SOA and Web services

- *Web services* offer one way to implement SOA
- Web services = WSDL, SOAP, and many other standards
  - Related to interoperability, orchestration, reliability, security, …, …
  - http://en.wikipedia.org/wiki/List_of_web_service_specifications
  - This collection of standards is sometimes also called *WS-**

2

Web-palvelukonsepti tarjoaa yhden tavan toteuttaa SOA. Tämä tapa perustuu ”Web-palvelustandardien” käyttöön: palvelut kuvataan WSDL-kielen avulla ja kommunikointi toteutetaan SOAPin avulla. Näihin kieliin palaamme myöhemmin. On kuitenkin todettava, että ”Web-palvelustandardeihin” liittyy myös paljon ongelmia. Toisaalta nämä standardit kehittyvät jatkuvasti ja toisaalta esimerkiksi palveluiden koordinointiin liittyen ei ”standardeista” olla aivan vielä päästy vastaavaan yhteisymmärrykseen. Sanaan ”standardi” kannattaakin yleisesti suhtautua varauksellisesti Web-palveluihin liittyvistä teknologioista puhuttaessa: kyseessä saattaa olla lähinnä ehdotus eikä varsinaisesti standardi. Ja toisaalta standardista käytetään usein rinnakkain useita eri versioita.

SOA and Web services

- SOA and Web services aim at automatically solving problems related to interoperability, configuration etc.
  - an agreement on the standards partly enables this
    - challenges: a whole range of ”standards”
  - possible problems:
    - genericity/flexibility vs. automation
- A lot of tool support for building Web services and client applications
  - varying features
  - support for Web service languages and recommendations, and their different versions varies

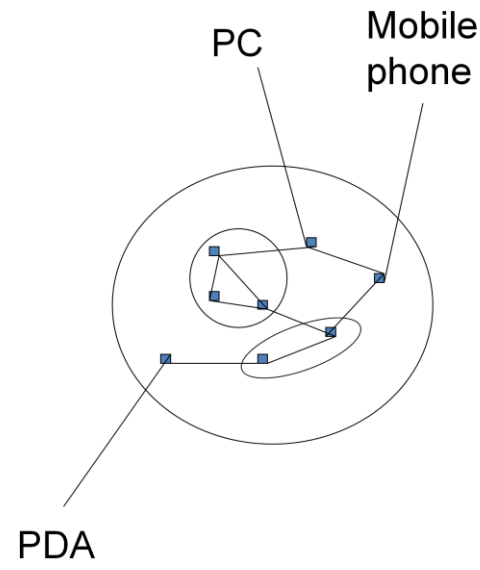
3

SOA ja Web-palvelut pyrkivät periaatteessa ratkaisemaan yhteentoimivuuteen, konfigurointiin jne. liittyvät ongelmat automaattisesti ja vieläpä ajonaikana. Esimerkiksi mikäli kutsuttavaa palvelua ei ole saatavilla tai yhteentoimivuusongelmia ilmenee, tulisi kutsuvan palvelun kyetä joko ratkomaan yhteentoimivuusongelmat tai korvaamaan ko. palvelu toisella vastaavan toiminnallisuuden omaavalla palvelulla. Tämä on kuitenkin vielä useissa tapauksissa kaukana käytännön toteutuksista siitä huolimatta, että useista standardeista onkin päästy jo yksimielisyyteen. Automaattisuuteen liittyy myös omat ongelmansa. Esimerkiksi muutosten tekeminen manuaalisesti voi olla vähintäänkin haasteellista.

Web-palveluiden ja asiakassovellusten tekemiseksi ja osin generoimiseksi on olemassa runsaasti työkalutukea. Esimerkiksi palvelun rajapintakuvaus (WSDL) voidaan generoida automaattisesti palvelun rajapinnan (esim. Java) perusteella. Nämä työkalut kuitenkin poikkeavat toisistaan sekä tarjottujen ominaisuuksien että toteutustapojen suhteen. Esimerkiksi samasta rajapinnasta (esim. Java) eri työkalut generoivat erilaisia WSDL-kuvauksia. Lisäksi eri työkalujen tarjoamatuki eri Web-palvelukielille, niiden eri versiolle ja eri suosituksille vaihtelee.

