

---



# Szolgáltatások és alkalmazások (VITMM131)



WWW, szolgáltatás integráció, Web szolgáltatások



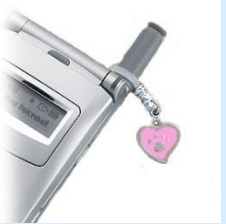
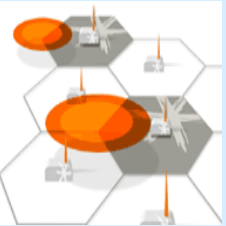
Vidács Attila  
Távközlési és Médiainformatikai Tanszék (TMIT)  
I.E.348, T:19-25, [vidacs@tmit.bme.hu](mailto:vidacs@tmit.bme.hu)

---

# Tartalom

---

- Internet-alapú szolgáltatások
- Web szolgáltatások (Web services)
  - Elosztott rendszerek
  - Szolgáltatás-orientált architektúra  
- Service Oriented Architecture (SOA)
  - Web szolgáltatás – definíciók, komponensek



# Internet szolgáltatási modell (ism.)

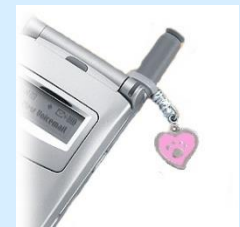
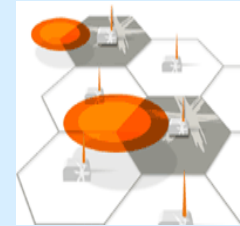
- Megjegyzés: *A megkülönböztetés kliens és szerver között kizárólag a szolgáltatásra és nem az Internetre vonatkozik!*
  - Az Internet hálózatán a kliens és szerver egyaránt egy hálózati állomás adott IP címmel.
  - Az IP címek használatosak az adatcsomagok továbbítására a forrástól a célállomásig (*routing*)
  
- Következmény: *Az útválasztás (routing) tekinthető az egyedüli szolgáltatásnak amit az Internet nyújt. Az egyes szolgáltatók ezt a szolgáltatást használva nyújtják a saját értéknövelt szolgáltatásaikat.*

Más szavakkal: az Internet *útválasztási* képessége elkülönül a szolgáltatásoktól, melyek az Internetet *használják*.



# World Wide Web mint szolgáltatási platform

- A Web-et és a Web böngészőket megelőzően a szolgáltatóknak ki kellett fejleszteniük és menedzselniük a saját vég-vég szolgáltatásaik erőforrásait.
- Az 1990-es évek-beli bevezetésétől a Web vált a szabványos szolgáltatási platformmá az Interneten.
  - A Web egy univerzális szolgáltatási protokollt nyújt a HTTP (**Hypertext Transfer Protocol**) képében;
  - és egy *kliens alkalmazást* a Web böngésző szerepében.
- Következmények:
  - A Web egy gyors piacra lépési időt (*time-to-market*) biztosít az új szolgáltatásoknak, és
  - drámaian csökkenti a végfelhasználók „tanulási idejét” egy konzisztens felhasználói interfész biztosításával a szolgáltatásokhoz.



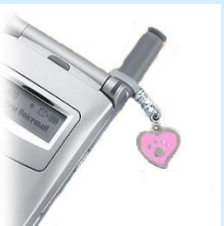
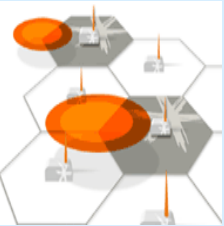
# Szolgáltatás integráció

- A nagyszabású szolgáltatások gyakran számos, *egymással interakcióban álló* szolgáltatást magában foglalnak.
  - Pl., egy e-kereskedelmi szolgáltatás használhat egy autentikációs biztonsági szolgáltatást a felhasználók azonosításához.
- Az együttműködő szolgáltatások valószínűsíthetően *különböző szolgáltatótól* származnak.
- **A különböző szolgáltatások együttműködésének biztosítását nevezük **szolgáltatás integrációnak**.**
- Az integráció megvalósításának egyik kulcs technológiája az **interfész** koncepció alkalmazása.
  - *Az interfész egy adott szolgáltatási komponens által kiejánlott **függvényhívások** vagy **metódus hívások** egy halmazára vonatkozik (pl. Web szerver, adatbázis, ...)*



# Szolgáltatás integráció - interfészek

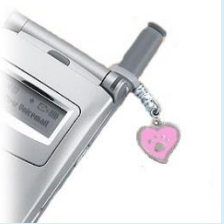
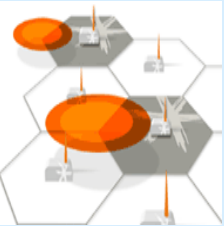
- Egy **metódus hívás** definíciója az alábbiakból áll:
  - a *neve*,
  - *be- és kimenő paraméterek* rendezett listája,
  - az *adattípusaik*, valamint
  - a *visszatérési érték* és *adattípus*.
- Egy **interfész** tekinthető metódus deklarációk egy halmazának (amelyek *nem* metódus definíciók!).
- Egy interfész **kiszolgálója** (szerver) az interfész metódusokat *implementáló* objektumok egy halmaza.
- Egy interfész **kliense** az interfész metódusokat *meghívó* objektumok összessége
- Egy ORB (**Object Request Broker**) kezeli az üzenetváltást egy interfész kliens és szerver oldala között.



# Tartalom

---

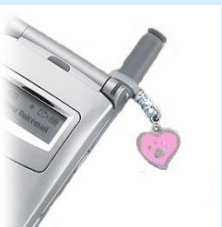
- Internet-alapú szolgáltatások
- Web szolgáltatások (Web services)
  - Elosztott rendszerek
  - Szolgáltatás-orientált architektúra  
- Service Oriented Architecture (SOA)
  - Web szolgáltatás – definíciók, komponensek



# Web szolgáltatások bevezető

---


- **Web szolgáltatások** (**Web Services**) a Web-en publikáltak/találhatóak meg és használhatóak.
- Web szolgáltatások...
  - *alkalmazás komponensek;*
  - *nyílt protokollok segítségével kommunikálnak;*
  - *ön-leíróak (self-describing) és kompaktak (self-contained);*
  - *felfedezhetőek (pl., UDDI használatával– lásd később);*
  - *használhatóak más alkalmazások által;*
  - *XML alapúak.*
- Bővebben: „Web- és e-szolgáltatások” VITMM132 (BME-TMIT, Magyar Gábor)





# W3C

---

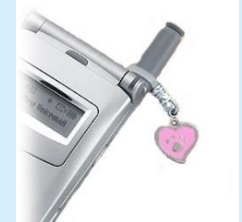
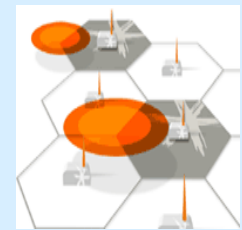
- 
- <http://www.w3c.org/>
  - **W3C** 1994-ben alakult a Web teljes „kiaknázására” általános protokollok kidolgozásával, amelyek elősegítik a fejlődést és biztosítják az együttműködő-képességet (interoperability).

