

hyväksymispäivä arvosana

arvostelija

## **Liikepalveluväylä ja sen merkitys palvelusuuntautuneissa järjestelmissä**

Antti Saukko

Helsinki 8.2.2009

Seminaarityö

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Tiedekunta/Osasto — Fakultet/Sektion — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen tdk		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author			
Antti Saukko			
Työn nimi — Arbetets titel — Title			
Liikepalveluväylä ja sen merkitys palvelusuuntautuneissa järjestelmissä			
Oppiaine — Läroämne — Subject			
Tietojenkäsittelytiede			
Työn laji — Arbetets art — Level		Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Seminaarityö		8.2.2009	17 sivua + 0 liitesivua
Tiivistelmä — Referat — Abstract			
<p>Liikepalveluväyläkonsepti on yritys integroida heterogeeniset palvelut ja sovellukset yhteisen väyläarkkitehtuurin varrelle. Palveluväyläarkkitehtuuri abstrahoi alla olevat tiedonsiirtoprotokollat ja kykenee hyväksikäyttämään näitä viestien välityksessä. Liikepalveluväylä kykenee muuttamaan yhteensopimattomat tietotyypit yhteensopiviksi hyväksikäyttäen palveluiden metainformaatiota.</p> <p>Suurimpia liikepalveluväylän etuja edeltäneisiin integraatiotekniikoihin ovat sen aikaisempaa parempi kyky skaalautua, sekä tukeutuminen olemassa oleviin palvelusuuntautuneiden järjestelmien arkkitehtuurimalleihin ja täten myös yleisesti käytettyihin standardeihin.</p> <p>ACM Computing Classification System (CCS):  D.2 [Software Engineering],  D.2.11 [Software Engineering, Software Architectures]</p>			
Avainsanat — Nyckelord — Keywords			
Liikepalveluväylä, palveluväylä, palvelusuuntautuneet järjestelmät, ESB			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — Övriga uppgifter — Additional information			

# Sisältö

<b>1 Johdanto</b>	<b>1</b>
<b>2 Liikepalveluväylä palvelusuuntatuneessa järjestelmässä</b>	<b>2</b>
<b>3 Liikepalveluväylän toiminnot</b>	<b>4</b>
3.1 Kommunikointi väylässä . . . . .	6
3.2 Integraatio-elementit . . . . .	11
3.3 Hallintamenetelmät . . . . .	14
<b>4 Yhteenveto</b>	<b>15</b>
<b>5 Loppusanat</b>	<b>16</b>
<b>Lähteet</b>	<b>17</b>

# 1 Johdanto

Nykyajan tärkeimpiä trendejä teollisuudessa ovat järjestelmien yhteensopivuus ja niiden mukautuvuus jatkuvasti muuttuvaan ympäristöön. Järjestelmät ovat heterogeenisiä ja niitä on toteutettu useamman eri teknologisen suuntauksen aikana. Liikepalveluväylä (*ESB, Enterprise Service Bus*) on yritys rakentaa silta moninaisten sovellusten välisille kommunikaatioille ja mahdollistaa parempi, dynaamisempi liiketoimintavuo hyödyntäen olemassa olevia sovelluksia ja teknologioita. Liikepalveluväylä on evoluutioaskel palvelusuuntautuneiden järjestelmien paremmassa integroinnissa (*SOA, Software Oriented Architecture*) ja se pyrkii jatkamaan siitä, mihin aikaisemmat yhteensovittamisarkkitehtuurit ovat jääneet.

Liikepalveluväylä on palvelusuuntautuneiden järjestelmien yksi kulmakivistä, ja se on liki välttämätön osa palvelusuuntautuneiden järjestelmien kokonaisuutta tuoden tärkeitä lisäarvoa tuottavia palveluja ohjelmistoketjuun. Liikepalveluväylän pääominaisuuksiin kuuluvat kyky hyödyntää olemassa olevia ohjelmistoja, kyky kommunikoida sovellusten välillä, löytää palveluja dynaamisesti, reitittää viestejä kehittyneesti, muokata viestin sisältöä ja rakennetta, kyky skaalautua, mahdollisuus luotettavaan viestintään, transaktioiden hallinta ja korkeamman tason ohjelmistojen hallinta [PvdH05].

Liikepalvelujenväylä pyrkii ohjaamaan yhdenmukaisuuteen sekä suosittelemaan avoimia ja hyväksi havaittuja standardeja. Vaikka liikepalveluväylä ei pyri spesifioimaan yhteistä implementaatiostandardia, on yritysten samankaltaisten tarpeiden myötä suuri osa yhteiskäyttöisistä komponenteista jokseenkin standardoitu.

Järjestelmien mukautuminen globaaliin talouteen, uudenlaiset tiedon tuottamis- ja jalostusmallit, sekä kyky hyödyntää olemassa olevaa infrastruktuuria vaativat palvelusuuntautuneita järjestelmiä. Liikepalveluväylä on tämän päivän ratkaisu mahdollistaen sovellusten välisen yhteistoiminnan voimakkaasti hajautetuissa järjestelmissä.

Tämä seminaarityö tutkii abstraktin liikepalveluväylän toiminnallisia vaatimuksia, vertailee niitä jo käytössä oleviin tekniikoihin ja pyrkii antamaan yleiskuvan liikepalveluväylän pyrkimyksestä tulla palvelusuuntautuneiden järjestelmien kulmakiveksi. Seminaaripaperin lopussa vedetään yhteen esitellyt tekniikat ja spekuloidaan liikepalveluväylän todellista merkitystä.

## 2 Liikepalveluväylä palvelusuuntauneessa järjestelmässä

Palvelusuuntautunut tietojenkäsittely (*SOC, Service-oriented Computing*) on arkkitehtuurimalli, jossa sovelluksen muodostaa sarja hajautettuja palveluja [Pap03]. Palvelusuuntautunut tietojenkäsittely tukeutuu abstraktiin hajautettujen järjestelmien malliin, jota kutsutaan myös palvelusuuntauneeksi arkkitehtuuriksi (*SOA, Service-oriented Architecture*).

Palvelusuuntautuneille järjestelmille ja arkkitehtuureille fundamentaalisia yhdistäviä tekijöitä ovat nykyiset web-teknologiat ja hyväksi havaitut standardit. Web -teknologioista tärkeimpinä mainittakoon<sup>1</sup> XML (*Extensible Markup Language*) -teknologiat: XSD (*XML Schema*), XSLT (*Extensible Stylesheet Language Transformation*), XQuery ja XPath, HTTP (*Hypertext Transfer Protocol*) ja SOAP (*Simple Object Access Protocol*). Hyväksi havaituilla yhteisillä teknologioilla on tarkoitus mahdollistaa parempi palvelujen yhteensopivuus keskenään. Palvelusuuntauneet järjestelmät pyrkivät täten voimakkaasti abstrahoimaan alla olevat suoritinarkkitehtuurit, käyttöjärjestelmät ja ohjelmointikielet.

