YLEINEN TIETOSUOJA-ASETUS

JA OHJELMISTOARKKITEHTUURI

Pro Gradu -tutkielma
Tietojenkäsittelytiede

Kalle Hjerppe

Johannes Holvitie
Ville Leppänen
Jukka Ruohonen

Turun yliopisto
Tulevaisuuden teknologioiden laitos
Joulukuu 2017

TURUN YLIOPISTO
Tulevaisuuden teknologioiden laitos
HJERPPE, KALLE: Yleinen tietosuoja-asetus ja ohjelmistoarkkitehtuuri

Pro Gradu -tutkielma, 94 s., 4 liites.
Tietojenkäsittelytiede
Joulukuu 2017

Yleinen tietosuoja-asetus on Euroopan Unionin (EU) säätämä laki, joka tulee koskemaan kaikkea EU:n kansalaisten henkilötietojen käsittelyä. Asetusta aletaan soveltaa keväällä 2018. Yleinen käsitys on, että asetus tuo uusia vaatimuksia henkilötietoja käsitteleville tietojärjestelmille.

Tässä tutkielmassa vastataan kahteen kysymykseen: 1) Mitä yleinen tietosuoja-asetus vaatii henkilötietoja käsitteleviltä tietojärjestelmiltä? 2) Millainen ohjelmistoarkkitehtuuri kattaisi nämä vaatimukset tehokkaasti? Kysymyksiä tarkastellaan tietyn ohjelmistoja kehittävän tapausyrityksen kontekstista. Konteksti määritellään tutkielman aluksi.

Tietosuoja-asetuksen vaatimukset selvitetään käymällä systemaattisesti lakiteksti läpi, johtaen siitä lopulta vaatimusmäärittelyn. Vaatimusmäärittely kattaa tekniset vaatimukset, joihin ohjelmistoarkkitehtuurin tulee ottaa kantaa. Työssä löydettiin yhdeksän erillistä kokonaisuutta vaatimuksiksi.

Toiseen tutkimuskysymykseen vastataan suunnittelemalla vaatimusmäärittelyn kattava esimerkkiarkkitehtuuri. Suunnittelussa hyödynnettiin aiempaa tutkimusta aiheesta ja tapausyrityksen ominaisuuksia.

Ratkaisu koostuu kuudesta tietosuojamoduulista, jotka kattavat uudelleenkäytettävästi suurimman työn asetuksen vaatimuksista. Ne ovat ydinhenkilötietomoduuli, suostumuskeskus, lokitusarkkitehtuuri, oikeuksien hallinta, pseudonymisointiprosessi ja `@PersonalData`-annotaatioprosessori. Kukin esitellään tarkemmin tutkielmassa.

Lopputuloksena löydettiin vähintään tapausyritykselle sopiva arkkitehtuuri. Työssä tehdyn arvioinnin mukaan se täyttää tietosuoja-asetuksen. Komponentteja voidaan käyttää muissakin konteksteissa. Tutkielmassa havaittiin myös mahdollisuuksia myöhemmälle työlle.

Asiasanat: *tietosuoja*, *GDPR*, *ohjelmistoarkkitehtuuri*, *suostumus*, *pseudonymisointi*

UNIVERSITY OF TURKU
Department of Future Technologies
HJERPPE, KALLE: General Data Protection Regulation and Software Architecture

Master's thesis, 94 p., 4 app. p.
Computer science
December 2017

The General Data Protection Regulation is a law issued by the European Union (EU). Processing of personal data of the citizens of EU will be within its scope. The regulation shall apply from May 2018. The common consensus is that the regulation will introduce new requirements to information systems that handle personal data.

This thesis will answer two questions: 1) What are the requirements of the regulation for information systems? 2) What kind of a software architecture would account for them efficiently? The research questions are considered from the point of view of a chosen case company. The context will be defined at the beginning of the thesis.

The requirements of the data protection regulation will be defined by systematically parsing the law text, resulting in a requirements specification. The scope of the specification will be the technical requirements for a software architecture. Nine such requirements are found in the thesis.

The second research question is answered by designing an example software architecture that matches the requirements. This work is based on previous research on the subject and the attributes of the case company.

The solution presented in the thesis consists of six reusable data protection modules, which tackle different requirements. They are a core personal data module, a consent center, a logging-architecture, a rights management module, a pseudonymization process, and `@PersonalData` annotation processor. Each module is examined closer in the thesis.

In conclusion, the study finds a suitable software architecture, which fulfills the requirements of the regulation. This applies at the very least for the case company. In addition, the components are reusable in other contexts. Options for further research were also noted in the thesis.

