

Modele arhitecturale de asistare a plasarii ordinelor la bursa

Claudiu VINTE
claudiu_vinte@yahoo.com

Continutul acestui material este intentionat a raspunde, într-o abordare de asamblu la nivel arhitectural, unei întrebări foarte precise: care sunt exigentele ce stau la baza proiectării unui sistem informatic dedicat asistării procesului de trimitere a ordinelor de vânzare/cumpărare către o institutie bursiera, respectiv de primire și interpretare a rezultatelor cu privire la ordinele executate – de la aceasta din urma – și care sunt conceptele arhitecturale care pot fi luate în considerare în procesul modelării unui sistem informatic integrat care să răspundă respectivelor cerințe.

Aspectele privesc în special o institutie financiara abilitata în efectuarea de tranzactii bursiere, în mod uzual pe multiple piete, și care își pune problema unei abordări integrate a activității de trading, pentru a putea veni în întâmpinarea unui diversificat portofoliu de clienti – banci, firme de investitii, companii, persoane fizice etc.

Aceasta delimitare este necesara datorita faptului ca exista piete (generic denumite OTC – over the counter) pe care, nu numai ca procesul de punere în corespondenta a cererii cu oferta se realizeaza total automat, dar participatii la activitatea tranzactionala pot si persoane fizice care au subscris la un serviciu de comunicatie cu piata respectiva, prin intermediul rețelei internet (cazul cel mai reusit fiind cel al pietei NASDAQ în Statele Unite).

Cuvinte cheie: piata bursiera, investitii, ordine, tranzactii, institutie bursiera

1. Exigente

În acest context, caracteristicile activității de *trading* într-o institutie specializata sunt reprezentate de:

- un numar relativ mare de persoane specializate în activitatea de *trading*, localizate spatial de asa natura încât să se poată realiza o coerenta a politicii și strategiilor de tranzactionare în functie de evolutia pietei;
- necesitatea unui acces partajat la o resursa de date locala, centralizata, pentru toti membrii implicati – eventuale măsuri de securitate, niveluri de control al accesului la date, filtre pe categorii de utilizatori etc.;
- nevoia unui motor sistem unic, integrat, de gestiune a tranzactiilor efectuate în cursul unei zile, de către toti participantii la activitatea de *trading*, cu facilitati de recuperare a datelor și refacere a stării sistemului în cazul unor disfunctionalitati la nivel *hardware*;
- interfata unica, comuna pentru toti utilizatorii, de accesare a institutiei bursiere, în mod uzual prin linii de comunicatie dedicate, cu protocol specific;

- activitatea de *trading* de realizeaza pe piete multiple, fiind necesara o abordare integratoare în proiectarea aplicatiilor utilizatorilor finali, astfel încât prin intermediul unei interfete flexibile să se asigure trasparența directionării ordinelor către piata bursiera specificata de utilizator (având în vedere faptul ca aceleasi instrumente financiare se pot tranzactiona pe multiple piete). Datorita acestor premise activitatea de tranzactionare bursiera într-o institutie specializata capata dimensiuni și abordări fundamentale diferite comparativ cu tratarea situației unui proces de *trading* la nivel individual, realizat de o persoana fizica înregistrata la un serviciu particular, prin intermediul rețelei *internet*, pe o anumita piata, cu instrumente specifice acelei pietei.

2. Abordari

Indiferent de abordarea arhitecturala, ca o consecinta directa a aspectelor prezentate anterior, componentele generice ale unui astfel de sistem informatic se indentifica prin:

- un motor sistem, care joaca rolul de manager pentru toate ordinele introduse în sistem cu intentia de a fi trimise catre o anumita piata si care, din punctul de vedere a intercomunicatiei între aplicatii si procese, va avea sarcina unui server de tranzactii;
- un modul care are sarcina de a asigura înregistrarea tuturor tranzactiilor pe un suport persistent si care, în coroborare cu serverul general de tranzatii, va trebui de furnizeze mecanisme de recuperare a datelor si de refacere a starii sistemului în cazul unor incidente hardware;
- o gama diversa de aplicatii client, destinate a raspunde diferitelor nevoi legate de activitatea de *trading* desfasurata de utilizatorii finali si care trebuie sa utilizeze o interfata (API) de comunicare cu motorul sistem sau cu celelalte aplicatii (functie de modelul arhitectural) comuna pentru întreg sistemul – aplicatii de plasare a ordinelor la bursa, într-o maniera singulara (ordin cu ordin), aplicatii pentru suportul *tradingului* în loturi/transe (*basket trading*), aplicatii de evaluare a tendintelor pietei, fundamentate pe preturile instrumentelor financiare furnizate de componente ca Reuther sau Tibco s.a.
- multiple linii de comunicatie capabile sa asigure conexiunea întregului sistem cu diversele piete pe care se intentioneaza reali-

zarea de tranzactii.

