

Curs 1: Introducere în baze de date distribuite și mobile

1.1 Definiții și concepte fundamentale

Baze de date distribuite.

O **bază de date distribuită** reprezintă o colecție de date gestionată de un SGBD, ale cărei componente de stocare sunt răspândite pe mai multe calculatoare interconectate[1]. Altfel spus, baza de date nu este stocată într-un singur loc central, ci pe **mai multe noduri** (servele sau computere) aflate fie în aceeași locație fizică, fie în locații geografice diferite, conectate prin rețea[2].



Fig. 1

<https://arzhost.com/blogs/wp-content/uploads/2022/12/distribution-database-848x566.png>

În esență, utilizatorii percep aceste date ca făcând parte dintr-o **singură bază de date logică**, sistemul ascunzând (pe cât posibil) detaliile distribuției. De exemplu, Oracle definește o bază de date distribuită ca fiind alcătuită din „două sau mai multe fișiere aflate în site-uri diferite”, putând rezida pe mai multe calculatoare locale sau în rețele diferite[2].

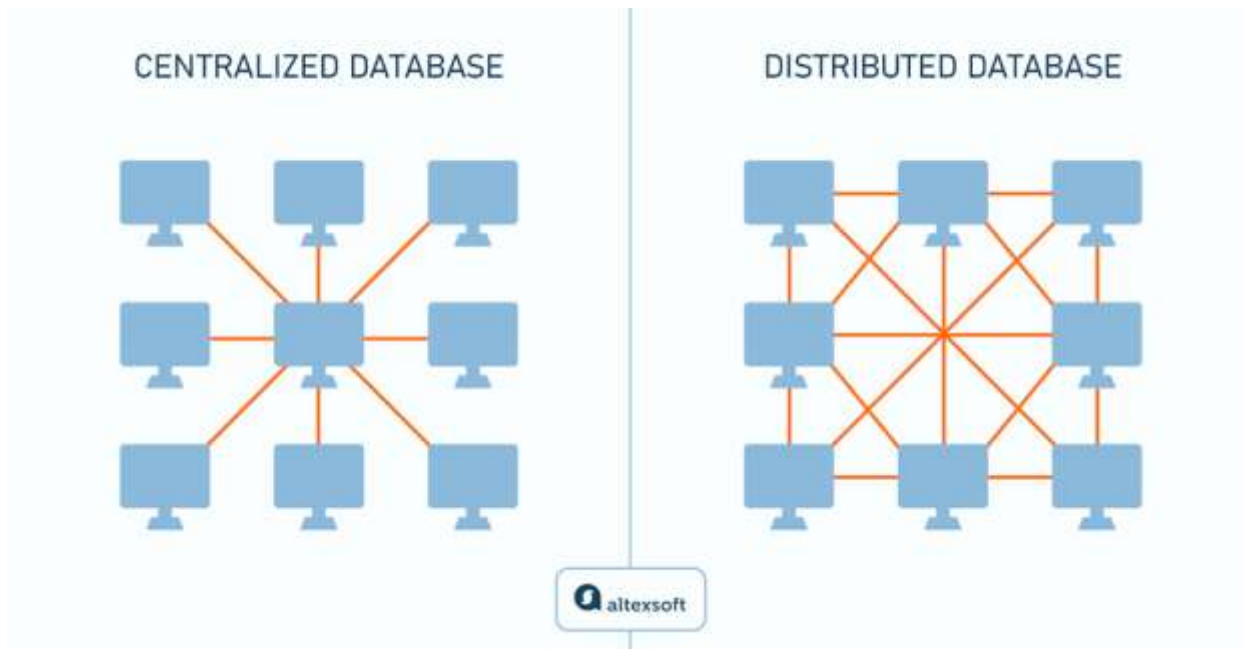


Fig. 2 <https://www.altexsoft.com/static/content-image/2024/8/7c285156-c182-4b66-8a69-ef1e7183b663.png>

Scopul este ca distribuția să fie **transparentă** pentru utilizatori – aceștia interacționează cu sistemul ca și cum ar fi o bază de date unică, **monolitică**, fără să cunoască pe ce nod se află efectiv datele[3][4].

