

Kimmo Haukimäki

Mobiilien vertaisverkkojen väliohjelmistot

Tietotekniikan
kandidaatintutkielma
21. kesäkuuta 2004

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Tekijä: Kimmo Haukimäki

Yhteystiedot: kikahauk@cc.jyu.fi

Työn nimi: Mobiilien vertaisverkkojen väliohjelmistot

Title in English: Mobile Peer-to-Peer Middlewares

Työ: Tietotekniikan kandidaatintutkielma

Sivumäärä: 31

Tiivistelmä: Tutkielma johdattelee tämän hetkiseen mobiiliin vertaisverkko maailmaan. Siinä määritellään keskeisimmät termit aiheeseen liittyen sekä tutkitaan yhteensä viiden mobiilin vertaisverkko väliohjelmiston rakennetta ja ominaisuuksia.

English abstract: This thesis is an introduction to the current mobile peer-to-peer networks. The most relevant terms will be defined. Structures and features of five mobile peer-to-peer middlewares will be presented.

Avainsanat: Mobiili vertaisverkko, väliohjelmisto

Keywords: Mobile peer-to-peer, middleware

Sisältö

1	Johdanto	1
2	Käsitteitä	1
2.1	Vertaisverkko	1
2.2	Mobiili vertaisverkko	2
2.3	Väliohjelmisto	3
3	MOBY - väliohjelmisto	5
3.1	Arkkitehtuuri	5
3.2	Tietoturva	7
3.3	Palveluiden etsiminen	7
3.4	Resurssien hallinta	8
3.5	Järjestelmän Jini palveluiden luokat	8
4	Proem - väliohjelmisto	10
4.1	Kokonaisuudet	10
4.2	Identiteetit, nimet ja profiilit	11
4.3	Sovellusympäristö	11
4.4	Protokollat ja viestit	12
4.5	Peerlet ja Peerlet moottori	14
4.6	Palvelut	14
5	JXTA - väliohjelmisto	16
5.1	Arkkitehtuuri	17
5.2	Protokollat	17
5.3	JXTA mobiililaitteissa	18
5.4	Mobiililaitteiden tietoturva	19
6	PeerWare	20
6.1	Käsitteitä	20
6.2	Toimintamalli	20
6.3	Arkkitehtuuri	21
6.4	Tietoturva	23

7 BlueCheese	24
7.1 Toiminta	24
7.2 Palveluita	25
8 Yhteenveto	26
Viitteet	27

1 Johdanto

Vertaisverkkomallin käytön yleistymisen datan siirrossa ja palveluiden tarjoamisessa on nopeasti muokannut olemassa olevien järjestelmien rakennetta. Palveluiden tarjoaminen ilman keskitettyä jakajaa helpottaa tiedon siirtymistä verkoissa, koska tieto voidaan hakea monesta eri paikasta. Langattomien ja liikuteltavien laitteiden vertaisverkossa, eli mobiilissa vertaisverkossa on kaksi kriittistä ominaisuutta verrattuna langalliseen vertaisverkkoon. Eri mobiililaitteiden välillä alusta rakenne voi vaihdella, jolloin ei voida olettaa sovelluksen toimivuutta jokaisessa laitteessa. Tätä varten on joko tehtävä sovellus uusiksi tai tehtävä väliohjelmisto, joka yhdistää palveluita laitteeseen. Toinen eroavaisuus on ad-hoc ympäristö, jossa yhteydet ovat dynaamisia ja muuttuvat laitteiden liikkuesssa. Tämä tuo suunnitteluun haasteita, mutta se tarjoaa myös mahdollisuuksia, mm. tiedon vaihtamista ja palveluiden etsimistä laitteille määritellyssä paikassa. Esimerkiksi auton kuljettaja voi tilata kartan mobiilipalvelun avulla ja etsiä näin vaihtoehtoisen reitin juututtuaan ruuhkaan.

Tämä tutkielma on tehty olemassa olevien artikkeleiden pohjalta. Tarkoituksena on ollut kartoittaa nykyisten mobiilien vertaisverkon väliohjelmistojen tietämystä, joka voisi edesauttaa BlueCheese (kappaleessa 7) väliohjelmiston kehitystä. BlueCheese mobiili vertaisverkko väliohjelmisto on kehitetty Jyväskylän yliopistossa tietotekniikan laitoksessa [1] ja Agora Centerissä [2]. Tutkielman rakenne on seuraava. Kappaleessa 2 määritellään aiheeseen olennaisesti liittyviä termejä. Kappaleissa 3 - 7 esitellään viisi mobiilia vertaisverkko väliohjelmistoa (MOBY, Proem, JXTA, PeerWare ja BlueCheese). Tutkielman yhteenveto on esitetty kappaleessa 8.

2 Käsitteitä

Tässä kappaleessa esitetään määritelmät tutkielman keskeisimpien termien osalta.

2.1 Vertaisverkko

Hajautettua verkkoarkkitehtuuria kutsutaan vertaisverkoksi (Peer-to-Peer, P2P), mikäli jokainen verkon päätelaite jakaa omia resurssejaan (prosessori tehoa, muistitilaa, printtereitä, tiedostoja, ...) [3]. Verkon vertaisilla (peer) on suora yhteys muiden jakamiin resursseihin, ilman välilaitteita (servereitä). Vertaisverkon laite on samaan aikaan sekä palvelun tarjoaja että palvelun pyytäjä.

Vertaisverkko (Peer-to-Peer, P2P) on arkkitehtuurimalli, jossa kaikki päätelait-

teet ovat samanarvoisia verrattuna toisiinsa. Verkko muodostetaan laitteiden välille, joilla on samanlainen ohjelmisto käytettävissä. Vertaisverkko on keskittämätön rakenne, joka tarkoittaa että laitteet voivat suoraan kommunikoida keskenään, eikä viestien välittämiseen tarvita keskitettyä palvelinta.

Vertaisverkkomalli on vaihtoehtoinen malli perinteiselle asiakas-palvelin mallille. Asiakas-palvelin mallissa on keskitetty palvelin, joka tarjoaa palveluita asiakkaille. Tässä mallissa kaikki viestit kulkevat palvelimen kautta, jonne asiakkaan on rekisteröidyttävä ennen viestien välittämistä. Vertaisverkossa ei ole keskitettyä palvelinta, vaan jokainen laite toimii sekä palvelimena että asiakkaana. Tämä mahdollistaa paljon nopeamman tiedon leviämisen vertaisverkossa verrattuna asiakas-palvelin verkkoon, koska palvelimesta ei muodostu pullonkaulaa tietoliikenteelle.

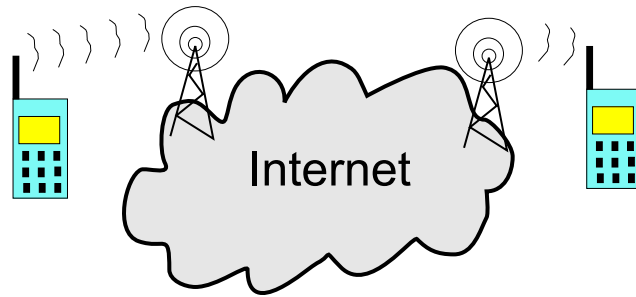
Hajautettu verkko on ns. puhdas vertaisverkko, mikäli se on vertaisverkko ja siitä voidaan poistaa mikä tahansa vertainen ilman että verkon palvelut tästä kärsivät [3]. Toisaalta hajautettu verkko on ns. hybridi vertaisverkko, mikäli se on vertaisverkko ja se vaatii keskitetyn laitteen osana palvelujen tarjontaa [3].

2.2 Mobiili vertaisverkko

Mobiili vertaisverkko (Mobile Peer-to-Peer, MP2P) on vertaisverkko, jossa vähintään yksi verkon vertaisista on mobiililaitte. Verkon rakenteelle on tunnusomaista dynaamisuus, koska verkko koostuu osittain tai kokonaan mobiileista solmuista. Mobiileilla vertaisverkoilla on kaksi rakennemallia: osittainen ja puhdas.

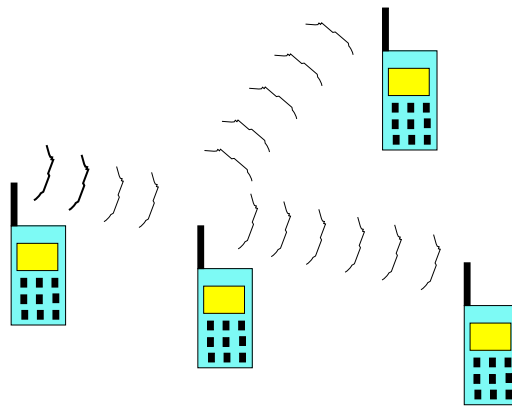
Vertaisverkko on ns. osittainen mobiili vertaisverkko, jos se koostuu sekä mobiileista laitteista että kiinteistä laitteista. Tällöin yksi tai useampi mobiililaitte on langattomasti yhteydessä kiinteän verkon laitteeseen. Tämä mahdollistaa palveluiden etsimisen kiinteän verkon välityksellä toiselta mobiililaitteelta. Tällöin palveluita voidaan käyttää huomattavasti laajemmalla alueella verrattuna puhtaaseen mobiiliin vertaisverkkoon. Osittaisen mobiilin vertaisverkon rakenne on esitetty kuvassa 1.

