

Integrarea sistemelor informatice



Suport curs nr. 3

Programator >> Arhitect

Sisteme software

2023-2024

C3 – Systeme software

Obiective

- Înțelegerea conceptelor teoretice legate de sisteme informatice
- Clasificarea sistemelor informatice din perspectiva integrării
- Identificarea nivelurilor de integrare în sisteme software

Sisteme software

Până acum am discutat în principal despre dezvoltare de software

Dezvoltarea reprezintă totuși doar o etapă în implementarea unei aplicații/unui sistem software

- Ce este un sistem software?
- Cine construiește acest sistem?

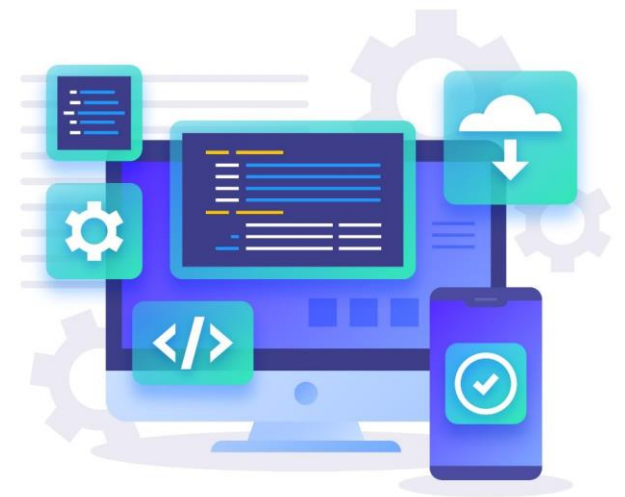
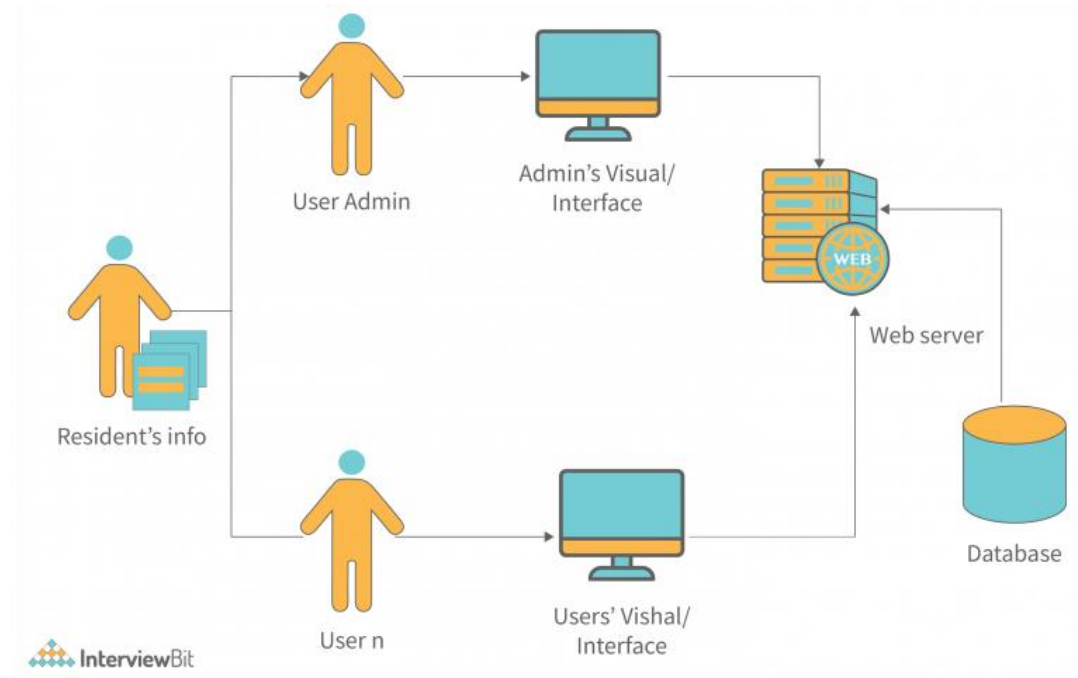


Image by Freepik

Sisteme software

Sistem

- Un ansamblu de software, hardware și de resurse umane care realizează împreună una sau mai multe funcții ale procesului de afaceri
- Un sistem poate fi pre-proiectat sau poate apărea ad-hoc prin acțiunea uneia sau a mai multor resurse umane sau software, participante



Sursa: InterviewBit, System Architecture – Detailed Explanation, 2023

Sisteme software

Proiectarea sistemului

- o specificație a structurii sistemului, incluzând perspectivele asupra componentelor, informațiilor și a proceselor
- o descompunere a funcțiilor sistemului în secvențe de subfuncții (grafice), atribuite subsistemelor componente, asociate cu o specificație pentru informațiile și materialele care trebuie să fie disponibile la nivelul interfețelor componentelor astfel încât subfuncțiile să poată fi realizate



Image by Freepik

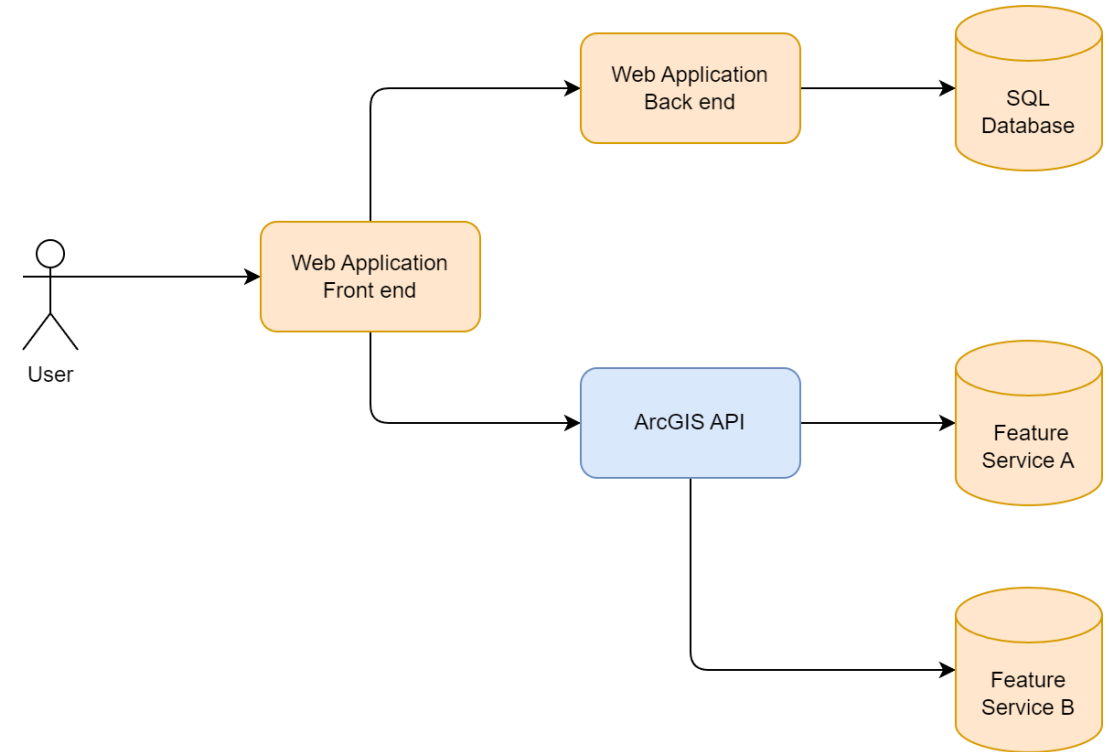
Sisteme software

Componentă (agent)

- un (sub)sistem care joacă un rol particular, prin furnizarea funcțiilor sale sub formă de subfuncții ale unui sistem mai mare

Componentă specifică sistemului

- un (sub)sistem conceput pentru a prelua un set specific de roluri definite într-un sistem mai mare pentru care sunt cunoscute apriori funcționalitățile și specificațiile de interfațare



Schemă bloc (exemplu)

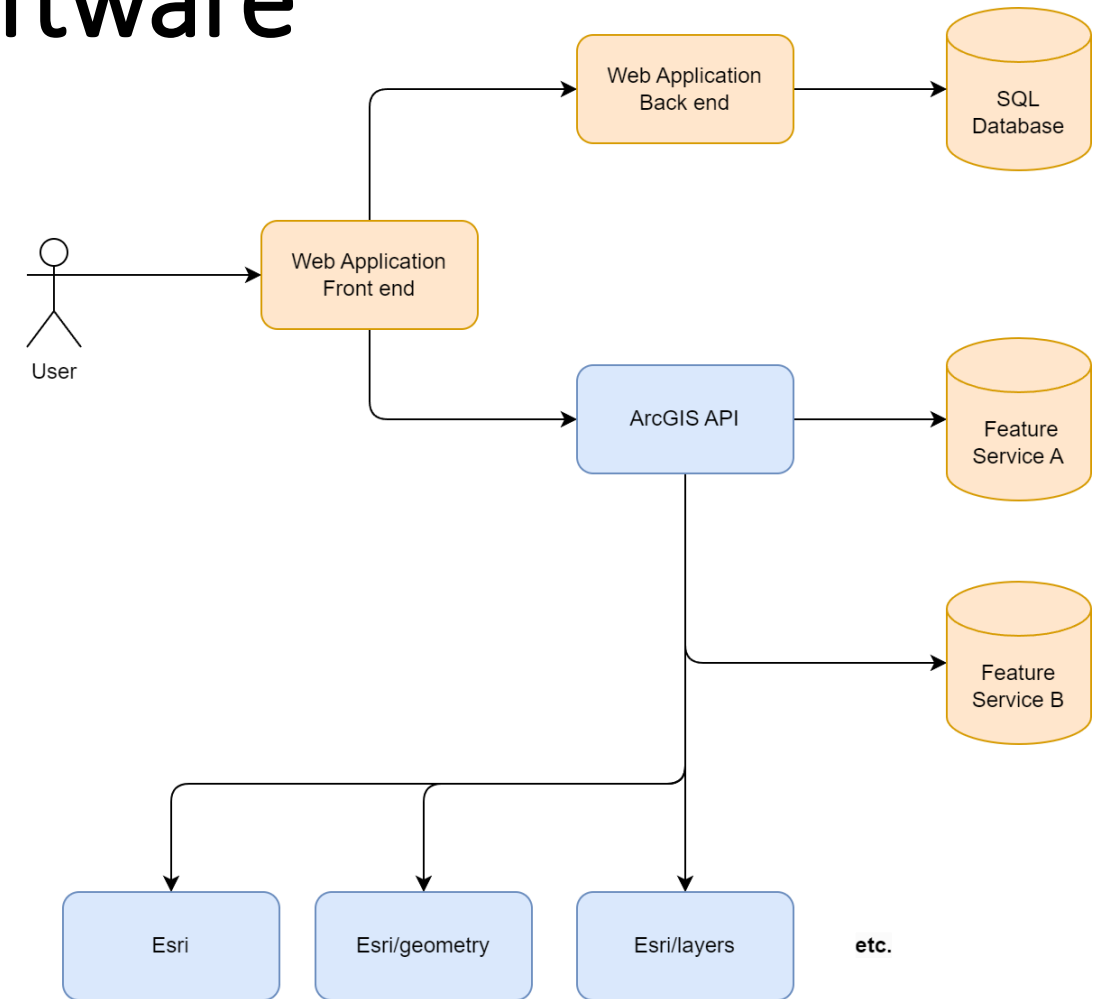
Sisteme software

Componentă reutilizabilă

- un sistem proiectat pentru a îndeplini un anumit set de roluri într-un sistem mai mare, fără ca proiectul sistemului mare și specificațiile de interfață pentru componentă să fie cunoscute apriori

Proiectarea unei componente reutilizabile

- Definirea fluxurilor de informații și a interfețelor explicite folosite
- Descrierea posibilităților de configurare pentru a expune mai multe variante comportamentale și interfețe, prin intermediul unor parametri

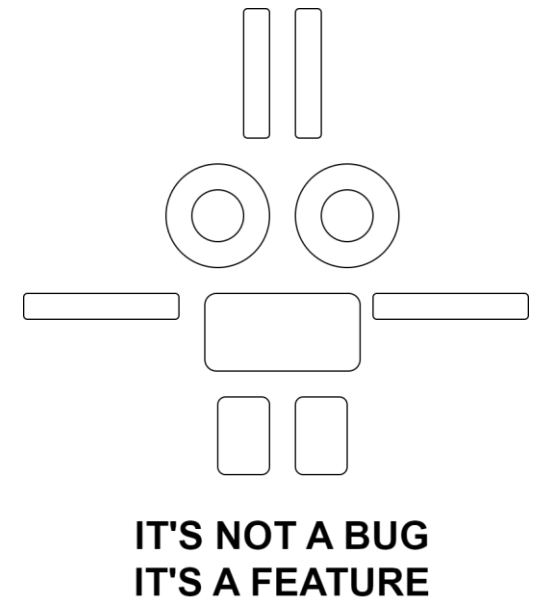


Schemă bloc (exemplu)

Sisteme software

Comportament

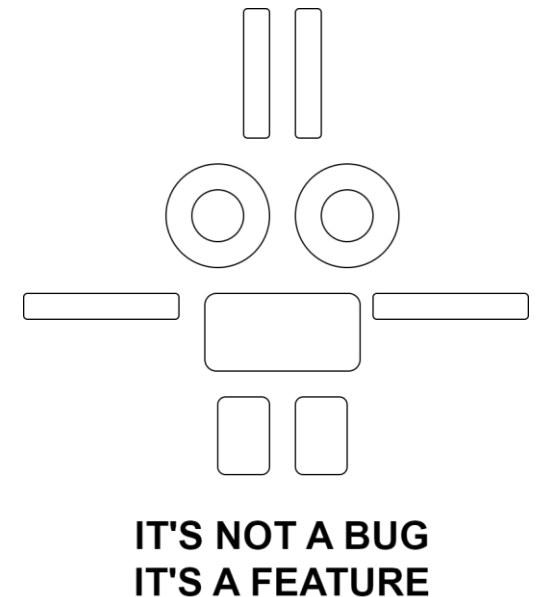
- Ceea ce execută sistemul/componenta – activitățile unei componente / sistem așa cum sunt percepute de sine, de alte componente ale sistemului și/sau de proiectantul sistemului, prin efectul asupra mediului și/sau prin attribute măsurabile.
- Comportamentul unui element al sistemului poate fi influențat de condițiile de mediu, de acțiuni din trecut, sau de configurare.



