

Kuali_PoC_raportti

Johdanto

Peppi-projektin ensimmäisessä vaiheessa selvitettiin Open ESB-tuotteen soveltuvuutta palveluiden koostamiseen ja julkaisemiseen Peppi-projektissa. PoC-projektin tuloksena OpenESB koettiin ongelmalliseksi, joten lopullista teknologiavalintaa lykättiin. Mielenkiintoiseksi vaihtoehdoksi nousi Kuali-yhteisön tarjoamat moduulit ja palvelut. Kuali PoC -projekti toteutettiin yhdessä Eduixin ja Metropolian asiantuntijoiden kanssa. Kuali PoC-projektin tavoitteet on kuvattu erillisessä dokumentissa, [PoC-projektin tavoitteet](#)

Kuali PoC-projektiin kuului osallistuminen Kuali Days -tapahtumaan, jossa Eduix esitteli edellisessä projektissa syntyneet skenaariot/vaativuudet Peppi-alustalle, luettavissa Kuali yhteisön wikissä, [skenaariot](#)

Kuali Days -tapahtuman raportti löytyy erillisestä dokumentista, [raportti Kuali-days tapahtumasta](#)

Termit

Palvelu, joukko yhteenliittyviä toimintoja, esim. opintosuunnitelman hakupalvelu.

Palvelualusta, tietojärjestelmä joka liittää palvelut yhteen.

Komponentti, palvelun toteutus.

Järjestelmä, palvelualusta ja siihen kiinteästi liittyvät muut tietojärjestelmät.

Kuali Rice -evaluointi

Palvelun kehittäjä

Palvelun kehittäjä tarkoittaa sitä tahoa, joka vastaa uuden palvelun kehittämisestä, hankkimisesta tai käyttöönotosta tai joka osallistuu varsinaiseen kehitystyöhön.

| Tilanne | Vaatus | Ongelma jos vaatimus ei toteudu | Testi | PoC tulos | Metropolian kommentit |
|---|---|---|---|---|-----------------------|
| Yksittäinen palvelu on ollut käytössä 6 vuotta ilman että palvelua on jouduttu kertaakaan muuttamaan. Palvelu tarjoaa rajapinnan ja tätä vasten on tehty 8 kriittistä integraatiota, joiden tulee olla jatkuvasti toiminnassa. Palveluun on tarkoitus tehdä uusia ominaisuuksia sekä laajentaa ja muuttaa palvelun ulkoista rajapintaa. | Palvelun uusi versio on voitava asentaa tuotantoon vanhan tilalle siten, että vanhaa rajapintaa vasten tehty 8 integraatiota säilyvät ehjinä asennuksen jälkeen ilman muutoksia integraatioihin ja mahdollinen päivityskatko on mahdollisimman lyhyt. | Palveluiden kehittäminen vaikeutuu merkittävästi. Yhden palvelun päivitys voi tuoda mukanaan huomattavan määrän piilokustannuksia. | - Toteuta ja julkaise yksinkertainen palvelu - Muuta julkaistua rajapintaa ja asenna palvelu uudelleen - Selvitä mitä ongelmia muutoksesta aiheutuu | Saatu palvelu toimimaan alustassa sekä palvelua käyttävä client. Alusta ei tue rajapintojen versiointia muuten kuin pakkausten nimiavaruutta muuttamalla. | |
| Liiketoiminta muuttuu ja tätä varten tarvitaan uusi palvelu tukemaan uutta tarvetta. Palvelulle löytyy valmis komponentti, mutta se on toteutettu eri tekniikalla/eri kielellä kuin muut alustaan liittyvät palvelut | Palvelu on voitava liittää palvelualustaan vaikka palvelun toteuttava komponentti on tehty eri tekniikalla kuin millä olemassaolevat komponentit on tehty. | Palvelua varten täytyy kehittää uusi komponentti vaikka tarjolla olisi valmis koeteltu komponentti. Pahimmassa tapauksessa komponentit voidaan ostaa vain tietyltä toimittajalta. | Koita integroida alustaan komponentteja jonkin standardin protokollan ja rajapinta teknologian avulla, esim . SOAP webservice | Ainut tapa integroida alustaan muita komponentteja on tehdä oma kääre (wrapper), johon ohjelmoidaan itse protokollaan liittyvät asiat | |

