

Integrarea sistemelor informatice



Suport curs nr. 8

Programator >> Arhitect

Procesul de integrare – Modelarea domeniului

2024-2025

Obiective

- Înțelegerea diferitelor aspecte ale integrării
 - Puncte de vedere
 - Niveluri de abstractizare
 - Modele de dezvoltare
- Înțelegerea conceptului de modelare a domeniului
 - Identificarea domeniilor de modelare
 - Utilizarea modelelor corespunzătoare nivelului de abstractizare

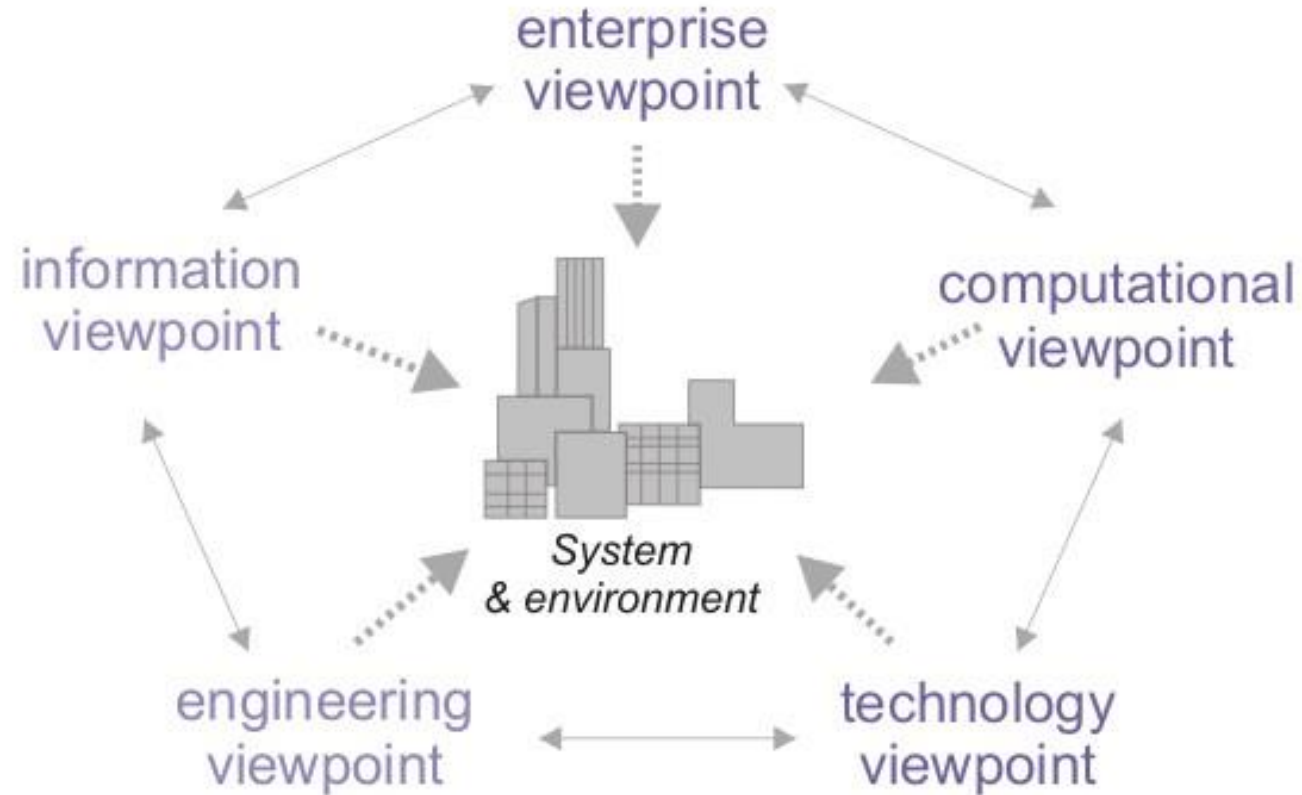
Modele de dezvoltare a sistemelor

Conform ISO Reference Model for Open Distributed Processing (RM-ODP)

| Modelul conceptual | Modelul tehnic |
|---|--|
| <ul style="list-style-type: none">• descrie sistemul în termeni de business• prezintă componentele computaționale și informațiile asociate• prezintă constrângerile• poate fi la diferite niveluri de abstractizare și de detaliu | <ul style="list-style-type: none">• specifică în termeni tehnici cum se implementează sistemul• prezintă deciziile legate de tehnologiile folosite• arată mecanismele care se vor folosi și detaliază modul lor de folosire |

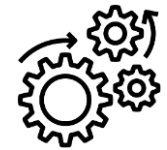
Puncte de vedere

Conform ISO **Reference Model for Open Distributed Processing (RM-ODP)**



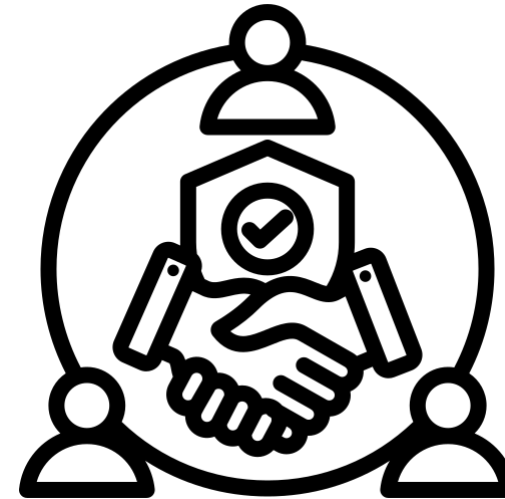
Puncte de vedere

- **punctul de vedere de întreprindere** – surprinde comportamentele relative la contextul **organizației**
- **punctul de vedere informațional** – surprinde **informațiile**, întrebuințarea și interpretarea lor
- **punctul de vedere computațional** – surprinde descompunerea sistemului în **componente** specifice care interacționează între ele prin **interfețe**
- **punctul de vedere ingineresc** – care dezvoltă **mecanisme** menite să suporte interacțiunea între componente
- **punctul de vedere tehnologic** – care alege cât mai precis tipul de tehnologie pentru **implementarea** sistemului



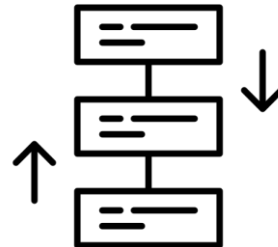
Aspecte de integrare

- Un **aspect de integrare** este un set de teme ce trebuie avute în vedere la integrare, privind compatibilitatea componentelor cu rolul lor în sistem, precum și cu actorii cu care interacționează într-un set dat de interacțiuni
 - Aspecte tehnice
 - Aspecte semantice
 - Aspecte funcționale
 - Aspecte de politică
 - Aspecte logistice



Aspecte de integrare – tehnice

- **Aspectele tehnice** corespund perspectivelor tehnologice si inginerești din cadrul RM-ODP (ISO Reference Model for Open Distributed Processing)
 - Aspectul de **integrare a coordonării** – “înțelegere” între componente la nivel de mecanism, interfață, protocol
 - Aspectul de **integrare a datelor** – adesea, aceste preocupări fac parte din convenția pentru interfețe: componentele “se pun de acord” asupra organizării, structurii și reprezentării informației în timp ce comunică



Aspecte de integrare

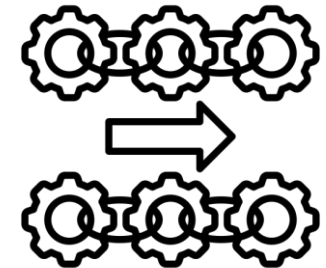
- **Aspectele de politică** se referă la abilitatea sistemului, ca întreg, de a derula procesele de afaceri
 - Asigură faptul ca totul se desfășoară conform **politicilor companiei**
 - De ex. securitate, încredere, disponibilitate, actualitate
- **Aspectele logistice** țin de impactul pe care îl are sistemul asupra afacerii în sine. Ele sunt strâns legate atât de procesul de integrare cât și de calitatea sistemului rezultat
 - De ex. **costuri, timp, resurse** materiale pentru procesul de integrare, costuri pe termen lung, consum de resurse și planuri de viitor pentru sistemul integrat



Integrarea orizontală și verticală

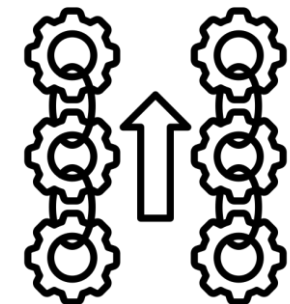
Integrarea pe orizontală

- folosirea tehnologiilor sistemelor distribuite care permit schimbul de date și informații între diferite sisteme sau subsisteme



Integrarea pe verticală

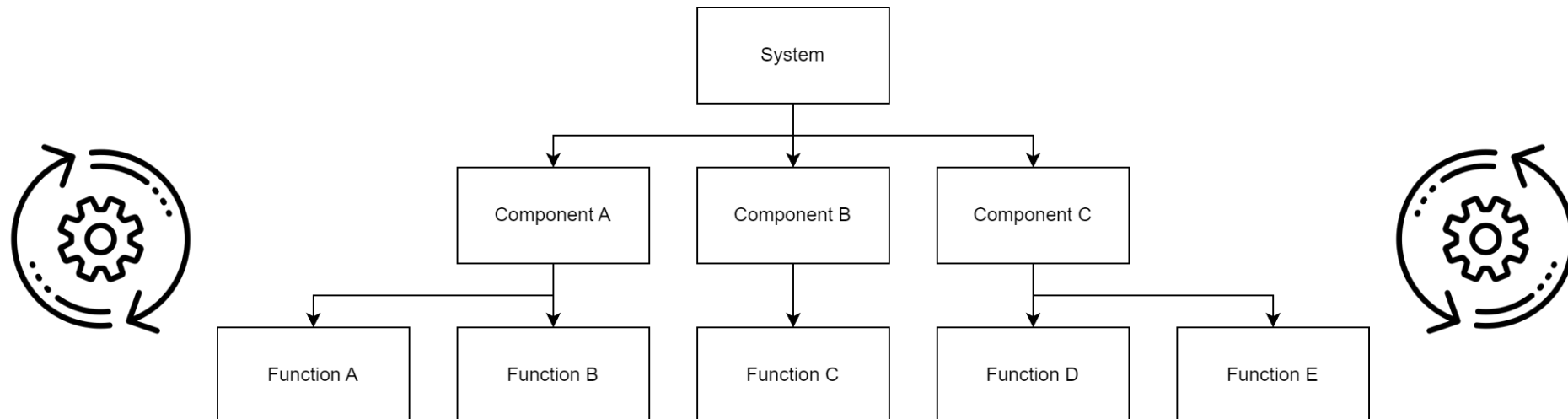
- folosirea tehnologiilor avansate pentru creșterea performanței și integrarea lor în cadrul sistemului



Integrare bidirecțională

Round-trip engineering

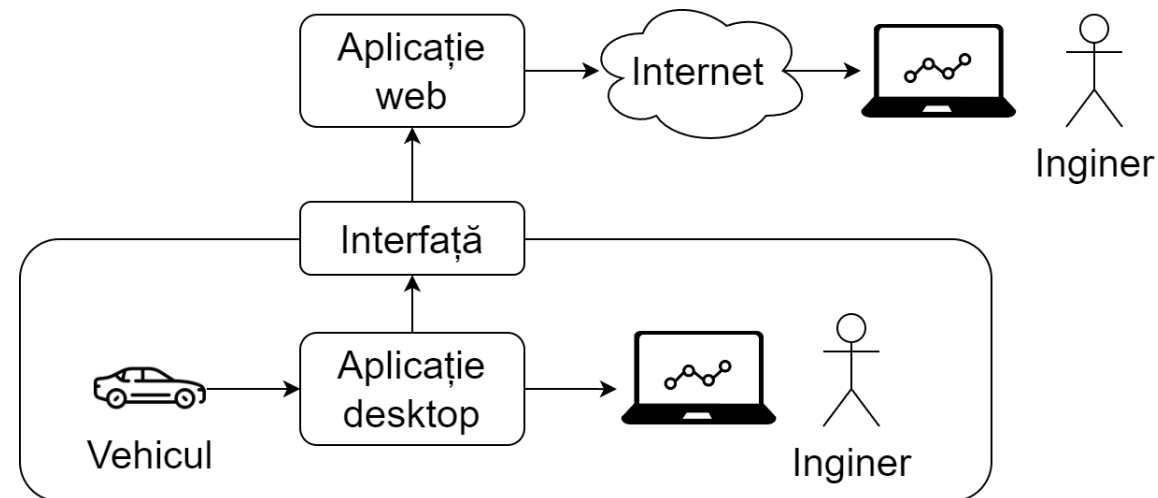
- **Forward engineering** – rezultatele activității la un nivel de abstractizare mai mare sunt transmise pe niveluri de execuție
- **Reverse engineering** – regăsirea informațiilor de modelare după efectuarea unor modificări în implementarea sistemului



Integrare bidirecțională

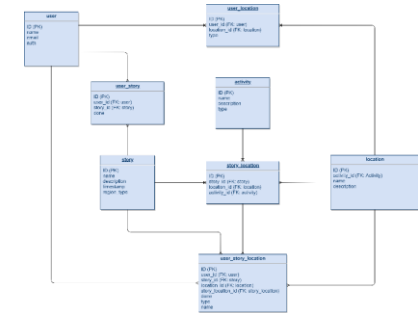
Problemă

- Proiectarea unui subsistem pentru transferul de date bidirecțional dintre 2 sisteme incompatibile (ex. aplicație desktop vs aplicație web)
- Studiu de caz: Transformarea unei aplicații desktop “legacy” pentru diagnoză auto într-o aplicație web modernă

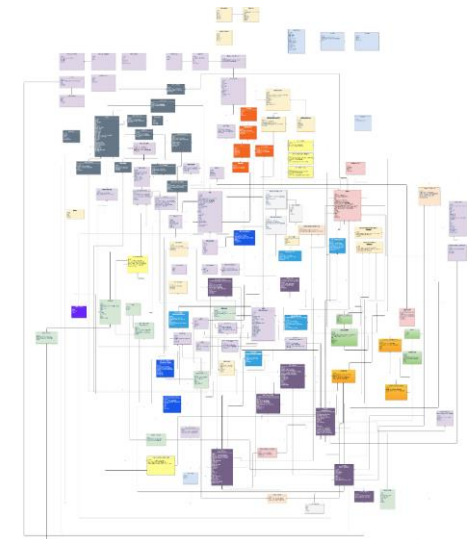


Modelarea domeniului

- Modelarea domeniului este o etapă esențială în proiectarea sistemului integrat
- Modelul datelor crește pe măsură ce crește complexitatea sistemului
- Devine dificil de înțeles în ansamblu – greu extensibil
- ..fără un model al domeniului