Sisteme software

Funcționalitate

- partea comportamentului unei componente care satisface anumite obiective sau scopuri
- Când comportamentul sistemului este modelat ca un set de tranziții ale stării în mediul observat, atunci funcția poate fi modelată ca subset al acestor tranziții care servește scopului sistemului.
- rezultatul generat de comportament

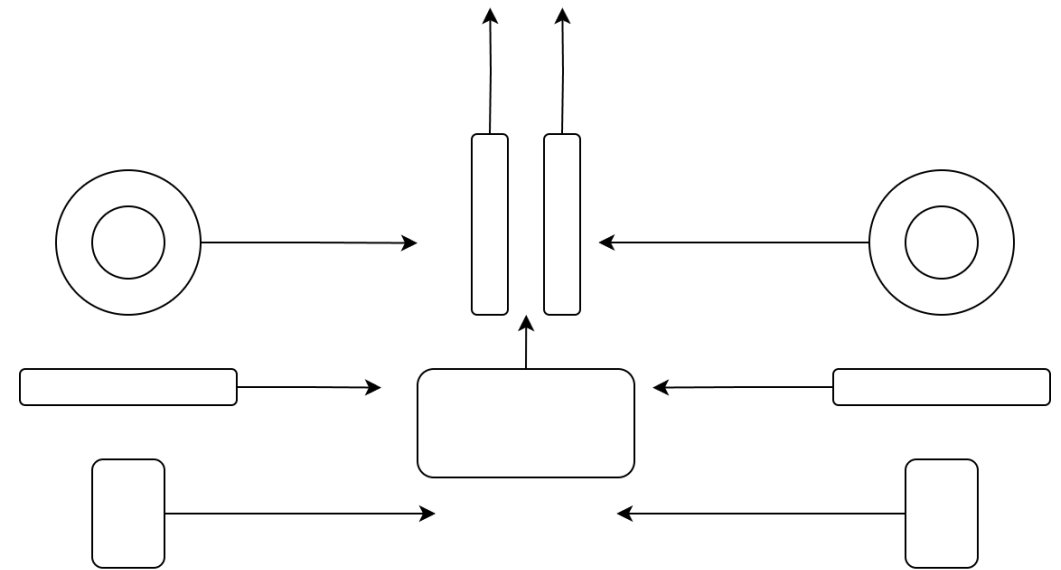


Sisteme software

Cum se proiectează o funcționalitate?

Proces

- descompunerea/detalierea unei funcționalități – ordonarea temporală sau logică (un grafic) de evenimente/**activități** care realizează o funcție sau mai multe funcții
- Un proces este considerat a fi **automat** dacă nu există nicio implicare umană în inițierea sau executarea acestuia.

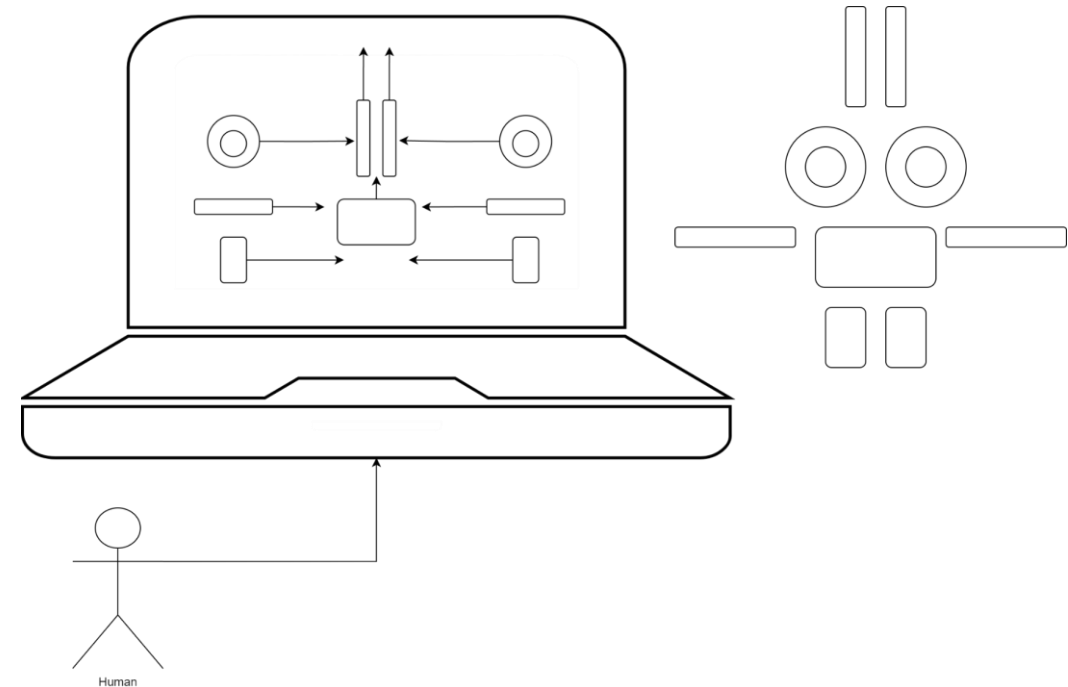


**IS IT A BUG?
IS IT A FEATURE?
NO, IT'S A PROCESS**

Sisteme software

Proces automatizat

- exclude **intervenția** umană după inițiere, cu excepția unor circumstanțe extraordinare.
- poate permite intervenția umană pentru a trece peste deciziile sale automate și poate solicita intervenția umană în cazul în care detectează o situație pentru care nu este prevăzut un algoritm de luare a **deciziilor**.



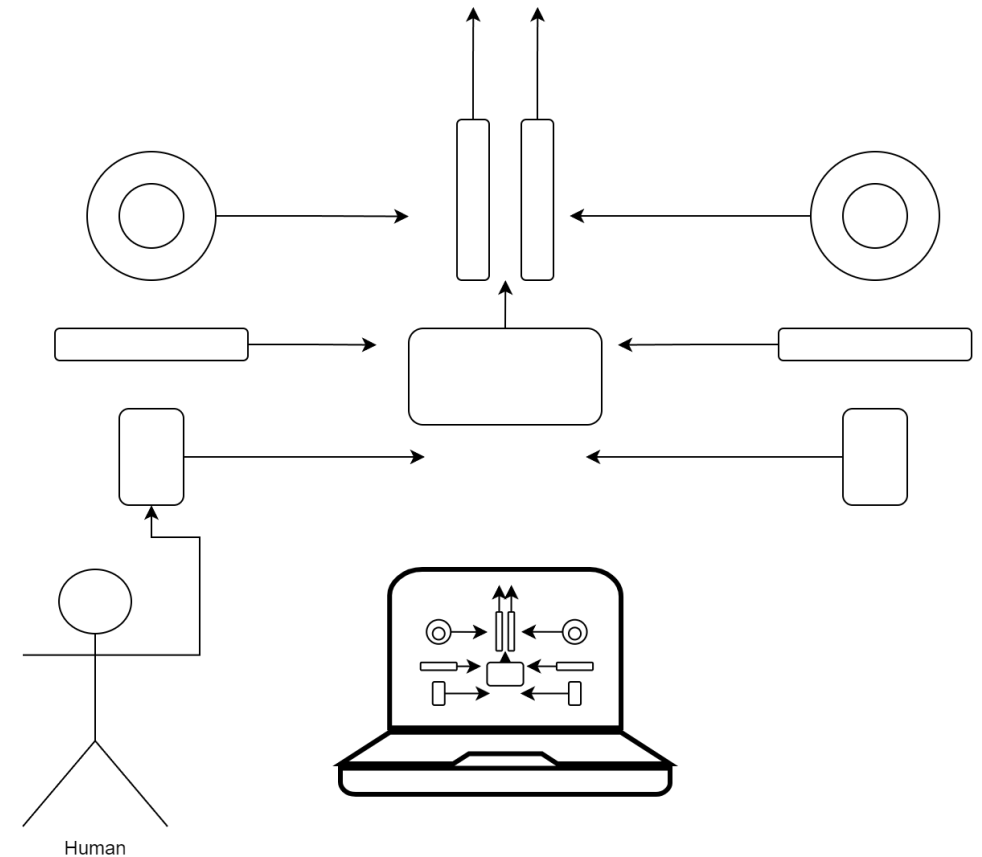
Sisteme software

Proces semi-automatizat asistat de calculator

- o parte din proces este automatizat, dar inteligența umană este angajată în procesul de luare a **deciziilor** importante și realizarea obiectivului

Proces manual

- nu există niciun ajutor semnificativ dat de calculator în executarea procesului



Sisteme software

Cum se dezvoltă un proces?

Resursă

- orice persoană, dispozitiv, sistem software, materiale, sau set de informații care efectuează, sau este utilizată în îndeplinirea unei sau mai multor funcții

Resursă activă (actor)

- o resursă care efectuează în mod direct una sau mai multe funcții și (de obicei) ia decizii în exercitarea funcției.

Agent

- o resursă activă, de regulă, un sistem software

Resursă pasivă

- materiale, echipamente sau informații folosite de resurse active în exercitarea unei funcții

Sisteme software

Acțiune comună

- Orice mod productiv în care mai multe resurse participă activ la realizarea unei funcții, inclusiv de coordonare, colaborare, cooperare, asistență involuntară, chiar și concurență
- Factorul critic este faptul că interacțiunea între resurse implică comunicarea din partea fiecărui actor participant, iar acțiunea comună realizează o parte dintr-o funcție a sistemului integrat

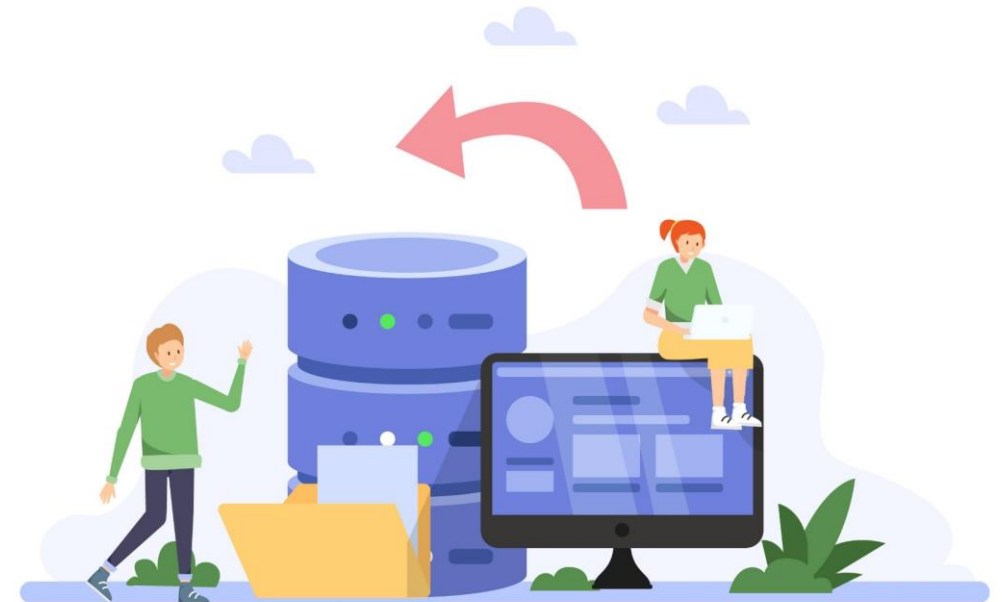


Image by Freepik

Sisteme software

Informația

- o secvență de simboluri, care poate fi interpretată ca un mesaj
- orice fel de eveniment care afectează starea unui sistem dinamic, care poate interpreta informații
- cunoașterea comunicată sau primită cu privire la un anumit fapt sau circumstanță
- un răspuns la o întrebare

