

Centralised computing versus Distributed Computing

Sisteme Distribuite

Mihai Zaharia

Bucătărie

- proful --->
- Regulile de evaluare --> ca și anul trecut
- Examenul --> vezi mai jos

Cum se utilizează cursul ?????

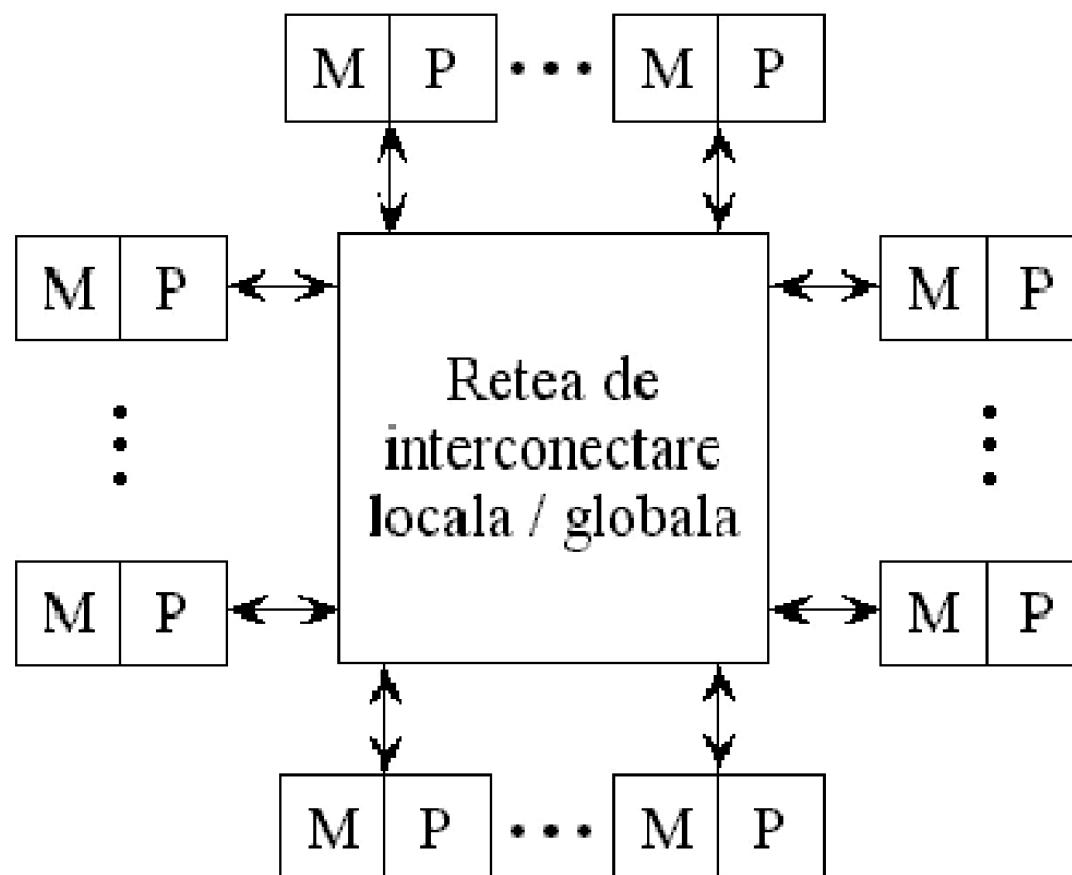
1. se iau notițe! (chiar dacă te doare mâna, neuronul etc)
2. nu este permisă înregistrarea sub nici o formă a titularului
- oricare ar fi el - (nu pentru că se supără ci pentru că se încalcă simultan trei legi în vigoare!)
3. te duci acasă și citești despre termenii din notițe și slide-uri până te lămurești apoi te uiți din nou pe slide-uri.
4. dacă mai rămân după atâta citit unii termeni mai neclari există profesor și asistenți
5. prezentarea la laborator fără efectuarea pașilor anteriori anulează 80% din efectul combinat curs+lab asupra pregătirii voastre!