Diversele abordari se diferentieza prin felul în care se realizeaza comunicatia între aceste module de baza, totul traducându-se în termeni de:

- viteza de raspuns a sistemului;
- parametri de securitate;
- toleranta la incidente neprevazute;
- flexibilitate în întreținere si extensii ulterioare.

Un prim model, cel mai simplu conceptual, care realizeaza simpla conectare a modulelor componente ale sistemului într-o maniera *liniara* este ilustrat în **figura 1**.

Caracteristic acestui model este faptul ca fiecare client se conecteaza la serverul dedicat managementului ordinelor (OMS) printr-o conexiune TCP/IP (unu-la-unu). În mod concret serverul creaza o conexiune socket pentru fiecare client în parte – folosind o politica de parcurgere circulara a unei liste de conexiuni socket, prin care primeste cereri de la fiecare client si trimite raspuns tuturor clientilor, pe rând, acestia având sarcina de a-si filtra, în mod individual, mesajele primite de la server pentru a le considera doar pe cele care îi privesc. Legatura cu baza de date se poate realiza în mod direct, respectiv serverul (OMS).

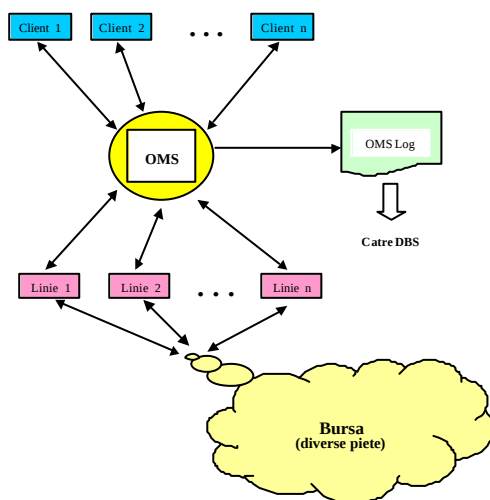


Figura 1

poate avea o conexiune directă cu un server de baze de date dedicat (Oracle, Sybase etc.) transmitându-i acestuia fiecare cerere pe ca-

re o primește de la clienți si toate răspunsurile trimise acestora. În practica însă, datorită faptului ca serverul de gestiune a or-

dinelor (OMS) si serverul de baze de date trebuie sa ruleze pe diferite masini fizice, si pentru a nu încarca fluxul principal de comunicare (scopul este întotdeauna trimiterea cât mai rapida a ordinelor clientilor catre

piata) cu datele trimise prin retea catre o serverul de baze de date aflat la distanta, se adopta un sistem simplu si robust, folosind doua fisiere text aflate pe aceasi masina fizica cu OMS, asa cum detaliaza **figura 2**.

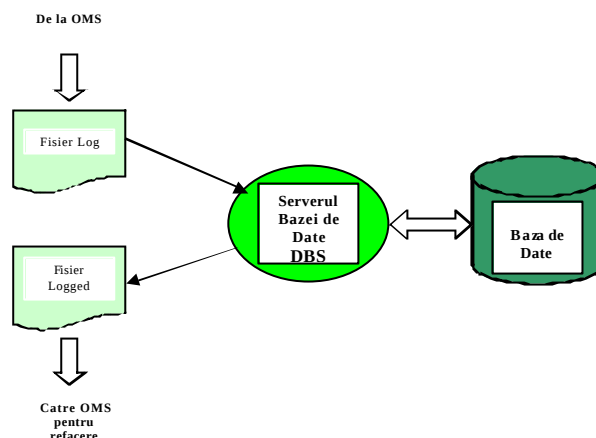


Figura 2.