*„The W3C **mission** is to lead the World Wide Web to its full potential by developing protocols and guidelines that ensure the long-term growth of the Web. Below we discuss important aspects of this mission, all of which further W3C's vision of One Web.”*

- W3C (**World Wide Web Consortium**)...
    - **1994. októberében** jött létre;
    - **Tim Berners-Lee** (MIT) vezetésével;
    - tagsági szervezet (Member Organization);
    - **Web szabványosításával** foglalkozik;
    - **WWW szabványok** (W3C ajánlások – recommendations);
-

---

# Web szolgáltatás - Példa



# Web Service – example use case

---



□ Web service use case: **Travel reservation**

□ A **travel agent** company wants to offer to people the ability to book complete vacation packages

■ plane/train/bus tickets, hotels, car rental, etc.



□ **Service providers** (airlines, bus companies, hotel chains, etc) are providing Web services to query their offerings and perform reservations.



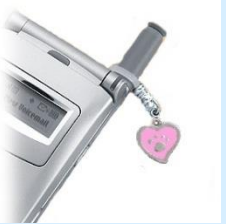
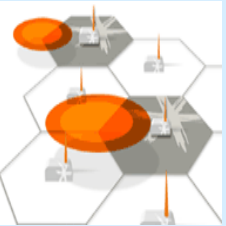
□ **Credit card companies** are also providing services to guarantee payments made by consumers.

---

# Web Service – example use case

---

- Due to the **loosely coupled-nature of Web services**, the travel agent **doesn't need to have a priori agreements** with service providers or credit card companies.
- This allows the **travel agent** to have access to more services, offering more options to its customers,
- the **credit card companies** to offer their services broadly and therefore make their customers happy, and
- the **service providers** can offer their services broadly and easily and therefore generating more business for themselves.
  - „Internet prices”



# Stakeholders / interests

---



**Travel agent:**

- provides a system to provide the user with options for his/her vacation;
- earns money by charging fees for each package bought.



**Service providers:**

- sell their services by making them available widely.



**Credit card company:**

- enable customers to use their credit cards in a very large number of cases;
- make profit with each money transaction.

**Consumer:**

- book vacation easily by choosing among a large variety of offers.
-

# Actors and goals

---



- **Consumer:**

- best combination of services and prices for his/her needs.

- **Travel agent:**

- customer satisfaction, sell packages.




- **Service providers:**

- sell services.

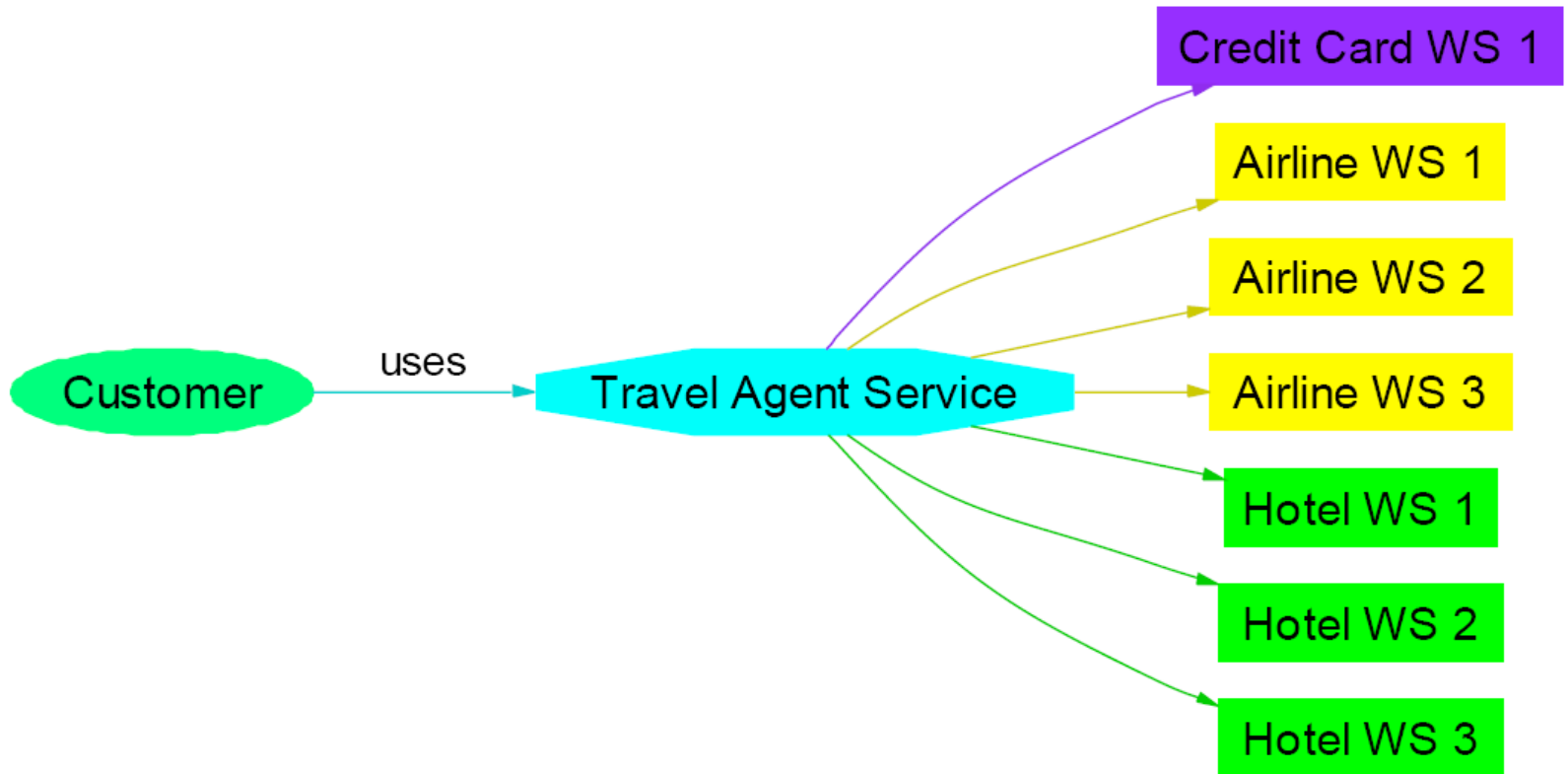
- **Credit company:**

- qualify buyer, do the payment.

