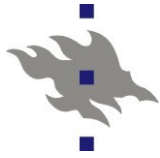


Palvelusuuntautunut ohjelmistotuotanto

Luento 4: Ei-toiminnalliset piirteet;
Väliohjelmistojen rooli palvelusuuntauneessa
ohjelmistotuotannossa
Toni Ruokolainen, 26.1.2010



Luennon runko

- Ei-funktionaaliset piirteet
 - Karakterisointi
 - Ei-toiminnallisten piirteiden hallinta
- Väliohjelmistojen rooli palvelusuuntautuneessa ohjelmistotuotannossa
 - Palveluperustaisten järjestelmien elinkaarten tukeminen
 - Ei-toiminnallisten piirteiden realisointi
 - Ohjelmistotuotantoprosessien tukeminen



Toiminnalliset ja ei-toiminnalliset piirteet

- Palveluiden toiminnalliset piirteet tuottavat niiden varsinaiset toiminnallisuuden
- Toiminnallisia piirteitä ovat esimerkiksi
 - Palvelurajapinnan rakenne
 - Kommunikoitavien dokumenttien rakenne
 - Toimintojen (operaatiot, metodit) semantiikka
 - Käyttäytyminen
- Ei-toiminnalliset piirteet tuovat jotain lisäarvoa palvelun toiminnallisuuteen
- Ei-toiminnallisia piirteitä ovat esimerkiksi
 - Salattu tiedonsiirto
 - Kolmannen osapuolen varmentamat interaktiot (notariaatti)
 - Erilaiset palvelunlaatuominaisuudet
 - Esimerkiksi saatavuus, vasteaika, luotettavuus, ...



Ei-toiminnalliset piirteet ja ominaisuudet

- Ei-toiminnallinen piirre (*non-functional feature, NFF*)
 - Määrittelee lisäarvoa tuottavan piirteen semantiikan ja arvojoukon
 - Esimerkiksi yleisesti hyväksytty käsite, jonkin standardin tai ontologian käsite.
 - Esimerkiksi: *Vasteaika*
- Ei-toiminnallinen ominaisuus (*non-functional property, NFP*)
 - Ei-toiminnalliselle piirteelle on annettu jokin kelvollinen arvo sen määrittelemästä arvojoukosta.
 - Esimerkiksi: *Vasteaika < 2s*
- Ei-toiminnallisia ominaisuuksia voidaan käyttää
 - Palveluiden etsimiseen
 - Etsitään joukko palveluita, jotka täyttävät annetut kriteerit
 - Palveluiden valintaan
 - Valitaan paras palvelu kandidaattijoukosta



Ei-toiminnalliset piirteet poikkileikkaavina ominaisuuksina

- **Horisontaalisesti samalla abstraktiotasolla**
 - Efektit toiminnallisiin piirteisiin (esim. käyttäytyminen ja struktuuri)
 - Efektit eri toimijoiden välillä: ei-toiminnalliset piirteet eivät välttämättä ole realisoitavissa pelkästään yhden toimijan varassa → palvelusopimukset
- **Vertikaalisesti: abstraktiotasojen väliset riippuvuudet**
 - Tulkintaongelmat käyttäjävaatimusten ja järjestelmävaatimusten välillä
 - Kuinka formalisoida esimerkiksi vertikaaliset riippuvuudet?
 - Liiketoiminnallinen palvelu: *"tarvitaan luottamuksellisuutta"* →
 - Komponenttipalvelu: *"vaaditaan vahvaa salaus"* →
 - Teknologinen palvelu: *"tulee käyttää 1024 bittistä RSA salausta"*
- **Piirteiden väliset interaktiot**
 - Ei-toiminnallisten ja toiminnallisten piirteiden välillä
 - Esimerkiksi viestinnän asynkroninen salaus vaatii avaimenvaihtoprotokollan mukaisen käyttäytymisen lisäämistä
 - Ei-toiminnallisten piirteiden välillä
 - Esimerkiksi sanomien salaus ja sen purkaminen vaikuttavat vasteaikaan



Ei-toiminnallisten piirteiden luonteesta

- Asiakkaan subjektiivisen arvioinnin kohteita
 - Esimerkiksi varakkaammalle asiakkaalle palvelun laatu on tärkeämpää kuin sen hinta.
- Ei-toiminnallisten piirteiden tärkeys riippuu palvelun luonteesta
 - Esimerkiksi "varaa hotellihuone" -palvelu vaatii yksityisyyttä ja luotettavuutta, kun taas "selaa vapaita hotellihuoneita" ei niinkään.
- Ei-toiminnallisten ominaisuuksien kuvaukset epävarmoja
 - Asiakkaan preferenssit ja arviointikriteerit ovat usein laadullisia ja epämääräisiä (*"vaadin voimakasta salausta"*), kun taas ei-toiminnallisten piirteiden arvojoukot ovat usein määrällisiä (*"käytän 1024 bitin salausavainta"*).
 - Joidenkin piirteiden "hyvyyden" vertaaminen on hankalaa.
 - Esimerkiksi poikkeuskäsittelyn taso



