

Palvelu- vai integraatioarkkitehtuuri? Vai molemmat?

Petteri Manninen, SysOpen

Artikkelissa pohditaan integraatioarkkitehtuurien ja palveluarkkitehtuurien yhteyttä toisiinsa sekä mahdollisuuksia hyödyntää integraatioarkkitehtuuria palveluarkkitehtuuria rakennettaessa.

Mitä ovat palveluarkkitehtuuri ja integraatioarkkitehtuuri

Palveluarkkitehtuuri (Service-Oriented Architecture, SOA) on hajautetun tietojenkäsittelyn ja integraation arkkitehtuurimalli, jonka keskeinen ajatus on tarjota korkean tason liiketoimintapalveluja helposti kutsuttavan rajapinnan kautta. Tavoitteena on muodostaa uusia palveluja koostamalla ne olemassa olevista palveluista ja siten reagoida nopeammin muuttuviin liiketoiminnan tarpeisiin. Viime aikoina suosituin malli on ollut Web Services + UDDI-hakemisto, joiden avulla palvelut on kutsuttavissa välineriippumattomasti olemassa olevia standardeja käyttäen. Vaikka palveluarkkitehtuurimalli on ollut olemassa

kauan (esim. pankkien keskusko-
nepohjaiset transaktioympäristöt),
sen suosio on kasvanut vasta nyt,
kun käytettävät välineet on stan-
dardoitu. Aiemmat hajautetun
tietojenkäsittelyn mallit, kuten
DCE ja CORBA, eivät ole saavut-
taneet vastaavaa suosiota lähinnä
teknologiavetoisuutensa ja toimit-
tajakohtaisten ratkaisujensa takia.

Integraatioarkkitehtuuri on yhte-
näistetty tapa liittää sovelluksia,
niiden osia ja tietoja toisiinsa.
Integraatiolla tarkoitetaan sovel-
luksille tarjottavaa yhtenäistä
tapaa käsitellä jo olemassa olevia
palveluja ja tietoa. Integraatioark-
kitehtuuriprojektien lopputulok-
sena syntyy kuitenkin malli, jonka
avulla voidaan koostaa uusia palve-
luja vanhoja yhdistämällä. Lisäksi

kokonaan uusienkin palvelujen
luominen helpottuu saman mallin
avulla.

Puhtaalta pöydältä lähdettäessä
palveluarkkitehtuuri on helppo
toteuttaa. Sen sijaan olemassa
olevan sovelluskannan muok-
kaaminen palveluja tarjoaviksi
komponenteiksi on vaativa
mutta välttämätön prosessi, jotta
palveluarkkitehtuurin rakenta-
misessa päästään edes alkuun.
Palveluarkkitehtuurista puhutta-
essa usein korostetaan pelkästään
palvelurajapintana toimivaa Web
Services -toteutusta, joka kuitenkin
on vain fasaadi todellisille palve-
luille ja siten yksinkertaisin osuus
palvelun rakentamisessa. Vaativin
osuus, todellisen palvelun suunnit-
telu ja toteutus "jätetään lukijalle".

Tavoitteet

Molempien arkkitehtuurimallien keskeinen tavoite on joustavuus: palvelujen yhtenäinen ja helppo kutsutapa sekä mahdollisuus reagoida nopeasti liiketoiminnan muutoksiin.

Palveluarkkitehtuurin keskeisiä teemoja ovat:

- korkean abstraktiotason sovelluspalvelut
- palvelujen löyhät liitokset
- sovellusten ja palvelujen komponentointi
- komponenttien uudelleenkäytettävyys
- palvelujen korkeampi käytettävyys
- palvelujen parempi hallittavuus
- nopea reagointi liiketoiminnan muutoksiin

Vaikka palveluarkkitehtuurin tavoitteet ovat luonteeltaan yleisiä ja sopisivat minkä tahansa arkkitehtuurin tavoitteiksi, on huomattava, että painopistealueena ei ole ajonaikainen suorituskyky vaan toteutuksen helppous. Huonompaa suorituskykyä voidaan kompensoida palvelujen korkealla tasolla, jolloin kutsujen määrä jää pieneksi.