Keywords: *data protection*, *GDPR*, *software architecture*, *consent*, *pseudonymisation*

Sisällysluettelo

Luku 1
Johdanto

Tässä tutkielmassa käsitellään ohjelmistoarkkitehtuuria ja tietosuojaa Euroopan unionin (EU) vuonna 2018 voimaantulevan tietosuoja-asetuksen [40] kannalta. Arkkitehtuuria tarkastellaan tapaustutkimuksena luvussa kaksi esiteltävän esimerkkiyrityksen kontekstissa. Tärkein käsiteltävä kohderyhmä on henkilötietoja rekisteröivät verkkosovellukset ja niiden taustajärjestelmät. Työ rajataan eurooppalaisten henkilötietoja käsitteleviin järjestelmiin. Tässä luvussa esitellään tutkimuksen rakenne, aihe, termejä ja työn tekemiseen johtanutta taustaa.

Aihe on ajankohtainen, koska lähes kaikki eurooppalaisten henkilöiden tietoja käsittelevät yritykset joutuvat jotenkin reagoimaan uuteen asetukseen. Yleinen tietosuoja-asetus (Asetus (EU) 2016/679) annettiin Euroopan parlamentin toimesta 27.4.2016 ja sitä aletaan soveltaa 25.5.2018 alkaen. Tietosuoja-asetus on tunnettu sen englanninkielisellä nimellä General Data Protection Regulation (GDPR). Asetuksen sisältöön perehdytään tämän työn luvussa kolme. Euroopan tietosuojalainsäädäntö kuitenkin lähtökohtaisesti sen myötä uudistuu. Voimme odottaa tiukempia vaatimuksia henkilötietorekisterien pitäjille, mitkä pitää huomioida tietojärjestelmiä kehittettäessä.

Tutkielman tavoitteena on muuntaa tietosuoja-asetuksen velvoitteet *vaatimusmäärittelyksi* valittavaan kontekstiin sopivalle ohjelmistoarkkitehtuurille. Työssä tarkastellaan muita aiempia tutkimuksia GDPR-yhteensopivista ohjelmistoarkkitehtuureista. Lopulta suunnitellaan valittuun ympäristöön sopiva arkkitehtuuri aiempaa tutkimusta hyväksi käyttäen niin, että tehdyn vaatimusmäärittelyn kohdat täyttyvät. Suunniteltua ratkaisua analysoidaan vertailtuna muihin mahdollisiin lähestymistapoihin. Lopputyön käytännön osuutena tehdään osasta tutkimuksen tuloksia *Proof of concept* -tyyppinen esimerkkitoteutus.

Tämän työn kontekstissa käsite *tietosuoja* on luonnollisen henkilön suojelua tämän henkilötietojen käsittelyn yhteydessä. *Henkilötiedon* määritelmään palataan luvussa kolme, jossa tarkastellaan, miten tietosuoja-asetus sen rajaa. Tietosuoja on Euroopan unionin

mukaan perusoikeus. EU:n perusoikeuskirjan [47] kahdeksannen artiklan kohdassa yksi todetaan seuraavaa: ”Jokaisella on oikeus henkilötietojensa suojaan”. Tähän perustuen Euroopan unionin parlamentti antoi uuden asetuksen. Huomion arvoista on myös se, että Suomen perustuslakikin [36] takaa yksityiselämän suojan. Pykälässä 7 todetaan, että ”jokaisen yksityiselämä, kunnia ja kotirauha on turvattu”, ja että ”kirjeen, puhelun ja muun luottamuksellisen viestin salaisuus on loukkaamaton”. Westin esittää *yksityisyyden* käsitteen olevan yksilöiden, ryhmien tai instituutioiden vaatimuksena siitä, koska, miten ja kuinka paljon informaatiota niistä kommunikoidaan muille [49]. Näihin määritelmiin viitataan kun tutkielmassa puhutaan edellä mainituista käsitteistä.

Tietosuojan tärkeydelle on siis perustavanlaatuinen kannatus lainsäätäjien puolelta, mikä voidaan tehdä johtopäätöksenä mainituista lakiteksteistä. Herääkin kysymys, miksi uusi tietosuoja-asetus on tarpeellinen? GDPR korvaa aiemman EU:n kannanoton tietosuojaan. Direktiivi 95/46/EY yksilöiden suojelusta henkilötietojen käsittelyssä ja näiden tietojen vapaasta liikkuvuudesta [39] kumotaan uudessa tietosuoja-asetuksessa, ja kaikki viittaukset siihen viittaavat jatkossa tietosuoja-asetukseen. EU perustelee aiemman direktiivin riittämättömyyttä uuden asetuksen pohjustuksessa viitaten erityisesti tietosuojan täytäntöönpanon hajanaisuuteen ympäri unionia. *Direktiivin* ja *asetuksen* ero EU:n parlamentin konteksissa onkin sen sovellettavuudessa. Direktiivi on ohje jäsenvaltioille, jonka mukaan niiden on laadittava oma lakinsa, kun taas asetus tulee voimaan sellaisenaan kaikissa jäsenvaltioissa [10].

