

# SUOMEN KUNTALIITTO

## Sairaalapalvelut

**TERVEYDENHUOLLON XXI ATK-PÄIVÄT**  
**29. - 30.5.1995**  
**Kalastajatorppa, Helsinki**

---

Konsultti Päivi Tamminen  
Twin-Systems Oy

**AVOIN ARKKITEHTUURI**  
**- Yhteistyön tekninen perusta**

## 1 Johdanto

Avoimuus on noussut aikaisempaa tärkeämmäksi kriteeriksi valittaessa tietojärjestelmiä. Avoimuus mahdollistaa toimittajien kilpailuttamisen ja toisaalta myös yhteistyön. Päätöksentekijät ovat huomanneet, että avoimilla ratkaisuilla voidaan organisaation tietotekniikkaratkaisut toteuttaa aikaisempaa halvemmalla. Lisäksi laman myötä valmiit ratkaisut ovat alkaneet kiinnostaa, enää ei haluta tehdä kaikkea itse tai yksin.

Tämä esitys koostuu kahdesta osasta. Ensimmäisessä osassa määritellään se, mitä avoimuudella tarkoitetaan. Toisessa osassa kuvataan Helsingin terveystieteiden keskuslaitoksen avoin sovellusarkkitehtuuriratkaisu.

## 2 Avoimuus

Pamela Gray, joka on yksi tunnetuimmista avoimien järjestelmien tutkijoista, määrittelee avoimuuden käsitteillä siirrettävyys, skaalattavuus ja liitettävyys. Hänen mielestään avoimen järjestelmä tulisi toteuttaa kaikki kolme ehtoa. (Gray 1991, 27.)

Siirrettävyydellä tarkoitetaan sitä, että sovellusohjelmiston tulee toimia eri tietokonejärjestelmissä. Siirrettävyyden tulisi olla mahdollista ilman, että sovellusohjelmistoon tehdään mitään muutoksia. Käytännössä siirrettävyys tarkoittaa sitä, että käyttäjä voi vapaasti valita sen laitteiston ja varusohjelmiston, jossa haluaa sovellusohjelmistonsa toimivan. (Gray 1991, 26.) Tästä syystä siirrettävyyttä pidetään usein avoimuuden synonyyminä ja ainoana ehtona.

Skaalattavuus tarkoittaa sitä, että ohjelmiston tulisi toimia erikokoisissa laitteistoissa. Tällöin käyttäjä voisi vapaasti valita oman organisaationsa kokoon nähden sopivan tietokonelaitteiston. (Gray 1991, 27.)

Liitettävyydellä Gray tarkoittaa sitä, että järjestelmien välinen kommunikointi tulisi olla mahdollista sovellusohjelmia muuttamatta (Gray 1991, 27). Sovellusten ja järjestelmien yleensä tulisi siis tarjota rajapinta, jonka kautta toiset järjestelmät pystyittäisiin niihin liittämään.

Seuraavissa luvuissa on kuvattu se, miten Helsingin terveystieteiden keskuslaitoksen on pyrkinyt näihin avoimuuden haasteisiin vastaamaan.

## 3 Helsingin terveystieteiden keskuslaitoksen arkkitehtuuriratkaisun tausta

Terveystieteiden keskuslaitoksen tietotekniikkajohto totesi keväällä 1993, että ennen uusien hankkeiden käynnistämistä täytyy laatia sovellusrakentamista ja -hankintaa ohjaava sovellusarkkitehtuuri. Tämän arkkitehtuurin avulla voidaan varmistua siitä, että ratkaisuista tulee avoimia.

Sovellusarkkitehtuurin laadinnalle asetettiin seuraavat vaatimukset:

1. Siirrettävyyksivaatimukset

Tietojärjestelmät pitää pystyä siirtämään laitteistosta ja varusohjelmistosta toiseen sovelluksia muuttamatta. Myös tietokannan hallintajärjestelmä on pystyttävä vaihtamaan.

2. Monilaiteympäristö

Arkkitehtuurin tulee tukea tilannetta, jossa sovelluspalvelut hajautuvat useille erimerkkisille laitteille. Käyttäjälle sovelluksen ilmiasun tulee olla samanlainen riippumatta siitä, mistä laitteistosta sovelluspalvelu saadaan.

