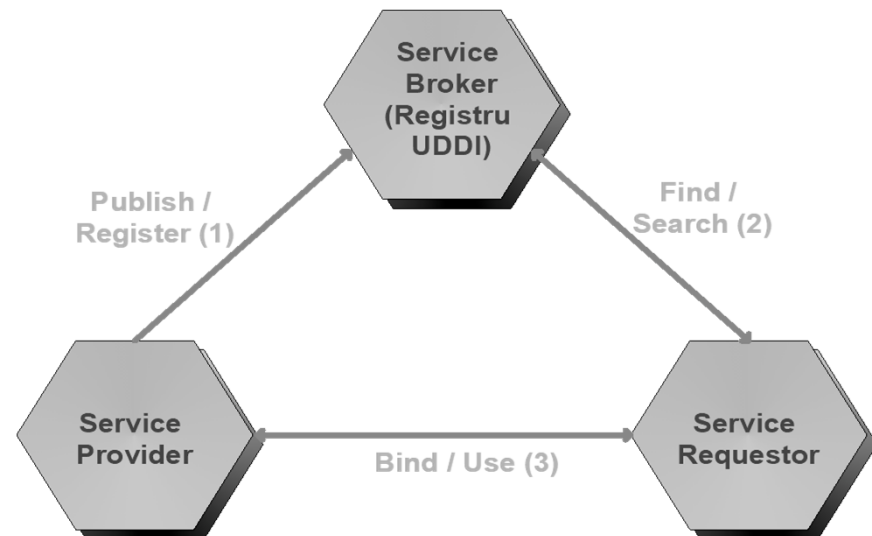


M. Caramihai, © 2020

**PROGRAMAREA
ORIENTATA
OBIECT**

CURS 7

SOA (Services Oriented Architecture)



Glosar termeni

- RPC = Remote Procedure Call
- CORBA = Common Object Request Broker Architecture
- APPC = Advances Program to Program Communications
- EDI = Electronic Data Interchange
- XML = Extensible Markup Language
- BPM = Business Process Management
- UDDI = Universal Description, Discovery and Integration
- WSDL = Web Services Description Language
- SOAP = Simple Object Access Protocol
- API = Application Program Interface
- ERP = Enterprise Resources Planning
- CRM = Custom Relationship Management
- EAI = Enterprise Application Integration
- CCI = Client side Communication Interface
- NEP = Network Equip. Provider

Economia bazata pe servicii (1)

- Evolutia (economica a) societatii s'a bazat pe:
 - Specializare
 - Standardizare
 - Scalabilitate
- In prezent, economia se bazeaza (aproape) in exclusivitate pe servicii:
 - Transport
 - Telecomunicatii
 - *Retail*
 - Sanatate
 - Financiar – bancar
 - ...

Economia bazata pe servicii (2)

- Obiectivele afacerii:

- Costuri mici
- Eficientizare
- Monitorizare usoara a activitatilor
- Detectia si gestiunea exceptiilor
- Raspunsuri rapide la probleme, etc
- solutia: **IT**

- Tendinte in afaceri:

- Spatial: intra-intreprindere → inter-intreprindere → global
- Temporal: manual → electronic → web
- IT: mainframe → baterii de servere → seturi de servicii web

Economia bazata pe servicii (3)

- Problemele IT: exista diferite:

- Sisteme de operare
- Functionalitati
- Formate de date
- Sisteme de securitate
- Infrastructuri hard
- Protocoale de interactiune
- Limbaje de programare

Solutie: integrare

Despre servicii

- Definitie serviciu: o unitate de lucru realizata de un furnizor / service provider, in vederea finalizarii unui rezultat pentru un client
- SOA → termen introdus de catre **Yefim Natis** in 1994
- Justificare:
 - Metodele OOAD (Object Oriented Analysis and Design) curente nu adreseaza cele 3 elemente cheie ale unui SOA: **servicii, fluxuri, componente.**
 - Este nevoie sa fie adresate explicit tehnicile si procesele necesare pentru a identifica, specifica si realiza serviciile, fluxurile si structura lor.

Servicii: attribute

- Bine definite, usor de utilizat, cu interfata standardizata
- *Self-contained* fara dependente vizibile spre alte servicii
- (uzual) Valabile intotdeauna (dar inactive pana la sosirea unei solicitari)
- Usor de accesat si utilizat, nu este necesara nici o "integrare"
- Independent de contextul utilizarii,
 - Dar un serviciu trebuie sa aiba un context
- Prin combinarea serviciilor existente se pot oferi unele noi
- Calitatea serviciilor poate fi cuantificabila
 - Nu se refera la "Ce" – ci la "Cum"
 - Performantae/Calitate
 - Cost
 - ...

Context, compozitie si stare

- Serviciile sunt deseori proiectate sa ignore contextul in cadrul caruia sunt apelate:
 - Serviciile sunt mai degraba independente de context
 - Serviciile pot fi utilizate intr'un context ce nu este cunoscut in momentul proiectarii
- Noi servicii pot fi create prin combinarea celor existente
 - Rezervarea (proiectarea) unei calatorii vs. proiectarea unui drum cu masina, avionul, ...

Interfata cu serviciile

- Nespecifica
 - Toti furnizorii de servicii ofera (in general) acelasi tip de interfata
- Polimorfism accentuat
- Implementarea se poate modifica utilizand diverse modalitati prin care utilizatorul sa nu fie afectat.
 - Serviciile interactioneaza cu foarte multi 'consumatori'

Oferta de servicii

- **Utilizatorul** de servicii isi prezinta 'cererea'
- **Furnizorul** de servicii isi prezinta 'oferta'
- Un '**facilitator**' (*broker*) are rolul:
 - Sa gaseasca cea mai buna oferta in raport cu o anumita cerere
 - Sa prezinte o oferta in diverse modalitati a.i. sa corespunda diverselor cereri
- Cerere / Raspuns reprezinta o forma particulara de aplicare a unui protocol cerere / oferta

Scalabilitatea serviciilor

- 'Consumatorul' poate utiliza / combina diverse tipuri de servicii fara sa trebuiasca sa cunoasca modul in care lucreaza serviciile respective
- Furnizorii de servicii pot sa'si largeasca aria de utilizatori prin optimizarea
 - Facilitatilor la nivelul interfetei
 - Posibilitatilor de acces
 - Reducerea cuplarii la nivelul elementelor sistemului informatic

SOA – abordare formală

- Un stil arhitectural în scopul cuplării diverselor tipuri de servicii prin intermediul unui protocol de comunicare.
- Un model computațional ce are la bază Internetul.
- Orientarea spre servicii se referă la descrierea lor dinamică, publicare și utilizare
- Serviciu web:
 - Arhitectura bazată pe XML pt transmiterea mesajelor, descrierea serviciilor și facilități extinse
 - Arhitectura distribuită ce permite utilizarea în comun a aplicațiilor slab cuplate (prin realizarea

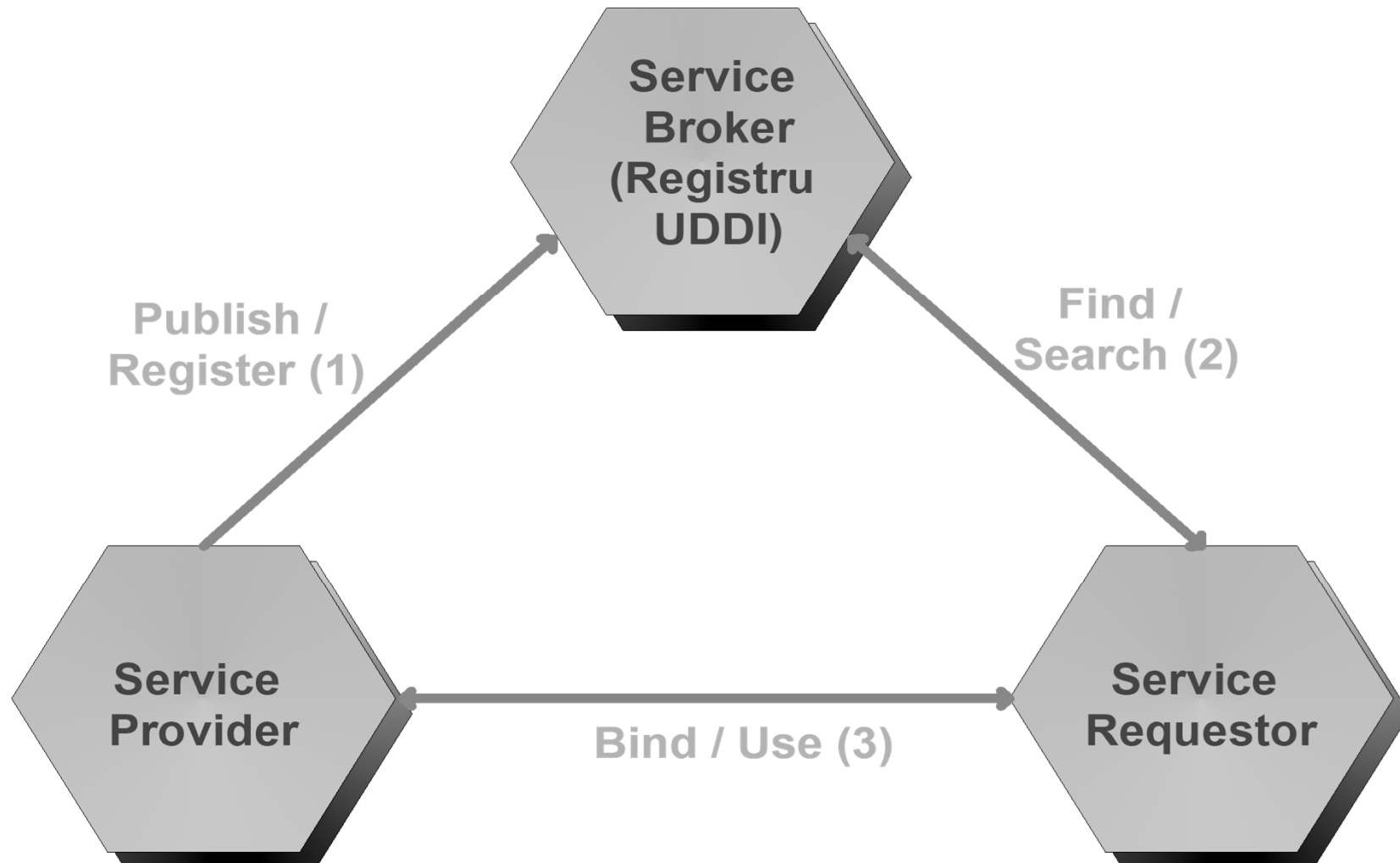
SOA – definitie

- SOA descrie un set de sabloane pentru a crea servicii slab cuplate care, datorita separarii interfetelor, implementarilor si protoacoalelor, furnizeaza o mare flexibilitate in receptivitatea la cerintele actuale de business functionale si non-functionale:
 - performanta
 - securitate
 - scalabilitate

SOA – objective

- Cuplare slaba: descompunerea in serv. independente ajuta la scaderea dependentei de un singur proces
- Neutralitatea de platforma: transmiterea de mesaje (d.e. XML) creste capabilitatea de a atinge neutralitatea de platforma
- Reutilizare a componentelor
- Scalabilitate: adaugarea de logici noi trebuie sa fie facila
- Standardizare: serviciul trebuie sa depinde doar de elementele sale descriptive

