

Oikeusministeriö
Uusi vaalitietojärjestelmä
Teknisen arkkitehtuurin päivitys

Oikeusministeriö

Uusi vaalitietojärjestelmä

Teknisen arkkitehtuurin päivitys

10.10.2005

Versiohistoria:

Versio	Pvm	Laatijat	Muutokset
0.0.1	6.10.2003	PLa	Dokumenttipohja ja alustava rakenne
0.1	9.10.2003	PLa & PKa	Sisällys
0.1.1	15.10.2003	PLa	Taustamateriaalin läpikäyntiä
0.1.5	6.11.2003	PLa	Asioiden lisäämistä
0.9	9.12.2003	PKa	Yhdistetty kokonaisuus. Toimitettu asiakkaalle.
0.91	2.1.2003	PKa	Lisätty riskianalyysistä puuttuneet osat

Tarkastanut: _____

Hyväksynyt: _____

Tallennus:

C:\Ohjelmatiedostot\TeamWARE\Toimisto\TEMP\MAIL\MV8\VAT_Arkk_Paivitys_v0.91.doc

DOKUMENTIN TILA:Alustava Valmis Tarkastettu Hyväksytty

SISÄLLYS

YHTEENVETO	5
1. JOHDANTO	6
1.1 DOKUMENTIN TARKOITUS	6
1.2 RAJAUKSET	6
1.3 TERMIT JA LYHENTEET	6
1.4 VIITTEET	7
2. UUDEN SUKUPOLVEN SOVELLUSPALVELINTEN MAHDOLLISUUDET	9
2.1 SOVELLUSPALVELINMARKKINAT	9
2.2 PÄIVITETTY IBM:N TARJONTA	10
2.3 PÄIVITETTY BEA:N TARJONTA.....	10
2.4 ORACLEN SOVELLUSPALVELINTARJONTA.....	11
2.5 AVOIMEN LÄHDEKOODIN J2EE-SOVELLUSPALVELINTARJONTA	12
2.6 YHTEENVETO SOVELLUSPALVELINMARKKINOISTA.....	14
3. PORTAALITEKNOLOGIAN JA INTEGRAATION MAHDOLLISUUDET	15
3.1 PORTAALITEKNOLOGIA	15
3.2 INTEGRAATIOTEKNIKAT J2EE-YMPÄRISTÖSSÄ.....	16
3.3 WEB SERVICES	17
4. LYHYT RISKIANALYYSI.....	19
4.1 TEKNOLOGIARISKIT.....	19
4.2 TUOTETASON RISKIT	20
4.3 RATKAISUTASON RISKIT.....	21
4.4 J2EE-PROJEKTIRISKIT	21
4.5 TOIMITTAJIIN LIITTYVÄT RISKIT	23
4.6 NYKYRATKAISUUN JA SIIRTÄMÄPOLKUUN LIITTYVÄT RISKIT	23
5. UUDEN VAALITIE TOJÄRJESTELMÄN TOTEUTUS.....	25
5.1 KOKONAISARKKITEHTUURIN JÄSENNYS	25
5.2 UUDELEENKÄYTÖN MAHDOLLISUUDET	26
5.3 ETENEMISMALLI.....	27

KUVALUETTELO

Kuva 1. Uuden vaalitietojärjestelmän kokonaisarkkitehtuuri.....	25
---	----

YHTEENVETO

Tämä raportti tarkentaa 23.1.2002 luovutetun uuden vaalitietojärjestelmän teknisen arkkitehtuurin jatkomäärittelyä siten, että teknologiassa ja tuotteissa tapahtuneet uudistukset on käydään läpi pääpiirteittäin ja vaalitietojärjestelmän kannalta keskeiset muutokset tuodaan esiin. Edellisen raportin kohdealueena olleiden IBM:n ja BEA:n sovelluspalvelintarjonnan lisäksi tarkastellaan myös Oraclen tuotetarjontaa ja avoimeen lähdekoodiin perustuvia ratkaisuja.

Perinteisen sovelluspalvelintoiminnallisuuden lisäksi tarkastellaan myös portaali-teknologiaa ja uuden sukupolven integraatioteknologian antamia mahdollisuuksia. Jälkimmäiseen tarkasteluun sisältyvät myös Web Services -teknologiat.

Uuden vaalitietojärjestelmän toteuttaminen J2EE-arkkitehtuurin avulla on yleisen suuntauksen mukainen valinta, mutta siihen sisältyy myös joukko riskejä. Näitä riskejä analysoidaan ja riskien hallintaan esitetään myös joukko ratkaisuja.

Lopuksi käsitellään uuden vaalitietojärjestelmän toteutukseen liittyviä kysymyksiä. Kokonaisarkkitehtuurille esitetään malli, jossa toiminnallisuus on jaettu itsenäisiin osajärjestelmiin ja nämä puolestaan liitetään toisiinsa erillisen integraatoratkaisun avulla. Näin voidaan pienentää kokonaishankkeen riskejä ja varautua mahdollisiin muutoksiin tavoitteena olevassa toiminnallisuudessa. Uudelleenkäytön osalta tarkastellaan erilaisia malleja, joissa toteutusprojektia voidaan tehostaa erilaisilla valmiilla tai puolivalmiilla ratkaisuilla. Viimeiseksi esitellään karkean tason etenemismalli, jonka avulla voidaan hallitusti siirtyä estettyyn osajärjestelmistä koostuvaan uuteen vaalitietojärjestelmään..

1. JOHDANTO

1.1 Dokumentin tarkoitus

Dokumentin on tarkoituksena on tarkentaa 23.1.2002 päivätyn uuden vaalitietojärjestelmän teknisen arkkitehtuurin jatkomäärittelyn /1/ sisältö vastaamaan teknologiassa tapahtuneita uudistuksia. Erityisesti käsitellään seuraavia osa-alueita:

- Sovelluspalvelinohjelmistojen uusien versioiden mukanaan tuomat muutokset toiminnallisuuteen, kehitysvälineisiin ja arkkitehtuuriratkaisuihin
- Viimeaikaisen teknologiakehityksen mukanaan tuomat muutokset, erityisesti portaaliteknologian vaikutukset ja integraatiotekniikoiden tuomat mahdollisuudet
- J2EE-arkkitehtuurin uudelleenikäytön tuomat mahdollisuudet ja kokemukset
- Uusien Web Services -tekniikoiden vaikutus arkkitehtuuriin
- Lyhyt riskianalyysi J2EE-arkkitehtuuriin pohjautuvien hankkeiden toteutustyöstä.

Raportissa pyritään tuomaan esiin erityisesti käytännön kokemuksista saatuja oppeja sekä uudistushankkeen onnistumiseen tähtääviä havaintoja.

1.2 Rajaukset

Dokumentti ei korvaa aiemmin tehtyä raporttia /1/, vaan tarjoaa siihen lisäinformaatiota useilta osa-alueilta. Kokonaisuuden hahmottaminen edellyttää perehtymistä molempiin dokumentteihin. Dokumentissa ei pyritä tarkastelemaan uuden vaalitietojärjestelmän varsinaista toiminnallisuutta, vaan käsittely rajataan ennen kaikkea J2EE-arkkitehtuurin hyödyntämiseen uuden vaalitietojärjestelmän tulevassa toteutuksessa.

1.3 Termit ja lyhenteet

Termi	Selite
APS	Application Platform Suite. Laajennetulla toiminnallisuudella varustettu sovelluspalvelin.
API	Application Programming Interface. Sovellusrajpinta.
B2B	Business to business. Yritystenvälinen (integraatio).
BAM	Business Activity Monitoring. Liiketoiminnan reaaliaikainen seuranta.
BPM	Business Process Monitoring. Liiketoimintaprosessien hallinta.
EJB	Enterprise Java Bean. Sunin määrittelemä Java-pohjainen tapahtumankäsittely-ympäristö ja siinä olevat komponentit.
Integraatiobroker	Järjestelmäintegraatioon tarkoitettu ohjelmisto, joka ohjaa, muuntaa ja hallitsee näiden välisiä tietovirtoja.
J2EE	Java 2 Enterprise Edition. Sunin määrittelemä yritystason järjestelmiin tarkoitettu teknologiakokonaisuus