Serverul de gestiune a ordinelor (OMS) va scrie, înainte de orice prelucrare, cererea primită de la client în fisierul text OMSLog, aflat pe masina locala. Se asigura în acest fel protectia datelor în cazul unei caderi a serverului si, totodata, se reduce substantial timpul de înregistrare a datelor pe un suport persistent, nefiind necesara trimiterea lor la distanta si asteptarea de mesaje de confirmare a înregistrării. Din acest moment, din punctul de vedere al OMS problema salvarii datelor este rezolvata, el putându-se concentra asupra rezolvarii concrete a cererii clientului si, eventual, trimerii ordinului de vânzare/cumparare catre liniile de comunicare cu institutia bursiera.

La acest stadiu al satisfacerii unei tranzatii intervine rolul unui al doilea server, care gestioneaza preluarea datelor din fsierul text OMSLog si transmiterea lor catre o baza de date aflata la distanta. Prin acest server (DBS) se asigura o anumita paralelizare a fluxului de prelucrare asociat serverului principal OMS, care este în acest mod eliberat de sarcina de a astepta mesaje de confirmare de la baza de date. În schimb, aceasta sarcina îi este atribuita unui server specializat (DBS) care, dupa primirea confirmarii faptului ca înregistrarea datelor s-a

realizat cu succes în baza de date, duplica datele initiale prin scrierea acestora într-un alt fisie text (OMSLogged) si verifica, pentru sincronizare, daca acestea coincid cu cele scrise de OMS în fisierul OMSLog. Serverul bazei de date (DBS) ruleaza pe aceasi masina cu OMS, pe care se gasesc si cele doua fisiere text sincronizate.

În aceasta maniera, în cazul unei caderi a serverului principal (OMS), la initializare, acesta trebuie sa aiba un mecanism de consultare si verificare a datelor din cele doua fisiere text detectând ce înregistrari nu au ajuns sa fie scrise în baza de date (pentru ca DBS scrie în fisierul text corespondent - OMSLogged - numai dupa ce a primit confirmarea înregistrării cu succes a articolelor în baza de date), fiind în masura sa refaca starea sistemului înainte de momentul caderii.

Comunicatia dintre serverul principal si liniile dedicate pietelor specifice se realizeaza în aceasta arhitectura dupa acelasi principiu. OMS deschide conexiune socket cu fiecare linie în parte si le *asculta* pe rând dupa aceeași politica *round-robin*. Serverul interpreteaza cererile clientilor si, în cazul în care se solicita trimiterea unui ordin catre o anumita piata, selecteaza linia de comunicare spe-

cifica (eventual poate alege una din mai multe linii specializate pentru o aceeași piață după un algoritm de încărcare optimă a acestora) și trimite ordinul corespunzător către aceasta. Liniile de comunicație specifice diferitelor piețe (adesea la aceeași instituție bursieră) au rolul de a converti pachetele de date din formatul propriu al sistemului de *trading*, în formatul cerut pentru piața respectivă de către instituția bursieră, și reciproc.

Principala trasatură a acestui model este faptul că fiecare componentă a sistemului comunică unu-la-unu cu o altă componentă cu care se află în conexiune directă și interfata prin care se realizează această comunicare (API) este de regulă unică în sistem, însă este greoaie și inflexibilă pentru că fiecare modul trebuie să poartă cu sine această componentă de comunicație iar modificări în cadrul protocolului și a structurilor de date folosite au impact asupra tuturor componentelor sistemului, de cele mai multe ori acestea fiind implementate în diferite limbaje de programare pe diferite platforme.

O variație pe această arhitectură, care își găsește frecvente implementări practice este prezentată în **figura 3**. Îmbunătățirea constă în realizarea unui nivel mai înalt de flexibilitate prin separarea atribuțiilor serverului unic (OMS) de gestiune a cererilor clienților și de direcționare a ordinelor specifice către liniile de comunicație corespunzătoare, introducându-se două servere distincte, specializare fiecare în satisfacerea de cereri grupate pe categorii de clienți.

Un server (OMS) va gestiona cererile aplicațiilor client asociate utilizatorilor sistemului (traderi, brokeri) și un alt server (MOMS) va fi însărcinat cu managementul comunicării cu piața prin liniile specializate. Arhitectura internă a celor două servere este similară, fiecare însă manipulând structuri de date specifice comunicăției cu clienții de care răspunde.

Pasarea mesajelor între cele două zone separate din punct de vedere logic și func-

țional se realizează printr-un program de rutare a pachetelor (Router).