Jotta palvelut olisivat palvelusuuntauneiden arkkitehtuurien mukaisia, tulisi niiden olla teknologia neutraaleja, löyhästi sidottuja (*loosely coupled*) ja sijaintituntemattomia (*location transparent*) [Pap03]. Näillä kolmella tavalla pyritään takaamaan, että tekniset ratkaisut olisivat pienimmän yhteiset nimittäjän määrittelemiä, ja täten mahdollisia toteuttaa millä tahansa ohjelmistoalustalla. Esimerkkinä pienimmän yhteisen teknologian tekijästä on esimerkiksi yleisesti käytetty tiedonsiirto-protokolla HTTP.

Sidonnalla ohjelmistokomponenteissa tarkoitetaan kahden toistensa kesken kommunikoivan komponentin välistä rajapintojen riippuvuutta [Har05]. Löyhällä sidonnalla palvelusuuntauneissa järjestelmissä tarkoitetaan myös monesti kommunikaatioformaattiriippumattomuutta, joka on olennainen osa liikepalveluväylä konseptia. Löyhä sidonta edesauttaa myös sijaintituntemattomuuden saavuttamista, joka on myös tärkeä määrittelevä tekijä palveluväylistä puhuttaessa. Sijaintiriippumattomuudella pyritään saavuttamaan tiedostamattomuus todellisen palvelun fyysisestä sijainnista. Tällä ominaisuudella mahdollistetaan myös tehokas palvelujen hajautus.

Palvelusuuntautunut arkkitehtuuri on siis implementaatiostrategia, ei teknologia [DSRR08].

---

<sup>1</sup>Protokollat luettavissa osoitteessa <http://www.w3.org/>

Palvelusuuntautuneissa järjestelmissä on kyse useamman hajautetun palvelun yhteistoiminnasta. Jokainen yksittäinen palvelu suorittaa sille mielekkään osakokonaisuuden, pyrkien noudattamaan palvelusuuntautuneiden järjestelmien arkkitehtuuria siten, että samaista palvelua voidaan käyttää uudelleen seuraavia tuotteita toteuttaessa. Yksittäinen palvelu voi olla sekä palvelun tuottaja, että palveluiden kuluttaja.

Palvelusuuntautuneiden järjestelmien rakentaminen ilman yhteistä integrointi- ja kommunikaatioväylää on ongelmallista. Kun yritykset haluavat ottaa käyttöön jo olemassa olevat sovellukset, sovellustenvälinen kommunikaatio voi pohjautua standardeihin, mutta toiset sovellukset/palvelut voivat puhua eri standardeilla. Liikepalveluväylä pyrkii ratkaisemaan tämän ongelman mallintaen arkkitehtuurin, joka on syntynyt palvelusuuntautuneiden järjestelmien soveltajien, eli yritysten, tarpeesta, ei pelkästä akateemisesta mielenkiinnosta [Cha04].

Palvelujen välinen kommunikaatio on usein myös tapahtumapohjaista (*event-driven*) ja tästä johtuen palveluille muodostuu vaatimuksia kommunikoida toistensa kesken asynkronisesti [PvdH05]. Asynkronisuus tarkoittaa ennalta-arvaamatonta tapahtumaketjua, jossa tapahtumaa ei olla erikseen tilattu. Tapahtumapohjaisella kommunikaatiolla palvelusuuntautuneissa järjestelmissä voidaan tarkoittaa myös tapahtumia, joissa suoritusvuossa yhtenä tekijänä on ihminen. Tästä hyvänä esimerkkinä toimii tapahtuman hyväksyntäprosessi, jossa on välttämätöntä saada hyväksyntä lainvoimaiselta tekijältä, ihmiseltä.

Yhtenäisen liikepalveluväyläkonseptin (*ESB, Enterprise Service Bus*) yksi tarkoituksista on mahdollistaa erilaisten sovellusten välinen kommunikointi abstrahoimalla muut viestipohjaiset väliohjelmistot (*MOM, Message-oriented Middleware*), sekä määrittämällä yhteisen kommunikaatiomekanismin. Liikepalveluväylä toimii siis liimana, joka mahdollistaa palvelusuuntautuneiden heterogeenisten järjestelmien yhteistoiminnan.

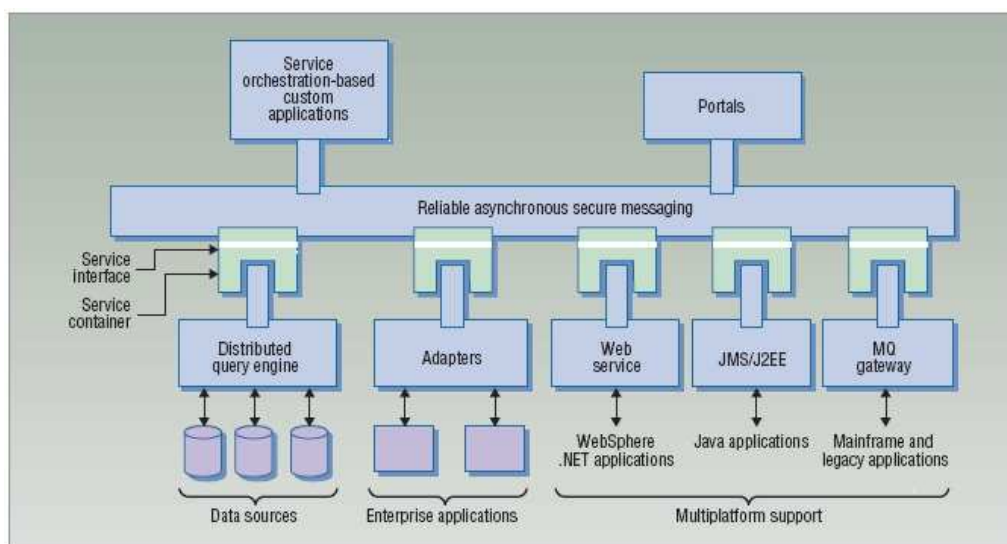
Vaihtoehtoinen lähestymistapa väylälle olisi ratkaista teknologioiden ja tietomallien yhteensopivuusongelma pakottamalla kehittäjät noudattamaan tarkasti yrityksen valitsemaa teknologiaratkaisuja [PvdH05]. Tässä lähestymistavassa törmätään hyvin nopeasti ongelmiin, jos halutaan integroitua esimerkiksi muiden palveluntarjoajien järjestelmiin.

Liikepalveluväyläkonsepti pyrkii viemään eteenpäin palvelusuuntautuneiden järjestelmiä kuvaamalla nopeamman ja paremman tavan adaptoitua uusiin ohjelmisto- ja haasteisiin ennemminkin ohjelmistoja konfiguroimalla, kuin niitä ohjelmoimal-

la [DSRR08].

### 3 Liikepalveluväylän toiminnot

Liikepalveluväylä on rikastettu kommunikaatiöväylä, joka pyrkii mahdollistamaan olemassa olevien sovellusten ja palveluiden integroimisen, sekä uusien lisäarvopalveluiden helpon käyttöönoton. Liikepalveluväyläkonsepti perustuu täysin avoimiin standardeihin ja se mahdollistaa palveluiden kehittämisen, käyttöönoton ja hallinnan [SHLP05] palvelusuuntauneissa heterogeenisissä järjestelmissä. Yhteistä soveltua standardimääritelmää liikepalveluväylälle ei ole olemassa, ja tästä syystä se jakaa vielä tänäkin päivänä ihmisten mielipiteitä asiasta: *"Onko liikepalveluväylä todellinen?"*.