Fragmentare și replicare. Două concepte fundamentale în bazele de date distribuite sunt **fragmentarea** datelor și **replicarea** acestora.

Fragmentarea presupune împărțirea unei relații (tabel) în părți mai mici, stocate pe noduri diferite – de exemplu, **fragmente orizontale** (submulțimi de rânduri) sau **fragmente verticale** (submulțimi de coloane) dintr-un tabel[5]. Fiecare parte a bazei distribuite, atașată unui nod, se numește *partiție* sau *fragment*[6].

Replicarea se referă la menținerea de copii identice ale unor fragmente pe mai multe noduri, cu scopul de a spori fiabilitatea și disponibilitatea sistemului[6]. Astfel, dacă un nod devine indisponibil, o copie a datelor de pe alt nod poate prelua sarcina, permițând sistemului să continue funcționarea fără întreruperi (analog conceptului de redundanță RAID din stocarea pe discuri)[7].

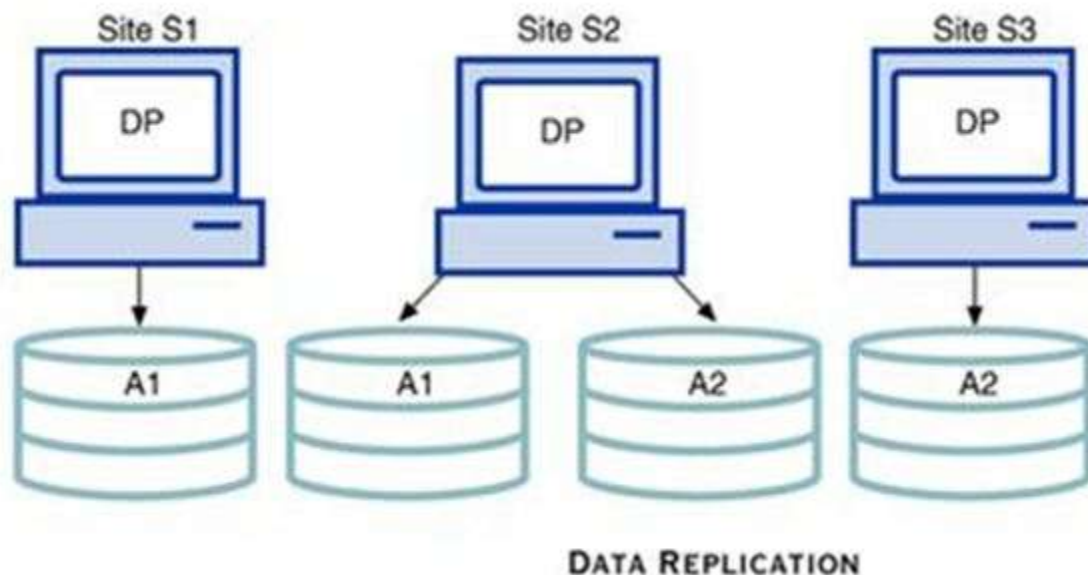


Fig. 3

http://www.myreadingroom.co.in/images/stories/docs/dbms/Distributed_Database_Design_-_data_replication.JPG

În practică, un sistem de baze de date distribuite trebuie să asigure mecanisme pentru menținerea **consistenței datelor** între fragmentele replicate și pentru coordonarea tranzacțiilor distribuite (de exemplu, folosind protocoale de tip two-phase commit pentru a garanta proprietățile ACID în mediu distribuit). Aceste mecanisme implică și ele un grad ridicat de complexitate. De pildă, actualizarea unei informații replicate necesită propagarea modificării pe toate copiile sale, ceea ce implică sincronizare și poate afecta performanța dacă nu este gestionat eficient[8][9].