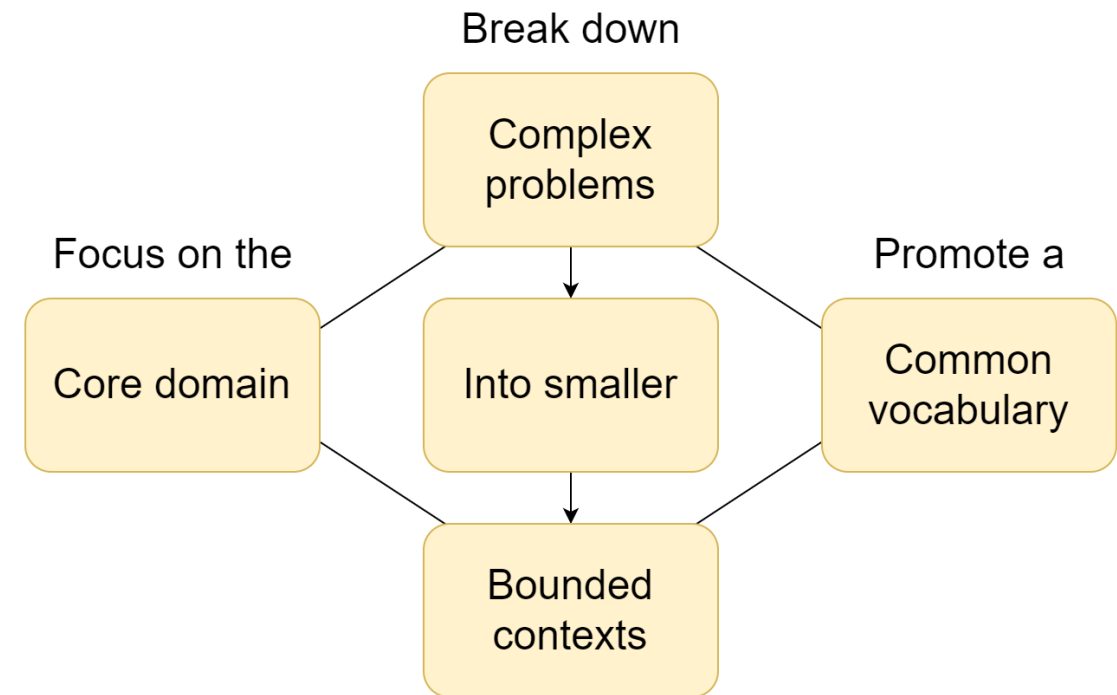
Ex. Schema bazei de date..



..a evoluat în timp

Modelarea domeniului

- Cum modelăm domeniul?
 - Problemele complexe se împart în **sub-probleme**, bine delimitate ca și context
 - Utilizarea unui **vocabular comun** (și a uneltelor de modelare standardizate) pentru a defini modelele

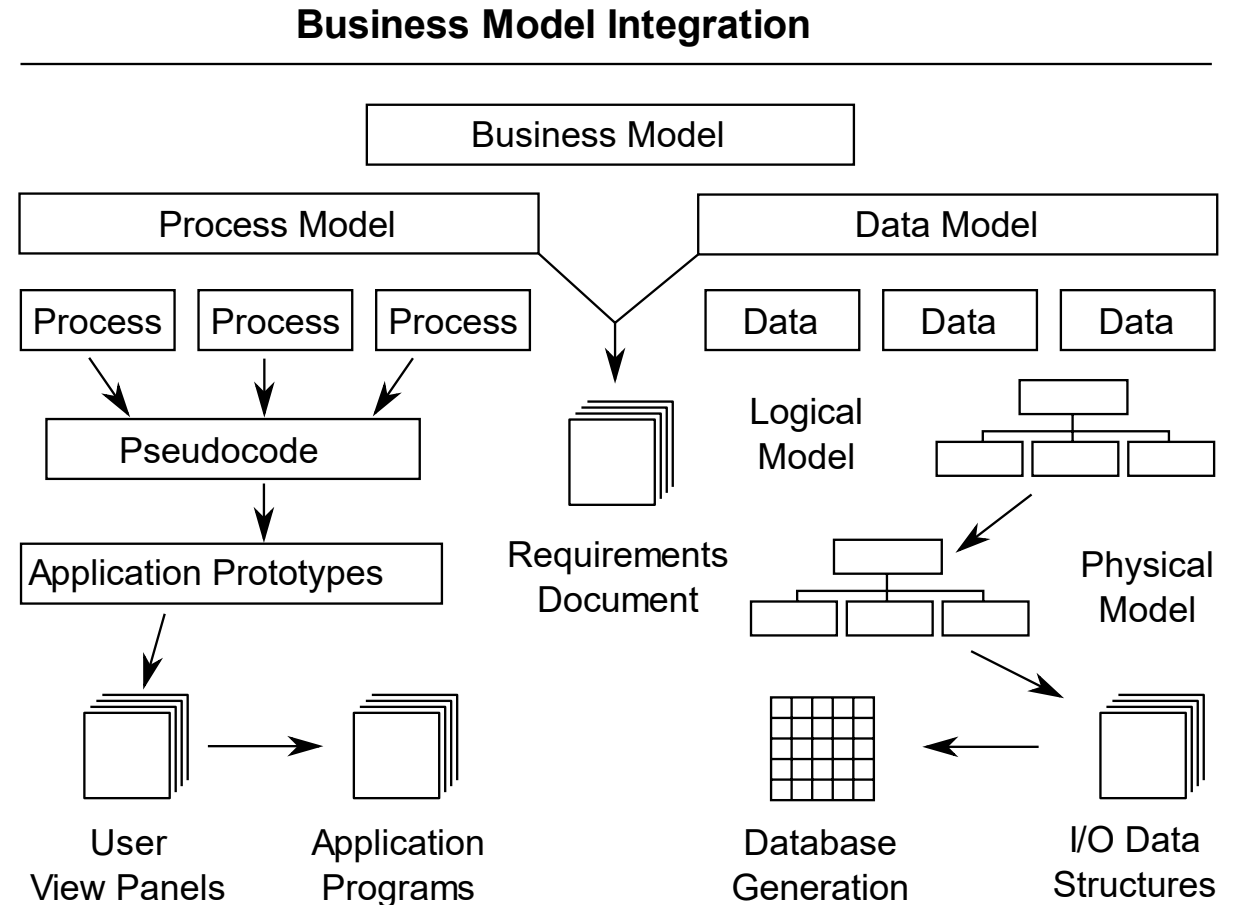


Modelarea domeniului

..începe prin definirea proceselor de afaceri

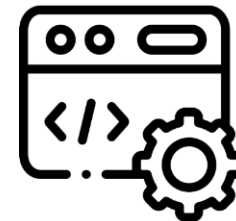
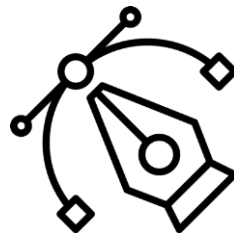
Modelul de afaceri

- Include modelul **proceselor** și modelul **datelor**
- Arată **corespondența** dintre nivelurile de abstractizare
- Se construiește în etapa de **analiză** a afacerii (en. business analysis)

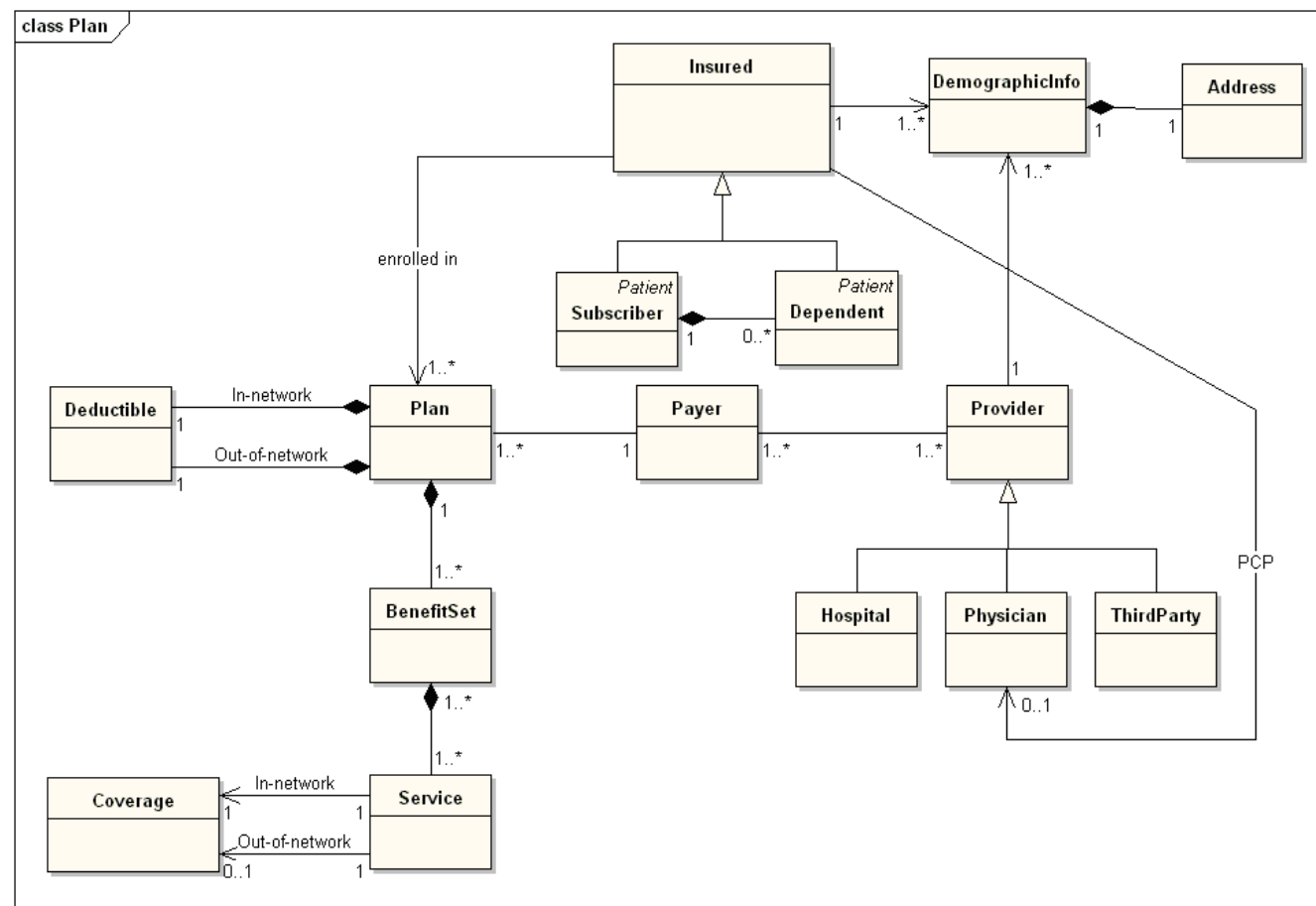


Modelarea domeniului – definiție

- **ISO/IEC/IEEE 24765: domain model** – “a product of domain analysis that provides a representation of the **requirements** of the domain.”
- **Domeniul** – reuneste conținuturi, concepte și idei necesare pentru descrierea **arhitecturii unui sistem software** particular



Modelarea domeniului – exemplu



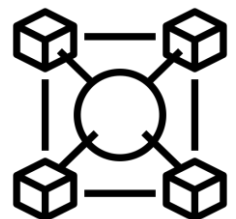
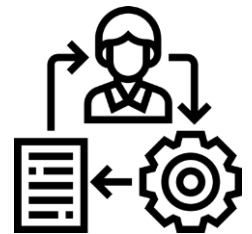
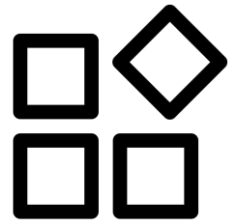
Diagramă de clase

Sursa: Kishorekumar 62 - Own work, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=5863269>

Modelarea domeniului – caracteristici

Caracteristici ale modelelor

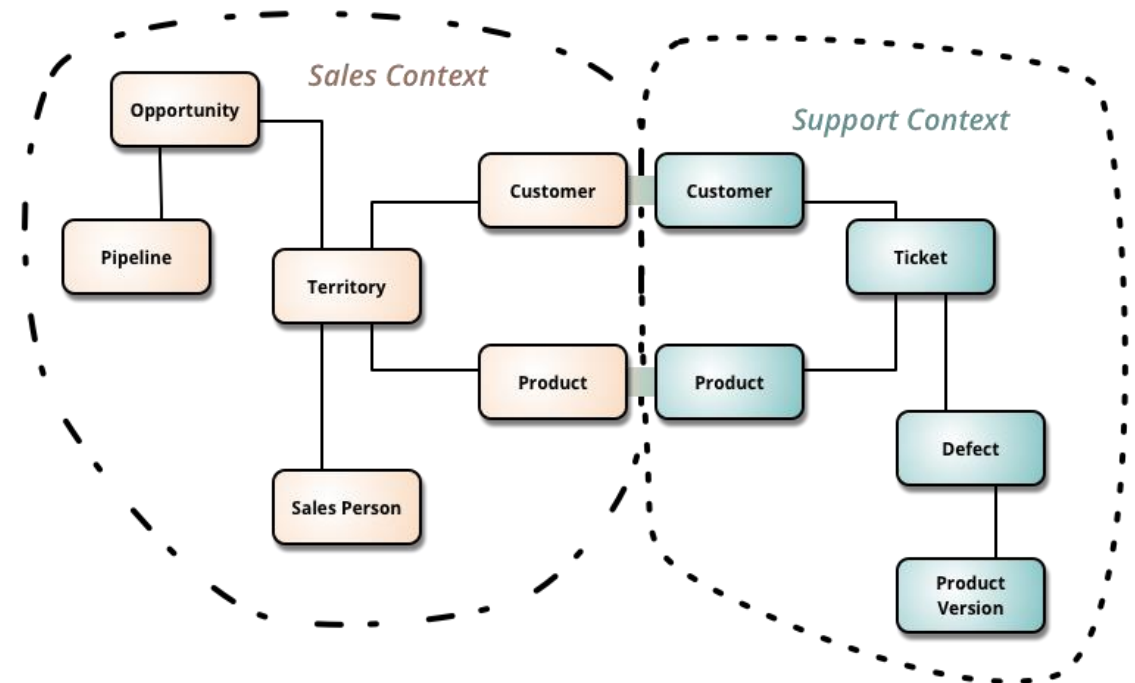
- Reutilizabilitatea
 - Cerințele funcționale sunt utile pentru organizații asemănătoare
 - Utile pentru descrierea extensiilor și modificărilor sistemului
- Portabilitatea
 - Metamodelle
 - Generatoare de cod, transformatoare între diferite niveluri de abstractizare
- Interoperabilitatea
 - Descrieri explicite și formale ale interfețelor



