Luottamuksenhallinta web-palveluympäristössä

Sini Ruohomaa

Helsinki 29.4.2005

Pro gradu -tutkielma

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Tiedekunta/Osasto — Fakultet/Sektion — Faculty
Matemaattis-luonnontieteellinen tiedekunta

Laitos — Institution — Department
Tietojenkäsittelytieteen laitos

Tekijä — Författare — Author
Sini Ruohomaa

Työn nimi — Arbetets titel — Title
Luottamuksenhallinta web-palveluympäristössä

Oppiaine — Läroämne — Subject
Tietojenkäsittelytiede

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Pro gradu -tutkielma	29.4.2005	68 sivua

Tiivistelmä — Referat — Abstract

Tutkielmassa käsitellään luottamuksenhallintaa web-palveluympäristössä. Dynaaminen toimintaympäristö asettaa vaatimuksia luottamuksenhallintajärjestelmälle, jota käytetään paitsi paikallisten pääsynhallintapäätösten tekemiseen, myös laajemman mittakaavan päätöksenteon tukena, useiden autonomisten toimijoiden muodostamien yhteisöjen hallinnassa. Tutkielma esittelee Trust Based on Evidence -projektissa kehitetyn luottamuksenhallintajärjestelmän tiedollisen ja toiminnallisen mallin, paikallisesta ja yhteisön näkökulmasta. Mallia selkeytetään web-palveluympäristöön sijoittuvan esimerkin avulla. Luottamuksen käsitteen rakentamiseksi esitellään myös eri osa-alueille sijoittuvia luottamuksen malleja ja luottamusta käyttäviä järjestelmiä.

Avoimessa verkkoympäristössä palveluntarjoaja joutuu tasapainottelemaan kahden osin vastakkaisen tavoitteen välillä: toisaalta järjestelmän tulisi olla mahdollisimman avoin, jotta se houkuttelisi käyttäjiä, toisaalta liiallinen avoimuus kasvattaa tietomurron riskiä. Kompromissin löytäminen on hankaloitunut edelleen saavutettavien käyttäjien määrän kasvaessa ja tarjottavien palvelujen monimutkaistuessa. Tehtävä vaatii toisaalta erikoistapauksien käsittelyä, toisaalta yleistettävyyttä laajan käyttäjistön suhteen.

Tietoturvan ylläpidon automatisointia ovat edistäneet muun muassa politiikkapäätösten erottaminen toteutuksesta ja mahdollisten tietomurron merkkien tarkkailun delegointi siihen erikoistuneille ohjelmille (IDS). Palvelujen käyttäjistön kasvaessa ja siirtyessä nimettömämmiksi kurinpito ja tarkkailu kuitenkin vaikeutuvat entisestään, eikä ylläpitäjiä riitä sidottavaksi jatkuvaan käyttäjien vahtimiseen. Monesti valvoja voikin vain poistaa käyttöoikeuden häiriköltä, jolloin esimerkiksi hieman lievemmälle sääntöjen “venyttämiselle” ei juuri voi tehdä mitään.

Luottamuksenhallinta helpottaa rikkomuksiin ja toisaalta hyvään käytökseen reagoimista asteittain. Sen pohjalta käyttäjien valvontaan, pääsynhallintaan ja resurssien rajoitukseen liittyvä hienosäätö voidaan tuoda ymmärrettäväksi osaksi ylläpitoa ja pitkälti myös automatisoida.

ACM Computing Classification System (CCS):
H.5.3 [Information Interfaces and Presentation: Group and Organization Interfaces]
K.4.4 [Management of Computing and Information Systems: Electronic Commerce]
K.6.5 [Management of Computing and Information Systems: Security and Protection]

Avainsanat — Nyckelord — Keywords
luottamuksenhallinta, Web Services

Säilytyspaikka — Förvaringsställe — Where deposited
Kumpulan tiedekirjasto, sarjanumero C-

Muita tietoja — övriga uppgifter — Additional information

Sisältö

Liitteet

1 Johdanto

Teemme päätöksiä mieluusti mahdollisimman tarkan tiedon pohjalta. Kun tietoa ei ole saatavilla, joudumme tekemään olettamuksia. Näiden olettamusten kohdistuessa toisten ihmisten motiiveihin ja toimintaan puhutaan joskus luottamuksesta, jonka negatiivinen vastine on epäluottamus. Luottamukseen liittyy vahvasti henkilökohtainen riskinotto olettamuksen perusteella; puhumme 'luottamuksesta' yleisemmän 'oletuksen' sijaan vain silloin, kun toisen toiminta vaikuttaa vahvasti myös meihin. Luotamme esimerkiksi, että sukulainen maksaa velkansa takaisin, koska muuten menettäisimme lainatut rahamme. Toisaalta oletamme kenties, että itsekseen vilkkaasti puhuvalla ohikulkijalla on kännykkä jota emme näe, mutta oletuksen osoittautuminen vääräksi ei juuri vaikuta elämäämme. Vähäisesti vaikuttavat asiat jäävät harmaalle alueelle, jossa sanankäyttö voi vaihdella. Voimme toki myös sanoa "luottavamme" siihen, että asuntomme ei pala maan tasalle tällä viikolla. Tämän tutkielman puitteissa keskitytään kuitenkin ihmisiin, tai tarkemmin heidän ohjelmallisiin edustajiinsa kohdistuvaan luottamukseen.

Tietoverkon kautta viestivät ihmiset eivät ole tekemisissä keskenään suoraan, vaan jonkinlaisen ohjelmallisen edustajan kautta. Nämä ohjelmat voivat yksinkertaisimmillaan välittää esimerkiksi kirjoitettuja tai nauhoitettuja viestejä, mutta ne voivat myös hoitaa hieman itsenäisempiä tehtäviä, kuten kerätä tietoa web-sivustojen linkkien kautta. Ihminen on toiminnassaan vapaampi kuin häntä edustava ohjelma, ja voi esimerkiksi valita useita erilaisia edustuksia itselleen—esimerkiksi eri käyttäjätunnuksia tiettyyn keskusteluryhmään. Toisaalta esimerkiksi sähköpostimatojen ja ns. troijalaisten hevosten vaikutuksesta ihminen ei aina kykene hallitsemaan itseään edustavia ohjelmia, vaan niiden toimintaan saattaa käyttäjän tietämättä vaikuttaa useita muitakin ihmisiä. Täten ohjelmalliseen edustajaan luotettaessa on sen käyttäjän hyvän tahdon lisäksi uskottava siihen, ettei edustajaa manipuloi joku muu [Jøs96]. Koska pelkän käyttäjätunnuksen jäljittäminen sen omistavaan ihmiseen on joskus jopa mahdotonta, tutkielmassa rajoitutaan ihmisen edustajiin kohdistuvaan luottamukseen. Tämä edustaja voi olla käyttäjätunnus tai muu pysyvä "virtuaalihenkilöllisyys". Luottaja puolestaan on ihminen tai joukko ihmisiä, joita edustaa heidän luottamusnäkemystään ohjeistuksen mukaan toteuttava tietojärjestelmä; järjestelmä itsessään ei sinänsä luota mihinkään, kuten Jøsang perustelee [Jøs96]. Ilmaisun yksinkertaistamisen vuoksi jatkossa tästä asetelmasta puhutaan kuitenkin järjestelmän "luottamuksena", jonka kohteena ovat käyttäjät.

Kun olemme muodostaneet yleiskäsityksen siitä, miksi ja miten ihmiset luottavat, voimme pohtia luottamuksen merkitystä tietoturvan kannalta. Ihmiset käyttävät luottamusta epävarmuuden ja siihen liittyvien riskien hallintaan. Tämä lienee tarpeen myös tietojärjestelmissä. Oletetaan, että luottamus tulisi ottaa osaksi itsenäisten toimijoiden tietoturvahallintoa, eikä sitä nykyjärjestelmissä kannata korvata muilla käsitteillä tai ratkaisuilla. Seuraavaksi tulee päättää, missä määrin seurata ihmisen luottamuksen mallia sekä missä määrin luottamuksenhallinta voi erota tästä mallista ja parantaakin sitä.

Tutkielmassa luottamusta ja sen hallintaa tarkastellaan web-palveluympäristön kontekstissa. Tässä ympäristössä olemme jatkuvasti tekemisissä useiden itsenäisten toimijoiden kanssa. Näiden autonomisuus lisää epävarmuutta ympäristössä, ja tapahtumien ennakointi vaikeutuu. Koska web-palveluympäristössä kunkin palvelun toteutus piilotetaan palvelun abstraktion taakse, emme voi tietää, mitä kaikkea palvelupyyntömme todellisuudessa saa aikaan. Jotta epävarmuus ei estäisi yhteistoimintaa, luottamus on jopa välttämätöntä [Sin03].

Vaikka luottamus ja sen hallinta ovat nousseet esille käsitteinä vasta viime vuosina [GS00], tietojärjestelmissä on jo aiemmin voitu havaita luottamuksen hallinnan esiasteita. Esimerkiksi Unix-ympäristössä on pääkäyttäjä, jonka kaikkivoipiin käsiin koko järjestelmä luovutetaan. Myöhemmin hallintaa on voitu tarkentaa niin, että tietyllä peruskäyttäjällä annetaan pääkäyttäjän oikeudet tietyn toimenpiteen suorittamiseen. Käyttäjien jaottelu erilaisiin ryhmiin ja pääsyoikeuksien jakaminen näille ryhmille on myös eräänlaista luottamuksen ilmaisua, pääsynhallinnan lisäksi. Nämä rakenteet ovat kuitenkin varsin jähmeitä: mikäli käykin ilmi ettei kyseinen käyttäjä ole enää luottamuksemme arvoinen, käyttäjän oikeuksia on muokattava käsin. Yksittäisten toimenpiteiden suhteellisen raskauden takia käyttäjien valvontaan ja järjestelmän pääsyhallinnan hienosäätöön riittää harvoin resursseja muuten kuin hyvin karkealla tasolla. Luottamuksella on myös käyttötapoja pääsynhallinnan ulkopuolella; myöhemmin esitellään näistä muutamia.

Tutkielmassa tutkitaan myös luottamuksenhallinnan eri puolten toteutusmahdollisuuksia. Yksi edellytys joustavalle luottamuksen hallinnalle on politiikan ja toteutuksen erottaminen toisistaan, jolloin järjestelmää ei tarvitse esimerkiksi kääntää jatkuvasti uudelleen luottamusarvioiden muuttuessa ajan myötä. Tässä tukena ovat politiikkakielet, joilla etenkin pääsynhallinnan sääntöjä voidaan ilmaista ja myös muuttaa dynaamisesti [DDLS01, UBJ04, KFJ03, TBJ+03].

Luvussa 2 käsitellään kohteeksi valittua web-palveluympäristöä ja sen tuomia vaatimuksia luottamuksenhallinnalle. Luvussa 3 tarkastellaan erilaisia näkökulmia luottamukseen, sen mallintamiseen ja käyttöön. Luvussa 4 esitellään luottamukseen ja sen avulla tehtäviin päätöksiin vaikuttavat tekijät sekä nimetään luottamustiedon lähteet. Tämän tietosisällöllisen kuvauksen jälkeen luku 5 kattaa luottamuksenhallinnan toiminnallisen kuvauksen paikallisesta näkökulmasta. Lopuksi luku 6 laajentaa paikallisen näkökulman suurempaan yhteisömittakaavaan, ja kuvaa luottamuksen käyttöä yhteisöjen toiminnassa. Yhteenvedossa nimetään jatkotutkimuksen haasteita ja hahmotellaan lähitulevaisuuden näkymiä.

2 Yhteistoiminnan haasteet

Luottamus helpottaa riskinhallintaa toimittaessa yhdessä muiden ihmisten kanssa. Sen käyttöönotto tietojärjestelmien suojauksessa voisi parantaa järjestelmän mukautumistehokkuutta uusiin tilanteisiin huomattavasti ja vähentää siten myös ihmisvalvonnan tarvetta. Tässä luvussa kuvaillaan aluksi yhteistoiminnan haasteita, joihin luottamus voi vastata, minkä jälkeen kuvaillaan tarkemmin web-palveluympäristöä.

Tämän jälkeen hahmotellaan karkea arkkitehtuuri luottamuksenhallinnan sijoittumisesta toimintaympäristöönsä sekä esitellään esimerkkikäyttötapaukset, joiden pohjalta järjestelmän toimintaperiaatteita voidaan selkeyttää.

2.1 Luottamus yhteistoiminnan perustana

Palveluja myyvässä yrityksessä kaksi lähes vastakkaista tavoitetta kohtaavat. Tietoturvavaatimusten täyttäminen ja riskinhallinta onnistuvat parhaiten täysin suljetussa järjestelmässä. Toisaalta myynnin kasvattaminen ja mahdollisimman monen asiakkaan tavoittaminen puolustavat palvelun avoimuutta. Avoimuuteen liittyy myös toteutettavuusrajoitteita suojaukselle, kuten asiakkaiden turhan rasittamisen välttämistä kilpailukyvyn ylläpitämiseksi. Tästä syystä esimerkiksi rekisteröityminen palvelun käyttäjäksi tulisi olla mahdollisimman vaivatonta, jolloin esimerkiksi tunnistamista ei voida suunnitella yksin tietoturvan tarpeista lähtien.

Sama avoimen ja suljetun järjestelmän välisen tasapainon etsiminen värittää lähes kaikkia yhteistoimintaympäristöjä. Tutkielman tarkastelun aiheena on web-palveluympäristö, jossa luottamussuhteita muodostetaan paitsi asiakkaan ja palveluntarjoajan välille, myös palveluntarjoajien yhteisöissä. Nämä yhteisöt toimivat vastapainona kehitykselle, jonka mukaan yrityksillä on tarvetta erikoistua muutamaan kapeaan ydinosaamisalaan. Kun asiakkaiden palveleminen vaatii yleisiä ratkaisuja, erikoistuneet palveluntarjoajat voivat liittyä yhteen kokonaisratkaisun luomiseksi. Esimerkiksi verkossa toimivan matkatoimiston voidaan katsoa muodostavan yhteisön lentoyhtiön, majoitustahon ja mahdollisen elämysmatkaan kuuluvan muun ohjelman järjestäjän kanssa. Tällöin kunkin palvelua tuottavan yhteisön toimijan tulee voida luottaa toisiinsa. Muun muassa matkatoimiston on luotettava siihen, että lentoyhtiö hoitaa lupaamansa lennon asiakasta tyydyttävällä tavalla, ja lentoyhtiö luottaa siihen, ettei matkatoimisto varaa lippuja väärin perustein, jolloin pahimmassa tapauksessa koneet lentävät tyhjinä.

Elämysmatkan järjestävä yhteisö on tällä hetkellä todennäköisesti pitkäaikainen kokonaisuus, joka on järjestetty ihmisvoimin. Web-palveluympäristössä yhteisöt eivät välttämättä ole pitkäikäisiä, vaan ne saatetaan tuoda yhteen yhden palveluinstanssin tarjoamiseksi ja hajottaa välittömästi tämän toteuduttua. Eri toimijat valitsevat roolinsa yhteisössä, ottaen vastuun itselleen sopivista tehtävistä. Tähän liittyvä hallinnointi hoituu mahdollisimman pitkälti automaattisesti. Kuten matkatoimistoesimerkissä, yhteisön jäsenet ovat itsenäisiä toimijoita, jotka muodostavat löyhän kokonaisuuden. Kunkin jäsenen on varmistuttava itse siitä, etteivät muut yhteisön jäsenet aiheuta sille haittaa. Ulkoisen kontrollin vähyydestä johtuen yhteisön toimintaan liittyy niin paljon epävarmuutta, ettei yhteistoiminta ole mahdollista ilman luottamusta. Luottamuksenhallintajärjestelmä voi auttaa kutakin toimijaa päättämään, onko jokin toiminto järkevää sallia.

Yhteisöjen dynaaminen muodostaminen ei ole vielä yleistynyt huomattavasti sähköisen kaupankäynnin piirissä. Perinteisesti palveluja verkossa tarjoava yritys on toteuttanut tai toteutuuttanut itse kaikki palvelunsa osat. Palvelujen yhdistämistä voi

peruskäyttäjäkin kuitenkin huomata etenkin maksuvaiheessa, kun tarvikkeet voidaan esimerkiksi maksaa ennakkoon pankin tarjoamalla verkkomaksupalvelulla tai vaikkapa PayPal-palvelun [Pay05] kautta. Suomessa jotkin verkossa toimivat kaupat, kuten Verkkokauppa.com [Ver05], mahdollistavat myös kuljetusten tilan tarkkailun verkossa Postin seurantajärjestelmän [Suo05] avulla. Tällöin kauppa käyttää oman palvelunsa osana pankkien tai Postin palveluja. Laajempaa yhdistelyä on harjoitettu lähinnä yrityksen sisäisesti.

Yhteisömaailma on laajuudessaan varsin monimutkainen, joten tässä tutkielmassa keskitytään sen yksinkertaiseen erikoistapaukseen: yhden asiakkaan ja kirjakauppapalvelun "pienoisyhteisöön". Tässä yhteisössä roolit ovat hyvin rajatut: kirjakauppa myy kirjoja ja asiakas ostaa niitä. Yhteisö syntyy dynaamisesti asiakkaan ensimmäisen palvelukutsun yhteydessä, ja sen muodostuessa ensi kerran asiakkaalta otetaan ylös riittävät tunnistetiedot, jotta hänet voidaan tunnistaa myöhemminkin. Lisäksi asiakkaan luottamus jätetään käsittelemättä, ja keskitytään kirjakaupan asiakkaaseen kohdistamaan luottamukseen. Kirjakauppa pyrkii suojaamaan itseään asiakkaan ostoilta ja tilauksilta, joista se ei saa haluamaansa vastinetta.

Kirjakaupan paikallisen luottamuksenhallintajärjestelmän kuvailun jälkeen palataan tutkielman lopussa vielä lyhyesti siihen, miten esimerkki laajenee koskemaan yhteisöä: kirjakauppa voi tarvittaessa muodostaa dynaamisesti yhteisön esimerkiksi harvinaisempien kirjojen välittäjän ja kuljetuspalvelun kanssa pystyäkseen vastaamaan tavallista haastavampaan palvelupyyntöön. Se toimii myös osaltaan suuremmassa kontekstissa, jossa eri kirjakaupat saattavat vaihtaa tietoa kokemuksistaan eri asiakkaiden kanssa, luoden osin yhteistä luottamustietoa.

Trust Based on Evidence (TuBE) -projekti [KVR05] pyrkii löytämään yhden yhteisö- ja paikallistasolla toimivan yleisratkaisun, mahdollisesti yhdistelemällä jo olemassa olevia ratkaisuja. Tässä tutkielmassa tavoitteena on seurata samaa periaatetta, tosin pienemmässä mittakaavassa, kehittäen luottamuksenhallinnan arkkitehtuuria edellä rajatulle esimerkille. Tutkielmassa on osin tehty päätöksiä eri tekijöiden edustuksesta ja vaikutuksesta toisiinsa, jotka eivät kaikki toistu koko projektin mittakaavassa. Projektin puitteissa on tutkielman lisäksi tehty muun muassa katsaukset luottamuksenhallinnan tutkimuksen [RK05] sekä tietomurtohälytintutkimuksen [Vil05a] tilaan. Edellisen kontekstissa hieman suppeammalta luottamuksenhallinnan alueelta valittuja esimerkkiprojekteja on luokiteltu luottamuspäätöksissä käytettyjen parametrien mukaan luottamusontologian pohjaksi [Vil05b].

2.2 Web Services -arkkitehtuuri avoimille palveluille

Maailmanlaajuinen tietoverkko on yhteistoiminnan edistämiseen omiaan. Internet on kuitenkin myös avoimuudessaan turvaton paikka, sillä asiakkaasta ei välttämättä voi varmasti tietää muuta kuin esimerkiksi tämän IP-osoitteen, joka sekin saattaa olla vain väliaikainen. Kuten edellä mainittiin, palvelun myynnin tehostamiseksi sitä halutaan tarjota mahdollisimman avoimesti kaikille mahdollisille asiakkaille. Vastapainona tällöin tietoturvan kannalta järjestelmän suojaus kuitenkin vaikeutuu.

Tässä aliluvussa kuvaillaan web-palveluympäristöä, sen taustaa ja siihen liittyviä luottamuksen haasteita.

Nopeasti muuttuvat, laajat käyttäjäkannat aiheuttavat haasteita nykyisille suojausjärjestelmille. Päätöksenteko tietyn hankalaksi muodostuneen käyttäjän pääsyoikeuksien rajoittamisesta vaatii ylläpitäjän puuttumista asiaan, useimmiten jollakin käyttäjäkohtaisella rajoituksella. Toisaalta myös asiallisesti toimivat käyttäjät kärsivät koko käyttäjäkantaan vaikuttavista rajoituksista, joilla yritetään suojata järjestelmää muutamalta huonosti käyttäytyvältä käyttäjältä.

Tietojärjestelmiin lisätään usein suunnitteluvaiheessa, kenties tiedostamattomasti, hieman implisiittistä ja karkeajakoista luottamuksen hallintaa. Epäluotettavaksi todetuilta käyttäjiltä saatetaan sulkea käyttäjätili tai tutulle käyttäjälle antaa hieman erityisoikeuksia, jotka helpottavat hänen toimintaansa. Tämä vaatii ylläpitäjän puuttumista asiaan. Toisaalta tunkeutumisenestojärjestelmä (engl. *intrusion prevention system, IPS*) tai palomuuri voidaan säätää katkaisemaan hyökkääjäksi arvellun käyttäjän yhteys järjestelmään jopa automaattisesti. Ikävä kyllä käyttäjien tarkkailu vaatii resursseja, jolloin valvonnan tarkentaminen tai esimerkiksi reaaliaikaistaminen säännöllisen lokitarkkailun sijaan voi suurella käyttäjäkannalla olla mahdoton toteuttaa.

Kun luottamus liitetään tietoisesti näkyväksi osaksi järjestelmää, tietoturvahallinnointia voidaan automatisoida ja hienojakoistaa järjestelmäriippumattomien työkalujen ja periaatteiden avulla. Luottamuksen käsitteistö voi myös helpottaa suojausjärjestelmästä keskustelemista, sillä sen periaatteet voi hahmottaa henkilökohtaisen kokemuksen pohjalta ilmankin kattavaa teknistä taustaa. Luottamus voi siis olla paitsi yleistettävissä oleva työkalupohja, myös tehokas paradigma. Ihmisen ja "koneellisen" luottamuksen erot [Ess97, Mar94] ovat kuitenkin vielä tutkittavina, eikä aina ole selvää, missä määrin ihmisten luottamuksen mekanismeja on järkevää yrittää toisintaa ohjelmallisesti.

Luottamuksen automatisointi, kuten muukin tietoturva-automaatio, vaatii käyttäjiensä hyväksynnän. Palveluntarjoajan on voitava luottaa järjestelmään, joka edustaa sen omia luottamussuhteita. Tähän luottamukseen kuuluu esimerkiksi vakaa usko siitä, että luottamusjärjestelmä toimii kuten sen kehittäjät kertovat sen toimivan, ja siten toteuttaa palveluntarjoajan luottamuspolitiikkaa kuten tämä on sen ilmaissut. Lisäksi järjestelmän tulisi kyetä riittävissä määrin suojaamaan itseään ulkopuolisten vaikutukselta; kenties sen asetukset on suojattu salasanalla ja pääsy fyysiseen laitteistoon on rajoitettua. Lisäksi järjestelmän tulee käsitellä verkosta saapuvat virheelliset ja oikeelliset kyselyt määritelmiensä mukaisesti, eikä sortua esimerkiksi puskurin ylivuotoon tai muihin ohjelmavirheisiin.

Luottamuksenhallinnan tutkimuksen alkuvaiheissa 1990-luvun loppupuolella yksi tärkeistä edistysaskelista on ollut pyrkimys erottaa politiikkapäätökset yksittäisten sovellusten toteutuksesta edellisten helpomman päivityksen mahdollistamiseksi [BFL96, BFK98, CFL+97]. Taustana oli havainto perinteisten järjestelmien jäykkyydestä; sovelluksen tai sovellusten joukon päivittämispakko aina uuden politiikan käyttöönoton yhteydessä vähentää minkä tahansa tietoturvajärjestelmän jous-

tavuutta huomattavasti.

Myös edellä mainittu palvelujen yhdistäminen kärsii perinteisten massiivisten järjestelmien jäykkyydestä. Komponenttiohjelmoinnille on tyypillistä tiukka kytkentä toiminnallisten kokonaisuuksien välillä, mikä hankaloittaa osien uudelleenkäyttöä. Tällöin kutsurajapintojen muutosten lisäksi ongelmallisia ovat esimerkiksi oletukset muiden komponenttien sisäisen tilan muutoksista tietyn kutsun seurauksena. Lisäksi yhdisteltävät osat on kenties toteutettu eri ohjelmointikielillä, ja niiden suoritusalustavaatimukset vaihtelevat.

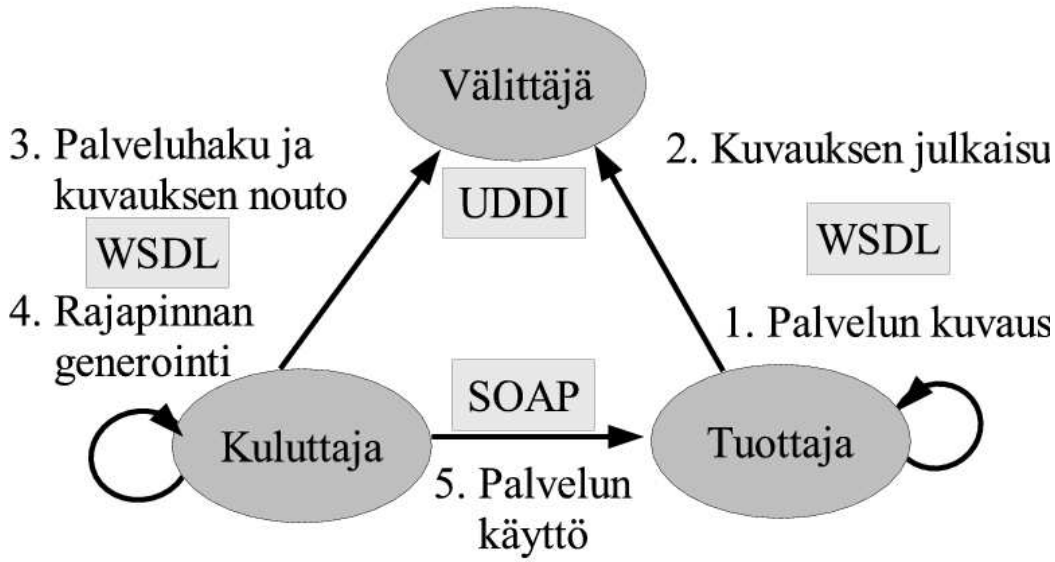
Kahdessa eri laitteessa ajettavat komponentit voivat viestiä esimerkiksi etäproseduurikutsuin (*Remote Procedure Call, RPC*). Tällöin RPC-ajuri piilottaa ohjelmointikielten ja alustojen väliset erot itse komponenteilta. Etäproseduurikutsu ei kuitenkaan sellaisenaan ratkaise tiukan kytkennän aiheuttamia ongelmia. Sitä on myös kritisoitu raskaskäyttöiseksi ja hitaaseen verkkoon heikosti sopivaksi [Lin01, s.168].

Palvelukeskeisen periaatteen mukaan osat tulisikin abstrahoida moduulien sijaan korkeamman tason palveluiksi, kuten matkatoimistoesimerkissä luontevasti lennon, majoituksen tai elämysohjelman varaamiseksi. Palvelua tulisi voida kutsua standardirajapinnan kautta ilman tietoa tai oletuksia palvelun tai asiakasohjelman sisäisestä tilasta. Lisäksi palvelun käyttöön vaadittavat tiedot tulisi voida saada välittäjältä, jolloin palvelun sijaintiakaan ei tarvitsisi tuntea ennalta [Pap03]. Palvelukeskeisyys ohjaa toteuttajaa löyhentämään palveluina tunnettujen komponenttien välisiä kytkentöjä niin pitkälle, että niitä voidaan paikantaa ja yhdistää ajonaikaisesti, kenties jopa niin, ettei ohjelmoijan tai muun valvovan ihmisen tarvitse lainkaan puuttua asiaan. Tämä on huomattava askel kohti dynaamisesti muodostuvia yhteisöjä.

Palvelukeskeisen laskennan (*Service-Oriented Computing, SOC*) periaatteita ja arkkitehtuuria edustaa web-sovellusten maailmassa Web Services -arkkitehtuuri. Arkkitehtuurin kuvauksessa määritetään yleiset elementit, jotka tarvitaan web-palvelujen yhteensopivuuden tueksi. Se ei ota kantaa itse palvelujen toteutukseen eikä rajoita niiden yhdistelyä [BHM+04]. Web-palvelujen rajapinnat kuvaillaan koneluettavalla standardikielellä, ja palveluja kutsutaan viestipohjaisesti.

Web Service -arkkitehtuuri on kaikkiaan hyvin laaja kokonaisuus, johon on ehdotettu toiminnan eri tasoille ja alueille yhteensä kymmeniä eri osa-arkkitehtuureja, protokollia ja standardeja, joista osa korvautuu uusilla tarpeiden tarkentuessa. Tähän on poimittu esiteltäviksi vain kourallinen keskeisimpiä esimerkkejä.

Web-palveluympäristön toimijat voidaan jakaa kolmeen rooliin: palvelun tarjoaja, sen kuluttaja ja mahdollinen välittäjä, joka tuo kaksi muuta toimijaa yhteen. Palvelun tarjoaja on toteuttanut jonkin palvelun, jolle hän haluaa käyttäjiä. Hän kuvailee palveluaan jollakin sovitulla kielellä, jota asiakkaat ja välittäjät ymmärtävät. Käsittelemme myöhemmin tähän tarkoitukseen kehitettyä kieltä, WDSL:ää. Mikäli palvelun tarjoaja tuntee jo valmiiksi kaikki tulevat asiakkaansa, tämä kuvaus voidaan lähettää suoraan asiakkaille. Nämä luovat sen pohjalta liittymän omaan järjestelmäänsä, ja kaikki osapuolet ovat valmiita palvelun käyttöönottoon. Tuonnempana käsitellään palvelun käyttämiseen suunniteltua viestintäprotokollaa, SOAPia.



Kuva 1: Web-palveluympäristön toimijaroolit ja esiteltävien standardien käyttöalueet.

Usein palvelun tarjoaja ja hänen asiakkaansa joutuvat kuitenkin etsimään toisensa. Tämän helpottamiseksi web-palveluympäristössä on palvelujen välittäjä, jolle sovitun muotoinen palvelukuvaus voidaan lähettää. Julkaisun jälkeen tulevat asiakkaat löytävät välittäjän hakemistosta haluamansa palvelun ja hakevat kuvauksen. Palvelunvälityksen keskeistä kuvausta, UDDIa, käsitellään myöhemmin. Kuvaillut kolme toimijaroolia, niiden väliset suhteet ja esiteltävien standardien sijainti tässä kolmikehässä on koottu kuvaan 1.

SOAP[1] määrittää palvelua käytettäessä lähetettävien viestien muodon ja yleisen käyttötavan. Viestiprotokolla rakentuu merkintäkieli XML:lle. Web-palvelua "kutsutaan" lähettämällä SOAP-viesti esimerkiksi HTTP-kuljetusprotokollan yli palvelukoneeseen. Tämä kutsu voidaan SOAPin puitteissa tehdä RPC-kutsun tapaan synkronisesti, jolloin kutsuja jää odottamaan vastausta, tai asynkronisesti. Kummassakaan tapauksessa palvelua ei kutsuta kuten funktiota, vaan hoidettavaksi lähetettävän viestin käsittelijä voidaan vaihtaa toiseen milloin vain ilman, että viestin rakennetta muutetaan. Palvelukuvaus, jonka pohjalta viestit rakennetaan, ei ota kantaa toteutukseen, joten tämä voidaan vaihtaa taustalla ilman, että kutsutapa muuttuu.

Web Services Definition Language, WSDL on kieli palvelun kuvaamiseksi. Palvelun toimintaa kuvaillaan syntaktisella tasolla, eikä toteutusta kiinnitetä lainkaan. WSDL-kuvaus kertoo myös, miten palvelua kutsutaan protokollan ja koodauksen tasolla, sekä määrää lähetettävien viestien muodot. WDSL-kuvaus toteutetaan SOAP-viestien tapaan XML-dokumenttina. WSDL-kuvaus ei kuitenkaan ole

[1] Termiä 'SOAP' ei ole protokollan versiosta 1.2 lähtien määritelty lyhenteeksi; alkujaan kirjaimet tulevat sanoista *Simple Object Access Protocol*. Web Services Architecture [BHM+04, s. 65] avaa kirjaimista myös havainnollistavamman termin *Service Oriented Architecture Protocol*.

yksin suurikaan apu sopivaa palvelua hakevalle asiakkaalle, joten palvelukuvausta julkaistaessa mukaan voidaan liittää myös muita tietoja palveluista. Rajapinta- ja tietotarpeet vaihtelevat riippuen siitä, tekeekö haun itsenäinen ohjelma-agentti vai ihminen. Joidenkin lisämääreiden ilmaisemiseen on suunniteltu koneluettavia kieliä, toisia ilmaistaan yhä lähinnä ihmislukijoille suunnattuna tekstinä.

Universal Description, Discovery and Integration, UDDI määrittää eräänlaisen metapalvelun, joka helpottaa varsinaisten web-palvelujen löytämistä ja käyttöönottoa. Spesifikaatio kuvaa hajautetun palveluhakemiston tietosisällön ja sen käyttötavat. Tiedon levitystä ja pyytämistä käsittelee oma osansa, *Technical Architecture*, kun taas tietosisältö määritellään osiossa *Business Registry*. Tietosisältöön kuuluu teknisen palvelukuvauksen lisäksi mm. tietoa palveluja tarjoavista yrityksistä tai yhteisöistä [BCE+02].

TuBE-projektin tavoitteena on lisätä tähän Web Services -arkkitehtuurin rajaamaan palveluympäristöön tuki luottamuksenhallinnalle. Arkkitehtuuriin on jo aiemmin esitetty myös erilaisia tietoturvaan liittyviä standardeja, joista yksi on WS-Trust [KN+04]. Tämä spesifikaatio ei kuitenkaan tue kaipaamaamme luottamushallinnointia, vaan keskittyy pikemminkin todentamiseen ja uskottavuuteen. WS-Trust käsittelee luottamusta henkilöllisyyden ja muiden väitteiden todisteisiin, ja rajoittuu siten tämän tutkielman ulkopuolisiin asioihin. Tutkielman käsittelemä yleisempi luottamus kohdistuu tiettyyn yksilöön esimerkiksi tietyn toimenpiteen ja sen kohteen suhteen.