Vertaisverkko on ns. puhdas mobiili vertaisverkko, jos kaikki verkkoon kuuluvat vertaiset ovat mobiililaitteita. Mobiililaitteet ovat yhteydessä toisiinsa langattomalla siirtotekniikalla (mm. Bluetooth [4], WLAN [5]). Koska langattomien siirtotekniikoiden kantama on tyypillisesti hyvin pieni (alle 100m), on myös resurssien saatavuus usein hyvin rajallinen. Pieni kantama ja mobiililaitteiden liikuteltavuus tekevät verkosta erittäin dynaamisen. Nämä ovat rajoitteita, jotka tulee ottaa huomioon sekä ohjelmistosuunnittelijoiden että palveluiden käyttäjien. Esimerkiksi autojen kohda-



Kuva 1: Osittainen MP2P.

nessa valtiolla ei mobiililaitteella ehditä siirtää kovinkaan suurta datamäärää. Puh-
taan mobiilin vertaisverkon rakenne on kuvassa 2.

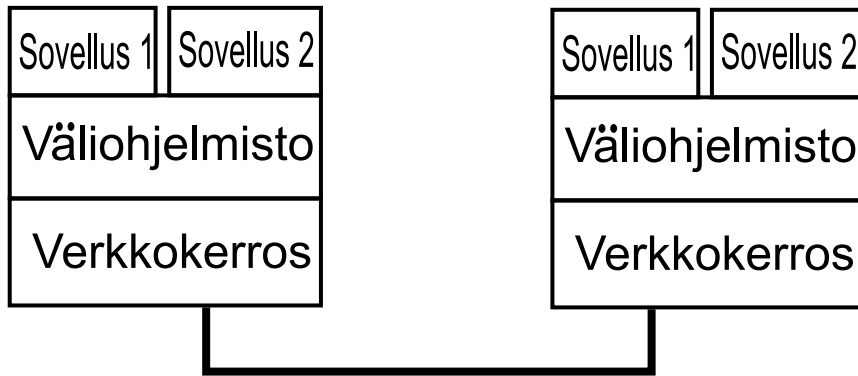


Kuva 2: Puhdas MP2P.

2.3 Väliohjelmisto

Väliohjelmisto (Middleware) on yhdistävä ohjelmisto, joka sisältää joukon palvelui-
ta, jotka mahdollistavat monen prosessin suorittamisen yhden tai useamman lait-
teen välillä verkon välityksellä [6]. Väliohjelmiston palvelut ovat ohjelmistojoukko-
ja, jotka ovat sovelluksen ja käyttöjärjestelmän tai sovelluksen ja verkkopalveluiden
välissä.

Väliohjelmisto sanan alkuperä hahmottuu helposti tutkimalla protokollapinin
rakennetta, jossa kerroksia on päällekkäin useampia. Protokollapinin väliohjelmis-
toksi sanotaan ohjelmistoa, joka on joidenkin kerrosten välissä, usein sovellus- ja
verkkokerroksen välissä. Kuva 3 havainnollistaa väliohjelmisto käsitettä.



Kuva 3: Väliohjelmisto.

Väliohjelmiston tarkoituksena on mahdollistaa sovellusten välinen tiedonsiirto. Väliohjelmisto piilottaa sovellukselta monimutkaiset toteutukset ja helpottaa näin sovellusohjelmoijaa. Väliohjelmistoon voidaan toteuttaa sellaisia toimintoja, joita tarvitsee moni sovellus, jolloin väliohjelmistosta saadaan yleiskäyttöinen, joka on yksi väliohjelmiston päätavoite.

Tyypillisiä väliohjelmistojen toimintoja ovat yhteyden muodostus, yhteyden purku, datan pilkkominen, datan kasaaminen, datan laadun tarkistus, datan eteenpäin välittäminen verkko- tai sovelluskerrokselle jne.

3 MOBY - väliohjelmisto

MOBY [7] on väliohjelmisto, joka pystyy tarjoamaan joukon palveluita (sovelluksia) tietyllä alueella. Jokainen asiakas toimii sekä datan lähteenä että välittäjänä ja tällä ratkaisulla jokainen mobiililaitte tuo lisäarvoa verkkoon.

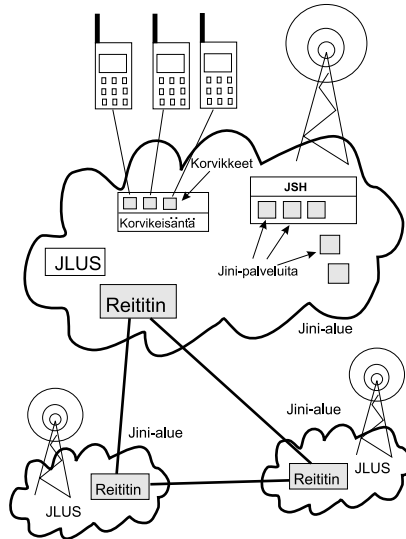
MOBY:n tavoitteena on täyttää seuraavat neljä ominaisuutta: (i) käyttäjän näkymättömyys (ii) läsnäolo (iii) sovellusten helppo lisääminen ja (iv) suorituskyky. MOBY:ssa on käytetty kahta teknologiaa tavoitteiden saavuttamiseen — Jini [8] ja Jini Technology Surrogate Architecture Specification. Jini tarjoaa palveluita sellaisille asiakkaille, joilla ei ole tietoa palveluista ja tavasta, jolla kommunikoidaan. Tämä tekniikka sopii langattomaan ad-hoc ympäristöön, jossa vertaiset liikkuvat paljon ja palveluiden on keskusteltava kapasiteettia vaihtelevien laitteiden kanssa. Surrogate architecture specification tekniikka on ratkaisu laitteille, jossa resurssit ovat rajoitettut, kuten kännykät. Laitte voi suorittaa palveluita korvikkeen avulla langallisessa laitteessa.

Lisäksi on muutama haaste, jotka pitää ratkaista. Java 2 Micro Editionista (J2ME) [9] puuttuu luokkien toimintojen kuormittaminen, joka tulee ottaa huomioon jos tarvitaan eri protokollia eri palveluille. Toinen haaste on palveluiden etsiminen Internetin yli. Myös luotettavuus, tietoturva, suorituskyky, palveluiden sijainti ja asiakkaan tunnistaminen ovat haasteita, jotka täytyy ratkaista. MOBY tarjoaa ratkaisut näihin kaikkiin haasteisiin.

3.1 Arkkitehtuuri

MOBY on arkkitehtuuriltaan osittainen mobiili vertaisverkko, jossa on käytetty hyödyksi Jini tekniikkaa. MOBY:n arkkitehtuuri on esitetty kuvassa 4. Arkkitehtuurin Jini alue koostuu seitsemästä osakokonaisuudesta:

1. Tukiasema (GSM-antenni), jonka kautta langaton laite (esim. kännykkä) on yhteydessä alueeseen.
2. Korvikeisäntä (Surrogate Host, SH) on kiinteän verkon osa johon langaton laite liitetään. Jokaiselle langattomalle laitteelle on oma korvikeisäntä.
3. Jinin etsimispalvelua (JLUS, Jini Lookup Service) käytetään palvelujen etsimiseen ja rekisteröimiseen.



Kuva 4: MOBY:n arkkitehtuuri.

4. Jini-palvelujen isännältä (Jini Services Host, JSH) voidaan ladata palveluja. JSH voi lisätä tai vähentää palveluja, joita on tarjolla.
5. Reitin (Mnode) on yhteydessä toisiin Jini alueisiin (Jini Domain).
6. Jini palvelu (Jini Service) on yksittäinen palvelu, jota JSH tarjoaa alueella.
7. Langaton laite (esim. kännykkä) on päätelaite, joka on tukiaseman välityksellä yhteydessä alueeseen.

Langaton laite muodostaa yhteyden tukiaseman kanssa tultuaan alueelle. Laitteelle lähetetään alustustietona korvike isännän IP-osoite ja porttinumero. Tämän jälkeen laite ajaa käynnistysohjelman, joka muodostaa yhteyden korvikeisännän kanssa. Laitteesta tulee näin verkon solmu (peer).

Korvikeisäntä käyttää Java 2 Platform Standard Editionia (J2SE) [10], joka mahdollistaa monilähetysviestien lähetyksen paikallisesti. Tämä tarkoittaa sitä, että korvikeisäntä voi etsiä palveluita Jini alueelta, mutta toisista alueista etsimiseen tarvitaan reitittimen apua. Reitittimen välityksellä pystytään etsimään palveluita mistä muusta tahansa alueesta.