Termi	Selite
JCA	J2EE Connector Architecture. Tekniikka liittää Java-sovellukset olemassa oleviin järjestelmiin.
JMS	Java Message Service. Java-ympäristöön määritelty sanomajonorajapinta.
JSR	Java Specification Request. Java-yhteisön standardikehityksen välivaihe.
MDB	Message Driven Bean. EJB-tapahtumankäsittely-ympäristössä toimiva sanomankäsittelijä.
OAS 10g	Oracle Application Server 10g. Oraclen sovelluspalvelintuote(perhe)
OASIS	Organization for the Advancement of Structured Information Standards. Sähköisen liiketoiminnan standardeja kehittävä organisaatio.
SOAP	Simple Object Access Protocol. Web Services -ympäristön sovellustason kommunikointiprotokolla.
SPE	Software Performance Engineering. Kokonaisvaltainen sovelluksen toteutusyökin aikana tapahtuva suorituskyvyn ja skaalautuvuuden varmistusmenettely.
UDDI	Universal Description, Discovery and Integration, Hakemistomäärittely Web Services -ympäristöön.
W3C	World Wide Web Consortium. Web-tekniologian standardeja kehittävä organisaatio.
Web Services	Joukko protokollatason määrittäjiä, joiden avulla voidaan toteuttaa hyvinkin monipuolisia järjestelmien ja organisaatioiden välisiä kommunikointimalleja
WebLogic	BEA:n sovelluspalvelintuote(perhe)
WebSphere	IBM:n sovelluspalvelintuote(perhe)
WSAD	WebSphere Application Studio. IBM:n sovelluskehitysympäristö J2EE-sovellusten toteuttamiseen.
WSDL	Web Services Description Language. Kuvausmekanismi Web Services -kommunikoinnin yksityiskohtien määrittämiseksi
WSRP	Web Services for Remote Portals. Web Services -tekniologioihin liittyvä portaalien yhteiskäyttöä tukeva standardi.
XML	Extensible Markup Language. Standardoitu tapa esittää rakenteisia dokumentteja.

1.4 Viitteet

- /1/ Uuden vaalitietojärjestelmän (VAT) teknisen arkkitehtuurin jatkomäärittely, Versio 1.6H, 23.1.2002.
- /2/ Uuden vaalitietojärjestelmän (VAT) teknisen arkkitehtuurin jatkomäärittely, tuotevertailu, 17.12.2001.
- /3/ Vaalitietojärjestelmän kehittäminen, Oikeusministeriö, Arto Jääskeläinen, 19.5.2003.
- /4/ Suomen vaalijärjestelmä, yleiskuvaus, kesäkuu 2002.
- /5/ Vaalitietojärjestelmän esimäärittely, 6.11.2003, Versio 1.31.
- /6/ AIM and Portal Market Forecast and Trends, 2002-2007, Gartner Dataquest, 18.8.2003.

2. UUDEN SUKUPOLVEN SOVELLUSPALVELINTEN MAHDOLLISUUDET

2.1 Sovelluspalvelinmarkkinat

Sovelluspalvelinmarkkinat ovat kehittyneet suuntaan, jossa sovelluspalvelin-toiminnallisuuden lisäksi tuotteet sisältävät joukon hyödyllisiä ja varsin laajoja lisäominaisuuksia. Tyypillinen kokonaisuus sisältääkin seuraavat piirteet:

- J2EE-standardin mukainen sovelluspalvelin
- Integraatiobroker
- Yritysportaali
- Kokonaisuuteen integroitu kehitysvälineistö

Syntyneitä uutta kokonaisuutta tutkimuslaitos Gartner kutsuu nimellä APS (application platform suite). Vuosina 2001-2002 puhdas sovelluspalvelinten dollarimääräinen lisenssimarkkina pienentyi vuosittain 7.4 %, mutta APS-markkinoiden markkina puolestaan kasvoi vuositasolla edelleen 11.7 % /6/.

APS-kehityksen arvioidaan laajentuvat myös uusiin alueisiin. Näitä ovat business-to-business-integraatio (B2B), business processing monitoring (BPM) and business activity monitoring (BAM). Näiden alueiden sovellusten toteuttaminen on mahdollista yhdellä ja samalla laajalla kehitysympäristöllä. Myös Web Services -tekniikat ovat siirtymässä osaksi APS-pakettia.

Seuraavassa taulukossa on keskeisten toimittajien markkinaosuuksia vuonna 2002 Gartner Dataquestin raportin mukaan /6/:

Toimittaja	Markkinaosuus sovelluspalvelinmarkkinoilla	Markkinaosuus koko APS-markkinoilla
IBM	37.7	20.5
BEA	30.0	11.3
Sun One	4.0	-
Hitachi	3.0	-
NEC	2.9	-
Oracle	< 1	6.2
Fujitsu	< 1	4.5

Alla on päivitetty kuvaus IBM:n ja BEA:n tuotteista sekä lyhyt kuvaus Oraclen tuotteesta ja avoimen lähdekoodin tarjonnasta.

2.2 Päivitetty IBM:n tarjonta

IBM:n WebSphere-sovelluspalvelimen nykyinen versio on 5.1. WebSphere-perheen alun perin erikseen toteutetut tuotteet ovat suuntauksen mukaisesti koottu yhdeksi kokonaisuudeksi, jossa on mukana myös Web Services -tuki ja joukko integraatio-ominaisuuksia. Sovelluspalvelimen lisäksi keskeisiä WebSphere-tuoteperheen osia ovat:

- WebSphere Application Studio, WSAD (kehitysympäristö)
- WebSphere Portal (yritysportaali)
- WebSphere Everyplace ja WebSphere Voice (puhekäyttöliittymien ja mobiiliulottuvuuden rakentaminen)
- WebSphere MQ ja Business Integration -tuotteet (integraatio)

Näiden lisäksi IBM:n WebSphere-tuoteperhe sisältää laajan kokonaisuuden eri tyyppisiä lisäkomponentteja.

Muutokset IBM:n tuoteperheessä

WebSphere-tuoteperheen pohjalla oleva sovelluspalvelin on kohdannut erittäin laajoja uudistuksia verrattuna aiempiin versioihin. Nämä uudistukset ovat tuoneet teknisellä tasolla parannuksia, joiden ansiosta IBM:n tuote on samanveroinen kilpailevien tuotteiden kanssa. Erityisesti aiemmin haasteelliset sovellusten käyttöönottoon ja hallintaan liittyvät piirteet ovat parantuneet. Samoin WSAD-kehitysympäristö on tuonut selkeitä tuottavuushyötyjä. WebSpheren keskeisiä vahvuuksia on sen käytettävyys hyvin laajassa ajoympäristökirjossa (mm. z/OS ja OS/390).

WebSphere Portal on kokoelma IBM:n eri tuotteita, jotka yhdessä tarjoavat yritysportaalin toiminnallisuuden. IBM:llä on tällä hetkellä kaksi eri kokoonpanoa, Portal Enable ja Portal Extend. Näiden kokoonpanojen keskeinen ero on siinä, että Extend sisältää myös IBM:n Lotus-tuoteperheen työryhmäsovellukset. Ominaisuuksiensa valossa se on ehkä markkinoiden laajin kokoonpano, jossa kaikki komponentit tulevat samalta valmistajalta. Tuotteen elinkaari on kuitenkin vasta alkuvaiheessa, sillä tuotteen piirteet vielä elävät ja myös vakaudessa on toivomisen varaa.

Integraatiotuotteiden osalta IBM etenee kahdella saralla. Perustana oleva MQ-sanomajonotuote on markkinoiden pitkäikäisimpiä ja laajimmin käytössä eri ympäristöissä. Se tarjoaa vankan pohjan integraatiolle. Sen sijaan Business Integration -tuotteet ovat vasta viime aikoina siirtyneet laajaan käyttöön. Käytännössä tuoteperhe on koottu joukosta toisiinsa liitettyjä erillistuotteita, joiden keskinäinen integraatio ei vielä ole täysin valmis.

2.3 Päivitetty BEA:n tarjonta

BEA:n WebLogic-sovelluspalvelimen nykyinen versio on 8.1. BEA on johdonmukaisesti rakentanut WebLogic-tuoteperhettä osa kerrallaan pyrkien säilyttämään

tuotekokonaisuuden yhtenäisyyden ja rakenteen selkeyden. Uusimman version myötä on päästy tilanteeseen, jossa perhe sisältää oleelliset APS-kokonaisuuden osat. Sovelluspalvelimen lisäksi BEA tarjoaa seuraavat elementit:

- WebLogic Workshop (uudentyyppinen tuottavuutta nostava kehitysväline)
- WebLogic Portal (yritysportaali)
- WebLogic Integration (integraatio)

Lisäkomponenttien osalta BEA tukeutuu yhteistyökumppaneihin. Esimerkiksi tekniseen ohjelmointityöhön BEA tarjoaa kehitysvälinevaihtoehdoksi Borlandin JBuilder-välineen WebLogicille räätälöityä versiota.

Muutokset BEA:n tuoteperheessä

BEA WebLogic Server on edelleen markkinoiden kypsän J2EE-sovelluspalvelin, jonka perusominaisuudet ja toiminnot ovat selkeitä ja luotettavia. Ylläpidon osalta toiminnot ja rajapinnat ovat monipuoliset. Esimerkiksi muut sovelluspalvelimet hyödyntävät käytännössä Apache-web-palvelinta, kun WebLogicissa tämä komponentti on valmiina – tästä syntyy etua järjestelmän konfiguroinnin ja ylläpidon suhteen.

