

hyväksymispäivä arvosana

arvostelija

OSDA ja UDDI palveluhakemistoina

Ilkka Vaahtoranta

Helsinki 20.09.2007

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

HELSINGIN YLIOPISTO h HELSINGFORS UNIVERSITET – UNIVERSITY OF
HELSINKIh

| | |
|---|--|
| Tiedekunta/Osasto h Fakultet/Sektion – Laitos h Institution h Department | |
| Faculty/Section | |
| Matemaattis-luonnontieteellinen tiedekunta | |
| Tietojenkäsittelytieteen laitos | |
| Tekijäh <input type="checkbox"/> Författare h Author | |
| Ilkka Vaahtoranta | |
| Työn nimih <input type="checkbox"/> Arbetets titel h Title | |
| OSDA ja UDDI Palveluhakemistoina | |
| Oppiaine h Läroämne h Subject | |
| Tietojenkäsittelytiede | |
| Työn lajih <input type="checkbox"/> Arbetets art h Level | Aikah <input type="checkbox"/> Datum h Sivumääräh <input type="checkbox"/> Sidoantal h Number of pages |
| | Month and year |
| | 12.9.2007 |
| | 15 sivua + 1 liitesivu |
| Tiivistelmäh <input type="checkbox"/> Referat h Abstract | |
| <p>Palvelusuuntautuneet Arkkitehtuurit (Service Oriented Architecture, SOA) ovat arkipäivää yritysten välisessä kommunikoinnissa, Palveluiden ja tietojen välityksessä. Palveluista tarvitaan haku- tai ilmoitusmahdollisuutta, mikäli palveluja rakennetaan julkisiksi monenvälisiksi palveluiksi. Palvelut sekä tietoa niiden laadusta ja toiminnasta on oltava saatavilla. Open Service discovery architecture for efficient cross domain service-provisioning (OSDA) on malli, joka kykenee yhdistämään heterogeenisiä palvelun tarjoajia useista eri toimintaympäristöistä ja tarjoaa tehokkaan hakumenetelmän palveluille. Universal Description, Discovery and Integration (UDDI) on Internetissä tarjolla oleva XML-pohjainen rekisteri, jossa palveluntarjoajat pystyvät julkaisemaan tietoja tarjoamistaan palveluista. UDDI on teollisuuden aloitteesta rakennettu olemassaoleva ympäristö. Tässä seminaarityössä ensin esittelen UDDI:n ja OSADA:n Arkkitehtuuri periaatteet.</p> | |
| Avainsanat – Nyckelord h Keywords | |
| OSDA, UDDI, palveluhakemisto | |
| Säilytyspaikka h Förvaringställe h Where deposited | |
| Muita tietoja h Övriga uppgifter h Additional information | |

Sisältö

| | |
|---|--------------------|
| 1 Johdanto..... | 4 |
| 2 Web Services..... | 5 |
| 3 Universal Description, Discovery, and Integration (UDDI)..... | 5 |
| 3.1 UDDI Tietomalli..... | 7 |
| 3.2 UDDI Rekisteri..... | 8 |
| 3.3 UDDI Replikointi..... | 9 |
| 4 Open service discovery architecture, OSDA..... | 11 |
| 4.1 OSDA Arkkitehtuuri..... | 12 |
| 4.2 USD palvelunkuvaus..... | 13 |
| 5 Yhteenveto..... | 14 |
| 6 Lähteet..... | 15 |

1 Johdanto

Nykyisen kaltaisiin Service Oriented Architecture (SOA) toteutuksiin kuuluu kolme osapuolta: palvelun toteuttaja, palvelun käyttäjä ja palveluhakemisto, josta palvelut on mahdollista löytää. Palvelut toteutetaan usein käyttäen nykyistä Internetin infrastruktuuria ja olemassa olevia protokollia.

Palveluiden tuottaja tarjoavat omaan toimintaansa tai toimialaansa liittyviä palveluja. Palvelut voivat olla palvelu- tai tietosisältökeskeisiä. SOA järjestelmien käyttäjät tarvitsevat palveluita toteuttaakseen omia palveluitaan ja käyttävät palveluhakemistoja löytääkseen tarvitsemansa palvelut. SOA järjestelmät tarvitsevat palveluhakemistoja pystyäkseen löytämään tarvittavat palvelut Internetistä [ERN02]. SOA toimintamalli tarjoaa yrityksille ja muille tahoille pääsyn palveluihin, jotka mahdollisesti sijaitsevat maantieteellisesti hajallaan.

