

## Enterprise Application Integration Using Java Technologies

Conf.dr. Felix FURTUNĂ, prep. Alexandru BĂRBULESCU  
Catedra de Informatică Economică, ASE București

*The current article points out some of the tasks and challenges companies must face in order to integrate their computerized systems and applications and then to place them on the Web. Also, the article shows how the Java 2 Enterprise Edition Platform and architecture helps the Web integration of applications. By offering standardized integration contracts, J2EE Platform allows application servers to play a key role in the process of Web integration of the applications.*

**Keywords:** Java technology, enterprise application, web application, framework.

### **I**ntegrarea aplicațiilor la nivel de întreprindere

Integrarea aplicațiilor la nivel de întreprindere (EAI - Enterprise Application Integration) determină integrarea aplicațiilor și sursele de date ale întreprinderii astfel încât acestea să redea cu ușurința procesele și datele afacerilor. Integrarea aplicațiilor și a surselor de date trebuie să se realizeze fără a necesita schimbări semnificative ale aplicațiilor și datelor existente.

Organizațiile încearcă să combine aplicații ce rulează de cele mai multe ori pe platforme hardware diferite și care nu conțin protocoale de comunicare cu alte pachete software din afara domeniului lor îngust. În acest sens, companiile au *insule* de funcții și date ale afacerilor și fiecare insulă are propriul său domeniu de probleme.

Cum încearcă o întreprindere să rezolve această situație? Compania va aduce o echipă de consultanți și se angajează într-un proces lung și costisitor de determinare a fezabilității integrării sistemelor sale, de proiectare a abordării integrării și, în final, de dezvoltare și implementare a procedurilor (atât manual, cât și computerizat) de realizare a integrării. Câteodată, faza de analiză stabilește că nu este economic sau că nu este posibil să se integreze anumite sisteme. Chiar și atunci când integrarea înaintază, s-ar putea să dureze ani ca ea să se realizeze. De cele mai multe ori, nu există o garanție a succesului. Proiectele au fost adesea abandonate din cauza costurilor mult prea mari sau a descoperirii târzii a unor dificultăți semnificative. Chiar dacă

proiectele s-au realizat în totalitate, soluția rezultată poate avea și ea un set de probleme. EAI reprezintă o abordare diferită a acestei probleme. EAI definește semanticele pentru integrarea aplicațiilor și datelor. Adică EAI definește o metodologie sau o abordare standard pentru ca aplicațiile și sursele de date să poată comunica. Folosind acest standard, aplicațiile pot comunica cu ușurință cu alte aplicații și surse de date. Piese puzzle-ului integrării – cum ar fi un sistem inferior de management al bazei de date (DBMS – Database Management System) – se pot schimba, dar datorită metodologiei uzuale, piesa înlocuitoare poate fi introdusă și comunicarea poate continua neîntreruptă.

Concentrându-se pe integrarea proceselor și datelor afacerii, EAI cuprinde atât distribuirea acestor procese și date, cât și conceptul de reutilizare a modulelor. Cel mai important, EAI abordează această integrare ca pe un proces separat de diferite aplicații. Cu alte cuvinte, cineva poate integra mai multe aplicații, unele cu altele și cu surse fundamentale de date, fără a fi nevoit să înțeleagă sau să știe detaliile ale aplicațiilor.

EAI se potrivește mai degrabă mediilor eterogene, decât celor omogene. Mediile eterogene sunt acelea ale căror aplicații și date nu aparțin în totalitate aceluiași mediu, cum se întâmplă în exemplul AT&T discutat mai sus. O companie poate ajunge în acest punct prin achiziții și asocieri cu alte companii, caz în care au fost obligate să absoarbă sistemele unei alte companii în propriul lor mediu. Ele pot încerca să-și crească capacitatea – sau să

evite să înlocuiască sistemele existente – prin adaptarea propriilor sisteme dezvoltate intern sau a altor sisteme achiziționate la sistemele lor centrale. Sau pot susține un număr mare de utilizatori pe sisteme distribuite cu o multitudine de platforme.

### **Integrarea Web a aplicațiilor**

O dată cu apariția Web-ului, integrarea aplicațiilor la nivel de întreprindere a căpătat o semnificație mai largă decât aceea de îmbinare a sistemelor de aplicații doar în cadrul întreprinderii. Server-ele dintr-o întreprindere întrețin și lucrează cu cantități enorme de date și cu procese de afaceri foarte complexe. Mai mult, deoarece Web-ul permite accesul facil la informații și servicii, el a devenit un mijloc principal de comunicare. O companie trebuie să fie capabilă să-și facă afacerea accesibilă altora, de la proprii angajați până la partenerii externi, furnizori și cumpărători. Angajații necesită acces la datele întreprinderii pentru a ține pasul cu politica și cu ritmul de dezvoltare ale companiei și pentru a desfășura activitatea internă a acesteia. De exemplu, angajații înregistrează rapoartele de cheltuieli cu ajutorul unei interfețe Web. Partenerii de afaceri pot transmite importante informații tehnice. Cumpărătorii și furnizorii necesită acces la datele firmei pentru a facilita procesele de comandă și livrare.

