

Internet szolgáltatások és alkalmazások

MPLS (+QoS), áttérés nyílt rendszerekre, SOA, Web szolgáltatások (példa)

Tartalom

- Átviteli szolgáltatásminőség az interneten
 - alap IP
 - integrált szolgáltatás (Integrated Services – IntServ)
 - differenciált szolgáltatás (Differentiated Services – DiffServ)
 - **Multi-protokoll címkekapcsolás (Multi-Protocol Label Switching – MPLS)**
- Áttérés nyílt rendszerekre
 - OSA – Open Service Access (nyílt szolgáltatás hozzáférés)

MPLS (+QoS)

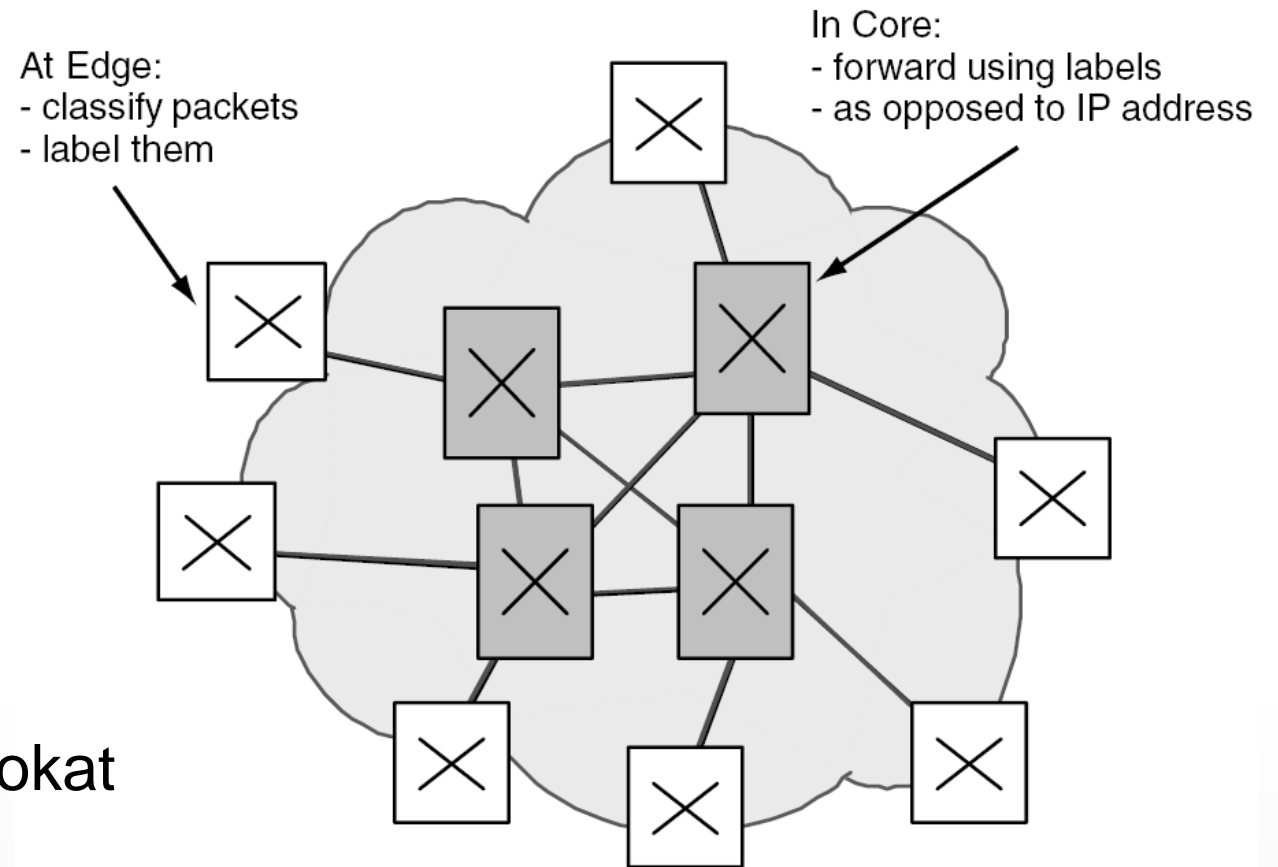
Multi-Protocol Label Switching (MPLS)

