

Curs 9 – Proiectarea arhitecturii sistemelor informatice

Cuprins

- ✓ Aspecte generale ale proiectării sistemelor informatice
- ✓ Principiul proiectării eşalonate a sistemelor informatice
- ✓ Proiectarea arhitecturii sistemului informatic
 - A. Arhitectura client server
 - B. Arhitecturi distribuite
 - C. Arhitectura orientată pe servicii
- ✓ Diagrama de componente
- ✓ Diagrama de desfăşurare
- ✓ Diagrama de pachete



Aspecte generale ale proiectării sistemelor informatice

- Proiectarea sistemului informatic constă în stabilirea **soluțiilor logice** și specificarea din punct de vedere **fizic a componentelor** noului sistem.
- Proiectarea se bazează în principal pe rezultatele obținute din cele două grupe de activități premergătoare:
 - definirea soluției de realizare a noului sistem
 - modelarea noului sistem

Obiectivele proiectarii

Activitati de analiza

Obiective: sa inteleaga:

- Evenimentele si procesele din companie
- Activitatile sistemului si cerintele de procesare
- Stocarea si necesarul de informatii

Modele de analiza

Documente

Activitati de proiectare

Obiective:

- Sa defineasca, sa organizeze, sa structureze componentele **sistemului solutie**

Autori: prof.dr. Ion Lungu, conf.dr. Anca Andreescu, conf. dr Ramona Bologa, lect.dr. Alexandra Florea

Aspecte generale ale proiectării sistemelor informatice

- Unele metodologii împart proiectarea sistemelor informatice în:
 - A. **proiectare generală**/proiectare de ansamblu / conceperea sistemului informatic
 - B. **proiectarea de detaliu** – include proiectarea detaliilor specifice ale programelor:
 - i. Proiectarea fiecărui caz de utilizare
 - ii. Proiectarea bazei de date
 - iii. Proiectarea interfețelor cu utilizatorul și cu alte sisteme
 - iv. Proiectarea controlului și a securității
- În cadrul acestor etape, sistemul este proiectat din punct de vedere **logic** și din punct de vedere **fizic**, separat sau nu.

Principiul proiectării eşalonate a sistemelor informatice

- La stabilirea **ordinii** de prioritate în abordarea structurilor sistemului informatic pot fi avute în vedere următoarele criterii:
 - prioritatea **obiectivelor** componentelor;
 - asigurarea **legăturilor** între componente;
 - disponibilitatea **resurselor**(limita fondurilor ce pot fi alocate în timp pentru realizarea sistemului informatic);
 - nivelul de **dotare cu tehnică de calcul existent** în etapa de concepere și cel prevăzut a fi atins în timp;
 - **forțele de proiectare** pe care le va antrena proiectul;
 - **personalul de specialitate** existent și în pregătire, la unitatea beneficiară, necesar pentru implementarea și exploatarea curentă a sistemului informatic).

Intrebari cheie ale activitatii de proiectare

Activitate de proiectare	Intrebare
Proiectarea mediului de executie	A fost specificat in detaliu mediul de executie software cu toate optiunile?
Proiectarea arhitecturii aplicatiei si software-lui	Au fost specificate in detaliu toate elementele software-lui si cum va fi realizat fiecare caz de utilizare?
Proiectarea interfetelor cu alte sisteme	A fost specificat in detaliu modul in care sistemul va comunica cu toate celelalte sisteme din interior/exteriorul organizatiei?
Proiectarea bazei de date	Au fost specificate in detaliu toate cerintele de stocare ale datelor, inclusiv toate elementele schemei BD?
Proiectarea controalelor si securitatii sistemului	Au fost specificate in detaliu toate elementele pentru a asigura siguranta sistemului, securitatea si protectia

i. Proiectarea mediului

- **MEDIUL** reprezinta toate **tehnologiile necesare** pentru a sustine aplicatia software
 - Calculatoare server sau desktop
 - Dispozitive mobile, sisteme de operare
 - Facilitati de comunicatie, de intrare si de iesire
- Se mai numeste si **Arhitectura tehnica**

ii. Proiectarea arhitecturii aplicatiei si software-lui

- Impartirea sistemului in **subsisteme**
- Definirea **arhitecturii software**
 - Pe trei niveluri sau model-view-controller
- Proiectarea detaliata a **fiecarui caz de utilizare**
 - diagrame de clase de proiectare
 - diagrame de secvente
 - diagrame de stari

iii. Proiectarea interfetelor cu utilizatorii

- Proiectarea dialogului **pleaca de la cerinte**
 - Fluxul de activitati din cazurile de utilizare
 - Diagramele de secventa ale sistemului
- In prezent este necesara proiectarea de interfete pentru **diverse medii si dispozitive**:
 - Telefoane smart
 - Tablete, iPad, etc

iv. Proiectarea interfetelor cu alte sisteme

- Sistemul informatic interactioneaza cu alte sisteme interne sau externe, fiind necesara integrarea acestora
- Conectarea la alte sisteme se poate realiza in multiple feluri:
 - Se pot salva datele utilizate de alte sisteme
 - Se pot utiliza date salvate din alte sisteme
 - Se pot face solicitari de informatii in timp real
 - Se poate apela la servicii software