Caracteristici cheie. Un sistem de gestiune a bazelor de date distribuite (DDBMS) oferă câteva facilități esențiale:

- **Transparența distribuției:** ascunde utilizatorilor detaliile privind locul unde sunt stocate datele, modul de fragmentare sau replicare. Interogările se formulează la fel ca într-o bază de date centralizată, DDBMS-ul ocupându-se de localizarea și combinarea datelor de pe noduri multiple[3][4].
- **Autonomie locală:** fiecare site (nod) poate avea un anumit grad de independență în administrarea propriilor date, ceea ce permite adaptarea la nevoile locale și reduce dependența de un nod central[10].
- **Toleranță la defecțiuni:** prin replicare, sistemul poate continua să funcționeze chiar dacă un nod sau o legătură de rețea cade, îmbunătățind **disponibilitatea** serviciului[11].
- **Scalabilitate și performanță:** distribuind datele și încărcarea pe mai multe mașini, se pot deservi mai mulți utilizatori și opera un volum mai mare de tranzacții în paralel. Datele pot fi plasate în apropierea punctelor de utilizare (de exemplu, datele unei filiale pe un server local acelei filiale), reducând latența și echilibrând încărcarea între noduri[12].

Baze de date mobile.

Noțiunea de **bază de date mobilă** se referă la sistemele de gestiune a datelor destinate dispozitivelor mobile (smartphone, tabletă etc.), care funcționează de obicei ca parte a unei arhitecturi distribuite mai largi. O bază de date mobilă poate fi văzută ca o instanță de bază de date locală pe dispozitivul mobil, capabilă să stocheze și să prelucreze date **offline**, sincronizându-se periodic cu o bază de date centrală sau cu servicii cloud. De exemplu, un telefon mobil poate găzdui o bază de date (precum SQLite) cu informații de contact, mesaje, date de aplicație, care se **sincronizează** cu serverul când conexiunea de rețea este disponibilă[13].

Comparativ cu bazele de date distribuite clasice (între servere fixe), bazele de date mobile aduc provocări suplimentare: conexiuni wireless **intermitente sau lente**, resurse hardware limitate și necesitatea de a funcționa **deconectat** pe perioade mai lungi. Din acest motiv, adesea se folosesc mecanisme de **cache local** și **replicare asincronă**: datele recente și tranzacțiile efectuate pe dispozitivul mobil sunt păstrate local și apoi retransmise serverului la reconectare[14]. Vom detalia aceste aspecte la secțiunea 1.3, dedicată aplicațiilor mobile.

