# Aspecte obligatorii ale securităţii în bazele de date

Anterior proiectării unui sistem sigur, prima alegere care trebuie făcută se referă la politica de securitate care va fi impusă de către sistem. Această politică este constituită, în general, de un set de reguli care vor asigura securitatea sistemului.

Politicile de securitate pot fi clasificate în următoarele două tipuri:

- politici obligatorii (*mandatory*)
- politici opţionale (*discretionary*).

Politicile *mandatory* sunt cele obligatorii prin natura lor, independente de aplicaţie. Politicile opţionale sunt cele specificate de către administrator sau de către un utilizator responsabil de mediul în care va opera sistemul. Cea mai cunoscută politică de securitate opţională este cea referitoare la controlul accesului.

În cadrul acestui curs, vom prezenta aspectele obligatorii ale securităţii în sistemele de gestiune a bazelor de date.

Cursul cuprinde:

- o pespectivă istorică a acestor aspecte, cu precădere asupra unui tip specific de securitate *mandatory*, securitatea multi-nivel (MLS/DBMS *Multi Level Secure Database Management Systems*)
- proiectarea acestui tip de sisteme, din care rezultă arhitecturi de securitate pentru SGBD-uri.

Scopul cursului este de a prezenta fundamentele diferitelor tipuri de MLS/DBMS-uri, inclusiv a sistemelor de baze de date relaţionale, distribuite şi orientate pe obiecte.

## 1. Dezvoltări anterioare

Multe contribuţii ale anilor ‘80 şi ‘90 în domeniul securităţii bazelor de date au fost orientate către MLS/DBMS. Aceste sisteme se mai numesc *Trusted Database Management Systems* (TDBMS). Într-un MLS/DBMS, utilizatorii sunt clasificati pe diferite niveluri de securitate, cum ar fi *Unclassified*, *Confidential*, *Secret* şi *TopSecret*. La rândul lor, datelor le sunt

atribuite niveluri de “sensibilitate”, denumite similar clasificării utilizatorilor. În general, se presupune că aceste niveluri de securitate formează o latice ordonată parțial.

De exemplu, *Unclassified* < *Confidential* < *Secret* < *TopSecret*. Ordonarea parţială provine din faptul că avem diferite compartimente, iar acestea pot să nu fie comparabile.

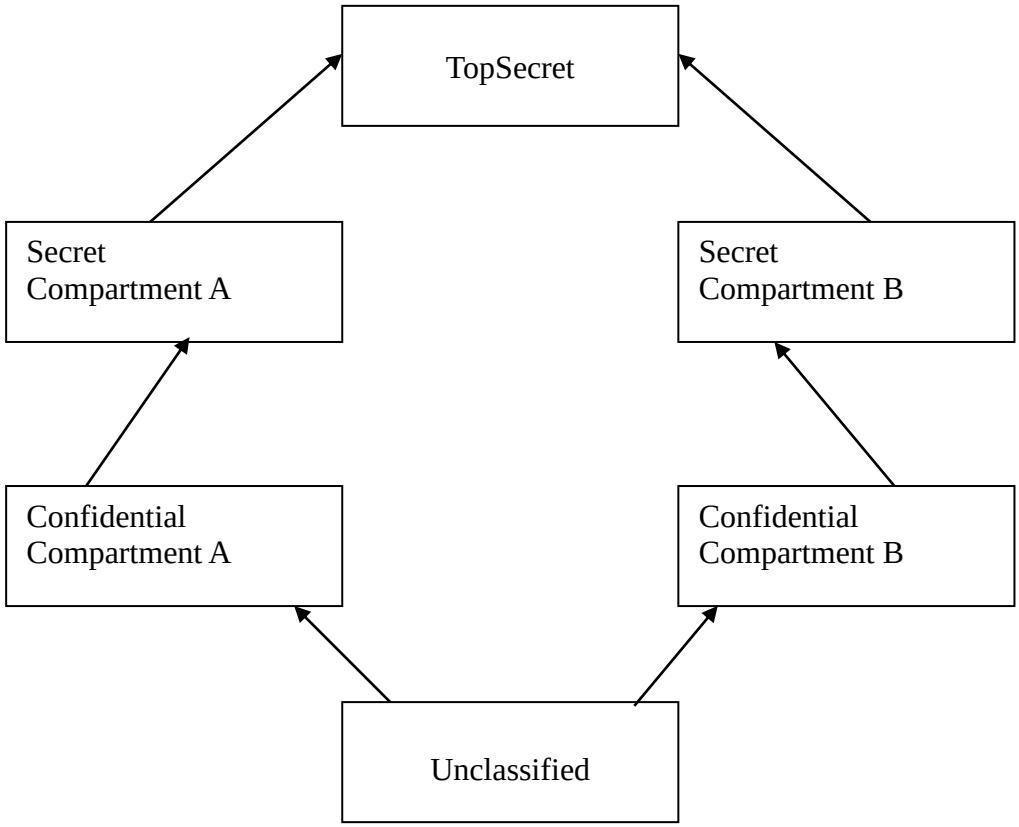


**Figura 1.** Niveluri de securitate

MLS/DBMS-urile au evoluat pornind de la dezvoltările din cadrul unor sisteme de operare securizate multi-nivel (de exemplu, MULTICS, SCOMP) şi de la dezvoltările din domeniul bazelor de date. Multe dintre eforturi s-au îndreptat către modelul de date relaţional.

La sfârşitul anilor ‘80, *National Computer Security Center* a demarat eforturile pentru interpretarea criteriilor de evaluare pentru sistemele de calcul sigure, în cazul sistemelor de baze de date. Această interpretare a fost denumită *Trusted Database Interpretation*.

În anii ‘90 cercetările s-au concentrat asupra sistemelor nerelaţionale, inclusiv asupra sistemelor de baze de date orientate obiect MLS. De asemenea, s-a lucrat asupra sistemelor de baze de date distribuite sigure, pe mai multe niveluri. Au apărut şi au fost investigate provocări precum modelele de date multi-nivel, problema inferenţei şi procesarea sigură a tranzacţiilor. De asemenea, au început să apară produse comerciale. Ulterior, eforturile încă s-au îndreptat către examinarea securităţii multi-nivel pentru noile tehnologii de gestiune a datelor.

