

LAPPEENRANNAN TEKNILLINEN KORKEAKOULU

Tietotekniikan osasto

## **INTERNET -PALVELUNHALLINTA**

Diplomityön aihe on hyväksytty Tietotekniikan osaston osastoneuvostossa

20.1.1999

Työn tarkastaja: Professori Olli Martikainen

Työn ohjaaja: DI Kari Koivisto

Lappeenrannassa 25.5.2000

Aarre Asikainen

Korkearannantie 7 H

54530 LUUMÄKI

Puhelin: +358 40 5177071

## **TIIVISTELMÄ**

Lappeenrannan teknillinen korkeakoulu

Tietotekniikan osasto

Aarre Asikainen

### **Internet -palvelunhallinta**

Diplomityö

2000

79 sivua, 25 kuvaa ja 3 taulukkoa

Tarkastaja: Professori Olli Martikainen

Hakusanat: Internet, Palvelu, Palvelunhallinta, Palveluntarjoaja, Sisällönhallinta

Keywords: Internet, Service, Service management, Service provider, Content management

Internet –palvelunhallinnan rooli kasvaa tulevaisuudessa ja sen kehittäminen on yhä tärkeämpää. Palvelut kehittyvät monipuolisemmiksi, niiden laatu paranee ja käyttömäärät kasvavat. Erityisesti palveluiden sisällönhallinnan rooli kasvaa ja palveluiden laadusta tulee merkittävä kilpailutekijä. Kokonaisvaltaisen palvelunhallinnan on pystyttävä vastaamaan joustavasti palveluntarjoajien uusiin tarpeisiin, jolloin sen on kehityttävä samaan suuntaan kuin palvelunhallinnan perinteisen telemaailman puolella. Palvelunhallinnan on tuettava kaikkia OSS – järjestelmien (Operating Support System) osa-alueita, jolloin sen laajamittaiseen toteuttamiseen tarvitaan siihen liittyvien erilaisten hallintarajapintojen standardointia. Palvelunhallintajärjestelmät on toteutettava mahdollisimman joustaviksi, jotta niiden piiriin saadaan uusien komponenttien lisäksi jo olemassa olevat komponentit. Mahdollinen palvelunhallinnan toteutustapa voi olla Javan ja CORBA:n (Common Object Request Broker Architecture) yhteiskäyttö, sillä ne takaavat järjestelmään tarvittavan joustavuuden muuttuvissa olosuhteissa kuten Internetissä.

## **ABSTRACT**

Lappeenranta University of Technology

Department of Information Technology

Aarre Asikainen

### **Internet service management**

Master's thesis

2000

79 pages, 25 figures and 3 tables

Supervisor: Professor Olli Martikainen

Keywords: Internet, Service, Service management, Service provider, Content management

The role of Internet service management will grow in the future and it's development becomes important. Services become more diverse, service quality gets better and service usage's grows. Especially the role of content management will grow and quality of service will become significant competitive advantage. Comprehensive service management has to be able to meet swiftly service provider's new needs. It has to evolve to the same direction than where the service management has evolved in the traditional telemanagement area. This means that Internet service management has to support all different OSS (Operating Support System) areas. Implementation of the service management in the large scale requires standardization of common management interfaces. Service management systems need to be flexible so that new and already existing service components can be brought under management. One possible way to implement the service management could use Java and CORBA technologies (Common Object Request Broker Architecture) together, because those can guarantee flexibility to the system in changing environment as Internet.

## **ALKUSANAT**

Kiitän työni tarkastajaa professori Olli Martikaista ja ohjaajaa Kari Koivistoa heidän arvokkaista näkemyksistään Internet –palvelunhallinnan alueelta ja ohjauksesta, joita heiltä sain diplomityöni aikana. Kiitokset myös kaikille Necsom Oy:n työntekijöille, jotka avustivat ja antoivat arvokkaita neuvoja tätä diplomityötä tehdessäni, erityisesti Front Officen väelle. Lisäksi haluan kiittää kaikkia niitä sukulaisiani, jotka kaikesta huolimatta uskoivat tämän työn valmistuvan jonakin päivänä.

Lopuksi suurimmat kiitokset vaimolleni Outille kaikesta kärsivällisyydestä, tuesta ja ymmärtämisestä, jota häneltä sain pitkän diplomityöni aikana.

Kiitokset kaikille.

## SISÄLLYSLUETTELO

1	JOHDANTO .....	1
1.1	Tausta .....	1
1.2	Ongelma .....	2
1.3	Tavoite .....	3
2	INTERNET PALVELUYMPÄRISTÖNÄ .....	4
2.1	Internet –verkkoympäristö .....	4
2.2	Palveluarkkitehtuurin kehitys.....	7
2.3	Palveluntarjoajat.....	9
2.3.1	ISP (Internet Service Provider) .....	10
2.3.2	ISP:t Suomessa.....	13
2.4	Sisällöntarjoajat.....	14
2.4.1	Portaalit .....	16
2.4.2	Ongelmat .....	17
3	PALVELUNHALLINTA.....	19
3.1	Telekommunikaatiopalvelut.....	19
3.2	Mitä on palvelunhallinta?.....	20
3.3	Klassinen palvelunhallinta .....	21
3.4	Laajennettu palvelunhallintamalli .....	22
3.5	Palvelut Internetissä .....	25
3.5.1	Palvelunjakelu ja siirtotiet.....	25
3.5.2	Peruspalvelut .....	26
3.5.3	Palvelut tulevaisuudessa.....	27
3.5.4	Palvelunhallinnan ongelma .....	28
3.5.5	Komponentit ja suunnittelumallit.....	28
3.6	Perinteisten lisäarvopalveluiden hallinta.....	29
3.7	Internet –palveluiden toteutusteknologiat .....	30
3.7.1	Socket –rajapinta ja CGI (Common Gateway Interface).....	31
3.7.2	DCOM (Distributed Component Object Model) .....	32
3.7.3	Java –pohjaiset teknologiat .....	36
3.7.4	CORBA (Common Object Request Broker Architecture) .....	42

3.7.5	Yhteenveto .....	48
4	HALLINNAN KOHTEET .....	51
4.1	Sisältö.....	51
4.2	Palvelut.....	53
4.3	Verkonhallinta.....	54
4.4	Asiakkaiden hallinta Internet -palveluntarjoajan ympäristössä.....	55
4.4.1	Asiakkaiden tunnistus .....	55
4.4.2	Tietoliikenteen salaus.....	56
5	HALLINTARATKAISUT .....	58
5.1	Sisältö.....	58
5.2	Palvelunhallinta.....	60
5.2.1	CORBA -pohjaiset toteutukset.....	60
5.2.2	Java 2 Enterprise Edition.....	62
5.2.3	Sovelluspalvelimet (Application Servers).....	65
5.3	Verkonhallinta.....	66
5.4	Verkkoelementtien ja palveluiden kehitys tulevaisuudessa .....	68
5.5	Kokonaisuuden hallinnan ratkaisut .....	71
6	JOHTOPÄÄTÖKSET.....	73
6.1	Saadut tulokset .....	73
6.2	Tulevaisuus .....	74
7	YHTEENVETO.....	76

## LYHENNELUETTELO

AH	Authentication Header
API	Application Programming Interface
ASP	Active Server Pages
CGI	Common Gateway Interface
CLSID	Class ID
CMIP	Common Management Information Protocol
CORBA	Common Object Request Broker Architecture
CoS	Common Object Services
DCOM	Distributed Component Object Model
DES	Data Encryption Standard
DII	Dynamic Invocation Interface
DLL	Dynamic Link Library
DNS	Domain Name Service
DSL	Digital Subscriber Line
DTD	Document Type Definition
EJB	Enterprise Java Beans
ESP	Encapsulating Security Payload
FTP	File Transfer Protocol
GPRS	General Packet Radio Service
GSM	Global System for Mobile communication
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
IDL	Interface Definition Language
IETF	Internet Engineering Task Force
IIOP	Internet Inter-ORB Protocol
IKE	Internet Key Exchange
IN	Intelligent Network
IP	Internet Protocol
IPSec	IP Security

ISDN	Integrated Services Digital Network
ISP	Internet Service Provider
ITU	International Telecommunications Union
J2EE	Java 2 Enterprise Edition
JDBC	Java Database Connectivity
JSP	Java Server Pages
LRPC	Lightweight Remote Procedure Call
MPEG2	Moving Picture Experts Group standard 2
OECD	Organization for Economic Cooperation and Development
OMG	Object Management Group
ORB	Object Request Broker
OSS	Operations Support System
PGP	Pretty Good Privacy
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RFC	Request for Comments
RMI	Remote Method Invocation
RPC	Remote Procedure Call
RSA	Rivest-Shamir-Adleman
SAP	Service Access Point
SDK	Software Development Kit
SE	Service Element
SGML	Standard Generalized Markup Language
SMS	Short Message Service
SNMP	Simple Network Management Protocol
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/ Internet Protocol
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
WAP	Wireless Application Protocol



WebCATV	Web Cable TV
WWW	World Wide Web
XML	Extensible Markup Language

# 1 JOHDANTO

## 1.1 Tausta

Nykyään viestintä- ja telekommunikaatioalojen teknologia kehittyy nopeasti. Globaalin tiedonsiirtoverkon eli Internetin kehittyminen ja kasvu on mullistanut viestintä- ja telekommunikaatioalaa. Heterogeenisen ja osittain kaottisenkin Internet –tiedonsiirtoverkon kasvu ja käyttäjämäärien jatkuva lisääntyminen luovat kasvavan tarpeen palveluiden hallinnalle ja palvelunlaadulle.

Perinteisessä telemaailmassa lisäarvopalvelut on toteutettu älyverkko eli IN –teknologialla (Intelligent Network). Älyverkkoihin pohjautuvissa järjestelmissä palveluiden hallintaan on pyritty luomaan formaalit menetelmät ja käsitteet, joiden avulla esimerkiksi uusien palveluiden luonti ja jakelu pystytään suorittamaan nopealla standardoidulla tavalla. Älyverkko teknologian standardoinnista vastaa kansainvälinen telekommunikaatioalan organisaatio ITU (International Telecommunications Union).

Internetin ympärille ei ole vielä saatu aikaiseksi samanlaisia palveluja ja niiden hallintaa kattavia standardeja. Tällä hetkellä ei ole vielä selviä ratkaisumalleja siitä, kuinka palveluiden hallinta olisi parhaiten hoidettavissa, mutta jo nyt Internet -maailmassa on näkyvillä selkeä kehityssuunta palvelukeskeiseen suuntaan.

Internet -palveluiden toteuttamiselle on olemassa useita kilpailevia teknologisia vaihtoehtoja, joista nopeimmin leviävä ja potentiaalisin on Java –ohjelmointikieli yhdistettynä sitä tukeviin selainohjelmiin. Uusia palveluja ja palvelumuotoja syntyy jatkuvasti lisää ja niiden yhtenäiselle hallinnalle etsitään keinoja. Järjestelmällisellä ja asianmukaisella palvelunhallinnalla pystytään takaamaan

asiakkaiden tarpeet täyttävä palveluiden laatu ja nopea kehitys. Internet - palvelunhallinnan kehittäminen on tulevaisuudessa asiakkaan tarpeiden tyydyttämisen ja palveluiden laadun kannalta hyvin keskeinen vaatimus.

## **1.2 Ongelma**

Palvelunhallinta on monimuotoinen ongelma, vaikka sitä käsiteltäisiin missä tahansa kohdeympäristössä. Tämän päivän viestintäjärjestelmissä palvelunhallinnan käsite ei ole uusi, sillä esimerkiksi perinteisen telekommunikaatioalan piirissä se on otettu huomioon ja siihen on kehitetty erilaisia ratkaisumalleja. Siellä palvelukulttuuri ja palveluiden tuotantologiikka ovat pitkälti vakiintuneita. Esimerkiksi GSM –verkon lisäarvopalvelut (Global System for Mobile communication) tuotetaan älyverkkoteknologialla. Perinteiseen puhelinverkkoarkkitehtuuriin kehitetyn älyverkkoteknologian taustalla oli ajatus luoda formaalit menetelmät ja tuki uusien palveluiden nopealle tuottamiselle ja esille tuomiselle. Edellä mainittujen ominaisuuksien lisäksi standardoitiin myös laitteistot ja palvelurajapinnat.

Internet on palveluympäristö, joka tarjoaa mahdollisuuden tuottaa ja välittää monipuolisia palveluja. Palvelut voivat sisältää tekstiä, ääntä, videota ja multimediaa. Tämä monimutkaistaa omalta osaltaan Internet -palvelunhallintaa ja tuo mukanaan uusia ongelmia. Ongelmaksi muodostuukin miten Internet -palvelut voidaan hallita yhtä luotettavasti ja tehokkaasti kuin palvelut perinteisen telekommunikaatioalan puolella.

### 1.3 Tavoite

Internet –palvelunhallinnan tavoitteena on saada aikaan toimiva palvelun tuotanto- ja jakelu ympäristö. Palveluiden tuotannon täytyy olla automatisoitua, jotta uudet palvelut saadaan toimitettua asiakkaiden ulottuville riittävän lyhyillä toimitusajoilla, kuitenkin kuluttamatta tarpeettoman suurta määrää rajallisia resursseja. Palvelunhallintaympäristön on myös kerättävä tarpeellinen määrä mittaus- ja laskutustietoa, jotta palvelunhallinta tukisi myös laskutusta ja mahdollistaisi järjestelmän toimivuuden seuraamisen.

Palvelunhallintaa voidaan tarkastella suppeamman klassisen palvelunhallintamallin tai laajemman palvelutuotantoprosessit ja niiden oheisjärjestelmät mukaan ottavan mallin kannalta. Tässä diplomityössä pyritään selvittämään ja tarkastelemaan palvelunhallintaa molempien mallien kannalta aloittaen perinteisestä palvelunhallintamallista ja laajentaen sitä kattamaan palveluntuotantoprosessit ja niiden oheisjärjestelmät. Työssä keskitytään palvelunhallinnan ongelmiin tuotannon, jakelun ja ylläpidon näkökulmasta. Erilaisia verkkoon liittyviä reititys- ja resurssienvarausongelmia ei käsitellä tämän työn piirissä, lukuun ottamatta reunareitittimien (edge routers) roolia tulevaisuuden palveluiden hallinnassa.

## 2 INTERNET PALVELUYMPÄRISTÖNÄ

### 2.1 Internet –verkkoympäristö

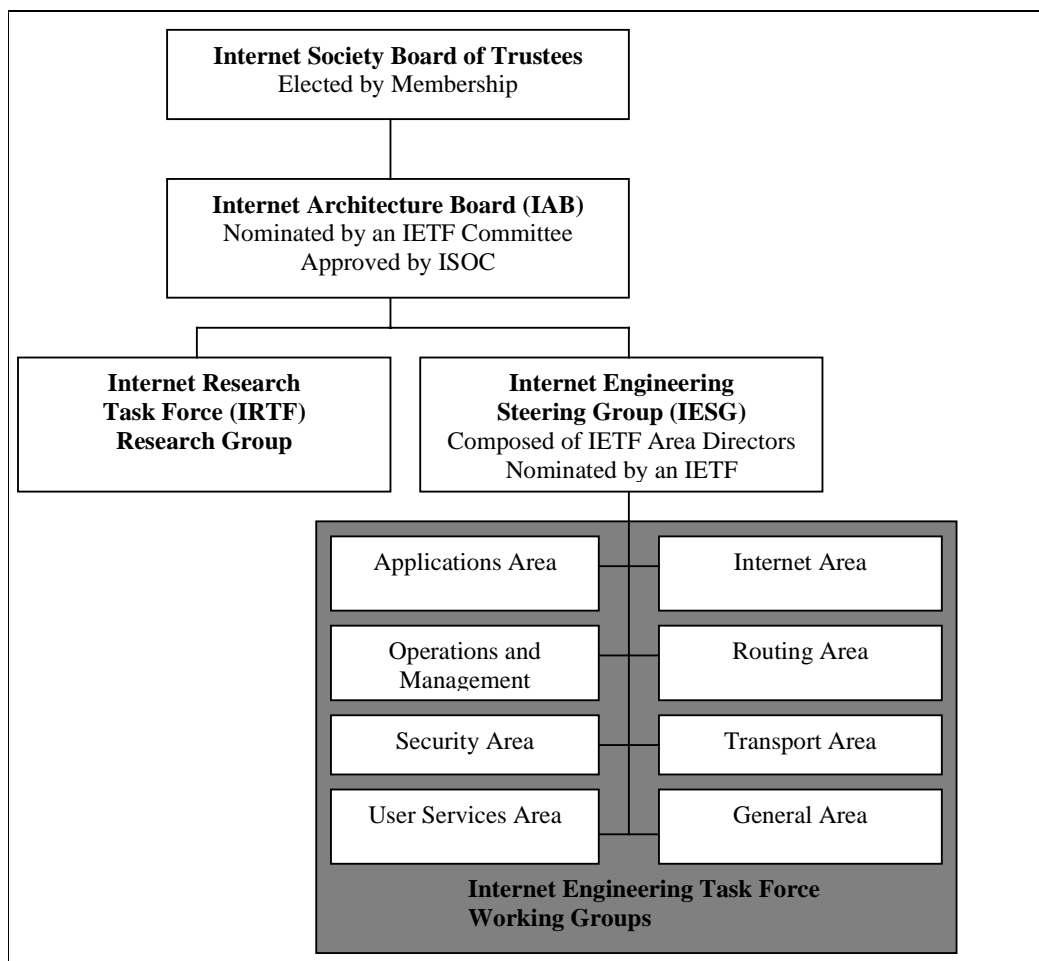
On arvioitu, että heinäkuussa 1999 Internetiin oli liittyneenä ainakin 56 miljoonaa tietokonetta ja vuoden 2000 alussa jo yli 72 miljoonaa tietokonetta (Lähde: Internet Software Consortium (<http://www.isc.org/> /1/). Käyttäjämäärien arviointi on hankalaa ja erityisesti se miten käyttäjä määritellään aiheuttaa vaihtelua arvioissa, joten arviot vaihtelivat 30 miljoonasta 70 miljoonaan käyttäjään vuonna 1999. Varmaa kuitenkin on, että käyttäjämäärät Internetissä tulevat edelleen lisääntymään.

Suomi on ollut edelläkävijä Internetin käytössä ja liikenneministeriön mukaan 1.7.1999 Suomessa olikin jo 111 Internetiin liittyntä konetta 1000 asukasta kohti /2/. Asukaslukuun suhteutettuna Suomi on Internetin käytön suhteen kärkimaita.

Internetin rakenne tällä hetkellä voidaan määritellä esim. itsestään järjestyväksi puoli-itsenäisten komponenttien kokoelmaksi. Teknisesti Internet rakentuu pakettipohjaisesta TCP/IP -protokollaa (Transmission Control Protocol / Internet Protocol) käyttävistä reitittimistä. Itse protokolla ei takaa palvelunlaatua (QoS, Quality of Service) eli se on ns. best effort –tyyppinen protokolla ja siitä käytössä oleva versio on IPv4. Kyseinen versio perustuu noin 20 vuotta vanhaan RFC 791:een (Request for Comments). IP –protokolla yksilöi verkkoon kytketyt laitteet kiinteämittaisen 32 bittisen osoitteen avulla. Yksittäinen osoite jaetaan verkkotunnisteeseen (network identifier) ja laitetunnisteeseen (host identifier). Osoitteet voidaan lisäksi luokitella A, B ja C –luokan osoitteisiin vaihtuvamittaisen verkkotunnisteen pituuden avulla /3/.

Tulevaisuudessa verkkoon kytkettyjen laitteiden lisääntymisen myötä siirrytään käyttämään versiota IPv6 (myös IPng), joka laajentaa protokollan tukemaa osoitevaruutta (128 bittinen osoite) ja tuo mukanaan myös muita parannuksia kuten esim. pakettien priorisoinnin. Kyseisten lisäysten avulla perusrakenteeltaan toimivaksi havaittu protokolla saadaan tukemaan laajempaa määrää IP –osoitteita, jotka nykyisen IPv4 osoitevaruuden käytössä ollessa uhkaavat loppua kesken. Kyseinen osoitteiden loppuminen onkin eräs merkittävä syy, miksi IPv6 – protokollaa on lähdetty kehittämään ja ottamaan käyttöön.

Internetissä ei ole mitään yksittäistä keskitettyä hallintoelintä tai sitä hallitsevaa organisaatiota, vaikka Internetin ylläpitoon ja siihen liittyviin standardeihin liittyikin muutamia niitä hoitavia yhteisöjä. Standardointiprosessiin liittyvät yhteisöt ja niiden väliset suhteet on esitetty kuvassa 1.



**Kuva 1:** Internetin standardointiprosessiin liittyvät yhteisöt /4/.

Internet on hyvin kevyesti organisoitu yhteisö, kun tarkastellaan sen käyttäjämääriä ja verrataan sitä moniin perinteisiin organisaatioihin. Sen vapaamuotoisuus mahdollistaakin mukautumisen jatkuvaan kasvuntarpeeseen välittämättä rajoituksista, joita järjestelmälliset ja tiukasti hallinnoidut prosessit toisivat mukanaan.

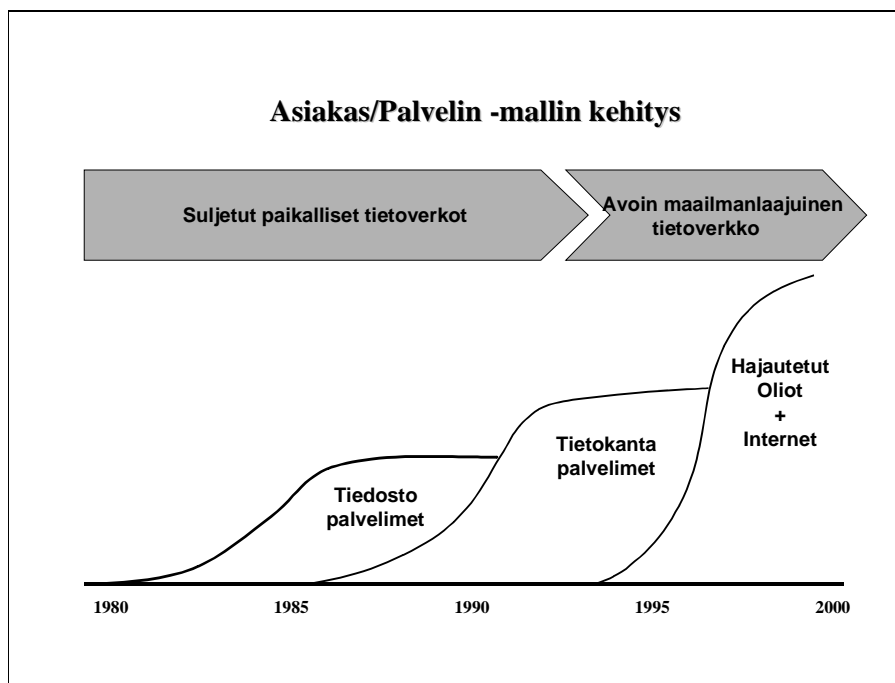
Tällä hetkellä Internet on myös kasvavassa määrin kaupallinen tiedonvälitysympäristö. Akateemisesta, yliopistot ja muut vastaavat instituutiot yhdistävästä, ei-kaupallisesta järjestelmästä on kehittynyt viime vuosien aikana globaali ja massiivinen, lähes jokaisen ihmisen ulottuvilla oleva kaupallinen

media. Internet –verkkoon syntyy koko ajan uusia palvelumuotoja ja yrityksiä, jotka haluavat tarjota asiakkailleen omia tuotteitaan ja palveluitaan perinteisten markkinointikeinojen rinnalla. Puhtaasti vain Internetissä toimivia yrityksiä on tällä hetkellä kuitenkin vielä lukumääräisesti vähän ja ne ovat yleensä tiettyjen erityisalojen yrityksiä esim. verkkokirjakauppoja. Niistäkin suurin osa on vielä tällä hetkellä taloudellisesti tappiollisia, mutta tulevaisuudennäkymät ovat hyvät ja kasvua tapahtuu koko ajan. Suuremman ryhmän muodostavat yritykset, jotka toimivat perinteisillä tavoilla, mutta hyödyntävät Internetin tuomat mahdollisuudet ja käyttävät sitä perinteisten markkinointi- ja tiedotuskanavien rinnalla. Tuo kasvu on luonut tarpeen myös erilaisille Internet palveluntarjoajille (ISP, Internet Service Provider), joita on tällä hetkellä kymmeniä tuhansia ympäri maailmaa.

## **2.2 Palveluarkkitehtuurin kehitys**

Internetin käyttäjämäärän kasvun ja tiedonsiirtoteknologioiden kehityksen rinnalla myös Internetissä sovellettavat palveluarkkitehtuurit ovat kehittyneet samalla, kun Internet on kasvattanut kaupallista painoarvoaan. Arkkitehtuuri on kehittynyt alkuaikojen keskustietokone ja pääte –arkkitehtuurista asiakas –palvelinmallin kautta nykypäivän hajautettuun oliopohjaiseen arkkitehtuurimalliin. Kuvassa 2 on esitetty Internetissä käytetyn asiakas / palvelin –mallin kehitys.