Integraatioarkkitehtuurin keskeisiä teemoja ovat:

- olemassa olevien järjestelmien yhteenliittäminen
- joustavuus, varautuminen tuleviin tarpeisiin
- palvelujen löyhät liitokset

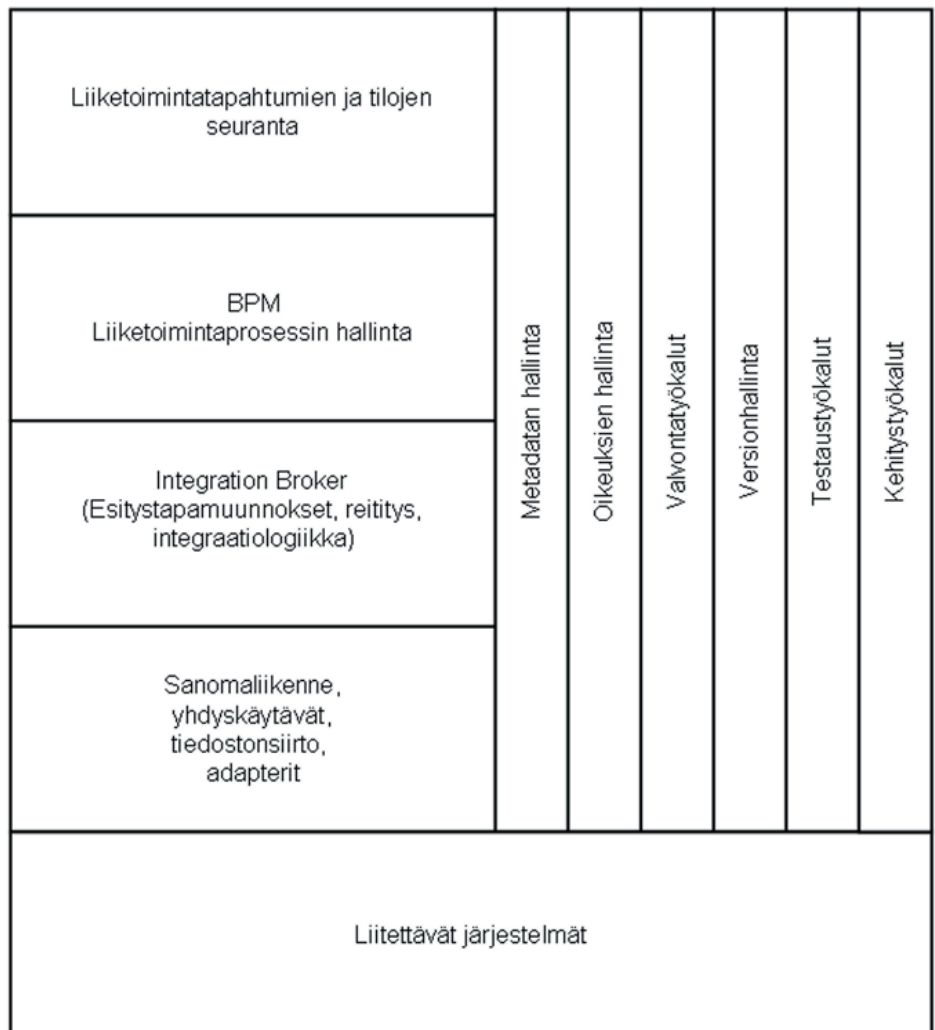
- monimutkaisuuden vähentäminen rajapintojen yhtenäistämisen kautta
- hallittavuus (keskitetty valvonta ja hallinta)
- korkea käytettävyys
- skaalautuvuus
- turvallisuus (yksittäisten palvelujen tasolla keskitetysti hallittuna)
- tiedon eheys
- ratkaisujen pitkät elinkaaret, komponenttien elinkaaret eivät ole sidottuja toisiinsa
- toteutus- ja tuotantokustannusten vähentäminen pitkällä aikajänteellä

