

Evaluarea abordărilor serviciilor Web pe platformele J2EE și .Net

Gabriela Rodica Hrin, Lucian Emanuel Anghel, Mihaela Tomescu, Daniel Savu, Marilena Iuliana Iiescu

*Institutul Național de Cercetare – Dezvoltare în Informatică – ICI
Rodica.Hrin@ici.ro*

Pentru efectuarea unei evaluări comparative a abordărilor serviciilor Web pe platforma J2EE furnizată de Sun Microsystems și respectiv pe platforma .Net furnizată de Microsoft, au fost considerate următoarele patru provocări:

- descrierea serviciului;
- implementarea serviciului;
- publicarea, descoperirea și conectarea serviciului;
- invocarea și executarea serviciului.

1. Descrierea serviciului

Pentru ca serviciile Web să fie utilizate pe scară largă este necesară descrierea acestora într-un mod structurat. În acest scop, limbajul de descriere a serviciilor Web (WSDL¹) a definit o gramatică XML² care permite descrierea serviciilor Web drept colecții de puncte terminale sau porturi acceptate de mesaje.

În WSDL, definiția abstractă a punctelor terminale și a mesajelor este distinctă de implementarea concretă a acestora sau de legăturile dintre ele. Specificațiile protocolului concret și ale formatului de date pentru un anumit tip de punct terminal constituie o legătură. Un punct terminal este definit prin asocierea unei adrese Web cu o legătură, iar o colecție de puncte terminale definește un serviciu.

• J2EE

Serviciile Web implementate pe platforma J2EE schimbă informații cu părțile interesate utilizând WSDL în scopul realizării unui acord privind formatul adecvat pentru fiecare document XML transferat. Pentru efectuarea de tranzacții de afaceri prin intermediul unei organizații care oferă servicii Web implementate pe platforma J2EE, părțile terțe pot căuta într-un registru informații referitoare la serviciile Web ale organizației.

• .Net

Pentru a se autodescrie, un serviciu Web .Net asigură suportul pentru specificația WSDL 1.1 și utilizează un document WSDL în același mod ca un serviciu Web J2EE. Diferența este dată de faptul că pentru un serviciu Web .Net se utilizează un spațiu de nume XML într-un document WSDL în scopul identificării unice a punctelor terminale ale serviciului Web.

.Net furnizează o componentă pe partea de client și o componentă pe partea de server. Componenta pe partea de client permite unei aplicații să invoce operațiile serviciului Web descrise într-un document WSDL. Componenta pe partea de server transpune operațiile serviciului Web în apeluri de metode de

¹ WSDL - Web Service Description Language - Limbaj de descriere a serviciilor Web

² XML - eXtensible Markup Language - Limbaj de marcare extins

obiecte COM³ în conformitate cu descrierile dintr-un document WSDL și dintr-un fișier care este scris în meta-limbajul pentru servicii Web (WSML⁴) și este necesar pentru implementarea Microsoft a SOAP⁵.

2. Implementarea serviciului

Implementarea serviciilor Web presupune structurarea datelor și operațiilor în interiorul unui document XML conform cu specificația SOAP. După implementarea unei componente a serviciului Web, între client și componentă are loc un schimb de mesaje sub formă de documente XML.

- **J2EE**

Clasele și aplicațiile Java existente pot fi împachetate utilizând API-ul Java pentru RPC⁶ bazat pe XML (JAX-RPC⁷) și pot fi expuse ca servicii Web. JAX-RPC utilizează XML pentru a efectua apelarea de la distanță ale procedurilor (RPC) și expune o API⁸ care poate fi utilizată atât pentru împachetarea și trimiterea parametrilor argumentelor, cât și pentru despachetarea acestora în scopul transmiterii și primirii apelurilor de proceduri.

Prin intermediul J2EE, serviciile pentru afaceri create sub formă de componente Enterprise JavaBeans (EJB) sunt împachetate și expuse ca servicii Web. Pachetul rezultat este un serviciu Web acceptat de SOAP care este conform cu o interfață WSDL bazată pe metodele EJB.

Arhitectura serviciilor Web J2EE se prezintă sub forma unui set de cadre de lucru bazate pe XML, care furnizează infrastructuri ce permit organizațiilor să integreze logica serviciului de afaceri în interfețe proprietare. J2EE asigură suportul pentru serviciile Web prin intermediul JAXP⁹ care permite analizarea manuală a documentelor XML de către dezvoltatorii care doresc să efectueze operații prin intermediul serviciilor Web.