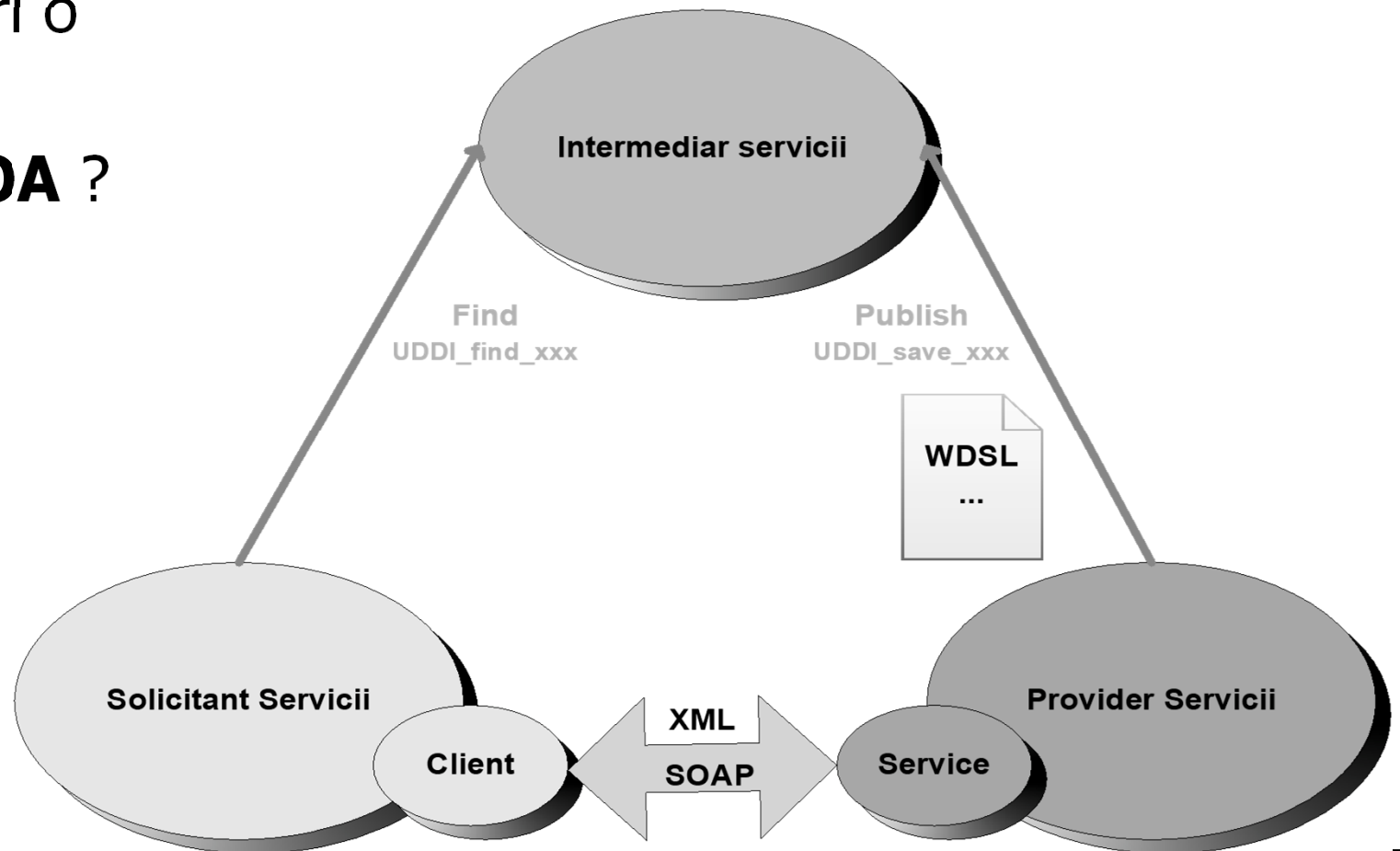
Trinitatea SOA



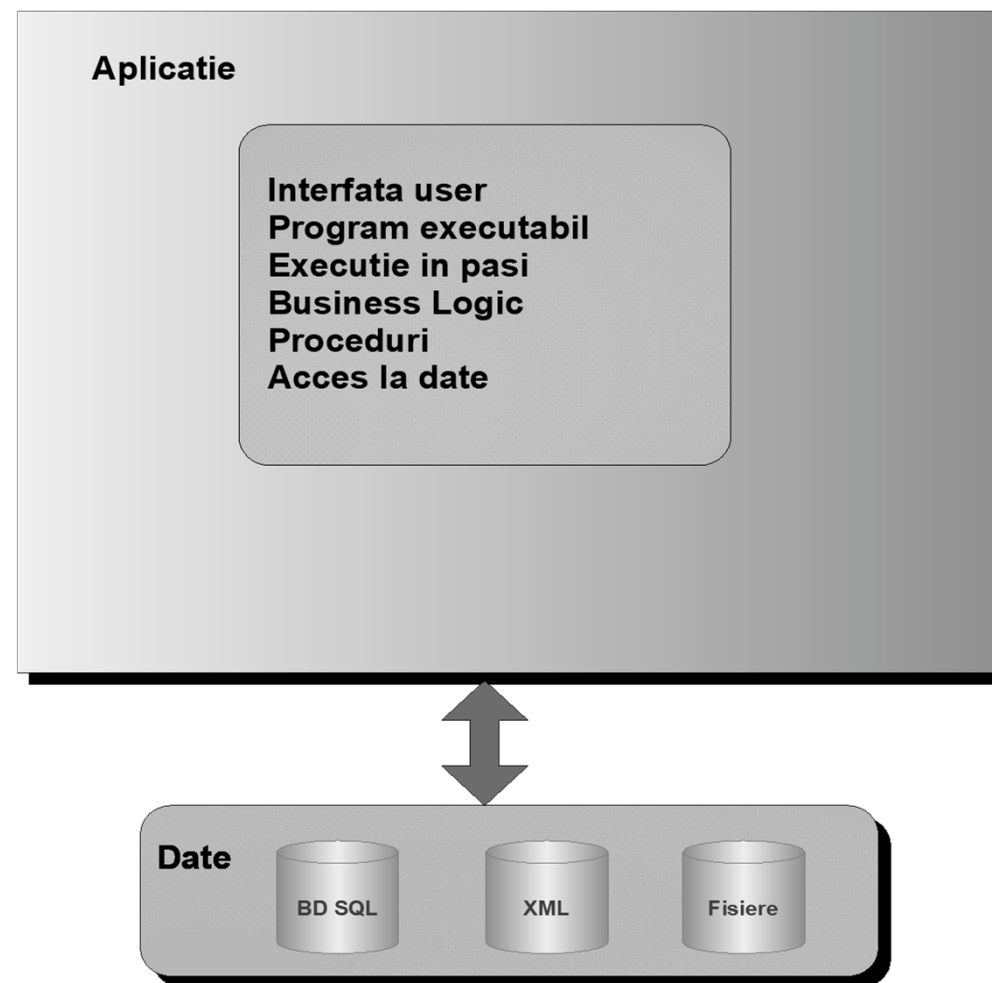
Servicii Web & SOA

Se face de multe ori o
confuzie:

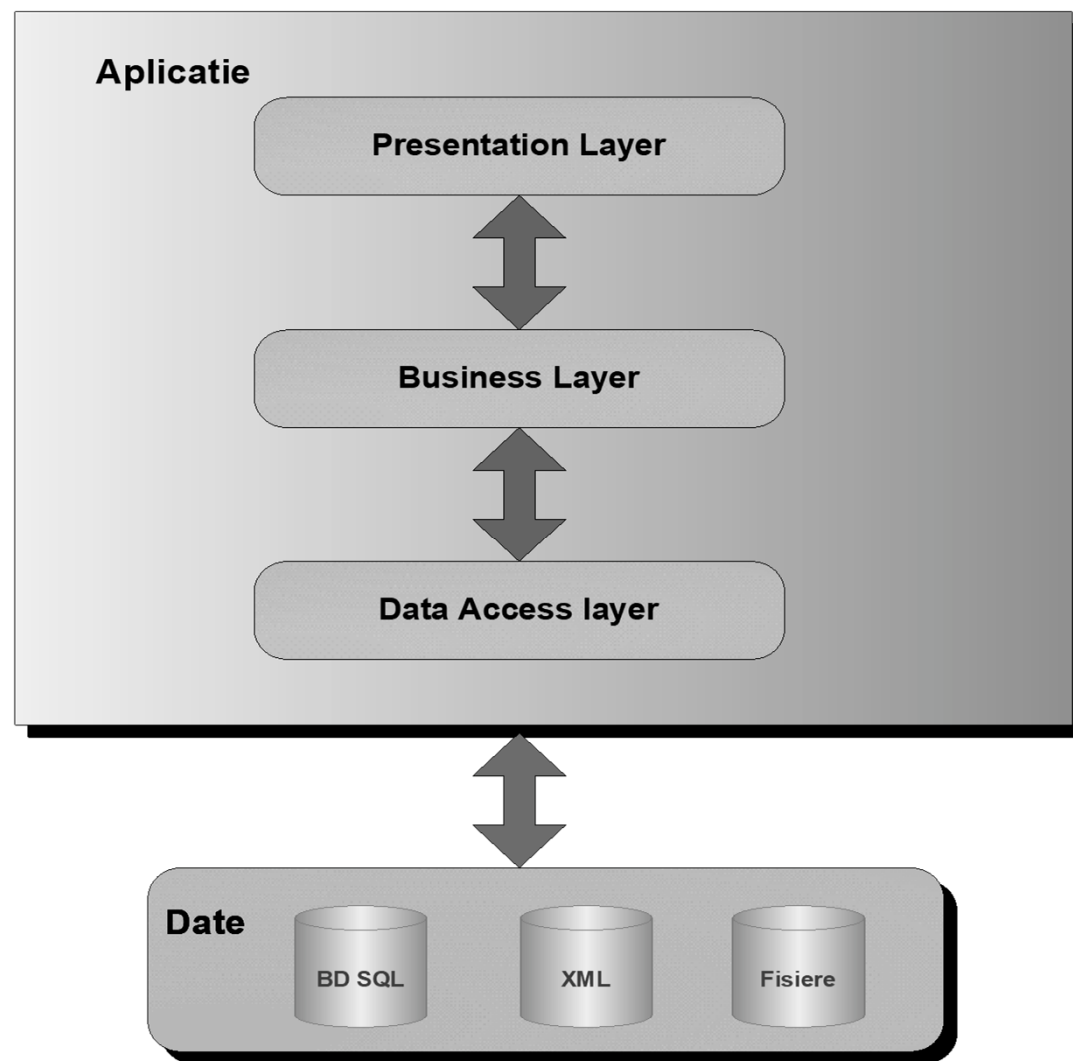
web service = SOA ?