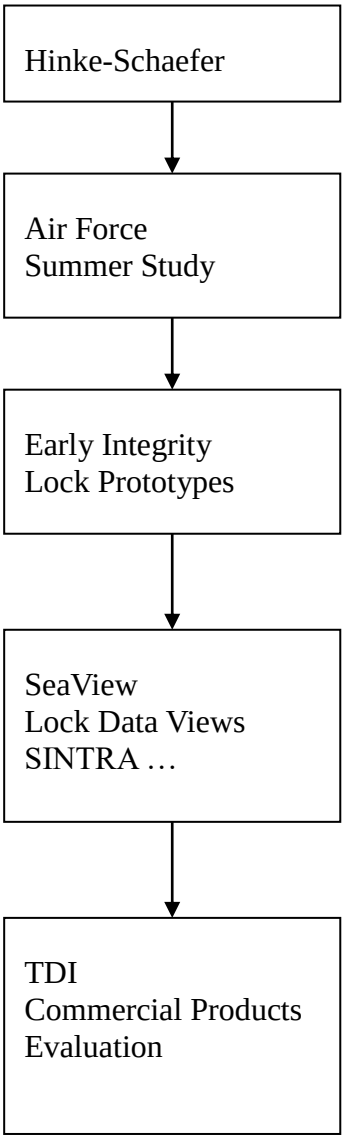


**Figura 2.** Evoluţia MLS/DBMS

## 1.1 Primele iniţiative

Un moment important în dezvoltarea MLS/DBMS-urilor l-a constituit *Air Force Summer Study*. Acest moment a fost la rândul său influenţat de către dezvoltările anterioare, cea mai notabilă dintre acestea fiind abordarea Hinke-Schaefer asupra furnizării securităţii obligatorii (*mandatory*) de către sistemele de operare.

Această abordare a dezvoltat un mod de a găzdui MLS/DBMS-urile pe sistemul de operare MULTICS MLS. Sistemul se baza pe cel relaţional şi ideea consta în partiţionarea relaţiei pe baza atributelor şi stocarea acestora în diferite fişiere, la diferite niveluri, urmând ca ulterior sistemul de operare să controleze accesul la fişiere.

***Air Force Summer Study***

Scopul acestui studiu a fost cel de a analiza problemele din domeniu şi de a propune abordări viabile pentru proiectarea MLS/DBMS.

Grupul de studiu a reunit experţi din domeniu şi a fost împărţit în trei subgrupuri. Primul subgrup s-a concentrat asupra abordărilor pe termen scurt pentru proiectarea MLS/DBMS. Aceste abordări au inclus pe cea referitoare la blocarea integrităţii şi la arhitectura distribuită.

Al doilea subgrup a avut în vedere modelul sistemului de mesaje militare pentru un MLS/DBMS. Acest model a fost dezvoltat la vremea respectivă, iar scopul grupului a fost cel de a examina aplicabilitatea lui pentru MLS/DBMS.

Al treilea subgrup a studiat abordările pe termen lung şi a examinat probleme precum clasificarea datelor bazată pe conţinut şi pe context, modelele de date relaţionale multi-nivel şi problema inferenţei.

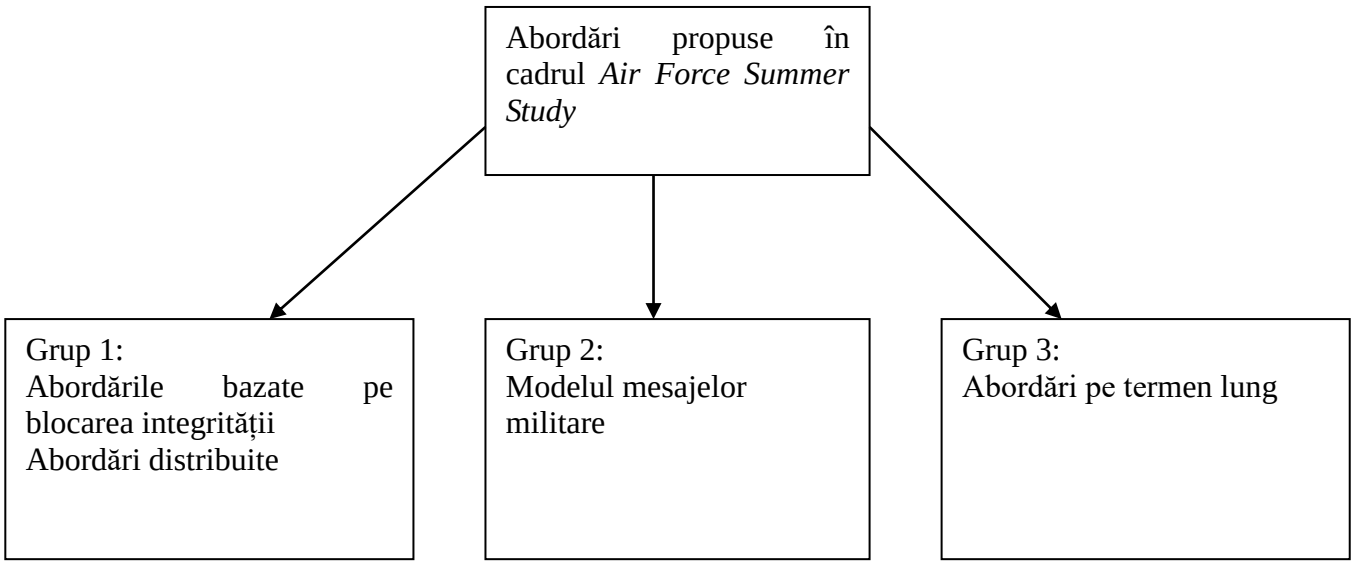


**Figura 3.** Abordări propuse de către *Air Force Summer Study*

## 1.2 Cercetări şi dezvoltări majore

Ulterior *Air Force Summer Study* au fost raportate numeroase contribuţii asupra proiectării şi dezvoltării bazelor de date sigure.

Dezvoltările iniţiale s-au bazat pe abordarea referitoare la blocarea integrităţii, dezvoltată la *MITRE Corporation*. Au fost proiectate şi dezvoltate două prototipuri, dintre care unul pentru sistemul *Ingres*.

Sistemele *SeaView System* şi *Lock Data Views System* pot fi, de asemenea, amintite.

Aceste două inițiative au fost luate de către *Rome Air Development Center* iar scopul a fost îndreptat către abordările pe termen lung propuse de către *Summer Study*. Ambele inițiative au influențat foarte mult dezvoltările comerciale ulterioare.

Alte trei inițiative importante sunt: sistemul SINTRA dezvoltat de către *Naval Research Laboratory*, sistemul SWORD dezvoltat de către *Defense Research Agency* (UK) și SDDBMS dezvoltat de *Unisys*. Sistemul SINTRA s-a bazat pe arhitectura distribuită propusă de către *Air Force Summer Study*, SWORD a propus alternative la modelele de date *SeaView* și *Lock Data Views*, iar SDDBMS a investigat abordările cu partiționare și replicare pentru proiectarea MLS/DBMS.