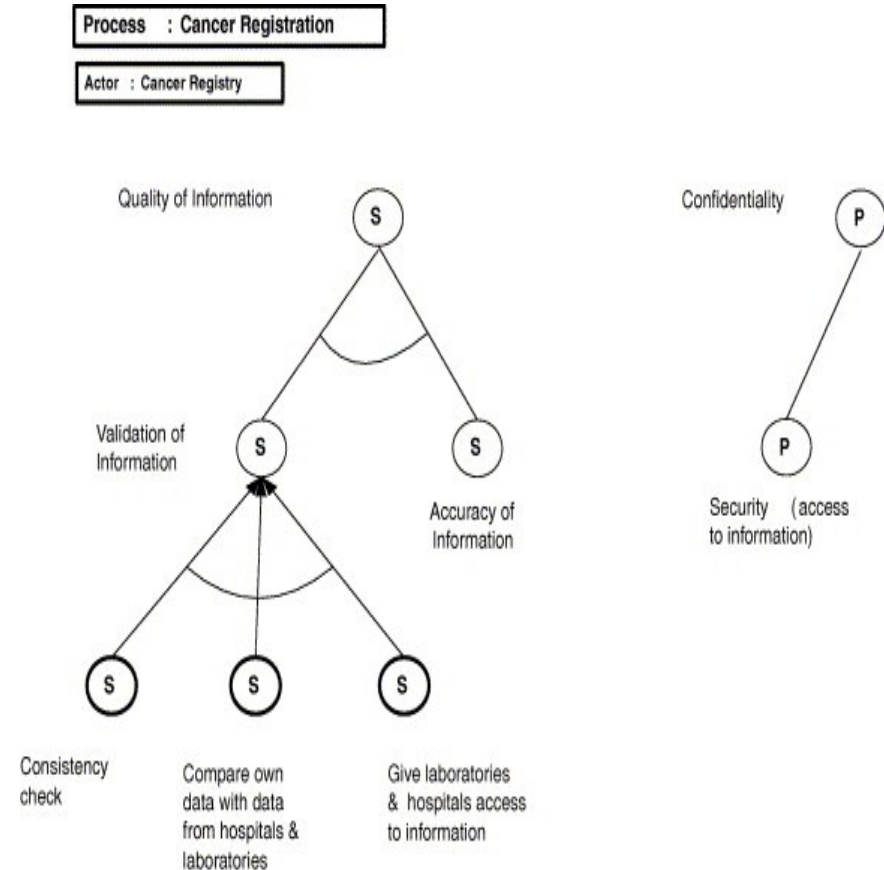
Ei-toiminnallisten piirteiden hallinta

- Ei-toiminnallisten piirteiden tarpeen tunnistaminen
 - Kuinka tunnistetaan liiketoiminnan kannalta välttämättömät piirteet?
- Ei-toiminnallisten piirteiden mallintaminen
 - Mikä on ei-toiminnallisen piirteen semantiikka?
 - Kuinka ei-toiminnalliset piirteet sidotaan toiminnallisuutta kuvaaviin malleihin?
- Ei-toiminnallisten ominaisuuksien yhteensopivuus
 - Tulee vastaan erityisesti palvelukoosteissa
 - Keskinäiset riippuvuudet ja priorisointi
 - Piirteiden ja arvoalueiden yhteensopivuus
- Ei-toiminnallisten piirteiden ja ominaisuuksien realisoiminen
 - Sopimusneuvottelut --> palvelusopimukset ei-toiminnallisista ominaisuuksista
 - Toimintaan saattaminen --> väliohjelmiston konfigurointi
 - Ominaisuuksien valvonta --> infrastruktuuripalveluiden käyttäminen valvontaan



Ei-toiminnallisten piirteiden tunnistaminen

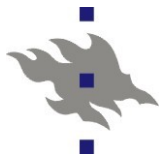
- Kehitetty erilaisia metodologioita ei-toiminnallisten piirteiden tunnistamiseen vaatimusmäärittelyssä
 - Voivat perustua esimerkiksi puoli-strukturoituihin haastatteluihin
 - Hyödynnetään toimialueriippumattomia käytänteitä ja haastattelukaavaimia, skenaarioita, tunnettuja riskejä, anti-patterneita,...
 - Iteroidaan haastattelua, kunnes saadaan tarpeeksi tarkat määrittelyt
 - Määritellään esimerkiksi ei-toiminnalliset piirteet tavoitteiksi (*goal*), jaetaan niistä alitavoitteiksi, kunnes ne ovat niin konkreettisia, että ne voidaan ilmaista toimintoina