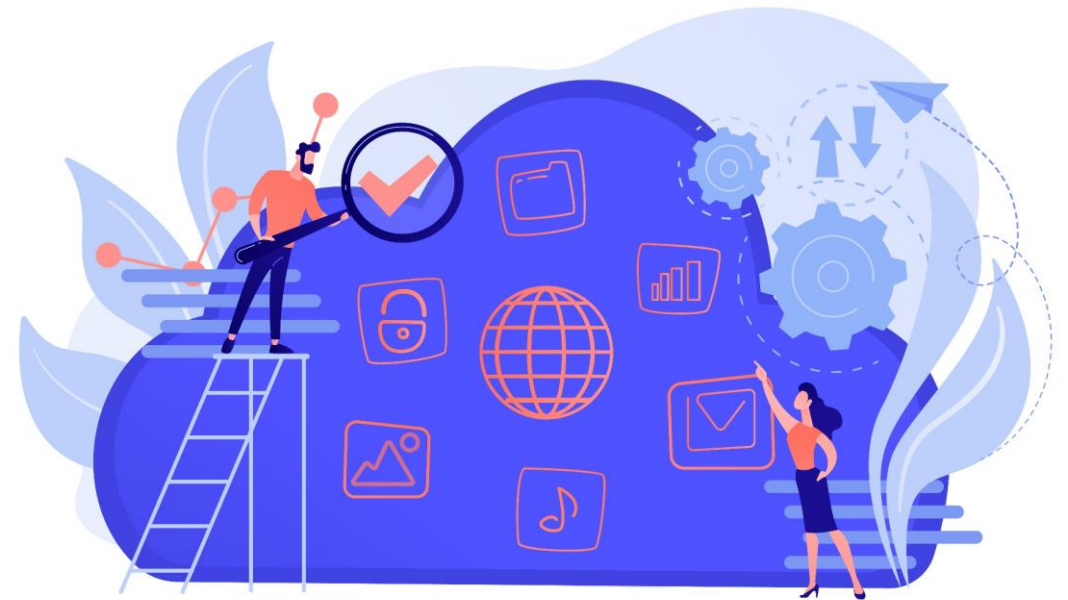


Image by Freepik

Sisteme software

Comunicația

- orice operație care are ca rezultat un flux de informații de la un actor la altul, inclusiv:
 - redarea de informații interne
 - generarea de date de către diferite mecanisme
 - interpretarea datelor primite de la surse interne
- schimbul de date între actori în exercitarea unei acțiuni comune, cu conținut semantic suficient conservat (prin fluxul de date și operațiuni de interpretare a datelor) astfel încât acțiunea comună să își atingă scopul



Image by Freepik

Sisteme software

Mecanism de comunicație

- modul în care are loc transferul de informație între resurse.
- caracterizat de comportamentul fundamental asociat operațiunilor de comunicare
- mecanism de nivel mai mare de abstractizare

Protocol de comunicație

- regulile detaliate ale fluxurilor de informație

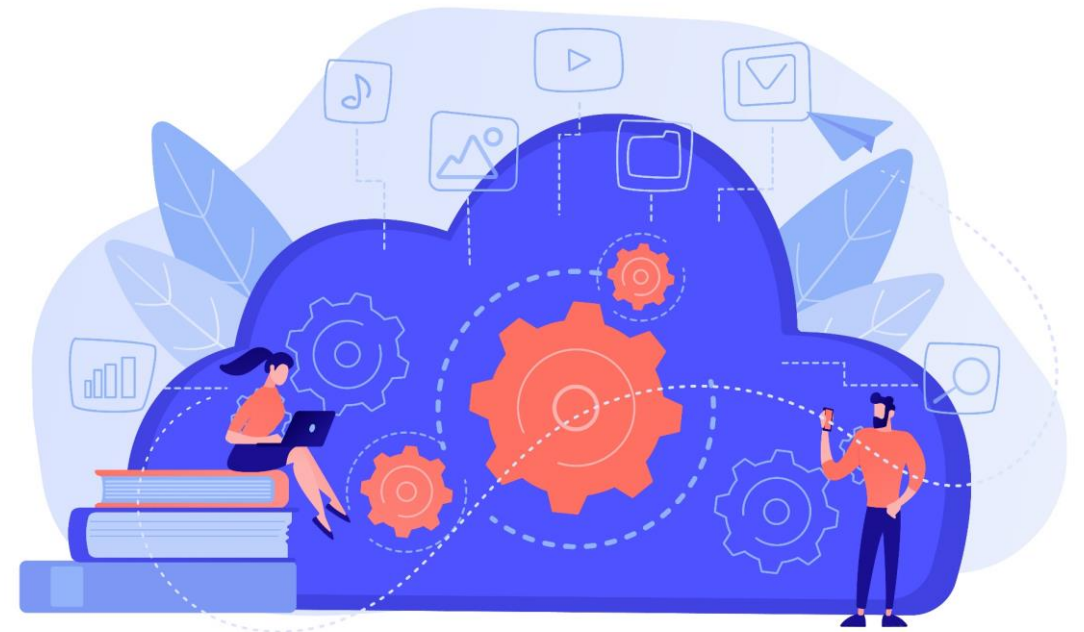


Image by Freepik

Sisteme software

Resurse de comunicație

- Fișiere sau documente
- baze de date partajate
- apel de procedură sau invocarea unei operații
- mesaje sau fluxuri de date (stream)
- semnale (fizice sau software)
- mecanisme de interfațare umană (display, tastatură, mouse, etc.)



Image by Freepik

Sisteme software auto-integratoare

Auto-integrarea

- cerințele tehnice de comunicare sunt soluționate în mod automat de către agentul de comunicare, spre deosebire de a fi pre-rezolvate în proiectarea sistemului/agentului și pre-programate în componente

Sistem auto-integrator

- un sistem capabil de a se reconfigura și/sau a configura comunicațiile astfel încât să coopereze cu alte sisteme pentru a satisface noile cerințe
 - Sistem configurabil prin proiectare
 - Sistem prevăzut cu “mecanisme de învățare”



Image by Freepik

Sisteme software auto-descriptive

Cerințe tehnice

- tehnologie comună pentru transmiterea și primirea datelor
- protocoale comune pentru transferul de informație
- structura și reprezentare comune, pentru date

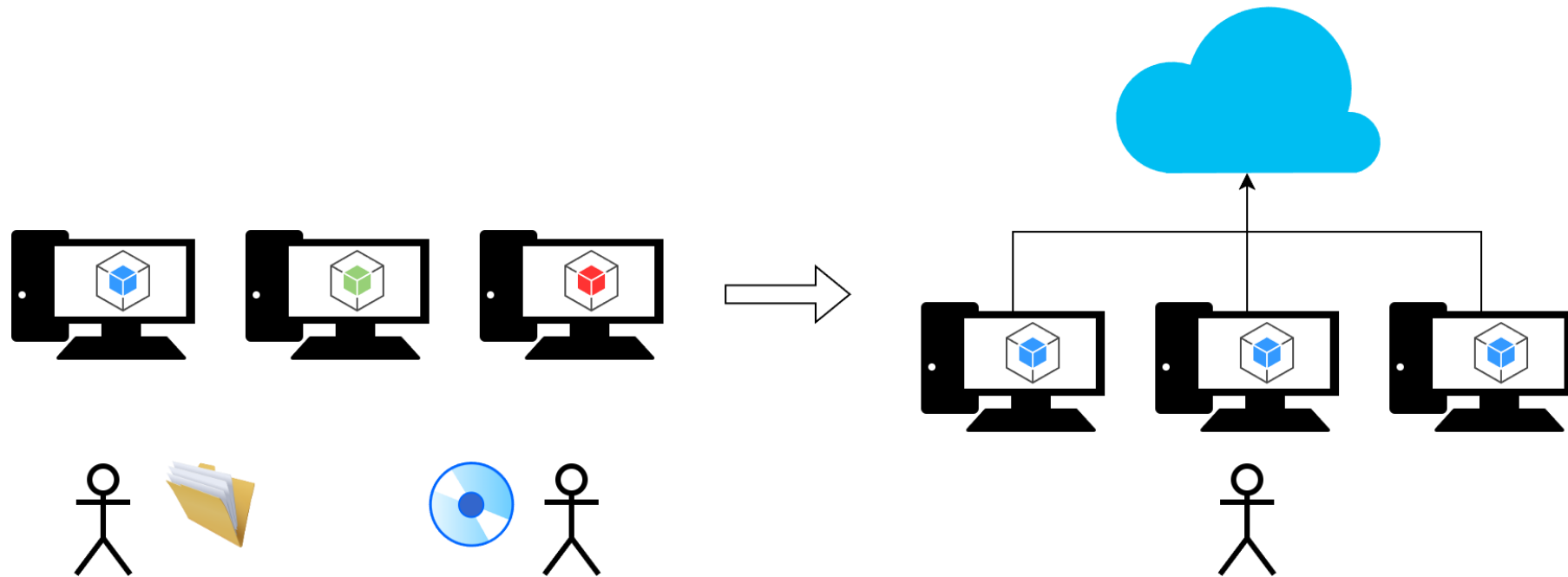
Sistem auto-descriptiv

- un sistem auto-integrator, conectat la alte sisteme în rețea, care își publică capacitățile (rolurile posibile) și determină în mod automat a capacitățile (rolurile) celorlalte sisteme
- scopul auto-descrierii este de a facilita re folosirea componentelor