Termeni si concepte

sisteme strâns cuplate

sisteme slab cuplate

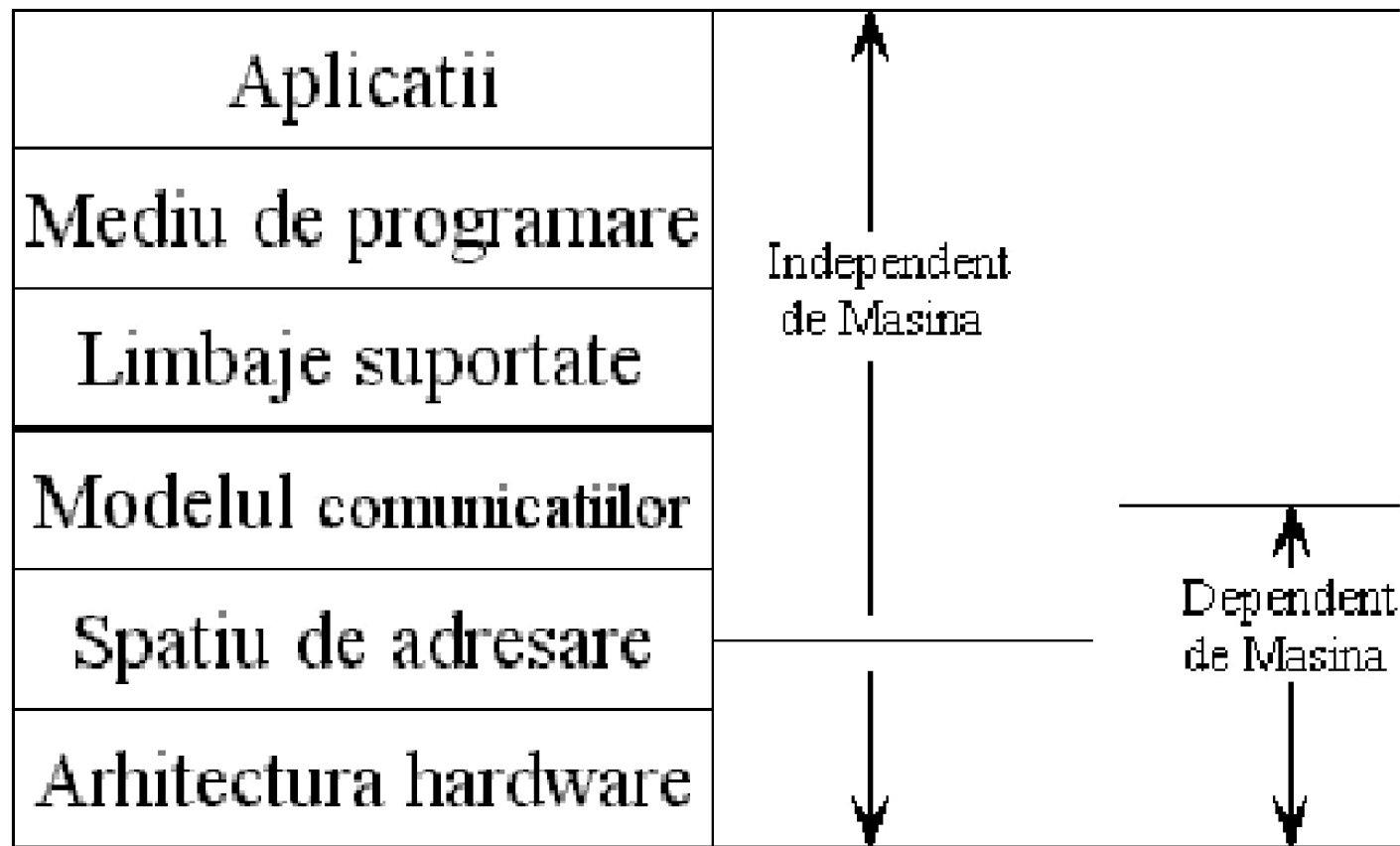
Modelul sistemelor HW slab cuplate



M = Memorie; P = Procesor

Multicalculatoare cu memorie distribuită

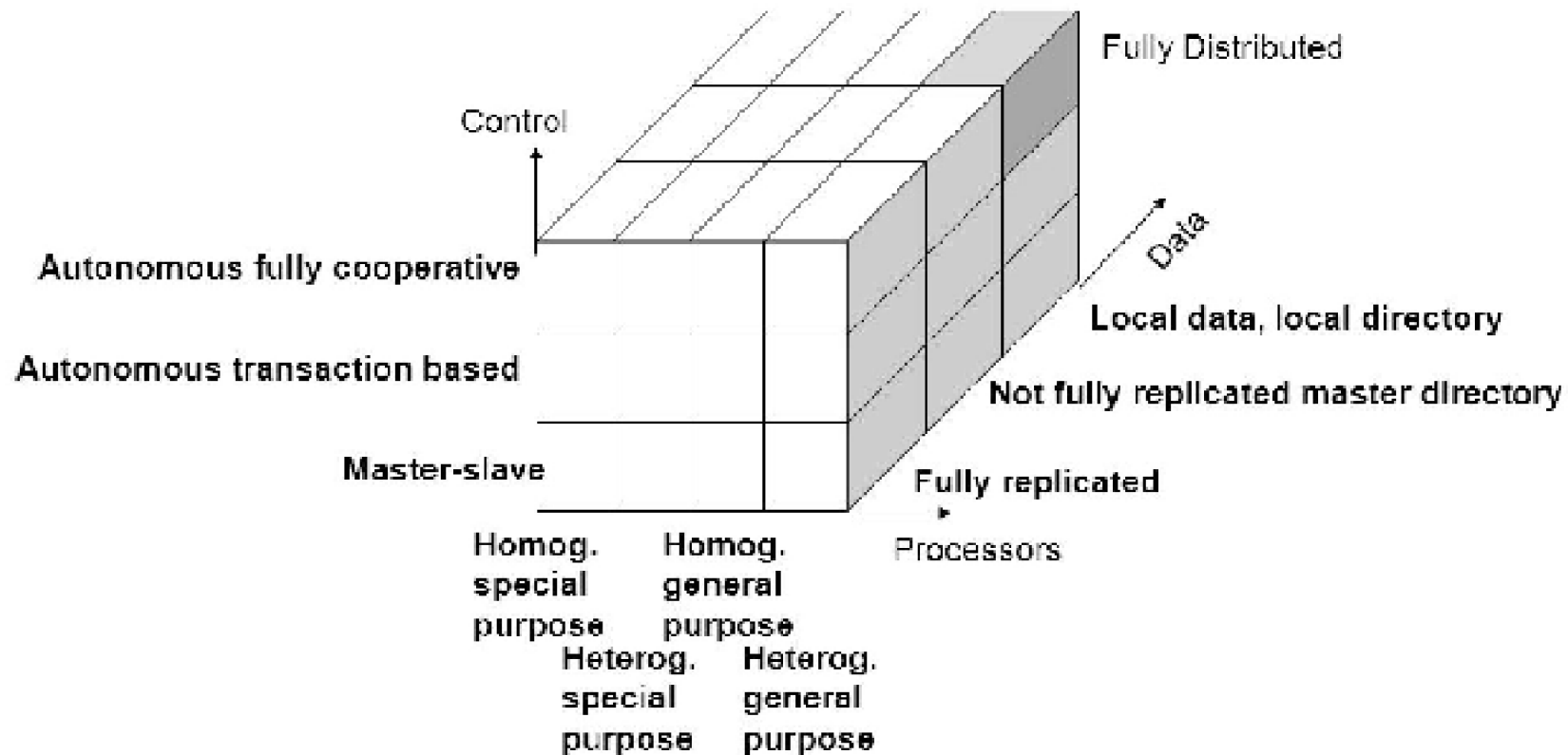
Modelul Li al unui sistem distribuit(hw+sw)