A vision of Web services

- Software will be assembled from a web of services
- Building applications just-in-time
- Discovering and coordinating services on the Web dynamically
- From distributed systems to de-centralized applications



4

Web-palveluiden alkuperäisen ja tavoiteltavan vision mukaisesti palveluja tulisi voida etsiä ja niitä tulisi voida käyttää dynaamisesti. Tietyn palvelun käyttöön sitominen tulisi siis olla ajonaikainen (dynaaminen) toimenpide, ei staattinen. Sovelluksia tulisi voida muodostaa olemassa olevia palveluita hyödyntäen aina kulloisenkin tarpeen mukaisesti. Nämä yhdessä edellyttävät lisäksi sen, että palveluiden koordinoinnin tulisi myös olla dynaamista.

Ehkäpä yksi oleellisimmista näkökulmista on, että Web-paveluiden myötä siirryttäisiin hajautetuista järjestelmistä ei-keskitettyihin järjestelmiin. Tämä merkitsisi sitä, että tarjolla olisi verkko erilaisia (hallinnan näkökulmasta itsenäisiä ja tasavertaisia) palveluita, joita mikä tahansa sovellus voi käyttää ja mahdollisesti yhdistellä uusiksi palveluiksi.

Nämä edellä esitetyt visiot ovat luonnollisesti vielä kaukana todellisuudesta. Esimerkiksi tietoturvakysymykset ja käyttöoikeudet asettavat reunaehtoja ja vaatimukia, joita ei vielä yleisellä tasolla olla täysin ratkaistu. Yksittäisiä ratkaisuja ja ratkaisuehdotuksia on esitetty, mutta yhtenäistä periaatetta ja käytäntöjä ei vielä ole. Näin ongelmiin palaamme myöhemmin.

Defining Web services

- W3C puts it: ”programmatic interfaces made available for application to application communication are referred to as Web services”
- ...or more precisely: ”A Web service is a collection of functions packaged as a single entity and published to the network for use by other applications”
  -> a hierarcy of Web services: more complicated ones are aggregated from simpler ones
- or perhaps:“XML applications mapped to programs, objects, databases, or business functions”