iv. Proiectarea bazei de date

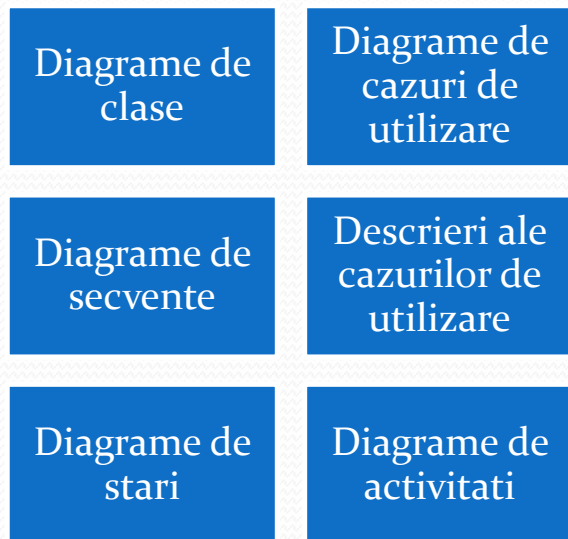
- Se pleaca de la **modelul claselor domeniului**
- Se alege **structura bazei de date**
 - De obicei se lucreaza cu baze de date relationale, dar pot exista platforme care lucreaza cu baze de date orientate obiect
- Se proiecteaza **arhitectura BD** (distribuita, etc)
- Se proiecteaza **schema bazei de date**
 - Tabelele si coloanele in relational
- Se proiecteaza **restrictiile de integritate referentiala**
 - Referinte prin chei externe

v. Proiectarea securitatii si controalelor

- Scopul este de a proteja bunurile companiei
- Este un element crucial in Internet si retele wireless
- Se vor proiecta controale la nivel de:
 - Interfata cu utilizatorul
 - Aplicatie
 - Baza de date

Modele de analiza si modele de proiectare

ANALIZA



PROIECTARE



Autori: prof.dr. Ion Lungu, conf.dr. Anca Andreescu, conf. dr Ramona Bologa, lect.dr. Alexandra Florea

Proiectarea arhitecturii tehnice (mediului)

1. **Proiectarea unui sistem cu implementare internă**
 - a. Sisteme software **stand alone** - care rulează pe un singur dispozitiv, fără conectare la rețea
 - b. Sisteme software **bazate pe o rețea internă** –
 - Arhitectura client-server, rețea LAN
 - Aplicații desktop și aplicații bazate pe browser
 - c. Arhitecturi **client server pe trei niveluri**
 - Nivelul datelor, nivelul aplicației și nivelul prezentare
 - Aplicații desktop și aplicații bazate pe browser

Proiectarea arhitecturii tehnice

2. Proiectare pentru implementare externa

- Configurare pentru implementare pe Internet – cu avantaje si riscuri
- Alternative de gazduire pentru implementare pe Internet
 - Colocatie
 - Servicii de management
 - Servere virtuale
 - Cloud computing
- Diversitatea dispozitivelor client
 - Laptop, tablete si notebook, telefon smart etc

Configuratie pentru implementare in Internet

- **Avantaje**

- Accesibilitate numar mare de utilizatori potentiali
- Costuri reduse ale comunicatiei
- Standarde utilizate pe scara larga – standarde Web arhicunoscute

- **Probleme potentiale:**

- Securitate – brese de Securitate pe serverele Web, atacuri hacker
 - https – Hypertext Transfer Protocol Secure
 - TLS – Transport Layer Security – ver. avansata a protocolului de retea SSL
- Volumul datelor trasmise prin retea - daca traficul este incarcat, volumul datelor trasmise si durata sunt ridicate

Proiectare pentru mediu distribuit, la distanta

- Se folosesc **doua interfete** ale aceleiasi aplicatii Web pentru utilizare **interna**, respective **externa**
 - Securitate mai redusa
- **Virtual private network (VPN)**
 - Retea inchisa cu securitate si acces inchis construita pe baza unei retele publice (Internet)

Proiectarea arhitecturii sistemului informatic

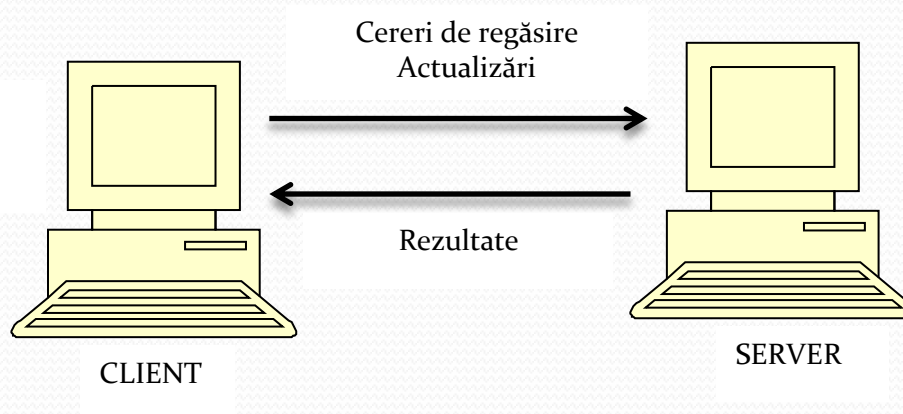
- Cele mai cunoscute **tipuri de rețele** sunt:
 - Rețea punct la punct (bus);
 - Rețea inel (ring);
 - Rețea stea (star);
 - Rețea ierarhică.
- După alegerea tipului de rețea, echipa de analiză trebuie să identifice **protocolul de comunicație**.
Cele mai cunoscute protocoale de comunicații sunt:
 - **TCP/IP**. Este un protocol indicat în cazul utilizării unei rețele Ethernet sau în cazul în care calculatoarele din rețea au arhitecturi diferite;
 - **SNA**. Este utilizat, în general pentru conectarea mainframe-urilor IBM.

A. Arhitectura client/server

- Arhitectura client/server este un ansamblu de trei componente principale: server, client și o rețea care conectează calculatoarele client la servere pentru a colabora la îndeplinirea sarcinilor.
- Tipurile de aplicații client/server sunt:
 - Sisteme cu baze de date
 - Poșta electronică (E-mail)
 - Sisteme de tip „groupware“
 - Sisteme moștenite
- O arhitectură distribuită presupune existența unor baze de date multiple (care se găsesc pe calculatoare distincte) și a unor aplicații care manipulează datele de la diferite stații de lucru locale cu ajutorul unor sisteme de gestiune a bazelor de date (SGBD).