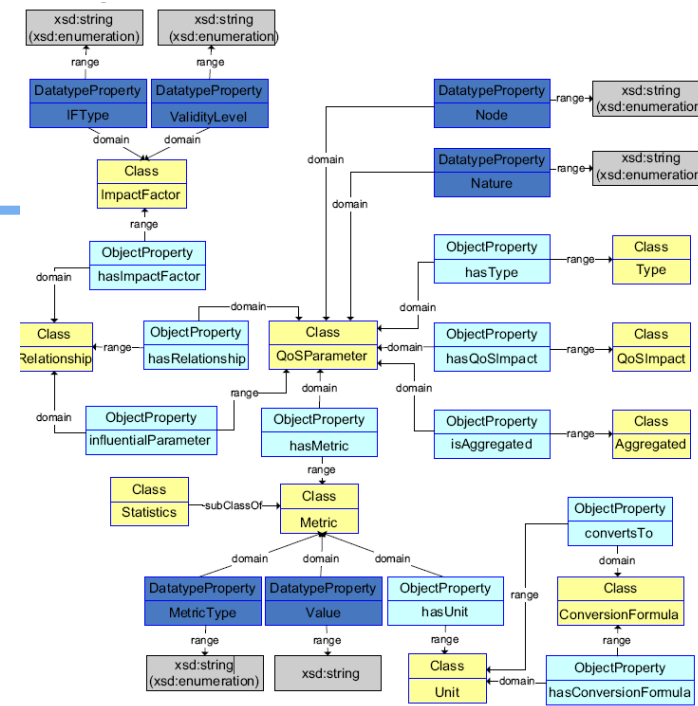
Ei-toiminnallisten piirteiden semantiikasta

- Jotta ei-toiminnallisista piirteistä voitaisiin sopiva autonomisten toimijoiden kesken, tulee niiden tulkinta olla yksikäsitteinen
- Ei-toiminnallisten piirteiden semantiikan määrittelyyn käytetään pääasiassa kahta lähestymistapaa
 - Ontologinen semantiikka
 - Ei-toiminnallisille piirteille on määritelty kategoriat
 - Kategoria määrittelee piirteen karakterisoivat ominaisuudet
 - Eräänlainen *aksiomaattisen semantiikan* variantti
 - Ontologia voidaan määritellä esimerkiksi OWL-kielellä
 - Muunnospohjainen semantiikka
 - Ei-toiminnallisen piirteen semantiikka määrittyy sen tuottaman efektin mukaisesti
 - Efekti voidaan määritellä joko samalla tai alemmalla abstraktiotasolla
 - Eräänlainen *operationaalisen semantiikan* variantti
 - Efektit voidaan määritellä esimerkiksi *mallimuunnoksen*
 - Näitä lähestymistapoja voidaan käyttää myös yhdessä

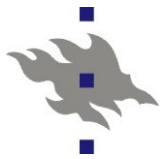


Esimerkki QoS-ontologiasta ja sen käytöstä

- Määritelty QoS-ontologia ja sitä vastaava kieli
 - Ontologia määrittelee *QoS-piirteiden* peruskäsitteet ja niiden väliset yhteydet
 - Ontologian luokkia voidaan instantioida ja siten laajentaa ontologiaa
 - Esim. *Accessibility* on eräs *QoSParameter*-luokan instanssi
 - Kielen avulla ekosysteemin toimijat voivat määritellä *QoS-ominaisuuksia*
 - QoS-kielillä määrittelystä QoS-profiilista muodostetaan analyysiä varten OWL-kuvaus

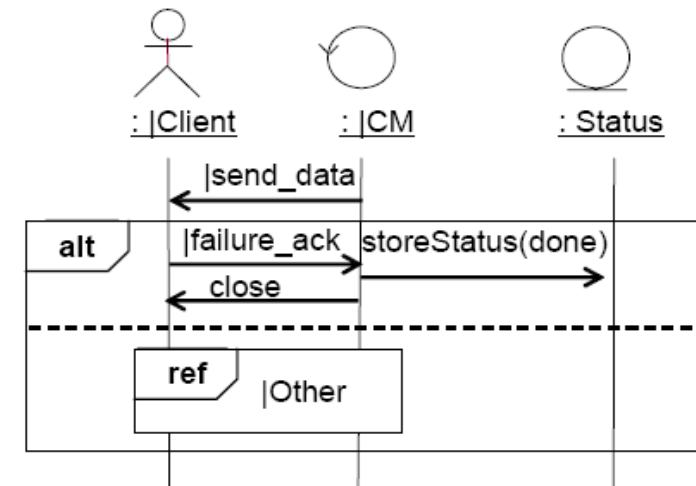
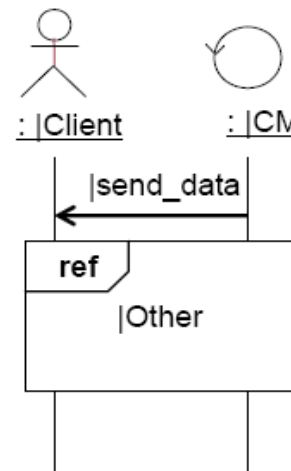