| | | | | | |
|--|--|--|--|---|--|
| Järjestelmään ollaan tekemässä uutta palvelua, joka toteutetaan koostamalla olemassaolevia palveluita. Palvelun toiminta tulee testata kunnolla muita palveluita vasten ennen sen tuotantoon asennusta. | Järjestelmästä voidaan luoda testiympäristö sekä olemassaolevia palveluita vastaavat testipalvelut. Testien aikana tietovirtaa palveluiden välillä voidaan tarkastella. | Tuotantoon asennetaan virheellisesti toimivia komponentteja | | Testaus hoidetaan stub-palveluiden avulla. viestijonossa olevat viestit ovat nähtävissä | |
| Palvelun kehittäjä on tekemässä uutta hakupalvelua, mutta ei tiedä että sellainen on jo olemassa. | Palvelualusta voidaan listata siiheen liitetyt palvelut ja niiden toimintaperiaatteet. | Tehdään päällekkäisiä palveluita, jolloin ylläpidon määrä ja kustannukset kasvavat. | Selvitä miten listaus asennetuista palveluista saadaan selville ja miten selvitetään palveluiden väliset riippuvuudet | Rice-käyttöliittymän avulla on mahdollisuus nähdä listaus asennetuista palveluista, mutta palveluiden riippuvuutta ei voi nähdä keskitetysti | |
| Palvelun kehittäjä tekee uutta palvelua ja huomaa että palvelussa voitaisiin hyödyntää olemassaolevia palveluita. Olemassaolevat palvelut eivät kuitenkaan tarjoa ulkoista rajapintaa. | Palvelualustaan on voitava tehdä palveluita siten, että muut palvelut voivat hyödyntää helposti palvelun tarjoamia rajapintoja. Vanhojen järjestelmien on pystyttävä kommunikoimaan palvelualustan kanssa tietokantayhteyden tai siirtotiedoston avulla. | Tehdään päällekkäisiä palveluita, jolloin ylläpidon määrä ja kustannukset kasvavat. | Selvitä miten alustaan integroidaan komponentteja suoran tietokantayhteyden tai siirtotiedoston avulla | Ainut tapa integroida alustaan muita komponentteja on tehdä oma kääre (wrapper), johon ohjelmoidaan itse protokollaan liittyvät asiat | |
| Palvelualustan kehitys ja tuki loppuu. Palvelualustaan on liitetty lukuisa määrä palveluita ja integraatioita. | Palvelualusta voidaan vaihtaa toiseen tuotteeseen kohtuullisella vaivalla ja ilman että palveluita joudutaan tekemään uudelleen. | Uusi palvelualusta ja integraatiot joudutaan toteuttamaan alusta alkaen, jolloin kustannukset ovat erittäin suuret. | Koita vaihtaa palvelualustan keskeisiä komponentteja | Käytännössä rice-pakkaukset paketoidaan samaan warriin jokaista palvelua kohden ja jokainen palvelu on suoraan yhteydessä rice-tietokantaan | |
| Palvelun kehittäjä on tekemässä integraatiota, joka johon liittyy 5 palvelua. Palveluiden rajapinnat eivät tue transaktioita. Palvelun kehittäjän tulee olla varma että häiriön sattuessa integraatio ei jää virheelliseen tilaan eikä tieto pääse vääristymään. | Palvelualustaan on voitava tehdä transaktionaalisia palveluita ja palvelualustan on tuettava hajautettuja transaktioita. | Monimutkaisia integraatioita ei voida tehdä tai ylläpito vaikeutuu merkittävästi. Integraation testaaminen on vaikeaa ja vauriolähtökohta kasvaa. | - Tee kaksi palvelua, joista molemmat käyttävät omaa tietokantaa. - Tee näistä palveluista koostainen palvelu - Aiheuta häiriö prosessin viimeiseksi ja selvitä molempien tietokantojen tila | Transaktionaalisuus on käytössä vain saman war-paketin sisällä | |
| Palvelualusta ei toimi odotetulla tavalla. | Palvelualustaan liittyvän dokumentaation taso tulee olla hyvä sekä mahdollinen tuki helposti saatavilla. Palvelualustan toiminta voidaan tarkastaa alustan lähdekoodista ja tarvittaessa voidaan suorittaa pikakorjaus itse palvelualustan lähdekoodiin. | Palvelut joudutaan tekemään toisin kuin on suunniteltu. Palvelua ei voida toteuttaa. Pahimmassa tapauksessa koko palvelualusta joudutaan vaihtamaan. | Arvioi dokumentaation määrä ja laatu | Dokumentaatio(1.0.3) sisältää virheitä ja palveluiden tekeminen pelkkien ohjeiden perusteella on hankalaa. PoC-projektin aikana jouduttiin kysymään neuvoa muutamaa otteeseen Kuali-sähköpostiryhmistä. Vastaukset tulivat nopeasti, mutta eivät olleet kattavia. Ainoa varma tapa saada palvelu toimimaan on lukea Rice-lähdekoodista kuinka konfiguraatiot tulee asettaa. | |

| | | | | | |
|---|--|---|--|---|--|
| On tehty uusi palvelu, joka on tarkoitus asentaa tuotantoon. | Uuden palvelun asennus ei saa aiheuttaa käyttökatkoja muissa alustaan liitetyissä palveluissa. | Ylläpito vaikeutuu ja hidastuu. Järjestelmäpäivitykset vaikeutuvat. | Arvioi kuinka paljon aikaa yksittäisen palvelun asennus vie ja vaikutukset muihin palveluihin | Asennus ei eroa normaalista java-sovelluksen asennuksesta. | |
| Komponenttien kehittämiseen tarkoitettu kehitystyökalu toimii huonosti. | Komponentti voidaan kehittää/toteuttaa usealla eri kehitystyökalulla. Komponentin rakenteen tulee olla niin yksinkertainen, että sitä voidaan tarkastella ilman erityistä kehitystyökalua. Komponentin konfiguraatiot ovat löydettävissä nopeasti. | Uusien palveluiden kehittäminen on hidasta ja kallista. | Tee palvelu ohjeissa mainitulla tavalla ja arvioi kuinka helppoa ja intuitiivista palvelun kehittäminen on. Selvitä miten palvelu asentuu alustaan | Rice-palvelun toteutus koettiin hankalaksi. Quali-moduulit on tehty lähes ainoastaan Oracle-tietokannan päälle, joka lisäsi konfiguroinnin hankaluutta. | |
| | | | | | |