- 
- Note: **Only the user** in the scenario **is a human being!**  
The travel agent service, airline, hotel and payment services that the travel agent service is interacting with, are machines.
-

# Usage scenario

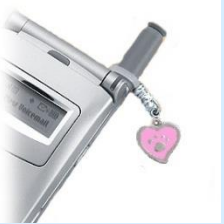
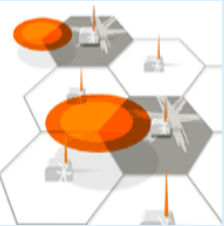
- Use case: how a user would make a reservation for a vacation package (flight and hotel room).



# Assumptions

---

- Assumption: All the services are **using common concepts** (e.g. flight, economy class, room, etc).
  - I.e., For the travel agent service to understand the airline services and to be able to send meaningful information to them, a **travel industry ontology** („szakzsargon”) needs to exist and be used by the Web services taking part in this scenario.
  
- An **ontology**...
  - ...is a **formal description** of a set of concepts and their relationships to each other.
  - ... **defines a standard vocabulary** that can be used to communicate those concepts.

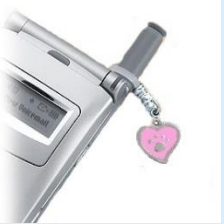
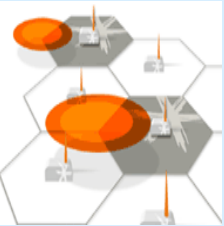




# Assumptions

---

- Note: Some additional technology is needed:
  - **context maintenance**
  - **reliability**
    - ...in order to make money, each step needs to happen.
  - **trust mechanisms**
    - ...for the services to do business with each other.
  - description of **orchestration of services**
    - ...if a reservation of a flight involves interacting with a couple of Web services, the airline would document in a machine readable way **how to interact** with the two single services in order to get the desired result,
    - including **how to handle errors** in the process fails before the operation is completed.
  - ...



# TASK 1


---



□ **TASK 1:** User requests availabilities about some travel dates



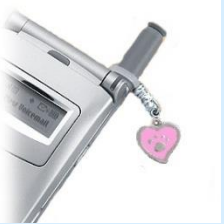
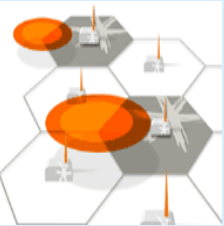
□ **Goal / Context**

- The **user** gets the location of a travel agent service via an unspecified way (search engine, URI in an email, service directory, etc).
    - E.g., [http://e-travelmanagement2.amadeus.com/login/HRG\\_BME](http://e-travelmanagement2.amadeus.com/login/HRG_BME)
  - The **user** provides a destination and some dates to the travel agent service.
  - The **travel agent** service inquires airlines about deals and presents them to the user.
- 

# TASK 1 steps

---

- Step 1: The user is presented with a form to fill in order to provide the travel agent service with details about dates of his/her travel and the destination.
- Step 2: The user submits the information to the service in order to get a list of flights corresponding to his/her schedule.
- Step 3: The travel agent service finds a list of airlines.
- Step 4: For each airline found:
  - The travel agent service requests a description of how to communicate with the service found.
  - The travel agent service requests a list of flights accommodating the user.
- Step 5: The travel agent service presents the results of the queries to the user letting him choose the best option.



# TASK 1 technologies / requirements

---



- Discovery technology:

- used by the travel agent service to find the airlines services.



- Description language:

- used by the airlines to describe their query services to the travel agent service.



- Response to queries:

- XML documents that the travel agent service processes and merge together.

- Ontologies:

- the data coming from different airline services and expressed with different XML vocabularies needs some semantics to be merged in a meaningful way.
-

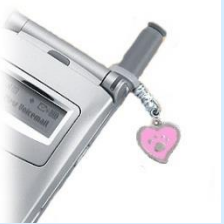
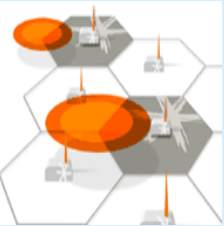
## TASK 2:

---

□ **TASK 2:** User chooses flight and looks for hotels

□ **Goal / Context**

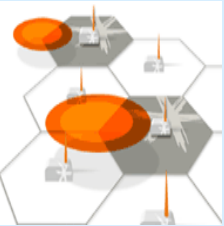
- The user has been presented with options for flights to go to his/her destination.
- The user chooses a preferred flight.
- The service puts the seats on hold, and goes on with proposing lodging options to the user.



## TASK 2 steps

---

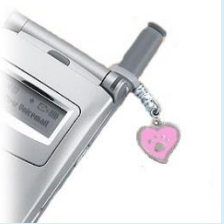
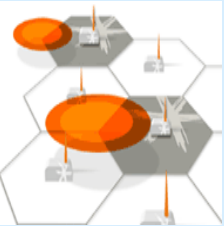
- Step 1: The user communicates his/her choice for the flight.
- Step 2: The travel agent service requests the chosen airline to put the flight on hold:
  - The travel agent service requests a description of how to put a seat on hold to the airline service.
  - The travel agent service sends the request accordingly.
- Step 3: The airline returns a confirmation number with an expiry date.
- Step 4: The travel agent service finds a list of hotels.
- Step 5: For each hotel found:
  - The travel agent service requests a description of how to communicate with the service found.
  - The travel agent service requests accommodation options for the period.



## TASK 2 steps (cont'd)

---

- Step 6: The travel agent service looks for payment services available, and builds a list of options for the user.
- Step 7: The travel agent service presents the results of the queries to the user letting him choose the best option, along with the payment options offered.



## Task 2: technologies/requirements

---



□ Description language:

- used by the airlines to describe their services to put tickets on hold to the travel agent service,
- by the hotels to describe their query services to the travel agent service.

□ Discovery technology:

- used by the travel agent service to find the hotels services.