Sistem client server

- Interfața utilizator
- Logica aplicației



- Conectare
- Acces la baza de date
- Logica aplicației
- Protecție și securitate

B. Arhitecturi distribuite - Distribuirea datelor

a) Distribuire prin fragmentare

- Reguli:
 - Completitudinea
 - Reconstrucția
 - Disjuncția
- Metode:
 - Metoda orizontală
 - Metoda verticală
 - Metoda mixta

Distribuirea datelor

b) Distribuire prin replicare.

- Soluția este utilă atunci când diferiți utilizatori, de la diferite noduri de rețea, au nevoie simultan de ultimele informații, iar baza de date este distribuită local.
- Din punct de vedere al replicării, proiectarea replicării se poate realiza astfel:
 - Datele nerePLICATE
 - Datele replicate parțial
 - Datele replicate total

Distribuirea datelor

c) Distribuirea mixtă.

d) Distribuirea prin încărcare.

- Tehnica se folosește atunci când datele sunt stabile, deci se actualizează rar, sau atunci când nu toți utilizatorii trebuie să aibă acces la datele de ultimă oră.
- După alegerea tipului de distribuire, se vor proiecta elementele specifice fiecărui tip.

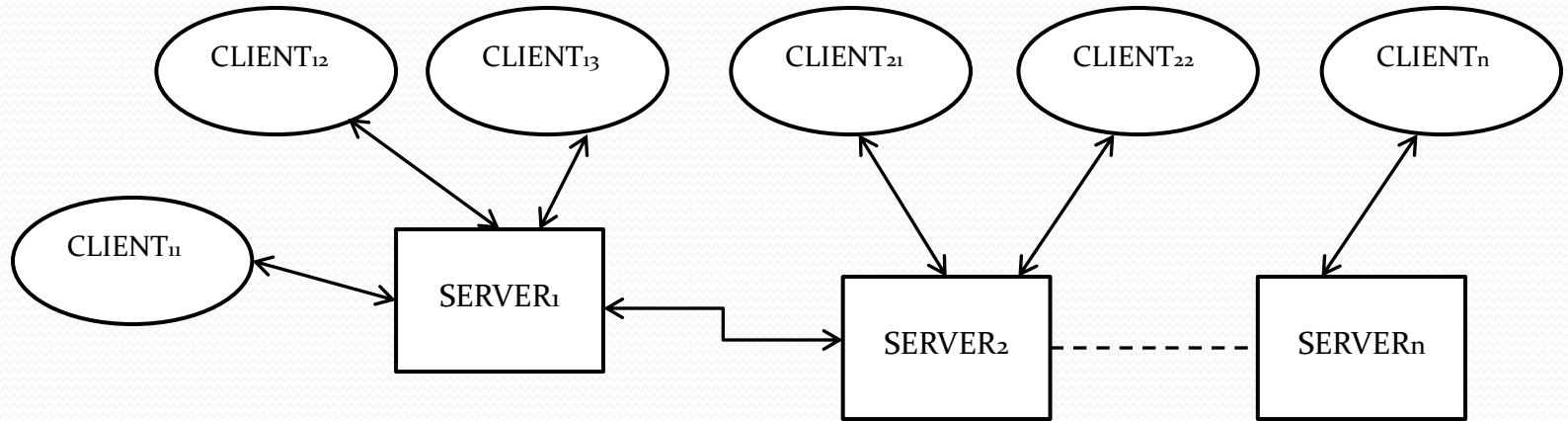
Distribuirea datelor din catalog

- **Catalogul bazei de date distribuite** conține:
 - informațiile despre **schema globală**: numele colecțiilor globale, numele caracteristicilor (câmpurilor) din fiecare colecție;
 - informații despre **fragmentare**: calificarea fragmentelor (pentru metoda orizontală), câmpurile din fragmente (pentru metoda verticală), arborele de fragmentare prin calificare și descrierea fragmentelor prin câmpuri (pentru metoda mixtă).
 - informații despre **alocare**: legăturile dintre fragmentele și imaginile fizice ale colecțiilor globale, precum și legăturile dintre imaginile fizice și datele (prin numele lor) memorate pe fiecare calculator din rețea.
 - informații despre **accesul la date**: metadatele de acces utilizate pe fiecare calculator din rețea (index, hash, pointeri etc.), restricțiile de integritate impuse la descrierea datelor, securitatea datelor (drepturi de acces etc.).
 - informații statistice: indicatori statistici care dau profilul bazei de date. .
- **Distribuirea catalogului** poate fi realizată în următoarele variante:
 - Catalogul replicat
 - Catalogul local
 - Catalogul centralizat
 - Catalogul mixt

Sistem distribuit

- După luarea în calcul a tuturor **avantajelor** și **dezavantajelor** diferitelor modalități de distribuire a sistemelor, datelor și catalogului de date și, mai ales, în funcție de complexitatea sistemului și de cerințele beneficiarului, se va realiza proiectarea sistemului distribuit