Kuva 1: Liikepalveluväylälle on monta nimeä, useimmiten siihen viitataan vain palveluväylänä tai kuvan tapauksessa luotettavaksi, tietoturvalle asynkroniseksi viestiväyläksi.

Liikepalveluväylä -termin esitti ensikertaa Roy Schulten, Gartner -nimisestä teknologiakonsulttiyrityksestä joulukuussa vuonna 2002 [Cha04]. On myös mahdollista että nimen on keksinyt hieman aikaisemmin Sonic Software- nimen yritys. Näin asian ilmaisee Microsoftin<sup>1</sup>. Vaikka liikepalveluväylä perustuukin avoimille standardeille,

<sup>1</sup><http://msdn.microsoft.com/en-us/library/aa475433.aspx>

suurin osa ensimmäisistä liikepalveluväylistä oli kaupallisia. Ensimmäiset liikepalveluväylät, jotka olivat myös kaupallisia, olivat Sonic ESB, Fiorano ESB, CapeClear ja PolarLake Integration Suite [HTL05].

Kuvassa 1 nähdään liikepalveluväylä ja siihen kytkeytyneitä palveluja. Liikepalveluväylä pyrkii määrittelemään palvelusuuntautuneiden järjestelmien integraatioinfrastruktuurin, uuden sukupolven väliohjelmiston ja arkkitehtuurimallin.

Sen lisäksi, että liikepalveluväylä mahdollistaa sovellusten ja palveluiden välisen kommunikaation, se mahdollistaa myös muita tärkeitä lisäarvopalveluja. Lukuisat lähteet [Cha04, PvdH05, IBM04, SHLP05, OJ07] määrittelevät liikepalveluväylille seuraavia merkittäviä ominaisuuksia:

- Kommunikaatiokyvykkyydet
  - Viestityskyvykkyydet
  - Reitityskyvyys
  - Yhteinen kommunikointikäytäntö
- Integraatio-elementit
  - Järjestelmä-adapterit
  - Tietomallimuunnokset
- Hallinta
  - Mahdollisuus muokata palvelun toimintaa (konfigurointi)
  - Palveluiden huomaamaton vaihto

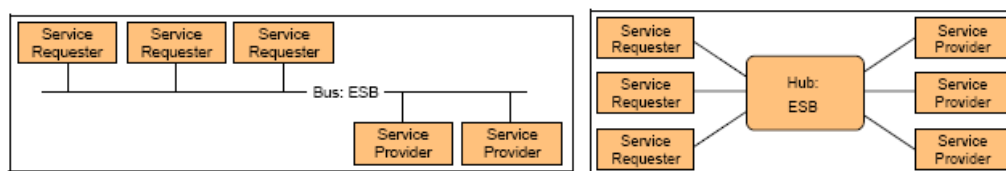
Liikepalveluväylä voidaan nähdä evoluutioaskeleena aikaisemmille sovellusten integraatiomalleille. Aikaisemmissa bisnessovellusten integraatioissa puhuttiin yleensä kaupallisista EAI -ratkaisuista (*Enterprise Application Integration*). Korkean tason funktionaalisia toiminnallisuuksia vertaillessa liikepalveluväylällä ja EAI -mallilla on vain vähän eroavaisuuksia. Merkittävimmät eroavaisuudet EAI -tyyppisellä ratkaisulla ja liikepalveluväylällä on arkkitehtuuri. Siinä missä EAI -tyyppiset arkkitehtuurit pohjautuvat usein Hub-and-Spoke -arkkitehtuuriin [DSRR08], liikepalveluväyläarkkitehtuuri tukeutuu väyläarkkitehtuuriin (kuva 2). *Hub-and-Spoke* -arkkitehtuurissa kaikki bisneskriittinen toiminta kulkee yksittäisen keskitetyn pisteen kautta. Keskitetyn pisteen ongelma on sen haavoittuvuus. Mikäli kyseinen piste



ei suoriudu tehtävästä, koko järjestelmä on käyttökelvoton. Suuremmissa järjestelmissä keskitetystä pisteestä tulee myös helposti järjestelmän pullonkaula. Väyläarkkitehtuurissa (*bus architecture*) pyritään ratkaisemaan tämä ongelma siten, että bisneskriittistä yksittäistä pistettä ei ole. Integroituneet järjestelmät toimivat hajautetusti ja mahdollistavat täten paremman laajennettavuuden.

Toinen merkittävä ongelma EAI -ratkaisussa on niiden epästandardillisuus. Kommunikaatio ja integraatiomekanismit ovat kaupallisten yritysten tapa sitoa asiakas tietyn yrityksen tuotteisiin. Hinnakkaan integroinnin seurauksena, ei siirtyminen toisen yrityksen tuotteisiin ole monesti mahdollista. Liikepalveluväyläkonseptin on tarkoitus tukeutua ainoastaan standardeihin rajapintoihin, ja täten mahdollistaa tarvittaessa helppo siirtyminen tuotteiden välillä [Cha04].

Edellisistä argumenteista huolimatta liikepalveluväylä on saanut suurimman osan sen toiminnallisista määritelmistään EAI -ratkaisusta. EAI -ratkaisuiden integraatiokomponentti (*integration broker*) on toiminnallisuudeltaan hyvin samanlainen kuin liikepalveluväyläkontti (*service container, kuva 1*). Näiden komponenttien vastuulla on molemmissa järjestelmissä mm. palveluihin kytkeytyminen, viestien reititys ja niiden tulkkaukset [PvdH05].

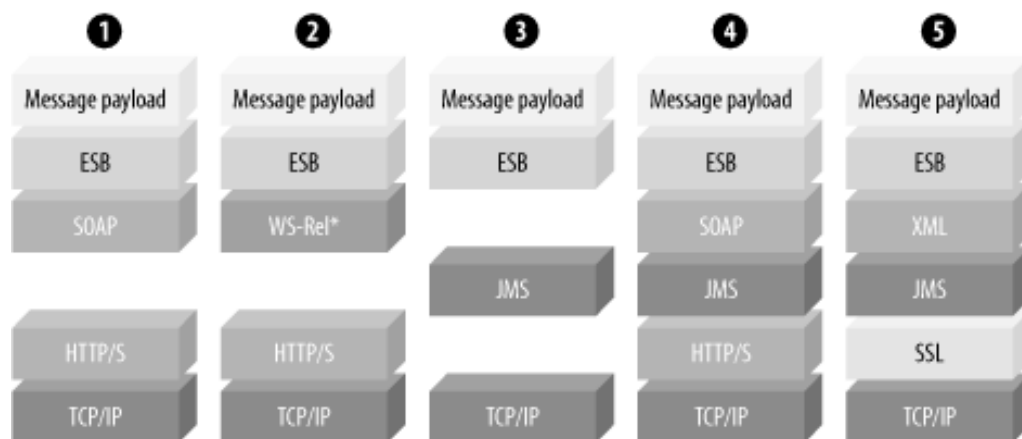


Kuva 2: Väylä ja keskitetty hub-and-spoke -arkkitehtuuri.