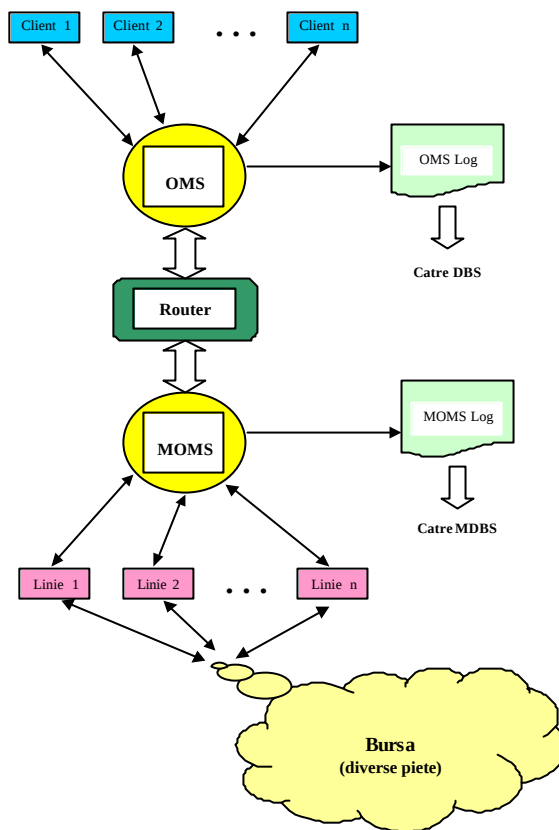


Figura 3.

Prin acest model arhitectural se introduce o întârziere în trimiterea ordinelor către piață, însă această întârziere este neglijabilă comparativ cu avantajele ce decurg din descentralizarea modelului – modularitate, flexibilitate în întreținere și extindere.

O abordare fundamental diferită constă în ideea de nu avea în mod necesar o conexiune unu-la-unu (TCP) între fiecare dintre componentele sistemului, ci posibilitatea de a putea comunica cu *toti* deodată, fiecare interpretând mesajele în funcție de rolul specific pe care îl joacă în sistem. Acest model este ilustrat de **figura 4**. În locul unei abordări liniare a fluxului de date între aplicațiile client și liniile de comunicație cu bursa, se introduce flexibilitatea unei concepții de comunicare unu-la-toti sau unu-la-mai-multi, fiecare modul putând fi conectat sau detașat din arhitectura în funcție de nevoile situației concrete.

Elementul esential al acestei arhitecturi îl constituie posibilitate de *broadcasting* (poate fi realizata si o implementare *multicasting*) a mesajelor fiecarui modul catre toate celelalte module asigurându-se, în acelasi timp, mecanisme de control al transmisiei si de retransmitere a eventualelor pachetelor pierdute. Se folosesc în acest model ambele protocoale – UDP si TCP – permitându-se o

comunicare rapida si asigurându-se o arhitectura extrem de flexibila. Componenta de comunicatie este separata integral de modulele functionale si functioneaza ca un strat intermediar între nivelul de comunicare TCP/UDP si nivelul aplicatiilor utilizator. Sa denumim aceasta componenta NetRouter (NR) si sa îi prezentam functia pe care o îndeplineste.

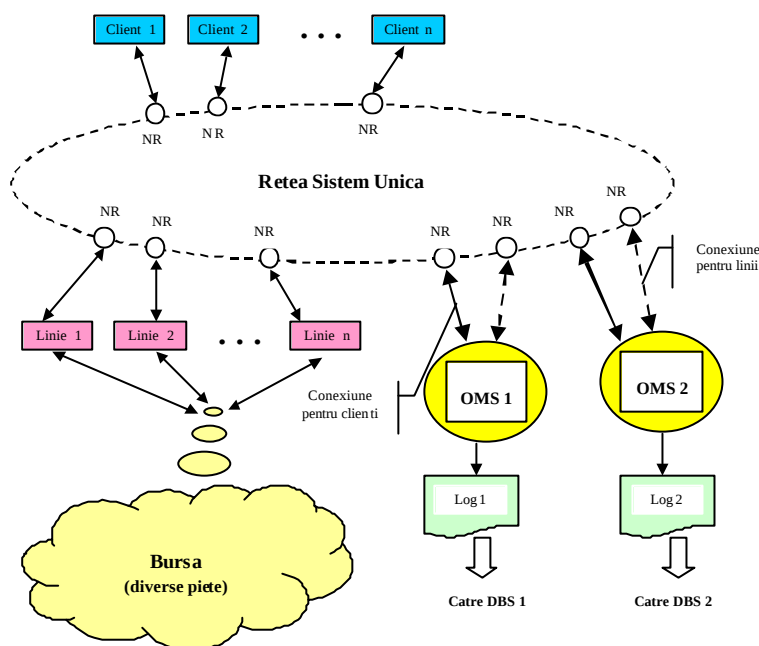


Figura 4.