Web Services -arkkitehtuuri ja siinä toteutuvat palvelukeskeiset suunnitteluperiaatteet mahdollistavat uusien sovellusten lisäksi myös alunperin muuhun toimintaympäristöön toteutettujen sovellusten (niin sanottu *legacy application*) käärimisen SOAP-viestikäyttöön. Arkkitehtuuri helpottaa myös palvelujen uudelleenkäyttöä yhdistelyn kautta. Esimerkiksi kirjakauppa voi hakea UDDI-kannan kautta tarvitsemaansa harvinaisten kirjojen välittäjää ja suuremman tilauksen tapauksessa myös kuljetuspalvelua. Se voi siten tarkistaa WSDL-kuvauksen perusteella, miten näiden palveluiden kanssa tulisi viestiä, ja hoitaa tilauksen lopulta SOAP-viestein. Tässä kuvailtu pääteknologioiden kolmikko ei kuitenkaan ratkaise vielä kaikkia yhteisönmuodostuksen ongelmia; WSDL-kuvaus ei sisällä semanttista tietoa eri parametrien merkityksestä, vaan se vaatii ihmisen lukemaan kuvauksen mukana tulevaa dokumentaatiota, jotta yhteensopiva viestinlähetys voidaan toteuttaa kirjakaupan sovelluksessa. Web-Pilarcos-projekti tutkii yhteisönmuodostusta ja hallinnointia web-palveluympäristössä, ja lisää perusteknologioiden valikoimaan tarpeellisia väliohjelmistopalveluja [KRMH05].

2.3 Käyttötapauksen kuvaus

Asiakkaan ja palveluntarjoajana toimivan kirjakaupan välillä on luottamussuhde, joka voi saada vaikutteita muilta toimijoita, esimerkiksi toisilta kirjakaupoilta. Kaikki luottamuspäätökset tehdään kuitenkin itsenäisesti. Luottamuksenhallintajärjestelmä toimii itsenäisesti kirjakaupan sovelluksen vierellä, vastaanottaen sille lähetetyt

SOAP-viestit ja tehden niiden pohjalta luottamuspäätöksen siitä, sallitaanko viestin edustama toiminto.

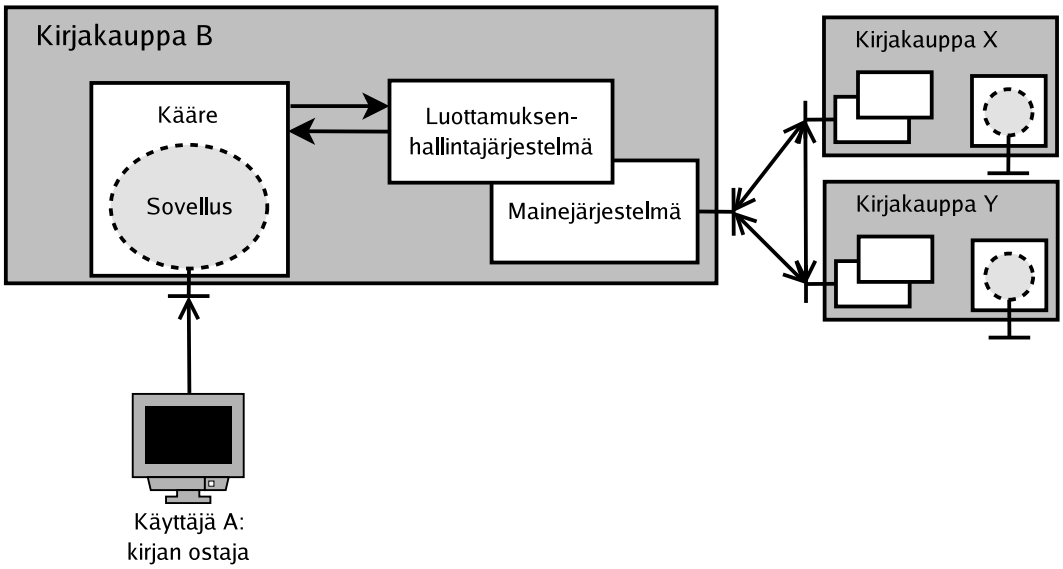
Tyypillinen käyttötapaus alkaa asiakasyrityksen tietokantapankkiin kehittämän sovelluksen saadessa toimintopyynnön käyttäjältä SOAP-viestin muodossa. Oletetaan esimerkin vuoksi että toiminto koskee kirjan ostamista. Sovellusta ympäröivä kääre (*wrapper*) tunnistaa SOAP-viestin tiettyyn toimintoon kuuluvaksi asiakasyrityksen asetusten mukaisesti ja kysyy luottamuksenhallintajärjestelmältä, sallitaanko ostotoiminto kirjakaupan luottamuspäätöspolitiikan mukaan. Myönteisestä luottamuspäätöksestä seuraa, että toiminto sallitaan ja kirjakauppasovellus valmistautuu palvelemaan pyyntöä. Mikäli luottamuspäätös on kielteinen, SOAP-viestiä ei tarvitse välittää lainkaan eteenpäin sovellukselle.

Yhden SOAP-viestin aloittama toiminto, kuten ostoesimerkin tapauksessa, saattaa kestää useiden viestien ajan: aluksi sovellus ilmoittaa kirjan saatavuustiedot, hinnan ja maksutapamahdollisuudet paluuviestissä, jolloin käyttäjä lähettää viestin, jossa valitsee maksutavan. Tämän jälkeen sovellus selvittää käyttäjältä tarvittavat tiedot lisäviestien myötä, mikä on yksinkertaisuuden vuoksi liitetty käyttäjän vastausviestin osaksi SOAP-viestien kuvauksessa (liitteet 1 ja 2). Lopulta kirjakauppa merkitsee kirjan ostetuksi. Tämän jälkeen sovellus saattaa vielä valmistella tarvittavat toimenpiteet kirjan toimittamiseksi käyttäjälle ja mahdollisesti sen laskuttamiseksi. Verkon yli kulkevissa SOAP-viesteissä luottamusta ei edusteta lainkaan, vaan luottamuksen käsittely ja ylläpito tapahtuu paikallisesti.

Mikäli luottamuspäätös moniviestiselle toiminnolle on myönteinen, se kattaa kaikki toimintoon liittyvät viestit, jolloin myöhemmiltä viesteiltä ei tarvita enää erillisiä luottamuspäätöksiä. Ne tulee kuitenkin voida yhdistää aiempaan päätökseen — ilman toiminnon aloitusviestiä lähetetyt viestit rikkovat odotettua järjestystä, joten ne voivat myös vaikuttaa maineeseen. Ellei kirjakauppasovelluksen katsota hylkäävän itse väärässä vaiheessa lähetetyt viestit, täytyy kääreen hoitaa karsinta.

Toiminnon edetessä luottamuksenhallintajärjestelmälle välitetään tietoa toimijan käytöksestä järjestelmässä. Poikkeavuudet ja epäilyttävä toiminta vaikuttavat kokemuksen kautta toimijan luotettavuutta mittaavaan maineeseen, samoin kuin odotusten mukainen käytös. Paikallinen mainekäsitys voidaan välittää yhteensopivaa mainejärjestelmää käyttävälle yhteisölle.

Korkean tason arkkitehtuuri on esitetty kuvassa 2. Käyttäjä A ottaa yhteyttä kirjakauppaan B palvelusovelluksen kautta ostaakseen kirjan. Sovellusta suojaava kääre keskustelee luottamuksenhallintajärjestelmän kanssa pyyntöön suostumisesta. Myönteisen päätöksen seurauksena pyyntö välittyy sovellukselle. Käyttäjän toiminnan seuraaminen tuottaa päivityksiä paikalliseen mainejärjestelmään, joka voi välittää nämä tiedot kirjakauppojen X ja Y mainejärjestelmille.



Kuva 2: Luottamuksenhallinta kontekstissaan; korkean tason arkkitehtuurikuvaus.

3 Luottamuksen käsite

Luottamus on yleinen termi, joka herättää kontekstista riippuen hieman erilaisia mielikuvia. Näistä mielikuvista valikoidaan edelleen tietyn käyttötarkoituksen kannalta hyödylliset osat määriteltäessä luottamusta, jolloin lopputulokset voivat olla jo hyvin erilaisia. Luottamusta käytetään ja tutkitaan varsin monenlaisissa ympäristöissä, joista tutkielmaa varten on rajoituttu web-palveluympäristöön. Tämä luku rakentaa luottamuksen tutkimuksen taustaa esittelemällä luottamuksen tutkimuksen kirjoa erityisesti tietojenkäsittelytieteen alalla. Ihmisten luottamuksen tutkimuksella esimerkiksi sosiologian alalla on myös vaikutuksia tietojenkäsittelytieteessä [FSD+02, JSTT04].

3.1 Luottamusmallit

Luottamus määritellään kirjallisuudessa eri tavoin sovelluksesta riippuen. Todentamisen (engl. *authentication*) yhteydessä luottamus saatetaan jopa typistää tiedoksi siitä, kuka kohde on tai mihin "luotettuun" ryhmään se kuuluu [BFK98, WSJ00], jolloin luottamus neuvotellaan vaihtamalla varmenteita. Luottamussuhteita tutkittaessa voidaan keskittyä siihen, onko yhteisön kahden toimijan välillä yhteistoimintaa, jossa esiintyy luottamusta vaativaa epävarmuutta [GMMZ04]. Tällöin havainnoidaan luottamussuhteita niiden tarpeen kautta. Tiedon ja sen lähteen luotettavuus on tullut esille erityisesti suositusjärjestelmissä, joissa eri käyttäjät suosittelevat esimerkiksi kirjoja toisilleen [MB04, GKRT04].

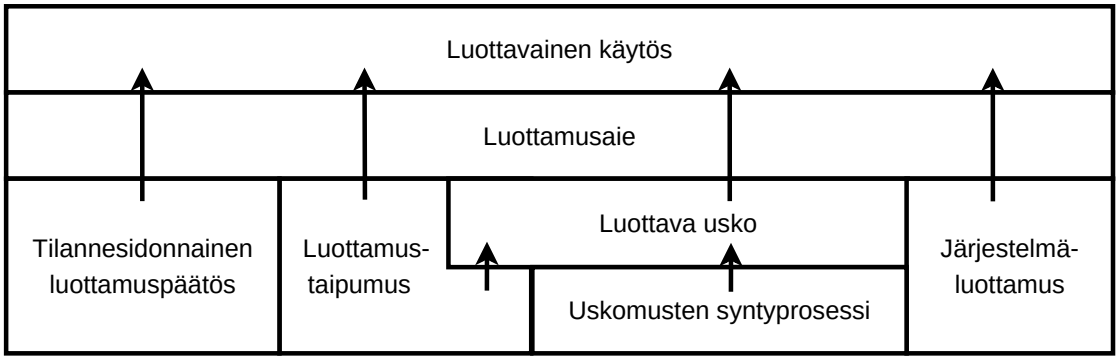
Luottamuspäätökset yleisemmin mahdollistavien määritelmien vaihtelu koostuu lä-

hinnä erilaisten parametrien vaihtelusta, kuten luottamuspäätökseen liittyvän riskin roolista [Mar94, MD95] ja siitä, tulkitaanko luottamus jonkinlaiseksi hyvän tuloksen todennäköisyydeksi [Gam00], päätökseksi vaiko vain halukkuudeksi ottaa riski [MD95]. Tässä tutkielmassa luottamus katsotaan esimerkiksi Jøsangia ja Lo Prestiä vapaasti mukaillen [JP04] halukkuudeksi osallistua tiettyyn yksittäiseen toimenpiteeseen tietyn kohteen kanssa, siinä uskossa että lopputulos on myönteinen. Tämä päätös tehdään tapauskohtaisesti.

Luottamus saattaa olla myös kulttuurisidonnaista. Hofstede on todennut kansainvälisessä tutkimuksessaan epävarmuuden välttämisen tarpeen vaihtelevan eri kulttuureissa [Hof93]. Tarpeen ollessa suuri epävarmuuden uhkaa heikennetään muun muassa kaikenkattavin säännöin ja asetuksin, jolloin joka tilannetta varten on olemassa ohje siitä, miten toimitaan. Ei ole täysin poissuljettua, etteikö myös käsitys kahden agentin välisen luottamussuhteen järkevyydestä vaihtelisi kulttuureittain. Jøsang kuvaa rationaalisten olioiden, siis esimerkiksi agenttien, välisen luottamuksen olevan mahdotonta [Jøs96], koska ne vain toteuttavat käyttäjänsä määräämää politiikkaa. Käyttäjä, ihminen, puolestaan on tunteellinen olio, ja täten kykenevä luottamaan toisiin ihmisiin. Sen sijaan ihminenkään ei voi luottaa agenttiin samalla tapaa kuin muihin ihmisiin, sillä luottamus voi koskea vain sitä, miten hyvin agentti kykenee vastustamaan toisen tunteellisen olion hyökkäyksiä tai murtoyrityksiä [Jøs96].

McKnight ja Chervany ovat analysoineet luottamuksen käsitettä ja sen alakäsitteitä johtamisen kontekstissa [MC96]. He nimeävät alustavasti neljä luottamuksen tyyppiä, jotka heijastavat luottamuksen riippuvuutta eri tekijöistä. Persoonaton tai rakenteellinen (*Impersonal/Structural*) luottamus perustuu sosiaalisiin tai institutionaalisiin rakenteisiin. Se on tilannesidonnaista, muttei riipu toimijoista tai näiden tiloista. Luottavaisuus (*Dispositional trust*) perustuu luottajan persoonallisuuteen, eikä riipu luottamuksen kohteista. Henkilökohtainen (*Personal*) luottamus vastaa yhden henkilön luottamusta toiseen henkilöön tai asiaan tietyssä tilanteessa, ja henkilöiden välinen (*Interpersonal*) luottamus kahden tai useamman henkilön tai ryhmän välistä molemminpuolista luottamusta. Tutkielman asetelmassa henkilökohtainen luottamus on keskeisintä, tosin luottava "henkilö" voi olla myös yhteisö.

McKnight ja Chervany erottavat kuusi luottamuksen kategoriaa: luottava usko, luottamusaie ja luottavainen käytös johtavat kukin seuraavaan; järjestelmäluottamus, luottamustaipumus ja tilannesidonnainen luottamuspäätös vaikuttavat kukin johonkin näistä tai useaan. Kategorioiden vaikutukset toisiinsa on esitetty kuvassa 3. Luottava usko (*Trusting Belief*) syntyy osin tiedostamatta. Siihen liittyy usko tällaisen luottamuksen kohteen hyväntahtoisuuteen, rehellisyyteen, kyvykkyyteen ja/-tai ennustettavuuteen. Sen sijaan luottamusaie (*Trusting Intention* heijastaa halukkuutta antautua riippuvaiseksi toisesta, vaikka negatiiviset seuraukset ovat mahdollisia eivätkä täysin hallittavissa. Tämä halukkuus on tilanne- ja henkilösidonnaista, sillä on suunta (tietty lähde ja kohde) ja se riippuu tietoisesta tahdosta. Luottavainen käytös (*Trusting Behaviour*) liittyy luottamusaikeesta seuraavaan toimintaan. Tällaista luottamusta ei voi havaita suoraan, vaan se ilmenee indikaattorien, esimerkiksi riskinoton, yhteistyön tai tiedon vaihtamisen kautta. Itse käytös ei ole sama kuin indikaattorit.



Kuva 3: McKnightin ja Chervanyn luottamuskategoriat ja niiden vaikutus toisiinsa ([MC96], tekijän suomentama).

Järjestelmäluottamus (*System Trust*) liittyy edellä kuvattuun persoonattomaan luottamukseen. Se kuvaa luottajan uskoa siihen, että riittävät persoonattomat rakenteet ovat tukemassa toiminnan onnistumista. Täten se tukee luottamusaietta, muttei vaikuta luottavaan uskoon, joka riippuu luottamuksen kohteesta eikä ympäristöstä. Luottamustaipumus (*Dispositional Trust*) on edellä kuvatun luottamustyypin vastine kategoriamallissa. Sen taustalla on kahden syyn yhdistelmä: usko ihmisiin yleensä ja siihen, että he ovat luottamuksen arvoisia (*Belief-In-People*), ja usko siihen, että ihmisiin luottaminen saa yhteistoiminnan sujumaan paremmin (*Trusting stance*). Ensimmäinen näistä tukee luottavaa uskoa ja sen kautta välillisesti luottamusaietta, kun taas toinen tukee suoraan luottamusaietta. Tilannesidonnainen luottamuspäätös (*Situational Decision to Trust*) liittyy aikomukseen asettua riippuvaiseksi kenestä tahansa toisesta toimijasta tietyssä tilanteessa—esimerkiksi jos toimintoon liittyy pieni riski ja mahdollisesti paljon hyviä seurauksia. Päätös ei riipu järjestelmän tuesta, johon järjestelmäluottamus vaikuttaa. Se on yksinkertaisesti luottajan tilannesidonnainen toimintastrategia, ja tukee siksi suoraan luottamusaietta.

Luottamukseen vaikuttavien tekijöiden tunnistaminen on yksi tärkeä osa luottamuksenhallinnan tutkimusta. Ilman selkeää mallia eri tekijöiden vaikutuksesta ei luottamustiedon pohjalta voida tehdä järkeviä päätöksiä. Tekijöiden valinta vaikuttaa pitkälti päätöshetkellä käytettävissä olevaan luottamustietoon ja sen luonteeseen. Luottamuspäätöksiä voidaankin tutkia erillään tekijöiden tutkimisesta lähinnä silloin, kun luottamustietojen päivitys on delegoitu täysin kuvaillun järjestelmän ulkopuolelle.

Mui, Mohtashemi ja Halberstadt määrittävät luottamukseen liittyviksi termeiksi maineen ja vastavuoroisuuden (*reciprocity*) [MMH02], ja korostavat maineen ja luottamuksen määritelmällisen erottamisen tärkeyttä. He toteavat näiden kolmen termin vaikuttavan syklisesti toisiinsa. Kohonnut maine parantaa luottamusta, ja luottamus parantaa vastavuoroisuutta lisäämällä sen todennäköisyyttä, että luottaja reagoi luottamuksen kohteen luottajan kannalta edulliseen toimintaan toimimalla vastaavasti itse kohteen kannalta edullisesti. Lisäksi toimijan vastavuoroisen käytöksen lisäys vaikuttaa myönteisesti tämän maineeseen. Vastavuoroisuus määritellään mo-

lemminpuoliseksi tekojen vaihdoksi; molemminpuolisuus koskee palvelusten lisäksi myös negatiivisten tekojen kostamista, joten maineen positiivinen lisäys vastavuoroisuuden lisääntyessä ei ole aina täysin selvää.

SECURE [CGS+03] tunnistaa luottamukseen vaikuttaviksi tekijöiksi kolmansien osapuolten suositukset, epäsuorista suosituksista koostuvan maineen sekä aiemman kokemuksen. Luottamuksen esityksessä kiinnitetään huomiota myös varmuuteen (*confidence*) tiedon paikkansapitävyydestä [EWN+03]. Lisäksi riski ja odotetut edut vaikuttavat luottamuspäätökseen, mutta ne arvioidaan lähinnä luottamuksen pohjalta sen sijaan että luottamus riippuisi niistä. Jøsang ja Lo Presti ovat keskittyneet tutkimaan riskin ja luottamuksen välistä suhdetta [JP04]. Essin huomioi mallissaan kontekstin vaikutuksen, samoin kuin kohteet, joihin toiminto vaikuttaa ja siihen liittyvät riskit tai hyödyt sekä luottamuksen kohteen henkilökohtaisen panoksen toiminnossa, tämän maineen ja tunnistamisen varmuuden [Ess97].

Luottamuksen käsittelyä formaalein menetelmin on lähestytty varsin erilaisista suunnista. Demolombe nimeää kolme itsenäisesti tutkittavissa olevaa ongelmaa tällä saralla [Dem04]: Ensimmäinen ongelma on luottamusta tukevien tekijöiden, esimerkiksi havainnot, maine ja tilanteen analyysi, määrittäminen. Ongelma on läheisessä suhteessa luottamuskäsitteen yksityiskohtaiseen määrittelyyn, mihin tämä aliluku keskittyi. Monessa projektissa tekijöiden määrittäminen on vain osatavoite, ja tutkimus keskittyy muihin ongelmiin.

Ongelma, johon Demolombe on itse kohdistanut työnsä, koskee säännönmukaisuuksia luottamuksessa: millaisin säännöin voidaan johtaa seurauksia annetusta luottamukseen liittyvästä oletusjoukosta. Esimerkkitutkimus siis liittää luottamuksen logiikkaan. Kolmas ongelma on tämän tutkielman aiheeseen suorimmin liittyvä haaste luottamustiedon käytöstä päätöksenteossa. Esimerkkinä tämän aiheen tutkimuksesta esitellään SECURE-projektin luottamuksenhallinnan malli.

Demolomben maalaaman luottamuksen maailmankuvan ulkopuolelle jää tutkimustyö, jonka luottamuksen määritelmät eroavat liiaksi hänen luottamuksen uskomusluonteeseen sidotusta määritelmästään. Määritelmien yhteensopivuuden tarkastaminen onkin yleismerkityksellisiä termejä käsittelevää tutkimusta arvioitaessa tärkeää, sillä omat käsityksemme aiheesta voivat vaikuttaa ratkaisevasti tulkintaamme. Esimerkiksi luvun alussa mainittu todentaminen käsittelee luottamusta eri merkityksessä kuin TuBE, samoin luottamussuhteiden olemassaolon mallintaminen. Luottamusta on myös käytetty jokseenkin samassa merkityksessä kuin mainetta, jolloin sen leviämisen mallintamista ja simulointia voidaan tutkia [GKRT04]. Luottamussuhteiden mallintamiseen tähtäävä kieli Security-Aware Tropos toimii edustavana esimerkkinä luottamuskäsitykseltään eroavista projekteista.

3.2 Luottamus teoreettisissa ja käytännön järjestelmissä

Ihmisten luottamus on varsin tunneperäistä ja epärationaalista. Sen merkitykset ja siihen liittyvät oletukset vaikuttavat myös luottamuksenhallinnan tutkimuksessa tulkintoihin kontrolloimattomasti. Tämän ehkäisemiseksi Demolombe pyrkii tuomaan luottamuksen logiikan sääntöjen piiriin [Dem04].

Demolombe erottaa toisistaan käsitteet usko (*belief*), vahva usko ja tieto; kukin koskee yhden toimijan käsitystä toisen toimijan jostakin ominaisuudesta. Usko on näistä heikoin, ja vaikka toimijalla on jokin perustelu uskolleen, hän tai se on tietoinen perustelun mahdollisesta virheellisyydestä. Vahvan uskon edellytyksenä on usko myös edellä mainitun perustelun paikkansapitävyydestä, jolloin erehtymisen mahdollisuutta ei tiedosteta vaikka sellainen olisikin. Tieto puolestaan vaatii kyseisen perustelun pitävän aidosti paikkansa. Luottamus määritetään yhden toimijan (a) vahvaksi uskoksi ($K_a()$) siitä, että toisella toimijalla (b) on tietty ominaisuus (o) jonkin kontekstin tai asian (p) suhteen, mikä merkitään $To_{a,b}(p)$. Tällainen ominaisuus voi olla esimerkiksi uskottavuus (*cred*), joka määritelmän mukaan tarkoittaa, että kun toimija b uskoo jonkin asian olevan totta, se on totta. Tällöin a luottaa b:n uskottavuuteen asiassa p jos ja vain jos a uskoo vahvasti siihen, että kun b uskoo asian p olevan tosi ($B_b p$), se myös on tosi; merkitään:

$$Tcred_{a,b}(p) =_{def} K_a(B_b p \rightarrow p)$$

Tarkkaavaisuus (*vigilance*) määritellään kääntämällä ehtonuoli: mikäli p on totta, b uskoo sen olevan totta. Uskon ja totuuden lisäksi malli käsittelee myös tiedottamista: rehellisyyden edellytyksenä on, että b kertoo asian p a:lle vain jos uskoo p:n olevan tosi, yhteistyökyky tarkoittaa, että mikäli b uskoo asian olevan totta, hän myös kertoo asiasta a:lle. Todenmukaisuus (*validity*) määrää, että jos b on kertonut asiasta p, p on totta. Kääntäen täydellisyydellä tarkoitetaan, että mikäli p on totta, b on kertonut asiasta. Järjestelmä tukee tiedonsiirron lisäksi konkreettisempaa toimintaa: toiminnan käsite ("tehdä p todeksi" jollakin p) käsitellään yhdessä deonttisen logiikan käsitteiden, luvallisuuden ja pakollisuuden, kanssa, ja muun muassa tottelevaisuus määritellään näiden avulla. Lopuksi Demolombe lisää käsitteistöön myös yrittämisen, jonka kautta määritellään kyvykkyys kontekstilla c rajattuna: mikäli b yrittää tehdä p:n todeksi kun c, p on tosi.

Kun kaikille eri termeille on määritetty päättelysäännöt, teoriaa eri ominaisuuksien välisistä suhteista voidaan kehittää puhtaasti typografisin säännöin. Tämä erotus järjestelmän termien ja niiden arkipäiväisten merkitysten välillä vapauttaa todistukset termeihin sidottujen merkitysten piilovaikutusten ongelmalta[2]. Toisaalta formalismi myös yksinkertaistaa ja sitoo sanoille uuden merkityksen, jolloin niiden ilmaisuvoima ei pysy samana.

Demolombe tuo itse esille suurimman ongelman, harmaan sävyjen puutteen. Mikäli b on tarkkaavainen, hän uskoo p:n olevan totta *aina* kun se on. Todellisuu-

[2]Nimityksen ja merkityksen erottamisen vaikeus oli keskeisenä tekijänä geometrian kehityksessä. Peruskouluissa käsiteltävä Euklideen geometria käyttää hyväkseen sanojen "piste" ja "suora" arkista merkitystä, johon liittyen oli itsestäänselvää olettaa, että annetun pisteen x kautta voidaan piirtää vain yksi suoran y kanssa yhdensuuntainen suora. Mikäli sen sijaan termit tulkitaan niin, että suoria voidaan piirtää useita tai ei yhtään, on tuloksena vastaavasti hyperbolinen tai elliptinen geometria. Ennen tätä havaintoa useat matemaatikot uskoivat yhden yhdensuuntaisen suoran piirtomahdollisuutta käsittelevän aksiooman olevan todistettavissa muiden aksioomien pohjalta; kului noin kaksi vuosituhatta ennen kuin aksiooma todistettiin vastaesimerkillä mahdottomaksi johtaa. Samalla luotiin ns. epäeuklidinen geometria [Hof79, luku 4].

dessa tarkkaavaisuudella on eri asteita, ja asteikon ääripäät ovat erikoistapauksia. Demolombe ehdottaakin astevaihteluiden ilmaisemismahdollisuutta laajennokseksi järjestelmäänsä [Dem04].

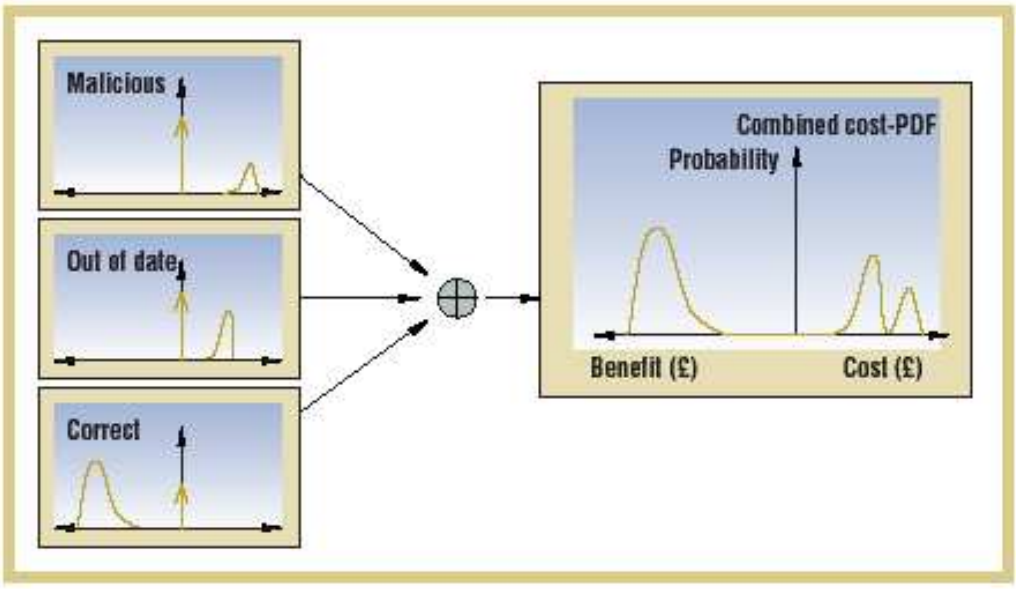
Luottamuksenhallinnan keskeinen ongelma koskee luottamustiedon pohjalta tehtävään päätöksentekoon. SECURE-projektin luottamuksen formalisoinnissa käsitellään päätöksen johtamista nykytiedosta ja tietojen päivittämistä kokemuksen perusteella. Toimijan a luottamus toimijaan b on luottamusarvo, joka kuuluu joukkoon T [EWN+03]. Joukon alkiot voivat olla esimerkiksi arvovälejä yksiulotteisella lukusuoralla, jolla suoran luvut ilmaisevat luottamuksen määrän. Mikäli luottamuksen määrän arvot asettuisivat välille "[0,1]" (0 edustaa vähäistä luottamusta, 1 suurta), yksi joukon T alkioista voisi olla "[0,1]". Tämä kuvastaa hyvin epävarmaa luottamusarviota, jossa toimijan varsinainen luotettavuus voisi havaintojen pohjalta olla mitä tahansa. Alkio "[0,49;0,51]" puolestaan kuvastaa keskinkertaista luottamuksen määrää, mutta arvion katsotaan olevan varsin tarkka [WCE+03].

Luottamusarvojen esitysmuotoa ei kuitenkaan rajata; se voi olla esimerkiksi hyvien kokemusten ja huonojen kokemusten lukumääräpari, mutta arvojoukolle on määriteltävä kaksi osittaisjärjestystä. Yksinkertaisin näistä kahdesta on arvion tarkkuuden mukainen osittaisjärjestys $(T, \sqsubseteq)$, joka järjestää luottamusarvot niiden ilmaisemien välien pituuden mukaisesti laskevaan järjestykseen. Tämän järjestyksen mukaan $m[0,4;0,6] \sqsubseteq [0,2;0,3]$, koska välin $[0,4;0,6]$ pituus on $0,6-0,4=0,2$, kun taas pienempi väli $[0,2;0,3]$ on pituudeltaan vain $0,3-0,2=0,1$ ja täten ilmaisu on tarkempi. Tämän osittaisjärjestyksen ensimmäinen eli epätarkoin elementti $\perp_{\sqsubseteq}$ on tässä tapauksessa koko arvovälin kattava $[0,1]$. Elementti kuvaa suurinta mahdollista epävarmuutta, ja se on kaikkien täysin tuntemattomien toimijoiden luottamusarvooletus. Koska SECUREn toimintaympäristö on hajanainen, epävarmuuden edustus on tarpeen; mikäli tietoa ei ole riittävästi, luottamuksenhallintajärjestelmän ei haluta välttämättä tekevän satunnaisia arvauksia vaan vastaavan yksinkertaisesti "en tiedä".

Arvion edustaman luottamuksen mukainen järjestys $\succeq$ on hieman monimutkaisempi. Edellisen esimerkin kannalta vertailu on suoraviivaista: $[0,2;0,3] \succeq [0,4;0,6]$, koska välin $[0,4;0,6]$ kaikki arvot ovat suurempia ja siten kuvastavat suurempaa luottamusta kuin välin $[0,2;0,3]$. Osittain tai täysin päällekkäisten välien, kuten $[0,2;0,3]$ ja $[0,24;0,26]$, järjestäminen ei sen sijaan onnistu. Arvojoukon T osittaisjärjestyksen $(T, \succeq)$ vaaditaan olevan täydellinen hila (*complete lattice*). Kahdelle arvolle X ja Y pätee tällöin $X \succeq Y$ jos ja vain jos X:n pienin yläraja ja suurin alaraja ovat vähintään yhtä suuret kuin Y:n vastaavat. Esimerkiksi pätee siis $[0,1;0,5] \succeq [0,2;0,6]$, mutta mikäli jälkimmäisen välin yläraja olisi 0,4, järjestystä ei voitaisi ilmaista.

Koska käsittelyn kohteena ovat näkemykset muista toimijoista, SECURElla termi "luottamus" sisältää myös osittain aliluvussa 4.1 määritellyn maineen merkityksen. Mallissa tunnetaan myös maineen ja suositusten käsitteet; suositukset ovat kolmansien osapuolten lausuntoja heidän kokemuksistaan ja luottamuksestaan tiettyyn kohteeseen, kun taas maine kattaa anonyymin yhteisöltä saadun luottamustiedon.

SECUREn luottamusmallin erikoisuus on riskianalyysin eksplisiittisyys [CGS+03].



Kuva 4: Esimerkki yhdistetyn hinta–hyöty -tiheysfunktion johtamisesta, kun yksittäisten lopputulosten (saatu väärät, vanhentuneet tai oikeat tiedot) tiheysfunktiot on laskettu [CGS+03].

Riski ilmaistaan hinta–hyöty -tiheysfunktion avulla (*cost probability density function*). Kaikki mahdolliset tietyn toiminnon lopputulokset tunnistetaan ja niille arvioidaan luottamusta ja muita kontekstitietoja käyttäen erisuuruisten haittojen (hinta) ja hyötyjen todennäköisyydet, jotka riippuvat lisäksi luottamuksen kohteelle lasketusta luottamusarvosta. Mikäli käyttäjä vastaanottaa toiselta käyttäjältä esimerkiksi jonkin yrityksen yhteystiedot, voidaan mahdollisina lopputuloksina pitää, että saadut yhteystiedot ovat joko oikeat, vanhentuneet tai pahantahtoisen käyttäjän jäljiltä täysin väärät. Mikäli tiedot ovat oikeat, ne eivät todennäköisesti ole haitallisia, vaan varsin hyödyllisiä.

