum de servicii** bazat pe fisierul WSDL al furnizorului de servicii=>Implementarea unui serviciu consumator e mai facila

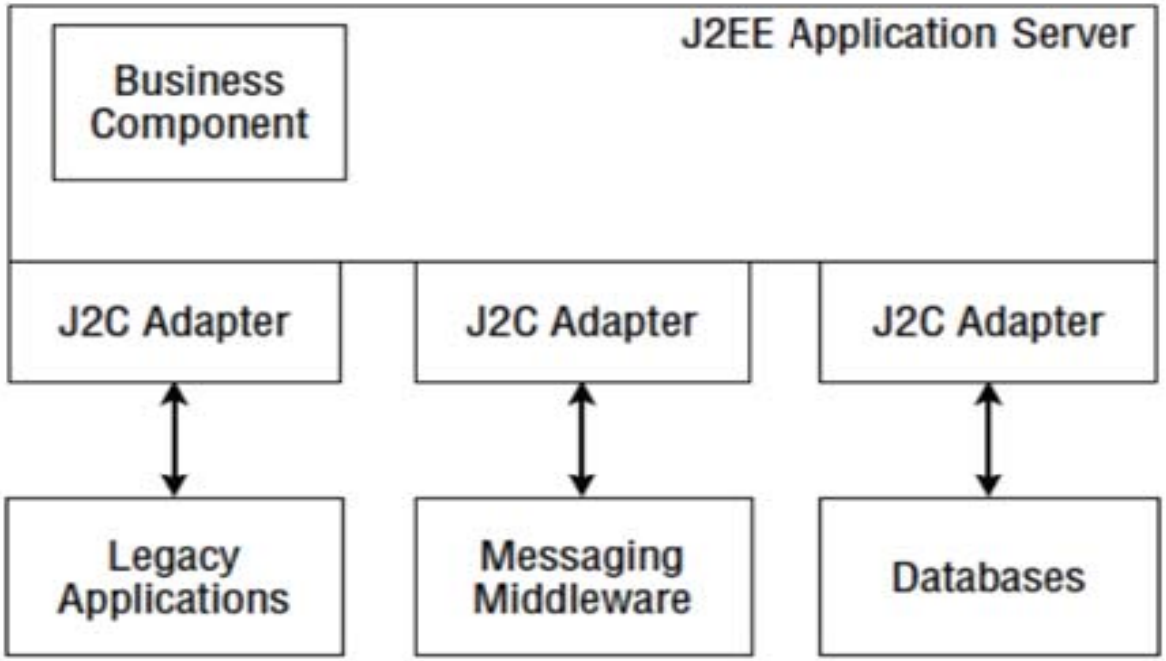


D. J2EE Connector Architecture

- **J2EE Connector Architecture J2C** sau **Java Connector Architecture JCA** = solutie standardizata pt integrare- 3 functii cheie:
  - **Common Client Interface** = API client uniform intre mai multe sist. infomatice
  - **Service Provider Interface** = defineste contracte la niv de sist. de gestionare a conexiunii, manag. tranzactiilor si de securitate intre un **server de aplicatii** si **adaptorul** folosit pt. o aplicatie a intrepr.
  - **Protocol de desfasurare si impachetare**
- Se recomanda cand implementarile serviciilor de business se bazeaza pe un server J2EE