Facilitând accesul la informații și servicii, integrarea Web a aplicațiilor determină ca întreprinderea să necesite o securitate mai mare. Întregul acces la server-ele companiei trebuie să aibă loc într-o manieră securizată. Nici o companie nu-și poate permite pierderea datelor sau, și mai rău, compromiterea integrității datelor.

În cele din urmă, este necesar ca toate acestea să se desfășoare într-un mediu partajabil. Indiferent de mărimea unei întreprinderi, accesul la sistemele sale trebuie să fie multiplu. O firmă nu poate risca să folosească un sistem care nu are capacitatea de a permite accesul mai multor utilizatori în același timp. De exemplu, o aplicație pentru vânzarea și cumpărarea de acțiuni oferită de industria serviciilor financiare trebuie să poată lucra cu tranzacții al căror număr crește rapid. Cel mai fa-

vorabil este atunci când întreprinderea își menține flexibilitatea de a dezvolta și adăuga noi aplicații și de a le extinde pe cele existente.

Până acum, aplicațiile erau considerate a fi fie front-office, fie back-office. Aplicațiile front-office sunt cele care privesc clientul sau utilizatorul final. Ele includ aplicații pentru managementul relațiilor cu clienții și pentru automatizarea activității de marketing. Aplicațiile back-office asigură infrastructura informațională pentru desfășurarea proceselor afacerilor back-end ale unei întreprinderii. Aplicațiile furnizate de un sistem de planificare a resurselor întreprinderii (ERP – Enterprise Resource Planning) sunt bune exemple de aplicații back-office. EAI tradițională se concentrează asupra integrării aplicațiilor front și back-office. Totuși, locul ei începe să fie luat de EAI pe Web. În loc să fie direcționate către punctul de început sau către cel de final, cele mai multe aplicații EAI sunt integrate acum pentru ambele puncte și sunt implementate pe Web.

Așa cum este imperativ ca un sistem informatic al întreprinderii (EIS – Enterprise Information System) să aibă o arhitectura Web, la fel este necesar ca aplicațiile întreprinderii să se desfășoare pe platforme de aplicații standard, folosite pe scară largă. Acum, întreprinderile privesc server-ele de aplicații ca platforme pentru dezvoltarea de aplicații Web.

Așa cum se vede din figura 1, serverele de aplicații sunt potrivite pentru domeniile companie-client (B2C – business-to-customer) și companie-companie (B2B – business-to-business) care tensionează atât de mult integrarea aplicațiilor. Server-ul de aplicații asigură un punct natural al integrării între sistemele informatice existente ale întreprinderii și aplicațiile Web. Server-ul de aplicații contribuie la desfășurarea tranzacțiilor și accesul la el poate fi partajat atunci când este nevoie. Platforma de aplicații J2EE este o tehnologie aleasă de mai multe întreprinderi și dezvoltatori de aplicații.

În figura 1 este reprezentată direcția spre care se îndreaptă în prezent companiile. Succesul limbajului de programare Java și al platfor-

mei J2EE contribuie de asemenea la această integrare Web a aplicațiilor, în mare parte da-

torită faptului că ele facilitează dezvoltarea și implementarea aplicațiilor pe Web.

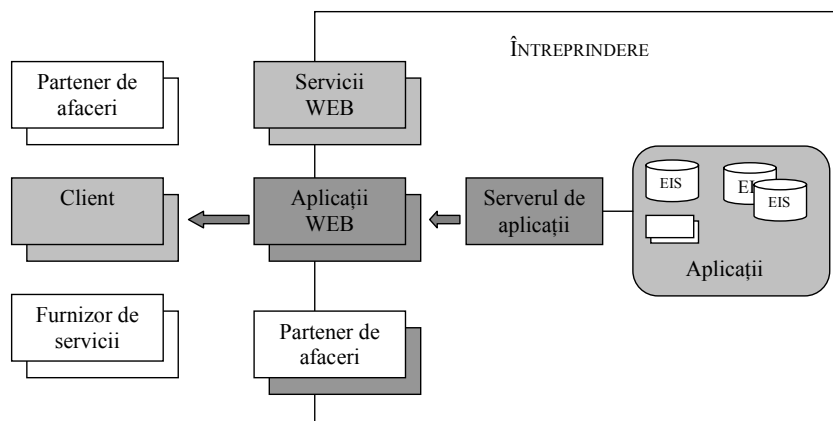


Fig.1. Implementarea Web a aplicațiilor

Pentru a optimiza integrarea Web a aplicațiilor, întreprinderile se îndreaptă din ce în ce mai mult către mediul de programare Java și platforma J2EE. Java este un limbaj independent de platforma care este proiectat pentru Web și este în același timp o platformă de dezvoltare a aplicațiilor întreprinderii larg adoptată.

