***Servicii Web (Web Services)***

**Profesor coordonator: Studenti:**

**Conf.dr.ing. Stefan Stancescu**

Matragociu Bogdan

Mirzac Adrian

442A

Cuprins

* *Matragociu Bogdan:*

1. Introducere
2. WSDL
3. SOAP

* *Mirzac Adrian:*

1. Securitate
2. Concluzii
3. Bibliografie
4. **Introducere**

Serviciile Web (Web Services) descriu o modalitate de acces la date si interactiunea cu programe ce ruleaza pe diferite platforme de operare în cadrul retelelor publice si de întreprindere. Spre deosebire de retelele extranet tipice, ce necesita interfete puternic integrate între membrii comunicarii, scopul serviciilor Web este acela de a oferi o singura interfata comuna care sa permita calculatoarelor sa ruleze programe, sa partajeze date si sa acceseze servicii diverse. Bazate pe un limbaj comun (XML) si un protocol comun de transport (HTTP), serviciile Web actioneaza ca un intermediar între cele doua entitati ce doresc sa comunice între ele. Serviciile Web sunt bazate pe limbaje si pr otocoale specifice (în afara de XML si HTTP), printre care, cele mai cunoscute sunt: Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) si Universal Discovery, Description and Integration (UDDI). SOAP este practic un mecanism de transport pentru mesajele XML. Un plic SOAP este utilizat pentru a transporta mesaje XML ce includ date sau programe de-a lungul unei retele, de regula prin HTTP. UDDI este un registru bazat pe XML ce permite furnizorilor sa prezinte serviciile Web pe Internet. WSDL (care este bazat tot pe XML) reprezinta o modalitatea de descriere de conexiune a clientilor (dintr-o perspectiva software) la furnizorii de servicii Web. Împre una cu XML si HTTP, aceste protocoale reprezinta fundamentul pe care se sprijina serviciile Web, permitând diverselor entitati sa caute si sa prezinte servicii, sa apeleze proceduri, sa ruleze programe si sa întoarca rezultate. [[1]](#footnote-1)

1. **WSDL**

* WSDL vine de la Web Services Description Language
* WSDL este scris în XML
* WSDL este un document XML
* WSDL este folosit pentru a descrie servicii Web
* WSDL este, de asemenea, utilizat pentru a localiza serviciile Web
* WSDL este o recomandare W3C

WSDL este un document scris în XML. Documentul descrie un serviciu Web. El specifică locația serviciului și a operațiunilor (sau metodele) expunese de serviciu.

**Structura documentului WSDL**

Un document WSDL descrie un serviciu web folosind aceste elemente majore:

* <types> Un container de definiții de tipuri de date utilizate de către serviciul web
* <Message> O definiție a datelelor care trebuie comunicate
* <portType> Un set de operațiuni susținute de către unul sau mai multe endpoint-uri
* <binding> Un protocol și formatul datelor pentru un anumit tip de port

Structura principală a unui document WSDL arată astfel:

<definitions>

<types>

data type definitions........

</types>

<message>

definition of the data being communicated....

</message>

<portType>

set of operations......

</portType>

<binding>

protocol and data format specification....

</binding>

</definitions>

Un document WSDL poate conține, de asemenea, alte elemente, cum ar fi elemente de extensie, și un element de serviciu care face posibilă gruparea mai multor servicii Web într-un document WSDL singular. 3

**Mesaje WSDL**

Elementul <mesaj> definește elementele unei operațiuni.

Fiecare mesaj poate consta din una sau mai multe părți. Părțile pot fi comparate cu parametrii unui apel de funcție într-un limbaj de programare tradițională.

**Tipuri WSDL**

Elementul <types> definește tipurile de date care sunt utilizate de către serviciul web.

Pentru neutralitatea maximă a platformei, WSDL utilizează sintaxa XML Schema pentru a defini tipurile de date.

## WSDL Bindings

## Elementul <binding> definește formatul de date și protocolul pentru fiecare tip de port.

## 3.SOAP

## SOAP vine de la Simple Object Access Protocol

## SOAP este un protocol de comunicare

## SOAP este folosit pentru comunicarea între aplicații

## SOAP este un format pentru trimiterea de mesaje

## SOAP comunică prin Internet

## SOAP este independent de platforma

## SOAP este independent de limbaj

## SOAP este bazat pe XML

## SOAP este simplu și extensibil

## SOAP vă permite sa treceti de firewall-uri

## SOAP este o recomandare W3C

Este important pentru dezvoltarea de aplicații, pentru a permite comunicarea între programele de pe Internet.