Termeni și concepte

arhitecturi software strâns cuplate - monolit	arhitecturi software slab cuplate - servicii/microservicii
--	---

Sistem distribuit Enslow 1978

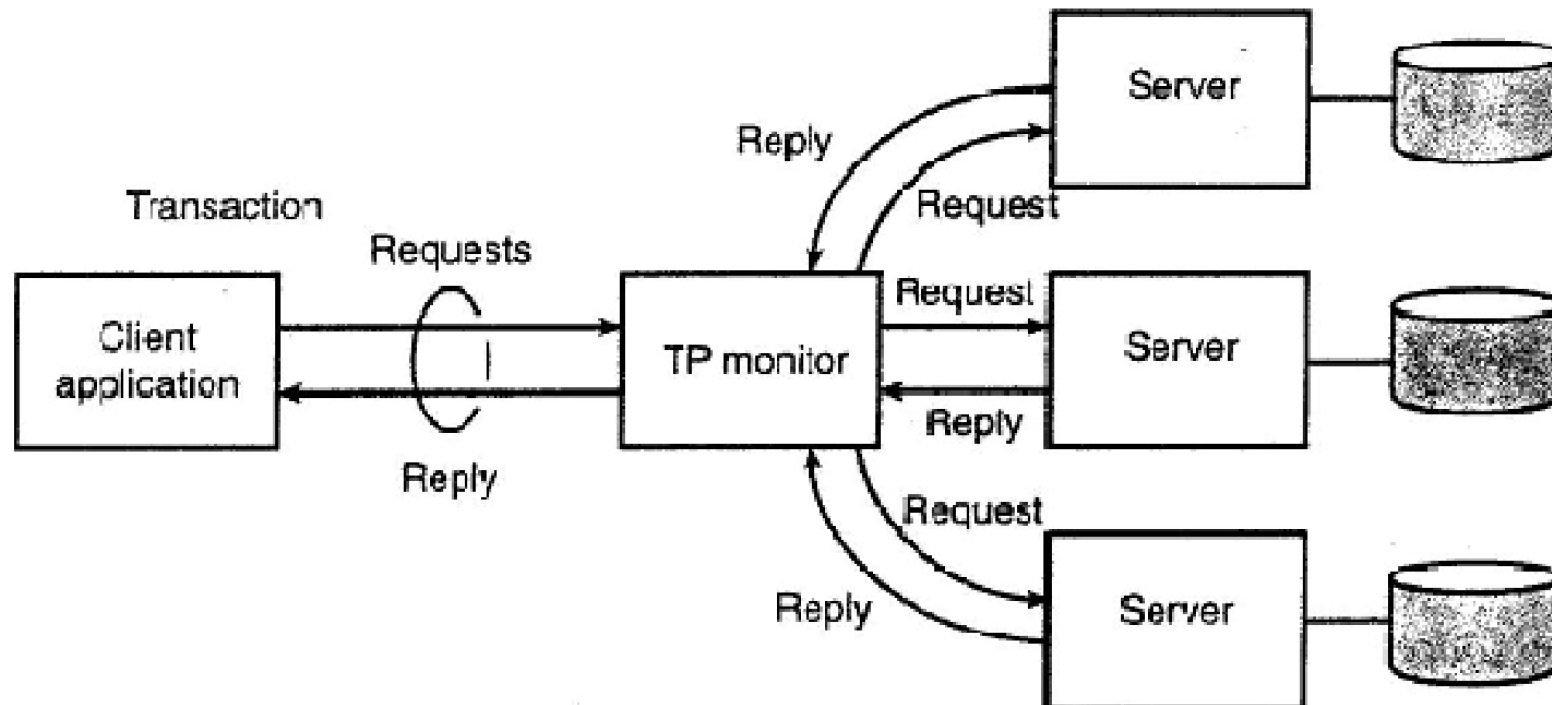


Hardware distribuit fizic/geografic

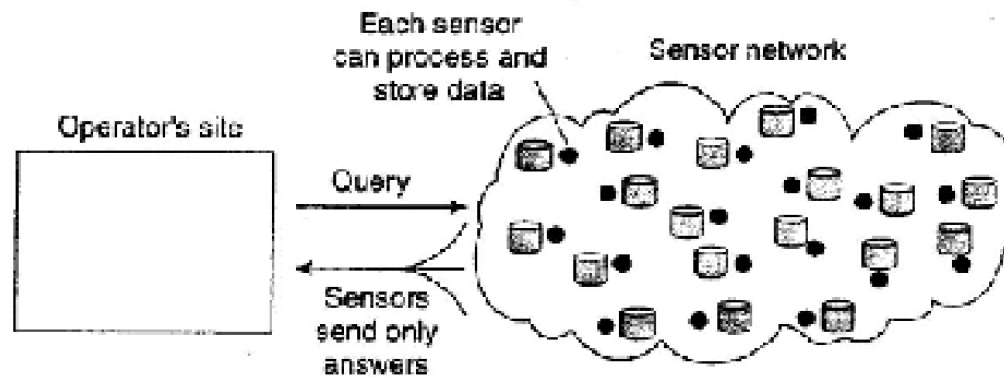
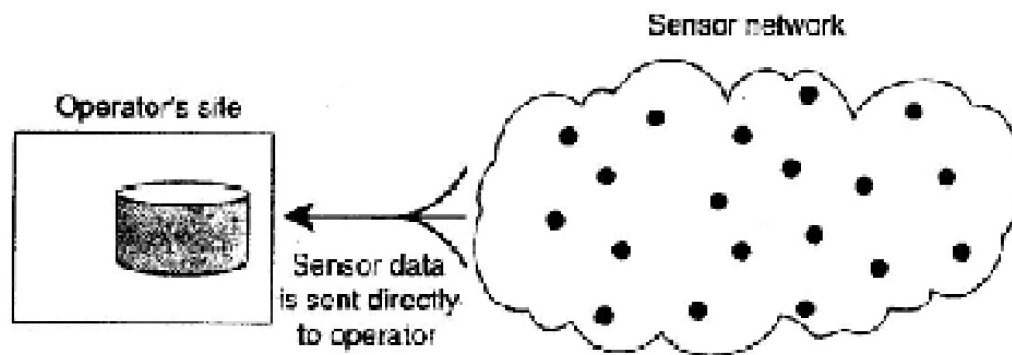
Control Distribuit