WebLogic Workshop on integroinut kehitystyökalut uudeksi kokonaisuudeksi. Kehitysympäristössä pyritään helppoon toteutukseen eikä J2EE:n syvälinen tuntemus ole välttämättömyys. Uudessa kehitysympäristössä matalan tason koodi korvataan kehitysvaiheessa annotaatioilla, joista asennuksen yhteydessä generoidaan varsinainen ajettava sovellus. Kirjoitettavaa ja ylläpidettävää koodia on merkittävästi vähemmän. Korkean tason työkaluissa ja automaattisissa koodigeneraattoreissa piilee kuitenkin vaaransa, ja Workshopin lopullinen suosio jääkin vielä nähtäväksi. Kehittäjien on tunnettava työkalut. Lisäksi koodin manuaalinen ylläpito on erittäin työlästä ellei jopa mahdotonta. Yhteensopivuus muiden Java-ympäristöjen kanssa voidaan menettää ja myös työkalujen uusien versioiden yhteensopivuus voi tuoda haasteita.

WebLogic Portal tarjoaa yritysportaalin peruselementit, muttei laajenna tarjontaansa tämän ulkopuolelle. Lisäpiirteiden osalta (esim. julkaisuvälineet, yhteisöllisyyspiirteet) BEA nojaa kolmansien osapuolten tuotteisiin.

WebLogic Integration oli BEA:n tuoteperheen edellisessä versiossa merkittävä uutuus, jolla BEA laajensi sovelluspalvelimensa integraatiopalvelimeksi. Nykyinen versio (8.1) sisältää jo varsin kattavan kokonaisuuden tarvittavia integraatiotekniikoita, ja lisäksi tuotteen käyttö on merkittävästi helpottunut ensimmäisen vaiheen tekniseen lähestymistapaan verrattuna.

2.4 Oraclen sovelluspalvelintarjonta

Oracle Application Server on siirtymässä vartenotettavien sovelluspalvelinten joukkoon Oraclen viimeaikaisen vahvan panostuksen myötä. Alla oleva arvio perustuu versioon 9iAS, mutta vastikään on julkistettu uusi versio 10g. Oraclen alkuperäinen

sovelluspalvelinarkkitehtuuri perustui tietokantapalvelimen sisälle upotettuun Java-ympäristöön, mutta ratkaisu osoittautui käytännössä kömpelöksi ja epävakaaksi. Epäonnistuneen yrityksen jälkeen Oracle lisensoi ruotsalaiselta Orionilta muiden valmistajien kaltaisen sovelluspalvelinratkaisun ja lähti kehittämään sitä edelleen. Oracle on sittemmin lisännyt mm. uusien standardien tukea, parantanut suorituskykyä ja rakentanut laajat klusterointiominaisuudet mukaan sovelluspalvelimen ytimeen. Varsinaisen sovelluspalvelimen lisäksi keskeisiä elementtejä Oraclen sovelluspalvelintarjonnassa ovat:

- JDeveloper-kehitysympäristö
- Oracle Application Server Portal
- Oracle Application Server Wireless
- Oracle Application Server Integration

Näiden lisäksi Oracle tarjoaa joukon lisäratkaisuja erityisesti tuotantokriittisten järjestelmien toteuttamisen (mm. välimuistitoteutuksia, hallintavälineistöä, korkeaa käytettävyyttä, integroitua tietoturvaa).

Oraclen tuoteperheen lyhyt arvio

Oraclen sovelluspalvelimessa on nopean kehitystyön myötä ollut joitakin lastentauteja, mutta käytännön projektityössä niihin on saatu toimittajalta nopeat ja toimivat korjaukset. Toiminnallisuuden ja piirteiden osalta Oraclen tuote on tässä vaiheessa kilpailukykyinen markkinajohtajien tuotteiden kanssa.

Vahva panostus portaaliominaisuuksiin, langattomuuteen, integraatioon ja muihin lisäarvopiirteisiin on viime aikoina laajentanut Oraclen tarjontaa merkittävästi. Osa toiminnallisuudesta on saatu toteutettua tietokannan ja Oraclen edellisen sukupolven teknologioiden pohjalta, mutta pääosa kehitystyöstä on tehty uudelle J2EE-puolelle. Näiden teknologioiden käyttö on Suomessa vielä rajallista, mutta arvatenkin käyttäjiä tulee löytymään erityisesti Oraclen perinteisistä tietokanta-asiakkaista. APS-markkinoilla Oracle on kuitenkin jo kolmannella sijalla /6/.

Oraclen vahvuutena on sen mahdollisuus integroida sovelluspalvelin mahdollisimman tiiviisti tietokannan kanssa myös hallintavälineiden osalta. Lisäksi Oraclen tuotepaketointi on tehty houkuttelevaksi yhdistämällä laaja joukko työkaluja, joita tarvitaan tyypillisissä tuotantokriittisen järjestelmän toteutustyössä. Oraclen heikkoutena on kuitenkin selvä altavastaajan asema sovelluspalvelinmarkkinoilla.

2.5 Avoimen lähdekoodin J2EE-sovelluspalvelintarjonta

Myös avoimeen lähdekoodiin perustuvat sovelluspalvelinratkaisut ovat nousseet käyttökelpoisiksi vaihtoehdoiksi kaupallisille tuotteille. Käytännössä J2EE-pohjaisen tuotantoympäristöjä on toteutettu mm. seuraavilla komponenteilla:

- Apache/Tomcat-web-palvelin

- JBoss-EJB-toteutus
- Jokin relaatiokanta (MySQL, PostgreSQL tai kaupallinen tuote)

Tällä yhdistelmällä on saavutettu suorituskykyisiä ja toimivia ratkaisuja. On lisäksi huomattava, että kaupalliset toimijat sovelluspalvelinmarkkinoilla (mm. IBM, BEA, Oracle) ovat jo pitkään käyttäneet avoimeen lähdekoodiin perustuvia komponentteja osana omia tuotteitaan.

Avoimeen lähdekoodiin perustuvat ratkaisut ovat käytössä myös Suomessa. Arvion mukaan avoimen lähdekoodin sovelluspalvelimet ovat käytössä 10 %:ssa J2EE-projekteista vuoden 2003 lopussa, mutta käyttö rajoittuu lähinnä edelläkävijätyyppeihin IT-organisaatioihin /6/.

Arvio avoimen lähdekoodin käytöstä

Avoimeen lähdekoodin käyttöön sisältyy joukko erikseen ratkaistavia lisähaasteita:

- Lisensointimallit ja niiden soveltuvuus ratkaisun pohjaksi on selvitettävä jokaisen komponentin osalta erikseen.
- Tuen saatavuus on rajallista. Jotkin avoimeen lähdekoodiin perustuvat tuotteet on kuitenkin kaupallistettu joko osittain tai kokonaan (mm. JBoss, MySQL). Tällöin on tyypillisesti saatavilla myös tukea ja koulutusta.
- Kehitysvälineistö on rakennettava itse. Myös kehitystyö voi tapahtua avoimeen lähdekoodiin perustuvilla työkaluilla, mutta oikean paketin kokoaminen jää tekijöiden vastuulle.
- Erilaiset lisäarvopiirteet (mm. integraatio- ja portaalitoiminnallisuus) on myös koottava ja liitettävä kokonaisuuteen itse. Sopivien komponenttien löytäminen voi olla haastavaa, sillä tarjontaa on runsaasti.
- Kokonaisuuden rakentaminen ja komponenttien yhteensovittaminen vaatii laajaa perehtymistä ja työtä. Esimerkiksi EJB-toiminnallisuuden tarjoaa JBossin lisäksi myös OpenEJB ja valinta näiden välillä voi olla hankalaa.
- Ratkaisut eivät aina sovellu tuotantokriittisiin ympäristöihin. Pulmana on lähinnä puuttuva tuki klusteroinnille, korkealle käytettävyydelle ja hallinnointimekanismeille.

Tuotantoympäristössä avoimeen lähdekoodiin sisältyy usein ylipääsemättömiä haasteita, mutta kehitysvälineistössä ja kehitysympäristöissä avoin lähdekoodi on erittäin laajassa käytössä. Tyypillinen J2EE-hankkeissa käytetty kehitysvälineistö perustuu laajaan avoimen lähdekoodin käyttöön, ja tällä saralla onkin kehittynyt laaja joukko toimivia työkaluja, joista osa on vakiintunut lähes standardin asemaan (mm. Eclipse-kehitysympäristö, CVS-versionhallintavälineistö, JUnit-yksikkötestausväline).