Aplicațiile de astăzi comunica prin apeluri de procedură la distanță (RPC) între obiecte cum ar fi DCOM și CORBA, dar HTTP nu a fost proiectat pentru acest lucru. RPC reprezintă o compatibilitate și o problemă de securitate, firewall-uri și servere proxy vor bloca în mod normal acest tip de trafic.

O cale mai bună de a comunica între aplicații este HTTP, deoarece HTTP este suportat de toate browserele de Internet și serverele. SOAP a fost creat pentru a realiza acest lucru.

SOAP oferă o modalitate de a comunica între aplicațiile care rulează pe sisteme de operare diferite, cu diferite tehnologii si limbaje de programare.

SOAP a devenit o recomandare W3C pe 24 Iunie 2003. 3

Un mesaj SOAP este un document XML obișnuit care conține următoarele elemente:

* Un element Envelope care identifică documentul XML ca un mesaj SOAP
* Un element Header care conține informații de antet
* Un element de Body care conține informații de apel și răspuns
* Un element Fault care conține erori și informații de stare

Câteva reguli importante de sintaxă:

* Un mesaj SOAP trebuie codificat cu ajutorul XML
* Un mesaj SOAP trebuie să utilizeze namespace-ul SOAP Envelope
* Un mesaj SOAP trebuie să utilizeze namespace-ul SOAP Encoding
* Un mesaj SOAP nu trebuie să conțină o referință DTD
* Un mesaj SOAP nu trebuie să conțină instrucțiuni de procesare XML

**Structura unui mesaj SOAP:**

<?xml version="1.0"?>  
<soap:Envelope  
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"  
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">  
  
<soap:Header>  
...  
</soap:Header>  
  
<soap:Body>  
...  
  <soap:Fault>  
  ...  
  </soap:Fault>  
</soap:Body>  
  
</soap:Envelope>

Namespace-ul xmlns:soap trebuie intotdeauna sa aiba valoarea : "http://www.w3.org/2001/12/soap-envelope".

Dacă se utilizează un namespace diferit, aplicația generează o eroare și aruncată înapoi mesajul.

Elementul SOAP Body conține mesajul SOAP real destinat pentru obiectivul final al mesajului.

Elementul optional SOAP Encoding Fault este folosit pentru a indica mesaje de eroare.

În cazul în care un element Fault este prezent, el trebuie să apară ca un element copil al elementului Body. Un element Fault poate să apară o singură dată într-un mesaj SOAP.

**Un exemplu SOAP**

În exemplul de mai jos, o cerere GetStockPrice este trimisa la un server. Cererea are un parametru StockName, și un parametru Price care va fi returnat în răspuns. Namespace-ul pentru aceasta funcție este definit în "http://www.example.org/stock".3

### **Cerere SOAP:**

POST /InStock HTTP/1.1  
Host: www.example.org  
Content-Type: application/soap+xml; charset=utf-8  
Content-Length: nnn  
  
<?xml version="1.0"?>  
<soap:Envelope  
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"  
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">  
  
<soap:Body xmlns:m="http://www.example.org/stock">  
  <m:GetStockPrice>  
    <m:StockName>IBM</m:StockName>  
  </m:GetStockPrice>  
</soap:Body>  
  
</soap:Envelope>

### **Raspuns SOAP :**

HTTP/1.1 200 OK  
Content-Type: application/soap+xml; charset=utf-8  
Content-Length: nnn  
  
<?xml version="1.0"?>  
<soap:Envelope  
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"  
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">  
  
<soap:Body xmlns:m="http://www.example.org/stock">  
  <m:GetStockPriceResponse>  
    <m:Price>34.5</m:Price>  
  </m:GetStockPriceResponse>  
</soap:Body>  
  