3.2 Tietoturva

Tietoturvattomassa verkossa palvelun pyytäjä voi saada väärää informaatiota, joka johtaa laitteiden resurssien tuhlaamiseen. Tämän takia MOBY tarjoaa tietoturvallisia ja luotettavia palveluita. Palvelun pyytäjällä on mahdollisuus tutkia mistäpäin palvelun tarjoaja on. Käyttäjä saa palvelun pyynnön yhteydessä kuvauksen palvelusta sekä lähiverkon tunnisteiden, jonka perusteella voidaan tunnistaa luotettavat palvelut, koska luotettavuus perustuu oletukseen, että luotettavat lähiverkot tarjoavat vain luotettavia palveluita.

3.3 Palveluiden etsiminen

Palveluiden etsimisen tuloksena saadaan määritelmä palvelusta. Määritelmä sisältää kaksi erillistä tietoa. Ensimmäinen on varsinainen kuvaus palvelusta ja toinen on tieto siitä mistä palvelun voi ladata. Mnode säilyttää ja ylläpitää tietoa kaikista paikallisverkon palveluista, jolloin etsintäkysely voidaan suorittaa suoraan Mnodeelta.

Palveluiden etsiminen MOBY:ssa suoritetaan yleislähetysviestinä, jossa TTL (Time To Live) on kiinteän mittainen. Ensimmäisellä kysely yrityksellä TTL asetetaan pieneksi ja jos tuloksia ei löydy voidaan TTL:ää kasvattaa seuraavalla kysely yrityksellä. Hakutoiminnossa käytetään UDP-protokollaa, koska se ei tarvitse yhteyden muodostamista ja ylläpitoa, ja se on näin ollen nopeampi ja kevyempi kuin TCP. UDP ei kuitenkaan takaa viestien perille menoa, jolloin tuloksia voi kadota matkalla. MOBY palvelu ei takaa palvelujen kuvausten yksilöllisyyttä, mutta se takaa Mnode kuvausten yksilöllisyyden. Tämä tarkoittaa sitä, että eri palveluilla voi olla täsmälleen samat kuvaukset.

Asiakkaan ja palvelun välinen kommunikointi tapahtuu palvelun edustajan välityksellä. Palvelun edustajalla on viite varsinaiseen palveluun ja kommunikointi näiden välillä tapahtuu jonkin protokollan avulla. Protokolla on laite riippuvainen ja tämä vaatii palvelun tarjoajalta ladattavia protokollia eri laitteisiin.

MOBY:ssa on kolmen tyyppisiä palveluita: Yleiset Jini-palvelut, Solmupalvelut ja Järjestelmän Jini-palvelut.

Yleiset Jini-palvelut ovat palveluita, jotka ylläpitävät palveluiden rekisteröintitietoja. Näihin palveluihin kuuluu sekä palvelun käyttäjien tietojen ylläpito että palvelun tietojen ylläpito. Tietojen muuttaminen ei tapahdu automaattisesti vaan ainoastaan palvelukutsua käyttäen. Näiden palvelutietojen etsimisessä laitteen korvike etsii ensiksi paikalliselta alueelta. Mikäli etsintä ei tuota tulosta suoritetaan etsintä

muista alueista Mnodea käyttäen.

Solmupalvelut ovat palveluita, joita voidaan ladata Java Archive (jar) paketteina. Tämän palvelutyyppin rekisteröidyt palvelut löytyvät alueen JLUS-komponentista, jolloin laitteen korvike voi etsiä haluamaan palvelua suoraan JLUS-komponentista. Etsinnän tuloksena saadaan URL, josta palvelu voidaan ladata.

Järjestelmän Jini-palvelut ovat verkon palveluita. Näiden palveluiden tarkoituksena on muiden palveluiden hallinta. Nämä järjestelmän palvelut voivat poistaa jonkin palvelun käytöstä mikäli se ei ole tarpeellinen.

3.4 Resurssien hallinta

Resurssien hallinnan ongelma tulee eteen palveluiden jakamisessa Jini alueiden välillä. Yleisten Jini-palveluiden tapauksessa riittää että palvelun tarjoaja hyväksyttää palvelun MOBY-ylläpitäjällä. Tämän jälkeen riittää että palvelu välitetään Mnodele, jonka kautta se pääsee leviämään muihin Jini-alueisiin. Solmupalvelujen tapauksessa resurssien hallinta ei myöskään ole monimutkaista. Palvelu hyväksytetään jälleen MOBY-ylläpitäjällä, jonka jälkeen ladattava paketti varastoidaan, jolloin se on näkyvillä kaikille laitteille ja korvikkeille.

Järjestelmän Jini-palvelut sen sijaan vaativat erityiskäsittelyä. Näihin palveluihin kuuluu MOBY:n toiminnan kannalta tärkeitä toimintoja, kuten resurssien varaus palvelua varten ja resurssien vapautus palvelun poiston yhteydessä. Nämä toiminnot vaativat säännöllistä tarkkailua ja resurssien hallintaa. Järjestelmän Jini-palvelut pyrkivät asiakkaiden sijainnin ja tarpeen mukaan tehostamaan palvelun viiveen vaihtelua siirtämällä, monistamalla tai tuhoamalla palveluita.

3.5 Järjestelmän Jini palveluiden luokat

Tähän asti ollaan puhuttu yleisesti MOBY:n rakenteesta ja sen palveluista, joita tarjotaan Jini-alueella. Seuraavaksi ajatellaan palveluita mobiilien palveluiden näkökulmasta. Palvelut on ryhmitelty kolmeen luokkaan: sisällön sovittimet, tietokanta perustaiset ja välimuistilliset sijaintiin kiinnitetyt.

Sisällön sovitin luokkaan kuuluu sellaisia palveluita, jotka muokkaavat asiakkaan pyytämän palvelun tai tiedoston sellaiseen muotoon, että se on yhteensopiva asiakkaan laitteen kanssa. Esimerkki tällaisesta palvelusta on kuvatiedostoformaatin (JPG) muuttaminen toiseksi (PNG), jos laite ei tue jotain formaattia.

Tietokanta perustaisten luokkaan kuuluu palveluita, jotka tarvitsevat jatkuvasti

tietokantaa toimintojensa suorittamiseen. Esimerkiksi keltaisten sivujen hakupalvelussa tarvitaan joka haulla tietokantaa. Tällöin tietokanta ladataan tiettyyn Jini alueeseen ja palvelu siirretään sinne.

Kolmas luokka on välimuistilliset sijaintiin kiinnitetyt, joiden palvelut hyödyntävät saman alueen käyttäjien samoja tarpeita. Esimerkiksi karttapalvelu, jossa käyttäjät haluavat saada alueen kartan, on tällainen palvelu. Tällöin palvelu voi varastoida tiedon käyttäjien laitteisiin, jolloin joka kyselyllä ei tarvitse hakea karttaa tietokannasta. Palvelut eivät tällöin pysty takaamaan uusimman version saatavuutta, joten tämän luokan palvelut ovat sisällöltään hyvin stabiileja.

4 Proem - väliohjelmisto

Proem [11] on avoin sovellusalusta, joka mahdollistaa mobiilien vertaisverkko sovellusten kehittämisen ja sijoittamisen. Proemin ominaisuuksiksi on valittu olemassa olevien mobiilisovellusten yhteisiä ominaisuuksia, jolloin tuloksena on syntynyt yleinen sovellusalusta. Proemin tavoitteet ovat seuraavat:

- *Monikäyttöisyys* takaa erityyppisten vertaisverkkosovellusten lisäämisen Proemiin.
- *Yhteentoimivuus* sallii toimivuuden heterogeenisten laitteistojen ja sovellusten välillä.
- *Alusta riippumattomuus* mahdollistaa sovellusten suunnittelun ohjelmointikielystä, järjestelmästä ja verkosta riippumatta.
- *Sisäänrakennettu tietoturva* takaa sovelluksille yksityisyyden, luottamuksellisuuden ja siirretyn ja varastoidun datan oikeellisuuden.
- *Korkean tason kehitystuki* on merkittävin tavoite Proemissa ja se mahdollistaa yksinkertaisen, mutta tehokkaan kehitysalustan toteuttaa mobiileja vertaisverkkosovelluksia.

4.1 Kokonaisuudet

Proemin arkkitehtuuri määrittelee seuraavat neljä kokonaisuutta:

- **Vertainen** on itsenäinen mobiililaite, joka voidaan yhdistää vertaisverkko yhteydellä.
- **Yksilö** on henkilö, joka omistaa yhden tai useamman vertaisen. Vertainen on henkilökohtaisesti yksilön omistuksessa eikä sitä voida jakaa muiden yksilöiden kanssa. Eli yksilö voi omistaa useamman vertaisen, mutta vertainen kuuluu aina vain yhdelle yksilölle.
- **Tietovarasto** on joukko dataa, jotka ovat hallittavissa ja käytettävissä joukolle vertaisia.
- **Yhteisö** on joukko kokonaisuuksia (vertaisia, yksilöitä, tieto varastoja ja toisia yhteisöjä). Jokainen kokonaisuus voi kuulua moneen yhteisöön ja yhteisö voi

sisältää erityyppisiä kokonaisuuksia. Yhteisö voi asettaa oikeuksia dataan ja toiminnallisuuksiin. Esimerkkejä yhteisön muodostamisesta ovat seuraavat:

- Yksilö omistaa monta vertaista.
- Tietovaraston käyttöoikeudet on annettu tietylle projektille.
- Kaveriporukan muodostama yhteisö, johon ei muilla ole oikeuksia.

