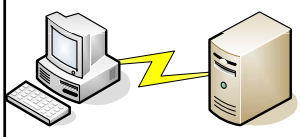
Hálószolgáltatások (Web Services)

Webalkalmazás-dedikált alkalmazás

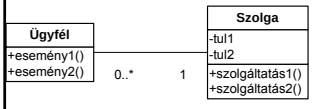


Célkiszolgáló-célügyfél (dedikált kiszolgáló/ügyfél)

- Mindkettő önálló alkalmazás

- Csak egymással tudnak kommunikálni

-Pl: Oracle server + SQLDeveloper



Web kiszolgáló-ügyfél

-Vékony ügyfél: web böngésző, mint GUI + dinamikus HTML lap

-Kiszolgáló: Web-kiszolgáló + Web alkalmazás tároló (Web-container) + Web-alkalmazás

- HTTP (+XML) protokollra épített kommunikáció

-Platform- (gép, oprendszer) -független

Statikus←→dinamikus HTML, XML

- HTML (Hipertext jelölőnyelv, H-Markup Language) lapok részei: statikus←→dinamikus, egyes részek tartalma
- Generalized Markup Language (GML) 1960-as években szabvány
- Standard Generalized Markup Language (SGML) 1986-ban ISO szabvány
- Dinamikus részek: XML→HTML

Extensive Markup Language (XML)

- Tim Bray: IBM+USA egyetemek 1986.
- XML1.0 1998. XML 1.1. 2004.
- Faszerkezetek leírására használt szabvány, a következő egyszerű szabálykészlettel:
  - Elemformátum: `<elemnév attribútum1="érték1" attribútum2="érték2" ...> tartalom...</elemnév>`, ahol „tartalom” további szövegeket és/vagy elemeket jelent
  - Egymásbaágyazott szerkezetek: az `<elem1><elem2></elem1></elem2>` szerkezet tiltott
- A fenti általános nyelvtant alkalmazva milyen konkrét XML nyelvtanokat engedünk meg? →Nyelvtanleírások
- Egy konkrét alkalmazása HTML/XHTML
- Legfontosabb használata: szoftverek közötti adatcserére, ha összetett, faszerkezetű adatelemekről van szó

```
<?xml version="1.0" encoding="UTF-8"?>
<Recept név="kenyér" elk_idő="5 perc„
  sütés_idő="3 óra">
  <cím>Egyszerű kenyér</cím>
  <összetevő mennyiség="3" egység="csésze">
    Liszt</összetevő>
  <összetevő mennyiség="10 egység="dekagramm">
    Élesztő</összetevő>
  <összetevő mennyiség="1.5" egység="csésze">
    Meleg víz</összetevő>
  <összetevő mennyiség="1" egység="teáskanál">
    Só</összetevő>
  <Utasítások>
    <lépés>Keverj össze minden összetevőt,
      aztán jól gyúrd össze!</lépés>
    <lépés>Fedd le ruhával és hagyd pihenni
      egy óráig egy meleg szobában!</lépés>
    <lépés>Gyúrd össze újra, helyezd bele
      egy bádog edénybe, aztán
      süsd meg a sütőben!</lépés>
  </Utasítások>
</Recept>
```

XML Nyelvtanleírások

- Document Type Definition (DTD): kicsit merev, egyetlen globális névtérrel, különböző nyelvtanok nem összeilleszthetők
- XML-sémák: primitív típusok, típusszűkítés és –bővítés, reguláris kifejezések használata, típusok közt metszetek és uniók stb. Elemtartalmak elemsorrendje, kötelező/lehetséges előfordulások, számosságok, alternatívák és sorozatok. Típusöröklődés és behelyettesítés.
- Relax NG: kicsi, sokoldalú, elegáns. Kevés hivatalos támogatás áll mögötte.

XML Faműveletek

- XPath nyelv: Fa-csomópontok szűrésére és kiválasztására használható
- XQuery nyelv: XML alapú lekérdező nyelv
- XML Structured Transformation Language (XSLT): az XML fákból más XML fákat létrehozó átalakító nyelv. Tartalmazza az XPath résznyelvet. A fákra minták illeszthetők, amelyből újabb részfák építhetők fel. Maga is XML nyelven van írva.
- Cascading Style Sheets (CSS): HTML és XML dokumentumok formattálását megadó leírás

XML programozása

- Simpe API for XML (SAX): eseményvezérelt API, amely szekvenciális elérési lehetőséget ad az XML adatokhoz
- Document Object Model (DOM): az XML dokumentumot objektumhálóba alakítja át.