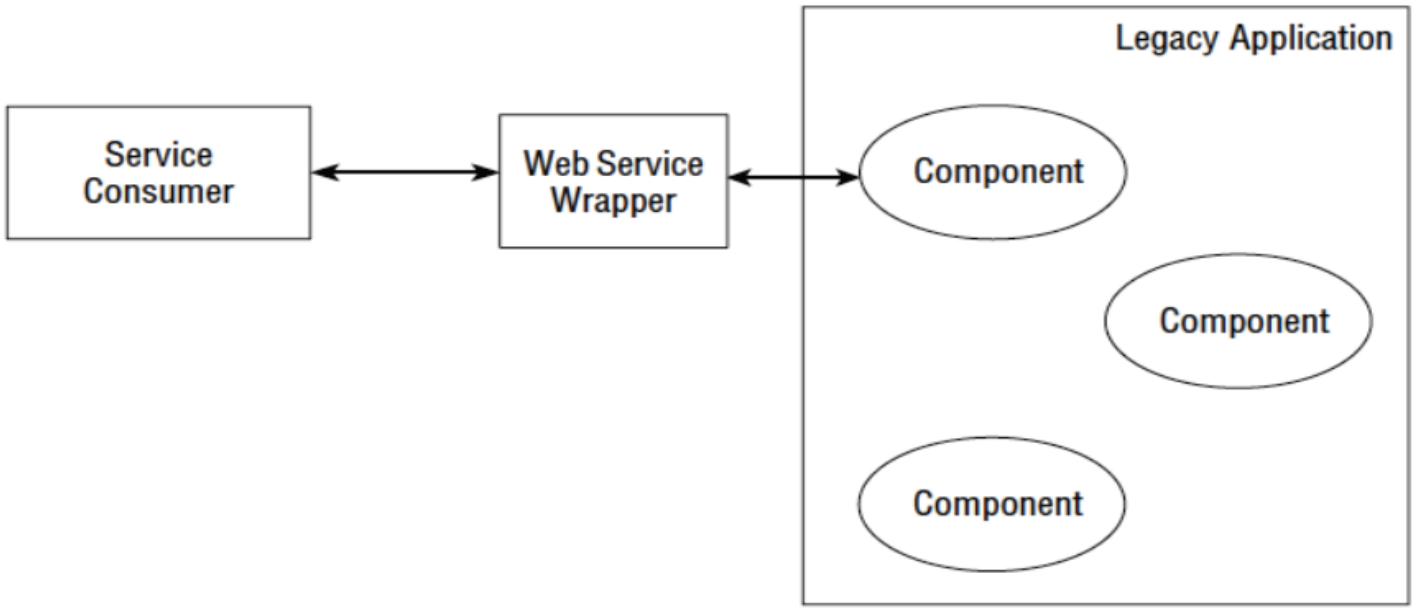
Adaptori JCA/J2C pentru integrare

- Adaptoare disponibile in prezent: pt ERP (SAP, Oracle, Baan, JD Edwards etc), SGBD (Oracle, DB2, SQL Server etc), sisteme de mesagerie (WebSphere MQ, Rendezvous TIBCO) etc



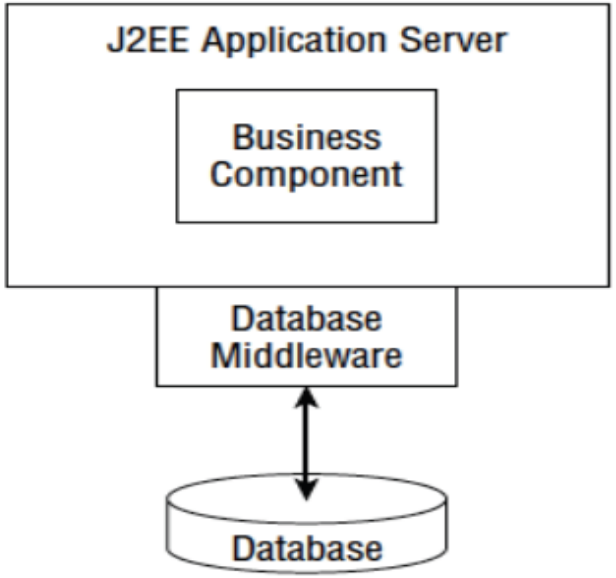
E. Impachetare prin servicii Web

- Servicul Web personalizat trebuie sa fie construit pe baza implementarilor existente
- Se presupune ca platformele de implementare ofera suport pt servicii Web
- Componentele care trebuie expuse vor fi incluse in interfata serviciului Web, iar componentele de business vor fi construite pe baza fisierelor WSDL



