

Arkkitehtuurivalintaan liittyviä ratkaisuja WWW:ssä

Simo Vuorinen,
TietoEnator Oyj

Arkkitehtuurin (software architecture) suunnittelu on kriittinen osa ohjelmistotuotantoprosessia. Arkkitehtuuria aletaan rakentaa jo järjestelmän määrittelyvaiheessa. Arkkitehtuuri voidaan jakaa usealla eri tavalla osaluokkiin, eikä yleisesti hyväksyttyä prosessimallia arkkitehtuurin rakentamiseen ole.

Tässä artikkelissa pohditaan WWW-sovellusten rakennetta yleisesti pitäen mielessä arkkitehtuurin asettama vaatimuskehys.

Mitä arkkitehtuuri on?

Arkkitehtuuri-sanaa käytetään nykyään melko paljon erilaisissa merkityksissä. Tietotekniikkayritykset rekrytoivat ”arkkitehteja” erilaisiin tarkoituksiin, kuten järjestelmäarkkitehteiksi, teknisiksi arkkitehteiksi, sovellusarkkitehteiksi, tietoturva-arkkitehteiksi – kaikki tehtävät, joissa suunnitellaan tietojärjestelmiä kokonaisuuksina tai joissa suunnitellaan jotain tietojärjestelmän osaa, on ”arkkitehtityötä”.

Arkkitehtuuri-sana tulee latinan sanasta *architectura*, joka tarkoittaa rakennustaitea. Arkkitehti on taas ”päärakentaja, joka on opillisesti arkkitehtuuriin perehtynyt henkilö, joka laatii rakennus- ja asemakaavasuunnitelmia ja valvoo niiden toteuttamista” (Uusi Sivistyssanakirja). Tätä analogiaa ei kuitenkaan kannata viedä liian pitkälle; rakennusarkkitehdin palkkaa usein asiakas suunnittelemaan rakennuksen ja valvomaan rakentamista, kun taas järjestelmäarkkitehti on yleensä sovellusta rakentavan tietotekniikka-yrityksen palveluksessa (Sommerville s.226).

Arkkitehtuurin suunnitteluun ei ole mitään yleisesti hyväksyttyä prosessimallia. Arkkitehtuurin taso riippuu järjestelmäarkkitehdin sovellustuntemuksesta, taidoista ja sisäisestä intuitiosta (Sommerville s.226). Tietotekniikassa arkkitehtuuri voidaan jakaa esimerkiksi

- loogiseen arkkitehtuuriin ja
- tekniseen arkkitehtuuriin.

Looginen arkkitehtuuri sisältää tietojärjestelmäarkkitehtuurin, hajautusratkaisut, tietovarannot ja sovellusarkkitehtuurin kuvaukset. Teknisen arkkitehtuurin kuvausiin kuuluvat tietoliikennearkkitehtuuri, käyttöjärjestelmät ja varusohjelmistot, laitteistoarkkitehtuuri sekä ohjelmistokehitysympäristö. Arkkitehtuuripohjainen kehitys yleensä sisältää kuitenkin sisältää seuraavat vaiheet:

- Arkkitehtuurivaatimusten määrittäminen
- Arkkitehtuurin suunnittelu
- Arkkitehtuurin dokumentointi
- Arkkitehtuurin analysointi
- Arkkitehtuurin toteutus (realization)
- Arkkitehtuurin ylläpito

Alla kuva arkkitehtuuripohjaisesta suunnitteluprosessista.

Tässä artikkelissa kuitenkin arkkitehtuurilla tarkoitetaan järjestelmän tai ohjelman komponenttien rakennetta, niiden välisiä suhteita, sekä periaatteita ja ohjeita, jotka ohjaavat niiden suunnittelua. Arkkitehti on henkilö, joka laatii tällaisen järjestelmän rakennussuunnitelman ja valvoo sen toteuttamista.

Mihin arkkitehtuuri ottaa kantaa?