Az XML használata

- Konkrét XML nyelvtan létrehozása
- Nyelvtanelemeket létrehozó (export), ill. azokat értelmező beolvasó (import) programok létrehozása.
- <catalog xmlns= "urn:oasis:names:tc:entity:xmlns:xml:catalog">
  <public publicId="-//OASIS//DTD XML DocBook V4.1.2//EN" uri="docbook/xml/docbookx.dtd"/>
  <system systemId="urn:x-oasis:docbook-xml-v4.1.2" uri="docbook/xml/docbookx.dtd"/>
  </catalog>

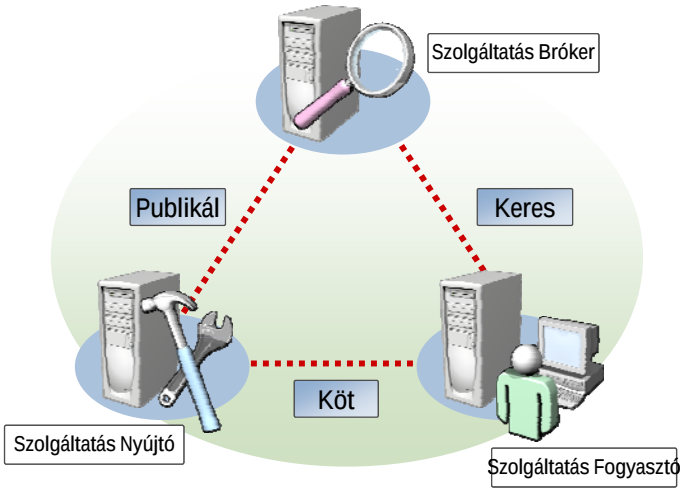
Az XML kezelése, műveletek

- XML editor: szerkesztésre
- XML, CSS, Schema, DTD vagy XSLT dokumentumok létrehozása
- XML ellenőrzése:
  - alapszintaxis alapján
  - DTD vagy XML séma alapján

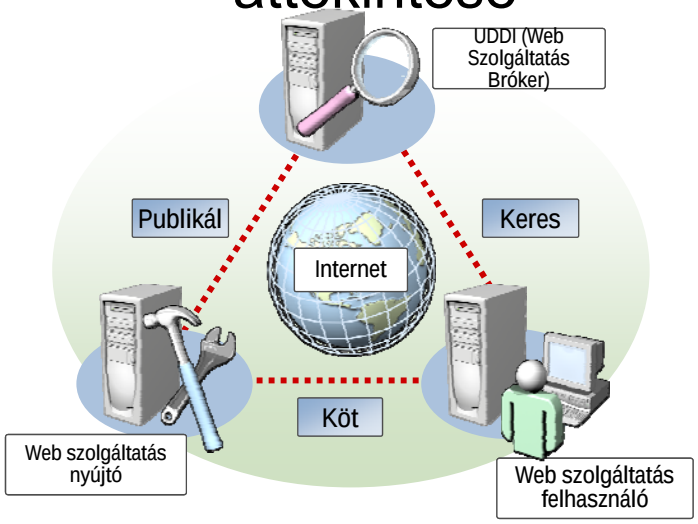
PI: Az XML használata a NetBeans-ben

- Project.xml file: a NetBeans project felépítését írja le
- Build-impl.xml: projekt „build” információk
- Persistence.xml: a projekt perzisztens objektumainak leírása
- Stb. stb.

Szolgáltatás Orientált Architektúra



Web szolgáltatás architektúra áttekintése



Web szolgáltatás szolgáltató

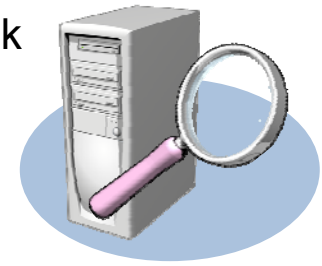
- Például:
  - Web szerverek
  - .NET Common Language Runtime

Web szolgáltatás fogyasztó

- Minimális funkcionalitás
- Szolgáltatás keresés
- Proxy-k
- Aszinkron hívások

Web szolgáltatás bróker

- Együttműködés a brókerek és a szolgáltatók között
- Együttműködés a brókerek és a fogyasztók között
- UDDI tárolók



Webszolgáltatások leírása

- A webszolgáltatás leírható: egy szolgáltatáshoz tartozik egy interface, és létezik egy ember számára is olvasható leírása (WSDL).
- A webszolgáltatás felkutatható: a létrehozott szolgáltatás publikálható (UDDI).
  - a felkutatás létrehozható centralizált, ill. decentralizált módon egyaránt. Centralizált esetben egy nyilvántartó rendszer felhasználásával.