### 3.1 Kommunikointi väylässä

Järjestelmien heterogeenisyyden vuoksi, liikepalveluväylän yksi tärkeimmistä tehtävistä on abstrahoida alla oleva viestinvälitysjärjestelmä. Liikepalveluväylän tarkoitus on siis kyetä kommunikoimaan useamman erilaisen järjestelmän kanssa. Väylään liittymisen edellytyksenä on yhtenäinen rajapinta (*service interface*) sovellus- ja palvelukerrokselle ja adaptaatiokerros jokaiseen erityyppiseen järjestelmään. Kun adaptaatiokomponentti liikepalveluväylälle on rakennettu, kykenee väylän palvelut

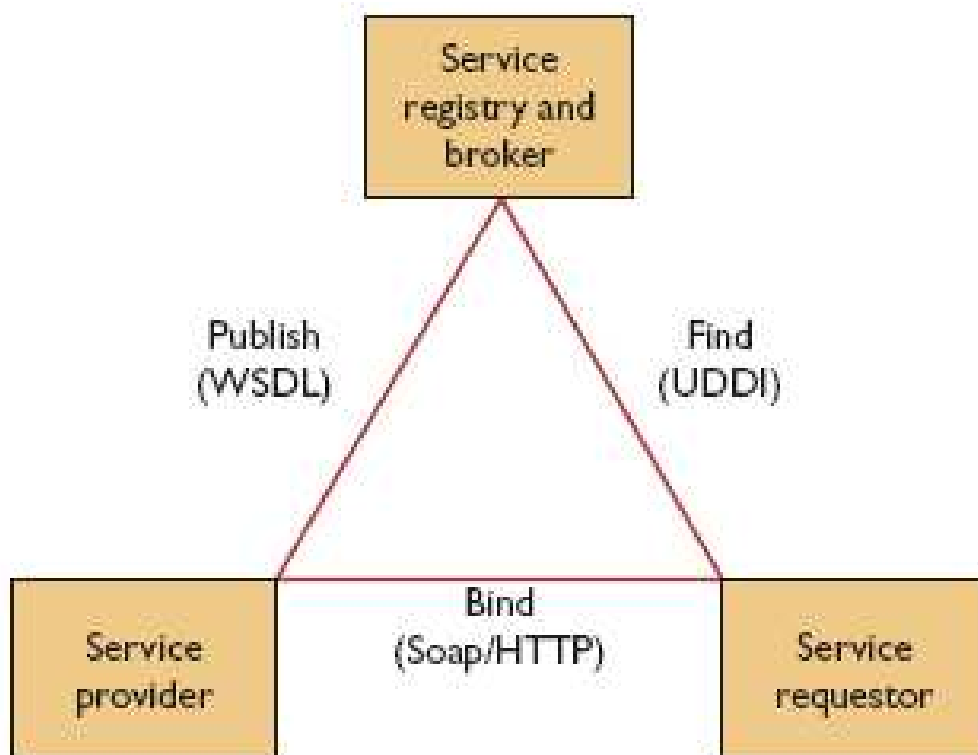
kommunikoidaan järjestelmän muiden komponenttien kanssa. Kommunikaatio väylään tapahtuu hyväksikäyttäen ainoastaan liikepalveluväylän viestitysmekanismeja. Liikepalveluväylä ei listaa välttämättömiä adaptaatiokomponentteja, mikä onkin mahdollistanut markkinat monelle liikepalveluväyläratkaisuja tuottavalle taholle. Suuri osa liikepalveluväyläkomponenteista vielä nykypäivänäkin on Java-pohjaisia, sillä muita järjestelmäriippumattomia komponentteja ei ole ollut tarjolla [Cha04].



Kuva 3: Protokollapinoesimerkkejä, joiden avulla voidaan kommunikoida liikepalveluväylään [Cha04].

Kuvassa 3 näemme viisi esimerkkiä mahdollisista protokollakonfiguraatioista, joissa pinon ylimpänä komponenttina ovat liikepalveluväylän viestin otsaketiedot ja itse konkreettinen hyötykuorma. Todellista hyötykuormaa voidaan toki kuljettaa myös lukuisilla muilla protokollakonfiguraatioilla. Sonic ESB mm. kertoo tukevansa yli 200 eri tiedonsiirtoprotokollaa [OJ07]. Sovellukset ja palvelut kommunikoivat toistensa kanssa liikepalveluväylän viestimekanismeilla, kuvan 3 ESB -rajapinnan kautta.

Palvelujen tuottajat ja käyttäjät käyttävät palveluja liikepalveluväylän kautta. Kun palveluntuottaja liittyy liikepalveluväylään, se julkistaa omat rajapintansa ja muut merkittävät ominaisuudet liikepalveluväylälle metadatatassaan. Kun taas puolestaan halutaan kuluttaa palveluja, käytetään liikepalveluväylää löytämään haluttu palvelu liikepalveluväylään kytketyistä palveluista. Liikepalveluväylä hyödyntää palvelusuuntautuneiden järjestelmien löydä-kytkeydy-julkista (*find-bind-publish*) -kolmikantaa [Pap03, HS05], joka on kuvattu kuvassa 4. Tämä malli mahdollistaa löyhän sidonnan ja täten helpon palveluiden korvaamisen, ilman että muu järjestelmä häiriintyy.



Kuva 4: Palvelusuuntautuneiden järjestelmien dynaaminen palveluiden haku ja sidonta [HS05].

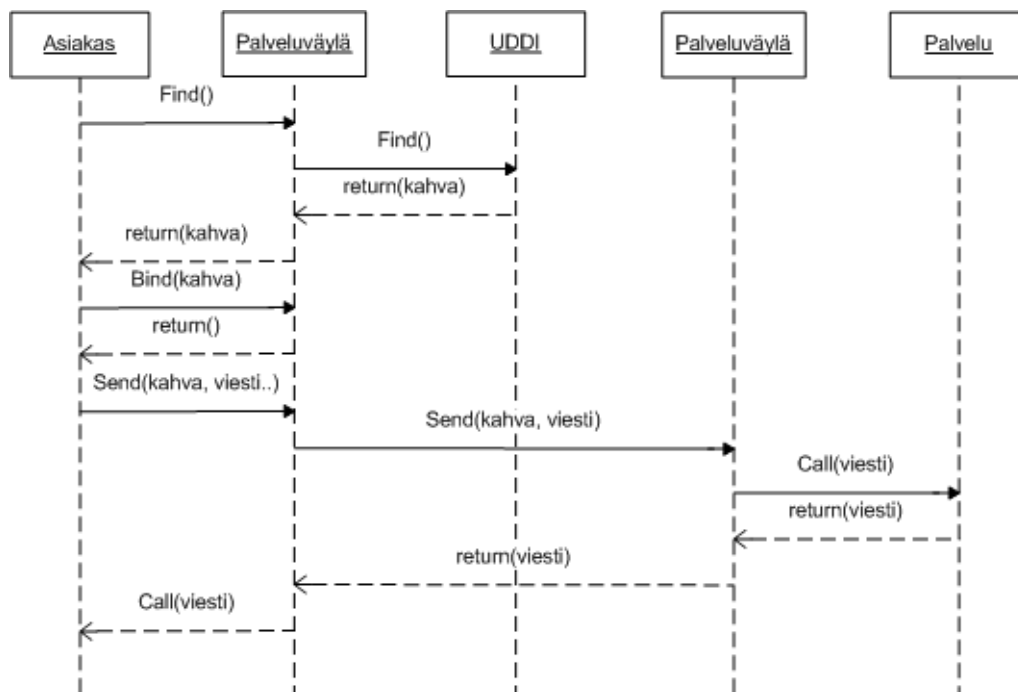
Kuvan 4 mallin mukaisesti palveluiden tuottajat käyttävät WSDL-kuvauskieltä (*Web services Description Language*) kertoakseen omista rajapinnoista ja niiden ominaisuuksista. Tärkeisiin ominaisuuksiin kuuluu mm. palvelun käyttämä kommunikaatioformaatti, parametrimääreet ja palvelun käyttämä kommunikaatiotyyppi. Tätä metatietoa käytetään hyväksi palveluita suunniteltaessa ja nämä määrittelevät liikepalveluväylän päätepiitteet (*end point*) [Cha04, DSRR08]. Päätepiitteitä käytetään hyväksi reityksessä ja koreografia- ja orkestrointipalveluissa.