- Az MPLS nem kimondottan egy QoS technológia, de támogathatja a QoS biztosítását az IP routing kiterjesztéseként.
- Alapötlet: MPLS alkalmazásakor...
 - csomagokat megcímkézzük, amikor belépnek a hálózatba, amelyek ezek után
 - egy fix LSP-t (**Label Switched Path** - **címkekapcsolt útvonal**) követnek.

Multi-Protocol Label Switching (MPLS)

- Az MPLS főbb sajátosságai: Az IP csomagok egy rövid címke alapján kerülnek továbbításra (címkekapcsolás - **label switching**) a leghosszabb prefix cím keresés (IP routing) helyett, így
 - a továbbítás *egyszerű és gyors*;
 - a címkének csak link-szintű jelentősége van,
 - a címke minden csomópontban megváltoztatható az útvonalon.
- LSP-khez a csomagok flow szinten rendelhetők.
 - pl., **egy forrás-cél páros között több LSP is lehetséges, eltérő QoS karakterisztikával.**

MPLS architektúra



- A határ csomópontoknál a csomagokat *osztályozzák és felcímkézik.*
- A csomagok a helyi LSR (**címkekapcsoló útválasztó - Label Switching Router**) konfiguráció alapján továbbítódnak a következő csomóponthoz.
- A maghálózatban csak a címke használható az útvonal megtalálásához.

MPLS és QoS

- MPLS használható QoS biztosítására, ha egyéb QoS technológiákkal kombináljuk.
 - PI.1, MPLS leképezi az LSP-eket ATM virtuális kapcsolataira, így IP QoS-t biztosítva LSP-nként az ATM QoS-re alapozva.
 - PI.2, MPLS + DiffServ: különböző LSP-k használata szolgáltatási osztályonként, így lehetővé téve az osztályok egymástól független útválasztását.

MPLS összegzés

- Az MPLS önmagában nem egy QoS technológia, de nyújt olyan funkcionalitást amely megkönnyíti a QoS biztosítását. QoS könnyebben biztosítható az útvonalak rögzítésével.
- A hálózatot monitorozva és a forgalom elterelése a torlódásos szakaszokról bizonyos szintű minőséget biztosít önmagában is.
- Másrésztől megkérdőjelezhető, hogy jó döntés-e még egy járulékos réteget hozzáadni az IP alá(?!)
 - menedzsment komplexitását növeli;
 - a hálózati rétegnek is többletköltséget jelent;
 - IP-re támaszkodik (jelzés a címkék terjesztéséhez és útválasztó protokollokhoz: RSVP-TE (Traffic Engineering)).

IntServ, DiffServ, MPLS összefoglalás

- **IntServ** (RSVP) biztos (matematikailag igazolható) QoS határértékeket, de **nem skálázható**.
- **DiffServ** gerinchálózatokban hasznos, ahol egyedi alkalmazás folyamatok helyett forgalom **aggregátumoknak biztosít garanciát**.
- **MPLS** önmagában nem egy QoS technológia, de nyújt olyan funkcionalitást amely **megkönnyíti a QoS biztosítását**.



Áttérés nyílt rendszerekre

API-k

Tartalom

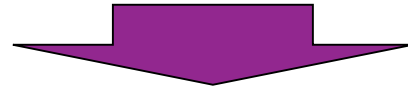
- Átviteli szolgáltatásminőség az interneten
 - alap IP
 - integrált szolgáltatás (Integrated Services – IntServ)
 - differenciált szolgáltatás (Differentiated Services – DiffServ)
 - Multi-protokoll címkekapcsolás (Multi-Protocol Label Switching – MPLS)

- **Áttérés nyílt rendszerekre**
 - OSA – Open Service Access (nyílt szolgáltatás hozzáférés)

Áttérés nyílt szolgáltatásokra

- **Konvergens hálózati trend**

- RÉGI: **zárt**, magántulajdonban kezelt rendszerek
 - természeténél fogva rugalmatlan, nem egykönnyen programozható
 - Új szolgáltatások bevezetése gyakran jelentős anyagi és idő-ráfordítást igénylő erőfeszítés.