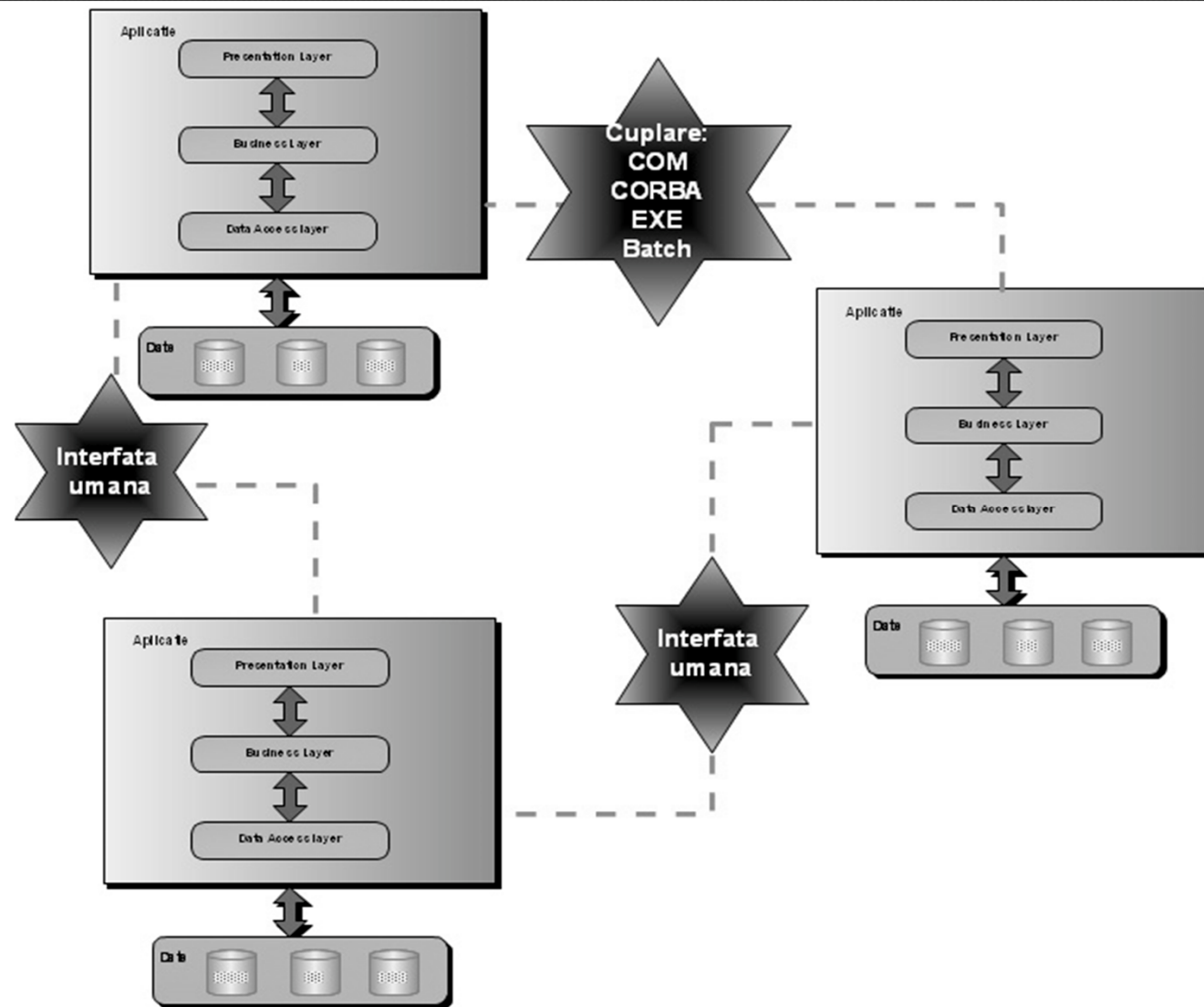
Arhitectura traditionala



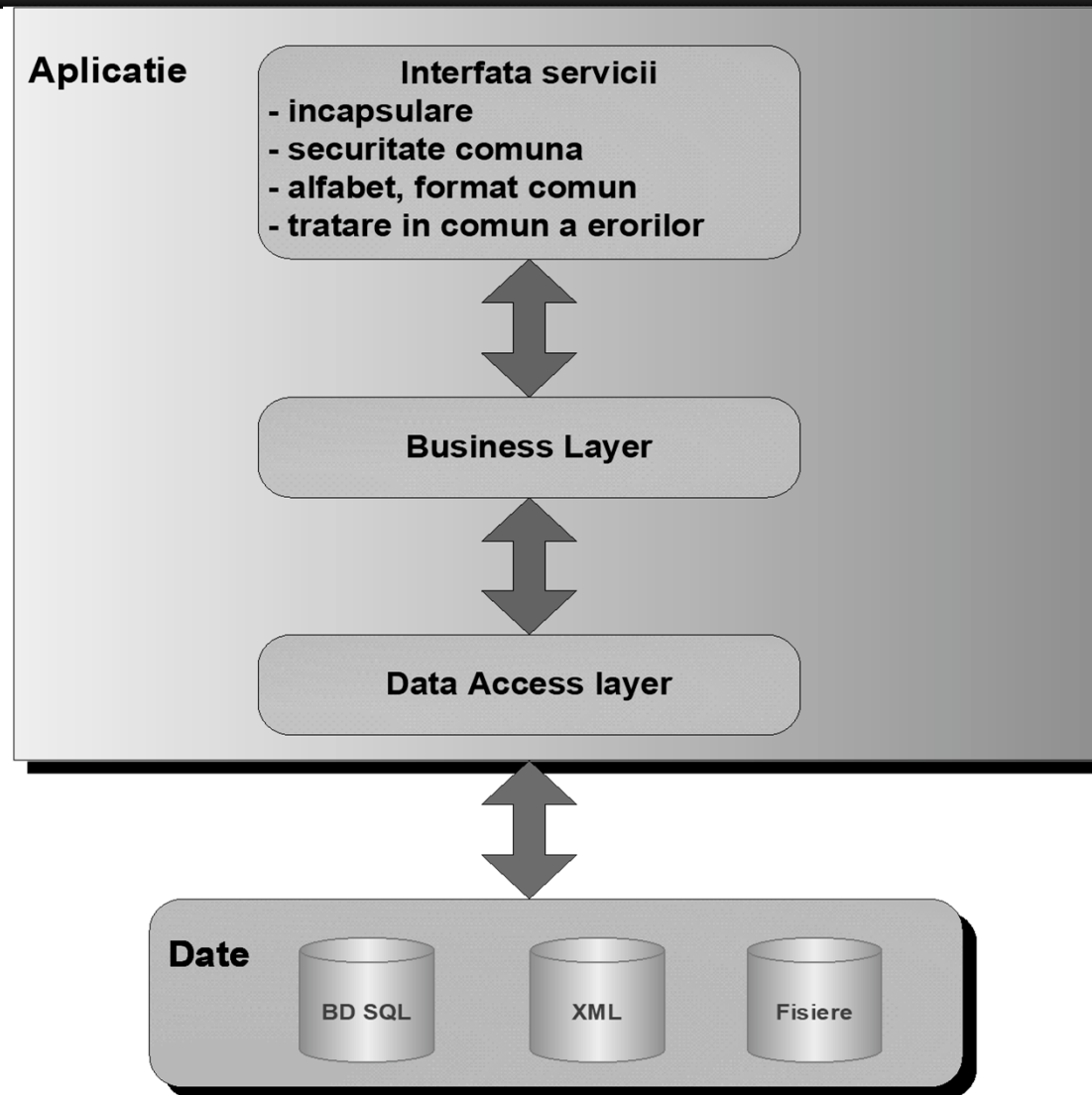
Arhitectura bazata pe componente



Probleme de implementare



Arhitecturi orientate spre servicii



Componentele SOA

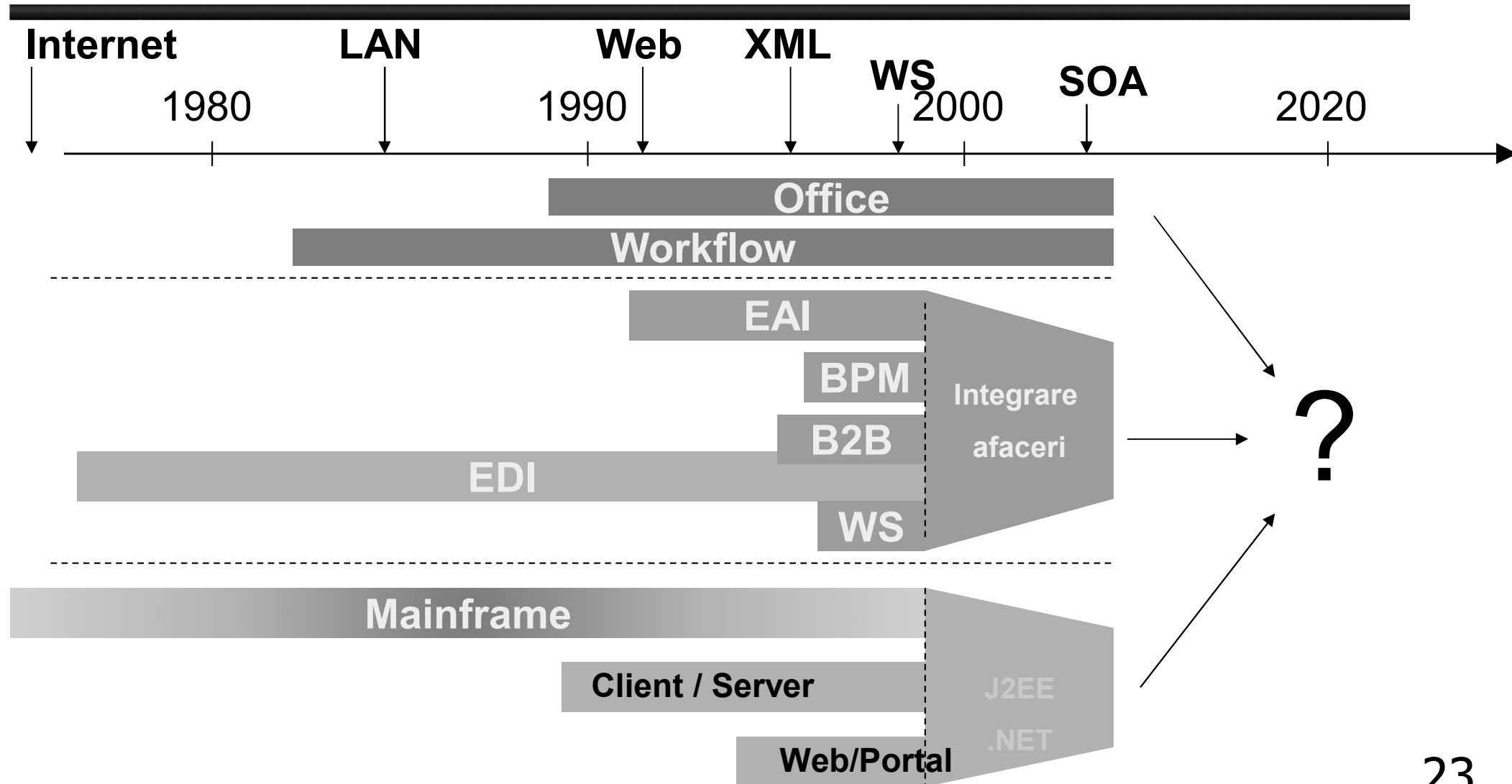
- **Serviciile in SOA au 3 proprietati fundamentale:**

- Independente de platforma
- Alocate si invocate dinamic
- Auto-mentinere

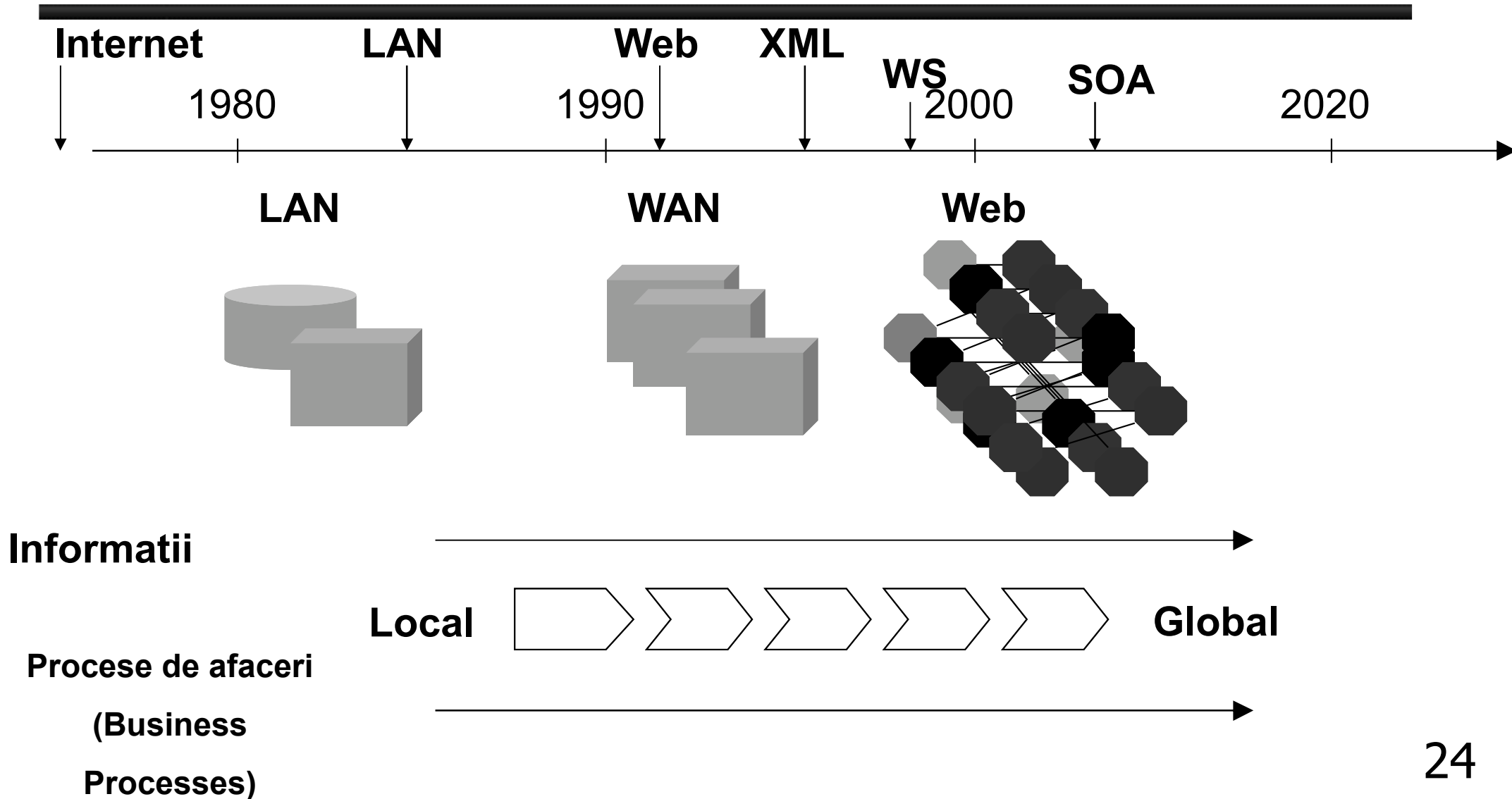
- **Mesaje** – Furnizorii si utilizatorii de servicii comunica intre ei prin mesaje.

- Independenta serviciilor de platforma si / sau limbaj: mesajele trebuie sa parvina aplicatiilor ce ruleaza pe orice platforma si sunt realizate in orice limbaj de programare.
- XML ofera functionalitatea, granularitatea si scalabilitatea solicitata de transmisia unui mesaj.

Convergente tehnologice



Conectivitate: procese globale si acces la informatie



Conectivitate: consecinte (1)

- **O aplicatie** NU mai ruleaza in cadrul unui singur sistem, pe o aceasi platforma si aferenta unei singure organizatii
- Se formeaza un continuum Obiect ... Document
- **Mesaje si servicii**
 - In opozitie cu conceptul de 'obiecte distribuite'
 - Interconexiunile sunt *peer-to-peer*
- Comunicatie **asincrona**
- **Concurenta** (la resurse) devine norma de baza

Conectivitate: consecinte (2)

- **Aliante si colaborari**

- ➔ In opozitie cu « integrare »

- **Limbaje**

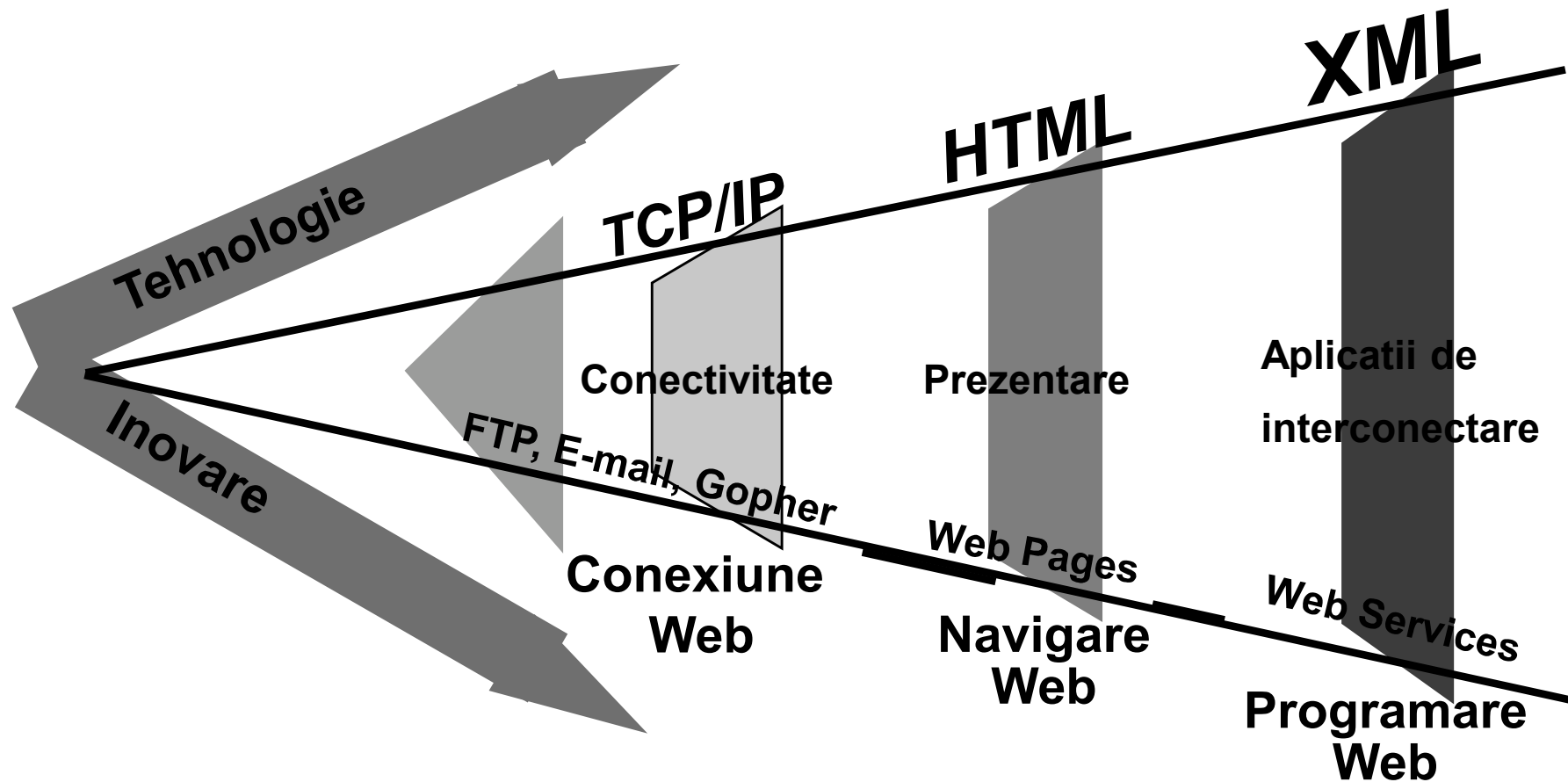
- ➔ Semantice (nu sintactice)

- ➔ Bazate pe modele si declaratii (nu procedurale)