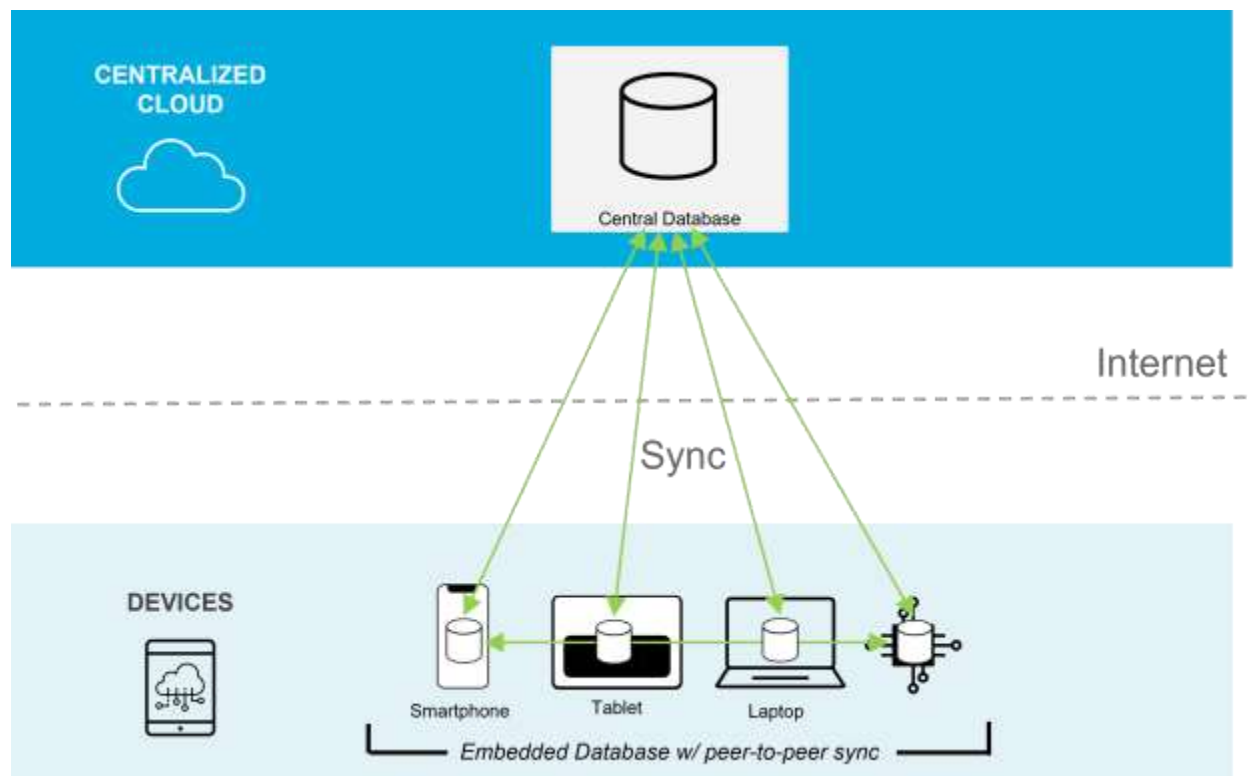


Fig. 4 <https://cdn.thenewstack.io/media/2022/09/b1c8fc8a-figure2-e1664461543825.png>

1.2 Evoluția și importanța bazelor de date distribuite

Evoluția către distribuire. Istoric, primele sisteme de gestiune a bazelor de date (SGBD) din anii 1960-1970 erau centralizate - rulate pe mainframe-uri sau servere unice, deserving organizații întregi. În anii 1980, odată cu popularizarea bazelor de date relaționale, majoritatea datelor erau tot centralizate într-un singur DBMS la nivel de organizație[15]. Totuși, pe măsură ce rețelele de calculatoare au evoluat și organizațiile au devenit tot mai descentralizate

(filiale multiple, echipe distribuite geografic), a apărut necesitatea ca datele să fie distribuite mai aproape de punctele unde sunt folosite. Astfel, în anii 1980-1990 se dezvoltă primele sisteme de baze de date distribuite (atât ca prototipuri academice cât și ca funcționalități integrate în produse comerciale). Un exemplu timpuriu îl reprezintă proiectele de cercetare ca **System R** și **R*** (IBM) care au experimentat execuția de interogări și tranzacții pe date aflate în locații multiple. De asemenea, protocolul **client-server** în arhitectura aplicațiilor a permis ca interfața și logica aplicației să fie separate de serverul de baze de date, deschizând calea pentru *replicarea* bazelor de date pe servere regionale și *federarea* datelor între sisteme eterogene.

Importanța și beneficiile.

Baza de date distribuită a devenit crucială în contextul creșterii volumului de date și al nevoii de **disponibilitate 24/7**.

Iată câteva beneficii majore (avantaje) care subliniază importanța folosirii bazelor de date distribuite în mediul actual:

- **Reflectarea structurii organizației:** Datele pot fi repartizate conform structurii companiei - de exemplu, fiecare filială își poate avea propriul fragment de bază de date, ceea ce corespunde firesc responsabilităților acelei filiale[10]. Acest lucru permite **autonomie locală**: departamentele sau sucursalele controlează mai direct propriile date, putând opera independent până la un punct[10].

- **Disponibilitate sporită și toleranță la erori:** O defecțiune la nivelul unui nod afectează, de obicei, doar fragmentul local de date, nu întregul sistem[11]. Celelalte noduri pot continua să funcționeze, ceea ce înseamnă că sistemul per ansamblu rămâne parțial operațional. În plus, prin replicare, dacă un nod cade, datele sale există redundant pe alt nod, asigurând continuitatea serviciului (failover). Această reziliență este esențială pentru aplicații critice care nu își permit timp de nefuncționare.