3. Vaiheittainen eteneminen

Sovellukset pitää pystyä hankkimaan tai/ja tuottamaan vaiheittain. Uusia sovellusosuuksia pitää pystyä lisäämään aikaisempia muuttamatta. Lisäksi uudet osuudet pitää pystyä vaihtamaan liittämään jo käytössä olevaan sovelluskantaan.

4. Valmisohjelmistojen käyttö

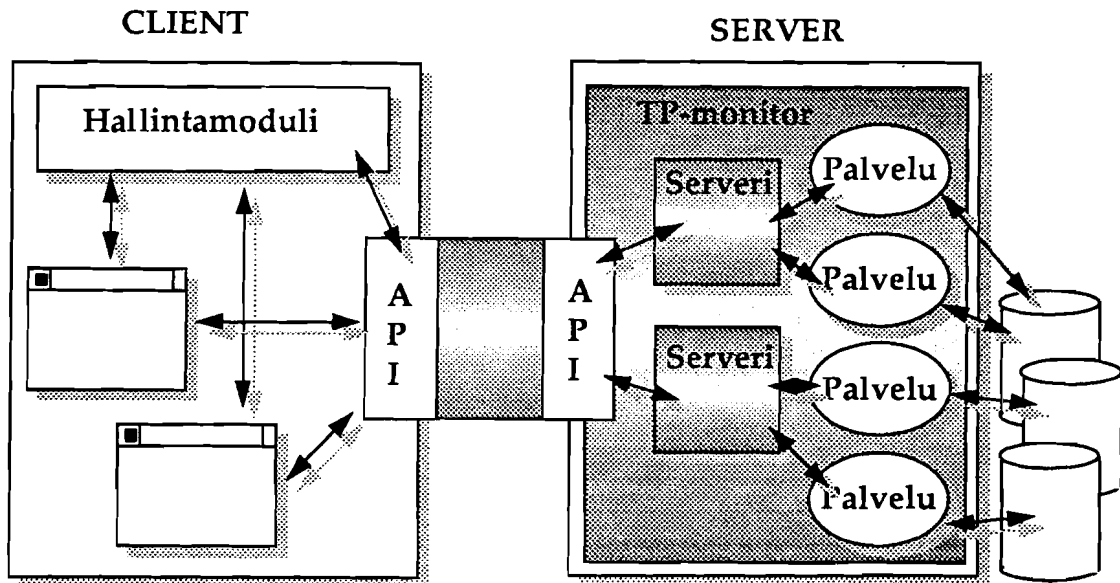
Sovellusarkkitehtuuri tulee rakentaa siten, että osa sovelluksista voidaan ostaa valmiina. Näiden valmissovellusten pitää kuitenkin pystyä hyödyntämään suoraan kaikille sovelluksille yhteistä tietovarastoa.

Terveysvirastossa oli jo aikaisemmin valittu käyttöliittymäksi Microsoftin Windows. Lisäksi terveystoimistossa oli käynnissä terveydenhuollon ja sosiaalitoimen kaikki toiminnot kattavan SQL-pohjaisen tietokannan määrittely.

#### 4 Helsingin terveystoimiston sovellusarkkitehtuuri

Helsingin terveystoimiston arkkitehtuuriratkaisu perustuu Työasema-palvelin -malliin. Siinä työasema ja palvelin keskustelevat keskenään ohjelmasta ja tietokannasta riippumattomien sanomien välityksellä. Työasemasovellus lähettää palvelimelle palvelupyynnön, jonka tämä käsittelee paikallisesti.

Helsingin terveystoimiston sovellusarkkitehtuuri voidaan jakaa kolmeen osaan, jotka on esitetty kuvassa 1. Nämä osat ovat työasemaosa (client), middleware ja palvelinosa (server).



Kuva 1. Sovellusarkkitehtuurin perusosat.

#### 4.1 Työasemaosa

Työasemassa sijaitsee osa sovelluksen logiikasta. Työasemasovellus hoitaa kommunikoinnin käyttäjän kanssa. Palvelinlaitteet ja tietokannat eivät näy suoraan työasemasovelluksille, vaan ne tekevät palvelupyyntöjä.

Terveysvirastossa työasemasovelluksia ja -palveluita tulee vähitellen lisää. Jotta välttyttäisiin päällekkäiseltä työltä ja pystyttäisiin hallitsemaan työaseman sovellusympäristö, työasemaosaan on rakennettu Modulinhallintaohjelmisto. Modulinhallintaohjelmisto huolehtii työasemasovellusten hallinnasta, sisäänkirjoittautumisesta, käyttöoikeuksista sekä palvelinlaitteiden palvelujen reititystiedoista.