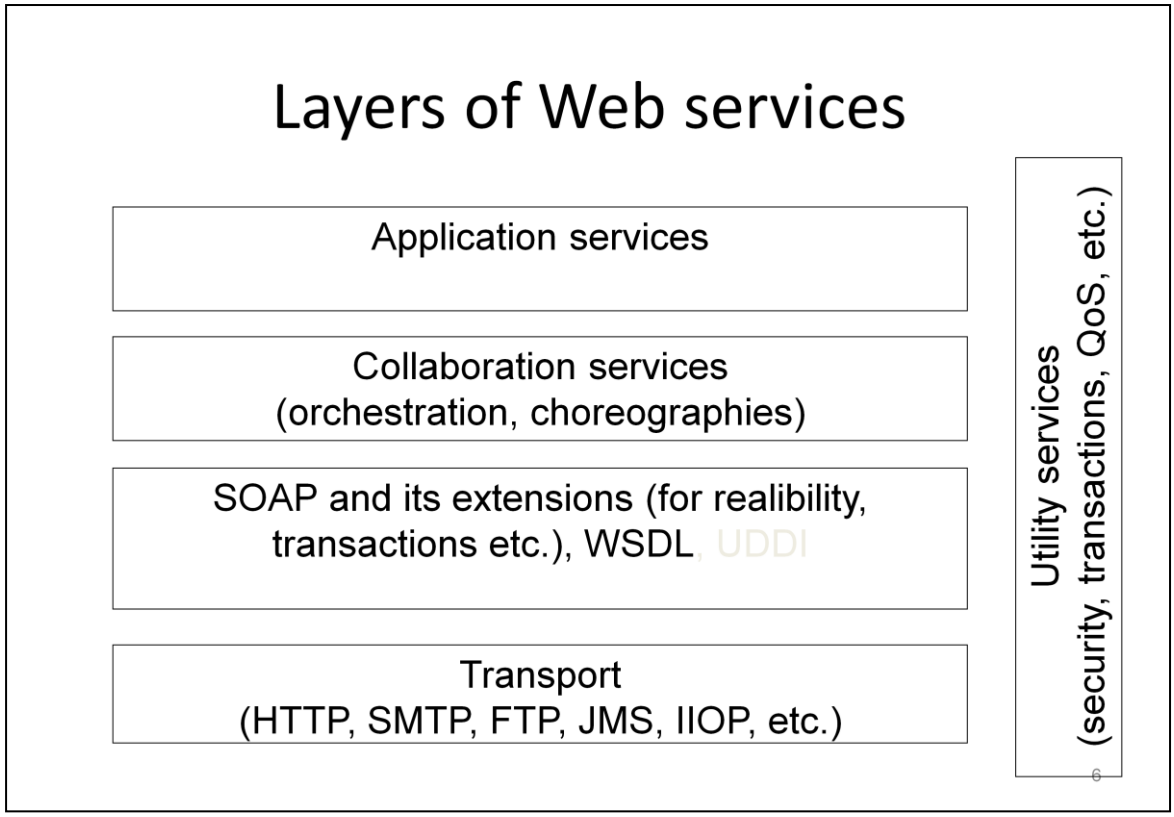
5

Web-palveluille on esitetty lukuisia määritelmiä. Yksinkertaisimmillaan niiden on sanottu olevan sovelluksia, joihin voidaan ottaa yhteys käyttäen standarditeknologioita (kuten XML ja HTTP). On myös sanottu, että Web-palvelu on käytännössä sama asia kuin SOAP-protokollan käyttö (tästä lisää myöhemmin). Nämä eivät kuitenkaan ole kovin hyviä määritelmiä, sillä ne ovat aivan liian laajoja. W3C puolestaan määrittelee Web-palvelun (vapaasti käännettynä) joukkona ohjelmiin liittyviä rajapintoina, jotka ovat käytettävissä sovellusten välisessä kommunikoinnissa. Tämäkin määritelmä on melko yleinen.

Hieman tarkemmin määriteltynä Web-palvelun on sanottu olevan joukko funktioita, jotka on **pakattu** yhdeksi kokonaisuudeksi ja **julkaistu** verkossa muiden sovellusten käytettäväksi. Tämä määritelmä on jo selvästi parempi. Oleellista tässä on se, että näitä tarjottuja funktioita voidaan yhdistellä ja pakata uusiksi palveluiksi. Se myös implikoi Web-palveluille hyvin oleellisen piirteen: Web-palveluhierarkian. Toisin sanoen yksinkertaisimmista palveluista voidaan koostaa monimutkaisempia palveluita. Toinen oleellinen asia tässä määritelmässä on palveluiden julkaiseminen. Jotta Web-palvelu olisi aidosti vapaasti etsittävissä ja käytettävissä, edellyttää se, että palveluiden käyttäjät voivat etsiä palveluita jostain yleisesti tunnetusta ”markkinapaikasta”.

Viimeinen määritelmä kuvaa Web-palvelut XML-sovelluksina, jotka on sidottu joihinkin ohjelmiin, tietokantoihin tai liiketoimintafunktioihin. Web-palveluissa käytetään XML-pohjaisia kieliä, mutta palveluiden kutsuminen XML-sovelluksiksi voi olla myös harhaanjohtavaa. Tässä määritelmässä oleellista on se, että itse palvelun toiminnallisuus voi mitä vain ja se on voitu toteuttaa millä tahansa halutulla tavalla. Palvelun käyttöä varten tulee kuitenkin toteuttaa käytettäviä standardeja ymmärtävä ja käsittelevä interaktiota tukeva kerros. Web-palvelukonseptin voidaan ajatella olevan myös mekanismi back-end systeemien paketoimiseksi (wrapping). Tällaisia back-end systeemejä voivat olla vaikkapa tietokanta, legacy-systeemi jne.