- **.Net**

Programele aplicațiilor .Net sunt compilate într-un cod binar intermediar denumit limbaj intermediar al Microsoft (MSIL¹⁰), nemaifiind executate în codul mașină nativ. Codul binar portabil este apoi compilat în codul nativ utilizând pe durata execuției un compilator JIT¹¹ și este executat într-o mașină virtuală denumită limbaj comun de execuție (CLR¹²). Acest mod de funcționare este similar cu modul de funcționare Java, cu excepția faptului că .Net conține mai multe limbaje. Fiecare dintre aceste limbaje este translatat în MSIL, care este executat în CLR utilizând JIT. Împreună cu platforma .Net, Microsoft oferă câteva limbaje bazate pe infrastructura comună a limbajelor (CLI¹³), cum sunt Managed C++, JScript, VB.Net și C#.

3. Publicarea, descoperirea și conectarea serviciului

După implementarea unui serviciu Web, acesta trebuie publicat pentru a permite părților interesate să-l găsească. Clienții trebuie să aibă acces la informații referitoare la modul în care se pot conecta la un serviciu Web și pot interacționa cu acesta. Aceste informații de conectare și interacțiune sunt referite ca informații de

³ COM - Component Object Model - Model de obiecte componente

⁴ WSML - Web Services Meta Language - Meta-limbajul pentru servicii Web

⁵ SOAP - Simple Object Access Protocol - Protocol simplu de acces la obiecte

⁶ RPC - Remote Procedure Calls - Apelare de la distanță a procedurilor

⁷ JAX-RPC - Java API for XML-based RPC - API Java pentru RPC bazat pe XML

⁸ API - Application-Programming Interface - Interfață de programare a aplicației

⁹ JAXP - Java API for XML Parsing - API Java pentru analizarea XML

¹⁰ MSIL - Microsoft Intermediate Language - Limbajul intermediar al Microsoft

¹¹ JIT - Just In Time - Imediat

¹² CLR - Common Language Runtime - Limbaj comun de execuție

¹³ CLI - Common Language Infrastructure - Infrastructura comună a limbajelor

legătură. Regiștrii sunt mijloacele principale de publicare, descoperire și interconectare a serviciilor Web. Regiștrii conțin structuri de date și taxonomii utilizate pentru descrierea serviciilor Web și a furnizorilor de servicii Web. Un registru poate fi găzduit de organizații private sau de părți terțe neutre.

IBM și Microsoft au colaborat la realizarea specificației limbajului pentru examinarea serviciilor Web (WSIL¹⁴) care permite aplicațiilor să navigheze prin servere Web pentru a găsi servicii Web XML. WSIL este complementar lui UDDI¹⁵ facilitând descoperirea serviciilor disponibile pe site-uri Web care nu sunt înregistrate în regiștrii UDDI.

• J2EE

Sun Microsystems consideră API-ul Java pentru regiștrii XML (JAXR¹⁶) ca fiind o interfață unică al cărei obiectiv este interoperarea mai multor tipuri de regiștri. JAXR permite clienților accesarea unor servicii Web furnizate de implementatorii care expun servicii Web create pe o implementare a specificației JAXR.

Tipurile de furnizori de JAXR sunt:

- JAXR Pluggable Provider: implementează caracteristici ale specificației JAXR ce sunt independente de orice tip specific de regiștri;
- Registry-specific JAXR Provider: implementează specificația JAXR într-un mod specific unui registru;
- JAXR Bridge Provider: nu este specific nici unui registru și servește drept punte către o clasă a regiștrilor, cum sunt ebXML¹⁷ sau UDDI.

• .Net

Într-o primă fază, Microsoft a oferit o modalitate de descoperire a serviciilor Web prin utilizarea unui mecanism simplu bazat pe HTTP¹⁸ (DISCO¹⁹) sub forma unui fișier de descoperire. Un fișier DISCO publicat este un document XML care conține legături la alte resurse ce descriu serviciul Web. Din momentul adoptării pe scară largă a UDDI, Microsoft a susținut acest set de specificații pentru interoperabilitate deoarece UDDI permite eficientizarea relațiilor de interoperabilitate dintre soluții.

SDK-ul UDDI oferă un server .Net UDDI și suport pentru Visual Studio .Net, bazându-se pe cadrul de lucru .Net. Unele produse Microsoft, cum este de exemplu Microsoft Office XP, oferă suport pentru descoperirea serviciilor prin intermediul UDDI.