## 1.3 Interpretarea criteriilor de evaluare pentru bazele de date sigure

În urma progreselor apărute în anii ‘80, *National Computer Security Center* (NCSC) a anticipat că urmau să apară produsele comerciale (SGBD) securizate. La vremea respectivă, existau deja sisteme de operare comerciale evaluate prin criteriile specifice elaborate pentru sistemele de calcul sigure. Aceste criterii constau din reguli pe care trebuie să le satisfacă un sistem pentru a fi certificat la un anumit nivel. Cel mai înalt nivel era A, iar cel mai scăzut era C1. Între acestea existau mai multe niveluri (C2, B1, B2, B3).

Pentru a evalua MLS/DBMS, aceste criterii nu mai erau suficiente, astfel încât a fost nevoie de interpretarea lor pentru sistemele de baze de date. Interpretarea a primit numele *Trusted Database Implementation* (TDI). TDI s-a concentrat asupra unei abordări denumite TCB (*Trusted Computing Base*). Aceasta este acea parte a sistemului care impune politicile de securitate.

TDI a fost publicat în 1991, moment la care existau deja suficiente produse comerciale (*Oracle, Sybase, Informix, Digital Equipment Corporation, Ingres*) pregătite să fie evaluate.

## 1.4 Tipuri de sisteme de baze de date sigure multi-nivel

Multe dintre MLS/DBMS-urile proiectate și dezvoltate se bazează pe modelul relațional. Au existat, însă, și eforturi îndreptate asupra altor modele, precum cele obiect și funcționale.

Tipuri de MLS/DBMS:

- MLS Relational Databases
- MLS Object Databases
- MLS Distributed Databases
- MLS Parallel Databases
- MLS Real-Time Databases

- MLS Functional Databases
- MLS Logic Databases

### 1.4.1 Sistemele de baze de date relaţionale

Primele eforturi în privinţa MLS au fost îndreptate asupra bazelor de date relaţionale. Aceste sisteme au reflectat atât proiectarea modelelor de baze de date relaţionale multi-nivel, cât şi furnizarea accesului pe bază de vizualizări. Aceste eforturi au formulat şi noţiunea de poliinstanţiere, în care utilizatorii pot vizualiza diferit un element în funcţie de nivelul lor de securitate.

Alte subiecte de cercetare examinate în cadrul MLS/DBMS-urilor bazate pe modelul relaţional au inclus procesarea sigură a tranzacţiilor, problema inferenţei şi procesarea constrângerilor de securitate.

### 1.4.2 Sisteme entitate-relaţie

În anul 1987, *Air Force Research Laboratory in Ro*me a avut iniţiativa de a proiecta un MLS/DBMS pe baza modelului entitate-relaţie, dezvoltat iniţial în 1976 de către Peter Chen şi utilizat ulterior pentru modelarea aplicaţiilor. Scopul acestei iniţiative a fost atât acela de a explora proprietăţile referitoare la securitate pentru modelul ER, cât şi de a explora utilizarea modelelor ER sigure pentru proiectarea SGBD-urilor.

Rezultatul a constat în modelele MLS ER, care au fost utilizate începând de atunci pentru modelarea aplicaţiilor sigure. Unele variante ale acestui model au fost utilizate pentru a explora problema inferenţei.

Această abordare a contribuit la proiectarea aplicaţiilor MLS, însă nu au existat iniţiative de a proiecta MLS/DBMS-uri pe baza modelului ER.

### 1.4.3 Sisteme de baze de date obiect

Proiectarea MLS/DBMS de acest tip se poate baza pe diferite modele. În general, se bazează pe proiectarea propusă pentru MLS/DBMS bazate pe modelul relaţional. În plus, trebuie securizate obiectele complexe şi gestionată execuţia sigură a metodelor.

Pe măsură ce cercetările au progresat în proiectarea MLS/DBMS bazate pe obiecte, au existat eforturi pentru utilizarea modelelor obiect pentru proiectarea aplicaţiilor sigure. Astazi, în urma dezvoltării UML, există iniţiative de a proiecta aplicaţii sigure pe baza acestui limbaj.

Modelele obiect au fost utilizate şi pentru reprezentarea bazelor de date multimedia sigure.

### 1.4.4 Sisteme de baze de date distribuite şi eterogene

Iniţiativa *Air Force Summer Study* a discutat proiectarea MLS DBMS bazate pe arhitecturi distribuite, însă partiţionând datele în funcţie de nivelul lor de securitate. Scopul era acela de a dezvolta sisteme de baze de date centralizate. Ulterior, s-a lucrat asupra proiectării şi dezvoltării MLS/DBMS distribuite (MLS/DDBMS). Mai apoi, atenţia s-a concentrat asupra proiectării şi dezvoltării sistemelor de baze de date distribuite eterogen.

### 1.4.5 Sisteme de baze de date deductive

Bazele de date deductive se bazează pe logică. Atunci când a fost investigată problema inferenţei, au fost proiectate sisteme de baze de date deductive sigure multi-nivel. Aceste sisteme se bazează pe logica NMTL (Nonmonotonic Typed Multilevel Logic). Aceasta furnizează capabilităţi referitoare la raţionament în nivelurile de securitate, care prin natura lor sunt nemonotone. Această logică încorporează construcţii pentru raţionamentul asupra aplicaţiilor la diferite niveluri de securitate.

### 1.4.6 Sisteme de baze de date funcţionale

Modelele de date funcţionale se bazează pe funcţii, iar evaluarea cererilor conduce la execuţia de funcţii. Au fost studiate modele MLS şi pentru acest tip de sisteme, a fost introdusă noţiunea de funcţie multi nivel şi a fost ridicată problema execuţiei acestor funcţii.

### 1.4.7 Sisteme de baze de date în timp real

Aceste sisteme au fost examinate pentru numeroase aplicaţii incluzând pe cele de comandă şi control, precum şi cele de control al proceselor. Provocarea este de a încorpora securitatea în procesarea real-time. Aceste două componente sunt, însă, în conflict. De exemplu, dacă trebuie făcute toate verificările de control al accesului, tranzacţiile pot să nu se mai termine la timp. Există şi potenţial pentru canalele ascunse (*covert*) atunci când are loc integrarea securităţii cu procesarea real time. Au fost investigaţi algoritmi siguri pentru controlul concurenţei în timp real, au fost tratate probleme referitoare la încorporarea securităţii, procesării în timp real şi a toleranţei la defecte. Problemele referitoare la aceste sisteme devin critice pentru multe aplicaţii, inclusiv sistemele îmbarcate.

## 1.5 Probleme importante

Anterior, am făcut o scurtă prezentare a diferitelor tipuri de MLS/DBMS. Câteva probleme importante şi dificile în MLS/DBMS sunt inferenţa, procesarea sigură a tranzacţiilor şi dezvoltarea unui model de date relaţional sigur multi-nivel.