Pe lângă platforma Java, companiile mai folosesc XML pentru a schimba informații corporative între domeniile de aplicații. XML este o modalitate, independentă de platformă, de a reprezenta formatele de date și este neprețuit în schimbul de date între diferite entități. Există o sinergie între XML și Java. XML reprezintă pentru date ceea ce Java reprezintă pentru serviciile de aplicații. Datorită independenței față de platforma a XML, acesta reprezintă baza curentă a tehnologiilor Web.

### Abordări ale integrării aplicațiilor întreprinderii

Există diferite abordări ale realizării integrării aplicațiilor la nivelul întreprinderii. Am identificat 5 abordări care considerăm că sunt folosite de obicei pentru integrarea sistemelor informatice existente în întreprindere cu aplicațiile acesteia. Acestea sunt:

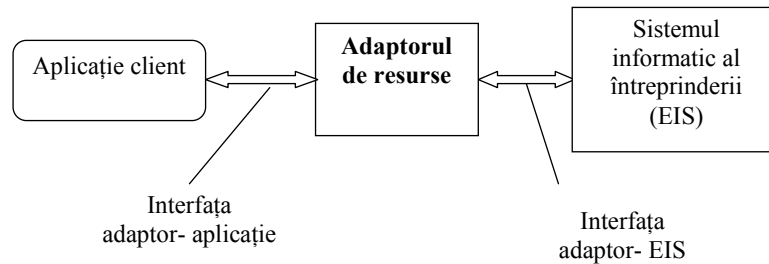
- Client – server cu două niveluri
- Adaptoarelor sincrone
- Adaptoarelor asincrone
- Abordarea bazată pe agent de mesaje
- Abordarea bazată pe server de aplicații

**A. Abordarea client – server cu două niveluri.** Această abordare este bazată pe un model client – server în două faze și nu este folosită de obicei pentru aplicații Web. Era folosită pe scară largă, înaintea apariției aplicațiilor Web, acum fiind mai puțin folosită. Cu această abordare, un EIS furnizează un adaptor ce definește un API pentru accesarea datelor și funcțiilor sale. O aplicație client tipică accesează datele și funcțiile oferite de un EIS prin interfața acestui adaptor. Clientul folosește API-ul programatic pus la dispoziție de adaptor pentru a se conecta și a accesa EIS. Adaptorul implementează suportul de comunicare cu EIS și asigură accesul la datele și funcțiile EIS. Comunicarea dintre adaptor și EIS folosește un protocol specific EIS. Acest protocol asigură suport pentru securitate și tranzacții. De asemenea, el susține propagarea conținutului de la o aplicație la EIS. Majoritatea adaptoarelor pun la dispoziție clientului un API care abstractizează detaliile protocolului inferior și mecanismul de distribuție dintre EIS și adaptor.

Un adaptor de resurse este specific unui anumit EIS. Totuși, un EIS poate furniza mai multe adaptoare decât poate folosi un client pentru a accesa EIS. Deoarece cheia adaptoarelor EIS este reutilizarea lor, sistemele informatice sau dezvoltatorii independenți de software (ISV – independent software vendor) încearcă să creeze adaptoare care folosesc un limbaj de programare utilizat pe scară largă și care prezintă un model de pro-

gramare al clientului care are cel mai mare grad de reutilizare. Un EIS asigură o formă simplă a adaptorului, în care acesta proiectează un API specific EIS ca pe un API standard, reutilizabil. Adesea, un astfel de adaptor este dezvoltat ca o bibliotecă. În acest caz,

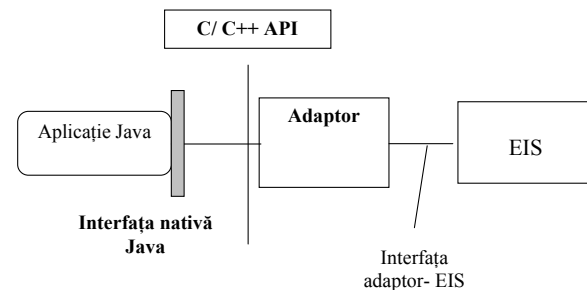
dezvoltatorul de aplicații poate folosi pentru realizarea adaptorului același limbaj de programare pe care îl folosește pentru a scrie o aplicație, fără a fi necesare modificări ale EIS.