Palvelualustan ylläpitäjä

Palvelualustan ylläpitäjällä tarkoitetaan tahoa, joka vastaa palvelualustan toiminnasta tai osallistuu sen ylläpitotyöhön.

| Tilanne | Vaatusus | Ongelma jos vaatusus ei toteudu | Testi | PoC tulos | Metropolian kommentit |
|--|---|---|--|-----------|-----------------------|
| Palvelualustassa on tapahtunut vakava häiriötilanne. | Ylläpitäjän tulee saada tieto tapahtuneesta häiriöstä. | Palvelun käyttökatko pitenee. | Aiheuta häiriö palvelussa ja selvitä miten vika saadaan selville | | |
| Ylläpitäjä on saanut viesti vakavasta häiriötilanteesta. | Ylläpitäjä voi nähdä yhdestä paikasta järjestelmän kokonaistilan sekä järjestelmässä tapahtuneet häiriöt ja niiden aiheuttajat. | Ylläpito vaikeutuu ja aiheuttaa pitkiä käyttökatkoja. | Arvioi miten nopeasti häiriön aiheuttaja saadaan paikallistettua | | |
| Ylläpitäjä A on saanut ilmoituksen, että palvelu A ei toimi oikein. Ylläpitäjä B on saanut ilmoituksen, että palvelu B ei toimi oikein. Ilmoituksista ei kuitenkaan näe häiriön syytä. Palvelut A ja B on integroitu kolmanteen palveluun, joka aiheuttaa häiriön molemmissa järjestelmissä. | Ylläpitäjät näkevät helposti järjestelmien väliset riippuvuudet ja löytävät häiriöaiheuttajan nopeasti. | Ylläpitäjät tekevät turhaan työtä etsiessään häiriön syytä väärästä paikasta. | | | |
| Palveluiden käyttö kasvaa rajusti. | Palvelualustan suorituskykyä voidaan lisätä klusteroinnin avulla. | Koko palvelualusta joudutaan vaihtamaan tai osasta palveluita luopumaan. | | | |
| Palvelualustaan on kytketty uusi palvelu, jossa oleva vika aiheuttaa liikaa kuormaa palvelualustalle säännöllisin väliajoin. | Palvelualustan hallintatyökaluilla voidaan tarkastella palvelun tilaa ja siihen liittyviä tietovirtoja. | Ongelmien selvitys vaikeutuu. Kynnys asentaa uusia palveluita tai päivityksiä kasvaa. | | | |

| | | | | | |
|--|--|---|--|--|--|
| Palvelu joudutaan päivittämään. Palvelun rajapintoja käyttää kriittinen integraatio, joka lähettää palvelulle viestejä asynkronisesti. Integraation täytyy pystyä toimimaan huoltokatkon ajan. | Palvelun päivityskatkos ei saa aiheuttaa häiriötilannetta integraatioihin mikäli käytetään asynkronista tiedonsiirtoa. | Lisää ylläpitotyötä ja kynnys asentaa uusia palveluita tai päivityksiä kasvaa. | | | |
| Järjestelmään tehdään tietoturva-auditointi tai epäillään että järjestelmään on yritetty tehdä tietoturvahyökkäys. | Palvelualustaan liitetty palvelut ja integraatiot voidaan listata ja tarkastaa niiden näkyvyys sisäverkon ulkopuolelle. Palvelualustassa tapahtuvaa tietovirtaa voidaan tarkastella. Ylläpitä saa kokonaiskäsitelyksen järjestelmässä käytössä olevista tietoturvaan liittyvistä teknologioista. | Riski hyökkäyksen onnistumisesta kasvaa, jos tietoturvan varmistaminen keskitetysti ei onnistu. | | | |
| Palvelimelle on asennettu 3 palvelua, jotka on liitetty toisiinsa. Palvelimen osoite vaihtuu. | Palveluiden on pystyttävä olemaan yhteydessä toisiinsa vaikka palvelimen osoite vaihtuisi, ilman että palveluita joudutaan konfiguroimaan uudelleen. | Ylläpidon määrä kasvaa. Riski siihen, että jokin vanha palvelu hajoaa muutoksen yhteydessä on erittäin suuri. | | | |
| | | | | | |
| | | | | | |

Yhteenveto Eduix

Yhteenveto Metropolia

ServiceMix/Student -evaluointi

Projektin aikana huomattiin, että Rice-alusta(KSB) ei välttämättä tällä hetkellä pysty vastaamaan Peppi-projektin tarpeisiin palveluiden toteuttamisessa ja koostamisessa, joten Rice-KSB korvattiin suositulla open source ServiceMix-tuotteella. ServiceMix valittiin, koska se käyttää samoja alikomponentteja kuin Rice-alusta, kts. kappale "Ehdotus Peppi-projektin tekniseksi arkkitehtuuriksi".