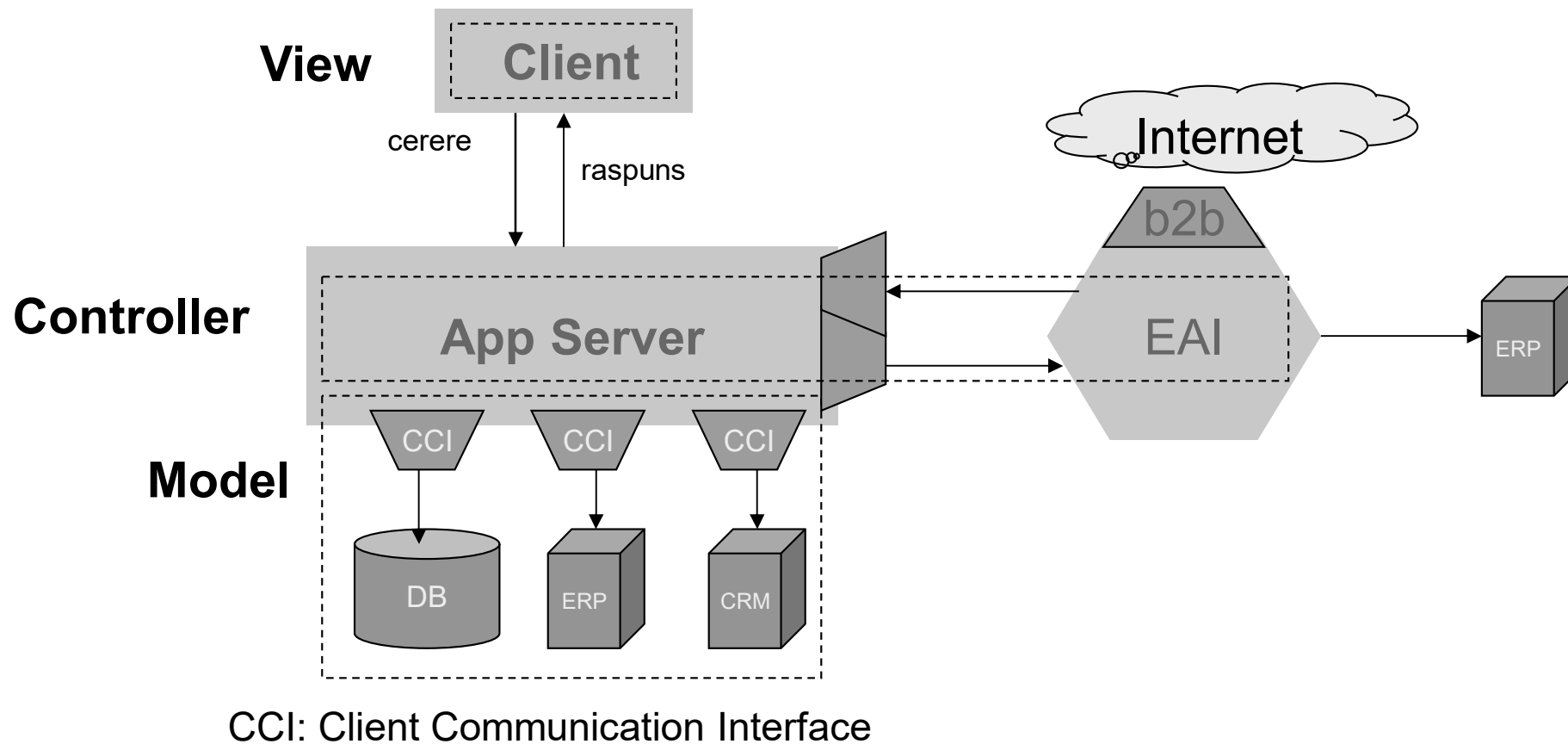
- Modele licentiate si operationale

- ...

Evolutie tehnologica



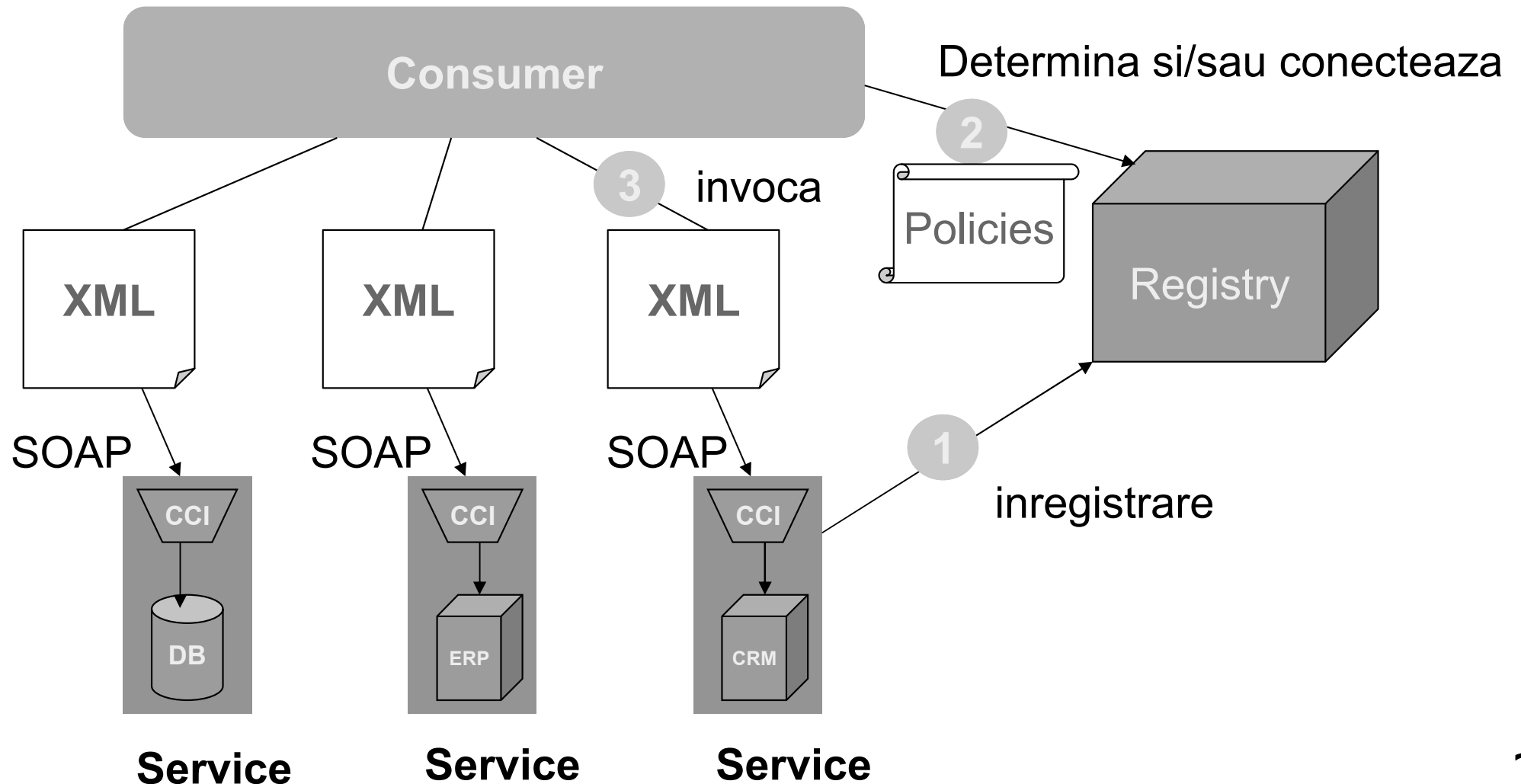
Softul in epoca web (J2EE, .Net, ...)



De ce un model nou ?

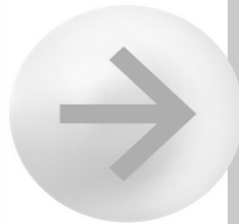
- Creșterea conectivității
 - ➔ D.e. J2EE, a fost proiectat pentru a dezvolta aplicații web 24x7
 - ➔ Ceea ce s'a și întâmplat !
- Există totuși o diferență față de afirmația: 'Vreau ca aplicațiile să execute *business logic* și în alte sisteme – deseori în conexiune dinamică cu sistemul în care mă aflu'
 - ➔ JCA (J2EE Connector Architecture) nu este suficient
 - ➔ Infrastructura EAI nu este suficientă

Service Running: in afara frontierei utilizatorului



Dela componenta la servicii (Web)

- Necesita o biblioteca client
- Configuratie C/S
- Extensibila
- Stateless
- Rapida
- Granularitate mica → medie



- Cuplare nelimitata prin
 - Schimb de mesaje
 - Politici
- Peer-to-peer
- Compozabile
- Independent de context
- Mai costisitoare
- Granularitate medie → mare

Servicii Web

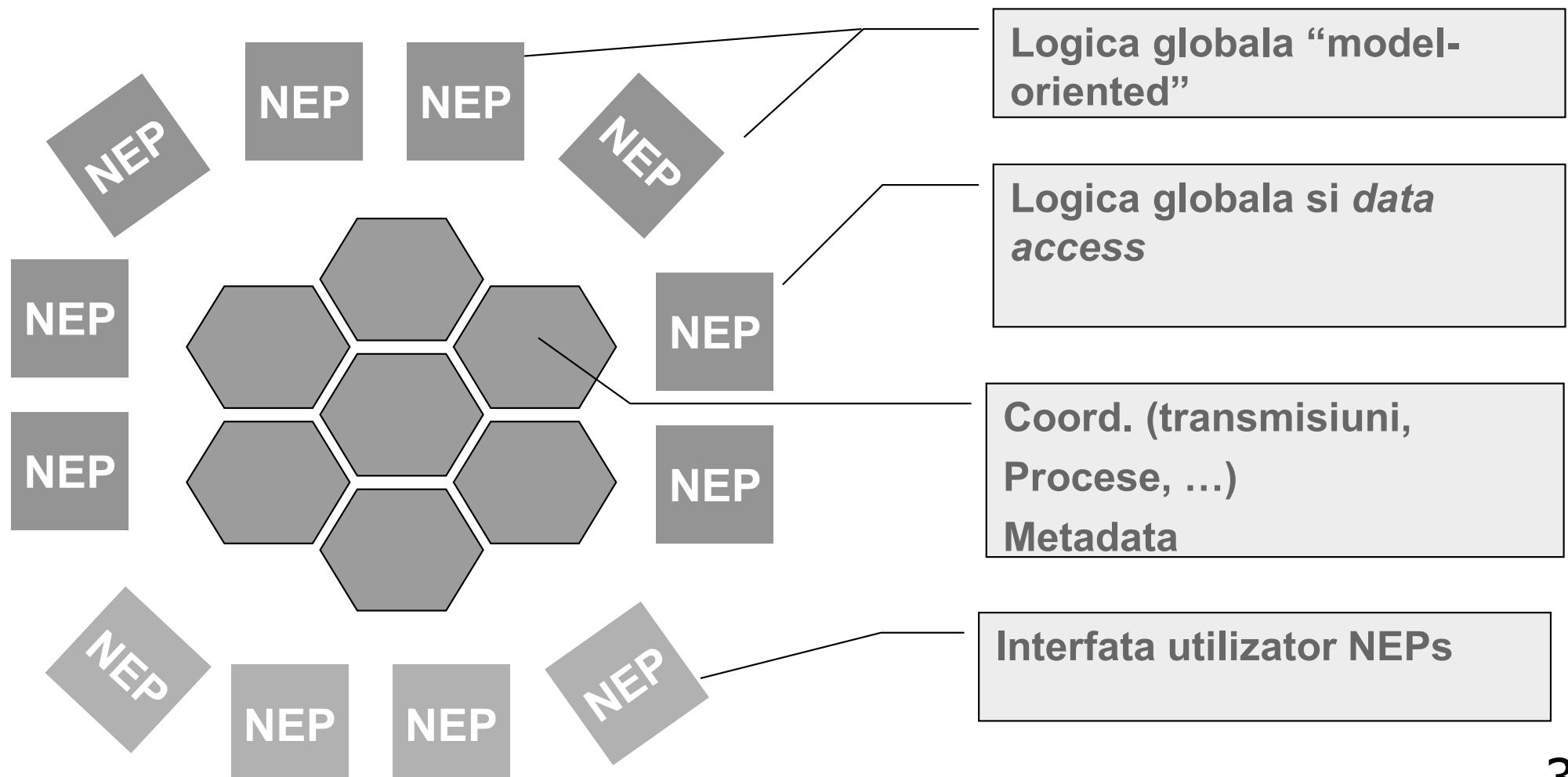
- Cuplare nelimitata

- Serviciile web nu necesita CCI (*Client side Communication Interface*)
 - Utilizarea unui serviciu implica putine componente
- Serviciile web pot accepta mesaje pe care nu le inteleg / suporta in totalitate
 - XML, WSDL
- Serviciile web se regasesc in limite (frontiere) foarte largi
 - Localizarea este independenta de mecanismul de invocare
 - *Directories*

- Interactiile *peer-to-peer* sunt posibile

- O interactiune de tip cerere / raspuns este in general limitativa
- Intr'o structura bazata pe servicii, este dificil de diferentiat clientul de server

Drumul spre SOA (1)



Drumul spre SOA (2)

- **Scop**

- Organizarea logicii globale independent de modul de implementare
 - In mod normal, logica globala "impacheteaza" serviciile web

- Re-implementarea *controller-ului* cu tehnologii de coordonare

- ...Intreaga filozofie de dezvoltare a aplicatiilor trebuie re-inventata

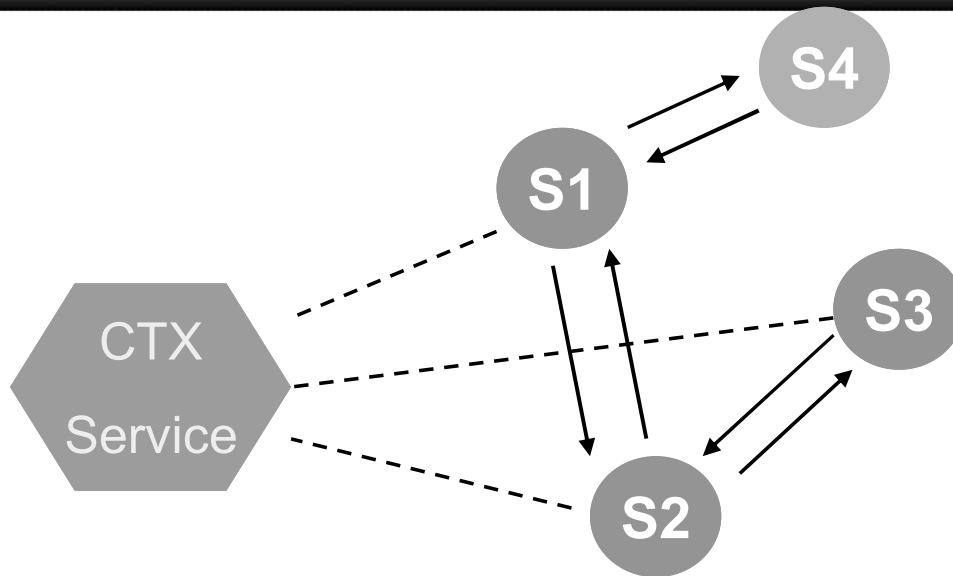
Nivelul de coordonare

- Exista multe concepte neincluse in structuri / arhitecturi:
 - ➔ *Composition*
 - ➔ *Orchestration*
 - ➔ *Choreography*
 - ➔ *Collaboration*
 - ➔ ...
- Care sunt relatiile intre toate aceste elemente?
- OASIS/WS-CAF
 - ➔ Management contextual
 - ➔ Coordonare
 - ➔ Managementul tranzactiilor