Arkkitehtuurikuvausten on tarkoitus kuvata, mistä osista tietojärjestelmä tai –järjestelmäkokonaisuus koostuu, miten se on ra-

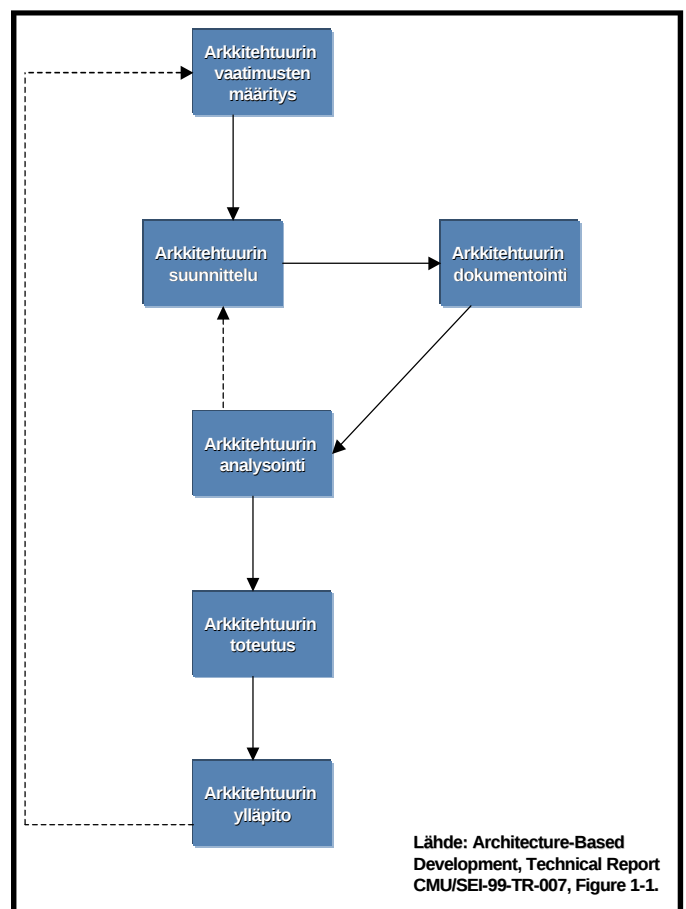
kennettu tai tullaan rakentamaan, minkälaisia tekniikoita ja teknologioita ratkaisuja käytetään, sekä järjestelmän eri osien väliset yhteydet ja suhteet.

Arkkitehtuuria suunniteltaessa käytetään apuna järjestelmän vaatimusmäärittelyä, jolloin on tiedossa perusvaatimukset ja mahdolliset rajoitukset.

WWW-arkkitehtuuri – asiakas/palvelin-arkkitehtuurin erikoistapaus

WWW-arkkitehtuurimalli on asiakas/palvelin-mallin erikoistapaus, jossa asiakas-sovellus on normaalisti WWW-selain, joka käyttää palvelimen tarjoamia palveluita HTTP-protokollan kautta.

WWW:ssä on kolme toisistaan erottuvaa virtuaalista verkkoratkaisua,



Lähde: Architecture-Based Development, Technical Report CMU/SEI-99-TR-007, Figure 1-1.

Kuva 1: Arkkitehtuuripohjainen suunnitteluprosessi. Läh-

intranet-, extranet- ja internet-ratkaisut. Näillä kolmella mallilla on tietynlaisia peruspiirteitä, joita läpikäymällä voi valita soveltuvan tavan rakentaa WWW-sovelluksen.

Samoin asiakassovelluksen käyttöliittymälle voidaan nähdä ainakin kolme perustyyppiä, joilla on tietyt ominaisuudet, ja joiden valintaan vaikuttaa usein yllämainitun verkkoratkaisun tyyppi.

Intranet-ratkaisut

Intranet-ratkaisut ovat ”huolettomimpia” näistä kolmesta virtuaalisesta verkkomallista. Intranet-sovellusten etuna on se, että ympäristö on yleensä varsin homogeeninen. Rakentajalla on tiedossa ympäristö, jonne sovelluksia rakennetaan. Yleensä tiedetään käytettävä selain ja sen versio, ja joissain tapauksissa voidaan rajoittaa sovellus toimimaan ko. selaimella ja sen tietyllä minimiversiotasolla.

Intranet-ratkaisuissa voidaan haluttaessa käyttää muitakin protokollia kuin HTTP:tä. Samoin pystytään yleensä myös kiertämään palomuuereista mahdollisesti aiheutuvat muut ongelmat, eikä salaamattoman tiedon kulku verkossa ole samalla tavalla ongelma kuin esimerkiksi avoimissa internet-ratkaisuissa.

Extranet-ratkaisut

Extranet-ratkaisut alkavat olla melko yleisiä esimerkiksi yritysten välillä, joiden on päästävä käyttämään joitain yhteisiä resursseja.