Palvelun kehittäjä

Palvelun kehittäjä tarkoittaa sitä tahoa, joka vastaa uuden palvelun kehittämisestä, hankkimisesta tai käyttöönotosta tai joka osallistuu varsinaiseen kehitystyöhön.

| Tilanne | Vaatus | Ongelma jos vaatimus ei toteudu | Testi | PoC tulos | Metropolian kommentit |
|---|--|--|---|---|-----------------------|
| Yksittäinen palvelu on ollut käytössä 6 vuotta ilman että palvelua on jouduttu kertaakaan muuttamaan. Palvelu tarjoaa rajapinnan ja tätä vasten on tehty 8 kriittistä integraatiota, joiden tulee olla jatkuvasti toiminnassa. Palveluun on tarkoitus tehdä uusia ominaisuuksia sekä laajentaa ja muuttaa palvelun ulkoista rajapintaa. | Palvelun uusi versio on voitava asentaa tuotantoon vanhan tilalle siten, että vanhaa rajapintaa vasten tehdyt 8 integraatiota säilyvät ehjinä asennuksen jälkeen ilman muutoksia integraatioihin ja mahdollinen päivityskatko on mahdollisimman lyhyt. | Palveluiden kehittäminen vaikeutuu merkittävästi. Yhden palvelun päivitys voi tuoda mukanaan huomattavan määrän piilokustannuksia. | - Toteuta ja julkaise yksinkertainen palvelu - Muuta julkaistua rajapintaa ja asenna palvelu uudelleen - Selvitä mitä ongelmia muutoksesta aiheutuu | Yksinkertaisen palvelun asentaminen ja julkaiseminen koettiin suoraviivaiseksi. Alustassa on mahdollista ajaa eri versioita samasta paketista pelkästään versionumeroa muuttamalla. | |

| | | | | | |
|--|--|---|--|---|--|
| Liiketoiminta muuttuu ja tätä varten tarvitaan uusi palvelu tukemaan uutta tarvetta. Palvelulle löytyy valmis komponentti, mutta se on toteutettu eri tekniikalla/eri kielellä kuin muut alustaan liittyvät palvelut | Palvelu on voitava liittää palvelualustaan vaikka palvelun toteuttava komponentti on tehty eri tekniikalla kuin millä olemassaolevat komponentit on tehty. | Palvelua varten täytyy kehittää uusi komponentti vaikka tarjolla olisi valmis koeteltu komponentti. Pahimmassa tapauksessa komponentit voidaan ostaa vain tietyltä toimittajalta. | Koita integroida alustaan komponentteja jonkin standardin protokollan ja rajapinta teknologian avulla, esim . SOAP webservice | Alusta tarjoaa useita tiettyyn protokollaan liittyviä komponentteja. Ajanpuutteen vuoksi Winhan webservice-rajapintaa ei ehditty tarpeeksi testaamaan. | |
| Järjestelmään ollaan tekemässä uutta palvelua, joka toteutetaan koostamalla olemassaolevia palveluita. Palvelun toiminta tulee testata kunnolla muita palveluita vasten ennen sen tuotantoon asennusta. | Järjestelmästä voidaan luoda testiympäristö sekä olemassaolevia palveluita vastaavat testipalvelut. Testien aikana tietovirtaa palveluiden välillä voidaan tarkastella. | Tuotantoon asennetaan virheellisesti toimivia komponentteja | | Testausta voidaan tehdä stub-palveluiden avulla. | |
| Palvelun kehittäjä on tekemässä uutta hakupalvelua, mutta ei tiedä että sellainen on jo olemassa. | Palvelualusta voidaan listata siiheen liitetyt palvelut ja niiden toimintaperiaatteet. | Tehdään päällekkäisiä palveluita, jolloin ylläpidon määrä ja kustannukset kasvavat. | Selvitä miten listaus asennetuista palveluista saadaan selville ja miten selvitetään palveluiden väliset riippuvuudet | Palvelut voidaan listata ja myös niiden riippuvuudet muista palveluista on nähtävissä. Monitorointiin tarkoitettua konsolia on myös helppo laajentaa, konsolin laajentaminen testattiin. | |
| Palvelun kehittäjä tekee uutta palvelua ja huomaa että palvelussa voitaisiin hyödyntää olemassaolevia palveluita. Olemassaolevat palvelut eivät kuitenkaan tarjoa ulkoista rajapintaa. | Palvelualustaan on voitava tehdä palveluita siten, että muut palvelut voivat hyödyntää helposti palvelun tarjoamia rajapintoja. Vanhojen järjestelmien on pystyttävä kommunikoimaan palvelualustan kanssa tietokantayhteyden tai siirtotiedoston avulla. | Tehdään päällekkäisiä palveluita, jolloin ylläpidon määrä ja kustannukset kasvavat. | Selvitä miten alustaan integroidaan komponentteja suoran tietokantayhteyden tai siirtotiedoston avulla | Alusta tarjoaa useita tiettyyn protokollaan liittyviä komponentteja. | |
| Palvelualustan kehitys ja tuki loppuu. Palvelualustaan on liitetty lukuisa määrä palveluita ja integraatioita. | Palvelualusta voidaan vaihtaa toiseen tuotteeseen kohtuullisella vaivalla ja ilman että palveluita joudutaan tekemään uudelleen. | Uusi palvelualusta ja integraatiot joudutaan toteuttamaan alusta alkaen, jolloin kustannukset ovat erittäin suuret. | Koita vaihtaa palvelualustan keskeisiä komponentteja | Vaihdettu alustan palveluita pyörittävä runtime toiseksi ongelmitta. Huomioitavaa että servicemix toimii felix:llä kun taas kaupallinen versio fuse esb toimii equinox:lla | |
| Palvelun kehittäjä on tekemässä integraatiota, joka johon liittyy 5 palvelua. Palveluiden rajapinnat eivät tue transaktioita. Palvelun kehittäjän tulee olla varma että häiriön sattuessa integraatio ei jää virheelliseen tilaan eikä tieto pääse vääristymään. | Palvelualustaan on voitava tehdä transaktionaalisia palveluita ja palvelualustan on tuettava hajautettuja transaktioita. | Monimutkaisia integraatioita ei voida tehdä tai ylläpito vaikeutuu merkittävästi. Integraation testaaminen on vaikeaa ja vauriolähtökohta kasvaa. | - Tee kaksi palvelua, joista molemmat käyttävät omaa tietokantaa. - Tee näistä palveluista koosteinen palvelu - Aiheuta häiriö prosessin viimeiseksi ja selvitä molempien tietokantojen tila | Saatu transaktio toimimaan yksittäisen palvelun sisällä helposti. Transaktioiden pitäisi toimia ongelmitta osgi-säiliön sisällä, mutta ajanpuutteen vuoksi transaktiot kahden eri tietokantaa käyttävän palvelun välillä jäi testaamatta. | |