Käyttäessään Web-palvelua ohjelma lähettää pyynnön/kyselyn XML-muodossa ja yleensä myös vastaanottaa vastauksen niin ikään XML-muodossa. Jotta kommunikaatio olisi mahdollista, täytyy viestien formaatista ja käytettävän palvelun rajapinnasta sopia. Tämän lisäksi tulee sopia siitä mekanismista, joka määrittelee miten Web-palveluja tulee julkistaa ja miten niitä voidaan etsiä.



Kalvolla esitetyn kuvan alimpana kerroksena ovat viestinvälitysprotokollat. Vaikka Web-palveluita usein sanotaan käyttävän HTTP:tä – ja näin käytännössä hyvin usein onkin – itse Web-palvelukonseptia ei ole sidottu tiettyyn viestinvälitysprotokollaan. Yhtä hyvin käytössä voisi olla esimerkiksi SMTP tai FTP. Kuvan seuraavan kerroksen muodostavat Web-palvelustandardit SOAP, WSDL ja UDDI. Näistä ensimmäistä käytetään kommunikointiin sovellusten kesken. WSDL-kieltä puolestaan käytetään kuvaamaan tarjottu Web-palvelu (tarjotut funktiot ja yhteydenottotapa). UDDI on yksi tapa toteuttaa palveluiden ”markkinapaikka” (rekisteri), mutta muitakin vaihtoehtoja on olemassa. Palvelun ”mainostaminen markkinapaikassa” ei palvelun pystytyksen kannalta toki ole välttämätöntä. Mikäli asiakaskunta tietää miten palveluun saa yhteyden ja miten sitä voidaan kutsua, on se tarpeetonta. Näin on usein esimerkiksi kun Web-palvelukonseptia käytetään rajoitetussa ympäristössä kuten yrityksessä. Tällöin mikäli käyttö tapahtuu palomuurien sisäpuolella, ei kommunikoinnin turvallisuuden takaamiseen tarvitse kiinnittää välttämättä huomiota.

Palveluverkosto (esimerkiksi toisiaan käyttävät palvelut) edellyttää palvelujen koordinointia. Koordinointipalvelut on esitetty kuvassa kolmantena kerroksena (collaboration services). Esimerkiksi sekvenssi yksittäisten palvelujen suorittamista operaatiosta voidaan haluta koostaa yhdeksi liiketoimintatransaktioksi. Palveluiden yhdistämiseen käytetään nk. *orkestointi-* ja *koreografiakieliä*. Näistä orkestrointi tarkoittaa palveluiden yhdistämistä yhdestä tietystä liiketoimintaprosessia suorittavasta näkökulmasta, kun taas palvelukoreografia sallii useita samanaikaisia ja tasavertaisia näkymiä liiketoimintaprosessiin. Näihin palataan vielä myöhemmin. Lopuksi kuvan ylimpänä kerroksena ovat itse Web-palvelut.

Kuvan kaikkia eri kerroksia koskee ja tukee joukko muita hyödyllisiä palveluja (utility services). Jotkin käytetyt ratkaisut koskevat kaikkia kerroksia ja voivat siten vaikuttaa käytettäviin formaatteihin, protokolliin ja API-määrittelyihin tai vaikkapa vaikuttaa niiden valintaan. Turvallisuusaspektit ovat

esimerkiksi hyvin tärkeitä Web-palvelujen kannalta (erityisesti liiketoimintakriittiset palvelut). Vaikka SOAP ei tällä hetkellä suoranaisesti tuekaan turvallista viestinvälitystä, on tämä puute huomioitu ja sen korjaamiseksi on esitetty ehdotuksia SOAP-laajennoksiksi ja esim. digitaalisten allekirjoitusten liittämiseksi SOAP viesteihin (SOAP Security Extensions: Digital Signature). Toisaalta turvallisuus alimmalla tasolla voi tarkoittaa vaikkapa SSL:n tai TLS:n käyttöä. Palvelun laadun huomioiminen (Quality of Service, QoS) puolestaan painottaa vastinaikojen, hinnan, transaktioiden ja muita liiketoimintainteraktioissa oleellisia asioita.