2.6 Yhteenveto sovelluspalvelinmarkkinoista

IBM ja BEA hallitsivat 67 %:a J2EE-sovelluspalvelinmarkkinoista vuonna 2002. IBM on kasvattanut osuuttaan palvelinmarkkinoista viime vuosina ja BEA:n teknologiaetumatka on jonkin verran vähentynyt. Lisäksi Oracle on noussut varteenotettavaksi haastajaksi. Avoimen lähdekoodin sovelluspalvelimet ovat käytössä arviolta 10 % J2EE- projekteista vuoden 2003 lopussa.

J2EE-perusteknologia on vahvasti standardoitu ja kilpailuetua on vaikea sillä saralla enää löytää. Markkina onkin muuttumassa siten, että teknologiatoimittajat pyrkivät tarjoamaan eri tyyppisiä lisäarvopiirteitä sovelluspalvelimen päälle. Keskeisiä piirteitä ovat portaalitoiminnallisuus, integraatiopiirteet ja tuottavuutta nostavat kehitysvälineet. Yhteistä toimittajille on se, että niiden tarjoamat lisäarvopiirteet eivät enää ole yhteensopivia muiden toimittajien ratkaisujen kanssa.

Tämä trendi vaikuttaa myös sovelluskehitykseen, sillä toimittajakohtaisten graafisten kuvausmenetelmien ja sovelluspalvelinten erityispiirteiden käytön myötä katoaa J2EE-ratkaisujen siirrettävyys. Edessä onkin yhä useammin valinta tuottavuuden ja standardinmukaisen ratkaisun välillä. Edellisen haasteena on mahdollinen elinkaaririski ja jälkimmäisen haasteena taas ohjelmistotyön tuottavuus.

3. PORTAALITEKNOLOGIAN JA INTEGRAATION MAHDOLLISUUDET

3.1 Portaaliteknologia

Portaaliteknologia on viime aikoina kehittynyt erityisesti yritysportaalien osalta. Yritysportaalin tavoitteena on tarjota yksi kanava yrityksen oleellisiin tietoihin ja palveluihin sekä organisaation omille käyttäjille että organisaation ulkopuolisille tahoille. Yritysportaali tarjoaa toisaalta pääsyn tietoon, palveluihin ja toisaalta antaa erilaisia mekanismeja käyttäjäyhteisöjen keskinäisen vuorovaikutuksen. Yleiskäyttöisyydestä huolimatta portaalin käyttökohdetta usein rajataan kattamaan jokin tarkka tarkkan osa-alue organisaation toiminnassa. Tyypillisiä esimerkkejä ovat informaatioportaali ja HR-portaali.

Erillisten sovelluskohtaisten käyttöliittymien asemesta yritysportaalit tarjoavat integroidun, hallitun ja yhdenmukaisen näkymän tietoon ja palveluihin. Tuotteet tarjoavat myös keinoja sisällön ulkonäön ja kokemuksen mukauttamiseen käyttäjän omien valintojen ja toiminnan mukana. Lisäksi sisällön näkyvyys ja sen syvyys voi muuttua käyttäjän rooliin tai henkilökohtaisten kiinnostusten perusteella.

Tyypillinen yritysportaalituote sisältää seuraavia uuden vaalitietojärjestelmän kannalta hyödyllisiä elementtejä:

- Valmista toiminnallisuutta erityisesti staattisen sisällön esittämiseen
- Julkaisutoiminnallisuuden tai liittymän ulkoiseen julkaisujärjestelmään
- Sovelluskehysten ja apuvälineitä omien laajennusten rakentamiseen
- Valmiita yhteisöllisyyspiirteitä, työryhmäpiirteitä jne.
- Personointimekanismeja käyttäjäkohtaiseen toiminnan muokkaamiseen
- Integraatiovälineitä liittymien rakentamiseen ulkoisiin järjestelmiin
- Valmiskomponentteja joidenkin tunnettujen ohjelmistojen integrointiin
- Mekanismit single-sign-on-toiminnallisuuden toteuttamiseen
- Mekanismeja käyttäjän vahvaan tunnistamiseen
- Käyttäjien ja ympäristön hallinnointivälineet
- Piirteitä monikanavaratkaisun toteuttamiseen (esim. mobiilikäyttö)

Portaaliteknologian standardointi on vasta käynnistynyt. Tällä hetkellä mielenkiintoisimmat määritykset ovat Java-yhteisön määrittelemä Portlet API (JSR 168) ja OASIS-ryhmän määrittelemä Web Services for Remote Portals (WSRP). Edellinen määrittelee Java-kielisen rajapinnan portaalin näyttöelementin (ns. portlet) toteuttamiseksi ja jälkimmäinen taas kuvaa, kuinka portaalille julkaistaan sisältöä Web Services -standardiperheen mekanismien avulla.

Portaalimarkkinat ovat noin viiden vuoden ikäiset ja edelleen vahvassa muutostilassa. Tuotteet ovat vasta viime aikoina päässeet pahimmista lastentaudeista, eivätkä ne ole vielä päässeet laajaan käyttöön. Erityisesti interaktiivisen toiminnallisuuden vieminen yritysportaaleihin on osoittautunut haastavaksi. Tutkimusyhtiö Gartner arvioi maaliskuussa 2003, ettei yksikään yritys ole vielä saavuttanut riittävää kyvykkyyttä toimittaa laajamittaista yritysportaalikokonaisuutta (ns. smart enterprise suite). Suunta on kuitenkin selvä: kaikki keskeiset sovelluspalvelintoimittajat ovat viemässä portaalipiirteitä osaksi sovelluspalvelimen ydintoiminnallisuutta.

Arvio portaaliteknologian käytöstä

Teknologian nuoruudesta huolimatta uuden vaalitietojärjestelmän toteutuksessa on syytä harkita portaaliteknologioiden käyttöä. Erityisesti ulkoisille sidosryhmille tarkoitettun monikanavaisen viestinnän ja toiminnallisuuden toteuttamisessa portaaliteknologia tarjoaa laajan joukon valmista toiminnallisuutta, jonka toteuttaminen J2EE-perusteknologialla ei ole tarkoituksenmukaista. Sen sijaan taustajärjestelmien toteutukseen portaaliteknologia ei välttämättä tuo sellaista lisäarvoa, joka oikeuttaisi nykyisellään varsin monimutkaisen ratkaisun käytön.

Portaaliteknologian käyttöönotto on kuitenkin suunniteltava huolella, jotta nuoren teknologian tuomat riskit voidaan minimoida.

3.2 Integraatiotekniikat J2EE-ympäristössä

Integraatioon käytetyt teknologiat ja tuotteet ovat viimeisen parin vuoden aikana käyneet läpi merkittävän muutoksen. Lähes kaikki keskeiset integraatiotuotteiden toimittajat ovat jo siirtyneet tai ovat siirtymässä käyttämään kahta perusteknologiaa tuotteidensa uusien versioiden pohjana: J2EE ja XML. Poikkeuksen muodostaa Microsoft, joka käyttää näistä vain XML:ää. Keskeiset J2EE-sovelluspalvelintoimittajat ovat mukana tässä kehityksessä ja sovelluspalvelimet ovatkin laajenemassa integraatiopalvelimiksi.

Erityisen mielenkiintoisia integraatiopiirteitä J2EE-palvelimissa ovat seuraavat kokonaisuudet tai piirteet:

- XML-pohjaiset integraatiovälineet (jäsentäjät, muuntimet jne.). Nämä soveltuvat sekä synkroniseen että asynkroniseen integraatioon.
- Sanomajonotuen liittäminen sovelluspalvelimen ydintoiminnallisuuteen: Java Message Service (JMS) ja Message Driven Bean (MDB). Tämä soveltuu erityisesti sanomapohjaiseen integraatioon.
- J2EE Connector Architecture (JCA), jolla voidaan liittyä olemassa oleviin järjestelmiin. Tämä soveltuu erityisesti synkroniseen integraatioon.
- Web Services -perusteknologian tuki, ja erityisesti sen uudet orkestraatiopiirteet, joilla voidaan perustason Web Services -kommunikoinnin lisäksi ohjata toimintaa myös prosessitasolla.

- Integraatiobroker-toiminnallisuus, jonka avulla laajoja ja monipuolisia tietovirtoja voidaan hallita tehokkaasti sovelluspalvelimesta käsin.
- Liiketoimintaprosessien kuvausmekanismit, joiden avulla voidaan komponenteista koostaa liiketoimintaprosessin tekninen toteutus.

Osa yllä olevasta toiminnallisuudesta perustuu standardeihin tai standardin kaltaisiin määrityksiin (mm. JMS, MDB, JCA, Web Services) ja osa taas on pitkälle toimittajakohtaista toiminnallisuutta (integraatiobroker, liiketoimintaprosessit). Näihinkin on tosin tekeillä standardeja erityisesti Web Services -rintamalla, mutta standardien valmistuminen vie varmasti aikansa.

