

hyväksymispäivä

arvosana

arvostelija

Aine: Service discovery and location transparency in Service-oriented computing

Toni Könnilä

Helsinki 14.10.2015

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Sisältö

1	Abstrakti	1
2	Johdanto	1
3	Palveluorientoituneiden arkkitehtuurien yhteispiirteitä	2
3.1	Monoliittisen ja palveluorientoituneen mallin vertailu	2
3.2	Palveluorientoituneen mallin haasteet	2
4	Palveluiden haku ja läpinäkyvyys	2
5	Yhteenveto	2

Liitteet

1 Malli ABC

1 Abstrakti

Service-oriented computing (SOC) on ajatusmalli, jossa yhden monoliittisen ohjelman sijaan keskitytään keskenään kommunikoiviin palveluihin. SOC nojaakin hyvin vahvasti palveluorientoituneeseen arkkitehtuurimalliin (myöhemmin SOA), joka ajaa tätä ideaa. SOA ei kuitenkaan tarjoa itsessään vastauksia siihen miten turvallisuus, palveluiden koordinointi ja ohjelman muu arkkitehtuuri on rakennettu.

Tässä aineessa keskitytään palveluorientoituneiden arkkitehtuurien yhteisiin piirteisiin ja erityisesti palveluidenhakuun (*service discovery*), ja siihen liittyen, palveluiden läpinäkyvyyteen (*location transparency*).

2 Johdanto

SOC:in on ajatusmalli, joka painottaa eri palveluiden käyttöä palvelun koostamiseksi. Yleensä lopputuloksena on siis palvelu, jonka ideana on kommunikoida muiden palveluiden kanssa, jotka ovat itsessään alustariippumattomia ja väljästi yhteydessä toisiinsa (*loosely coupled*). Business-taustaisena ideana SOA-arkkitehtuureissa (*Service-oriented architectures*) on tuottaa hyvin keskenään toimivia, mahdollisimman atomisia palveluita, jotka ovat uudelleenkäytettäviä, helposti korvattavissa ja ylläpidettävissä tuottaen nopeasti kustannustehokkaita hajautettuja palveluja lyhyellä kehitys-julkaisuikkunalla (*time-to-market*).

Yksittäisen palvelun tehtävä voi olla yksittäinen pyyntö, tai monimutkaisempi businessprosessi, ja tällaisten palveluiden tarjoajat (olipa ne sitten organisaatioita tai mitä vain) mahdollistavat ohjelmallisen pääsyn palveluihinsa ja dataansa internetin yli käyttäen standardoituja välitystapoja, käyttäen yleensä XML-perustaisia tai REST-mallin mukaista HTTP-protokollaan perustuvia rajapintoja käyttäen. Näiden palveluiden tarjoajien (*service providers*) vastuulla on antaa palvelukuvaus ja ylläpitää palvelua, ja antaa palveluun liittyvää teknistä sekä muuta tukea. Palvelua kutsuvat tahot voivat olla muita ohjelmia, tai yksittäisiä tahoja, esimerkiksi kotipalvelin joka pyytää resurssia REST-rajapinnan toteuttavalta palvelulta internetin yli. Tästä seuraa, että näillä näillä palveluilla on tiettyjä vaatimuksia ja edellytyksiä.

- *Teknologiariippumattomuus*: Palvelun ei kuulu olla liian teknologiariippuvainen, vaan sitä pitää pystyä kutsumaan alalla vallitsevilla työkaluilla ja alan standardien mukaisesti.

- *heikko linkittyneisyys (loosely coupled)*: palvelun tulee olla irti mahdollisesta kontekstista, ja toimia itsenäisesti (ainakin ulospäin) ilman tietoa muista palveluista tai komponenteista.
- *Palvelun haettavuus (service discovery) ja tuki sijaintiläpinäkyvyydelle (location transparency)*: Palveluiden määrittelyt ja varsinaiset sijaintitiedot tulee olla tallennettuna johonkin - esimerkiksi UDDI[footnote tms. plz] repositorioon, ja asiakkailla/tahoilla pitää olla pääsy palveluun tietämättä missä se varsinaisesti sijaitsee.

3 Palveluorientoituneiden arkkitehtuurien yhteispiirteitä

Palveluorientoituneet arkkitehtuurit toteuttavat tätä SOC:in ideaa, mutteivat ota kantaa esimerkiksi turvallisuuteen, transaktioiden hallintaan tai koordinaatioon [viite papazoglouhun]. Tästä syystä onkin kehitelty xSoa pyramidi [oisko liite - papazoglou pyramid] joka ottaa nämä huomioon.

3.1 Monoliittisen ja palveluorientoituneen mallin vertailu

3.2 Palveluorientoituneen mallin haasteet

4 Palveluiden haku ja läpinäkyvyys

5 Yhteenveto

Liite 1. Malli ABC

Liitteet ovat tässä vain sisällysluettelon ja esitystavan mallina. Jokainen liite aloitetaan yleensä uudelta sivulta, jonka alkuun tulee liitteen numero ja nimi. Kunkin liitteen sivut numeroidaan erikseen.

Liite on paitsi dokumenttia täydentävä osuus myös itsenäinen kokonaisuus. Liite ei siten voi olla pelkästään kuva tai ohjelmanpätkä, vaan liitteessä on ilmaistava sen sisällön laatu ja tarkoitus.