Usage examples of Web services

- Software migration and platform integration
  - wrapping legacy systems to act like a Web service
  - ”agreement on disagreement”
- Agile business processes
  - flexible integration
  - business collaborations and agreements
  - e.g., streamlining business operations, invoicing, shipping operations etc.
- Service registries and searching
  - searching goods or services according to given requirements (e.g., best price)
  - coordinating travel tickets or hotel reservations for a given date
- RPCs
  - location transparency!
- And many, many more…

7

Web-palvelut voivat käytännössä olla mitä tahansa. Paketoimalla vanha legacy-systeemi Web-palveluksi sovitaan käytännössä erimielisyydestä: legacy-systeemien logiikka tai toteutusta ei tarvitse muuttaa ja ne voivat olla hyvinkin eri tavoin toteutettuja.

Erilaiset ketterät liiketomintaprosessit ovat myös potentiaalisia Web-palvelukonseptin käyttökohteita. Esimerkiksi liiketoimintaoperaatiot (laskutus, tilaukset jne.) voidaan tarjota Web-palveluina. Liiketoimintasopimuksien ja liiketoimintaprosessien määritykset ovat erityisesti painotettuja RosettaNet ja ebXML –kosepteissa, jotka ovat tietyssä mielessä vaihtoehtoisia näkökulmia Web-palveluihin. Näihin palataan myöhemmin.

Yksi palvelun muoto voi olla vaikkapa muiden palveluiden etsintään tarkoitettu rekisteri. Rekisteristä voidaan etsiä palveluita annetuin kriteerein (esim. halvin matka). Palvelu voi myös hyödyntää muita palveluita. Esimerkiksi matkanjärjestämispalvelu voi käyttää hyväkseen sekä palvelua, jonka avulla voidaan etsiä halvimmat lennot kohteeseen annetulla aikavälillä, että palvelua, joka etsii sopivimman hotellin (annettujen kriteerien mukaisesti) matkakohteessa.

Web-palveluja käytetään – tietyssä mielessä vastoin sen alkuperäistä käyttötarkoitusta ja visiota – etäkutsujen toteuttamiseen. Se onkin tällä hetkellä yleisin käyttömuoto. Tämä voidaan tehdä siten, että kutsun muoto on kutsuttavan ohjelman sijainnista riippumaton eli se ei siis näy kutsun muodosta (location transparency). Etäkutsujen toteuttamiseen on kuitenkin jo olemassa useita eri menetelmiä.

Web services - a silver bullet or reinventing the wheel?

- A silver bullet? No
  - Web services provides rather a new layer, not a replacement for existing computing infrastructure
  - Web services play an important role as a tool for bridging technology domains
  - bridging is based on standard formats
- Reinventing the wheel? No
  - Web services are not language nor platform dependent like DCOM, RMI etc.
  - Web services are dynamic and disconnected; connections are transient and temporary
  - Web service infrastructure assumes that parties can be connected without prior knowledge of each other, i.e., any client can bind to any published service (as long as the service description and security requirements are followed)
  - From distributed applications to de-centralized applications

8

Web-palvelu ei tarjoa mitään mullistavan uutta. Web-palvelua voidaan ajatella ylimääräisenä kerroksena, joka mahdollistaa sovellusten välisen interaktion. Se ei korvaa eikä sen ole tarkoitus korvata olemassa olevia tekniikoita (esim. hajautustekniikat) eikä olemassa olevia ohjelmistoja. Koska kyseessä on kevyt XML-pohjainen integrointi käyttäen yleisesti hyväksyttyjä standardeja, antaa Web-palvelukonsepti mahdollisuuden silloittaa eri teknologioita. Esimerkiksi palvelu voi olla toteutettu .NET ympäristössä kun taas sen asiakas voi J2EE-toteutus.