- **Performanță și scalabilitate îmbunătățite:** Distribuind datele, interogările pot fi deservite de nodul care are local informația, reducând timpii de acces. De asemenea, încărcarea (cererile utilizatorilor) poate fi împărțită între noduri diferite, permițând operarea în **paralel**. De exemplu, dacă o anumită partiție primește un volum intens de cereri, celelalte partiții nu sunt afectate și își deservesc în continuare utilizatorii[12]. Adăugarea de noduri noi (scalare orizontală) poate crește capacitatea sistemului aproape linear, spre deosebire de sistemele centralizate unde resursele hardware au limite fizice.

- **Economie și cost eficient:** Este adesea mai rentabil să folosim mai multe calculatoare de putere moderată distribuite într-o rețea, decât un singur mainframe foarte puternic și costisitor[16]. Arhitectura distribuită oferă și **modularitate** – componentele sistemului pot fi adăugate sau înlocuite treptat, fără a opri întregul serviciu și fără a perturba alte părți ale sistemului[17].

Provocări și progrese recente.

În ciuda avantajelor, distribuirea bazei de date aduce și provocări: crește **complexitatea** administrării (asigurarea transparenței, întreținerea mai multor noduri, securizarea tuturor locațiilor, menținerea consistenței între copii)[18][19]. De exemplu, o operație simplă ca un *JOIN* între două tabele devine complicată dacă datele respective sunt pe noduri diferite - poate necesita transfer de volume mari de date prin rețea și coordonare între servere[20]. Totodată, costurile inițiale de implementare pot crește datorită infrastructurii de rețea și a licențelor multiple, iar personalul trebuie să aibă expertiză în administrarea sistemelor distribuite (ceea ce nu este încă foarte răspândit)[19].

Cu toate acestea, tendințele tehnologice actuale au dus la soluții ce atenuează multe dintre aceste probleme. De la sfârșitul anilor 2000, odată cu explozia **aplicațiilor web la scară mare** și a volumelor de “big data”, s-a observat o orientare către sisteme distribuite de stocare non-relaționale (**NoSQL**). Acestea au apărut ca răspuns la nevoia de scalare orizontală și de procesare rapidă a datelor semi-structurate sau nestructurate generate de internet[21]. Bazele de date NoSQL sacrificau uneori consistența strictă (conform principiului CAP) în favoarea disponibilității și partiționării tolerate la scară, oferind *eventual consistency* în loc de *strict consistency*. Exemple includ Cassandra, MongoDB, Redis etc., folosite intensiv în aplicații precum rețele de socializare, comerț online sau IoT.

Mai nou, în ultimii ani, asistăm la apariția conceptului de **Distributed SQL (NewSQL)** - o nouă generație de SGBD distribuite care încearcă să combine ce e mai bun din ambele lumi: **consistența tranzacțională și limbajul SQL** al bazelor relaționale, cu **scalabilitatea și toleranța la defecțiuni** ale arhitecturilor NoSQL[22]. Aceste sisteme (precum Google Spanner, CockroachDB, Amazon Aurora, Yugabyte, TiDB ș.a.) oferă un singur spațiu de date logic, replicat sincron sau asincron pe multiple noduri la nivel global, menținând tranzacții ACID distribuite și SQL standard[22][23]. Un exemplu notabil este **Google Spanner**, prezentat inițial într-o lucrare din 2012 ca primul SGBD relațional distribuit global, cu sincronizare bazată pe ceas atomic, care oferă tranzacții consistente extern la scară planetară[24]. Aceste progrese demonstrează importanța continuă a bazelor de date distribuite în era cloud: organizațiile moderne au nevoie ca datele lor să fie disponibile în multiple centre de date din lume, aproape de utilizatori, și să poată crește elastic odată cu cererea, fără a renunța la garanțiile tranzacționale pentru date critice.