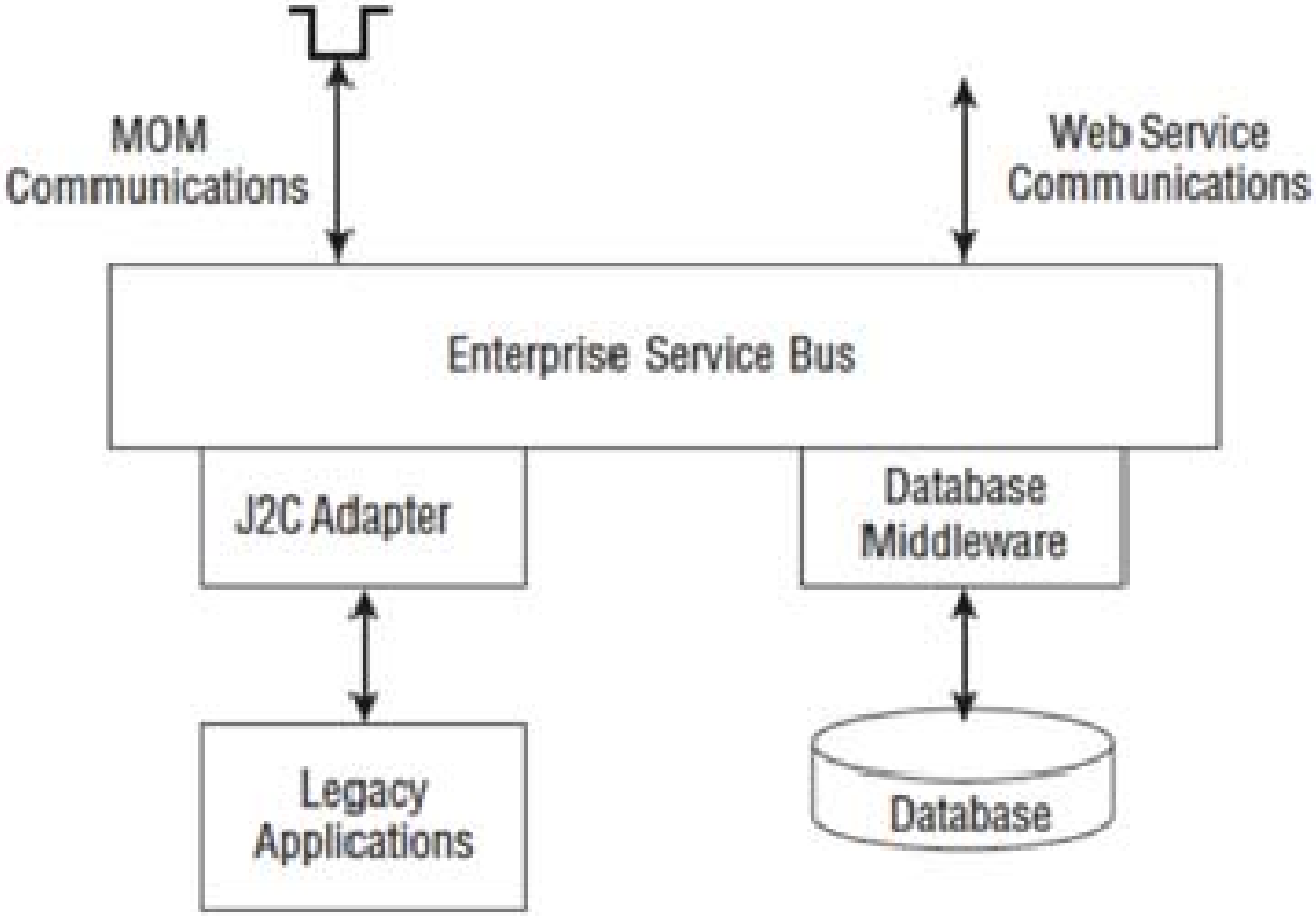
F. Acces direct la baze de date

- Standardizarea programelor de acces la BD, de ex. JDBC (Java Database Connectivity)
- JDBC se bazeaza de obicei pe J2C si functioneaza la fel: middleware-ul ofera acces la BD pt interogare si returneaza rezultatele



G. Enterprise Service Bus

- Suporta majoritatea mecanismelor de integrare prezentate mai sus, ofera suport pentru diverse tipuri de middleware
- Magistrala are o arhitectura deschisa, bazata pe standarde introduse de servicii Web, combinate cu produse traditionale EAI, evitand natura lor proprietara, complexitatealor sau inflexibilitatae furnizorulor
- Ofera capacitati de transformare
- Sustin dezvoltarea independenta a serviciilor

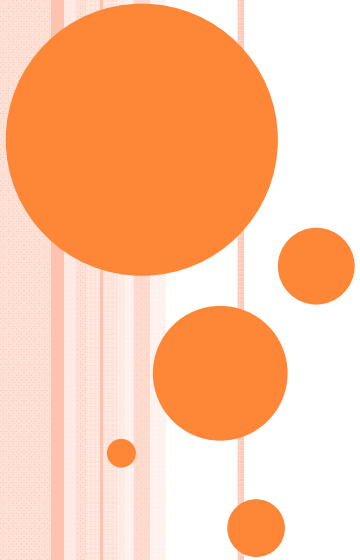

MOM
Communications
Web Service
Communications
Enterprise Service Bus
J2C Adapter
Database
Middleware
Legacy
Applications
Database

I.2. Oracle Application Integration Architecture

- solutie modularizata si bazata pe standarde cu rolul de a integra aplicatiile din cadul unei companii chiar daca aceste aplicatii sunt Oracle sau nu
- realizat pe baza SOA şi BPM Oracle Fusion Middleware.
- soluții prefabricate de integrare care reduce costul integrării sistemelor şi minimizează riscurile;
- cele mai bune practici de afaceri bine documentate sunt disponibile pentru integrare în aplicațiile existente;
- arhitectură deschisă, bazată pe standarde
- un depozit de servicii de afaceri (Business Service Repository) care permite dezvoltarea sau extinderea aplicațiilor.