Arvio integraatiotekniikoiden käytöstä

Uudet J2EE-integraatiotekniikat ovat merkittävässä asemassa uuden vaalitietojärjestelmän toteutuksessa. Tähän on kaksi perustelua.

Ensinnäkin, laajojen J2EE-pohjaisten järjestelmien keskeisenä haasteena on usein liian kunnianhimoinen monoliittiseen kokonaisratkaisuun tähtäävä rakenne. Uuden vaalitietojärjestelmän osalta sama uhka on olemassa. Jotta järjestelmän rakentaminen ja jatkokehittäminen olisi hallittua, kokonaisuus on syytä pilkkoa hallittaviksi osakokonaisuuksiksi, joiden liittäminen edellyttää selkeää integraatiokerrosta ja sen määrittelevää integraatioarkkitehtuuria. Tämä on syytä perustaa vakiintuneisiin J2EE-integraatoratkaisuihin

Toiseksi, uuden vaalitietojärjestelmän toteutuksessa rakennetaan laaja joukko liittymiä sekä nykyiseen toimintaympäristöön että mahdollisiin uusiin sidosryhmiin, tiedotuskanaviin jne. Tämänkin ulkoisen integraatiotarpeen toteuttamiseen on hyvä käyttää J2EE-arkkitehtuurin mukaisia ratkaisuja. Edellä mainittu ratkaisu sisäinen integraatioarkkitehtuuri voidaan laajentaa koskemaan myös ulkoista integraatiota.

3.3 Web Services

Web Services -määritykset ovat viime aikoina nousseet keskeiseksi malliksi tulevien integraatoratkaisuiden suunnittelussa. Teknisessä mielessä kyseessä on joukko protokollatason määrityksiä, joiden avulla voidaan toteuttaa hyvinkin monipuolisia järjestelmien ja organisaatioiden välisiä kommunikointimalleja. Web Services -teknologioiden kasvava suosio perustuu kolmeen seikkaan:

- Se on ensimmäinen järjestelmien välisen kommunikoinnin malli, joka on kaikkien keskeisten markkinaosapuolten hyväksymä (mm. kaikki J2EE-sovelluspalvelintoteuttajat ja Microsoft).
- Se määrittelee rajapinnat toteutusteknologiariippumattomasta ja toimii siten neutraalina välikerroksena eri teknologioiden välillä.
- Se sisältää riittävät keinot laajojen, rakenteisten ja monipuolista vuorovaikutusta sisältävien integraatoratkaisujen tekoon.

Yhdessä nämä ovat johtaneet tilanteeseen, jossa Web Services -teknologioilta odotetaan merkittävää helpotusta integraatiohaasteisiin. Web Services -teknologian ydin sisältää seuraavat elementit:

- Simple Object Access Protocol (SOAP): XML-viesteihin perustuva sovellusten välinen kommunikointiprotokolla
- Web Services Description Language (WSDL): kuvausmekanismi osapuolten välisen kommunikoinnin yksityiskohtien määrittämiseksi
- Universal Description, Discovery and Integration (UDDI): hakemistopalvelu

Näiden lisäksi on valmisteilla laaja joukko määrittämiä mm. tietoturvan hallintaan, tapahtumankäsittelyn toteuttamiseen ja liiketoimintaprosessien kuvaamiseen. Keskenäisyyden vuoksi Web Services -standardeja ei ole vielä otettu laajaan julkiseen käyttöön, mutta organisaation sisällä ja myös suljetuissa organisaation välisissä ratkaisuihin Web Services on jo käyttökelpoinen malli.

Järjestelmän toteutuksen kannalta Web Services tarjoaa ennen kaikkea mahdollisuuden aidon palveluarkkitehtuurin toteuttamiseen. Tällöin järjestelmän toiminnallisuus jaetaan palveluihin, joita voidaan kutsua teknologiarajojen ylitse. Tyypillisesti palvelut kerrostetaan siten, että ylempien kerrosten käyttäjäläheiset palvelut koostetaan alempien kerrosten teknisistä palveluista. Hyvin toteutetun palveluarkkitehtuurin keskeisiä etuja on samojen palveluiden uudelleenkäyttämismahdollisuus.

Arvio Web Services –tekniikan käytöstä

Web Services -standardien käyttöä on syytä harkita uuden vaalitietojärjestelmän toteutuksessa, sillä standardit ovat jo tässä vaiheessa tulleet keskeiseksi osaksi J2EE-sovelluspalvelinten toiminnallisuutta. Web Services -määritykset ovat hyvää vauhtia laajenemassa kohti tehokasta ja laajasti käyttökelpoista kokonaisuutta, jonka avulla kaikki Internet-pohjainen sovellusten välinen kommunikointi voidaan toteuttaa teknologiavalinnoista riippumatta.

Uuden vaalitietojärjestelmän integraatoratkaisu ei kuitenkaan kokonaan voi perustua Web Services -standardeihin. Toiminnallisten kokonaisuuksien sisällä on syytä käyttää tehokkuussysteistä myös J2EE-riippuvia integraatiotekniikoita silloin, kun tiedetään ettei palvelun käyttö ylitä teknologiarajaa tai osakokonaisuuden rajaa. Vastaavasti eräpohjaiset liittymät on syytä tehdä XML-tiedostojen tai vastaavien mekanismien avulla.

4. LYHYT RISKIANALYYSI

Tässä kappaleessa käsitellään J2EE-hankkeisiin liittyviä riskejä erityisesti vaalitietojärjestelmän uudistamisen näkökulmasta. Riskit on jaoteltu seuraaviin ryhmiin:

- Teknologiariskit
- Tuoteriskit
- Ratkaisutason riskit
- Projektiriskit
- Toimittajariskit
- Nykyratkaisuun ja siirtymäpolkuun liittyvät riskit

Riskeille pyritään myös esittämään tilanteeseen sopivia hallintakeinoja.

4.1 Teknologiariskit

Riski 1. J2EE-tekniikan jatkuva muutos. Java-maailman keskeiset teknologiatoimittajat jatkavat edelleen J2EE-arkkitehtuurin kehitystä ja näiden lisäksi laaja Java-yhteisö kehittää avoimen koodin muodossa lisää toiminnallisuutta. J2EE-arkkitehtuuri ei täysin vakiinnu koskaan.

Erityisesti pitkän elinkaaren J2EE-sovelluksissa on syytä pitäytyä vain vakiintuneessa J2EE-arkkitehtuurin ytimessä, joka on pysynyt pitkälti samana jo muutaman vuoden. Haittapuolena on mahdollinen tuottavuuden lasku, mutta tällöin tuottavuuden nostamiseen voidaan käyttää muita keinoja kuten sovelluskehityksiä ja edistynyttä kehitysympäristöä.

Riski 2. J2EE on liian monimutkainen. J2EE-arkkitehtuuri kaikessa laajuudessaan antaa runsaasti erilaisia vaihtoehtoja sovellusten rakentamiseen. Valinta tekniikoiden välillä voi olla vaikeaa. Lisäksi tekijöiltä vaaditaan runsaasti osaamista ja ratkaisu saattaa muuttua vaikeaselkoiseksi sillisalaatiksi.

Kaikkia J2EE:n tarjoamia vaihtoehtoja ei ole syytä käyttää, vaan projekteissa on syytä rajoittaa valikoituun osajoukkoon. Vaalitietojärjestelmä sisältää siinä määrin erilaisia kokonaisuuksia, että tässä valinta on syytä tehdä osakokonaisuuksittain (esim. portaalivälineistö joissakin osissa, ja tietokantaintensiivinen käyttöliittymävälineistö toisaalla). Valinta on syytä dokumentoida riittävällä tasolla (*suunnittelumallit* ja *best practices*) ja lisäksi on syytä ottaa käyttöön osakokonaisuuksittain *sovelluskehitykset*, jotka tukevat parhaiten kunkin osakokonaisuuden tavoitteita.

Keskeisenä ratkaisuna monimutkaisuuden hallintaan on myös vahvan *arkkitehtuurin* määrittely. Arkkitehtuuri on syytä määrittellä ainakin kolmella tasolla.

- Järjestelmäarkkitehtuuri vastaa perinteistä varusohjelmisto- ja laitetason arkkitehtuuria. Integraatioarkkitehtuuri jakaa kokonaisuuden osasovelluksiin sekä määrittelee niiden väliset rajapinnat ja kuvaa myös integraation toteutustavan.
- Integraatioarkkitehtuuri on erityisen tarpeellinen laajojen perinneosia sisältävien kokonaisuuksien rakentamisessa. Uusi vaalitietojärjestelmä on tyyppiesimerkki tällaisesta järjestelmästä.
- Sovellusarkkitehtuuritasolla puolestaan kuvataan kunkin sovelluksen sisäinen ratkaisu sekä kaikille osille yhteiset sovelluskomponentit.