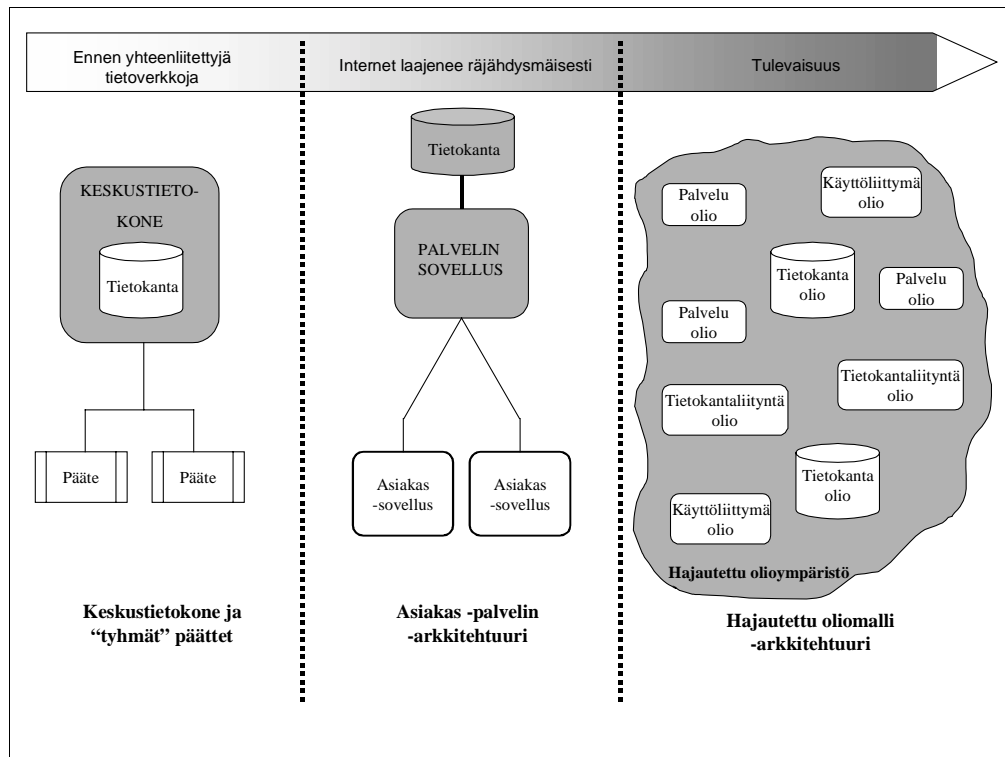


**Kuva 2:** Asiakas / Palvelin -mallin kehitys /5/.

Kun erilaisten tietokoneverkkojen tuomat edut huomattiin, myös perinteinen keskustietokone ja pääte –arkkitehtuuri kävi riittämättömäksi. Syntyi ns. asiakas / palvelin -arkkitehtuuri, jonka pohjalta edelleen tuotetaan ohjelmistojärjestelmiä myös Internet –ympäristöön. Puhdas asiakas –palvelin malli toimiikin suhteellisen hyvin kunhan järjestelmät eivät kasva liian suuriksi ja monimuotoisiksi. Kun erilaisten kommunikoivien järjestelmien lukumäärät kasvoivat ja syntyi tarve saada nämä usein toteutukseltaan heterogeeniset järjestelmät toimimaan yhdessä, kävi puhdas asiakas –palvelinmallin soveltaminen riittämättömäksi.

Kompleksisten järjestelmien kehittämistä varten tarvittiin uusi arkkitehtuurimalli, jonka oli tuettava hajautusta ja erilaisten kommunikointirajapintojen määrittelyä. Järjestelmiä alettiin suunnitella ja toteuttaa oliopohjaisesti. Näin syntyi hajautettu oliopohjainen arkkitehtuurimalli, jonka pohjalta Internetiin saadaan tuotettua luotettavia vikasietoisia järjestelmiä. Juuri tähän tarpeeseen luotu ratkaisu on Javan ja OMG:n (Object Management Group) CORBA -teknologian (Common Object Request Broker Architecture) yhdistelmä, joka pohjautuu oliopohjaisen

hajautettuun arkkitehtuurimallin. Kuvassa 3 on esitetty palveluarkkitehtuurin evoluutio kohti hajautettua oliomallia.



**Kuva 3:** Palveluarkkitehtuurin evoluutio kohti hajautettua oliomallia.

### 2.3 Palveluntarjoajat

Palveluntarjoajat käsitetään Internetissä vielä nykypäivänä lähes yksinomaan ISP:inä (Internet Service Provider), vaikka muitakin palveluntarjoajia jo löytyy. Itse palveluntarjonta on käsitteenä paljon laajempi. Sen alle voidaan ajatella kuuluvan erilaiset Internetin ympärillä toimivat osapuolet, jotka tuottavat ja tarjoavat palveluja muille niitä käyttäville osapuolille. Internetin resurssien eli tiedonsiirtokapasiteetin moninkertaistuminen ja teknologian kehittyminen tuovat kasvavassa määrin mukaan uudenlaisia palveluntarjoajia esimerkiksi julkaisutoimintaan ja elektroniseen kaupankäyntiin. Mahdollisuudet ovat

kuitenkin lähes rajattomat, ja mikäli Internet kehittyy maailmanlaajuiseksi laajakaistaiseksi palveluverkoksi, saattaa se ainakin osittain korvata nykyiset perinteiset yhdenpalvelun verkot. Esimerkiksi nykyään on jo mahdollista soittaa Internet –puheluita mihin päin maailmaa tahansa tietokoneelta mikäli omistaa tarvittavan ohjelmiston ja oheislaitteet.

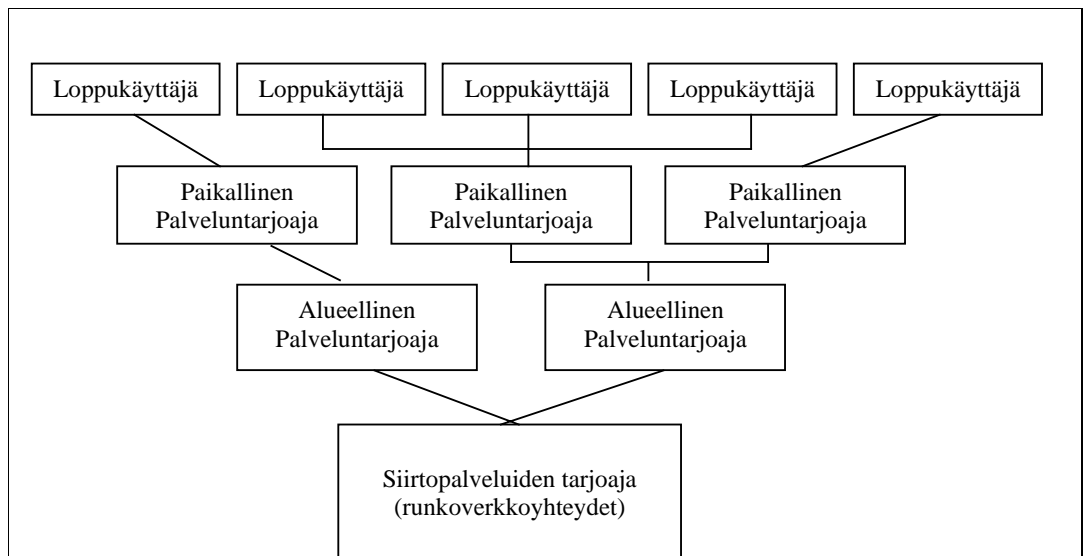
Nykyinen yleinen puhelinverkko (PSTN, Public Switched Telephone Network) ja Internet tulevatkin todennäköisesti pitkään täydentämään toisiaan. On oletettavaa, että osa puheluliikenteestä siirtyy IP -pohjaiseksi Internetiin ja puhelinverkko pysyy osalle loppukäyttäjistä keinona päästä Internetiin eli ns. access –verkkona. Internet saattaa myös kehittyessään ja nopeutuessaan osittain korvata muita viestintävälineitä kuten nykyisen televisioverkon. Myös tässä tapauksessa muutos voi olla kaksisuuntainen, sillä jo nyt on olemassa kaapelimodeemiyhteyksiä Internetiin, jotka hyödyntävät kaapelitelevisioverkkoa ja sen siirtotietä.

### **2.3.1 ISP (Internet Service Provider)**

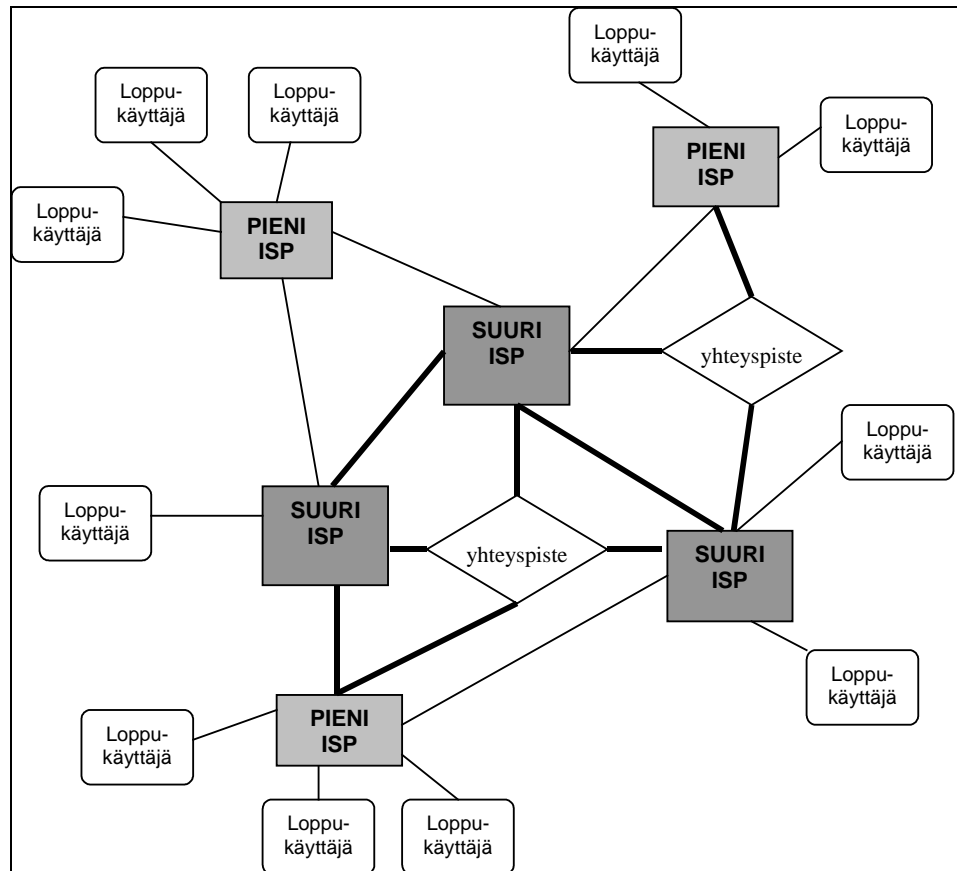
ISP on yritys, joka tarjoaa Internet -yhteyksiä ja niihin liittyviä palveluja omien resurssiansa kautta korvausta vastaan asiakkailleen. ISP:ita on useaa eri koko luokkaa ja ne profiloituvat yleensä tiettyihin asiakasryhmiin. Karkeasti jaettuna voidaan ajatella, että pienemmät paikalliset ISP:t keskittyvät pelkästään loppukäyttäjiin (yksityiset henkilöt sekä pienet yritykset) ja koon kasvaessa asiakkaina voi olla myös yrityksiä tai organisaatioita. Suuren tiedonsiirtokapasiteetin omaavat ISP:t myös jälleenmyyvät tiedonsiirtokapasiteettia ja yhteyksiä pienemmille palveluntarjoajille. On myös olemassa palveluntarjoajia, jotka tarjoavat yhteyksiä laajakaistaiseen runkoverkkoon. Esimerkiksi Suomessa toimivan Funetin päätuote on tarjota yhteyksiä runkoverkkoon akateemisille oppilaitoksille ja muille instituutioille. Tämä toimintamalli juontaa juurensa ajalta, jolloin Internet oli vielä puhtaasti akateeminen oppilaitosten välinen tiedonsiirtoverkko. ISP:t ovat yleensä syntyneet

täyttämään jonkin tietyn asiakaskunnan tarpeen. Tästä johtuen on syntynyt myös palveluntarjoajia, joiden päätuote poikkeaa edellä mainituista linjoista yrityksen koosta riippumatta. Tällä hetkellä lähes kaikki ISP:t kuitenkin ovat myös jossain muodossa mukana tarjoamassa yhteyksiä loppukäyttäjille. Internetin kehittyminen ei-kaupallisesta tiedonsiirtoverkosta kaupalliseksi verkoksi tulee todennäköisesti aiheuttamaan joitakin rakenteellisia muutoksia.

Ideaalisessa jälleenmyyntiin perustuvassa hierarkiamallissa loppukäyttäjät ostavat yhteyden ja palvelunsa paikalliselta pieneltä ISP:ltä, joka ostaa tietyn tiedonsiirtokapasiteetin ja yhteyden ulkomaailmaan suuremmalta alueelliselta ISP:ltä. Tämä puolestaan on liittynäänä siirtopalveluiden tarjoajaan, joka tarjoaa liittymän ja palvelut Internet –runkoverkon yhteyksiä varten. Kuvassa 4 on esitetty Internetin hierarkkinen rakenne palveluntarjoajien kannalta. Reaalimaailmassa tilanne ei kuitenkaan ole näin yksinkertainen, sillä eri ISP:ien välillä on sekä kahden- että monenkeskisiä palvelusopimuksia. Tällaisissa tapauksissa tietoliikenne ohjautuu yhteispisteiden kautta (exchange point), joissa useat palveluntarjoajat voivat vaihtaa liikennettä. Kuvassa 5 on esitetty Internetin todellinen rakenne ISP:n näkökulmasta.



**Kuva 4:** Internetin hierarkkinen rakenne palveluntarjoajien kannalta.



**Kuva 5:** Internetin todellinen rakenne ISP:n näkökulmasta.

Osa perinteisistä telekommunikaatioalan operaattoreista on alkanut myös toimia ISP:nä muiden toimintojensa ohella. Operaattorien rooli tulee tulevaisuudessa muuttumaan. Yhä suurempi osa tietoliikenteestä on dataa (IP -liikennettä) ja perinteisen puhelinliikenteen suhteellinen osuus pienenee. ISP:n ja operaattorin raja saattaa tulevaisuudessa hämärtyä. Osa Internetissä toimivista operaattoreista voi keskittyä IP -liikenteen reitittämiseen runkoverkoissa, jolloin niitä voi verrata nykyisiin kaukopuheluoperaattoreihin. Toiset taas voivat keskittyä uusien palveluiden tarjontaan loppukäyttäjille. Suuri osa operaattoreista ja ISP:sta fokuoitetu tulevaisuudessa jollekin suhteellisen kapealle hallitulle sektorille saavuttaakseen hyvän kilpailukyvyyn. ISP:ien rakenteissa, toimintamalleissa ja yrityskulttuureissa tulee olemaan suuria eroja riippuen siitä onko kyseessä alunperin puhtaasti Internet -maailmaan keskittynyt yritys vai perinteiseltä

telekommunikaatio puolelta siirtynyt operaattori. Internet -maailmaan siirtyneet teleoperaattorit ovat yleensä kooltaan suurempia kuin nuoret pelkästään Internet – maailmaan keskittyneet yritykset.

On arvioitu, että tammikuussa 1998 eripuolilla maailmaa oli yli 30000 Internet palveluntarjoajaa. Myös ISP:ien lukumääriä, kuten Internetin käyttäjämääriä, on hankala arvioida tarkasti, joten arvio on suhteellisen epätarkka ja muuttuu koko ajan. Internetiin liittyvät tunnusluvut ovatkin yleensä tilastollisia arvioita /6/.

### **2.3.2 ISP:t Suomessa**

Tällä hetkellä on selvästi nähtävillä kehityssuunta, jossa ISP:ien lukumäärä vähenee ja koko kasvaa yritysostojen ja fuusioiden kautta. On oletettavaa, että samanlainen suuntaus jatkuu myös ajan myötä maailmanlaajuisesti ja näin syntyy suuria globaalisti toimivia palveluntarjoajia. Kuitenkaan pienet paikalliset palveluntarjoajat eivät tule aivan täysin häviämään.

Suomen Internet -markkinat ovat muiden telemarkkinoiden tapaan kehittyneet pitkälle muuhun maailmaan verrattuna. Vapaa markkinakilpailu näkyy mm. Internet-palveluntarjoajien suurena määränä sekä alhaisina hintoina. Internet-palveluntarjoajia (ISP) on Suomessa kymmeniä, vaikkakin muutaman viime vuoden aikana on Internet-palveluiden tarjonnassa tapahtunut myös keskittymistä. Kolmen suurimman palveluntarjoajan (Sonera Internet, Finnet/HPY/Kolumbus ja Saunalahden Serveri) markkinaosuus alkuvuonna 1999 oli yksityisasiakkaiden osalta yli 80 %. Useat Internet -palveluntarjoajat toimivat vain alueellisesti, esimerkiksi yhdellä telealueella /7/.

Suomen Internet -palveluiden hintataso on OECD -maista halvin. Hinnoittelu on nykyään ns. "flat rate" -hinnoittelua, mikä tarkoittaa, että käytöstä maksetaan

kiinteä kuukausimaksu riippumatta yhteysajasta. Ainut aikaveloitukseen perustuva osa on tavallinen puhelumaksu ajalta, jolloin on oltu yhteydessä verkkoon /7/.

Suomessa toimivien ISP:ien suhteellisen pienestä koosta johtuen on oletettavaa, että niillä on käytössä paljon ilmaisia tai ainakin edullisia ohjelmistoja ja laitteita. Pienten palveluntarjoajien on ollut mahdotonta investoida suuria määriä rahaa kalliisiin laitteisiin ja ohjelmistoihin. Onkin luultavaa, että suuri osa tämän hetken pienistä ISP:sta Suomessa tuottaa palvelunsa tavallisella suhteellisen tehokkaalla PC -laitteistolla. Käyttöjärjestelmänä on usein Linux, joka jo nyt alkaa vallata sijaa muilta käyttöjärjestelmiltä myös yritysmaailmassa. Onkin ennakoitu, että esimerkiksi Saksassa Linux kasvattaa edelleen nopeimmin markkina-asemaansa. Linux käyttöjärjestelmän suosio perustuu sen ilmaisuuteen, luotettavuuteen ja avoimuuteen. Sen päälle onkin syntynyt jo kattava valikoima ilmaisia sovelluksia ja kaupalliset sovelluksetkin lisääntyvät kasvavalla tahdilla.

Linux käyttöjärjestelmä kasvattaa markkina-asemaansa myös maailmanlaajuisesti. International Data Corporation (IDC) on arvioinut, että Linux käyttöjärjestelmällä oli kymmenen miljoonaa käyttäjää vuoden 1998 lopussa ja että 17 prosenttia Intel -pohjaisista palvelinratkaisuista sisälsi Linuxin. Lisäksi teknologia-alan tutkimusyritys Forrester on arvioinut Linux / Apache -palvelimen yhdistelmän olevan 50 prosentissa kaikista verkkopalvelimista (Markku Sillanmikko, Linux and IBM -esitelmä, 14.10.1999 Linux@work seminaari /8/).

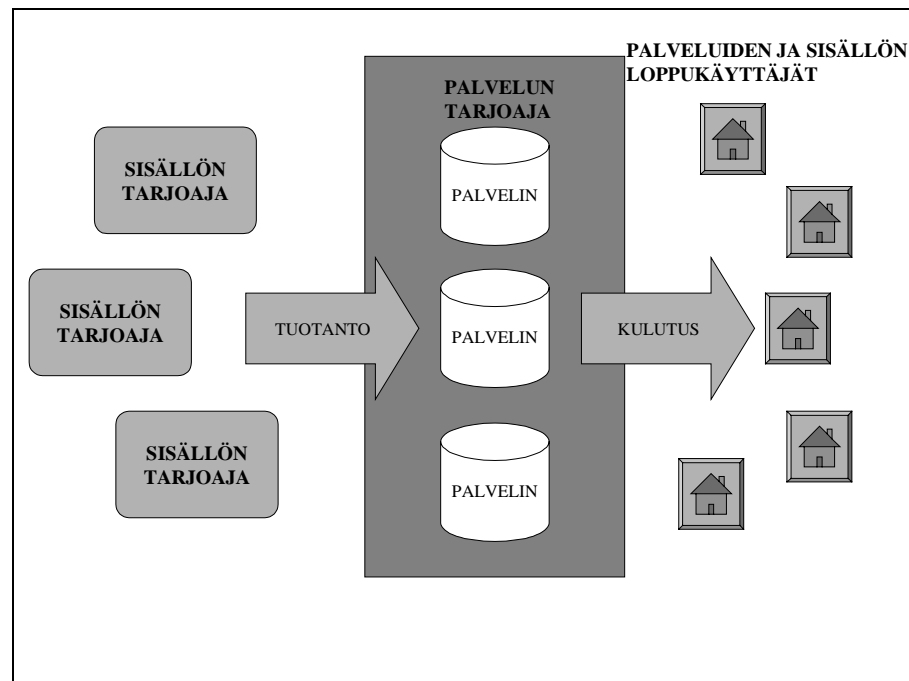
## **2.4 Sisällöntarjoajat**

Sisällöntarjoajaa Internetissä voidaan pitää omana suhteellisen itsenäisenä käsitteenä. Periaatteessa sisällöntarjoajaksi voitaisiin kutsua mitä tahansa osapuolta, joka tuottaa sisältöä palveluna Internetiin kenen tahansa saataville. Nykyään sisällöntarjoajalla yleensä kuitenkin tarkoitetaan osapuolta, joka pyrkii tuottamaan sisältöä tai palveluja liiketaloudellisin perustein joko rahoittamalla

toimintansa mainosrahoituksella tai laskuttamalla palvelun käyttäjiä. Myös eräät organisaatiot voivat toimia sisällöntarjoajina ja tällöin ne hankkivat tarpeelliset varat muita keinoja käyttäen eivätkä pyri tuottamaan voittoa. Sisältöön perustuvan liiketoiminnan merkitys Internetissä kasvaa. Palveluiden käyttäjät osaavat tulevaisuudessa vaatia yhä erikoistuneempia ja heidän tarpeensa täsmällisesti kattavia palveluja. Sisällön merkitys Internet –palveluissa kasvaa sekä yritys- että yksityisasiakkaiden piirissä.

Tällä hetkellä Internetissä toimii sisällöntarjoajia, jotka omistavat itse tarvittavat laitteet ja resurssit sisällönjakelua ja verkkoon liittymistä varten. Nekin kuitenkin yleensä joutuvat ostamaan liittynnän ulkoverkkoon joltain ISP:ltä. Tulevaisuudessa syntyy sisällöntarjoajia, jotka tulevat keskittymään puhtaaseen sisällöntarjontaan. Siihen keskittyvät yritykset ja organisaatiot ostavat tarvitsemansa Internet toiminnan edellyttämät resurssit ja palvelut yleensä ISP:iltä. Kyseinen toimintamalli tuo kustannus- ja resurssisäästöjä, koska yritys voi keskittyä olennaiseen eli sisällöntarjontaan tarvitsematta välittää teknisistä yksityiskohdista ja laitteista. Lisäksi se tuo mahdollisuuden myös muille kuin alan asiantuntijoille tulla Internetiin tarjoamaan palvelujaan. Palveluntarjonta monipuolistuu tämän hetken markkinointi ja mainonta keskeisestä ajattelumallista. Tarjottu sisältö voi olla muodoltaan mitä tahansa verkossa välitettyä dataa esim. tekstiä, ääntä, liikkuvaa kuvaa tai multimediaa. Tulevaisuudessa sisällöntarjontaan erikoistuneet organisaatiot ja yritykset tulevat yleistymään entisestään ja keskittyvät yleensä omille tarkoin määritellyille suhteellisen kapeille palvelusektoreille, jonka tietotaito niillä on hyvin hallussaan. Kuvassa 6 on kuvattu Internetin palveluiden ja sisällön tuotanto- ja jakelumalli.





**Kuva 6:** Internet Palveluiden ja sisällön tuotanto- ja jakelumalli.

#### 2.4.1 Portaalit

Portaali on eräänlainen yhdistelmäpalvelu, joka tarjoaa kokoelman Internet – palveluita samassa paikassa. Käytännössä portaalija voi olla erilaisia riippuen siitä mihin tarkoitukseen ne on suunniteltu. Tulevaisuudessa käyttäjä voi halutessaan muokata portaalin tai osia siitä omien tarpeidensa mukaan. Nykyään Internet portaalit ovat enemmän tai vähemmän vielä erilaisia linkki- eli osoitekokoelmia. Suomessa toimivia portaalija ovat esimerkiksi [www.mtv3.fi](http://www.mtv3.fi) ja [www.soneraplaza.fi](http://www.soneraplaza.fi). Portaalit kuitenkin kehittyvät yhä enemmän palvelukeskeiseen suuntaan. Tulevaisuudessa syntyy suuria ja monipuolisia portaalija, jotka profiloitetaan massoja varten. Ne tarjoavat jokaiselle jotakin mutta eivät ole kovin spesifisiä ja niiden käyttäjäkohtaiset yksilöintimahdollisuudet ovat vähäisiä. Niiden rinnalle syntyy erikoistuneita pienempiä portaalija, jotka täyttävät tietyn tarkkaan määritellyn tarpeen ja joita käyttäjä voi muokata itselleen

sopivaksi. Portaali tunnistaa käyttäjän ja palvelee tätä luodun asiakasprofiilin mukaisesti. Portaalien avulla käytettäviä palveluja voivat olla esim. matkapuhelimiin liittyvien palveluiden asetukset ja muutokset. Portaalit tuovat mukanaan mahdollisuuden kohdistaa mainontaa asiakasprofiilien mukaisesti. Tämä mahdollistaa kohdistetun mainonnan eli käyttäjälle välitetään vain sellaisia mainoksia, joista hän on todella kiinnostunut.