În concluzie, **baza de date distribuită** este astăzi coloana vertebrală a multor sisteme informatice la scară mare, de la aplicații enterprise la servicii web globale. Importanța sa rezidă în **capacitatea de a oferi atât performanță și scalabilitate, cât și continuitate și fiabilitate**, adaptându-se nevoilor distribuite ale organizațiilor moderne.

1.3 Aplicații mobile și necesitatea bazelor de date mobile

Contextul aplicațiilor mobile.

Odată cu proliferarea smartphone-urilor și tabletelor, aplicațiile mobile au devenit omniprezente, permițând utilizatorilor acces la informație și servicii oriunde s-ar afla. Aceste aplicații manipulează adesea volume semnificative de date – de la datele utilizatorului (contacte, mesaje, poze) până la conținut de aplicație (hărți offline, fluxuri de știri, produse dintr-un catalog etc.).

Baza de date mobilă este tehnologia care asigură stocarea și gestionarea acestor date pe dispozitivul mobil în mod eficient, sigur și sincronizat cu infrastructura backend.

Necesitatea bazelor de date mobile derivă din **caracteristicile mediului mobil**, care diferă de cele ale mediului desktop sau server tradițional:

- 1) **Utilizare offline și conexiune intermitentă:** Spre deosebire de aplicațiile web care presupun conexiune permanentă la un server central, utilizatorii mobili se așteaptă ca aplicațiile lor să funcționeze chiar și fără internet (de exemplu în avion, într-o zonă fără semnal sau dacă doresc să economisească trafic de date). Prin urmare, **aplicațiile mobile trebuie să poată lucra fără conexiune de rețea**, cel puțin pentru anumite operații[14]. Pentru a realiza acest lucru, datele necesare sunt **descărcate și stocate local** într-o bază de date pe dispozitiv (cache local) atunci când există conexiune, iar eventualele modificări efectuate offline sunt salvate și apoi **sincronizate** cu serverul central la

reconectare[14]. De exemplu, o aplicație de note sau de email descarcă mesajele noi pe telefon și permite citirea sau compunerea unor mesaje și offline; odată revenit online, modificările sunt trimise serverului și se preiau eventualele date noi. Baza de date mobilă facilitează acest mod de lucru, asigurând stocarea tranzacțiilor locale temporare și integritatea datelor până la sincronizare.

- 2) **Limitări de bandă și costul rețelei:** Conexiunile mobile (3G/4G/5G) pot avea **lățime de bandă limitată** și, în unele cazuri, costuri asociate traficului de date. Este inefficient și costisitor să transferi frecvent seturi mari de date între server și client. De aceea, bazele de date mobile ajută la **conservarea lățimii de bandă** – odată ce datele au fost sincronizate și stocate local, utilizarea aplicației nu mai necesită trafic permanent[25]. Doar diferențele (modificările) se transmit la sincronizare, eventual în mod comprimat sau lotizat. Acest model *offline-first* îmbunătățește și experiența utilizatorului, eliminând întârzierile cauzate de rețea pentru operațiile locale (de ex., deschiderea unei liste de contacte se face instant din baza locală, nu printr-un apel la server).
- 3) **Resurse hardware limitate (CPU, memorie, baterie):** Dispozitivele mobile dispun de resurse mult mai reduse față de servere: procesoare mai lente, memorie limitată și, mai ales, alimentare pe baterie. Astfel, **eficiența** devine crucială. Bazele de date mobile sunt adesea **optimizate pentru un footprint redus** și pentru a efectua operații cu un minim de consum energetic. De exemplu, SQLite – un SGBD relațional embedded foarte folosit pe Android și iOS – are o amprentă mică și operează direct într-un fișier pe dispozitiv, cu overhead minim. De asemenea, operațiile costisitoare (compresii, indecșii complecși, tranzacții mari) pot consuma baterie, deci designul aplicației mobile trebuie să țină cont de asta. Un SGBD mobil bun poate gestiona erori și optimiza accesul la date, astfel încât să **prevină descărcarea bateriei** și să ruleze fluent pe CPU-ul dispozitivului[26].
- 4) **Necesitatea sincronizării și rezolvării conflictelor:** În scenariul modern, același utilizator poate folosi **mai multe dispozitive** (telefon, tabletă, laptop) pentru a accesa aceleași date. De exemplu, un manager poate modifica o intrare de calendar de pe telefon și adăuga altă intrare de pe tabletă în lipsa conexiunii, așteptând ca ambele modificări să se **sincronizeze** cu serverul central Exchange/Google Calendar. Acest lucru impune ca sistemul de baze de date mobile să suporte un model robust de **sincronizare bidirecțională**, incluzând detectarea conflictelor (când aceeași înregistrare e modificată concurent pe două dispozitive) și **fuzionarea datelor**. Soluțiile tipice includ marcarea versiunilor de date, jurnalizarea schimbărilor și aplicarea unor reguli de *merge* la server (de exemplu, ultima modificare câștigă, sau unificarea listelor etc.)[27]. Multe platforme oferă deja astfel de facilități: Couchbase Lite, Realm, Firebase Realtime Database/Firestore ș.a. permit sincronizare aproape în timp real, cu **replicare automată** în momentul în care dispozitivul recâștigă conectivitatea. Un **sistem de baze de date mobile modern** ar trebui să ofere unelte pentru replicare automată și concilierea variantelor de date modificate offline de utilizatori multipli[28]. Astfel, se evită pierderea modificărilor și se asigură consistența datelor între dispozitiv și server.