Tarkastellaan seuraavaksi tietosuojan vaikutuksia ja aiemmin mainittujen lakien tarpeellisuutta. Henkilötietojen kerääminen ja jakaminen ovat nousseet dramaattisesti vuodesta 1995, jolloin aiempi tietosuojadirektiivi julkaistiin, nykyhetkeen, tai siihen aikaan kun yleistä tietosuoja-asetusta alettiin valmistella. Teknologia mahdollistaa tiedon keräämisen ja käsittelyn aivan eri mittakaavalla kuin aiemmin. Sosiaalisen median palvelut ovat hyvä esimerkki vapaaehtoisesta tiedon luovuttamisesta. EU:n selvityksessä GDPR:n vaikutuksista [31] mainitaan myös, että yksityishenkilöt saattavat henkilötietoja julki yhä enemmän, ei välttämättä seuraamuksia ymmärtäen. Tämä johtaisi luottamuksen heikentymiseen digitaalisia palveluja kohtaan ja olisi esteenä ekonomiselle kasvulle. Myös julkisen sektorin verkkopalvelut vaativat luottamusta käyttäjiltään. Yleinen luottamuspula haittaisi siten niitäkin.

Eurobarometri 359 [30] on kyselytutkimus, joka julkaistiin vuonna 2011. Sen tulokset antavat hyvän kuvan tilanteesta, jossa yleistä tietosuoja-asetusta valmisteltiin vuoden 2012 alussa. Kyselyssä haastateltiin eurooppalaisia yli 15-vuotiaita heidän asenteistaan tietosuojaa kohtaan. Tulokset tukevat väitettä siitä, että luottamus EU:n tietosuojaan on vaakalaudalla. Kolme neljäsosaa eurooppalaisista kokee henkilökohtaisten tietojen luovuttamisen olevan kasvavassa osassa nykyaikaista elämää, noin puolelta on kysytty verkkopalvelussa enemmän henkilötietoja kuin tarpeen, ja yli puolet ovat huolissaan seurannasta

maksukorttien kautta.

Yrityksille, joiden liiketoiminta sisältää henkilötietojen käsittelyä, ei tietosuojan vaaliminen kuitenkaan ole itsestäänselvää. Tietosuojan valvominen on kuluttajan kannalta vaikeaa, koska omien henkilötietojen siirtyminen yrityksestä toiseen ei ole seurattavissa. Yritykset saavat taloudellista hyötyä henkilötietojen käytöstä, mikä luonnollisesti on ristiriidassa käsittelyn rajoittamisen kanssa. Lisäksi globaaleja palveluja tarjoavat yritykset eivät aina huomioi Euroopan lainsäädäntöä. Tästä on esimerkkinä vuonna 2015 julkaistu tapaus [48], jossa Belgian yksityisyyskomissio totesi erään tunnetun sosiaalisen median päivitetyt käyttöehdot EU:n säädösten vastaiseksi. Uuden tietosuoja-asetuksen myötä EU on varautunut rikkomuksiin pelotevaikutuksella: hallinnollisten sakkojen maksimisuuruus on jopa 20 miljoonaa euroa, tai 4% yrityksen liikevaihdosta (suurempi vaihtoehdoista). Koska asetusta aletaan soveltaa vasta vuonna 2018, ei vielä voida ennustaa, kuinka ankarasti sakkoja tullaan tuomitsemaan.

Tämä johtaakin tutkielman tavoitteeseen: suunnitella sopiva ohjelmistoarkkitehtuuri, joka kattaa tietosuoja-asetuksen vaatimuksen henkilötietojen käsittelystä. Suunnittelussa otetaan huomioon liiketoiminnalliset vaatimukset tietosuojavaatimusten ohella. Työssä tarkastellaan esimerkkiyritystä, joka esitellään seuraavassa luvussa. Tavoitteena ei ole luoda yleismaailmallista ratkaisua, koska konteksti on tärkeä osa arkkitehtuurin suunnittelua – ja yksityiskohdat vaikuttavat oleellisesti lopputulokseen. Tärkeää on kuitenkin löytää ja eristää yleistettävissä olevia kohtia, ja pyrkiä arvioimaan arkkitehtuureja mahdollisimman objektiivisella tavalla. Tieteellistä kontribuutiota tutkielmassa on tavoitteena tehdä seuraavin osin: vaatimusmäärittely tietosuoja-asetuksesta, yleisen tason tietosuoja-arkkitehtuuri, ja tehdä ratkaisuja yksittäisiin vaatimuksiin.