Modelarea domeniului – studiu de caz

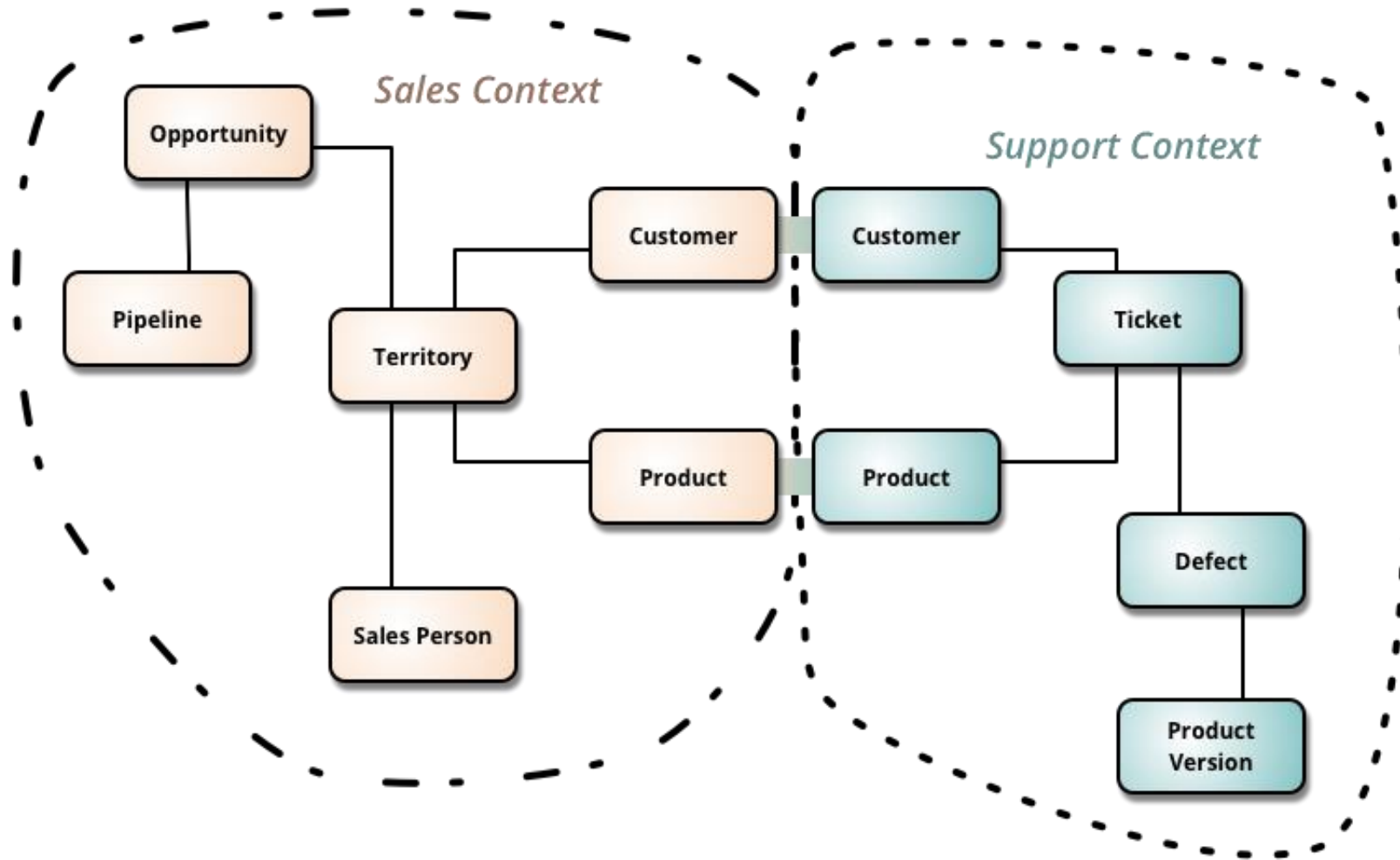
Domain-Driven Design (DDD) este o paradigmă – colecție de principii și modele care ajută dezvoltatorii să creeze sisteme pornind de la **cerințele de business**

- Aplicat corect poate duce la abstracții software numite modele de domenii
- Aceste modele **încapsulează logica de afaceri complexă**, reducând decalajul dintre realitatea afacerii și cod



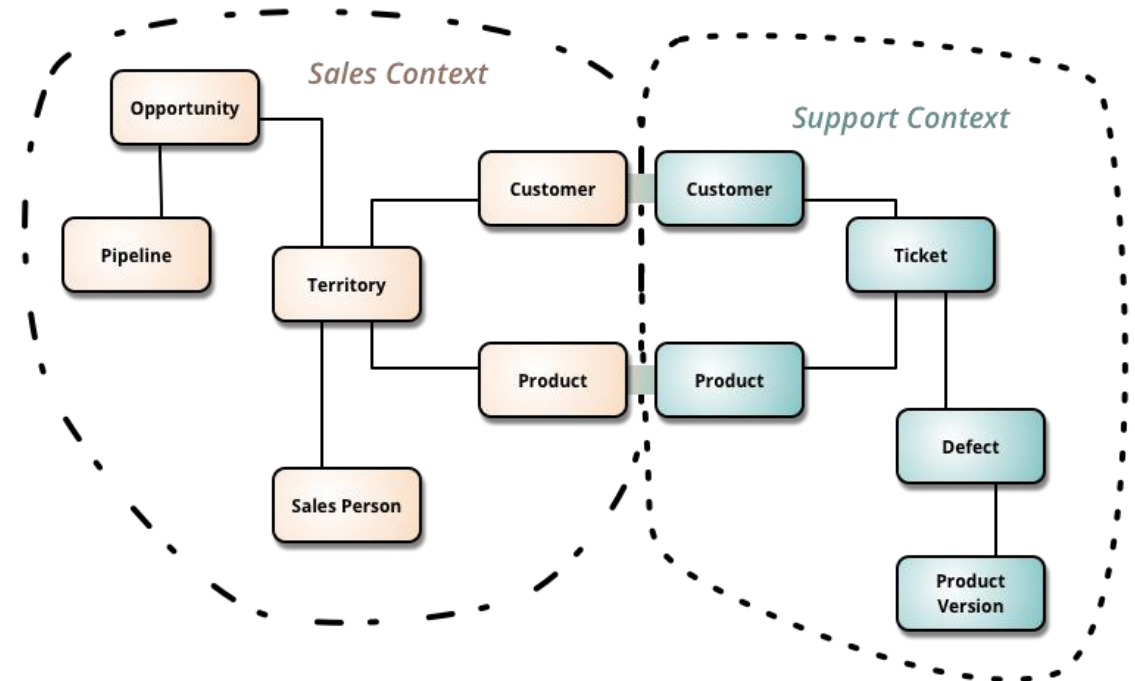
Sursa: Martin Fowler, BoundedContext, martinFowler.com, 2014

Modelarea domeniului – studiu de caz



Modelarea domeniului – studiu de caz

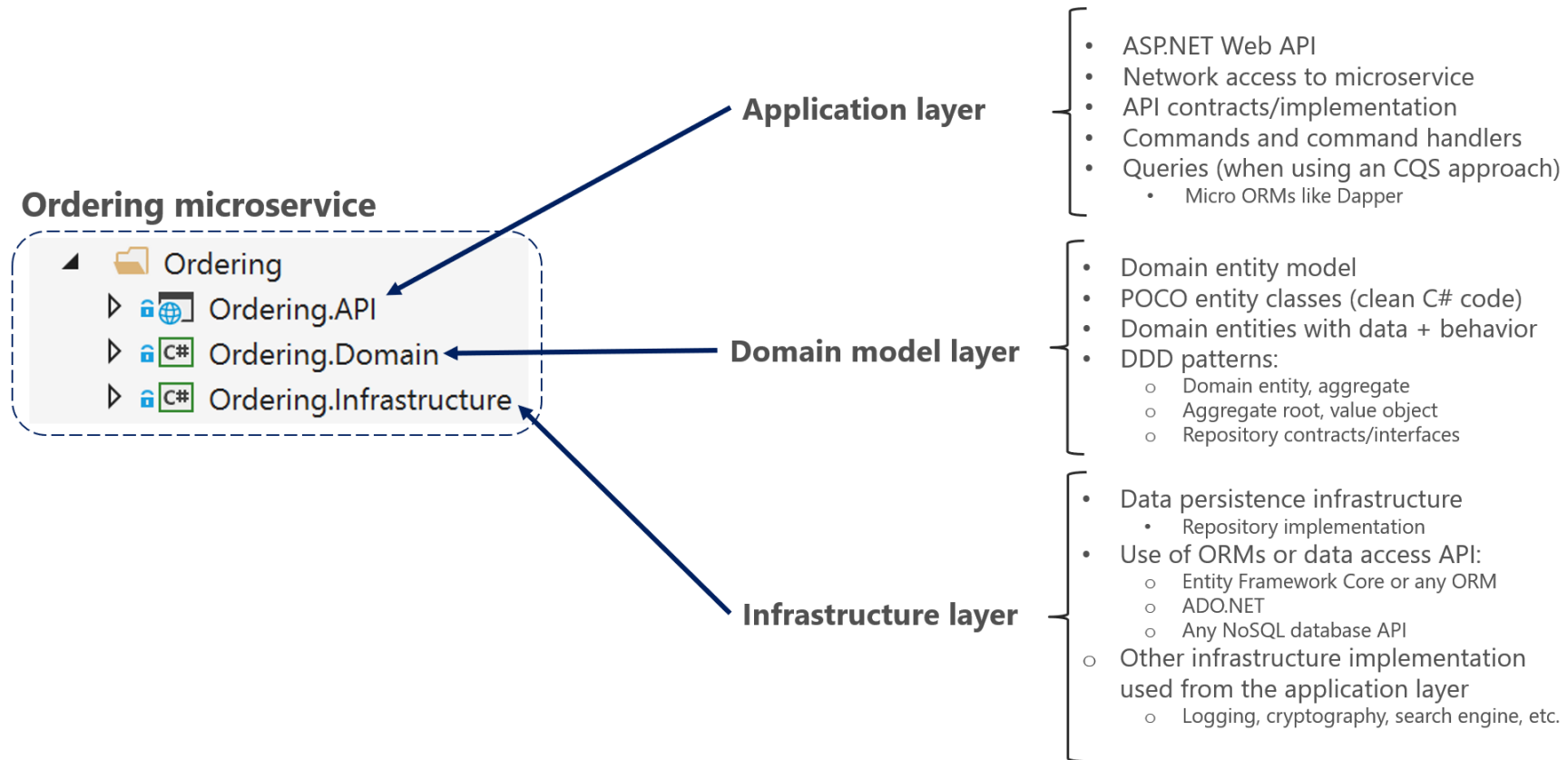
- Ce facem dacă modelul nu se potrivește cu implementarea?
 - Refactorizarea implementării
 - Sau redefinirea modelului?
- Soluții
 - Agregarea obiectelor similare din **perspectivă semantică**
 - Asigurarea **înțelegerii domeniului** pe tot parcursul implementării