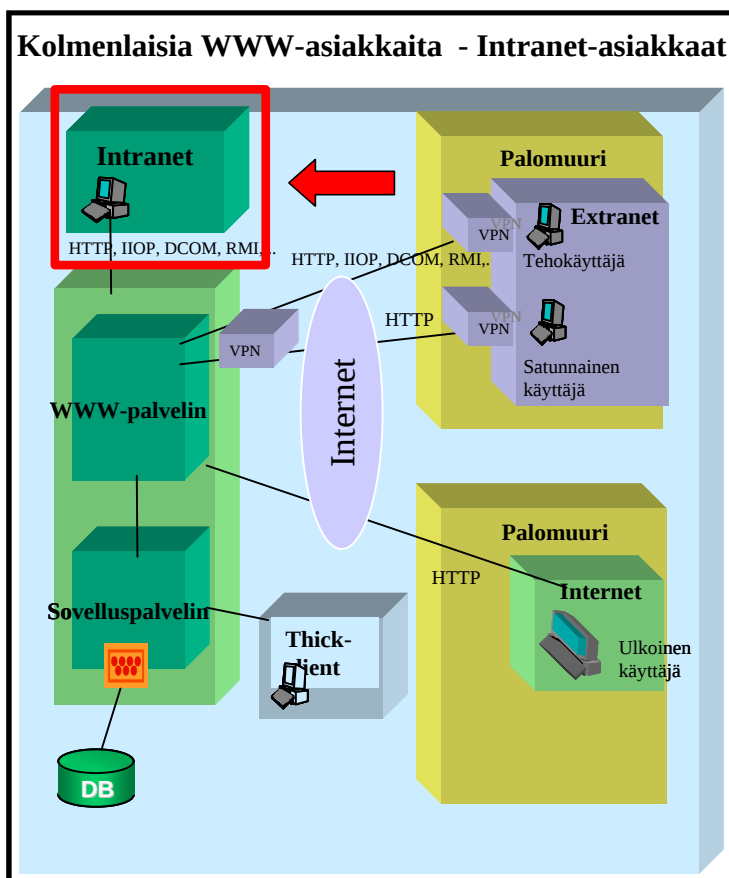
Extranet on ympäristönä heterogeenisempi kuin intranet, ja yritysten välillä voi olla eroja tietoteknisessä ympäristössä. Periaatteessa edelleenkin on mahdollista, että tunnetaan käyttäjien laitteistoympäristö ja selain versioineen, mutta selainstandardit voivat olla yrityksissä erilaisia, ja lisäksi saattaa olla versiokohtaisia eroavaisuuksia.

Tietoturva on otettava huomioon toisessa mittakaavassa kuin intranetissa ja yllätyksiä saattaa tulla myös palomuurien ja proxy-palvelinten kanssa.

Internet-ratkaisut

Internet-ratkaisussa ei välttämättä tiedetä käyttäjän selaimesta tyyppiä (Microsoft IE, Netscape Navigator, jne.) eikä versiota. Kuljetaan avoimessa verkossa, jossa tietoturva on huomioitava hyvin. Käyttäjän sisäänkirjaustietoja ei ole hyvä kuljettaa suojaamattoman yhteyden kautta, sillä data kulkee tällöin verkossa selväkielisenä.

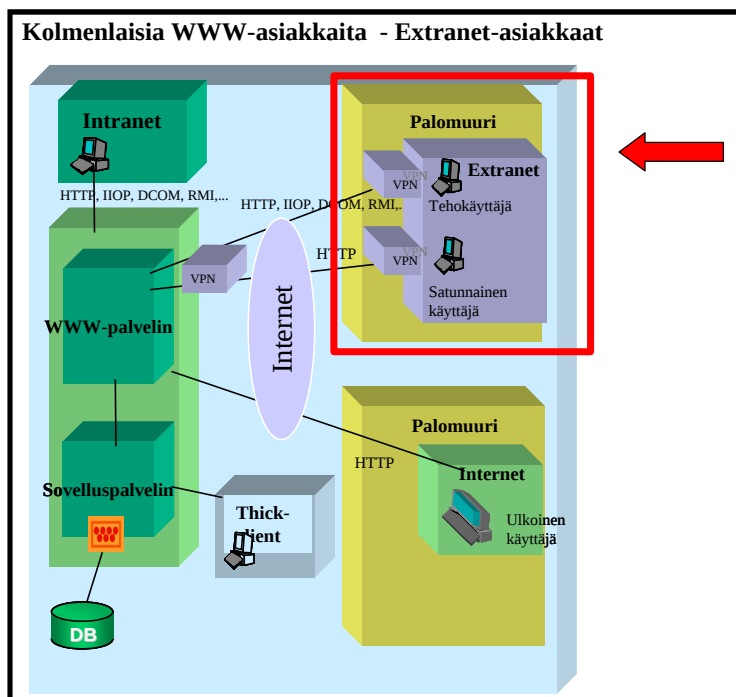
Yleensä joudutaan rajaamaan ulos muut protokollat kuin HTTP, sillä palomuurit on useimmiten konfiguroitu siten, että niiden läpi kulkee vain HTTP-liikenne, ja siitäkin tarkastetaan haluttaessa vielä, ettei mukana ole Java-sovelmia, ActiveX-komponentteja tai muuta aktiivista sisältöä.