Acest NetRouter asigura conectarea tuturor aplicatiilor care ruleaza pe o masina din retea la sistemul integrat de *trading*, fiind la rândul sau o aplicatie implementata într-o maniera care sa permita portarea ei pe orice platforma ce ar putea fi folosita în conditiile concrete. Pentru a fi performanta, aceasta aplicatie trebuie sa aiba fire de executie distincte pentru *ascultarea* rețelei, pentru primirea mesajelor de la aplicatiile ce ruleaza pe masina locala si pentru *broadcasting* în rețeaua sistem pentru celelalte module (descrierea implementarii unei astfel de aplicatii va face obiectul unui articol viitor).

Principial, modulele sistemului își mentin functionalitatea. Cele doua servere de tranzactii ilustreaza diferenta fundamentala în

privinta flexibilitatii unui astfel de model arhitectural: posibilitatea conectarii mai multor servere de gestiune a tranzactiilor, în acelasi timp, asigurându-se o repartizare a încărcării pe mai multe componente cu aceeasi functionalitate, pe de o parte, si realizarea unui mecanism de prevenire a caderii întregului sistem în cazul caderii unuia din servere – celelalte servere preluând în mod consistent atributiile acestuia – pe de alta parte. Trebuie remarcat ca fiecare din cele doua OMS poate fi conceput ca o *baterie de servere* dupa modelul prezentat în arhitectura *liniara*, cele doua conexiuni socket cu cele doua NetRouter-e (sau cu un acelasi NetRouter, altfel) putând fi realizate pentru serverul specializat în tranzactii cu apli-

catiile utilizator (OMS) si pentru serverul dedicat transmiterii ordinelor catre liniile de comunicatie cu bursa (MOMS).

O rafinare în plus se poate realiza prin disocierea întregului flux de mesaje în doua subretele distincte (doua segmente de retea), asa cum se prezinta în **figura 5**. Prin aceasta separare se previne încărcarea excesiva a unei rețele sistem unice prin *broadcasting-ul* realizat de toate modulele implicate. În acest model posibilitatea înlocuirii fiecarui server cu o baterie de servere specializate devine si

mai pregnantă. Poate fi folosit același model de comunicare cu baza de date ca cel descris în modelul liniar sau, datorita facilitatii de *broadcasting* se poate realiza o arhitectura *stea* în jurul bazei de date, fiecare componenta a sistemului având posibilitatea de a accesa în mod nemijlocit baza de date, consistenta accesului realizându-se printr-o politica de blocare a înregistrărilor si deblocare a acestora după modificarea câmpurilor vizate (**figura 6**).