I.3. Enterprise Services Architecture

- Interpretarea SAP pt arhitectura SOA, extinde conceptul de serviciu Web, înlocuindu-l cu conceptul propriu de **serviciu al întreprinderii (enterprise service).**
- Dpdv tehnic, serviciile de întreprindere se bazează pe servicii Web, oferind acces la elementele şi funcțiile afacerii în termeni economici.
- încapsulează funcționalitățile întreprinderii şi le expune ca servicii reutilizabile, care pot fi apoi combinate cu alte servicii pentru a răspunde unor cerințe noi.
- Organizațiile pot adopta această arhitectură prin implementarea platformei **SAP NetWeaver®.**



II. Tehnologia middleware

II. Tehnologia middleware

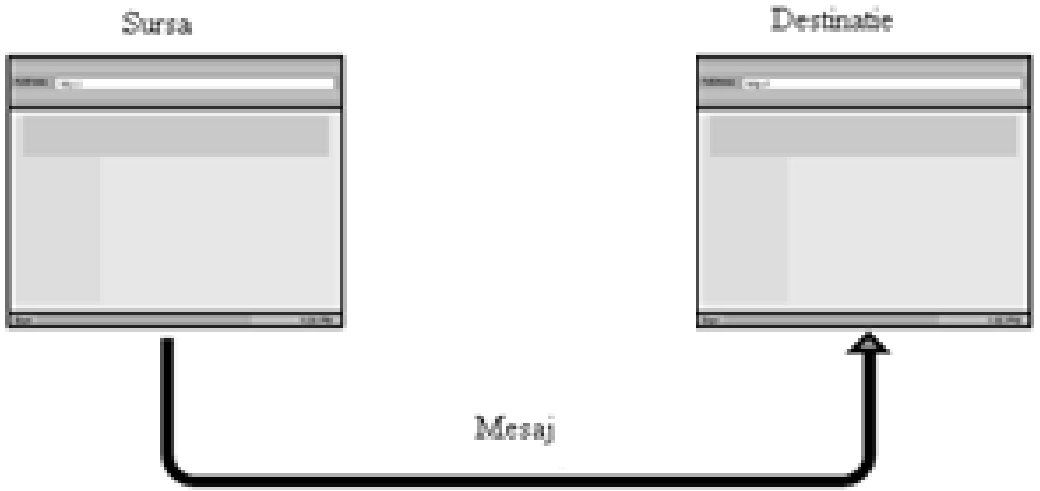
- Middleware este un mecanism care permite unei entități (bază de date sau aplicație) să comunice cu o altă entitate (sau cu mai multe entități).

MODELE DE MIDDLEWARE

- Există două modele de middleware: logic şi fizic.
- **Modelul logic** descrie cum are loc transferul de date conceptual. Configurațiile asociate modelului logic sunt:
  - unu la unu,
  - mulți la mulți şi
  - sincron versus asincron.
- **Modelul fizic** descrie metodele precum şi tehnologia folosite pentru transferul de informație. Modelului fizic îi sunt asociate modele bazate pe mesaje.