Vanhentuneet tiedot tulkitaan SECUREn esimerkissä lähinnä haitalliseksi; niiden haitallisuus voi tosin olla vähäisempää, mikäli esimerkiksi tietojen pohjalta tehtyyn vikasoittoon vastaa taho, joka osaa kertoa soittajalle uuden numeron. Täysin väärä vastaus voi toisaalta olla epätodennäköisin tulos, mutta sen haitta on suurin. Epäluotettava taho voi antaa tahallaan vääriä vastauksia, joten sitä arvioitaessa väärän vastauksen vaihtoehto voi olla kaikkein todennäköisin. Kaikki lopputulosten arviot ilmaistaan omina tiheysfunktioinaan, jotka yhdistetään lopulta yhdeksi. Esimerkkiä havainnollistaa kuva 4.

Luottamuspäätösten tutkiminen tähtää päätösten tukemiseen koneellisesti, niihin vaikuttamiseen ja mahdollisesti myös niiden automatisointiin. Sen sijaan järjestelmän toimijoiden välisten luottamussuhteiden mallintaminen organisaatiotasolla ei välttämättä johda aktiivisen luottamussuhteiden hallintajärjestelmän luomiseen. Analyysi voi sen sijaan tukea parempaa järjestelmäsuunnittelua, jonka tuloksena tietoturvasta saadaan saumaton osa järjestelmää jälkikäteen lisätyn “lisätoiminnon”

sijaan.

Mallinnusjärjestelmä Security-aware Tropos [GMMZ04] laajentaa alkuperäistä Tropos-mallinnusjärjestelmää neljällä suhteella, jotka koskevat toimijoita ja tavoitteita, tehtäviä tai resursseja. Kaksi jälkimmäistä ovat välineitä jonkin tavoitteen saavuttamiseksi. Luottamus on kahden toimijan (*agent*) ja palvelun suhde: toimija A luottaa toimijaan B tavoitteen G suhteen. Delegaatio on samoin kolmiosainen suhde: A delegoi tavoitteen, suoritusluvan tai resurssin X eksplisiittisesti toimijalle B. Tarjonta on kaksiosainen suhde: toimija A tarjoaa mahdollisuuden saavuttaa tavoitteen suorittamalla tehtävän tai tarjoamalla resurssin. Omistus on samoin kaksiosaista: toimija A omistaa tavoitteen, tehtävän tai resurssin. Myös muu kuin omistaja voi tarjota esimerkiksi resurssia käyttöön; tämä vaatii omistajan suostumusta [GMMZ04].

Security-Aware Tropos erottaa toisistaan organisaation luottamusmallin ja toiminnallisen mallin. Luottamusmallissa kuvataan ainoastaan omistus- ja luottamussuhteita, kun taas tominnallisessa mallissa kuvataan delegaatiota ja tarjontaa, luvanannon ja Tropos-riippuvuuksien avulla. Mallinnusta varten voidaan käyttää sekä kuvallista että leksikaalista, formaalia merkintätapaa [GMMZ04]. Grønmo ja Oldevik toteavat mallinnuskielten arvioinnissaan, että kuvallinen merkintätapa sopii erityisesti nopeaan yleiskuvan hahmottamiseen, mutta skaalautuu suuriin kuvauksiin jokseenkin huonosti, kun taas formaali merkintätapa mahdollistaa monimutkaisten tai yksityiskohtaisten mallien viestinnän [GO05]. Security-Aware Tropoksen kuvallinen merkintätapa on hienoisesta epäintuitiivisuudestaan huolimatta tehokas yksinkertaisten mallien esittämisessä, mutta skaalautuvuusongelma on havaittavissa jo sairaalaympäristöön sijoitetun esimerkkitapauksen [GMMZ04] mallinnuksessa, jolloin mallikuvien yksiselitteisenä säilyttämisen takia resursseja joudutaan esittämään kuvasssa kerran jokaista käyttöluvanantoa kohden—esimerkiksi yksi lupa henkilökohtaisten tietojen käyttöön potilaalta terveysviranomaisille, yksi viranomaiselta sairaalalle ja yksi sairaalalta hoitavalle lääkärille. Yksityiskohdat, jotka voi täten olla vaikea mahduttaa kuvalliseen notaatioon, voidaan mallintaa yksinkertaisemmin formaalin merkintätavan kautta.

Security-Aware Tropos kuvaa luottamussuhteita tiedon käyttöluvan antamisen ja tehtävien delegaation yhteydessä. Tällöin toimijoiden suoran havainnoinnin sijaan ilmaistaan näiden rooleihin liittyvät luottamusuhteiden tarpeet. Malli ei ota kantaa siihen, luottaako tietyn roolin täyttävä toimija aktiivisesti luottamussuhteen toiseen osapuoleen, vaan toimija ei välttämättä ole edes tietoinen muista vaihtoehdoista ja voi siten olla myös hyvin epäluuloinen luottavaisuuden sijaan. Mikäli valinnan mahdollisuutta ei ole tai sitä ei havaita, luottamusta ei välttämättä ole eikä sitä tarvita. Security-Aware Troposin luottamusmalli auttaakin osoittamaan erityisesti epävarmuuskohdat järjestelmässä. Näitä voidaan paikata luottamuksella.

Luottamussuhteiden todellisen tilan ilmaisemiseen tarvitaan jokin toinen kieli. Esimerkiksi Sultan-luottamuksenhallintajärjestelmä [GS02] tallettaa luottamussuhteet politiikkalausekkeiden tapaan. Luottamussuhde on perusmuodossaan "toimija A luottaa toimijaan B arvolla n". Politiikan tapaan luottamuksella voi olla ehtoja. Toinen Sultanin tuntema käsite, suositus, voi toimia myös ehtona: "toimija A luottaa toi-

mija X:ään arvolla n jos toimija Y suosittelee X:ää arvolla m". Myös suositukset ilmaistaan politiikkoina, joiden ehtona voi olla muun muassa luottamus. Erillinen analyysijärjestelmä tunnistaa ristiriitaiset määritelmät. Sultan-luottamuskieltä voidaan käyttää myös yhdessä Ponder-politiikkakielen [DDLS01] kanssa, jolloin Ponder-lausekkeita voidaan käyttää Sultan-lauseiden ehtoina sekä päinvastoin.

Luottamuksen lähde vaikuttaa TuBEn luottamuksenhallinnassa hyvin vahvasti käsiteltäviin arvoihin. Tässä suhteessa voidaan nähdä web-palveluympäristön rahaa ja muita voimavaroja koskevien päätösten suurin vaikutusero verrattuna vaikutukseltaan kevyempiin luottamuspäätöstilanteisiin, kuten esimerkiksi vertaisverkkojen reitityksen tai kirjasuositusten "luotettavuuden" mukaan järjestämisen yhteydessä. Kun päätöksen vaikutus on pieni ja luottajan tuntemat luottamuksen kohteet ovat harvassa, järjestelmä voi tarjota tukea luottamuspäätöksen delegoimiselle. Se voi jopa tehdä tästä oletuksen, jolloin luottamuksesta tulee osittain transitiivista: mikäli käyttäjä X ilmaisee luottavansa käyttäjään Y ja käyttäjä Y on ilmaissut luottavansa käyttäjään Z, katsotaan käyttäjän X ilmaisevan samalla transitiivisen luottamuksensa käyttäjään Z. Tällöin esimerkiksi Z:n kirja-arvostelut tarjotaan X:lle "luotettuina". Luottamuksen transitiivisuus on suosittu tutkimuksen aihe juuri ympäristöissä, joissa luottamukseen liittyvä riskinotto koskee korkeintaan lievää ajan haaskausta. Suositusjärjestelmät ovat yleinen tutkimusaihe tällaisille luottamuksen verkoille [MB04, GKRT04]. Niiden yhteydessä on tutkittu myös maineen käyttämistä eräänlaisena järjestelmän sisäisenä valuuttana, jolloin myönteinen yhteisön toimintaan osallistuminen sallii sen palvelujen käytön [FKÖD04].

Käsitellyn luottamustutkimuksen yhteenvetona on joitakin tekijöitä koottu taulukkoon 1. Suositusjärjestelmien luottamustutkimusta edustaa Epinionsin luottamussuhteita tutkineet Guha *et al.* [GKRT04]. Taulukkoon on lisätty myös TuBEn luottamusmallin vastaavat arvot.

Sovellusympäristö kuvaa ympäristöä, jonka vaatimusten kautta luottamusta tutkitaan. McKnight ja Chervany esittävät hallintoesimerkin luottamuksen määrittelyn lopuksi, mutta hallinnon vaikutus ei näy erityisen vahvasti tuloksissa. Luottamuksenhallintaan keskittyneet järjestelmät, SECURE, Sultan ja TuBE, eroavat hienoisesti sovellusympäristöltään: SECURE keskittyy kaikkialla läsnäoleviin ja paikasta toiseen liikkuviin älykkäisiin laitteisiin kuten kämmenmikroihin (*Ubiquitous, Roaming Entities*). Sultan rajautuu esimerkkiensä perusteella sähköiseen kaupankäyntiin, mikä ei eroa suuresti TuBEn sovellusympäristöstä. *Luottamuksen kohde* tuo esiin mallien yleisyyden. Suurin osa keskittyy luottamaan ihmisiin, vaikkakin suositusjärjestelmissä välissä on käyttäjätunnus. Luottamuksen automatisoinnin yhteydessä ihmisen lisäksi käsitellään yleisempiä toimijoita, joita nimitetään yhteisesti agenteiksi. Security-Aware Troposin mallissa luottamus kohdistuu erityisesti rooliin, ei sen omistajaan. Luottamus voi kohdistua myös laajempaan yksikköön, osaorganisaatioon.

Luottamustiedon käsittely -sarake erottaa hajautetut mallit, joissa jokaisella luottajalla on oma subjektiivinen luottamustietokokoelmansa joita se käsittelee paikallisesti, ja keskitetyt mallit, joissa usean luottajan tietoja käsitellään keskitetysti.

Projekti	Sovellus-ympäristö	Kohde	Käsittely	Erottelu	Ilmeneminen
McKnight, Chervany	(hallinto)	ihminen	hajautettu	tyyppi	"luottava käytös"
Demolombe	logiikka	ihminen	hajautettu	ol. ominaisuus	uskomus
SECURE	mobiili-laitteet	agentti	hajautettu	aste	auktorisointi
Sec.-Aware. Tropos	järjestelmä-suunnittelu	rooli, org.	keskitetty	olemassa-olo	delegaatio, tiedonjako
Sultan	e-kauppa	agentti	keskitetty	aste	mm. auktorisointi
Guha *et al.*	suositus-järjestelmä	ihminen	keskitetty	aste	tiedon uskottavuus
TuBE	web-palvelut	agentti	hajautettu	aste	auktorisointi, personointi

Taulukko 1: Esitellyn luottamustutkimuksen yhteenveto.

McKnightin ja Chervanyn sekä Demolomben malleissa luottamustiedon käsittelyyn ei oteta kantaa, joten ne on sijoitettu hajautettujen käsittelijöiden kategoriaan ihmisten hajautettujen luottamuskäsitysten perusteella. Sultanissa luottamuslausunnot kootaan yhteen kantaan, joka kykenee tällöin analysoimaan esimerkiksi luottamuspolitiikkojen välisiä ristiriitoja. Security-Aware Troposilla mallinnetaan yksi, "ulkoinen" näkemys, joka kattaa koko organisaation. Suositusjärjestelmässä luottamussuhteet säilötään suositusjärjestelmän kanssa keskitetylle palvelimelle.

Luottamussuhteiden erottelu -sarake kuvaa luottamuksen ominaisuutta, jonka perusteella kahden luottamussuhteen voidaan sanoa olevan erilaisia. Security-Aware Tropos ei erottele luottamussuhteita millään muulla tavalla kuin niiden olemassaolon perusteella. Demolombe ei ota suoraan kantaa suhteiden luokitteluun, mutta määritelmän perusteella luottamus koskee tietyn mielenkiintoisen ominaisuuden olemassaoloa. Luottamuspäätöksiä käsittelevissä järjestelmissä keskeisin erottelija on luottamuksen aste tai määrä. *Luottamuksen ilmeneminen* -sarake kuvaa lähintä seurausta, jota luottamuksen olemassaololla tai positiivisella luottamuspäätöksellä on. Demolomben mallissa luottamus on uskoa jonkin toimijan tietyn ominaisuuden olemassaoloon, eikä tämän uskon seurauksiin puututa. Muissa malleissa luottamuksesta seuraa jokin McKnightin ja Chervanyn "luottavaa käytöstä" indikoivaa toimintaa: tieto hyväksytään, toiminto auktorisoidaan tai tietoa luotetaan toisen käyttöön. Luottava käytös on siten yläkäsite, jota muissa malleissa tarkennetaan. Sultan ei määrää, mitä luottamuksesta seuraa; Ponder-politiikkakieleen yhdistäminen mahdollistaa muun muassa auktorisoinnin. TuBE sallii auktorisoinnin lisäksi toiminnon muokkauksen luottamuksen määrän mukaiseksi esimerkiksi tarkentamalla tarkkailua tai vähentämällä saatavilla olevia resursseja.

4 Luottamuksen tietomalli TuBE-järjestelmässä

Kun luottamusta käytetään pääsynhallintaan liittyvissä päätöksissä, siihen vaikuttaa paitsi kohteen arvioitu luotettavuus, myös paikallinen järjestelmä. Jotkin pääsynhallinnalla suojatut toiminnot ovat tarkkaan vartioituja, kun taas toiset kuuluvat löyhemmän kontrollin piiriin. Toimintojen tarvitsema pääsynhallinnan taso voi myös vaihdella ajan myötä. Tässä luvussa esitellään luottamuspäätöksiin vaikuttavat tekijät ja niiden lähteet.

4.1 Luottamuspäätökseen vaikuttavat tekijät

TuBE-projektin mallin mukaan luottamuksenhallinnassa tasapainotellaan halukkuutta lisääviä tekijöitä, kuten vastapuolen luotettavuutta tai toimintojen suorituksen tarpeellisuutta, ja sitä hillitseviä tekijöitä, kuten riskiä. Tasapainoon voi vaikuttaa sekä pienentämällä riskiä että pyrkimällä vahvistamaan myönteisiä tekijöitä, esimerkiksi parantamalla luottajan käsitystä luottamuksen kohteesta. Tämä käsitys, maine, on yksi luottamuksen tärkeimmistä osatekijöistä.

Luottamus määritellään *halukkuudeksi sallia annetun partnerin suorittaa tietty toiminto tietyssä tilanteessa, kun huomioidaan kannusteet toiminnon suoritukselle ja siihen liittyvä riski* ([KVR05], taustalla muun muassa [MD95, JP04]). Luottaja on Internetissä toimiva palveluntarjoaja, ja toiminnot liittyvät tarjottujen palvelujen hyödyntämiseen. Luottamuksen kohde on yhteistyökumppani tai asiakas, jota edustaa tämän tunnistettavissa oleva verkkoidentiteetti.

Luottamuspäätös on joko myöntävä tai kieltävä. Se koskee tiettyä luottajaa, luottamuksen kohdetta, toimintoa ja järjestelmän tilaa. Riski ja tärkeys määritellään toimintokohtaisesti, ja järjestelmän tilaan kuuluu muun muassa tieto siitä, mitä rajoitteita on voimassa riskin rajoittamiseksi. Nämä rajoitteet voivat olla käyttäjäkohtaisia ja koskea esimerkiksi tämän tarkkailua järjestelmässä, asetettuja vakuuksia, sitovia sopimuksia tai esimerkiksi todistettua tietyn maan kansalaisuutta, mikä voi merkitä palveluntarjoajalle parempaa lain suojaa palveluntarjoajalle varalta. Koko järjestelmää koskeva riskien rajoittaminen, esimerkiksi vakuutus, otetaan huomioon riskien määrittämisessä. Kieltävän luottamuspäätöksen yhteydessä voidaan selvittää, olisiko mahdollista lisätä tilanteeseen joitakin tällaisia rajoitteita, joiden avulla päätöksestä tulisi myönteinen. Eri tekijöiden vaikutusta luottamuspäätökseen käsitellään tarkemmin luvussa 5.3.

Riski liittyy varallisuuteen ja voimavaroihin (*assets*) kohdistuvaan uhkaan [BS04]. Tietokantapalvelulle toimivat palvelimet ovat voimavara, jolloin esimerkiksi hyökkäyksen aiheuttama toimintahäiriö on vastaava uhka. Riski on uhan vakavuuden mitta. Lyhyeen toimintahäiriöön saattaa liittyä suhteessa pienempi riski kuin esimerkiksi tietokantapalvelimella olleiden luottamuksellisten tietojen leviäminen vääriin käsiin. Järjestelmiin voi kohdistua myös yllättäviä uhkia, mutta niiden arviointi on mahdotonta ja tärkeimmät riskit oletetaan kartoitetuiksi.

Toimintoon liittyy riskin lisäksi myös odotettu hyöty. Jotkin toiminnot voivat ol-

la jopa elintärkeitä toiminnan jatkumiselle, jolloin luottaja voi olla halukas sallimaan myös huonomaineisten yhteistyökumppaneiden suorittaa tällainen toiminto. Toiminnon **tärkeys** kuvastaa tätä tarpeellisuuden ja odotetun hyödyn yhteisvaikutusta myönteisen päätöksen kannustimina. Tietyn toiminnon riski- ja tärkeysarvot voivat molemmat olla korkeita tai matalia yhtä aikaa tai erikseen, eikä yhtä voi päätellä toisesta.

Maine on *käsitys, jonka toimija on aiemmilla teoillaan antanut pyrkimyksistään ja normeistaan* (myötäillen [MMH02]). Maine ilmenee vain yhteisössä joka tarkkailee jäseniään tavalla tai toisella, eikä sillä ole objektiivista merkitystä lähdeyhteisön ulkopuolella. Joskus tietyn yhteisön määrittämä maine voi olla lähempänä toimijan todellisia pyrkimyksiä ja normeja kuin toisen yhteisön, mutta toisaalta myös pyrkimykset ja normit voivat muuttua sen mukaan, missä yhteisössä toimitaan. Koska tietyn toimijan maine yhdessä yhteisössä kuitenkin antaa usein osviittaa siitä, millainen maine samalle toimijalle kehittyisi toisessa yhteisössä, mainetietojen viestimisestä yhteisöjen välillä voi olla hyötyä erityisesti kun toimija on tunnettu pitkään lähdeyhteisössä, mutta on vasta aloittamassa toimintaansa kohdeyhteisössä. Maine perustuu kokemukseen, ja kokemusta tietyn toimijan suhteen kertyy vain tämän kanssa asioivalle osalle yhteisön jäsenistä. Myös tämä osayhteisö joutuu viestimään mainetiedon jotenkin yläyhteisölleen.

Maineen viestinnässä käytetään **lausuntoja** (engl. *positive/negative recommendation*). Lausunto ei välttämättä vaikuta suoraan vastaanottavan yhteisön mainetietoihin, vaan lausuntojen hyödyntämisessä käytetään yhteisön omaa harkintaa, jota toteuttaa yhteisöjen välinen mainejärjestelmä—järjestelmä, joka kerää, yhdistää ja erityisesti levittää mainetietoa [RZFK00]. Joitakin lausuntoja voidaan jättää huomiotta sen perusteella, että lausunnon antaja ei ole uskottava. Joitakin lausuntoja ei voida käyttää siksi, että lausunto koskee liian erilaista toimintojen kontekstia, jotta se voitaisiin muokata vastaanottavan yhteisön käyttöön sopivaksi. Lausuntoja voidaan myös liittää mainetietokantaan "epävarmana tietona", jolloin niitä painotetaan vähemmän kuin omakohtaista kokemusta. **Mainejärjestelmät** ehdottavat eri tapoja, joilla lausuntojen pohjalta voidaan tuottaa yhteisölle yhtenäinen mainekäsitys.

4.2 Toimijoiden tunnistaminen

Jotta luottamustieto, erityisesti maine, voidaan liittää tiettyyn toimijaan, tulee tämä voida tunnistaa. Tunnistusmenetelmien valikoima yltää käyttäjätunnuksen ja salasanan yhdistelmistä aina X.509-varmenteisiin [MWR89]. Pitkäaikaiset identiteetit ovat erityisesti yksi hyvin toimivan mainejärjestelmän edellytyksistä [RZFK00]; mikäli mainetietoa ei ole tukemassa luottamuspäätöstä, on luottamusjärjestelmän toiminta uhattuna.

Tunnistus voidaan sitoa henkilöllisyyteen, esimerkiksi ihmiskäyttäjän henkilötunnukseen. Se voi myös perustua yksinkertaisesti toimijan erottamiseen muista toimijoista, jolloin tunnistaminen on riittävän tarkkaa mikäli aiemmat kokemukset toimijasta voidaan yhdistää toisiinsa. Tällä yhdistämisellä on eri kattavuusastei-

ta. Esimerkiksi monet ilmaiset sähköpostipalvelut mahdollistavat sen, että yhdellä toimijalla on monta eri käyttäjätunnusta. Tällöin eri tunnuksia ei voida yhdistää toisiinsa.

Toisaalta uusien käyttäjien luonti voidaan pyrkiä järjestämään siten, että useiden käyttäjätunnusten haaliminen tulee kalliiksi, vaikka tuskin koskaan täysin mahdottomaksi. Esimerkiksi yksittäisten käyttäjätunnusten maksullisuus pienentää kiinnostusta luoda useita, muttei suinkaan poista sen mahdollisuutta. Ratkaisu ei välttämättä kuitenkaan sovi esimerkiksi kirjakaupalle, joka haluaa pitää palveluun liittymisen kynnyksen mahdollisimman matalana asiakkailleen, eikä voi siten käyttää erityisen tiukkoja kriteerejä asiakkaidensa valitsemiseen. Yksityisasiakkaat eivät välttämättä suostu läpikäymään henkilöllisyyden todistamisen raskasta prosessia vain päästäkseen katsomaan esimerkiksi pienen verkkokaupan sivustoa, joten heille joudutaan avoimuuden nimissä myöntämään käyttäjätunnuksia vähäisemmin perustein.

Usean tunnuksen luontia voidaan tehdä vähemmän houkuttelevaksi myös esimerkiksi laskemalla uuden käyttäjän oletusmainetta, mutta korjaukset tehdään jälleen myös uusien liittymisen joustavuuden kustannuksella. Mikäli uuden käyttäjän oletusmaine lasketaan niin alas, ettei se riitä pääsyyn palvelun keskeisiin osiin, ei asiakkaan kiinnostus palvelua kohtaan välttämättä kestä niin kauan, että tämä jäisi odottamaan maineensa kehittymistä. Järjestelmän suunnittelussa tulisi ottaa huomioon monella tunnuksella toimivien käyttäjien mahdollisuus, ellei tätä ole estetty, lähinnä sitomalla tunnukset henkilöllisyyteen tavalla tai toisella.

ISO-standardi X.509 (ISO DIS 9594-8) [IET98, MWR89] määrittää varmennejärjestelmän, joka mahdollistaa käyttäjätunnuksen liittämisen luontivaiheessa tiettyyn henkilöön. Tämä kuitenkin vaatii luotettua kolmatta osapuolta, varmentajaa, joka lisää digitaalisen allekirjoituksensa X.509-varmenteeseen varmistuttuaan ensin hakijan henkilöllisyydestä, sekä varmistaa julkisen avaimensa saatavuuden varmenteen todentamista varten. Käyttäjän kannalta X.509-varmenteen hankinta on suhteellisen monimutkainen ja kallis toimenpide, erityisesti jos sen ainoa hankintasyy on web-palvelu, jota käyttäjä kenties kokeilee pari kertaa ennen siirtymistään muualle.

Yritysasiakkailla käyttäjätunnuksen myöntäminen voi perustua tehtyyn sopimukseen, jolloin toinen osapuoli joudutaan myös tunnistamaan. Yhteistyö voi odotetusti kestää pitkään ja sisältää useita transaktioita, jolloin toimintaa aloitettaessa maksettava hinta luotettavammasta tunnistusjärjestelmästä ei ole enää yhtä merkittävä. Yritysten välisessä yhteistyössä on jo tietoverkkojen ulkopuolella käytetty mainejärjestelmiä [Tan03], jotka maanlaajuinen rekisteröinti ja sen kautta saatavat yksiselitteiset tunnisteet tekevät mahdolliseksi.

Mikäli toimijan tunnistetta palvelussa ei sidota reaalimaailman yksilön henkilöllisyyteen, järjestelmän käytön aloittaminen saadaan joustavammaksi. Joustavuuden kääntöpuolena on tyhjästä aloittamisen mahdollistaminen, sillä järjestelmä ei voi havaita varmasti kahden eri käyttäjätunnuksen yhteyttä samaan käyttäjään. Monen tunnuksen luomismahdollisuus heikentää huonon maineen tehokkuutta sanktiona, sillä maineen laskiessa uuden käyttäjän mainetason alapuolelle käyttäjä voi luoda uuden tunnuksen, jonka maine on täten oletustasolla.

Web-palvelukontekstissa käyttäjän henkilöllisyyden todentaminen henkilöllisyystunnuksen tarkkuudella on raskas, ellei mahdotonkin operaatio. Lisäksi luottamuksen implisiittisenä kohteena on varsinaisen käyttäjän lisäksi myös joukko muita toimijoita, jotka voivat vaikuttaa viestinvaihtoon käyttäjän ja palvelun välillä. Esimerkiksi web-palvelu, jota kutsutaan toisesta web-palvelusta tietyn käyttäjän aloitteesta, ei joudu luottamaan pelkästään käyttäjään vaan myös kutsuvan palvelun tarjoajaan ja siihen, että tämän palvelu edustaa käyttäjän tahtoa. Lisäksi troijalaiset hevoset ja muut haittaohjelmat voivat vaikuttaa käyttäjän havaittavaan toimintaan, jolloin niiden tekijät ja käyttäjät kuuluvat myös osaltaan implisiittisen luottamuksen piiriin. Mikäli jokin näistä aiheuttaa epäilyttävän palvelupyynnön tietyn käyttäjän nimissä, ei vastaanottava palveluntarjoaja voi yleisessä tapauksessa reagoida muuten kuin olettamalla, että käyttäjä on itse vastuussa pyynnöstä, jolloin tämän maine todennäköisesti laskee.

X.509-varmenteet yhdistävät kryptografisesti käyttäjää edustavan julkisen avaimen tämän henkilöllisyyteen. Vaikka järjestelmä toimisi ilman X.509-varmenteita, se voi kuitenkin hyötyä esimerkiksi mahdollisuudesta erotella erilaisia käyttäjäryhmiä toisistaan, yliopisto-opiskelijoista Plussa-kortin haltijoihin. SPKI/SDSI-varmenteilla voidaan ilmaista haltijan ominaisuuksia, kuten “yliopisto-opiskelija”, ottamatta kantaa tämän henkilöllisyyteen [Ell04]. Karabulutin ja Biskupin hybridimalli pyrkii yhdistämään X.509- ja SPKI/SDSI-mallien parhaat puolet [Kar03].

Palvelut kuten laskujen sähköisen lähetyksen mahdollistava Netposti [Suo05] käyttävät suomalaisten, verkkopalveluja tarjoavien pankkien sisäänkirjautumisjärjestelmää apunaan käyttäjien tunnistamisessa, jolloin käyttäjän ei välttämättä tarvitse käydä fyysisesti postissa todistamassa henkilöllisyyttään. Netposti-palvelu luottaa tällöin pankkien tunnistusmekanismin olevan riittävä käyttäjän henkilöllisyyden toteamiseksi. Tällöin Netpostin tunnukset voidaan yhdistää välillisesti henkilöllisyyteen, minkä seurauksena useiden käyttäjätunnusten luominen saman henkilön käyttöön voidaan tehdä jo huomattavan hankalaksi. Myös sähköiset henkilöllisyystodistukset tekevät tuloaan. Nämä kaikki voivat toimia eräänlaisina henkilöllisyyden todistavina varmenteina, joita voidaan joko käyttää tunnistamisen välineenä suoraan tai vain uusien paikallisten käyttäjätunnusten luomisen yhteydessä.

Jatkossa oletamme, että käyttäjää edustaa tunnus, jonka tarkistaminen on hoidettu palvelun kannalta riittävällä tarkkuudella, varmenteen, salasanan tai muun menetelmän avulla. Tätä tunnusta ei välttämättä voi jäljittää tiettyyn henkilöön, mutta sen avulla joukko toimintoja voidaan yhdistää tiettyyn tunnukseen. Oletamme myös, että uusien tunnusten luonti on jokseenkin vaikeaa, jolloin käyttäjät yleensä ottaen pitäytyvät yhden tunnuksen luontiin eivätkä ohita mainejärjestelmää luomalla uusia käyttäjiä maineensa laskiessa. Tämä mahdollistaa jatkuvuuden sessiosta toiseen, mikä on välttämätöntä kokemustiedon pitkäaikaisen keräämisen kannalta. Sanaa “käyttäjä” käytetään yhteydessä yhteen käyttäjätunnukseen.

4.3 Mainetiedon lähteet

Luvussa 4.1 luottamuspäätökseen vaikuttavat tiedot jaettiin neljään osaan: käyttäjän maineeseen, toimintokohtaisiin tietoihin kuten sen riskiin ja tärkeyteen luottajan kannalta, sekä kontekstiin, jossa luottamuspäätös tehdään. Näistä mainetieto on selkeimmin jatkuvia päivityksiä vaativa, konteksti vaatii päivityksiä tilanteiden muuttuessa ja riski- sekä tärkeystietojen voidaan olettaa olevan jokseenkin pysyviä.

Tietyn käyttäjän luotettavuusarvioita varten tarvittavat mainetiedot voidaan koota karkeasti jakaen kolmesta eri lähteestä. Arvio voi perustua kolmansien osapuolten lausuntoihin käyttäjän luottamuksesta eli tämän ulkoiseen maineeseen tai paikallisesti valvonnan pohjalta koottuun tietoon käyttäjän käytöksestä tässä järjestelmässä eli paikalliseen maineeseen. Lisäksi se voi ennakkotietojen puutteessa pohjautua yksinomaan järjestelmään asetettuun tuntemattoman käyttäjän oletusmaineeseen. Viimeistä vaihtoehtoa voidaan myös kutsua kyseisen järjestelmän tai sen ylläpitäjän luottavaisuudeksi, sillä se vastaa ylläpitäjän käsitystä jokseenkin keskimääräisen uuden käyttäjän luotettavuudesta.

Mikäli käyttäjä tunnetaan samalla tai usealla mutta toisiinsa yhdistettävissä olevalla tunnuksella eri järjestelmissä, ne voivat vaihtaa keskenään mainetietoa ja hyötyä siten toistensa paikallisista kokemuksista. Mainetiedon ylläpito ei ole uusi ajatus yritysmaailmassa; erilaiset yritysrekisterit ovat jo pitkään pitäneet kirjaa esimerkiksi yritysten arvioidusta maksukyvystä verotietojen perusteella. Tällaisten palvelujen avulla on voitu hillitä epävarmuutta elektronisessa kaupankäynnnissä myös yli valtiorajojen [Tan03]. Yksityishenkilöille vastaavia palveluja on rajatummin, ja niihin liittyy erityisiä haasteita yksityisyyden suojaamiseksi [SJ04].

Yksityisyyden suojan kannalta lienee parasta, että käyttäjä voi itse kontrolloida, mitä mainetietoa hänestä kulkee järjestelmästä toiseen. Tämä voidaan mahdollistaa esimerkiksi muuntamalla lausunnot varmennemuotoisiksi “todistuksiksi” hyvästä käytöksestä, jotka käyttäjä toimittaa itse järjestelmälle halutessaan. Päätökseen voi vaikuttaa muun muassa järjestelmän luotettavuus käyttäjän silmissä [Obr04, WSJ00]. Toinen lähestymistapa mahdollistaa käyttäjän eri tunnuksilla erilaisissa järjestelmissä kokoaman maineen yhdistämisen kontrolloidusti käyttäjän valvonnassa [SJ04].

Käyttäjälähtöisessä maineviestinnässä kääntöpuolena on tiedon epätäydellisyyden korostuminen, sillä käyttäjä todennäköisesti haluaa estää itselleen epäedullisten tietojen leviämisen. Tällöin ulkoisista lähteistä saatu mainetieto on auttamatta varsin yksipuolista. Järjestelmän hallinnoima mainetiedon levitys, johon käyttäjä ei voi vaikuttaa yhteisöön liittymisensä jälkeen, onkin yhä hyvin yleinen lähestymistapa mainejärjestelmissä. Esimerkiksi verkkohuutokauppa eBay [eBa05] näyttää maailmalle kaikkien järjestelmän kautta kauppaa käyneiden paikallisen maineen, esitettynä muiden käyttäjien antamien pisteytysten ja lausuntojen muodossa. Käyttäjistä näytetään tällöin kuitenkin vain käyttäjätunnus, ja tunnus yhdistetään tunnistaviin yhteystietoihin vasta kun kaupankäynti käyttäjän kanssa tätä vaatii.

Järjestelmän hallitsema maineviestintä varmistaa, että kaikki tarkkailtujen maine-

järjestelmien antamat lausunnot ovat käytettävissä. Tosin, kuten edellä todettiin, käyttäjä voi yhä esiintyä eri järjestelmissä eri nimillä, jolloin muiden mainejärjestelmien lausunnoista ei ole hyötyä ellei käyttäjätunnuksia voida yhdistää toisiinsa luotettavasti.

Kukin mainejärjestelmä kattaa yhden yhteisön, jonka jäsenet ovat yhtä mieltä maineen määräytymisperusteista kyseisessä järjestelmässä. Mekanismi, jolla eri mainejärjestelmät voivat vaihtaa lausuntoja keskenään, vaatii maineen esitysmuotojen yhteisen tulkinnan lisäksi myös sopimuksen siitä, miten maine määräytyy ja mistä. Mikäli yksi mainejärjestelmä kerää ainoastaan tietoa siitä, miten hyvin sen arvioimat käyttäjät ovat maksaneet laskunsa, sen lausunnot eivät välttämättä ole lainkaan hyödynnettävissä mainejärjestelmässä, jossa arviointiperusteena on hyödyllisimpien tuotearvostelujen kirjoittaminen tai standardinmukaisuus viestinnässä. Esimerkiksi kirjastopalvelu voisi sen sijaan hyötyä sekä laskunmaksutiedosta että hyvien arvostelukirjoittajien tunnistamisesta, mutta sen tulee voida luottaa siihen, ettei näiden tietojen lähdejärjestelmissä hyvä maine kerry paljon nopeammin kuin sen omassa arvioinnissa. Muutoin sen täytyy tulkita lausunnot vähemmän merkitseviksi, etteivät ne saa liiallista vaikutusta palvelun paikallisessa näkemyksessä.