#### **2.4.2 Ongelmat**

Palveluiden ja sisällön tuottajien on tärkeää tarjota asiakkaalle helppo tapa hoitaa palveluiden käytöstä maksaminen. Tällä hetkellä Internet –maksaminen kehittyikin nopeasti ja tarjolla on useita erilaisia ratkaisumalleja. Mikromaksamiseen perustuvat mallit tuottavat laskutustietoja eli maksutapahtumia asiakkaan käyttäessä palvelua. Tietty palvelu tuottaa tietyn määrän maksutapahtumia eli ns. sykäyksiä. Sykäyksellä on tietty hinta ja asiakasta laskutetaan tämän perusteella. Toiset mallit perustuvat ns. elektroniseen kukkaraan eli asiakkaalla on valmiina tietty määrä verkkorahaa ladattuna itselleen ja hän maksaa sillä käyttämänsä palvelut. Varsinaisia tuotteita ja malleja kyseiseltä alueelta on jo useita ja niitä ei tarkemmin tarkastella tämän työn piirissä. Tulevaisuuden kannalta on kuitenkin tärkeää, että löydetään yhtenäinen teknologia tai standardi, jonka avulla eri Internet -maksamiseen liittyvät teknologiat saadaan luotettavasti ja saumattomasti toimimaan yhdessä. Internet -maksamiseen liittyy myös läheisesti asiakkaan autentikointi eli käyttäjän identiteetin varmistaminen. Autentikointi -malleja on olemassa myös useita erilaisia, joista toiset sopivat Internetiin paremmin kuin toiset. Kaikki Internetissä toimivat autentikointi -mallit pohjautuvat periaatteessa kuitenkin jonkinlaisen sähköisen allekirjoituksen käyttöön. Toteutustavat ja menetelmien luotettavuus vaihtelevat. Eräs mielenkiintoisimmista autentikointi -malleista soveltaa julkiseen avaimen perustuvaa RSA (Rivest-Shamir-Adleman) algoritmia, jota pidetään

suhteellisen luotettavana tapana hoitaa käyttäjän tunnistus avoimessa Internet - ympäristössä.

Verkkopalveluiden käyttäjille on tarpeen taata riittävän laajakaistainen tiedonsiirto yhteyden päästä päähän. Tulevaisuuden verkkopalvelut tarvitsevat yhä nopeamman tiedonsiirtoyhteyden ja kyseinen asia on ratkaistava jollakin tekniikalla. Yritysasiakkaille tämä ei ole yleensä kustannusongelma mutta yksityisasiakkaat tarvitsevat edullisen ja toimivan laajakaistateknologian. Tällä hetkellä käytettyjä liityntäteknologioita ovat ISDN (Integrated Services Digital Network), eri DSL –teknologiat (Digital Subscriber Line) ja kaapelimodeemit.

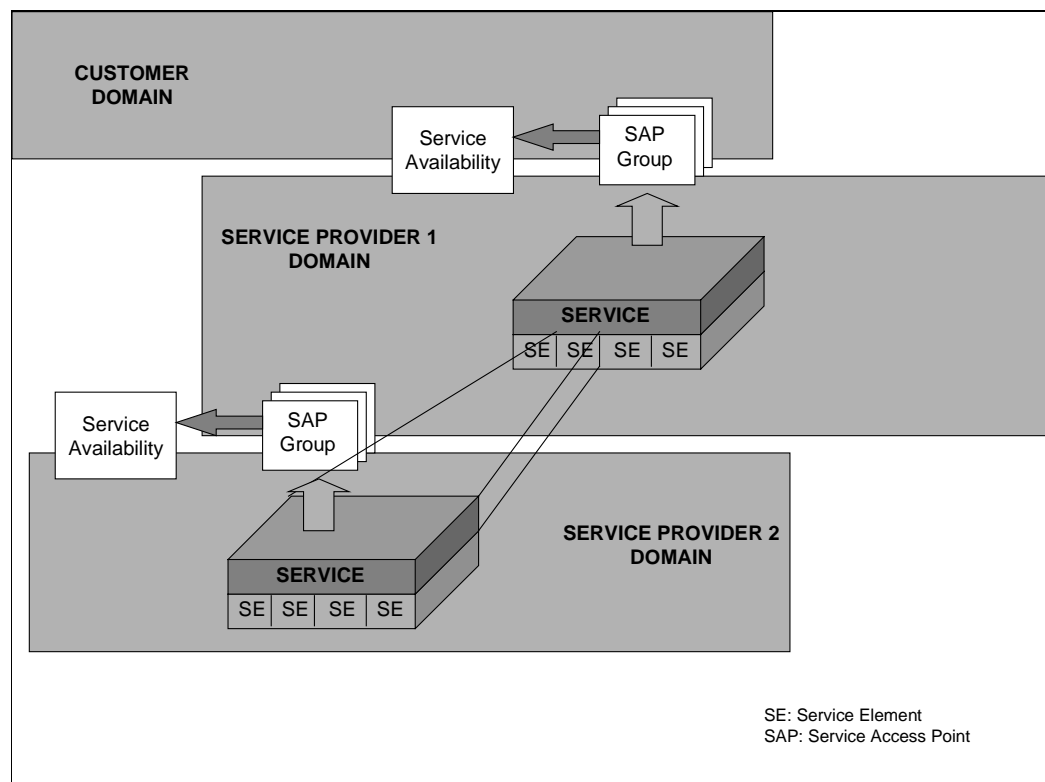
Palveluiden tarjonnan on kasvettava huomattavasti, jotta niiden käyttö omaksuttaisiin laajassa mittakaavassa. Eräs merkittävä tekijä palveluiden käytön yleistymisessä on se, että uudet palvelut suunnitellaan asiakaslähtöisesti ja niille on olemassa asiakkaan puolelta selkeä tarve. Esimerkiksi erilaisia pankkipalveluja voi jo käyttää Internetissä ja niiden suosio onkin kasvanut nopeasti. Myös Internetin jatkuva kasvu vaikuttaa suuresti palveluiden käytön yleistymiseen, sillä palvelut ovat jokaisen Internetiä käyttävän saatavilla. Yksittäisen käyttäjän kannalta ongelmaksi voi myös tulevaisuudessa nousta se, että tarjolla on suuri joukko palveluja joista hänen on vaikea valita.

### 3 PALVELUNHALLINTA

#### 3.1 Telekommunikaatiopalvelut

Telekommunikaatiopalvelu on joukko itsenäisiä toimintoja, jotka ovat kiinteä osa yhtä tai useampaa liiketoimintaprosessia. Tämä toiminnallinen joukko muodostuu laitteistoista ja ohjelmistokomponenteista sekä myös niiden alla sijaitsevasta tiedonsiirtomediasta /9/.

Nykyään monipuolisten palveluiden tuottaminen voi vaatia useiden erilaisten omien ja ulkopuolisilta tahoilta (esimerkiksi muilta palveluntarjoajilta) hankittujen komponenttien yhdistelyä. Palveluiden kerroksellinen luonne ilmenee kuvassa 7, jossa näkyvät myös palveluiden keskinäiset suhteet sekä rajapinnat asiakkaisiin.



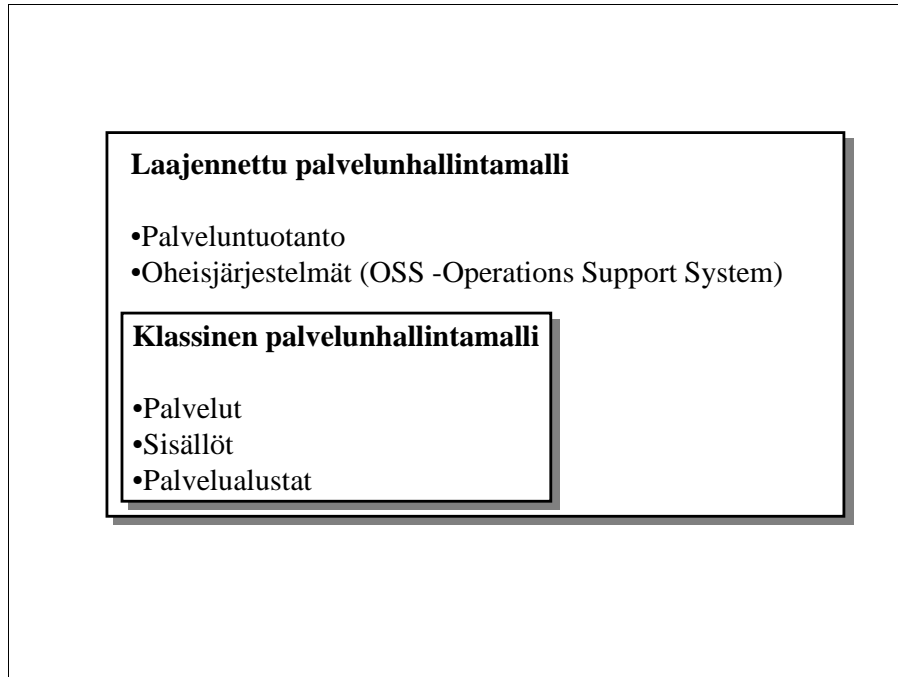
**Kuva 7:** Kerroksellinen palvelumalli /9/.

### 3.2 Mitä on palvelunhallinta?

Käsitteenä palvelunhallinta voidaan määritellä esimerkiksi tietyn loogisen toiminnallisuuden tai niiden sarjan hallinnaksi. Toiminnallisuuksista muodostuu asiakkaalle tai kohderyhmälle mielekäs kokonaisuus, joita he ovat valmiita käyttämään yleensä tiettyä taloudellista korvausta vastaan. Palvelu on abstrakti käsite suoritteesta tai tapahtumaketjusta, joka toteuttaa tietyn mielekkään asiakokonaisuuden tai tapahtuman. Palvelua ei yleensä voi mitata määrällisesti tai fyysisesti. Palvelut tuotetaan erilaisilla prosesseilla, joihin liittyy olennaisena osana logistiikka. Palvelunhallinta on noiden tapahtumaketjujen tai suoritteiden hallintaa niin, että hallinta on osittain automatisoitua. Erityisesti erilaiset rutiinit ja logistiikka pyritään automatisoimaan, jolloin kulutetaan mahdollisimman vähän palveluidentarjoajan resursseja kuten työtä, aikaa tai rahaa. Internet -palveluiden logistiikka eroaa perinteisten palveluiden logistiikasta siten, että palvelulla on olemassa jo jakelukanava (Internet). Tuota jakelukanavaa pitkin palvelu saadaan välitettyä asiakkaille, mutta sitä varten on toteutettava sekä asiakas- että palveluntarjoajan päässä tarvittava jakelutoiminnallisuus. Yleisesti Internetissä toimiessa pyritään siihen, että palvelun logistiset toiminnot olisivat pitkälle automatisoituja ja siihen, että asiakas pystyisi hoitamaan tarvittavat toiminnot itsepalveluperiaatteella. Esimerkiksi asiakas pystyy lataamaan haluamansa palvelun suoraan palveluntarjoajan palvelimelta itselleen, jolloin palveluntarjoaja tunnistaa miten asiakas on käyttänyt palvelua ja laskuttaa tämän perusteella.

Perinteisesti palvelunhallinta on kattanut jo olemassa olevien palveluiden, sisältöjen ja palvelualustojen ylläpidon ja monitoroinnin. Palvelunhallintaa voidaan kuitenkin käsitteenä laajentaa kattamaan myös palvelutuotanto ja muut siihen liittyvät oheisjärjestelmät. Järjestelmän käyttäjän näkökulmasta laajempi ja kattavampi tarkastelu on mielekkäämpi, koska keskitetty ja kokonaisvaltainen palvelunhallintajärjestelmä helpottaa kompleksisen palvelunhallintamekanismin

ymmärtämistä ja käyttöä. Kuvassa 8 on esitetty kokonaisvaltainen palvelunhallintamalli.



**Kuva 8:** Kokonaisvaltainen palvelunhallintamalli /10/.

### 3.3 Klassinen palvelunhallinta

Klassinen palvelunhallintamalli kattaa olemassa olevat palvelut, niiden sisällöt ja erilaiset palvelualustat. Se ei ota kantaa palveluntuotantoon tai palveluihin liittyviin oheisjärjestelmiin vaan kattaa palveluiden jakeluun liittyvät välttämättömät toiminnot. Käytännön toiminnan kannalta on tärkeää, että mallia laajennetaan kattamaan myös palveluntuotanto ja erilaiset oheisjärjestelmät kuten esimerkiksi asiakkaiden hallinta ja laskutus. Klassiseen eli perinteiseen palvelunhallintaan kuuluvia ominaisuuksia ovat: palvelualustojen ja palveluiden tilan monitorointi, virhetilanteista ilmoittaminen ja niistä toipuminen sekä tarvittaessa palvelun sisällön oikeellisuuden tarkistaminen. Palvelunhallinta

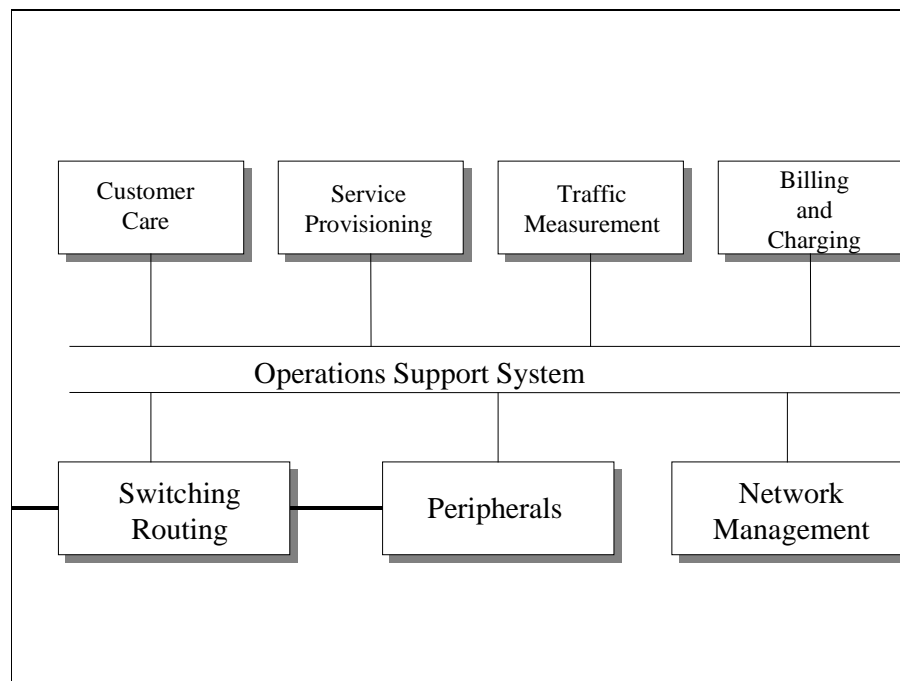
puhtaasti klassisen palvelunhallintamallin mukaisesti on mahdollista mutta yleensä siihen on liitetty myös mittaus- ja laskutusjärjestelmiä. Ne ovat välttämättömiä, mikäli palveluntarjoaja toimii liiketaloudellisin perustein ja haluaa laskuttaa asiakkaitaan kyseisten palveluiden käytöstä. Joissakin tapauksissa laskutus- ja mittausjärjestelmät voivat olla täysin palvelunhallinnan ulkopuolella olevia järjestelmiä, mutta tällöin laskutus on usein kiinteä hintaista eikä perustu varsinaiseen palveluiden käyttöön. Vaikka esimerkiksi Suomessa tällä hetkellä useimmat Internet -yhteyksiä tarjoavat ISP:t laskuttavat kiinteä hintaisia kuukausimaksuja on oletettavaa, että teknologian kehittyessä ja liikennemäärien kasvaessa laskutus tulee ainakin osittain perustumaan tiedonsiirtokapasiteetin käyttöön. Palvelunhallintaa onkin mielekkäämpää tarkastella kokonaisuutena, joka kattaa kaikki palveluiden ja niiden jakeluun liittyvät osa-alueet. Laajennettu palvelunhallintamalli käsittelee kyseistä kokonaisuutta.

### **3.4 Laajennettu palvelunhallintamalli**

Laajennettu palvelunhallintamalli ottaa huomioon myös palveluntuotantoon liittyvät näkökulmat ja systeemin tukijärjestelmät eli OSS:n (Operations Support Systems). OSS toteuttaa tarvittavat rajapintamuunnokset ja viestien välityksen tukijärjestelmien eri komponenttien välillä (Lähde: Professori Olli Martikainen, Telecom Visions -esitelmä /10/).

OSS:n avulla erilliset palvelunhallintaan liitetyt alijärjestelmät saadaan kommunikoidaan ja toimimaan yhdessä, jolloin esimerkiksi asiakaspalvelua voidaan automatisoida ja helpottaa niin, että tietystä asiakkaasta saadaan tarvittava tieto useasta eri alijärjestelmästä. OSS toimiikin eräänlaisena tukijärjestelmät yhteen kokoavana komponenttina ja se tarjoaa niille yhteisen kommunikointiväylän. OSS järjestelmä voidaan toteuttaa esimerkiksi CORBA ORB:n (Object Request Broker) avulla, sillä se mahdollistaa jo olemassa olevien ohjelmistojen mukaan tuomisen kokonaisjärjestelmään. OSS:n piiriin kuuluvat

seuraavat liiketoiminnan kannalta tärkeät prosessit: asiakkaiden hallinta (customer care), palvelun hinnoittelu (service provisioning), liikenteen mittaaminen (traffic measurement), laskutus ja veloitus (billing and charging). Lisäksi se kattaa myös seuraavat matalamman tason prosessit ja komponentit: Verkonhallinta (network management), oheislaitteistot (peripherals) sekä kytkentä ja reitittäminen (switching and routing). Kuvassa 9 on kuvattu OSS –järjestelmän eri komponentit.



**Kuva 9:** Operations Support System (OSS) /10/.

Internet –palvelut ovat jo tällä hetkellä monimuotoisia, eikä niille ole olemassa mitään standardia palveluiden toteuttamisesta tai niiden rajapinnoista. Näin ollen palvelunhallintaakaan ei ole juuri toteutettu. Java –ohjelmointikielen kehittyessä sen ympärille on alkanut tulla palvelukeskeisiä arkkitehtuuri- ja toimintamalleja. Esimerkiksi Java Beans ja Enterprise Java Beans (EJB) ovat palvelunäkökulmasta määriteltyjä komponenttimalleja. Näiden avulla toteutettuihin palveluihin voidaan tuoda mukaan ominaisuuksia, jotka mahdollistavat niiden hallinnan. Java –ohjelmointikielen mukaan tuleminen Internet –maailmaan onkin mahdollistanut



aivan uudenlaisten palveluiden kehittämisen. On luultavaa, että tulevaisuudessa Javan edelleen kehittyessä ja sen ympärillä tapahtuvien standardointiprosessien jatkuessa Java –pohjaisten Internet –palveluiden määrittelyyn, tuottamiseen ja hallintaan syntyy aivan uusia malleja ja standardeja. Vaikka Javalle onkin olemassa myös kilpailevia vaihtoehtoja, näyttää se olevan potentiaalinen ja laajasti käytetty apuväline Internet –palveluiden kehittämisessä. Vanhat HTML:n (Hyper Text Markup Language) ja CGI:n (Common Gateway Interface) avulla toteutetut järjestelmät käyvät nykyään vain ominaisuuksiltaan rajoittuneisiin ja suhteellisen vähän käytettyihin palveluihin.

Tällä hetkellä kokonaisvaltaisia Internet –ympäristöön suunniteltuja palvelunhallintajärjestelmiä ei vielä ole markkinoilla. Asiaan vaikuttaa voimakkaasti palveluiden, järjestelmien ja laitteistojen heterogeenisuus, joka vaikeuttaa palvelunhallinnan kehitystä. Ehkä myös Internetin nopeassa kehityksessä keskitytään vielä tällä hetkellä itse palveluiden kehittämiseen eikä niinkään niiden hallintaan, sillä palvelunhallinnan merkitystä ei ehkä vielä ole täysin ymmärretty suuressa mittakaavassa. On kuitenkin selvää, että palveluiden määrän kasvaessa ja niiden ominaisuuksien monipuolistuessa, tulevaisuudessa on välttämätöntä kiinnittää huomiota myös palvelunhallintaan.

Palvelunhallinnan piiriin kuuluvia tai sitä sivuavia tuotteita löytyy eri valmistajilta, mutta ne keskittyvät yleensä vain tiettyyn palvelunhallinnan osa-alueeseen kuten verkonhallintaan. Siihen tarkoitettuja SNMP -protokollaan (Simple Network Management Protocol) perustuvia ohjelmistoja on tarjolla useita mutta niitä ei voi integroida muihin palvelunhallintaan liittyviin järjestelmiin. SNMP –protokollaan pohjautuvat verkonhallintajärjestelmät toimivat laitetasolla mutta eivät juuri tarjoa integrointimahdollisuuksia palvelutasolla. Erilaisten valmiiden palvelunhallintaan liittyvien järjestelmien liittäminen kokonaisjärjestelmään on perusedellytys, koska hallittava alue on erittäin laaja ja niiden toteuttaminen erikseen tuotteeseen vaatii suunnattomasti resursseja. Lisäksi asiakkailta voi jo olla käytössään järjestelmiä, joita he eivät ole valmiita

korvaamaan uusilla. Yhtenäisten standardien ja järjestelmien välisten rajapintojen puuttuminen kyseiseltä alueelta on vaikeuttanut hallintajärjestelmien kehitystä.

Kuitenkin erilaisten järjestelmien ja arkkitehtuurien yhteensovitusta varten on olemassa valmiita toimivia ratkaisumalleja esimerkiksi OMG:n CORBA - teknologia. Sen ominaisuudet ovat hyödyllisiä myös palveluissa, sillä niihin integroidaan yhä enemmän toimintoja jo olemassa olevista järjestelmistä. Lisäksi käyttöjärjestelmäriippumaton ohjelmointikieli Java on tuonut mukanaan mahdollisuuden siirtää yksi ja sama ohjelmisto käyttöjärjestelmästä toiseen. Nykyään OMG:n CORBA –arkkitehtuuria ja Javaa (Sun Microsystems, Inc.) kehitetään tiiviissä yhteistyössä.

### **3.5 Palvelut Internetissä**

Internet -palveluihin ja niiden ominaisuuksiin vaikuttavat erityisesti palveluiden jakelun toteutusmallit ja siirtoteiden kaistanleveys. Tällä hetkellä Internet – palveluilla tarkoitetaan pääasiassa peruspalveluja mutta tulevaisuudessa palvelut monipuolistuvat ominaisuuksiltaan ja palveluiden tarjonta kasvaa. Tällöin palvelunhallinnan merkitys kasvaa.

#### **3.5.1 Palvelunjakelu ja siirtotiet**

Tällä hetkellä Internetissä tarjotut palvelut rajoittuvat käytössä olevan kaistanleveyden takia pääasiassa HTML –pohjaisiin sivustoihin. Nekin voivat sisältää niin suuria data määriä (tekstiä, kuvia, liikkuvaa kuvaa ja ääntä), että modeemiyhteyksillä niiden lataaminen voi olla erittäin hidasta. Koska suurin osa loppukäyttäjistä kuitenkin vielä käyttää Internet –palveluita suhteellisen hitailla yhteyksillä, on tarjotut palvelut pyritty optimoimaan tiedonsiirtonopeuden mukaan. Nykyään loppukäyttäjälle tuleva kaistanleveys saadaan tarvittaessa

moninkertaistettua modeemiyhteyksien tarjoamaan kaistanleveyteen verrattuna. Se voidaan tehdä esimerkiksi erilaisilla DSL -tekniikoilla (Digital Subscriber Line) tai kaapelimodeemeilla. Kyseisten teknologioiden etuna on pidettävä sitä, että ne hyödyntävät jo olemassa olevaa tiedonsiirtoinfrastruktuuria kuten puhelinlinjaa tai kaupunkialueilla kaapelitelevisioverkkoa. Lisäksi lähitulevaisuudessa kolmannen sukupolven matkapuhelinjärjestelmät (UMTS, Universal Mobile Telecommunications System) tarjoavat nopeaa tiedonsiirtokapasiteettia (Mbit/s) ja edelleen neljännen sukupolven matkapuhelinjärjestelmät tulevat moninkertaistamaan kapasiteetin. Uusien teknologioiden yleistyessä ja käyttökustannusten halventuessa niitä tullaan myös ottamaan yhä suuremmassa mittakaavassa käyttöön.

On luultavaa, että erilaiset nopeat optisiin ratkaisuihin perustuvat teknologiat tulevat käyttöön runkoyhteyksissä, sillä ne tarjoavat kilpailukykyisen ratkaisun kun tarvitaan erittäin suurta kapasiteettia. Ne eivät kalleutensa vuoksi luultavasti saavuta suosiota loppukäyttäjien tiedonsiirtoteknologiana. Kapasiteetin tarve Internetin runkoyhteyksissä lisääntyy kasvun ja uusien palveluiden myötä.

### **3.5.2 Peruspalvelut**

Puhuttaessa Internet –palveluista tarkoitetaan usein jo vakiintuneita peruspalveluja, joita Internet -yhteyksiä tarjoavat ISP:it tuottavat asiakkailleen. Näitä peruspalveluja ovat WWW (World Wide Web), sähköposti, tiedostonsiirto eli FTP (File Transfer Protocol) ja uutisryhmäpalvelu. WWW –liittymä on palvelu, jonka kautta asiakas saa yhdistettyä käyttämänsä selainohjelman Internetiin. Tällä hetkellä palvelunhallinnasta puhuttaessa tarkoitetaan usein em. palveluiden tai niihin läheisesti liittyvien palveluiden hallintaa (esim. DNS eli Domain Name Service). Peruspalveluiden hallintaa varten löytyy sekä kaupallisia että ilmaisia tuotteita, jotka yleensä keskittyvät vain tiettyyn palveluun.

### 3.5.3 Palvelut tulevaisuudessa

Peruspalvelut säilyttävät asemansa myös tulevaisuudessa, mutta niiden rinnalle kehittyy uusia monipuolisempia palveluja. Uudet palvelumuodot ovat usein sisältökeskeisiä ja ne vaativat suurempia tiedonsiirtonopeuksia kuin peruspalvelut. Uusien palveluiden käytön yleistymisen ja niiden palvelunlaadun takaamiseksi on pystyttävä kasvattamaan loppukäyttäjien tiedonsiirtonopeuksia huomattavasti nykyisestä. Taulukkoon 1 on listattu erilaisia palvelukategorioita ja esimerkkejä tulevaisuudessa niihin kuuluvista uusista palvelumuodoista (Lähde: Professori Olli Martikainen, Telecom Visions -esitelmä, 1999).