Cea mai notabilă dintre aceste probleme este cea a inferenţei. Aceasta constă în procesul de a formula cereri şi a deduce informaţii sensibile din răspunsurile legitime primite. În literatura de specialitate au fost prezentate diferite abordări în scopul rezolvării acestei probleme. A fost demonstrat că problema generală a inferenţei este nerezolvabilă. A fost explorată utilizarea constrângerilor de securitate şi a structurilor conceptuale pentru a trata diferite tipuri de inferenţă. Un caz particular de inferenţă este problema agregării.

În ceea ce priveşte procesarea sigură a tranzacţiilor, multe eforturi au fost îndreptate către reducerea canalelor ascunse (*covert*) la procesarea tranzacţiilor în MLS/DBMS. Aşa cum am precizat anterior, provocarea constă în a proiecta MLS/DBMS-uri care funcţionează în timp real. Aceasta presupune nu numai faptul că tranzacţiile trebuie să fie sigure, ci să respecte şi constrângeri referitoare la timp.

Referitor la dezvoltarea unui model de date relaţional sigur multi-nivel au fost dezvoltate mai multe propuneri. Problema este cauzată de faptul că diferiţi utilizatori au vizualizări diferite asura aceluiaşi element Dacă utilizăm valori multiple pentru a reprezenta aceeaşi entitate, atunci nu respectăm integritatea bazelor de date. Dacă nu impunem poliinstanţierea, rezultă un potenţial de signaling channels. Problema este încă deschisă.

## 1.6 Tehnologii noi

Pe măsură ce apar noi tehnologii, pot fi examinate problemele specifice referitoare la securitatea multi-nivel. La ora actuală există multe tehnologii noi precum data warehousing, sistemele de comerţ electronic, sistemele multimedia, bibliotecile digitale şi web. Cu toate acestea, a apărut un număr limitat de iniţiative referitoare la investigarea securităţii multi-nivel pentru noile sisteme de gestiune a datelor. Acest lucru se datorează în parte faptului că şi în cazul sistemelor relaţionale există probleme dificile de rezolvat referitoare la securitatea multi-nivel. Pe măsură ce sistemele devin mai complexe, dezvoltarea unor sisteme multi-nivel sigure devine o provocare tot mai mare. De exemplu, cum s-ar putea dezvolta sisteme sigure multi-nivel pentru librăriile digitale şi sistemele de comerţ electronic? Cum s-ar putea obţine o performanţă acceptabilă? Cum pot fi verificate sisteme uriaşe (de exemplu, World Wide Web)? În prezent,

încă există foarte multe lucruri de făcut în privinţa securităţii opţionale pentru aceste tipuri de sisteme.

# 2. Principii ale proiectării

În continuare, vom descrie principiile de proiectare pentru MLS/DBMS, prezentând o taxonomie pentru realizarea diferitelor arhitecturi MLS/DBMS.

Tipul de control al accesului utilizat de catre MLS/DBMS este cel mandatory. Vom prezenta acest tip de control, vom introduce politicile de securitate Bell-La Padula si interpretarea lor pentru MLS/DBMS. Ulterior, vom prezenta diferite arhitecturi de securitate MLS/DBMS.

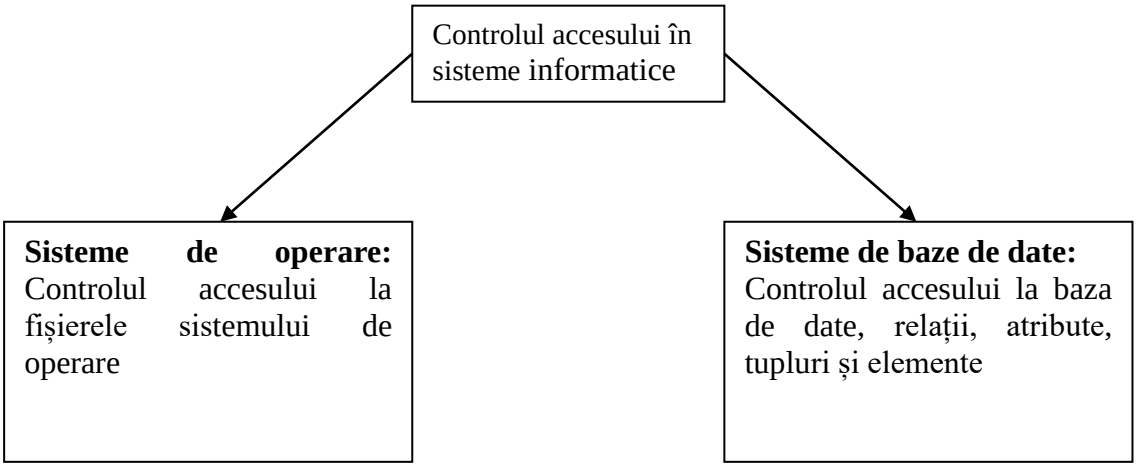


**Figura 4.** Controlul accesului în sisteme de operare şi sisteme de baze de date

## 2.1 Controlul de tip *mandatory* al accesului

Deși SGBD-urile trebuie să trateze majoritatea problemelor de securitate pe care le abordează sistemele de operare (identificare si autentificare, control al accesului, audit), există caracteristici specifice care generează probleme suplimentare.

De exemplu, obiectele din SGBD tind să aibă dimensiune variabilă și pot avea granularitate fină (relații, atribute și elemente). Acest fapt contrastează cu sistemele de operare, unde granularitatea tinde să fie mai mare (fișiere, segmente). Din cauza acestei granularități fine a MLS/DBMS (denumite adeseori TDBMS), obiectele asupra cărora este efectuat accesul MAC (*Mandatory Access Control*) și DAC (*Discretionary Access Control*) pot să difere. În sistemele de operare MLS (*Trusted Operating Systems*), MAC și DAC sunt realizate de obicei asupra aceluiași obiect (de exemplu, un fișier).

De asemenea, există anumite diferențe funcționale între sistemele de operare și SGDB-uri.

- Sistemele de operare lucrează cu subiecți care doresc să modifice obiecte. SGBD-urile sunt folosite pentru a partaja date între utilizatori și a furniza acestora mijloace prin care să lege diferite obiecte.
- SGBD-urile sunt dependente de sistemele de operare, care le furnizează resurse (de exemplu, comunicare între procese și gestiune a memoriei). Prin urmare, proiectarea SGBD-urilor sigure trebuie sa ia în considerare modul în care securitatea este tratată de către sistemele de operare.