</soap:Envelope>

## Securitate

## Credem ca nu mai este nevoie de o pledoarie in favoarea securitatii datelor, mai ales cand este vorba de date sensibile care sunt transmise intre cele doua entitati ce comunica. In general serviciile web asigura autentificarea, autorizarea, confidentialitatea si integritatea datelor. Autentificarea reprezinta procedeul prin care se stabileste identitatea uneia dintre partile implicate in comunicare. Dup ace procesul de autentificare a luat sfarsit , autorizarea stabileste care sunt drepturile de actiune pentru acea entitate. Confidentialitatea implica faptul ca mesajele transmise din partile implicate in comunicare sunt transparente pentru alte entitati, fie intentionat sau nu. Cel mai utilizat mechanism pentru asigurarea confidentialitatii este criptarea. Integritatea presupune ca datele transmise nu au fost alterate , ele sunt primate de catre destinatar exact asa cum au fost trimise de expeditor- acest lucru se face prin intermediul semnaturilor digitale.

**Criptarea XML**

Criptarea XML (XML Encryption) reprezinta o specificatie dezvoltata de Consortiul World Wide Web care descrie procesul criptarii datelor si reprezentarea acestora în format XML. Aplicatiile care respecta aceasta specificatie trebuie sa asigure criptare simetrica si asimetrica, precum si algoritmi uzuali de criptare ca AES (Advanced Encryption Standard) sau Triple DES. Specificatia XML Encryption precizeaza ca Implementarile ofera suport pentru standard uzuale de chei de criptare, precum Diffie Hellman. De asemenea, în cadrul acesteia se defines mecanisme pentru criptarea întregului mesaj sau doar a unei parti a acestuia. Acest lucru permite un acces flexibil la datele din cadrul mesajelor XML. Într-o tranzactie cu o carte de credit, spre exemplu, o aplicatie poate necesita acces la numele clientului, numarul cartii de credit, limita de credit si nu acces la alte informatii, cum ar fi cele legate de alte achizitii. Un al tert implicat în tranzactie poate sa aiba acces doar la numele clientului si la limita de credit, nimic în plus.

**Semnatura XML**

Semnatura XML este o specificatie elaborata de W3C si IETF si reprezinta o metoda de reprezentare a semnaturilor digitale folosind sintaxa XML. De asemenea, tot aici se descriu procedurile de calcul si de verificare a acestor semnaturi. O semnatura digitala asigura integritatea si autentificarea mesajului. Aceasta este creata prin rularea unui algoritm hash pe un set specific de date (document) si criptarea mesajului rezultat cu cheia privata de expeditorului. Destinatarul ruleaza acelasi algoritm hash asupra documentului ce conduce la un alt rezultat al mesajului. Destinatarul utilizeaza acum cheia publica a expedi- torului pentru a decripta primul mesaj rezultat. Daca cele doua rezultate coincid, atunci expeditorul este asigurat ca datele au fost receptionate corect.

O componenta importanta în cadrul specificatiei Semnaturii XML este aducerea la forma canonica. Metodele clasice de analiza morfologica pot duce la pierderea de informatii neesentiale într-un mesaj XML sau poate duce la schimbarea prezentarii acestuia chiar daca sensul ramâne acelasi. Acest lucru reprezinta o problema pentru semnaturile digitale deoarece, daca analizorul morfologic modifica chiar si cu un bit mesajul XML, rezultatul calculat la destinatar va fi diferit de cel de la expeditor si astfel mesajul nu va fi autentificat. Aducerea la forma canonica creaza o forma denormalizata a documentului. Într-o tranzactie expeditorul creeaza si semneaza o forma canonica a documentului, apoi se trimite atât documentul cât si varianta semnata. Destinatarul aduce la forma canonica documentul si apoi compara rezultatul obtinut cu forma semnata.