Riski 3. Puutteellinen skaalautuvuus ja suorituskyky. J2EE-sovellukset ovat usein osoittautuneet haastaviksi skaalautuvuuden ja suorituskyvyn suhteen.

Käytännön esimerkit osoittavat, että J2EE-arkkitehtuurilla voidaan saavuttaa riittävä suorituskyky laajoissakin kaupallishallinnollisissa järjestelmissä. Haasteena on kuitenkin perusarkkitehtuurin monimutkaisuus, joka usein toteuttajien osaamattomuuden kanssa johtaa suorituskyvyn kannalta huonoihin ratkaisuihin. Riittävä suorituskyky ja skaalautuvuus voidaan kuitenkin taata hyvällä työskentelymallilla, johon sisältyy mm. seuraavia keinoja:

- Sovellusarkkitehtuurin suorituskykyanalyysi jo projektin alkuvaiheessa
- Kriittisten osien suorituskyvyn varmistaminen erillisillä proof-of-concept-toteutuksilla
- Perinteisten Software Performance Engineering (SPE) -menettelyjen käyttö (transaktioanalyysi, ratkaisun suorituskykymallinnus tai simulointi ennen toteutustyön aloittamista jne.)

Uuden vaalitietojärjestelmän suorituskykyyn on syytä keskittyä jo vaatimusmäärittelyvaiheessa, mm. ei-toiminnalliset vaatimukset ja mahdollisten kuormien arviointi.

4.2 Tuotetason riskit

Riski 3. Tuotteiden kypsymättömyys. J2EE-teknologian kehityksen myötä tuotteiden päivityssykli on erittäin nopea ja tuotteet ovat tyyppillisesti hiukan puutteellisia markkinoille tullessaan. Uusien versioiden käyttö hidastaa projektityötä ja saattaa myös johtaa kestäättömiin virityksiin.

Kehitysvaiheessa on syytä soveltaa konservatiivista tuotepolitiikkaa. Välineistö kannattaa vakioda kunkin osaprojektin kohdalla mahdollisimman pitkälle ja lisäksi välineistöön ei kannata valita aivan uunituoreita komponentteja. Tuotantoympäristö on syytä pitää mahdollisimman samana kuin kehitysympäristö, jotta vältetään turhat testausyykliä. Kokemuksen perusteella parhaiten onnistuvat projektit, joissa käytetään toimittajan toiseksi viimeistä sovelluspalvelinversiota.

Riski 4. Valittu lisäarvotuote poistuu markkinoilta. J2EE-tuotteiden mukana tulevat lisäarvokomponentit ja sovelluskehikset saattavat poistua markkinoilta hyvinkin nopeasti.

Syynä on usein kilpailevan ja standardiksi muodostuvan ratkaisun syntyminen Java-yhteisön piirissä.

Pitkän elinkaaren järjestelmissä J2EE-standardin ulkopuolisia sovelluskehyskiä, lisäarvokomponentteja tms. ei ole syytä käyttää, mikäli niiden lähdekoodiin ei ole pääsyä tai ne eivät muuten ole saavuttaneet laajaa suosiota. Sen sijaan tuottavuuden nostoon kannattaa käyttää sovellustoimittajien omaan käyttöön kehittämiä sovelluskehyskiä, sovelluskohtaisia kehyskiä, tai avoimeen lähdekoodiin perustuvia ratkaisuja. Joissain tilanteissa toimittajakohtaiset piirteet voi peittää erityisen rajapinnan avulla, jolloin säilytetään mahdollisuus siirtyä uuteen toteutukseen, jos valittu toteutus poistuu markkinoilta.

4.3 Ratkaisutason riskit

Riski 5. Ratkaisun ylläpidettävyys. J2EE-sovellukset voivat olla vaikeita ylläpitää erityisesti silloin, kun käytetään arkkitehtuurin koko kirjoa ja tekijöillä on erilaisia taustoja.

Hyvä arkkitehtuurisuunnittelu ja toteutustyön ohjeistus ovat ylläpidettävyuden keskeisiä takeita. Lisäksi projektikäytäntöön on syytä ottaa mukaan katselmoinnit, jotta varmistetaan myös arkkitehtuurin ja ohjeiden noudattaminen. Sovelluskehyskiä käyttö tyypillisesti määräämuotoistaa tekemistä ja helpottaa ylläpitoa.

Riski 6. Ratkaisun hallittavuus. J2EE-sovellukset ovat tyypillisesti monimutkaisia ja niiden hallinta on usein hankalaa. Ajoympäristö saattaa sisältää laajan joukon valmisohjelmistoja (sovelluspalvelin, kantapalvelin, web-palvelin, hakemistopalvelin, turvaratkaisut, jne.), ja niiden saaminen stabiiliin tilaan on haastavaa.

Palvelinympäristön hallintaan on erityisesti paneuduttava arkkitehtuuria suunniteltaessa ja sovellushankkeen alkuvaiheessa myös sovellustasolla. Hallittavuutta voidaan parantaa mm. oikealla sovellusrakenteen suunnittelulla, turvaratkaisun hyvällä suunnittelulla, hakemistopalvelun käytön vakioinnilla jne. Jälkikäteen näiden korjaaminen on usein liian työlästä, sillä valinnoilla on vaikutusta myös sovelluskoodiin. Hallittavuutta voidaan myös parantaa sopivalla varusohjelmistovalinnalla.

4.4 J2EE-projektiriskit

Riski 7. Perinteiset menetelmät eivät toimi. Perinteisissä sovelluskehitysmenetelmissä ei ole tukea komponenttitekniikan erityisvaatimuksille.

Ratkaisuna on menetelmän laajentaminen tukemaan myös J2EE-tekniikan erityispiirteitä. Toimittajavalinnassa on syytä painottaa toimittajan menetelmän kypsyyttä erityisesti komponenttitekniikan osalta. Tyypillisiä J2EE-hankkeen vaatimia lisäpiirteitä menetelmään ovat mm.

- Arkkitehtuurin/sovelluskehyskiä valinta tai minikehyskiä toteutus ei-toiminnallisten vaatimusten pohjalta

- Uudelleenkäyttö osaksi menetelmää (mm. projektin alussa komponenttivaraston kartoitus, lopussa sadonkorjuun valmistelu)
- Komponentti-infrastruktuurin haltuunotto ja konfigurointi
- Komponentointivaihe: rajapinnat ja niiden analyysi
- Nopean kierron toteutus: uusien kehitysvälineiden tehokas hyödyntäminen, versionhallintavälineet
- Yksikkötestauksen vienti komponenttitasolle (JUnit)

Riski 8. Perinteinen projektimalli ei toimi. Laajojen J2EE-projektien läpivienti on osoittautunut erityisen haastavaksi. Osaavatkin projektipäälliköt ovat epäonnistuneet hankkeissa. Haasteena on mm. uuden Java-sukupolven erilainen työkulttuuri, teknologian nopea muuttuminen projektin aikana, liian suuret odotukset uuden teknologian mahdollisuuksista sekä liiketoimintatavoitteiden muuttuminen hankkeen aikana.

Ratkaisuna on projektimallin uudistaminen usealla tasolla:

- Iteratiivinen projektimalli, jossa välietappien yhteydessä voidaan muuttaa ja tarkentaa tavoitteita ja työsuunnitelmaa
- Laajan hankkeen jakaminen pieniksi itsenäisiksi osakokonaisuuksiksi
- Työn vastuuttaminen toiminnallisiin osakokonaisuuksiin, sillä perinteinen vastuuttaminen teknisiin osakokonaisuuksiin johtaa vaikeuksiin
- Formaalin muutoshallinnan käyttöönotto
- Testausmenettelyn tuominen osaksi koko projektin elinkaarta
- Uudenlainen projektimiehitys, jossa projektipäällikkö saa tukea sekä järjestelmäarkkitehdiltä että sovellusarkkitehdiltä.
- Katselmointien ottaminen osaksi projektityötä

Toimittajavalinnassa on syytä painottaa käytettävän projektimallin valmiutta erityisesti komponenttiprojektien läpiviennissä. Lisäksi projektiorganisaatioon on syytä erikseen nimetä joukko uusia rooleja. Tyypillisiä lisärooleja ovat (a) *sovellusarkkitehdin* nimeäminen perinteisen järjestelmäarkkitehdin rinnalle, (b) *integraatio työn* nostaminen omaksi osatehtäväkseen (mm. erillisen integraatioarkkitehtuurin määrittely), (c) *mentorien* käyttö projektiryhmän osaamistason nostamiseen.