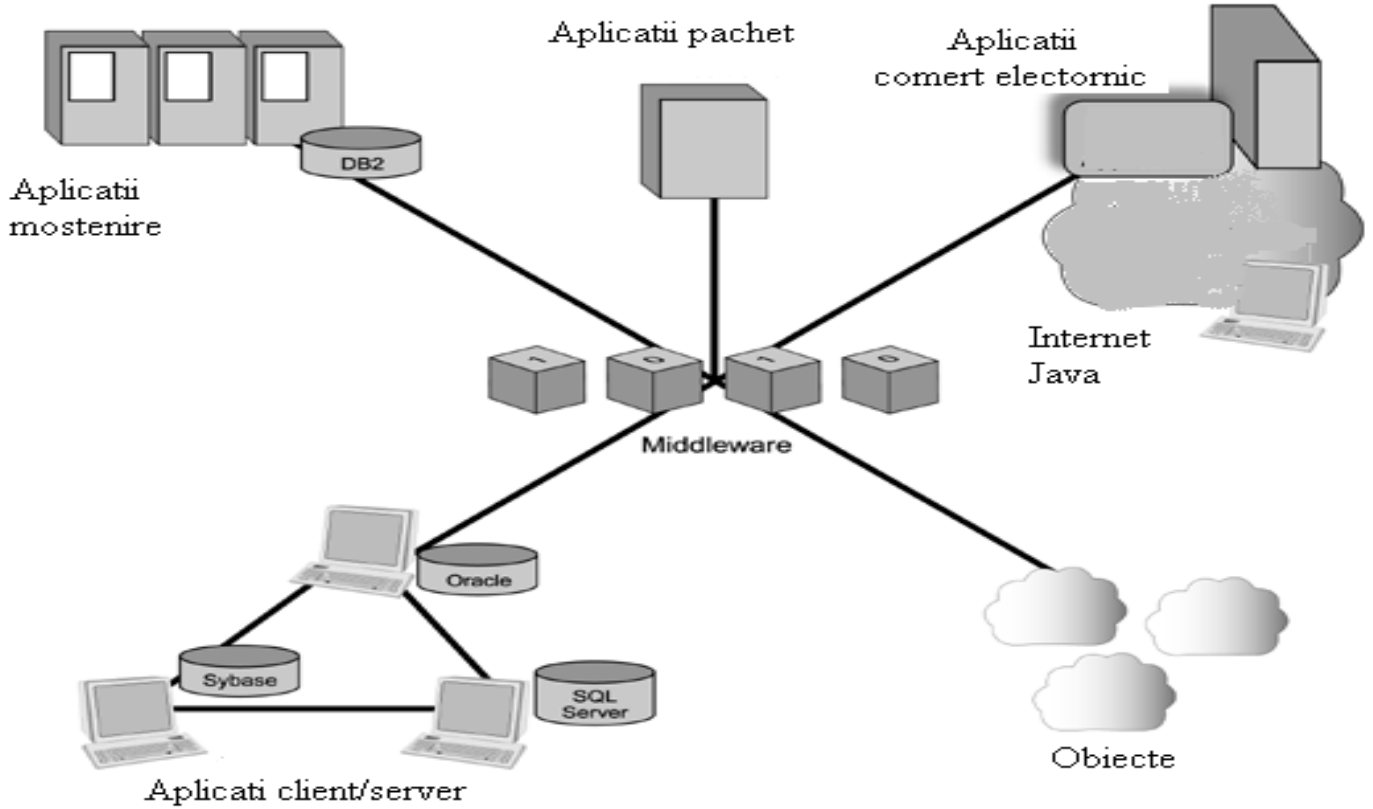
MIDDLEWARE UNU LA UNU

- foloseşte o conexiune software temporară între două programe sau comenzi (pipe) pentru a permite unei aplicații să acceseze altă aplicație.
- considerăm, spre exemplu, două aplicații A şi B. Când aplicația A încearcă să comunice cu aplicația B închide conexiunea folosind un **apel de procedura sau un mesaj**



MIDDLEWARE MULTI LA MULTI

- leagă mai multe aplicații între ele.



- cel mai puternic middleware logic, oferă atât flexibilitate cât şi adaptabilitate problemei integrării.
- Exemple : integrare la nivel de servere, middleware tranzacțional (servere aplicații şi monitori TP) şi chiar obiecte distribuite.

Middleware-ul Asincron

- **transferul de informație** între mai multe aplicații în mod **asincron**, ceea ce presupune că software-ul middleware se poate deconecta de la aplicația sursă sau destinație.
- aplicațiile **nu sunt dependente** de alte aplicații conectate pentru procesare.
- procesul care permite acest lucru are **aplicațiile plasate într-o coadă de aşteptare**, fiecare cu un mesaj asociat şi fiecare rulează independent, răspunsul de la celelalte aplicații primindu-se mai târziu.
- Avantajul principal este acela că middleware-ul **nu blochează** celelalte aplicații din procesare.

MIDDLEWARE-UL SINCRON

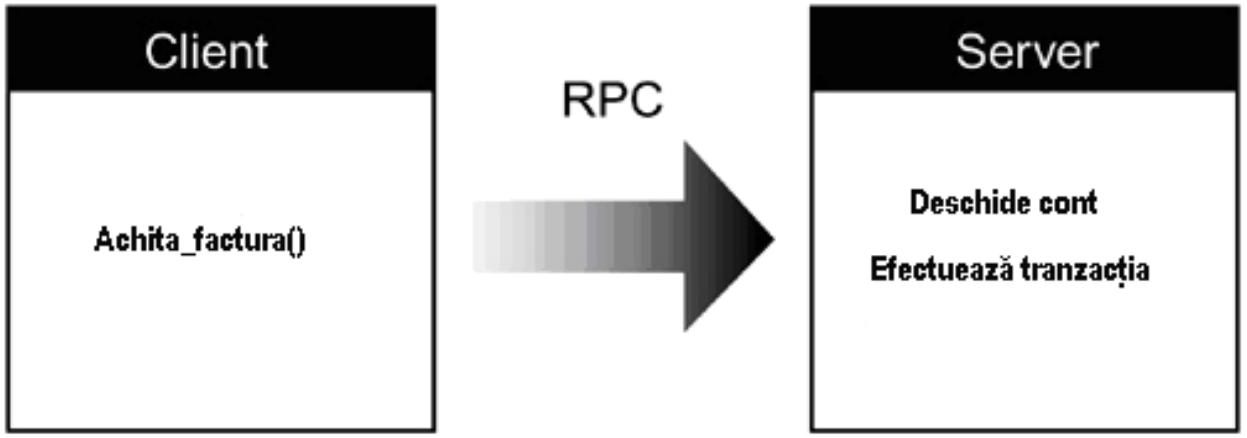
- este strâns cuplat la aplicații.
- aplicațiile sunt **dependente de middleware** pentru a procesa unul sau mai multe apeluri funcție pe o aplicație la distanță.
- aplicația care apelează trebuie să oprească procesarea pentru a **aştepta răspunsul aplicației** aflate la distanță.
- Dezavantajul acestui model este **cuplarea aplicațiilor la middleware** şi la aplicația la distanță.