Service Oriented Architecture (SOA)

- WEB szolgáltatások: olyan kiszolgáló komponensek, amelyek /valamiféle dedikált hálózati protokoll helyett/ WEB-kiszolgálókon/ általuk értelmezett protokollon keresztül/ állnak rendelkezésre ➔ csak a protokoll/ adatformátum/ rögzített, a megvalósítás nem➔platform függetlenség
  - RESTful Web Services: Universal Resource Identifiers+HTTP. Minden egyéb információ a HTML-be van beágyazva
  - Java API for XML based Remote Procedure Calls (JAX-RPC)… webszolgáltatások és ügyfelek megvalósítása J2EE1.4 környezetben (1998. UserLand Software Frontier)
    - A legegyszerűbb wevszolgáltatás
    - XML és HTTP alapú
    - Nincsenek objektumok
  - Java API for XML WebServices (JAX-WS). Java EE5-től, a JAX-RPC továbbfejlesztése

Service Oriented Architecture (SOA)

- A JAX-RPC és a JAX-WS a következő specifikációkon alapul:
  - Simple Object Access Protocol: webszolgáltatás meghívásának és az adatok átadásának/ átvételének protokollja (SOAP 1.2 specifikáció)
  - Web Service Definition Language (WSDL): webszolgáltatások külső felületének leírása (WSDL 1.1 specifikáció)
  - Universal Discovery, Description and Integration (UDDI): Webszolgáltatásokat tartalmazó regisztrációs adatbázisok szabványa (beleérve a WSDL leírások és a futó szolgáltatások helyét).
  - Java Architecture for XML Binding (JAXB): A JAX-RPC-vel szemben a JAX-WS a JAXB szabványon alapul (JAXB2.0/2.1, ill. JAX-WS2.0/2.1 párhuzamos szabványok)

Webszolgáltatások keresése

- Webszolgáltatás kereső honlap: http://seekda.com

```
<definitions name="Calculator”
  targetNamespace=
  "http://www.inf.bme.hu/~dobe/gsoap/calc.cgi/
  Calculator.wsdl">
  <types>
  <schema targetNamespace=http://tempuri.org/ns.xsd
    elementFormDefault="unqualified„
    attributeFormDefault="unqualified">
  <import namespace=
    "http://schemas.xmlsoap.org/soap/encoding/"/>
  <complexType name="operandusok">
    <sequence>
      <element name="a" type="xsd:double" minOccurs="1„
        maxOccurs="1"/>
      <element name="b" type="xsd:double" minOccurs="1„
        maxOccurs="1"/>
    </sequence>
  </complexType>
  <!-- operation request element -->
  <element name="add">
    <complexType>
      <sequence>
        <element name="ops" type="ns:operandusok„
          minOccurs="0" maxOccurs="1" nillable="true"/>
      </sequence>
```

Webszolgáltatások készítése és használata NetBeans rendszerrel

- A NetBeans az XML szerkesztés helyett vizuális (GUI szerkesztő eszközt bocsát rendelkezésre)

Nyilvános Hálószolgáltatás: MNB árfolyam elérése

1. Hozzunk létre Java alkalmazást
2. A Java projekthez hozzunk létre új webszolgáltatás-hivatkozást (WebServiceReference, URI:http://www.mnb.hu/arfolyamok.asmx?wsdl
3. A program main eljárásába másoljuk be a `getExchangeRates` webszolgáltatás-hívást, és módosítsuk a paraméterezést értelemszerűen.

Feladat I: készítsünk számológép Háló-kiszolgálót

1. Indítsuk el a GlassFish Hálókiszolgálót!
2. Háló-kiszolgáló létrehozása: File/New/Project/Web/WebApplication név: SzamlaloWSAlk
3. Hálószolgáltatás létrehozása: SzamlaloWSAlk/JobbGomb/New/WebService név:SzamlaloWS, package:org.me.szamlalo
   1. Adjunk hozzá egy új eljárást!
   2. Vegyünk fel paramétereket! (add(i:int,j:int):int)
   3. Source nézetben valósítsuk meg az eljárást!
4. Telepítés: SzamlaloWSAlk/JobbGomb/Deploy
5. Tesztelés: SzamlaloWS/JobbGomb/Test Web Service

Feladat II: Egyszerű ügyfél a számológéphez

1. Java alkalmazás létrehozása:
   File/New/Project/Java/JavaApplication név:
   SzamlaloWSUgyfel, CreateMainClass:yes
2. Hálószolgáltatáshoz ügyfél létrehozása:
   SzamlaloWSUgyfel/JobbGomb/New/WebServiceClient
   Browse:SzamlaloWSAlk
3. Java-hivatkozás a szolgáltatásra: Szamlalo
   SzamlaloWSAlk/JobbGomb/Web Service References/…
   húzzuk át a használni kívánt (add) műveletet a main Java kód belsejébe.
4. Tesztelés: írjuk át a kódot úgy, hogy a program parancssor paramétereiből vegye a hívás paramétereit!
5. Bővítsük a szolgáltatásokat a többi alapművelettel, a parancssorból olvassuk be a műveletkódot, és teszteljük!