Kun jokin väylän palveluista haluaa käyttää palvelua, etsii se sen käyttäen UDDI-protokollaa (*Universal Description, Discovery, Integration*). Kun haluttu palvelu on löydetty, otetaan se käyttöön sitomalla (*bind*) sen kahva (*handle*). Kuvassa 4 kuvataan perinteinen malli käyttää palvelua SOAP-protokollan yli [HS05, IBM04]. Palveluväylässä ei ole tarkoituksenmukaista käyttää suoraan SOAP-protokollaa, kuten kuva 3 antaa ymmärtää. Edellä mainitut protokollat ovat osa WS-I:n<sup>1</sup> (*Web*

<sup>1</sup>Tarkat WS-I:n määrittelemät protokollat on lueteltu osoitteessa <http://www.ws-i.org/>

*Services Interoperability organisation*) määrittelemiä, joita myös kaupalliset palveluväylien toimittajat suosivat [PTDL07, DSR08].

Palveluiden dynaaminen sidonta käy ilmi kuvan 5 sekvenssikaaviosta.



Kuva 5: Liikepalveluväylä abstrahoi konkreettisen viestinvälitysprotokollan ja hoitaa kommunikoinnin tämän puolesta.

Kuvassa 5, asiakas haluaa kommunikoida palvelun kanssa. Ensin se pyytää liikepalveluväylän hakupalvelua (find) tarjoamaan palvelumääritelmän toteuttavan kahvan palvelulle. Kun liikepalveluväylä löytää sopivan palvelun, sitoo (bind) asiakas itsensä kyseiseen palveluun. Seuraavana asiakas pyytää liikepalveluväylän välittämään viestin palvelulle. Kommunikointi palvelulle tapahtuu asiakkaan tuntematta sen sijaintia tai sen käyttämää tietomallia. Liikepalveluväylä siis abstrahoi konkreettisen viestiprotokollaimplementaation ja ainut yhteinen viestitysprotokolla on se, mikä on sovittu liikepalveluväylän kanssa. Koska viestitysprotokolla on asiakkaalle abstrakti, voi palveluväylä toteuttaa haluamansa luotettavan ja tietoturvallisen kommunikoinnin osapuolten välillä.

Perinteisen etäkutsun hyödyntäminen suoraan sovelluksesta on ongelmallista monessa suhteessa palvelusuuntautuneissa järjestelmissä. Etäkutsut (*RPC, Remoter*

*procedure call*) toimivat usein hyvin synkronisesti [Cha04]. Etäkutsuissa sovellus suorittaa etäkutsun, joka näyttää aivan samalta kuin mikä tahansa muu funktiokutsu. Kutsun ottaa vastaan tynkä-implementaatio, joka tämän jälkeen pyytää palvelua sen todelliselta implementoilijalta. Etäkutsun suorituksen jälkeen tulokset palauteaan kutsun pyytäjälle ja jatketaan paikallisen sovelluksen suorittamiseen. Nykyiset liikepalveluväylät pyrkivät ennen kaikkea asynkroniseen, tapahtumapohjaiseen kommunikointiin [Cha04, DSRR08, PTDL07].

Ongelmana perinteisessä etäkutsukäytännön muodostuu hajautus ja virheiden käsittely. Hajautetuissa sovelluksissa olisi toivottavaa yhteinen virheiden käsittely ja mahdollisesti epäonnistuneen pyynnön suorittaminen jossain toisessa palvelussa. Suorien etäkutsujen käyttö ei mahdollista yhtenäistä virheiden käsittelyä. Liikepalveluväylän tuoma lisäästraktio pyrkii ratkaisemaan tämänkin ongelman [PvdH05] ottamatta kantaa todelliseen implementaatioon, sillä palvelupyyntöjen onnistuminen taataan liikepalveluväylän puolesta.

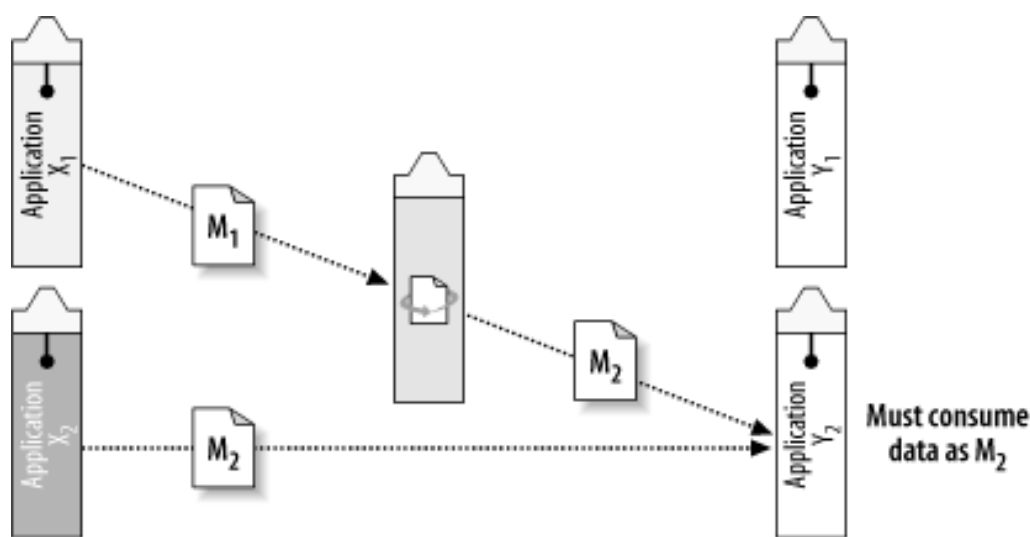
Liikepalveluväylän etuna sen abstrahoidessa viestinvälitysprotokolla on yhtenäistää kommunikaatio, yhdenmukaistaa virheiden käsittely ja kyky siepata (*intercept*) kommunikaatio välistä. Kyky siepata kommunikaatio mahdollistaa myös tiedon muuttamisen (*mediation*), tyyppikonversiot (*schema transformation*) ja viestin uudelleenreitityksen [Cha04, DSRR08]. Kuvan 6 esimerkissä hyödynnetään abstraktion mahdollistamaa tulkkausta ja tehdään tyyppimuutos. Nämä tekniikat edesauttavat bisnessovellusten integroinnissa [HTL05].