SOA mallit nostavat oliopohjaisen suunnittelun seuraavalle tasolle. Palveluiden käyttö ei rajoitu käytettyyn käyttöjärjestelmään tai ohjelmoitikielen. Mikäli toimisimme käyttöjärjestelmätasolla, SOA tyyppiset palvelut eivät olisi mahdollisia. Palveluiden helppo saatavuus mahdollistaa uusien business -mallien synnyn ja nopean tuotekehityksen esimerkiksi mashup tyyppisillä palveluilla, jossa kootaan yhteen monesta eri toimittajan palvelusta koostuva palvelu [CST07].

SOA:n peruseriaatteisiin kuuluvat löyhä- ja myöhäinen kytkentä, sekä Palveluiden uudelleen käytön. Löyhällä kytkennällä tarkoitetaan mahdollisuutta eristää ohjelmisto moduulit hyvin määritellyn rajapinnan taakse. Myöhäinen kytkentä tarkoittaa tässä ajon aikaista palvelun valintaa. Valinta voi myöhäisessä kytkennässä perustua esimerkiksi saatavuuteen tai vasteaikoihin.

Palveluiden ja resurssien julkaiseminen yhteistoimintaa varten voidaan toteuttaa usealla eri tekniikalla. Tässä työssä tarkoitukseni on käydä läpi palveluhakemistojen markkinastandardi Universal Description, Discovery, and Integration (UDDI) palveluhakemiston peruseriaatteet

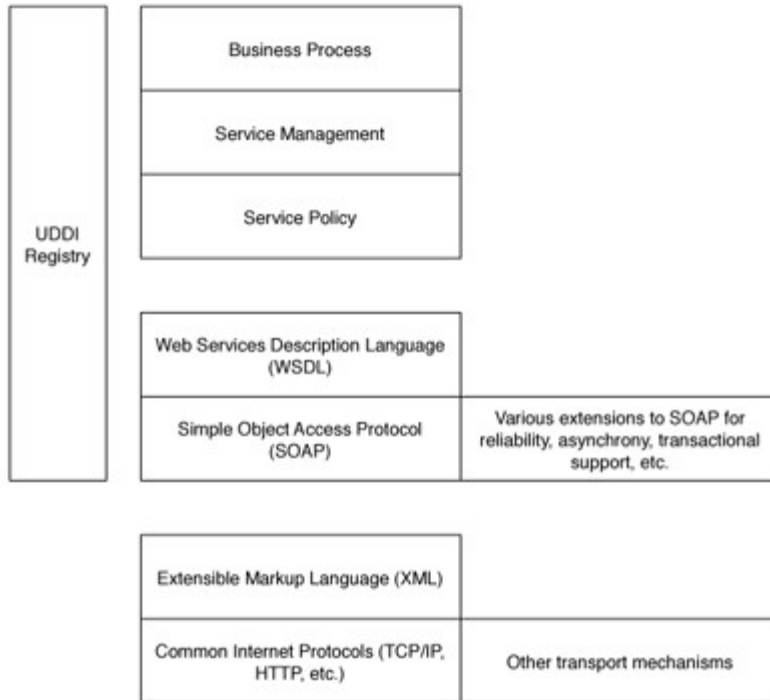
ja tutustua Open Service Discovery Architecture (OSDA) malliin. OSDA on ehdotus väliohjelmistopalveluhakemistomallista, joka toteuttaa yritysten sisäisen ja ulkoisen Palveluiden löytämistarpeet. OSDA yhdistää useita Palveluiden ja resurssien löytämisteknologioita samaan palveluhakemistoon.

2 Web Services

Markkinastandardina toimivan Web Services SOA malliin kuuluu kokoelma olemassa olevia Web standardeja. Mallia ei ole varsinaisesti suunniteltu kokonaisuutena. Web Services on kehittynyt markkinoiden valitessa standardeista parhaiten käyttöönsä sopivia ja yleisesti hyväksytyjä protokollia. Tärkeimpiä tällä hetkellä Web Services SOA mallin käytössä olevista standardeista ovat XML viestien ja sisällön kuvaukseen, WSDL Palveluiden kuvaukseen, SOAP kommunikointiin ja UDDI Palveluiden julkaisuun sekä löytämiseen.