| | | | | | |
|---|--|--|--|---|--|
| Palvelualusta ei toimi odotetulla tavalla. | Palvelualustaan liittyvän dokumentaation taso tulee olla hyvä sekä mahdollinen tuki helposti saatavilla. Palvelualustan toiminta voidaan tarkastaa alustan lähdekoodista ja tarvittaessa voidaan suorittaa pikakorjaus itse palvelualustan lähdekoodiin. | Palvelut joudutaan tekemään toisin kuin on suunniteltu. Palvelua ei voida toteuttaa. Pahimmassa tapauksessa koko palvelualusta joudutaan vaihtamaan. | Arvioi dokumentaation määrä ja laatu | ServiceMix dokumentaatio vanhentunutta, mutta Fuse esb-sivuilla dokumentaatiota varsin hyvin. Lisäksi ServiceMixiin on saatavilla koulutusta. | |
| On tehty uusi palvelu, joka on tarkoitus asentaa tuotantoon. | Uuden palvelun asennus ei saa aiheuttaa käyttökatkoja muissa alustan liitetyissä palveluissa. | Ylläpito vaikeutuu ja hidastuu. Järjestelmäpäivitykset vaikeutuvat. | Arvioi kuinka paljon aikaa yksittäisen palvelun asennus vie ja vaikutukset muihin palveluihin | Asennus ei aiheuta katkoksia muiden palveluiden käyttöön. Jokaisella palvelulla on lisäksi oma classloading, jolloin kirjastojen lataukset eivät ole yhtä kriittisiä kuin normaalissa java-sovelluspalvelimessa | |
| Komponenttien kehittämiseen tarkoitettu kehitystyökalu toimii huonosti. | Komponentti voidaan kehittää/toteuttaa usealla eri kehitystyökalulla. Komponentin rakenteen tulee olla niin yksinkertainen, että sitä voidaan tarkastella ilman erityistä kehitystyökalua. Komponentin konfiguraatiot ovat löydettävissä nopeasti. | Uusien palveluiden kehittäminen on hidasta ja kallista. | Tee palvelu ohjeissa mainitulla tavalla ja arvioi kuinka helppoa ja intuitiivista palvelun kehittäminen on. Selvitä miten palvelu asennuu alustaan | Palvelun asentuminen alustaan toimii suositulla maven kehitystyökalulla. palveluiden konfiguraatiot on löydettävissä helposti. | |
| | | | | | |

Palvelualustan ylläpitäjä

Palvelualustan ylläpitäjällä tarkoitetaan tahoja, joka vastaa palvelualustan toiminnasta tai osallistuu sen ylläpitotyöhön.