Reititys keskitetyssä hub-and-spoke -arkkitehtuurissa (kuva 2) oli ongelmallista järjestelmien koon kasvaessa. Tässä arkkitehtuurissa kaikki viestit reititetään aina keskitetyn pisteen kautta. Liikeväyläarkkitehtuurissa reititysmenetelmiä on kaksi.

Ensimmäinen reititys pohjautuu matkakuvausreititykseen (*itinerary-based routing*). Tässä mallissa viestin reitti on tiedossa heti palvelun ensimmäisestä pisteestä lähtien. Toisin sanottuna, kuvaus reitistä liitetään viestin otsaketietoihin ja tietoa kuljetaan väylässä viestin mukana [Cha04].

Toinen reititysmenetelmä on sisältöpohjainen (*CBR, Content-based routing*) reititystä. Tätä menetelmää kutsutaan myös älykkääksi reititykseksi. Sisältöpohjaista reititystä tapahtuu, kun liikepalveluväylä tutkii viestin otsakkeen tai hyötykuorman sisältöä, ja päättää muuttaa viestin kulkureittiä. Sisältöpohjainen reititys on osa palvelukontin (*service container*) toimintoa, ei osa viestin metadatta [Cha04].

Esimerkkinä sisällön aiheuttamasta viestin uudelleen reitittämisestä on tietotyypin



Kuva 6: Liikepalveluväylä havaitsee ristiriidan asiakkaan tuottamassa tietomallissa ja muuntaa sen palvelulle sopivaan muotoon.

yhteensopimaattomuus, joka vaatii tietomallikonversio. Tietomallikonversio tapahtuu joko liikepalveluväylän erillisenä palveluna (*infrastruktuuripalveluna*) tai sidottuna asiakkaan omaan sovellusympäristöön, tämän palvelukonttiin (service container) [Cha04]. Tietomallikonversiot tapahtuvat usein XML-pohjaisella merkintäkielillä, hyväksikäyttäen XSLT:tä (*Extensible Stylesheet Language Transformation*).

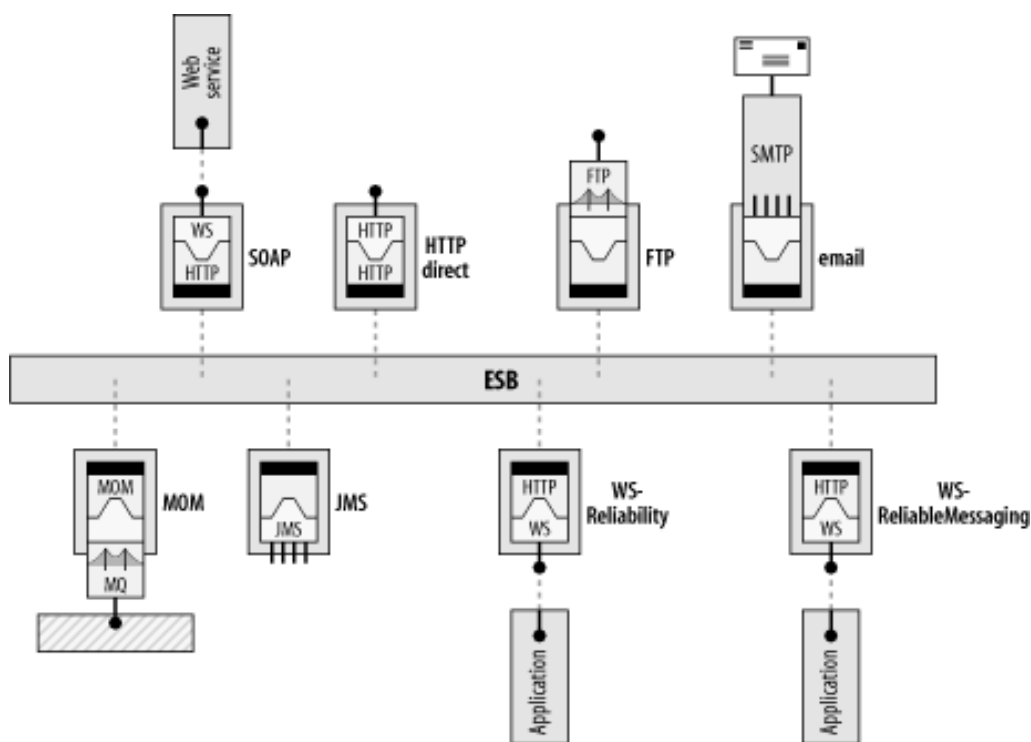
Reitityssäännöstöehdotelmissa [Cha04] puhutaan skriptikielistä, XPathista, mutta myös BPEL4WS<sup>1</sup>:stä, joka puolestaan on täydellinen bisnesprosessien kuvauskieli.

## 3.2 Integraatio-elementit

Liikepalveluväylään liittyviä tietolähteitä on monia. Tietolähteinä voi toimia tietokannat, aikaisemmat EAI -sovellukset, Web-palvelut kuin myös muut viestipohjaiset järjestelmät. Tämä kertoo siitä, että liikepalveluväylän tukee kyetä kommunikoidaan myös muiden kuin WS-I -standardien mukaisesti. WS-I -organisaatio ei määrittele mm. yleisiä internetissä käytettyjä protokollia kuten sähköpostiprotokollat (SMTP, POP, IMAP) tai tiedonsiirtoprotokolla FTP. Kyseiset integraatorajapinnat ovat kuitenkin välttämättömiä, mikäli halutaan liittää bisneslogiikkavuohon esimerkiksi hyväksyntäprosesseja, jotka hyväksikäyttävät sähköpostia. Ilman näitä

<sup>1</sup><http://docs.oasis-open.org/wsbpel>

yleisiä standardeja tulisi jälleen kerran määritellä vastaava toiminnallisuus, ja näin ollen hyvin helposti olisimme samassa tilanteessa missä EAI -tyyppiset integraatiomallit. Ongelmana näissä oli rajallinen hajautettavuus, kyvyttömyys kommunikoida oman verkon ulkopuolelle, markkinoiden fragmentoituminen ja uuden protokollan määrittelyyn ja implementointiin vaadittava työ.



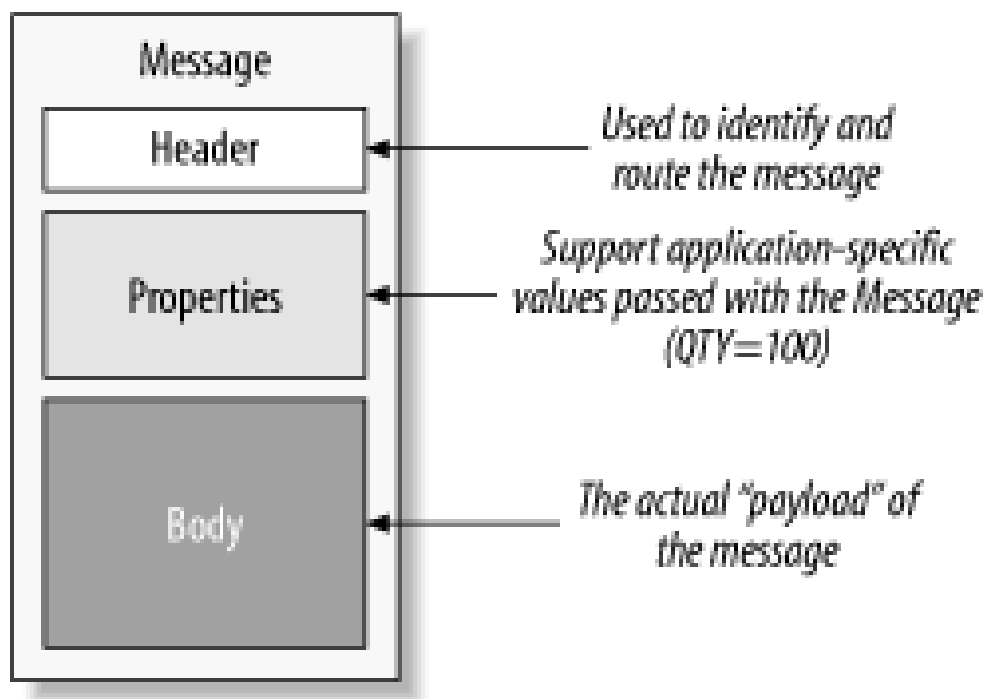
Kuva 7: Liikepalveluväylään kytkeytyneitä palveluja [Cha04].