Integraatioarkkitehtuurin teemat ja tavoitteet ovat luonteeltaan

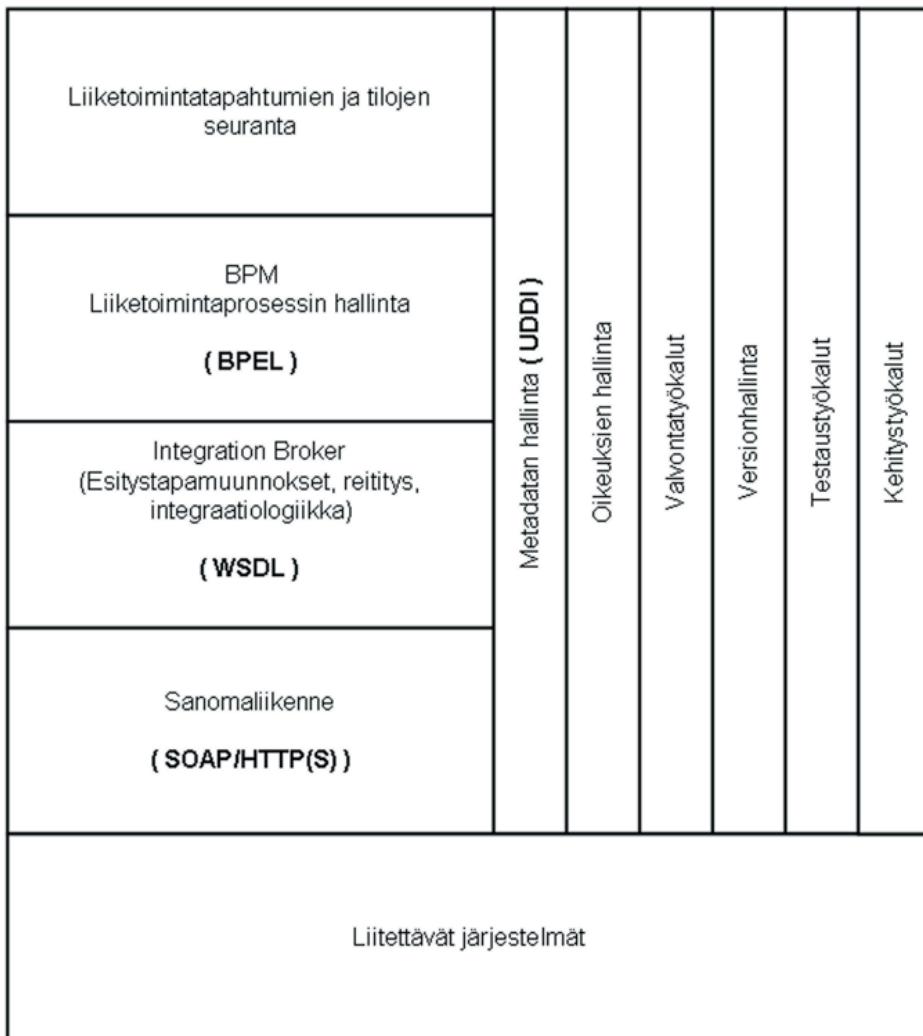
teknisempiä ja pyrkivät kriittisille järjestelmille tyypilliseen toimintavarmuuteen. Integraatioarkkitehtuurissa peruspalveluista koostetaan korkeamman tason palveluja käyttämällä integraatiotuotteiden workflow-ominaisuuksia.

Rakenne

Palvelu- ja integraatioarkkitehtuurien rakenteet ovat hyvin samankaltaisia, koska palveluarkkitehtuuria voidaan pitää yleisen integraatioarkkitehtuurin yhtenä erikoistapauksena. Palveluarkkitehtuurissa on kuitenkin tehty rajoituksia, joita ei yleisessä integraatioarkkitehtuurissa voida tehdä.