| Tilanne | Vaatus | Ongelma jos vaatimus ei toteudu | Testi | PoC tulos | Metropolian kommentit |
|--|---|---|--|---|-----------------------|
| Palvelualustassa on tapahtunut vakava häiriötilanne. | Ylläpitäjän tulee saada tieto tapahtuneesta häiriöstä. | Palvelun käyttökatko pitenee. | Aiheuta häiriö palvelussa ja selvitä miten vika saadaan selville | Fuse HQ-tuotteen avulla järjestelmään voidaan tehdä häilytyksiä. Tämän testaamista ei kuitenkaan ehditty tehdä. | |
| Ylläpitäjä on saanut viesti vakavasta häiriötilanteesta. | Ylläpitäjä voi nähdä yhdestä paikasta järjestelmän kokonaistilan sekä järjestelmässä tapahtuneet häiriöt ja niiden aiheuttajat. | Ylläpito vaikeutuu ja aiheuttaa pitkiä käyttökatkoja. | Arvioi miten nopeasti häiriön aiheuttaja saadaan paikallistettua | Häiriöt voidaan jäljittää lokin perusteella. Lisäksi palveluiden tila voidaan nähdä listauksesta. | |

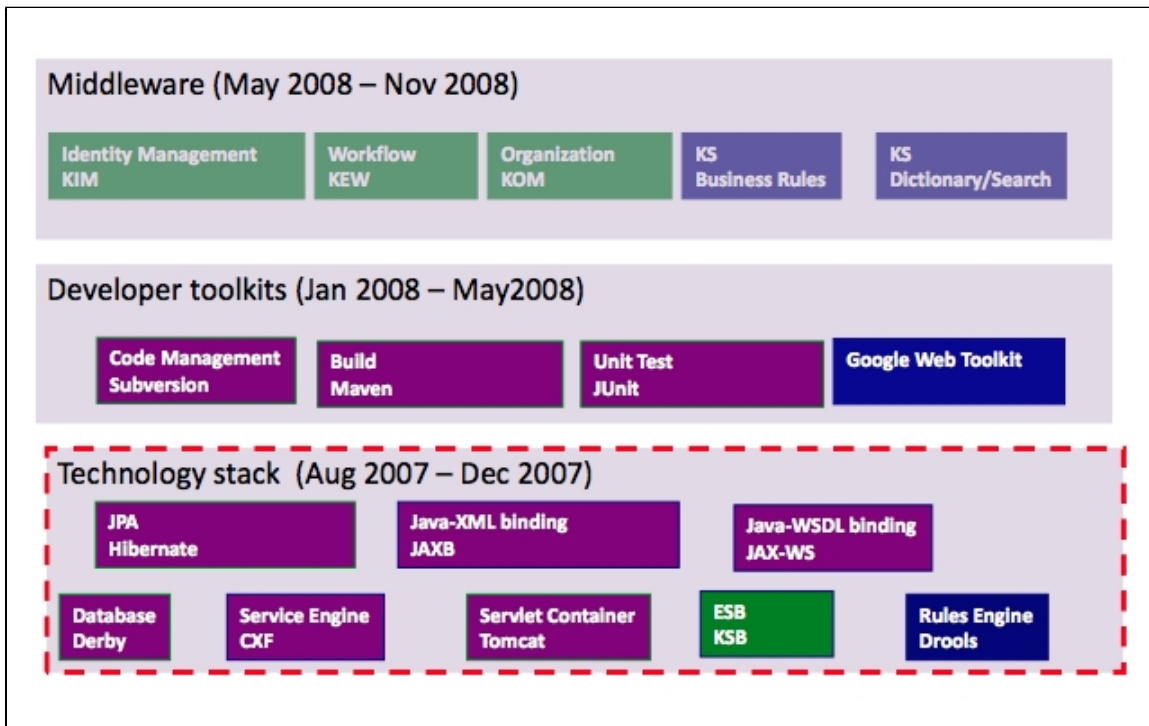
| | | | | | |
|--|--|--|--|--|--|
| Ylläpitäjä A on saanut ilmoituksen, että palvelu A ei toimi oikein. Ylläpitäjä B on saanut ilmoituksen, että palvelu B ei toimi oikein. Ilmoituksista ei kuitenkaan näe häiriön syytä. Palvelut A ja B on integroitu kolmanteen palveluun, joka aiheuttaa häiriön molemmissa järjestelmissä. | Ylläpitäjät näkevät helposti järjestelmien väliset riippuvuudet ja löytävät häiriöaiheuttajan nopeasti. | Ylläpitäjät tekevät turhaan työtä etsiessään häiriön syytä väärästä paikasta. | | Palveluiden riippuvuussuhteet voidaan selvittää helposti. | |
| Palveluiden käyttö kasvaa rajusti. | Palvelualustan suorituskykyä voidaan lisätä klusteroinnin avulla. | Koko palvelualusta joudutaan vaihtamaan tai osasta palveluita luopumaan. | | Testattu klusterointia lähettämällä viestejä eri servicemix-instanssien välillä. | |
| Palvelualustaan on kytketty uusi palvelu, jossa oleva vika aiheuttaa liikaa kuormaa palvelualustalle säännöllisin väliajoin. | Palvelualustan hallintatyökaluilla voidaan tarkastella palvelun tilaa ja siihen liittyviä tietovirtoja. | Ongelmien selvitys vaikeutuu. Kynnys asentaa uusia palveluita tai päivityksiä kasvaa. | | Viestien lähetykseen ja rajapintojen toimintaan on mahdollista liittää toiminnallisuutta interceptor-luokkien avulla. | |
| Palvelu joudutaan päivittämään. Palvelun rajapintoja käyttää kriittinen integraatio, joka lähettää palvelulle viestejä asynkronisesti. Integraation täytyy pystyä toimimaan huoltokatkon ajan. | Palvelun päivityskatkos ei saa aiheuttaa häiriötilannetta integraatioihin mikäli käytetään asynkronista tiedonsiirtoa. | Lisää ylläpitotyötä ja kynnys asentaa uusia palveluita tai päivityksiä kasvaa. | | Mikäli integraatio käyttää servicemixin jms-jonoa(activeMQ), on viestit mahdollista persistoida lähetyksen ajaksi. Tämän avulla varmistutaan että viesti toimitetaan perille käyttökatkoksista huolimatta. | |
| Järjestelmään tehdään tietoturva-auditointi tai epäillään että järjestelmään on yritetty tehdä tietoturvahyökkäys. | Palvelualustaan liitetyt palvelut ja integraatiot voidaan listata ja tarkastaa niiden näkyvyys sisäverkon ulkopuolelle. Palvelualustassa tapahtuvaa tietovirtaa voidaan tarkastella. Ylläpitä saa kokonaiskäsityksen järjestelmässä käytössä olevista tietoturvaan liittyvistä teknologioista. | Riski hyökkäyksen onnistumisesta kasvaa, jos tietoturvan varmistaminen keskitetysti ei onnistu. | | Palvelut voidaan listata helposti. | |
| Palvelimelle on asennettu 3 palvelua, jotka on liitetty toisiinsa. Palvelimen osoite vaihtuu. | Palveluiden on pystyttävä olemaan yhteydessä toisiinsa vaikka palvelimen osoite vaihtuisi, ilman että palveluita joudutaan konfiguroimaan uudelleen. | Ylläpidon määrä kasvaa. Riski siihen, että jokin vanha palvelu hajoo muutoksen yhteydessä on erittäin suuri. | | Osgi-palveluita ajetaan samassa JVM:ssä ja ne hakevat viittaukset muihin palveluihin palvelurekisteristä, jolloin osoitteita ei tarvitse palveluihin konfiguroida. Käytettäessä viestin välitykseen JMS-jonoa on osoitteet mahdollista piilottaa palveluilta klusteroinnin avulla. | |
| | | | | | |