Kuvassa 7 näemme liikepalveluväylään kytkeytyneitä muita sovelluskerroksen palveluja, jotka ovat kykeneväisiä kuljettamaan liikepalveluväylän viestejä.

Viestinvälityksessä tavanomaiset tietotyypit ovat binäärimuotoinen data ja teksti. Moni kuitenkin määrittelee yhdeksi tietotyypeistä myös XML-formaatin. Useat liikepalveluväylätoimittajat tarjoavat myös omat erikoistietotyypinsä [DSRR08]. Omilla tietotyypeillä pyritään usein optimoidumpaan toimintaan erikoistilanteissa.

Liikepalveluväylässä kommunikointi voi tapahtua kolmella tavalla: yksi (*one-way*) tai kaksisuuntaisesti (*request-response*) tai tilauspohjaisesti (*publish-and-subscribe*). Esimerkiksi yksisuuntaiset FTP ja SMTP -protokollat eivät kykene kaksisuuntaiseen kommunikointiin, koska ne eivät voi palauttaa viestiä takaisin väylään [DSRR08]. Ti-

laus pohjaisessa kommunikoinnissa väylältä tilataan jonkun otsikkotiedon mukaisia viestejä, ja kun ehdot täyttyvät, viestit välitetään tilaajalle [Cha04]. Koska liikepalveluväylä abstrahoi kommunikaatioväylän tiedonsiirtoprotokollan, ei kaikki protokollat sovellu toteuttamaan kaikkia edellämainittuja viestitysmekanismeja. On kuitenkin huomattava, että tämä on täysin salittua.



Kuva 8: Viesti voi sisältää hyötykuorman lisäksi myös viestin rakenteeseen liittyvää tärkeää tietoa [Cha04].

Viestit kulkevat liikepalveluväylässä viestivuona palvelusta toiseen [Cha04]. Palveluiden määränpäinä on liikepalveluväylän päätepisteet (*endpoint*). Alustava kulkureitti on määriteltynä viestin ensimmäisestä pisteestä lähtien. Tieto reitistä kulkee osana viestin otsaketietoja. Viestin kulkureitin määrittelee bisneslogiikka- ja koreografiapalvelut.

Kun esimerkiksi bisneslogiikka on vaatinut tiedoston siirtoa FTP -protokollaa hyväksikäyttäen ja tämän jälkeen tiedoston lähetyksen sähköpostilla (SMTP) eteenpäin, kulkee reititustieto osana viestin otsaketietoja (kuva 8). Bisneslogiikka toimii korkeamman tason reitittäjänä, kun taas liikepalveluväylä ratkoo yhteensopivuus



ja transformaatio-ongelmat. Kun viesti vastaanotetaan liikepalveluväylän sähköpostipalveluun, tulkitaan viestin lisäotsakekentistä tieto viestin sisällöstä, liitetiedostoista, vastaanottajista ja muista tärkeistä tekijöistä, jotta palvelu osaa käsitellä sähköpostin ja lähettää sen.

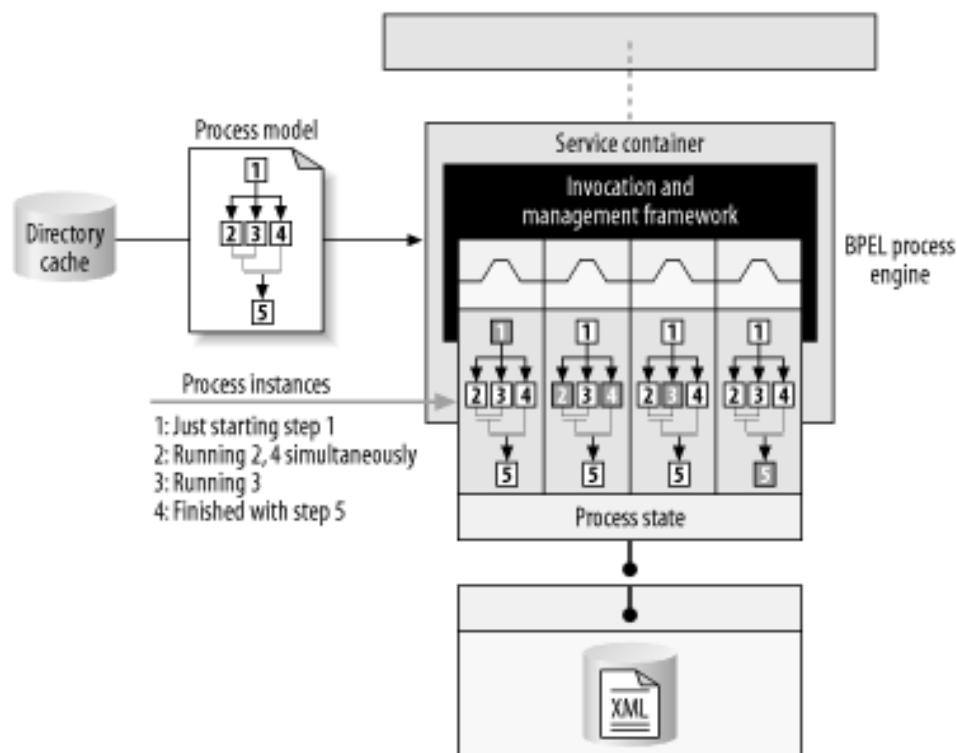
### 3.3 Hallintamenetelmät

Jokainen liikepalveluväylän palvelu rakennetaan palvelukontin (service container, kuva 1) päälle. "Palvelukontti on fyysinen ilmentymä abstraktista päätepiesteestä ja se toteuttaa palvelurajapinnan." [Cha04]. Yksinkertaisimmillaan palvelukontti on kevyt prosessi tietokoneessa, jonka päällä palvelua suoritetaan. Konttia voisi verrata virtuaalikoneeseen tai hiekkalaatikkoon, kuitenkin ilman tulkkausominaisuuksia. Palvelukontin palveluita suoritetaan rikastetussa ympäristössä, johon mahdollistetaan pääsy konfiguraatio- ja hallintatyökaluilla (kuva 9).