□ Ontologies:

- the data coming from different accommodation services and expressed with different XML vocabularies needs some *semantics* to be merged in a meaningful way.
-



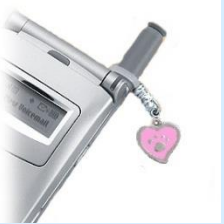
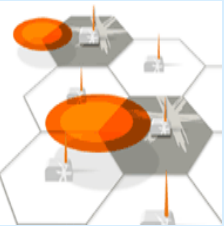
# TASK 3

---

□ **TASK 3:** User books hotel room and flight


□ **Goal / Context**

- The user has been presented with options for hotels to go to his/her destination and a means of payment.
- The user chooses a hotel option.
- The travel agent service contacts a bank for payment authorization.
- The service books the hotel and confirms the flight, using the payment authorization from the bank.



# TASK 3 steps

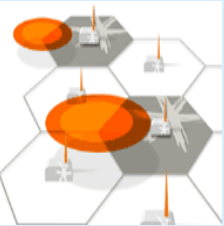
---

- 
- Step 1: The user communicates his/her accommodation choice to the travel agent service.
  - Step 2: The travel agent service contacts the bank service that the user chose to confirm payment:
    - The travel agent service requests a description of how to guarantee payment of the total amount.
    - The travel agent service send the request accordingly.
    - The response indicates success with an authorization number, signed by the payment authority.
  - Step 3: The travel agent service books the hotel room:
    - The travel agent service requests a description of how to book a room to the chosen hotel service.
    - The travel agent service sends a request in order to find out how to cancel the reservation should a problem occur later in the process.
    - The travel agent service sends the request accordingly, communicating the payment service chosen and the signed authorization number from this service.
-

## TASK 3 steps (cont'd)

---

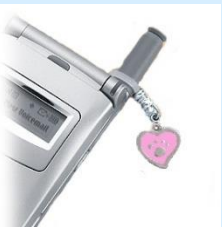
- Step 4: The travel agent service confirms the flight reservation:
  - The travel agent service requests a description of how to buy a ticket on hold to the airline service.
  - The travel agent service sends a request in order to find out how to cancel the reservation should a problem occur later in the process.
  - The travel agent service sends the request accordingly, communicating the payment service chosen and the signed authorization number from this service.
- Step 5: The travel agent service charges a fee to the user:
  - The travel agent service requests a description of how to request payment to the payment service.
  - The travel agent service sends the request accordingly, along with the authorization number signed by the payment service.
- Step 6: The service provides the user with various confirmation numbers and wishes the user a good vacation.



# TASK 3 technologies / requirements

---

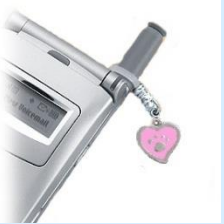
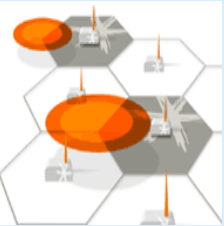
- Service description technology:
  - used by the payment authority to describe its confirmation service,
  - by the hotel to describe its room booking service, and
  - by the airline to describe its service to buy tickets by confirming seats on hold.
- Authentication technology:
  - used by the payment authority to sign the payment authorization to be trusted by the hotel service, the airline service and the travel agent service.
- Encryption technology:
  - used by the payment service and the travel agent service to communicate the user's payment information confidentially.
- Ontologies:
  - the payment confirmation needs to be used in a way meaningful to the travel service, hotel and airline services; in other words, **the output of one service needs to be used as the input to other services** that might use different vocabularies.



## Example - remarks

---

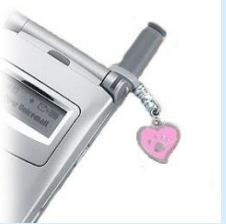
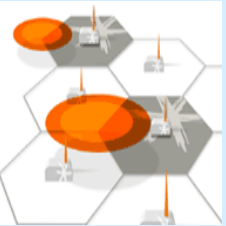
- This scenario illustrates how a program, the **travel agent service**, can **interact dynamically** with airline services, hotel services, **without a priori knowledge** of them or of the way they work.
- Thanks to the ontologies used, **the program can adapt** to variations of formats that an airline service might be using and adapt to the introduction of new products.
- However, there is a limit to what the travel agent service can understand.
  - For example, it is likely to be able to understand the introduction of a new class of tickets
    - E.g., class Z tickets can only be bought more than 60 days in advance and with a valid international student identification
  - The travel agent service will need to implement the extra logic to make it understand this new type of restriction, including validating the student identification.



# Tartalom

---

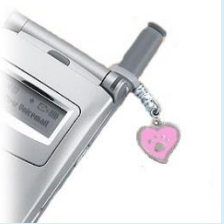
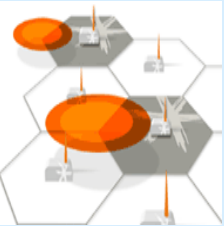
- Internet-alapú szolgáltatások
- Web szolgáltatások (Web services)
  - **Elosztott rendszerek**
  - **Szolgáltatás-orientált architektúra  
- Service Oriented Architecture (SOA)**
  - Web szolgáltatás – definíciók, komponensek



# A Web mint elosztott rendszer

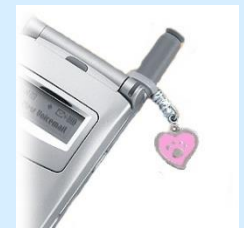
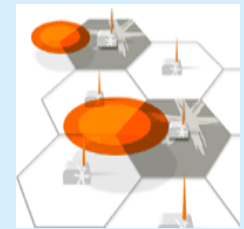
---

- Egy **elosztott rendszer** *különféle* szoftver ágensekből áll, amelyeknek *együtt kell működniük* egy adott feladat ellátásában.
- Az **ügynök** (**agent – ágens**) különböző számítási környezetben működhet, ezért hardver/szoftver protokollok segítségével kell kommunikálnak a hálózaton.
- Az elosztott rendszerek **architektúrális kihívásai**:
  - a mögöttes hálózat *késleltetése és megbízhatatlansága*,
  - *közös osztott memória hiánya*,
  - *részleges meghibásodások*,
  - távoli erőforrások konkurens hozzáférésének kihívásai,
  - *inkompatibilis frissítésekből* adódó sérülékenység.