Yhteenveto Eduix

Yhteenveto metropolia

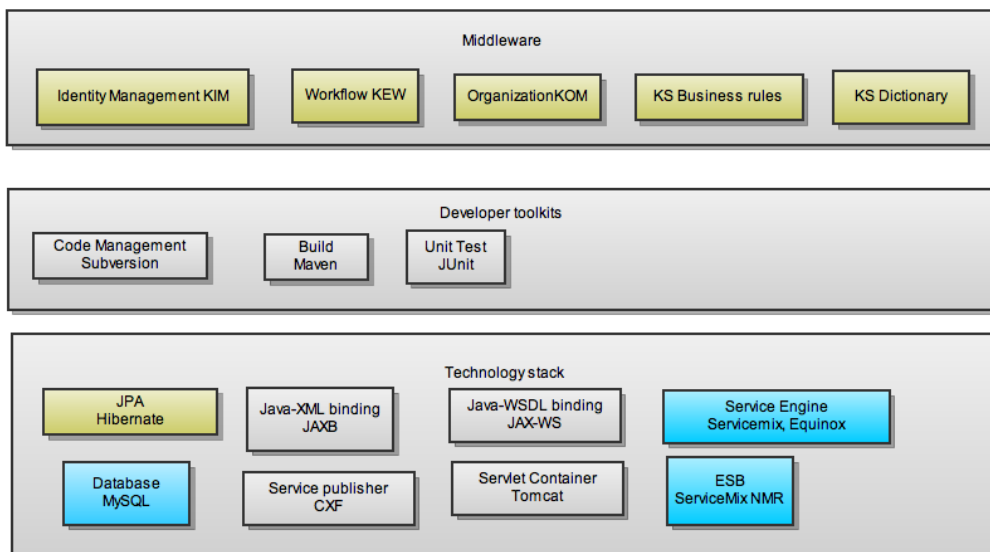
Ehdotus Peppi-projektin tekniseksi arkkitehtuuriksi

PoC-projektin aikana selvisi ettei Kuali Rice-projektia voida käyttää palveluiden toteutukseen. Student-projektin lähdekoodi todettiin kuitenkin laadukkaaksi ja rajapinnat on eristetty moduulien sisäisistä ratkaisuista hyvin. Peppi-projektin ja Kuali-student projektien tekniset ratkaisut ovat varsin samankaltaiset.



Kuali Rice/Student teknologiat, lähde <http://kuali.org/ks/documentation>

Peppi-projektissa käytettävät teknologiat, vihreällä olevia teknologioita ei keretty PoC-projektin aikana tarpeeksi testaamaan, mutta joita todennäköisesti tullaan toteutusvaiheessa käyttämään. Sinisellä olevat ovat uusia verrattuna Kuali-alustaan, mutta joiden käyttöön todennäköisesti tullaan myös Kuali-projektissa myöhemmin siirtymään, arvio perustuu Kuali Days -tapahtumasta saatuihin kokemuksiin ja kualin wikissä ja keskusteluryhmissä käytyihin keskusteluihin.