Sistem cu aplicații distribuite client server



C. Arhitectura orientată pe servicii

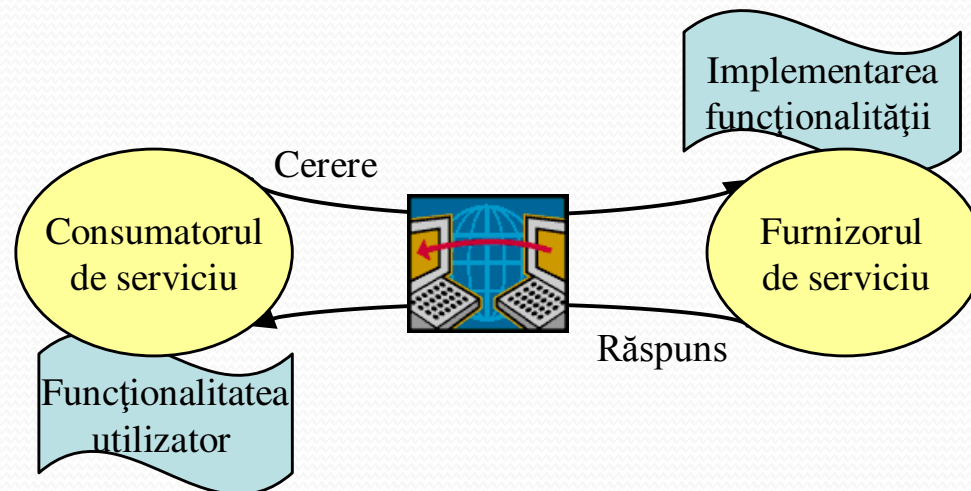
SOA- Service Oriented Architecture

- SOA se bazează pe un mecanism de cerere/răspuns convențional: un **consumator de serviciu** invocă un **furnizor de serviciu** prin rețea și așteaptă până când se va realiza operația la furnizor.
- Astfel, SOA divide o aplicație într-un **coordonator de serviciu**, care reprezintă funcționalitatea la utilizator și **furnizorii de servicii**, care implementează funcționalitatea.

Arhitectura orientată pe servicii

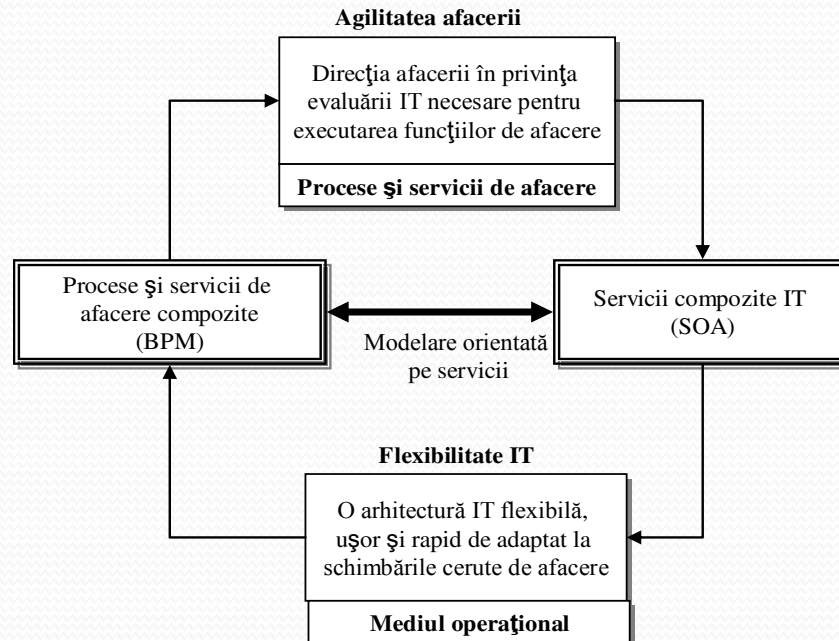
SOA- Service Oriented Architecture

- În timp ce coordonatorul tinde să fie unic pentru o aplicație particulară, **un serviciu** poate fi reutilizat și partajat de multiple aplicații compozite.
- Coordonatorul serviciului, în mod explicit, specifică și invocă serviciile dorite .