Sursa: Martin Fowler, BoundedContext, martinFowler.com, 2014

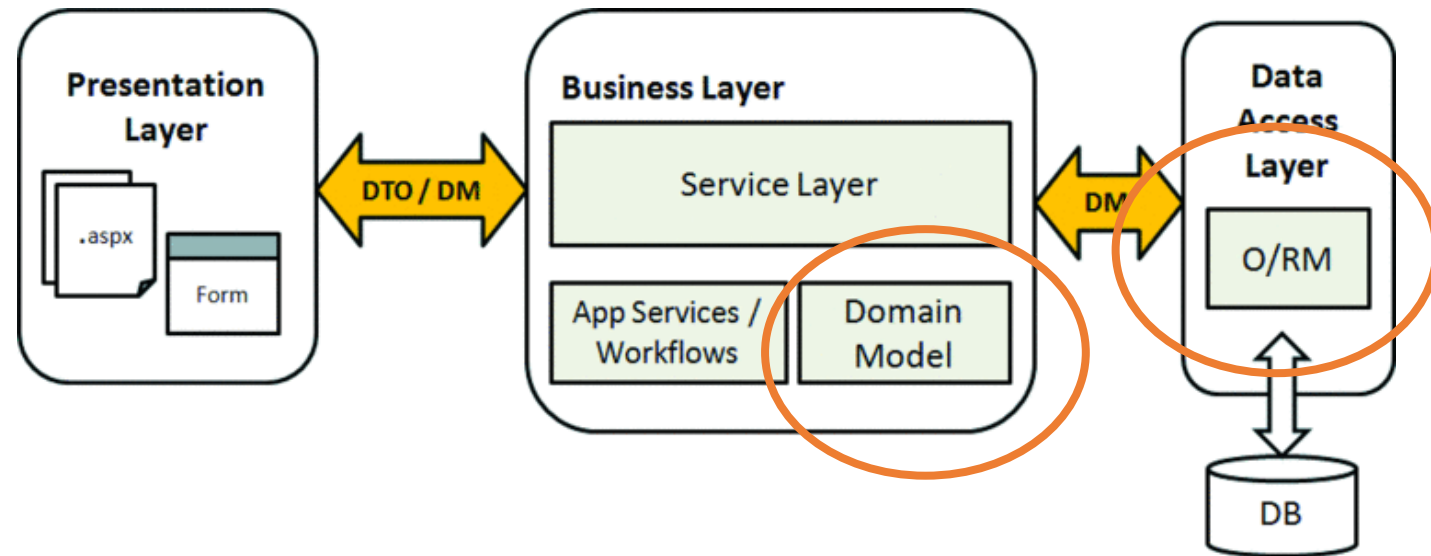
Modelarea domeniului – studiu de caz

Layers in a Domain-Driven Design Microservice



Modelarea domeniului – studiu de caz

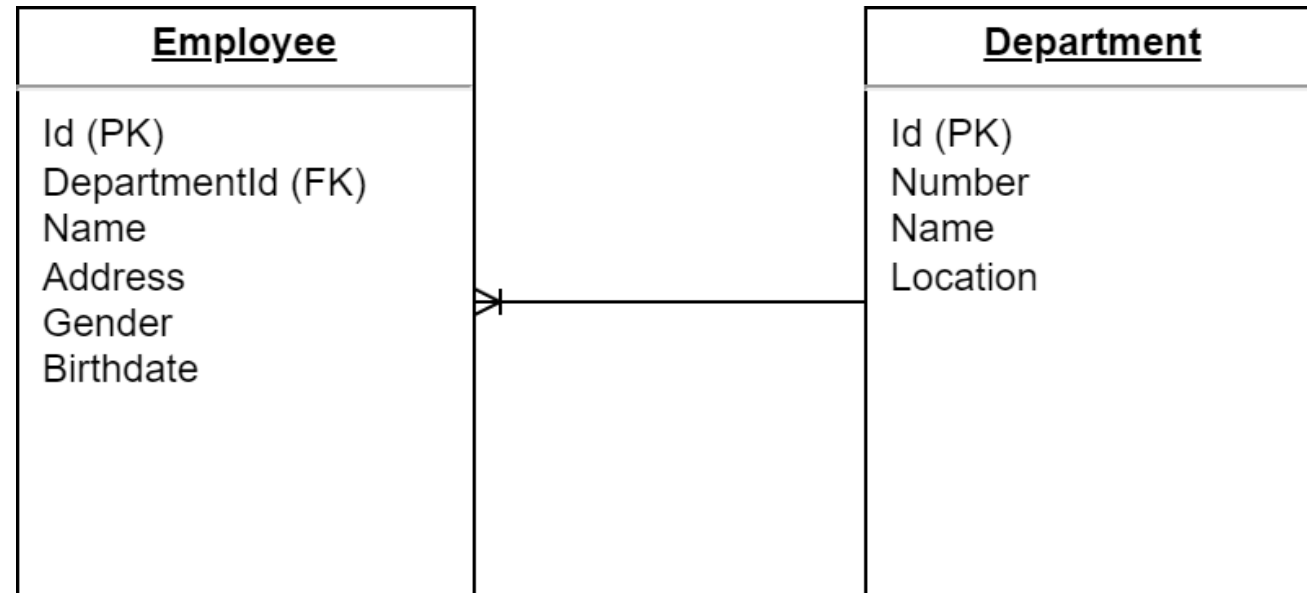
- Modelul datelor – Domain Model / Entity Data Model
- Modelul funcțional – Business Logic
- Nivelul de translație
 - DTO (Data Transfer Object)
 - ORM (Object Relational Mapping)



Sursa: Microsoft Learn

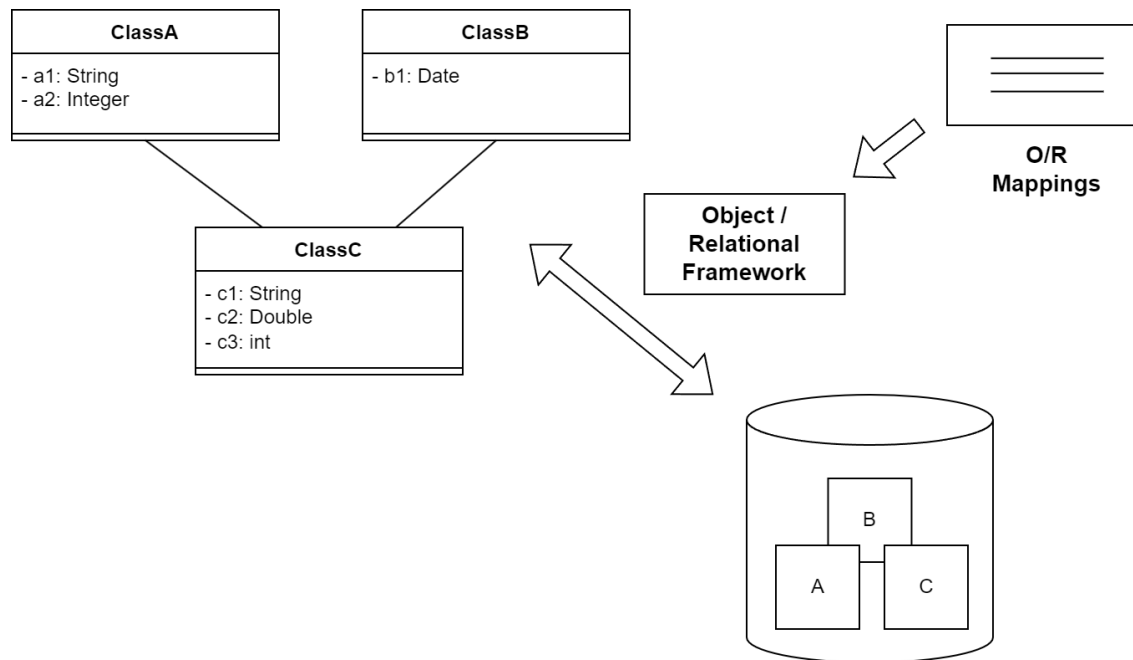
Modelarea domeniului – studiu de caz

- Schema bazei de date relaționale



Modelarea domeniului – studiu de caz

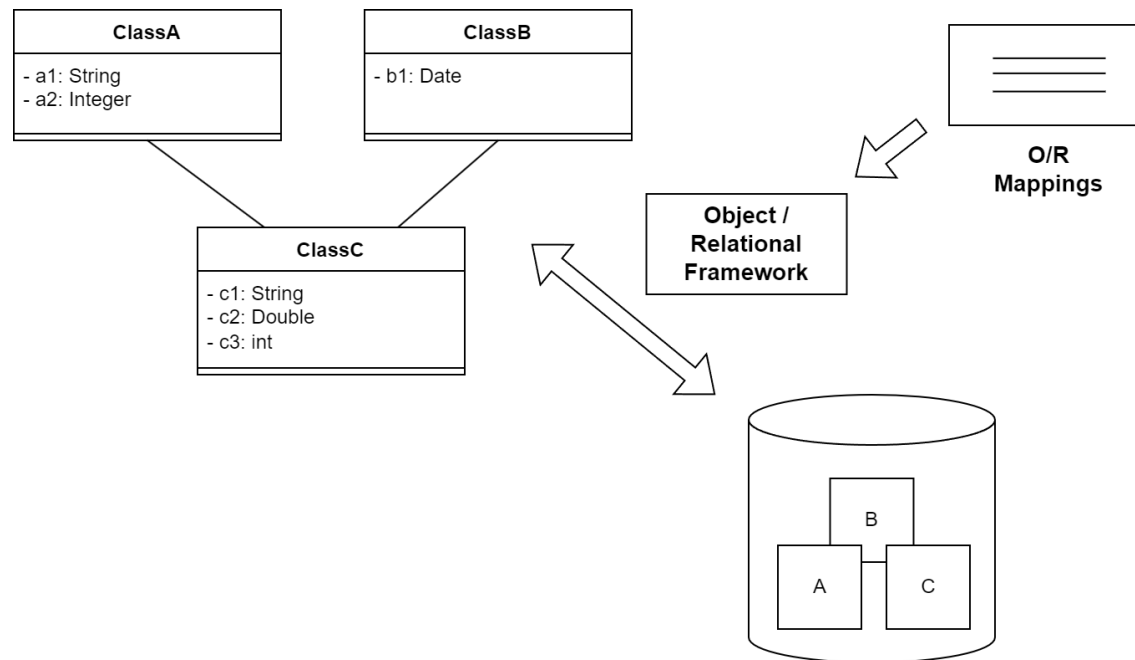
- Reprezentarea entităților
 - Node.js
 - [Sequelize ORM](#)



```
var Employee = sequelize.define('Employee', {  
  id: {  
    type: DataTypes.INTEGER,  
    primaryKey: true,  
    autoIncrement: true  
  },  
  departmentId: {  
    type: DataTypes.INTEGER,  
    allowNull: true,  
    references: {  
      model: 'department',  
      key: 'id'  
    }  
  },  
  name: {  
    type: DataTypes.STRING(100),  
    allowNull: true  
  },  
  // ...  
});
```


Modelarea domeniului – studiu de caz

- Reprezentarea relațiilor
 - Node.js
 - [Sequelize ORM](#)



```
Employee.associate = (models) => {  
  Employee.belongsTo(models.Department, {  
    foreignKey: 'departmentId'  
  });  
};  
Department.associate = (models) => {  
  Department.hasMany(models.Employee, {  
    foreignKey: 'departmentId'  
  });  
};
```

Modelarea domeniului – studiu de caz

- Accesarea datelor – afișarea angajaților cu detalii despre departamentul lor

SQL

```
con.query(`
SELECT emp.name, dep.name
FROM Employee AS emp
INNER JOIN Department as dep
ON emp.departmentId = dep.id;
`, (err, result, fields) => {
  if (err) throw err;
  console.log(result);
});
```



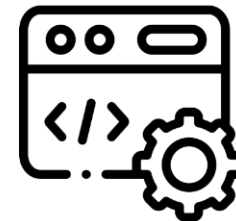
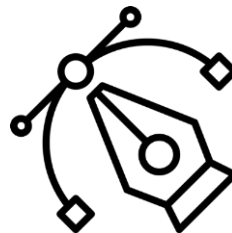
Sequelize ORM

```
var result = await models.Employee.findAll({
  include : [{
    model: models.Department
  }]
});
console.log(result);
```

Modelarea domeniului – etape

Ingineria domeniului

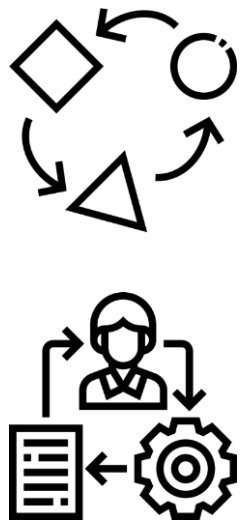
- O procedură de stabilire a cerințelor domeniului, având ca obiectiv **identificarea structurilor și a dependențelor** între ele
 - **Analiza** domeniului (domain analysis)
 - **Proiectarea** domeniului (domain design)
 - **Implementarea** domeniului (domain implementation)



Modelarea domeniului – proceduri

Programarea generativă

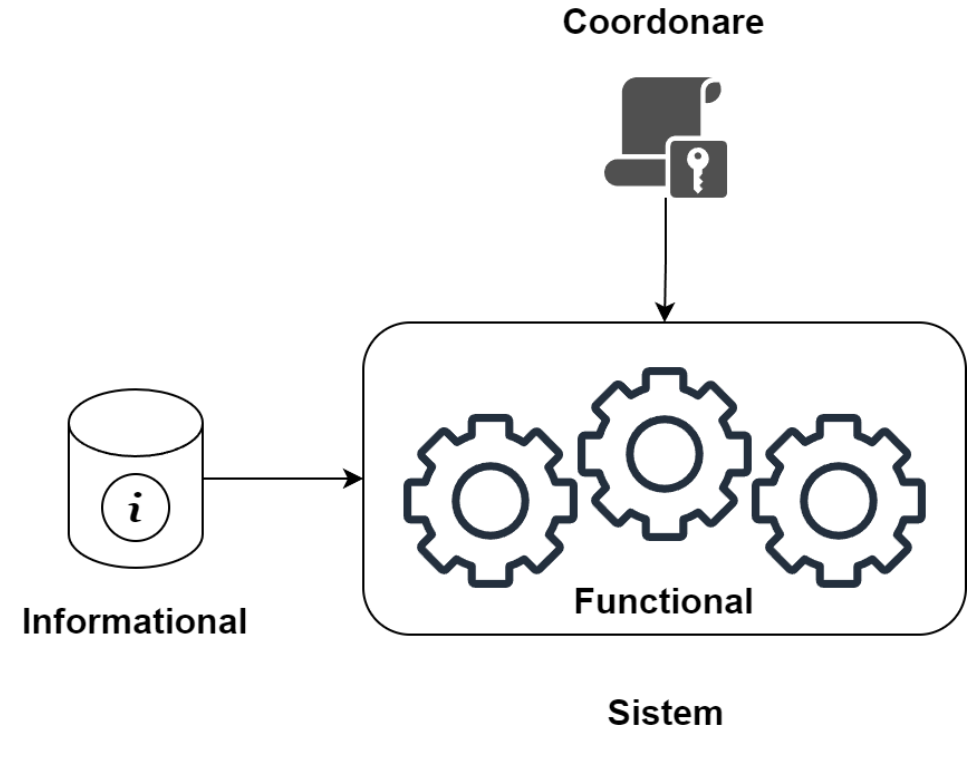
- Descrie proceduri prin care se realizează componente care pot rula automat, respectiv se realizează cod compilat din domeniul modelului – generatoare
 - **Transformari orizontale** (Query View Transformation)
 - Transformă descrierea modelului în alt model, la același nivel de abstractizare (în UML – M2M)
 - **Transformări verticale**
 - Asigură trecerea de la un nivel de abstractizare la altul (din UML în cod)



Domenii de abstractizare

Pentru a separa nivelurile de abstractizare, modelele se diferențiază prin perspective sau prin componenta specifică a sistemului pe care o descriu