Kuva 1: Integraatioarkkitehtuurin rakenne



Kuva 2: Palveluarkkitehtuuri sovitettuna integraatioarkkitehtuuriin

Kuvassa 1 on esitetty tyypillinen integraatioarkkitehtuurin rakenne. Alimmalla tasolla on kuvattu liitettävät järjestelmät, joiden päälle integraation tarvitsemat kerrokset on rakennettu. Kahta ylintä kerrosta (liiketoimintaprosessin hallinta ja seuranta) ei aina toteuteta, mutta ne tulee suunnitellussa huomioida. Kaavion oikeaan reunaan kuvatut tukitoiminnot eivät suoraan vaikuta yksittäisen palvelupyynnön käsittelyyn, mutta ovat mm. hallittavuuden takia pakollisia.

Vastaava palveluarkkitehtuurirakenne on esitetty kuvassa 2 täydennettynä kuljetuskerroksen (SOAP/HTTP(S)), esitystavan (XML),

palvelun kutsumisen (WSDL) ja hakemiston (UDDI) osalta.

”...erot löytyvät tarkasteltaessa yksittäisten pyyntöjen kulkua järjestelmän läpi.”

Suorituspolku

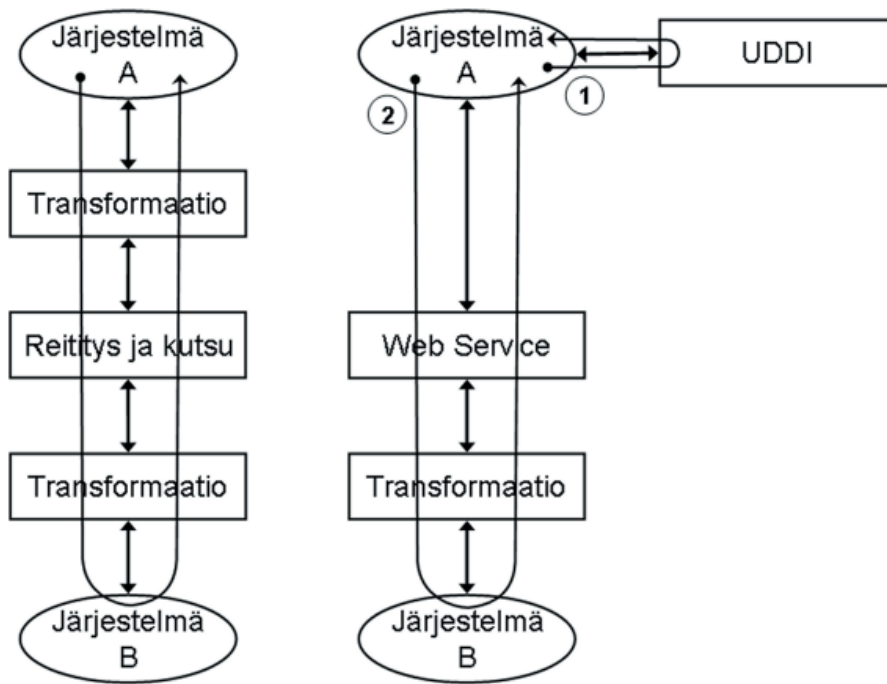
Vaikka arkkitehtuurit vaikuttavat täysin yhteneväsiltä, erot löytyvät tarkasteltaessa yksittäisten pyyntöjen kulkua järjestelmän läpi (Kuva 3). Integraatiojärjestelmä toimii jokaisella pyynnöllä samalla tavoin. Kaikki pyynnöt ja vastaukset kulkevat transformaatio- ja reiti-

tyskerrosten kautta, koska osapuolilla ei ole tietoa toisen osapuolen rakenteesta, tiedon esittämistä vasta, käytetystä kuljetuskerroksesta jne. Toinen osapuoli voi yhtä hyvin olla saman järjestelmän osa samassa koneessa tai vaikka älykäs mobiilipäätte julkiverkon takana toisella puolella maata.