Datele distribuite

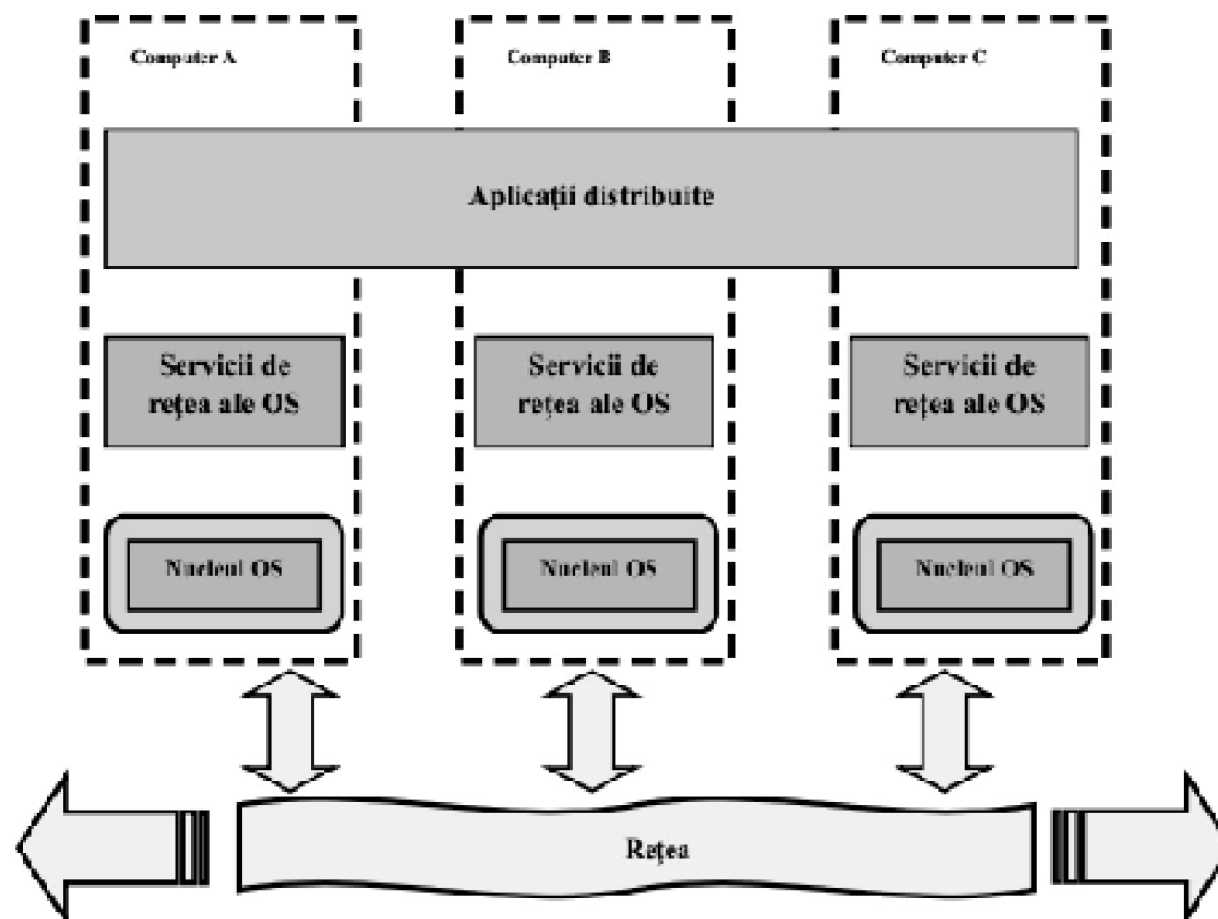
Example



Example

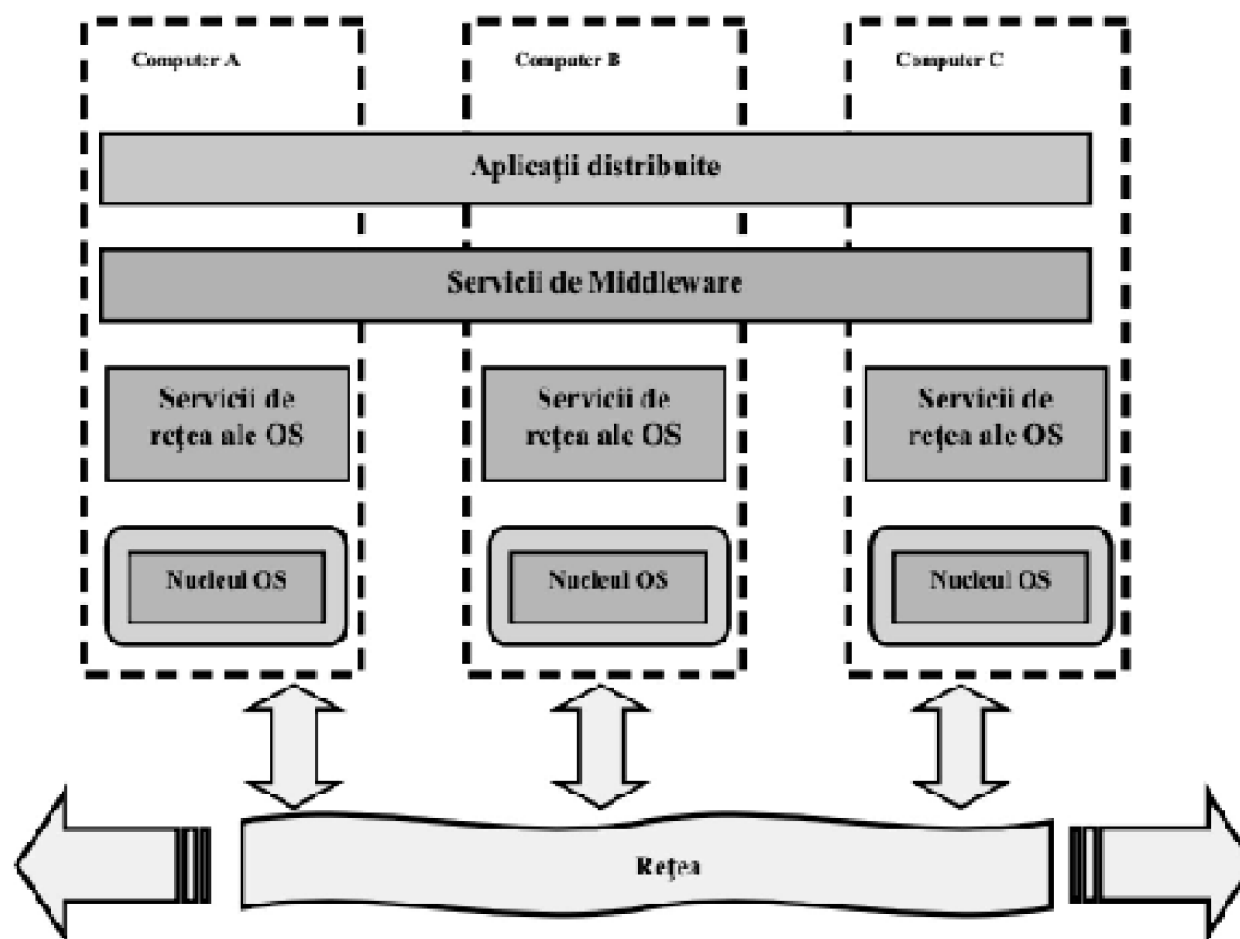