TIPURI DE MIDDLEWARE

a. RPC,
b. MOM,
c. obiecte distribuite,
d. middleware orientat pe baza de date,
e. middleware tranzacțional (include monitori TP şi servere de aplicații)
f. servere de integrare

A. Remote Procedure Calls (RPC)

- oferă dezvoltatorilor capacitatea de a apela o funcție dintr-un program şi de a o executa într-un alt program, pe o maşină la distanță
- RPC sunt modele sincron de middleware. Pentru ca RPC să fie activat, execuția programului trebuie oprită.
- nu au o performanță foarte bună într-o rețea înceată, cum este Internetul.



B. *MIDDLEWARE-UL ORIENTAT PE MESAJE (MOM)*

- un software care foloseşte ca mecanism de transfer mesajele, unități de informație care sunt interschimbate de aplicații
- se bazează pe modelul **asincron**: mesajul intră într-o coadă de mesaje, iar managerul cozii hotărăşte când este trimis mesajul la destinația finală. Mesajele care se întorc la aplicația apelantă vor fi prelucrate când aplicația va fi disponibilă.
- Există **două tipuri de modele MOM**:
  - unu la unu şi
  - coadă de mesaje (Message Queue - MQ).
- Deoarece software-ul MQ (seria MQ de la IBM, MSMQ de la Microsoft) gestionează distribuția de mesaje de la un program la altul, managerul cozii poate optimiza performanța folosind **metode de acordare a priorității**.
- MQ permite ca mesajele să fie declarate drept persistente sau stocate pe disc la anumite intervale de timp

C. *OBIECTELE DISTRIBUITE*

- **Obiectele distribuite** sunt programe-aplicații de mici dimensiuni care **folosesc interfețe şi protocoale standard** pentru comunicare.
- De exemplu, dacă se creează **un obiect distribuit CORBA** care rulează pe un **sever UNIX** şi un altul care rulează pe un **server NT**, folosind un **protocol standard de comunicație**, obiectele pot face interschimb de informație şi funcții (acelaşi standard – CORBA, acelaşi protocol – IIOP (Internet InterORB Protocol)).
- Există **două tipuri de obiecte distribuite**:
  - **CORBA** este creat de OMG în 1991 şi este mai mult un standard decât o tehnologie oferind specificații pentru crearea unui obiect distribuit.
  - **COM** (Component Object Model) este creat de Microsoft şi include interfețe standard şi protocoale de comunicație.