3 Universal Description, Discovery, and Integration (UDDI)

UDDI kokoaa olemassa olevia Web standardeja yhteen ja muodostaa niillä Web Services hakemisto palvelun. Palvelun tarkoitus on tarjota toimittajatietoon, dataan ja metadataan nojautuva Palveluiden hakupalvelu. UDDI on itsessään Web Service mallin mukainen palvelu. Kuvassa 1 kuvataan UDDIn rakentumista olemassa olevien standardien päälle. UDDI rekisteri kommunikoi ulkoisesti ja sisäisesti SOAP protokollalla ja käyttää tietojen kuvaukseen XML kieltä. Palveluiden rajapintakuvaukset toteutetaan WSDL määritelmän mukaan. UDDI standardi on tällä hetkellä versiossa 3.0. UDDI on yksi Organization for the Advancement of Structured Information Standards (OASIS) yhteisön ylläpitämistä standardeista.



Kuva 1: UDDI suhteessa muihin WEB Services protokollisiin [UDD08]

UDDI koostuu kahdesta peruspalvelusta. Ensimmäisessä palvelun tuottaja julkaisee kuvauksen itsestään ja tarjoamastaan palvelusta. Toisessa palvelun kuluttaja voi hakea haluamiaan palveluja niiden kuvausten tarjoaman tiedon perusteella. Kuvaukset ja niiden tietosisältö ovat keskeisessä asemassa, kun palveluja haetaan UDDI hakemistosta.

UDDIn hakuominaisuudet ovat kuitenkin rajatut. Palveluita voi etsiä muutamien muuttujien avulla annetuista BusinessEntity, BusinessService, BindingTemplate ja tModel kuvauksesta. UDDI hakuominaisuuksiin onkin ehdotettu parannuksia, jotka tekevät palveluiden löytämisestä helpompaa. Suurimmat haasteet ovat Palveluiden saatavuus, muutosten päivittyminen ja semanttisen tiedon puute palvelukuvauksissa. A. ShaikhAli et al. [ASH03] kuvaavat UDDIe palvelu, joka laajentaa UDDI rekisterin hakuominaisuuksia. UDDIe tuo WSDL palvelukuvauksiin lisää metatietoa palveluiden laadusta ja tehokkuudesta sekä tehostaa hakuja, kun ne perustuvat numeeriseen dataan. UDDIe tarjoaa myös Palveluiden aikasidonnaisen tarjoamisen niin, että muutokset saataviin palveluihin tulevat julkiseksi nopeasti.

3.1 UDDI Tietomalli

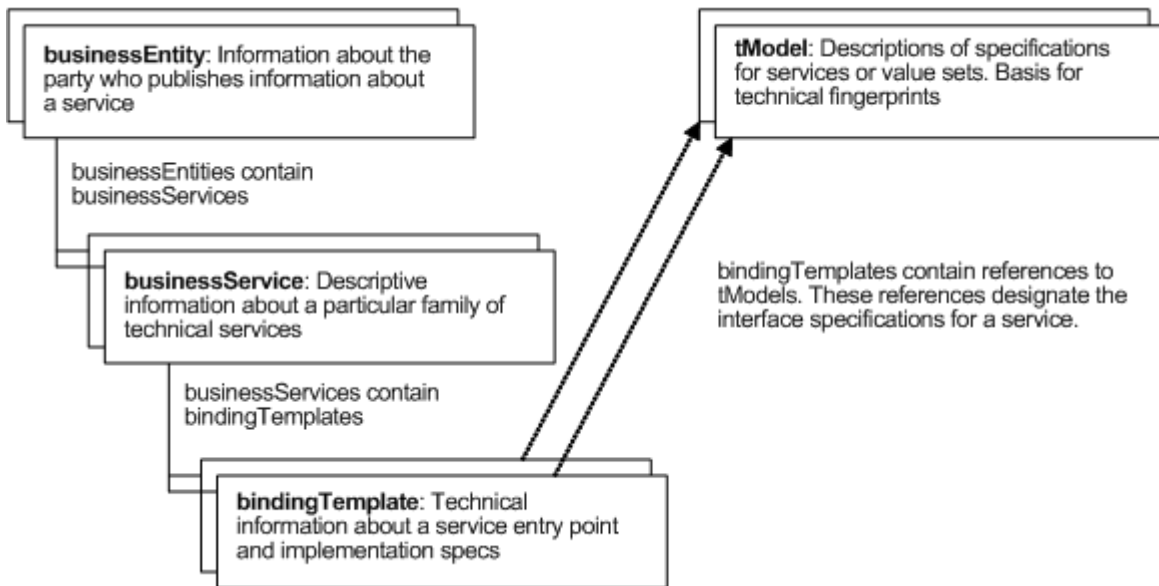
Korkein taso UDDIn tietomallissa on BusinessEntity. Se sisältää tietoa siitä, kuka tai mikä kyseinen organisaatio on. Tällä tiedolla ilmaistaan myös, millä toimialalla yritys tai yhteisötoimii ja mitä palveluja se tuottaa. [UDD03]

BusinessService tieto kuvaa palvelun tarjoavan osapuolen muita vastaavia palveluja. Se kerää palveluja samaan kategoriaan. [UDD03] Jokainen BusinessService kuuluu vain yhdelle businessEntity elementille.