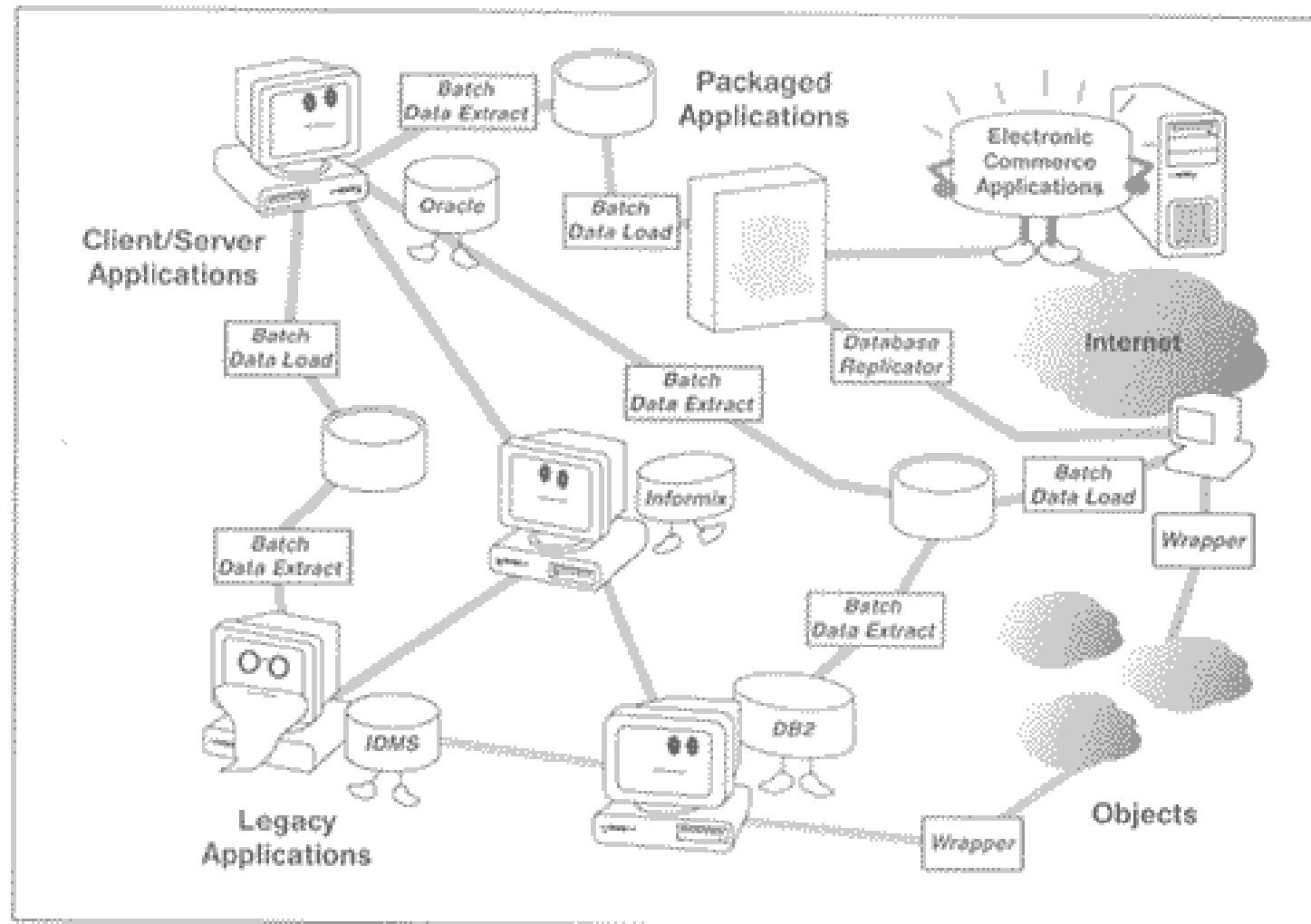


Image by Freepik

Clasificarea sistemelor informatice din perspectiva integrării



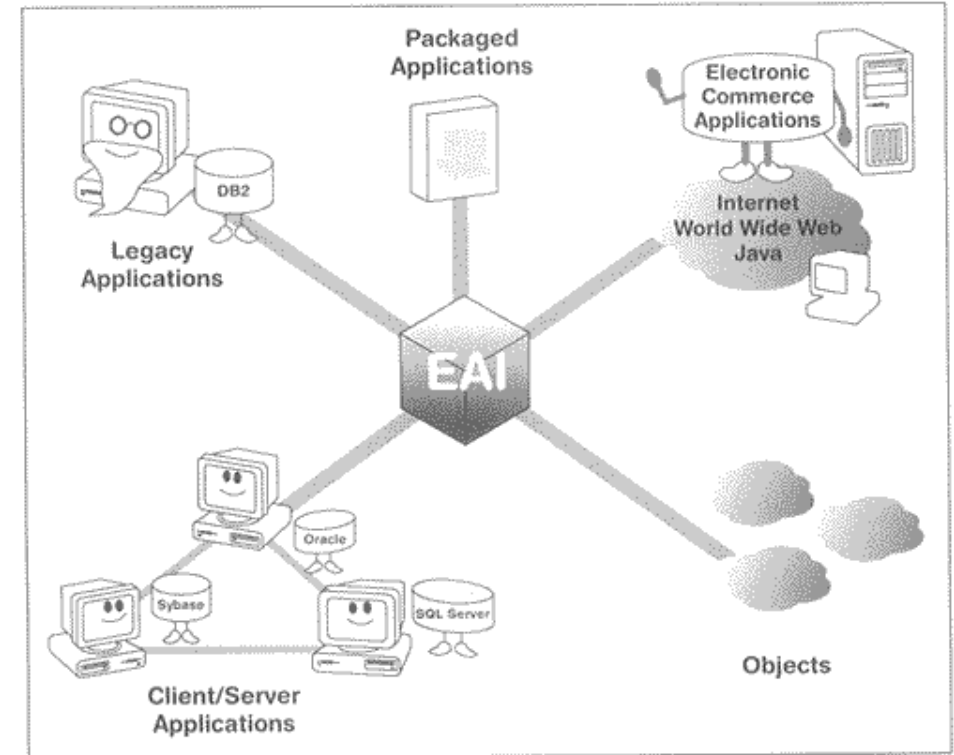
Sisteme informatice eterogene



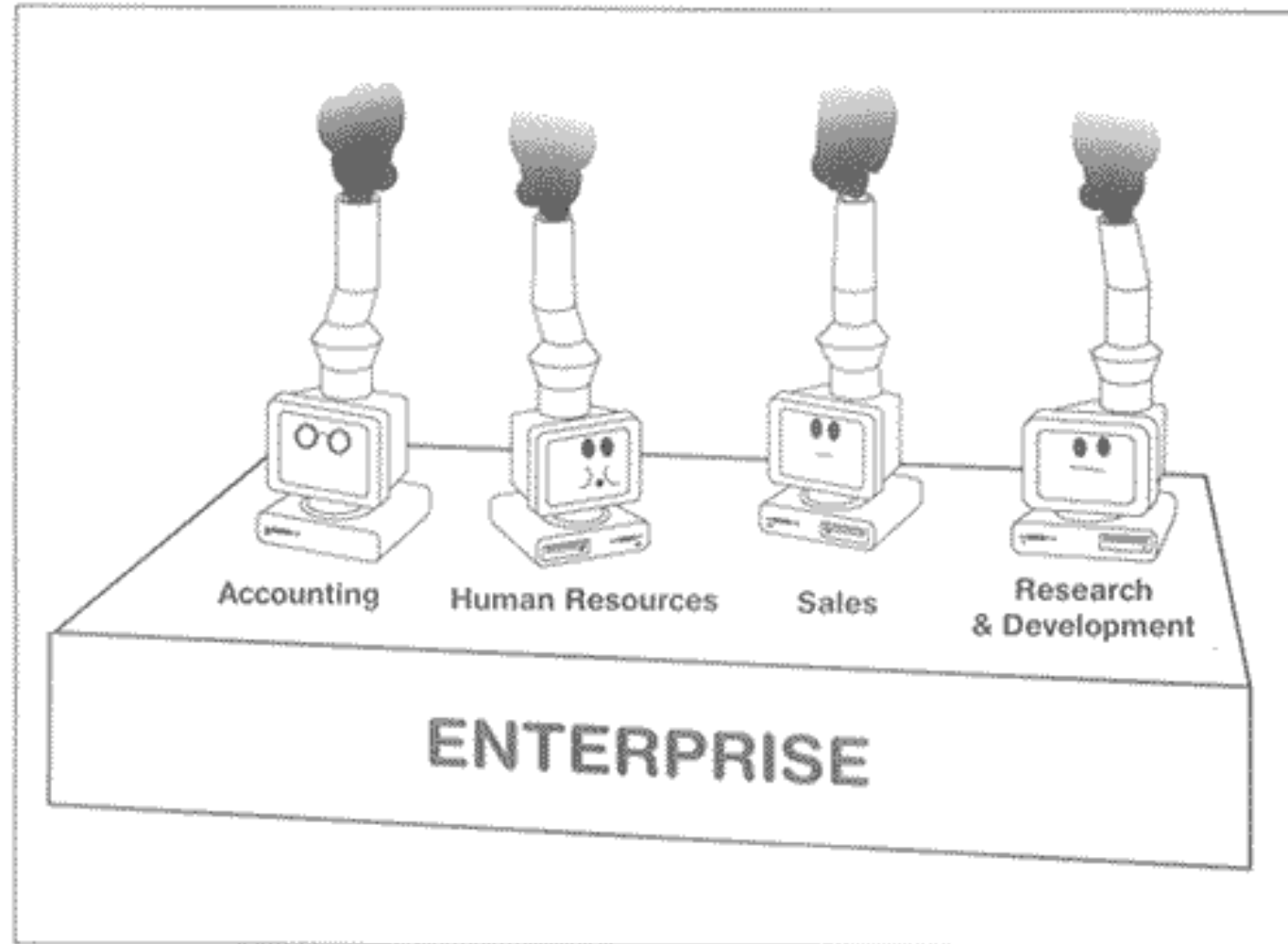
Sisteme informatice integrate

Proiectarea sistemelor integrate

- Înțelegerea arhitecturii de ansamblu, a proceselor și datelor existente, într-un context dat
- Identificarea componentelor care necesită schimb de informație (și sub ce formă)
- Utilizarea tehnologiilor moderne pentru a facilita integrarea aplicațiilor



Aplicații independente – “Stovepipe”



Sisteme tradiționale / legacy

- Sistemele tradiționale sunt aplicații independente disponibile în sisteme centralizate
- Ați auzit de calculatoarele mainframe care ocupau o cameră întreagă?
 - Da, chiar mai există, și sunt folosite tot mai mult în aplicații bancare, științifice, guvernamentale, etc.
- Caracteristici:
 - Aplicații independente
 - Arhitecturi de tip “monolit”
 - integrarea cu alte aplicații este dificilă sau inexistentă



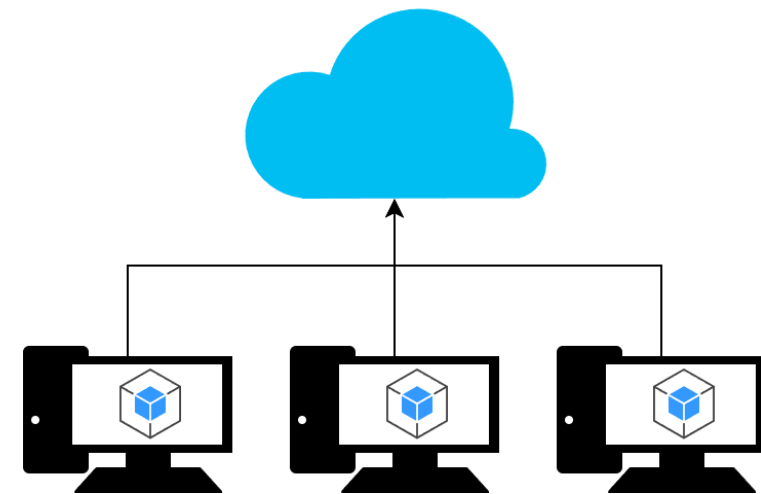
Sisteme de tip microcalculator / PC

- Aplicațiile PC sunt destinate pentru a rula local pe calculatoarele personale
 - integrarea informațiilor este dificilă (formatul datelor, diferențe de configurare)
 - procesele care rulează nu pot fi eficient accesate din exterior

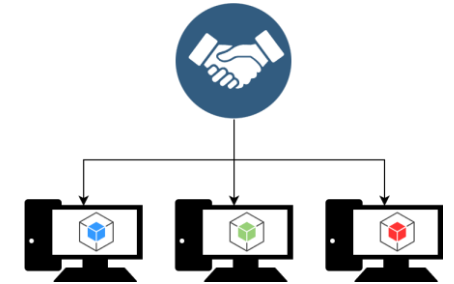


Sisteme distribuite

- Sisteme cu multiple componente care rulează pe mașini diferite, interconectate prin rețea
 - client/server
 - internet/intranet
 - arhitecturi de prelucrarea a obiectelor distribuite
- Beneficii: scalabilitate, paralelizare, toleranță la defecte
- Dezavantaje: dificil de implementat



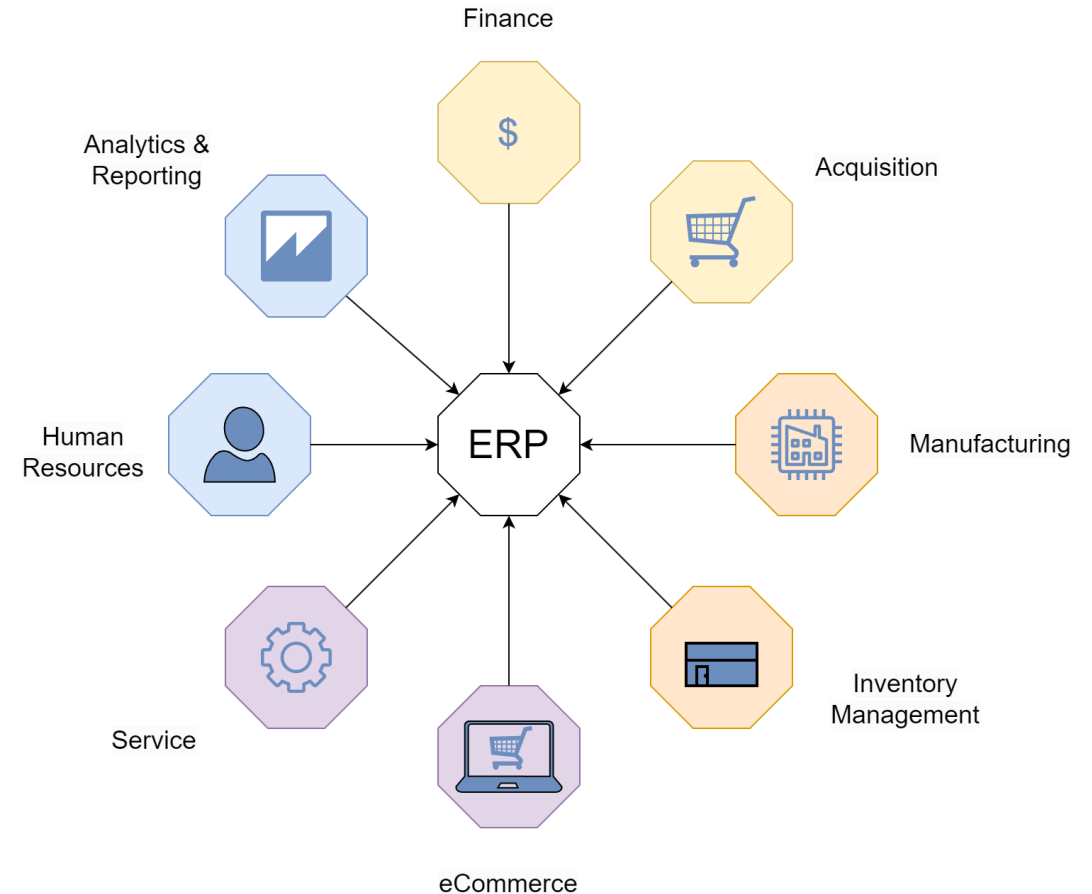
Aplicații încapsulate



- “Packaged Applications” – aplicații care sunt în general achiziționate, nu dezvoltate, eventual personalizate
- Conțin procese de afaceri reutilizabile, bazate pe bune practici și nu presupun efort mare de dezvoltare
- Exemple