- ÚJ: **nyílt rendszerek** lehetővé teszik bárki számára szolgáltatás nyújtását a felhasználóknak.

De milyen „rendszerekről” is van szó?

Áttérés nyílt rendszerekre – „minek?”

A hálózat megnyitásának célja: **Lehetővé tenni az alkalmazások számára a közcélú hálózatok kommunikációs képességeinek elérését.**

Motivációk:

- A **(hálózati) szolgáltatók** érdekeltek a forgalom (így a **bevétel!**) növelésében, attraktív szolgáltatások nyújtása által.
- A **stratégia**: **független szoftverfejlesztők** bátorítása olyan alkalmazások kifejlesztésére, amelyekkel a (hálózati) kommunikációs szolgáltatások fejleszthetők.
- Az **információs szolgáltatások fejlesztői** keresik a hálózati kommunikációs lehetőségeket saját IT alkalmazásaik kiterjesztéséhez.
 - (pl., banki tranzakció értesítők)

Áttérés nyílt rendszerekre – „hogyan?”

Mire van szükség?

- Az alkalmazások (azaz felhasználók) *elválasztása* a generikus funkcionalitástól.
- *Biztonságos, megbízható és számlázható* felhasználói kontroll biztosítása a *hálózati funkciók* kezeléséhez, amelyeket különböző hálózati elemek (pl. kapcsolók, adatbázisok) biztosítanak.
- Fontos: *Míg az internet természetesen bátorítja és elősegíti (pl. intelligens végberendezések alap átviteli szolgáltatással) a nyílt hálózati felhasználást, addig a távközlési szolgáltatók hálózatainak megnyitásánál figyelembe kell venni a hálózat **szolgáltatási képességeit**.*
- Ez megtehető egy **nyílt interfész** biztosításával, azaz az **API** (alkalmazás programozói interfész) koncepció használatával.

Nyílt szolgáltatás hozzáférés (OSA)

- **OSA** (**Open Service Access** – **nyílt szolgáltatás hozzáférés**).
- 3GPP szabvány definíciója szerint: „*az OSA egy koncepció az új szolgáltatások bevezetésének gyártófüggetlen módjára*”.
- A 3GPP szabvány definiál egy **OSA architektúrát** amely lehetővé teszi a távközlési szolgáltató (telco) hálózatán *kívüli* alkalmazás fejlesztők számára, hogy...
 - *kihasználják a hálózati funkcionalitást, és*
 - *információt kapjanak a hálózatból***...szabványos interfészeken (API) keresztül.**

Az API-k koncepciója

- **API** (**Application Programming Interface** – alkalmazás programozói interfész)
 - Def: **Szolgáltatás képes API-k** alatt a hálózat által nyújtott funkciók halmazát értjük, amely lehetővé teszi programozók számára a felhasználóknak szolgáltatást nyújtó programok írását.
- A **hálózati operátorok** által hozzáférhetővé tett **nyílt szabványos API-k** konvergens hálózatokban:
 - Lehetővé teszi **független szolgáltatás fejlesztők** számára a hálózat által támogatott alkalmazások fejlesztését **anélkül, hogy ezzel veszélyeztetnék a hálózatot vagy annak szolgáltatásait.**

Rétegek, protokollok, API-k

- A **távközlő hálózatok** világában a hangsúly a (hálózati) *rétegek specifikálásán* van.
 - A protokollok specifikálása és szabványosítása teszi lehetővé a hálózatok együttműködését.
 - Pl. TCP/IP stack: (HTTP + TCP + IP protokollok)
 - Pl. SS7 stack: (ISUP, TUP, MTP-x protokollok)
- A **szoftverek** világában a hangsúly a *rétegek közötti interfészek*en (API – Application Programming Interface) van.
 - *API-k szabványosításával platformfüggetlen alkalmazások* hozhatók létre.
 - Pl. POSIX (Unix és Linux rendszerek)
 - Pl. WIN32 (Microsoft Windows 95/98/2000, stb.)