Pakolliset palvelun käyttöön liittyvät teknisiedot kuvataan bindingTemplate osiossa. Yksi bindingTemplate kuvaa yhden Web Servicen ja ilmaisee, kuinka palvelua käytetään ja miten siihen päästään käsiksi. Jokainen bindingTemplate kuuluu vain yhdelle businessService elementille.

Yleis- tai uusiokäyttöiset palvelun tyypit kuvataan teknisissä malleissa eli tModel osiossa. Osio voi kuvata esimerkiksi palvelun tyyppiä, protokollaa tai kategoria järjestelmää. Tämän tietomallin osa auttaa Palveluiden hakijoita selvittämään, minkälaisesta ja miten käyttäytyvästä palvelusta on kyse. [UDD03]

BusinessEntityjen väliset suhteet esitetään publisherAssertion osassa. Subscription osa kertoo palvelun voimassaoloajan. Kuva 2 esittää tietosisältöjen suhdetta toisiinsa. [UDD03]

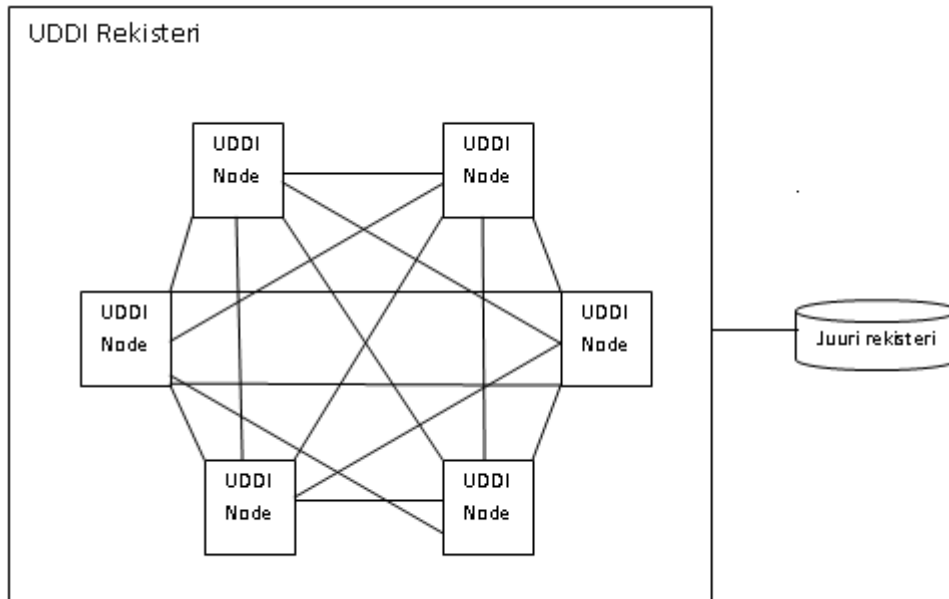


Kuva 2: UDDI tietomallin suhteet [UDD03]

3.2 UDDI Rekisteri

Web Services SOA ratkaisu tarvitsee palveluhakemistoa, jotta palveluja voidaan tarjota ja löytää tehokkaasti. Palveluhakemiston tulisi olla saatavilla ja näkyä käyttäjälle yhtenä ilmentymänä.

UDDI palvelurekisteri rakentuu solmukkeista (node) ja rekisteristä (register). Solmukkeen tehtävänä on tarjota palveluille julkaisu- ja hakuyksikkö. Yhdistelemällä voidaan solmukkeista koota rekisteri. Rekisteri koostuu siis yhdestä tai useammasta solmukkeesta. UDDIn mallissa jokainen rekisterin solmuke tarjoaa saman rajapinnan palveluiden käyttöön ja hakuun. Jokainen solmuke sisältää lisäksi saman rekisterin kaikkien muiden somukkeiden tietosisällön. Näin käytettäessä mitä tahansa rekisterin solmuketta rekisteri esiintyy käyttäjälle yhtenä loogisena kokonaisuutena.