Ce suport ne oferă sistemele de operare?



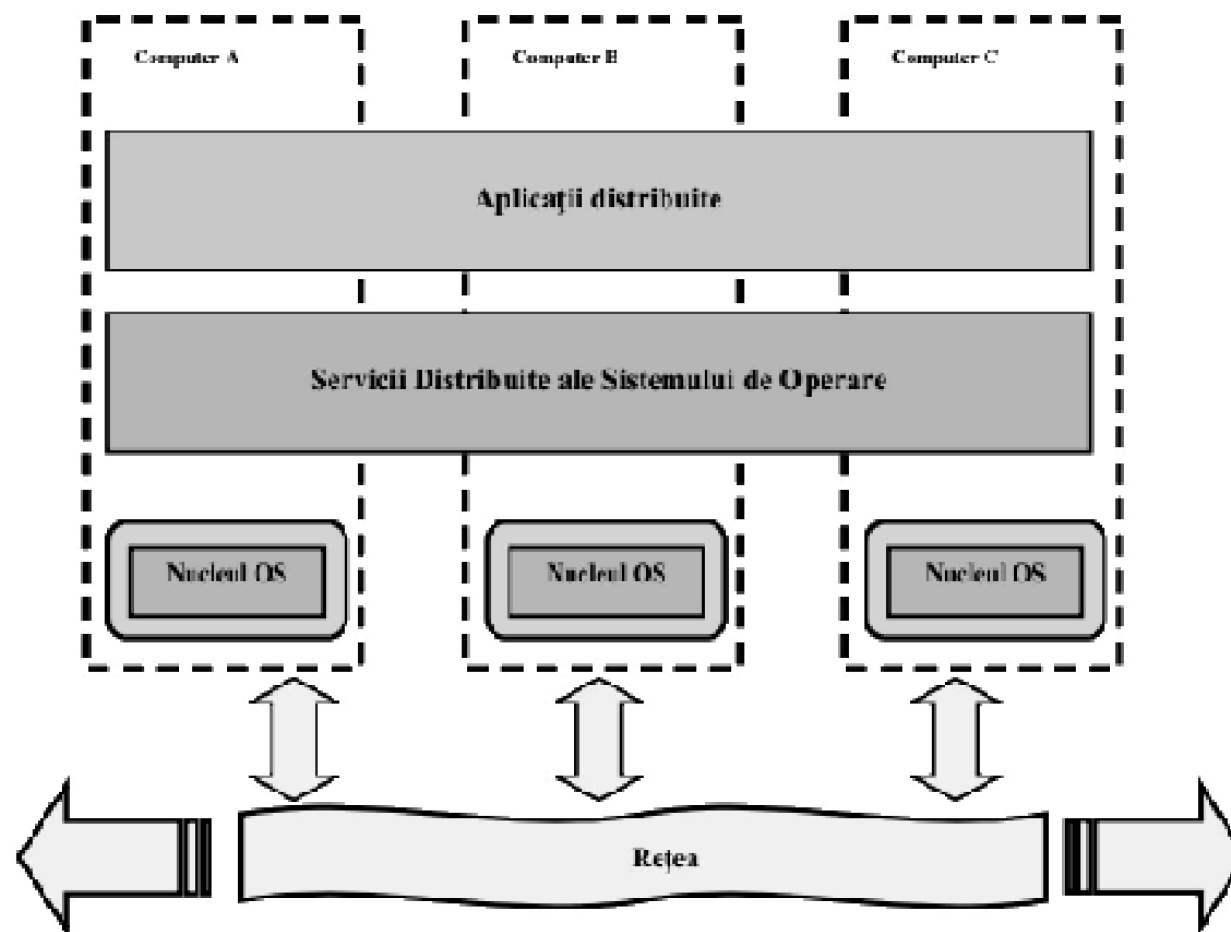
Structura generală a unui sistem de operare în rețea