#### 4.2 Middleware

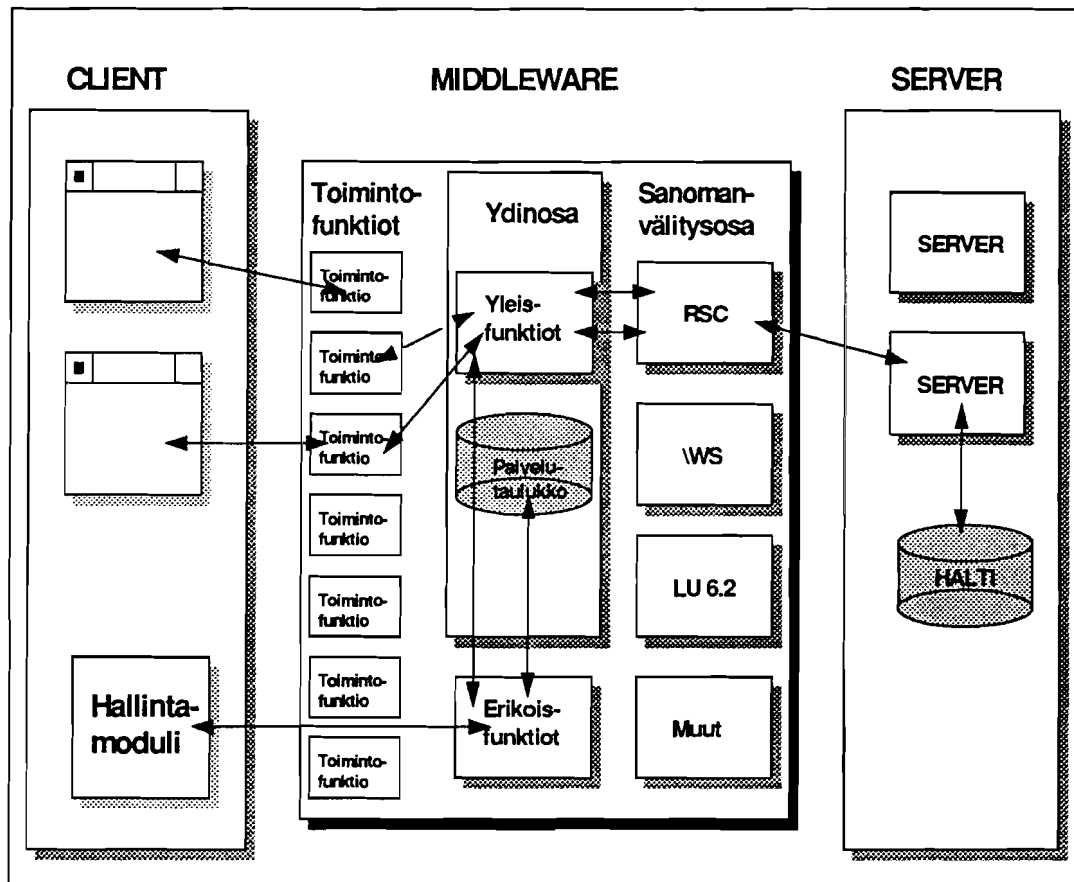
Työasemasovellusten ja palveluiden välinen kommunikointi hoidetaan sanomilla, joita Middleware välittää pisteestä toiseen. Sanomat eivät näy suoraan työasemasovellukseen, sillä tietoliikenne API on toteutettu niin että se peittää sovellukselta varsinaisen sanomien muodostamisen, lähettämisen ja vastaanottamisen.

Middleware on vielä jaettu kolmeen osaan, jotka ovat toimintofunktio-osa, ydinosa ja sanomanvälitysosa (kts. kuva 2). Toimintofunktiot muodostavat rajapinnan työasemasovelluksille. Ne tarjoavat korkeimman tason API:n sovellusohjelmoijalle.

Ydinosa tarjoaa yleispalveluita toimintofunktioille. Tällaisia yleispalveluita ovat tietotyyppien tarkis-

taminen ja merkkijonojen vienti palvelusanomalle. Lisäksi ydinosa huolehtii palvelusanomien reitityksestä.