TCP	Transmission Control Protocol	HTTP	HyperText Transfer/Transport Protocol
SS7	Signalling System Number 7		ISUP ISDN User Part
TUP	Telephony User Part	MTP	Message Transfer Part
POSIX	Portable Operating System Interface		

Rétegek, protokollok, API-k (2)

- Az **API koncepció** használatával a **távközlésben** lehetőség nyílik **hordozható, technológia- és protokollfüggetlen alkalmazások fejlesztésére**.
 - Pl. konferenciahívás esetén egyes résztvevők PSTN-en, mások mobilon vagy hálózati terminálokön kapcsolódnak.
- Megjegyzés: *Kb. 3 millió professzionális szoftverfejlesztő áll szemben kb. 10 ezer specializálódott távközlési szolgáltatás fejlesztővel!*

API-k jellemzői

- A **nyílt szolgáltatás hozzáférés API-k** hét **tulajdonsága**:
 - **Nyíltság:**
 - Széles körben elfogadott szabvány definiálja.
 - **Biztonság:**
 - A szolgáltatás interfészek hozzáférése csak hitelesített és felhatalmazott fél részére megengedett.
 - **Integritás (sérthetetlenség):**
 - A hálózat funkcionalitása nem veszélyeztethető az alkalmazás esetleges túlzott hívásai következtében.
 - **Rugalmasság:**
 - Az interfész lehetővé teszi változatos alkalmazások támogatását, és az interfész implementációja biztosítja különféle hordozó hálózatok használatát.
 - ...**(folyt.)**

API-k jellemzői (folyt.)

- A **nyílt szolgáltatás hozzáférés API-k** hét **tulajdonsága**:
 - ...
 - **Absztrakció:**
 - A mögöttes hálózat és annak heterogenitása rejtve kell maradjon az alkalmazás programozó előtt. (Azaz a programozótól nem elvárható, hogy megértse az adott hálózati technológiát és jelzési protokolljait.)
 - **Technológia semlegesség:**
 - Az API definíciója nem kötött egyetlen adott nyelvhez vagy implementációs technológiához.
 - **Szolgáltatás felderítés:**
 - Egy alkalmazás képes felderíteni a rendelkezésre álló szolgáltatásokat egy kiszolgáló node-ban.

Hálózatok megnyitása

- **Az *API koncepció* használatával a távközlésben lehetőség nyílik *hordozható, technológia- és protokollfüggetlen alkalmazások fejlesztésére.***
- Megjegyzés: *Kb. 3 millió professzionális szoftverfejlesztő áll szemben kb. 10 ezer specializálódott távközlési szolgáltatás fejlesztővel!*



Web szolgáltatások, SOA

Web services

Tartalom

- Internet szolgáltatási modell

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

- Web szolgáltatások technológiái
 - XML, HTTP, WSDL, SOAP, UDDI

Internet szolgáltatási modell

kliens-szerver, WWW, szolgáltatás integráció, interfészek

Internet szolgáltatási modell: kliens-szerver (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 (ism)

- 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 nevezzük **szolgáltatás integrációnak**.
- Az integráció megvalósításának egyik kulcs technológiája az **interfész** koncepció alkalmazása.
 - *Az interfész egy adott szolgáltatási komponens által kiajánlott 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.