Kuva 2: Intranet-ratkaisut

Selaimessa käytettävät asiakassovellustyypit

Asiakassovellustyyppinä, joita WWW-selaimessa käytetään, on niin-
ikään kolme perustyyppiä. Nämä ovat puhdasta HTML:ää sisältävä thin
client, HTML-sivun lisäksi skriptikieltä sisältävä ”smart client”, ja kol-
mantena vaihtoehtona vielä WWW-selaimessa käytettävä thick client, joka
voi olla esimerkiksi Java-sovelma tai ActiveX-komponentti.



Kuva 3: Extranet-ratkaisut

Thin client – pure HTML

Thin client-sovelluksen etuina on, että se yleensä toimii varmin, koska se ei sisällä muuta kuin HTML:ää. Tämän vuoksi se on myös varsin nopea latautumaan, ja palomuurit eivät sitä hyljeksi.

Toisaalta sen ongelmina on se, että koska se sisältää vain HTML:ää, se ei sisällä älyä; kaikki tarkistukset tehdään vasta palvelimella. Tämä tarkoittaa sitä, että jos HTML-sivulla on lomake, jonka käyttäjä täyttää, ja hän on syöttänyt lomakkeeseen virheellistä tietoa tai jokin pakollinen tieto puuttuu kokonaan, tämän tarkistus tapahtuu vasta, kun käyttäjä on lähettänyt lomakkeen submit-painikkeella vastaanottavalle palvelimelle. Tämä saattaa turhauttaa käyttäjää, ja toimenpiteen läpivienti voi olla melkoisen hidasta.

On tietenkin jo suunnitteluvaiheessa mietittävä, mikä on sovelluksen käyttäjien kohderyhmä, jotta voidaan päättää myös käytettävä HTML-versio. On hyvä muistaa, että kaikilla ei ole viimeisimpiä selainversioita käytössään.

Tämän tyyppinen ratkaisu tulee kyseeseen tiedon haku-, jake- lu- ja tallennussovelluksissa, joissa ei tarvita toiminnallisuutta vaan tieto on saatava selaimen nopeasti ja varmasti.

”Smart client” –HTML & skriptikieli

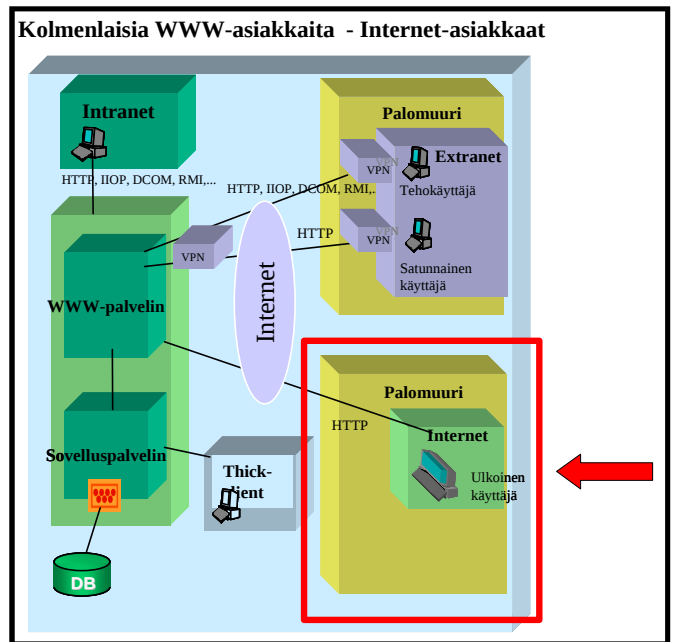
”Smart client” saattaa olla hiukan hitaampi latautumaan kuin thin client, mutta ero on todennäköisesti marginaalinen. Sen sijaan ”smart client”-ratkaisu on huomattavasti käyttäjä-ystävällisempi kuin puhdas HTML-sivu. ”Smart client”-vaihtoehdossa osa lomakkeen tarkistuksista voidaan tehdä jo lomakkeen vielä ollessa käyttäjän selaimessa. Tällaisia tarkistuksia on esim. pakollisten tietojen puuttuminen, laskutoimitukset, numeerisuustarkistukset, jne. Selainten oliomallit kuitenkin eroavat toisistaan, joten kannattaa tarkistaa, mikä on skriptikielen versiosta tuettuna halutuissa selaimissa, ja rajoitua niihin toimintoihin, jotka toimivat halutuissa selaimissa. Skriptikielen käytössä on otettava huomioon sekä selaintyyppi että selaimen versio.