Yhteisö käsite eroaa perinteisestä hajautettujen järjestelmien ryhmä käsitteestä. Yhteisö on avoin joukko kokonaisuuksia. Jäsenyys ei ole hallittavissa ryhmän omistajan toimesta, mutta jäsenyys voidaan antaa jäsenen toimesta mille tahansa kokonaisuudelle. Jäsenyyden täydellinen hallinta on mahdotonta, koska yksi ei hallitse koko yhteisöä ja jäseniä voi liittyä milloin tahansa. Keskustellakseen yhteisön jäsenen kanssa täytyy jäsenelle lähettää salattu jäsenyysmerkki salakirjoitetussa muodossa. Jäsenyysmerkki on yksilöllinen yhteisölle ja yhteisöt ovat luotettavia alueita. Todistaakseen jäsenyys, kokonaisuuden täytyy muodostaa jäsenyysmerkki, joka on hyväksytty joukolla yhteisön jäseniä. Yhteisön jäsen voi joko hyväksyä tai hylätä jäsenyysmerkin todisteena jäsenyydestä.

4.2 Identiteetit, nimet ja profiilit

Kokonaisuus tunnistetaan nimen perusteella. Nimet on määritelty Uniform Resource Identifiers (URI) toimesta. Jokaisella kokonaisuudella voi olla useampi nimi. Nimi voi olla esimerkiksi *proem:peer:1100* tai *proem:dataspace:2133*. Proem ei sisällä keskittettyä nimivarastoa, joten ei voida tietää kuuluuko kaksi eri nimeä samalle kokonaisuudelle vai kahdelle eri kokonaisuudelle. Nimi on yksilöllinen ja se kuuluu vain yhdelle kokonaisuudelle.

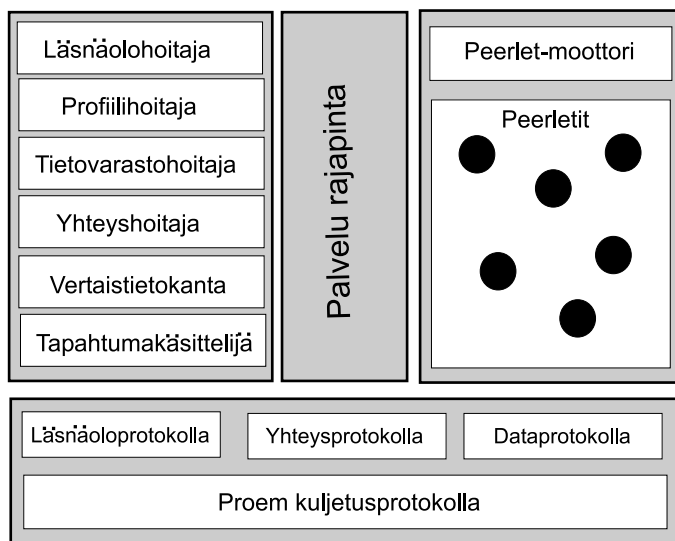
Proem mahdollistaa myös kokonaisuuksien määrittelyn muullakin tavalla kuin nimen avulla. Kokonaisuudelle voidaan määrittellä profiili, joka on tietoa kokonaisuudesta (meta-data). Profiili määritellään XML-tiedostolla, jossa voidaan kuvata esimerkiksi yksilöstä nimi, osoite, sähköpostiosoite tai vertaisesta mihin tietovarastoihin sillä on oikeudet. Profiilien avulla voidaan verkossa mainostaa kyseistä kokonaisuutta.

4.3 Sovellusympäristö

Proemin sovellusympäristö koostuu kahdesta pääkomponentista: Proem-kehitysympäristöstä ja Proem-suoritusjärjestelmästä. Proemin kehitysympäristö on tarkoitettu sovellus-

kehittäjälle ja se sisältää korkean tason rajapinnan nopeaan ja tehokkaaseen Proem sovelluskehitykseen. Proemin suoritusjärjestelmän arkkitehtuuri on esitetty kuvassa 5 ja se koostuu kolmesta komponentista:

- Protokollapino, joka koostuu neljästä protokollasta (lisää kappaleessa 4.4).
- Peerlet-moottori, joka hallitsee Peerlettejä (lisää kappaleessa 4.5).
- Palveluiden joukko, joka tarjoaa sovellukselle mekanismit mm. nimeämiseen, datan hallintaan, tapahtumien käsittelyyn ja käyttäjän tiedon hallintaan (lisää kappaleessa 4.6).



Kuva 5: Proemin arkkitehtuuri.

4.4 Protokollat ja viestit

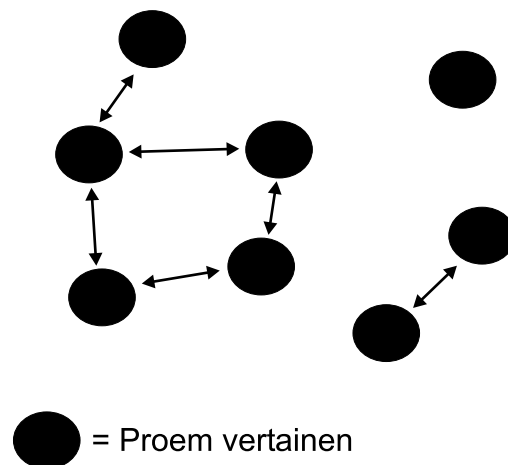
Proem voidaan ajatella joukkona toimintoja, joita sovellukset voivat käyttää, sekä alustana joka määrittelee viestin rakenteen, jonka avulla sovellukset voivat vaihtaa dataa keskenään. Määritelmä ja vertaisverkkoprotokollien käyttö takaavat yhteentoimivuuden eri laitteistojen ja sovellusten välillä.

Proem sisältää neljä protokollaa, yhden matalan tason kuljetusprotokollan ja kolme ylemmän tason protokollaa.

Proemin kuljetusprotokolla on yhteydetön asynkroninen yhteysprotokolla. Kuljetusprotokolla voidaan toteuttaa monen protokollan päällä, kuten TCP:n, UDP:n

tai HTTP:n. Data siirretään protokollalla yhdessä atomisessa yksikössä. Koska protokolla on epäluotettava, data voi kadota tai vahingoittua matkalla tai se voi saapua väärässä järjestyksessä perille. Protokollassa ei ole erikseen kuittausmekanismia pakettien saapumisen varmistamiseksi, mutta kuittaus voidaan ulkoisesti määrittellä protokollalle. Data muutetaan siirron ajaksi XML-muotoon, joka mahdollistaa vertaisten toteuttaa protokolla sellaiseksi kuin haluavat. Etenkin Proem-vertaiset voidaan toteuttaa millä ohjelmointikielellä tahansa.

Joukko Proem-vertaisia, jotka ovat yhteydessä toisiinsa Proemin kuljetusprotokollan välityksellä muodostavat Proem-verkon (Kuva 6).



Kuva 6: Proem verkko.

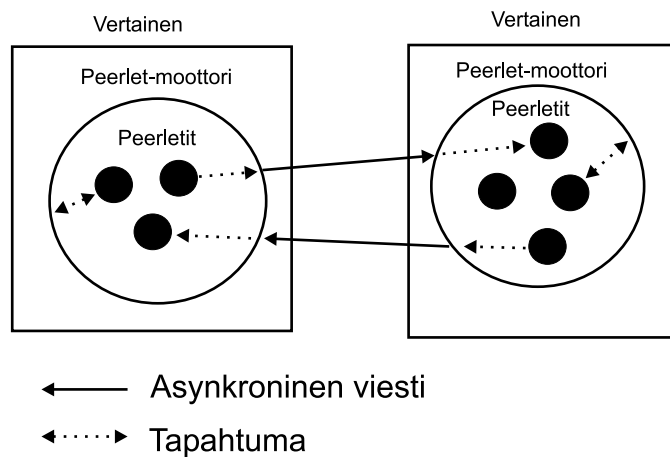
Proemin ydinprotokollat ovat:

- **Läsnäoloprotokolla** sisältää viestit, jotka mahdollistavat vertaisen ilmoittaa verkon yli olemassa olostaan ja saatavilla olevista kokonaisuuksista. Protokollan yleisin viestityyppi on profiilit.
- **Dataprotokolla** sisältää viestit, joiden avulla voidaan jakaa ja synkronoida dataa.
- **Yhteysprotokolla** sisältää viestit jäsenyyden hakemiseen, tarkistamiseen ja oikeuksien muuttamiseen.

Näiden protokollien lisäksi sovelluskehittäjä voi luoda omia protokollia laajentaakseen sovelluksensa toimintaa.