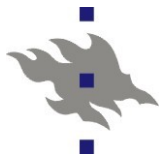
Accessibility	0.99000
Availability	0.99995
Capacity	200
Scalability	0.80
Performance	
(max) Jitter	1 (msec)
(max) ErrorRate	10 ⁻⁵
(max) Latency	300 (msec)
(min) Throughput	384 (Kbps)
ResponseTime	0.01 (sec)
Cost	20.00 (Euro/month)
Configuration	
SupportedStandards	“UDDI 3.0”
SupportedStandards	“WSDL 1.1”
SupportedStandards	“SOAP 1.0”
Reliability	
MTBF	36,000,000 (sec)
MTTR	1,800 (sec)



Esimerkki: malliperustainen ei-toiminnallisten piirteiden määrittely

- Ei-toiminnallisten piirteiden semantiikka voidaan määrittellä niiden tuottamien efektien perusteella
- Esimerkiksi virnehallinnan tuottava ei-toiminnallinen piirre lisää käyttäytymiskuvaukseen uusia toiminteita
- Effektien määrittely voidaan antaa mallimuunnoksien kuvauksina
 - Esimerkiksi käyttämällä mallikaavaimia, joissa määritely muunnoksen alkutila (LHS) ja lopputila (RHS)





Ei-toiminnallisten piirteiden ja ominaisuuksien realisointi

■ Suunnitteluajana mallinnetaan tarvittavat piirteet

- Tunnistetaan ei-toiminnalliset vaatimukset ja niitä vastaavat ei-toiminnalliset piirteet
- Yhdistetään ei-toiminnalliset piirteet toiminnallisiin malleihin
- Analysoidaan piirteiden yhteentoimivuus ja vastaavuus vaatimusten kanssa

■ Palvelun elinkaareissa

- Julkaisu- & toiminta-aikana ylläpidetään NFF tietoutta
- Toiminta-aikana
 - Mahdollistetaan dynaaminen sidonta ei-toiminnallisten piirteiden perusteella
 - Valvotaan palvelun suoritusta

■ Yhteistyöverkoston elinkaareissa

- Neuvottelut ei-toiminnallisista ominaisuuksista
- Yhteistyön konfigurointi sopimusten mukaisesti
- Palvelusopimusten valvonta
- Annetaan palautetta yhteistyön laadusta

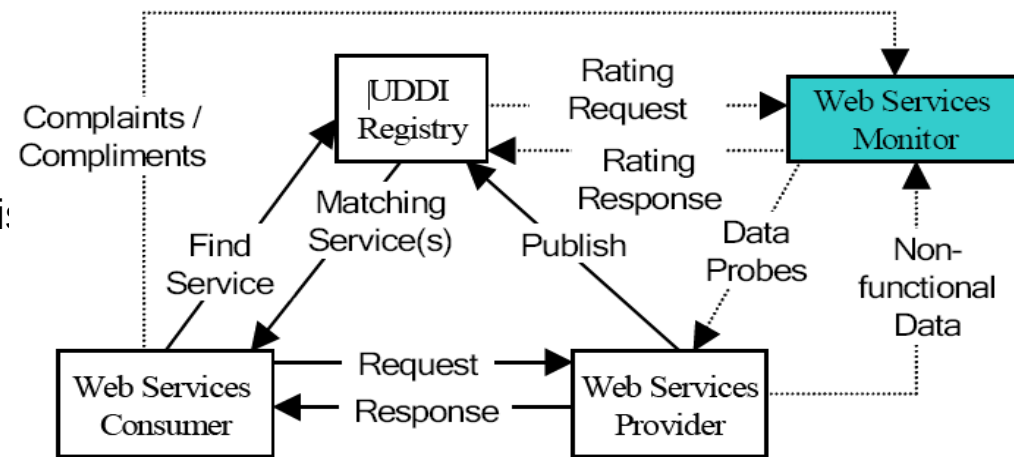


Figure 3: The Proposed Architectural Extension

Kuva: Singhera, Z. U. and Shah, A. A. 2006. Extended web services framework to meet non-functional requirements. In Workshop Proceedings of the Sixth international Conference on Web Engineering (Palo Alto, California, July 10 - 14, 2006). ICWE '06, vol. 155. ACM, New York, NY, 21.



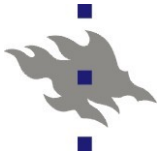
Väliohjelmistojen rooli palvelusuuntautuneessa ohjelmistotuotannossa

■ Väliohjelmistot

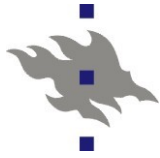
- Määritelmä
- Väliohjelmistoalustat

■ Roolit

- Palveluperustaisten järjestelmien elinkaarten tukeminen
 - Jaetun, abstraktin alustan ja kehyksen toteuttaminen
 - Yhteisen / yhtenäisen kommunikointiabstraktion luominen
 - Esim. kanavat, pubsub, shared tuple space
 - Palveluiden etsintä
 - Palveluiden sitominen
 - Yhteistoiminnan koordinointi (ja hallinta)
- Ei-toiminnallisten piirteiden hallinta ja realisointi
- Ohjelmistotuotantoprosessien tukeminen
 - Tietämyksen hallinta ja jakaminen
 - Palveluiden asentaminen (*deployment*)



Tauko..