Web szolgáltatások

Bevezető

Tartalom

- Internet szolgáltatási modell

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

- Web szolgáltatások technológiái
 - XML, HTTP, WSDL, SOAP, UDDI

Web szolgáltatások bevezető

- **Web szolgáltatások (Web Services)** a Web-en publikáltak, elérhetőek 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.**

W3C

- **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.” -

www.w3c.org

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

Web szolgáltatás – példa

Utazás menedzsment rendszer

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 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

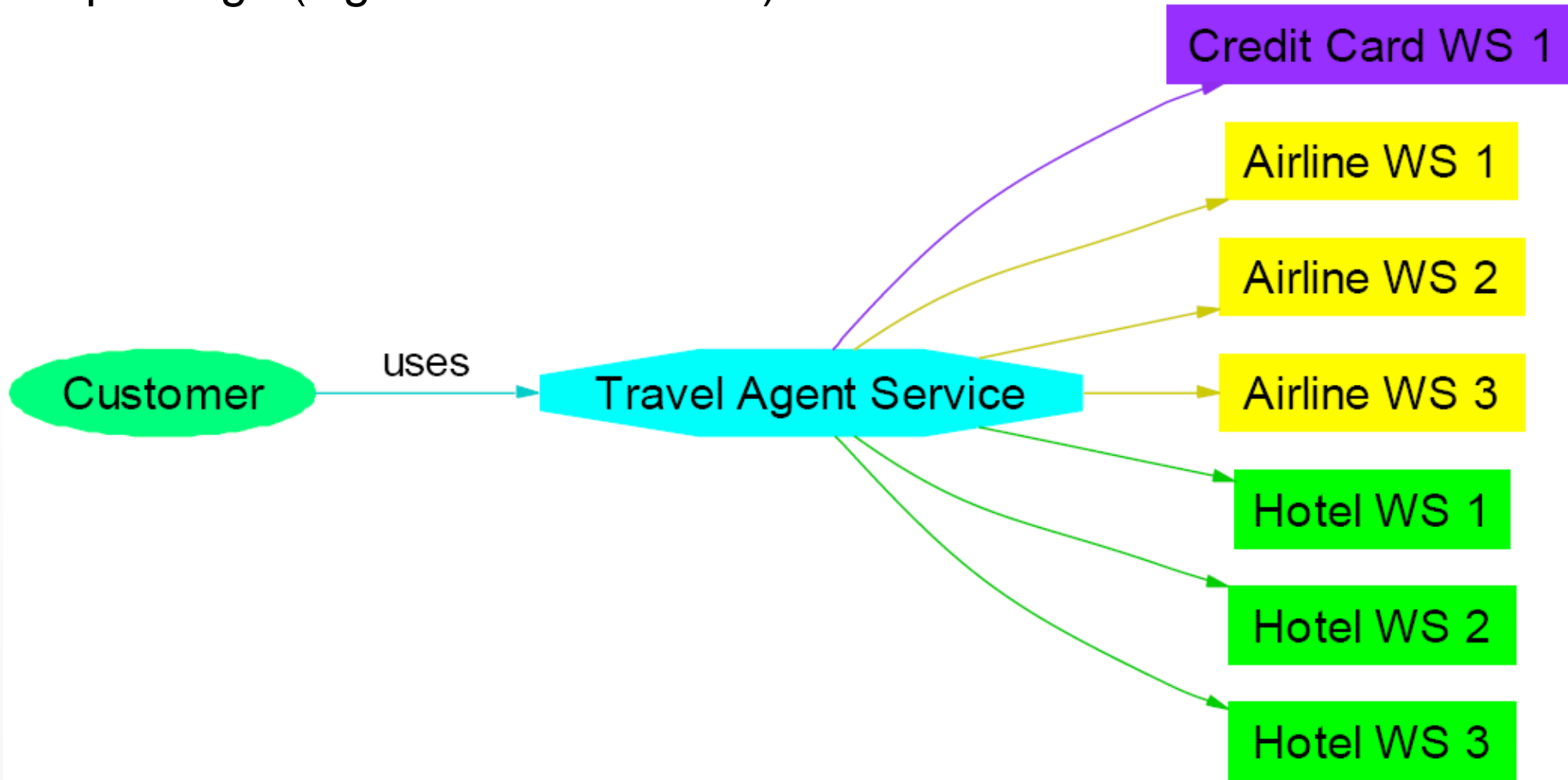
- Note:

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
 - 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.