4.5 Peerlet ja Peerlet moottori

Peerlet on sovellus, joka on toteutettu tapahtumapohjaisesti Proemin vaatimalla tavalla. Peerlet-moottori on Peerlettejä hallinnoiva komponentti. Peerlet-moottori voi hallita useampaa Peerlettiä yhtä aikaa ja se on vastuussa Peerlettien rekisteröinnistä, suorittamisesta ja tuhoamisesta. Peerletin ja Peerlet-moottorin välinen yhteyden pito hoituu tapahtumia välittämällä. Peerlet voi tällä tavalla ilmoittaa tilansa muutoksista sekä lähettää tai vastaanottaa viestejä. Peerletit saavat ja käsittelevät tapahtumia asynkronisesti. Kuvassa 7 on esitetty Peerlettien, Peerlet moottorin ja vertaisten välinen yhteys viestien ja tapahtumien avulla.



Kuva 7: Peerlettien, Peerlet moottorien ja vertaisten välinen yhteys.

4.6 Palvelut

Proemin palvelut ovat toteutettu joukkona palvelukomponentteja:

- **Läsnäolohoitaja** on vastuussa ilmoittaa vertaisen läsnäolo sekä etsiä lähellä olevia toisia vertaisia.
- **Profiilihoitaja** ylläpitää vertaisen profiilitietoa.
- **Tietovaraston hoitaja** on vastuussa tietovarastoista ja oikeuksista niihin.
- **Yhteyshoitaja** pitää lukua vertaisten jäsenyyksistä yhteyden aikana ja suorittaa luotettavuustarkistuksia vertaisista, jotka ovat pyrkimässä yhteyteen.

- **Vertaistietokanta** ylläpitää tietoa muista vertaisista ja sallii Peerlettien tallettaa tietoa toisista vertaisista. Vertaistietokanta sisältää tiedot milloin mikäkin vertainen on viimeksi kohdattu.
- **Tapahtumakäsittelijä** sallii tapahtumapohjaisen yhteydenpidon järjestelmän komponenttien kesken, kuten myös Peerlettien. Menetelmä toimii julkaise-jatila tekniikalla, joka sallii datan vaihdon. Järjestelmä ja Peerlet-komponentit voivat ilmoittaa saatavilla olevasta datasta ja lähettää kiinnostuneille data hyväksymällä päivitystapahtuma. Tapahtumia käytetään myös ilmoittamaan Peerleteille vertaisten ilmaantumisesta ja poistumisesta.

5 JXTA - väliohjelmisto

JXTA [12] on vertaisverkkoteknologia, joka mahdollistaa kehittäjiä helposti luoda jaettuja ohjelmistoja. JXTA (engl. juxtapose) tukee myös mobiiliverkkoja, ja se sallii monen alustaltaan erilaisen mobiililaitteen ajamisen.

JXTA on alunperin kehitetty vertaisverkon väliohjelmistoksi ja myöhemmin siihen on lisätty tuki mobiililaitteille, joten se on rakenteeltaan osittainen mobiilivertaisverkko. JXTA sisältää vertaisverkkosovellusten kehittämistä varten seuraavat toiminnot:

- Vertaisten löytäminen verkosta palomuurien yli dynaamisen etsimisen avulla.
- Tiedostojen ja laskentaresurssien jakaminen verkossa.
- Vertaisryhmien muodostaminen, jossa vertaiset voivat löytää toisensa.
- Vertaisten toimintojen valvonta.
- Salattu yhteys verkossa toisten vertaisten kanssa.

JXTA määrittelee joukon omia termejä, joita se käyttää kokonaisuuden hahmottamiseen:

- **Vertainen** on mikä tahansa verkon laite, joka toteuttaa yhden tai useamman JXTA protokollista (kappaleessa 5.2).
- **Vertaisryhmä** on joukko vertaisia, joilla on yhteisiä kiinnostuksen kohteita.
- **Putki** on virtuaalinen suuntaamaton yhteys vertaisten välillä. Putket toimivat asynkronisesti ja erikoistapauksessa voivat olla liitettynä useampaan pääte-laitteeseen.
- **Mainos** on kuvaus, jostakin JXTA:n osasta (vertainen, vertaisryhmä, putki, palvelu, tms.).
- JXTA määrittelee erikoisvertaisia, jotka voivat varastoida mainoksia tai välittää vertaisten lähettämiä pyyntöjä tai molempia. Näitä erikoisvertaisia kutsutaan **kohtausvertaisiksi**. Kohtausvertaisia käytetään esimerkiksi mainosten etsimiseen toisista verkkosegmenteistä. Kohtausvertainen voi olla yhteydessä toiseen kohtausvertaiseen ja ne voivat välittää pyyntöjä toisilleen. Vertaiset oppivat kohtausvertaisten sijainnit bootstrap-algoritmeilla.

5.1 Arkkitehtuuri

JXTA arkkitehtuuri koostuu kolmesta kerroksesta: JXTA ytimestä, JXTA palveluista ja JXTA sovelluksista (Kuva 8).



Kuva 8: JXTA:n arkkitehtuuri.

JXTA ydin on vastuussa perustoiminnoista, kuten yhteys. Sen toimintoihin kuuluu myös yleiset tietoturvamenetelmät sekä JXTA:n protokollat. Ytimen ominaisuuksina on vertaisryhmien ja putkien hallinta, sekä vertaisten valvonta.

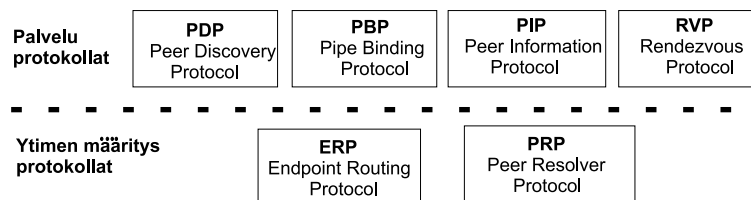
JXTA palvelut sisältää korkean tason toiminnallisuutta, kuten etsiminen, resursien jakaminen ja varastointi. JXTA palveluita tarvitaan alustariippumattomaan yhteistyöhön.

JXTA sovellukset on sovelluksia, jotka on liitetty JXTA väliohjelmistoon.

5.2 Protokollat

JXTA määrittelee yhteensä kuusi protokollaa [13]. Protokollat ovat jaettu kahteen kategoriaan: *ytimen määrittämisprotokollat* ja *palveluprotokollat*. Ytimen määrittämisprotokollia on kaksi: *ERP (Endpoint Routing Protocol)* ja *PRP (Peer Resolver Protocol)*, jotka vaaditaan toteuttamaan JXTA yhteensopivuuden saavuttamiseksi. Palveluprotokollia on neljä: *RVP (Rendezvous Protocol)*, *PDP (Peer Discovery Protocol)*, *PIP (Peer Information Protocol)* ja *PBP (Pipe Binding Protocol)*. Palveluluokan protokollien toteuttaminen ei ole pakollista, mutta on erittäin suositeltavaa suuremman yhteensopivuuden saavuttamiseksi muiden toteutusten kanssa. Kuvassa 9 on esitetty JXTA:n protokollapino.

ERP on protokolla, jonka avulla vertainen voi etsiä reittejä toisiin vertaisiin. Protokollan avulla voidaan myös etsiä reitti vertaiseen, johon joudutaan käyttämään toisia vertaisia välisolmuina. Vertaisverkoille on hyvin tyypillistä verkon topologian



Kuva 9: JXTA:n protokollapino.

muutokset, jolloin jo löydettyä reittiä ei voida enää käyttää. ERP osaa etsiä uuden reitin tilanteessa, jolloin vanhaa ei enää voida käyttää.

PRP protokollan avulla vertainen voi lähettää yleisiä kyselyviestejä ja saada niihin vastauksia. PRP protokollalla voi lähettää yleislähetysviestejä yhdelle tai useammalle käsittelijälle ryhmän sisällä ja sovittaa ne vastauksiin. Jokainen kysely on määriteltä käsitelijän nimellä, jonka perusteella kysely ja vastaukset tunnistetaan. Käsitelijän nimi ei viittaa yhteenkään vertaiseen.

RVP on protokolla, jonka avulla vertaiset voivat hyväksyä tai olla hyväksyjä levityspalvelussa. Vertaisryhmässä vertainen voi olla joko kohtauspaikkavertainen tai vertainen, joka kuuntelee kohtauspaikkavertaisia. RVP sallii viestien lähettämisen kaikille palvelua kuuntelevalle vertaiselle.

PDP protokolla mahdollistaa vertaisen julkaista omia mainoksiaan. PDP:llä vertainen voi etsiä muiden mainoksia sekä levittää omiaan toisille vertaisille. PDP käyttää PRP protokollaa viestien lähettämiseen ja levittämiseen.

PIP protokollalla vertainen voi vaihtaa tilatietoja (tila, liikennemäärä jne.) muiden vertaisten kanssa. PIP käyttää PRP protokollaa lähetykseen ja levitykseen.