Väliohjelmistot

- Väliohjelmisto (*middleware*) on (hajautettu) ohjelmistojärjestelmä, joka tuottaa palveluita sovellusten välisen yhteistoiminnan mahdollistamiseksi
- Väliohjelmiston elementit
 - Väliohjelmistoalusta
 - Ohjelmointirajapinta (*application programming interface, API*)
 - Asiakas- ja palvelinedustajat (*proxy*) sekä tyngät (*stub*)
- Väliohjelmiston ominaisuudet määrittelevät
 - Yhteistoiminnan luonteen
 - Synkronista / asynkronista, hajautuksen luonteen (esim. keskitetty toimitila vs. täysin hajautettu toimijat)
 - Ohjelmistojen toteutustyylin
 - Etäproseduurikutsuihin perustuva (*remote-procedure call, RPC*)
 - Viestiperustainen (*message-oriented middleware*)
 - Hajautettuihin objekteihin / komponentteihin perustuvat lähestymistavat (*distributed object / component middleware*)



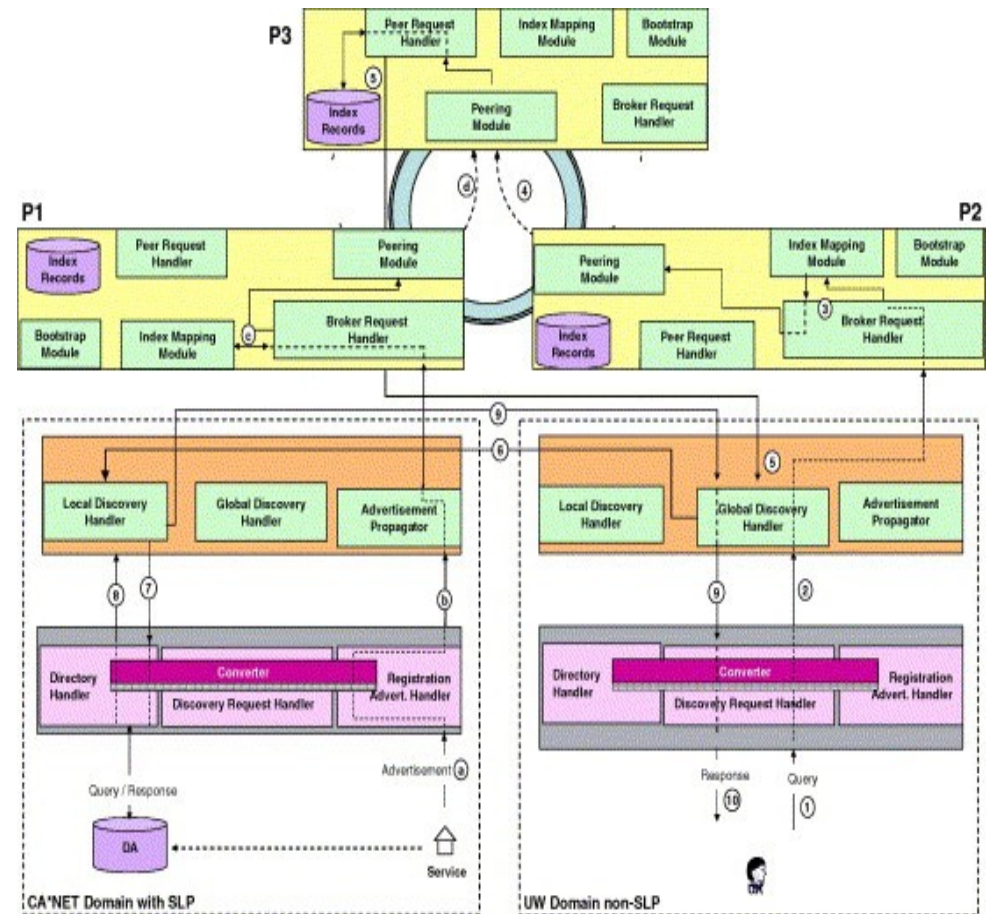
Väliohjelmistoalustat yhtenäisen, jaetun yhteistoimintaympäristön toteuttajina