Näkymäkerros

Näkymäkerros perustuu johonkin yleisesti käytössä olevaan teknologiaan, mutta se rajataan seuraavasti:

- Näkymäkerros ei ole transaktionaalinen. Mikäli näkymäkerroksessa kutsutaan eri palveluita ja huomataan että kutsut tulee suorittaa transaktiossa, tarkoittaa se että palvelukerrokseen on tehtävä uusi palvelu tätä toimintoa varten.
- Viestit näkymäkerroksen ja palvelualustan välillä ovat XDW-määrittymiin perustuvia XML-viestejä. Huomioitavaa on että XDW-määrittymiset eivät välttämättä sovellu sellaisenaan suunnittelun työvälineisiin, vaan skeemaa täytyy mahdollisesti laajentaa.
- Oikeustarkistuksia tehdään ainoastaan näkymien suodattamista varten, varsinaiset oikeustarkistukset tehdään palveluissa erikseen.

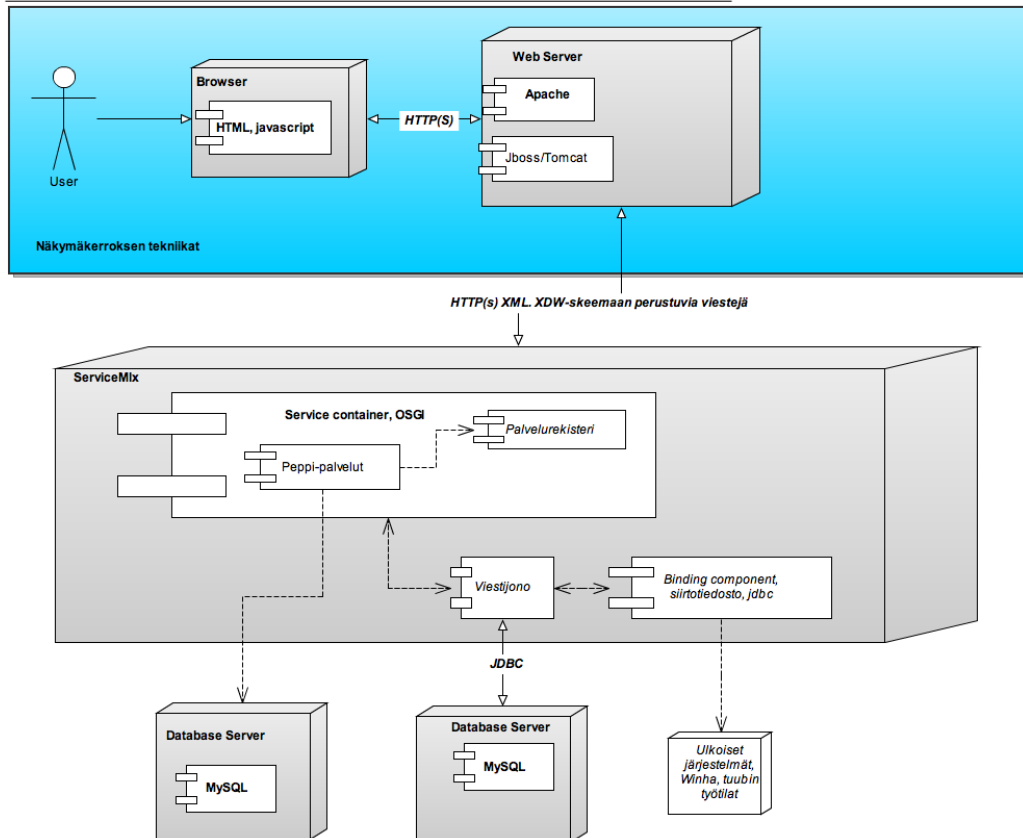
Peppi-palvelut

Peppi-palvelut toteutetaan OSGI-paketteina, jotka ovat yhteydessä toisiinsa ainoastaan rajapinta-luokkien avulla. Palvelut julkaisevat XDW-määrittymiin perustuvia rajapintoja näkymäkerroksen käyttöön. Palveluiden toteutuksessa hyödynnetään Quali Student -projektin lähdekoodia ja ratkaisumalleja. Suunnittelussa huomioidaan, että tulevaisuudessa Quali-projektissa Rice ksb saatetaan myös korvata ServiceMix-tuotteella.

Integraatiot ulkopuolisiin järjestelmiin

Peppi-palvelut integroidaan esim. Winhaan ServiceMixin viestijonon avulla. Viestit on mahdollista säilöä tietokantaan lähetyksen ajaksi, joka takaa viestin lähetyksen perille. Mikäli viestintäohjelmassa tapahtuu häiriö, jää viesti tietokantaan ja teknisen pääkäyttäjän tehtävänä on korjata häiriötilanne.

Peppi-komponentit



Metropolian arvio ehdotetusta arkkitehtuurista