**WS-Security**

Specificatia Web Services Security dezvoltata tot de Organizatia OASIS se ocupa cu schimbul securizat de mesaje SOAP. Se permit aplicatiilor sa ataseze o semnatura digitala si headere de criptare mesajelor SOAP pentru integritate si confidentialitate. Atât headerul cât si corpul mesajului SOAP pot fi criptate sau semnate, precum si alte entitati atasate mesajului SOAP. Specificatia descrie mecanisme de atasare a unor jetoane de securitate mesajelor SOAP. Aceste jetoane de securitate includ o varietate de informatii de securitate, de la nume de utilizatori pâna la certificate digitale X.509 sau tichete Kerberos. Jetoanele de securitate sunt folosite pentru a valida pretentiile exprimate de expeditor. De exemplu, un mesaj SOAP poate pretinde sa aiba acces la anumite resurse de calcul. Capetele comunicarii pot aplica politici de securitate pentru a cere jetoane speciale de securitate pentru validarea unor astfel de pretentii. WS Security coopereaza cu specificatiile XML Signature si XML Encryption. De asemenea, ofera suport pentru utilizarea modelelor de securitate standard precum PKI, Kerberos, SSL (Secure Socktes Layer)/TLS (Transport Layer Security).Pe lânga aceste specificatii mai exista si altele în faza de proiectare, printre care WS Policy (descrie politicile de securitate dintre corespondenti), WS Trust (face referire la modele de încredere pentru o interoperare securizata) si WS Privacy (permite furnizorilor de servicii Web sa-si stabileasca preferinte si practici personale).[[2]](#footnote-2)

**Principiile securitatii serviciilor web**

**1: Aplicarea apararii in adancime**:

Se folosesc mai multi portari pentru a tine atacatorii la distanta. Apararea in adancime inseamna ca nu trebuie sa te bazezi pe un singur strat de securitate, sau consideri ca unul dintre straturile tale ar putea sa fie compromis.

**2: Verificarea la poarta:**

Se implementeaza la prima poarta metodele de autentificare si autorizare a accesului clientilor.

**3: Compartimentizarea**: Izolarea si constrangerea problemei. Se aplica principiul de separare a ,,problemelor”. Firewall-urile, conturile si codurile cu privilegii reduse , reprezinta un exemplu de compartimentizare.

**4: Strapungerea controlata a sistemului:**

Daca o aplicatie cedeaza, nu lasa datele sensibile ,,la vedere” . Utilizatorului ii sunt afisate mesaje de eroare in care nu se dau foarte multe detalii legate de problemele interne ale sistemului.

**5: Reducerea suprafetei de atac:**

Pentru a implementa acest principiu se dezactiveaza sau se elimina serviciile, protocoalele si functionalitatile neutilizate.

**6: Securizarea celor mai vulnerabile legaturi**:

Cea mai slaba legatura, daca este strapunsa poate duce la aparitia unui lant de probleme legate de securitatea sistemului.[[3]](#footnote-3)

## Concluzii

## Serviciile web sunt unele dintre elementele-cheie ale programarii Web. Acestea sunt elemente de software extrem de versatile, care au într-adevăr potențialul de a deschide o nouă eră în software: era interoperabilitatii. Serviciile web pot fi utilizate în mod eficient pentru înființarea tranzacțiilor business-to-business. Acestea sunt expuse funcționalitatii software pentru clienți și pentru integrarea platformelor eterogene.

## Servicii web sunt bazate exclusiv pe protocoale generale deschise acceptate de Internet. Acesta este principalul lor atu, dar, de asemenea, prezinta si slabiciuni. Puterea, care permite interoperabilitatea cu adevărat, vine la prețul de viteză scăzuta, care este grav afectata de lățimea de bandă. Servicii de web nu au o gama larga de utilizabilitate dar reprezintă cu siguranță o categorie de agenți software din ce in ce mai cautata.

## Abstractizarea deferitelor tipuri de sercivii web este posibila, desi mecanismul trebuie sa fie foarte extensibil.

## Tema stiintifica a serviciilor web a fost aplicata cu succes pe un scenariu din lumea reala.

## Integrarea serviciilor web in Drupal a fost realizata in functie de efortul programarii.

1. **Bibliografie**

* Securitatea serviciilor web - Revista Informatica economica Conf. dr. Razvan Daniel Zota [1]
* <http://msdn.microsoft.com/en-us/library/ff648318.aspx> [2]
* [www.w3schools.com](http://www.w3schools.com) [3]
* [www.wikipedia.com](http://www.wikipedia.com) [4]

1. Securitatea serviciilor web-Revista Informatica economica Conf. dr. Razvan Daniel Zota [↑](#footnote-ref-1)
2. Securitatea serviciilor web-Revista Informatica economica Conf. dr. Razvan Daniel Zota [↑](#footnote-ref-2)
3. http://msdn.microsoft.com/en-us/library/ff648318.aspx [↑](#footnote-ref-3)