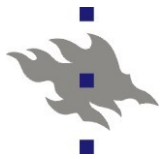
- Väliohjelmistoalusta (*platform*) toteuttaa väliohjelmiston tukemat palvelut
 - Eli tuottaa infrastruktuuripalvelut ekosysteemiin
- Väliohjelmiston palveluita voi olla usealla abstraktiotasolla
 - Teknologiainfrastruktuuripalveluita: yhtenäinen näkemys käyttöjärjestelmä- ja kommunikointipalveluihin
 - Kommunikointipalveluita: palveluiden välinen kommunikointi
 - Yleispalveluita: ilmoitukset, turvallisuus, transaktiot, ...
 - Sovellusaluekohtaisia palveluita: lennonavigointialgoritmeja, potilastietokantamalleja, ...
- Esimerkkejä väliohjelmistoalustoista
 - CORBA (Common Object Request Broker Architecture)
 - J2EE
 - .Net



Palveluiden etsintä ja väliohjelmistot

- Palveluiden etsintään kehitetty useita väliohjelmistoja
 - "Broker" -arkkitehtuuriin perustuvia
 - CORBA trader: palveluiden etsintä nimiperustaisesti; valinta QoS-parametrien avulla; automaattinen (suljettu universumi)
 - UDDI: palveluiden etsintä nimen tai luokitusten mukaan; ei mahdollisuutta valintakriteereiden määrittelyyn
 - P2P -arkkitehtuuriin perustuvia
 - Useita tutkimusprototyyppejä
- Esimerkki: OSDA (*open service discovery architecture*)
 - P2P -perustainen palveluiden etsintä
 - Toimii yhtenäistävänä kerroksena eri palvelunetsintäteknologioiden välillä





Palveluiden sitominen ja väliohjelmistot

■ WS-Binder tutkimusprototyyppi

- Teknologisten palveluiden sidonta dynaamisesti ja QoS-perustaisesti koosteeseen
 - Kooste määritelty abstraktina WS-BPEL prosessina
- Toimintaperiaate
 - Ennen koosteen suorittamista
 - Konfigurointi käyttäjän preferenssien mukaisesti
 - Palveluiden etsintä
 - Palveluiden valinta: parhaiten QoS-vaatimukseen sopivat
 - Valittujen palveluiden sidonta ja edustajien (*proxy*) instantiointi
 - Koosteen suorituksen aikana
 - Edustajat välittävät palvelukutsut sidotuille palveluille
 - Edustajat toteuttavat uudelleensidonnan, jos QoS-vaatimuksia rikotaan