Diferențele discutate anterior conduc la concluzia că abordările tradiționale utilizate în proiectarea sistemelor trebuie să fie adaptate pentru SGBD-uri. La ora actuală, nu există o abordare arhitecturală standard pentru MLS/DBMS. Au fost propuse diferite modalități pentru proiectarea și construirea MLS/DBMS. O parte a acestor abordări vor fi prezentate in continuare. În esență, MLS/DBMS-urile au fost proiectate pe baza uneia dintre arhitecturile care vor fi prezentate.

Figura 4 ilustrează diferența dintre controlul accesului in sistemele de operare și controlul accesului în SGBD-uri.

## 2.2 Politici *mandatory* de control al accesului

Politicile MAC specifică accesul pe care subiecții îl au asupra obiectelor. Multe dintre SGBD-urile comerciale se bazează pe politicile lui Bell și La Padula specificate pentru proiectarea sistemelor de operare. Prin urmare, vom prezenta aceste politici, iar apoi modul în care ele au fost adaptate pentru SGBD-uri. Alte politici *mandatory* le includ pe cele referitoare la noninferenta (Goguen, Messeguer).

În politicile Bell – La Padula, subiecţii sunt asociați unor niveluri și pot opera inclusiv până la nivelul pe care se află. Obiectelor le sunt atribuite niveluri de sensibilitate. Cele două reguli ale politicii B-LP sunt:

1) Proprietatea securității simple: Un subiect are acces pentru citire asupra unui obiect dacă nivelul său de securitate domină nivelul corespunzător obiectului.

2) Proprietatea * : Un subiect are acces pentru scriere asupra unui obiect dacă nivelul de securitate al subiectului este dominat de cel al obiectului.

Aceste proprietăți se aplică și pentru SGBD-uri. În cazul acestora, proprietatea * este modificată astfel:

Un subiect are acces pentru scriere asupra unui obiect dacă nivelul subiectului este acelaşi cu cel al obiectului. Prin urmare, un subiect poate modifica relaţii care au nivelul său de securitate.

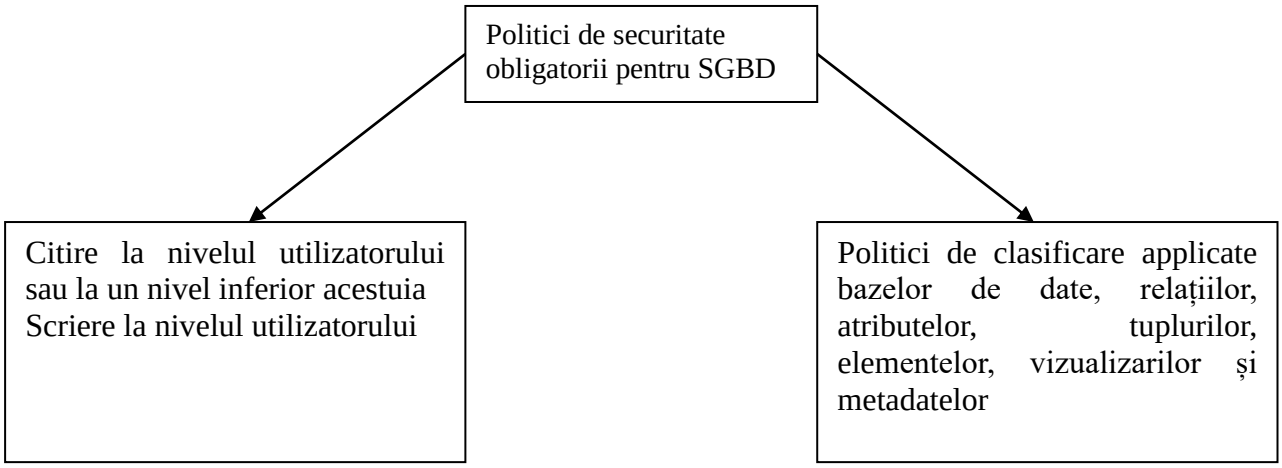


**Figura 5.** Politici obligatorii pentru SGBD-uri

Un aspect important care face parte din politicile de securitate pentru sistemele de baze de date este poliinstanţierea. Aceasta presupune ca acelaşi obiect poate avea diferite interpretări și valori la diferite niveluri. De exemplu, pe nivelul *Unclassified* salariul unui angajat poate fi 30000, iar la nivelul *Secret* acesta poate fi 70000. În cadrul modelelor relaționale multi-nivel putem avea ambele intrări, însă însoțite de nivelurile de securitate corespunzătoare, ca atribut suplimentar.

Una dintre motivațiile pentru utilizarea poliinstanțierii este evitarea “canalelor ascunse”. De exemplu, dacă avem o intrare la nivelul *Secret* care furnizează valoarea 70000 pentru salariul lui X, iar un subiect *Unclassified* doreşte să introducă valoarea 30000 pentru același salariu și actualizarea nu este permisă, acest lucru ar putea semnala un canal de la un nivel mai înalt către unul aflat la nivel mai jos. În timp, acesta ar putea deveni un canal ascuns. Asupra poliinstanțierii au avut loc multe discuții și dezbateri, însă nu a fost atins un consens.

Diferitele sisteme au implementat modelele de date relationale multi-nivel in moduri diferite.

## 2.3 Arhitecturi de securitate

Arhitecturile de securitate sunt arhitecturi de sistem, care au fost proiectate urmărind anumite aspecte ale securității. În continuare vom prezenta 5 arhitecturi pentru MLS/DBMS. Așa cum am menționat anterior, acestea furnizează o taxonomie pentru MLS/DBMS-uri.

Aceste arhitecturi sunt caracterizate de: blocarea integrității (*integrity lock*), activarea

securității obligatorii de către sistemul de operare, extensiile nucleului, subiectul sigur, arhitectura distribuită. Mai departe, arhitectura distribuită se împarte pe baza abordării partiționate, respectiv a celei replicate.

### 2.3.1 Arhitectura Integrity lock

Aceasta arhitectură utilizează un SGBD nesigur (*back end*), cu acces la datele din baza de date, o interfata (*front-end*) nesigură care comunică cu utilizatorul și o aplicație *front-end* sigură care utilizează criptarea (figura 6). Componentele nesigure sunt izolate unul de celalalt astfel încât nu există comunicare între cele două fără medierea filtrului sigur. SGBD-ul este menținut la nivelul sistemului, acesta fiind considerat cel mai înalt nivel suportat de către sistem. Sunt menținute instanțieri multiple ale *front-end*-ului, câte o instanță pentru fiecare nivel de utilizator. Filtrul sigur este și el mentinut la nivelul sistemului.