Web-palvelu ei kuitenkaan ole myöskään pyörän uudelleen keksimistä. Web-palveluissa ei ole kyse niinkään hajautusteknologiasta (kuten CORBA, DCOM, RMI) vaan siinä pyritään ei-keskitettyyn järjestelmään (ainakin periaatteessa). Lisäksi Web-palvelut eivät ole ohjelmointikieli- tai alustariippuvaisia kuten esimerkiksi RMI ja DCOM. Edelleen voidaan sanoa, että yhteydet on tarkoitettu transienteiksi: palveluihin kytkeydytään dynaamisesti aina tarpeen mukaan. Palvelun käyttäjän ei tarvitse tietää palvelun toteutuksen yksityiskohtia vaan sille riittää ainoastaan palvelun kuvaus. Web-palveluissa on myös oleellista se, että palvelun käyttäjän ja palvelun välillä ei tarvitse olla ennalta määriteltyä sopimusta (tämä on kuitenkin mahdollista ebXML-konseptissa) vaan periaatteessa mikä sovellus tahansa voi käyttää julkaistua palvelua, edellyttäen että palvelun kuvausta ja mahdollisia turvallisuusvaatimuksia noudatetaan. Tämä luonnollisesti pätee annetussa ympäristössä: mikäli kyseessä on esimerkiksi yrityksen sisäisessä verkossa tarjottu palvelu, voi palvelua käyttää vain sovellukset, joiden on mahdollista ottaa palveluun yhteys.

Formats and protocols

- Web Service Description Language (WSDL)
  - a format for describing a Web service interface, data, message types, interactions patterns, and protocol mappings
- Simple Object Access Protocol (SOAP)
  - a message format for invoking a Web service
  - binds the WSDL description

9

Edellä esitetty Provider-Requestor-Broker -roolijako kuvaa erilaiset Web-palvelujen käsitteet ja konseptit. Se ei vielä ota kantaa siihen, mitkä ovat käytetyt formaatit ja APIt. Tosin myös siitä on päästy yleisesti yhteisymmärrykseen. Palvelujen kuvaukset ja viestinvälitys suoritetaan käyttäen WSDL ja SOAP -formaatteja (alunperin IBM:n ja Microsoftin ym. yhteisenä ehdotuksena). SOAP ja WSDL ovatkin jo vakiinnuttaneet asemansa. Palvelurekisteri voidaan toteuttaa esimerkiksi UDDI-rekisterinä, mutta sen rinnalle on tullut muitakin kandidaattiteknologioita pääosin UDDI:n tietynlaisen rajoittuneisuuden vuoksi.

WSDL ja SOAP -spesifikaatioit kehittyvät edelleen: uuden versiot seuraavat toisiaan ja niihin liittyviä muita suosituksia ja ehdotuksia (W3C) tehdään jatkuvasti. Eri versioiden käyttö luonnollisestikin johtaa yhteentoimivuusongelmiin. Lisäksi nämä spesifikaatiot sisältävät optionaalisia sääntöjä ja kaikki toteutukset eivät välttämättä toteuta samoja optionaalisia piirteitä. Tämä saattaa myös aiheuttaa yhteentoimivuusongelmia, vaikka eri osapuolet käyttäisivätkin samoja versioita. Web Service Interoperability (WS-I) yhteisö pyrkiikin pureutumaan näihin ongelmiin antamalla suosituksia näiden spesifikaatioiden käyttötavoista ja niiden eri versioiden yhteiskäytöstä. Vaikka tavoite onkin varsin hyvä, on tehtävä silti varsin haasteellinen. WS-I yhteisön roolia käsitellään tällä kurssilla myöhemmin.