**Taulukko 1:** Palvelut Internetissä /10/.

<b>TCP/IP –pohjaiset palvelut Internetissä</b>	<b>Esimerkkejä</b>
1. Henkilökohtainen tiedonsiirto	IP –puhelu ja videoneuvottelu Matkapuhelin ja viestintä
2. Verkkopalvelut	Elektroninen kaupankäynti Julkaisutoiminta ja elektroninen media
3. Audiovisuaaliset palvelut	Interaktiivinen multimedia verkossa (pelit, virtuaali- todellisuus, Online – tapahtumat) Broadcast –kanavat verkossa (WebCATV, MPEG2)

### **3.5.4 Palvelunhallinnan ongelma**

Palvelunhallinnan ongelmana on Internet -palveluiden heterogeenisuus eli monimuotoisuus, sillä niille ei ole määritelty yksikäsitteistä palvelumallia. Internetin luonteesta johtuen siellä voi olla useita eri tyyppisiä palveluita, jotka toteutukseltaan ja arkkitehtuuriltaan poikkeavat huomattavasti toisistaan. Pahimmassa tapauksessa jokaiselle uudelle palvelulle syntyy uusi hallintaohjelmisto, joka toimii erillään muiden palveluiden hallintajärjestelmistä.

Internet tulee luultavasti tulevaisuudessakin kehittymään useita eri teknologioita kattavaksi ja tukevaksi ympäristöksi. Näiden heterogeenisten teknologioiden avulla luotujen palveluiden välille on vaikeaa kehittää kattavaa palvelunhallintaa ilman yhteisiä standardeja. Esimerkiksi yhtenäisten kommunikointirajapintojen määrittely on avainasia palvelunhallinnan kannalta. Jotta erityyppisten palveluiden hallintaa varten saadaan luotua toimivat standardit, tarvitaan luultavasti jonkinlaista palveluiden luokittelua. Ne saatetaan esimerkiksi jakaa luokkiin, joiden perusteella kyseisille palveluille on mielekästä määritellä yhtenäiset rajapinnat hallintaa varten. Eräs vaihtoehto luokkiin jakamiselle voi olla esimerkiksi taulukon 1 mukainen luokittelu.

### **3.5.5 Komponentit ja suunnittelumallit**

Nopeasti muuttuvassa Internetissä palvelut on kehitettävä sekä tuotettava nopeasti ja lisäksi ne on saatava potentiaalisten asiakkaiden ulottuville. Palveluiden kehittämiseen käyvät samat tehokkuutta lisäävät ratkaisumallit kuten muuhunkin ohjelmistonkehittämiseen. On tarpeellista luoda uudelleen käytettäviä komponentteja, joita yhdistelemällä ja parametroimalla palvelu saadaan luotua nopeasti ja resursseja turhaan kuluttamatta. Komponenttimallin rinnalla voidaan käyttää suunnittelumalleja (Design Patterns, /11/), jotka ovat yleisesti hyväksi

havaittuja ja testattuja toimintamalleja. Ne ratkaisevat tietyn rajatun ongelman ja tarjoavat siihen yksiselitteisen ratkaisumallin. Opittuaan käyttämään niitä suunnittelija saa rutiinin eräiden yleisten ongelmien ratkaisemiseen ja näin ollen välttään osasta virheistä, joita voisi syntyä mikäli suunnittelija yrittäisi itse ratkaista loogisen ongelman ilman valmiita ratkaisumalleja. Suunnittelumallit voivat toimia esim. koodausstandardin osana ja näin koodista saadaan selkeämpää ja ymmärrettävämpää. Tämä tuo etuja varsinkin, jos suunnittelijat vaihtuvat tai heitä on useita.

### **3.6 Perinteisten lisäarvopalveluiden hallinta**

Perinteisillä lisäarvopalveilla tarkoitetaan tässä yhteydessä palveluja, joita asiakas on voinut käyttää perinteisen julkisen kytkävän puhelinverkon välityksellä. Perinteisiä lisäarvopalveluja on kehitetty lukuisia ja puhelinverkon digitalisoituminen on tuonut mukanaan useita erilaisia lisäarvopalveluja, kuten puhelinvastaajat, puhelunsiirrot ja lukuisat palvelunumerot. Lisäarvopalveluiden hallinta on kuitenkin huomattavasti yksinkertaisempaa tässä ympäristössä verrattuna avoimeen Internet –tiedonsiirtoverkkoon. Puhelinverkko on arkkitehtuuriltaan lähes suljettu ja sillä on erillinen signaalintijärjestelmä, joten se on suhteellisen turvallinen.

Asiakkaan autentikointi eli tunnistus ja laskutus voidaan tehdä kiinteässä puhelinverkossa tilaajanumeron perusteella, mutta ongelmana myös siellä on se, että kyseisestä numerosta soittava asiakas ei välttämättä ole liittymän omistaja. Asiakkaiden hallinta on kuitenkin yksinkertaisempaa kuin Internetissä ja asiakas voidaan yleensä yhdistää tiettyyn tilaaja –eli puhelinnumeroon. Palvelut toteutetaan älyverkko –teknologialla, jolla perinteiseen kiinteään puhelinverkkoon saadaan tuotua älykkyyttä sisältäviä palveluja. Myös palveluiden hallinta perustuu älyverkkoteknologiaan, jolloin se on luonteeltaan lähempänä tietotekniikkaa kuin tietoliikennetekniikkaa. Perinteisen puhelinverkon lisäarvopalvelut ovat

vakiintuneita ja siirtoteknologiat sekä laitteet ovat pitkälle kehittyneitä ja toimintavarmoja. Palvelunhallinta tapahtuu OSS –järjestelmän kautta, joka mahdollistaa esimerkiksi laskutustiedon välityksen laitteistoilta laskutusjärjestelmään.

On kuitenkin luultavaa, että julkinen kytkevä puhelinverkko ja Internet –verkko tulevat yhdistymään ainakin osittain tulevaisuudessa. Jo tällä hetkellä on mahdollista soittaa ns. Internet –puheluja ympäri maapalloa suhteellisen edullisesti. Internet ei vielä kuitenkaan pysty tarjoamaan tarpeellista palvelutasoa suuremmissa mittakaavassa tapahtuvalle puhelinliikenteelle.

Tiedonsiirtonopeuksien kasvaessa ja infrastruktuurin kehittyessä on täysin mahdollista, että puhelinliikenne siirtyy ainakin osittain tietoverkkoihin.

Matkapuhelin puolella puhelinliikenteen palvelut ovat jo nykyään lähellä Internet –palveluita. Pääasiassa palvelut ovat tällä hetkellä tekstipohjaisia ja perustuvat SMS -palveluun (Short Message Service), mutta uuden WAP –protokollan (Wireless Application Protocol) markkinoille tulo tuo myös graafisia palveluja uusiin matkapuhelimiin. WAP:in avulla matkapuhelinpalveluiden tuottamiseen ja kehittämiseen saadaan tuotua Internetin kaltaista lisäarvoa. Internet -pohjaisia palveluja on jo nyt saatavissa kehittyneimpiin matkapuhelinmalleihin ja suomalaiset teleoperaattorit ovat edellä kävijöitä myös tällä alueella.

### **3.7 Internet –palveluiden toteutusteknologiat**

Internet –palveluita voidaan toteuttaa useilla eri tavoilla. Puhtaasti Socket –rajapinnan avulla toteutettuja palveluja on vähän mutta kaikki muut teknologiat käyttävät Socket -rajapintaa hyväkseen. HTTP / CGI –pohjaiset (Hypertext Transfer Protocol, Common Gateway Interface) palvelut ovat käyneet suorituskyvyltään riittämättömiksi, vaikka niitä vielä esiintyykin Internetissä. Kilpailukykyisiä palveluiden toteutusteknologioita ovat erilaiset Java -teknologiat, DCOM –malli (Distributed Component Object Model) ja CORBA.

### 3.7.1 Socket –rajapinta ja CGI (Common Gateway Interface)

Socket –rajapinta on ohjelmointi rajapinta (API, Application Programming Interface), joka on toteutettu suoraan TCP ja UDP kerrosten päälle TCP/IP -protokollapinossa. Internet -palveluja on mahdollista toteuttaa puhtaasti tämän rajapinnan päälle. Koska kyseinen API on matalan tason rajapinta, se on tehokas ja nopea tapa kommunikoida Internetissä. Laajojen palveluiden toteutus on kuitenkin työlästä pelkästään Socket –rajapinnan avulla, koska se ei tarjoa mitään sovellustason protokollaa eikä muita toimintoja vaan ne on kaikki toteutettava itse. Rajapinta tarjoaa ainoastaan keinot luoda yhteys tiettyyn IP –osoitteeseen tiettyyn porttiin ja mahdollisuuden joko kirjoittaa tai lukea tavuja. Käytännössä Internet -palveluita ei nykyään enää toteuteta puhtaasti Socket -rajapinnan avulla, vaan muut kehittyneemmät teknologiat käyttävät hyväkseen sitä kommunikoinnissaan.

Vielä muutama vuosi sitten toteutetut palvelut olivat suurimmaksi osaksi HTTP (Hypertext Transfer Protocol) ja CGI –pohjaisia. Kyseisten palveluiden hallinta oli täysin käsityötä ja palvelut olivat toiminnaltaan hitaita ja toteutukseltaan raskaita. CGI (Common Gateway Interface) on standardi, jolla palvelin välittää HTTP -pyyntöjä ja niiden parametrejä asiakasohjelmilta sovellusohjelmille. Sovellusohjelma suorittaa pyynnön ja palauttaa tulokset HTML -muodossa (Hypertext Markup Language) palvelimelle (web server) käyttäen CGI –protokollaa. Tämän jälkeen palvelin palauttaa tulokset asiakasohjelmalle. CGI –protokollan suurimpina puutteina voidaan pitää sen hitautta ja tila käsitteen puuttumista. Vaikka nykyäänkin vielä osa palveluista on toteutettu CGI:llä, on uusien palveluiden toteuttamiseen ja implementointiin muita toimivampia ratkaisuja kuten esimerkiksi Microsoftin ASP –sivut (Active Server Pages) tai erilaiset Java pohjaiset ratkaisut. ASP –sivut ovat HTML –sivuja, joihin on sijoitettu pieniä upotettuja ohjelmia (script). Ohjelmat prosessoidaan Microsoft IIS palvelimessa, jossa niistä generoidaan staattinen HTML –sivu ennekuin se lähetetään selaimelle. ASP:n avulla voidaan räätälöidä asiakkaan selaimelle



lähetettävä sivu, jolloin esim. saadun syötteen perustella voidaan tietokannoista hakea tarvittavat tiedot palvelin päässä ja muokata lähetettävää HTML -sivua niiden antaman tiedon perusteella. ASP –sivuja voidaankin toimintaperiaatteeltaan verrata CGI –teknologiaan ja Javan Servlet -ominaisuuksiin, sillä kaikissa em. vaihtoehtoissa lähetettävä HTML –sivu generoidaan palvelimessa. Etuna lähestymistavassa on se, että asiakkaan selaimen ei tarvitse tukea muuta kuin HTML –standardia.

### **3.7.2 DCOM (Distributed Component Object Model)**

Nykyään Internet palveluja kehitetään yhä enemmän hajautetun oliomallin pohjalta. OMG:n standardoiman CORBA -arkkitehtuurin rinnalla on kehittynyt myös kilpaileva olioarkkitehtuuri DCOM. Ehkä parhaiten tunnettuja DCOM -olioita ovat ActiveX -komponentit. Ne ovat DCOM -arkkitehtuurin päälle rakennettuja ohjelmistokomponentteja, jotka on suunniteltu toimimaan verkossa ja ne mahdollistavat koodin dynaamisen lataamisen selaimen. ActiveX -komponentit voivat olla kontrolleja (ActiveX controls), dokumentteja (ActiveX documents) tai pieniä suoritettavia ohjelmia eli ns. skriptejä (ActiveX scripts). Esimerkiksi ActiveX -kontrollit ovat ladattavia interaktiivisia ohjelmistokomponentteja, joita voidaan sijoittaa HTML –sivuille. Niiden avulla on mahdollista luoda erilaisia interaktiivisia palveluja ja tarjota niitä asiakkaille Internetin välityksellä. ActiveX -komponentit toimivat kuitenkin vain Windows käyttöjärjestelmässä.

DCOM on Microsoftin kehittämä de facto aseman saanut arkkitehtuuri ja näin ollen laajalle levinnyt. Koska se on mukana Windows NT, Windows 98 ja Windows 2000 käyttöjärjestelmissä, se on edelleen varteen otettava kilpailija CORBA -arkkitehtuurille.

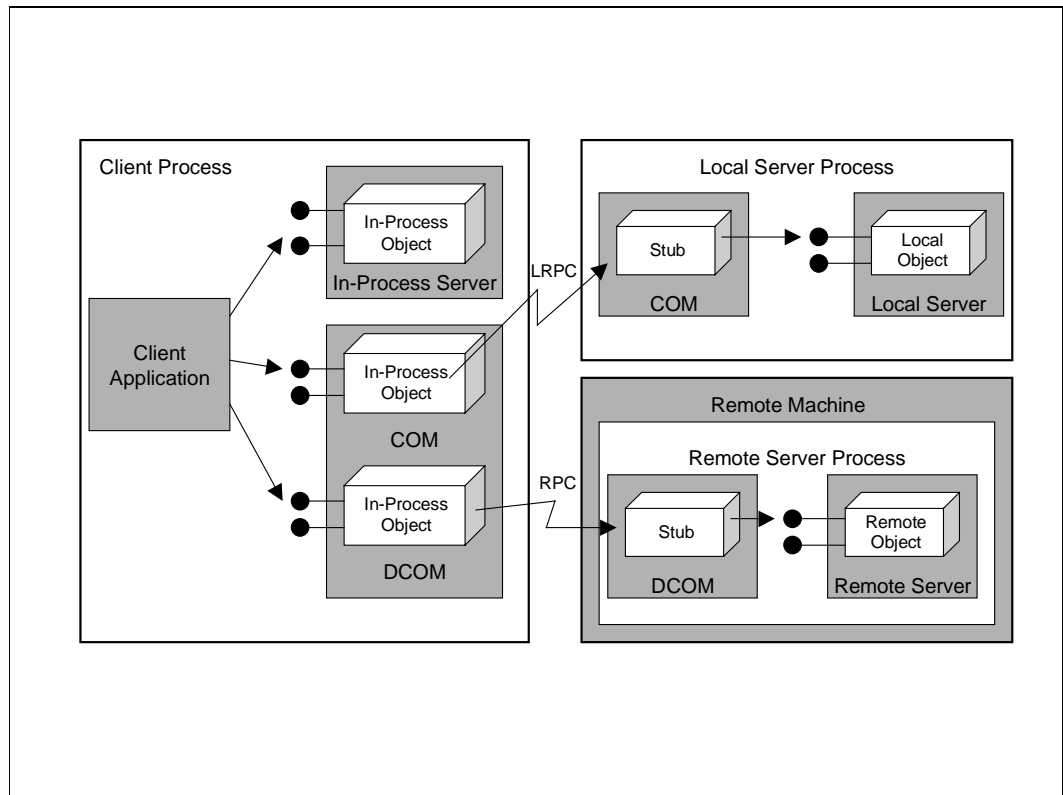
DCOM ja CORBA ovat osittain samankaltaisia ja sisältävät samoja piirteitä ja ominaisuuksia. Ne kuitenkin eroavat toisistaan tietyissä asioissa huomattavasti. DCOM on kehitetty puhtaasti Windows maailmaan kun taas CORBA on käyttöjärjestelmä riippumaton. DCOM erottaa olioiden kommunikointirajapinnat niiden implementaatiosta ja rajapinnat on määriteltävä käyttäen rajapintojen määrittelykieltä (IDL, Interface Definition Language). On huomattava, että kyseessä on Microsoftin kehittämä rajapintojen kuvauskieli ja se ei ole sama kuin CORBA:n IDL. Kyseisiä rajapintamäärittelyä käytetään luomaan rungot asiakas- ja palvelinprosesseille. DCOM -oliot eivät oikeastaan ole perinteisen oliomallin mukaisia olioita. DCOM -olioilla ei ole tilaa ja niistä ei voi luoda uniikkeja instansseja eli olioita joilla on yksilöllinen oliotunniste. DCOM -oliot ovat siis osoittimia tilattomiin rajapintoihin. DCOM -rajapinta on oikeastaan ryhmä toisiinsa liittyviä funktioita ja niitä käyttäville asiakasohjelmille välitetään osoitin noihin funktioihin. Osoitin ei ota kantaa olioiden tilaan, eli DCOM -asiakasohjelma ei voi ottaa yhteyttä uudestaan samaan olioinstanssiin samassa tilassa kuin aikaisemmin. Se voi vain ottaa yhteyden saman luokan rajapintaosoittimeen. Tästä johtuen DCOM -oliot eivät säilytä tilaansa eri yhteyksien välillä, mikä voi aiheuttaa ongelmia esim. ympäristöissä joissa on katkeilevia yhteyksiä.

DCOM -luokalla on yksilöllinen tunniste CLSID (Class Identifier) ja se voi toteuttaa useita eri rajapintoja. Asiakasohjelmat pyytävät palvelinprosessilta oliota tietyllä CLSID:llä. DCOM ajaa palvelinohjelman ja pyytää sitä luomaan tuon kyseisen luokan olion. Kun olio on luotu asiakasohjelmalle palautetaan sen rajapinnan osoitin. Sana palvelin tarkoittaaakin DCOM -maailmassa palvelevaa agenttia, eikä niinkään palvelinta sanan vakiintuneessa merkityksessä.

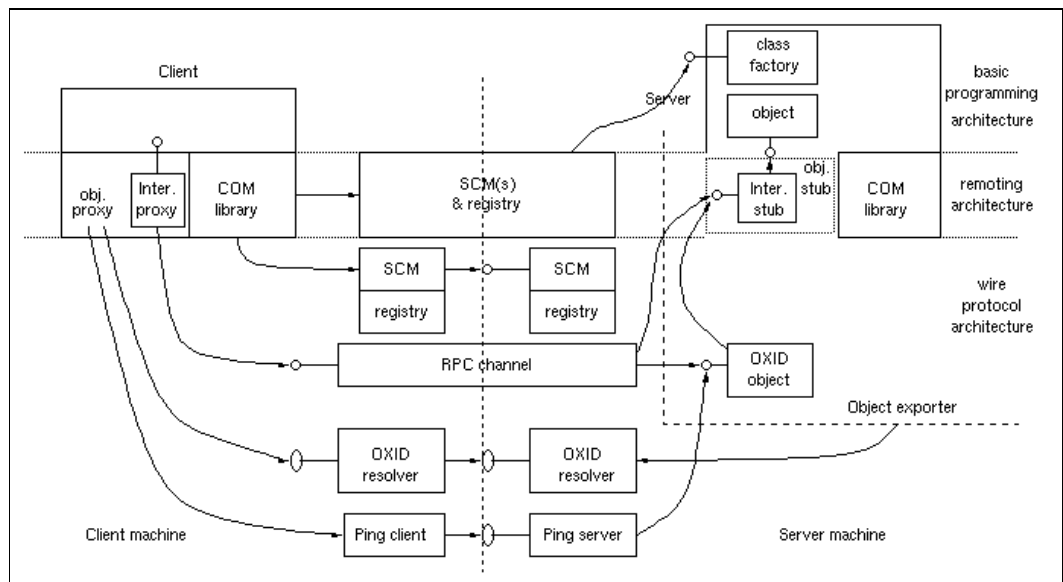
DCOM -palvelin voi toimia kolmella tavalla:

- Prosessin sisäinen palvelin (in-process server) ajetaan samassa prosessitilassa kuin asiakasohjelmakin. Windows ympäristössä ne on toteutettu DLL – kirjastoina (Dynamic Link Library), jotka ladataan suoraan asiakasprosessiin.
- Paikalliset palvelevat agentit eli palvelut suoritetaan erillisinä prosesseina mutta samassa koneessa. Asiakasohjelmat käyttävät LRPC (Lightweight Remote Procedure Call) mekanismia kommunikointiin paikallisen palvelun kanssa. Paikalliset palvelut ovat itsenäisiä ajettavia tiedostoja.
- Etäpalvelimet ajetaan omana prosessina toisessa koneessa (mahdollisesti toisessa käyttöjärjestelmässä). Asiakasohjelmat käyttävät RPC –mekanismia (Remote Procedure Call) kommunikoidakseen näiden kanssa.

Teoriassa DCOM mahdollistaa asiakasohjelmien läpinäkyvän kommunikoinnin palvelinolioiden kanssa riippumatta siitä missä ne sijaitsevat. Käytännössä kuitenkin implementaatioissa joutuu väistämättä käyttämään prosessiin liittyvää koodia, koska osoittimen on oltava prosessiin liittyvä. Eli toiminnallisuuden toteuttamiseen tarvitaan yksityiskohtaista tietoa myös vastapään toiminnasta ja määrittelyistä. Kuvissa 10 ja 11 on kuvattu DCOM:n asiakas- ja palvelinmallin mukaiset rajapinnat sekä arkkitehtuuri.



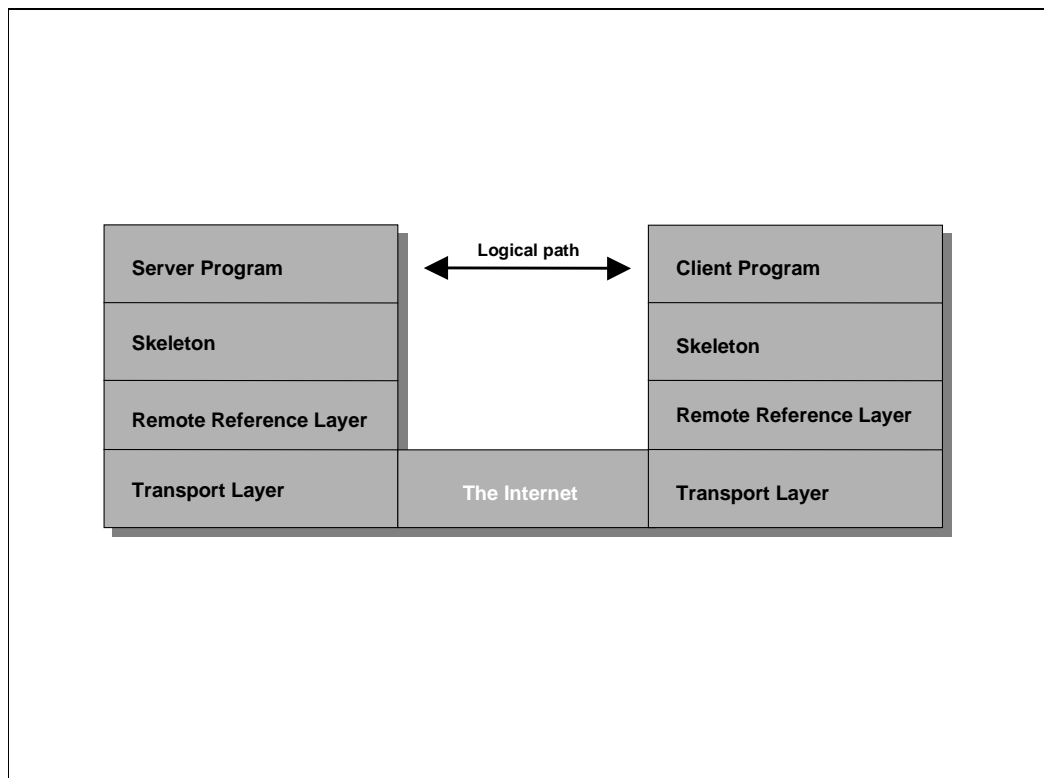
**Kuva 10:** DCOM:n asiakas- ja palvelinmallin rajapinnat /12/.



**Kuva 11:** DCOM -arkkitehtuuri /13/.

### 3.7.3 Java –pohjaiset teknologiat

RMI (Remote Method Invocation) antaa mahdollisuuden toteuttaa hajautettuja Java –pohjaisia sovelluksia eli Java –ohjelma voi tehdä metodikutsuja eri virtuaali- ja isäntäkoneissa sijaitseviin ohjelmiin. RMI –pohjainen kutsu voidaan suorittaa, kun kutsuttavan olion olioviite on tiedossa. Kyseinen olioviite voidaan hankkia joko RMI:n itseohjautuvan nimipalvelun (bootstrap-naming service) kautta tai vastaanottamalla kyseinen olioviite argumenttina tai palautusarvona. RMI –pohjainen asiakasohjelma voi kutsua palvelinpuolen ohjelmaa, joka taas voi toimia asiakaspuolena jollekin toiselle RMI –pohjaiselle ohjelmalle. RMI käyttää olioiden sarjamuotoon muuntamista (object serialisation) parametrien välityksen keinona ja se tekee sen hävittämättä tietoa. Näin ollen se tukee aitoa oliosuuntautunutta polymorfismia. Kuvassa 12 on kuvattu RMI –kerromalli.