În această abordare fiecare tuplu care este inserat în baza de date are asociată o etichetă de securitate și o sumă (*checksum*) criptografică. Eticheta de securitate este criptată, iar datele sunt necriptate. Sumele sunt calculate de către filtrul sigur la inserare și recalculate la regăsire. La inserare, filtrul sigur calculează suma și SGBD-ul nesigur ia datele (de exemplu, tuplurile), eticheta asociată și suma și le stochează în baza de date. La operația de regăsire, *back end*-ul regăsește tuplurile de date și le trimite filtrului sigur care recalculează sumele pe baza tuplului și a etichetei regăsite. Dacă filtrul determină că datele nu au fost încă afectate, le transmite utilizatorului via *front-end*-ul nesigur.

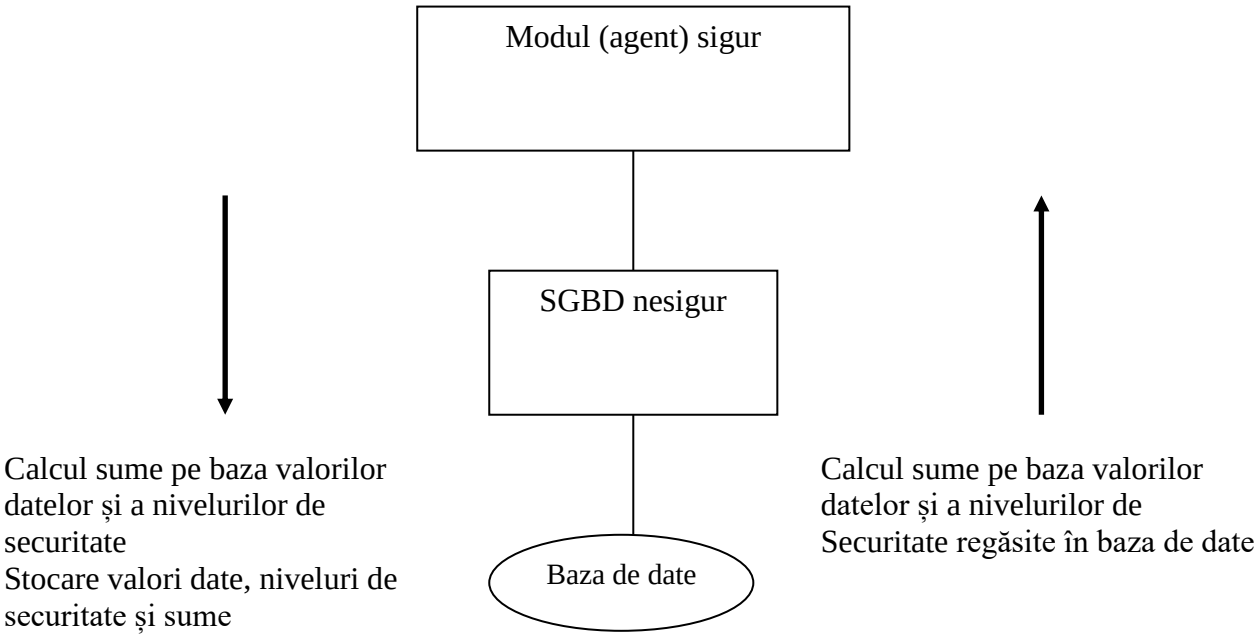


**Figura 6.** Arhitectura Integrity lock

Avantajul acestei abordări este că doar o cantitate mică de cod sigur suplimentar este cerută de către MLS/DBMS, iar performanța este independentă de numărul de niveluri de securitate care apar. Dezavantajul este că aceasta abordare constituie subiectul unor amenințări legate de interferență. Această amenințare apare deoarece *back end*-ul nesigur poate sa vadă date secrete, să le cripteze ca pe o serie de tupluri de date accesibile și să transmită tuplurile criptate *front-end*-ului sigur. Deoarece tuplurile de date sunt accesibile, filtrul sigur nu va fi capabil să detecteze operațiile ascunse asupra SGBD-ului nesigur.

### 2.3.2 Controlul accesului prin intermediul sistemului de operare

Această abordare (figura 7), cunoscută și sub denumirea Hinke-Schaefer, utilizează sistemul de operare sigur pentru a media controlul accesului. Nicio mediere a controlului accesului nu este realizată de către SGBD. Obiectele SGBD-ului (de exemplu, tuplurile) sunt tratate similar obiectelor sistemului de operare (de exemplu, fișiere). Astfel, tuplurile de pe nivelul Secret sunt stocate în fișiere Secret, iar cele Top Secret sunt stocate în fișiere Top Secret.

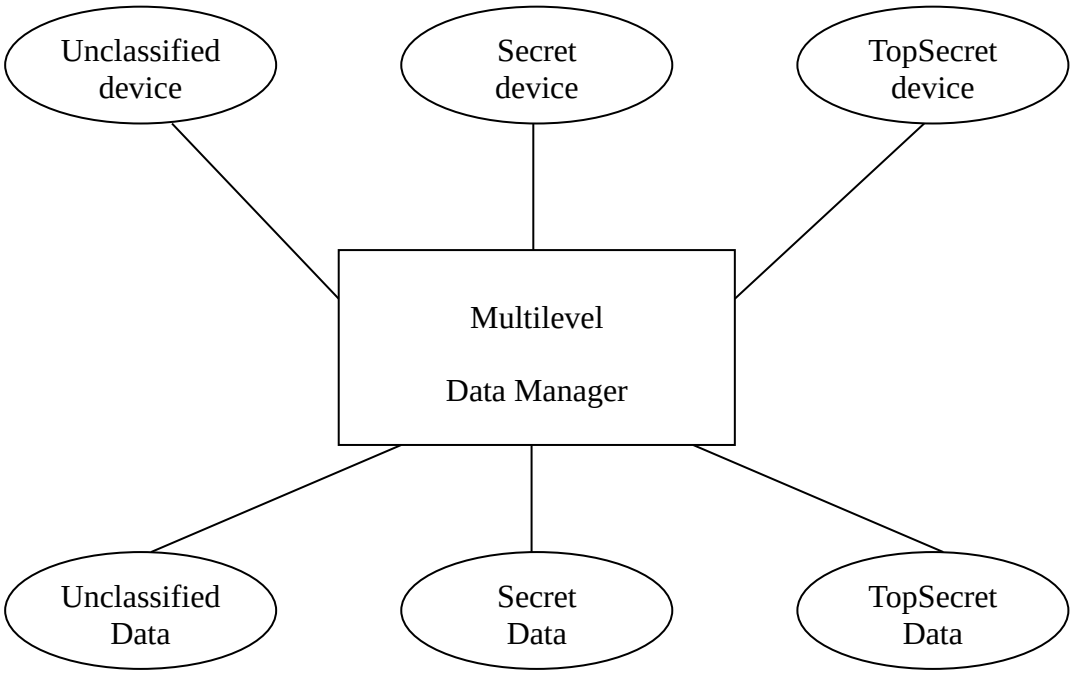


**Figura 7.** Controlul obligatoriu al accesului prin sistemul de operare

Utilizând această abordare, nu există niciun SGBD care să aibă acces la toate datele din baza de date. Există o instanțiere a SGBD-ului pentru fiecare nivel de securitate. Avantajul acestei arhitecturi constă în faptul că este simplă și sigură. Dezavantajul este acela că performanța crește odată cu numărul de niveluri de securitate. Această arhitectură mai poartă numele de arhitectură cu un singur nucleu.