Pe lângă cele de mai sus, mai există aspecte precum **securitatea datelor mobile** (dispozitivele pot fi pierdute sau furate, deci baza de date locală trebuie criptată și protejată cu autentificare) și **gestionarea schemelor de date** pe mobil (uneori aplicațiile mobile stochează doar sub-seturi relevante de date, nu întreaga bază de date centrală, pentru a economisi spațiu).

Exemple și aplicații. Aproape orice aplicație mobilă non-trivială folosește o formă de bază de date mobilă:

- Aplicațiile de mesagerie (WhatsApp, Signal) stochează local istoricul mesajelor și fișiere media într-o bază de date (de exemplu WhatsApp folosește SQLite). Astfel, poți căuta sau vedea mesaje vechi și offline ; la reconectare, noile mesaje sosite sunt sincronizate de pe server.



Fig. 5 <https://static1.makeuseofimages.com/wordpress/wp-content/uploads/2024/01/whatsapp-and-signal-download-page-side-by-side.jpg>

- Aplicații de navigație și hărți (Google Maps, Here WeGo) permit descărcarea **offline a hărților**. Aceste date cartografice sunt stocate într-o bază de date locală, permițând calcularea rutei și vizualizarea hărții fără conexiune. La recăpătarea conexiunii, se pot actualiza segmentele de hartă sau rapoartele de trafic.



Fig. 6 https://s1.cdn.autoevolution.com/images/news/gallery/here-wego-route-options-showing-google-maps-how-its-done_4.jpg

- Clienții de email și calendar își stochează mesajele și evenimentele recente local, pentru a permite citirea și compunerea lor offline. Odată online, schimbările (email trimis, eveniment adăugat) se sincronizează cu serverul (Gmail, Exchange etc.).