4. Invocarea și executarea serviciului

SOAP este un protocol simplu bazat pe XML care definește un cadru de lucru de transmitere a mesajelor ce asigură schimbul de date și informații structurate prin intermediul Web-ului.

O specificație SOAP constă din următoarele părți principale:

- un plic obligatoriu pentru încapsularea datelor;
- reguli opționale de codificare a datelor pentru reprezentarea tipurilor de date definite de aplicație și un model pentru serializarea modelelor de date nesintactice;
- un șablon de schimb de mesaje cerere / răspuns;
- o legătură opțională între SOAP și HTTP.

¹⁴ WSIL - Web Services Inspection Language - Limbaj pentru examinarea serviciilor Web

¹⁵ UDDI - Universal Description, Discovery, and Integration - Descrierea universală, descoperirea și integrarea

¹⁶ JAXR - Java API for XML Registries - API Java pentru regiștrii XML

¹⁷ ebXML - Electronic Business Extensible Markup Language - Limbaj de marcare extensibil pentru afaceri electronice

¹⁸ HTTP - Hypertext Transfer Protocol - Protocol de transfer hipertext

¹⁹ DISCO - Discovery Protocol - Protocol de descoperire

SOAP poate fi utilizat în combinație cu orice protocol sau mecanism de transport care poate să transporte plicul SOAP.

Destinatarii serviciului Web operează ca ascultători SOAP și pot anunța părțile interesate (alte servicii Web sau aplicații) atunci când este primită o cerere de serviciu Web. Ascultătorul SOAP validează un mesaj SOAP în funcție de schemele XML corespunzătoare, conform definiției dintr-un fișier WSDL. În continuare, ascultătorul SOAP despachetează mesajul SOAP. În ascultătorul SOAP, distribuitorii mesajului pot invoca implementarea codului corespunzător serviciului Web.

În final, este invocată logica de afaceri în scopul obținerii unui răspuns. Rezultatul logicii de afaceri este transformat într-un răspuns SOAP și transmis apelantului serviciului Web.

• J2EE

J2EE utilizează JAX-RPC în scopul trimerii de apeluri de metode SOAP la „părțile” aflate la distanță și pentru primirea de rezultate. JAX-RPC permite dezvoltatorilor Java să creeze servicii Web în care este încorporată funcționalitate RPC bazată pe XML în conformitate cu specificația SOAP 1.1.

După definirea și implementarea serviciului JAX-RPC, acesta este integrat într-un sistem de procese de execuție JAX-RPC pe partea de server. Integrarea se bazează pe tipul componentei utilizată pentru implementarea serviciului JAX-RPC. De exemplu, un serviciu bazat pe EJB este integrat într-un container EJB.

În timpul integrării unui serviciu JAX-RPC, instrumentul de integrare configurează una sau mai multe legături de protocole pentru serviciul JAX-RPC. O legătură asociază o definiție abstractă a serviciului cu un protocol bazat pe XML și cu protocolul de transport. Un exemplu de legătură este SOAP 1.1 peste HTTP.

Un client al serviciului Web poate utiliza un serviciu JAX-RPC invocând metode de la distanță pe un port al serviciului descris într-un document WSDL.

• .Net

Prin intermediul cadrului de lucru .Net oferit de Microsoft, utilizatorii pot avea acces la un serviciu Web implementând un ascultător al serviciului Web. În acest scop, un sistem trebuie să înțeleagă mesaje SOAP, să genereze răspunsuri SOAP, să furnizeze un contract WSDL pentru serviciul Web și să facă publicitate serviciului prin intermediul UDDI.

Dezvoltorii .Net care crează ascultători și consumatori de servicii Web bazate pe SOAP au posibilitatea să aleagă una din următoarele variante:

- utilizarea claselor de mesaje SOAP încorporate în .Net;
- crearea manuală a unui ascultător SOAP utilizând MSXML²⁰, ASP²¹, ISAPI²² etc.;
- utilizarea Microsoft SOAP Toolkit 2.0 în scopul creării unui ascultător de servicii Web care se conectează la interfața de afaceri implementată utilizând COM.

Microsoft SOAP Toolkit 2.0 oferă o componentă pe partea de client care permite unei aplicații să invoce operații ale serviciilor Web descrise într-un document WSDL.