Palveluarkkitehtuurissa palvelua pyytävä osapuoli on standardoitu, joten pyyntö tulee aina sovitussa muodossa. Tämä antaa mahdollisuuden keventää reititystä ja transformaatioita, jolloin palvelua tarvitseva voi pyytää palvelun sijainnin hakemistosta ja kutsuu sitä suoraan ilman reititystä. Vaikka palvelun pyytjä ei transformaatioita ja reititystä tarvitse, on palvelun toteutettava osapuoli usein rakennettu siten, että se kutsuu useita taustajärjestelmiä ja niiden palveluita.

Molemmissa tapauksissa pyydetty palvelu voi itsessään olla synkroninen tai asynkroninen, mutta Web Services -toteutukset ovat luonteeltaan synkronisia, jolloin asynkro-

nisen rajapinnan tuominen palvelurajapinnaksi aiheuttaa pollaamista. Jotta tavoite nopeasta reagoinnista liiketoiminnan muutoksiin toteutuisi, on palveluiden kehittäminen ja muuttaminen oltava yksinkertaista. Ilmeinen ratkaisu tähän tarpeeseen on integraatiojärjestelmä ja sen hyödyntäminen palveluiden koostamisessa.



Kuva 3: Pyynnön suorituspolku molemmissa arkkitehtuurimalleissa

Yhteenveto

Palveluarkkitehtuuri on täsmälleen yhtä hyvä ja joustava kuin komponentit, joista palvelut muodostuvat. Työkalut korkean abstraktiotason palvelukomponenttien koostamiseen nopeasti ja helposti tarjoaa hyvin suunniteltu integraatioarkkitehtuuri.

Palveluarkkitehtuuri on liiketoi-

komponentit, jotka muodostavat ko. palvelut, koodataan tiukoin sidoksin, jolloin uusien palveluiden luonti hankaloituu ja vanhojen palveluiden ylläpidettävyys kärsii. Tällöin integraatioarkkitehtuuria hyödyntämällä voidaan nämä liitokset muuttaa löyhiksi.

Integraatioarkkitehtuurin kannalta on sama, onko kutsuja sovellus, toinen komponentti vai palvelu-

”Palveluarkkitehtuuri on integraatioarkkitehtuurin erikoistapaus, jossa käytettävien tekniikoiden määrää on rajattu.”

mintälähtöinen ja sen tavoitteena on nopea reagointi muuttuviin liiketoimintaolosuhteisiin. Tämä toteutetaan yleensä siten, että sovelluksen palvelut on liitetty toisiinsa löyhin liitoksin. Usein kuitenkin

komponentti. Perinnejärjestelmien tarjoamat palvelut vaativat joka tapauksessa rajapintoja, joita ei kannata ratkaista ainoastaan palveluarkkitehtuurintarpeisiin. Vaarana on ratkaista tarvittavat rajapinnat

siten, että koosteisten palveluiden tuottaminen myöhemmin osoittautuu vaikeaksi.

Web Services -toteutusten avulla tapahtuva palveluiden jakaminen useaan organisaatioon laajentaa tiettyyn palveluun sitoutuneiden käyttäjien määrää ja samalla vaikuttaa ylläpidon helppouteen/vaikeuteen ja ylläpitoonopeuteen. Hyvällä suunnittelulla estetään lukkiutuminen tiettyyn palveluun ja sen versioon ja pystytään tukemaan useita rinnakkaisia versioita palvelusta. Samoin on oltava mahdollista muuttaa palvelun toteutusta ilman että muutos näkyy palvelun käyttäjille. Vaikka tämä on mahdollista perinteisinkin tekniikoin, vähentää integraatioarkkitehtuuri tarvittavia työmääriä merkittävästi. Vaikka palveluarkkitehtuuri ei ole integraatioarkkitehtuuri, siitä muodostuu integraation merkittävä käyttäjä.

Palveluarkkitehtuuri on integraatioarkkitehtuurin erikoistapaus, jossa käytettävien tekniikoiden määrää on rajattu. Vain integraatioarkkitehtuurin hyvällä suunnittelulla on palveluarkkitehtuurin käyttöönotto mahdollista.