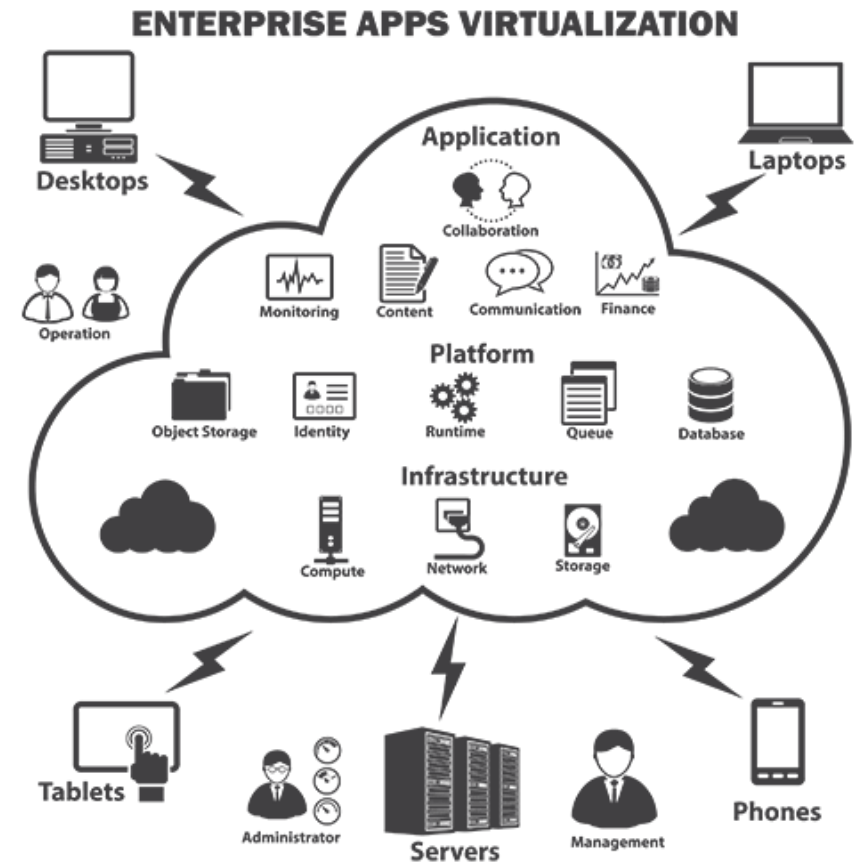


- ERP (Enterprise Resource Planning)
- CRM (Customer Relationship Management)
- Aplicații de suport vânzări



Sisteme virtuale

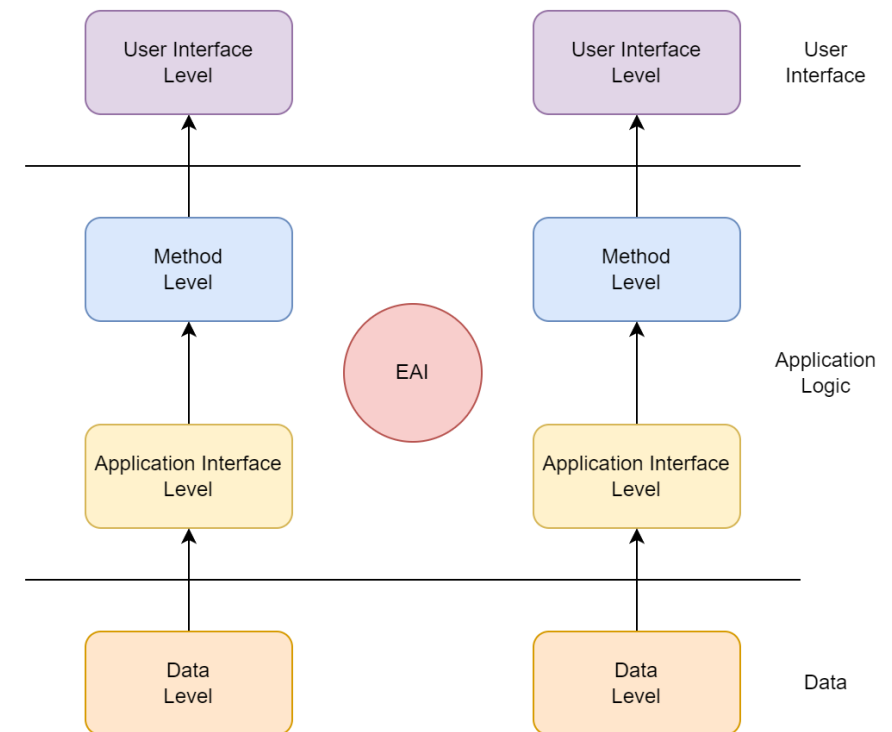
- Cel mai înalt nivel de integrare
- Aplicațiile diferite sunt disponibile într-un format unificat și universal accesibil
- Exemple
 - Microsoft 365
 - Google Workspace
 - Adobe Creative Cloud



Sursa: Enterprise Apps Virtualization

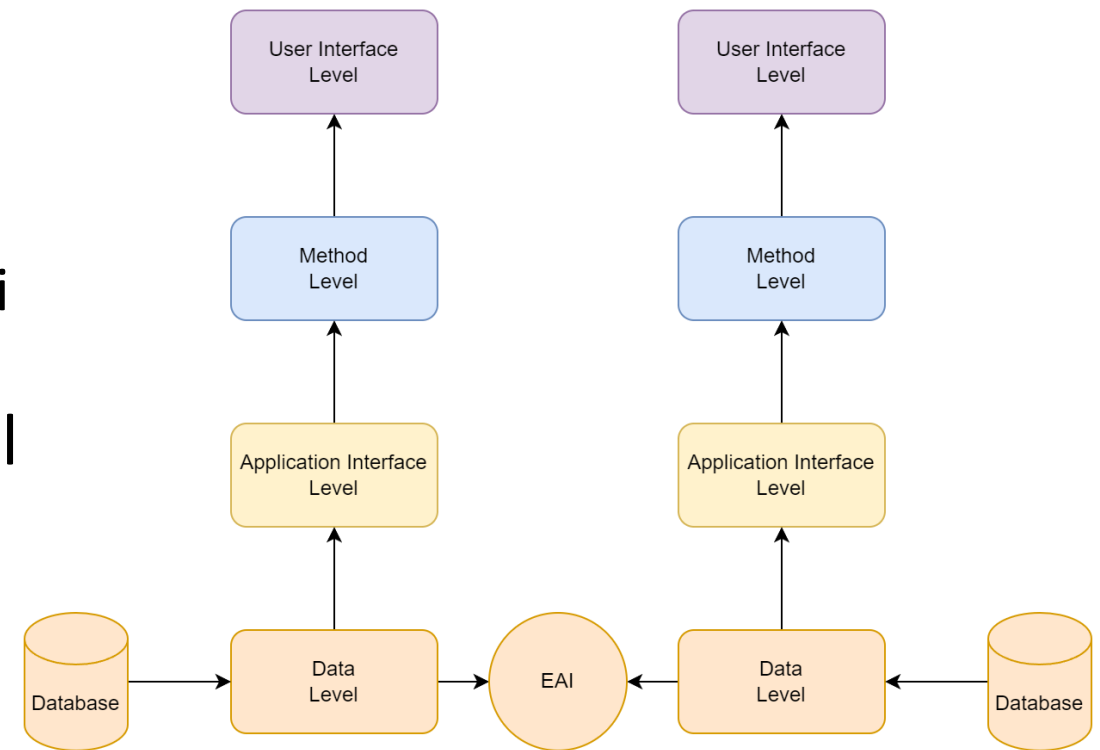
Tipuri de integrare a aplicațiilor pentru întreprinderi

- **EAI** – Enterprise Application Integration
- Pentru a integra mai multe aplicații, organizațiile trebuie să înțeleagă atât procesele de afaceri cât și datele (informațiile) aferente
- Niveluri de integrare
 - Nivelul **de date**
 - Nivelul **funcțional** / logică aplicație
 - Nivelul **interfeței utilizator**



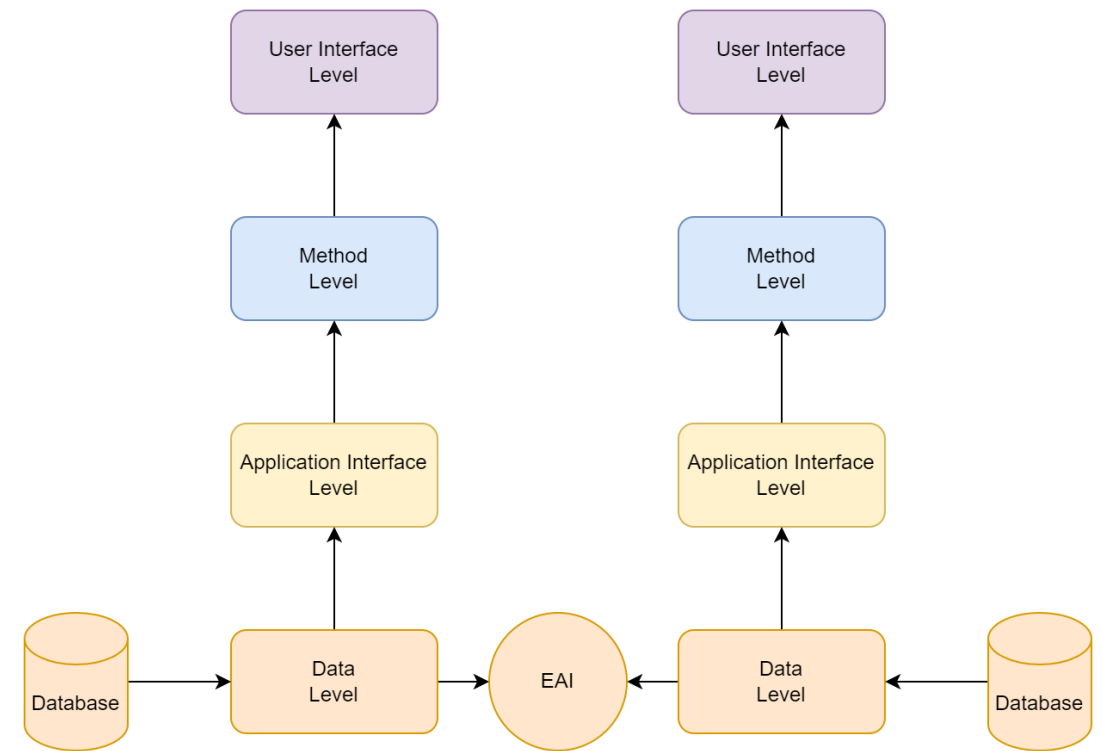
Integrarea sistemelor – nivelul de date

- Procesul de transfer al datelor între diferite locații – implică tehnici și tehnologii pentru **ETL**
 - (extract, transform, load)
 - **Extragerea** informației
 - **Prelucrarea**/transformarea informației
 - **Actualizarea**/încărcarea în altă locație
- Sursa datelor (modelul de date, tipul bazei de date)
 - Baze de date **relaționale**
 - Baze de date **orientate pe object**
 - Baze de date **multidimensionale** (data warehousing)
 - Alte tipuri