5. Alegerea între platformele J2EE și .Net

Cunoscând abordările serviciilor Web pe platformele J2EE și .Net, un implementator trebuie să decidă ce versiune va implementa. Din punct de vedere pur tehnic, fiecare metodă are avantajele și dezavantajele sale.

Abordarea serviciilor Web pe platforma .Net are avantajul cheie că aceasta a fost proiectată special în acest scop, în timp ce platforma J2EE este readaptată prin adăugarea de API-uri suplimentare.

Unul dintre avantajele utilizării platformei J2EE este posibilitatea de alegere dintr-un număr mare de organizații implicate în vânzarea de API-uri pentru software-uri precreate (în majoritatea cazurilor servere

²⁰ MSXML - Microsoft Extensible Markup Language - Limbajul de marcare extins oferit de Microsoft

²¹ ASP - Active Server Pages - Pagini active de server

²² ISAPI - Internet Server API - Interfață programabilă a aplicației pentru server de Internet

de aplicații), din care fac parte numeroase proiecte cu sursă deschisă. În multe privințe, serverele de aplicații J2EE cu sursă deschisă sunt mai apropiate de standardul stabilit de Sun Microsystems, deoarece nu adaugă extensii proprietare pentru rezolvarea problemelor.

În cazul în care o organizație nu crează un sistem nou, alegerea în ceea ce privește platforma pentru implementarea de servicii Web este influențată de sistemul existent în momentul respectiv. Dacă o organizație are o echipă de programatori specialiști și un sistem de afaceri, atunci va dori să utilizeze sistemul existent bazat fie pe platforma J2EE, fie pe platforma .Net.

6. Glosare de termeni

6.1. Glosar de termeni referitori la servicii Web XML și la SOA

- **BPEL (Business Process Execution Language - Limbajul de execuție a proceselor de afaceri)**

BPEL este un limbaj de programare serializat în XML având drept obiectiv facilitarea programării în ansamblu. Conceptele de programare în ansamblu și programare la nivel de parte componentă se aplică în funcție de modul de scriere a tipului proceselor asincrone cu execuție pe termen lung, precum procesele de afaceri.

- **BPEL4WS (Business Process Execution Language for Web Services - Limbajul de execuție a proceselor de afaceri pentru servicii Web)**

BPEL4WS este o inițiativă a organizațiilor BEA Systems, IBM, Microsoft, SAP AG și Siebel Systems având drept obiectiv asigurarea interoperabilității pentru descrierea proceselor de afaceri bazate pe servicii Web și comunicația între acestea. BPEL4WS este o notație pentru specificarea comportamentului proceselor de afaceri bazate pe serviciile Web. BPEL4WS permite proceselor să exporte și să importe funcționalitate utilizând exclusiv interfețele serviciilor Web.

- **DTD (Document Type Definition - Definiția tipului de document)**

DTD este un limbaj pentru descrierea structurii documentelor XML. XML-ul standardizează modul în care sunt reprezentate datele. Pentru ca datele să fie utile, aplicațiile trebuie să aibă acces la descrierea structurii datelor. DTD oferă un mecanism pentru descrierea unui astfel de conținut.

- **SAML (Security Assertions Markup Language - Limbajul de marcare pentru aserțiunile de securitate)**

SAML este un cadru de lucru pentru schimbul de informații de autentificare și autorizare. Securitatea implică de obicei verificarea certificatelor de acreditare prezentate de o „parte” pentru autentificare și autorizare. SAML standardizează reprezentarea acestor certificate de acreditare într-un format XML denumit „aserțiuni”, ceea ce are drept rezultat creșterea interoperabilității între aplicații.

- **SOA (Service-oriented Architecture - Arhitectură orientată pe servicii)**

O arhitectură orientată pe servicii (SOA) este o colecție de servicii care comunică între ele. Comunicația poate fi efectuată fie prin simpla trimitere a datelor, fie prin implicarea a două sau mai multe servicii care coordonează o anumită activitate. Anterior apariției serviciilor Web, au existat alte arhitecturi orientate pe servicii, precum DCOM²³ și CORBA.

- **SOAP (Simple Object Access Protocol - Protocolul simplu de acces la obiecte)**

SOAP este un standard care definește structura la nivel de aplicație pentru mesaje. Pentru ca două aplicații să se integreze, ele trebuie să agreeze o structură explicită a mesajului. SOAP oferă o structură a mesajului la nivel de aplicație în scopul de a fi utilizată pentru un număr mare de protocoale de transport. Aplicațiile care folosesc SOAP pot schimba cu ușurință informații cu alte aplicații care folosesc SOAP, facilitând integrarea între sisteme.