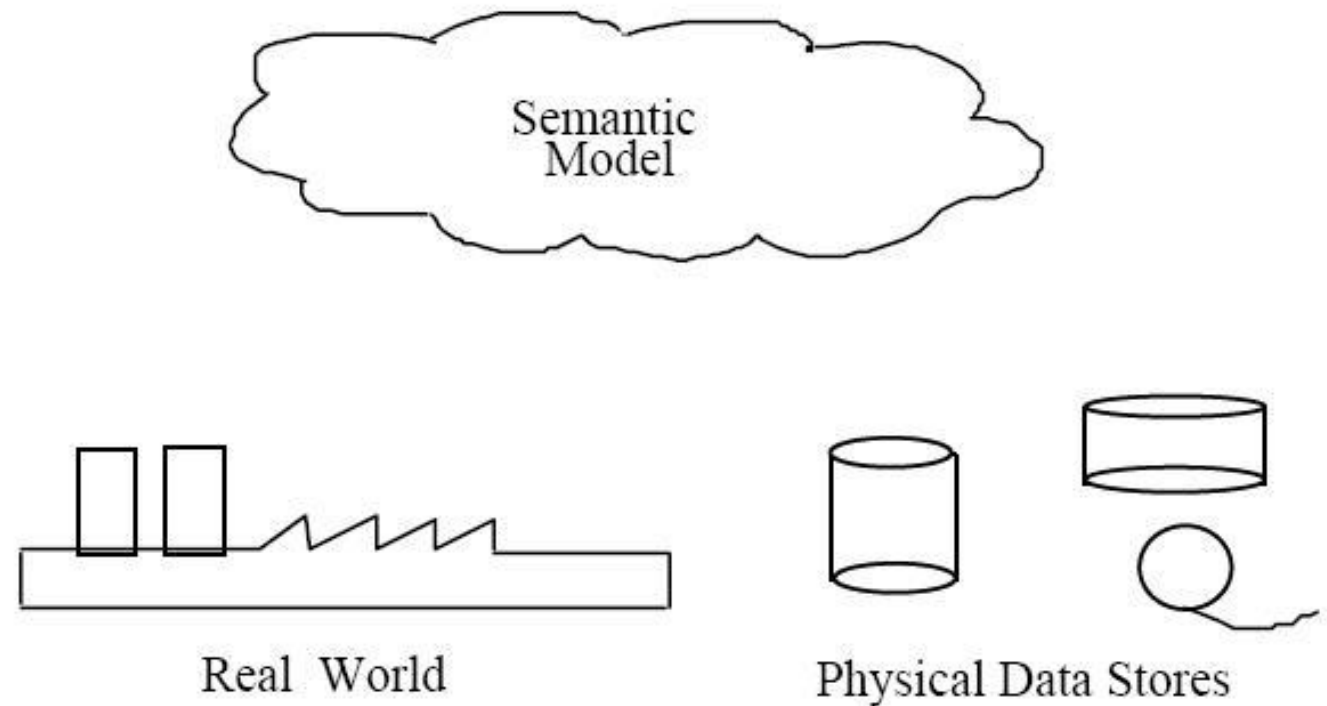
- Domeniul **informațional**
- Domeniul **funcțional**
- Domeniul de **coordonare**
- Domeniul de **sistem**





>> Domeniul informațional

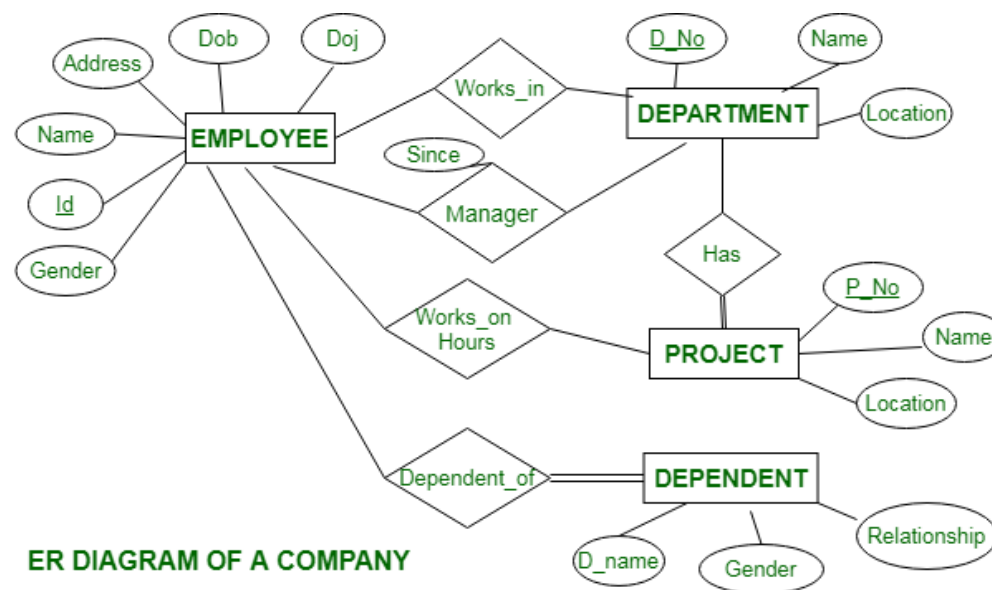
- Cuprinde **modele**
 - **Informaționale** → entități
 - **Semantice** → semnificații
 - **de date** → reprezentări





Domeniul informațional

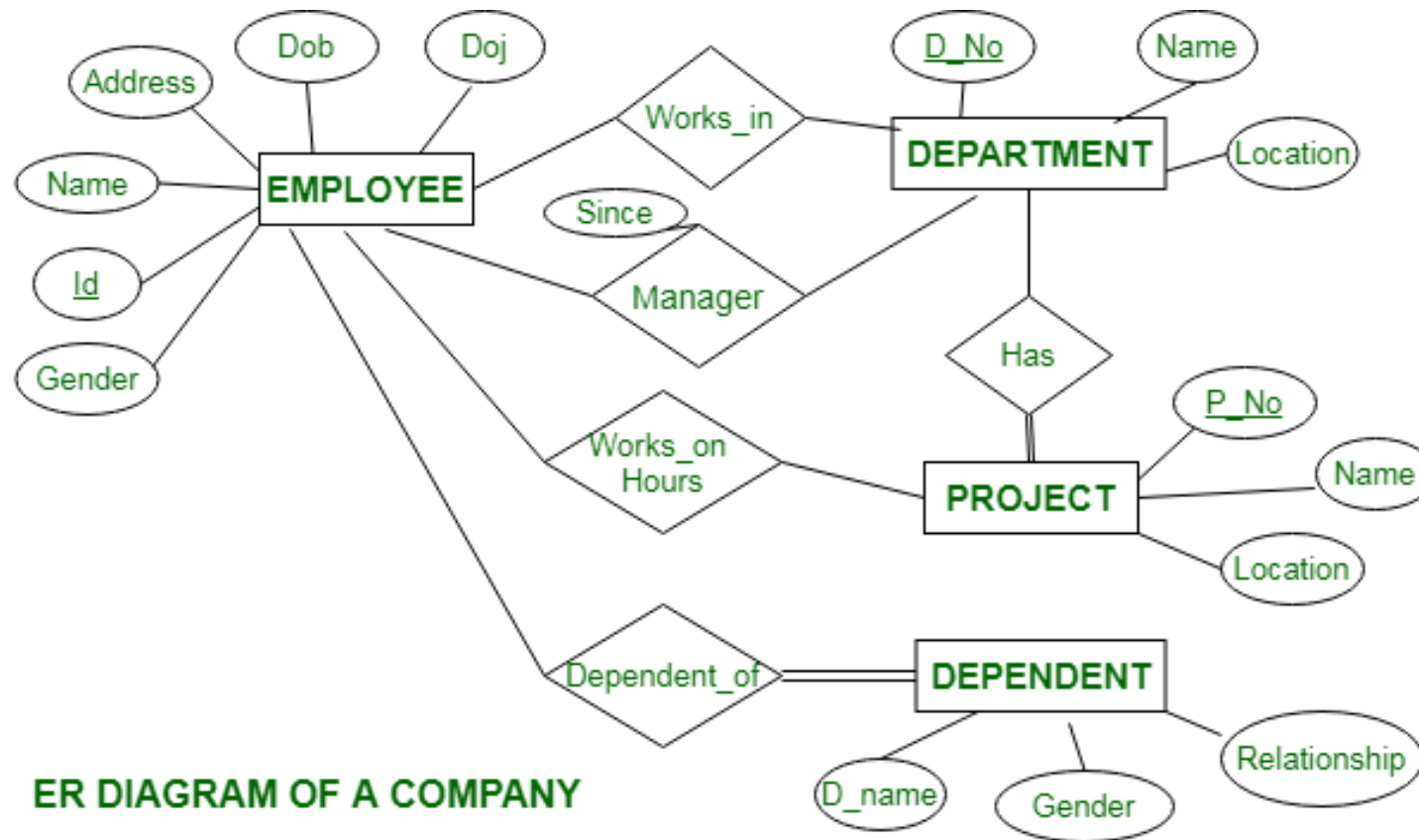
- **Modelul informațional** descrie obiectele din procesele de afaceri, relațiile dintre ele și proprietățile acestora
- Este un model conceptual – descrierea obiectelor nu ține seama de implementare





Domeniul informațional


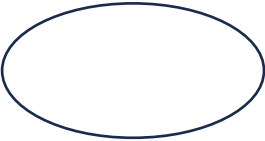
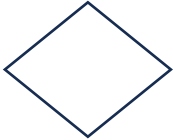
Ex: diagrama ER (Entity-Relationship Diagram) a unei companii





Domeniul informațional

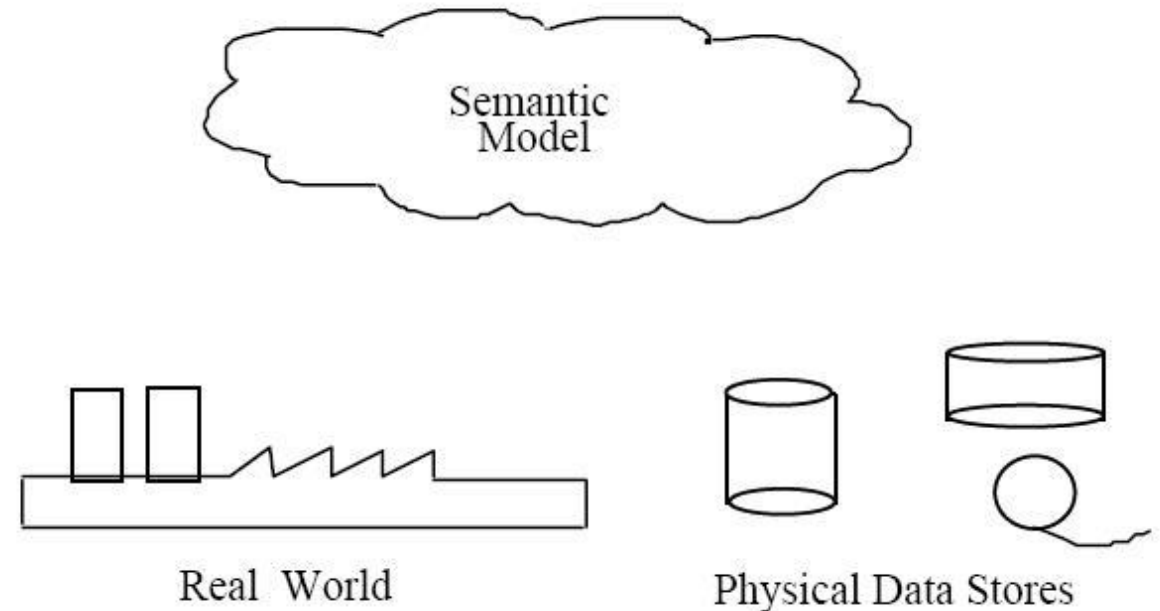
- **Modelul informațional** – diagrama Entitate-Relație
 - Definește structura logică a bazei de date
 - Corespondență directă cu schema bazei de date

| Figură | Simbol | Semnificație |
|------------|---|--|
| Dreptunghi |  | Entitate (puternică – independentă/slabă/asociativă) |
| Elipsă |  | Atribut (cheie/compus/multiplu/derivat) |
| Romb |  | Relație (1-la-1 / 1-la-N / M-la-N) |



Domeniul informațional

- **Modelul semantic** este un model conceptual care descrie prin reprezentări grafice înțelesul obiectelor, prin raportare la context
 - Descrie **semnificația** termenilor folosiți
 - Deseori folosit în aplicații de AI (Inteligență Artificială)
 - NLP (procesarea limbajului natural)
 - Algoritmi de clasificare
 - etc.