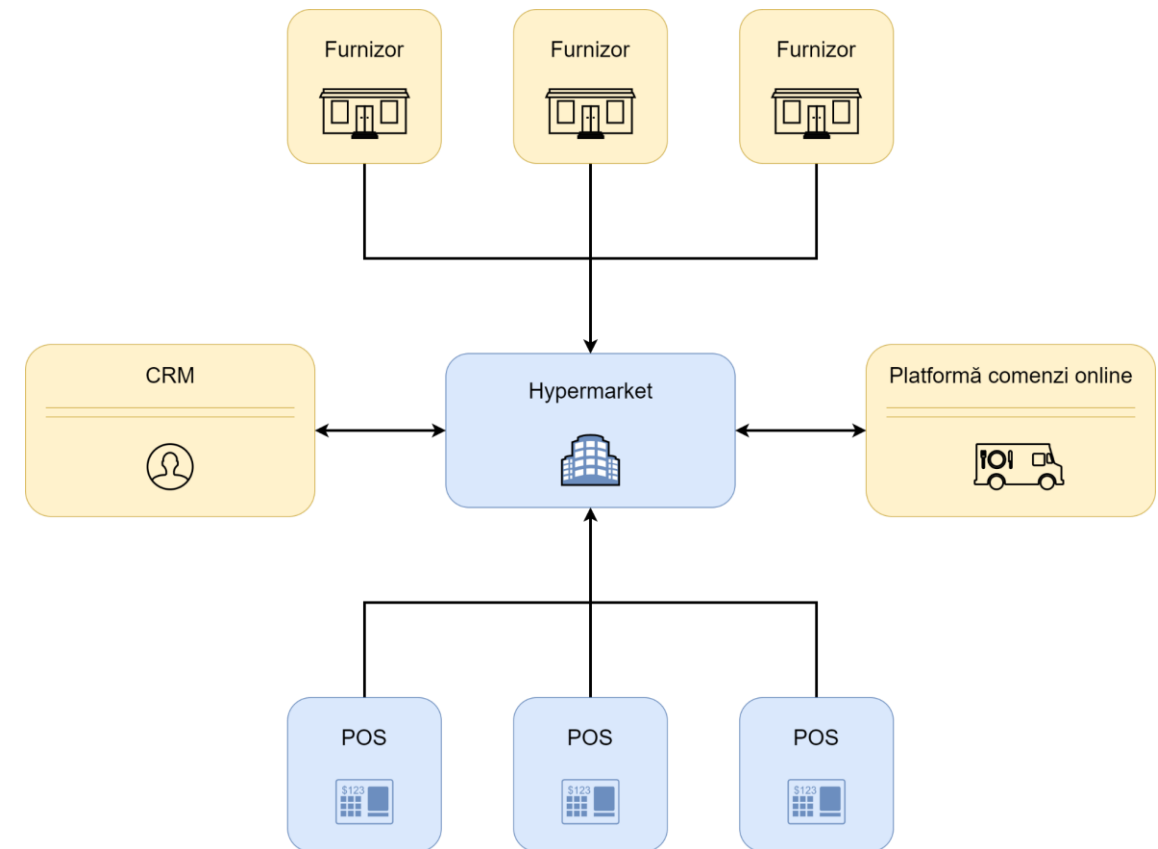
Integrarea sistemelor – nivelul de date

- **Beneficii** – Cost scăzut față de alte metode de integrare, deoarece nu presupune modificări majore de cod sau teste extensive
- **Dificultăți** – Migrarea datelor (adaptarea aplicațiilor de suport al proceselor de afaceri la tipul și structura datelor), când se folosesc concomitent diferite tipuri de baze de date



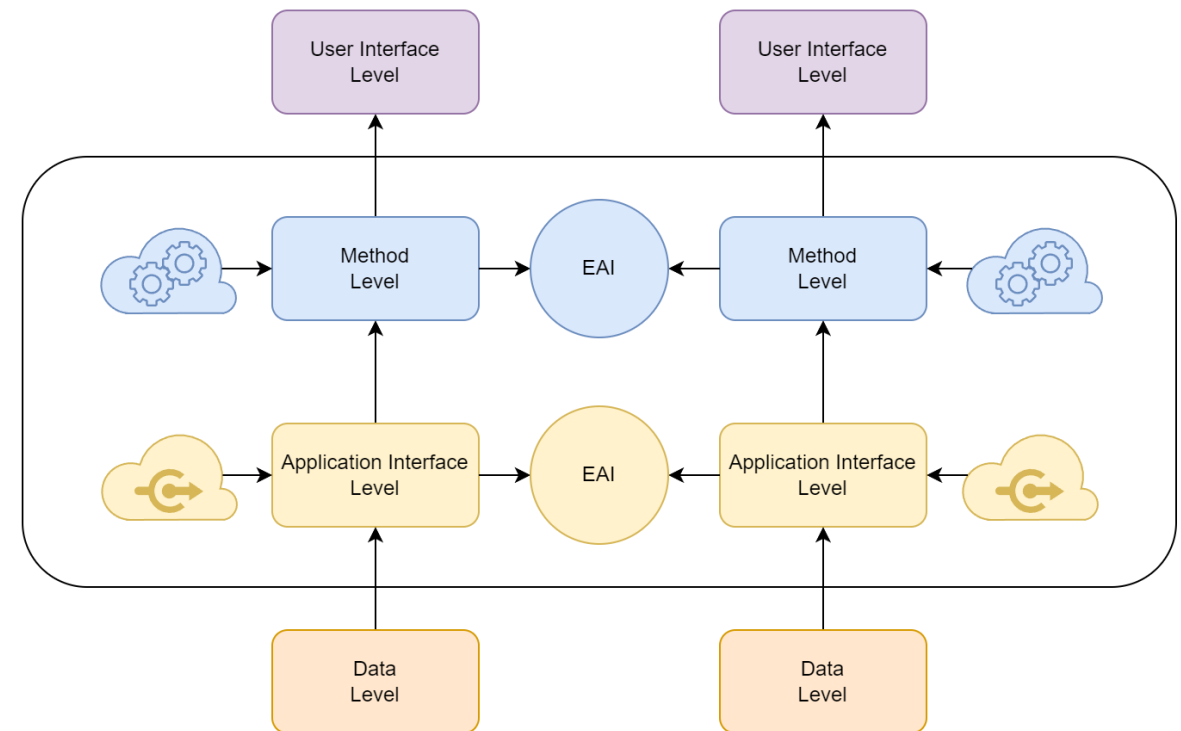
Integrarea sistemelor – nivelul de date

- Modul cel mai frecvent de integrare, în care datele sunt folosite de diverse aplicații, unele chiar independente
- Accesul la baze de date este relativ simplu, fiind standardizat pentru fiecare tip folosit – ODBC (Open Database Connectivity)
- Exemplu – Sistem de gestionare a inventarului într-un hypermarket



Integrarea sistemelor – nivelul funcțional

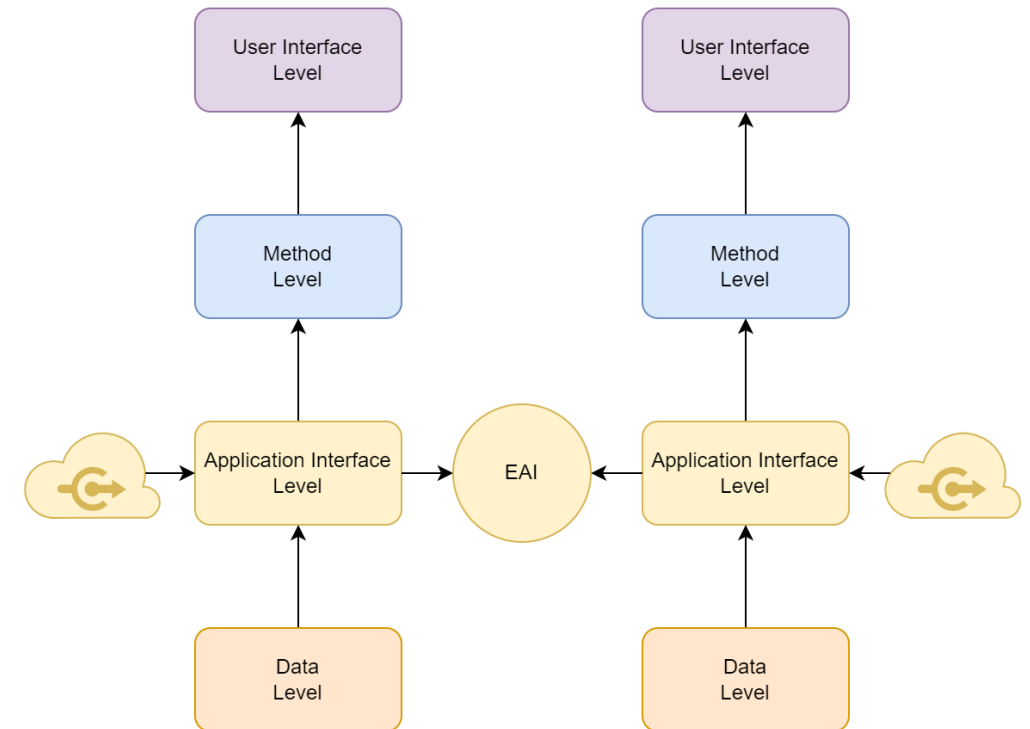
- Se referă la utilizarea **interfețelor** expuse de aplicații pentru a accesa datele și/sau procesele de afaceri
- Dezvoltatorii trebuie să aibă în vedere doar caracteristicile specifice și funcționalitățile interfețelor aplicațiilor
- Se utilizează frecvent în cazul integrării aplicațiilor încapsulate care expun interfețe către procesele și datele lor
 - ex: SAP, PeopleSoft, Baan, etc.
- Soluția cea mai frecventă este prin broker de mesaje



Integrarea sistemelor – nivelul funcțional al interfețelor

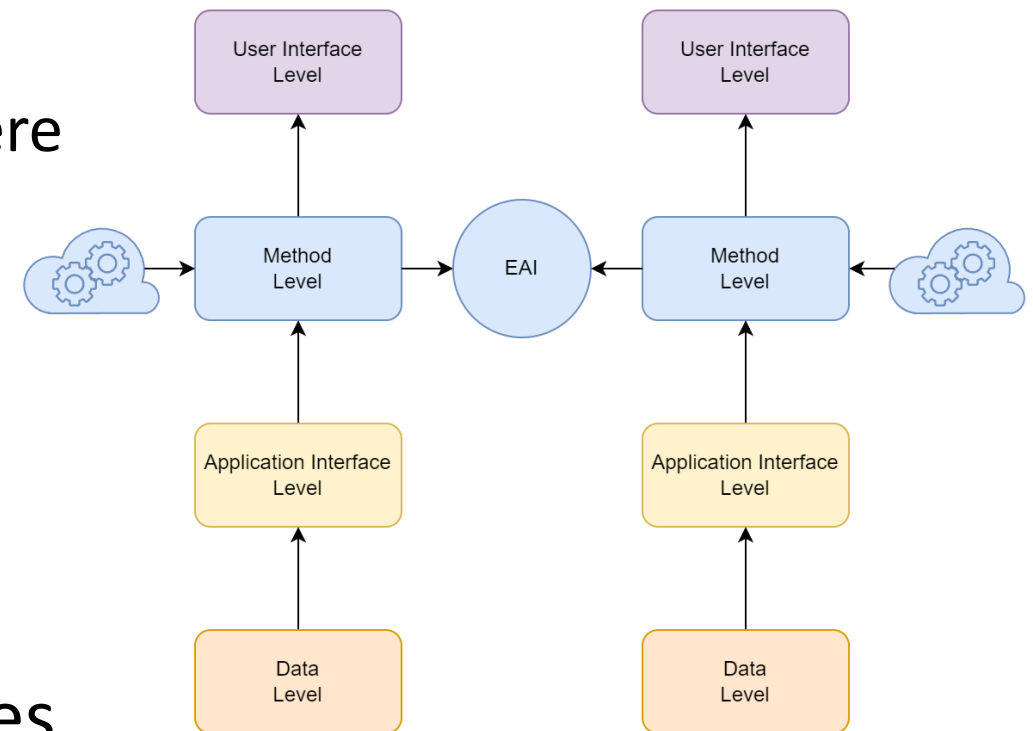
Interfețele aplicațiilor

- Sunt interfețele pe care le expun dezvoltatorii pentru a avea acces la diferitele niveluri sau servicii ale aplicațiilor (fără acces la cod)
- Unele interfețe au capabilități multiple, "feature-rich", altele sunt restrictive. Accesul poate fi la procese de afaceri, la date sau la ambele
- Exemple – aplicații care folosesc API-ul ChatGPT