- **UDDI (Universal Description, Discovery and Integration - Descrierea universală, descoperirea și integrarea)**

²³ DCOM - Distributed Component Object Model - Model de obiecte componente distribuite

UDDI este un protocol bazat pe XML care oferă un director distribuit permițând organizațiilor să-și publice informații (de exemplu, nume, produs, locație sau servicii Web oferite) pe Internet și să descopere alte servicii.

- **WSCI (Web Service Choreography Interface - Interfața de coreografie a serviciilor Web)**

WSCI este un limbaj de descriere a interfețelor bazat pe XML care tratează fluxul mesajelor schimbate de un serviciu Web ce participă în interacțiuni coreografiate cu alte servicii. WSCI nu abordează definiția și implementarea proceselor interne care conduc schimbul de mesaje, ci descrie comportamentul observabil al unui serviciu Web prin intermediul unei interfețe orientată pe flux de mesaje. Astfel, dezvoltatorii, arhitecții și instrumentele au posibilitatea să descrie și să compună o vedere globală a dinamicii schimbului de mesaje prin înțelegerea interacțiunilor cu serviciul Web.

- **WSDL (Web Services Definition Language - Limbajul de descriere a serviciilor Web)**

WSDL este un limbaj pentru descrierea modului de interfațare cu servicii bazate pe Web. Există mai multe standarde XML care oferă structura conținutului schimbat de aplicații. Un astfel de conținut poate fi trimis și primit prin intermediul unei diversități de protocoale de transport. Pentru a invoca un serviciu, o aplicație consumatoare trebuie să cunoască interfața serviciului, inclusiv modul de structurare a conținutului și protocolul de transport utilizat. WSDL descrie explicit această interfață într-un format standardizat, care poate fi citit de mașină și care permite consumarea prin intermediul instrumentelor.

- **WSDM (Web Services Distributed Management - Managementul distribuit al serviciilor Web)**

Comitetul tehnic al OASIS pentru WSDM definește două seturi de specificații: MUWS (Management Using Web Services - Managementul utilizând servicii Web) și MOWS (Management Of Web Services - Managementul serviciilor Web).

- **WS-I Basic Profile (Profilul de bază WS-I)**

WS-I Basic Profile este un set de standarde pe care organizațiile implicate în vânzarea de servicii Web trebuie să-l respecte pentru asigurarea interoperabilității cu produsele SOAP. În prezent, există o multitudine de specificații care au ca obiectiv promovarea integrării serviciilor Web în cadrul organizațiilor și între acestea. Aceste standarde abordează modul de descriere, descoperire și invocare a serviciilor Web de-a lungul unei rețele. Standardele existente evoluează și apar în permanență standarde noi. WS-I (Web Services Interoperability Organization - Organizația pentru interoperabilitatea serviciilor Web) are drept obiectiv accelerarea adoptării de standarde pentru serviciile Web prin furnizarea de profile și linii directoare de testare în scopul utilizării lor de către organizațiile implicate în vânzarea de servicii Web. Fiecare profil constă dintr-o listă de specificații cu numerele versiunilor, clarificări ale ambiguităților din specificații și modul de utilizare a caracteristicilor. WS-I Basic Profile indică modul de utilizare a specificațiilor: XML Schema 1.0, SOAP 1.1, WSDL 1.1 și UDDI 2.0 de către organizațiile implicate în vânzarea de servicii Web pentru ca produsele lor să fie interoperabile. Profilele viitoare vor aborda interoperabilitatea în domenii specifice, precum securitatea, transmiterea fiabilă a mesajelor, tranzacțiile și managementul proceselor de afaceri.

- **WS-Policy (Web Services Policy Framework - Cadrul de lucru pentru politica serviciilor Web)**

WS-Policy oferă o gramatică flexibilă și extensibilă pentru exprimarea capacităților, cerințelor și caracteristicilor generale ale entităților dintr-un sistem bazat pe servicii Web XML. WS-Policy definește cadrul de lucru și modelul pentru exprimarea acestor proprietăți drept politici.