Lisäksi kannattaa muistaa, että JavaScript ei ole sama asia kuin Java. JavaScript on kohtalaisen kevyt, HTML-sivujen ja sen elementtien ohjailuun tarkoitettu skriptikieli, kun taas Java on täysimittainen ohjelmointikieli.

Smart client-tyyppinen asiakassovellus on käyttökelpoinen sekä intranet- että extranet-tyyppisissä ratkaisuissa. Tämän tyyppisiä sovelluksia on kyllä internet-käytössäkin, mutta niissä esiintyy jonkin verran ongelmia.

Thick client –järeä sovellus selaimessa

Selaimessa on mahdollista myös suorittaa varsin järeitäkin sovelluksia, jotka mahdollistavat käyttäjälle samat piirteet kuin mikä tahansa työasemasovellus.



Kuva 4: Internet-ratkaisut

Java-sovelmat (applets) voivat olla täysimittaisia sovelluksia kunnollisine graafisine käyttöliittymineen, taulukkokomponentteineen ja puu-elementteineen. Myös Microsoftin ActiveX-komponenteilla saadaan aikaan vaativatkin käyttöliittymät selaimen.

Jos käytetään tämän tyyppistä täydellistä sovellusta selaimen kautta, voidaan sovelluslogiikka jakaa halutulla tavalla selaimen ja palvelimen välillä.

Näissä kummassakin on kuitenkin omat ongelmansa. Aktiivista sisältöä tuottavan selaimen ladattavan ohjelman ollessa kyseessä joudutaan kiinnittämään erityistä huomiota selainkäyttäjän turvallisuuteen. Tällaiset sovellukset saattavat myös olla melkoisen suuria, joten kun sovellus käynnistetään selaimen kautta ensimmäistä kertaa, sovelluksen latautuminen palvelimelta saattaa kestää jonkin aikaa.

Thin Client

TietoEnator

NTVPTest

- Search Ntvptest
- SupportGroup-sites
- TechSupport
- Distribution Lists
- TechSupport Persons
- Bulletinboard
- TechSupport Newsgroups
- VP Books
- Download
- FAQ
- Back to NTVPTest Home Page
- Links
- News & Announcements
- Archive
- Upcoming Events
- Add a bulletin board, news, or an event
- MentorTeam

✓Tapahtuma: LINUX/XML
Järjestäjä: Java- ja Internet-tukiryhmä
Aika: 12.10 klo 9.00-11.00
Paikka: Otae1007
Ilmoittautuminen: simo.vuorinen@tietoanator.com
Kuvaus: *Risto Raitio: Linux
*Janne Kallunki: XML

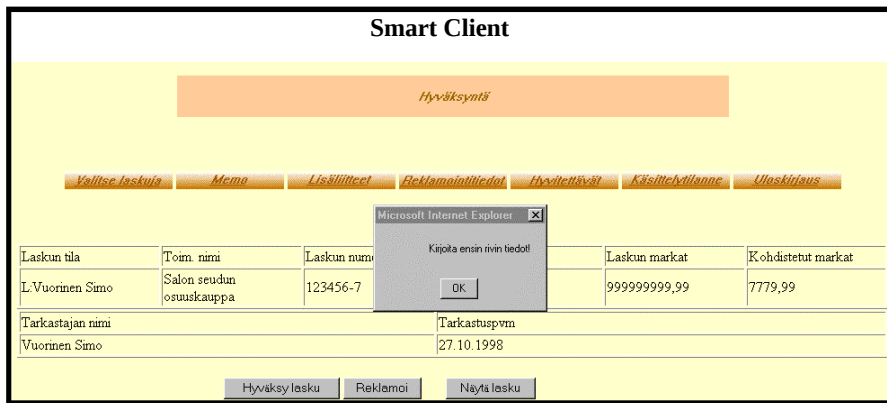
✓Tapahtuma: VP:n YLEISKÄYTTÖISET JAVA-KIRJASTOT
Järjestäjä: Java- ja Internet-tukiryhmä
Aika: 21.9 klo 9.00-11.00
Paikka: OTAE 1007
Ilmoittautuminen: simo.vuorinen@tietoanator.com
Kuvaus: Java- ja Internet-tukiryhmä proudly presents:

VP:n Yleiskäyttöiset Java-kirjastot

*Risto Pohjasmäki: Itutit-pakkaus - lokikirjoitus, virhekasittely, oliopuulaus, istunnonhallinta, ym.