Riski 9. J2EE-hankkeen ostaminen on haastavaa. Useat esimerkit osoittavat, että uusilla sovelluspalvelimilla toteutetut hankkeet voivat johtaa hallitsemattomaan kustannusten nousuun ja projektin pitkittymiseen.

Ratkaisuja löytyy usealta tasolta. Keskeisenä keinona on hankkeen kokonaisymmärryksen pitäminen omilla käsissä. Vaatimusten hallintaan, muutoshallintaan ja testaukseen on syytä panostaa jo hankkeen alkuvaiheesta lähtien, sillä näiden kautta ostava taho pystyy ohjaamaan hankkeen kulkua. Tyypillinen ostajan virhe on luottaa liikaa toimittajaan, joka

usein itsekin on ymmällään muuttuvassa J2EE-maailmassa. Määrämuotoinen ja riippumattoman tahon katselmointi on syytä ottaa käytännöksi jo hankkeen alusta pitäen.

Jos hankekokonaisuus pidetään omissa käsissä (esim. päätoiminen hankepäällikkö), niin keskeinen riskinhallinnan keino on hankkeen jakaminen usealle toimittajalle. Tällöin yksittäisen toimittajan ongelmat esim. teknologiahaasteiden osalta eivät muodostu uhaksi kokonaishankkeelle. Samoin voidaan seurata ja vertailla eri toimittajien tuottavuutta ja puuttua ajoissa poikkeamiin.

Osakokonaisuuksien välinen integraatio on syytä vastuuttaa erikseen ja pitää se omana osaprojektina. Sen sijaan eri osakokonaisuuksia ei ole syytä pakottaa samaan muottiin, kunhan ne toimivat määritellyn integraatoratkaisun puitteissa. Näin taataan kunkin osakokonaisuuden onnistuminen omien tavoitteidensa suhteen.

Projektin johtoryhmätyöskentelyyn on panostettava enemmän kuin perinteisissä projekteissa. Syynä on jatkuva J2EE-maailman muuttuminen sekä myös uuden vaalitietojärjestelmän tavoitetilan muuttuminen edessä olevan pitkän hankkeen aikana.

4.5 Toimittajiin liittyvät riskit

Riski 10. Sovellustoimittajan puutteelliset valmiudet. Ohjelmistotoimittajilla ei välttämättä ole käytettävissä riittävästi osaavia tekijöitä, projektipäälliköitä ja vaalitietojärjestelmän toteuttamiseen soveltuvaa menetelmää. Erityisesti J2EE-teknologiaa laajasti osaavien tekijöiden määrä on edelleen liian pieni tekeillä olevien hankkeiden määrään nähden.

Ratkaisuna on toisaalta useamman toimittajan käyttäminen ja toisaalta riittävän osaamistaseen varmistaminen kultakin toimittajalta. Tekijöiden osaaminen on syytä varmistaa jopa henkilötasolla. Uusi vaalitietojärjestelmä on laaja kokonaisuus, joten toimittajien voidaan olettaa myös kouluttavan uusia tekijöitä hanketta varten. Näin ollen toimittajien valmius ja menettelyt kohdennettuun *koulutukseen* ja *mentorointiin* on myös varmistettava.

Riski 11. Tuotetoimittajan halu myydä erikoisratkaisuja. Tuotetoimittajat pyrkivät sovelluspalvelinten ohella myymään monenlaisia lisäarvokomponentteja. Nämä voivat kuitenkin muodostua ongelmaksi ylläpidon, suorituskyvyn ja elinkaaren osalta.

Ratkaisuna on pitäytyminen J2EE-ydinteknologiassa uuden vaalitietojärjestelmän osalta. Lisäkomponentteja voidaan toki käyttää, mutta ne on syytä eristää rajapinnan taakse, jotta ne voidaan tarvittaessa korvata vaihtoehtoisilla ratkaisuilla. Erityisesti valmistajakohtaisen portaaliteknologian käyttö voi kuitenkin olla tarkoituksenmukaista ja sen käyttö onkin suunniteltava huolellisesti.

4.6 Nykyratkaisuun ja siirtymäpolkuun liittyvät riskit

Riski 12. Nykyratkaisun vaikea uudelleenkäytettävyys. Nykyinen järjestelmä tarjoaa laajan toiminnallisuuden, jota ei kerralla kannata korvata. Vähittäinen korvaaminen

edellyttää nykyisen ja uuden ratkaisun yhteiskäyttöä ja olemassa olevan toiminnallisuuden uudelleenkäyttöä välivaiheen ratkaisussa. Nykyistä ratkaisua ei kuitenkaan ole suunniteltu tällaista uudelleenkäyttöä silmälläpitäen.

Ratkaisuna on vahva panostaminen integraatioarkkitehtuurin hankkeen alusta pitäen. Alkuvaiheessa on syytä jäsentää kokonaisuus selkeiden rajapintojen avulla osiin ja kuvata rajapinnat tarkasti nykyjärjestelmän ja uuden ratkaisun välillä. Uudistushankkeen hallittu toteuttaminen saattaa edellyttää integraatioon liittyvää kehitystyötä myös nykyisen ratkaisun sisällä.

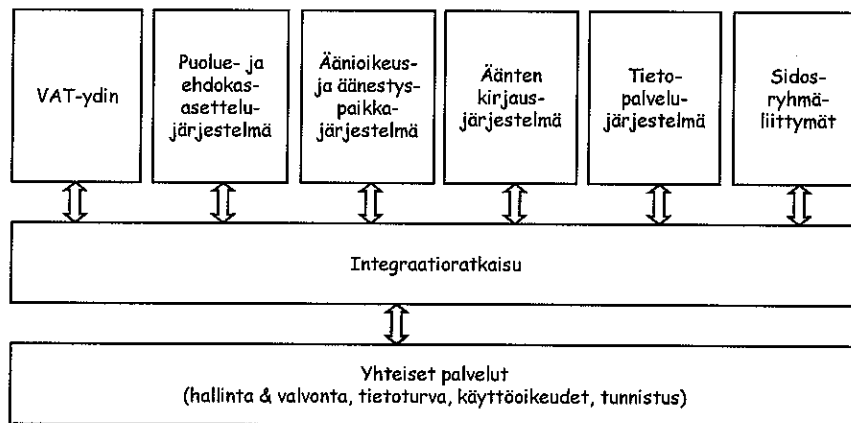
Riski 13. Tavoitteiden muuttuminen hankkeen aikana. Yhteiskunnan ja teknologian kehityssuuntia on vaikea ennakoida ja pitkälle ajalle ajoittuvan järjestelmä hankkeen aikana on luultavaa, että tavoitteena oleva toiminnallisuus ja ratkaisun rakenne muuttuu.

Ratkaisuna on järjestelmän jäsentäminen siten, että siinä varaudutaan lähtökohtaisesti muutoksiin. Avaimena on toisaalta palveluarkkitehtuurin soveltaminen (ks. Web Services -luku) ja toisaalta vahvan integraatoratkaisun käyttöönotto. Tällöin muuttuvien tavoitteiden seuraukset kokonaisratkaisuun ovat rajallisia.

5. UUDEN VAALITIE TOJÄRJESTELMÄN TOTEUTUS

5.1 Kokonaisarkkitehtuurin jäsenitys

Riskien hallinnan kannalta uuden vaalitietojärjestelmän kokonaisarkkitehtuuri on syytä jakaa useaan osajärjestelmään. Tätä puoltaa myös integraatiotekniikoiden viimeaikainen kehitys ja siirtyminen osaksi J2EE-ydinteknologiaa. Seuraavassa kuvassa on lyhyesti hahmoteltu mahdollista kokonaisarkkitehtuuria tältä pohjalta.



Kuva 1. Uuden vaalitietojärjestelmän kokonaisarkkitehtuuri.

Arkkitehtuurin kannalta keskeisessä roolissa on integraatoratkaisu, joka yhdistää osajärjestelmät toisiinsa ja yhteisiin palveluihin. Integraatoratkaisun avaintekniikoita voivat olla Web Services (synkroninen integraatio), sanomajonot (asynkroninen integraatio) ja XML-siirtotiedostot (eräpohjaiset siirrot). Myös muut tekniikat ovat mahdollisia, mikäli niiden käyttöön on tarvetta esimerkiksi sidosryhmien vaatimuksesta tai tietoturvan varmistamiseksi (mm. käyttäjien autentikointi). Lopulliset tekniikat ja niiden käyttötapa on syytä valita teknologioiden kypsyyden ja käyttötarpeen mukaan. On huomattava, että sama integraatoratkaisu voi toimia pohjana sekä kokonaisjärjestelmän sisäiselle integraatiolle että sidosryhmien kanssa toteutettavalle ulkoiselle integraatiolle.

Kaikille osajärjestelmille yhteisiä palveluita käytetään pääsääntöisesti integraatoratkaisun kautta. Tällä pyritään teknologiariippumattomuuteen ja samalla helpotetaan osajärjestelmien toteutushankkeita.