Ce suport ne oferă sistemele de operare?



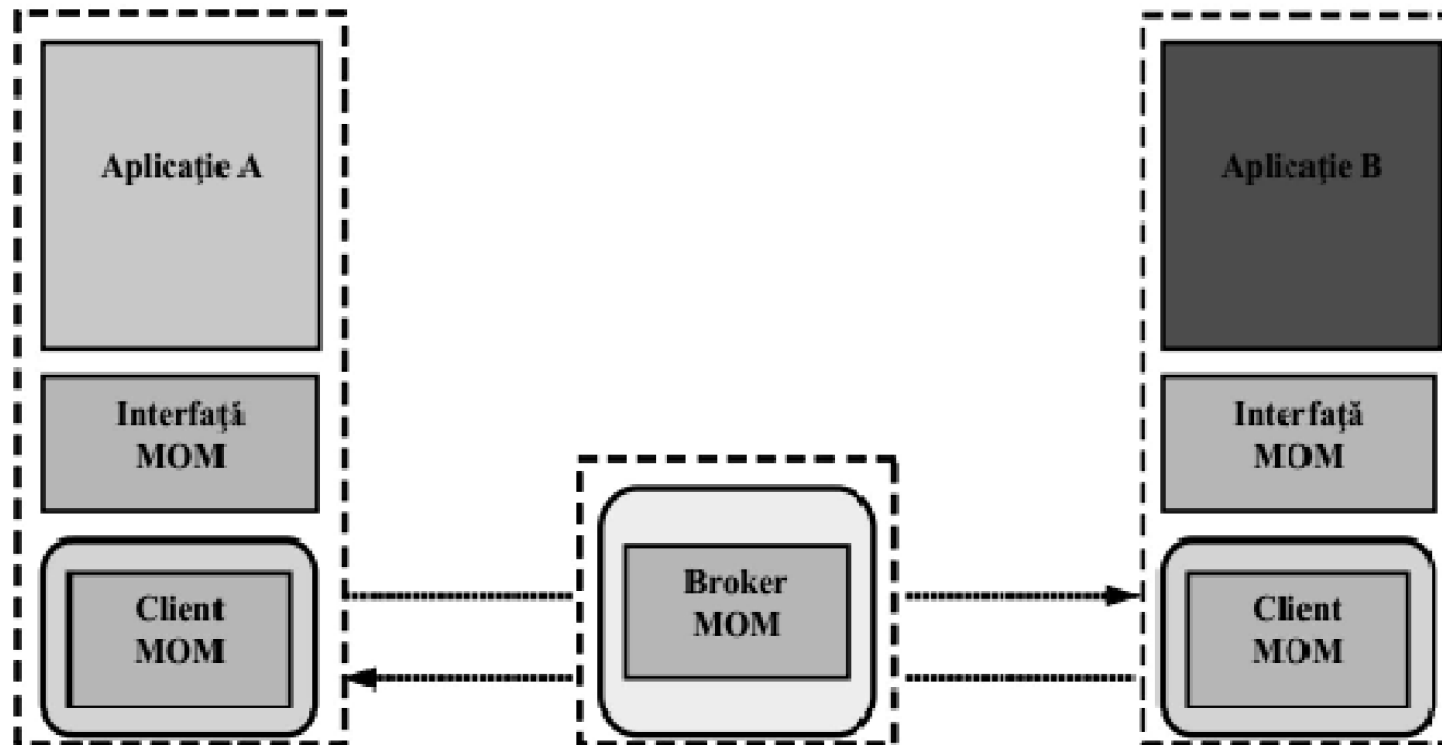
Structura generală a unui sistem distribuit ca și Middleware

Ce suport ne oferă sistemele de operare?



Structura de ansamblu a unui sistem de operare multicomputer

Ce suport ne oferă sistemele de operare?