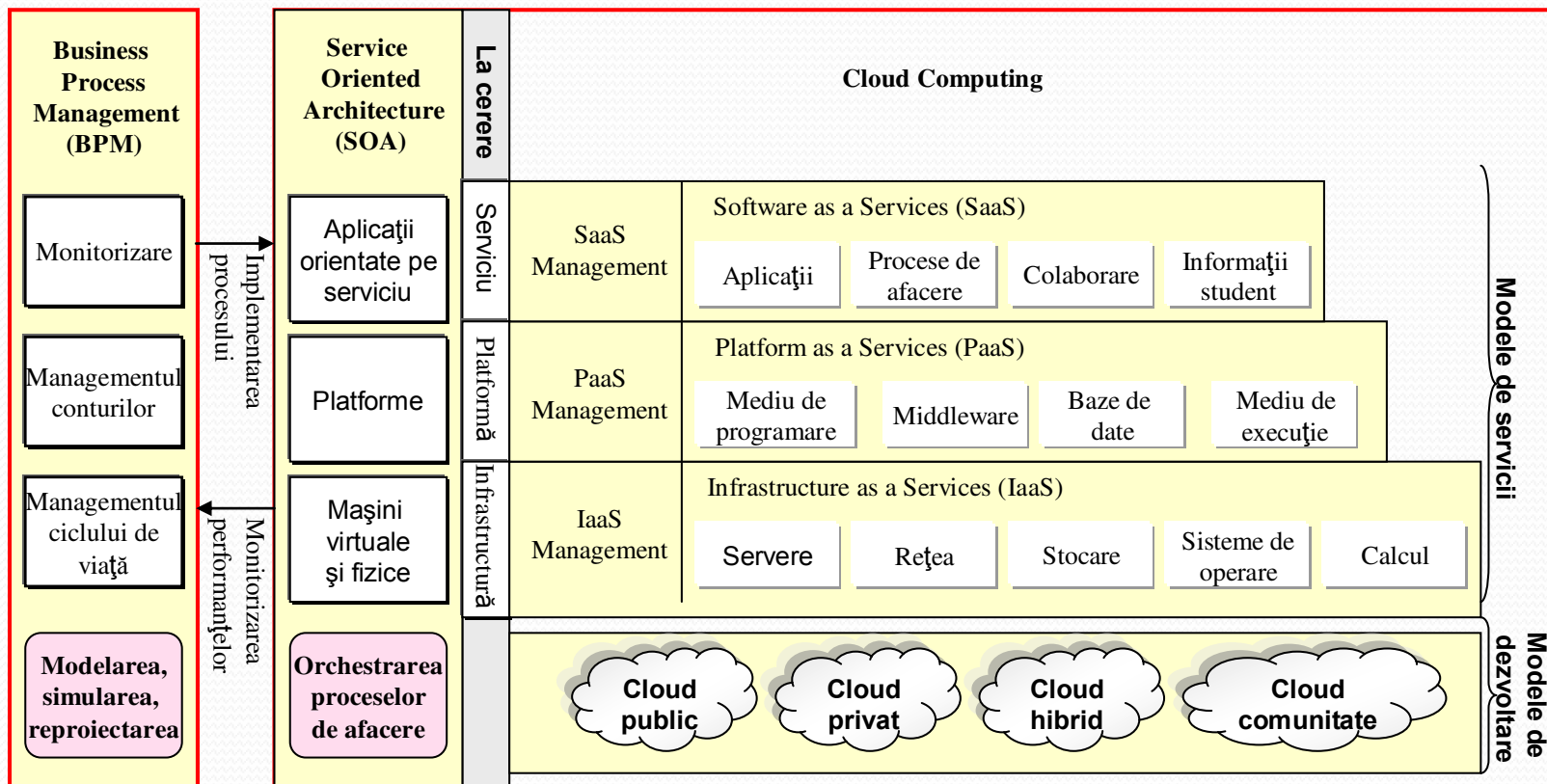
Relația dintre BPM și SOA

- O arhitectură orientată pe servicii poate aduce multe beneficii, cum ar fi: promovarea reutilizării, abilitatea de a combina serviciile pentru crearea de noi aplicații compozite, utilizarea serviciilor decuplate prin interfața standard, oferind în același timp un comportament tehnologic pentru dezvoltare de soluții de afaceri.
- Gestionarea eficientă a acestor schimbări implică utilizarea de soluții BPM, contribuind totodată la guvernare SOA și automatizarea proceselor .



Relația dintre Cloud computing și SOA/BPM

- În ceea ce privește relația dintre Cloud Computing și SOA / BPM se poate spune că: 1. la nivel de organizație Cloud Computing reprezintă o extindere a SOA și 2. Cloud Computing creează noi oportunități pentru furnizorii de produse și servicii BPM.