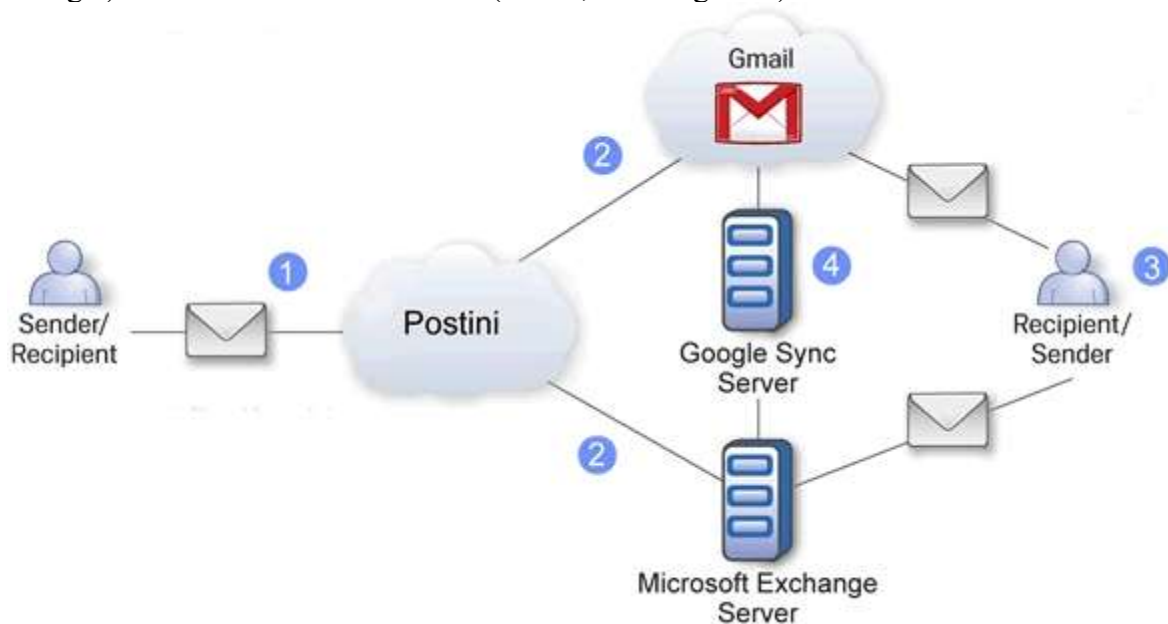


Fig. 7 https://image.itmedia.co.jp/enterprise/articles/1012/10/you_gmail.jpg

- Aplicații enterprise de tip CRM/SFA (Customer Relationship Management / Sales Force Automation) pe tablete ale agenților de vânzări: aceștia își descarcă catalogul de produse,

informațiile despre clienți și comenzi într-o bază de date pe tabletă la începutul zilei și pot înregistra comenzi sau interacțiuni offline în timpul vizitelor; la finalul zilei, tableta se conectează și sincronizează noile comenzi cu serverul central al companiei.



Fig. 8 <https://www.greensoft.ro/wp-content/uploads/2019/02/infografic-sfa-en.png>

Toate aceste exemple ilustrează **necesitatea bazelor de date mobile**: fără o soluție de stocare locală și sincronizare, aplicațiile ar fi dependente de rețea și ar oferi o experiență slabă utilizatorilor (lag, indisponibilitate). Prin folosirea bazelor de date mobile, aplicațiile devin **mai robuste, mai rapide și mai fiabile** în mediul mobil, permițând utilizatorilor acces neîntrerupt la datele lor. În concluzie, baza de date mobilă extinde paradigma distribuției datelor până la marginea rețelei (edge), adică până la dispozitivul utilizatorului, asigurând că aplicațiile mobile pot funcționa eficient **oricând și oriunde**, chiar și în absența unei conexiuni constante.

Bibliografie :

- [1] [3] [5] [6] [7] [10] [11] [12] [16] [17] [18] [19] [20] Bază de date distribuită - Wikipedia
https://ro.wikipedia.org/wiki/Baz%C4%83_de_date_distribuit%C4%83
- [2] [15] [21] Ce este o bază de date? | Oracle România
<https://www.oracle.com/ro/database/what-is-database/>
- [4] Transparencies in DDBMS - GeeksforGeeks
<https://www.geeksforgeeks.org/dbms/transparencies-in-ddbms/>
- [8] [9] Distributed database - Wikipedia
https://en.wikipedia.org/wiki/Distributed_database
- [13] [14] [25] [26] [27] [28] Mobile database - Wikipedia
https://en.wikipedia.org/wiki/Mobile_database
- [22] [23] [24] What is distributed SQL? The evolution of the database
<https://www.cockroachlabs.com/blog/what-is-distributed-sql/>