D. *MIDDLEWARE-UL ORIENTAT PE BAZA DE DATE*

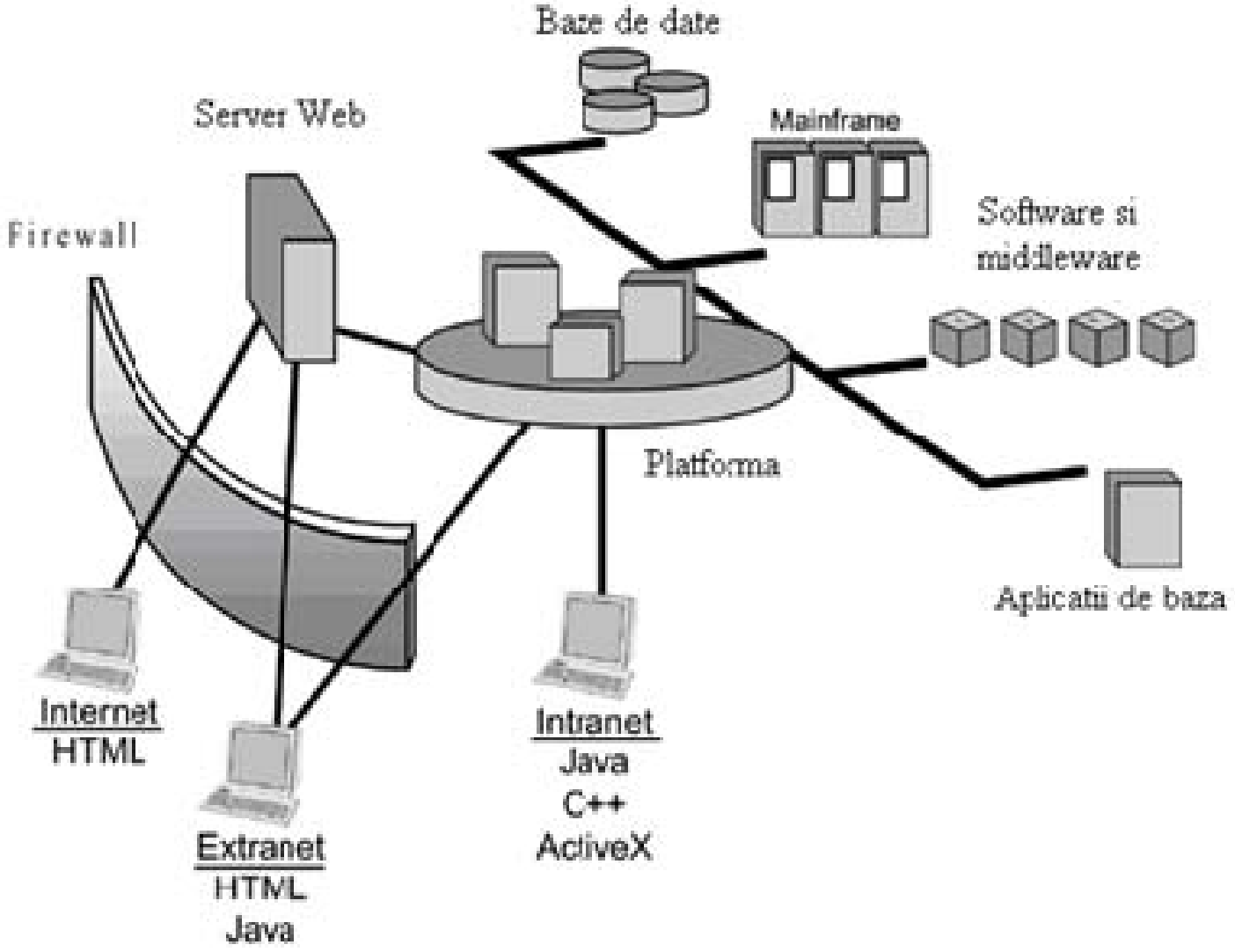
- Orice middleware care facilitează comunicația fie cu o BD, fie cu o aplicație, fie între mai multe BD.
- E un mecanism de extragere a informației dintr-o BD locală sau la distanță (autentificarea, cererea de informație şi procesarea informației care a fost extrasă din BD)
- Două tipuri :
  - **CLI (Call Level Interface) = API-uri comune,** care oferă acces la BD folosind o interfață bine definită. Ex: **Open DataBase Connectivity (ODBC)** de la Microsoft.- foloseşte o interfață pentru a facilita accesul la BD şi drivere pentru a gestiona diferențele dintre bazele de date. Oferă acces simultan la BD multiple, printr-un **driver manager** care să faciliteze comunicația între diferite BD
  - **Java DataBase Connectivity (JDBC) e** alt exemplu de CLI. JDBC este o interfață standard care foloseşte un singur set de metode Java pentru a facilita accesul la baze de date multiple. JDBC se aseamănă cu ODBC şi **funcționează de pe orice aplicație Java**: applet, servlet, Java Server Pages (JSP), Enterprise JavaBean (EJB).
  - **middleware nativ pe baza de date.**

E. MIDDLEWARE TRANZACTIONAL – MONITORII TP

- prima generație de servere de aplicații
- oferă un mecanism pentru a facilita comunicarea între două sau mai multe aplicații, precum şi o **locație pentru logica aplicației**
- se bazează pe **tranzacții**, văzute ca unități de lucru cu un început şi un final. O tranzacție are doar două stări: poate sa fie ori **finalizată**, ori **anulată complet**.
- oferă **servicii** care garantează **integritatea tranzacțiilor** (serviciul de tranzacție), iar pe de altă parte, un monitor TP asigură **managementul resurselor** şi servicii de management pe termen lung
- oferă **conectori la resurse** ca ***baze de date***, alte ***aplicații*** sau ***cozi***. Aceşti conectori necesită o dezvoltare de aplicație sofisticată pentru a comunica cu tipurile variate de resurse. Odată conectate, aceste tipuri de resurse sunt integrate în tranzacții.