Comunicarea între aplicații prin intermediul unui middleware orientat pe mesaje

SWOT - SD - Avantaje

costuri reduce

SWOT - SD - Avantaje

modularitate și flexibilitate

SWOT - SD - Avantaje

- **Fiabilitate și integritate**

SWOT - SD - Avantaje

performanța

SWOT - SD - Dezvantaje

Lipsa cunoștințelor despre starea globală

SWOT - SD - Dezvantaje

Lipsa unui timp global

SWOT - SD - Dezvantaje

Nedeterminismul

SWOT - SD - Dezavantaje

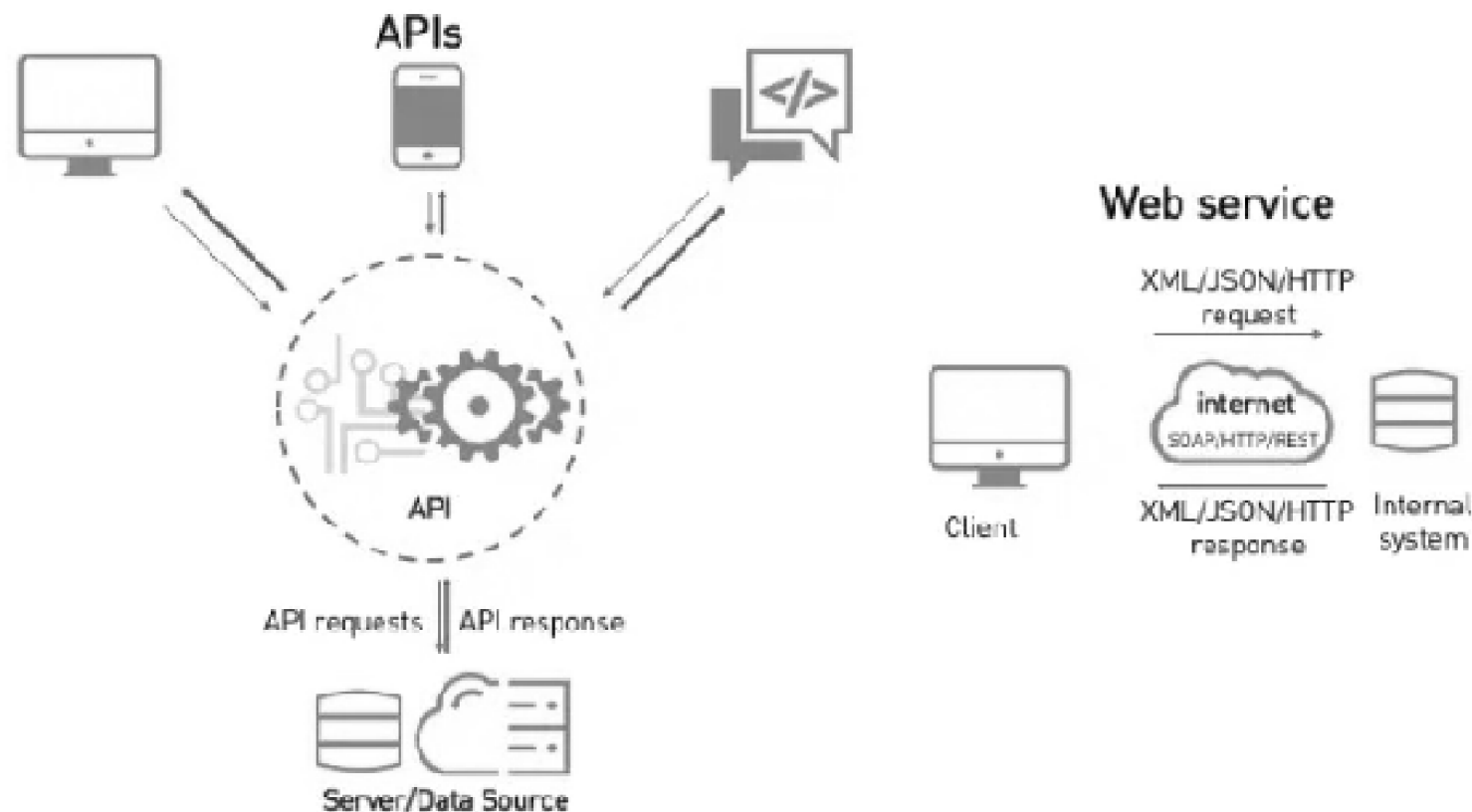
comunicațiile

SWOT - SD - Dezvantaje

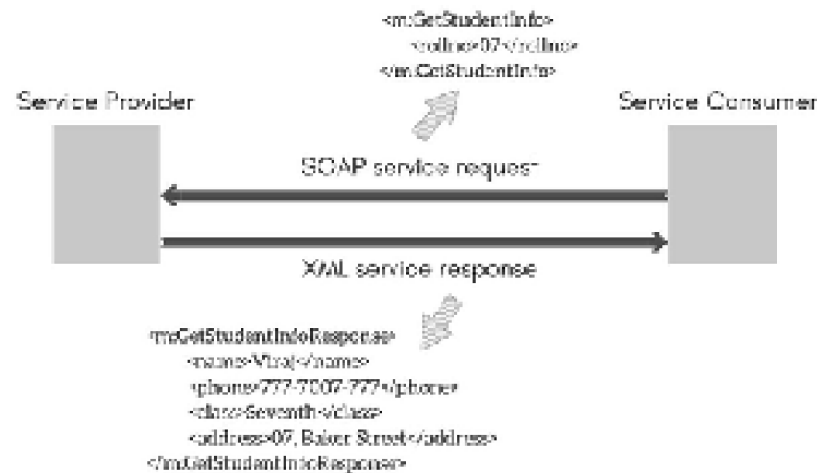
securitatea

Ce sunt Serviciile WEB?

- cum au apărut?



Standarde clasice în serviciile WEB



REST vs SOAP

REST

REST este o arhitectură software.

REST acronimul lui **Representational State Transfer**

REST poate utiliza SOAP deoarece fiind model de proiectare arhitectural poate utiliza orice protocol ar dori

REST utilizeaza URI pentru a expune logica de afaceri. Deoarece el foloseste cereri HTTP același URI poate fi utilizat pentru diverse tipuri de operații

REST preia/are securitatea protocolului de transport utilizat

REST acceptă diverse formate de date, fișier text, HTML, JSON, XML etc.

SOAP

SOAP este un protocol sau un set de standarde

SOAP acronimul lui **Simple Object Access Protocol**

SOAP nu poate utiliza rest REST deoarece este un protocol.

SOAP utilizeaza interfața serviciului pentru a expune logica de afaceri.

SOAP își definește propriul standard de securitate.

SOAP lucrează numai cu formatul XML.