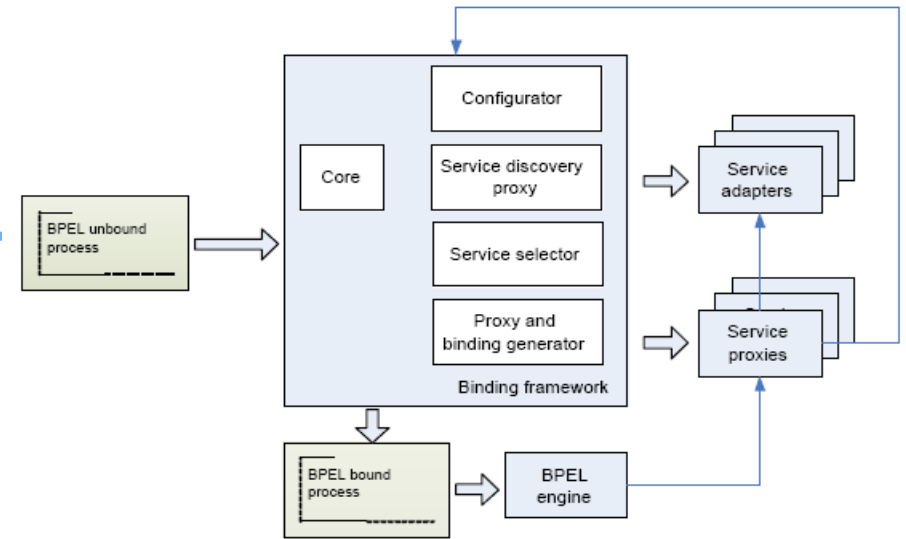


Figure 1: The WS Binder architecture

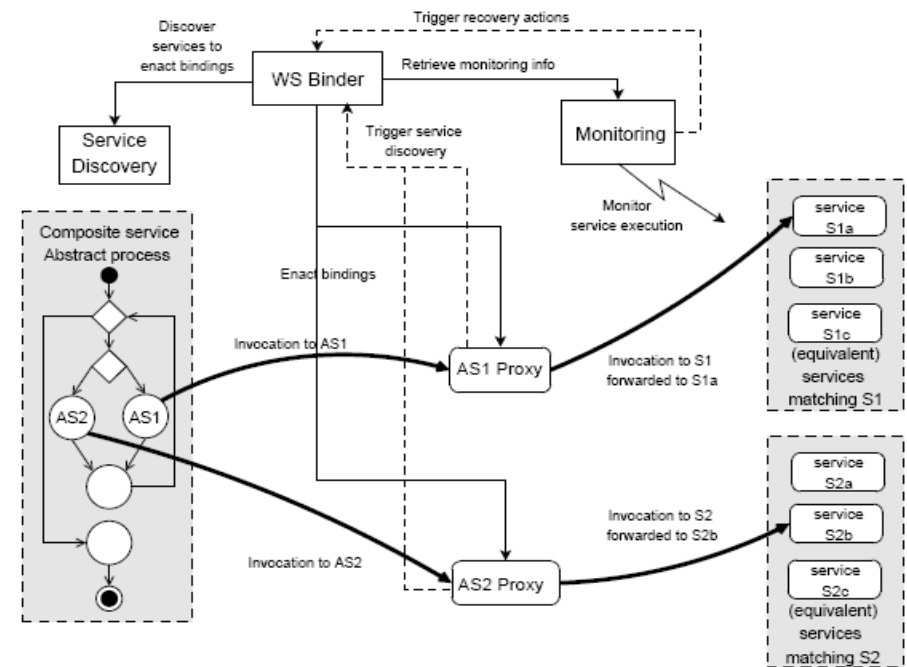


Figure 2: Enacting bindings through proxy services



Yhteistoiminnan koordinointi ja hallinta sekä väliohjelmistot

- **Palvelukoosteiden koordinointi**
 - Hajautettujen koreografioiden koordinointi
 - Esimerkiksi tapahtumankäsittelyyn perustuen
 - Keskitettyjen orkestraatioiden suorittaminen
 - Esimerkiksi WS-BPEL prosessien suorittaminen
- **Palvelusopimusten valvonta**
 - Tarvitaan infrastruktuuripalveluita palveluiden valvontaan sekä sopimusten ylläpitoon
 - Mahdollisista sopimusrikkomuksista tulee huomauttaa asianosaisille ja / tai suorittaa niitä vastaavat korjaustoimenpiteet (kompensointi, sopimuksen purkaminen, uudelleen neuvottelu)
- **Yhteentoimivuuden takaaminen**
 - Palveluiden adaptointi esimerkiksi syntaktisten ja semanttisten epäyhteensopivuuksien yhteydessä
 - Yhteentoimivuustarkistukset metainformaation julkaisun yhteydessä



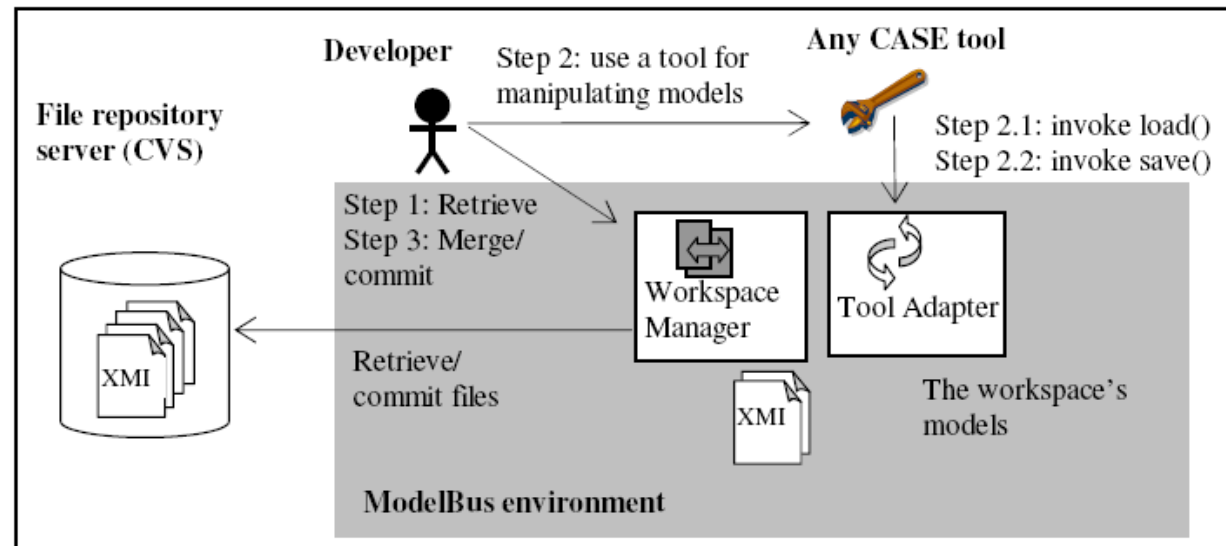
Tietämyksen hallinta ja jakaminen

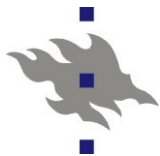
- Hajautetuissa ohjelmistotuotantoprosesseissa tarvitaan infrastruktuuripalveluita tietämyksen jakamiseen

- Tietämys voi koostua
 - Palveluita ja koosteita kuvaavista malleista
 - Ohjelmistometodologioita kuvaavista malleista
 - Vapaamuotoisista dokumenteista