PBP on protokolla, jolla vertainen voi luoda putkia yhden tai useamman vertaisen välille. Vertainen voi käyttää PBP:tä sitoakseen kaksi tai useampia putken päätä (sisäänmeno, ulostulo) fyysisen vertaisen päätepisteeseen. PBP käyttää PRP:tä putken sidontapyyntöjen lähetykseen ja levitykseen.

5.3 JXTA mobiililaitteissa

Osa JXTA:n protokollista mahdollistaa epätavallisen käytön niiden tarkoitukseen nähden. Esimerkiksi kiinteä vertainen voidaan synkronoida mobiilin vertaisen kanssa. Synkronointi tehdään SyncML [14] standardin mukaisesti, jolla voidaan laitteen sisällä synkronoida tietovarastoja, mutta myös laitteiden välinen synkronointi saadaan aikaan rakentamalla SyncML jonkun protokollan päälle. Tällä tavalla saavute-

taan vertaisverkon toiminnallisuus monessa laitteessa.

Mobiililaitteissa täytyy kuitenkin ottaa huomioon rajoittuneet resurssit, kuten alhainen laskentateho ja pieni kaistanleveys. JXTA käyttää XML:ä otsikoiden muodostamiseen ja se vaatii paljon laskentatehoa laitteelta, ja tämän takia JXTA:an tullaan tulevaisuudessa kehittämään protokolla, joka ei käytä XML:ä. JXTA ei kuitenkaan käytä säännöllisiä viestejä, kuten naapureiden etsiminen tai linkin tarkistus, ja tämä on erittäin käytännöllistä mobiililaitteille, koska se säästää energiaa ja kaistanleveyttä.

Paitsi että mobiililaitteissa on pieni kaistanleveys, niissä on myös rajallinen kantama. Langattomat tiedonsiirtotekniikat, kuten Bluetooth [4], WLAN [5] ja IrDA [15] ovat kantamaltaan hyvin heikkoja (n. 10-100m). Kohtausvertaisten avulla voidaan kuitenkin muodostaa yhteys mobiililaitteiden välille, jotka ovat suoran kantaman ulkopuolella. Tällöin kohtausvertainen asetetaan paikkaan, jossa se on molempien mobiililaitteiden ulottuvilla ja pystyy näin toimimaan viestien välittäjänä. Kohtausvertaisen täytyy kuitenkin tukea mobiililaitteiden ominaisuuksia pystyäkseen olemaan liittäjänä.

5.4 Mobiililaitteiden tietoturva

JXTA mahdollistaa yleisiä mekanismeja tietoturvaan JXTA:n ydinkerroksella. Se toteuttaa kattavan joukon perustietoturva ominaisuuksia tukemaan eri tietoturvaratkaisuja JXTA palveluille ja sovelluksille. Ne sisältävät seuraavat ominaisuudet:

- Hajautusfunktioita, symmetrisiä ja asymmetrisiä salausalgoritmeja.
- Oikeaksi tunnistamiskehys.
- Salasana - perustuen sisäänkirjautumiseen.
- Oikeuksien hallintamekanismi vertaisryhmiin perustuen.
- Datansiirron tietoturvamekanismi.

Kaikki nämä mekanismit perustuvat standardeihin ja yleisiin salaustekniikoihin ja -algoritmeihin, kuten RSA [16] julkisen avaimen vaihtoon ja RC4 [17] tavukoodiin.

6 PeerWare

PeerWare [18] on väliohjelmisto, joka on kehitetty tukemaan ryhmätyöskentely palveluita. Se on toteutettu mukautuvaksi, jotta se voidaan rakentaa yhteistyöllä yrityksen tarpeiden mukaiseksi. Koska ohjelmisto on tarkoitettu ryhmätyöskentelyyn on sovellusten tuettava suuren ihmisjoukon käyttöä yritysverkon tai Internetin välityksellä. PeerWare mahdollistaa dokumenttien ja jäsenten etsimisen organisaatiossa, kuten myös mahdollisuuden hyväksyä uusia tietoja. PeerWare on ohjelmisto etäryhmätyöskentelyyn, joten siihen on toteutettu myös liikkuvien ja langattomien laitteiden tuki. PeerWare on siis väliohjelmisto liikkuvaan ryhmätyöskentelyyn, joka on toteutettu osittaisen mobiilin vertaisverkon rakenteella.

6.1 Käsitteitä

Kappaleessa määritellään muutamia käsitteitä, jotka liittyvät PeerWaren toimintaan, ja ominaisuuksia, joita käytetään PeerWaressa.

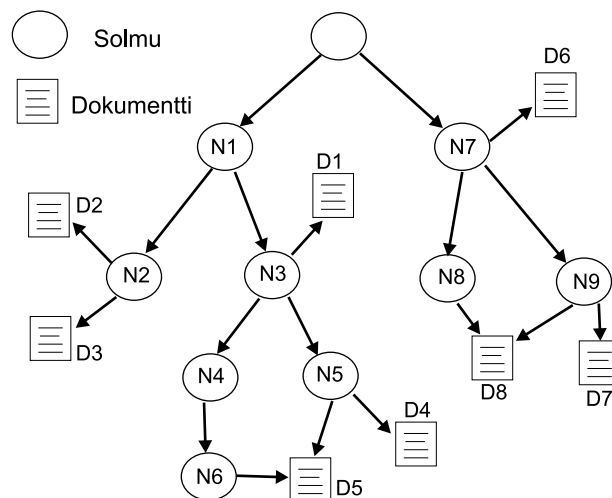
Koodin liikuteltavuus on mahdollisuus sijoittaa ajonaikana ohjelmistokomponentti uuteen paikkaan. Sovelluksen muokkausta varten vaadittava koodi voidaan kuljettaa tai se voidaan hakea ajonaikana.

GVDS (Global Virtual Data Structure) on yhteyden metamalli mobiiliin ympäristöön. Se sisältää tiedot alueella olevien mobiilien laitteiden paikoista ja jaetuista tiedostoista. Se on globaali, koska se sisältää kaiken datan, jotka yhteydessä olevat komponentit paikallisesti tarjoavat. Se on virtuaalinen, koska data ei fyysisesti sijaitse sen rakenteessa.

6.2 Toimintamalli

PeerWare käyttää toimintamallissaan hyväksi GVDS:ä. Kaikista yhteydessä olevista vertaisten tietovarastoista muodostetaan yhteinen hetkellinen jaettu tietorakenne. PeerWare:n hallitsema tietorakenne sisältää solmuja ja solmujen omistamia dokumentteja. Yhteen dokumenttiin voi olla omistus monella solmulla. Rakenne on havainnollistettu kuvassa 10.

Kun vertainen ei ole yhteydessä muihin verkon vertaisiin, sillä on mahdollisuus käyttää ainoastaan omia dokumentteja. Yhteyden muodostamisen jälkeen vertaisella on oikeus virtuaaliseen tietorakenteeseen ja kaikkiin muiden vertaisten jakamiin dokumentteihin. Kuvassa 11 on esimerkki miten GVDS tietorakenne muodostuu.



Kuva 10: Peerwaren tietorakenne.

PeerWaren [19] yksi tavoitteista on tehdä siitä mahdollisimman yksinkertainen. Tämän takia PeerWarena ei ole käytetty kuin kolmea GVDS:n toiminnoista:

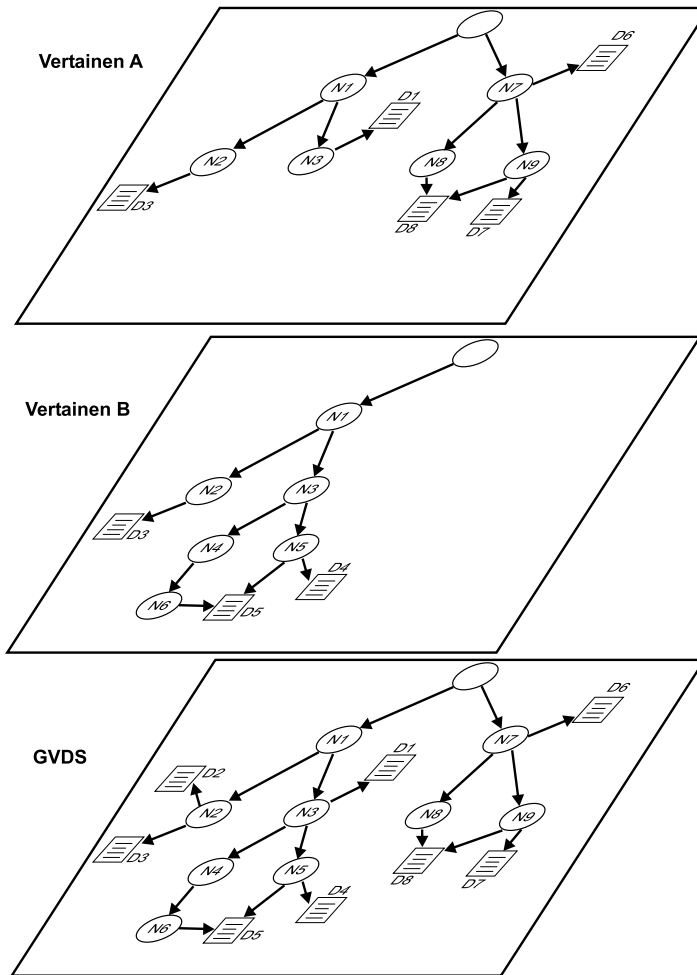
- **Suoritus**-toiminto sallii vertaisten suorittaa mielivaltaisen koodin osan yhteydessä olevan vertaisen valittavista osista.
- **Julkaisu**-toiminnon avulla vertainen voi ilmoittaa tapahtuman ilmaantumisesta.
- **Hyväksymis**-toiminnolla vertainen voi hyväksyä haluttuja tapahtumia itselleen.