Hallintatyökalut mahdollistavat yksittäisten palvelujen ympäristön muokkaamisen, seuraamisen ja hajauttamisen. Hallintatyökaluja voidaan käyttää myös analysoidessa ja valvottaessa liikepalveluväylän suorituskykyä, järjestelmien käyttöastetta ja mahdollisia pullonkauloja. Hallintatyökaluilla on myös suuri merkitys palveluiden orkestroinnissa ja koreografiassa.

Bisnesprosessin hallintaan käytetään kahta menetelmää, koreografiaa ja orkestrointia. Kumpainkaan näistä tekniikoista ei ole osa liikepalveluväylän infrastruktuuripalvelua [IBM04], vaan kyse on erillisistä palveluista. Näitä tekniikoita käytetään hallitsemaan prosessi- eli etenemisvuota, eli reititystä. Orkestroinnissa on ainoastaan yksi keskitetty koordinaattori ja koreografiassa ei. Koreografiapalveluita käytetään yleensä pitkän aikavälin toimintaprosesseissa [IBM04, Cha04, PvdH05]. Näitä tekniikoita käyttäen kuvataan kuinka prosessissa edetään. Kun prosessia suunnitellaan, käytetään palveluiden julkaisemia rajapintatietoja (*WSDL*) hyväksi mallintaessa prosessin etenemistä. Koska prosessin etenemisen voi kuvata täysin graafisilla työkaluilla, tulee tästä sanonta, että palveluita tulisi konfiguroida, ei ohjelmoida. Bisnesprosessin kuvaukseen on myös hyödynnettävissä useita standardeja. Näistä merkittävin ehkä WS-I -organisaation WS-BPEL (*Web Services Business Process Execution Language*).

Palveluiden orkestrointi on tärkeä osa palveluvuota ja täten yleensä osa kuvattua liikepalveluväylätoiminnallisuutta.



Kuva 9: Palvelua voidaan hallita palvelukontin (Service Container) hallintarajapinnan kautta [Cha04].

## 4 Yhteenveto

Järjestelmien koon kasvaessa aikaisemmissa EAI -intergraatoratkaisuissa käytetty keskitetty arkkitehtuuri oli usein pullonkaulana. Liikepalveluväyläarkkitehtuuri kykenee aikaista tehokkaampaan palveluiden hajautukseen. Liikepalveluväylä pyrkii myös nopeuttamaan uusien ohjelmistoratkaisujen kehittämistä abstrahoimalla hankalat ja kaupalliset väliohjelmistot, mutta määrittelemään kuitenkin sovituserroksen näille.

Liikepalveluväylä tukeutuu voimakkaasti olemassa oleviin standardeihin ja hyväksii havaittuihin protokolleihin, samalla esitellen olemassa olevien ratkaisujen heikkoja kohtia pyrkien korjaamaan niitä. Hajautettuja palveluja hallitaan ja seuraataan palvelukonttien kautta.

Tässä seminaarissa keskityttiin liikepalveluväylään ja sen ominaisuuksiin. Seminaarissa käsiteltiin palvelusuuntautuneiden järjestelmien merkittävimmät tekijät: nykyistä edeltäneet arkkitehtuurit, yleiset tiedonsiirtoprotokollat, sidontamenetelmät, palveluiden asynkronisuus, tietomallitransformaatiot, viestien rakenteet sekä niiden merkitys liikepalveluväyläarkkitehtuurissa.

## 5 Loppusanat

Liikepalveluväylien merkityksestä kiistellään paljon. Useat näkevät tämän arkkitehtuurimallin täysin tarpeettomana, toiset taas kokevat sen olleen katalyyttinä monelle teollisuuden kaipaamalle standardille. Tästä huolimatta, se, että liikepalveluväyliä on tarjolla nykyään useita kymmeniä, jollei satoja kertoo jotain. Liikepalveluväylät tai kehittyneet EAI -arkkitehtuurit ovat liike oikeaan suuntaan.

Vaikka liikepalveluväylä konseptina noteerattiinkin vasta 2004, on sen ympärillä tapahtunut paljon. Jos vertaamme liikepalveluväylien ensiaskelia nykypäivään, havaitsemme, että kaupallisten protokollien lisäksi, suuri osa järjestelmistä tukee myös standardoituja avoimia tekniikoita.

Nykytrendi näyttääkin olevan avoimempi kuin olisi voinut kuvitella. Ei pelkästään avoimen lähdekoodin liikepalveluväylien osalla, mutta myös suurten ohjelmistoyritysten. Tänä päivänä monet suuret ohjelmistoyrityksen tarjoavat liikepalveluväyläalustat käyttöön ilmaiseksi, sillä asiakkaat eivät halua lukittautua yhteen valmistajaan. Kilpailuasetelma on selvästi muuttunut, ja nyt myyntivalttina onkin eri valmistajien integraatioarsenaali vanhoihin järjestelmiin ja yritysten uskottavuus markkinoilla.

Vaikka vastausta kysymykseen onko *"liikepalveluväylä vain entiset standardit puettuna uuteen termiin"* ei tässä välttämättä saada, voimme kuitenkin olla varmoja, että liikepalveluväylä on edesauttanut palvelusuuntautuneita järjestelmiä lähemmäksi määränpäättään, parempaa yhteensopivuutta.

## Lähteet

- Cha04 Chappell, D., *Enterprise Service Bus*. O'Reilly, July 2004.
- DSRR08 Davies, J., Schorow, D., Ray, S. ja Rieber, D., *The Definitive Guide to SOA: Oracle Service Bus*. Apress, Berkely, CA, USA, 2008.
- Har05 Harsu, M., *Ohjelmointikielet: Periaatteet, käsitteet, valintaperusteet*. Talentum, Helsinki, Finland, 2005.
- HS05 Huhns, M. N. ja Singh, M. P., Service-oriented computing: Key concepts and principles. *IEEE Internet Computing*, 9,1(2005), sivut 75–81. URL <http://dx.doi.org/10.1109/MIC.2005.21>.
- HTL05 Herault, C., Thomas, G. ja Lalanda, P., Mediation and enterprise service bus: A position paper. *MEDIATE2005*, Hepp, M., Polleres, A., van Harmelen, F. ja Genesereth, M. R., toimittajat, osa 168 sarjasta *CEUR Workshop Proceedings*. CEUR-WS.org, 2005, sivut 67–80.
- IBM04 *Patterns: implementing an soa using an enterprise service bus*. IBM Corp., Riverton, NJ, USA, 2004.
- OJ07 Ortiz Jr., S., Getting on board the enterprise service bus. *Computer*, 40,4(2007), sivut 15–17.
- Pap03 Papazoglou, M. P., Service -oriented computing: Concepts, characteristics and directions. *Web Information Systems Engineering, International Conference on*, 0, sivu 3.
- PTDL07 Papazoglou, M. P., Traverso, P., Dustdar, S. ja Leymann, F., Service-oriented computing: State of the art and research challenges. *Computer*, 40,11(2007), sivut 38–45.
- PvdH05 Papazoglou, M. P. ja van den Heuvel, W.-J., Service oriented computing: approaches, technologies and research issues. *The VLDB Journal*, 16, sivut 389–415.
- SHLP05 Schmidt, Hutchison, Labros ja Phippen, The enterprise service bus: Making serviceoriented architectures real. *IBM Systems Journal*, 0.