- **WS-Security (Web Services Security - Securitatea serviciilor Web)**

WS-Security este un mecanism care permite încorporarea informațiilor de securitate în mesaje SOAP. Deși SOAP furnizează o tehnică flexibilă pentru structurarea mesajelor, nu abordează direct modul de securizare a acestor mesaje. WS-Security se bazează pe specificația SOAP pentru structurarea utilizării capacităților esențiale de securitate. Astfel, WS-Security utilizează jetoane binare pentru autentificare, semnături digitale pentru integritate și criptare la nivel de conținut pentru confidențialitate. Prin structurarea securității SOAP, WS-Security facilitează includerea elementelor de securitate în SOAP utilizând instrumentele și aplicațiile pentru întreprindere.

- **WS-Trust (Web Services Trust Language - Limbajul de încredere pentru serviciile Web)**

WS-Trust este un bloc de creare care este utilizat împreună cu alte protocoale pentru serviciile Web și protocoale specifice aplicațiilor în scopul adaptării unei game variate de modele de securitate. WS-Trust

este o extensie la WS-Security care furnizează metode pentru editarea, înnoirea și validarea jetoanelor de securitate, precum și moduri de stabilire a relațiilor de încredere între agenți / brokeri.

- **XML (Extensible Markup Language - Limbajul de marcare extins)**

XML este un limbaj de marcare pentru documente care conțin informații structurate. Limbajul de marcare este un mecanism care permite identificarea structurilor dintr-un document. Specificația XML definește un mod standard de adăugare a marcajelor la documente. XML a devenit standardul de facto pentru reprezentarea datelor.

- **XML Encryption (Criptarea XML)**

XML Encryption este un proces pentru criptarea și decriptarea unor părți ale documentului XML. Majoritatea schemelor actuale de criptare utilizează tehnici la nivel de transport care criptează un întreg flux de cereri și răspunsuri între un expeditor și un destinatar, asigurând faptul că intermediarii nu au posibilitatea să vizualizeze conținutul schimbat. Criptarea la nivel de conținut convertește fragmente ale documentelor în text cifrat ilizibil, în timp ce alte elemente rămân lizibile ca text în clar. XML Encryption permite controlul granular asupra elementelor vizibile sau private dintr-un mesaj.

- **XML Schema (Schema XML)**

XML Schema este un limbaj pentru descrierea structurii și aplicarea de constrângeri asupra conținutului documentelor XML. Utilizarea cu succes a XML-ului presupune existența unui mecanism pentru descrierea conținutului documentelor XML. XML Schema permite dezvoltatorilor să realizeze blocuri de creare care definesc conținutul, incluzând elementele, atributele și tipurile de date. Un document XML este considerat o instanțiere a unei scheme XML dacă este conform cu structura și constrângerile stabilite în schemă. În comparație cu DTD-ul, XML Schema asigură suportul pentru a gamă mai mare de capacități, cum sunt spațiile de nume și tipurile de date.

- **XML Signature (Semnătura XML)**

XML Signature este un mecanism care permite asigurarea originii și integrității documentelor XML. Semnăturile digitale oferă o metodă electronică pentru „timbrarea” unei identități în conținut. Destinatarul conținutului pot examina „timbrul” pentru a se asigura că expeditorul care a creat și trimis conținutul este cel așteptat. De asemenea, acest „timbru” oferă destinatarului siguranța că, pe durata tranzitului, conținutul a rămas intact și nemodificat. XML Signature standardizează procesul de semnare și de inserare a conținutului XML în documente XML, permițând destinatarilor să le verifice originea și integritatea.

6.2 Glosar de termeni referitori la tehnologii Java pentru business intelligence

- **API-ul JavaMail**

API-ul JavaMail asigură crearea și transmiterea mesajelor MIME²⁴ standard pe Internet și accesul la mesajele din depozitul standard de e-mail-uri. Platforma J2EE trebuie să asigure suportul pentru JavaMail.

- **APM (Application Programming Model - Modelul de programare a aplicațiilor)**

APM definește modul de utilizare și combinare a caracteristicilor platformei J2EE în scopul creării de soluții destinate diferitelor domenii ale aplicațiilor pentru întreprindere.

- **CORBA (Common Object Request Broker Architecture - Arhitectura agenților / brokerilor de cereri de obiecte comune)**

CORBA este un standard industrial elaborat de OMG²⁵, care ia în considerare comunicația dintre obiectele distribuite ce pot rula pe platforme diferite în cadrul unei rețele.

- **Componente**

În schema J2EE, dezvoltatorii de aplicații utilizează componente precum: EJB-uri, JB-uri și servleți, iar furnizorii de platforme J2EE utilizează containere și conectori. Astfel, dezvoltatorii de aplicații se pot concentra pe problemele de specialitate, precum business intelligence.