**Fig.2.** Abordarea adaptorului de resurse EIS la EAI

De exemplu, un dezvoltator de aplicații Java poate folosi un adaptor realizat în Java pentru a accesa un EIS ce are la bază un limbaj sau o platformă diferită de Java. Un adaptor EIS poate fi dezvoltat ca o bibliotecă C (figura 6). O aplicație Java folosește o interfață nativă Java (JNI – Java Native Interface™) pentru a accesa acest adaptor asemănător unei biblioteci C. JNI reprezintă interfața nativă de programare pentru Java și face parte din Java Developers Kit (JDK™). JNI permite codului Java ce rulează pe o mașină virtuală Java (Java Virtual Machine) să opereze cu aplicații și biblioteci scrise în alte limbaje, cum ar fi C sau C++. Programatorii folosesc de obicei JNI pentru a scrie metode native când nu pot scrie întreaga aplicație în Java. Acesta este și cazul în care o aplicație Java trebuie să acceseze o bibliotecă sau aplicații deja existente scrise în alte limbaje de programare (JNI era folosită mai ales înainte de apariția platformei J2EE, acum locul ei fiind luat de arhitectura J2EE Connector).

Interfața JNI a adaptorului de resurse permite aplicațiilor Java să comunice cu biblioteca C a adaptorului. Deși această abordare este funcțională, este greu de folosit. Aplicația Java trebuie să înțeleagă modul de a invoca metodele prin interfața JNI. Această abordare nu asigură suportul J2EE pentru tranzacții, securitate și partajare. Dezvoltatorul trebuie să facă față complexității lucrului cu aceste servicii la nivel de sistem și trebuie să facă aceasta folosind complexa interfață JNI.

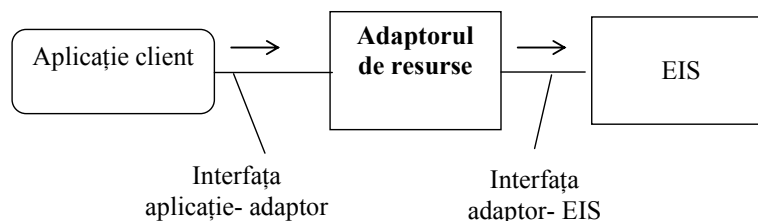


**Fig.3.** Folosirea interfeței native Java

O altă formă mai complexă a adaptorului EIS poate realiza *adaptarea* între diverse modele de componente, platforme de calcul distribuite și arhitecturi. De exemplu, un EIS poate dezvolta un adaptor distribuit care are și aptitudinea de a comunica dirijat cu EIS. Acest tip de adaptor prezintă un model client bazat pe o arhitectura a modelului de componente. Adaptorii folosesc diferite nivele de abstractizare și prezintă diferite API-uri bazate pe aceste abstractizări, depinzând de tipul EIS. De exemplu, pentru anumite tipuri de EIS, un adaptor poate prezenta un API aparte al aplicației client de apel al funcțiilor. Astfel, o aplicație client folosește acest API pentru a realiza interacțiunile cu EIS. Un adaptor pentru alte tipuri de EIS poate oferi dezvoltatorului aplicației client un model de programare a datelor. Prezintă acest tip de model de programare, o aplicație client accesează datele EIS folosind fie un model al reprezentării datelor sau de acces specific EIS, fie un model de date relațional. Este de asemenea posibil ca un adaptor să se construiască pe un API (API-ul de apel al funcțiilor di-

rijate sau API-ul de acces la date) prezentat de EIS. Cu alte cuvinte, adaptorul mai dezvoltat poate folosi nivelul inferior de abstractizare oferit de EIS pentru a construi un nivel superior al proceselor activității sau al abstractizării obiectelor activității pentru dezvoltatorii de aplicații client.

**Folosirea adaptoarelor sincrone.** Un adaptor poate prezenta o modalitate de comunicare între aplicațiile client și EIS fie sincronă, fie asincronă. Figura 4 ilustrează folosirea adaptoarelor proiectate pentru comunicarea sincronă. Aceste adaptoare furnizează un model sincron de comunicare cerere – răspuns între aplicație și EIS.