Sanomavälitysosa eristää Middlewären ylemmät osat varsinaisesta tietoliikenneraajapinnasta. Ylempillä osilla on standardimuotoiset komennot, joilla ne kutsuvat sanomavälitysosan palveluita. Sanomavälitysosa mahdollistaa useiden eri tietoliikenneraajaintojen käytön (RSC, TUXIDO/WS ja LU 6.2).



Kuva 2. Middleware ja sen kommunikointi työasemasosan ja palvelimen kanssa.

#### 4.3 Palvelinosa

Sovelluksen palvelinosa koostuu palveluista. Palvelu toteuttaa työasemasovelluksen antaman palvelupyynnön. Palvelu ohjelmoidaan laite- ja varusohjelmistoriippumattomasti, ja se on siksi siirrettävissä ympäristöstä toiseen. Palvelujen siirrettävyyden ja ylläpidettävyyden takaamiseksi kutakin palvelutyyppiä varten on laadittu palvelurunko, josta ympäristöriippuvaiset osat on eristetty. Kaikki palvelut tuotetaan näitä runkoja käyttäen.

Yksi tai useampi palvelu yhdessä laiteriippuvan osuuden kanssa muodostaa ns. serveriohjelman. Serveriohjelman laite- ja varusohjelmistokohtainen osuus toimii palveluiden pääohjelmana, joka vastaanottaa saapuvan sanoman

ja ohjaa sen oikealle palvelulle. Serveriohjelman laiteriippuva osuus toteutetaan erikseen kuhunkin laite- ja tapahtumamonitorympäristöön.

#### 4.4 Tietokannan käsittely

Tietokannan käsittely voitaisiin toteuttaa täysin käytetystä tiedonhallintajärjestelmästä riippumattomaksi käyttämällä ns. gateway-tuotteita. Tällöin tietokantakäsittelyt olisivat suoraan siirrettävissä ilman koodikorjauksia tiedonhallintajärjestelmästä toiseen.

Gateway-tuotteiden käyttö johtaisi kuitenkin siihen, että kaikki tietokantakäsittelyt jouduttaisiin kääntämään ja optimoimaan ajonaikaisesti. Tämä hidastaa tietokannan käsittelyn nopeutta ja sitä kautta edellyttää suurempia laiteresursseja kuin upotetun SQL:n käyttäminen.

Terveysvirastossa käytetään volyymisistä tietokannan käsittelyssä upotettua SQL:ää. Avoimuus saavutetaan siten, että tietokannan käsittelyssä käytetään ainoastaan ANSI SQL:n ominaisuuksia.

#### 5 Arkkitehtuuriyhteistyö

Helsingin avointa sovellusarkkitehtuuria käsitellään tällä hetkellä Hyks-toimikunnan tietojenkäsittelytyöryhmässä. Tavoitteena on arkkitehtuuriratkaisu, joka yhteinen Helsingin yliopistollisen keskussairaalan, Uudenmaan sairaanhoitopiirin ja Medici Datan kanssa.

Tarkoituksena tässä yhteistyössä on sopia avoimen arkkitehtuurin yhteisistä osista. Tätä esitystä kirjoitettaessa on syntynyt yhteinen näkemys arkkitehtuuriratkaisusta työasemasovelluksen ja middlewaren osalta. Palvelujen osalta ratkaisu on vielä tekeillä.

Tällä hetkellä näyttää siltä, että yhteinen ratkaisu tulee päälinjauiltaan noudattelemaan tässä esityksessä kuvattua Helsingin terveystarvaston ratkaisua.

#### 6 Lopuksi

Avoimuutta voidaan edistää luomalla ja käyttämällä erilaisia standardirajapintoja. Rakentamalla sovellukset edellä kuvattua arkkitehtuurin mukaisesti voidaan minimoida sovelluksen siirtokustannukset ympäristöstä toiseen.

Pelkät rajapintaratkaisut eivät kuitankaan riitä. Sovellusten rakentajat ja toimittajat on myös motivoitava käyttämään avoimia ratkaisuja.

Pamela Grayn mukaan avoimet ratkaisut tulevat dominoimaan markkinoita. Käyttäjien on siis ajoissa valmistauduttava avoimiin järjestelmiin, muutoin he jäävät sul-

jettujen järjestelmien vangeiksi. (Gray 1991, 43.)

Lähteet

Gray, Pamela (1991) **Open Systems - A Business Strategy for the 1990s**. McGraw-Hill, London.