# Szolgáltatás központú architektúra (SOA)

- A **szolgáltatás-központú architektúra** (SOA – **Service Oriented Architecture**) az *elosztott rendszer architektúra* egy formája a következő jellemzőkkel:
  - (**Logikai nézet**): A szolgáltatás egy elvont, logikai nézete a tényleges programoknak, adatbázisoknak, stb., által megfogalmazva, hogy „mit csinál” az adott szolgáltatás.
  - (**Üzenet központúság**): A szolgáltatás a szolgáltatást nyújtó és az azt igénybe vevő ügynökök **üzenetcsereje** által definiált, nem pedig az ügynökök jellemzői alapján. (Más szavakkal: egy SOA-ban *nem* szükséges ismerni, hogyan valósítunk meg egy szolgáltatást nyújtó ügynököt.)
  - (**Leírás központú**): Egy szolgáltatás gépek által is feldolgozható meta-adatokkal **írható le**.
  - (**Platform semleges**): Az üzenetek platform semleges, szabványosított formában, *interfészek*en keresztül kerülnek kézbesítésre. (Az XML a legnyilvánvalóbb formátum.)

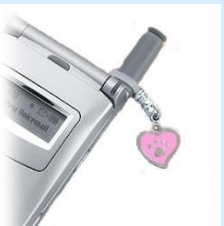
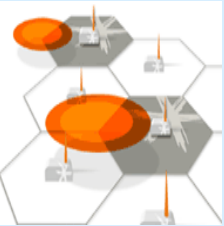




# Web szolgáltatások és SOA

---

- A **Web szolgáltatás technológiák** lehetséges jelöltek a SOA implementálására.
- SOA és a Web szolgáltatások a legmegfelelőbbek olyan alkalmazások/szolgáltatások számára, ahol...
  - Az **Interneten** kell működniük (ahol a sebesség és megbízhatóság nem garantálható);
  - az elosztott rendszer komponensei **különböző platformokon,** más-más gyártók eszközein futnak.
    - Pl., Java virtuális gép kliensek, Microsoft hoszt szerver.



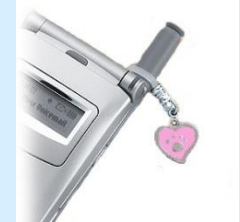
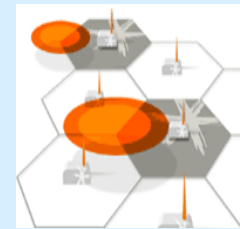
# Mi a Web szolgáltatás?

---

## □ Definíció:

- *A Web szolgáltatás egy hálózati **gép-gép együttműködés** támogatására tervezett szoftver rendszer.*
- *Egy automatikusan feldolgozható leírással (konkrétan **WSDL**) megadott interfésszel rendelkezik.*
- *Más rendszerek a leírásban megadott módon **SOAP** üzenetekkel működnek együtt a Web szolgáltatással, tipikusan HTTP átvitelrel XML-t használva.*

- ## □ Megjegyzés: A „Web szolgáltatások” nem összekeverendők a Web-en elérhető szolgáltatásokkal, amelyeket leginkább Web-alapú szolgáltatásoknak neveznek!



# Ügynökök és szolgáltatások

- A **Web szolgáltatás** egy absztrakt fogalom, amelyet egy konkrét **ügynöknek** kell implementálnia.
- Az **ügynök**...
  - egy konkrét szoftver és/vagy hardver elem, amely
  - *üzeneteket küld és fogad.*
- A **szolgáltatás**...
  - az **erőforrás**, amely
  - *a nyújtott funkcionalitás egy halmazával jellemzett.*
- *Pl., Egy adott Web szolgáltatást az egyik nap egy adott ágens implementál, míg a következő nap egy (másik programnyelven megírt) különböző ágens. Az ágens lecserélhető, de a web szolgáltatás ugyanaz marad.*



# Igénybevevő és nyújtó

- Egy Web szolgáltatás *célja* adott *funkcionalitás nyújtása*.
- A **szolgáltatást nyújtó entitás** (**provider entity**) (személy vagy szervezet) *biztosít* egy megfelelő ágenst amely egy adott *szolgáltatást implementál*.
- A **szolgáltatást kérő entitás** (**requester entity**) (személy vagy szervezet) egy Web szolgáltatást szeretne *használni*.
- A kérő entitás egy **kérő ágenst** (**requester agent**) használ az *üzenetek cseréjéhez* a **nyújtó ágenssel** (**provider agent**).
- Megjegyzés: A szakirodalomban a „szolgáltató” (**service provider**) kifejezést egyaránt használják a szolgáltatást nyújtó entitás (pl. személy) és/vagy a szolgáltatást implementáló ágens (pl. szoftver) megnevezésére.



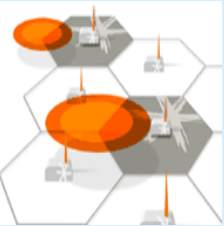
# Tartalom

---

- Web szolgáltatások (Web services)
  - Elosztott rendszerek
  - Szolgáltatás-orientált architektúra  
- Service Oriented Architecture (SOA)

- **Web szolgáltatások technológiái**

- XML, HTTP, WSDL, SOAP, UDDI
- Áttérés nyílt rendszerekre
- Nyílt szolgáltatás hozzáférés (OSA – Open Service Access)
- Alkalmazás programozói interfész (API – Application Programming Interface (API))



# Szolgáltatás leírás

- A *kérő entitás* és a *nyújtó entitás* **meg kell egyezzenek** az üzenetváltás „**mechanikájáról**” és „**szemantikájáról**”.
- Az üzenetváltás **mechanikája** a **Web szolgáltatás leíróban** (WSD – **Web Service Description**) dokumentált.
- A WSD egy...
  - gépek által feldolgozható *specifikációja* a Web szolgáltatás *interfészének*,
  - **WSDL**-ben írva (**Web Service Description Language**).
  - Definiálja az *üzenet formátumokat, adattípusokat, átviteli protokollokat*.
  - Megad egy vagy több *hálózati elérhetőséget* ahol a szolgáltató ügynök meghívható..