*Kan Marttila: servlet-pakkaus - edelliset implementoitui servlet-kehikseen ja muita käyttökelpoisia piirteitä

Kuva 5: Thin client



Kuva 6: Smart client

Thick client-tyyppisiä ratkaisuja voi käyttää intranet-sovelluksissa helpommin kuin extranet- ja Internet-ratkaisuissa, ja nämä soveltuvat käyttöliittymiltään vaativampiin sovelluksiin.

Java-sovelmat (applets)

Java-sovelman kanssa joudutaan ottamaan huomioon tarvittava Java-virtuaalikoneen versio. Vaikka selaimissa vielä toistaiseksi on selainvalmistajan puolesta virtuaalikone, niiden versiot kulkevat selaimen tyyppin ja version mukana. Käytännössä, jotta voidaan varmistua sovelman toimivuudesta, olisi suositeltavaa käyttää Sunin Java Plug-inia. Näin voitaisiin varmistua siitä, että sovelmaa suoritetaan selaisella virtuaalikoneella joka toimii varmasti riippumatta siitä, mikä selaintyyppi tai selainversio on kyseessä. Tällä hetkellä Java Plug-in on saatavilla Microsoftin ja Netscapen selaimiin.

Java Plug-in kykenee haluttaessa varastoimaan sovelman plug-inin välimuistiin, joten sovelmaa ei tarvitse enää ladata uudestaan joka kerran, kun sivulla käydään. Tämä onkin ollut melkoinen koetinkivi sovelmien käytölle.

Java-sovelman oikeuksia päästä käsiksi käyttäjän resursseihin voidaan rajoittaa vielä Javan hienojakoisella turvallisuusmallilla.

ActiveX-komponentit

ActiveX-komponentti voi olla täysimittainen Windows-sovellus. Microsoftin ActiveX-komponenttien turvallisuusmalli perustuu luottamukseen. Tämä tarkoittaa sitä, että käyttäjän on hyväksyttävä ko. komponentin latautuminen koneeseensa, jotta voi hyödyntää komponentin ominaisuuksia.

ActiveX:n käyttöä on rajoittanut se, että ko. komponenttimalli toimii ainoastaan Microsoftin selaimissa. Netscapen

selaimiin on saatavissa joitakin plug-ineja, jotka mahdollistavat ainakin joidenkin ActiveX-komponenttien käytön myös Netscapesta käsin.

Kolmitaso- tai monitasoiset arkkitehtuurit

Arkkitehtuurissa otetaan usein kantaa myös suorituskyykyyn, skaalautuvuuteen, luotettavuuteen ja turvallisuuteen.

Kolmitasoarkkitehtuuriratkaisussa asiakassovelluksen ja tietovaraston väliin asetetaan palvelukerros, joka toteuttaa osan liiketoimintalogiikasta, tai sen kokonaan. Lisäksi välikerros toteuttaa yllämainittuja piirteitä sovelluskehittäjälle ”ilmaiseksi”, joten sovelluskehittäjä voi keskittyä oleelliseen. Alla kuva thin client-vaihtoehdosta kolmi- tai monitasoarkkitehtuurissa.

Toinen vaihtoehto on rakentaa itse dynaaminen sivupalvelin, joka on toteutettu esim. Java Servlet- tai MS:n Active Server Pages-teknologialla. Näillä saadaan itsessäänkin jo kevyt suoritussympäristö WWW-sovellukselle. Näissä teknologioissa on itsessään ominaisuuksina palvelupyyntöjen säikeistys, kevyt istunnonhallinta ja muita piirteitä, jotka helpottavat sovelluskehitystä. Alla kuva thin client-vaihtoehdosta Java Servlet-ympäristössä.