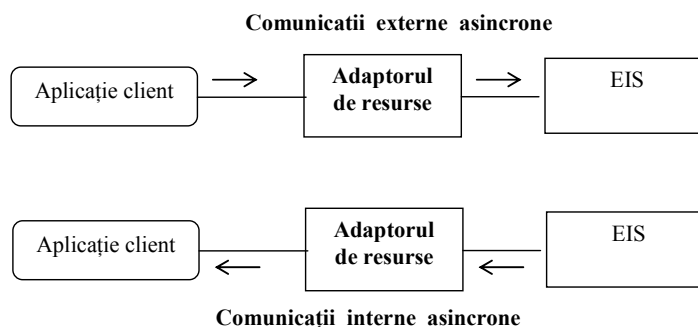


**Fig.4.** Folosirea unui adaptor sincron

Cum funcționează un adaptor sincron? Să considerăm ca exemplu un adaptor care definește un API ce include o funcție dirijată apelabilă de o aplicație. Această funcție creează în EIS un articol de contabilitate a clienților. Când o aplicație dorește să interacționeze cu EIS pentru a crea un astfel de articol, ea invocă această funcție. Aplicația se execută și întoarce apelantului răspunsul ce conține rezultatele funcției în EIS. O astfel de interacțiune este considerată sincronă deoarece execuția aplicației apelante așteaptă în mod sincron în timp ce funcția se execută în EIS.

O formă a adaptorului sincron permite comunicarea sincronă bidirecțională între o aplicație și un EIS. Acest tip de adaptor permite unui EIS să apeleze sincron o aplicație.

**Folosirea adaptoarelor asincrone.** Adaptoarele asincrone asigură o altă abordare a integrării aplicațiilor. Să folosim același exemplu al unui adaptor care prezintă un API cu o funcție dirijată ce permite unei aplicații să interacționeze cu EIS și să creeze un articol de contabilitate a clienților. Această funcție este apelabilă de o aplicație.



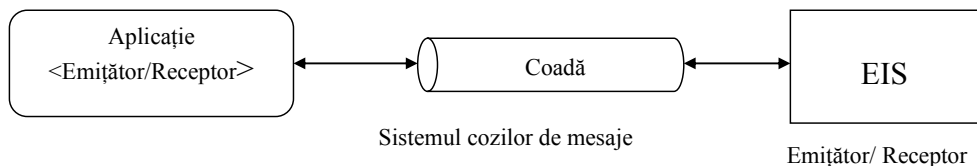
**Fig.5.** Folosirea unui adaptor asincron

În comunicarea asincronă o aplicație apelează funcția dirijată pentru a crea un nou articol de contabilitate a clienților în cadrul EIS. Aplicația face apelul dirijat și imediat se întoarce și continuă propria procesare. Funcția dirijată este trimisă la EIS. Acesta prelucrează funcția și întoarce ca răspuns aplicației o oarecare informație ca o invocare asincronă separată. Adaptorul de resurse transmite ape-

lul asincron de la EIS la aplicație. Un fapt important de ținut minte este acela că aplicația nu își suspendă execuția cât timp funcția dirijată rulează în cadrul EIS. Astfel aplicația își continuă activitatea și primește la un moment dat notificări cu privire la rezultatele invocării funcției dirijate. Așadar, un EIS poate invoca sau apela asincron o aplicație.

**Abordarea folosind agentul de mesaje.** Comunicarea asincronă bazată pe mesaje poate fi folosită de asemenea pentru integrarea aplicațiilor și sistemelor informatice ale întreprinderii. Există două forme de transmitere asincronă a mesajelor:

•transmiterea bazată pe coadă de mesaje  
•transmiterea bazată pe publicare – înregistrare

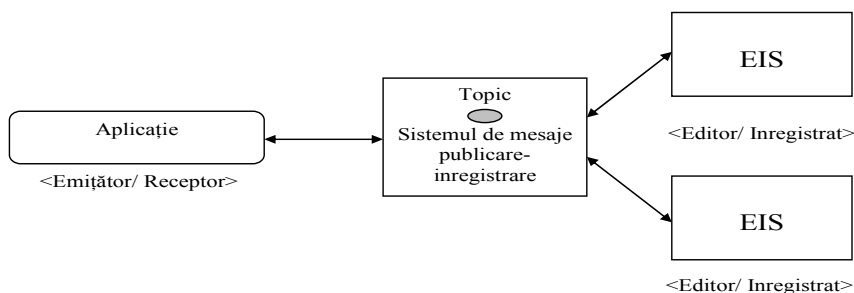


**Fig.6.** Folosirea cozilor de mesaje pentru integrarea EIS

Figura 6 ilustrează comunicarea bazată pe coadă de mesaje. Acest tip de comunicare, care mai este numită și mesagerie punct – la – punct, presupune existența unei aplicații ce trimite mesaje într-o coadă de mesaje. În comunicarea bazată pe coadă de mesaje, o astfel de coadă independentă față de aplicațiile emițător și receptor se comportă ca un

buffer de mesaje între aplicațiile comunicante. Aplicația emițător trimite un mesaj în această coadă, iar aplicația receptor primește mesajele din aceeași coadă.