Kolmansien osapuolten antamia lausuntoja olennaisempaa mainetietoa saadaan keräämällä omaa kokemustietoa, jolloin mainekäsitystä voidaan hienojakoistaa toimintokohtaiseksi paikallisen järjestelmän toimintojaon mukaan. Kokemustietoa saadaan tarkkailemalla käyttäjien toimintaa toisaalta suhteessa erilaisiin politiikkasääntöihin, toisaalta suhteessa käyttäjien aiempaan toimintaan. Sääntöjen rikkomisen vaikutus maineeseen on suoraviivaisen kielteinen, mutta miten reagoida käyttäjään, joka toimii epätavallisesti vain suhteessa aiempaan käytökseensä? Käytöksen raju muutos voi olla oire esimerkiksi siitä, että käyttäjä ei enää hallitse palvelua käyttävää ohjelmistoaan tai että hänen käyttäjätunnuksensa on tavalla tai toisella jonkun muun käytössä, jolloin ongelma liittyy myös tunnistamiseen ja välikäsien näkymättömyyteen, mitä käsiteltiin luvussa 4.2. Vaikka tällainen epäily ei välttämättä intuitiivisesti vaikuttaisi luottamukseen, jota järjestelmä tuntee käyttäjää kohtaan, järjestelmän tulee kuitenkin suojata itseään tätä uutta, arvaamattomampaa toimintaa vastaan. Tällöin maineen negatiivinen muutos kuvastaa kenties enemmänkin käyttäjätunnuksen uskottavuuden laskua kuin sen edustaman, järjestelmän kannalta varsin näkymättömän käyttäjän oletettujen arvojen ja normien muutosta.

Käyttäjien tarkkailua on automatisoitu aiemmin lähinnä järjestelmään tunkeutumisen ja tietomurtojen havaitsemiseksi. Tarkkailua voidaan tehdä joko reaaliaikaisesti tai lokien perusteella jälkeenpäin; erityisesti edellinen lähestymistapa kärsii suorituskyvyn asettamista rajoituksista. Kokemustiedon keräämistä käsitellään lähemmin luvussa 5.8.

Kun järjestelmään saapuu käyttäjä, josta ei ole saatavilla mitään aiempia tietoja, tälle täytyy asettaa jonkinlainen järjestelmäkohtainen oletusmaine. Oletusmaineen taso määrää, miten hankalaa uusien käyttäjien on käyttää palveluja ja miten kauan heidän on kerättävä mainetta ennen pääsyä järjestelmän hyödyllisiin palveluihin. Toisaalta korkea oletusmaine myös altistaa järjestelmää väärinkäytöksille, sillä ole-

tusmainetta alempi maine voidaan aiemmin tehdyn oletuksen mukaan nostaa oletusmaineeksi luomalla huonomaineisen käyttäjätunnuksen tilalle uusi. Oletusmaineen tasoa määrättäessä on tarpeen muistaa, etteivät kaikki "uudet" käyttäjät ole itse asiassa järjestelmässä ensimmäistä kertaa, mikäli yksi toimija voi luoda itselleen useita käyttäjätunnuksia.

Vaikka luottamuspäätös tehdään tapauskohtaisesti, on luottamuksen yleistäminen jollakin tasolla tarpeen. Mikäli jokainen käyttäjä on toistuvasti "uusi" pyytäessään jotakin palvelua, jota ei aiemmin ole pyytänyt, kokemuksen keräämisen vaikutuksia päästään hyödyntämään vain hitaasti — mainearvojen toimija–toiminto -matriisi on harva.

Kenties siis toimintoon ja kontekstiin sidottua arviota voidaan käyttää hyväksi jossakin toisessa tilanteessa, jossa luottamuksen kohde on sama? Tällainen yleistäminen lienee mahdollista sopivin rajoituksin. Toiminnot eivät vastaa suoraan toisiaan, kuten edellä todettiin, mutta esikuvana voidaan käyttää ihmisiä, jotka kehittävät kuitenkin yksittäisten arvioiden pohjalta yleiskuvaa luottamuksen kohteesta. Jossakin vaiheessa ihminen katsoo tuntevansa jonkin henkilön riittävän hyvin ja ilmaisee kenties hänen olevan mielestään yleisesti ottaen luottamuksensa arvoinen, tarkentamatta tiettyyn toimintaan. Tällöin kyse on eräänlaisen hyväntahtoisuuden ja kyvykkyyden, tai välinpitämättömyyden ja kyvyttömyyden, mielikuvan liittämisestä kohteeseen. Tämä mielikuva toimii pohjana luottamuksen ja erityisesti maineen määrittämiseksi uusissa konteksteissa.

Marsh on todennut kokemuksen vaikuttavan ihmisluottajan asenteisiin: onnistunut riskinotto positiivisen luottamuspäätöksen jälkeen yllyttää ottamaan riskin myös toistamiseen, kun taas epäonnistuminen tekee varovaiseksi [Mar94]. Tämä näkyy luottamuksenhallintajärjestelmässä suorimmin maineen muutoksessa kokemuksen perusteella ja päivitetyn mainetiedon käytössä myöhemmissä luottamuspäätöksissä. Vaikutus on kuitenkin myös tiettyä luottamuksen kohdetta koskevia päätöksiä yleisempi, ja kokemus voisikin vaikuttaa myös uusien käyttäjien oletusmaineeseen.

Oletusmaine, luottavaisuus, voisi siis vaihdella jokseenkin hitaasti kokemusten karttuessa, laskien pettymysten jälkeen ja nousten hitaasti hyvien kokemuksien aikana. Kuitenkin siitä tai tutummassa tilanteessa muista luottamuksen tyypeistä johdettavaan lopulliseen luottamispäätökseen voi vaikuttaa myös järjestelmän muu tila, minkä vaikutus voi ilmetä esimerkiksi yrityksessä riskinottokyvyn heikkenemisenä talouden tiukentuessa. Lisäksi tieto uudesta nopeasti levinneestä viruksesta tai muu epäilys ulkomaailman muuttumisesta äkillisesti vihamielisemmäksi voi luoda tarpeen luottavaisuuden laskemiseksi väliaikaisesti. Tällöin, vaikka luottamus ei varsinaisesti ole muuttunut, luottamispäätökset saattavat olla useammin kielteisiä. Tämä tilanteesta johtuva korjaus ei sisälly perusluottamukseen eli luottavaisuuteen, vaikka se vaikuttaakin päätöksiin yleisesti. Se erotetaankin oletusmaineen arvosta selkeyden vuoksi, ja kuuluu mainevaikutteisen kontekstin piiriin.

Maineella on erikoinen vaikutus myös luottamuksen kohteisiin, silloin kun järjestelmän toimijoita ohjaavat yksityiset ihmiskäyttäjät, jotka tietävät tai kykenevät päättelemään mainetasonsa järjestelmässä. Korkean maineen muodossa osoitettu luot-

tamus vaikuttaa keskimäärin ihmisiin kimmokkeena käyttäytyä luottamuksen arvoisesti; vastaavasti käyttäjä voi reagoida luottamuksen puutokseen käyttäytymällä tavallista epäluotettavammin, mikä voi luoda noidankehän [MD95].

4.4 Toimintokohtaiset tiedot: riski ja tärkeys

Toimintokohtaiset tiedot ovat toimijakohtaisten tietojen tapaan tärkeä tekijä luottamuspäätöksessä. Kullekin toiminnolle määritetään sen toteutumiseen liittyvä riski sekä toiminnon tärkeys luottajan kannalta. Riskin merkitys on todettu useissa lähteissä [JP04, ETW04], mutta tärkeys saatetaan jättää riskin peilikuvaksi.

Riskin ja tärkeyden määritteleminen ei välttämättä ole aivan suoraviivaista. Vaikka esimerkiksi rahallisen riskin arviointi olisi suoraviivaista, toimintoihin voi liittyä myös muunlaisia riskejä, kuten luottamuksellisen tiedon vuotaminen tai negatiivinen julkisuus, joiden vertailu rahamääriin ei välttämättä ole tarkoituksenmukaista. Lisäksi kuhunkin toimintoon liittyvä rahallinen tai muu riski voi vaihdella parametrien mukaan, kuten esimerkiksi jos toiminto määritetään tarkkuudella “osto luotolla”: tällöin toiminnon kohde, oli se sitten jauhosäkki tai rannekello, vaikuttaa hintansa kautta rahalliseen riskiin. Riskin arvioinnin helpottamiseksi olisi toivottavaa, että käytettävissä olisi mahdollisimman tarkka semanttinen tieto sekä toiminnosta että sen parametreistä, ja toisaalta että järjestelmä tukisi erilaisten toimintojen ryhmittelyä luokiksi, joihin liittyy samansuuruinen riski jokseenkin samoista syistä. Myös tärkeys voi riippua paitsi odotetuista tuloista, myös toiminnan jatkuvuuden kannalta todetusta yleisemmästä tarpeellisudesta. Esimerkiksi myyntitoiminnon lopullinen hyöty voi jäädä hyvin vähäiseksi, jos sitä seuraava perille kuljetuksen toiminto jää toteutumatta.

Riskitiedot, kuten myös toimintojen tärkeydet, täytynee syöttää jokaiseen järjestelmään erikseen, sillä toimintojako sekä kuhunkin liitettävä riski ja tärkeys voivat vaihdella järjestelmästä toiseen. Riskianalyysiin on erilaisia apuvälineitä. Esimerkiksi CORAS-riskinhallintajärjestelmä [BS04] tukee riskien tunnistamisen ja luokittelun lisäksi myös vastatoimenpiteiden etsimistä kullekin riskille. CORASin käyttäjä asettaa kullekin riskille vakavuusarvon ja toteutumistodennäköisyyden, joiden perusteella johdetaan arvo kullekin riskille. Jotta eri tekijöiden arvojen vertailu luottamuspäätöksen yhteydessä mahdollistuu, tulee riski- ja tärkeysarvojen semantiikan vastata maineen semantiikkaa riittävällä tasolla.

Riskin ja tärkeyden arvo voidaan saada myös osin ulkopuolelta. Erilaisten toimintojen yleisiä riski- ja tärkeysarviointipohjia voidaan tallettaa esimerkiksi tietyn yritysyhteisön käyttöön. Lisäksi yhteisö voi vaikuttaa paikallisiin riski- ja tärkeysarvoihin esittämällä mielipiteensä oman kokonaistoimintansa kannalta erilaisten toimintojen tärkeydestä ja kenties myös riskistä. Kaikki nämä tiedot tulee kuitenkin tarkistaa paikallisesti, sillä luottamuksenhallinnan ensisijainen tehtävä on suojata paikallista järjestelmää. Mikäli yhteisön sallitaan sanella esimerkiksi toimintojen tärkeystiedot, se voi pakottaa jäsenensä tekemään paikallisesti katsottuna vaarallisia päätöksiä, eikä suojattava järjestelmä ole tällöin enää täysin autonominen.

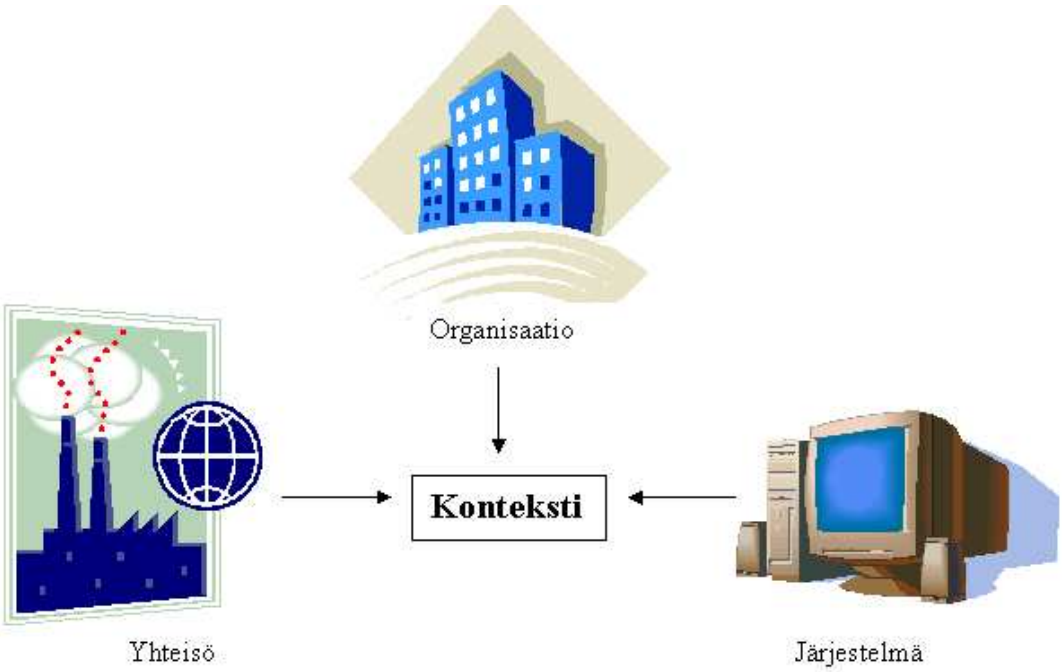
CORAS-riskinhallintajärjestelmää käytettäessä voidaan valita keinoja riskien pienentämiseen. Jotkin näistä keinoista pienentävät riskiä kerralla pysyvästi, ja niiden vaikutus voidaan ottaa huomioon ennen riskiarvojen syöttämistä luottamuksenhallintajärjestelmään. Jotkin keinot, kuten esimerkiksi automaattisen tarkkailun tehostaminen hiljaisina aikoina, vaikuttavat riskiin vain ajoittain ollessaan käytössä. Luottamuksenhallintajärjestelmä voi ottaa huomioon kulloinkin käytössä olevat riskiin tai muihin luottamuspäätöksessä käytettäviin suureisiin vaikuttavat toiminnot ja rajoitukset, kunhan sille on syötetty tarvittavat tiedot kunkin vaikutuksesta ja tieto siitä, milloin kukin on käytössä. Vastaavasti luottamuksenhallintajärjestelmä voi ottaa joitakin toimenpiteitä käyttöön automaattisesti, mikäli positiivinen luottamuspäätös tällaista vaatii. Mahdollisuuksia tutkitaan tarkemmin luvussa 5.4. Kaikki väliaikaisia muutoksia koskevat tiedot kuuluvat kontekstinhallinnan piiriin, eikä niitä tallenneta pysyvämpien riski- ja tärkeysarvojen päälle. Kontekstitiedon lähteitä käsitellään seuraavassa aliluvussa.

4.5 Konteksti

Joskus on tarpeen vaikuttaa luottamuspäätökseen väliaikaisesti. Järjestelmä voi esimerkiksi olla palvelunestohyökkäyksen alla tai toipumassa riskinottokykyä laskevasta kassavajeesta. Toisaalta esimerkiksi tietyn käyttäjän maineen voidaan haluta olevan korkeampi niin kauan kun hän kuuluu ryhmään “kanta-asiakkaat”. Palvelunestohyökkäys merkitsee väliaikaisesti suurempaa riskiä, kassavaje voi vaikuttaa lisäksi myyntitoimintojen tärkeyteen.

Kontekstiasetukset voidaan jakaa kategorioihin myös niiden lähteen mukaan. Kontekstilähteiden tyypit on esitetty kuvassa kuvassa 5. Jotkin asetukset liittyvät järjestelmän sisäiseen tilaan, kuten edellä mainittu hyökkäys tai yhtä käyttäjää koskeva toiminnan rajoittaminen, jotta tätä koskeva luottamuspäätös saataisiin riskiä pienentämällä myönteiseksi. Jotkin asetukset puolestaan liittyvät luottavan organisaation tilaan, kuten kassavaje tai se, että tietty käyttäjä on asetettu kanta-asiakkaaksi. Yhteisötasolla voidaan tunnistaa myös kolmas kontekstiasetusten kategoria, yhteisön tila. Mikäli yhteisö on hajoamassa, joidenkin toimintojen tärkeyspainotukset voivat erota huomattavastikin painotuksista, jotka liittyvät juuri luotuun yhteisöön. Esimerkkejä on monenlaisia, kuten myös niiden vaikutuksia.

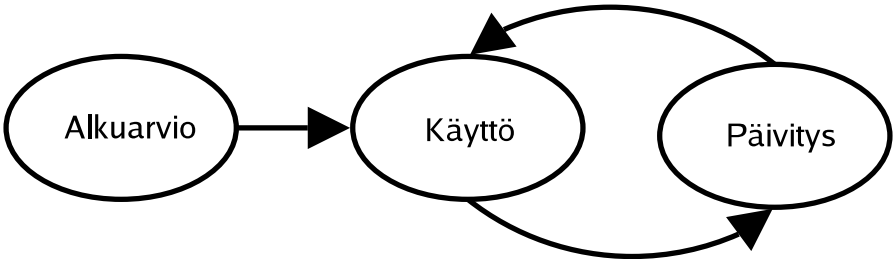
Tässä luvussa nimettiin luottamusarvion perusteena toimivan tiedon tekijät ja kuvattiin niiden lähteitä. Lopulta tiedoista pelkistetään päätös, joka on tarkimmillaankin muotoa “luotan”, “en luota” tai mahdollisesti “luotan, mikäli seuraavat toimenpiteet otetaan käyttöön”. Eri tekijöiden ylläpito mahdollisimman tarkkoina ennen tätä päätöstä helpottaa kuitenkin niiden ylläpitoa ja tuo eri tekijöiden vaikutukset päätökseen näkyville. Seuraavassa luvussa käsitellään tarkemmin näiden eri tietojen käyttöä luottamuspäätöksessä.



Kuva 5: Kontekstilähteiden tyypit.

5 Luottamuksen toiminnallinen malli TuBE-järjestelmässä

Luottamuksen hallinnan lisäämistä osaksi järjestelmää perusteltiin edellä erityisesti järjestelmän mukautuvuuden parantamisella. Tältä pohjalta olisi epäjohdonmukaista, mikäli luottamusarvo, kun se kerran on asetettu jonkin lähteen pohjalta, pysyisi aina samana. Luottamuksen hallinnan elinkaari ulottuukin laajimmillaan ennakoivasta luottamuksen arvioinnista ennen käyttäjän ensikirjautumista aina lokitietojen säännölliseen tarkastukseen, jolloin viimeisin yhteys on jo saattanut päättyä. Luottamuksen käyttö ja seurauksien tarkkailusta saatava kokemus seuraavat toisiaan syklissä, kuten kuva 6 esittää. Tämä luku käsittelee TuBE-luottamuksenhallintajärjestelmän osien toimintaa ja niiden merkitystä paikallisessa päätöksenteossa.



Kuva 6: Luottamuksen hallinnan elinkaari.

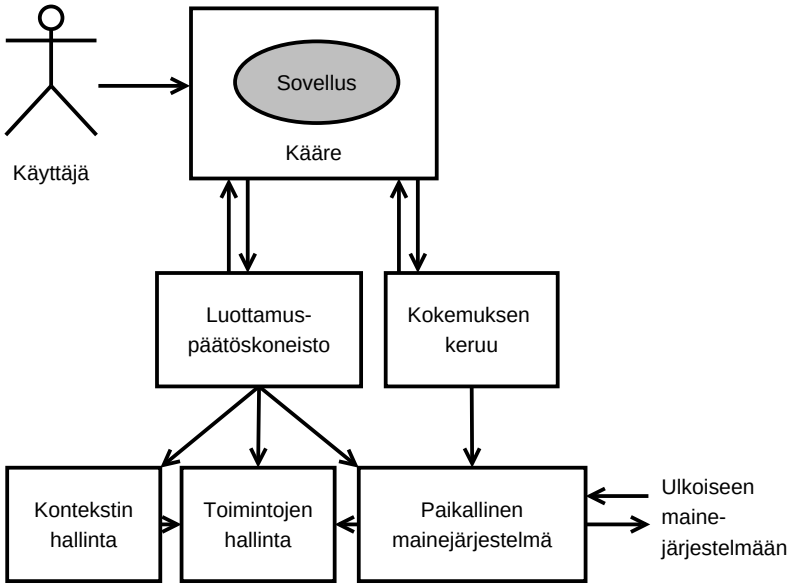
5.1 Luottamuksen hallinnan elinkaari

Luottamus rakentuu ajan myötä kokemusten perusteella. Kokemus vaikuttaa erityisesti maineeseen, joka luottamuksen kohteelle luottajan näkökulmasta kehittyy. Mikäli kohde on ennestään tuntematon, maineelle asetetaan jokin lähtötaso. Luottaja voi pyytää ulkoisilta yhteisöiltä mainelausuntoja helpottamaan ensimmäisen arvion tekoa, mikä edellyttää että kohde tunnetaan näissä yhteisöissä. Lausunto voi joissakin tapauksissa olla myös jonkin hyvämaineisen ryhmän jäsenyyden todistava varmenne. Kohde on saattanut kuitenkin toimia muissa yhteisöissä eri nimellä ja vaikuttaa siksi aivan tuntemattomalta. Alkuarvio voidaan joutua tekemään täysin ilman pohjatietoja, jolloin arvion optimistisuus riippuu tällöin ainoastaan luottajasta. Taipumusta arvioida muita luotettaviksi näiden henkilöllisyydestä riippumatta kutsutaan perusluottamukseksi tai luottavaisuudeksi (engl. *trust propensity*) [Mar94]. Perusluottamus voi sekin muuttua ajan myötä, kokemusten perusteella.

Luottamuspäätöksessä arvioidaan toiminto- ja toimijakohtaisesti, tukevatko järjestelmän nykyiset tiedot toiminnon sallimista. Toimijan hyvä maine vaikuttaa myönteisesti, kun taas toimintoon liittyvät korkeat riskit vaikuttavat kielteisesti. Päätökseen vaikuttavia muita tekijöitä ovat esimerkiksi toiminnon suorittamisen tärkeys. Yksinkertaistettuna esimerkkinä tietokantapankin asiakasyrityksenä toimivan verkkokaupan toimintoihin “osta 50 kg jauhoa luotolla” ja “osta rannekello luotolla” voi molempiin liittyä saman suuruusluokan lähinnä rahallinen riski, jolloin hyvämaineisen kanta-asiakkaan sallitaan tehdä tämänkokoinen tilaus mutta toisen, aiemmin laskuja maksamatta jättäneen asiakkaan huonontunut maine ei aivan riitä riskin vastapainoksi, jolloin kumpaakaan ostosta ei sallita. Toisaalta kaupalla saattaa olla väliaikaisia varastointiongelmia, mistä syystä se pyrkii eroon tilaavievistä jauhosäkeistään. Tällöin “osta 50 kg jauhoa luotolla” -toiminnon tärkeyttä nostetaan suhteessa vähemmän tilakriittiseen rannekello-ostokseen, jolloin huonompimaineinenkin asiakas pääsee ostamaan jauhoja, muttei vieläkään rannekelloa.

Edellämainitun asiakkaan maine oli laskenut kaupan silmissä laskujen jäätyä maksamatta. Luvun 4.1 mukaan maine on kokemukseen perustuva käsitys toimijan pyrkimyksistä ja normeista. Kokemus perustuu tekoihin, joita tulee siten tarkkailla. Tarkkailu voidaan tehdä reaaliaikaisesti tai säännöllisin väliajoin lokitietojen perusteella. Saadut tiedot voivat koskea politiikan vastaisesti toimimista, kuten laskun maksamatta jättämistä, tai vain muuten hyvin epätavallista käytöstä, kuten äkillistä 5 000 rannekellon tilaamista. Jos mitään epätavallista ei tapahdu, tarkkailutiedot tukevat järjestelmän käsitystä toimijan ennalta-arvattavuudesta ja sitä kautta eräänlaisesta luotettavuudesta—käyttäjän toimintaan liittyvä vähentynyt epävarmuus vaikuttaa tulevaisuuden yhteistoiminnan ennustettavuuteen, jolloin luottamusta tarvitaan vähemmän. Tietojen perusteella luottamuksenhallintajärjestelmä korjaa arviotaan käyttäjän maineesta. Lisäksi ulkopuolisten yhteisöjen suorittaman tarkkailun tuloksia voidaan käyttää mainetiedon päivitykseen.

Luvussa 2.3 esiteltiin järjestelmän yleinen työnjako: kun käyttäjä lähettää pyyntöviestin web-sovellukselle, sitä ympäröivä kääre tulkitsee pyynnön osaksi jotakin toimintoa, ja lähettää kyselyn luottamuksenhallintajärjestelmälle ennen pyynnön



Kuva 7: Järjestelmän yleiskuva.

välittämistä itse sovellukselle. Sovelluskääre on toisaalta itsekin sovelluskohtaisesti muuttuva osa luottamuksenhallintajärjestelmää. Järjestelmä jakautu alijärjestelmiin, mikä on esitetty kuvassa 7. Seuraavissa aliluvuissa käsitellään kunkin alijärjestelmän rakennetta ja tehtäviä, ja lopuksi palataan osien yhteistoimintaan esimerkin avulla.

5.2 Sovelluksen yhteys luottamusjärjestelmään

Luottamuksenhallintajärjestelmä on sidoksissa sovellukseen, jota se suojaa ja tarkkailee. Vahvimmin sovellusriippuvainen osa, sovelluskääre, sieppaa sovellukselle lähetetyt viestit. Mikäli viesti on osa jo valtuutettua, käynnissä olevaa toimintoa, kääre välittää sen suoraan sovellukselle. Mikäli viesti aloittaa uuden toiminnon, kääre pidättää sen ja kysyy valtuutusta. Päätöksen tekevä moduuli kokoaa tarvittavat tiedot toimintoon liittyvästä riskistä ja sen tärkeydestä, käyttäjän toimintokohtaisen maineen sekä kontekstiasetusten määräämät muokkaukset näihin kaikkiin. Tällöin se käyttää määrättyä politiikkaa tietojen yhdistämiseen luottamuspäätökseksi, joka on myöntävä tai kieltävä. Mikäli päätös on myöntävä, kääre sallii toiminnon alkaa ja lähettää aloittaneen viestin eteenpäin sovellukselle. Tämän jälkeen samaan toimintoon liittyvistä viesteistä kerätään kokemusta käyttäjästä.

Mikäli kääre ja sovellus kehitetään yhdessä, kuten kirjakauppaesimerkin rajauksessa päätettiin, voidaan luottamuksenhallinta huomioida sovelluksen suunnittelussa. Tällöin kääreen ja sovelluksen kautta voidaan saada tarvittaessa hyvinkin yksityiskohtaista ja semanttisesti rikasta tietoa sovelluksen toiminnasta. Sovelluksen kehittäjällä lienee syytä olla tukenaan kattava dokumentaatio, sillä tältä ei voi lähtökohtaisesti

odottaa asiantuntemusta luottamuksenhallinnasta. Kehittäjällä on kuitenkin muun muassa vaatimusanalyysin perusteella paras käsitys siitä, mitä sovelluksen on tarkoitus tehdä, joten luottamuksen käsitteen lisääminen toiminnan määrittelyyn on oletettavasti varsin suoraviivaista.

Sovelluksen ylläpitäjä on lisäksi vastuussa SOAP-viestien jakamisesta tiettyihin toimintoihin kuuluviksi. Mikäli toimintojen aloitusviesteissä on päällekkäisyyksiä, jako täytyy miettiä uudelleen. Sovelluskääre tarvitsee tiedon jo ensimmäisen viestin perusteella voidakseen tunnistaa tietyn SOAP-viestin vastaanotettuaan, minkä toiminnon suhteen luottamusta tulee arvioida. Ellei toimintoa voida tunnistaa välittömästi, joudutaan myös luottamuspäätöstä lykkäämään keskelle toimintoa. SOAP-viestin tietyksi toiminnoksi tulkitsemiseen tarvittava analyysi on aikakriittistä, joten ulkoisten tiedonlähteiden käyttö viestien tunnistamisessa ei ole toivottavaa. Myöhempi kirjakaupan toiminnonjakoesimerkki kuitenkin osoittaa, ettei sovelluskääre välttämättä voi selvitä tehtävästä täysin ilman ulkoista apua.

TuBEn luottamuksenhallintajärjestelmässä on kullekin toiminnolle mahdollista ilmaista riski- ja tärkeysarvot toiminnon (eli SOAP-viestien) parametrien perusteella. Tällöin esimerkiksi kirjaoston riski voidaan määrittää kirjan hinnan funktiona, eikä toimintoa tarvitse jakaa kahdeksi erilaiseksi ostotoiminnoksi vain riski- ja tärkeysarvojen erillistä määrittämistä varten. Myöhemmässä esimerkin läpikäynnissä toiminto jaetaan kahtia, ja ratkaisua verrataan parametrisointivaihtoehtoon.

5.3 Luottamuspäätöskoneisto

Yhteistyön käynnistyttyä luottamustiedon pohjalta voidaan tehdä päätöksiä siitä, mitä toimintoja tietylle käyttäjälle kulloinkin sallitaan. Päätös tehdään vertaamalla toimintoon liittyvää riskiä sekä käyttäjän mainetta ja toiminnon tärkeyttä toisaalta mahdollisiin yksittäisiin raja-arvoihin, toisaalta tulkinnalla rikastettuna toisiinsa. Mikäli päätös on positiivinen, järjestelmä sallii toiminnon. Mikäli päätös on negatiivinen, järjestelmä voi vielä tarkistaa riskiin vaikuttavat toimenpiteet, joita se voi ottaa käyttöön automaattisesti. Tällainen toimenpide voisi olla esimerkiksi käytössä olevien resurssien rajoittaminen tai tiukennettu valvonta. Mikäli jokin tällainen toimenpide laskisi riskin riittävän alhaiselle tasolle, jotta luottamuspäätös muuttuu positiiviseksi, järjestelmä voi ottaa sen käyttöön ja sallia toimenpiteen pienentyneen riskin perusteella.

Tavoitteena on luoda järjestelmä, joka ottaa luottamukseen vaikuttavat tekijät huomioon päätöksenteossa olematta liian monimutkainen oikeaan käyttöön. Tässä haasteeksi tulevat luottamuksen erilaiset käyttötavat—jotkin luottamuksenhallintajärjestelmää käyttävät toimijat voivat haluta painottaa tiettyjä tekijöitä eri tavoin kuin toisia.

Jotta selkeystavoite pysyisi etusijalla, aloitetaan tarkastelu esitystavaltaan yksinkertaisimmasta osasta: luottamuspäätöksen tulos on binääriarvo, joko toimenpiteeseen suostutaan tai siitä kieltäydytään. Binääriarvoon päädyttäessä joissakin tilanteissa päätös on helpompi kuin toisissa. Välimaastoon jäävien rajatapausten kohdalta

voidaan löytää myös laajennettuja vastausvaihtoehtoja.

Kolmas vastausarvo, “en tiedä”, merkitsisi tässä ympäristössä hitaan ihmisen, ylläpidon, mukaan tuomista päätöksentekoon. Tällöin lopputulos on kysyvän sovelluksen kannalta pitkälti sama kuin jos pyynnöstä kieltäydyttäisiin. Voidaan kuitenkin tarkastella vaihtoehtoa, jossa rajatapauksessa kieltäytymisen yhteydessä annetaan ilmoitus joko ylläpitäjälle, käyttäjälle tai molemmille siitä, millä toimenpiteillä päätös voidaan muuttaa tulevaisuudessa positiiviseksi: “ei, mutta...”. Toimenpiteet saattavat vaatia järjestelmän ulkopuolista toimintaa, kuten esimerkiksi vakuuksien maksamista tai lisätositteiden esittämistä käyttäjän maineen korottamiseksi TrustBuilderin [WSJ00] hengessä. Toisaalta ne voivat myös liittyä järjestelmän tilaan: uudelleen yrittäminen myöhemmin voi auttaa, mikäli toiminto on arvioitu väliaikaisesti riskialttiimmaksi esimerkiksi hyökkäyksen tai ruuhkan takia.

Mikäli tarvittavat toimenpiteet ovat sellaisia, että luottamuksenhallintajärjestelmä voi tehdä ne itsekin, mahdollistuu myös kolmas laajennettu vastausmuoto: “kyllä, jos...”. Luottamuspäätös voidaan saada rajatapauksesta automaattisesti positiiviseksi esimerkiksi asettamalla tämä aktiivisemman tarkkailun alaiseksi tai rajoittamalla hänen käytössään olevia resursseja. Kontrollin kohteena oleva ihmiskäyttäjä todennäköisesti kaipaa tästä ilmoitusta tai kysymystä jatkamishalukkuudesta, koska tarkkailun tason arvailu tai esimerkiksi yhteyden selittämätön hidastuminen aiheuttavat helposti negatiivisia tunteita, joka voi johtaa asiakkaan siirtymiseen muualle. Itsenäisesti toimiva ohjelma sen sijaan ei järin välitä palvelun ei-toiminnallisten ominaisuuksien muutoksesta ellei sitä erikseen käsketä niistä välittämään.

Luottamuspäätökseen liittyvä valinta perustuu kahden tekijäluokan välisen tasapainon etsimiseen. Tasapainon löytämiseksi arvot tarvitsevat jonkinlaista semantiikkaa, joka mahdollistaa niiden välisen vertailun. Mikäli päätökset tehdään vain sen perusteella, ylittääkö kukin tekijä jonkin raja-arvon, menetetään osa hyödystä, joka syntyy vain kun kaikki nämä tekijät ovat saatavilla yhtäaikaisesti.

Päätöksessä vaikuttavat toisaalta luottamushalukkuutta vähentävät tekijät, erityisesti toimintoon liittyvä riski, toisaalta halukkuutta lisäävät tekijät, erityisesti toimijan maine ja toiminnon tärkeys, joka liittyy myös odotettuun tuottoon palveluntarjoajan kannalta. Kaikki nämä tekijät riippuvat kolmesta yhteisestä parametrista, luottamuksen lähteestä ja kohteesta sekä toiminnosta. Koska erityisesti maine muuttuu ajan myötä, on neljäs määräävä parametri ajan hetki jona päätös tehdään. Aika kuitenkin vaikuttaa vain järjestelmän tilan kautta, mutta tästä seuraa, ettei luottamuspäätösten väli- ja lopullisia tuloksia voi vaaratta tallettaa pitkiksi ajoiksi välimuisteihin, vaikka tämä voisi nopeuttaa laskentaa. Historiatietojen ylläpito mahdollistaisi ajan liittämisen parametriksi, mutta tämä tilan- ja tehonsäästön välinen vaihto ei liene yleisessä tapauksessa tarpeen; valitaan siis yksinkertaisin ratkaisu. Lisäksi luottamuksen lähde ilmenee lähinnä vain siinä, että yhteisössä kukin yhteisön jäsen tekee paikallisesti omat luottamuspäätöksensä suojatakseen oman järjestelmänsä.