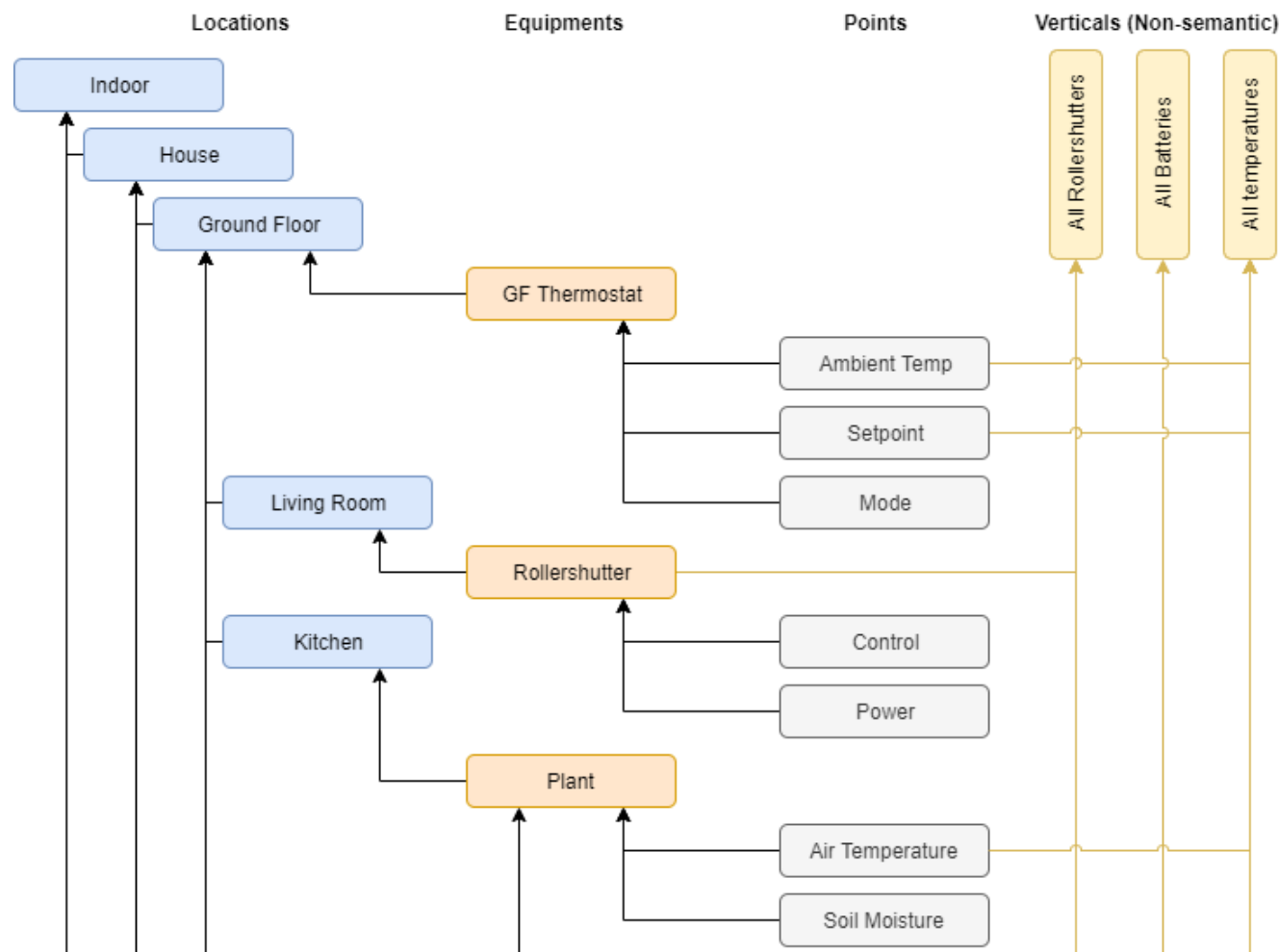




Domeniul informațional

Studiu de caz – OpenHab

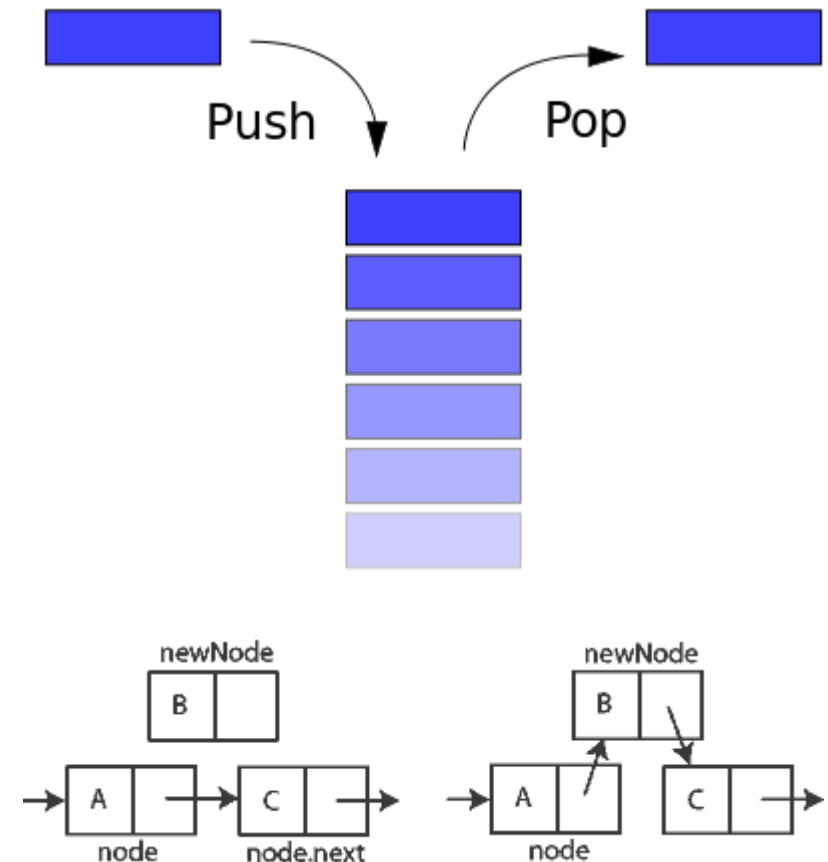
- Platformă open source pentru automatizarea locuinței (vezi curs IoT)
- Integrează multiple dispozitive și sisteme, printr-o interfață comună la nivel de utilizator și aplicație
- Model semantic
 - Locație – locație fizică (clădire, cameră, etc.)
 - Echipament – grupare de elemente
 - Punct – element / interfață





Domeniul informațional

- **Modelul datelor** este un model tehnic care definește organizarea și reprezentarea informației (recepționare, stocare, transfer)
- Modelul include:
 - Modele de **baze de date**
 - Limbaje de **transfer de date** (ex. XML, JSON)
 - **Structuri de date** ale limbajelor de programare

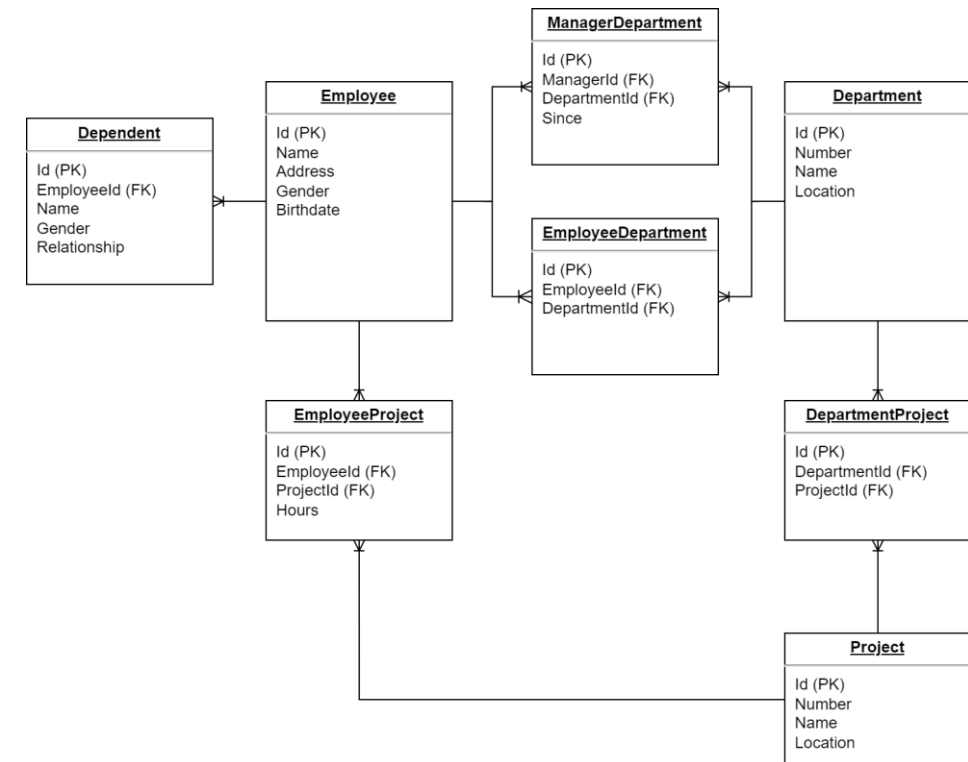
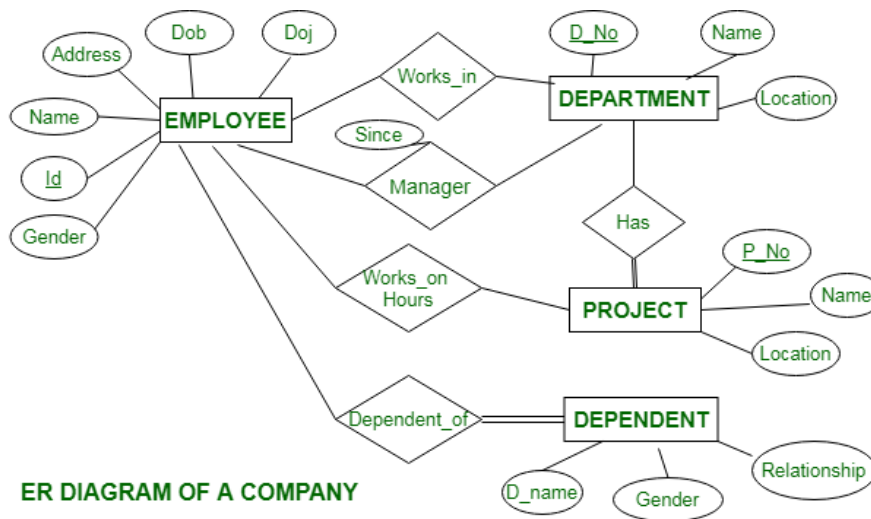




Domeniul informațional

Ex: diagrama ER (Entity-Relationship Diagram) a unei companii

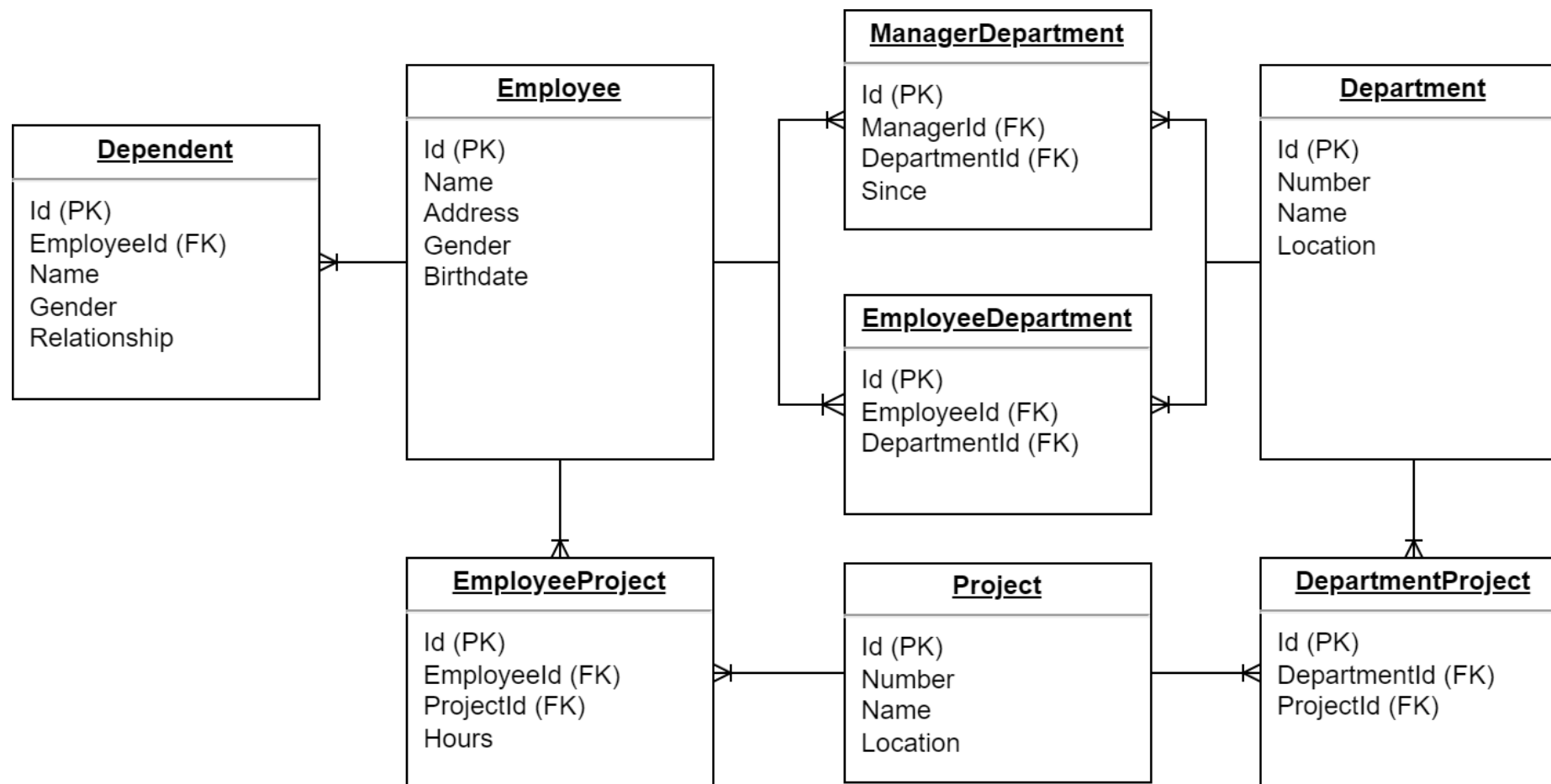
→ Schema bazei de date





Domeniul informațional

Ex: Schema bazei de date a unei companii





>> Domeniul funcțional

- Modelele din acest domeniu descriu **comportamentul** sistemului și al componentelor
 - **comportamentul** este ceea ce face sistemul
 - **funcția** este comportamentul dorit al sistemului
- Cuprinde modele comportamentale din UML
 - Diagrama **cazurilor de utilizare** (en. Use Case Diagram)
 - De **execuție**: **diagrame de secvență, de interacțiune, de activități**
 - De **proces**: **diagrame de stare (ASM)**



Domeniul funcțional

Un **model funcțional** – o reprezentare structurată a funcțiilor (activități, acțiuni, procese, operații) în cadrul unui sistem

Diagrama cazurilor de utilizare

- Prezintă descompunerea funcțională
- Este un model incomplet, deoarece ignoră anumite aspecte ale comportamentului (de ex. modificările de stare)