# Szemantika

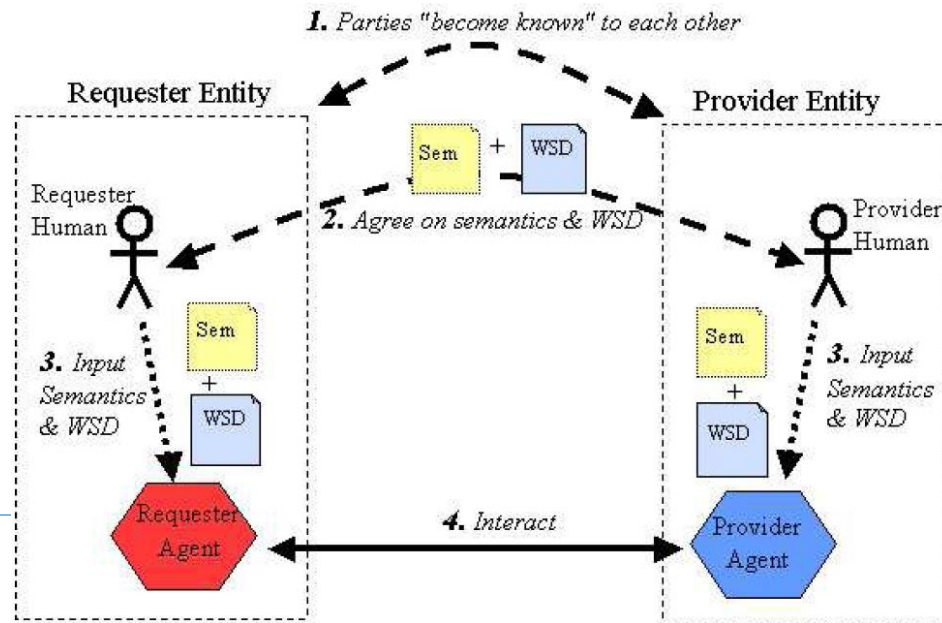
---

- A **szemantika** a „szerződés” a szolgáltatást kérő entitás és a szolgáltatást nyújtó entitás között, az interakció *célját* és *következményeit* tekintve.
- Lehet explicit vagy implicit, szóbeli vagy írásbeli, jogi vagy informális.



# Web szolgáltatás használata

- Általánosságban a következő lépések szükségesek:
  1. A szolgáltatást kérő és a szolgáltatást nyújtó entitások **„ismertté válnak** egymás előtt”;
  2. **Megegyeznek** az ügynökök a szolgáltatás leírásában és az interakciót szabályozó szemantikában.;
  3. A szolgáltatás leírás és szemantika **megvalósul** a nyújtó és kérő ügynökök közreműködésével;
  4. A kérő és nyújtó ügynökök **üzeneteket váltanak**.

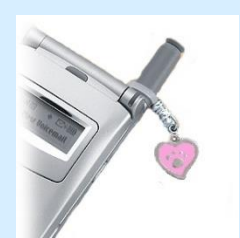
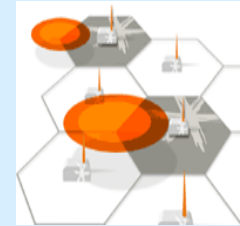




# Web szolgáltatás technológiái

---

- A Web szolgáltatások kapcsán gyakran idézett **technológiák** a következők:
  - **XML** (EXtensible Markup Language)
    - *adatok struktúrálása és tárolása, fókuszban magával az adattal.*
  - **HTML** (HyperText Markup Language)
    - *adatok megjelenítése, fókuszban az adat kinézetével.*
  - **SOAP** (Simple Object Access Protocol)
    - *specifikálja az üzenetváltások formátumát.*
  - **WSDL** (Web Services Description Language)
    - *leírja a Web szolgáltatás által elfogadott kérés üzeneteket, a visszaadott válasz üzeneteket, valamint elérhetőségének helyét az Interneten..*
  - **UDDI** (Universal Description, Discovery and Integration)
    - *Web szolgáltatások hirdetésére és keresésére szolgál nyilvános szolgáltatás tárházakban.*



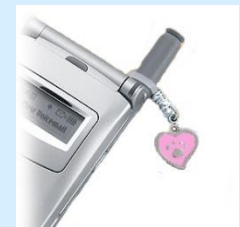
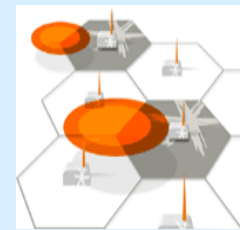
# XML

## □ EXTensible Markup Language (XML)

- W3C ajánlás;
- nem *csinál* semmit;
- adatok *struktúrálására, tárolására és átvitelére* való;
- *tisztán információ tag-ekbe csomagolva*;
- egyszerű szöveg;
- mindenhol(!) megtalálható.

## □ Példa: Egy feljegyzés Tove-tól Janinak, XML-ként tárolva:

```
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```



# HTML

---

HTML egy nyelv weboldalak leírásához.

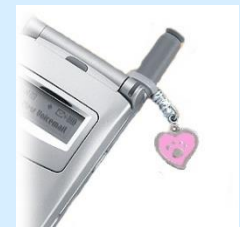
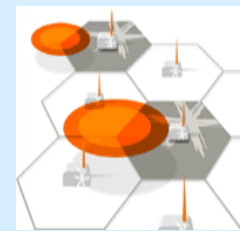
□ **HTML** (**Hyper Text Markup Language**)...

- *nem* egy programnyelv,
- leíró (markup) nyelv;
- leíró tag-ek halmaza weboldalak leírásához.

□ **HTML dokumentumok...**

- HTML tag-eket és egyszerű szöveget tartalmaz;
- weboldalaknak is hívják.

□ Egy Web-böngésző (pl. IE, Chrome, ...) feladata HTML dokumentumok olvasása és megjelenítése weboldalként.





# SOAP

---

- 
- **SOAP** (**Simple Object Access Protocol**) egy egyszerű XML-alapú protokoll az alkalmazások számára információcseréhez HTTP felett.

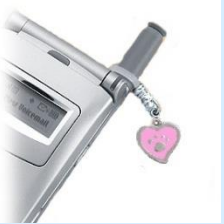
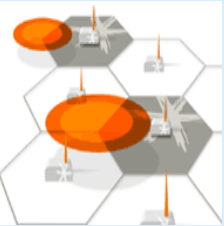
(Vagy egyszerűbben: egy protokoll Web szolgáltatások eléréséhez.)

- 
- 
- SOAP...
    - kommunikációs protokoll;
    - XML alapú;
    - Interneten kommunikál;
    - egyszerű és kiterjeszthető;
    - nyelvfüggetlen;
    - platformfüggetlen;
    - W3C ajánlás;
    - átjut a tűzfalakon.
-

# Miért SOAP?

---

- Napjaink alkalmazásai távoli eljárás hívások (**Remote Procedure Calls** – RPC) segítségével kommunikálnak az Interneten, de a HTTP-t nem erre tervezték.
- RPC kompatibilitási és biztonsági problémát jelent; tűzfalak és proxy szerverek alapvetően blokkolják ezt a fajta forgalmat.
- HTTP felett kommunikálni jobb, mert a HTTP-t támogatja az összes Internet böngésző és szerver. **SOAP-ot ezzel a céllal tervezték.**



# SOAP üzenet váza