Arhitectură pentru servicii web care să permită BPMS

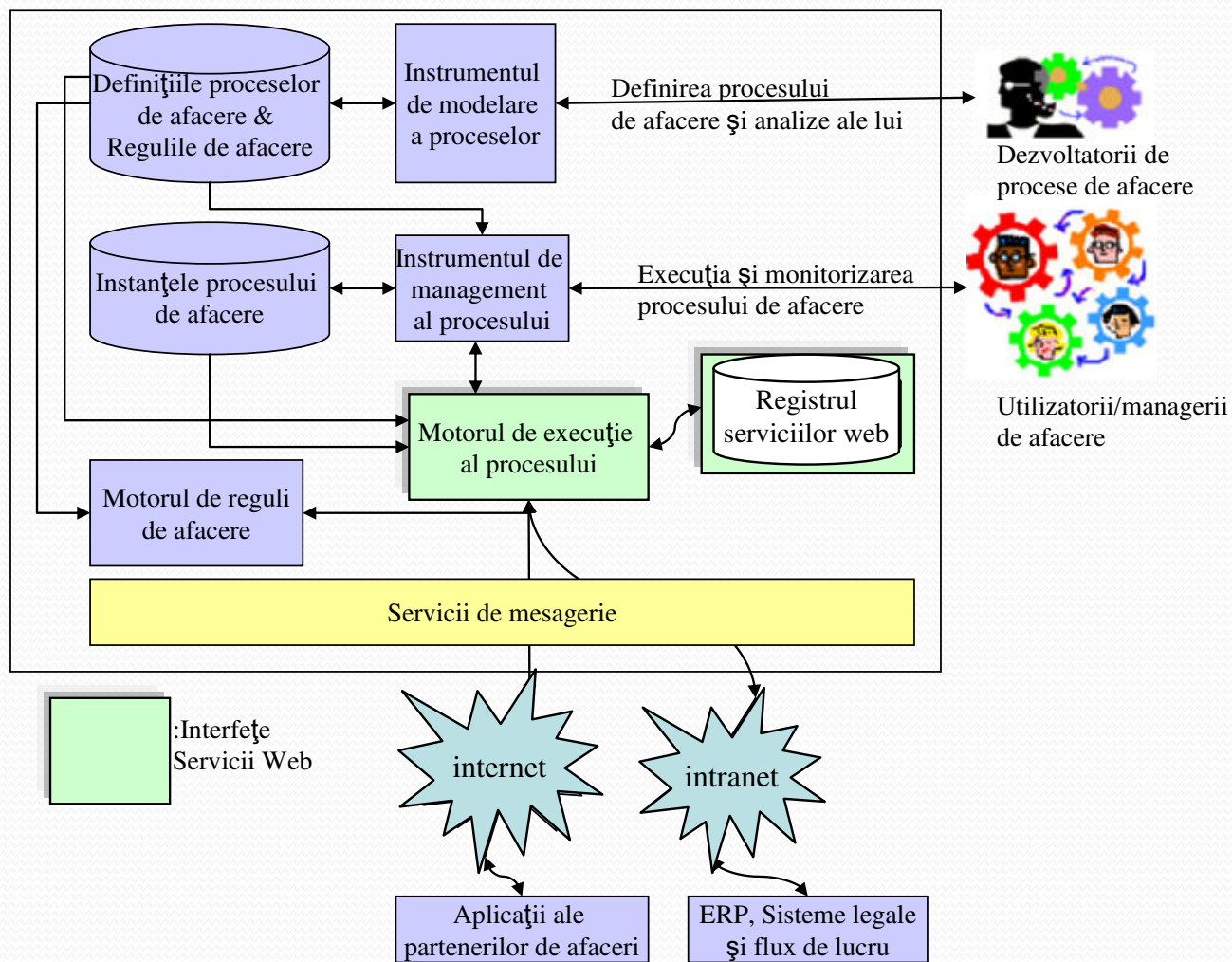


Diagrama de componente

- O diagramă de componente prezintă dependențele existente între diverse componente software ce compun un sistem informatic.
- Aceste **dependențe** sunt:
 - **statice** - au loc in etapele de compilare sau link-editare
 - **dinamice** - au loc in timpul execuției
- **O componentă** este un modul soft (cod sursa, cod binar, dll, executabil etc) cu o **interfață** bine definita(API)
- **API** – Application Program Interface - setul de metode publice disponibile din exterior

Diagrama de componente

- În general numele unei componente este **numele fișierului** reprezentat de componentă.
- Obiectele implementate de o instanță a componentei se reprezintă grafic în interiorul simbolului instanței componentei.



Reprezentarea grafică a componentelor în UML

Diagrama de componente

- Diagrama de componente este un graf de componente între care există relații de **dependență** sau de **compunere** (componente incluse fizic în alte componente).
- **Dependențele** între componente se reprezintă grafic prin linii întrerupte între o *componentă client* și o *componentă furnizor* de servicii, orientate spre componenta furnizor.
- Relația de dependență semnifică faptul că *clasele incluse în componenta client pot moșteni, instanția sau utiliza clase incluse în componenta furnizor*.
- De asemenea, pot exista **relații de dependență între componente și interfețe** ale altor componente, relații care semnifică faptul că un client utilizează operații ale componentei furnizor

Diagrama de componente

- Exemple de stereotipuri predefinite pentru componente:
 - programe principale (<<Main Program>>)
 - subprograme (<<SubProgram>>)
 - pachete (<<Package>>)
 - Biblioteci cu legare dinamica (<<DLL>>)
 - procese (<<Task>>)
 - executabile (<<EXE>>)