**Kuva 12:** RMI -kerromalli /14/.

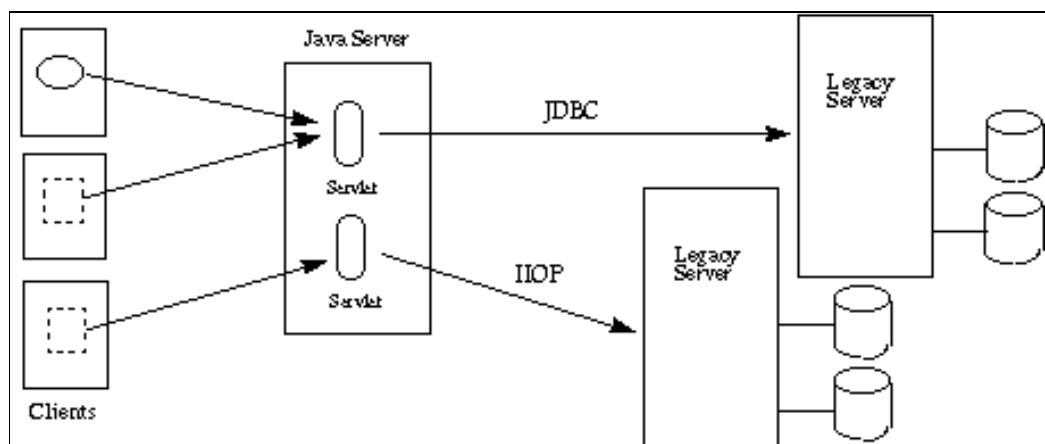
RMI -tekniikan edut ovat pääasiassa samoja kuin Javan yleisesti mukanaan tuomat edutkin. Kuten muutkin Javalla toteutetut ohjelmat, myös RMI –pohjaiset toteutukset ovat käyttöjärjestelmäriippumattomia ja helposti siirrettäviä. Ohjelmoijan kannalta etäoliot ja niiden metodit ovat samanlaisia käyttöä kuin paikallisetkin oliotkin ja metodit eli niiden implementaatio on piilotettu ohjelmoijalta. Toteutettujen palveluiden hallintaan RMI ei ota kantaa eikä tuo mukanaan mitään hallintaan liittyviä komponentteja eli palveluiden hallinta on toteutettava erikseen.

Servlet:it ovat alustasta riippumattomia Java -komponentteja, joiden avulla voidaan toteuttaa sovelluksia ja palveluita Internet -ympäristöön. Niiden avulla on mahdollista laajentaa verkkopalvelimien toimintaa. Servlet -komponentteja ei ole sidottu mihinkään tiettyyn protokollaan mutta yleisimmin niitä käytetään HTTP –protokollan ja HTTP -palvelimien kanssa. Servlet -komponentteja voidaan ladata toimivaan palvelimeen laajentaen näin ollen sen toiminnallisuutta ja ominaisuuksia. Servlet -komponentin tavukoodi (bytecode) voidaan lukea paikalliselta levyllä tai verkosta ja pääasiassa niitä käytetään CGI –ohjelmien tapaan tuottamaan dynaamisesti dataa asiakasohjelmia varten. Servlet -komponenteilla on kuitenkin muutamia ominaisuuksia, joilla ne eroavat CGI –ohjelmista.

- Servlet voi jatkaa toimintaansa vaikka se olisikin jo prosessoimassa pyynnön (request). Näin se on valmiina prosessoimaan seuraavan pyynnön ilman, että sitä tarvitsee käynnistää uudelleen kuten CGI –ohjelmien tapauksessa. Lisäksi Servlet voi käyttää hyväkseen Java -säikeitä (thread) hoitaakseen tehokkaasti yhtäaikaista pyyntöä.
- Servlet voi kommunikoida asiakkaana toimivan Java Applet:in kanssa interaktiivisesti. CGI –ohjelma kykenee vain vastaanottamaan pyynnön ja lähettämään vastauksen, jonka jälkeen yhteys katkaistaan. Servlet:it ja Applet:it voivat keskustella keskenään ja tehdä useita datan siirtoja.

- Servlet voi olla peräisin asiakaspään ohjelmasta. Servlet -komponentteja ajetaan turvallisessa rajatussa ympäristössä (sandbox) ja näin ollen palvelimen ei tarvitse välittää vihamielisistä Servlet -komponenteista.
- Servlet:it ovat askel kohti agenteja. Sama Servlet voidaan esimerkiksi ladata useaan palvelimeen ja se voi toteuttaa saman toiminnallisuuden niissä vuoronperään. Javan tietoturvaominaisuudet takaavat sen ettei palvelimen tarvitse pelätä Servlet:in aiheuttavan vahinkoa. Javan siirrettävyys taas takaa sen, että Servlet:it eivät välitä palvelimen alla olevasta käyttöjärjestelmästä. Edellä mainittujen ominaisuuksien avulla on mahdollista rakentaa älykkäitä agenteja. Servlet:it eivät kuitenkaan ole täysin agenttimallin mukaisia komponentteja, sillä ne on aina jonkin muun osapuolen käynnistettävä ja siirrettävä haluttuun paikkaan. Aidot agentit liikkuvat ja toimivat itsenäisesti.

Servlet -komponenttien avulla voidaan toteuttaa palveluja, jotka hyödyntävät muita valmiita järjestelmiä kuten tietokantoja. Näin on mahdollista tuottaa monipuolisia palveluja Internetiin ja erilaisia sovelluskohteita löytyy monia. Kuvassa 13 on kuvattu kuinka Servlet -komponentteja voidaan käyttää apuna esim. toteutettaessa WWW –pohjaisia käyttöliittymiä erilaisille suljetuille järjestelmille.



**Kuva 13:** Esimerkki Servlet -komponenttien käytöstä /15/.

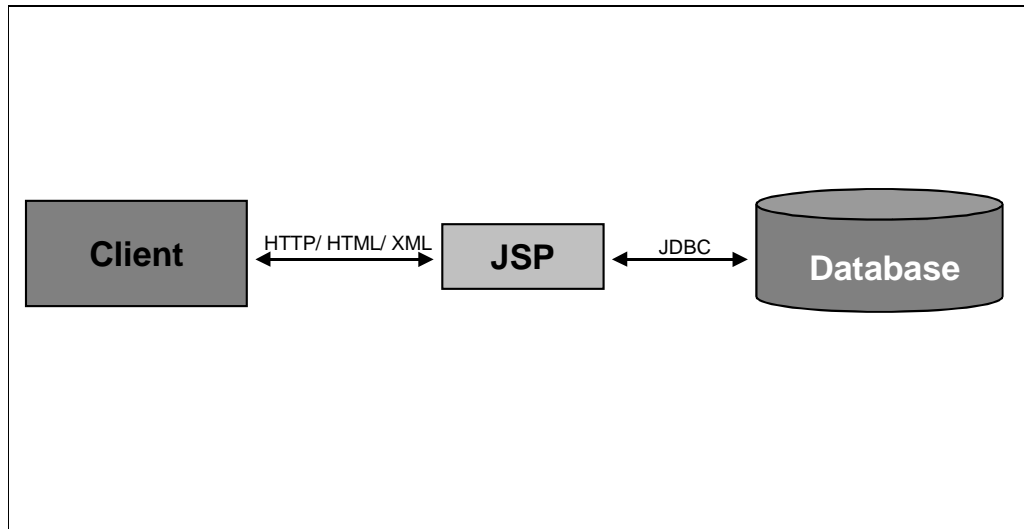
JSP:t (JavaServer Pages) eli Java -palvelinsivut ovat komponentteja, joiden avulla voidaan tuottaa dynaamisesti generoituja HTML- tai XML –sivuja (Extensible Markup Language). JSP -sivuja voidaan verrata Microsoftin ASP -sivuihin (Active Server Pages), mutta ne ovat järjestelmäriippumattomia. JSP –sivut käännetään Servleteiksi kun niitä kutsutaan, joten ne ovat sen suora laajennus. Alkuperäisissä Servleteissä tietyt staattiset elementit, kuten teksti, on sijoitettu suoraan liiketoimintalogiikan (business logic) sekaan, joten niiden muuttaminen vaatii uudelleen kääntämisen ja itse elementtien löytäminen voi olla jo joissakin tapauksissa hankalaa. JSP –sivut toimivat päinvastaisella tavalla ja sovelluslogiikka upotetaan standardeihin HTML- tai XML –malleihin (templates). Sivujen päivittäjät voivat muuttaa ongelmitta sivujen ulkonäköä ja rakennetta vaikuttamatta itse sovelluslogiikkaan. JSP –varastoelementti (JSP container) kääntää automaattisesti sivut Servlet -komponenteiksi kun niitä kutsutaan, joten uusien sivujen tekeminen ja ylläpitäminen on yksinkertaisempaa kuin pelkällä Servlet -teknologialla tehdyissä ratkaisuisissa. Varsinainen sovelluslogiikka voi olla seuraavanlaisissa muodoissa HTML- tai XML –sivuilla:

- Lyhyissä Java -ohjelmissa (Java scriptlets), jotka suorittavat joitakin yksinkertaisia toimintoja.
- JSP –leimoina (tag), jotka kutsuvat ulkopuolisia funktioita, kuten esimerkiksi JavaBean –komponentteja.
- Mukautettuina JSP -leimoina (customised tag). Ohjelmoijat voivat tehdä itse leimakirjastoja, joita sivujen päivittäjät voivat käyttää.

JSP:ssä on otettu huomioon joitakin palveluiden hallintaan liittyviä tekijöitä. Sovelluslogiikan erottaminen itse sivun sisällöstä ja mahdollisuus tuottaa dataa sivuille dynaamisesti tuo mukanaan mahdollisuuden toteuttaa mitä erilaisimpia palveluja Internet -ympäristöön. JSP –teknologia on tällä hetkellä vielä suhteellisen uutta ja on oletettavaa, että sen käyttö palveluiden toteutusteknologiana tulee lisääntymään. Kuvan 14 esimerkitapauksessa JSP:tä on käytetty ns. välikerroksen (middle-tier) implementoinnissa. Se hoitaa liittynän



tietokantaan JDBC:n (Java Database Connectivity) avulla ja välittää halutun tiedon asiakaspäälle.



**Kuva 14:**Esimerkki JSP:n käytöstä /16/.

Tällä hetkellä kehittyneiden Internet -palveluiden toteutukseen ja hallintaan on olemassa avoin Java -pohjainen EJB –komponenttiarkkitehtuuri (Enterprise JavaBeans). Se tarjoaa standardoidun avoimen arkkitehtuurin ja rajapintamallin Internet –palveluiden kehittämistä, jakelua ja hallintaa varten. EJB -teknologia on looginen jatko aikaisemmin kehitetylle JavaBeans –komponenttimallille, joka tarjosi asiakasohjelmistoille uudelleen käytettäviä komponentteja. EJB ei itse asiassa suoranaisesti liity JavaBeans –komponentti malliin vaikka sen nimestä voisi ehkä niin päätelläkin. EJB –komponenttiarkkitehtuurin pohjana oli tarve saada aikaan avoin ohjelmistoarkkitehtuuri Internetiin, jonka avulla voidaan helposti tuottaa skaalautuvia, turvallisia, laitealustasta riippumattomia sovelluksia ja palveluja uudelleenkäytettävänä palvelinpuolen komponentteina. EJB –komponenttien tehtävänä on loogisesti yhdistää sovelluksessa käsiteltävä tieto käsittelyrutiineihin eli liiketoimintalogiikkaan. EJB –arkkitehtuuri on kehittynyt hajautetuksi monikerroksiseksi (multi-tier) palvelinkeskeiseksi komponenttimalliksi. Tämä tuo mukanaan mahdollisuuden keskittyä palveluiden kehityksessä ja hallinnassa oleelliseen, eli palvelu- ja liikeidean toteuttamiseen.

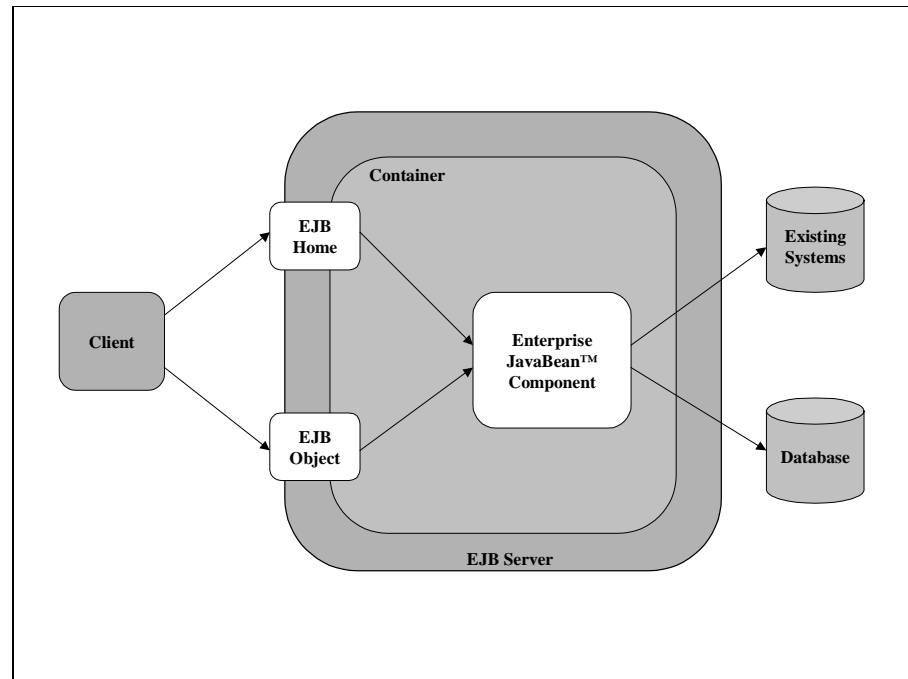
EJB –arkkitehtuurimalli myös mahdollistaa saumattoman Java –pohjaisten palveluiden ja niiden hallintajärjestelmien liittämisen yritysten jo olemassa oleviin IT – resursseihin (Information Technology).

Laajennetun palvelunhallintamallin mukainen palveluntuotannon näkökulma on mukana myös EJB –arkkitehtuurimallissa, sillä palveluntuotanto pyritään saamaan siinä mahdollisimman tehokkaaksi ja laadukkaaksi. EJB tukee tätä lähestymistapaa uudelleenkäytettävyyden avulla ja lisäksi standardoitu palveluiden suoritusympäristö ja komponenttimalli mahdollistavat nopean palveluiden lataamisen asiakkaiden saataville. Palveluun voidaan tuoda mukana valmiina komponentteina ennen vaikeasti toteutettavia piirteitä, kuten esimerkiksi tietoturva- ja luotettavuusominaisuuksia. EJB –arkkitehtuurimalli luo Internet - palveluille infrastruktuurin, joka tarvittaessa poistaa aikaa vievän systeemitason ohjelmoinnin palveluiden toteutuksesta.

EJB –arkkitehtuuri ei ota kantaa asiakaspään toteutukseen eikä määrittele mitään tarkkaa olioiden välistä kommunikointiprotokollaa. Palvelin voi tukea useita protokollia kuten esim. RMI:tä, DCOM:ia ja IIOP:ta (Internet CORBA Inter-ORB Protocol). Palvelun sisältävä EJB -palvelin voi näin ollen implementoida sen protokollan, joka on asiakasohjelmistojen kannalta tärkein ja järkevin. Asiakaspään ohjelmistoa ei myöskään ole välttämätöntä toteuttaa Javalla.

EJB –arkkitehtuuri perustuu Container –komponenttiin, joka toimii suoritusympäristönä EJB –komponenteille. Container –komponentti tarjoaa EJB -komponenteille hallinnollisia palveluja, kuten palveluhakemiston, transaktioidenhallinnan, olioiden tilojen ylläpidon, resurssien hallinnan, säikeistykseen (multithreading) ja tiettyjä tietoturvaominaisuuksia. EJB –komponentteja voidaan dynaamisesti ladata Container –komponentteihin. Ne myös tarjoavat palvelut EJB –komponenttien jakelua ja sovellusten välistä kommunikointia varten. Container –komponentteja ylläpitää EJB –palvelin

(middleware –ohjelmisto). Kuvassa 15 on esitetty Enterprise JavaBeans –arkkitehtuurimalli.

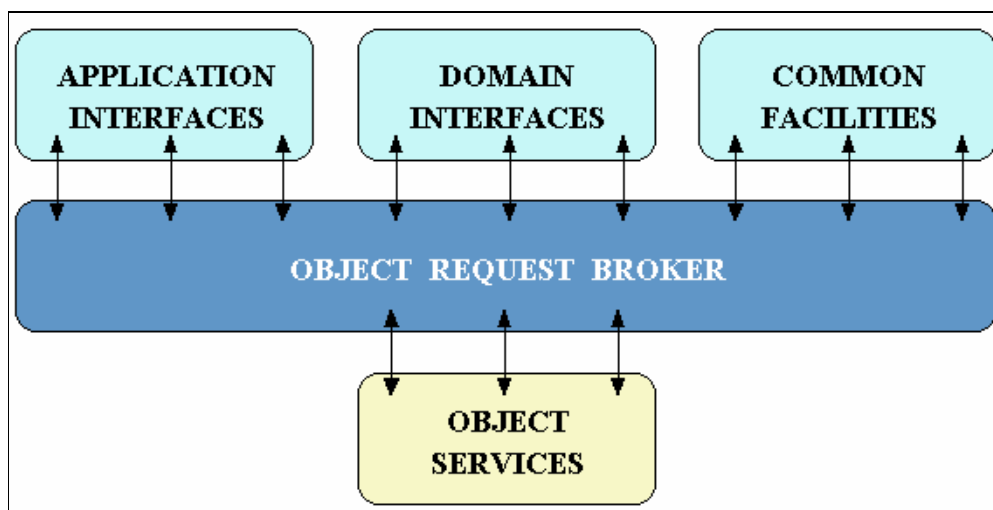


**Kuva 15:** Enterprise JavaBeans -arkkitehtuuri /17/.

### 3.7.4 CORBA (Common Object Request Broker Architecture)

OMG:n CORBA –arkkitehtuuri on kehitetty hajautettujen olioiden arkkitehtuurimalliksi. Se soveltuu näin ollen myös Internet -palveluiden arkkitehtuurimalliksi, koska kyseisten palveluiden tuottaminen hajautettujen olioiden avulla on mielekästä. Aikaisemmin CORBA –pohjaisesti toteutettiin pääasiassa kompleksisia ja skaalautuvuutta tarvitsevia räätälöityjä sovelluksia, mutta nykyään sillä toteutetaan yhä enemmän palveluja myös Internetiin. CORBA:n IDL (Interface Definition Language) ei ota kantaa toteutuksessa käytettävään ohjelmointikielen vaan sille löytyy ohjelmointikielisiä useille eri kielille. Palvelut voidaan toteuttaa esimerkiksi käyttämällä sekä CORBA- että

Java -teknologioita yhdessä. Kummatkin täydentävät toisiaan Internet palveluiden osalta. Java tarjoaa infrastruktuurin siirrettäville olioille (ohjelmille), jotka toimivat lähes kaikissa käyttöjärjestelmissä. CORBA puolestaan tarjoaa infrastruktuurin hajautetuille olioille, joiden avulla voidaan toteuttaa verkoista, komponenteista ja käyttöjärjestelmistä riippumattomia sovelluksia. CORBA -teknologian avulla voidaan toteuttaa uudelleen käytettäviä komponentteja palveluja varten ja sen arkkitehtuuri mahdollistaa tarvittaessa suurtenkin palveluiden implementoinnin Internet ympäristössä. Lisäksi CORBA -teknologian avulla on ollut mahdollista tuoda jo toteutettuja palveluja ja sovelluksia Internetin käyttäjien ulottuville suhteellisen pienillä muutoksilla alkuperäiseen toiminnallisuuteen ja logiikkaan. Kuvassa 16 on esitetty CORBA -viitemallin arkkitehtuuri.



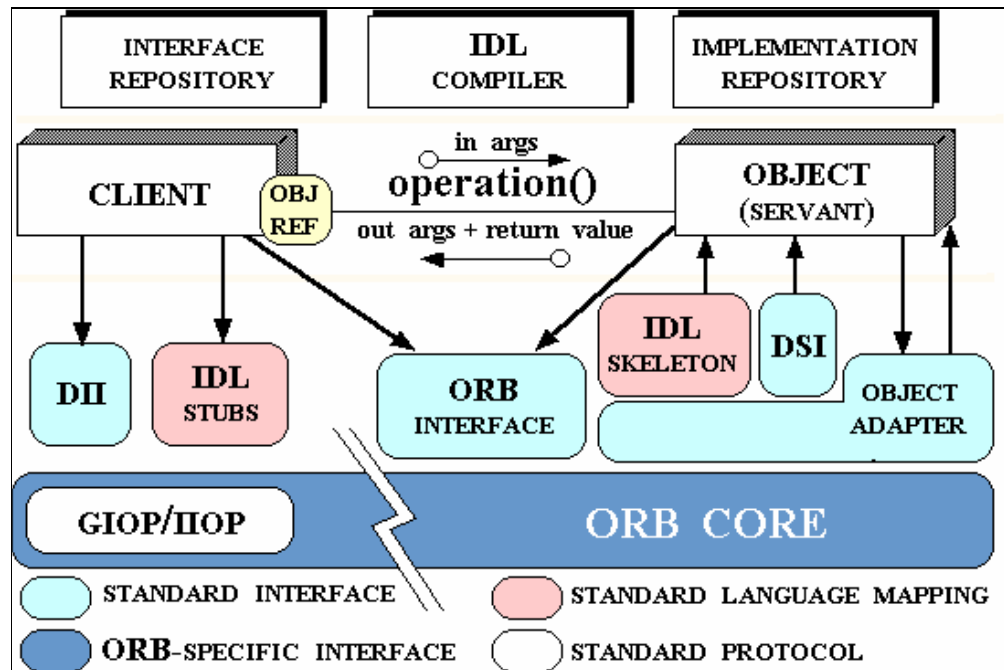
**Kuva 16:** CORBA -viitemallin arkkitehtuuri /18/.

CORBA -arkkitehtuuri perustuu DCOM -arkkitehtuurin ohella komponentteihin. Tärkeimpänä komponenttina siinä toimii ORB (Object Request Broker), joka toimii olioiden kommunikaatioväylänä (object bus). Se tarjoaa olioille mahdollisuuden läpinäkyvästi kutsua tai saada vastauksia toisilta olioilta, jotka voivat olla paikallisia tai muualla sijaitsevia.

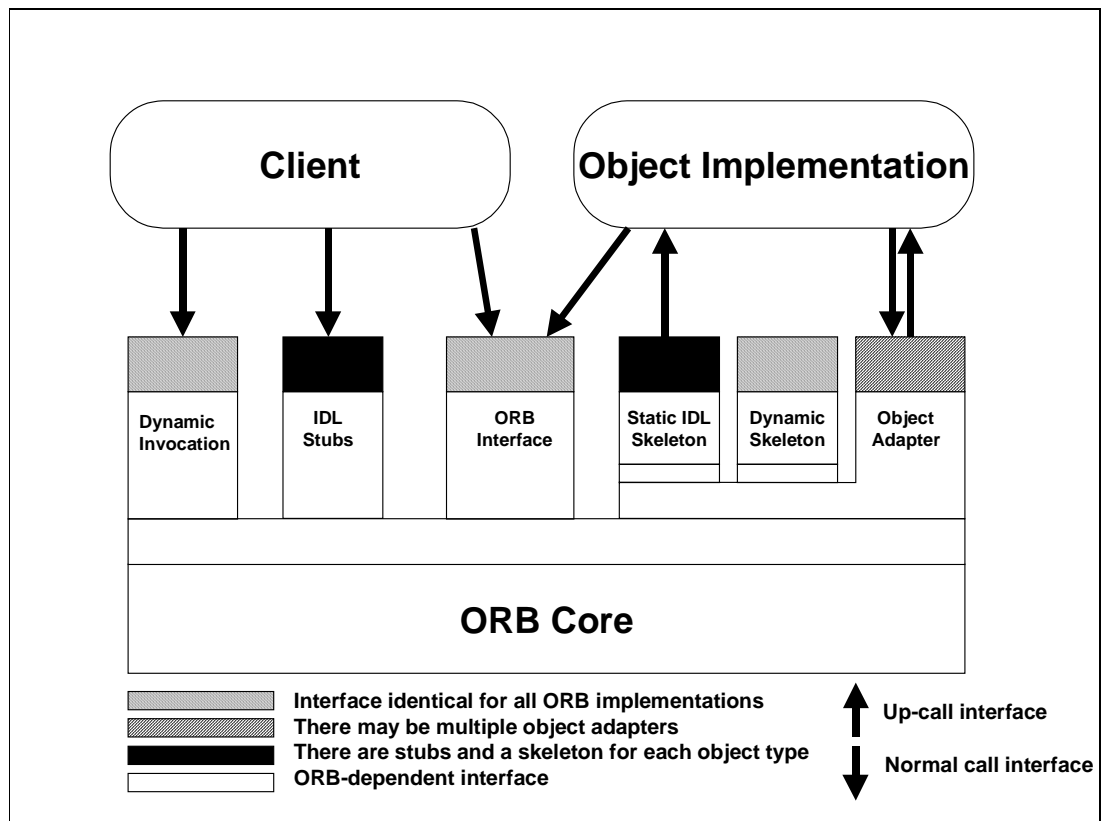
Asiakasoliot (Client) kommunikoivat staattisten IDL –rajapintojen (Interface Definition Language) tai dynaamisen kutsurajapinnan (Dynamic Invocation Interface) kautta. CORBA -asiakasoliot eivät ole tietoisia em. rajapintojen takana tapahtuvasta toiminnallisuudesta. Ne suorittavat kutsut (requests) yhtenäisellä tavalla riippumatta rajapinnasta, jota ne käyttävät kommunikointiin. Kutsut suoritetaan kohdeolion olioviitteen avulla (object reference). Olioviite on tieto, joka määrittelee yksilöllisesti ORB:iin kytkeytyneen olion. Sekä asiakkailta että palvelevana osapuolena toimivilla olioimplementaatioilla (object implementation) on käytössään näkyvä merkintätapa olioviitteestä. Se riippuu käytössä olevasta ohjelmointikielystä ja on näin ollen eristetty oikeasta ORB:n käyttämästä olioviitteestä. Yleensä eri ohjelmointikieli-implementaatiot tarjoavat jonkinlaisen olioviitettä vastaavan tietotyypin kuten esimerkiksi näkyvän osoittimen. Tämän avulla on mahdollista suorittaa kutsuja, toisilla ohjelmointikielillä toteutettuihin olioihin. Kutsun suorittava olio (asiakas) tietää kohdeolion tyyppin ja operaation, jonka se haluaa suorittavan. CORBA -olion asiakasrooli on vain käsitteellinen ja näin ollen tietty olio voi toimia asiakkaana ja palvelevana oliona samaan aikaan toisille CORBA -olioille. Olioiden (sovellusten) ei tarvitse olla toteutettu samalla ohjelmointikielillä, kun ne kommunikoivat keskenään ORB:n kautta.