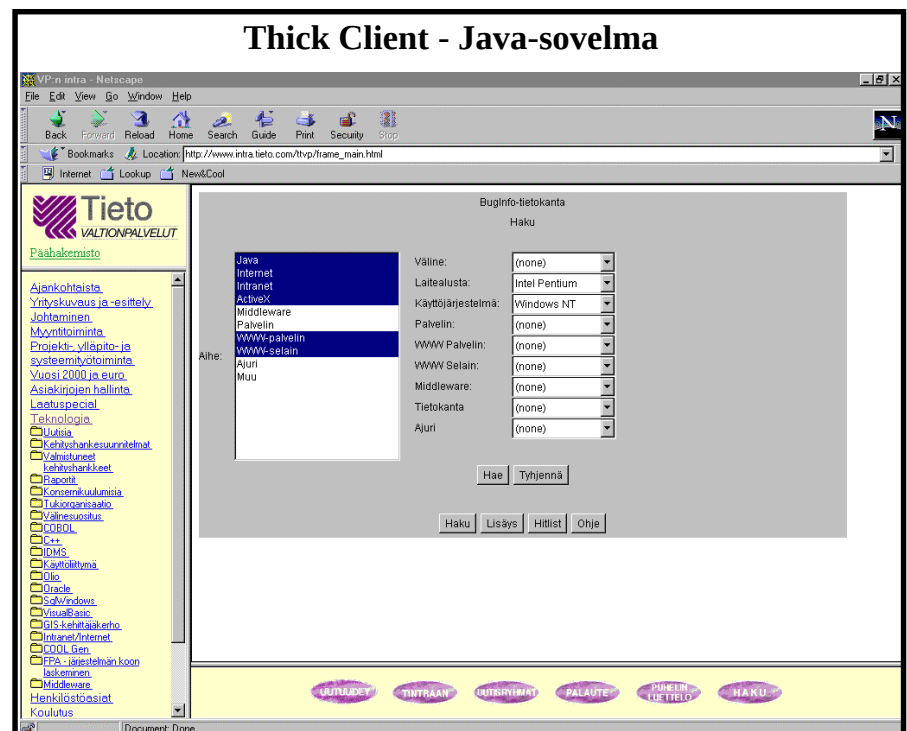
Sovelluspalvelinten valtteina on kuitenkin mm. parempi tilanhallinta, resurssien hallinta, transaktioiden hallinta, ja mahdollisuudet ympäristön monitorointiin. Lisäksi monet sovelluspalvelimet sisältävät tarpeellisia turvallisuus-, hakemisto- ja muita ominaisuuksia. On kuitenkin tapauskohtaisesti punnittava, mikä on paras mahdollinen ratkaisu.

Yhteenveto

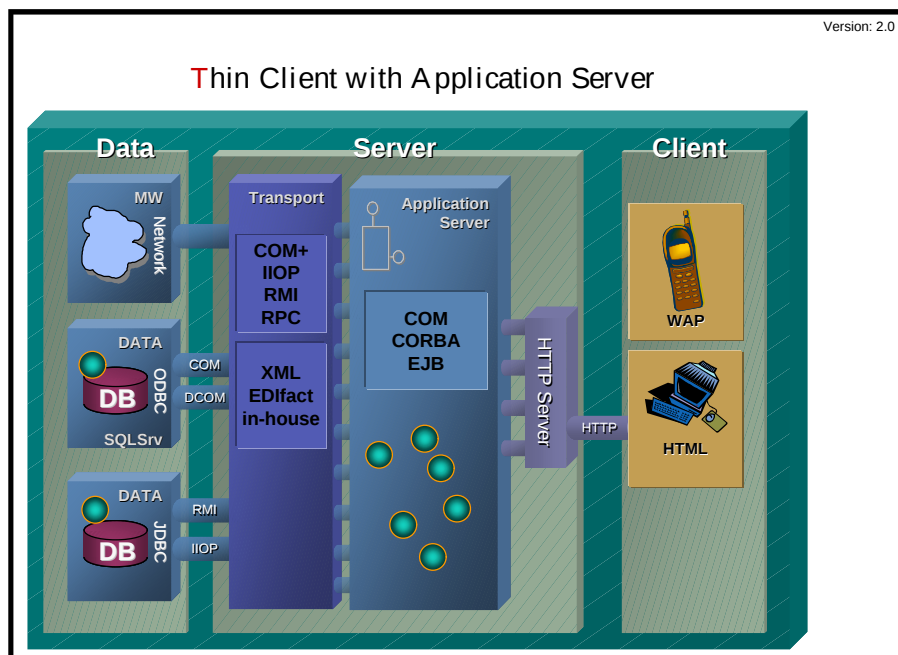
Arkkitehtuurin suunnitteluun ei ole mitään yleisesti hyväksyttyä prosessimallia, vaan arkkitehtuurin taso riippuu arkkitehdin sovellustuntemuksesta, taidoista ja sisäisestä intuitiosta.

Tietotekniikassa arkkitehtuuri voidaan jakaa esimerkiksi loogiseen arkkitehtuuriin ja tekniseen arkkitehtuuriin.