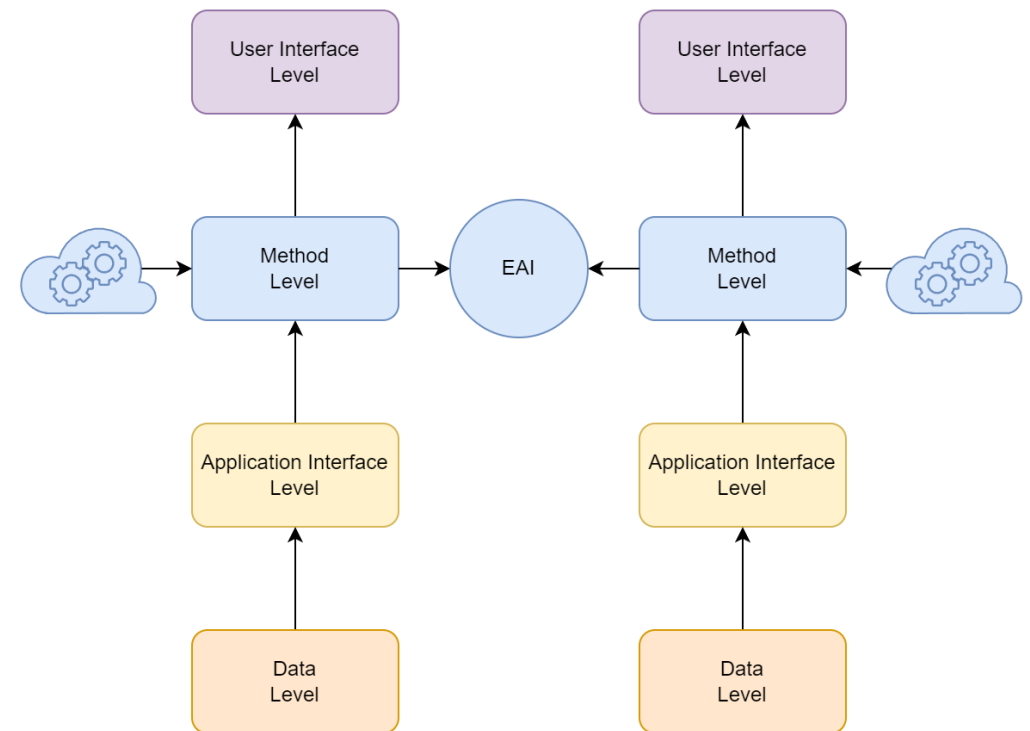
Integrarea sistemelor – nivelul funcțional al metodelor

- Folosirea în comun a unor **metode** de către mai multe aplicații
 - Se poate crea un set comun de servere de aplicații (**servicii web**)
 - Se pot utiliza în comun metode existente în aplicații, folosind tehnologii de utilizare în comun ale metodelor, cum ar fi obiectele distribuite
- În acest fel se pot folosi în comun procese de afaceri (cu sau fără acces la cod)



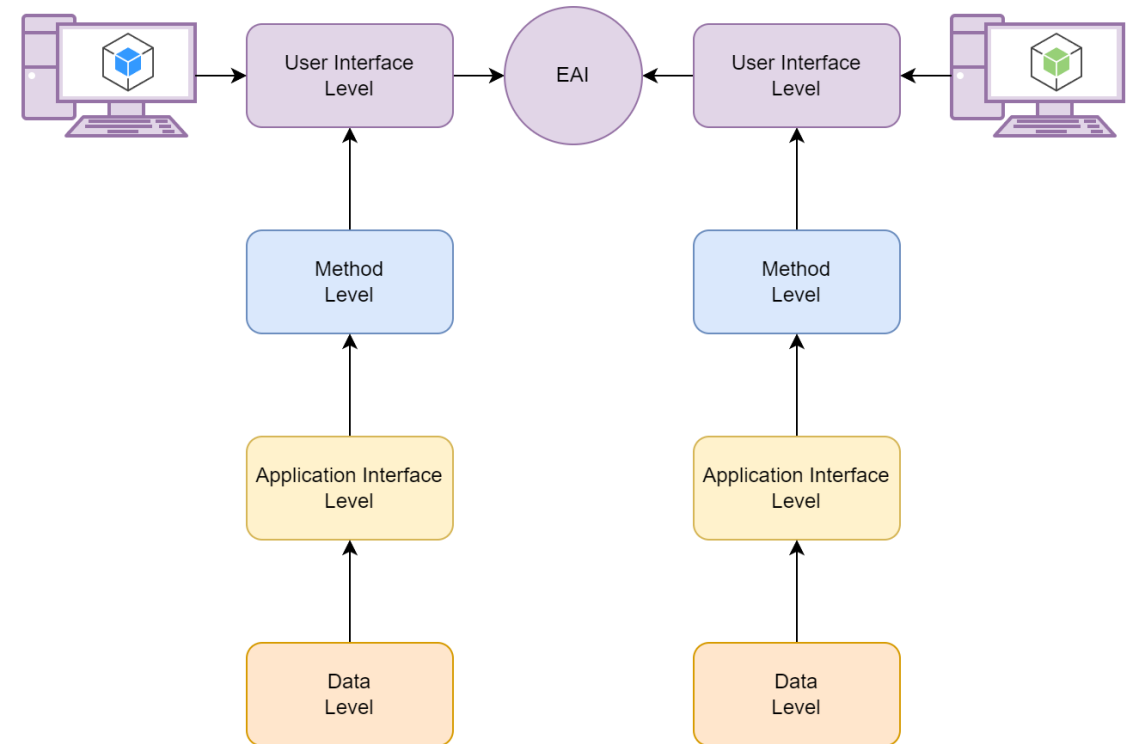
Integrarea sistemelor – nivelul metodelor

- Metodele includ
 - Scenarii
 - Reguli
 - Logică
 - Date
 - Obiecte
- Exemple de utilizare
 - Framework-uri
 - Obiecte distribuite
 - Servere partajate



Integrarea sistemelor – nivelul interfeței utilizator

- Integrarea se face prin cumulara mai multor interfețe utilizator (screen scraping)
- Se poate folosi în aplicațiile care nu permit acces la nivelul bazei de date sau al proceselor de afaceri
- În anumite cazuri, utilizatorul este singurul capabil să acceseze logica și datele într-un mod coerent, reprezentativ pentru afacere
- Nu necesită modificări ale aplicațiilor integrate



Integrarea sistemelor – nivelul interfeței utilizator

- Etape
 - Înțelegerea aplicației
 - Crearea unui catalog al ecranelor
 - Maparea ecranelor
- Exemple
 - Widget-uri de prognoză meteo
 - Microsoft 365, Google Workspace (partea de integrare a interfețelor aplicațiilor Office)



Bibliografie

- M. Mocanu, curs Integrarea Sistemelor Informatice, 2022
- [David S. Linthicum, Enterprise Application Integration](#)
- [IBM, Concept: Enterprise Application Integration](#)

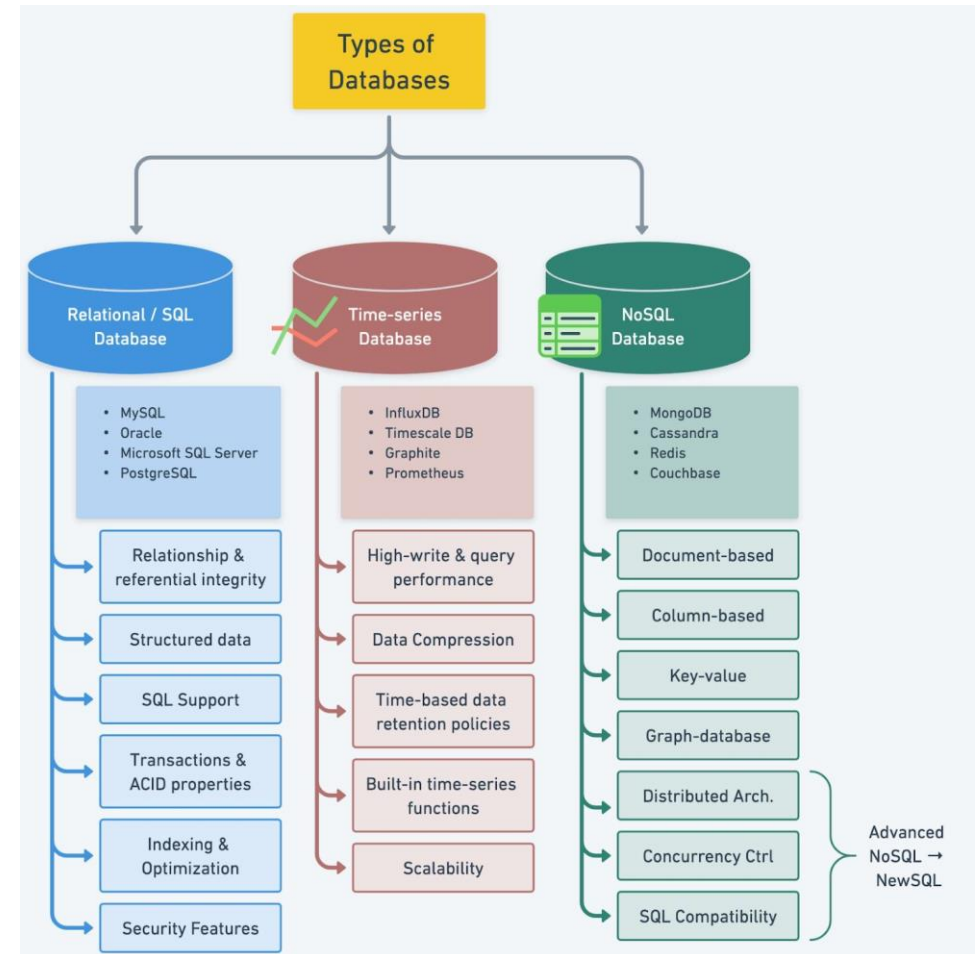


C3/p – Integrarea sistemelor în practică

Nivelul de date

Baze de date

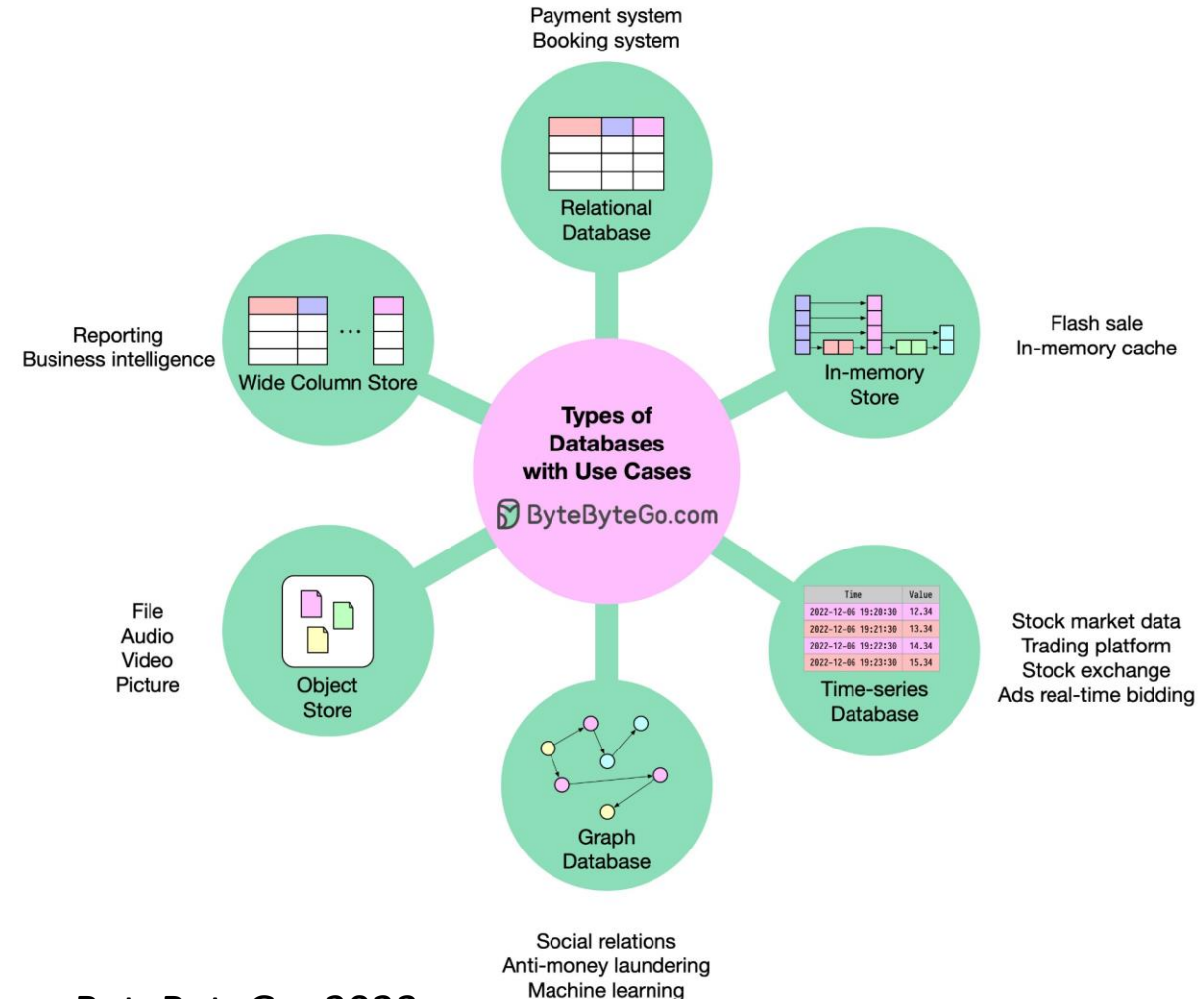
- Relaționale / SQL
- Nerelaționale / NoSQL
 - Documente
 - Grafuri
 - Cheie-valoare (en. key-value)
- Serii de timp / Time-series