Luottamuksen kohde määrää ensisijaisesti mainearvoja, mutta se voi ajoittain myös vaikuttaa toissijaisesti riski- ja tärkeysarvoihin. Kohde vaikuttaa näihin arvoihin

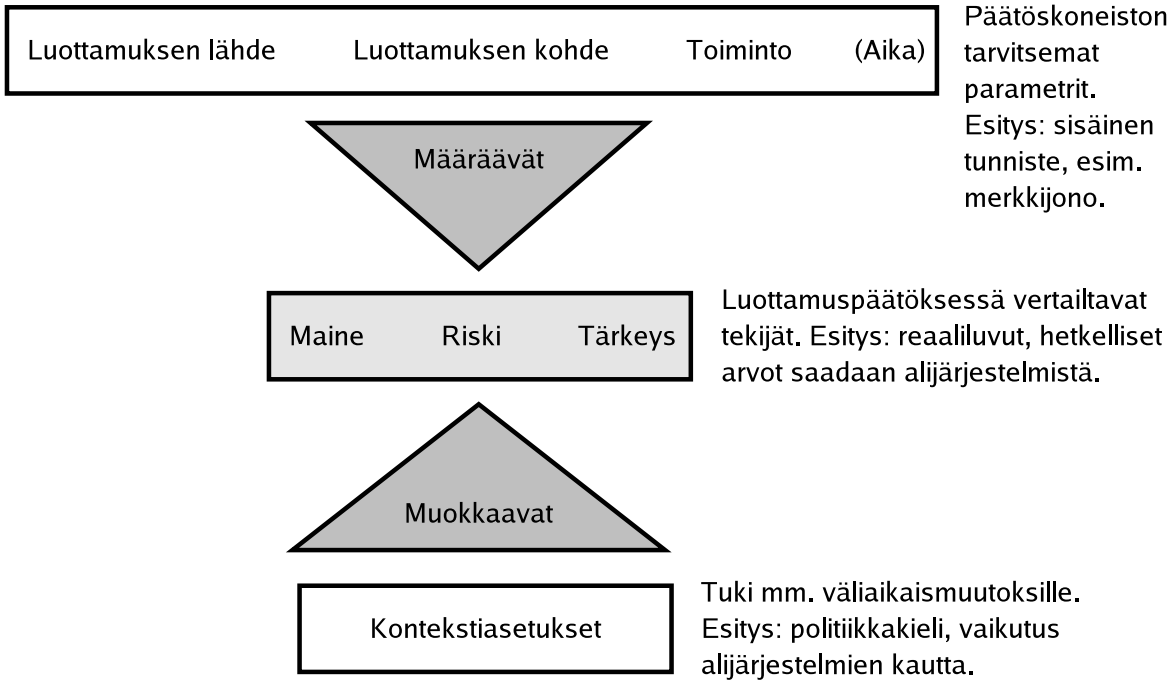
kontekstiasetusten kautta; esimerkiksi vakuuden maksaminen pienentää riskiä, jolloin kontekstiasetus voi määrittää vakuuden maksajien ryhmälle joidenkin toimintojen riskiarvot tavallista pienemmiksi. Kontekstiasetuksia käsitellään yksityiskohtaisesti myöhemmin. Kohteen määrittämiseksi tarvitaan mekanismi käyttäjien tunnistamiseksi.

Edellä todettiin, ettei järjestelmän tulisi käytännön syistä olla riippuvainen pysyvistä identiteeteistä. Luottamusta ei kohdennetakaan ensisijaisesti henkilöön vaan tämän edustukseen, joka voidaan tunnistaa. Luottamuksen kohdetta edustaa siten sisäänkirjautuneen käyttäjän tunnus. Tunnuksen takana voi olla aktiivisen ihmiskäyttäjän sijaan myös itsenäisesti toimiva sovellus, joka pyytää pääsyä palveluun esimerkiksi säännöllisesti tai oman käyttäjäkuntansa edustajan pyynnön seurauksena. Tällöin sovellus on välillisesti luottamuksen kohteena ihmiskäyttäjän tapaan.

Toiminto vaikuttaa kaikkiin luottamuspäätöksessä tasapainoteltaviin tekijöihin. Luottamuksen kohteen perusteella määräytyvä maine tarkennetaan toimintokohtaiseksi. Lisäksi toimintojen riski ja tärkeys riippuvat selvästi tästä määräävästä tekijästä. Toiminto voi vastata yhtä WSDL-kuvauksessa esitettyä kutsua, mutta yleisesti lienee järkevintä yhdistää yhteenkuuluvat kutsut yhdeksi toiminnoksi tai toimintojen luokaksi. Esimerkiksi tuotteen tilaaminen, johon liittyy saatavuuskysely, maksutavan valinta ja yhteystietojen lähettäminen, voidaan jakaa kolmeksi asiakkaan toiminnoksi (saatavuus, maksutapa, yhteystiedot) tai yhdeksi kokonaisuudeksi (tilaus). Käytännössä kukin osa voi vielä vaatia useita WSDL-viestejä, joten tiettyyn pisteeseen asti abstraktiotason nostaminen selkeyttää järjestelmää sekä vähentää riski- ja tärkeysarvojen asettamisen työläyttä pienentämällä toimintojen määrää. Yhdistämistä puolustaa myös se, että mikään kolmesta edellä mainitusta toiminnosta ei yksin ole erityisen riskialtis tai luottamuspäätöstä vaativa, mutta kun ne yhdistetään, niiden seurauksena lähetettävä tuote voi joutua hukkaan, mihin liittyy taloudellinen riski.

Toisaalta toiminnon parametrit voivat myös vaikuttaa riskiin: esimerkiksi ostoksen maksutavan valinnalla voi olla merkitystä. Mikäli käyttäjä arvioidaan luotettavaksi, hänelle voidaan tarjota mahdollisuus maksaa laskulla, kun taas muille tarjolla on kenties vain postiennakko tai nouto myymälästä. Laskua ei kenties maksetakaan, jolloin tuote on menetetty ja rahat saadaan korkeintaan karhuamalla, mutta postiennakko ja nouto pakottavat asiakkaan maksamaan ennen tuotteen luovutusta. Yleisemmin myös tilattava tuote, tässä saatavuuskyselyn parametri, vaikuttavat riskin suuruuteen. Jokaiselle tuotteelle tuskin on järkevää määritellä omaa riskiarvoaan, mutta rahallisen riskin suhteen esimerkiksi tuotteiden jaottelu muutamaan eri hintaluokkaan voi helpottaa riskin asettamista. Kuten edellä mainittiin, TuBEssa toiminnon parametrit otetaan huomioon ilmaisemalla riski ja tärkeys parametrien funktioina pohja-arvojen sijaan, kun taas esimerkissä tämä toteutetaan toimintoja jakamalla parametrien luonteen takia.

Luottamuksenhallintajärjestelmän kannalta “toiminto” voi koostua siis lopulta joukosta alitoimintoja, ja riippua joistakin näiden alitoimintojen parametreistä. Tämä näyttää johtavan siihen, että luottamuspäätöksen lopullinen tulos voisi vaihtua kes-



Kuva 8: Luottamuspäätökseen vaikuttavien tekijöiden roolit ja niiden väliset suhteet.

ken toiminnon: alussa tiedetään vain tuotteen hinta, jolloin tilaus näyttää hyväksyttävältä, mutta kenties kun maksutavaksi valitaankin lasku, riski nousee yllättäen liian korkeaksi. Käytännössä kokoelmatoiminnon aikana maine- ja muiden tietojen voidaan katsoa pysyvän samoina, jolloin tulos voidaan laskea eri vaihtoehdoille, jolloin luottamusjärjestelmää käyttävä sovellus voi tarjota käyttäjälle vain sallitut maksuvaihtoehdot saatujen tulosten perusteella. Jos ennustettavien vaihtoehtojen määrä kasvaa suureksi, lienee toiminnon uudelleenjärjestely tarpeen. Tällöin alkuperäinen toiminto tulee joko jakaa kahdeksi toiminnoksi tai, mikäli tämä onnistuu luontevasti, selvittää vaihtoehtoja eniten rajaavan valinnan tulos jo ennen ensimmäisen luottamuspäätöksen kysymistä.

Luottamuspäätökseen vaikuttavien tekijöiden väliset suhteet on esitetty kuvassa 8. Kun luottamuspäätöstä varten on selvitetty luottamuksen lähde, kohde ja toiminto, voidaan niiden avulla selvittää maine, toiminnon riski ja sen tärkeys.

Koska lopputuloksena on valinta myöntävän ja kieltävän vastauksen välillä, tuntuu luontevalta perustaa päätös jonkinlaiseen vertailuun. Tämä vertailu, kuten edellä todettiin, vaatii jonkinlaista semantiikan lisäämistä arvoihin. On siis voitava ilmaista, että jos toimintoon liittyy suuri riski, tekijän on vastaavasti oltava hyvämaineinen, mutta jos toiminto on hyvin tärkeä, maineen merkitys vähenee. Näiden vertailujen suhde voi olla lineaarinen ja vaatia jonkin kertoimen käyttöä, mutta teorian tasolla mikään ei estä myöskään polynomisia tai eksponentiaalisia vertailusuhteita: jos riski nousee yhden pisteen, kaivattu mainetaso kaksinkertaistuu.

Mikäli oletetaan, että luottamuspäätöksen kolme vaikuttavaa tekijää ovat kaikil-

ta arvoiltaan vertailtavissa keskenään, voidaan kunkin tekijän eri arvoille määrittää muiden tekijöiden avulla järjestys. Tästä seuraa, että emme menetä tietoa tekemällä järjestelmän selkeyteen liittyvän valinnan esittää kaikkien arvot positiivisina reaalilukuina[3]. Tällöin esimerkiksi huono maine on lähellä nollaa, kun taas hyvä maine on korkeampi. Vastaavasti vakavaa riskiä vastaa suurempi luku kuin pientä riskiä, ja suurta tärkeyttä edustetaan samoin suurilla luvuilla.

Reaalilukujen sijaan olisi voitu rajoittua myös kokonaislukuihin, mutta reaalilukuja käytettäessä järjestelmään on helpompi lisätä uusia arvoja vanhojen väliin tarvittaessa. Toisaalta positiivisiin lukuihin rajoittuminen tekee niiden tulkinnasta hieman suoraviivaisempaa: maine ja tärkeys vaikuttavat päätökseen aina "positiivisesti", joskin joskus hyvin vähän mikäli maine on huono. Vertailussa tutkitaan siis, riittävätkö maine ja tärkeys kattamaan toimintoon liittyvän riskin. Vertailun tarkka luonne määritellään luvussa 5.4, mutta suoraviivaisena esimerkkinä luottamuspäätöslausekkeen "maine + tärkeys $>$ riski" tuloksen tulisi helpon ymmärrettävyyden nimissä olla positiivinen, jos riski ja tärkeys ovat hyvin pienet mutta tärkeys hieman suurempi. Esimerkiksi tuotekuvaston selaaminen on lähes riskitöntä, ja saattaa edesauttaa ainakin pieniä ostoja myöhemmin. Toisaalta jos maineen sallitaan olla negatiivinen arvo, lauseketta täytyy muokata monimutkaisemmaksi, jotta tämä haluttu tulos ei esty.

Nämä kolme pääarvoa yhdistetään luottamuspäätökseksi. Arvojen ylläpitoon voi kuitenkin liittyä muita arvoja; esimerkiksi mainetta päivitettäessä lienee tarpeen tarkkailla toimintokohtaisen maineen lisäksi näistä yleistettyä kohteen perusmainetta, jota voidaan käyttää oletuksena kohteen pyytäessä toimintoja, joiden suhteen hänen toiminnastaan ei vielä ole kokemuksia. Nämä apuarvot voidaan kuitenkin piilottaa maineen-, riskin- ja tärkeydenhallinnan alijärjestelmiin, eikä luottamuspäätöskoneiston tarvitse niitä nähdä. Eri alijärjestelmiä käsitellään tarkemmin tuonnempana. Mikäli apuarvoja kaivataan osaksi luottamuspäätöstä, niiden vaikutusta voidaan lisätä alla lyhyesti kuvatun kontekstimekanismin kautta.

Luottamuspäätökseen voidaan myös vaikuttaa väliaikaisesti yhden tai useamman tekijän kautta. Kontekstiasetukset edustavat hetkellisiä muutoksia pääarvoon. Muutokset voivat liittyä järjestelmän, palvelua tarjoavan organisaation tai sitä ympäröivän yhteisön tilaan. Kontekstiasetus voi olla voimassa kerrallaan viikkoja tai kuukausia, mutta sen luonteeseen kuuluu väliaikaisuus; pysyvät muutokset tehdään suoraan pääarvoihin. Olkoon kontekstiasetus yksi tällainen, johonkin tiettyyn arvoon vaikuttava tilanne.

Mikäli asetukseen liittyvä tapahtuma vaikuttaa useisiin arvoihin, näille on tehtävä eri kontekstiasetukset, jotka otetaan käyttöön ja poistetaan käytöstä yhtäaikaa. Mikäli vaikutus yhteen arvoon sen sijaan riippuu toisesta arvosta, on tulkintamme mukaan tällöin kyse pikemminkin luottamuspolitiikan muutoksesta, jolloin korjaus ei enää kuulu kontekstimekanismin piiriin ja sen käyttöönotto johtaa luottamuspäätöksen muuttamiseen, mitä on kuvailtu seuraavassa aliluvussa. Kontekstimekanismin

[3] SECUREn luottamuksen hilamalli on yksi esimerkki tilanteesta, jossa kaikkien arvojen vertailtavuuden sijaan vaaditaan vain osittaista vertailtavuutta [CGS+03].

käsittelyä jatketaan myös seuraavassa.

5.4 Luottamuspäätöksen politiikka

Ihmiset käyttävät vaihtelevia metodeja erilaisten tietojen perustella tehdessään luottamuspäätöksiä, ja tulokset voivat olla hyvinkin yllättäviä [FSD+02, JSTT04]. Koska luottamuksenhallintajärjestelmä palvelee asiakasjärjestelmiä, joiden käyttäjinä ja valvojina on lopulta ihmisiä, päätöksenteon tulee olla säädettävissä näiden erilaisten tarpeiden mukaiseksi.

Luvussa 5.3 todettiin, että luottamuspäätös tehdään lopulta kolmen, mahdollisesti kontekstiasetuksin muokatun tekijän pohjalta: toiminnon tärkeyden, siihen liittyvän riskin sekä luottamuksen kohteen maineen perusteella. Näiden arvojen yhdistämistä luottamuspäätökseksi määrää politiikka, joka ilmaistaan luottamusjärjestelmälle politiikkakielellä. Mikäli politiikka pysyy samana, tekijöiden yhtenevät arvot johtavat samaan päätökseen deterministisesti. Luottamuspäätös on luonteeltaan funktio, mutta ei ole määrätty, etteikö päätös voisi olla aina negatiivinen tai aina positiivinen, eikä tuloksesta voida yleisessä tapauksessa päätellä sitä määrääviä arvoja; luottamuspäätöksen funktio $\mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \{0, 1\}$ ei siis aina ole surjektio eikä koskaan injektio[4] .

Arvojen asettelu vertailulausekkeessa, kuten edellä esimerkissä “maine + tärkeys > riski”, ehdottaa hyvin suoraviivaista muotoa politiikkakielelle: mikäli lauseke on tosi, luottamuspäätöskin on positiivinen, muussa tapauksessa negatiivinen. Tämänmuotoiseen lausekkeeseen ei kuitenkaan vielä sisälly tietoa siitä, millainen olisi rajatapaus josta ylläpitäjä haluaisi tiedonannon. Vaihtoehtona olisi tehdä kielestä ohjelmointikielen kaltainen, kuten Ponder-politiikkakieli [DDLS01]. Tällöin esimerkki, laajennettuna rajatapauksen kuvailulla, voisi näyttää seuraavanlaiselta:

```
// Tässä suoritus päättyy joko komentoon 'hyväksy' tai
// 'hylkää', sen sijaan 'ilmoita' jättää ilmoituksen ja jatkaa
// suoritusta.
if(maine + tärkeys > riski)
   hyväksy;
// Jos ero on 'hyvin pieni', mainitse siitä ylläpitäjälle.
else if(|maine + tärkeys - riski| < 0,5)
   ilmoita;
hylkää;
```

Ohjelmointikielimäisyydessä on etuna ohjelmoijan kannalta intuitiivinen laajennettavuus, sillä politiikkakieleen voidaan tällöin lisätä ohjelmointikielen rakenteina täysin uudenlaisia lauseketukia. Toisaalta tässä yhteydessä ei ole tarkoituksenmukaista

[4] Funktiomäärittelystä on jätetty selkeyden vuoksi pois kaikki “esikäsittelyyn” liittyvä, kuten kontekstiasetukset. Mikäli asetuksia vaihdetaan päätöksen seurauksena sen tuloksen muuttamiseksi, funktion arvo määritetään muuttuneilla arvoilla uudelleen.

rakentaa kovin monimutkaisia lausekkeita, joten vähempikin riittää; tällöin vältytään myös edelliseen esimerkkiin liittyvältä suorituksen päättymisen monimutkaisuudelta. Laajennetaan siis matemaattis-loogista lauseketta siten, että ilmoitusehdot voidaan liittää mukaan. Politiikkakielen ilmaisu koostuu siis kahdesta lausekkeesta, joista toisen totuusarvo määrää, sallitaanko toiminto, ja toisen totuusarvo ilmaisee raportointitarpeen. Lausekkeet tulee erottaa toisistaan jollakin tavalla; seuraavassa esimerkissä tähän käytetään varattua sanaa "ilmoita". Ensimmäinen lauseke määrää toiminnon sallimisen, toinen raportointitarpeen.

```
maine + tärkeys > riski ilmoita |maine + tärkeys - riski| < 0,5
```

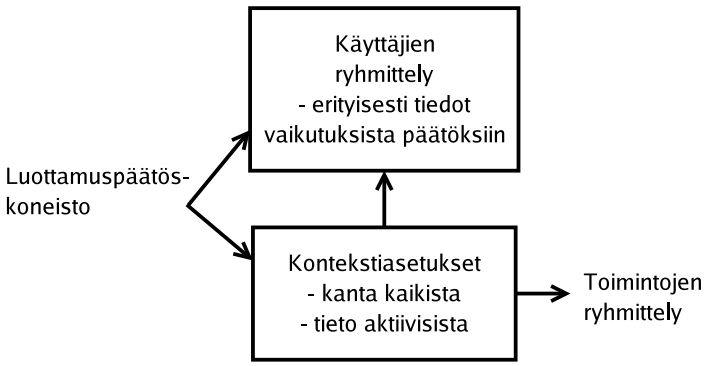
Esimerkin suoraviivainen summa ja vertailu on osin ristiriidassa termiensä aiemmin oletetun rajoittamattomien arvojen kanssa. Maineen, tärkeyden ja riskin arvot voivat tällöin olla vertailukelvottomia: maine voi olla luku väliltä [1,50], tärkeys väliltä [0,1] ja riski väliltä [100,1000]. Ellei vertailusemantiikan löytämisen edellyttämiä rajoituksia toteuteta toisessa osassa järjestelmää, kuten esimerkiksi käyttöliittymässä, ilmaisuun tulee sisällyttää normitus tietylle [minimi, maksimi] -välille. Vapaasti asetettavien tekijöiden arvojen normitus vaatii, että järjestelmä tuntee suurimman ja pienimmän käytössä olevan arvon tietylle tekijälle, ja tulos muuttuu mikäli jokin arvo ylittää tai alittaa nämä rajat. Käytännössä lienee siis järkevintä asettaa järjestelmäkohtaiset rajat näille arvoille toisaalla järjestelmässä, mikäli niitä halutaan verrata keskenään. Tämä helpottanee myös tietyn arvon merkityksen arviointia ihmiskäyttäjälle—esimerkiksi onko "5" korkea vai matala riski.

Itse lausekkeen ilmaisua varten käytettävissä ovat yksinkertaiset matematiikan välineet, kuten reaalilukuvakiot, kerto- ja jakolasku, yhteen- ja vähennyslasku; mahdollisesti myös potenssiinkorotus sekä neliöjuuret ja logaritmi, mutta näiden laskenta voi jo pahimmillaan haitata luottamuspäätöksen nopeutta. Sen sijaan maksimi tai minimi kahdesta luvusta sekä pyöristys ylöspäin, alaspäin tai lähimpään kokonaislukuun voivat olla hyödyllisiä operaatioita. Vertailurelaatioista ovat käytettävissä $=$, $>$, $<$, $\leq$ (ohjelmointikielissä usein "<=") ja $\geq$ (">="). Tarvitaan myös ryhmittelyä, joka voidaan merkitä sulkein, "(" ja ")". Useiden ehtojen sallimiseksi voidaan käyttää loogisten lausekkeiden tapaan yhdistäviä sanoja tai merkkejä "ja" (&&, $\bigwedge$) sekä "tai" (||, $\bigvee$).

Muokattavalle politiikalle on tarvetta jo ennen luottamuspäätöstä. Luvussa 5.3 mainittiin kontekstiasetusten vaikuttavan maine-, tärkeys- ja riskiarvoihin. Tässä yhteydessä mainittiin myös arvojen määräytyminen osa-arvoistaan, nimeämättä jälkimmäisiä yksityiskohtaisemmin. Kontekstiasetusten päätarkoitus on toimia väliaikaisina säätöinä näihin kolmeen arvoon, jolloin esimerkiksi voidaan lisätä tiettyjen toimintojen tärkeyttä väliaikaisesti.

5.5 Kontekstin hallinta

Kontekstinhallintajärjestelmän rakenne on esitetty kuvassa 9. Tieto aktiivisista kontekstiasetuksista ja niiden vaikutuksista välitetään luottamuspäätöskoneistolle, joka muokkaa toimintojen tärkeys- ja riskiarvoja sekä käyttäjien mainetta niiden vaiku-



Kuva 9: Kontekstinhallintamoduulin rakenne.

tuksen mukaan. Muokkaus voidaan myös sijoittaa tehtäväksi itse kontekstinhallintajärjestelmässä, mutta tällöin olisi järkevintä välittää luottamuspäätöskoneistolle jo valmiiksi muokatut riski-, tärkeys- ja mainearvot.

Kuvan 9 kontekstinhallintamoduuli sisältää myös kontekstiasetusten kannalta tarpeellista tietoa käyttäjien kuulumisesta erilaisiin ryhmiin, kuten "tarkkaan valvottavat" tai "käyttäjät, joille tarjotaan maksumahdollisuudeksi myös laskua". Ryhmittely vaikuttaa joidenkin kontekstiasetusten ehtoihin. Erityisesti ryhmittelytieto on tarpeen päätettäessä, asetetaanko tietty käyttäjä johonkin ryhmään, jotta luottamuspäätös saadaan positiiviseksi; tällöin kyseessä on lähinnä riskiä vähentävä käyttäjän asettaminen vapauksiltaan rajoitetumpaan tai tarkemmin tarkkailtavien käyttäjien ryhmään. Tämä lisätoiminto monimutkaistaa kontekstinhallintamoduulia, jonka tulee voida vastata luottamuspäätöskoneiston kyselyyn myös siitä, millaisia kontekstiasetuksia erilaisine vaikutuksineen tulee ottaa huomioon, mikäli käyttäjää lisätään näihin vapaasti muokattaviin ryhmiin.

Kontekstiasetukset esitetään luottamuspäätöksen funktiovalinnan tapaan politiikkakielellä. Kontekstiasetuksen politiikkalauseke palauttaa totuusarvon sijaan reaaliluvun. Luottamuspäätöksen politiikassa käytetyt merkintätyövälineet eivät siten välttämättä riitä, mikäli arvon muutosta säätelee jokin monimutkainen ehto. Niiden lisäksi tarvitaan if-else-rakennetta vastaava ehdollinen haarautuminen, jolla voidaan säädellä palautettavia arvoja kun tietyt ehdot täyttyvät. Ehdot voivat liittyä joko määrääviin tekijöihin, kuten toiminnon tunnisteeseen tai luottamuksen kohteeseen, itse muokattavan tekijän, kuten maineen, nykyarvoon tai edellä mainittuihin osa-arvoihin. Näille kaikille tarvitaan siten varatut sanat, jotta niihin voidaan viitata. Merkintätapa voi edelleen olla ohjelmointikielen kaltainen tai muistuttaa matemaattis-loogista lauseketta. Jälkimmäisessä vaihtoehdossa if-else-rakenne täytyy muuten tehdä näkyväksi, esimerkiksi muodossa "**jos** ehto **niin** arvo ehdon täyttyessä **muuten** arvo mikäli ehto ei täyty".

Koska samaan arvoon vaikuttavia kontekstiasetuksia voi olla useita, järjestelmän tulee tuntea niiden välinen prioriteetti, suoritusjärjestys. Erillinen tietorakenne selkeyttää järjestämistä, mutta vaatii toisaalta jokaiselle asetukselle tunnisteen jonka avulla asetukseen viittaaminen onnistuu. Tunnisteet lienevät joka tapauksessa tar-

peen asetusten käyttöönoton ja poiston tukena.

Mikäli tietokantapankin asiakasyrityksen, esimerkiksi verkkokirjakaupan, kassa on lähes tyhjä, yritys haluaa kenties suosia ostotoimintoa tavallista enemmän. Olkoon kontekstiasetus muotoa `tunniste; palautettava_lauseke`. Tällöin ostotoiminnon suosimiseksi sen tärkeyttä voidaan nostaa lausekkeella `kassavajaus;` **jos** `toiminto = ''osto''` **niin** `tärkeys * 1,5` **muuten** `tärkeys`. Tällöin ostotoiminnon tärkeyttä nostetaan 1,5-kertaiseksi, kun taas muiden toimintojen tärkeys pysyy samana.

Toisaalta hyvin erihintaisten tuotteiden myynnissä varsinainen rahallinen riski voi olla varsin erilainen. Yritys voi täten haluta jakaa ostotoiminnon useaksi eri toiminnoksi tuotteen tai tuotteiden hinnan mukaan, kenties myös maksutavan perusteella. Tällöin esimerkiksi toiminnon "pieni osto (1-5 EUR)" riski voi olla vain kymmenesosa toiminnon "suuri osto (15-50 EUR)" riskistä. Mikäli edellä kuvattu kassavajauksen kontekstiasetus haluttaisiin ilmaista kaikille eri ostoluokille yhtäaikaa, lausekkeen ehto-osassa tehtyjen vertailujen määrä voi kasvaa varsin suureksi. Hallinnoinnin helpottamiseksi otetaan käyttöön toimintoluokan käsite: yksi toimintoluokka sisältää useita toimintoja ja mahdollisesti muita toimintoluokkia.

Toimintoluokkaa voi käyttää paitsi kontekstiasetuksen ehdoissa yksittäisten toimintojen tunnisteiden sijaan, myös alkuperäisten riski- ja toiminnon tärkeysarvojen asettamisessa silloin kun ne ovat yhteneviä kaikille luokan toiminnoille. Kenties olisi hyödyksi myös mahdollistaa mainemuutoksen yleistys joissakin tapauksissa muihinkin saman luokan toimintoihin, esimerkiksi pienen oston epäonnistuminen maksun jäädessä suorittamatta voisi vaikuttaa myös suurta ostoa koskevaan maineeseen; toisaalta maineen lisäys pienen oston suhteen ei välttämättä kohottaisi suurta ostoa koskevaa mainetta tietyn rajan yli, sillä tällöin yhdellä suurella ostolla voitaisiin muuttaa pienostoksilla kerätty maine kerralla rahaksi jättämällä viimeinen lasku maksamatta.

Toimintoluokan "ostot" avulla edellä kuvattu kontekstiasetus olisi `kassavajaus;` **jos** `toimintoluokka = ''ostot''` **niin** `tärkeys * 1,5` **muuten** `tärkeys`, mikä ei luettavuudeltaan juuri eroa alkuperäisestä yhden ostotoiminnon esimerkistä, vaikka asetuksia voidaan nyt muokata tarkemmin. Käytännössä toiminto voinee kuulua useampaan kuin yhteen luokkaan kerrallaan, jolloin"="-operaattori ylläolevassa esimerkissä on riittämätön. Merkitköön se kuitenkin tässä yhteydessä kaikkien toiminnon luokkien vertaamista yhtäaikaa. Konditionaalilauseke ei välttämättä ole paras mahdollinen tapa esittää kontekstiasetuksia, joista suurin osa koskee vain tiettyä käyttäjä- tai toimintoluokkaa. Se toisaalta mahdollistaa myös ilmaisutapoja joita näissä yksinkertaisissa esimerkeissä ei tule ilmi.

Laskulla tuotteen toimituksen jälkeen maksamisen salliminen sisältää suuremman riskin kuin etukäteismaksun edellyttäminen. Muun muassa kännykkäoperaattorit saattavatkin vaatia etenkin maksuhäiriöisiltä asiakkailta etukäteismaksun, jolla tätä riskiä pienennetään. Maksua ei oletuksena käytetä laskujen maksamiseen, vaan se palautetaan asiakassuhteen päättyessä, ellei ongelmia ilmennyt. Tällainen järjestelmä voidaan ilmaista myös kontekstiasetusten avulla: Takuun maksaneet käyttäjät luokitellaan, kuten edellä toiminnot, käyttäjäryhmään "taatut". Kontekstiasetus voi

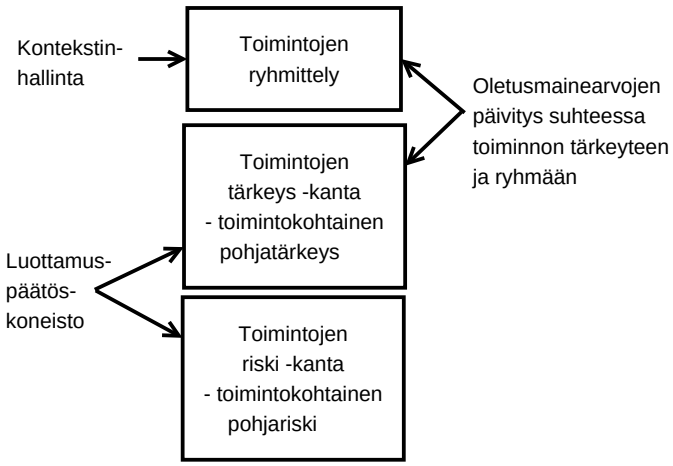
tällöin olla seuraavanlainen: `takuumaksut;` **jos** `käyttäjäluokka = ''taatut''` **niin** `riski - 50` **muuten** `riski`.

Kontekstiasetukset voidaan jakaa kahteen ryhmään sen mukaan, minkä tilaa ne kuvaavat. Organisaation tilaa kuvaava kontekstitieto, kuten esimerkiksi ilmoitus kassavajeesta, päivitetään järjestelmän ulkopuolelta. Järjestelmän tilaan liittyvä kontekstitieto puolestaan saadaan ja asetetaan järjestelmän sisäisesti. Joidenkin kontekstiasetusten jaottelu ei ole täysin suoraviivaista. Esimerkiksi luottamusjärjestelmän ulkopuolinen, koko organisaation verkossa toimiva tunkeutumisenestojärjestelmä (IDS) voi raportoida havaitsemistaan hyökkäyksistä myös luottamuksenhallintajärjestelmälle. Jos tämä asettaa tietyn kontekstiasetuksen raportoinnin mukaisesti, ei ole aivan selvää, oliko asetus järjestelmän sisäinen vai ulkopuolinen.

Luvussa 4.5 ehdotettiin, että joitakin kontekstiasetuksia voisi asettaa myös järjestelmän aloitteesta luottamuspäätökseen vaikuttamiseksi. Tähän sopisivat lähinnä järjestelmän sisäistä kontekstia kuvaavat asetukset, joiden vaikutus tehtävään luottamuspäätökseen on positiivinen. Riskin pienenemisellä on positiivinen vaikutus päätökseen, joten esimerkiksi tarkkailun lisääminen tai käytettävissä olevien resurssien rajaaminen voivat muuttaa vastaavien kontekstiasetusten kautta luottamuspäätöksen tulosta positiivisesti.

Jos esimerkiksi asiakasyrityksen maine ei aivan riitä sellaisenaan raskasta tietokantaoperaatiota vaativan palvelupyynnön toteuttamiseen, toiminto voidaan silti sallia asettamalla automaattisesti kyseinen käyttäjä väliaikaisesti käyttäjäryhmään, jonka jäsenien omistamien prosessien käytössä olevaa suoritustehoa lasketaan. Lisäksi joitakin riskiltään verraten suuria toimintoon liittyviä valintoja voidaan sulkea pois kyseisen toiminnon ajaksi, kuten esimerkiksi jälkikäteen maksamisen mahdollisuus (vaaditaan esimerkiksi postiennakkoa). Tarkkailun tiukentaminen soveltunee parhaiten varsin laajoihin vapauksiin, jotka ovat verrattavissa esimerkiksi komentotulkin käyttöoikeuteen jollakin palvelimella (jolloin suoritettuja komentoja ja niiden seurauksia saatetaan tarkkailla tavallista tiukemmin) tai Wikipedian [Wik05] kaltaisen suuressa ryhmässä muokattavan teoksen kirjoitusoikeus (jolloin mitattavia tietoja muokkauksista, kuten esimerkiksi merkkimäärän muutos, voidaan analysoida vandalismin ennaltaehkäisemiseksi ja käyttäjätunnus tallentaa muokkausten ohella jäljitettävyyden parantamiseksi).

Vaikka kontekstiasetukset tekevät luottamuksenhallintajärjestelmästä joustavamman, voi niiden ylenmääräinen käyttö hidastaa luottamuspäätöksen tekemistä. Aikakriittisissä sovelluksissa päätöstilanteessa, sovelluksen odottaessa, tehtävän laskennan ja toisaalta tiedon haun määrän minimointi on tärkeintä, mutta myöskään arvojen muutosten aikana tapahtuva päivitys ei saisi aiheuttaa liiallista suoritustehon laskemista. Mikäli kontekstiasetuksia on käytössä useita ja ne ovat usein päällekkäisiä, vaikuttaen esimerkiksi samaan toimintoon, toteutuksessa voi olla hedelmällisintä ottaa lähtökohdaksi eräänlainen koontitaulukko, jossa eri kontekstiasetusten vaikutukset on yhdistetty. Tällöin taulukosta kävisi esimerkiksi ilmi, että kaikkien kontekstiasetusten yhteisvaikutuksena tietyn toiminnon muokattu arvo on esimerkiksi 3,5. Luottamuspäätöstä tehtäessä arvot olisivat tällöin jo valmiina. Toisaalta taulukointi



Kuva 10: Toimintojenhallintamoduulin rakenne.