Yksittäiset osajärjestelmät kattavat tulevan vaalijärjestelmän koko toiminnallisuuden. Kuvan osajärjestelmäjako perustuu Jukka Leinon 2.12.2003 esittämään malliin, mutta kokonaisarkkitehtuuri ei muutu vaikka jako olisikin toteutusvaiheessa jokin muu. Osajärjestelmäjakoa valittaessa on oleellista, että kukin osajärjestelmä on toiminnaltaan ja

rakenteeltaan kohtuullisen homogeeninen. Näin saavutetaan tilanne, jossa osajärjestelmät ovat itsessään toimivia ja selkeitä kokonaisuuksia, vaikka kokonaisuudessa onkin luonteeltaan hyvin erilaisia toimintoja ja jopa ristiriitaisia vaatimuksia.

5.2 Uudelleenkäytön mahdollisuudet

Uuden vaalitietojärjestelmän toteutustyötä voidaan tehostaa uudelleenkäytön avulla. Erityisesti seuraavia uudelleenkäytön malleja on syytä harkita:

- Projektitoimittajan sovelluskehys
- Sovelluspalvelintoimittajan sovelluskehys
- Nykyisen vaalitietojärjestelmän hyödyntäminen
- Avoimeen lähdekoodiin perustuvat komponentit

Seuraavassa arvioidaan lyhyesti kutakin näistä.

Projektitoimittajan sovelluskehys. Kokeneilla J2EE-projektitoimittajilla on käytössään itse toteutettuja ja projekteissa toimiviksi työstettyjä sovelluskehyskehyksiä. Nämä tyypillisesti määrämuotoistavat J2EE-projektin toteutustavan ja antavat järjestelmästä riippuen vaihtelevan määrän valmiista ohjelmakoodia. Sovelluskehysten käyttökelpoisuutta arvioidessa on syytä selvittää sen alkuperäinen kohdealue. Jotkin kehykset on tarkoitettu erityisesti keskuskeskonesovellusten web-käyttöliittymien toteutukseen, kun taas toiset tarjoavat myös tietokantaintensiivisten sovellusten toteuttamiseen tarvittavia piirteitä. Eri osajärjestelmien toteutus voi perustua eri projektitoimittajien sovelluskehyskehyksiin, sillä sovelluskehysten ensisijaisena tarkoituksena on nostaa tuottavuutta, parantaa ylläpidettävyyttä ja vähentää riskejä. Tämä tapahtuu yleensä parhaiten projektitoimittajan omilla välineillä.

Sovelluspalvelintoimittajan sovelluskehys. Keskeisillä sovelluspalvelintoimittajilla on tyypillisesti useita sovelluskehyskehyksiä eri tarpeisiin. Näiden käyttöön liittyy tiettyjä riskejä, joista keskeisin on ylläpidon loppuminen ilman mahdollisuutta omaehtoiseen ylläpidon jatkamiseen. Tästä huolimatta näiden kehysten käyttö on harkitsemisen arvoista, sillä ne sisältävät laajan joukon toiminnallisuutta, jonka toteuttaminen ei yleensä ole tarkoituksenmukaista sovellusprojekteissa. Erityisesti sovelluspalvelintoimittajan portaalitoteutukseen tarkoitettujen kehyksien käyttö voi osoittautua hyödylliseksi tietopalveluratkaisun toteuttamisessa.

Nykyisen vaalitietojärjestelmän hyödyntäminen. Olemassa olevan vaalitietojärjestelmän hyödyntäminen siirtymävaiheessa on välttämätöntä, jos uusi vaalitietojärjestelmä halutaan ottaa käyttöön vaiheittain. Tällainen migraatiohanke tapahtuu tyypillisesti useassa vaiheessa. Aluksi tunnistetaan ja määritellään olemassa olevan järjestelmän tarjoamat rajapinnat. Sen jälkeen toteutetaan nämä rajapinnat olemassa olevaan järjestelmään. Lopuksi rajapinnan takana oleva toiminnallisuus voidaan vaiheittain toteuttaa uudella tekniikalla. Avainasemassa tällaisessa migraatiosuunnitelmassa on selkeä integraatioarkkitehtuuri ja sen johdonmukainen käyttö uutta ratkaisua toteutettaessa.

Avoimeen lähdekoodiin perustuvat komponentit. Avoimen lähdekoodin käyttö voi tarjota merkittäviä hyötyjä verrattuna kokonaan projektissa toteutettuihin ratkaisuihin. Tällöin on tietysti varmistettava, että lisensointiehtot ovat hankkeen kannalta tarkoituksenmukaisia ja että koodi on asetettujen laatuvaatimusten mukaista. Useat avoimeen lähdekoodiin perustuvat ratkaisut ovat kuitenkin saavuttaneet laajan suosion ja ovat siirtyneet jopa osaksi kaupallisia tuotteita. Esimerkkinä tällaisesta ratkaisusta on HTML-käyttöliittymän toteutukseen tarkoitettu Struts-tekniikka, jota käytetään mm. IBM:n ja BEA:n sovelluspalvelinympäristössä.

5.3 Etenemismalli

Kun yllä kuvattu ositettu arkkitehtuurimalli otetaan lähtökohdaksi, uuden vaalitietojärjestelmän tekninen toteutustyö on tarkoituksenmukaista tehdä vaiheittain siten, että lopputulos lähestyy askel askeleelta tavoiteratkaisua. Toteutustyö voidaan jakaa esimerkiksi seuraaviin karkeisiin osatehtäviin:

1. **Esimäärittely** sisältää tavoitteena olevan toiminnallisuuden karkean kuvauksen sekä käytön kannalta tavoitteellisen aikataulutuksen (valmistui 6.11.2003).
2. **Osakokonaisuuksien tunnistaminen ja alustava toiminnallisuuden jako.** Tämän vaiheen tarkoituksena on jäsentää kokonaisarkkitehtuuri osiin siten, että tavoitteena oleva toiminnallisuus saavutetaan ja hankkeen riskit voidaan minimoida.
3. **Integraatoratkaisun määrittely.** Tämä vaihe tuottaa arkkitehtuuritason määrittelyksen uuden vaalitietojärjestelmän integraatoratkaisulle. Ratkaisussa valitaan käytettävät integraatiotekniikat ja -tuotteet sekä määritellään niiden käyttötapa.
4. **Integraatoratkaisun toteutus ja osakokonaisuuksien toteutuksen tuki.** Tämä vaihe käynnistyy ennen osakokonaisuuksien toteutusprojekteja (toteutusosuus) ja jatkuu niiden aikana (tukiosuus). Työn tekee integraatiotiimi, joka hoitaa osaprojektien välissä olevan ns. harmaan alueen. Tätä varten on syytä määritellä myös osaprojektien noudattamat menettelyt (mm. rajapintojen versiointi, muutoksenhallinta, testausmenettelyt)
5. **Rajapintojen sovellustason määrittely.** Tämän vaiheen tarkoituksena on määritellä sovellustason rajapinnat eri osajärjestelmien välille integraatoratkaisun määrittelemillä mekanismeilla. Työ etenee tyypillisesti rinnakkain osakokonaisuuksien toiminnallisen määrittelyn kanssa (kohta 6).
6. **Osakokonaisuuksien toiminnallinen määrittely.** Tässä vaiheessa määritellään uuden vaalitietojärjestelmän tarkka toiminnallisuus osakokonaisuuksittain. Työn aikana voidaan myös tarkentaa rajapintojen määrittelyä (kohta 5).
7. **Nykyisen VAT-järjestelmän liittäminen integraatoratkaisuun.** Ensin tunnistetaan ja määritellään olemassa olevan järjestelmän tarjoamat rajapinnat osana vaihetta 5. Sen jälkeen toteutetaan nämä rajapinnat nykyiseen järjestelmään.
8. **Osakokonaisuuksien toteuttaminen vaiheittain.** Työ on syytä aloittaa uudesta toiminnallisuudesta, jolloin olemassa oleva VAT-järjestelmä voi edelleen toimia ratkaisun ytimenä.

9. **Olemassa olevan VAT-ytimen korvaava toteutus.** Tämä osuus on syytä toteuttaa viimeisenä, sillä se on edelleen toimiva kokonaisuus.

Etenemismallin tavoitteena on tuottaa itsenäisiin osajärjestelmiin perustuva kokonaisuus, joka viimeisen vaiheen jälkeen kattaa kaikki uuden vaalitietojärjestelmän toiminnalliset tavoitteet. Integraatioon perustuvan ratkaisun vahvuutena on sen joustavuus muuttuvien vaatimusten suhteen sekä toteutustyöhön sisältyvän riskin pienentyminen verrattuna monoliittiseen kokonaisratkaisuun.