Palvelevana puolena toimiva implementaatio tarjoaa semantiikan oliolle. Yleensä tämä tapahtuu määrittelemällä oliosta luodun instanssin data ja metodit. Usein olioimplementaatio käyttää hyväkseen muita olioita tai ulkopuolisia ohjelmistoja toteuttaakseen tarvitsemansa toiminnallisuuden kuten tietokantoja.

Kuvaan 17 on sijoitettu komponentit, jotka liittyvät ORB:n läpi tehtävään CORBA kutsuun. Asiakas (client) kutsuu olioviitteen avulla ORB:n kautta oliototeutuksen (object) operaatioita tai attribuutteja. Mikäli palautusarvo on määritelty se toimitetaan kutsuvalle osapuolelle, kun operaation on suoritettu. Lisäksi on mahdollista palauttaa myös operaation argumentteja takaisin asiakkaalle. Kuvassa 18 on esitetty CORBA –olioiden kommunikointirajapinnat.



Kuva 17: ORB -arkkitehtuuri /19/.



**Kuva 18:** CORBA -olioiden kommunikointirajapinnat /20/.

CORBA -arkkitehtuuri tarjoaa myös mahdollisuuden käyttää useita erilaisia valmiita palveluja, jotka toteuttavat jonkin tietyn tarkasti määritellyn ja standardoidun perustoiminnan. Näitä komponentteja kutsutaan yleisiksi palveluiksi (Common Object Services, CoS) ja niillä on kaikilla standardoidut rajapinnat, jotka on määritelty IDL -kielellä. Yleisiä palveluja on tällä hetkellä määriteltynä 17 erilaista, mutta niitä syntyy tulevaisuudessa lisää ja toisaalta jotkin saattavat hävitä tarpeettomina. Esimerkkeinä palveluista ja niiden toiminnallisuudesta ovat:

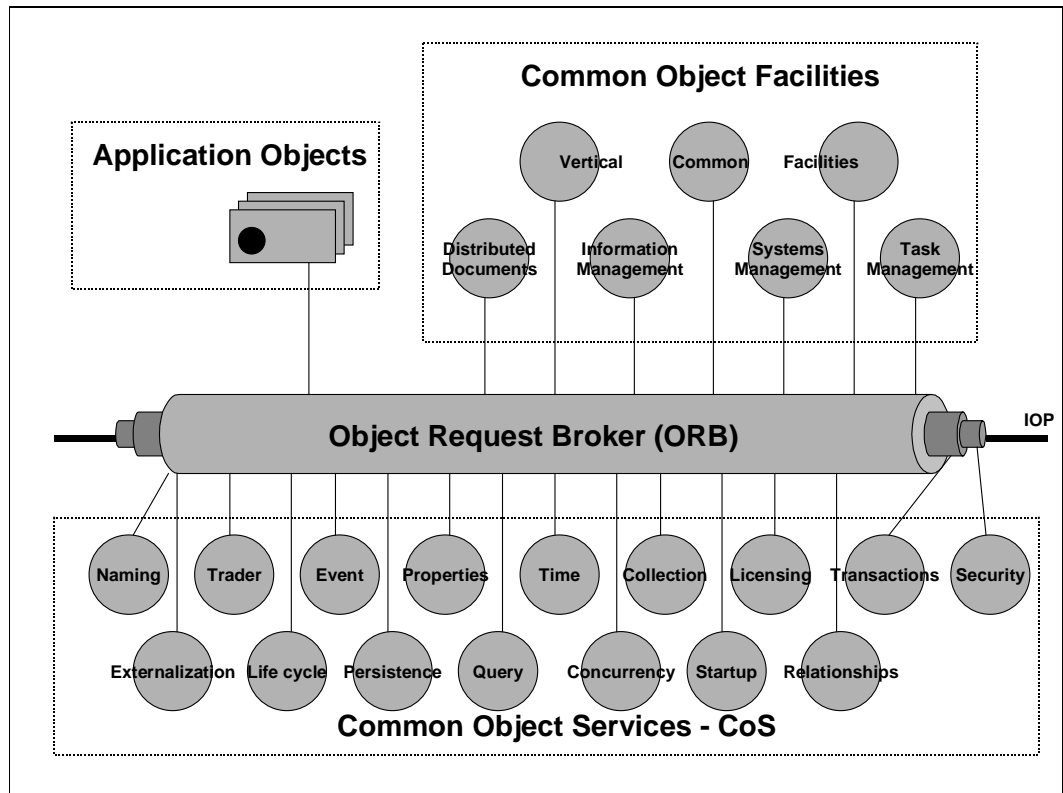
- Nimipalvelu (Naming Service), joka antaa mahdollisuuden löytää muita CORBA -olioita nimen perusteella. Se mahdollistaa myös CORBA -palvelimien rekisteröitymisen eli ne voivat tuoda itsensä julki muille olioille. Nimipalvelua voidaan verrata esim. puhelinluetteloon eli oliot voivat löytää

sen avulla toisen olion osoitteen (olioviitteen) kyseisen olion nimen avulla. Nimet ovat käyttäjän tajuttavia ja ne identifioivat olion. Nimeäminen liittyy nuo nimet olioviitteisiin.

- Salauspalvelu (Security Service) tarjoaa salaus- ja autentikointitoiminnot CORBA -olioille.

Yleisten palveluiden lisäksi OMG on määritellyt korkeamman tason palveluita (Common Object Facilities). Ne ovat horisontaalisesti orientoituneita palvelukokonaisuuksia ja tarjoavat loppukäyttäjälle näkyviä korkeamman tason sovelluspalasia. Näitä ovat esimerkiksi dokumenttien- ja järjestelmänhallinnan tarjoavat palvelukokonaisuudet. Edellisten lisäksi OMG on määritellyt myös vertikaalisia alakohtaisia palvelurajapintoja. Ne keskittyvät tietyille kapeille sektoreille kuten esim. telekommunikaatioon tai lääketieteeseen. Kuvassa 19 on CORBA –viitemalli palveluineen.





**Kuva 19:** CORBA –viitemalli palveluineen /21/.

### 3.7.5 Yhteenveto

Internet ja intranet -palveluiden toteuttamiselle on olemassa useita vaihtoehtoisia teknologioita. Teknologiakehitys onkin mahdollistanut yhä monipuolisempien palveluiden tuottamisen. Lupaavimpia teknologioita ovat Java ja CORBA yhdessä käytettynä sekä Javan EJB -komponentit. Myös palveluiden hallintaan on olemassa osittaisia valmiita ratkaisuja tai ainakin komponentteja joiden avulla palvelunhallintaa voidaan kehittää. Taulukossa 2 on vertailtu kuutta eri toteutusteknologiaa. Paras arvo on neljä tähteä ja mikäli jotain tiettyä ominaisuutta ei teknologiassa ole lainkaan on arvona viiva. Alkuperäinen taulukko löytyy englannin kielisenä Robert Orfalin ja Dan Harkeyn kirjasta Client / Server Programming with JAVA and CORBA /22/, josta löytyy myös kattavampi selvitys

eri teknologioiden ominaisuuksista sekä eroista ja puutteista. Alkuperäiseen taulukkoon verrattuna taulukkoon 2 on tehty pieniä kosmeettisia muutoksia. Taulukosta voidaan nähdä, että teknologioista monipuolisin on CORBA / IIOP mutta myös muilla on omat vahvuutensa.

**Taulukko 2:** Toteutusteknologioiden vertailutaulukko /22/.

OMINAISUUS	CORBA/IOP	DCOM	RMI	HTTP/CGI	SERVLETS	SOCKETS
Abstraktion taso	* * * *	* * * *	* * * *	* *	* *	*
Saumaton Java integraatio	* * * *	* * * *	* * * *	* *	* *	* *
Käyttöjärjestelmätuki	* * * *	* *	* * * *	* * * *	* * * *	* * * *
Java pohjaisuus	* * * *	*	* * * *	* *	* * * *	* * * *
Tuki tyypitetyille parametreille	* * * *	* * * *	* * * *	*	*	*
Configuroinnin helppous	* * *	* * *	* * *	* * *	* * *	* * *
Tuki hajautetuille metodikutsuille	* * * *	* * *	* * *	-	-	-
Tilakasite kutsuissa	* * * *	* * *	* * *	-	* *	* *
Tuki dynaamiselle olioiden löytämiselle ja metadatalle	* * * *	* * *	* *	-	*	-
Dynaamiset kutsut	* * * *	* * * *	*	-	-	-
Suorituskyky (ping)	* * *	* * *	* * *	-	*	* * * *
Ping arvot (ms)	3,5	3,8	3,3	827,9	55,6	2,1
Tietoturva	* * * *	* * * *	* * *	* * *	* * *	* * *
Hajautetut transaktiot	* * * *	* * *	-	-	-	-
Pysyvät olioviitteet	* * * *	*	-	-	-	-
URL pohjainen nimeäminen	* * * *	* *	* *	* * * *	* * * *	* * *
Monikieliset metodikutsut	* * * *	* * * *	-	* * *	-	* * * *
Ohjelmointikieli riippumaton siirtoprotokolla	* * * *	* * * *	-	* * * *	* * * *	-
Skaalautuvuus	* * * *	* *	*	* *	* *	* * * *
Avoin standardi	* * * *	* *	* *	* * * *	* *	* * * *

## 4 HALLINNAN KOHTEET

### 4.1 Sisältö

Sisällönhallinta on laaja käsite ja sen piiriin kuuluu kaikki se palveluiden sisältö ja informaatio, joita erilaiset palvelut tarjoavat. Sisältö voi olla lähes mitä tahansa dataa, jota voidaan välittää IP –pohjaisesti Internetissä. Perinteisten tekstipohjaisten palveluiden rinnalle on jo kehittynyt ääntä ja liikkuvaa kuvaa sisältäviä palveluja esim. urheilulähetyksiä. Nykyinen HTML –sivunkuvauskieli ei pelkästään tule riittämään tulevaisuudessa, koska sisältö tulee monipuolistumaan nykyisestä. Tällä hetkellä pääosin staattista tekstiä ja kuvia sisältävistä HTML –sivuista kehitty useita erilaisia sisältöobjekteja käsittäviä kokonaisuuksia, jolloin HTML:n ominaisuudet käyvät riittämättömäksi. Nykyisessä HTML –pohjaisessa toteutuksessa erilaisia sisältöobjekteja on vaikea hallinta, koska HTML ei ota kantaa itse sisältöön vaan keskittyy pelkästään esitysmuotoon. Kyseisen ominaisuuden eli objektien ns. metatiedon puuttuminen onkin HTML:n suurimpia puutteita. HTML -standardi ei myöskään ota kantaa dokumentin sisäiseen rakenteeseen, jolloin sen järkevyys jää täysin dokumentin tekijän varaan. Tällä hetkellä suurin osa selainohjelmista tukee uusinta 4.0 versiota HTML –kielestä.

HTML -kielen korvaaja saattaa olla monipuolisempi ja laajempi XML –kieli (Extensible Markup Language). XML on joustava ja laajennettavissa oleva SGML:n (Standard Generalized Markup Language) alijoukko. SGML on standardi, jonka avulla voidaan määritellä erilaisia merkintäkieliä (myös HTML on määritelty SGML:n avulla). XML kehitettiin SGML:n pohjalta, koska SGML on erittäin hankala toteuttaa käytännön sovelluksissa kuten selainohjelmissa. Lisäksi siinä on useita ominaisuuksia, joita käytetään harvoin tai ei koskaan.

XML on yhtenäinen tapa kuvata rakenteellisen dokumentin data ja datan konteksti eli ns. metadata. Perusajatuksena on, että dokumentin luoja voi lisätä omia tietotyyppisiä eli leimoja (tag) määrittelemällä oman merkintäkielen kyseistä dokumenttia varten. Kyseinen määritelmä eli DTD (Document Type Definition) sisältää dokumentin metadatan, jonka avulla itse dokumentti ja sen sisältö voidaan ymmärtää oikein sitä käsittelevissä sovelluksissa. Tämä mahdollistaa myös sen, että mikä tahansa sisältöobjekti voidaan sijoittaa dokumenttiin. Muut sovellukset osaavat tulkita sen DTD:n sisältämien tietojen avulla ja toimia sen mukaisesti mikä milloinkin on tarpeellista. XML erottaa toisistaan sisällön ja esitysmuodon, joten sama sisältö voidaan esittää tarvittaessa eri muodoissa eri sovelluksissa. XML:n laajennettavuus syntyy ominaisuudesta, jonka avulla käyttäjä voi määrittellä omia tyyppisiä dokumentteihinsa luomalla uusia leimapareja (tags). Koska XML -dokumentti voi sisältää mitä tahansa uusia leimoja, ja niiden sisälle voidaan upottaa mitä tahansa dataa, on periaatteessa mahdollista upottaa jopa ajettavia sovelluksia dokumentteihin. Kyseinen malli vaatii vain sen, että dokumentin vastaanottava pää osaa tulkita metadatan ja käsitellä varsinaisen sisältöobjektin asianmukaisella tavalla. XML 1.0 on World Wide Web Consortium:in (<http://www.w3.org>) uusin suositusversio ja XML:n käyttö yleistyy erilaisissa sovelluksissa ja palveluissa koko ajan. Markkinoille ovat tulleet jo ensimmäiset XML -pohjaiset selainohjelmat.

Palveluiden kannalta eräs oleellinen ominaisuus on miten itse sisällön avulla voidaan ohjata palvelun toimintaa. Esimerkkinä edellä mainitusta ominaisuudesta voidaan mainita HTML -sivujen hyperlinkit, joiden avulla selain voidaan ohjata uudelle hyperlinkin osoittamalle dokumentille. HTML -kielen rajoittuneisuuden vuoksi sillä ei voi juurikaan tehdä muunlaista palvelun ohjausta, mutta esim. XML mahdollistaa jo monipuolisempia ratkaisumalleja. Ladattu XML -dokumentti voi sisältää useita eri objekteja, jotka ohjaavat vastaanottavassa päässä palvelun toimintaa halutulla tavalla. Dokumentissa voi olla mukana esim. ajettavia ohjelmia, jotka suoritetaan asiakaspäässä vain tarvittaessa tai palvelu voi tarvittaessa valita mitä kentiä se käyttää XML -dokumentista.

## 4.2 Palvelut

Internet -sivut ovat kehittyneet muutamassa vuodessa staattisista HTML – pohjaisista tekstisivuista huomattavasti kohti vuorovaikutteisempia ja sisällöltään monipuolisempia kokonaisuuksia. Internetissä on jo tarjolla erilaisia palveluja, jotka toimivat vuorovaikutteisesti käyttäjän kanssa. Vuorovaikutteisuus on välttämätön ominaisuus kaupallisissa Internet palveluissa, sillä käyttäjä on pystyttävä tunnistamaan luotettavasti ja tämä yleensä vaatii kaksisuuntaista kommunikointia. On luultavaa, että ainakin osa palveluntarjoajista monipuolistaa tarjontaansa nopeassa aikataulussa kilpailuedun saavuttamiseksi. Palveluiden kehityksen edellytyksenä on ollut Internet ympäristöön kehittyneiden teknologioiden nopea kehitys ja nykyään palveluiden toteuttamiseen on valittavana useita erilaisia teknologioita kuten edellä on mainittu.

Palvelut monipuolistuvat tulevaisuudessa ja syntyy aivan uudenlaisia palvelutyyppisiä. Esimerkiksi palvelut, joissa käyttäjä lataa tarvitsemansa ohjelmiston koneelleen verkosta (verkkotietokone) voivat yleistyä. Käyttäjät lataavat tarvitsemansa ohjelmat koneelleen vain tarvittaessa ja maksavat käytöstä esim. per käyttökerta. Näin ollen käyttäjän ei tarvitse ostaa itselleen kallista ohjelmistoa, jota hän käyttää vain harvoin. Vaikka erilaisten toimisto-ohjelmistojen (tekstinkäsittely, taulukkolaskenta, esitysgrafiikka, kuvankäsittely jne.) hinnat ovatkin tulleet alaspäin, kyseinen palvelumalli voisi toimia erityisesti sillä alueella. Viihdepalveluiden puolella esimerkiksi tietyt pelit voisivat olla asiakkaiden itselleen ladattavissa saman mallin mukaisesti. Vaikka verkkotietokonemalli ei yleistyisikään varsinainen palvelumalli voi silti toimia, sillä kyseisten palveluiden käyttöasteet voivat olla yleistyessään suuria. Asiakkaan kannalta palvelumalli toimii hyvin, koska hänellä on aina käytettävissään haluamastaan ohjelmasta uusin versio eikä hyllyssä lojuisi kasapäin vanhoja tarpeettomia ohjelmia. Tavallaan palvelumallia sovelletaan rajoittuneesti,

varsinkin yritysmaailmassa kaupallisissa tuotteissa, jo tänä päivänä, sillä yhä useammin tuotteen dokumentaation on saatavilla vain verkosta ladattavana dokumenttina. Mallin eräs hyvä puoli on se, että tarjolla on aina uusin ja oletettavasti myös toimivin tai ajan tasalla oleva versio.

Palvelut ovat tulevaisuudessa yhä enemmän prosesseja, joista monipuolisimmat saattavat sisältää vuorovaikutuksia useiden eri järjestelmien ja komponenttien kanssa. Tämä vie palvelunhallintaa yhä enemmän prosessienhallinnan suuntaan, sillä ongelmaksi muodostuu kuinka hallita monimutkaisia prosesseja Internet - ympäristössä. Samalla monipuoliset palvelut ja kasvavat käyttäjämäärät asettavat vaatimuksia varsinaiselle palvelunjakelutekniikalle, sillä sen on oltava riittävän suorituskykyinen, vikasietoinen ja skaalautuva. Lisäksi tarvitaan joustavat ylläpito ja päivitys ominaisuudet, jotta palveluntarjoaja voi nopeasti ja tehokkaasti muokata palvelujaan tarpeen mukaan. Palvelunhallinta muodostaa tulevaisuudessa oman itsenäisen kokonaisuuden, kun siihen kuuluvia ominaisuuksia on tähän asti liitetty muihin hallintajärjestelmiin tai palvelunhallintaa ei ole ollut lainkaan. Sen merkitys kasvaa tulevaisuudessa voimakkaasti, sillä se on avainasemassa luotaessa uusia palveluja liiketoiminnallisista lähtökohdista.

### **4.3 Verkonhallinta**

Verkonhallinta keskittyy erilaisten verkkoelementtien hallintaan. Hallinta on keskitettyä ja mahdollisimman pitkälle automatisoitua. Hallittava verkko ja sen elementit voivat maantieteellisesti sijaita laajallakin alueella. Edellä mainitusta syystä johtuen verkonhallinta on pyritty keskittämään. Nykyään IP –verkossa voi olla useita erilaisia elementtejä, jotka kommunikoivat keskenään. Hallittavia elementtejä ovat esimerkiksi verkkoon liitetyt reitittimet, kytkimet, sillat ja verkkoon liitetyt tietokoneet sekä lisäksi kasvavassa määrin myös erilaiset lisälaitteet kuten esim. tulostimet. On oletettavaa, että tulevaisuudessa elementtien

määrä moninkertaistuu lukumäärällisesti ja että uusia elementtityyppejä tulee mukaan kuvioon.

#### **4.4 Asiakkaiden hallinta Internet -palveluntarjoajan ympäristössä**

Palveluntarjoajien on ratkaistava asiakkaiden hallintaan liittyvät ongelmat. Asiakkaat on pystyttävä tunnistamaan luotettavasti, jotta heitä voidaan laskuttaa palveluiden käytöstä ja luottamuksellisten tietojen siirron täytyy tapahtua salattuna. Nykyään käytössä olevat sovelluskohtaiset autentikointi- ja salausmenetelmät ovat hankalia päivittää ja vaihtaa. Ratkaisu onkin todennäköisesti IP Security –protokolla (IPSec), joka hoitaa tunnistuksen ja salauksen täysin verkkokerroksella.

##### **4.4.1 Asiakkaiden tunnistus**

Asiakkaiden luotettava tunnistaminen eli autentikointi on välttämätön ominaisuus, jotta palveluista voidaan laskuttaa asiakasta oikein. On oletettavaa, että laskutus suurimmassa osassa tapahtuu käytön perusteella. Se voi palvelun luonteesta riippuen olla käyttöaikaan tai käyttökertoihin perustuvaa ja eräissä erikoistapauksissa myös siirrettyyn datamäärään pohjautuvaa. Asiakkaalla on oltava digitaalinen identiteetti eli sähköinen allekirjoitus. Yksinkertaisimmillaan asiakas voidaan tunnistaa käyttäjätunnuksen ja salasanan perusteella, mutta muitakin malleja löytyy. Esimerkiksi erilaisiin älykortteihin pohjautuvat ratkaisut yleistyvät Suomessa viranomaistoiminnassa. Niiden ongelmana on kuitenkin se, että ne vaativat yleensä jonkin uuden lisälaitteen kuten kortinlukijan.



#### 4.4.2 Tietoliikenteen salaus

Tiedonsiirron turvallisuus, siirretyn tiedon eheys ja varmuus siitä, että siirrettyä dataa ei ole luvattomasti muutettu siirron aikana, tulee jatkossakin olemaan tärkeä ratkaistava ongelma avoimessa Internet –verkossa. Tähän asti salaus on hoidettu sovellustasolla, mutta kyseinen lähestymistapa tuo mukanaan muutamia vaikeasti ratkaistavia ongelmia. Sovellustasolla selaimissa on käytetty esimerkiksi SSL –protokollaa (Secure Sockets Layer) ja sähköpostin salaamisessa PGP -ohjelmaa (Pretty Good Privacy). Internetin tapaisessa nopeasti muuttuvassa verkossa tarvitaan helppoa ja yksinkertaista turvallisuusominaisuuksien päivitettävyyttä. Tätä ei kuitenkaan voida järkevästi toteuttaa puhtaasti sovellustasolla. Ratkaisu voi olla IP Security –protokolla (IPSec), jota kehitetään IPSec –työryhmässä IETF:n (Internet Engineering Task Force) alaisuudessa ja josta on tarkoitus tulla kiinteä osa IPv6 -protokollaa.

IPSec on täysin verkkokerroksella toimiva autentikoiva ja salaava protokolla. Sen etuna on helppo päivitettävyyys ja käytettävyys. Turvaominaisuuksien käyttö on helppoa kaikille sovellusohjelmille, koska ne on implementoitu jo verkkokerroksella. IPSec:in käyttöönotto vaatii TCP/IP –protokollapinon päivityksen tai erillisen moduulin käyttöönoton protokollapinon alla. IPSec -komponentteja ovat: pakettitason salauksen hoitava ESP –protokolla (Encapsulating Security Payload), autentikoinnin hoitava AH -protokolla (Authentication Header) ja avainten hallinnan hoitava IKE –protokolla (Internet Key Exchange). ESP –protokolla käyttää erilaisia salausmenetelmiä tiedon koskemattomuuden takaamiseksi ja AH –protokolla digitaalisia allekirjoituksia tiedon lähteen varmistamiseksi. IKE –protokollan avulla kommunikoivat osapuolet voivat neuvotella käytettävät salausmenetelmät, salausavaimet ja sen milloin ja millä logiikalla avaimia vaihdetaan. Lisäksi IPSec tukee X509.V3 standardin mukaisia sertifikaatteja, joiden avulla voidaan autentikoida laitteistoja kuten reitittämiä.

ESP -protokolla tukee useita erilaisia symmetrisiä salausalgoritmeja. Oletusarvona käytetään 56 bittistä DES –algoritmia (Data Encryption Standard) mutta algoritmit ovat vaihdettavissa tarpeen mukaan. ESP- ja AH –protokollien toiminnallisuuksissa on hieman päällekkäisyyttä, koska myös ESP –protokolla tukee jossain määrin autentikointia. ESP -data sijaitsee IP -datagrammissa standardin mukaisen IP –otsikon (IP header) jälkeen ja sisältää sekä datan että korkeampien kerrosten protokollien otsikkokentät. Korkeamman kerroksen protokollat käyttävät IP –protokollaa reitityksessä. Koska paketissa on standardi IP –otsikko, verkkoelementit voivat reitittää sen kuten minkä tahansa IP – datagrammin. IPSec onkin taaksepäin yhteensopiva teknologia verkkoelementtien osalta, sillä myös siitä tietämättömät verkkoelementit osaavat reitittää paketit oikein. Taulukkoon 3 on kerätty IPSec –protokollan tärkeimmät ominaisuudet ja edut.