### 2.3.3 Arhitectura cu extensii ale nucleului

Această arhitectură (figura 8) este o extensie a celei precedente. Sistemul de operare este utilizat pentru a furniza medierea MAC și DAC de bază. MLS/DBMS va suplimenta această mediere a accesului furnizând medierea controlului accesului. De exemplu. MLS/DBMS poate furniza DAC dependent de context asupra vizualizărilor. Aceasta abordare diferă de cea a subiectului sigur deoarece politicile impuse de către MLS/DBMS nu depind de cele ale sistemului de operare .

Această abordare are aceleași probleme de performanță ca și în cazul celei cu un singur nucleu. Însă, deoarece furnizează mecanisme de control al accesului mai sofisticate, poate răspunde anumitor nevoi de asigurare a controlului accesului din lumea reală.

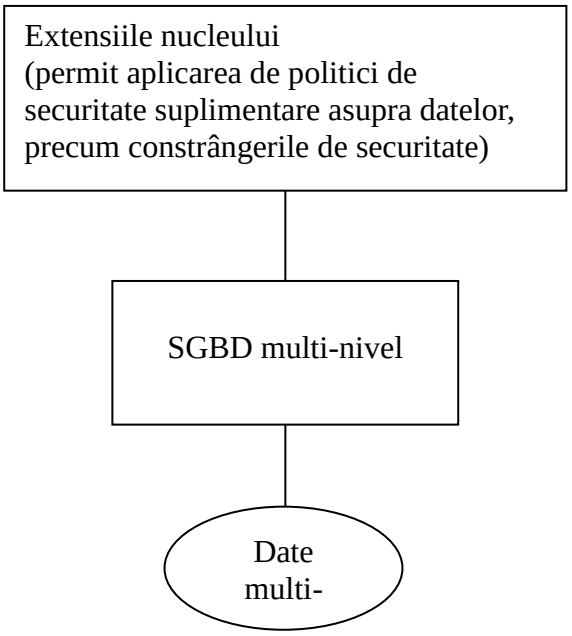


**Figura 8.** Arhitectura cu extensii ale nucleului

### 2.3.4 Arhitectura cu subiect sigur

Această abordare, denumită şi arhitectură dual bazată pe nucleu, nu se bazează pe sistemul de operare pentru a realiza medierea controlului accesului. Accesul la înregistrările SGBD-ului este mediat de către SGBD-ul sigur. Denumirea arhitecturii provine din faptul că, de obicei, SGBD-ul este un subiect (sau proces) sigur, găzduit pe sistemul de operare. În mod esenţial, SGBD-ul are acces la datele din baza de date.

Avantajul acestei arhitecturi este acela că poate furniza o bună securitate, iar performanţa este independentă de numărul de niveluri de securitate care intervin. Dezavantajul constă în faptul că trebuie să fie sigur codul SGBD-ului care realizează medierea accesului. Aceasta

presupune că abordarea poate necesita o cantitate mare de cod sigur.

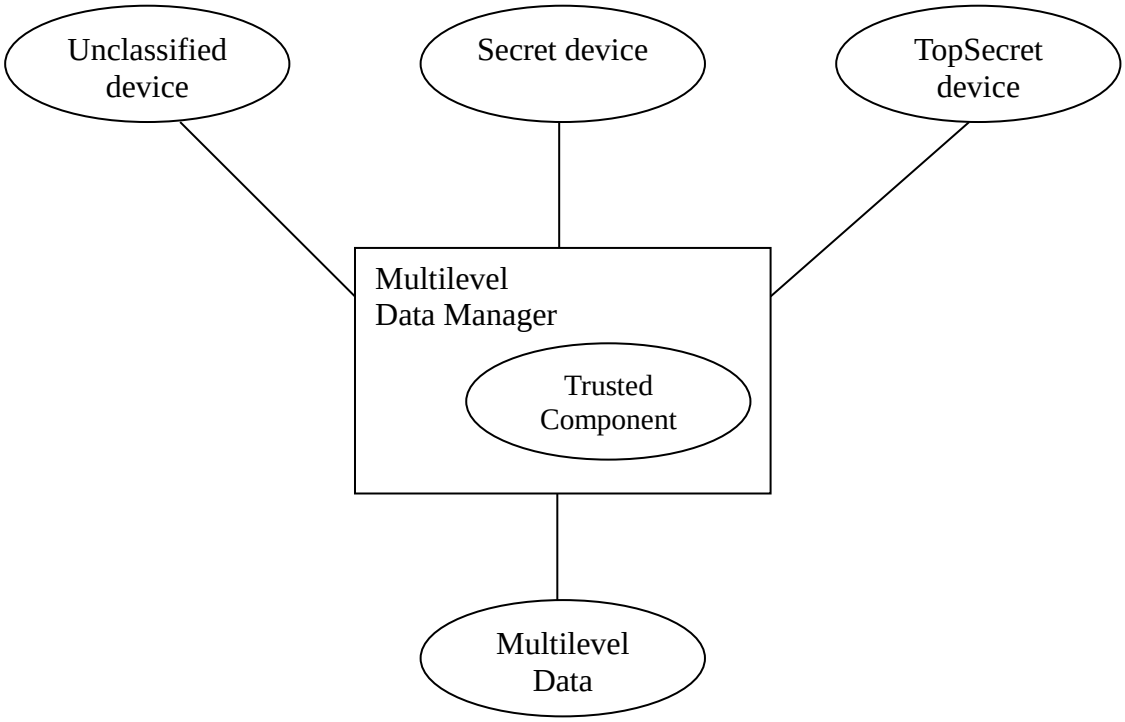


**Figura 9.** Arhitectura cu subiect sigur

### 2.3.5 Arhitectura distribuită

În cadrul acestei arhitecturi există mai multe SGBD-uri *back-end* nesigure şi un singur SGBD *front-end* sigur. Comunicările dintre SGBD-urile *back-end* au loc prin intermediul celui *front-end*. Există 2 abordări principale ale acestei arhitecturi. În prima dintre acestea, fiecare SGBD *back-end* conține date la un anumit nivel şi operează la acel nivel (figura 8). Aceasta presupune, de exemplu, că SGBD-ul *back-end* de la nivelul Secret va gestiona date Secret, iar cel de la nivelul Top Secret va gestiona date Top Secret. Această abordare se numeşte partiţionată. În cadrul celei de-a doua abordări (figura 9), datele de la nivelurile inferioare sunt replicate la niveluri mai înalte. Astfel, SGBD-ul de la nivelul Secret va gestiona datele Secret, Confidential şi Unclassified. Această abordare se numeşte replicată.