Nivelul de coordonare

- Exista multe concepte neincluse in structuri / arhitecturi:
 - ➔ *Composition*
 - ➔ *Orchestration*
 - ➔ *Choreography*
 - ➔ *Collaboration*
 - ➔ ...
- Care sunt relatiile intre toate aceste elemente?
- OASIS/WS-CAF
 - ➔ Management contextual
 - ➔ Coordonare
 - ➔ Managementul tranzactiilor

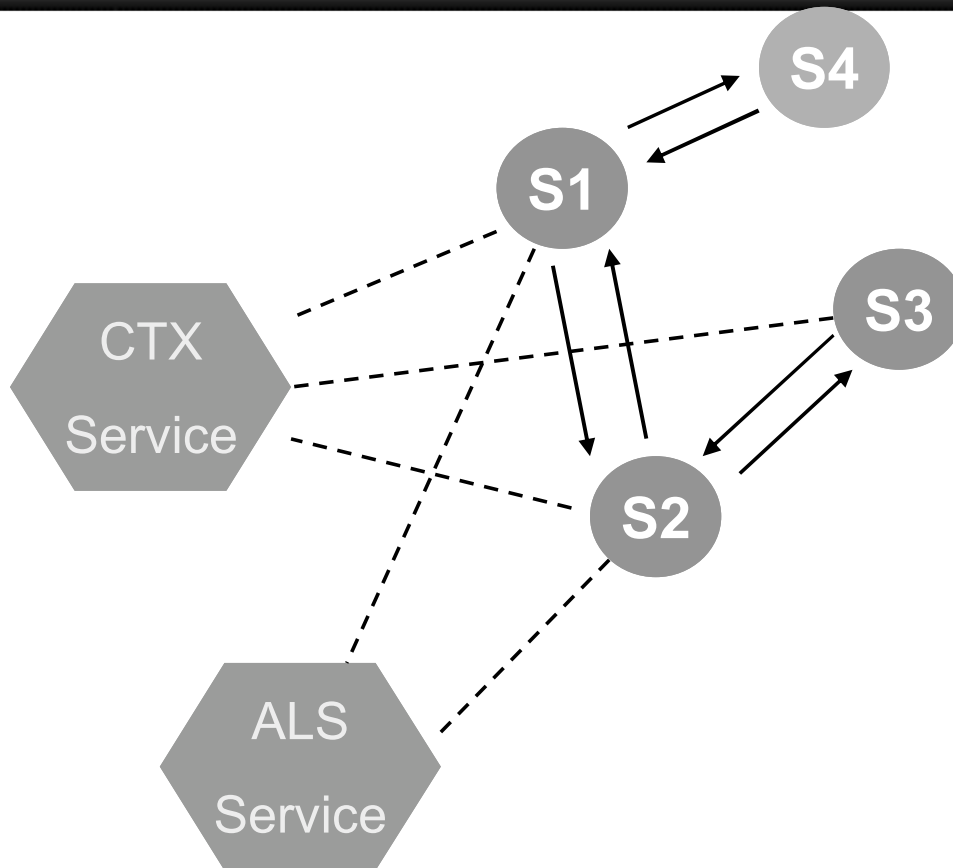
Ce este contextul ?



Interactiunile *peer to peer* determina un service contextual

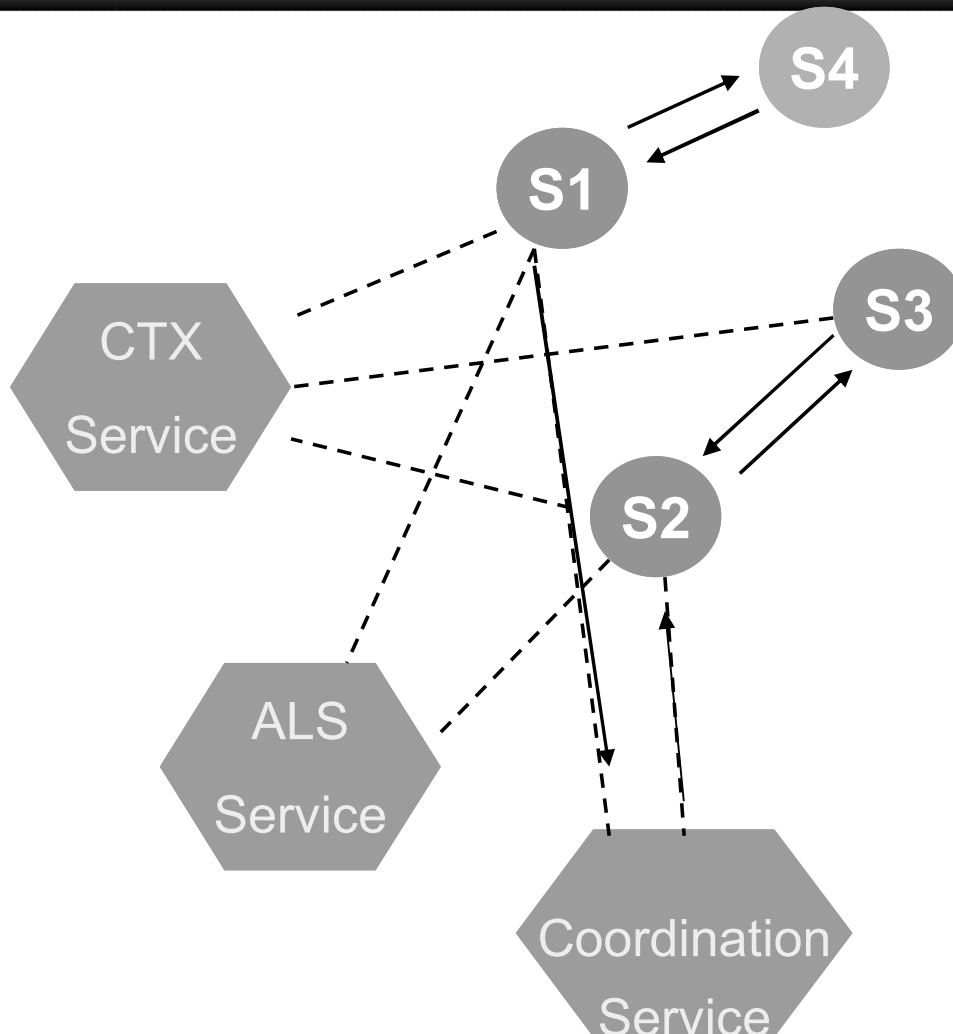
(d.e. S3 va trebui sa cunoasca nivelul interactiunii dintre S1 si S4 pentru a'si putea livra serviciile

Ce este o activitate *Lifecycle Service* (ALS)?



ALS permite demarcarea unitatilor de lucru *shared* intre diverse servicii

Coordonare



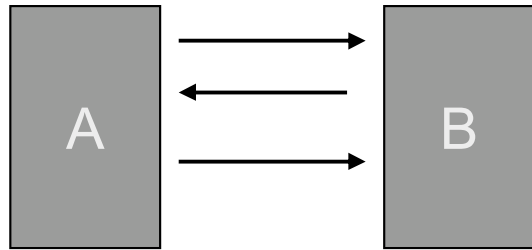
Un coordonator reprezinta un component activ intr'o arhitectura

Poate coordona servicii sau poate furniza el insusi servicii

Mai multe obiective:

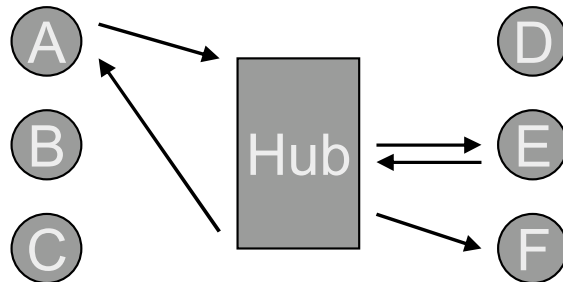
- ☐ Tranzactii
- ☐ Orhestrare
- ☐ compunerea de secvente de activitati (*Choreography*) ...

Topologii de coordonare



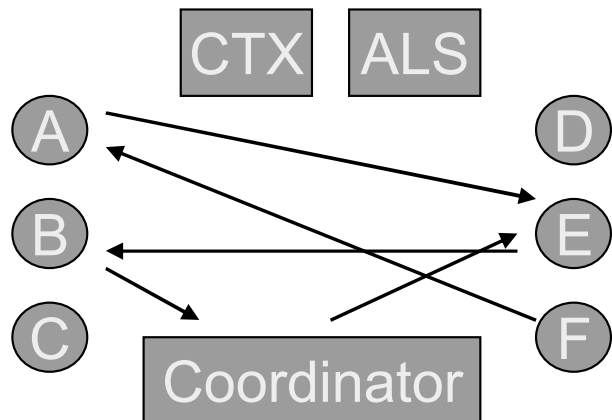
Relatii binare

- Context si Activitate sunt implicite
- Autocoordonare



Relatia Hub and Spoke

- Context si Activitate sunt coordonate de catre hub
- Coordonarea este controlata in exclusivitate de catre hub



Relatii multidimensionale *peer-to-peer*

- Context si Activitate sunt explicite
- Context, ALS si Coordination sunt controlate de catre *fabric*

Coordonare: concept abstract

- Conexiune cu *fabric services* pentru stabilirea tipurilor de coordonare
 - Nu orice este un proces...
- Pot fi compuse coordonari de diferite tipuri
 - O tranzactie poate include definirea unei orchestrari (ca si componenta a unei activitati)
 - Definirea unei orchestrari poate contine numeroase tranzactii

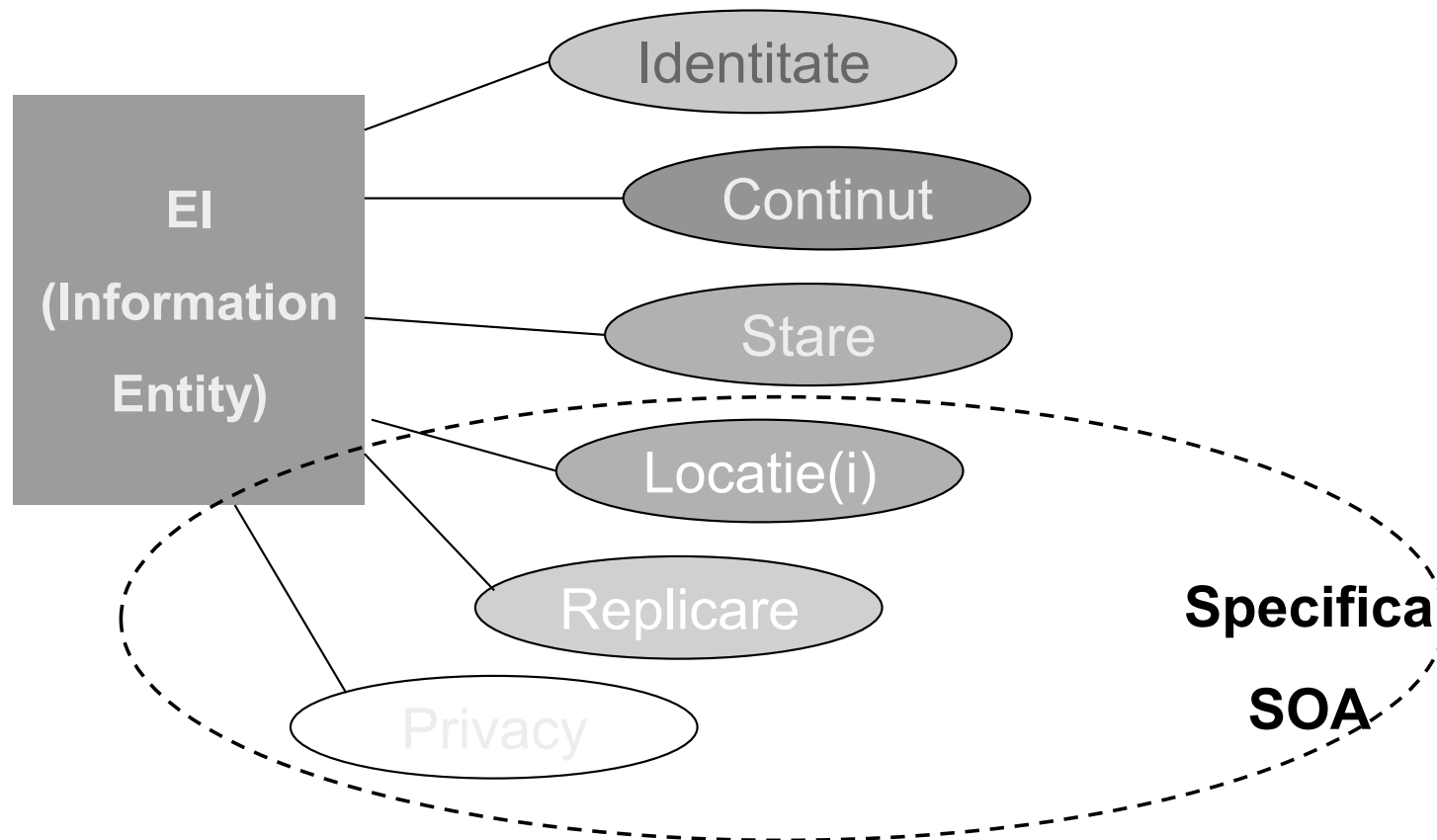
IE – Information Entity in SOA

- “in miezul aspectelor legate de Web services exista o problema foarte complexa: impreuna cu aplicatiile distribuite apare si necesitatea datelor distribuite”
 - Identificari si echivalente
 - autentificare
 - Autorizare
 - mediere
 - sincronizare

Sursa: The Dataweb: An Introduction to XDI, Drummond Reed *et al.*

IE in SOA

- Atunci cand se gestioneaza un agregat informational (*Information Aggregate*) intr-o SOA, trebuie luate in considerare mai multe dimensiuni



Principalele probleme

- Izolarea (*Isolation*)

- Nu putem garanta ca informatia pe care o detinem este informatia pe care o are si sistemul de inregistrari (*the system of record*)