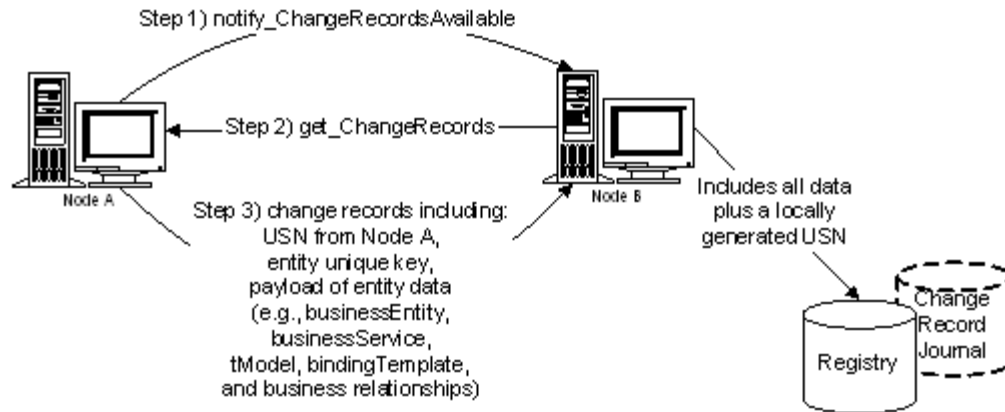
Kuva 3: UDDI rekisteri koostuu solmukkeista

UDDI spesifikaatio määrittelee liittymät rekisterin ja solmukkeiden välille. Se jättää kuitenkin auki sen, miten rekisteri arkkitehtuurin tulisi järjestäytyä [NLI06]. UDDI ei siis ole itsessään tuote vaan spesifikaatio.

3.3 UDDI Replikointi

Solmukkeen perusominaisuuksiin kuuluu kaikkien muiden solmukkeiden tietosisältö rekisterin loogisen kokonaisuuden tarjoamiseksi [UDD03]. Solmukkeen pitää pystyä replikoimaan tietojansa ajoitetusti muiden solmukkeiden kanssa. Lisäksi jokaisen solmukkeen pitää pystyä selviämään virhetilanteista ilman tietojen menetystä.

Muutosten replikointi tapahtuu UDDI rekisterissä käyttäen ilmoitus- ja hakumenetelmää. Solmuke ilmoittaa viestillä kaikille muille rekisterin solmuille, että sillä on muutos tietueita lähetettävänä. Jokainen solmuke päättää itse, koska se replikoi muutokset omaan hakemistoonsa. Solmukkeiden välinen kommunikointi tapahtuu SOAP viestein.



Kuva 4: UDDI replikointi

UDDI replikointi suositellaan käynnistettäväksi ajoituksella, jotta koko rekisterin sisältö päivittyy kaikkien solmukkeiden kesken. UDDI määritelmä suosittelee solmukkeiden välillä kehässä kulkevaa replikointimekanismia. UDDI rekisteriin voi lisätä uusia solmukkeita joustavasti. Uudet solmukkeet lisätään rekisteriin käyttäen UDDI replikointia ja sen tarjoamia rajapintoja.

UDDI solmukkeiden replikointi tuo hakuihin potentiaalisesti viivettä. Koska replikointi on usein ajastettu niin, että ajastus väliyksikkönä on tunti, kaikki muutokset eivät näy kaikille solmuille välittömästi. M. Jeckle ja B. Zengler [MJZ03] ehdottavat UDDI rekisteriin laajennusta Active UDDI välityspalvelulla. Perus UDDI ei tarjoa vaihtoehtoisia palveluja virhetilanteissa, joissa liikennöinti verkkoon tulee muutoksia tai palvelun tarjoaja katoaa verkosta. Active UDDI toteuttaa välityspalvelimen rekisterin ja rekisterin asiakkaiden välillä. Välityspalvelin ylläpitää tietoa rekisterin sisältämistä palveluista. Active UDDIn välityspalvelin myös piilottaa haettaessa ne palvelut, jotka eivät ole enää saatavilla.

UDDI spesifikaatio ei tue suoranaisesti rekistereiden välistä replikointia. Tämä on kuitenkin mahdollista niin, että palvelu tunnistetaan samaksi palveluksi jokaisessa rekisterissä, johon sen tiedot on siirretty. Rekistereiden tunnisteavain logiikka on rakennettu niin, että tämä on mahdollista. Aikaisemmissa UDDI versioissa 1.0 ja 2.0 rekistereiden välinen replikointi oli kielletty.

Ongelmaksi rekistereiden välisessä replikoinnissa muodostuvat palveluiden tunnisteet. Kahdella tunnisteella ei saa olla samaa tunnistetietoa. UDDI spesifikaatio suosittaa ns. Juurirekisterin (root registry) käyttöä (kuva3). Juurirekisterissä ylläpidetään tunnisteita kaikista siihen liittyvistä rekistereistä. Kyseisten rekistereiden väliseen kommunikointiin liittyneet rekisterit voivat tarkastaa juurirekisteristä että sen tunnisteet ovat uniikkeja. Juurirekisteriin liittyneitä rekistereitä kutsutaan liitos rekisteriksi (Affiliate register).