**B. Abordarea de tip publicare – înregistrare.** Abordarea de tip publicare – înregistrare funcționează diferit de cea bazată pe coadă.



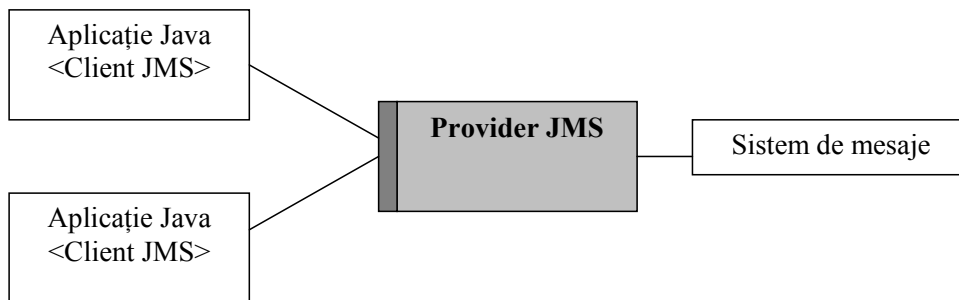
**Fig.7.** Folosirea unui sistem de publicare – înregistrare pentru integrarea EIS

Figura 7 ilustrează mesageria de tip publicare – înregistrare. Aceasta ar putea fi un serviciu de acțiuni care publică – modifică prețurile acțiunilor – pentru aplicațiile portofoliilor înregistrate. În mesageria de tip publicare – înregistrare există editoare de mesaje care creează mesajele și înregistratoarele de mesaje, care înregistrează interesul pentru un anumit mesaj. Există de asemenea o facilități de publicare – înregistrare care funcționează ca un punct de integrare – editoarele publică mesajele către această facilități și ea trimite aceste mesaje către înregistratoarele.

Iată cum funcționează mesageria de tip publicare – înregistrare. O aplicație de publicare publică mesaje pe diferite teme, cum ar fi cota de ultim moment a bursei. Mai multe aplicații se pot înregistra pentru această temă și primesc mesajele publicate de editor. Facilități de publicare – înregistrare are respon-

sabilitatea de a transmite mesajele publicate către aplicațiile înregistrate în funcție de temele avute în vedere.

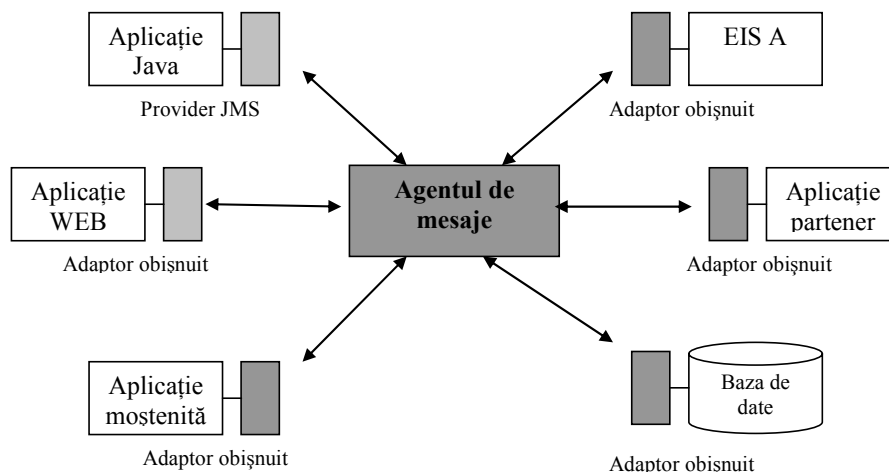
Atunci când o aplicație necesită folosirea fie a mesageriei bazate pe coadă de mesaje, fie a celei de tip publicare – înregistrare, ea trebuie să se conecteze la un sistem de mesagerie ce furnizează aceste mecanisme. De obicei, aplicația folosește un API oferit de sistemul de mesagerie pentru a accesa serviciile acesteia. Sistemul de mesagerie folosește un adaptor de mesaje, numit și furnizor, pentru a implementa API-ul de mesaje. Serviciul de mesaje Java (JMS – Java Message Service) asigură un API pentru sistemele de mesagerie ale întreprinderii. Aplicațiile, numite clienți JMS, folosesc API-uri JMS pentru a accesa serviciul de mesaje și fie sistemul de mesaje bazat pe coadă, fie cel de tip publicare – înregistrare (figura 8).