Java Beans vs Enterprise Java Beans

EJB	JAVABEANS
A Java API that allows modular construction of enterprise software	Classes that encapsulates many objects into a single object
EJB requires an application server or EJB container to run EJB applications	JavaBeans should be serializable, have a zero argument constructor and allow access to properties using getter and setter methods
EJB is complex than JavaBeans	JavaBeans is simpler than EJB
Programmer can concern about the business logic as the application server manages services such as transactions and exception handling	JavaBeans allow another application to use properties and methods of the Bean

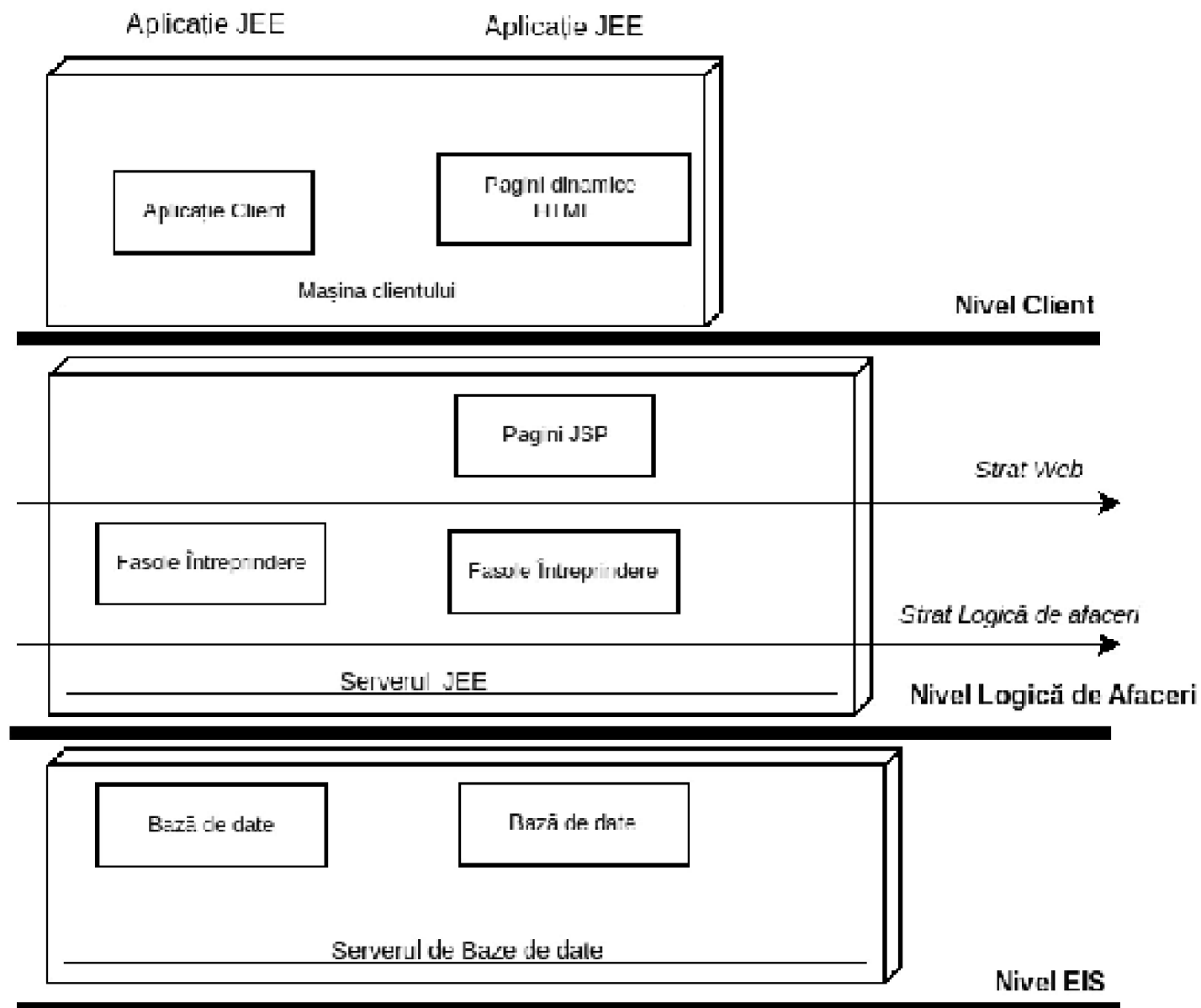
POJO vs Java Beans

POJO	Java Bean
<ul style="list-style-type: none">• Nu are alte restricții decât cele impuse de limbajul JavaNu permite un control foarte strict al membrilorNu poate implementa interfața <code>Serializable</code>Câmpurile pot fi accesate direct prin numele lorCâmpurile pot avea orice nivel de vizibilitateEste permisă dar nu obligatorie utilizarea unui constructor fără argumente (no-arg).Este recomandat a fi utilizat când nu se dorește nici un fel de restricții asupra membrilor iar utilizatorul poate avea acces complet la entitatea creată	<ul style="list-style-type: none">• Este un caz particular de POJO care are unele restricții.Permite controlul total asupra membrilorTrebuie să implementeze interfața <code>Serializable</code>Câmpurile pot fi accesate numai prin <i>getteri</i> și <i>setteri</i>.Câmpurile pot fi numai <i>private</i>Este obligatorie utilizarea unui constructor fără argumente (no-arg)Este utilizat atunci când se dorește furnizarea unui acces restricționat a utilizatorului la unele părți din entitatea creată (interfață contract etc)

Arhitectura JEE

- **multi-nivel--multistrat**
- **cadru pentru dezvoltare rapidă a aplicațiilor**
- **instalarea implică hardware heterogen**

Arhitectura JEE



Componentele unei aplicații Java EE