În abordarea partiţionată, *front-end*-ul sigur este responsabil de a asigura că cererea este direcţionată către *back end*-ul corect, dar şi de a efectua operaţiile de *join* asupra datelor trimise de la SGBD-urile *back end*. Deoarece cererea ar putea conţine informaţie clasificată pe un nivel mai înalt decât SGBD-ul *back-end* (de exemplu, coloanele referite în clauza WHERE a cererii), această abordare poate avea problema canalului *high signaling*. Aceasta apare deoarece cererile sunt trimise către SGBD-uri care operează la niveluri mai joase decât nivelul utilizatorului.

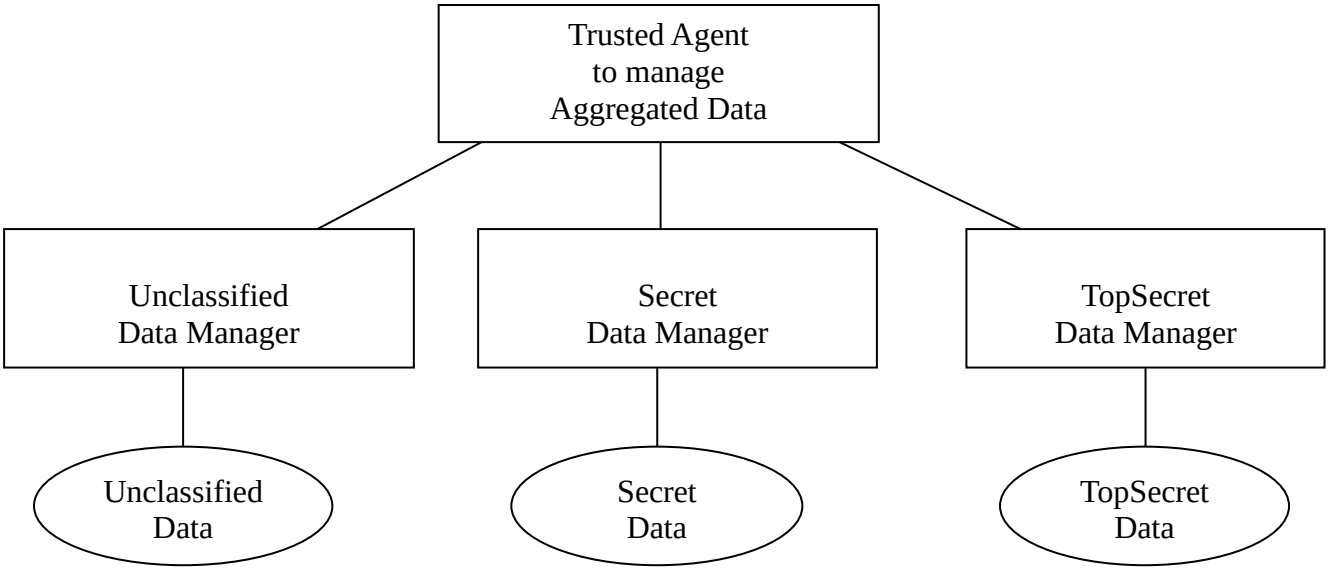


**Figura 10.** Arhitectura distribuită: abordarea partiţionată

În abordarea replicată, *front end*-ul sigur asigură faptul că cererea este direcţionatâă către un singur SGBD. Deoarece sunt interogate doar SGBD-urile care operează la acelaşi nivel ca şi utilizatorul, această abordare nu are problema semnalelor de mai sus. Mai mult, nu sunt necesare SGBD-uri *front-end* pentru efectuarea operaţiilor de *join*. Deoarece datele sunt replicate, *front-end*-ul sigur trebuie să asigure consistenţa datelor menţinute de către diferite SGBD-uri.

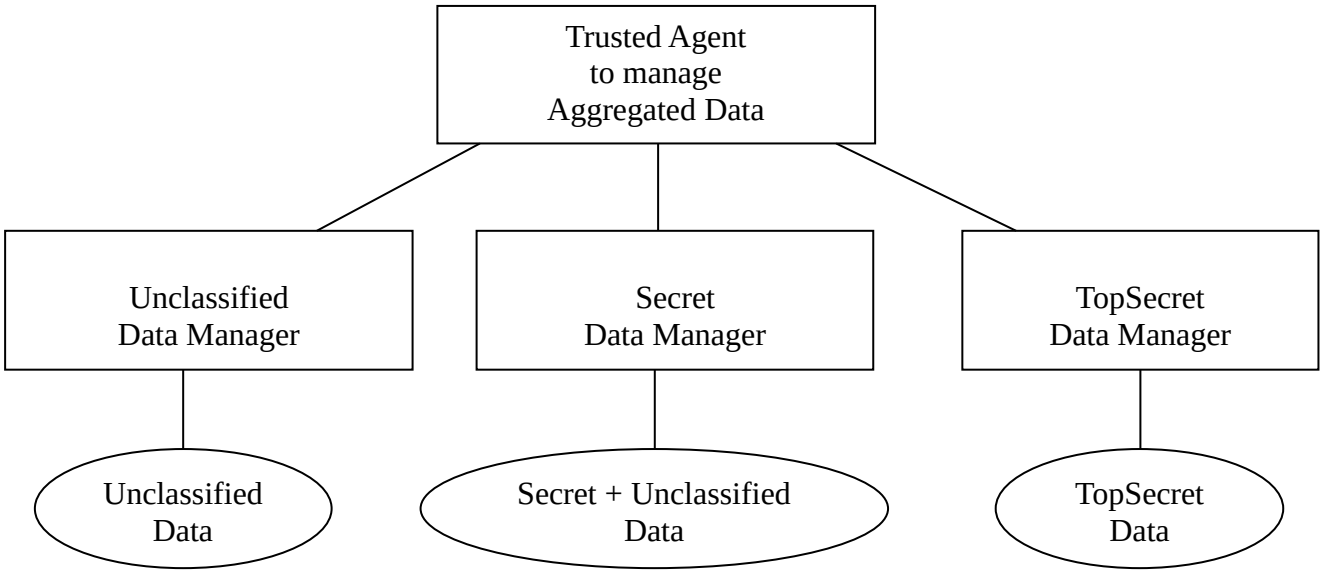


**Figura 11.** Arhitectura distribuită: abordarea cu replicare