²⁴ MIME - Multipurpose Internet Mail Extensions - Extensii de poștă electronică pe Internet cu scopuri multiple

²⁵ OMG - Object Management Group - Grupul pentru managementul obiectelor

- **Componentă „bean” de entităţi**

Componenta EJB „bean” de entităţi persistă între sesiunile utilizatorilor şi de obicei reprezintă o bază de date sau un obiect de afaceri. O platformă J2EE trebuie să ofere suport pentru crearea şi persistenţa componentelor „beans” de entităţi.

- **Componentă „bean” de sesiune**

Componenta EJB „bean” de sesiune lucrează cu un singur client pe durata unei sesiuni şi este eliminată atunci când se încheie sesiunea.

- **Componentă „bean” bazată pe mesaje**

Componenta EJB „bean” bazată pe mesaje permite aplicaţiilor J2EE să prelucreze mesajele în mod asincron. De obicei, o componentă „bean” bazată pe mesaje acţionează ca un ascultător de mesaje JMS.

- **Conectori**

Conectorul este o interfaţă între componente J2EE şi sisteme informatice existente în întreprindere.

- **Containere**

Containerele furnizează pe partea de server mediul procesului de execuţie pentru EJB-uri, servleţi şi JSP-uri. Containerul administrează interacţiunile dintre o componentă şi mediul înconjurător. Pe partea de client, browser-ul Web este containerul care susţine pagini HTML²⁶ şi apleţi.

- **Descriptor al implementării**

O caracteristică esenţială a specificaţiei EJB 1.1 este utilizarea XML în scopul descrierii, într-un document al descriptorilor implementării, a modului în care un EJB sau o colecţie de EJB-uri cooperează cu alte componente.

- **EJB (Enterprise JavaBeans)**

Specificaţia Enterprise JavaBeans crează o arhitectură de componente pentru dezvoltarea de aplicaţii de server portabile. Enterprise JavaBeans constituie nucleul aplicaţiilor J2EE care se execută pe o platformă J2EE. Există trei tipuri de componente EJB: „beans” de sesiuni, „beans” de entităţi şi „beans” bazate pe mesaje.

- **IDL (Interface Definition Language - Limbajul de definire a interfeţei)**

IDL este un limbaj utilizat pentru descrierea obiectelor într-un mediu CORBA de obiecte distribuite. În J2EE, JavaIDL permite aplicaţiilor să acceseze obiectele CORBA utilizând protocolul IIOP.

- **IIOP (Internet Inter-ORB (Object Request Broker) Protocol – Protocolul Internet între agenţi / brokeri de cerere a obiectelor)**

IIOP este un standardul pentru asigurarea conectivităţii prin intermediul CORBA. Specificaţia J2EE 1.2 solicită obligatoriu asigurarea suportului pentru IIOP.

- **J2ME (Micro Edition of Java 2 – Microediţia Java 2)**

J2ME este un subset cu cerinţe minime în ceea ce priveşte memoria şi este destinat pentru a fi utilizat în produse oferite consumatorilor, cum sunt dispozitivele de tip „hand held”. Aplicaţiile J2ME sunt compatibile cu platformele J2SE şi J2EE.

- **J2SE (Standard Edition of Java 2 – Platforma Java 2 ediţia standard)**

J2SE este utilizată de obicei pentru crearea de clienţi, pagini JavaServer şi servleţi. O platformă J2SE de dezvoltare sau pentru procesul de execuţie nu trebuie să asigure suportul pentru Enterprise JavaBeans.

- **Java ARchive (Fişier JAR)**

Java ARchive este un standard pentru împachetarea fişierelor claselor Java şi a altor resurse. Enterprise JavaBeans sunt distribuite dezvoltatorilor de aplicaţii drept fişiere JAR. Formatul fişierului Java este utilizat, de asemenea, pentru fişierele Enterprise ARchive.

²⁶ HTML - Hypertext Mark-up Language - Limbaj de marcare hipertext

- **Java Community Process (Procesul comunității Java)**

Java Community Process este un proces creat de organizația Sun Microsystems în anul 1998 pentru a permite comunității de utilizatori Java să participe la dezvoltarea rapidă a noilor API Java.