**Taulukko 3:** IPSec ominaisuudet ja edut /23/.

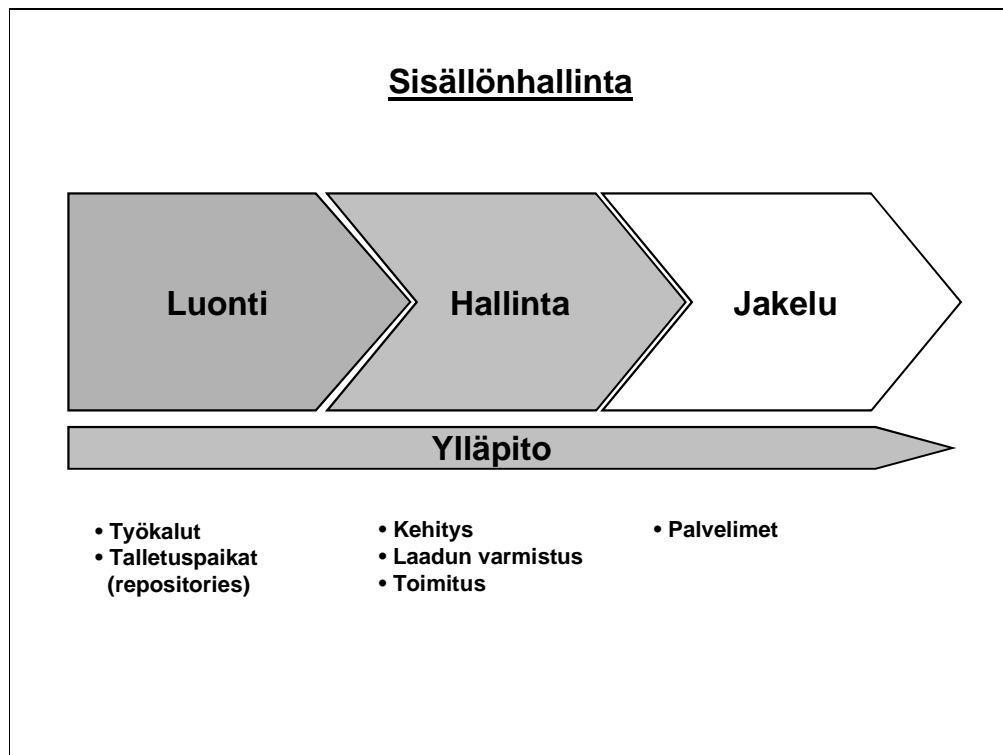
OMINAISUUS	KUVAUS/KOMMENTTI	ETU
Tiedon luottamuksellisuus, koskemattomuus ja aitous	IPSec tarjoaa nämä verkon peruspalvelut salaus- ja autentikointiteknologioiden avulla	Datan lähetyks voidaan tehdä julkisen verkon läpi ilman pelkoa, että joku tarkkailee, muuttaa tai kopioi sitä. Tämä mahdollistaa VPN (Virtual Private Network), extranet ja etäkäyttösovellukset.
Integroitu ratkaisu	IPSec on tarjolla ohjelmistona eli tarvitaan ainoastaan verkon infrastruktuurin päivitys	Turvallisuus voidaan toteuttaa ilman kalliita muutoksia jokaiseen tietokoneeseen. Tämä tuo suuria kustannussäästöjä, koska vain verkon infrastruktuuria tarvitsee muuttaa
Tuki sertifiikaateille	Laitteet tunnistetaan automaattisesti käyttäen digitaalisia sertifiikaatteja	Tämä ominaisuus skaalautuu suuriinkin verkkoihin, jotka vaativat salattuja yhteyksiä monien laitteiden välille
IKE	IKE protokolla neuvottelee automaattisesti turvallisuusliitot (Security Association)	Mahdollistaa turvallisen tietyillä parametreilla tapahtuvan kommunikoinnin ilman kallista manuaalista ennalta tehtyä konfigurointia
Joustavat turvallisuusvaihtoehdot	Liikenne voidaan valita salattavaksi laajennettujen pääsyylojien (access list) avulla	Vain haluttu liikenne voidaan salata suorituskyyvyn nostamiseksi. Erilaila luokiteltu data voidaan salata eri avaimilla ja algoritmeilla
Standardi ratkaisu	IPSec on kehityksen alla oleva IETF standardi	IPSec mahdollistaa eri valmistajien verkkoelementtien, pc:iden ja muiden laitteiden yhteistoiminnallisuuden

## **5 HALLINTARATKAISUT**

### **5.1 Sisältö**

Palveluiden sisällönhallinta on tällä hetkellä monilta osiltaan vielä käsityötä vaikka työkaluja erilaisten Internet -palveluiden sisällönhallintaan jo löytyykin. On kuitenkin oletettavaa, että keskitettyyn sisällönhallintaan aletaan kiinnittää tulevaisuudessa yhä enemmän huomiota, sillä sen merkitys kasvaa palveluiden kehittyessä ja palveluiden sisällön monipuolistuessa. Ongelmana onkin kuinka hallita laajoja kokonaisuuksia, joissa on paljon dynaamista dataa ja jotka vaativat usein päivityksiä. Lisäksi varsinaisen sisällön tuottajat eivät useinkaan ole teknillistä henkilökuntaa ja heidän olisi pystyttävä hoitamaan tarvittavat sisältöön liittyvät lisäykset ja päivitykset hallitusti ja helposti.

Kattavaan sisällönhallintaan voidaan katsoa kuuluvaksi kuvassa 20 olevan ketjun toiminnot: sisällön luonti, hallinta ja jakelu.



**Kuva 20:** Sisällönhallintaketju.

Toimivan sisällönhallintajärjestelmän tulisi pitää sisällään seuraavat ominaisuudet joustavuuden ja käytettävyyden takaamiseksi:

- Mahdollisimman pitkälle automatisoitu sisällönhallintaprosessi.
- Helppokäyttöiset käyttöliittymät ja liitynnät ulkopuolisiin työkaluihin.
- Sisällönhallintajärjestelmän ylläpidettävyys.
- Ryhmätyöskentelyn mahdollisuus eli usean yhtäaikaisen päivityksen tai lisäyksen mahdollisuus.
- Mahdollisuus käyttää valmiita pohjia ja rakenteita (templates).
- Mahdollisuus myöntää erilaisia oikeuksia eri sisällöntuottajille.
- Tuki heterogeenisille sisällöntuotanto- ja palvelinympäristöille.
- Laadun varmistus.
- Tietoturva.

Vaikka sisällönhallinta onkin suhteellisen uusi käsite, markkinoilta löytyy jo tuotteita kyseiselle alueelle. Esimerkiksi Interwoven tuotteet TeamSite ja OpenDeploy (<http://www.interwoven.com>), Documentumin 4i (<http://www.documentum.com>) ja Inktomin Content Distributor (<http://www.inktomi.com>) ovat sisällönhallintaan tehtyjä ohjelmistoja. Tuotteiden taso vaihtelee ja kattavammat ratkaisut voivat olla tänä päivänä vielä erittäin kalliita.

## **5.2 Palvelunhallinta**

Palvelunhallinnan ratkaisut ovat vielä nykyään puutteellisia. Kokonaisvaltaista palvelunhallintamallia tukevia ratkaisuja ei ole. On selvää, että tulevaisuudessa palvelunhallintaan kiinnitetään yhä enemmän huomiota ja sen toteuttamiseen etsitään keinoja. Kattavan ja kokonaisvaltaisen palvelunhallinnan toteutumisen ehtona on, että palveluille saadaan standardoitua avoimet hallintarajapinnat. Palvelunhallinta voidaan toteuttaa esimerkiksi puhtaasti CORBA –pohjaisesti tai J2EE –standardin (Java 2 Enterprise Edition) mukaisesti. Palveluiden kehittämistä ja jakelua varten on olemassa J2EE –standardiin pohjautuvia sovelluspalvelin (Application Server) tuotteita.

### **5.2.1 CORBA -pohjaiset toteutukset**

CORBA –teknologian tapauksessa palveluilla tarkoitetaan yleensä sen omia valmiiksi määriteltyjä palvelukomponentteja. Kyseiset palvelukomponentit toteuttavat yleensä tarkoin määritellyn, suhteellisen rajatun, yleisesti hajautetuissa oliosovelluksissa tarvittavan toiminnon, joita voidaan käyttää hyödyksi toteutettaessa Internet -palveluja.

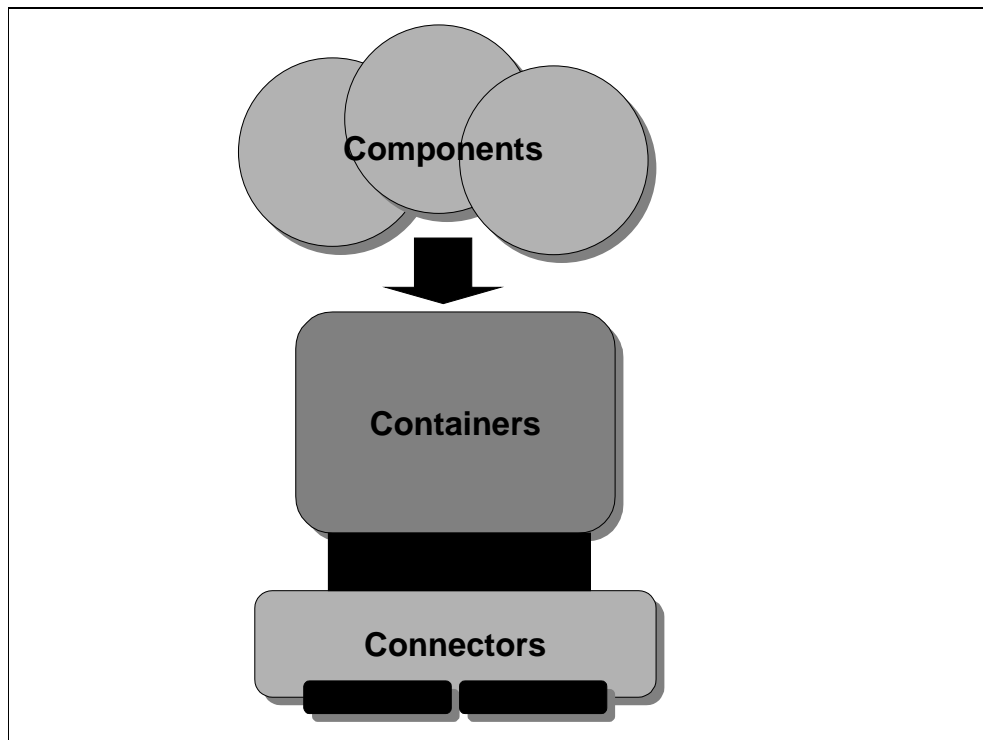
Palvelunhallinnan kannalta palveluja voidaan tarkastella CORBA –olioina, joilla on tietyt muille olioille näkyvät rajapinnat. Hallinta voidaan toteuttaa ennalta määritellyn hallintarajapinnan avulla, jonka palveluoliot toteuttavat. Rajapintoja voidaan tarvittaessa laajentaa, mikäli jokin palvelun toteuttava olio sitä vaatii. Jotta palvelunhallinta voidaan toteuttaa CORBA:lla, on pystyttävä määrittelemään hallintarajapinnat. Yhtenäisen hallintarajapinnan määrittely voi olla vaikea tehtävä, sillä palvelut voivat poiketa toisistaan ominaisuuksiltaan paljonkin. Ongelmaa voidaan lähestyä ryhmittelemällä samankaltaisia palveluja yhteen ja määrittelemällä eri palveluryhmille omat hallintarajapinnat. Hallintasovellukset ja hallittavat palvelut kommunikoivat keskenään ORB:n kautta ja tarvittaessa hallintasovellukset voivat käyttää apunaan olemassa olevia CORBA –palveluja. Järjestelmä on myös mahdollista toteuttaa siten, että hallintasovellus voi dynaamisesti hakea itselleen haluamansa palvelutoteutuksen hallintarajapinnan tiedot CORBA:n dynaamisten oliokutsujen avulla (DII, Dynamic Invocation Interface). Näin hallintasovellus voi mukautua erilaisiin palveluiden tarpeisiin ja samaa hallintasovellusta voidaan käyttää myös uusien palveluiden hallintaan.

CORBA pohjaisen palvelunhallinnan etuja ovat:

- Palveluiden ja hallintasovellusten toteutukset voidaan vaihtaa joustavasti, kunhan ne vain edelleen toteuttavat tarvittavat rajapinnat. Molemmat voidaan esim. päivittää suorituskykyisempään versioon ajonaikaisesti.
- Tarvittaessa järjestelmä voi toimia dynaamisesti.
- Eri komponentit voivat olla toteutettuja millä tahansa ohjelmointikielellä ja pystyvät silti kommunikoimaan keskenään.
- Muut CORBA -teknologian yleisesti tuomat edut kuten luotettavuus ja skaalautuvuus.

### 5.2.2 Java 2 Enterprise Edition

J2EE eli Java 2 Enterprise Edition (Sun Microsystems Inc.) on standardi monikerroksisten (multitier) sovellusten kehittämiseen. Se soveltuu myös Internet -palveluiden kehittämiseen. J2EE -arkkitehtuurimalli jakaa sovellukset kolmeen periaatteelliseen osaan: komponentteihin (Components), Säiliöihin (Containers) ja liittimiin (Connectors). Kuvassa 21 on esitetty J2EE –komponenttimalli.



**Kuva 21:** J2EE -komponenttimalli /24/.

Komponentit ovat sovellusohjelmia, joita sovelluskehittäjät kehittävät kun taas systeemitoimittajat keskittyvät säiliöiden ja liittimien kehittämiseen. Standardi määrittelee neljä eri komponenttityyppiä, joita J2EE -tuotteiden on tuettava. Komponentit voivat olla Javalla toteutettuja sovellusohjelmia, Applet tai Servlet –komponentteja, JSP –sivuja tai EJB –komponentteja. Säiliöt vaikuttavat

komponenttien ja liittimien välillä ja tarjoavat näille erilaisia palveluita läpinäkyvästi. Ne tarjoavat ajonaikaista tukea komponenteille. Kyseisen mallin avulla saavutetaan ominaisuus, jonka avulla on mahdollista toteuttaa ajonaikaisesti päätettäviä toiminnallisuuksia komponentteihin (vrt. koodiin toteutettu toiminnallisuus). Etuna kyseisessä lähestymistavassa on se, että asiakaspään ohjelmistoja (komponentteja) ei tarvitse aina päivittää, vaan päivitykset toiminnallisuuteen voidaan tehdä palvelin päässä. Liittimet toimivat sovellusrajapintoina olemassa oleviin järjestelmiin ja tuovat mukanaan mahdollisuuden kehittää palveluja, jotka käyttävät hyväkseen jo olemassa olevia teknologioita.

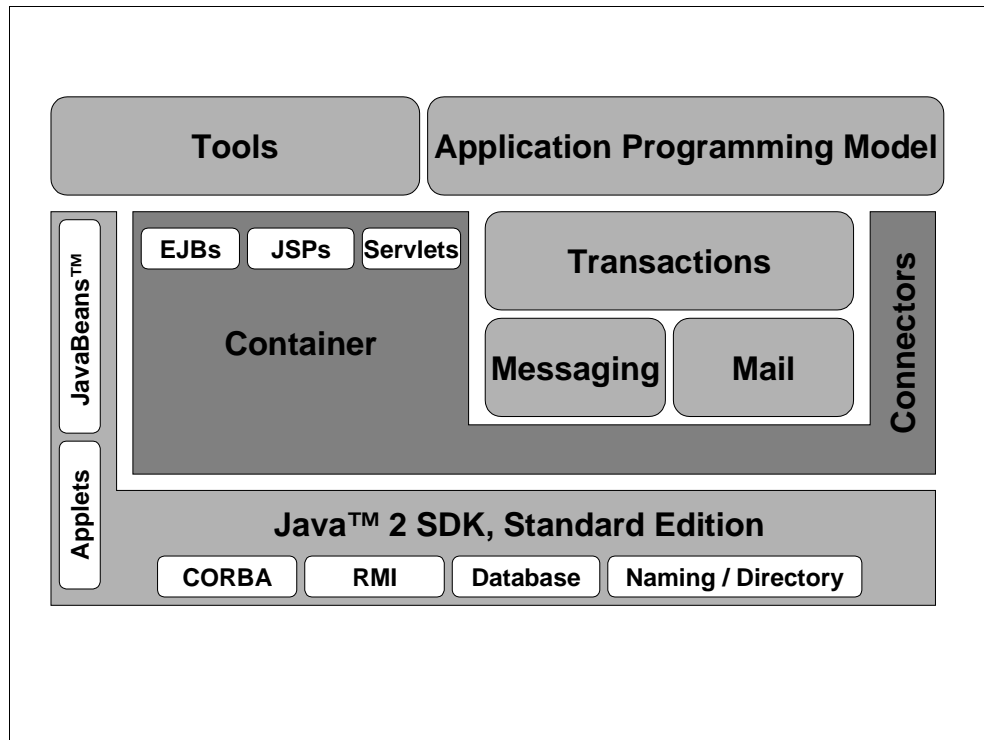
J2EE –standardi rakentuu Java 2 SDK:n (Software Development Kit) eli ohjelmistokehityspaketin päälle. Se tarjoaa ohjelmistokehittäjille ja ylempien kerroksien komponenteille erilaisia palveluja ja ohjelmistokomponentteja. Erilaisia Java 2 SDK:n tarjoamia komponentteja ovat mm. Applet:it sekä JavaBeans:it ja lisäksi se tarjoaa erilaisia sovelluskehitysrajapintoja (API, Application Programming Interface), joita ovat:

- CORBA,
- RMI,
- Database,
- Naming / Directory.

Database eli tietokantarajapinta tarjoaa liittynät erilaisiin tietokantoihin ja Naming / Directory –rajapinta tarjoaa nimi- ja hakemistopalveluiden perustoiminnallisuudet. Toteutettavat palvelut voivat suoraan käyttää hyväkseen kyseisiä palveluja mutta niitä käyttävät myös korkeamman tason palvelukokonaisuudet. J2EE –komponenttimallin mukaisia komponentteja ovat EJB -komponentit, JSP –sivut ja Servlet:it. Lisäksi standardi tarjoaa korkeamman tason palveluja kuten transaktoiden hallinnan, sähköposti- ja viestipalvelun. Edellä mainittujen yläpuolella ovat vielä erilaiset työkalut ja yleinen



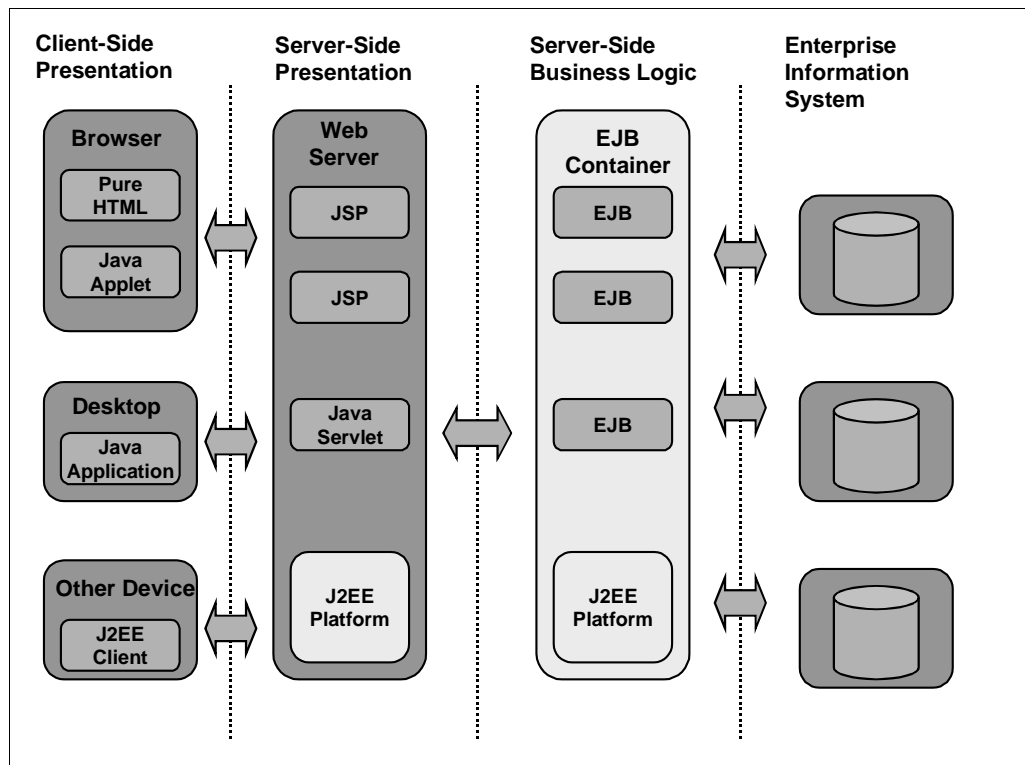
sovelluskehitysmalli. Kuvassa 22 on esitetty kokonaisuudessaan J2EE – arkkitehtuurimalli.



**Kuva 22:** J2EE -arkkitehtuuri /25/.

Kuvassa 23 on J2EE -standardin mukainen sovellusmalli. Se voidaan jakaa neljään periaatteelliseen kerrokseen tai tasoon. Asiakaskerroksella toimivat erilaiset loppukäyttäjän sovellukset, joiden avulla varsinainen palvelu saadaan käyttäjän ulottuville. Ne voivat olla selainpohjaisia, itsenäisiä sovelluksia tai muihin ratkaisuihin perustuvia (esim. matkapuhelimiin). Yhteisenä ominaisuutena näillä on se, että ne kaikki ottavat yhteyden palvelinpuoleen verkkopalvelimen (web server) kautta. Palvelinpuolen esityskerroksella tarjotaan tarvittaessa palveluiden visuaalinen ulkoasu asiakasohjelmille. Se voidaan tuottaa JSP – sivuilla tai tavallisina HTML –sivuina. Mikäli asiakassovellus ei tarvitse esityskerroksen tuottamaa dataa, se kommunikoi Servlet:ien välityksellä palvelinpuolen logiikkakerroksen (business logic) kanssa. Liiketoimintalogiikka

on sijoitettu EJB –komponentteihin, joita ajetaan EJB –ympäristössä. Kyseinen kerros voi tarvittaessa kommunikoida ja käyttää apunaan ulkopuolisia suljettuja järjestelmiä. Viimeisen tason muodostavatkin yrityksen muut suljetut järjestelmät, joita voidaan käyttää apuna palveluja toteuttaessa ja luotaessa. Kuvassa 23 on kuvattu J2EE –sovellusmalli ja sen eri kerrokset.



**Kuva 23:** J2EE –sovellusmalli /24/.

### 5.2.3 Sovelluspalvelimet (Application Servers)

Erityisesti palveluiden kehitystä varten, mutta myös palvelunhallintaan Internetissä, on kehitetty tuoteryhmiä, joita usein kutsutaan ns. sovelluspalvelimiksi eli Application Server -tuotteiksi. Ne pyrkivät tarjoamaan suorituskykyisen, skaalautuvan, vikasietoisen ja helppokäyttöisen tuotanto- ja hallintajärjestelmän Internet -palveluille. Ne ovat yleensä J2EE -standardin

toteutuksia, jonka avulla on saavutettu järjestelmäriippumaton skaalautuva ja olemassa oleviin järjestelmiin liitettävä kehitys- ja jakeluympäristö palveluille. Pääasiassa sovelluspalvelimien palvelut tuotetaan EJB-, JSP- tai Servlet -komponenteilla ja lisäksi ne tarjoavat muita valmiita komponentteja monipuolisten Internet -palveluiden toteuttamiseen. J2EE -standardin mukaisten komponenttien lisäksi sovelluspalvelimissa on usein mukana myös erilaisia XML –ominaisuuksia, suorituskykyisiä tietokantaliityntöjä, tietoturva ominaisuuksia ja liityntöjä sovelluskehittäjiin. Lisäksi eri sovelluspalvelimissa on valmistajakohtaisia lisäyksiä ja ominaisuuksia. Sovelluspalvelimien tuotanto- ja jakelumallit perustuvat komponenttimalliin, jonka avulla kehitetyt palvelukomponentit ovat uudelleenkäytettäviä ja laajennettavia. Mallin etuna on myös se, että palvelua käyttävä asiakassovellus voi olla toteutettu monella eri tavalla, eli se voi olla selainohjelma, Java –sovellus, puhdas CORBA –sovellus, ActiveX –sovellus tai jopa matkapuhelin. Valmiita sovelluspalvelin tuotteita tai tuoteperheitä ovat esimerkiksi BEA WebLogic (<http://www.bea.com/products/weblogic/>) ja IBM WebSphere (<http://www-4.ibm.com/software/webservers/appserv/>).

### 5.3 Verkonhallinta

Verkonhallinta (network management) pitää sisällään muutakin kuin vain palveluntarjoajan järjestelmän tarkkailun. Sen piiriin kuuluvat seuraavat asiat: Verkkoelementtien vikatilanteiden havainnointi ja niistä toipuminen, konfiguraatiot, tietoturva, suorituskyky ja käyttäjienhallinta. Verkonhallinta on huomattu tärkeäksi osa-alueeksi ja esimerkiksi kansainvälinen ei-kaupallinen TeleManagement Forum (<http://www.nmf.org>) toimii sen kehityksen edistämisen puolesta. Palveluntarjoajan kannalta verkonhallinta on tärkeä hoitaa kunnolla, koska sen avulla palveluntarjoaja pystyy takaamaan riittävän palvelunlaadun asiakkailleen. Lisäksi alusta alkaen hyvin hoidettu verkonhallinta takaa yleensä laajentumistilanteissa palveluntarjoajalle paremmat keinot selvittää laajentumisesta

ongelmitta. Palveluntarjoajien intresseissä on lähes aina kasvattaa asiakaskuntaa, mikä usein johtaa myös muutoksiin palveluntarjoajan verkossa.

Tällä hetkellä verkonhallinnan ja siihen liittyvän verkkoelementtienhallinnan yksi tärkeimpiä välineitä on SNMP –protokolla (Simple Network Management Protocol). Se on protokolla, jonka avulla voidaan välittää hallintatietoa eri elementtien välillä. SNMP –protokollasta on olemassa kaksi versiota: SNMP version 1 ja SNMP version 2. Jälkimmäiseen versioon on tehty parannuksia esimerkiksi lisäämällä uusia operaatioita ja korjattu ensimmäisessä versiossa esiintyneitä tietoturva-aukkoja. Versio 2 laajensi protokollaa huomattavasti. Edellisten lisäksi SNMP –protokollasta kehitetään edelleen seuraavan sukupolven versiota IETF:n (Internet Engineering Task Force) alaisuudessa SNMPv3 työryhmässä (SNMPv3 Working Group) ja sen työn pohjalta onkin jo julkaistu ensimmäiset standardit koskien SNMPv3:sta. Lähes kaikki verkkoelementit tukevat SNMP –protokollaa jossain mittakaavassa ja sen päälle on toteutettu useita tuotteita, jotka edellä mainittua protokollaa käyttäen toteuttavat verkonhallintaan liittyviä toimintoja.