```
<?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>
```



# SOAP és HTTP kapcsolata

---

- Egy **SOAP metódus** egy HTTP kérés/válasz amely megfelel a SOAP kódolási előírásoknak.
- **HTTP + XML = SOAP**
- Egy SOAP kérés lehet egy HTTP POST vagy egy HTTP GET kérés.



# SOAP példa

- Példa: *GetStockPrice* kérés a szervernek. A kérésben van egy *StockName* paraméter, és egy *Price* paraméter a várt visszatérési érték.

- A SOAP kérés:

```
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">
```

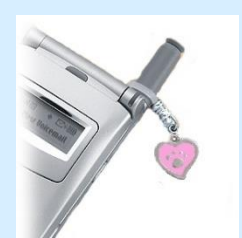
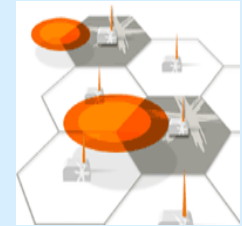
```
    <m:GetStockPrice>
```

```
      <m:StockName>IBM</m:StockName>
```

```
    </m:GetStockPrice>
```

```
  </soap:Body>
```

```
</soap:Envelope>
```





# SOAP példa (folyt.)

## □ Egy SOAP válasz:

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">
```

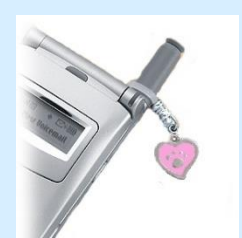
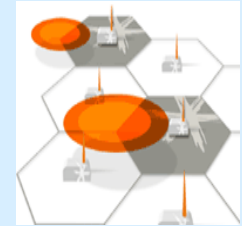
```
    <m:GetStockPriceResponse>
```

```
      <m:Price>34.5</m:Price>
```

```
    </m:GetStockPriceResponse>
```

```
  </soap:Body>
```

```
</soap:Envelope>
```



# WSDL

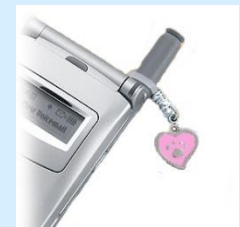
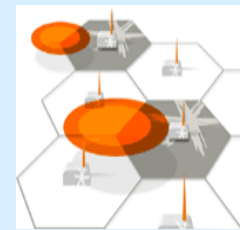
---

- **WSDL** (**Web Services Description Language – Web szolgáltatás leíró nyelv**) egy XML-alapú nyelv Web szolgáltatások leírására és a hozzáférhetőség megadására.

- WDSL...

- W3C ajánlás;
- Web szolgáltatások leírása;
- Web szolgáltatás elérhetőségének megadása;
- XML-alapú.

- Egy **WSDL dokumentum** egy egyszerű XML dokumentum.



# WSDL dokumentum struktúra

---

- Egy WSDL dokumentum fő struktúrája:

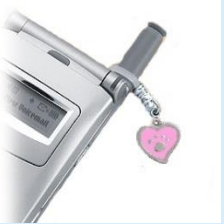
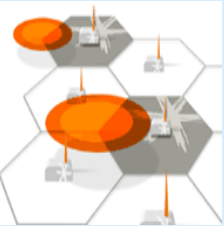
```
<definitions>
  <types>
    definition of types.....
  </types>

  <message>
    definition of a message....
  </message>

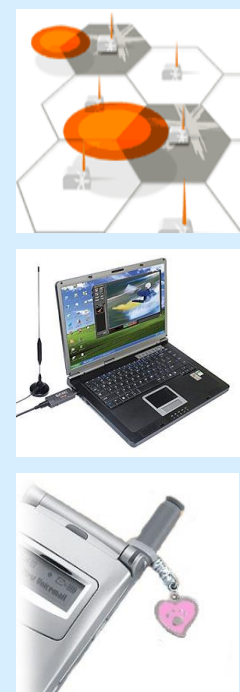
  <portType>
    definition of a port.....
  </portType>

  <binding>
    definition of a binding....
  </binding>

</definitions>
```



# WSDL példa



```
<message name="getTermRequest">  
  <part name="term" type="xs:string"/>  
</message>
```

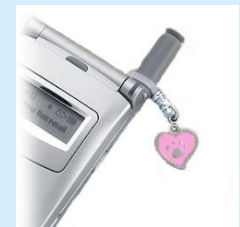
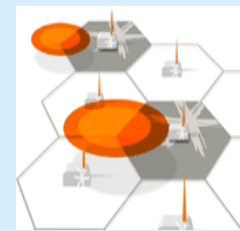
```
<message name="getTermResponse">  
  <part name="value" type="xs:string"/>  
</message>
```

```
<portType name="glossaryTerms">  
  <operation name="getTerm">  
    <input message="getTermRequest"/>  
    <output message="getTermResponse"/>  
  </operation>  
</portType>
```

- ❑ A **<portType>** elem definiálja a "glossaryTerms"-t mint egy **port** nevét, és a "getTerm"-et mint egy **művelet** nevét.
- ❑ A "getTerm" műveletnek van egy "getTermRequest" nevű **input üzenete**, és egy "getTermResponse", **output üzenete**.
- ❑ A **<message>** elemek definiálják az összes üzenet **részeit** (parts) és a hozzárendelt adattípusokat.

# UDDI

- **Universal Description, Discovery and Integration** (UDDI – **univerzális leírás, felderítés és integrálás**) egy **telefonkönyv szolgáltatás** (úm. „sárga oldalak”, „regiszter”) ahol Web szolgáltatások regisztrálhatók és kereshetők.
- UDDI...
  - egy telefonkönyv információ tárolására Web szolgáltatásokról;
  - telefonkönyv Web szolgáltatás WSDL segítségével leírt *interfészekről*;
  - SOAP segítségével kommunikál.
- UDDI támogatók
  - UDDI egy közös ipari erőfeszítés az összes vezető platform és szoftvergyártó (pl. Dell, Fujitsu, HP, Hitachi, IBM, Intel, Microsoft, Oracle, SAP, and Sun) és a piaci operátorok és e-biznisz vezetők közösségének részvételével.
  - Az UDDI közösségnek több mint 220 cég tagja.