Käyttämällä näitä toimintoja PeerWare mahdollistaa riittävät edellytykset tarpeelliselle toiminnallisuudelle. Vertaiset voivat lähettää kyselyjä, hyväksyä ja vastaanottaa tapahtumailmoituksia.

6.3 Arkkitehtuuri

PeerWarena käytettävät GVDS:n toiminnot vaativat reititystä viestien levittämiseen. Malli ei kuitenkaan määrää miten reititetään esim. mitä algoritmia tulee käyttää ja onko verkon rakenteella ennalta määrätty muoto.

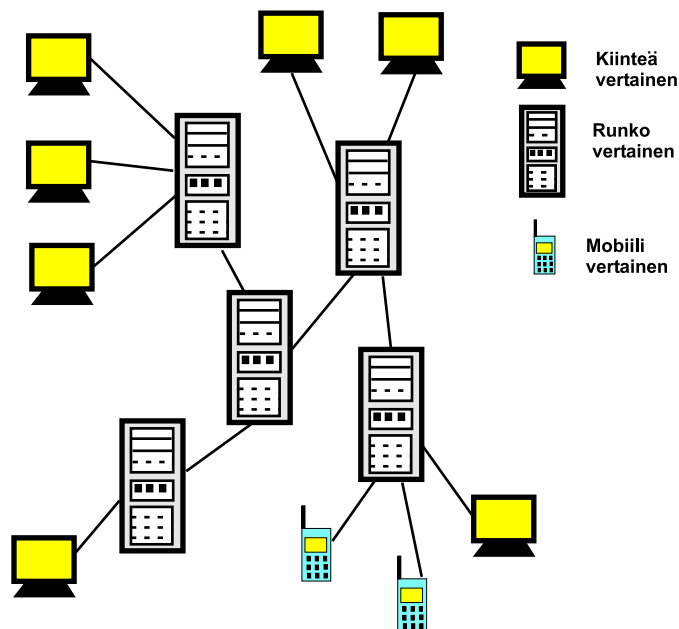
Tämä on mahdollistanut kehittää PeerWaren kaksi erilaista toteutusta. Ensimmäinen on suunnattu kiinteään verkkoon, jossa yhteydet on määritetty selkeästi



Kuva 11: GVDS rakenteen muodostaminen PeerWaressa.

sisään- ja uloskirjautumisella. Toinen toteutus on suunnattu ad-hoc verkkoja varten, jossa yhteydet muodostetaan loogisen kirjautumisen sijaan fyysisellä yhteydellä. Kumpikin toteutus on kuitenkin samanlainen arkkitehtuuriltaan, jossa verkko muodostuu kiinteistä runkovertaisista ja niihin kiinnittyneistä mobiileista tai kiinteistä vertaisista. Rakenne on havainnollistettu kuvassa 12.

Nämä kaksi eri toteutusta eroavat toisistaan varsinkin reititysalgoritmin suhteen. Kiinteässä verkossa, joka on rakenteeltaan stabiili, voidaan reititys tehdä hierarkiaan perustuen. Tämä on erittäin tehokas tapa kiinteälle verkolle, mutta ad-hoc verkoissa rakenne on dynaaminen ja paikallinen hierarkia tietämystä ei välttämättä ole ajan tasalla.



Kuva 12: PeerWare arkkitehtuuri.

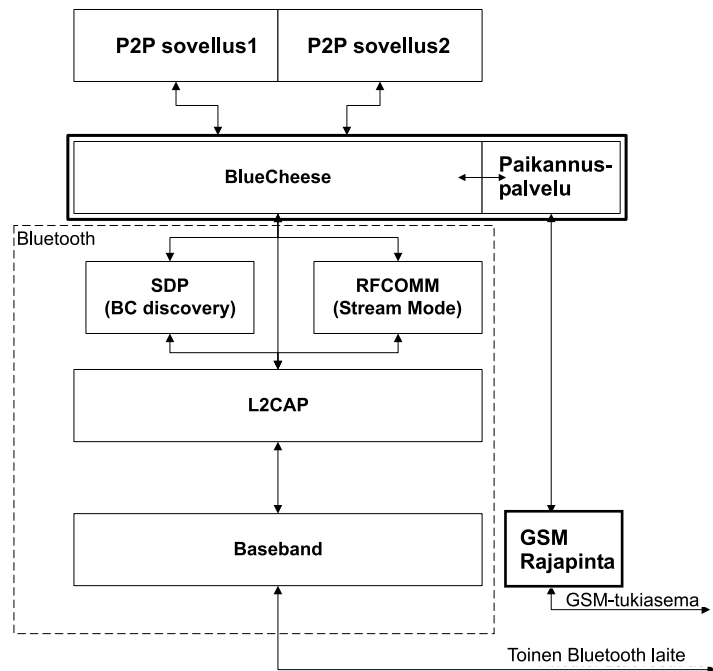
6.4 Tietoturva

Kulunvalvonta ja tietoturva ovat erittäin tärkeitä ominaisuuksia alueella, johon PeerWare on suunnattu. PeerWarella on kaksi mekanismia tietoturvan takaamiseen. Toisessa voidaan muodostaa vertaisten välille salattuja yhteyksiä ja tarkistaa vertaisten oikeellisuus. Toisessa mekanismissa vertaisille voidaan määrittää tarvittavia oikeuksia.

Jotta PeerWare olisi joustava ympäristö, on sen tietovarasto ja tietoturva komponentit toteutettu erillisinä. Näin on saatu ympäristö, joka voidaan helposti muokata erityisten tarpeiden mukaan.

7 BlueCheese

BlueCheese [20] on mobiilin vertaisverkon väliohjelmisto, joka tukee täydellisen mobiilin vertaisverkon rakennetta. Se tarjoaa sovelluksille joukon palveluita, joita ne voivat käyttää rekisteröitymisen jälkeen. BlueCheese mahdollistaa yhden tai useamman sovelluksen ajamisen samanaikaisesti. Datansiirto tekniikkana käytetään Bluetoothia [4], jonka protokollista SDP:tä (Service Discovery Protocol) käytetään palveluiden etsimiseen ja L2CAP:ia (Logical Link Control and Adaptation Protocol) datan siirtoon. Yhtenä palveluna on paikannuspalvelu, jonka informaatio perustuu GSM verkkorakenteeseen. Kuvassa 13 on esitetty BlueCheesen sijoittuminen Bluetoothiin ja sovellusten väliin.



Kuva 13: BlueCheesen arkkitehtuuri.

7.1 Toiminta

BlueCheesen ja sovellusten välinen keskustelu tapahtuu tapahtumien avulla. Sovelluksen kutsuessa tarjolla olevaa palvelua, funktiokutsu muutetaan tapahtumaksi, joka käsitellään heti kun siihen on aikaa. Vastaavasti sovellus voi saada väliohjelmistolta viestejä tapahtumien avulla, kuten saapunut data tai aukaistu yhteys. Jot-

ta tapahtumapohjainen keskustelu voidaan tehdä, täytyy sovellukseen toteuttaa tapahtumankäsittelyluokka.

Käyttäkseen palveluita sovelluksen on ensiksi rekisteröidyttävä BlueCheeseen. Rekisteröityminen tapahtuu käyttämällä siihen tarkoitettua palvelua, joka vaatii tiedon tapahtumankäsittelijä luokasta sekä sovellukselle määritellyn muuttumattoman nimen. Nimestä BlueCheese generoi sovellukselle tunnistenumeron, jonka avulla sovellus tunnistetaan. Tunnistenumero on sovellukselle aina sama, jolloin sen perusteella voidaan tarkastella mitä sovelluksia laitteesta löytyy.