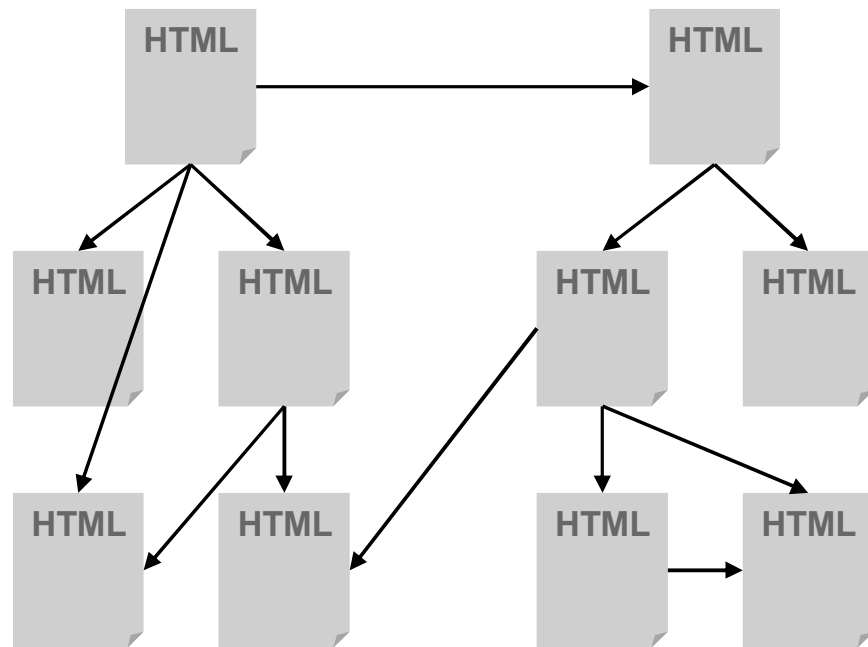
- *Containment*

- Nu putem garanta ca un utilizator al serviciului va aplica acelesi reguli de securizare (*privacy*) informatiei ce ii va fi oferita

Standarde Web *vs.* standarde Dataweb

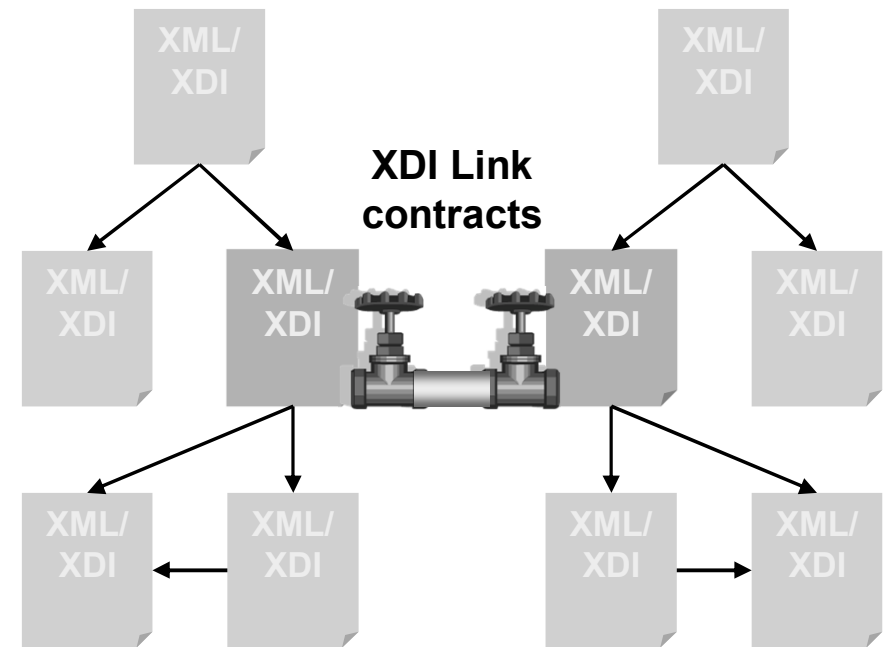
	Web	Dataweb
100% adresabilitate a resurselor	URIs	XRIs
Reprezentare si linking format comune	HTML	XML/XDI
Protocol de transfer (interchange) comun	HTTP	XDI/HTTP XDI/SOAP

Websites vs. Dataweb sites



**Website
A**

**Website
B**



**Dataweb site
A**

**Dataweb site
B**

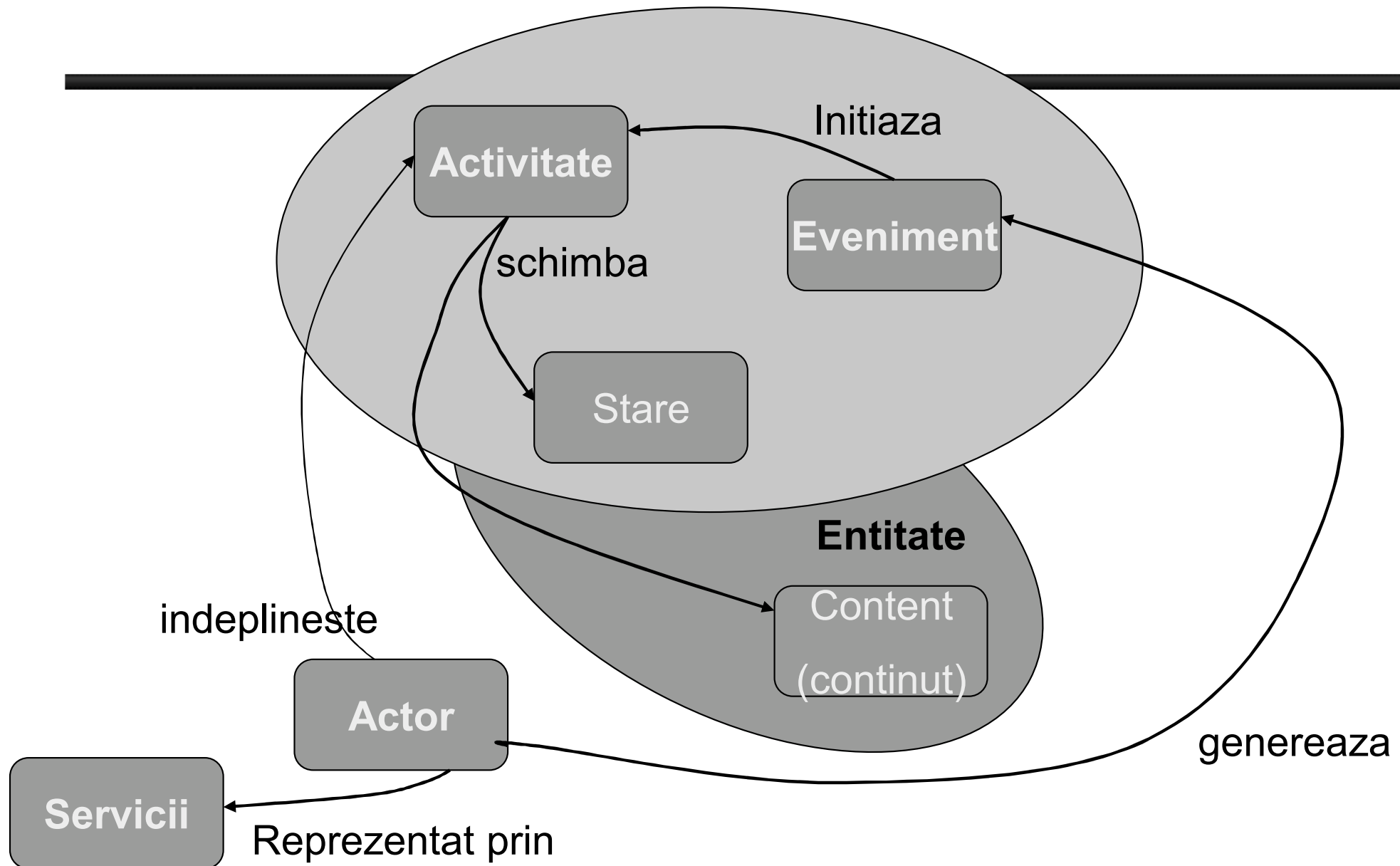
Elemente si Agregate Informationale

- Pana acum abia daca am "zgariat" suprafata problemei...
- Ideea ca am putea merge mai departe spunand ca pur si simplu schimbam mesaje intre servicii este eronata
- SOA nu ar exista fara conceptul sau de entitate informatională (information entity)
 - Evident, *entity beans* nu au reprezentat o solutie prea buna
 - .NET ofera conceptul de *DataSet* care pare mai adecvat

SOA si BPM

- SOA se refera la elaborarea de componente software care pot fi reutilizate in contexte necunoscute in momentul proiectarii
 - ➔ Compunere vs. Extensie (OO)
- BPM se refera ca capabilitatea de a modela precis si eventual a schimba contextul in care sunt utilizate componentele intreprinderii
- **Cum pot fi reconciliate aceste doua concepte ?**

Elementele BPM



BPM *vs.* SOA

- Doua abordari

- ➔ **Event Oriented (orientata eveniment)**

- BPML, BPEL
 - Pi-Calculus (si Event Calculus)

- ➔ **Activity Oriented (orientata activitate)**

- WfMC
 - Petri nets (Retele Petri)

- Cele doua abordari trebuiesc combinate si conceptul de stare trebuie sa faca parte din model

- Conceptul de "Turing complete" reprezinta o scuza pentru o abordare "pura" (i.e. event oriented)

BPEL

-
- “BPEL este un limbaj XML language pentru definirea compunerii serviciilor (web) in alte servicii (noi)” (Paul Brown, FiveSight)
 - BPEL este mai curand un simplu limbaj de orchestrare si nu un limbaj BPL (Business Process Language)
 - BPEL cere ca fiecare proces:
 - Sa aiba un “centru” de executie
 - Sa fie compus dintr-o multime vasta de definitii de orchestrare care interactioneaza intre ele
 - In pi-calculus, chiar si o variabila este un proces...
 - BPEL porneste de la premisa ca procesele business pot fi complet continute intr-o singura definitie care sa includa toate exceptiile (exception paths) posibile
 - Nu este sigur ca este intotdeauna o premisa corecta...

Business process

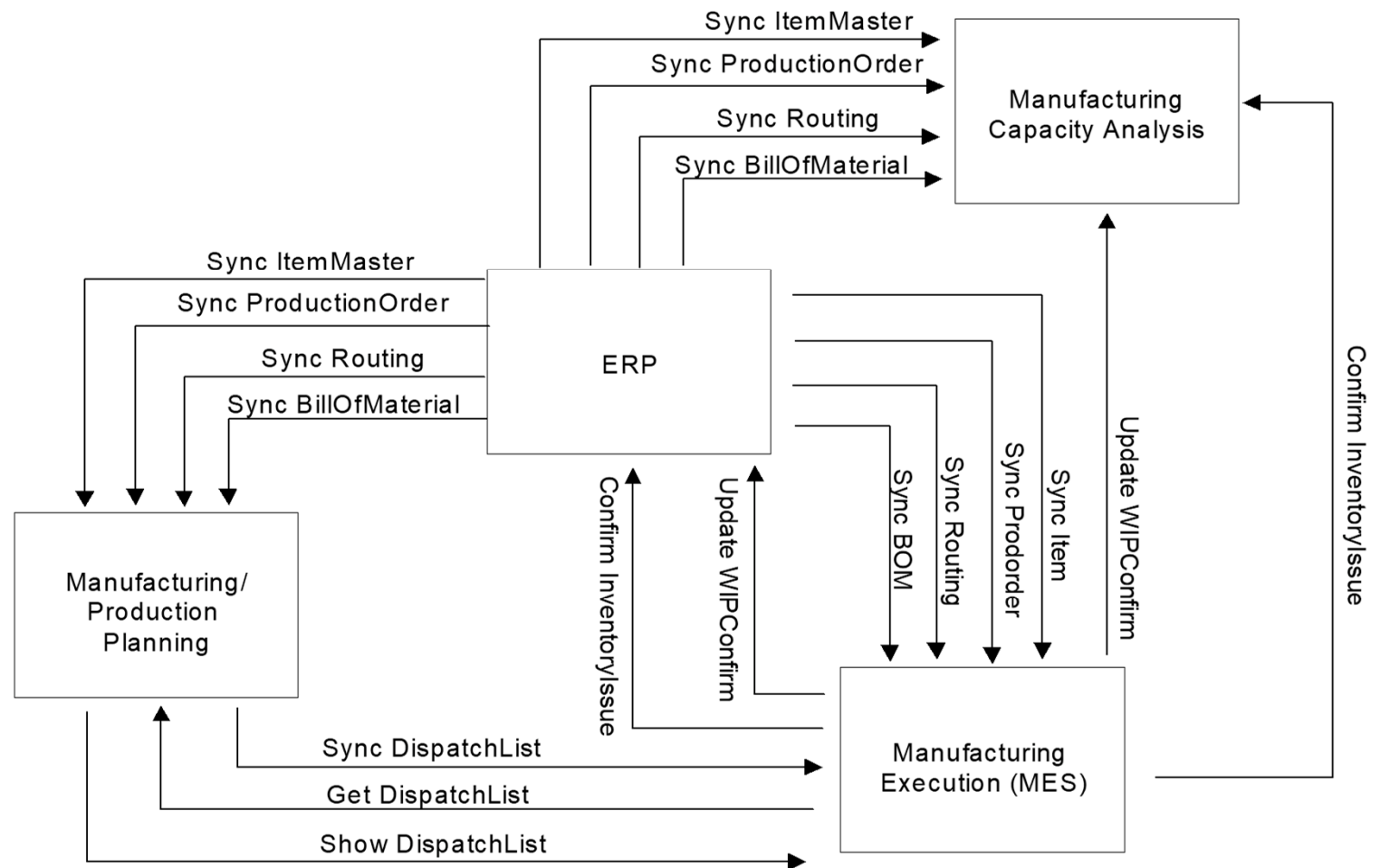
Un business
process

nu are un
“centru”...

de facto este
“peer-to-peer”