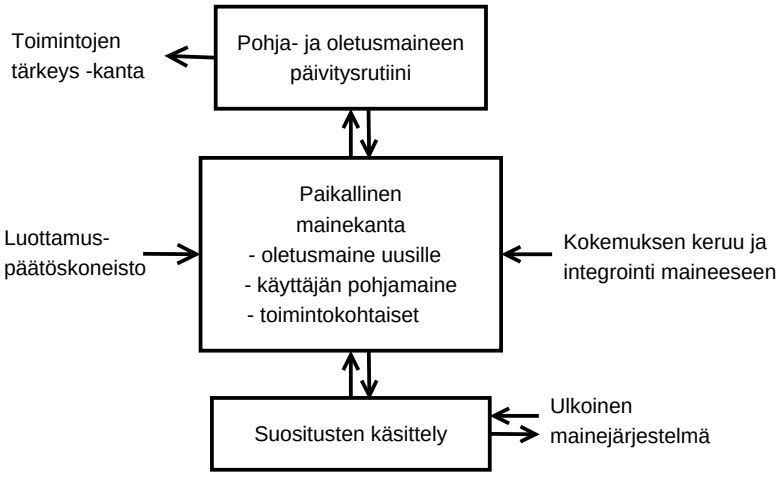
on raskas lähestymistapa, jos kontekstiasetusten muutoksia tehdään usein suhteessa luottamuspäätösten määrään, sillä muutoksen seurauksena suuria osia taulukosta voidaan joutua päivittämään.

TuBEn luottamuksenhallintajärjestelmässä kontekstiasetukset järjestetään ketjuksi, joka käydään läpi jokaisen päätöksen yhteydessä. Ketjun määräämät relevantit muutokset tehdään toimintojen tiedoista ja mainejärjestelmästä saatuihin pohja-arvoihin juuri ennen niiden käyttöä luottamuspäätöksessä. Tämä lähestymistapa edellyttää, että yhtä aikaa aktiivisina olevien kontekstiasetusten määrä ei nouse erityisen suureksi. Yksi mahdollinen lähestymistapa tämän takaamiseksi voisi olla muutamien samaan arvoon vaikuttavien kontekstiasetusten yhdistäminen yhdeksi ketjuun liitettäväksi asetukseksi. Yhdistämisestä on eniten hyötyä, kun yhdistettävien asetusten mahdolliset ehdot ovat samat tai lähes samat.

5.6 Toimintojen hallinta

Edellä todettiin tarve ryhmitellä toiminnot samankaltaisuuden tai yhtenäisten kontekstivaikutusten mukaan toimintoluokiksi. Luokittelu, samoin kuin toimintojen tärkeys- ja riskitiedot, tallennetaan toimintojen hallinta -moduuliin, jonka tarkempi rakenne on esitetty kuvassa 10. Kahta jälkimmäistä käytetään luottamuspäätöksissä sellaisenaan. Toimintojen ryhmittelytietoja tarvitaan lisäksi kontekstinhallintajärjestelmässä, jotta ryhmittelyjen avulla kirjatut kontekstiasetukset voidaan tulkita yksittäisiä toimintoja koskeviksi. Ryhmittely- ja tärkeystiedot ovat käytössä maineen yleistyksessä, jota kuvailtiin edellä.

Tärkeys- ja riskiarvot sekä toimintojen ryhmittely täytetään ensisijaisesti ihmisvoimin. Riskin arvioinnissa tukena voidaan käyttää riskianalyysin työkaluja. Tärkeyden arvioinnissa perusteena voi olla toisaalta suora taloudellinen hyöty, toisaalta muut ulkoiset vaatimukset, kuten varastotilan puute, jolloin varaston tyhjenemistä edistävät toiminnot ovat tärkeitä, tai erilaiset toiminnan jatkuvuuden asettamat vaati-



Kuva 11: Paikallinen mainejärjestelmä -moduuli. “Sisäiset” suositukset oletetaan kuvassa tuotettaviksi ulkoisen mainejärjestelmän kautta.

mukset, jolloin tietty toimitus pitää tehdä sopimusrikon välttämiseksi. Toimintojen ryhmittely ei ole järjestelmän toiminnan kannalta välttämätöntä, mutta onnistunut jako parantaa mainejärjestelmän toimivuutta sekä helpottaa kontekstiasetusten luontia ja ylläpitoa.

5.7 Paikallinen ja ulkoinen mainejärjestelmä

Maine on keskeisin toimijakohtaisen luotettavuuden mittari. Sen ylläpitämiseksi kerätään paikallisesti kokemusta aiemmasta yhteistoiminnasta, ja vastaanotetaan ulkopuolisia suosituksia mainepäivityksille. Maine riippuu tarkimmassa muodossaan sekä toimijasta että toiminnosta, jonka pohjalta kokemusta on saatu. Jotta mainetieto ei olisi liian harvaa, täytyy muutamaan kokemukseen sidotusta maineesta voida johtaa jonkinlaisia oletusarvoja sille, millainen maine kyseisellä käyttäjällä tulisi olla jonkin uuden toiminnon suhteen. Näistä vain luottamksen kohteista riippuvasta pohjamainearvoista voidaan edelleen johtaa arvo luottavaisuudelle, joka on siis pohjamaineen oletusarvo ennalta tuntemattomille käyttäjille.

Mikäli toiminnot ovat enimmäkseen erillisiä, toimintokohtaisen maineen tulisi olla ensisijainen päivityksen kohde, jolloin käyttäjän pohjamainetta käytetään vain uusien toimintojen maineoletusarvojen löytämiseksi. Pohjamaineen päivitys voi täten olla silloin tällöin tapahtuva rutiinitoiminta. Tämä oletus on pohjana maineenhallintajärjestelmän rakenteen tarkemmassa kuvauksessa, joka on esitetty kuvassa 11. Itse mainekanta on lähinnä tietovarasto; erillinen moduuli päivittää pohjamainetta sopivin väliajoin, ja toiminnan tarkkailusta vastaava moduuli muodostaa myös kerätystä tiedosta seuraavat mainepäivitykset. Yhteys ulkoiseen mainejärjestelmään suodattuu erillisen suositustenkäsittelymoduulin kautta.

Käyttäjäkohtaista pohjamainetta käytetään oletuksena, kun käyttäjälle tulee aset-

taa mainearvio suhteessa toimintoon, jota tämä ei ole aiemmin yrittänyt. Pohjamaine vastaa siis ihmismielen taipumukseen yleistää: tarkemman luottamuksen kohteeseen tutustumisen jälkeen luottaja voi alkaa osoittaa luottavansa tähän tietyssä määrin henkilönä, ilman toimintokohtaisia tarkennuksia. Tulkitsen, että tällaisissa tapauksissa on erityisesti kyse luottajan "paikallisesta" mainetiedosta. Luottaja kykenee tällöin yhä erottamaan toimintoja, joiden suhteen kohde katsotaan erityisen luotettavaksi, ja kenties joitakin, joiden suhteen tähän luotetaan vähemmän kuin yleistetty luottamus antaisi ymmärtää. Yleistetyn luottamuksen tason johtaminen ei kuitenkaan ole aivan ongelmatonta: esimerkiksi keskiarvon laskeminen tunnettujen toimintojen pohjalta johtaisi siihen, että harvoihin toimintoihin erikoistunut luottamuksen kohde voisi nauttia huomattavaa, jopa katteetonta luottamusta jonkin aivan toisenlaisen toiminnon suhteen.

Yksi edellä kuvatun ongelman osa koskee toimintojen hyödyllisyyseroa palveluntarjoajan kannalta. Käyttäjän ei tulisi voida koota riittävästi mainetta esimerkiksi auton ostamiseksi luotolla pelkästään tekemällä lukemattomia määriä onnistuneita saatavuuskyselyitä. Toisaalta kirjakauppa voisi katsoa esimerkiksi käyttäjien kirjoittamien tuotearvioiden olevan niin hyödyllisiä myynnille, että se olisi valmis sallimaan niillä kerätyn maineen vaikutuksen ostotoimintoihin asti.

Ongelman ratkaisemisessa voidaan käyttää hyväksi ennalta määrättyjä tärkeysarvoja kullekin toiminnolle: tärkeiden toimintojen tulisi painottua yleisempiä arvoja johdettaessa. Toimintojen tärkeys voidaan suoraviivaisimmin ottaa mukaan oletusarvojen laskentaan käyttämällä sitä painotuksena laskettaessa toimintokohtaisista mainearvoista keskiarvoa, joka siten asetetaan käyttäjäkohtaiseksi maineoletusarvoksi. Tämä kuitenkin antaa toimintojen tärkeydelle kaksoismerkityksen, jolloin on tärkeää varmistaa, etteivät painotukset eroa käytännössä. TuBEn luottamuksenhallintajärjestelmässä toiminnan tärkeyden käyttöön mainepäivityksissä ei ole vielä otettu kantaa. SECURE esittää, että kokemuksesta johdetun mainearvon tulisi heijastella sellaisen käyttäjän mainetta, joka toimii samalla tapaa alinomaa, ja tuloksena saatu arvo johdetaan tämän ja nykymaineen välille [WCE+03]; kenties maineen yleistämisessä voisi käyttää vastaavanlaista lähestymistapaa.

Toinen ongelman osa koskee joidenkin toimintojen heikkoa vertailtavuutta. Esimerkiksi hyvä maine kirjaostajana ei kerro paljoakaan sopivasta mainearviosta arkaluontoisten tietojen käsittelyssä. Toisaalta ihmisellä on taipumus yleistää luottamus rajoitetussa määrin useille elämän aloille. Luvussa 5.4 esitettiin toimintojen ryhmittelyä toimintoluokiksi, muun muassa kontekstiasetusten ehtojen selkeyttämiseksi. Näitä toimintoluokkia voisi käyttää myös pohjana oletusmainearvon löytämiseksi uudelle toiminnolle, joka kuuluu samaan toimintoluokkaan kuin maineeltaan tunnettu toiminto. Koska toimintoluokkia voidaan kuitenkin luoda myös muussa merkityksessä kuin aidosti samankaltaisten toimintojen yhdistämiseksi, vaatinee tämä luokittelun käyttötapa erityisen eksplisiittisen ilmaisun luokkaa määriteltäessä siitä, että toiminnot ovat riittävän samankaltaisia, jotta niistä voidaan päätellä maine muihin luokan toimintoihin. Yhden toimintoluokan sisällä oletusmaineen johtaminen tapahtuisi siten samaan tapaan, toimintojen tärkeyttä painottaen, kuin johdettaessa käyttäjäkohtaista oletusarvoa koko tunnettujen toimintojen joukosta. Tätä

arvoa tulisi käyttää ensisijaisena tiedonlähteenä uusille toimintokohtaisille mainearvoille. Toissijaisesti voidaan käyttää täyden painotetun keskiarvon kautta laskettua käyttäjäkohtaista maineoletusarvoa.

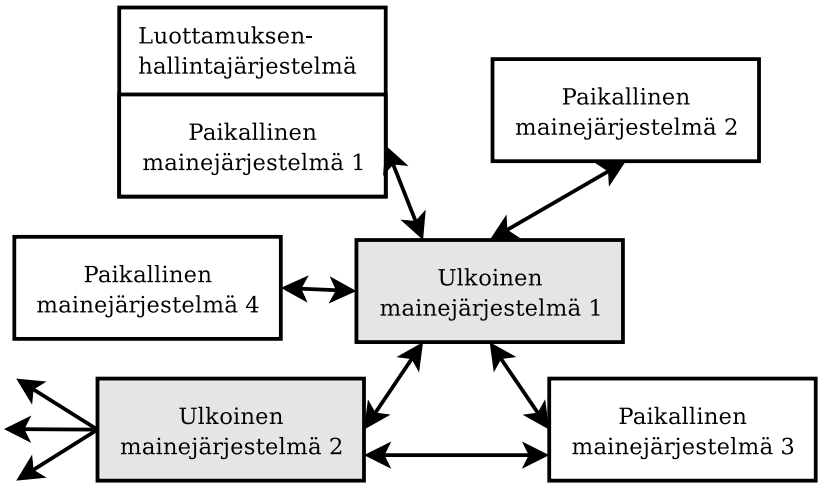
Maineen kehityksen suhteen eri luottajien tarpeet eronnevat toisistaan. Lieneekin järkevää mahdollistaa myös erilaiset lähestymistavat kokemuksen liittämiseksi maineeseen. Esimerkin kirjakaupan käyttämä maineen korotus- ja laskemisalgoritmi määräsi ostotoimintojen erilaisten lopputulosten vaikutuksen maineeseen: laskun maksamatta jättäminen laski mainetta eri kaavan mukaan kuin viestinvaihdon kesken jättäminen aikaisemmassa vaiheessa.

Uuden käyttäjän maineen määrittämiseksi on aiemmissa luvuissa kuvailtu kaksi lähestymistapaa. Mikäli käyttäjä tunnetaan jo jossakin toisessa järjestelmässä, voidaan kyseisen järjestelmän suosituksia käyttää pohjana mainearvon päättämiseksi. Suositukset voivat olla suoraan paikallisen mainejärjestelmän suositushallintamoduulin kanssa yhteensopivia tiedonantoja, jotka voidaan liittää järjestelmään automaattisesti. Tämä vaatii kuitenkin käyttäjän paikallisen identiteetin luotettavaa yhdistämistä suosittajajärjestelmän käyttäjäkantaan: mikäli käyttäjästä ei tiedetä muuta kuin paikallinen, tämän itse valitsema käyttäjätunnus, ei tällainen yhdistäminen ole mahdollista. Suositukset voivat kuitenkin tulla myös mainejärjestelmän ulkopuolelta. Esimerkiksi käyttäjä voi esittää varmenteen tai vastaavan dokumentin, joka todistaa hänen kuulumisensa johonkin hyvämaineiseen yhteisöön, jolloin ylläpitäjä voi lisätä luottamuksenhallintajärjestelmän mainetietoihin yhteisön jäsenille varatun erityisen oletusmaineen.

Esimerkin kirjakaupassa suositusten käsittelyä ei voida täysin automatisoida uusille käyttäjille, koska heistä ei ole riittäviä taustatietoja. Uusi asiakas voi kuitenkin esimerkiksi todistaa olevansa jonkin yhteistyökumppanin tietty asiakas varmenteen avulla, jolloin hänet saadaan sijoitettua kyseisen yhteistyökumppanin mainejärjestelmän nimiavaruuteen, ja mainetietoja voidaan vaihtaa. Myös henkilöllisyyden todistaminen voi toimia maineen lisääjänä, sillä se vähentää jäljityksen mahdollistaessaan halukkuutta väärinkäytöksiin. Tämä tosin voitaisiin myös tulkita riskin vähenemiseksi, juuri jäljityksen ja tilille saamisen mahdollisuuden takia.

Täysin tuntematon käyttäjä, josta ei ole minkäänlaista taustatietoa myöskään muiden mainejärjestelmien kautta, voidaan maineen osalta arvioida jonkin järjestelmän oletustason mukaisesti. Kirjakauppa voisi esimerkiksi haluta oletusmaineen olevan riittävän korkea, jotta käyttäjä voi ostaa kirjoja maksamalla ne välittömästi verkkopankissa, muttei riittävän korkea laskulla maksamiseen.

Oletusarvoa voidaan päivittää myös kokemusten pohjalta, mutta muutos tulee ottaa erikseen huomioon mainearvojen käyttötapaa suunniteltaessa. Oletustason ajantasaisuuden voisi kuitenkin kuvitella olevan siinä määrin epäkriittistä, että sen päivitys olisi luontevinta jättää ihmisten tehtäväksi. Päivitystä automatisoitaessa pohjana voitaisiin käyttää esimerkiksi käyttäjäkohtaisten maineiden keskiarvoa, mutta tällöin ylläpitäjän tulee olla vahvan tietoinen järjestelmän käyttäytymisen monimutkaistumisesta ja sen mahdollisista seurauksista; kaikki käyttäjät ovat joskus uusia käyttäjiä, ja kuten aiemmin on todettu, oletusmainetasolla (“oletusluottamuksella”)



Kuva 12: Esimerkki ulkoisten ja paikallisten mainejärjestelmien yhteistyöstä. Kukin paikallinen mainejärjestelmä on edelleen osa luottamuksenhallintajärjestelmäänsä.

on selkeitä vaikutuksia järjestelmän avoimuuden ja käytettävyyden sekä suojauksen ja väärinkäytettävyyden suhteen.

Edellä esimerkiksi laskulla maksamisen ehdotettiin olevan kirjakaupassa uusien käyttäjien saavuttamattomissa, mutta oletusmaineen suuri nousu voisi tuoda sen näidenkin ulottuville. Tällainen oletusmaineen tason korotus voisi tulla kyseeseen silloin, kun yritys toteaa olevansa riittävän vakavarainen avatakseen palvelunsa myös uusille hyökkäysmahdollisuuksille palvellakseen käyttäjäkuntaansa paremmin. Toisaalta oletusmaineen kontrolloimaton lasku voisi asettaa uudet käyttäjät tilanteeseen, jossa nämä eivät voi ostaa kirjoja lainkaan.

Mainejärjestelmien toiminnan edellytyksenä ovat jatkuvasti levitettävät mainetiedon päivitykset, joko pelkästään paikallisen kokemuksen perusteella tai pohjaten ulkoisista mainejärjestelmistä peräisin oleviin lausuntoihin. Paikallinen mainejärjestelmä toimii lähinnä paikallisen kokemustiedon ja mahdollisten hallinnon tekemien paikallisten lausuntojen perusteella silloin kun esimerkiksi joidenkin luotettujen ryhmien jäsenyys voidaan todistaa parhaiten esimerkiksi jäsenkorttia näyttämällä, mitä järjestelmä ei voi itse havaita. Tämän maineen viestimiseksi muille järjestelmille tarvitaan toinen, ulkoinen mainejärjestelmä, joka käsittelee erilaisilta paikallisilta ja muilta ulkoisilta mainejärjestelmiltä saapuvat lausunnot käyttäjäjärjestelmiensä yhteisen sopimuksen mukaisesti ja välittää mainepäivitykset eteenpäin käyttäjäjärjestelmilleen. Kuvassa 12 kuvataan ulkoisten ja paikallisten mainejärjestelmien yhteistyötä.

Kuten edellä mainittiin, minkä tahansa paikallisten mainejärjestelmien yhdistäminen ulkoisen mainejärjestelmän avulla ei aina ole hyödyllistä. Koska maine riippuu vahvasti myös toiminnosta, jonka suhteen sitä on tarkasteltu, yhdistettävien mainejärjestelmien toimintoryhmien täytyy olla riittävän yhteneviä, jotta niiden mainekäsitykset voisivat vastata toisiaan ainakin teoriassa. Kun yhteensopivuudesta on

varmistuttu, järjestelmien kesken täytyy tehdä sopimus siitä, miten mainetta kussakin tulkitaan, jotta yhteisessä mainejärjestelmässä tietyntasoisen maineen ansaitsee kussakin alajärjestelmässä jokseenkin vastaavanlaisella toiminnalla.

Paikalliset mainejärjestelmät voivat lisäksi antaa erilaisia painotuksia ulkoisten mainejärjestelmien lausunnoille. Kuvan 12 oikean alakulman paikallinen mainejärjestelmä 3 on kytkeytynyt kahteen ulkoiseen mainejärjestelmään, jotka lisäksi saavat tietoa toisiltaan. Se saattaa haluta antaa ulkoisen mainejärjestelmä 1:n lausunnoille enemmän painoa kuin mitä se saisi suodatettuna ulkoisen mainejärjestelmä 2:n läpi, mutta haluaa kuulla myös jälkimmäisen tuoman lisätiedon ilman suodatusta ulkoisen mainejärjestelmä 1:n läpi.

Identiteetti ja sen pysyvyys ovat keskeisiä maineen keräämiseksi. Maineen liittämiseksi identiteettiin on tärkeää kyetä toisaalta erottamaan eri luottamuksen kohteet toisistaan, toisaalta tunnistamaan tietty kohde samaksi kuin aiemmin tavattu. Esitelty luottamuksenhallintajärjestelmä ei oleta identiteetin olevan täysin sidottu tiettyyn yksilöön, vaan luottamus kohdistetaan tunnistettavaan toimijaan, joka voi olla esimerkiksi säännöllisesti palvelua käyttävä, pitkälti itsenäisesti toimiva sovellus. Tavoitteena on kyetä yhdistämään saman käyttäjän eri yhteydenotot toisiinsa vähintään silloin, kun tämä ei sitä itse aktiivisesti pyri estämään. Jotta ulkoisilta mainejärjestelmiltä saatua tietoa pystyttäisiin hyödyntämään käyttäjän maineen pohjana, on lisäksi tiedettävä käyttäjän nimitys tietolähteenä toimivassa järjestyksessä, mikä ei välttämättä ole yksinkertaista.

Tietokantapankin mainejärjestelmässä maineenhallinta keskittyy enimmäkseen paikalliseen maineeseen. Luottamuspäätöksessä käytettävä maine on toimija- ja toimintokohtaista. Kun toiminto etenee, käyttäjän toimintaa tarkkaillaan. Epäilyttävä käytös saattaa laskea mainetta, kun taas positiivinen käytös nostaa sitä. Maineen laskiessa tietyn rajan alle luottamuksen kohde ei välttämättä enää voi käyttää yhtäkään omalta kannaltaan tarpeellista toimintoa järjestelmässä. Mikäli esimerkiksi kirjakauppa lakkaa myymästä kirjoja, kun tietty mainetaso alittuu, asiakkaan saama hyöty palvelusta laskee hyvin vähäiseksi ja hänen on pakko hoitaa kirjaostoksensa jatkossa muualla. Paikallisesta näkökulmasta tämä on erityisen huono tilanne, mutta myös yhteisötasolla maineen palautuminen toimimalla hyvin yhteisön ulkopuolella ei välttämättä riitä. Käyttäjän ei voida olettaa aina olevan tekemisissä jonkin toisen samaan maineyhteisöön kuuluvan palveluntarjoajan kanssa, jolloin tämän suositusten kautta käyttäjän maine voisi palautua paikallisessa järjestelmässä.

Käyttäjät, joiden maine on laskenut alle järjestelmän peruskäyttöön vaadittavan rajan, tarvitsevat siis jonkinlaisen toipumismekanismin huonomaineisuudestaan noustakseen. Tässä tilanteessa luottajana toimiva palveluntarjoaja joutuu jälleen valitsemaan järjestelmänsä avoimuuden ja suojauksen välillä. Katsomalla huonomaineisen käyttäjän olevan parantumaton kunnes toisin todistetaan, palveluntarjoaja menettää lopulta asiakkaansa kilpailijalle, joka kenties suostuu hanakammin tarjoamaan tälle jonkinlaista palvelua. Lisäksi koska kirjakaupan kaltaiselle palveluntarjoajalle on eduksi pitää asiakkaaksi liittymisen vaiva mahdollisimman pienenä, huono maine ei toimi yhtä lailla rankaisuna asiakkaan voidessa luoda uuden käyttäjän. Internet-

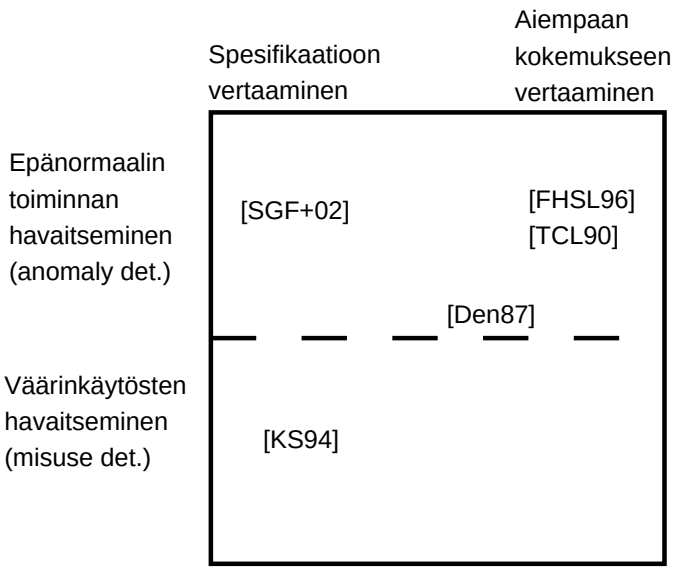
kirjakauppiaat tuskin yhdistyvät saman maineyhteisön alle yhden huonomaineisen yksityishenkilön hillitsemiseksi. Toisaalta sadoittain lentoja päivässä varaava matkatoimisto, joka ei maksa laskujaan, voi huomata huonon maineen sulkevan sen varsin monien lentoyhtiöiden palvelujen ulkopuolelle.

5.8 Kokemuksen keruu

Käyttäjän toimintaa on perinteisesti tarkkailtu lähinnä hyökkäysten havainnoinnin toivossa, mistä juontaa myös tarkkailuvälineiden yhteisnimitys "tunkeutumisenestojärjestelmä" (engl. *Intrusion Detection System*, IDS). Vaikka parhaassa tapauksessa kaikki hyökkäykset voidaankin ennaltaehkäistä, käyttäjän tai ohjelman toiminnan tarkastelu voidaan usein tehdä vasta jälkikäteen, tietyin väliajoin, sillä tehtävä on laskennallisesti raskas. Tällöin tyydytään kirjaamaan arvio tapahtuneesta, jolloin järjestelmän ylläpitäjä saa luettavakseen ikään kuin tiivistettyjä lokitietoja. Tunkeutumisenestojärjestelmä voisi myös selkeän väärinkäytöksen ilmetessä päättää itsenäisesti sulkea käyttäjän suojatun järjestelmän ulkopuolelle, mutta laajat harmaat alueet on jouduttu jättämään ihmishallinnon ratkaistavaksi. Jos päätettävänä on vain se, saako käyttäjä tämän tekonsa jälkeen toimia lainkaan, pikkurikkeet jätetään pitkälti huomiotta, jottei järjestelmästä tulisi liian ankara.

Hienojakoisempi luottamuksenhallinta mahdollistaa tähän "jokseenkin epäilyttävään" käytökseen reagoinnin automaattisesti. Se voi laskea luottamusta käyttäjään vain vähän, jolloin rikkeen mahdollisesti kehittyessä tavaksi käyttäjä saattaa päätyä tarkemmin tarkkailtavaksi tai toisaalta, jos hänen muu käytöksensä on moitteetteetonta, vaikuttaa tuskin lainkaan. Täysin perusteettomat hälytykset ovat kuitenkin yhä ongelmana samassa määrin kuin tavallisissa tunkeutumisenestojärjestelmissä, joten tarkkailusääntöjen hienosäätö on tarpeen.

Tunkeutumisenestojärjestelmät voidaan jakaa kolmeen eri luokkaan sen mukaan, minkä periaatteen mukaan ne erottavat hyvän käytöksen huonosta. Väärinkäytösmalleja tunnistava järjestelmä (engl. *misuse intrusion detection*) mallintaa tunnetut hyökkäykset ja vertaa käyttäjien toimintaa niihin [KS94]. Tällöin arvio hyökkäyksen etenemisestä voidaan delegoida ihmisasiantuntijalle, ja väärien hälytysten määrä on suhteessa varsin pieni. Toisaalta hyökkäys täytyy tuntea etukäteen, jotta sitä voidaan havaita lainkaan, jolloin järjestelmä voi jäädä jälkeen hyökkääjistä. Totutusta käytöksestä poikkeamisen tunnistava järjestelmä (engl. *anomaly intrusion detection*) taas tarkkailee käyttäjien normaalia käytöstä ja raportoi käytöksen muutoksista [TCL90, FHSL96]. Vaikka käyttäjät voidaankin profiloida varsin onnistuneesti, tämä lähestymistapa kärsii herkästi vääristä hälytyksistä. Toisaalta, mikäli käyttäjä tuntee tarkkailevan järjestelmän riittävän hyvin, hän voi teoriassa hitaasti "kouluttaa" profiloijan pitämään muuttunutta, haitallista käytöstään normaalina, jolloin hyökkäystä ei kenties enää huomata. Tunkeutumisenestojärjestelmän opettamiseen ei tulekaan käyttää jatkuvasti kerättävää tarkkailutietoa sen sisältöä tarkistamatta, sillä tällöin myös tiedossa mukana olevat, havaitsematta jääneet hyökkäykset mielletään yhä vahvemmin normaalikäytökseksi.



Kuva 13: Eri lähestymistapoja tunkeutumisen havaitsemiseksi.

Kahden edellisen mallin parhaat puolet, vääriä hälytyksiä vähentävä asiantuntijatieto ja uusimpiakin hyökkäystyyppejä havaitseva normaalin käytöksen tunteminen, voidaan yhdistää määritelmäpohjaiseksi järjestelmäksi (engl. *specification-based intrusion detection*) [SGF+02]. Siinä otetaan huomioon sovelluksen kannalta odotettava suoritusjärjestys, joka selvitetään esimerkiksi sen lähdekoodista tai muusta määrittelevästä dokumentaatiosta. Tästä poikkeava toiminta katsotaan siten merkiksi hyökkäyksestä. Määritelmien kehittäminen on kuitenkin varsin työlästä etenkin suuremmille sovelluksille, joten tarkkailu saatetaan rajata esimerkiksi vain etuoikeutettuina ajettaviin sovelluksiin tai niiden osiin. Paras tulos saataisiin varmaankin yhdistelemällä erityyppisiä järjestelmiä sopivassa suhteessa.

Esimerkiksi IDES yhdistää väärinkäytösmallit ja vertailun käsitykseen normaalista toiminnasta [Den87]. Järjestelmien yhdistämistä ja tarkkailun hienojakoistamista kuitenkin rajoittavat sekä tulosten ihmistulkinnan tarve että tarkkailun ja analysoinnin laskennallinen raskaus. Luottamuksenhallintajärjestelmässä voikin olla järkevää käyttää luottamusta myös sen määrittämiseen, miten tarkkaan käyttäjän toimia seurataan. Tällöin tarkkailua voidaan vähentää riittävän luotettujen käyttäjien kohdalla. Tunkeutumisenestojärjestelmien esimerkkien lähestymistavat on koottu kuvaan 13. Esimerkkiä kokemuksesta opittavaan väärinkäytösten havaitsemineen ei löytynyt; todennäköisesti hyökkäyksiä tarkkailemalla niitä tunnistamaan oppiva järjestelmä vaatii niin paljon hyökkäyksiä oppimateriaalikseen, että yleisemmin ihmiset vain käyvät eri lähteistä saamaansa materiaalia läpi ja määrittävät sen pohjalta suoran spesifikaation järjestelmää varten.

Tarkkailu voidaan myös ulkoistaa käyttämällä paikallisen kokemustiedon sijaan "luotettuja" mainejärjestelmiä [RZFK00], jotka esimerkiksi kaupankäynnin yhteydessä kokoavat asianosaisilta, joskus myös ulkopuolisilta todistajilta, arvion toimituksen

onnistumisesta. Tällöin näiden mainejärjestelmien käsitys toimijan maineesta otetaan käyttöön sen tarkemmin arvioimatta; eli tarkkailu on täysin delegoitu käytettyjen mainejärjestelmien ylläpitäjille. Tarkkaileva maineyhteisö rankaisee sisäisten sääntöjensä rikkojaa huonolla maineella, jonka seurauksena tämän kanssa ei haluta käydä kauppaa tai sille asetetaan epäedullisempia ehtoja. Tällöin arvioijina ja luottamuspäätöksen tekijöinä ovat yleensä ihmiset, jotka saattavat pohjata päätöksensä järjestelmän ylläpitämän luottamusarvion lisäksi toistensa antamille vapaamuotoisille lausunnoille arvioinnin kohteen käytöksestä. Silloin kun arviot ovat täysin koneluettavia, mainejärjestelmät voivat antaa arvokkaan lisän käyttäjää koskeviin luottamustietoihin. Toimintojen ollessa erilaisia yhteisön eri jäsenillä voi tämä muilta saatava mainetieto jäädä kuitenkin välttämättömien tulkintojen ja yleistysten takia informaatisisällöltään köyhäksi.

Pelkän maineen käytön etuna on tarkkailuun tarvittavan työn ja laskentakapasiteetin säästyminen. Tällöin on kuitenkin luotettava täysin muiden tarkkailuun. Yhteisö ei luonnollisesti voi toimia, jos kaikki odottavat muiden hoitavan tämän tehtävän, mutta delegaation avulla tarkkailuun erikoistuneet yhteisön jäsenet voivat tarjota kokemuksiinsa perustuvaa tietoa mainepalveluna [Tan03].

Sekä tunkeutumisenesto- että mainejärjestelmät tarkkailevat käyttäjiä tilanteissa, joihin liittyy sekä ulkopuolelta havaittavaa että piilevää tietoa. Verkkoliikennettä tarkkaileva järjestelmä esimerkiksi ei näe paketteja tulkitsevan sovelluksen sisäistä tilaa, eikä kaupankäynnin todistaja kykene havaitsemaan osapuolten välistä viestintää tarkkailemansa tilanteen ulkopuolella. Mitä enemmän tällaista “sisäpiiritietoa” on saatavilla, sen totuudenmukaisempi kuva tarkkailijoilla on tilanteesta. Sovellustasolla toimiva tunkeutumisenestojärjestelmä ymmärtää sovelluksen toimintaa ja voi siten tehdä tarkempia arvioita tilanteesta kuin IP-tason verkkoliikennettä tarkkaileva järjestelmä. Diego Zamboni ehdottaa sisäisten “sensorien” liittämistä tarkkailtaviin sovelluksiin [Zam01]. Tämä ei yleisessä tapauksessa ole välttämättä toteutettavissa, sillä vaikka sovelluksen lähdekoodi olisi saatavilla ja muokattavissa, sensorien koodin lisääminen oikeisiin kohtiin alkuperäissovelluksen koodia vaatii syvällistä sovelluksen toiminnan tuntemista. Toisaalta mikäli esimerkin rajauksen mukaisesti sovelluksiin lisätään luontivaiheessa tukea luottamuksenhallintaan, jolloin myös tällaisten sensorien lisääminen mahdollistuu. Jotta luottamuksenhallintajärjestelmä olisi mahdollisimman pitkälti erotettavissa sovelluksesta, jaetaan sovelluksen piiriin mahdollisimman pieni osuus tarkkailuvastuuta ja -semantiikkaa. Sovellukseen voidaan esimerkiksi jättää vain koukut, jotka kutsuvat tietyissä tilanteissa sovelluskääreen koodia. Kääre hoitaa tällöin tilanteen arvioinnin ja tiedon lähettämisen eteenpäin kerättäväksi kokemukseksi.