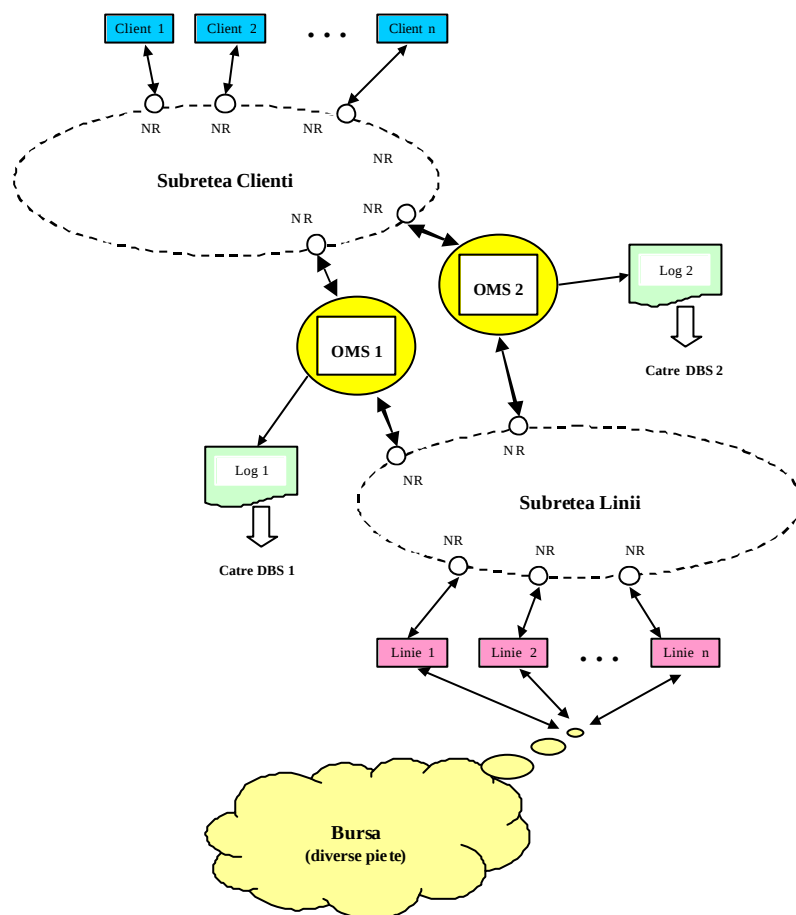


Figura 5.

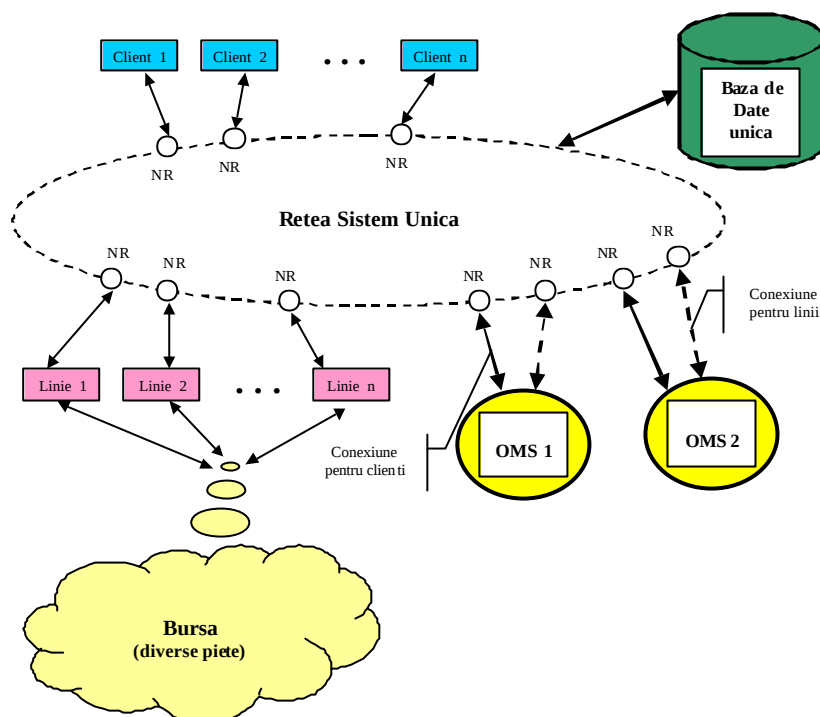


Figura 6.

Modelul prezentat în **figura 6.** diferă esențial de modelul din **figura 5.** în ceea ce privește filosofia de transmitere a mesajelor între module. Datorită posibilității accesului nemediat al tuturor componentelor sistemului la baza de date comună, nu mai este necesar un *broadcasting* de pachete care să conțină întreaga informație utilă legată de o tranzacție, ci este suficient să se trimită un pachet minimal conținând o cheie de regăsire a articolului vizat în baza de date – componenta interesată în acea tranzacție urmând să prelucreze din baza de date întreaga informație necesară. Modelul prezintă avantajul unei simplificări generale a arhitecturii însă întregul flux de comunicație este concentrat în jurul acestei baze de date unice, fapt care poate genera strângeri în funcționare și timpi de răspuns în general mai mari. Totuși, dacă pentru anumite instrumente bursiere activitatea de trading nu se bazează atât de mult pe un timp de răspuns extrem de mic (diferențele pot fi de ordinul secundelor sau zecimilor de secunde între cele două abordări, funcție de performanța componentei de stocare persistentă), cum ar fi cazul *tradingului* de *futures* și *options*, atunci modelul în *stea*

în jurul unei baze de date poate fi o soluție mai facilă de implementat, de menținut și cu un nivel ridicat de flexibilitate.