UDDI rekistereiden välisestä replikoinnista on ollut kansainvälinen esimerkki UDDI Business Registry (UBR), jonka tehtävänä oli näyttää että UDDI rekistereiden välinen kommunikointi on vakaa ja kestävä [UMS10]. UBR sisälsi replikoinnin kaikkiaan yli 50 000 yritys ja yhteisö UDDI rekisteriin .

4 Open service discovery architecture, OSDA

Tässä luvussa käsitellään Open service discovery architecture (OSDA) palveluhakemisto mallia. Mallia on ehdottanut ryhmä tutkijoita, jotka yrittävät liittyä yhteen OSDA mallilla monimuotoiset standardi palveluiden löytömekanismit. OSDA mallista ei ole tehty kaupallisia versioita, ja sen toiminta todellisessa maailmassa on testattu vain testiympäristössä. OSDA malli käsitellään tässä työssä, koska UDDI on defacto standardi vain Web Services palveluille ja sen evoluutiossa on selkeitä merkkejä puutteista rekistereiden skaalautuvuudessa ja palveluiden hakumenetelmissä.

Toisin kuin UDDI, OSDA malli ei pyri tarjoamaan hakmistopalvelua ainoastaan HTTP yli toimiville Web Services tyyppisille palveluille. UDDI on vain yksi monesta eri Palveluiden löytö- ja tarjoamismekanismista. Sen sijaan OSDA pyrkii yhdistämään resurssien jako ja löytötekniikoita yhteen palvelumalliin. OSDAn suunnittelussa ovat olleet lähtökohtana muunmuassa vikasietoisuus, suorituskyky, skaalautuvuus, löytömekanismitekniikoiden yhteistoiminta, alusta riippumattomuus, standardit ja valmiin implementaation saatavuus. OSDA ryhmän tavoitteena on luoda yleiskäyttöinen väliohjelmisto palveluhakemistoksi.

Mallin arkkitehtuurissa on käytetty olemassa olevien järjestelmien toimivia löytömekanismeja. Mallin komponentit ovat tilattomia, modulaarisia ja löyhästi kytkettyjä. OSDAn arkkitehtuuri perustuu käytössä oleviin Web standardeihin.

4.1 OSDA Arkkitehtuuri

UDDI palveluhakemisto koostui rekisteristä, joka voi koostua yhdestä tai useammasta solmukkeesta. UDDI malli ei ota huomioon, minkälaisissa ympäristöissä rekisteri sijaitse. OSDA malli pyrkii nimenomaan ratkaisemaan verkkojen ja teknologioiden monimuotoisuuden lähestymällä skaalautuvuusongelmaa kolmitasoisella arkkitehtuurilla.

Korkeimmalla tasolla on peer-to-peer (P2P) teknologiaan perustuva palveluhakemistojen verkosto. Palveluiden julkaisut ja haut suoritetaan tässä verkostossa. Verkosto on tarkoitettu alueiden ja verkkojen väliseen kommunikointiin.