- **JB (JavaBeans)**

API-ul JavaBeans este cadrul de lucru pentru componentele interfeței utilizator folosite pe partea de client a aplicațiilor J2EE. Crearea interfețelor pe partea de client având la bază standardul JavaBeans simplifică proiectarea interfeței și promovează dezvoltarea de elemente reutilizabile. Toate sistemele importante de dezvoltare Java asigură dezvoltarea rapidă a interfeței utilizând JavaBeans.

- **JDBC (Java DataBase Connectivity - Conectivitatea dintre Java și baze de date)**

API-ul JDBC furnizează acces la o gamă largă de baze de date și în prezent beneficiază de o susținere generală din partea organizațiilor implicate în vânzarea de baze de date. Platforma J2EE trebuie să asigure suportul pentru JDBC 2 pentru accesul la baze de date.

- **JMS (Java Message Service - Serviciul Java de mesaje)**

API-ul JMS asigură livrarea garantată a mesajelor între aplicații în mod asincron. Utilizând această API, există siguranța că nu se pierd informații vitale dacă un server este temporar indisponibil.

- **JNDI (Java Naming and Directory Interface - Interfața Java pentru acordare de nume și pentru directoare)**

JNDI furnizează o interfață unică pentru serviciile de numire și de directoare. Astfel, un EJB poate localiza toate resursele care îi sunt necesare prin intermediul unei interfețe consistente admisă de container.

- **JSP (JavaServer Pages - Paginile serverului Java)**

API-ul JavaServer Pages asigură încorporarea codului Java în contextul unei pagini Web. Acest fapt permite generarea dinamică a paginilor Web pe baza informațiilor furnizate de Enterprise JavaBeans. Nu este necesar ca un proiectant de pagini Web să știe să programeze în Java pentru a utiliza JSP-uri. Toate platformele J2EE trebuie să asigure suportul pentru JSP-uri.

- **Modul**

Modulul J2EE este constituit dintr-o colecție formată din una sau mai multe componente de același tip și un descriptor al implementării în XML.

- **MOF (Meta Object Facility - Facilitatea de metaobiecte)**

MOF reprezintă un set de interfețe IDL care definesc o arhitectură pentru crearea și partajarea metadatelor bogate din punct de vedere semantic.

- **OLAP (OnLine Analytical Processing - Prelucrare analitică online)**

OLAP este utilizată în cadrul aplicațiilor de business intelligence pentru analiza rapidă online a datelor extrase din baze masive de date.

- **OMG (Object Management Group - Grupul pentru managementul obiectelor)**

OMG este responsabil cu elaborarea de standarde în industria de prelucrare a informațiilor, precum CORBA și MOF.

- **Platforma J2EE (Java 2 Platform, Enterprise Edition - Platforma Java 2 ediția destinată întreprinderii)**

Platforma J2EE este un sistem de servere care susține întreaga gamă de containere și interfețe solicitate de specificație. O platformă poate rula pe orice hardware și se poate conecta la aplicații scrise într-o varietate de limbaje. O platformă poate susține conectarea la sisteme moștenite utilizând CORBA sau interfețe de tip Socket (soclu) de nivel inferior pentru ca orice aplicație de business intelligence care se execută pe o platformă J2EE să se poată executa pe toate platformele.

- **RMI (Remote Method Invocation - Invocarea de la distanță a metodei)**

RMI este API-ul Java standard care permite programelor să invoce metode în obiecte aflate la distanță, supuse restricțiilor de securitate.

- **Set J2EE de teste de compatibilitate**

Setul J2EE de teste de compatibilitate asigură conformitatea unui produs creat pe platforma J2EE cu standardul.

- **Servlet**

API-ul Java pentru servleți furnizează răspunsuri formate programelor care se execută în contextul unui server Web, răspund cererilor HTTP și generează HTML. Într-o aplicație J2EE, servleții și paginile JavaServer asigură conexiunea între clienți utilizând browsere Web și Enterprise JavaBeans și implementând logica de afaceri. Servleții înlocuiesc rapid scripturile CGI pe serverele Web datorită flexibilității lor mai mari și ușurinței programării. O platformă J2EE trebuie să asigure suportul pentru servleți.

- **SSL (Secure Socket Layer - Stratul soclurilor securizate)**

O platformă J2EE trebuie să asigure suportul pentru SSL care este o formă de comunicație Web sigură prin intermediul HTTPS²⁷.

²⁷ HTTPS - Hyper Text Transfer Protocol Secure sockets - Socluri sigure pentru protocolul de transfer hipertext