- Esimerkki: ModelBus

- Väliohjelmisto mallien jakamiseen
 - Mallien hajautettu päivittäminen ja synkronointi
- Mahdollistaa heterogeenisten mallinnus- ja kehitystyökalujen käytön hajautetun kehitysryhmän sisällä





Palveluiden toimintaan saattaminen väliohjelmistoalustoilla

- Ongelma: kuinka sujuvasti konfiguroida palvelut asiakkaan vaatimusten mukaisesti?
 - Esimerkissä: CRM sovellus SaaS (*software as a service*)-lähestymistavan mukaisesti
- Ratkaisu: automaattinen teknologisten palveluiden ja väliohjelmiston konfigurointi SLA -mallien mukaisesti
 - Väliohjelmistopalveluina
 - Provisioning planner: tuottaa asiakkaan vaatimuksia vastaavan palveluasennussuunnitelman
 - Provisioning engine: toteuttaa palveluasennussuunnitelmaan tarvittavat toiminnot

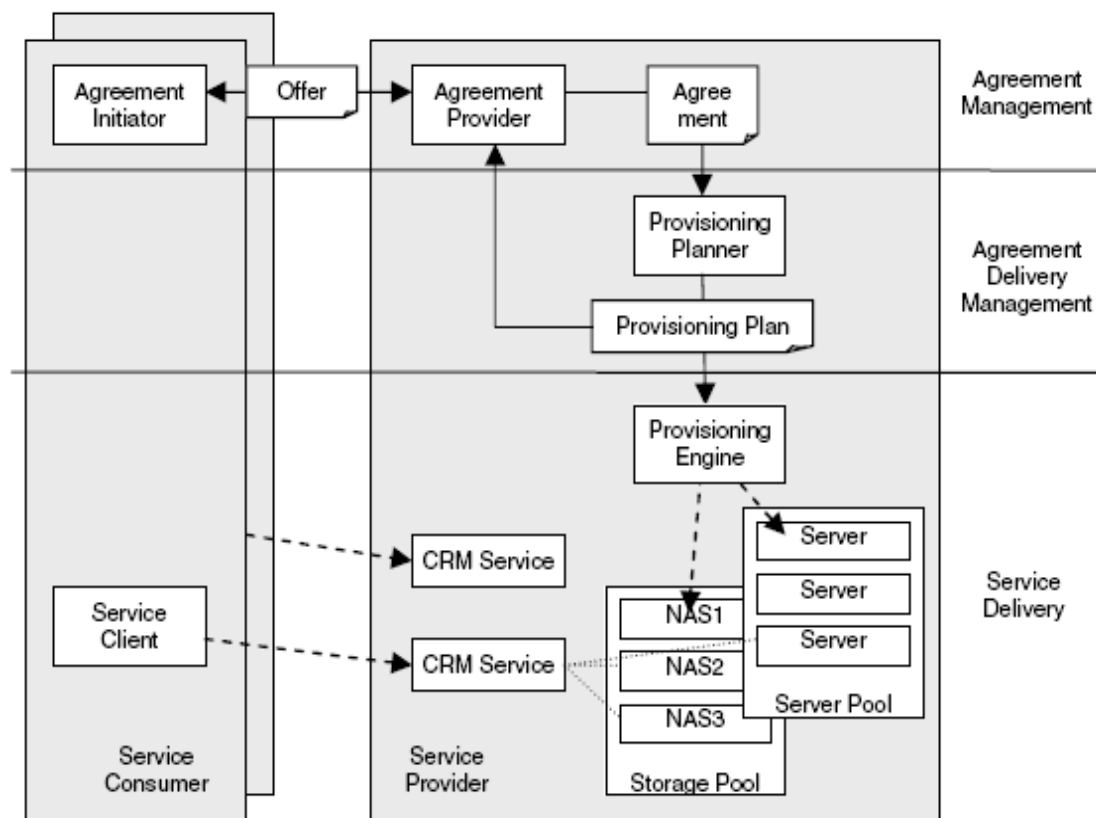


Fig. 1. Provisioning in an Agreement-Driven Service Architecture



Yhteenveto

- Ei-toiminnallisten piirteiden hallinta on tärkeää palveluekosysteemeissä, sillä ne
 - määrittävät palveluiden laatuominaisuuksia
 - Vasteaika, saatavuus, ...
 - mahdollistavat yhteistoiminnan riskien hallintaa
 - Luottamus, yksityisyys, turvallisuus, ...
 - määrittävät myös elinkaarien hallinnan ominaisuuksia
 - Ylläpidettävyys, evoluutiosieto, ...
- Palveluperustaisten järjestelmien toiminta perustuu väliohjelmistoihin
 - Toteuttavat ekosysteemin toimintaan vaadittavat infrastruktuuripalvelut
 - Tukevat
 - Elinkaarten hallintaa
 - Ei-toiminnallisten piirteiden hallintaa ja realisointia
 - (Hajautettuja) ohjelmistotuotantoprosesseja