OAGIS 8.1

Scenario 50



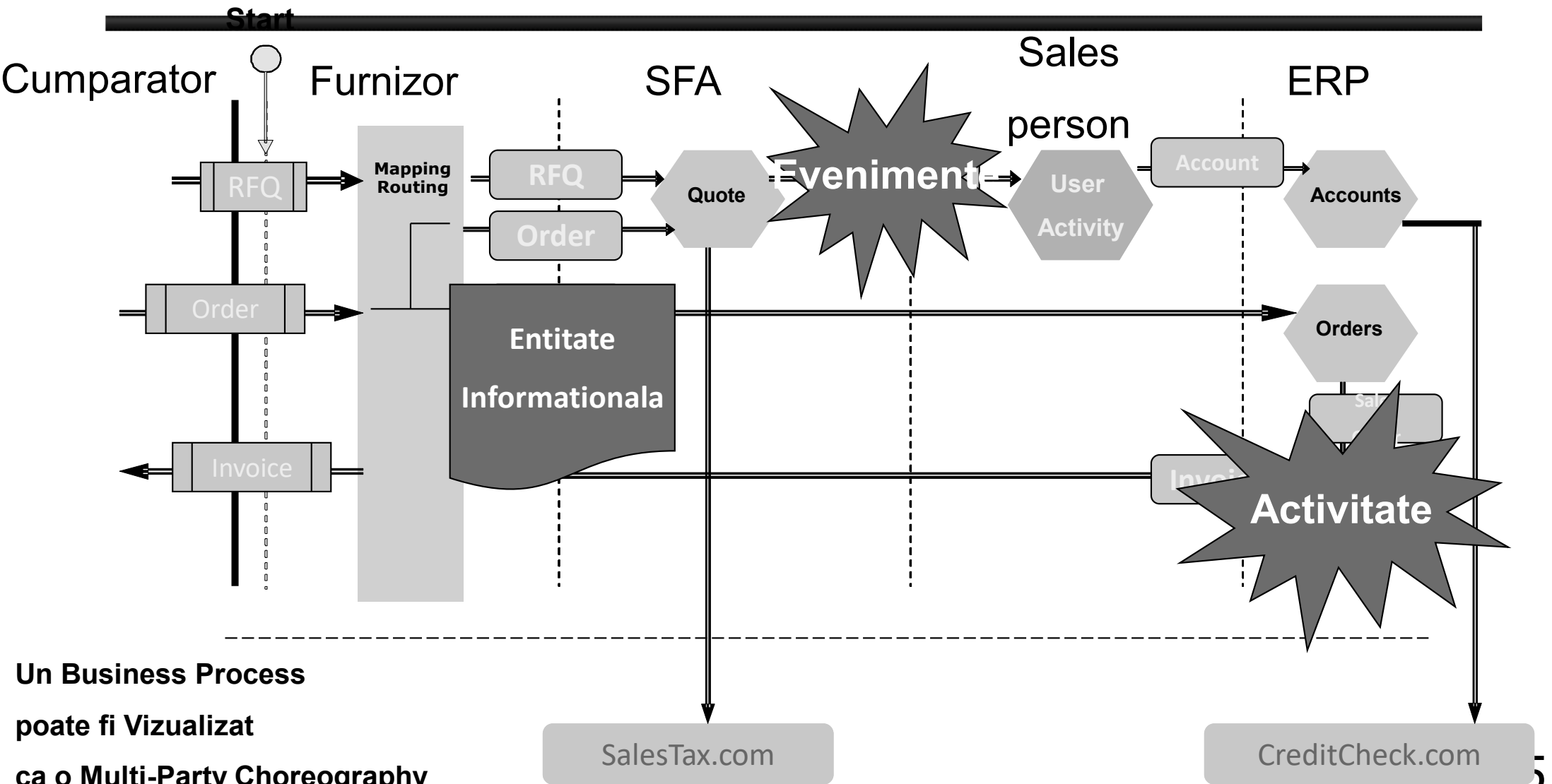
Exemplu

- Un cumparator comanda niste bunuri de la un furnizor
- Furnizorul parcurge o serie de pasi pentru indeplinirea comenzii
 - ➔ Accepta comanda
 - ➔ Actualizeaza sistemul de primire comezi (the order entry system)
 - ➔ Actualizeaza sistemul de incasari (the billing system)
 - ➔ ...

Limbaje de orchestrare

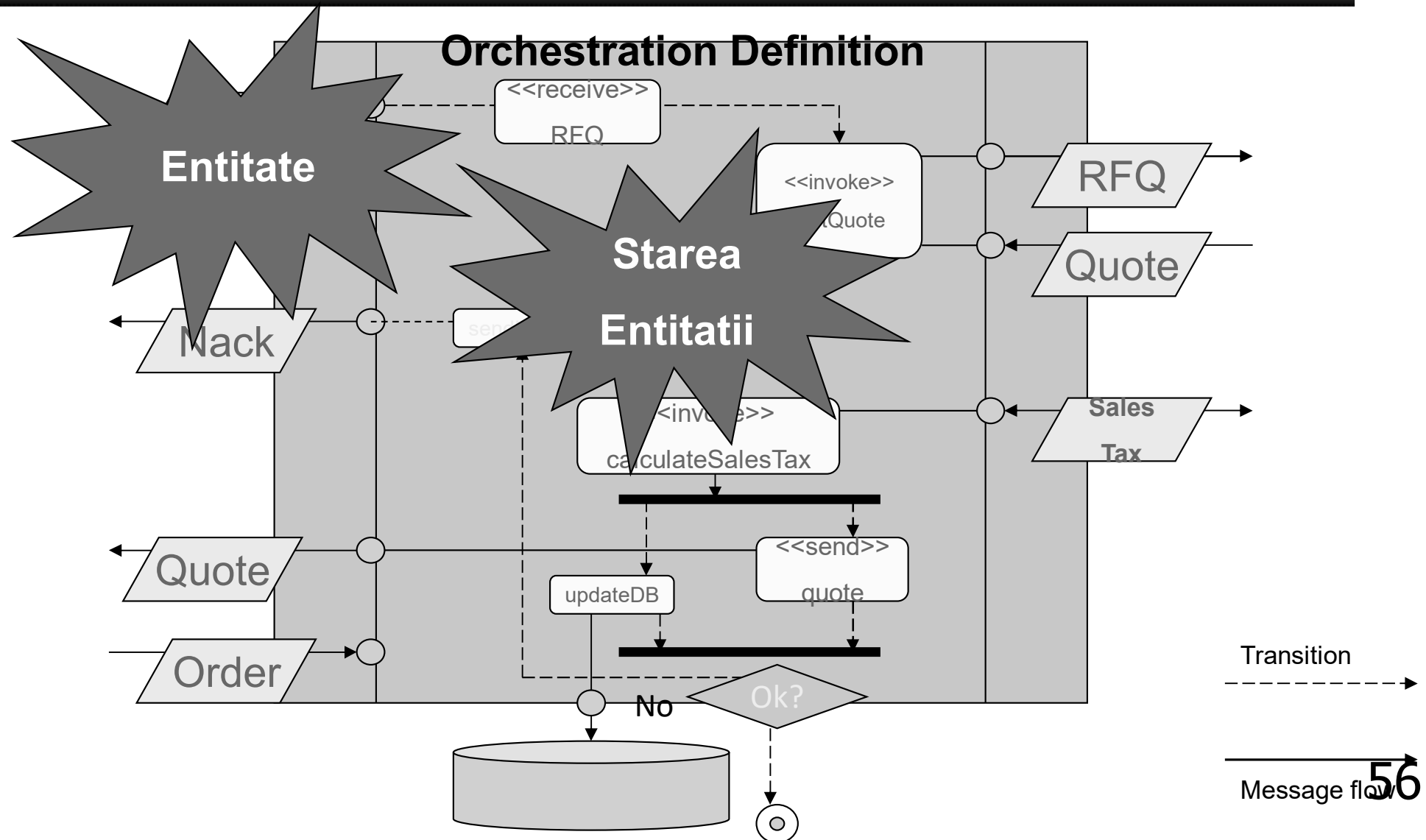
- Ele permit implementarea serviciilor complexe care implica:
 - Rulare pe termen lung (de la cateva ore la cateva luni),
 - Logica de tip evolutie bazata pe evenimente (event driven)
- Nu se refera la modelarea proceselor de business prin ele insele
 - Diferite orchestratii (i.e. servicii diferite) pot rula in cadrul aceluasi business process
 - Si *vice versa*

Vizualizarea unui *Business Process*

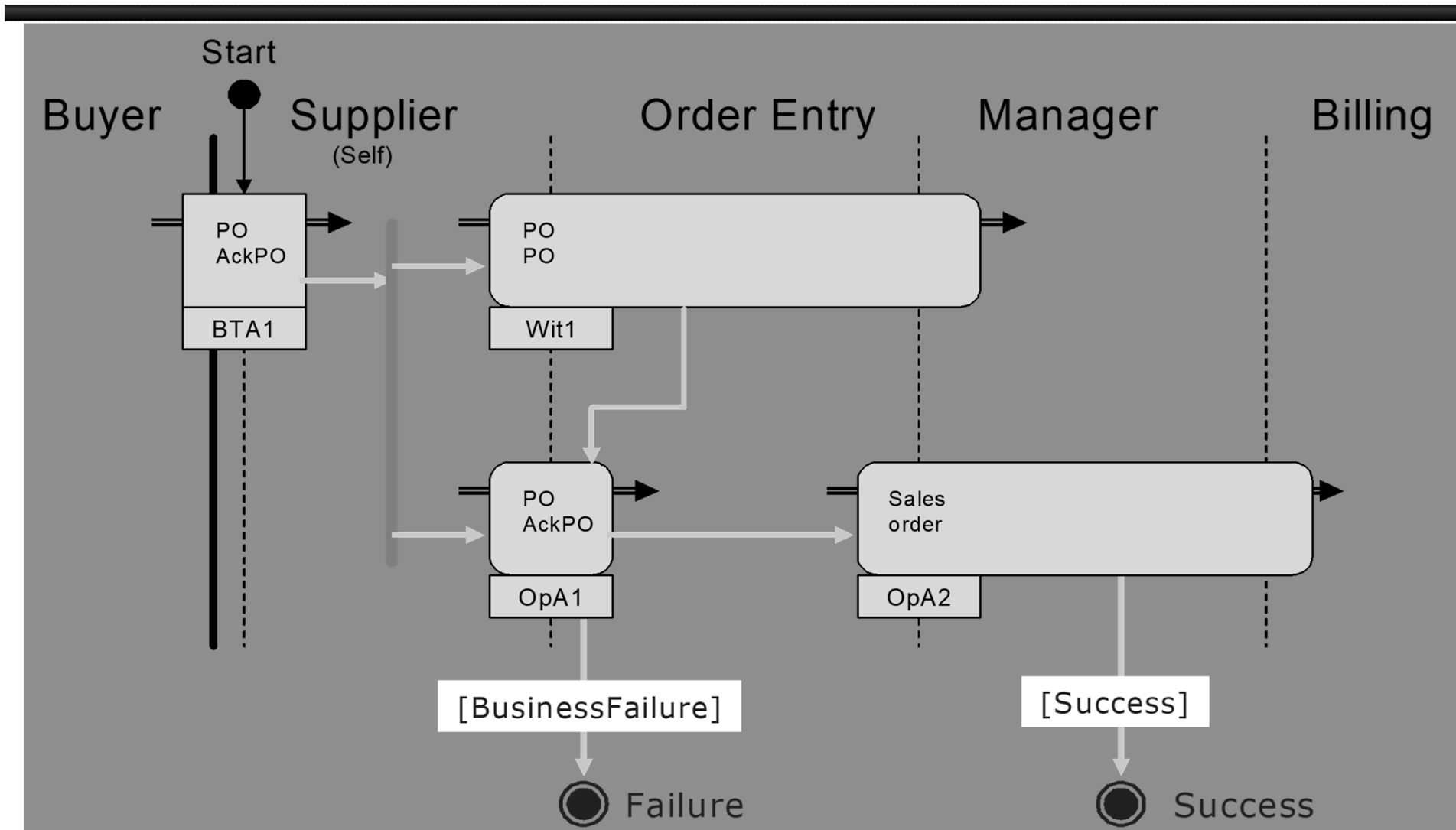


Serviciile SOA

Quote Service



Coregrafia



57

O coregrafie (*Choreography*) furnizeaza un model al fluxului de evenimente intre activitati

Cateva concluzii

- **Orchestrarea**

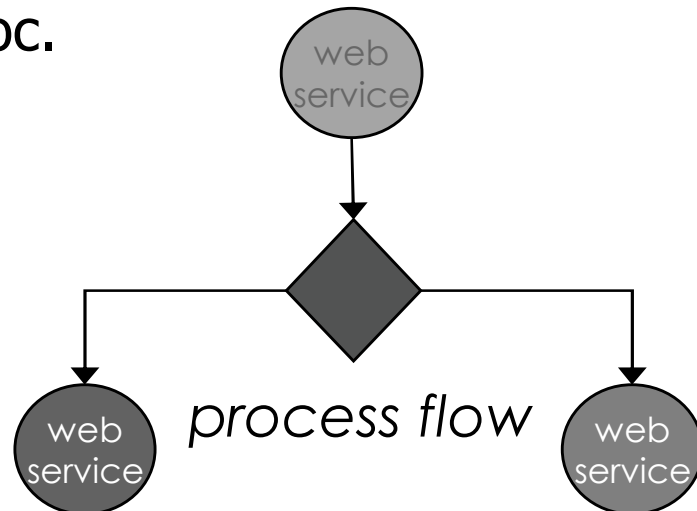
- « ... este un concept emergent care ofera **analistilor** o metoda de a descrie formal procese care stau la baza aplicatiilor business, astfel incat sa poata fi reprezentate si legate cu procese din cadrul altor aplicatii »

- **Coregrafia (Choreography)**

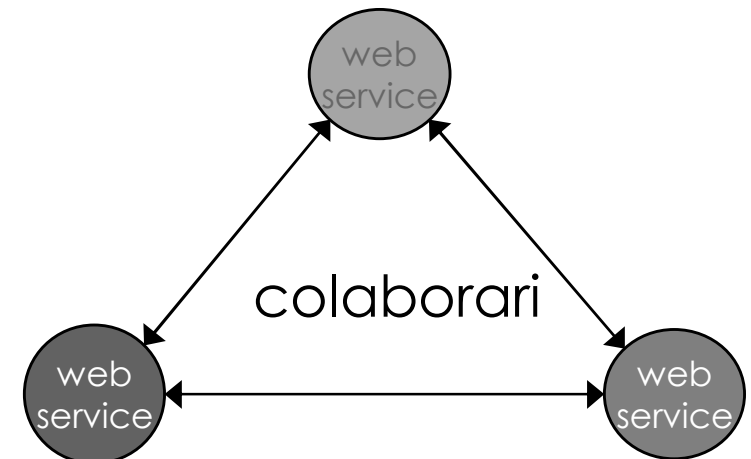
- Este un concept care evidentiaza modul in care aceste procese sunt interconectate in cadrul intreprinderii
- Coregrafia poate fi « activata » atunci cand sunt necesare maparea si rutarile
- Ambele concepte reprezinta forme ale **Coordonarii**

Orchestrare vs. coregrafie

- Include activitatile rezultate in urma interactiunilor dintre servicii
- Descrive *process flow*
- Poate include servicii web interne / externe, dar procesiul este controlat dintr'un singur loc.