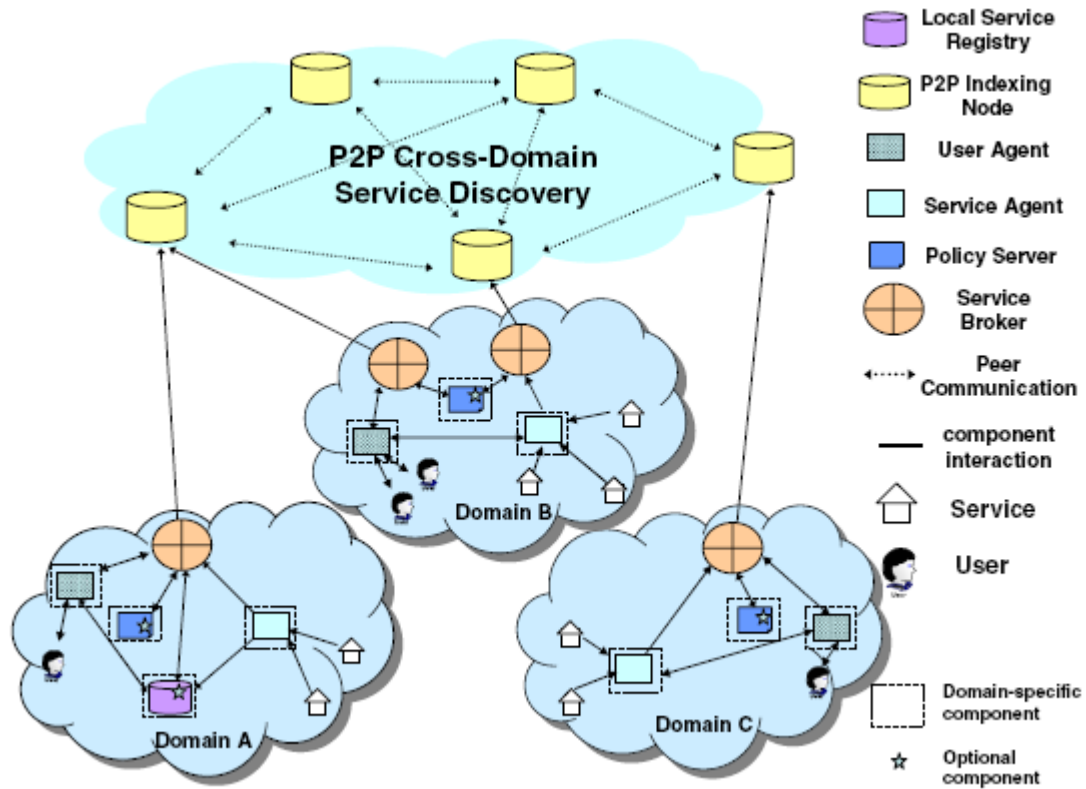
Toisena kerroksen P2P tason alla ovat ohjelmoitavat Palveluiden välittäjät (service broker). Nämä toimivat välityspalvelimien tavoin kuljettaen palvelun tarjous- ja käyttöpyynnöt domain alueiden sisältä julkiseen P2P verkkoon. Palveluiden välittäjät välittävät viestejä paikallisten haku- ja julkaisuagenttien kanssa. Agentit toimivat rajapintana paikallisen teknologian kanssa.

Kolmantena loogisena tasona ovat Palveluiden kuvaukset. Kuvauksissa käytetään Unified Service Description Scheme (USD) tiedostoa, joka on koottu käyttäen XML standardia.

Mallin toiminnalliset komponentit ovat luonteeltaan tilattomia ja löyhästi kytkettyjä. Jotta mallin toteutus olisi mahdollisimman avoin, toteutetaan kaikki mallin komponentit ja niiden välinen kommunikointi käyttäen avoimia standardeja kuten Java, SOAP, XML ja HTTP.

Tosin kuin UDDI:ssä OSDAn, palveluiden kuvausten välitys P2P verkon solmukkeissa ei tapahdu replikoimalla. OSDA malli käyttää muissa P2P verkoissa käytettyä Distributed Hash Tables (DHT) rakennetta. DHT:n ominaisuuksiin kuuluu hajautettu järjestys ilman keskitettyä koordinoitua. DHT:n mukainen arkkitehtuuri pystyy skaalautumaan miljooniin solmukkeisiin

ilman, että se aiheuttaa viestintään ja hakuihin tehottomuutta. Koska solmukkeita voi olla useita tuhansia on DHT kuvattu monissa lähteissä virheille immuuniksi [MNU01].



Kuva 5: OSDA hakemisto arkkitehtuuri

4.2 USD palvelunkuvaus

Kuten UDDIn WSDL ja SOAP viestit, perustuu USD myös XML kieleen. USD viesti sisältää tarvittavat tiedot Palveluiden julkaisuun OSDA verkossa. OSDA viesti sulkee sisäänsä uniikin palvelu tunnuksen (serviceID), joka koostuu yhdistetystä domain- ja palvelutunnuksesta. Domain tunnuksiin suositellaan esim DNS nimiä, jotta ne olisivat uniikkeja. OSDA ei itsessään sisällä mitään keskitettyä nimipalvelua, joka määrittäisi uusille domaineille uniikit tunnuksat. Muita tärkeitä palveluiden hallinnointiin liittyviä tietoja ovat voimassaoloaika ja tarkoitettu saatavuus alue. Itse palvelun kuvaus suositellaan toteuttamaan XML Schema Definition (XSD) mallia käyttäen. Palvelun sisäänpääsy tieto (accessInfo) sisältää kohdeosoitteen

liittymätyypin tai esimerkiksi WSDL tiedoston. palvelun kuvausviestin lasti (Payload) sisältää tiedon palvelun alkuperäisestä teknologiasta ja alkuperäisen palvelun kuvauksen. USD toimii Wrapper -viestinä alkuperäiselle palvelunkuvaukselle. USD mahdollistaa monipuolisten palvelunkuvausten hakemisen OSDA palveluhakemistosta.