Vaikka tunkeutumisenestojärjestelmät tarkkailevat enimmäkseen transaktioiden epäonnistumista, mainejärjestelmissä on kiinnitetty myös erityistä huomiota onnistumisiin [Obr04]. Maineen ylläpitämiseksi onkin tärkeää mahdollistaa myös sen paraneminen toimintojen edetessä suunnitelmien mukaan, sillä muuten jopa hyvin harvoin tapahtuvat poikkeamat johtavat lopulta kaikkien käyttäjien maineen huonontumiseen kohti nollatasoa. Myös käyttäjää tarkkailevien tunkeutumisenestojärjestelmien tulee siis voida välittää “ei ongelmia” -kokemustietoa mainejärjestelmälle, mikäli ne

eivät havainneet minkäänlaista hyökkäystä.

Koska web-palvelujärjestelmät perustuvat viestien lähetykseen, toiminta on toiseen ääripäähän, paikallisiin metodikutsuihin, verrattuna huomattavasti läpinäkyvämpää. Tämä mahdollistaa järkevämpien turvallisuuspäätösten tekemistä viestien sisällön pohjalta. Toisaalta viestien kuvaavuus ei aina riitä, sillä esimerkiksi tililtä nostaminen ja tilillepano saattavat näyttää viesteinä niin samanlaisilta, ettei niitä voi suoraan erottaa toisistaan [BHM+04]. Tähän tarvitaan lisätietoa sovelluksen ja viestien semantiikasta, jota WSDL ei sellaisenaan tarjoa. Joidenkin viestien parametrit voivat olla tärkeitä luottamuksen kannalta, kuten tilisiirtoesimerkissä (vertaa tilisiirto(A, B, 300) ja tilisiirto(A, B, -300)). Tällöin on kyseessä siirrettävän datan semantiikka.

Toisaalta viestien järjestys voi olla merkitsevä, esimerkiksi oston jälkeen tulisi seurata maksu, ja tietyt viestit tulkitaan kuuluvaksi tiettyyn toimintoon. Näiden suhteen tarvitaan semanttista tietoa kontrollin kulusta. Esimerkiksi yhteisöjen hallintaa tukevassa web-Pilarcos-järjestelmässä tarkkaillaan yhteisösopimuksen seuraamista, vaikkakin reaktiomahdollisuudet ovat vielä kehityskohteena. Yhteisösopimus määrittää muun muassa odotetun tapahtumien järjestyksen, jota voidaan käyttää valvonnan pohjana [KRMH05].

5.9 Käyttöesimerkki

Luottamusjärjestelmää käyttöönotettaessa kirjakauppasovelluksen kehittäjä osoittaa sovelluksen toiminnan kannalta sopivimman SOAP-viestien jaon eri toimintoihin ja toimintoluokkiin, sekä määrittää sovellukseen tarvittavat tarkistuspisteet kunkin toiminnon onnistumisen arvioimiseksi. Tässä hänellä on tukena kattava dokumentaatio.

Liitteessä 1 on esimerkki kirjan oston yhteydessä tarvittavien viestien WSDL-kuvauksesta, ja liitteessä 2 esitellään WSDL-kuvauksen mukainen viestienvaihto. Oletetaan, että kirjakauppa on jakanut ostotoiminnon kahdeksi riskiltään eriäväksi toiminnoksi kirjan hinnan mukaan, sillä kalliimpien kirjojen myyntiin liittyy toisaalta suurempi tärkeys liikevaihdon lisäämiseksi, mutta myös suurempi rahallinen riski, mikäli maksua ei jostakin syystä saada. Jotta sovelluskääre voisi erottaa nämä kaksi hintaluokaltaan eroavaa ostotoimintoa toisistaan, tulisi asiakkaan ensimmäisestä viestistä käydä ilmi kirjan hintaluokka. Yleisesti ottaen käyttäjä ei kuitenkaan ole paras lähde hinnan kysymiseen: hän on ensisijaisesti kiinnostunut tietystä kirjasta, eikä tietyn rahamäärän kuluttamisesta. Lisäksi käyttäjä ei suinkaan ole paras lähde kertomaan kirjan hintaa, vaan tarkin tieto saadaan kirjakaupan omasta tietokannasta.

Kirjan ostaminen alkaakin käyttäjän viestistä “olen kiinnostunut kirjasta, jonka ISBN on *X*” (BuyBookMsg). Hintatietojen puutteen takia viesti ei sovellu toiminnon aloittamiseen sellaisenaan. Sen sijaan sovelluskääre joutuu kysymään sovellukselta tai suoraan kirjakaupan tietokannasta kyseisen kirjan hintaa voidakseen muotoilla kyselynsä luottamuksenhallintajärjestelmälle. Kääre voi tehdä haun viestissä anne-

tun ISBN-numeron perusteella ja käyttää saamaansa tietoa kirjan hinnasta toiminnon tunnistamiseksi.

Mikäli ostotoimintoa ei olisi jaettu kahtia hintaluokan mukaan, sen riski voitaisiin myös määrittää toiminnon parametrien funktiona. ISBN ei kuitenkaan sellaisenaan sovi riskiarvoa laskevan funktion parametriksi sen enempää kuin toiminnon jakoperusteeksikaan, vaan kirjan hinta on yhä selvitettävä tietokannasta. Jotta riskilausekkeihin ei tarvitsisi sisällyttää aritmeettisia toimintoja monimutkaisempaa parametrikäsittelyä, esimerkin kannalta kahteen toimintoon jako on parempi vaihtoehto: sen seurauksena vain sovelluskääre on tavallista tiukemmin kytketty itse sovellukseen, kun taas luottamuksenhallintajärjestelmän toteutus voi muuten pysyä sovellusneutraalina.

Oletetaan, että ostettava kirja sijoittuu kalliimpaan hintaluokkaan (esimerkiksi yli EUR 15 maksaviin kirjoihin). Kirjakauppa on jaotellut jotkin maksutavat muita riskialttiimmiksi ja haluaa sovelluksensa kertovan vastausviestissä, mihin maksutapoihin käyttäjän maine riittää. Tämä voidaan selvittää kahdella tavalla. Mikäli luottamuspäätöksen tekevän alijärjestelmän vastaustapa halutaan pitää yksinkertaisena, kyllä/ei -muotoisina, kääreen tulee tehdä yksi kysely kutakin maksutavan riskiluokkaa kohden. Toinen vaihtoehto on käyttää räätälöityjä kontekstiasetuksia, jotka luottamuspäätöskoneisto tietää voivansa asettaa itse: esimerkiksi "`ei laskulla maksamista; jos toiminto = osto niin riski / 1,4 muuten riski`". Tällöin päätöskoneisto vastaa joko kerralla "kyllä" kaikkeen, tai rajoitetusti "kyllä, jos seuraavat kontekstiasetukset ovat voimassa: ei laskulla maksamista", mikäli riskin laskeminen on tarpeen. Kontekstiasetusten tällainen käyttö kuitenkin taipuu tehtävään hieman huonosti, sillä lisäksi olisi tarpeen rajata laskulla maksaminen pois vain tältä käyttäjältä eikä koko asiakasyrityksen järjestelmältä kerrallaan.

"Kyllä, jos..." -muotoisten luottamuspäätösten tulisikin rakentua erityisille sisäisesti asetettaville käyttäjäryhmille, jolloin niiden seurauksena tulisi voimaan haluttu kontekstiasetus ryhmään kuuluvuutta testaavan ehtolauseen täyttyessä. Kaikki käyttäjäryhmät eivät ole automaattisesti asetettavissa; esimerkiksi käyttäjää ei kenties voi liittää "avainasiakas"-ryhmään ennen kuin hän on käynyt henkilökohtaisesti solmimassa erityisen sopimuksen luottamuksen lähteen kanssa. "Erityistarkkailu", "laskutusmahdollisuus estetty" tai muut järjestelmän sisäisesti asetettavissa olevat ryhmät vaativat oman tietorakenteensa, josta käy ilmi ryhmän vaikutus luottamuspäätökseen ja toimenpiteet, jotka ryhmään liittäminen aiheuttaa—kuten esimerkiksi erityistarkkailun käynnistäminen. Ryhmän vaikutus luottamuspäätökseen saadaan toki selvitettyä kontekstiasetuksista, mutta luottamuspäätöskoneisto tarvitsee myös päätösvaiheessa nopeasti saatavilla olevan tiedon siitä, millaisilla ryhmiin liittämisillä se voi saada niukasti kielteisen luottamuspäätöksen muutettua positiiviseksi.

Oletetaan, että edellä kuvatun esimerkin kirjakauppa on jaotellut käyttäjien maineen lähinnä neljälle tasolle, 1–4, joista 1 on hyvin epäilyttävä, 2 oletusmaine ja 4 erityisen luotettu asiakas. Tavalliset asiakkaat asettuvat muutaman ostoksen jälkeen hieman tason 3 alle, kun taas pitkän, säännöllisen asiakassuhteen tuloksena voidaan nousta tasolle 3,5. Koska kauppa tarjoaa vain hyvin vähän toimintoja, yksinkertaistetaan

toimintokohtainen maine koskemaan kaikkia toimintoja.

Luottamuspäätöksessä käytetään kaavaa "riski <= maine + tärkeys". Toimintojen tärkeys on suhteessa tähän päätöstapaan asetettu arvoihin 0-1, joista tärkeysarvon 0 toiminto kuuluu palveluun, mutta ei varsinaisesti hyödytä yritystä, esimerkiksi kirjojen saatavuuden selailu, ja 1 vastaa yritystä suoraan hyödyttävä toimintaa, kuten kirjojen ostamista. Riskiarvot vaihtelevat välillä 1-5, jossa 5 on lähinnä teoreettinen maksimi.

Kirjojen saatavuuden selailu on hyvin matalariskistä, joten sen riskiarvo on asetettu 1:ksi—jopa hyvin epäilyttävät asiakkaat voivat selailla varastoa. Toiminto on kuitenkin luottamuksenhallinnan piirissä, joten esimerkiksi ruuhkautumisen seurauksena selailua voidaan rajoittaa kontekstiasetuksin. Yksittäisten kirjojen ostaminen välittömästi maksaen (luottokortilla tai verkkopankin kautta) on riskarvoltaan tasolla 3. Tällöin uudet käyttäjät, oletusmaineeltaan 2, voivat ostaa kirjoja (3 <= 2 + 1). Postiennakkoa käytettäessä riski on hieman suurempi, sillä uusia käyttäjiä luomalla hyökkääjä voisi tilata suuren määrän kirjoja tekaistuun osoitteeseen, jolloin kirjat eivät ole myytävissä ennen palautumistaan. Mikäli uuden käyttäjän yhtäaikaisten postiennakkotilausten määrää rajoitetaan, riskinä on yhä useiden uusien käyttäjien luonti. Ellei tätä kyetä rajoittamaan luottamuksenhallintajärjestelmän ulkopuolisin keinoin, lienee parasta estää postiennakkomyynti aivan uusilta käyttäjiltä nostamalla sen riskiarvo esimerkiksi 3,1:een. Laskulla saavat maksaa vain useita kirjoja aiemmin tilanneet asiakkaat, joten laskutetun ostamisen riski on 4, mikä vaatii tason 3 mainetta. Kalliit kirjat ovat ennakkoon maksettuna samaa riskitasoa, mutta postiennakon ja laskun riskiarvot ovat 0,1 ja 0,2 yksikköä suuremmat, 3,2 ja 4,2.

Koska ostotoiminto on jaettu kahtia vain kalleuden suhteen, täytyy korotetut riskiarvot ilmaista joko toiminnon parametrien tai järjestelmän tilaa kuvaavien kontekstiasetusten avulla. Oletetaan, että luottamuspäätös on vain muotoa kyllä/ei, jolloin eri vaihtoehtojen selvittämiseksi tulee tehdä kolme kyselyä: yksi perusriskitasolla, yksi postiennakkoa varten ja yksi laskulla maksua koskien. Ensimmäisenä tarkistetaan perustaso, sillä suurimman osan käyttäjistä voidaan olettaa olevan joko uusia tai melko uusia käyttäjiä ja siten maineeltaan lähellä 2:a, mikä sallii verkkopankkiostoksen muttei laskulla maksua. Käyttäjää voidaan siis positiivisten luottamuspäätösten jälkeen siirtää ryhmiin "postiennakko sallittu" ja "laskulla maksu sallittu", mitkä liittyvät kontekstiasetuksiin muotoa "`kallis osto laskulla;` **jos** `käyttäjä = laskulla maksu sallittu & toiminto = suuri osto` **niin** `riski + 0,2` **muuten** `riski`". Huomataan, että kalliin oston pohjariski on sama kuin halvemman oston, ja jaottelua käytetään hyväksi vain kontekstiasetusten kautta.

Liitteen 2 kirja on hinnaltaan kalliin rajan (EUR 15) yläpuolella, joten sen ostamista koskevat riskit 3, 3,2 ja 4,2. Oletetaan, että kirjaa tilaava käyttäjä on ostanut muutaman kirjan aiemmin, ja hänen maineensa on noussut näiden onnistuneiden toimintojen oletusarvosta 0,2 yksikköä arvoon 2,2. Perustasolla luottamuspäätöskoneisto vastaa, että osto sallitaan: 3 <= 2,2 + 1. Käyttäjä lisätään kokeeksi ryhmään "postiennakko sallittu", jolloin luottamuspäätöskoneisto vastaa yhä myönteisesti: 3,2 <= 2,2 + 1. Sen sijaan kun käyttäjä lisätään ryhmään "laskulla maksu sallittu",

luottamuspäätös muuttuu negatiiviseksi: $4{,}2 > 2{,}2 + 1$. Täten sovellukselta kääreen kautta lähtevässä paluuviestissä (ks. liite 2, SOAP-vastaus 1) ilmoitetaan, että maksutapoina sallitaan postiennakko, luottokortti tai verkkopankki.

Kirja on kenties merkitty hetkellisesti varatuksi, jotta sitä ei voida myydä kahteen kertaan. Sovellus ja kääre tarkkailevat nyt käyttäjää, ja niillä on oltava tieto siitä, miten käyttäjän maine muuttuu eri vakavuustason rikkeistä. Mikäli kirjan myynti onnistuu, käyttäjän maine kasvaa: kalliista ostosta 0,2 yksikköä, normaalihintaisesta 0,1 yksikköä. Tällöin yhden kirjan ostettuaan uusi käyttäjä voi tilata samanhintaisen tai halvemman kirjan seuraavalla kerralla postiennakolla, koska hänen maineensa on noussut tähän tarvittavan rajan (2,1 tai 2,2) yli. Noin viiden kalliin kirjan tai kymmenen halvan kirjan ostamisen jälkeen käyttäjä katsotaan jo riittävän luotettavaksi laskulla maksamiseen ainakin halpojen kirjojen suhteen. Mikäli laskua ei makseta, se siirretään yhä perintään, mutta kerralla laskutukseen siirrettäviä summia on kenties tarpeen rajoittaa. Maineen korottaminen lopetetaan tarpeettomana, kun se nousee arvoon 4. Matemaattisesti kauniimmin tämä voidaan myös toteuttaa jonkin yhteenlaskua monimutkaisemman, neljää ikuisesti lähestyvän ja loputtomiin kasvultaan hidastuvan kaavan avulla, mutta tämä ei ole esimerkin kannalta olennaista.

Ostos voi myös päättyä epäonnistuneesti useiden eri syiden takia, joista kaikkia ei voida havaita yksin järjestelmästä. Mikäli käyttäjä ei koskaan reagoi ensimmäiseen vastausviestiin ilmoittamalla maksutietojaan, voidaan tietyn ajan kuluttua päättää esimerkiksi laskea tämän mainetta 0,05 yksiköllä rikkeen toistuttua esimerkiksi viidesti lyhyen ajan sisällä, ja tämän jälkeen kerran joka toiston jälkeen aina pohjaarvoon 1 asti. Hidas lasku estää erityisesti uusien käyttäjien maineen laskemisen ostorajan alle yksittäisen yhteyden katkeamisen takia, ja mahdollistaa yhteyskatkoista huolimatta luotettavaan asiakkuuteen yltämisen. Viestinvaihdon keskeyttäminen on esimerkki jokseenkin harmittomasta käytöksestä, josta kenties kuitenkin halutaan olevan jonkinlaisia seuraamuksia sen aiheuttaman haitan (kirjan varaaminen hetkeksi) takia.

Esimerkki hieman vakammasta rikkeestä on kirjan tilaaminen postiennakolla tekaistuun osoitteeseen. Tämä varaa kirjan pidemmäksi aikaa, mutta tuote palautuu ajallaan kauppaan. Kauppa on päättänyt rankaista tästä saman verran kuin mitä vastaavan kirjan onnistuneesta ostamisesta palkitaan, jolloin mainetta lasketaan kirjan hinnasta riippuen 0,1 tai 0,2 yksikköä, kunnes se on arvossa 2. Tällöin tämänlaatuinen hyökkäys ei enää onnistu, koska käyttäjän maine ei riitä postiennakolla ostamiseen ennen kuin tämä on jälleen ostanut kaupasta kirjoja. Kutakin ostettua kirjaa vastaan voidaan yksi tilata postiennakkona ja jättää hakematta, jolloin jatkuva postiennakkotilauksilla tehtävä vandalismi käy käyttäjälle kalliiksi.

Olettaen että kirjakaupan sovellukseen ei sisälly vakavaa tietomurron riskiä, on käyttäjälle suurin mahdollinen rike tilata kirja laskulla ja jättää se maksamatta. Kirjakauppa laittaa toki laskun perintään, ja sen taloudellinen riski on yhä jokseenkin pieni, mutta rikkeestä seuraa kaupalle muita kuluja siinä määrin, että rangaistus on päätetty asettaa 0,6 maineyksikön laskuun. Tämä vastaa kolmen kalliin kirjan ostamisesta tulevaa mainekertymää. Maine ei voi yksittäisen laskun maksamatta

jäämisen takia koskaan laskea alle ostamiskyvyn kannalta kriittisen arvon (2), koska lähtöarvo on vähintään 3, mutta useiden laskujen jäädessä yhtäaikaa maksamatta käyttäjä ei kenties voi enää lainkaan ostaa kaupasta—luomatta uutta käyttäjää.

Sekä postiennakon väärinkäyttö että laskun maksamatta jättäminen ovat molemmat toimintoja, joita ei voida havaita yksin järjestemän sisällä. Aikaväli tilauksesta laskun eräpäivään tai postiennakkolähetyksen viimeiseen hakupäivään on järjestelmän kannalta hyvin pitkä, ja ilmoitus väärinkäytöksestä tai oston onnistuneesta päätökseen tuomisesta tulisi saada järjestelmän ulkopuolelta. Laskun erääntyminen voidaan hoitaa myös sisäisellä ajastimella, mutta laskuihin erikoistunut järjestelmä on parempi tiedonlähde. Tällöin yhden toiminnon ajallinen kesto venyy hyvin pitkäksi, mikä puolestaan monimutkaistaa tarkkailua. Tarkkailuyksikkö saa nyt tietoa myös järjestelmän ulkopuolelta, ja muokkaa tiettyyn toimintoon liittyvää mainetta vasta pitkään toiminnon aloittamisen jälkeen. Jonkinlaista tilatietoa on tarpeen säilyttää tänne asti.

Esimerkkijärjestelmässä on tähän mennessä määritelty vain kolme toimintoa, kaksi ostotoimintoa ja saatavuustietojen selaus. Palveluntarjoaja on hyvin tiukasti erikoistunut. Laajempi järjestelmä sen sijaan voi koostua useista kymmenistä toiminnoista, joita kaikkia yksi käyttäjä ei välttämättä tule koskaan käyttäneeksi. Kirjakaupassa vain kaksi toimintoa riippuu maineesta, ja koska molemmat olivat ostotoimintoja, maineen suora yleistäminen koskemaan molempia yhtäaikaa sopi paitsi esimerkin pitämiseen yksinkertaisena, myös järjestelmän toimintaan. Tällaisessa tilanteessa toimintokohtaisen maineen tuoma lisätieto on merkitykseltään pientä. Kuitenkin kauppaesimerkissäkin käytettiin eräällä tapaa myös toimintokohtaista mainetta: kalliin kirjan ostamisesta sai kaksinkertaisen mainekorotuksen halpaan kirjaan verrattuna. Tämä on verrattavissa siihen, että halvan kirjan ostamiseen liittyvä maine vaikuttaa puolta vähemmän käyttäjän pohjamaineeseen kuin kalliin kirjan.

Paikallisessa järjestelmässä maineella on merkitystä on lähinnä kokemustietojen tiivistäjänä. Yhteisötasolla myös maineen välittäminen järjestelmien välillä tulee mahdolliseksi. Lisäksi mainetta voidaan käyttää yhteisöjä muodostettaessa sen jäsenien valinnan tukena. Seuraavassa luvussa laajennetaan yhden palveluntarjoajan paikallinen luottamuksenhallinta usean, itsenäisiä luottamuspäätöksiä tekevän toimijan yhteisöihin.

6 Luottamuksenhallinta yhteisöissä

Yhteisö koostuu joukosta itsenäisiä toimijoita, ja se muodostetaan palvelemaan yhteistä tarkoitusta. Prosessin käynnistää yhteisön luonnista kiinnostunut yksittäinen toimija. Yhteisön muodostusta ohjaa yhteisömalli, joka määrittää halutun yhteispalvelun toteuttamisessa tarvittavat roolit ja niiden välisen toiminnan. Roolien täyttäjiksi valitaan yhteisöä muodostettaessa joukko toimijoita. Yhdistämisen teknisiä vaikeuksia voidaan vähentää palvelukeskeisen Web Services -arkkitehtuurin avulla; ne piilottavat toteutusyksityiskohdat SOAP-viestien taakse, jolloin rajapintojen epäyhteensopivuuden ongelmat pienenevät huomattavasti.

Yhteisön elinkaaressa on kolme keskeistä vaihetta, joissa kussakin luottamuksella on oma roolinsa. Yhteisöä luotaessa luottamusta käytetään kahdella tasolla: roolien täyttäjien valinnassa ja näiden yhteensopivuutta varmistettaessa. Yhteisömalli voi sisältää erilaisia muodostustavoitteita ja vaatimuksia, jotka yhteisön tietyn roolin toteuttavan toimijan tulee täyttää. Yksi tällainen tavoite voi olla maineen maksimointi; tällöin korkeamaineiset palveluntarjoajat ovat muita paremmassa asemassa valittaessa tietyn roolin täyttäjää. Myös itse roolin palveluun liittyviä parametrejä voidaan huomioida, kuten esimerkiksi hinta, toimitusaika tai lisäpalvelut, kuten takuut ja vakuudet. Mikäli roolin palvelulle on olemassa tietynlaisia standarditoteutuksia, voidaan tällaisille määrittää jopa tärkeys- ja riskiarvot, joita voidaan yhtä lailla pyrkiä minimoimaan tai maksimoimaan. Täysin yleisessä tapauksessa riskin ja tärkeyden arviointi yhteisön kannalta ilman tarkempia ennakkotietoja voi kuitenkin olla mahdotonta; itse palvelun tarjoaja voi ehdottaa tähän arvoja, mutta sen intresseissä voi olla ensisijaisesti itselleen myönteisten arvojen ilmoittaminen niiden oikeellisuuden sijaan.

Yhteensopivuutta varmistettaessa rooleja täyttävien palveluntarjoajien on tarkistettava, että näiden paikalliset luottamuksenhallintajärjestelmät sallivat nykytiedon, erityisesti muiden toimijoiden maineen perusteella rooliin liittyvien palveluiden kutsumisen. Mikäli jokin palveluntarjoaja ei lähtökohtaisesti salli toisen toimijan käyttää palvelua, jota tämän täyttämä rooli kutsuu yhteisömallissa, ei yhteisö voi toimia sellaisenaan. Tilannetta voidaan korjata vaihtamalla roolien täyttäjiä tai käymällä luottamusta muodostavia neuvotteluja. Toimija, jonka maine on riittämätön, voi vedota toisten toimijoiden mainejärjestelmistä saataviin suosituksiin. Se voi myös esittää todisteita tiettyihin hyvämaineisiin ryhmiin kuulumisesta, esimerkiksi varmenteiden muodossa. Lisäksi voidaan neuvotella vakuuksia tai rajoituksia palveluntarjoajan riskin pienentämiseksi, lähinnä tämän omasta aloitteesta, jolloin luottamuspäätös voidaan kääntää myönteiseksi vaikka toisen toimijan maine pysyisikin samana.

Yhteisön toiminnan aikana sen jäsenet tekevät luottamuspäätöksiä toisten yhteisön jäsenten päästämisestä käyttämään palvelujaan, ja keräävät kokemusta näiden toiminnasta. Vaikka yhteensopivuus varmistetaankin yhteisöä luotaessa, voi kesken toiminnan jonkin toimijan maine laskea joko toisten jäsenten huonojen kokemusten pohjalta, tai jonkin yhteisön jäsenen mainejärjestelmän vastaanottaessa negatiivisia suosituksia yhteisön ulkopuolelta. Yhteisö ole erillään muusta maailmasta; kukin toimija voi yhtäaikaisesti kuulua useisiin yhteisöihin ja toimia niissä eri tavoin. Mikäli toimijan maine romahtaa yhtäällä, se voi laskea rajusti myös yhteisössä negatiivisten suositusten vaikutusten levitessä muiden yhteisön jäsenien mainejärjestelmien kautta niiden luottamuspäätöksissään käyttämiin mainearvoihin. Mikäli toimijan maine laskee niin alas, ettei se saa enää käyttää yhteisön kannalta tarpeellista palvelua, yhteisö voi hajota. Tätä ennen voidaan kuitenkin aloittaa uudelleen luottamuksen muodostamisen neuvottelut, jotka etenevät samaan tapaan kuin luontivaiheessa.

Yhteisön kannalta jotkin toiminnot voivat olla tärkeämpiä kuin ne olisivat yksin paikallisessa kontekstissa, ja tämä tieto voidaan ottaa huomioon paikallisissa toimintojen tärkeysarvoissa. Sen tarkkaan huomiointitapaan voidaan myös ottaa kantaa

yhteisön toimintaa sitovassa sopimuksessa. Tästä johtuva korotus voisi muiden edellä kuvattujen toimenpiteiden tapaan saada palveluntarjoajan luottamuspäätöksen muuttumaan negatiivisesta positiiviseksi, mutta sen teho pohjautuu ainoastaan sopimiseen. On huomattavaa, että koska luottamuksenhallinnan tehtävä on paikallisen järjestelmän suojaaminen, ei se salli yhteisön sanella luottamuspäätöksiin vaikuttavia tekijöitä ilman lähdekritiikkiä. Palveluntarjoajan olisi siis oltava muiden jäsenten kanssa samaa mieltä siitä, että vaikka se ei sallisi tietyn yhteisön jäsenen käyttää palveluaan tavallisesti, se sallii palvelun käytön yhteisön toiminnan mahdollistamiseksi.

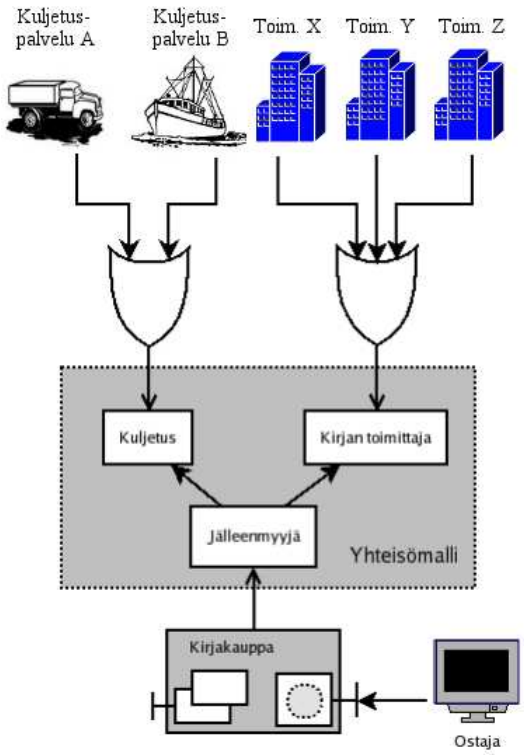
Yhteisön toiminta päättyy joko luottamusrikon takia tai sille asetetun tavoitteen tultua täytetyksi. Yhteisön hajoamisvaiheessa yhteisösopimukset puretaan ja niissä sovitut jatkotoimenpiteet otetaan käyttöön; esimerkiksi jonkin tuotteen valmistanut yhteisö joutuu sopimaan, miten valitusvastuu jaetaan sen varalta, että asiakas palauttaa tuotteen vaikkapa valmistusvirheen takia. Yhteisön hajoamisen luonteesta riippuen sen jäsenet voivat nyt kirjata itselleen muiden jäsenien suhteen joko myönteistä tai kielteistä kokemusta.

Kirjakauppa voi synnyttää yhteisön erikoisen kirjatilauksen täyttämistä varten. Sillä on asiakas, joka haluaa kirjan, ja se tarvitsee tilaukselle toimittajan ja kuljetuksen. Kirjakauppa käynnistää siis yhteisön luonnin. Kuljetusyritys ei todennäköisesti kykene täyttämään tilausta, vaan se erikoistuu kuljettamiseen. Harvinaisten kirjojen toimittaja ei kenties saa mitään tietoa tilaavasta asiakkaasta, joten se tarvitsee yhteisöä myynnin mahdollistamiseksi, ja kirjakauppa tuo yhteisöön asiakkaan, jonka se kenties on saanut juuri panostamalla näkyvyyteen ja hyvään palveluun myös harvinaisten kirjojen saamiseksi.

Kirjakaupan kolmen toimijan yhteisössä on sen oman roolin lisäksi kaksi yhteisöroolia tai palvelurunkoa, jotka se haluaa täytettäviksi: harvinaisen kirjan toimittajan ja kuljetuspalvelun. Mikäli yksi taho voisi täyttää molemmat roolit, ei tämä olisi haitaksi, mutta ilman yhden roolin täyttöä ei toisenkaan täytöstä ole hyötyä kirjakaupalle. Kirjakauppa määrittää tiettyjen roolien täyttämiseen haluamansa ehdot ja tavoitteet. Seuraavaksi ryhmä palveluntarjoajia, jotka kirjakauppa on etsinyt itse tai jotka ovat kuulleet tarjoutuneesta tilaisuudesta välityspalvelujen kautta, tekee tarjouksen siitä, miten he toteuttaisivat kyseisen palvelun.

Kuljetuspalvelu A ehdottaa, että kuljetuspalvelu toteutetaan autolla ovelta ovelle. Kuljetuspalvelu B tarjoaa satamasta satamaan -laivakuljetusta huokeasti. Kirjakauppa on kenties määrittänyt kuljetuspalvelun ehdoissa, että kuljetuksessa tulee maksimoida nopeus, joten rooliin valitaan kuljetuspalvelu A. Vaikka kuljetuspalvelu A osallistuu nyt yhteisöön, se voisi yhä kieltäytyä kirjan kuljettamisesta jos tehtävä rikkoisi sen sisäisiä sääntöjä. Jos kuljetuspalvelun toimipaikka olisi esimerkiksi hyvin uskonnollisella alueella, se saattaisi kieltäytyä kuljettamasta pyhäinhäväistykseksi katsottuja kirjoja. Tällöin yhteisö saattaa hajota ennen tavoitteensa toteutumista myös sen toiminnan jo alettua.

Hankittava harvinainen kirja ei kenties ole kaikkialla yhtä harvinainen, joten toimittajia saattaa ilmaantua kolme. Toimittaja X lupaa keskihintaisen toimituksen,



Kuva 14: Yhteisön rakentaminen mallin sisältämien roolien ympärille.

ilmoittaen ostavansa kirjan yhteyshenkilöltä. Toimittaja Y lupaa samoin keskihintaisen toimituksen, ilmoittaen hankintametodikseen vaihtamisen. Nämä toteutustavat on standardoitu, ja kirjakauppa on määrittänyt niihin liittyvän yhteisön kannalta varsin pienen riskin. Toimittaja Z lupaa erityisen halvan toimituksen; se toteuttaa hankinnan varastamalla. Toimittaja Z on toiminut kenties näin jo kauan pienessä mittakaavassa ja tietää, ettei toimintoon liity sen kannalta kovin suurta riskiä. Sen sijaan laaja kirjakauppa saattaa olla hyvin huolissaan julkisesta kuvastaan ja arvioi siksi toteutustavan riskin yhteisön kannalta liian suureksi. Toimittaja Z jää näin yhteisön ulkopuolelle, ja jäljelle jäävät vaihtoehdot ovat jokseenkin samankaltaiset toimittaja X ja toimittaja Y. Tässä valinnassa maine voi olla avuksi: kenties kirjakauppa on ollut tekemisissä molempien toimittajien kanssa jonkin verran, mutta on ollut erityisen tyytyväinen toimittajaan X, jonka maine onkin kirjakaupan järjestelmässä korkea tässä suhteessa. Mikäli kirjakauppa on asettanut riskiehtojen lisäksi roolin täyttämisessä tavoitteeksi myös maineen maksimoinnin, lopullinen valintapäätös tehdään nyt sen perusteella. Yhteisön rakentaminen roolien ympärille on esitetty kuvassa 14.

Lopuksi kirjakauppa, toimittaja X ja kuljetuspalvelu A solmivat sopimuksen, jossa määritetään yhteisön toimintasäännöt. Nämä säännöt voivat esimerkiksi ilmaista, että palvelupyyntöä ja sen toteutumista tulee seurata tietty palkkionmaksuprotokolla, ja että kirjakauppa vastaa yksin asiakkaan palautusoikeuden täyttämisestä; se luottaa yhteisökumppaniensa toimittavan sille oikean kirjan, eikä usko asiakkaan

olevan siihen tyytymätön.

Toimijat voivat sopimuksesta huolimatta yhä suojata itseään luottamuspäätöksin, joihin niiden partnereiden maine vaikuttaa. Mikäli esimerkiksi toimittaja X saisi tietää mainejärjestelmänsä kautta kesken yhteistoiminnan jotakin kirjakaupan toimista jotka romahduttaisivat sen maineen toimittajan X silmissä, se ei välttämättä enää luottaisi kirjakaupan esimerkiksi maksavan tilaamaansa harvinaista kirjaa sen saavuttua, ja kieltäytyisi yhteistyöstä. Tämän seurauksena voisi olla uudelleenneuvottelua esimerkiksi vakuuksista tai toimintajärjestyksestä (maksu ennen toimitusta), tai yhteisön hajoaminen ja mahdollisesti uuden yhteisön muodostaminen, mikäli toimittaja Y olisi yhä halukas täyttämään tyhjäksi jäävän roolin.