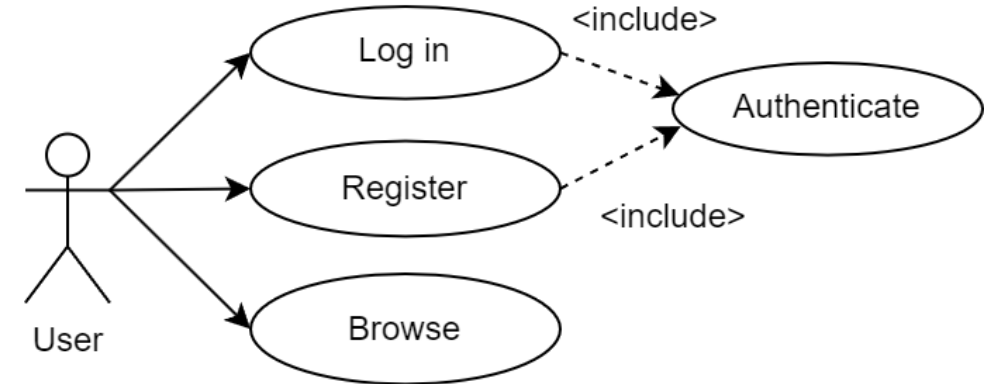


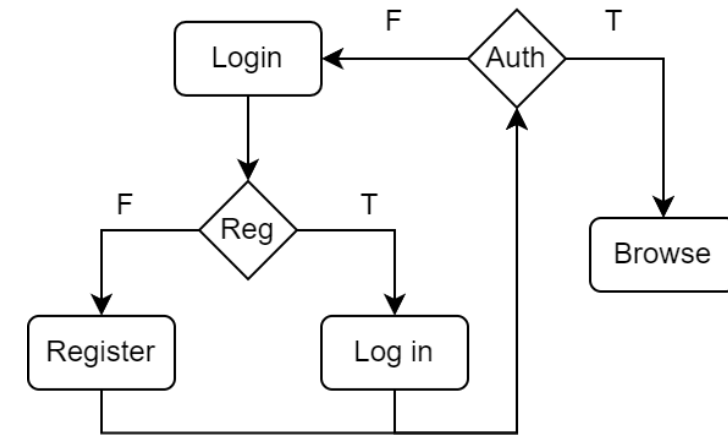
diagrama cazurilor de utilizare (use case)



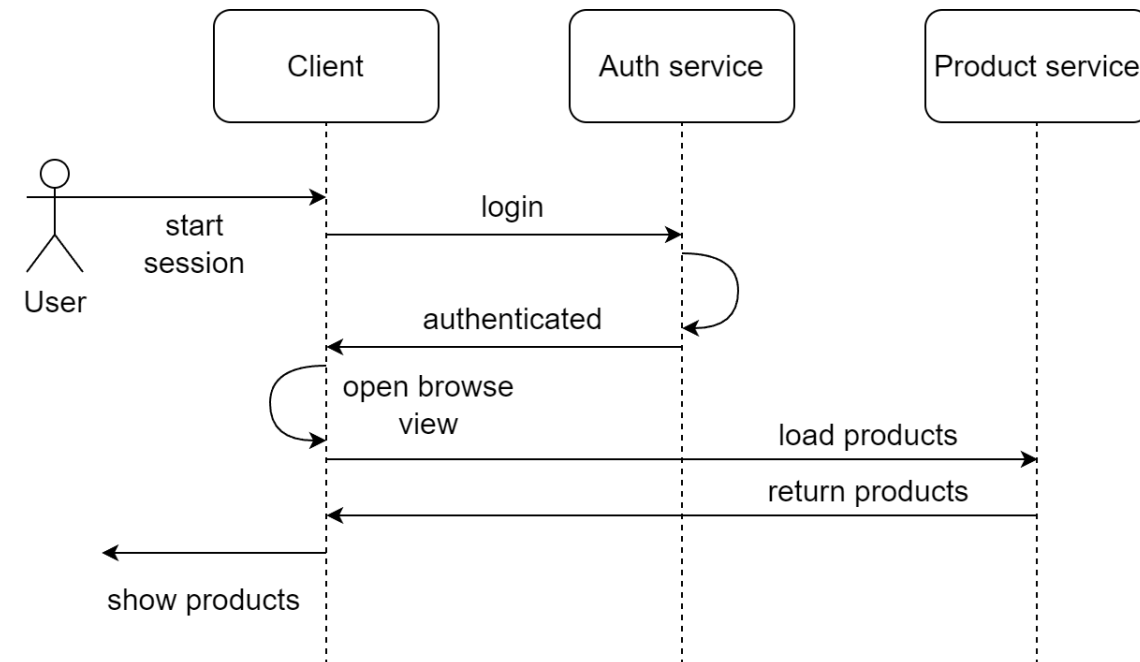
Domeniul funcțional

Modelul de execuție – reflectă modificările sistemului datorate unor acțiuni (elementare)

- model **procedural** – acțiuni secvențiale
- model **bazat pe evenimente** – acțiuni ca răspuns la stimuli externi (event-driven)
- model **bazat pe reguli** – acțiunile se desfășoară pe baza unor reguli predefinite (rule-based)
- Diagrame UML
 - de secvență
 - de interacțiune
 - de activități



diagramă de activități (workflow)



diagramă de secvență (sequence)



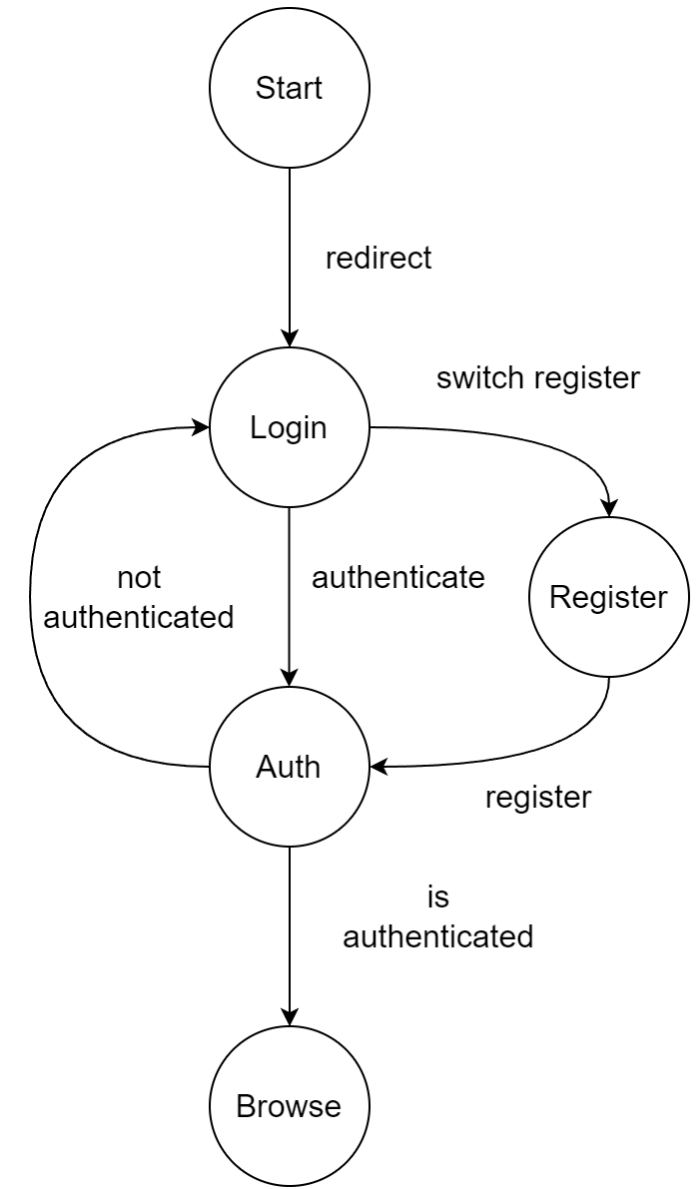
Domeniul funcțional

Modelul de stare reflectă comportamentul sistemului ca răspuns al activității acestuia într-un context dat

- Diagrame UML: de stare (ASM)

Modelul de proces reprezintă descompunerea unei funcții (graf temporal de activități ale unei componente)

- descrie modul în care un sistem îndeplinește o funcție

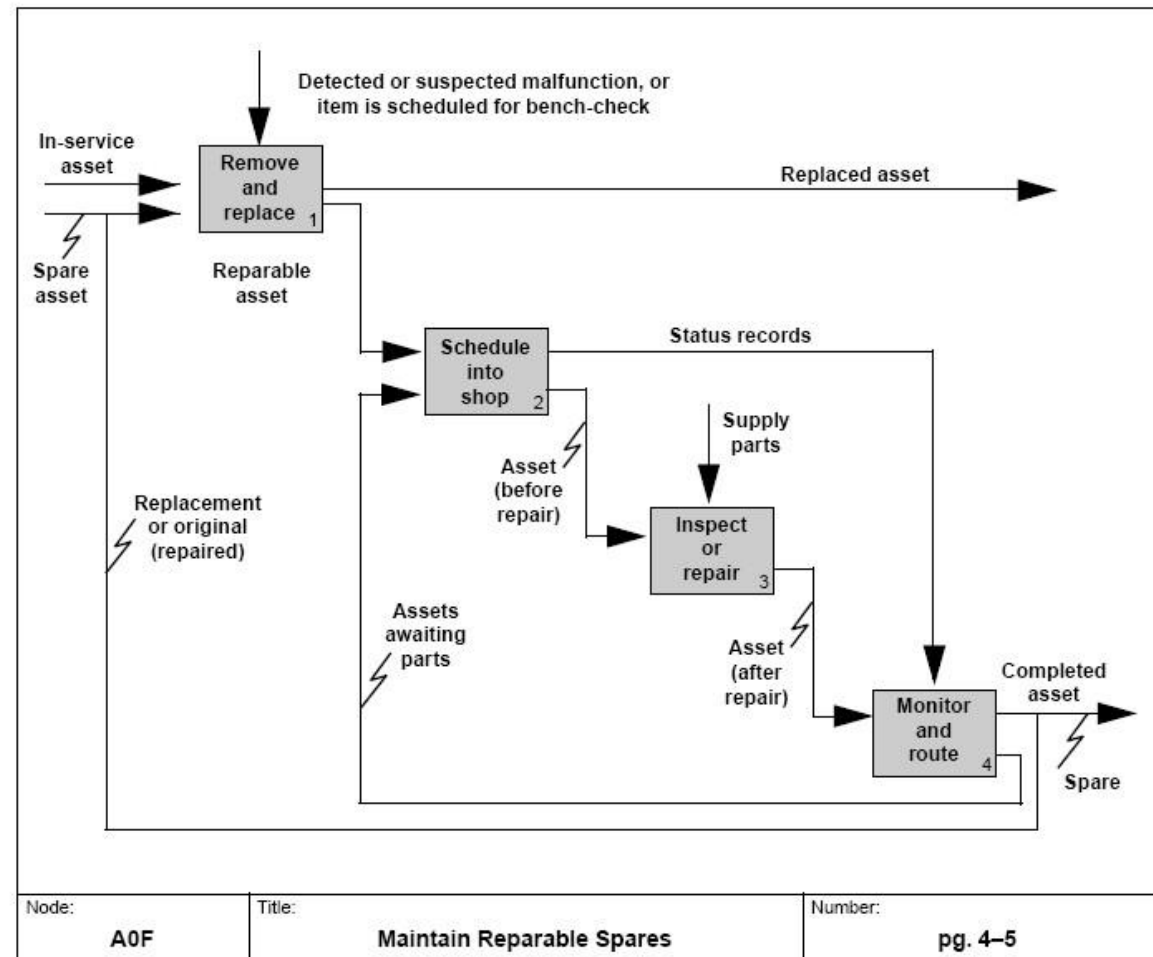


diagramă de stare



Domeniul funcțional – model de proces

Exemplu: procesul de mentținere a pieselor de schimb reparate





>> Domeniul de interfațare/coordonare

Modelele de interfațare specifică mecanismul prin care o componentă contribuie la interacțiune

- Interfața este un set de specificații comune componentelor care interactionează
- Interfețele depind de tehnologie/caracteristicile tehnice
- Exemplu: interfața USB, interfata grafică, interfața Java, interfețe fizice de conectare ale unor dispozitive externe, etc.





Domeniul de interfațare/coordonare

Modelele de coordonare specifică interacțiunile între componente (fluxul de informație și control)

- Modele tehnice care descriu **interacțiunile** ce apar odată cu execuția procesului în termenii expuși de interfața componentelor
- **Protocoloalele** sunt reguli de implementare și funcționare (secvențiere) la nivel de interfață
- Sunt situații în care specificarea interfeței depinde de protocolul folosit

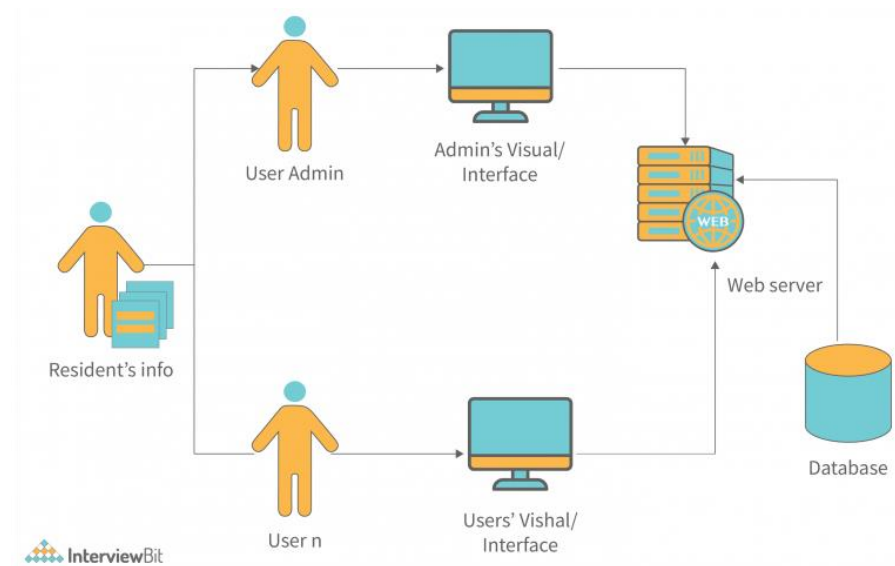
Modelele de prezentare sunt cazuri particulare ale modelelor de interfațare, unde componentele sau mecanismele de interacțiune implică **acțiune umană**