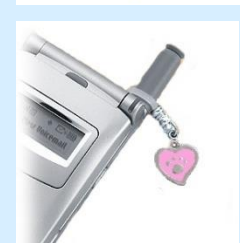
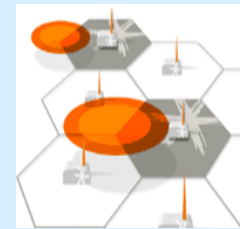


# UDDI „use case”

---

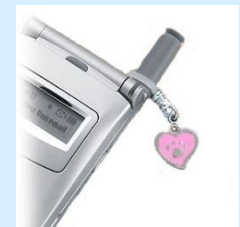
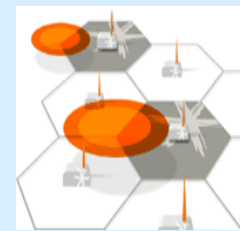
Példa UDDI alkalmazásra:

- 1. UDDI szabvány publikálása repülőjegyek árazására és foglalására.*
- 2. Légitársaságok regisztrálhatják a szolgáltatásaikat egy UDDI telefonkönyvbe.*
- 3. Utazási irodák kikereshetik az UDDI telefonkönyvben a légitársaságok foglalási rendszerének interfészeit.*
- 4. Ha az interfész megvan, az utazási iroda kapcsolatba léphet a szolgáltatással a jól definiált intefész használatával.*



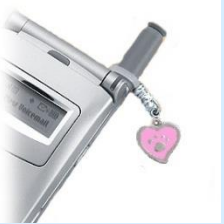
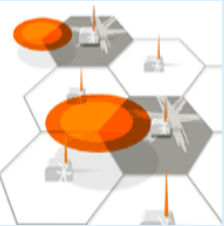
# Web szolgáltatás létrehozása és használata

- Egy Web szolgáltatás létrehozásának és telepítésének folyamata:
  1. WSDL interfészek létrehozása.
  2. WSDL interfészek publikálása UDDI használatával.
  3. Szolgáltatás implementálása egy szerveren, várakozás a kliensek jelentkezésére.
- Web szolgáltatás használatának folyamata:
  1. WSDL file megkeresése.
  2. WSDL segítségével egy kliens implementálása.
  3. A kliens segítségével a Web szolgáltatás meghívása a szerveren.
- A Web szolgáltatások erőssége a *laza csatolás* a szerver és a kliensek között.
  - A szerver és a kliensek különféle platformon lehetnek implementálva, eltérő gyártók termékein.



# Felderítés: Telefonkönyv, index vagy peer-to-peer?

- Három megközelítés létezik egy Web szolgáltatás felderítés szolgáltatás megvalósítására:
  - **telefonkönyvként** (registry),
  - **indexként**,
  - **peer-to-peer** (p2p) rendszerként.
  
- **A telefonkönyv megközelítés**
  - *központilag kontrollált* információtárolás;
  - A publikálás a *szolgáltatást nyújtó aktív részvételét* követeli meg;
  - A telefonkönyv tulajdonosa határozza meg, hogy kinek van joga publikálni;
    - Pl., egy tetszőleges harmadik fél nem publikálhatja egy tetszőleges személy szolgáltatásait szabadon.
  - A telefonkönyv tulajdonosa dönti el, hogy mi kerül a telefonkönyvbe.
  - UDDI-t gyakran a telefonkönyv megközelítés példájának tekintik, pedig indexként is használható.

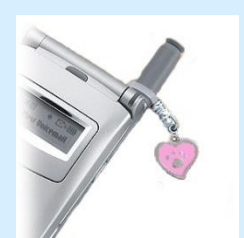
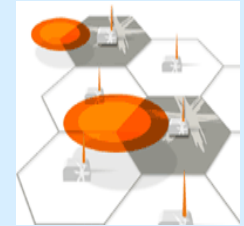




# Felderítés: Telefonkönyv, index vagy peer-to-peer?

## □ Az **index megközelítés**

- Az index egy *útmutató máshol elérhető* információhoz;
- *nem központilag kontrollált*, hogy milyen információkra hivatkozik;
- a *publikálás passzív*;
  - PI: A szolgáltatást nyújtók megjelenítik a szolgáltatásaik funkcionális leírását a Weben, és akik érdeklődnek (pl. indexek működtetői) begyűjthetik azokat.
- *bárki létrehozhatja a saját indexét*;
  - Miután a leírások nyilvánosan elérhetőek, szabadon begyűjthetők és indexelhetőek (ld. Google);
- a hivatkozott információ elavult lehet;
  - ...de ellenőrizhető használat előtt!
- harmadik féltől származó információt is tartalmazhat;
- különböző indexek különböző információt (szűkszavúbb – részletesebb) nyújthatnak.



## Felderítés: Telefonkönyv, index vagy peer-to-peer?

### A **telefonkönyv** kontra **index** megközelítés

- A lényegi különbség a kontrollban van: Ki és mit kontrollál, és hogyan kerülnek a leírások a gyűjteménybe?
  - *Telefonkönyv* esetén: a *tulajdonos* dönt.
  - *Index* esetén: a „piac” szabályoz.
    - Szabadpiaci mechanizmusok határozzák meg, hogy melyik indexet hányan és hogyan használják.
  
- **Peer-to-peer** (p2p) felderítés
  - A P2P megoldás egy alternatívát jelent *központi egység használata nélkül*;
  - nincs kritikus potenciális meghibásodási pont;
  - (ld. egyéb P2P előnyöket/hátrányokat).



## Felderítés: Telefonkönyv, index vagy peer-to-peer?

---

„Szóval, telefonkönyv, index vagy p2p?”

- ❑ **P2p rendszerek** alkalmasabbak *dinamikus* környezetben.
- ❑ Központosított **telefonkönyvek** megfelelőek statikus és szabályozott környezetben ahol a tárolt információ nem változik gyakran.
- ❑ **Indexek** alkalmasak olyan esetekben, amikor a jól skálázhatóság követelmény, és bizonyos verseny és diverzitás lehetséges a indexelési stratégiák között.



# Nem esett róla szó...

---

- Fontos aspektusok, amelyekről a kurzus keretében nem esik szó:
  - Web szolgáltatások biztonsága
  - Web szolgáltatások megbízhatósága
  - Web szolgáltatások menedzsmentje