3. Tendințe

Cele prezentate până la acest moment trasează liniile generale ale manierei în care este abordată problema unui sistem integrat de *trading* într-o firmă specializată în efectuarea de tranzacții bursiere. Principial, considerentele de viteză sunt primordiale, însă la o implementare atentă a oricăruia dintre modelele prezentate se constată o uniformizare generală a timpilor de răspuns, astfel încât aspectele legate de flexibilitatea arhitecturii la satisfacerea nevoilor concrete ale unui client particular au ajuns în prezent să prevaleze. Cu toate că modelul liniar are implementări performante, incluzând mecanisme de conectare a mai multor servere de tranzacții în paralel, pentru asigurarea continuității activității tranzacționale în cazul în care se produce o cadere *hardware* (un așa numit mecanism *hot- standby* prin care fiecare server – primar – este dublat de un al doilea – secundar – pregătit în orice moment să preia sarcina serverului primar în cazul caderii

acestui) acest model prezinta limitari în ceea ce privește extinderile dinamice, ce se pot

realiza pe parcursul desfășurării *tradingului*, funcție de încărcarea sistemului.

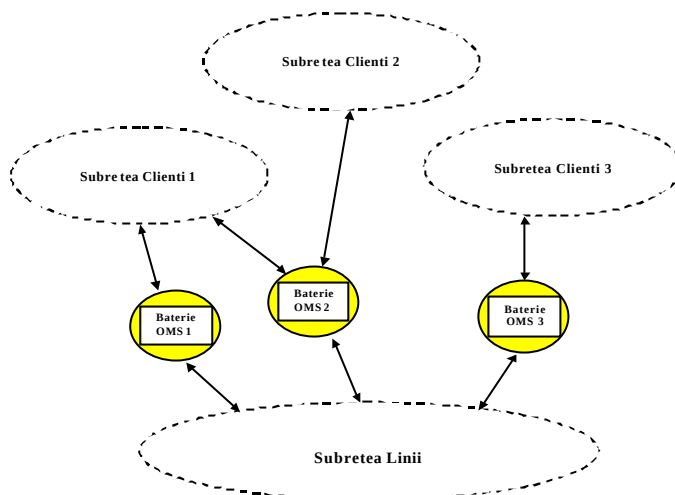


Figura 7.

De asemenea, facilitatea de *broadcasting* elimina protocoale specifice de comunicație între diferite componente particulare, pachetele ce se vehiculează având o structură standardizată, fiecare aplicație extragându-și componenta de date în care este interesată. Posibilitățile de interconectare a componentelor permit configurații care pot veni în întâmpinarea unei largi palete de contexte reale, așa cum ilustrează **figura 7.**, alăturată.

Bateriile de servere pot satisface subrețele separate atât la nivel logic cât și fizic (segmente diferite de rețea) permițând o separare a utilizatorilor pe clase de aplicații și tipuri de activități tranzacționale, realizându-se în acest fel atât îmbunătățiri în ceea ce privește încărcarea rețelei cât și mecanisme de securitate și de acces la date mult mai ușor de controlat și întreținut. Principial, se poate *alimenta* o subrețea unică de linii de conectare la instituția bursieră sau, în cazul unei multitudini de linii conectate la piețe diferite se pot face rafinări arhitecturale și la acest nivel, avantajul fiind că sistemul poate fi configurat dinamic, fără un efort suplimentar de

proiectare și implementare de componente auxiliare, necesare interconectării.

Multe firme de investiții ca Goldman Sachs, Morgan Stanley, Lehman Brothers, J.P. Morgan sau Barclays Capital folosesc diverse variante derivate dintr-un model conceptual *liniar*. Altele, ca Merrill Lynch sau Salomon Smith Barney au deja proiecte și implementări parțiale, utilizând facilități de *broadcasting*. Terenul este extrem de predispus la mutații!

Bibliografie:

- David M. Weiss - *After the Trade Is Made* - Second Edition, New York Institute of Finance, 1993
- Marcia Stigum - *The Money Market* - Third Edition, McGraw-Hill Company, Inc., 1990
- Burton G. Malkiel - *A Random Walk Down Wall Street* - W.W. Norton & Company, Inc., 1996
- Tokyo Stock Exchange, Research Dept. Press Releases
- New York Stock Exchange, Research Dept. Press Releases