**Fig.8.** Folosirea unui provider JMS

Figura 8 ilustrează folosirea unui agent de mesaje pentru integrarea EIS. A se observa că un adaptor permite unei aplicații să acceseze agentul de mesaje. În acest scenariu, un adaptor proiectează interfața la nivel de aplicație pentru agentul de mesaje până la mecanismele inferioare de mesagerie asincronă susținute de agentul de mesaje. În plus, adaptorul proiectează formatele mesajelor susținute de agentul de mesaje. (De exemplu, mecanismele inferioare de mesagerie asincronă

pot fi bazate pe coadă sau pe publicare – înregistrare). Unele adaptoare adaugă un plus de funcționalitate între aplicații și agentul de mesaje. De exemplu, ele adaugă capacitatea de a transforma mesajele – un adaptor poate transforma mesaje specifice aplicațiilor într-un format cerut de agentul de mesaje. Acesta din urmă convertește apoi mesajul într-un format cerut de receptorul sau înregistratorul de mesaje.



**Fig.9.** Folosirea unui agent de mesaje pentru integrarea EIS

Aplicațiile și sistemele informatice ale întreprinderii, folosind un agent de mesaje pentru înregistrarea și transmisia de mesaje, se pot comporta atât ca și consumatoare de mesaje, cât și ca producătoare. De exemplu, o aplicație de contabilitate financiară se poate înscrie pentru mesaje ce conțin informații despre tranzacțiile financiare. O aplicație de gestionare a comenzilor poate trimite prin agentul de mesaje, mesaje ce modifică un cont de credit în aplicația de contabilitate. Dezvoltatorii de agenți de mesaje furnizează adaptori specifici lor pentru EIS recunoscute. Când o aplicație și un EIS comunică folosind

mesajele asincrone se consideră ca ele sunt slab cuplate. Există atât avantaje, cât și dezavantaje ale integrării slab cuplate. În cazul acestui tip de integrare între un EIS țintă și o aplicație, aplicația poate continua prelucrarea cererilor clienților fără a afecta performanța EIS sau fără a avea defecțiuni de comunicare. Aceasta crește gradul de partajare. Totuși, dezvoltatorii de aplicații pot găsi dificilă programarea unui model de mesagerie asincronă. În același timp, aceste sisteme de mesaje asincrone nu susțin întotdeauna propagarea securității și contextele tranzacționale. Un agent de mesaje furnizează servicii adiționale

pentru integrarea aplicațiilor întreprinderii. Aceste servicii sunt: generarea mesajelor, gestionarea tranzacțiilor, transmisia sigură a mesajelor, prioritizarea și ordonarea mesajelor și transformarea mesajelor.

### Integrarea aplicațiilor bazată pe server

Un server de aplicații este un punct natural pentru integrarea aplicațiilor, deoarece el asigură o platformă pentru dezvoltarea, desfășurarea și gestionarea aplicațiilor Web ale întreprinderii. Server-ele de aplicații reprezintă o platformă pentru aplicații dezvoltate folosind o arhitectură pe mai multe nivele. O aplicație tipică pe mai multe nivele conține de fapt 3 nivele: nivelul client, nivelul de mijloc și nivelul EIS. Nivelul de mijloc implementează de obicei logica activității pentru o aplicație. Ca parte a implementării funcționalității logicii activității unei aplicații, nivelul de mijloc poate accesa date și funcții asociate aplicațiilor ce rulează pe nivelul EIS. Nivelul de mijloc servește în același timp ca și conținut al prezentării statice și dinamice

pentru nivelul client. Nivelul EIS conține sistemele ce rulează aplicațiile și bazele de date existente în întreprindere. Așa cum am mai spus, aceste EIS pot fi aplicații customizate sau pot fi aplicații gata făcute. Nivelul client este compus din diferite tipuri de aplicații client. Un client poate fi o aplicație peer sau un client HTML bazat pe un browser Web. Un server de aplicații obișnuit susține un model bazat pe componente pentru dezvoltarea aplicațiilor. În cadrul acestui model, o aplicație se poate compune din diferite tipuri de componente, cum ar fi componentele Web sau componente ale activității. Server-ul de aplicații asigură suportul pentru desfășurarea și timpul de execuție al acestor componente ale aplicațiilor. Ca urmare, un server de aplicații asigură o platformă deosebit de folositoare dezvoltării aplicațiilor Web, tranzacționale, securizate, distribuite și partajabile. Aceasta crește importanța server-ului de aplicații pentru integrarea aplicațiilor întreprinderii.