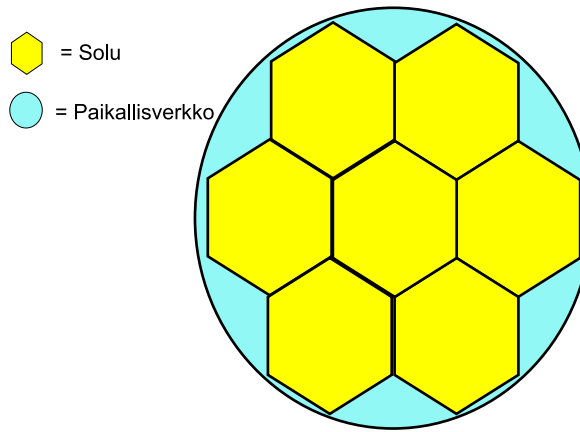
BlueCheese mahdollistaa yhteyden muodostamisen ja datansiirron vain yhden BlueCheese laitteen kanssa kerrallaan. Laite asetetaan jonoon odottamaan palveluvuoroa, mikäli ne yrittävät yhteyttä jo yhteydessä olevaan laitteeseen. Jonossa olevaa laitetta palvellaan, kun aikaisempi yhteys on purettu.

Yhteyden muodostuksen yhteydessä BlueCheese laitteet vaihtavat tiedot siitä mitä sovelluksia laitteeseen on rekisteröity. Sellaisille sovelluksille, jotka sijaitsevat molemmissa laitteissa, ilmoitetaan tapahtumalla, että yhteys on muodostettu ja datan siirto on mahdollista. Tämän jälkeen on sovelluksen vastuulla siirtävätkö ne dataa. Jotta BlueCheese osaa katkaista yhteyden toiseen laitteeseen, täytyy sovelluksen ilmoittaa BlueCheeselle milloin se on lopettanut datan siirron.

7.2 Palveluita

Paikannuspalvelu on yksi BlueCheesen tarjoamista palveluista. Se mahdollistaa sovelluksen tiedustella nykyistä sijaintia. Sijaintitieto perustuu GSM-verkkorakenteeseen. GSM-verkko muodostuu paikallisverkoista, jotka sisältävät soluja (Kuva 14). Tällä tiedolla sovellus voi vertailla eri sijaintitietojen etäisyyttä karkeasti tai pyytää BlueCheesea tekemään tämän vertailun. Vertailun tuloksena saadaan jokin neljästä arvosta:

- **Kaukana**, jos laitteet eivät ole samassa paikallisverkossa.
- **Aika lähellä**, jos laitteet ovat samassa paikallisverkossa, mutta eivät samassa solussa.
- **Hyvin lähellä**, jos laitteet ovat samassa solussa.
- **Ei tietoa**, jos vertailua ei voida tehdä esimerkiksi eri operaattoreiden takia.



Kuva 14: GSM verkkorakenne.

8 Yhteenveto

Vertaisverkkojen suosion myötä on alettu kehittää mobiileja vertaisverkko rakenteita. Mobiilisuus luo haasteita kehittäjille sen dynaamisuuden johdosta, mutta samalla se tuo monia hyödyllisiä ominaisuuksia sovellusten käyttäjille, kuten tiedon vaihtuvuus. Mobiili vertaisverkkomalli on tulevaisuuden tekniikka, joka ei vielä kuitenkaan ole vakiinnuttanut paikkaansa mobiililaitteiden tekniikkana. Tätä mallia kehittäjät ovat lähestyneet väliohjelmiston näkökulmasta, jonka avulla pystytään luomaan pohja monelle sovellukselle.

Tämän tutkimuksen kappaleissa 3 - 7 esiteltiin yhteensä viisi mobiilin vertaisverkon väliohjelmistoa (MOBY, Proem, JXTA, PeerWare ja BlueCheese) ominaisuuksiineen. Kolme näistä on rakenteeltaan osittaisia mobiileja vertaisverkkoja (MOBY, JXTA ja PeerWare), yksi on puhdas mobiili vertaisverkko (BlueCheese) ja yhden rakenteesta ei ole täyttä varmuutta (Proem). Todennäköisesti Proem on osittainen MP2P, koska se on rakennettu IP:n päälle eikä yhteysteknologiaan ole otettu kantaa.

Lukuunottamatta PeerWarea, joka räätälöidään asiakkaan tarpeiden mukaan, kaikkien esiteltyjen väliohjelmistojen yhtenä päätavoitteena on sovellusten helppo lisääminen siihen. Rakenteiden kehittämisen lähtökohdat ovat olleet väliohjelmistoilla erilaisia. Tyypillisesti osittaisten MP2P-väliohjelmistojen kehitys on aloitettu kiinteän vertaisverkon pohjalle, esimerkiksi JXTA on alunperin kiinteä P2P ohjelmisto ja MOBY on rakennettu Jini-tekniikan päälle. Vastaavasti puhtaasti MP2P-verkon kehityksessä ei ole valmista pohjaa, josta voitaisiin aloittaa.

Osittaisen MP2P:n etuja puhtaaseen MP2P:hen verrattuna on tiedon haun laa-

juus, kun tietoa voidaan etsiä kiinteästä verkosta ympäri maailmaa. Puhdas MP2P-tekniikka sopii puolestaan tilanteisiin, jossa kiinteään verkkoon ei ole mahdollista liittyä tai tiedon vaihtaminen pitää tapahtua erittäin nopeasti. Tulevaisuuteen jää kuitenkin nähtäväksi kumman tyyppiset MP2P väliohjelmistot tulevat saavuttamaan suuremman suosion, vai pystytäänkö kehittämään rakenne joka toimii hyvin molemmissa ympäristöissä.

Viitteet

- [1] *Jyväskylän yliopisto - tietotekniikan laitos*, saatavilla WWW-muodossa <URL: <http://www.mit.jyu.fi>>, 16.6.2004.
- [2] *Agora Human Technologies Center, Agora Center*, <URL: <http://www.jyu.fi/agora/>>, 16.6.2004.
- [3] R. Schollmeier, *A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications*, julkaistu First International Conference on Peer-to-Peer Computing (ICPPC'01) konferenssissa, 2001, s. 101-102.
- [4] *Bluetooth, The Official Bluetooth Wireless Info Site*, <URL: <http://www.bluetooth.com>>, 10.05.2004.
- [5] *Wireless LAN Community*, <URL: <http://www.wlan.org>>, 06.05.2004.
- [6] *Software Engineering Institute (SEI), Middleware*, saatavilla WWW-muodossa <URL: <http://www.sei.cmu.edu/str/descriptions/middleware.html>>, 9.1.2004.
- [7] T. Horozov, A. Grama, V. Vasudevan ja S. Landis, *MOBY - A Mobile Peer-to-Peer Service and Data Network*, julkaistu International Conference on Parallel Processing (ICPP'02) konferenssissa, IEEE 2002.
- [8] *Sun Microsystems, Jini Architecture Specification*, saatavilla PDF-muodossa <URL: http://www.sun.com/software/jini/specs/jini2_0.pdf>, kesäkuu 2003.
- [9] *Sun Microsystems, Java 2 Platform Micro Edition*, saatavilla WWW-muodossa <URL: <http://java.sun.com/products/j2mewtoolkit>>, 26.11.2003.

- [10] Sun Microsystems, *Java 2 Platform, Standard Edition (J2SE)*, saatavilla WWW-muodossa <URL: <http://java.sun.com/j2se>>, 25.8.2003.
- [11] G. Kortuem, *Proem: A Peer-to-Peer Computing platform for Mobile Ad-hoc Networks*, Oregonin Yliopisto.
- [12] N. Maibaum, T. Mundt *JXTA: A Technology Facilitating Mobile Peer-To-Peer Networks*, julkaistu International Mobility and Wireless Access Workshop (MobiWac'02) konferenssissa, IEEE 2002.
- [13] B. Traversat, A. Arora, M. Abdelaziz, M. Duigou, C. Haywood, J-C. Hugly, E. Pouyoul, B. Yeager, *Project JXTA 2.0 Super-Peer Virtual Network*, Sun Microsystems, Inc., saatavilla PDF-muodossa <URL: <http://www.jxta.org/project/www/docs/JXTA2.0protocols1.pdf>>, 25.3.2003.
- [14] Open Mobile Alliance, OMA Technical Section — Affiliates — SyncML, <URL: <http://www.syncml.org>>, 25.5.2004.
- [15] IrDA, The Infrared Data Association — IrDA, <URL: <http://www.irda.org>>.
- [16] Wikipedia, *RSA* — *Wikipedia, the free encyclopedia*, saatavilla WWW-muodossa <URL: <http://en.wikipedia.org/wiki/RSA>>, 11.6.2004.
- [17] Wikipedia, *RC4(cipher)* — *Wikipedia, the free encyclopedia*, saatavilla WWW-muodossa <URL: http://en.wikipedia.org/wiki/RC4_%28cipher%29>, 29.5.2004.
- [18] G. Cugola, G-P. Picco, *PeerWare: Core Middleware Support for Peer-to-Peer and Mobile Systems*, Milano, Italia, 2001.
- [19] European Research Consortium for Informatics and Mathematics, ERCIM News, saatavilla PDF-muodossa <URL: http://www.ercim.org/publication/Ercim_News/enw54/EN54.pdf>, Heinäkuu 2003, s. 19-21.
- [20] O. Alanen, K. Haukimäki, T. Juonoja, P. Rönkkö, *BlueCheese*, <URL: <http://kotka.it.jyu.fi/mopedi>>, tammikuu 2004.