5 Yhteenveto

Työssä esiteltiin aluksi tämän hetken Web Services teknologian periaatteet ja standardit XML, SOAP ja WSDL. UDDI rekisteristä käytiin läpi ne yksityiskohdat, jotka ovat ominaisia UDDIn Arkkitehtuurille ja tietosisällölle. OSDAssa pyrin pysymään samalla tasolla kuin UDDIn kuvauksessa, jotta näiden periaate-ero tulisi helposti selville.

Suurin ero UDDIn ja OSDAn välillä on itse käyttötarkoitus. UDDI on käytännön standardi Web Services palveluiden julkaisemiseen Internetissä tai muussa verkko ympäristössä. OSDA on tarkoitettu laajempaan käyttöön palveluiden ja resurssien jakamisessa tehokkaasti domain- ja verkkoalueiden välillä.

UDDIn keskitetty rekisteriratkaisu kärsii hitaasta muutosten leviämisestä replikointiviiveen takia. OSDAssa muutokset P2P verkostossa ovat nopeita, koska sen palvelukuvaukset päivitetään soft-state menetelmällä. Palvelut poistetaan hakemistosta mikäli ne eivät mainosta itseään uudestaan. UDDIssa palveluiden kuvaukset säilyvät rekistereissä kunnes ne poistuvat sieltä ajastetusti.

On mielenkiintoista nähdä, mikä palveluhakemistoarkkitehtuuri ratkaisee tulevaisuudessa ne haasteet, joita tässä työssä on kuvattu.

6 Lähteet

- [ERN02] Understanding Web services: XML, WSDL, SOAP, and UDDI, Eric Newcomer, Addison-Wesley, 2002 ISBN 0201750813, 9780201750812
- [CST07] Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services, Christoph Schroth and Till Janner, IT Pro May,June 2007
- [UDD01] About UDDI, Available at <http://UDDI.xml.org/UDDI-101>, 1.3.2009
- [WIKI01] XML, löydettävissä <http://en.wikipedia.org/wiki/XML>, 8.3.2009
- [WIKI02] SOA, löydettävissä http://en.wikipedia.org/wiki/Service-oriented_architecture, 8.3.2009
- [WIKI03] WSDL, löydettävissä http://en.wikipedia.org/wiki/Web_Services_Description_Language, 13.3.2009
- [WIKI04] OWL, löydettävissä http://en.wikipedia.org/wiki/Web_Ontology_Language, 13.9.2009
- [UDD02] UDDI Tutorials, <http://www.tutorialspoint.com/UDDI/index.htm>
- [UMS01] UBR Shutdown FAQ, löydettävissä <http://UDDI.microsoft.com/about/FAQshutdown.htm> 16.3.2009
- [PHL08] Semantization of Legacy web services: From WSDL to SAWSDL, 978-0-7695-3163-2/08 \$25.00 © 2008 IEEE DOI 10.1109/ICIW.2008.119
- [UDD03] UDDI Version 3.0.2, UDDI Spec Technical Committee Draft, Dated 20041019, löydettävissä http://UDDI.org/pubs/UDDI_v3.htm#_Toc85908159 16.3.2009
- [ASH03] UDDIe: An Extended Registry for Web Services, Ali ShaikhAli, Omer F. Rana, Rash Al-Ali David W. Walker, Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT-w'03) 0-7695-1873-7/03 \$17.00 © 2003 IEEE
- [MJZ03] Active UDDI – An Extension to UDDI for Dynamic and Fault-Tolerant Service Invocation, M. Jeckle and B. Zengler, [Lecture Notes in Computer Science](#) 2003, sivut 91-99
- [NLI06] OSDA: Open service discovery architecture for efficient, Noura Limam a,*, Joanna Ziembicki a, Reaz Ahmed a, Youssef Iraqi a, Dennis Tianshu Li a, Raouf Boutaba a, Fernando Cuervo, cross-domain service www.elsevier.com/locate/comcom Available online 11 January 2006
- [Wiki05] Distributed Hash Table, löydettävissä http://en.wikipedia.org/wiki/Distributed_hash_table, 17.3.2009
- [REA05] Service Discovery Protocols A Comparative Study, löydettävissä <http://bcr2.uwaterloo.ca/~alcatel/publications/IM05/IM05-discovery.pdf>, 17.3.2009
- [MNU01] A Simple Fault Tolerant Distributed Hash Table, Moni Naor_ and Udi Wieder, The Weizmann Institute of Science Rehovot 76100 Israel