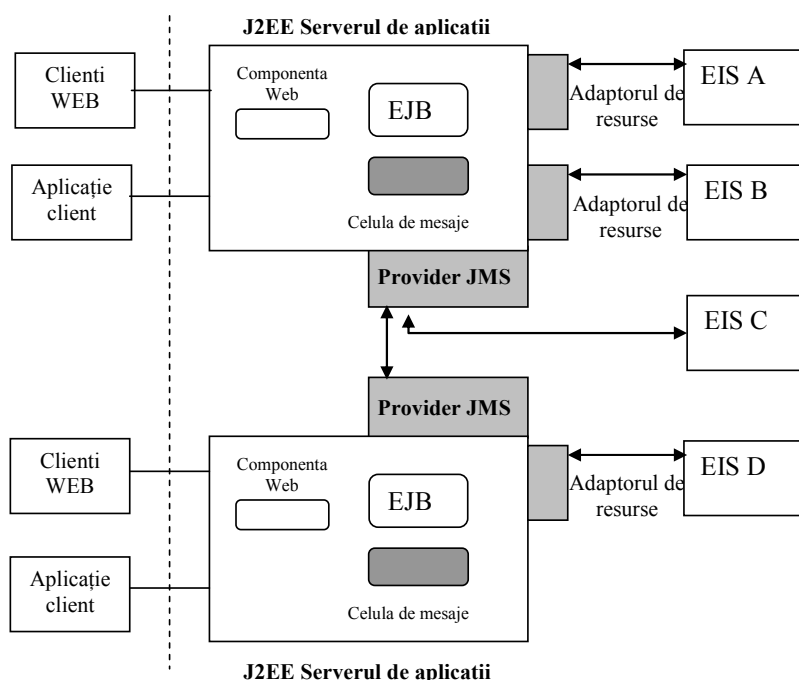


Fig. 10. EAI bazată pe serverul de aplicații

Un server de aplicații furnizează un set de servicii de execuție pentru componentele sale aflate în desfășurare. Aceste servicii de execuție sunt ascunse componentelor aplicației printr-un model simplificat de programare a aplicațiilor. Serviciile furnizate includ: suport

pentru tranzacții, securitate, durata de încărcare și eroarea la încărcare, accesul la baza de date, mesageria asincronă, comunicațiile distribuite, protocoalele Web, suportul XML. Este posibil să se dezvolte și să se ruleze aplicații pe un server de aplicații astfel încât



aplicațiile să poată conecta și agrega accesul la sisteme informatice eterogene și la aplicațiile existente în întreprindere. Atunci când sunt realizate cu această capacitate de a accesa sistemele informatice multiple și eterogene, componentele Web și cele ale afacerii ce se desfășoară pe nivelul de mijloc (sau pe server-ul de aplicații) folosește adaptoare pentru a accesa datele și funcțiile asociate cu aplicațiile acestor EIS-uri.

Componentele aplicației care rulează pe server-ele de aplicații folosesc adaptoare de resurse sincrone pentru a conecta și accesa EIS. Componentele aplicației pot folosi de asemenea un adaptor pentru ca un agent de mesaje să se integreze cu sistemele informatice bazate pe mesageria asincronă.

### Concluzii

Există un trend clar printre organizații din punct de vedere al integrării aplicațiilor și sistemelor lor informatice deja existente cu aplicații și servicii Web. Din ce în ce mai mult, companiile trebuie să-și facă simțită prezența pe Web și să-și activeze serviciile și pentru potențialii clienții de pe Web. Totuși,

în același timp, o companie nu-și poate permite să renunțe la sistemele și aplicațiile lor deja existente, trebuind să le determine să devină profitabile. Platforma J2EE pune la dispoziție o soluție flexibilă, simplă, necostisitoare și ușor de adaptat la necesitățile organizației. Datorită tehnologiei alese există posibilități de dezvoltări ulterioare a aplicațiilor, precum și posibilitatea distribuirii aplicației prin configurarea serverului de aplicație în sistem cluster.

### Bibliografie

- [1] \* \* \*, *Java 2 Platform, Enterprise Edition – Documentation*, <http://java.sun.com/j2ee/docs.html>
- [2] \* \* \*, *Java Foundation Classes*, <http://java.sun.com/products/jfc/index.jsp/>
- [3] Boian, F.,M., Boian, R.,F., *Tehnologii fundamentale Java pentru aplicații Web*, Editura Albastră, Cluj-Napoca, 2004
- [4] Graham, S., Simeonov, S., Boubez, T., *Servicii Web cu Java: XML, SOAP, WSDL și UDDI*, Teora, București, 2003
- [5] Tănasă, S., Olaru, C., *Dezvoltarea aplicațiilor Web folosind Java*, Polir om, Iași, 2005