SNMP –protokollan rinnalle on kehitetty myös CMIP –protokolla (Common Management Information Protocol). Se kehitettiin korvaamaan SNMP 1980 –luvun loppupuolella ja kehitystyössä oli mukana suuria organisaatioita ja yrityksiä. CMIP kehitettiin SNMP:n pohjalta ja sen tarkoituksena oli poistaa SNMP:ssä olevat puutteet ja siitä oli tarkoitus tehdä kattavampi ja yksityiskohtaisempi verkonhallintaprotokolla. CMIP –protokollan etuna verrattuna SNMP –protokollaan oli sen mahdollisuus välittää parametrien eli muuttujien (variables) lisäksi myös tehtäviä (tasks), joita voitiin suorittaa verkkoelementissä. Vaikka CMIP –protokolla olikin kattavampi kokonaisuus verkonhallintaan kuin SNMP –protokolla, se ei kuitenkaan saavuttanut suurta suosiota. Syy tähän oli se, että CMIP –protokollan täydellinen implementoiminen vei verkkoelementiltä liikaa resursseja (erityisesti muistia).

Joidenkin laitteistojen hallinnassa käytetään myös HTTP –protokollaan ja pieniin palvelimiin (web server) perustuvaa mallia. Tässä hallinta voidaan hoitaa suoraan selainohjelmasta käsin ja liittynän laitteeseen tarjoaa sinne suoraan sijoitettu pienikokoinen HTTP –palvelin. Kyseinen palvelin tarjoaa mahdollisuuden hakea ja muuttaa haluttuja parametrejä laitteessa. Minimaalinen HTTP –palvelin ei vie paljoa tilaa ja tarjoaa kuitenkin riittävät ominaisuudet. Mikäli palvelimen koko ei ole kriittinen resurssi liikenne voidaan hoitaa salattuna esim. SSL –protokollan (Secure Sockets Layer) avulla. Kyseinen ratkaisumalli toimii hyvin tapauksissa, joissa ei ole tarvetta pitää yllä suurta määrää laitteita ja eikä tarvita keskitettyä tietokantaa laitteista ja niiden ominaisuuksista.

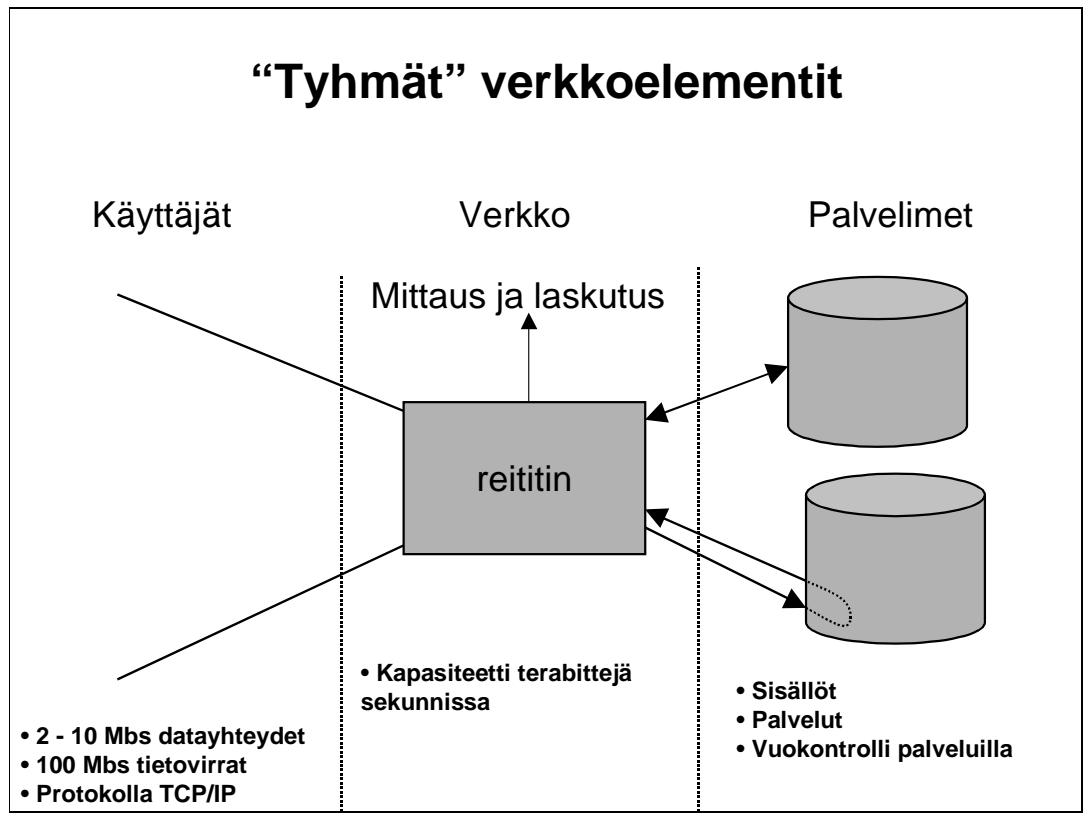
#### **5.4 Verkkoelementtien ja palveluiden kehitys tulevaisuudessa**

Palveluiden jakelu Internetissä voidaan tehdä kahdella toisistaan periaatteellisesti poikkeavalla tavalla. Arkkitehtuurisesti on mahdollista suunnitella ja toteuttaa palveluiden jakelu joko ns. tyhmiä verkkoelementtien kautta tai älykkäiden verkkoelementtien kautta. Jos verkkoelementit ovat tyhmiä, palveluiden logiikka ja kaikki älykkyys järjestelmässä sijaitsee palvelimella. Tällöin reitittimiltä vaaditaan suurta läpäisyä eli niiden tiedonsiirtokapasiteetti on luokkaa terabittejä sekunnissa. Loppukäyttäjillä tiedonsiirtoyhteydet ovat normaalisti luokkaa 2 – 10 megabittiä sekunnissa riippuen teknologiakehityksestä. Mikäli välitettävä data on datavirta (data stream, esim. liikkuvaa kuva) tyyppistä, on luultavasti käytössä vähintään 100 megabitin tiedonsiirtokaista. Tiedonsiirrossa käytettynä protokollana on todennäköisesti TCP/IP. Palvelut hallitsevat vuokontrollia ja reitittimet vain reitittävät datan oikeaan paikkaan.

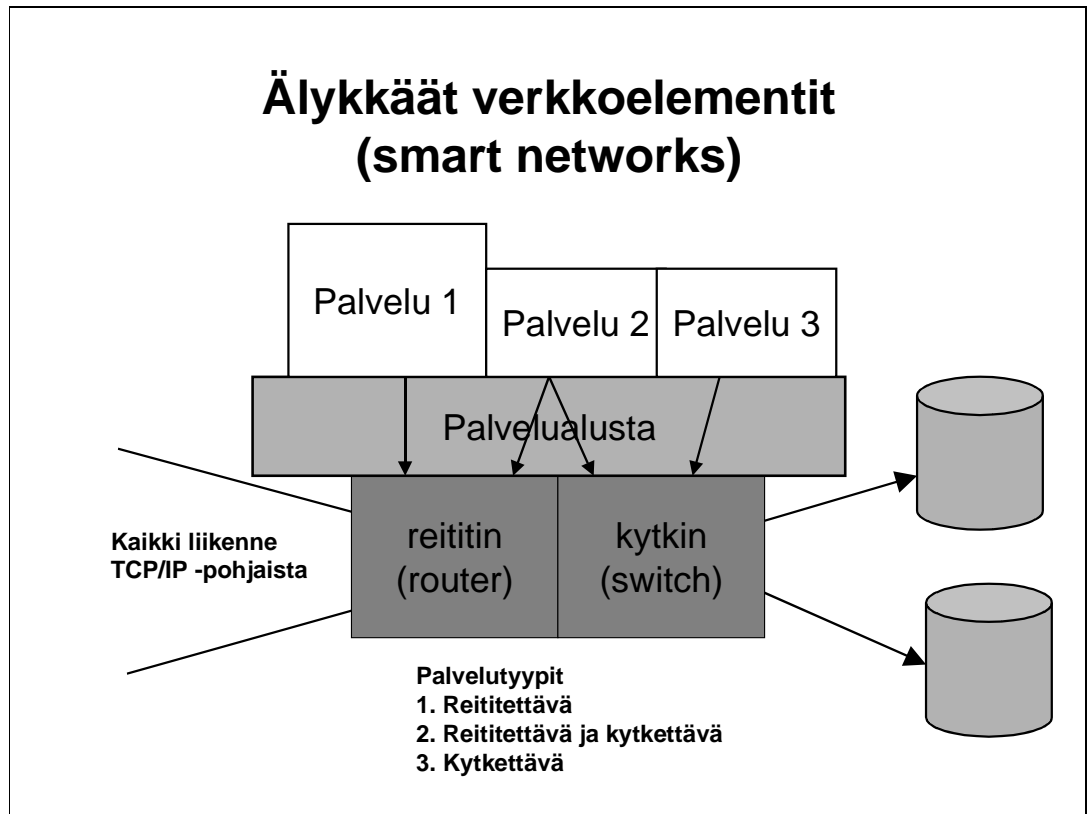
Mikäli taas käytetään älykkäitä verkkoelementtejä, osa älykkyudesta on toteutettu jo niissä ja palvelut toimivat palvelualustan päällä käyttäen hyväkseen palvelimilla sijaitsevia tietokantoja ja muita resursseja. Tällöin reitittimet luultavasti jakaantuvat kahteen ryhmään, eli runko- ja reunareitittimiin (edge router).

Runkoreitittimet ovat erittäin suorituskykyisiä ja lähinnä vain välittävät dataa. Reunareitittimet taas sisältävät enemmän älykkyyttä ja palvelut on toteutettu pääasiassa niissä.

Palvelulogiikka on toteutettu palvelualustan tarjoaman rajapinnan päälle ja palvelualusta tarjoaa palveluiden tuottamiseen peruskomponentit. Kyseinen malli on samankaltainen kuin nykyinen älyverkkoarkkitehtuuri. Palvelut voivat olla luonteeltaan joko reititystä tai kytkentää tarjoavia tai jopa edellisten yhdistelmää käyttäviä. Palvelun laadusta riippuen verkkoelementit joko reitittävät tai kytkevät välitetyn datan. Reititystä käytetään ei-reaaliaikaisissa palveluissa ja kytkentää, jos kyseessä on reaaliaikaisuutta tarvitseva palvelu kuten esimerkiksi liikkuva kuva. Kuvissa 24 ja 25 on kuvattu palvelunjakelun eri vaihtoehtoja tulevaisuudessa. Erityisesti palvelunjakeluun vaikuttavat ns. reunareitittimet (edge routers) ja niiden roolit (Lähde: Professori Olli Martikainen, Telecom Visions –esitelmä, 1999).



**Kuva 24:** Palvelunjakelu "tyhmien" verkkoelementtien kautta /10/.



**Kuva 25:** Palvelunjakelu älykkäitten verkkoelementtien kautta /10/.

## 5.5 Kokonaisuuden hallinnan ratkaisut

Kattavan palvelunhallintajärjestelmän eli OSS:n toteuttaminen Internet – ympäristössä on erittäin vaativa ja laaja tehtäväkenttä, sillä palveluiden, palvelualueiden ja laitteistojen monimuotoisuus tuo mukanaan useita ongelmia. Mikäli lisäksi halutaan hallintaan sisällyttää myös kaikki palvelunhallinnan alapuolella sijaitsevat komponentit, niin järjestelmän kompleksisuus moninkertaistuu. Toimivan hallintajärjestelmän toteuttaminen ilman olemassa olevia standardeja eri komponenttien välisistä rajapinnoista ja yhteyksistä on lähes mahdotonta. Tällä hetkellä kattavaan palvelunhallintaan ei vielä löydy standardeja eikä siitä johtuen valmiita tuotteitakaan, vaikka Javan J2EE -standardi on jo askel siihen suuntaan. Sekin on kuitenkin vielä suhteellisen kaukana kokonaisvaltaisesta palvelunhallinnan toteuttamisesta ja keskittyy pääasiassa palvelujen tuottamiseen.



Yksi syy palvelunhallinnan puuttumiseen onkin se, että itse palvelutkin ovat vielä suurelta osin kehityksen alla. Palvelunhallintaa on mahdotonta kehittää valmiiksi ennen kuin itse palvelut ja niiden toteutusteknologiat ovat selvillä ja vakiintuneita. On kuitenkin luultavaa, että mitä enemmän uudenlaisia palveluita ja palveluntarjoajia ilmestyy markkinoille, sitä suuremmaksi kasvaa paine palvelunhallinnan järjestelmällisestä hoitamisesta.

## 6 JOHTOPÄÄTÖKSET

### 6.1 Saadut tulokset

Tällä hetkellä palvelunhallinta on Internet –ympäristössä suhteellisen kypsymätöntä mutta tarve hyvin hoidetulle palvelunhallinnalle on selkeästi nähtävissä ja sen merkitys tulee kasvamaan tulevaisuudessa. Internetin kautta tarjottavat palvelut monipuolistuvat ja muuttuvat yhä enemmän sisältökeskeisiksi, jolloin varsinkin palveluiden sisällönhallinta kasvaa merkittäväksi tekijäksi. Sisällönhallinnasta muodostuu oma kokonaisuutensa, johon kehitetään kaupallisia siihen keskittyviä tuotteita. Poiketen esimerkiksi verkonhallintaan kehitetyistä tuotteista, jotka usein toimitetaan erilaisten järjestelmätoimitusten mukana, ne ovat itsenäisiä kokonaisuuksia ja riippumattomia (ainakin ideaalisessa tapauksessa) järjestelmätoimituksista.

Tällä hetkellä nykyisten Internet -palveluiden hallinta on tiukasti kytköksissä niiden toteutusteknologioihin eli niistä riippumatonta palvelunhallintaa ei oikeastaan vielä ole. Erilaisia hallintaominaisuuksia on jo mahdollista toteuttaa eri vaihtoehtojen avulla implementoiduille palveluille, mutta millään tuotteella ei ole valmiina kokonaisvaltaista, koko palvelunhallinnan kentän kattavaa järjestelmää. Palveluita voidaan toteuttaa esim. Javan J2EE -standardin mukaisten sovelluspalvelimien avulla. Palvelunhallinta näkökulman puuttuminen tai vaillinaisuus johtuu pääosin siitä, että kilpailukykyiset toteutusteknologiat ovat vielä varsin uusia. Varsinkin J2EE –standardi alkaa vasta nyt vakiinnuttamaan asemaansa ja CORBA on keskittynyt enemmän hajautetun oliopohjaisen infrastruktuurin luomiseen. Lupaavimmilta teknologioilta vaikuttavatkin juuri J2EE –standardin mukaiset sovellukset, vaikka myös ne vaativat vielä paljon kehitystä palvelunhallinnan näkökulmasta katsoen. J2EE –standardin vahvuutena on sen CORBA -yhteensopivuus ja Javan mukanaan tuomat edut. Nykyinen versio

siitä kuitenkin keskittyy pääasiassa palveluiden luontiin. Mikäli palveluita toteutetaan tulevaisuudessa yhä enemmän useilla eri toteutusteknologiolla, voi CORBA olla potentiaalinen vaihtoehto toteutettaessa palvelunhallintaa. Sen avulla sekä palvelut että palvelunhallintaan tarkoitetut järjestelmät voivat olla toteutuksiltaan heterogeenisiä ja kommunikoida keskenään standardoitujen rajapintojen kautta. Palvelunhallinnan kehittymisen kannalta on tärkeää, että hallintarajapinnat määritellään avoimiksi standardeiksi. Näin vältetään suljetuilta ja vain tietyn valmistajan tukemilta osittaisilta ratkaisuilta sekä varmistetaan järjestelmien yhteensopivuus.

## 6.2 Tulevaisuus

Tulevaisuudessa Internet –palvelut kehittyvät monipuolisemmiksi ja tarjonta laajenee, koska kasvava käyttäjäkunta tarvitsee uusia palveluja. Internetin käyttäjämäärien kasvu jatkuu ja se aiheuttaa paineita myös infrastruktuurin uudistamiselle. Sekä runko- että loppukäyttäjyhteyksien on oltava entistä nopeampia, jotta kasvava tiedonsiirtokapasiteetin tarve pystytään täyttämään.

Loppukäyttäjien modeemi- ja ISDN- yhteyksien merkitys vähenee niiden riittämättömän kaistanleveyden vuoksi. Nykyisen olemassa olevan parikaapeleihin pohjautuvan yhteyden korvaaminen uusilla vaihtoehtoilla loppukäyttäjille asti, kuten valokaapelilla, on suuressa mittakaavassa liian kallista. Valokaapeliyhteydet ovatkin pääasiassa runkoverkkojen ratkaisuja, suuren tiedonsiirtokapasiteettinsa vuoksi. Onkin oletettavaa, että yleisimpänä loppukäyttäjien siirtotienä säilyy parikaapeli mutta sitä hyödyntävät teknologiat muuttuvat. Eri DSL – teknologioihin pohjautuvat ratkaisut yleistyvät tarjoten yhä suurempaa kaistanleveyttä loppukäyttäjille. Kaupunkialueilla myös muunlaiset siirtotiet, kuten kaapeli verkkoon pohjautuvat ratkaisut, voivat yleistyä. Kiinteiden yhteyksien rinnalla langattomat yhteydet voivat tarjota kilpailukykyisen siirtotien, esimerkiksi tulevaisuudessa UMTS (Universal Mobile Telecommunications

System) tai nykyisessä GSM -verkossa toimiva GPRS (General Packet Radio Service) voivat tarjota riittäviä tiedonsiirtonopeuksia osalle loppukäyttäjistä. Langattomat teknologiat voivatkin olla jo lähitulevaisuudessa varteenotettava kilpailija fyysisille yhteyksille, mikäli ne vain hinnoitellaan kilpailukykyisesti.

Myös runkoverkkoyhteyksissä tulee tapahtumaan muutoksia. Itse siirtotienä tullaan tulevaisuudessa yhä enemmän käyttämään valokaapelia ja runkoverkon reitittimet toimivat optisesti. Optisella teknologialla voidaan taata tarpeeksi suuri tiedonsiirtonopeus, sillä tulevaisuudessa siirrettävät datamäärät moninkertaistuvat. Oletettavasti myös loppukäyttäjien lähellä eli runkoverkon reunoilla sijaitsevat reitittimet (reunareitittimet) muuttuvat. Niiden on pystyttävä tarjoamaan riittävä siirtonopeus (useita gigabittejä sekunnissa) palveluiden laadun takaamiseksi ja samalla niihin tullaan istuttamaan enemmän älykkyyttä nykyisiin reitittämiin verrattuna. Näin syntyy mahdollisuus luoda yhä monipuolisempia palveluja loppukäyttäjille.

Palvelunhallinnalle asetetaan tulevaisuudessa suuria vaatimuksia, koska sen on pystyttävä vastaamaan palveluiden käytön nopeaan kasvuun ja niiden monipuolistumiseen. Palvelunhallinnalta vaaditaan joustavuutta, jotta se pystyy mukautumaan nopeasti muuttuvassa ympäristössä ja lisäksi on pystyttävä takaamaan palveluiden laatu tilanteesta riippumatta. Palveluiden ja älykkyiden siirtyessä osaksi reunareitittämiin palvelunhallinnan on pystyttävä takaamaan riittävät hallintaominaisuudet, jolloin ratkaistavaksi nousee erityisesti hallintaan liittyvät tietoturvaongelmat.

## 7 YHTEENVETO

On oletettavaa, että palvelunhallinnan merkitys kasvaa tulevaisuudessa huomattavasti. Siihen johtavia tekijöitä ovat: palveluiden ja niiden käytön määrällinen lisääntyminen, niiden laadullinen parantuminen ja monipuolistuminen sekä tarve joustavalle palvelunhallinnalle. Palvelunhallintaan aletaan kiinnittää yhä enemmän huomiota, kun sen huomataan tuovan lisäarvoa myös liiketaloudellisessa mielessä. Palvelunhallinta on välttämätöntä, koska asiakkaille tarjottava palvelujoukko kasvaa ja palvelut ovat toteutuksiltaan ja ominaisuuksiltaan heterogeenisiä. Palveluiden laatuun ja toimivuuteen aletaan kiinnittää yhä enemmän huomiota ja sitä aletaan käyttää kilpailutekijänä. Palveluiden laatu onkin merkittävä menestystekijä Internetin kaltaisessa avoimessa verkossa, jossa asiakkaan on helppo vaihtaa palveluntarjoajaa. Järjestelmien heterogeenisyydestä johtuen on myös välttämätöntä, että palvelunhallintaa varten saadaan luotua kattavat avoimet standardit. Ilman niitä on vaarana, että itse palvelunhallinta pirstoutuu ja siitä ei saada maksimaalista hyötyä. Hallinnan toteutuskeinoina, kuten myös ehkä palveluiden toteutusteknologioina, voivat olla esim. Javan ja CORBA:n yhteiskäyttöön perustuvat mallit.

## LÄHTEET

- /1/ Internet Domain Survey. <http://www.isc.org/ds/>. 1999.
- /2/ Liikenneministeriö, Televiestintätalasto 1999 – Taulukot ja kuviot, <http://www.mintc.fi/www/sivut/suomi/telemarkkina/telemarkkina/tilasto/fig126.htm>.
- /3/ Fred Halsall, Data Communications, Computer Networks and Open Systems. 4<sup>th</sup> edition. Addison-Wesley, 1995. pp. 497. ISBN 0-201-42293-X.
- /4/ Huston, Geoff. ISP Survival Guide. New York: Wiley, 1999. pp. 85. ISBN 0-471-31499-4.
- /5/ Robert Orfali, Dan Harkey. Client Server Programming with JAVA and CORBA. 2<sup>nd</sup> edition. John Wiley & Sons, Inc. 1998. pp. 49. Figure 2-4. ISBN 0-471-24578-X. Kuvaa muokattu.
- /6/ Huston, Geoff. ISP Survival Guide. New York: Wiley, 1999. pp. 15. ISBN 0-471-31499-4.
- /7/ Liikenneministeriö, Internet palvelut, <http://www.mintc.fi/www/sivut/suomi/telemarkkina/telemarkkina/tilasto/net.htm>.
- /8/ Markku Sillanmikko, Linux and IBM -esitelmä, Helsinki 14.10.1999  
Linux@work seminaari.
- /9/ TeleManagement Forum. Performance Reporting Concepts & Definitions – TMF 701. Evaluation Version Issue 1.1. May 1999. pp. 17.
- /10/ Professori Olli Martikainen, Telecom visions -esitelmä. 1999.

/11/ Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns – Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994, ISBN 0201633612.

/12/ Robert Orfali, Dan Harkey. Client Server Programming with JAVA and CORBA. 2<sup>nd</sup> edition. John Wiley & Sons, Inc. 1998. pp. 339. Figure 15-5. ISBN 0-471-24578-X.

/13/ P. Emerald Chung, Yennun Huang, Shalini Yajnik, DCOM and CORBA Side by Side, Step by Step And Layer by Layer. [HTTP://www.bell-labs.com/~emerald/dcom\\_corba/Paper.html](http://www.bell-labs.com/~emerald/dcom_corba/Paper.html).

/14/ Elliotte Rusty Harold, Java Network Programming. O'Reilly & Associates. 1997. pp. 352. Figure 14-1. ISBN 1-56592-227-1.

/15/ Sun Microsystems, Inc. The Java Servlet API White Paper. <http://java.sun.com/products/servlet/whitepaper.html>.

/16/ Sun Microsystems, Inc. The Java Server Pages, White Paper. [http://java.sun.com/products/jsp/jsp\\_servlet.html](http://java.sun.com/products/jsp/jsp_servlet.html).

/17/ Sun Microsystems, Inc. ENTERPRISE JAVABEANS TECHNOLOGY – Server Component Model for the Java Platform. Modified illustration 2. [http://java.sun.com/products/ejb/white\\_paper.html#archit](http://java.sun.com/products/ejb/white_paper.html#archit).

/18/ Douglas C. Schmidt. Overview of CORBA. Figure 1. <http://www.cs.wustl.edu/~schmidt/corba-overview.html>.

/19/ Douglas C. Schmidt. Overview of CORBA. Figure 2. <http://www.cs.wustl.edu/~schmidt/corba-overview.html>.

/20/ Object Management Group. CORBA / IIOP 2.3.1 Specification. Figure 2-2. <http://www.omg.org/corba/corbaiiop.html>.

/21/ Robert Orfali, Dan Harkey. Client Server Programming with JAVA and CORBA. 2<sup>nd</sup> edition. John Wiley & Sons, Inc. 1998. pp. 8. Figure 1-2. Kuvan ulkonäköä muutettu ja palveluita ryhmitelty. ISBN 0-471-24578-X.

/22/ Robert Orfali, Dan Harkey. Client Server Programming with JAVA and CORBA. 2<sup>nd</sup> edition. John Wiley & Sons, Inc. 1998. pp. 375. Table 16-1. ISBN 0-471-24578-X. Taulukko suomennettu ja ulkonäköä muutettu.

/23/ Cisco Systems, Inc. ,IP Security --- IPsec, IPsec Features and Summary – taulukko suomennettuna.  
[http://www.cisco.com/warp/public/cc/cisco/mkt/ios/tech/security/prodlit/ipsec\\_ov.htm](http://www.cisco.com/warp/public/cc/cisco/mkt/ios/tech/security/prodlit/ipsec_ov.htm).

/24/ Sun Microsystems, Inc. Sun BluePrints Design Guidelines for J2EE,  
<http://www.javasoft.com/j2ee/blueprints/>.

/25/ Oracle Corporation. ORACLE and Java™ 2 Platform, Enterprise Edition. Sivu 8. [http://iseminar.oracle.com/user/ipatform/Developing\\_Java\\_2\\_Enterprise\\_Edition\\_Applications\\_on\\_Oracle.PDF](http://iseminar.oracle.com/user/ipatform/Developing_Java_2_Enterprise_Edition_Applications_on_Oracle.PDF)