Nivelul de date

Cum alegem tipul bazei de date în funcție de **tipul aplicației**

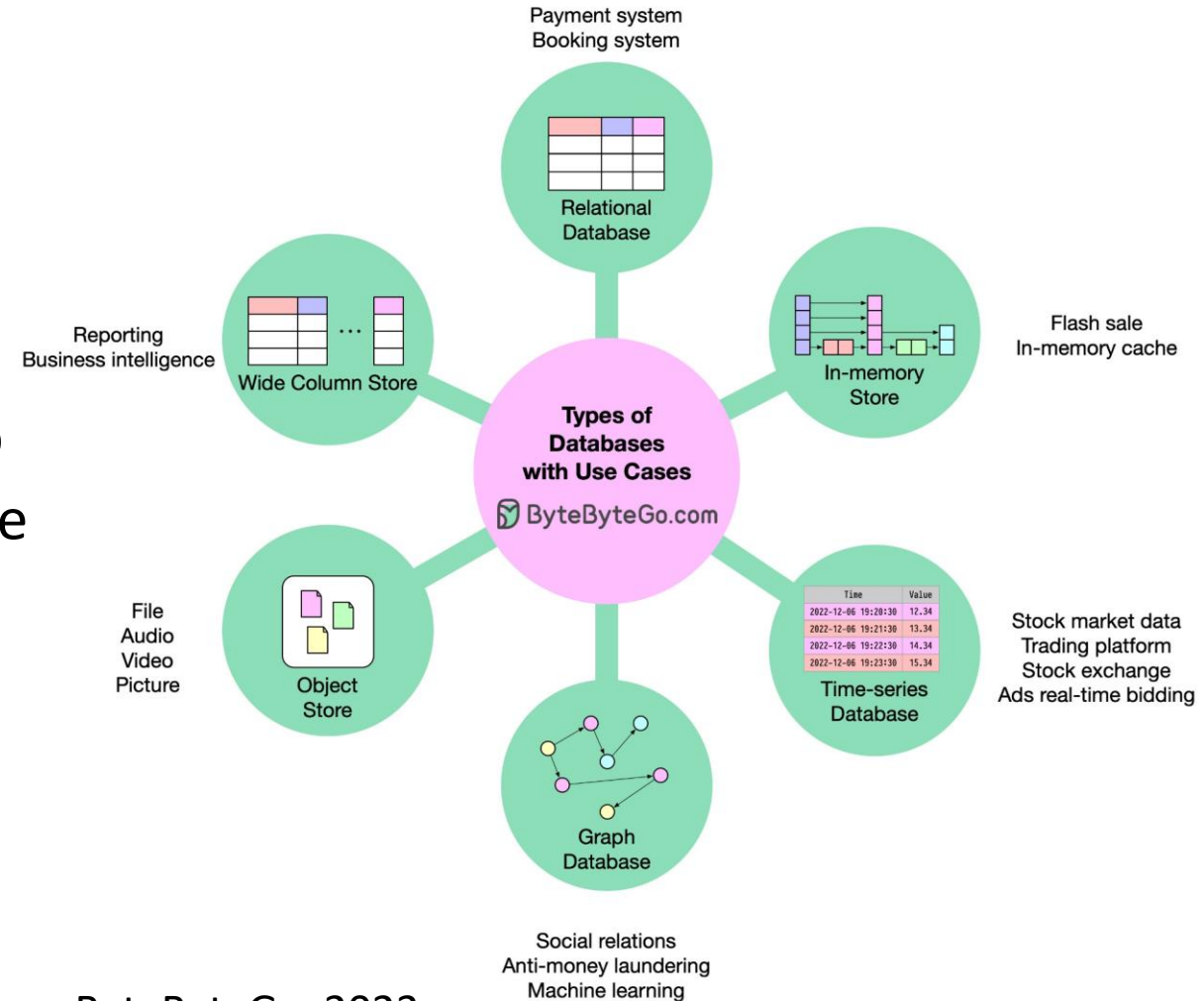
- Sisteme de plată sau rezervări
- Sisteme de raportare / analytics
- Sisteme multimedia
- Rețele sociale
- Platforme de tranzacționare
- Aplicații e-commerce – promoții



Nivelul de date

Cum alegem tipul **bazei de date** în funcție de tipul aplicației

- Relaționale / SQL – aplicații uzuale
- In-memory – operații rapide
- Time-series – operații cu serii de timp
- Grafuri – relații complexe între obiecte nestructurate
- Documente – date imutabile de dimensiuni mari
- Coloane largi – big data, analytics, raportări

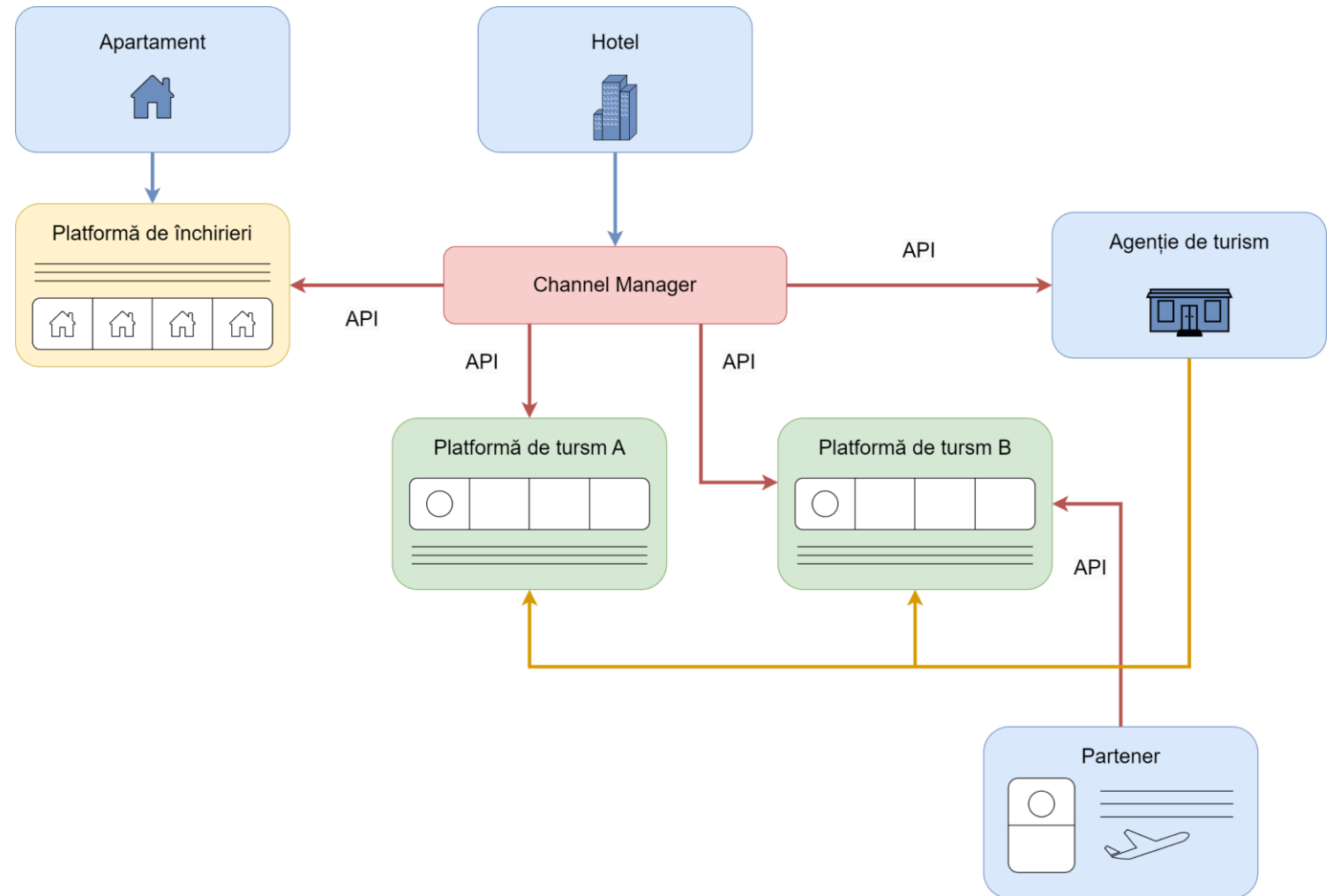


Nivelul funcțional – API

Integrarea mai multor aplicații prin intermediul interfețelor – API

- Sincronizare date
- Interacțiuni funcționale

Studiu de caz: Ecosistemul platformelor de turism online

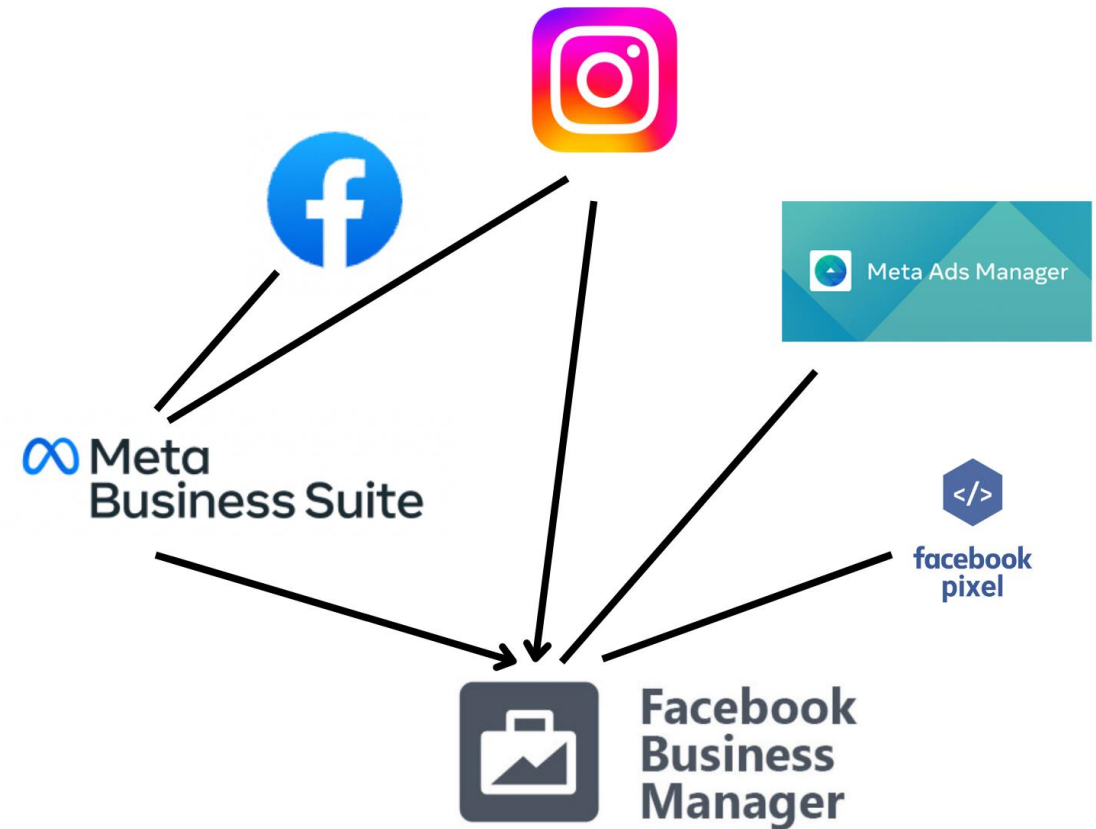


Studiu de caz: Meta Business Suite

Concept: Interfață centralizatoare
“all-in-one” pentru gestionarea
facilă a paginilor business pe
Facebook și Instagram

Context: un ecosistem în continuă
schimbare

Rezultat: o serie de interfețe
imbricate, dificil de navigat



Întrebări?