Exemplu de diagramă de componente

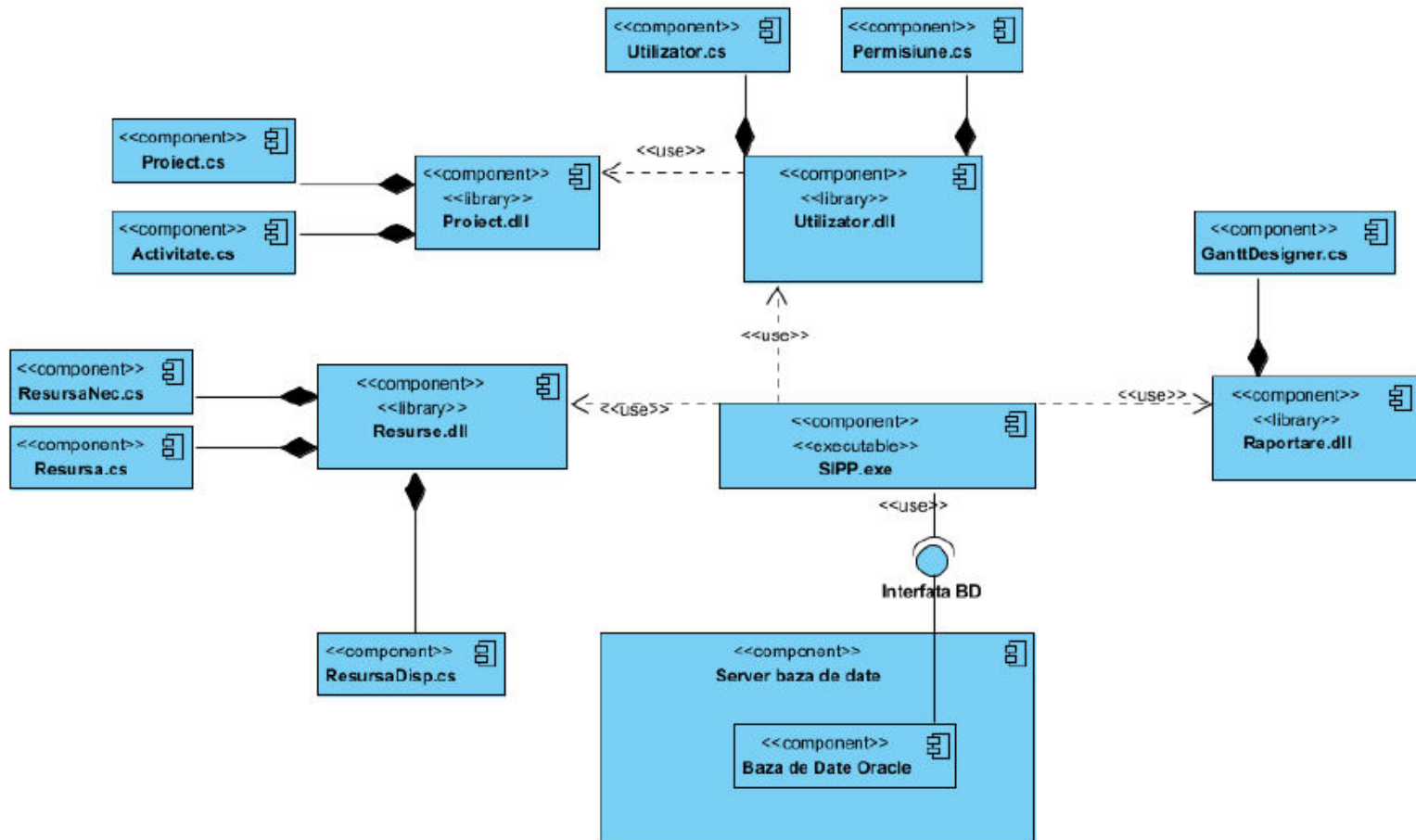


Diagrama de desfășurare

- Diagramele de desfășurare prezintă configurația elementelor de procesare din timpul execuției și componentele, procesele și obiectele care le conțin.
- O diagramă de desfășurare este un graf de **noduri** conectate prin asocieri de **comunicare**.
- **Un nod** este o entitate fizică ce reprezintă o resursă de procesare, având o memorie și anumite capabilități de procesare (dispozitive de calcul, resurse umane, resurse de procesare mecanică).
- Un nod este reprezentat grafic prin intermediul unui **paralelipiped**. Un tip de nod are asociat un nume, iar o instanță a unui nod are asociate (opțional) un nume de instanță și un nume de tip (*nume instanță : nume tip*). O **asociere între două noduri** indică existența unei căi de comunicare între noduri.

Diagrama de desfășurare

- Diagramele de desfășurare pot fi utilizate pentru reprezentarea **componentelor** ce pot aparține anumitor noduri prin imbricarea grafică a simbolului componentei în cadrul simbolului ce reprezintă nodul.
- Între componente pot exista și relații de dependență.

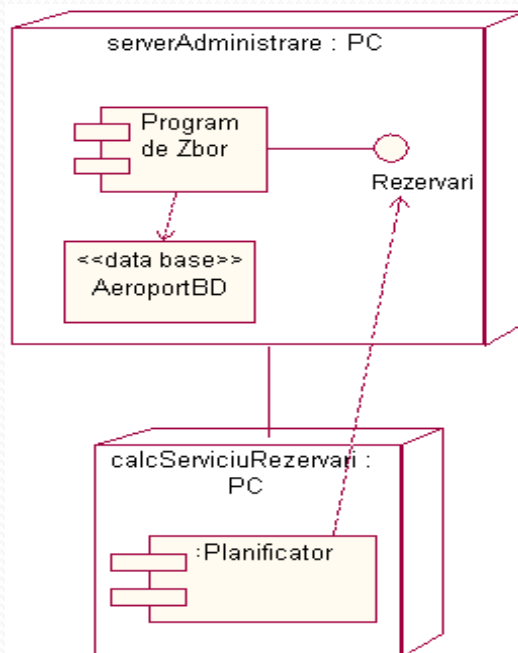


Diagrama de desfășurare

- Diagramele de desfășurare conțin două tipuri de noduri: **medii de execuție** și **dispozitive**.
 - **Mediile de execuție** reprezintă componente hardware capabile să execute programe.
 - **Dispozitivele** reprezintă componente hardware fără putere de calcul. Numele asociat unui dispozitiv este în general unul generic (ex. imprimanta, modem, terminal etc)
- O conexiune reprezintă o legătură hardware (în general bidirecțională) între două dispozitive sau procesoare.