Arkkitehtuurikuvausten on tarkoitus kuvata, mistä osista tietojärjestelmä tai –järjestelmäkokonaisuus koostuu, miten se



Kuva 7: Thick client

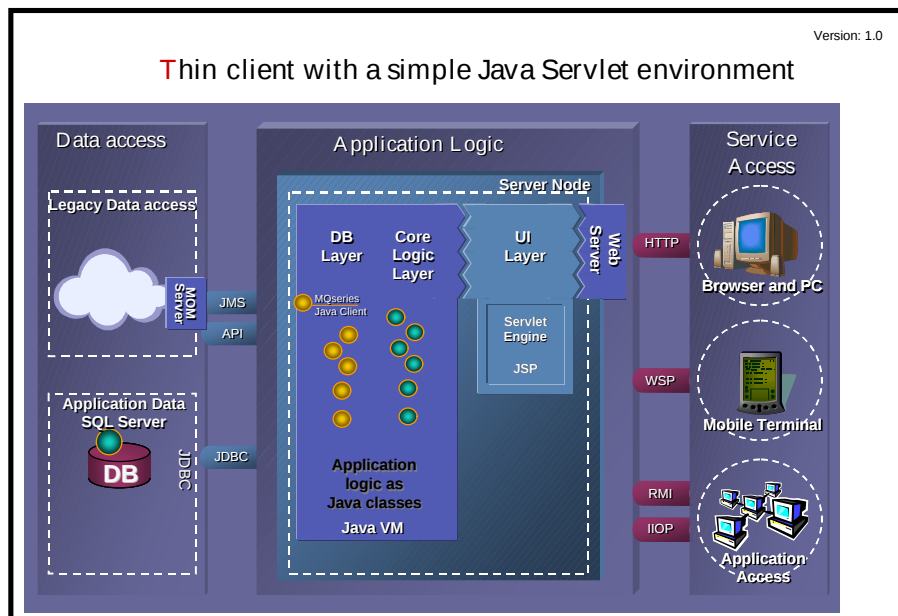


Kuva 8: Thin client sovelluspalvelinratkaisussa

on rakennettu tai tullaan rakentamaan, min-käläisiä tekniikoita ja teknologioita käytetään, sekä järjestelmän eri osien väliset yhteydet ja suhteet.

WWW-arkkitehtuurimalli on asiakas/palvelin-mallin erikoistapaus, jossa asiakkaana on normaalisti WWW-selain, joka käyttää palvelimen palveluja HTTP-protokollan kautta.

WWW-ratkaisuissa voidaan erottaa toisistaan intranet-, extranet- ja internet-tyyppiset ratkaisut, joilla on kullakin omat piirteensä ja rajoitteensa. Selaimessa suoritettavia asiakassovellustyyppisiä on myös kolme helposti erotettavissa. Nämä ovat pelkkää HTML:ää sisältävä thin client, HTML:n ja skriptikielen yhdistelmällä tuotettava "smart client" sekä thick client, joka voi olla esim. Java-sovelma tai Microsoft ActiveX-komponentti.



Kuva 9: Thin client Java Servlet-ympäristössä.

Palvelin voidaan toteuttaa dynaamisena sivupalvelimenä esim. Java Servlet-teknologialla tai Microsoftin Active Server Pages-teknologialla, joissa kummassakin on kevyen WWW-sovelluksen perustarpeet istunnonkäsittelyyn ja palvelupyyntöjen säikeistykseen.

Skaalautuvuutta, hallittavuutta, resursien hallintaa ja transaktioiden käsittelyä voidaan hankkia sovelluspalvelinvälikerroksella. Sovelluspalvelin-pohjainen monitasoarkkitehtuuri tarjoaa valtteina parempaa tilanhallintaa, resurssien hallintaa, transaktioiden hallintaa ja monitorointimahdollisuuksia. Valinta ratkaisujen välillä tehdään tapauskohtaisesti.

Lähdeluettelo ja luettavaa

Aikio A, Vornanen R.. Uusi sivistys-sanakirja, Otava. Helsinki, 1993.

Bass L., Kazman R. Architecture-Based Development. Carnegie-Mellon University Technical Report CMU/SEI-99-TR-007, 1999.

Monson-Haefel R. Enterprise JavaBeans, O'Reilly & Associates. Sebastopol, CA, 1999.

Pressman R.. Software Engineering – a practitioner's approach. McGraw-Hill, 1997.

Russell M. Introduction to Architecture. Proceedings in OOPSLA 1996. WWW:stä 12.11.2000 (<http://www.chimu.com/publications/oopsla96tutorial23/part8.html>)

Sommerville I.. Software Engineering, 5th edition. Addison-Wesley, 1995.

Simo Vuorinen,
TietoEnator Oyj