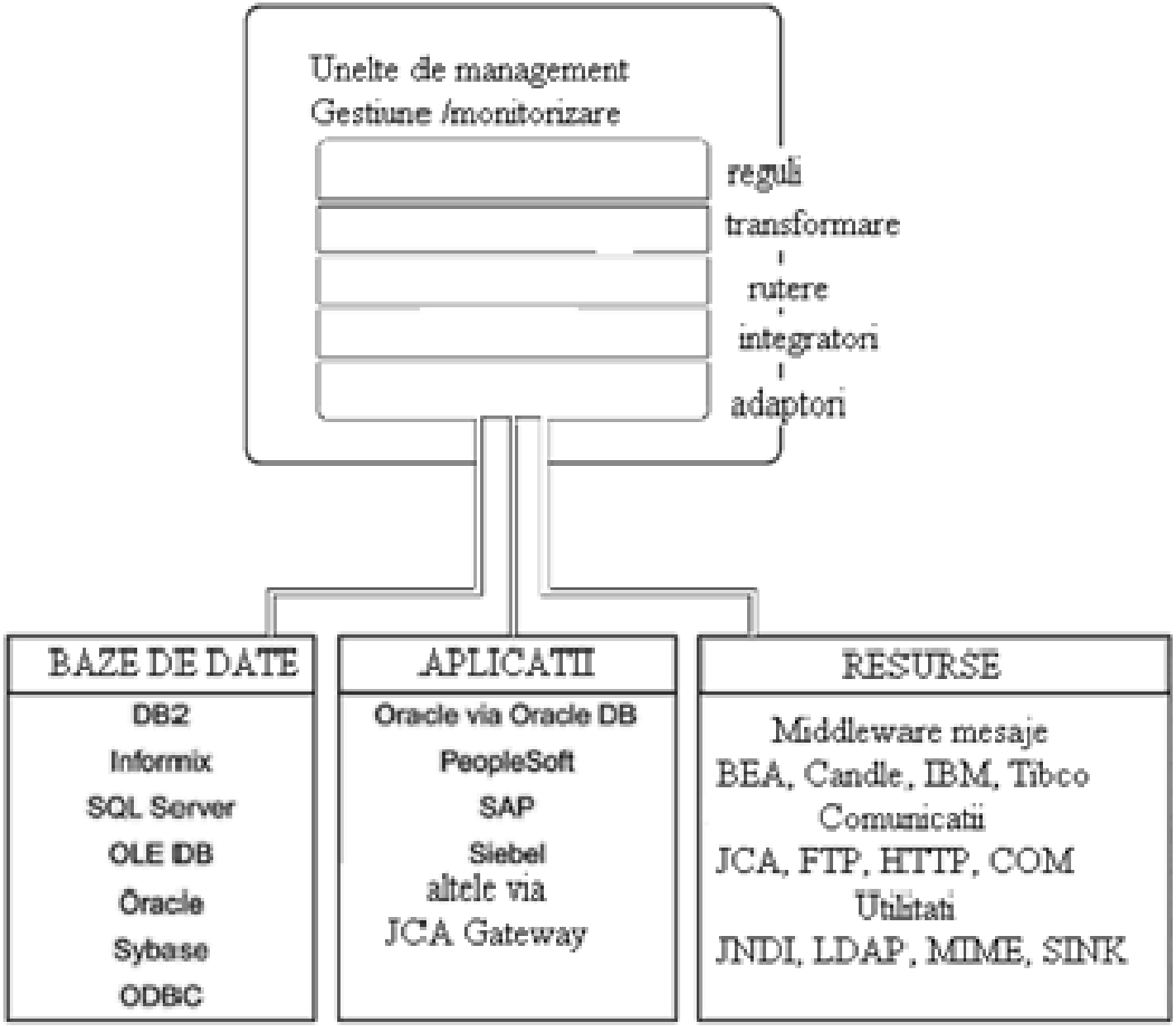
E. MIDDLEWARE TRANZACTIONAL - *SERVERELE DE APLICAȚIE*

- Majoritatea sunt **middleware Web** şi procesează tranzacții aparținând aplicațiilor Web.
- folosesc limbaje moderne, cum este Java, în locul celor tradiționale
- asigură posibilitatea accesului la alte aplicații şi fac posibilă procesarea şi stabilirea **resurselor** necesare conexiunilor: baze de date, aplicații ERP şi chiar aplicații tradiționale de tip mainframe.
- Serverele de aplicații oferă mecanisme de dezvoltare a interfeței utilizator şi mecanisme de amplasare a aplicațiilor pe platforma Web
- Producătorii de servere de aplicații consideră că produsele lor au o tehnologie ce permite rezolvarea problemelor de integrare a aplicațiilor


Baze de date
Server Web
Mainframe
Firewall
Software si
middleware
Platforma
Aplicatii de baza
Internet
HTML
Extranet
HTML
Java
Intranet
Java
C++
ActiveX

F. *SERVERELE DE INTEGRARE*

- partea de vârf a tehnologiei middleware pentru integrarea aplicațiilor
- pot facilita schimbul de informație între două sau mai multe aplicații sursă sau destinație şi pot face diferența între semanticile aplicației şi platforme.
- Serverele de integrare pot apărea în multe aplicații folosind reguli comune şi motoare de rutare.
- Ele pot **transforma schema şi conținutul informației** pe durata transferului între aplicații şi baze de date.
- gestionează mesaje între două sau mai multe aplicații sursă sau destinație.
- pot avea într-adevăr funcții adiționale, incluzând **un motor şi o interfață de integrare a proceselor**, precum şi **un mecanism de management.**
- **nu** sunt o tehnologie de **dezvoltare** a aplicațiilor ci mai degrabă una care permite **comunicarea** între mai multe aplicații


Unelte de management
Gestiune /monitorizare
reguli
transformare
rutere
integratori
adaptori
BAZE DE DATE
DB2
Informix
SQL Server
OLE DB
Oracle
Sybase
ODBC
APLICATII
Oracle via Oracle DB
PeopleSoft
SAP
Siebel
altele via
JCA Gateway
RESURSE
Middleware mesaje
BEA, Candle, IBM, Tibco
Comunicatii
JCA, FTP, HTTP, COM
Utilitati
JNDI, LDAP, MIME, SINK