Exemplu de diagramă de desfășurare

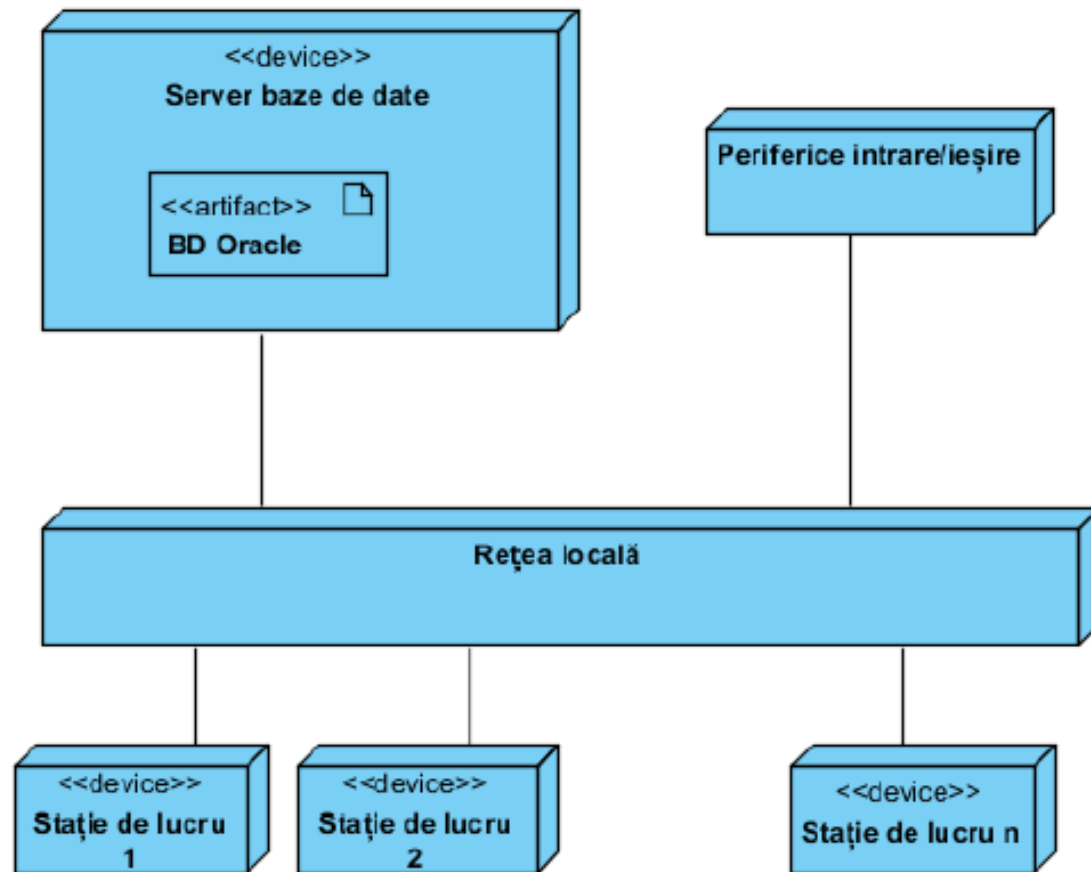


Diagrama de pachete

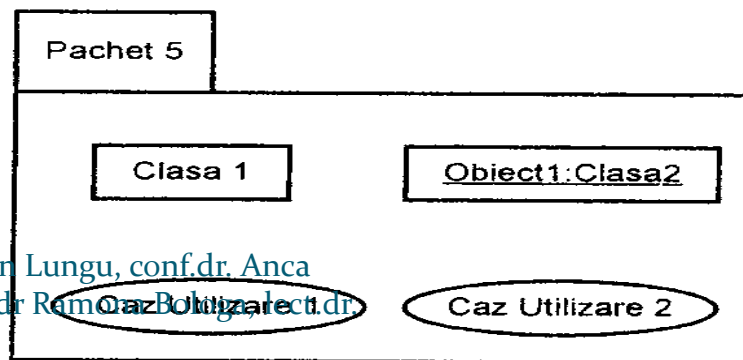
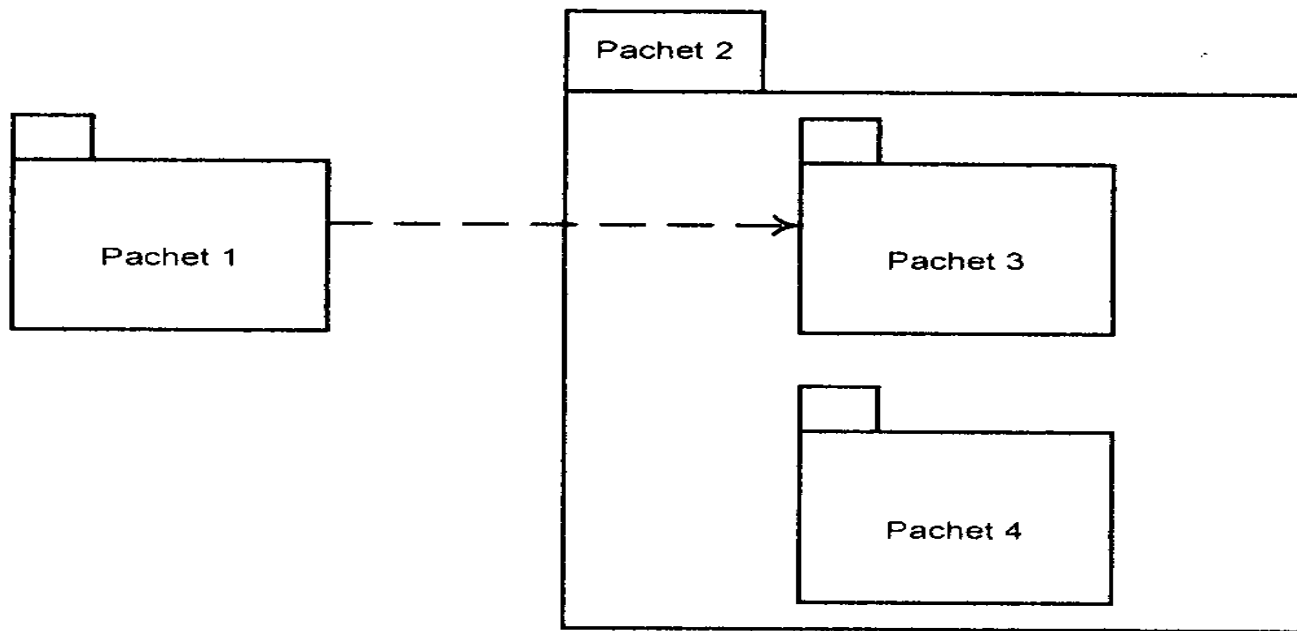
- Elementele din cadrul diagramelor pot fi grupate in pachete.
- Într-un pachet pot fi incluse alte pachete, clase, cazuri de utilizare, colaborări etc.
- Un pachet arată doar structurile pe care le conține, nu și comportamentul elementelor sale.
- Un element de modelare aparține unui singur pachet, dar alte pachete pot consulta acest element. Dacă se arată explicit conținutul pachetului, atunci numele pachetului se trece pe etichetă.
- Un pachet este un mecanism destinat unor scopuri generale, care organizează elementele în grupuri. Fiecare pachet are un nume care poate fi simplu sau cu cale (path name).

Diagrama de pachete

- Un pachet poate conține clase, interfețe, componente, noduri, colaborări, cazuri de utilizare, diagrame și chiar alte pachete. „A conține“ este o relație compusă, ceea ce înseamnă că fiecare element este declarat în pachet.
- Din punct de vedere al vizibilității, pachetele se comportă precum clasele.
- UML definește cinci stereotipuri care se aplică pachetelor:
 - *facade* - un pachet care este doar o vizualizare a altui pachet;
 - *framework* - un pachet care conține în principal modele (*patterns*);
 - *stub* - este un proxy pentru conținutul public al altui pachet;
 - *subsystem* - un pachet ce reprezintă o parte independentă a sistemului modelat;
 - *system* - un pachet ce reprezintă tot sistemul modelat

Autori: prof.dr. Ion Lungu, conf.dr. Anca
Andreescu, conf. dr Ramona Bologa, lect.dr.
Alexandra Florea

Exemplu de diagramă de pachete



Autori: prof.dr. Ion Lungu, conf.dr. Anca
Andreescu, conf. dr. Remus Boliga, lect.dr.
Alexandra Florea