- Inregistreaza secventa mesajelor (cu mai multe surse / componente)
- Se asociaza cu schimbul public de mesaje (NU cu procese executabile)
- Accent pe colaborari



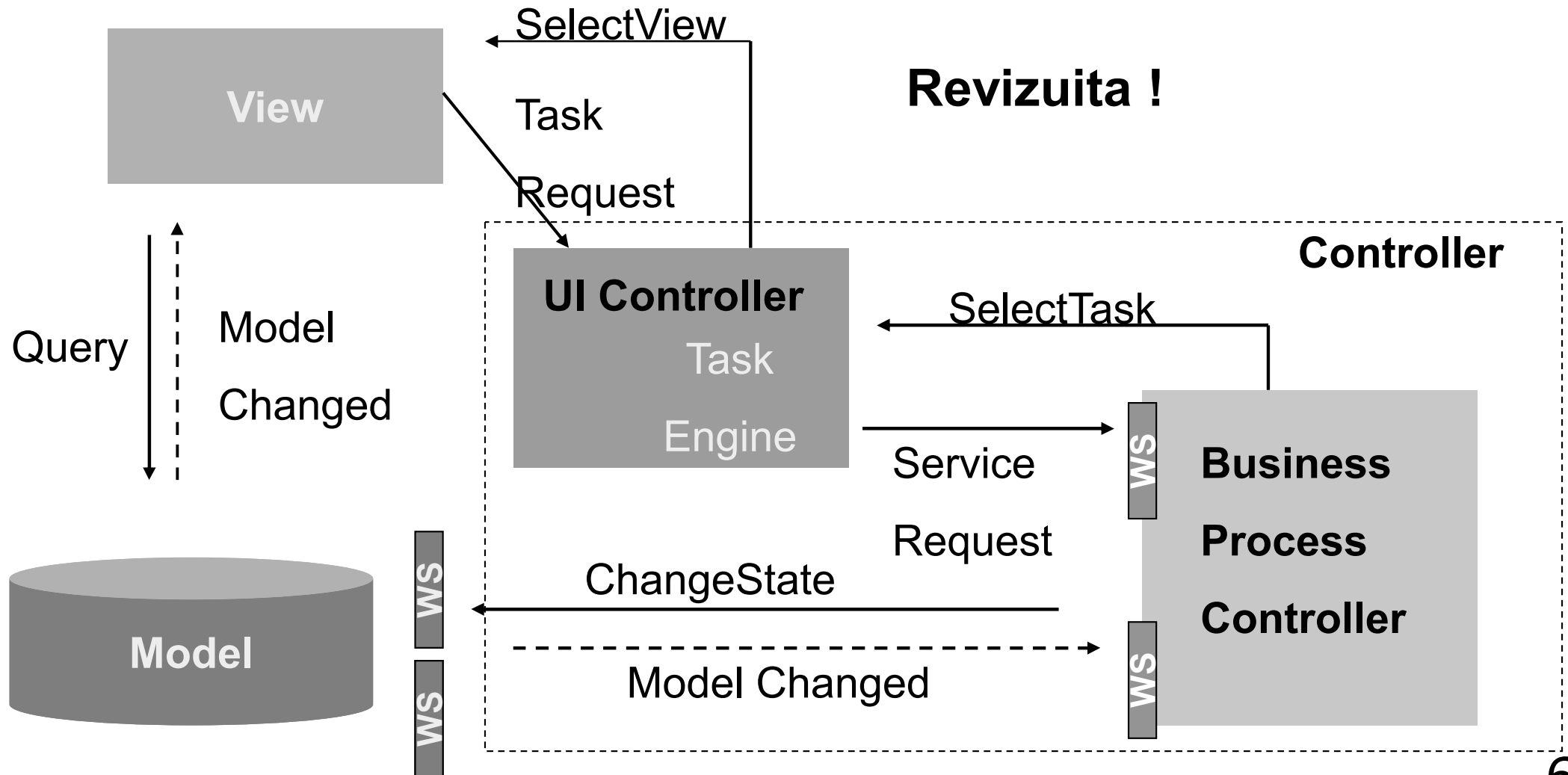
BPM + SOA

- Necesita un suport teoretic puternic
 - Lipseste inca o "piesa" importanta: "starea"
 - Comutarea de la orchestrare la *Business Process Definition*
- Odata stabilit un fundament teoretic, vom vedea ca apar Domain Specific Languages (*DSL*), care sunt mult mai apropiate de modul in care gandesc utilizatorii (si nu analistii) despre procesele de business (*Business Process*)

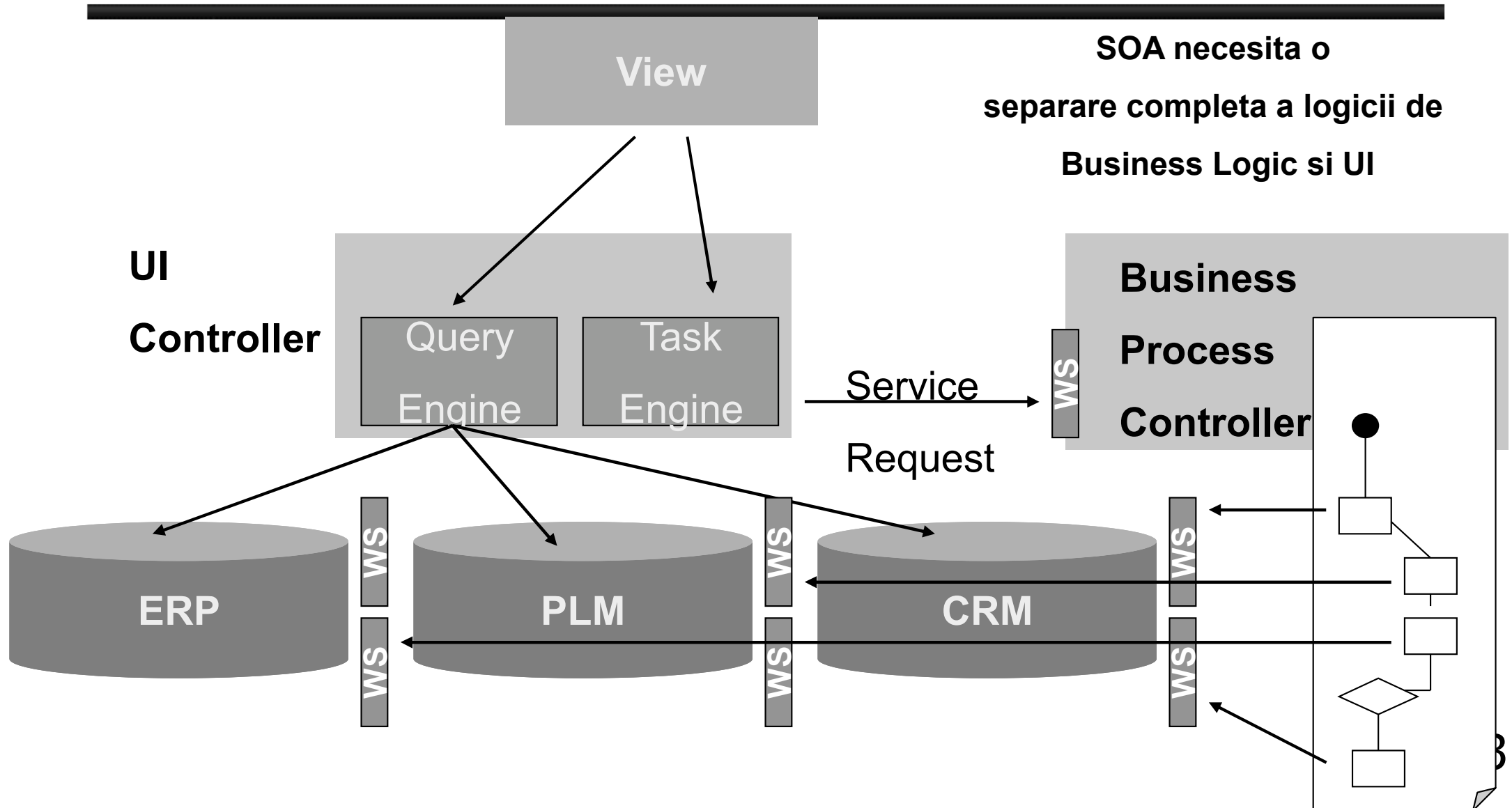
Un Model Tehnic pentru elaborarea de *Software* in SOA

- Trebuie sa adaptam fundamentul teoretic la arhitectura moderna a aplicatiilor
 - Model-View-Controller
- Model → Servicii
- Controller → Coordonare
- View → ?

Structura *Model-View-Controller*



Business Logic si UI

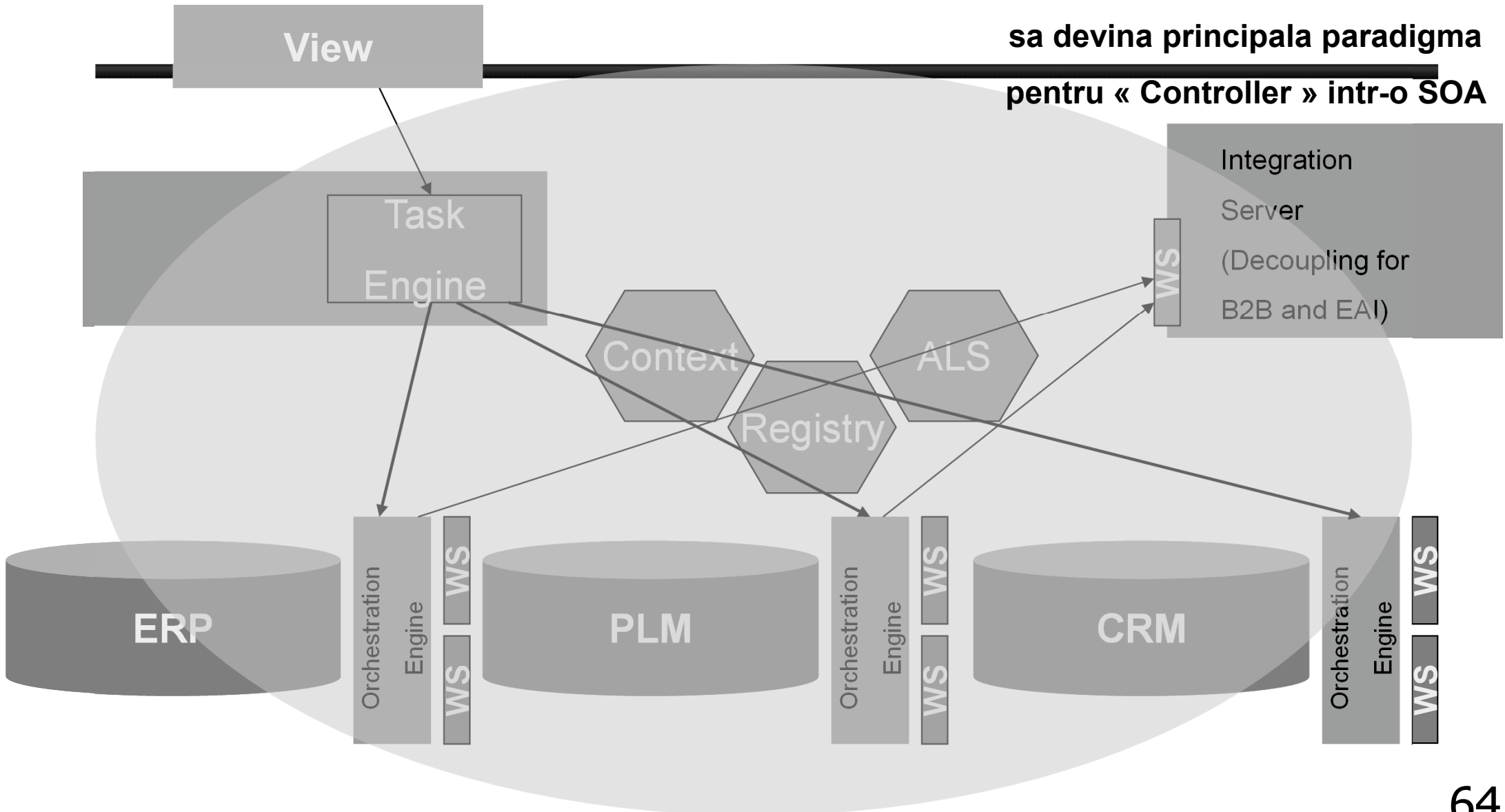


BPM si SOA

Business Process Execution

Este posibil ca BPM

sa devina principala paradigma
pentru « Controller » intr-o SOA



O noua paradigma: *Service Orientation*

- Nu este ca un nume nou pentru
 - ➔ API
 - ➔ Componente
- Este o multime de concepte valide cu care se poate construi un nou tip de software
 - ➔ Este cel puțin la fel de semnificativ ca cel de "Object Orientation"
- În particular, *Service Orientation* (SO) ne forțează să ne gândim la modul în care o aceeași secvență de cod poate fi utilizată (leveraged)
 - ➔ De un număr mare de consumatori
 - ➔ Într-un context neprevăzut

“SOA este inca departe ...”

- Inca mai trebuiesc precizari legate de ceea ce inseamna SOA cu adevarat
 - Am prezentat pana acum trei concepte, dar mai sunt inca multe si inca si mai multe aspecte inca neacoperite pana astazi
 - BPM si SOA sunt complementare
- Este necesara multa munca pentru a obtine si furniza adevarata “valoare SOA” si a trece dincolo de aplicatiile “de jucarie” ale SOA
 - In cel mai bun caz, suntem capabili sa oferim servicii web
- Standardizarea in domeniu reprezinta in acelasi timp un blestem si o binecuvantare
 - Deschide calea, dar aduce “ceata” si confuzie, datorita lipsei de ...coordonare

Noi directii...

- ... o noua "goana dupa aur" pentru a detine si a publica aplicatii de tip "servicii"
 - Aplicatiile de tip Mainframe si Client Server (ERP, CRM, SCM...) vor fi afectate in mod dramatic
- ... software mult mai "bogat" si mai "istet"
 - Ceea ce poate deveni si un cosmar, avand in vedere toate problemele de securitate care apar in ziua de astazi
- In orice caz, anumite tehnologii cheie inca lipsesc ...
 - "Convergenta" standardelor extrem de importanta