>> Domeniul de sistem

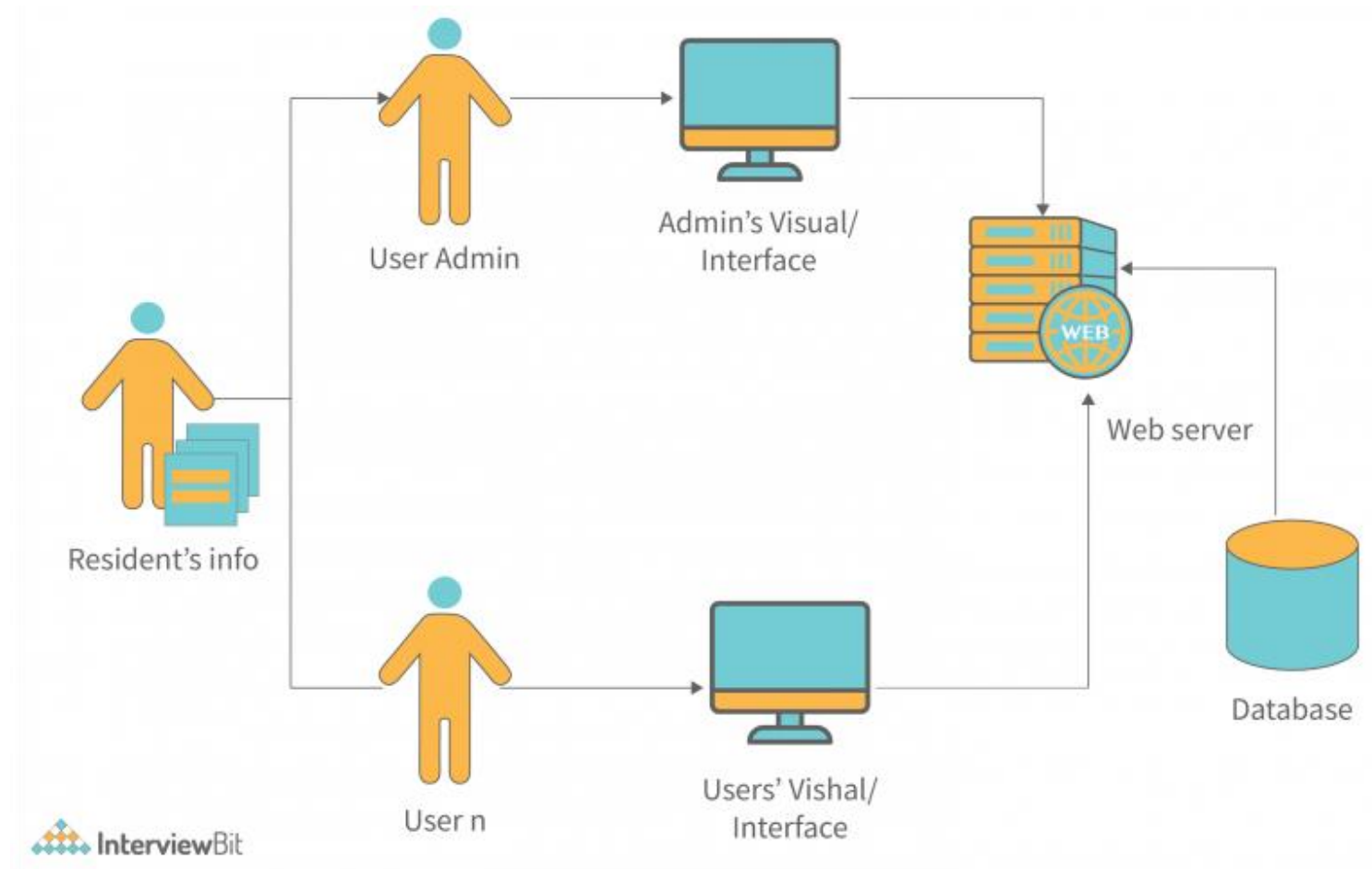
Modelele structurii de sistem descriu organizarea resurselor în sistem, inclusiv rolurile componentelor și legăturile dintre ele

- Modelele de nivel înalt (high-level) sunt perspective de întreprindere
- Modele mai detaliate sunt perspective ingineresti





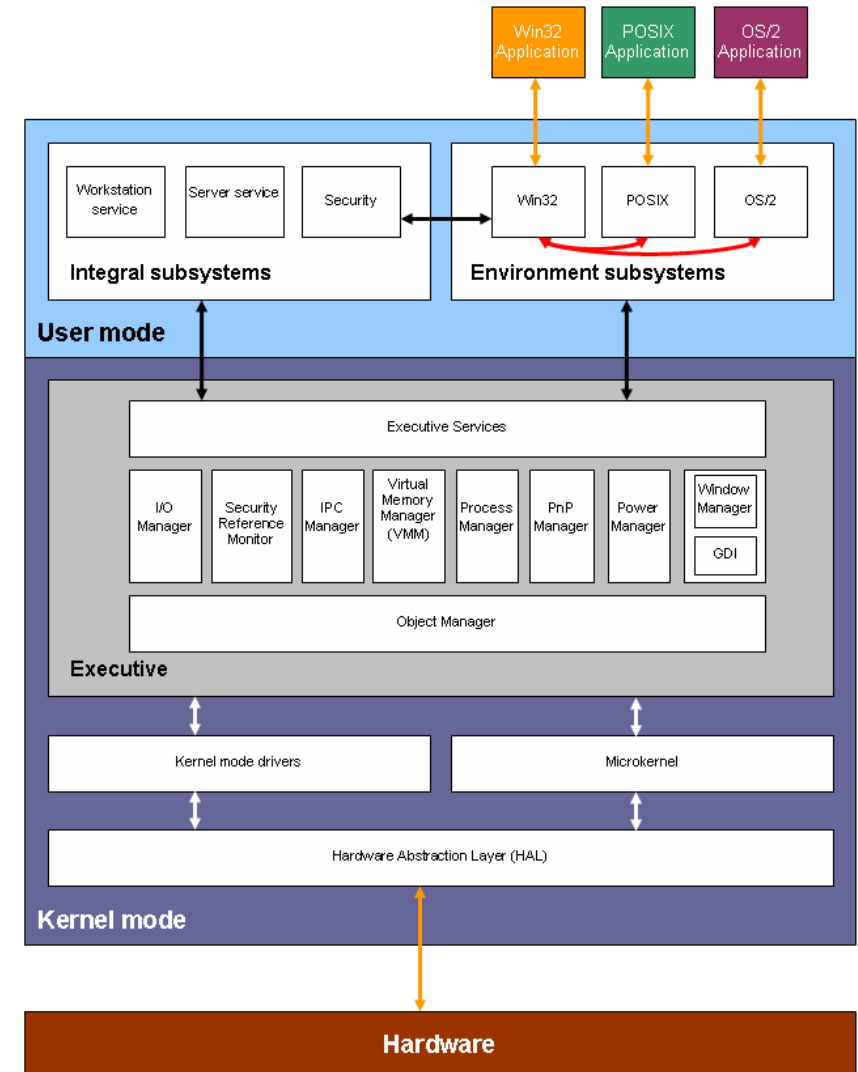
Domeniul de sistem



Domeniul de sistem – arhitectura

Schema bloc

- Perspectivă inginerască
- Componentele sistemului sunt reprezentate prin blocuri (black box)
- Blocurile sunt conectate pentru a evidenția legăturile dintre ele



Schema bloc a arhitecturii Windows 2000



Domeniul de sistem

Modelele de politică sunt modele conceptuale care specifică regulile pe baza cărora se derulează procesele de afaceri și în particular cele pentru siguranța sistemului, a securității

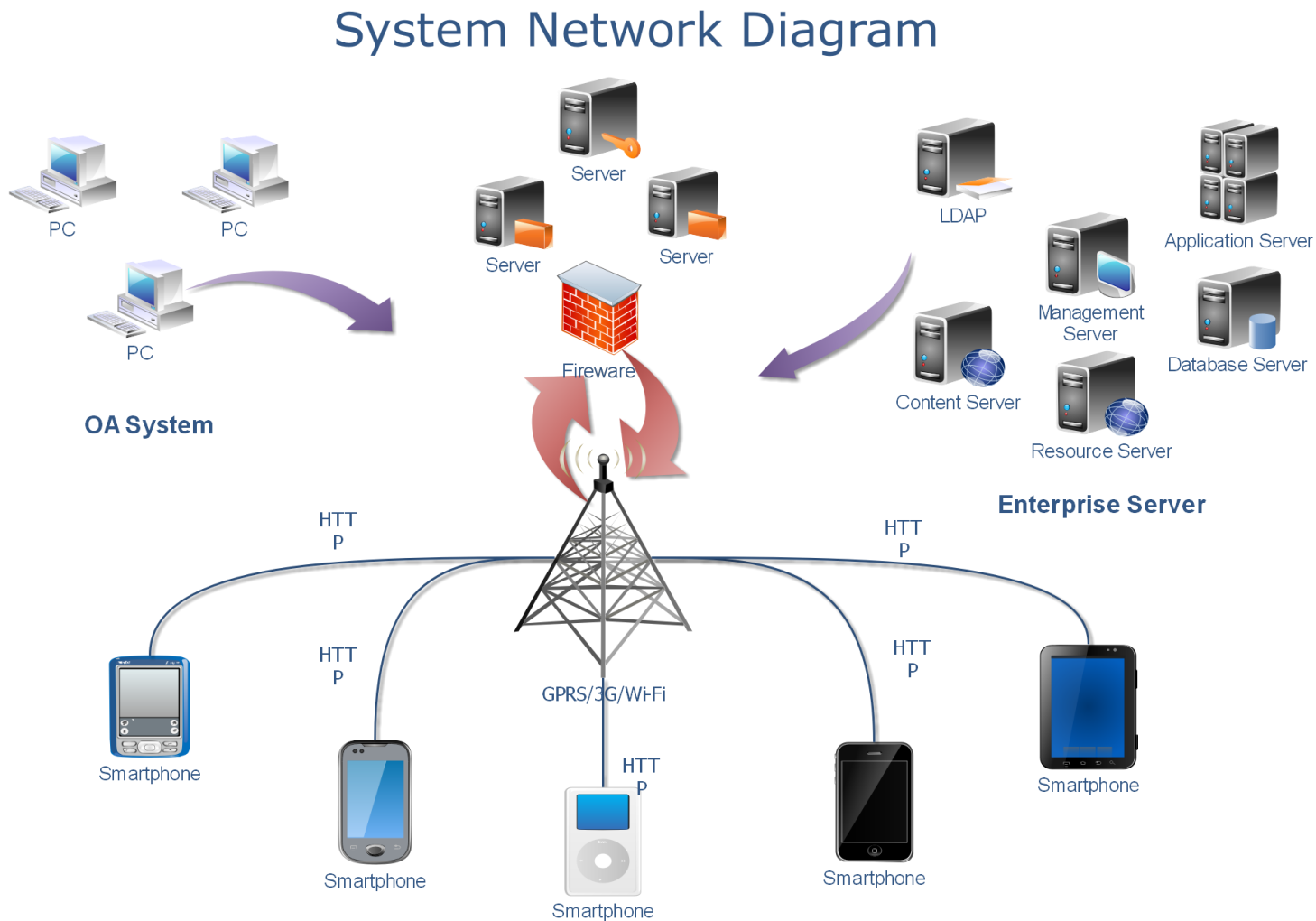
- Se încadrează în perspectivele de întreprindere

Modelele de rețea specifică structura logică și/sau fizică a sistemului din perspectiva comunicațiilor



Domeniul de sistem

Exemplu



Recapitulare – Domenii de abstractizare

Pentru a separa nivelurile de abstractizare, modelele se diferențiază prin perspective sau prin componenta specifică a sistemului pe care o descriu

Domeniul informațional

- Modelul informațional
- Modelul semantic
- Modelul datelor

Domeniul funcțional

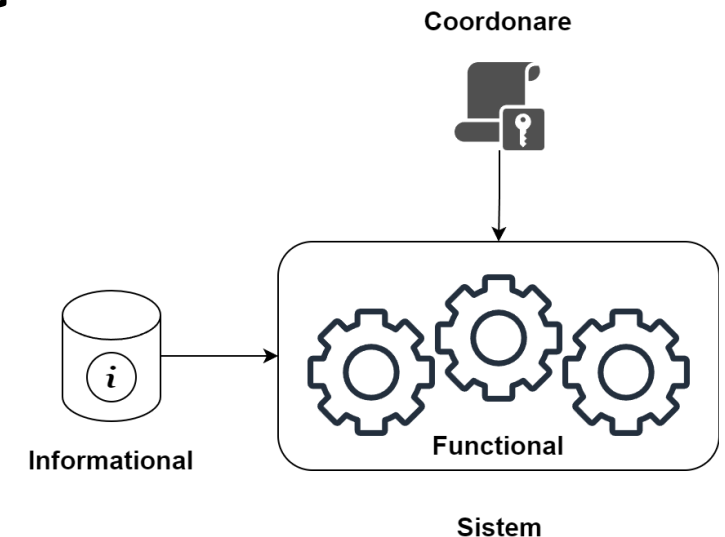
- Modele funcționale
- Modele de execuție
- Modele de stare
- Modele de proces

Domeniul de coordonare

- Modele de interfațare
- Modele de coordonare
- Protoale
- Modele de prezentare

Domeniul de sistem

- Modele de structură
- Modele de politică
- Modele de rețea



Modelarea domeniului și aspecte de integrare – studiu de caz

Scenariu

- În cadrul alegerilor prezidențiale recente, un candidat de extremă dreaptă a surprins prin câștigarea primului tur, deși nu era favorizat în sondaje sau mass-media.
- Succesul său a fost atribuit unei campanii de dezinformare pe TikTok, unde mesajele sale au devenit virale.
- Platformele tradiționale de monitorizare a alegerilor nu au putut identifica și contracara eficient răspândirea rapidă a dezinformării, evidențiind lipsa de integrare și capacitate de reacție a acestor sisteme.

Modelarea domeniului și aspecte de integrare – studiu de caz

Identificați aspectele de integrare și modelați sistemul având în vedere următoarele aspecte

- Monitorizare în timp real: Sistemele existente nu pot urmări eficient conținutul viral pe platformele descentralizate.
- Analiză de date: Detectarea informațiilor false în format scurt (video/meme) necesită algoritmi avansați.
- Integrare între platforme: Lipsa coordonării între TikTok și alte rețele sociale complică monitorizarea.
- Aspecte etice: Supravegherea social media ridică probleme de confidențialitate și reglementare.

Întrebări?