Yhteisön muodostaminen ja toiminta voi olla Web Services -teknologian tuesta huolimatta hankalaa viestipohjaisten rajapintojen hankalan tulkittavuuden ja sopimuksenmuotoilun semanttisten erojen takia. Web-Pilarcos-projekti [KRMH05] tarjoaa väliohjelmistotukea yhteisönmuodostukseen. Esimerkiksi tavoitteiden asettelu roolien täyttämisessä tiettyjen parametrien maksimoimiseksi on erityisen populoijapalvelun tehtävä. Projekti tukee myös sopimushallintaa ja esimerkiksi palvelutarjousten säilömistä yhteisen tai yhteensopivan ontologian mukaisesti. Web-Pilarcos-väliohjelmistopalvelut eivät kuitenkaan sellaisenaan sisällä minkäänlaista tukea luottamukselle; TuBE voi lisätä luottamuksenhallinnan palveluvalikoimaan [KVR05].

7 Yhteenveto

Autonomisten järjestelmien yhteistyö vaatii näiltä kykyä suojata itsensä myös muilta toimijoilta. Nopeasti muuttuvassa ympäristössä perinteinen pääsynhallinta ei riitä tähän tarkoitukseen. Parempi lähestymistapa onkin perustaa suojaus kokemuksen mukaan päivittyvälle luottamukselle. Aiemmat luvut kuvailevat mallin luottamuksenhallintajärjestelmän toteuttamiseksi. Tutkielmassa on keskitytty päätöksessä käytettävien tietojen tunnistamiseen ja niiden käsittelyyn järjestelmässä.

Pitkälle viety luottamukseen liittyvän tiedonkäsittelyn ja päätöksenteon automatisointi helpottaa suurien käyttäjämäärien käsittelyä ja vähentää jatkuvan inhimillisen valvontatyön tarvetta. Toisaalta kaiken päätösvallan siirtäminen koneelle ei välttämättä ole tavoiteltava päämäärä: ihminen haluaa yleensä sananvaltaa tärkeäksi kokemissaan asioissa [BHM+04]. Paras tulos saadaankin yhdistämällä: luottamuksenhallintajärjestelmä voi hoitaa rutiinipäätökset ja vapauttaa siten ihmiset muun muassa pääsynhallintapäivitysten rasitteesta, mutta rajatapaukset ja muut erikoistilanteet voidaan yhä jättää ihmisen selvitettäväksi.

Tietyn yksilön luotettavuuden keskeisin mittari on tämän maine. Paikallisesta näkökulmasta maine on ensisijaisesti kokemustiedosta koottu yhteenvetoarvo. Yhteisötasolla sen merkitys kasvaa, sillä mainetietoja voidaan välittää suositusten avulla luottamuksenhallintajärjestelmästä toiseen. Mainejärjestelmät ovat vastuussa paikallisen kokemuksen ja ulkoisten suositusten yhdistämisestä paikalliseksi mainekäsitykseksi.

Luottamuksenhallinta ei ole täysin ongelmatonta toteuttaa. Suurimmat haasteet liittyvät älykkään ja automatisoidun tarkkailun toteuttamiseen, maineen levitykseen sekä järjestelmän käyttöönotossa tarvittavaan tietomäärään. Maineen yleistämisen haasteita on käsitelty luvussa 5.7.

Nykyiset tarkkailujärjestelmät ovat lähinnä tunkeutumisenestojärjestelmiä, jotka toimivat varsin matalalla tasolla, IP-verkkopaketteja tai käyttöjärjestelmän palvelukutsuja seuraten. Sovellustason tarkkailu on tärkeää havaintojen saattamiseksi suurempaan kontekstiin; emme ole kiinnostuneita pelkästään vääränmuotoisten verkkopakettien välttämisestä, vaan myös siitä, sisälsikö SOAP-viesti esimerkiksi yhden vai tuhannen kirjan tilauksen ja hoituuko maksu luottokortilla vai postiennakkona. Tapahtumien ymmärtämisen automatisointi tällä tasolla vaatii kattavaa semanttista kuvausta viestien sisällöstä ja eri parametrien merkityksestä. Tämä ja yksittäisten havaintojen taustan ymmärtäminen "tavallisen" käytöksen määrittämiseksi edes käyttäjäkohtaisesti ovat melkoisia haasteita tarkkailujärjestelmän tekoälyn kehityksessä.

Maineen levittämisessä on kaksi haastetta, merkityksen välittäminen ja yksityisyyden suojaaminen. Yhteisön jäsenten suosituksia levittävät mainejärjestelmät voivat käsitellä hyvin erilaisiin toimintojoukkoihin liittyvästä kokemuksesta johdettua mainetta. Jos maineeseen ei liitetä toimintotietoa, vaan järjestelmien välillä levitetään lähinnä käyttäjäkohtaista pohjamainetta, tarvitaan maineyhteisössä yhä sopimus siitä, millä perusteella maine voi kohota tiettyyn arvoon. Koska toimijat ovat kuitenkin lopulta autonomisia, tulee suositusten vastaanottajan itse arvioida, miten yhtäpitäviä tämän havainnot ovat aiemmin olleet suosituksen lähettäjän kanssa, ja valita kriittisyytensä määrä vastaavasti.

Erityisesti yksityiskäyttäjien tapauksessa maineen levittäminen palveluntarjoajien kesken saattaa rikkoa yksityisyyden suojaa. Mikäli käyttäjä voi vaikuttaa maineen kulkuun suoraan, hän voi poistaa systemaattisesti kaikki negatiiviset suositukset levityksestä, jolloin tulos on vinoutunut. Toisaalta mikäli käyttäjä ohjautuu suojaamaan yksityisyyttään toimimalla joka järjestelmässä eri nimellä, sekä hän että luottamuksenhallintajärjestelmät menettävät hyödyn tietojen yhdistämisen mahdollisuudesta. Tähän ongelmaan on pohdittu ratkaisua mm. luomalla käyttäjille pseudonyymejä, jotka nämä voivat halutessaan todistaa toisiaan vastaaviksi [SJ04]. Tällöin käyttäjä voi esimerkiksi hyötyä eri paikkakunnan kirjakaupassa toisen kirjakaupan suosituksista, ja jättää ilmoittamatta esimerkiksi hyvää mainettaan radikaalipuolueen järjestelmässä.

Esitelty järjestelmä on varsin pitkälti muokattavissa erilaisten luottajien tarpeisiin. Luottamuspäätöksessä painotus eri luottamusvektorin arvojen välillä voidaan valita vapaasti. Lisäksi toimintojen eri lopputulosten vaikutukset maineeseen voidaan määritellä eri luottajille eri tavoin—kuten myös mahdolliset erilaiset lopputulokset kullekin toiminnolle; mutta tämä jää sovelluksen ja sitä ympäröivän kääreen koodin taakse. Kunkin toiminnon tärkeys- ja riskiarvot asetetaan yksilöllisesti, samoin oletusmaine uusille käyttäjille.

Muokattavuuden varjopuolena on käyttöönoton raskaus: järjestelmä vaatii paljon

tietoa toimiakseen hyvin. Päätösten politiikka-asetuksille voidaan asettaa järkeviä oletuksia, joista voidaan valita luottajalle parhaiten sopiva. Asetusten ja arvojen muokkaukseen voidaan tarjota käyttöliittymä, joka ohjaa esimerkiksi arvovalintoja tietyille väleille, jotka sopivat hyvin yhteen valitun politiikka-asetuksen kanssa. Riskiarvojen löytämiseen voidaan kehittää työkaluja; esimerkiksi CORAS-työkalu [BS04] tukee riskien löytämistä ja pisteyttämistä, vaikkakin pisteiden tarkoitus ei liitykään luottamuksenhallintajärjestelmän konfigurointiin vaan ennaltaehkäisyyn. Asiantuntija-apua tarvittaneen kuitenkin myös, vaikka ohjaava ja hyvin dokumentoitu käyttöliittymä voi auttaa jo huomattavasti.

Haasteista huolimatta luottamuksenhallinta on yksi luottamuksen tutkimuksen kasvavia osa-alueita. Jatkossa pyritään liittämään TuBE-luottamuksenhallintajärjestelmä web-Pilarcosin arkkitehtuuriin, mihin liittyen yhteisökontekstissa tapahtuvaa luottamuksenhallintaa tarkastellaan lähemmin.

Lähteet

BCE+02 Bellwood, T., Clément, L., Ehnebuske, D., Hately, A., Hondo, M., Husband, Y. L., Januszewski, K., Lee, S., McKee, B., Munter, J. ja von Riegen, C., *UDDI Version 3.0. UDDI Spec Technical Committee Specification, 19 July 2002.* UDDI.org, heinäkuu 2002. URL `http://uddi.org/pubs/uddi-v3.00-published-20020719.htm`.

BFK98 Blaze, M., Feigenbaum, J. ja Keromytis, A. D., KeyNote: Trust management for public-key infrastructures (position paper). *Proceedings of Security Protocols: 6th International Workshop, Cambridge, UK, April 1998.* Springer-Verlag, LNCS 1550/1998, huhtikuu 1998, sivut 59–63, URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=1550&spage=59`.

BFL96 Blaze, M., Feigenbaum, J. ja Lacy, J., Decentralized trust management. *Proceedings of the IEEE Symposium on Security and Privacy.* IEEE, toukokuu 1996, URL `http://ieeexplore.ieee.org/iel3/3742/10940/00502679.pdf`.

BHM+04 Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C. ja (Eds), D. O., Web Services architecture. W3C Working Group Note 11 February 2004. Tekninen raportti, World Wide Web Consortium, helmikuu 2004. URL `http://www.w3.org/TR/ws-arch/`.

BS04 Brændeland, G. ja Stølen, K., Using risk analysis to assess user trust - a net-bank scenario. *Proceedings of Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004.* Springer-Verlag, LCNS 2995/2004, maaliskuu 2004, sivut 146–160, URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=146`.

CFL+97 Chu, Y.-H., Feigenbaum, J., LaMacchia, B., Resnick, P. ja Strauss, M., REFEREE: Trust management for Web applications. *Computer Networks and ISDN Systems*, 29,8–13(1997), sivut 953–964. URL `http://citeseer.ist.psu.edu/58910.html`.

CGS+03 Cahill, V., Gray, E., Seigneur, J.-M., Jensen, C., Chen, Y., Shand, B., Dimmock, N., Twigg, A., Bacon, J., English, C., Wagealla, W., Terzis, S., Nixon, P., Serugendo, G. D. M., Bryce, C., Carbone, M., Krukow, K. ja Nielson, M., Using trust for secure collaboration in uncertain environments. *Pervasive Computing*, 2,3(2003), sivut 52–61. URL `http://ieeexplore.ieee.org/iel5/7756/27556/01228527.pdf`.

DDLS01 Damianou, N., Dulay, N., Lupu, E. ja Sloman, M., The Ponder policy specification language. *Workshop on Policies for Distributed Systems and Networks (Policy2001), HP Labs Bristol, 29-31 Jan 2001*, osa 1995, tammikuu 2001, sivut 18–, URL `http://citeseer.ist.psu.edu/damianou01ponder.html`.

Dem04 Demolombe, R., Reasoning about trust: A formal logical framework. *Proceedings of Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings*. Springer-Verlag, LNCS 2995/2004, maaliskuu 2004, sivut 291–303, URL `http://springerlink.metapress.com/link.asp?id=yalyru2brpq4fq2u`.

Den87 Denning, D., An intrusion-detection model. *IEEE Transactions on Software Engineering*, 13,2(1987), sivut 222–232. URL `http://www.cs.georgetown.edu/denning/infosec/ids-model.rtf`.

eBa05 The eBay online marketplace, 2005. `http://www.ebay.com/`. [24.6.2004]

Ell04 Ellison, C., SPKI/SDSI certificate documentation, marraskuu 2004. `http://world.std.com/~cme/html/spki.html`. [3.11.2004]

Ess97 Essin, D. J., Patterns of trust and policy. *Proceedings of 1997 New Security Paradigms Workshop*. ACM Press, 1997, URL `http://doi.acm.org/10.1145/283699.283738`.

ETW04 English, C., Terzis, S. ja Wagealla, W., Engineering trust based collaborations in a global computing environment. *Proceedings of Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings*. Springer-Verlag, LNCS 2995/2004, maaliskuu 2004, sivut 120–134, URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=120`.

EWN+03 English, C., Wagealla, W., Nixon, P., Terzis, S., McGettrick, A. ja Lowe, H., Trusting collaboration in global computing systems. *First International Conference on Trust Management*, toukokuu 2003, sivut 136–149, URL `http://springerlink.metapress.com/link.asp?id=yfhe8gg1398w088%e`.

FHSL96 Forrest, S., Hofmeyr, S., Somayaji, A. ja Longstaff, T., A sense of self for Unix processes. *1996 IEEE Symposium on Security and Privacy, May 6–8, 1996, Oakland, California*, toukokuu 1996, URL `http://ieeexplore.ieee.org/iel3/3742/10940/00502675.pdf`.

FKÖD04 Fernandes, A., Kotsovinos, E., Östring, S. ja Dragovic, B., Pinocchio: Incentives for honest participation in distributed trust management. *Proceedings of Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004*. Springer-Verlag, LNCS 2995/2004, maaliskuu 2004, sivut 64–77, URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=63`.

FSD+02 Fogg, B., Soohoo, C., Danielson, D., Marable, L., Stanford, J. ja Tauber, E. R., How do people evaluate a web site's credibility? Tekninen raportti, Stanford Persuasive Technology Lab, lokakuu 2002. URL `http://www.consumerwebwatch.org/news/report3_credibilityresearch/stanfordPTL_abstract.htm`.

Gam00 Gambetta, D., Can we trust trust? *Trust: Making and Breaking Cooperative Relations*, sivut 213–237. URL `http://www.sociology.ox.ac.uk/papers/gambetta213-237.pdf`. Electronic edition.

GKRT04 Guha, R., Kumar, R., Raghavan, P. ja Tomkins, A., Propagation of trust and distrust. *Proceedings of the Thirteenth International World Wide Web Conference, New York*. ACM, toukokuu 2004, sivut 403–412, URL `http://www.www2004.org/proceedings/docs/1p403.pdf`.

GMMZ04 Giorgini, P., Massacci, F., Mylopoulos, J. ja Zannone, N., Requirements engineering meets trust management—model, methodology, and reasoning. *Proceedings of Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004*. Springer-Verlag, LNCS 2995/2004, maaliskuu 2004, sivut 176–190, URL `http://www.springerlink.com/link.asp?id=k9gpgmbgf6hh0x21`.

GO05 Grønmo, R. ja Oldevik, J., An empirical study of the UML Model Transformation Tool (UMT). *Pre-proceedings of the First International Conference on Interoperability of Enterprise Software and Applications—INTEROP-ESA'2005, Geneva, Switzerland, February 23–25, 2005*, helmikuu 2005, sivut 511–522. Springer painaa varsinaisen proceedingskirjan, joka odottaa julkaisua.

GS00 Grandison, T. ja Sloman, M., A survey of trust in Internet applications. *IEEE Communications Surveys and Tutorials*, 3,4(2000), sivut 2–16. URL http://www.comsoc.org/livepubs/surveys/public/2000/dec/grandison.html.

GS02 Grandison, T. ja Sloman, M., Specifying and analysing trust for Internet applications. *Proceedings of 2nd IFIP Conference on e-Commerce, e-Business, e-Government I3e2002, Lisbon, Portugal*, lokakuu 2002, URL http://citeseer.ist.psu.edu/grandison02specifying.html.

Hof79 Hofstadter, D., *Gödel, Escher, Bach: an Eternal Golden Braid*. Harvester Wheatsheaf, 1979.

Hof93 Hofstede, G. *Kulttuurit ja organisaatiot—mielen ohjelmointi*, luku 5. WSOY, 1993. Suomennos Ritva Liljamo. Alkuperäisteos Cultures and Organizations: Software of the Mind, McGraw-Hill, 1991.

IET98 IETF X.509 Working Group, Public-key infrastructure (X.509), 1998. URL http://www.ietf.org/html.charters/pkix-charter.html [3.11.2004].

JP04 Jøsang, A. ja Presti, S. L., Analysing the relationship between risk and trust. *Proceedings of Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004*. Springer-Verlag, LNCS 2995/2004, maaliskuu 2004, sivut 135–145, URL http://springerlink.metapress.com/link.asp?id=mklyh19x5yb1c8n9.

JSTT04 Jonker, C. M., Schalken, J. J. P., Theeuwes, J. ja Treur, J., Human experiments in trust dynamics. *Proceedings of Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004*. Springer-Verlag, LNCS 2995/2004, maaliskuu 2004, sivut 206–220, URL http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=206.

Jøs96 Jøsang, A., The right type of trust for computer networks. *Proceedings of the ACM New Security Paradigms Workshop*. ACM, 1996, URL http://security.dstc.edu.au/staff/ajosang/papers/trdsyst.ps.

Kar03 Karabulut, Y., Implementation of an agent-oriented trust management infrastructure based on a hybrid PKI model. *Proceedings of Trust Management: First International Conference, iTrust 2003, Heraklion, Crete, Greece, May 28–30, 2003*. Springer-Verlag, LNCS 2692/2003, toukokuu 2003, sivut 318–331, URL http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2692&spage=318.

KFJ03 Kagal, L., Finin, T. ja Joshi, A., A policy language for a pervasive computing environment. *Proceedings of IEEE 4th International Workshop on Policies for Distributed Systems and Networks (POLICY 2003)*. IEEE, kesäkuu 2003, sivut 63–74, URL `http://ieeexplore.ieee.org/iel5/8577/27164/01206958.pdf`.

KN+04 Kaler, C., Nadalin, A. et al., *Web Services Trust Language (WS-Trust)*, toukokuu 2004. URL `ftp://www6.software.ibm.com/software/developer/library/ws-trust.pdf`. Version 1.1.

KRMH05 Kutvonen, L., Ruokolainen, T., Metso, J. ja Haataja, J.-P., Interoperability middleware for federated enterprise applications in web-Pilarcos. *Pre-proceedings of the First International Conference on Interoperability of Enterprise Software and Applications—INTEROP-ESA'2005, Geneva, Switzerland, February 23–25, 2005*, helmikuu 2005, sivut 196–208. Springer painaa varsinaisen proceedings-kirjan, joka odottaa julkaisua.

KS94 Kumar, S. ja Spafford, E. H., A Pattern Matching Model for Misuse Intrusion Detection. *Proceedings of the 17th National Computer Security Conference, Baltimore, Maryland, October 1994*, lokakuu 1994, sivut 11–21, URL `http://citeseer.ist.psu.edu/kumar94pattern.html`.

KVR05 Kutvonen, L., Viljanen, L. ja Ruohomaa, S., The TuBE approach to trust management in inter-enterprise systems, 2005. Käsikirjoitus lähetetty arvioitavaksi.

Lin01 Linthicum, D. S., *B2B Application Integration – e-Business-Enable Your Enterprise, 2nd Printing*. Addison-Wesley, Information Technology Series, elokuu 2001.

Mar94 Marsh, S., *Formalising Trust as a Computational Concept*. Väitöskirja, University of Stirling, Department of Computer Science and Mathematics, 1994. URL `http://citeseer.ist.psu.edu/marsh94formalising.html`.

MB04 Massa, P. ja Bhattacharjee, B., Using trust in recommender systems: An experimental analysis. *Proceedings of Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004*. Springer-Verlag, LNCS 2995/2004, maaliskuu 2004, sivut 221–235, URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=221`.

MC96 McKnight, D. H. ja Chervany, N. L., The meanings of trust. Tekninen raportti, University of Minnesota, MIS Research Center, 1996. URL `http://misrc.umn.edu/workingpapers/fullPapers/1996/9604_040100.pdf`. Taulukot liitteinä verkkoversiossa.

MD95 Mayer, R. C. ja Davis, J. H., An integrative model of organizational trust. *The Academy of Management Review*, 20,3(1995), sivut 709–734. URL http://links.jstor.org/sici?sici=0363-7425%28199507%2920%3A3%3C709%3AAIMOOT%3E2.0.CO%3B2-9.

MMH02 Mui, L., Mohtashemi, M. ja Halberstadt, A., A computational model of trust and reputation. *35th Annual Hawaii International Conference on System Sciences (HICSS'02)*, osa 7. IEEE Computer Society, tammikuu 2002, URL http://csdl.computer.org/comp/proceedings/hicss/2002/1435/07/14350188.pdf.

MWR89 Mitchell, C., Walker, M. ja Rush, D., Ccitt/iso standards for secure message handling. *IEEE Journal on Selected Areas on Communications*, 7, sivut 517–524. URL http://ieeexplore.ieee.org/xpl/tocresult.jsp?isNumber=648.

Obr04 Obreiter, P., A case for evidence-aware distributed reputation systems overcoming the limitations of plausibility considerations. *Proceedings of Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004*. Springer-Verlag, LNCS 2995/2004, maaliskuu 2004, sivut 33–47, URL http://www.springerlink.com/link.asp?id=ucc149f215dhyj0y.

Pap03 Papazoglou, M. P., Service-oriented computing: Concepts, characteristics, and directions. *WISE*. IEEE Computer Society, 2003, URL http://maximus.uvt.nl/sigsoc/pub/Papazoglou%20-%20Service-oriented%20computing%20-%20Concepts,%20characteristics%20and%20directions.pdf.

Pay05 The PayPal online payment site, 2005. http://www.paypal.com/. [24.6.2004]

RK05 Ruohomaa, S. ja Kutvonen, L., Trust management survey. *Proceedings of the iTrust 3rd International Conference on Trust Management, 23–26, May, 2005, Rocquencourt, France*. Springer-Verlag, LNCS 3477/2005, toukokuu 2005. Odottaa julkaisua.

RZFK00 Resnick, P., Zeckhauser, R., Friedman, E. ja Kuwabara, K., Reputation systems. *Communications of the ACM*, 43,12(2000), sivut 45–48. URL http://doi.acm.org/10.1145/355112.355122.

SGF+02 Sekar, R., Gupta, A., Frullo, J., Shanbhag, T., Tiwari, A., Yang, H. ja Zhou, S., Specification-based anomaly detection: a new approach for detecting network intrusions. *Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC, USA*, 2002, sivut 265–274, URL http://doi.acm.org/10.1145/586110.586146.

Sin03 Singh, M. P. *Trustworthy Service Composition: Challenges and Research Questions*, osa 2631/2003 sarjasta *Lecture Notes in Artificial Intelligence*, sivut 39–52. Springer-Verlag, 2003. URL http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2631&spage=39.

SJ04 Seigneur, J.-M. ja Jensen, C. D., Trading privacy for trust. *Proceedings of Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004*. Springer-Verlag, LNCS 2995/2004, maaliskuu 2004, sivut 93–107, URL http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=93.

Suo05 Suomen Posti Oyj, 2005. http://www.posti.fi/. [24.6.2004]

Tan03 Tan, Y.-H., A trust matrix model for electronic commerce. *Proceedings of Trust Management: First International Conference, iTrust 2003, Heraklion, Crete, Greece, May 28–30, 2003*, osa LNCS 2692/2003, toukokuu 2003, sivut 33–45, URL http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2692&spage=33.

TBJ+03 Tonti, G., Bradshaw, J. M., Jeffers, R., Montanari, R., Suri, N. ja Uszok, A., Semantic Web languages for policy representation and reasoning: A comparison of KAoS, Rei, and Ponder. *The SemanticWeb - ISWC 2003*. Springer-Verlag, LCNS 2870/2003, lokakuu 2003, sivut 419–437, URL http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2870&spage=419.

TCL90 Teng, H. S., Chen, K. ja Lu, S. C.-Y., Adaptive real-time anomaly detection using inductively generated sequential patterns. *1990 IEEE Symposium on Research in Security and Privacy, May 7–9, 1990*. IEEE Computer Society, toukokuu 1990, sivut 278–284, URL http://ieeexplore.ieee.org/iel2/300/2323/00063857.pdf.

UBJ04 Uszok, A., Bradshaw, J. M. ja Jeffers, R., KAoS: A policy and domain services framework for grid computing and Semantic Web services. *Proceedings of Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004*, osa Springer-Verlag, LNCS 2995/2004, maaliskuu 2004, sivut 16–26, URL http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=16.

Ver05 Verkkokauppa.com, Suomen vanhin ATK-alan verkkokauppa, 2005. http://www.verkkokauppa.com/. [24.6.2004]

Vil05a Viljanen, L., A survey on application level intrusion detection. Tekninen raportti, Helsingin yliopisto, Tietojenkäsittelytieteen laitos, 2005.

Vil05b Viljanen, L., Towards an ontology of trust, 2005. Käsikirjoitus lähetetty julkaistavaksi.

WCE+03 Wagealla, W., Carbone, M., English, C., Terzis, S. ja Nixon, P., A formal model on trust lifecycle management. Teoksessa *Workshop on Formal Aspects of Security and Trust (FAST2003) at FM2003*, osa IIT TR-10/2003, IIT-CNR, Italy, syyskuu 2003, sivut 184–195. URL `http://www.iit.cnr.it/FAST2003/fast-proc-final.pdf` (TR-10/2003).

Wik05 Wikipedia, vapaa tietosanakirja, 2005. `http://www.wikipedia.org/`. [24.6.2004]

WSJ00 Winsborough, W. H., Seamons, K. E. ja Jones, V. E., Automated trust negotiation. *DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings*, osa 1. IEEE, tammikuu 2000, sivut 88–102, URL `http://ieeexplore.ieee.org/iel5/6658/17862/00824965.pdf`.

Zam01 Zamboni, D., *Using Internal Sensors for Computer Intrusion Detection*. Väitöskirja, Purdue University, 2001. URL `http://www.cerias.purdue.edu/homes/zamboni/pubs/thesis-techreport.pdf`.

Liite 1. Yksinkertaistettu WSDL-kuvaus kirjamyynnille

```
<?xml version="1.0"?>

<!-- root element wsdl:definitions defines set of related
     services -->
<definitions name="Bookstore"
    targetNamespace="http://example.com/bookstore.wsdl"
    xmlns:tns="http://example.com/bookstore.wsdl"
    xmlns:xsd1="http://example.com/bookstore.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

     <!-- wsdl:types encapsulates schema definitions of
          communication types; here using xsd -->
     <wsdl:types>
        <!-- Type definitions for data used in messages. -->
        <xsd:schema
            targetNamespace="http://example.com/bookstore.xsd"
                xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">

             <!-- xsd definition: BookBuyRequest
                  [... ISBN string ...] -->
             <xsd:element name="BookBuyRequest">
                <xsd:complexType>
                    <xsd:all>
                        <xsd:element name="ISBN"
                          type="string"/>
                    </xsd:all>
                </xsd:complexType>
             </xsd:element>


             <!-- PriceAndPayment [... price float,
                  payMethodsAvailable string...]
                  (the latter is a list of payment types available;
                  it requires parsing but simplifies the
                  WSDL description) -->
             <xsd:element name="PriceAndPayment">
                <xsd:complexType>
                    <xsd:sequence>
                        <!-- Assume common currency, eg. EUR -->
                        <xsd:element name="price" type="float"/>
```

```
                    <xsd:element name="payMethodsAvailable"
                        type="string" />
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>

        <!-- PaymentInfo [... ISBN string, method string,
             info string, delivery string ...]
             All the info required of the customer, such as
             eg. credit card number, is hidden behind "info" -
             this simplification leaves the WSDL almost
             unusable by itself. 'delivery' contains information
             for delivering the book to the buyer. -->
        <xsd:element name="PaymentInfo">
            xsd:<complexType>
                <xsd:sequence>
                    <xsd:element name="ISBN"
                      type="string"/>
                    <xsd:element name="type" method="string"/>
                    <xsd:element name="info" type="string" />
                    <xsd:element name="delivery" type="string"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>

        <!-- SaleConfirmation [... message string ...]
             ('message' contains information to show to a user.) -->
        <xsd:element name="SaleConfirmation">
            xsd:<complexType>
                <xsd:all>
                    <xsd:element name="message" type="string"/>
                </xsd:all>
            </xsd:complexType>
        </xsd:element>

    </xsd:schema>
</wsdl:types>

<!-- Binding messages (their bodies) to specific aforedefined
     types. -->
<wsdl:message name="BuyBookMsg">
    <wsdl:part name="body" element="xsd1:BookBuyRequest"/>
</wsdl:message>

<wsdl:message name="SellingBookMsg">
```

```
    <wsdl:part name="body" element="xsd1:PriceAndPayment"/>
</wsdl:message>

<wsdl:message name="PayingForBookMsg">
    <wsdl:part name="body" element="xsd1:PaymentInfo"/>
</wsdl:message>

<wsdl:message name="BookSoldMsg">
    <wsdl:part name="body" element="xsd1:SaleConfirmation"/>
</wsdl:message>

<!-- wsdl:portType describes what one gets in return when
     sending in a particular sort of message.
     Note how we describe two operations but cannot express
     the order in which they should happen in relation to
     each other. -->
<wsdl:portType name="BookStorePortType">

    <wsdl:operation name="BuyBook">
       <wsdl:input message="tns:BuyBookMsg"/>
       <wsdl:output message="tns:SellingBookMsg"/>
    </wsdl:operation>

    <wsdl:operation name="PayForBook">
       <wsdl:input message="tns:PayingForBookMsg"/>
       <wsdl:output message="tns:BookSoldMsg"/>
    </wsdl:operation>
</wsdl:portType>

<!-- Bind the service: explain it uses SOAP. -->
<wsdl:binding name="BookstoreSoapBinding"
              type="tns:StockQuotePortType">

    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>

    <wsdl:operation name="BuyBook">
       <soap:operation
         soapAction="http://example.com/BuyBook"/>
       <wsdl:input>
           <soap:body use="literal"/>
       </wsdl:input>
       <wsdl:output>
           <soap:body use="literal"/>
       </wsdl:output>
```

```
        </wsdl:operation>

        <wsdl:operation name="PayForBook">
            <soap:operation
              soapAction="http://example.com/PayForBook"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>

    <!-- Finally, take the bookstore binding and make it a
         service. -->
    <wsdl:service name="BookstoreService">
        <wsdl:documentation>The book-selling service.</documentation>

        <wsdl:port name="BookstorePort"
                   binding="tns:BookstoreSoapBinding">

            <!-- give the binding an address -->
            <soap:address
              location="http://example.com/bookstore"/>
        </wsdl:port>
    </wsdl:service>

</wsdl:definitions>
```

Liite 2. Kirjan ostotilanteessa vaihdettavat SOAP-viestit

SOAP-pyyntö 1: Kirjan ostopyyntö

Allaoleva viesti aloittaa kirjan osto -toiminnon. Siinä ilmoitetaan halutun kirjan ISBN, tässä 951-745-197-0 (ainejärjestö Limes ry:n julkaisema Vaihtoehtoinen opintoopas 2002).

```
<SOAP–ENV:Envelope
   xmlns:SOAP–ENV="http://schemas.xmlsoap.org/soap/envelope/"
   SOAP–ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP–ENV:Body>
      <m:BuyBookMsg xmlns:m="http://example.com/bookstore.wsdl">
          <ISBN>951−745−197−0<ISBN>
      </m:BuyBookMsg>
   </SOAP–ENV:Body>
</SOAP–ENV:Envelope>
```

SOAP-vastaus 1: Kirjan hinta ja maksutavat

Palvelija palauttaa viestin, jossa ilmoitetaan kirjan saatavuushinta ja saatavilla olevat maksutavat. Yksinkertaistetussa WSDL-kuvauksessa ei ole määritelty virheilmoitusviestejä, joten oletuksena kirja on saatavilla aina jollakin hinnalla. Hinta on euroissa; tämä tieto ei myöskään välity suoraan WSDL-kuvauksesta. Saatavilla olevat maksutavat ovat postiennakko, luottokorttimaksu ja verkkopankki; käyttäjää ei ole arvioitu riittävän luotettavaksi laskulla maksamiseen.

```
<SOAP–ENV:Envelope
   xmlns:SOAP–ENV="http://schemas.xmlsoap.org/soap/envelope/"
   SOAP–ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP–ENV:Body>
      <m:SellingBookMsg xmlns:m="http://example.com/bookstore.wsdl">
          <price>24.05</price>
          <payMethodsAvailable>
             payment on delivery#credit card#Internet bank
          </payMethodsAvailable>
      </m:PriceAndPayment>
   </SOAP–ENV:Body>
</SOAP–ENV:Envelope>
```

SOAP-pyyntö 2: Maksutiedot

Toisessa pyyntöviestissä asiakas liittää jälleen mukaan kirjan ISBN:n—tämä on palvelimen vähäkontekstisuutta helpottava yksityiskohta. Lisäksi viestissä valitaan maksutavaksi luottokortti ja ilmoitetaan kortin tiedot.

```
<SOAP–ENV:Envelope
  xmlns:SOAP–ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP–ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP–ENV:Body>
    <m:PayingForBookMsg xmlns:m="http://example.com/bookstore.wsdl">
      <ISBN>951−745−197−0<ISBN>
      <method>credit card</method>
      <info>Card: Visa; Number: 999999999; Expires: 99.99.9999</info>
      <delivery>
        John Doe
        Po. Box 9
        FIN−99999 Helsinki
        Finland
      </delivery>
    </m:BuyBookMsg>
  </SOAP–ENV:Body>
</SOAP–ENV:Envelope>
```

SOAP-vastaus 2: Myynnin varmistus

Toiseen pyyntöviestiin vastataan myynnin varmistuksella. Virheenhallinta on jätetty jälleen yksinkertaisuuden nimissä pois; viesti itsessään kertoo toimenpiteen onnistuneen, mukana tulevat lisätiedot ovat lähinnä ihmisluettavaksi tarkoitettuja.

```
<SOAP–ENV:Envelope
  xmlns:SOAP–ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP–ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP–ENV:Body>
    <m:BookSoldMsg xmlns:m="http://example.com/bookstore.wsdl">
      <message>
        The book Vaihtoehtoinen opinto−opas 2002, ISBN 951−745−197−0,
        will be mailed to your address John Doe, Po. Box 9,
        99999 Helsinki, Finland. Your credit card has been charged
        for EUR 24.05. Thank you for shopping with BookStore.
      </message>
    </m:PriceAndPayment>
  </SOAP–ENV:Body>
</SOAP–ENV:Envelope>
```