Tässä johdannossa esiteltiin tutkielman aihe, sen rakenne ja sen rajaukset korkealla tasolla. Aiheeseen liittyviä tärkeimpiä käsitteitä määriteltiin. Käytiin läpi työn taustaa, eli Euroopan unionin lakimuutos. Esiteltiin perusteita ja motivaatiota työlle. Yleinen tietosuoja-asetus käsiteltiin vielä vasta pintapuolisesti, joten tutkielman haaste selkenee tarkemmin myöhemmissä luvuissa. Luvussa kaksi esitellään työhön valittua tapausyritystä, jonka tarpeiden mukaan arkkitehtuuri suunnitellaan. Tästä saamme rajauksia ja vaatimuksia tutkielmalle. Luvussa kolme käydään läpi yleinen tietosuoja-asetus ja muodostetaan sen datasta vaatimusmäärittely. Luvussa neljä tarkastellaan aiempaa tutkimusta työn aiheesta, johon pohjaudutaan arkkitehtuuria suunnitellessa. Luvussa viisi käydään läpi tutkielman oma kontribuutio: suunniteltu arkkitehtuuri yleisellä tasolla ja eri moduulien esittely. Luvussa kuusi analysoidaan suunniteltua arkkitehtuuria, pyrkien validoimaan tehdyt ratkaisut. Luku seitsemän on koko työn yhteenveto ja esittää tehdyt johtopäätökset.

Luku 2
Taustan esittely

Tutkielman tutkimuskohteena olevan esimerkkiyrityksen ympäristön ja kontekstin esittely. Rajauksia valitun yrityksen mukaan. Näistä johdetaan ominaisuuksia, joiden kautta yleisen tietosuoja-asetuksen vaatimuksia tarkastellaan.

Yksi tutkielman tärkeimmistä tavoitteista on suunnitella yleistä tietosuoja-asetusta vastaava ohjelmistoarkkitehtuuri. Arkkitehtuuria ei ole mielekästä suunnitella tyhjiössä, koska vaikeimmat kysymykset tulevat varsinaisen liiketoimintalogiikan lisäksi ympäristön aiheuttamista vaatimuksista ja niiden ominaisuuksien välisistä vaihtokaupoista. Tämän vuoksi tutkielmalle pitää määrittää konteksti, jossa aihetta tarkastellaan. Tämän kontekstin rajaa luvussa esiteltävä tapausyritys.

Tarkoitus ei ole ottaa kantaa siihen, millaista liiketoimintaa järjestelmän puitteissa tehdään, vaan määritellä rajat, joihin sopivat yritykset voisivat hyötyä tutkimuksen arkkitehtuurista. Tässä luvussa esitellään tapausyritystä ja arvioidaan sen luonnetta ominaisuuksin. Tavoitteena on löytää arkkitehtuuriympäristön ominaisuuksia ja tietosuoja-asetuksen ulkopuolisia vaatimuksia. Myöhemmin näihin viitataan GDPR:n vaatimuksia läpikäydessä, sovittaen lakitekstiä ja ominaisuuksia yhteen, jotta voidaan tehdä valintoja lopullista arkkitehtuurin määrittelyä varten. Luku on pohjustava, taustoja keräävä osio.

Luvussa käytetään löydetyistä arkkitehtuurin ominaisuuksista notaatiota *(ON: ominaisuus)*, vaikka esimerkiksi *(O5: uudelleenkäytettävyys)* viittaa viidenteen ominaisuuteen esittelyjärjestyksessä.

2.1 Tutkielman tapausyritys

Esimerkkiyrityksen valinta tietystä luokasta ei ole itseisarvo tutkielmalle; koska kaikki (Euroopassa toimivat) yritykset joutuvat kuitenkin noudattamaan tietosuoja-asetusta, jokaiselle olisi arvoa tilanteen tutkimisesta. Kuitenkaan ei ole mielekästä yrittää löytää niin kutsuttua *"Silver bullet"* -ratkaisua [4], vaan suunnitella optimaaliset ratkaisut kullekin yritykselle samalla etsien yleistettäviä kohtia. Yksi mahdollinen tutkimiskulma on vertailla, miten eri kokoiset yritykset joutuvat suhtautumaan yleiseen tietosuoja-asetukseen,

mutta se ei kuulu tämän tutkielman piiriin. Työhön voidaan valita siis mikä tahansa yritys, joten on perusteltua käyttää yritystä, johon on mahdollisimman hyvä näkyvyys – päädytään siis tarkastelemaan työnantajani kaltaisia ohjelmistoyrityksiä. Tutkielman esimerkkiyrityksen toimintaympäristöä kuvailevat ominaisuudet perustuvatkin useassa ohjelmistoalan yrityksessä tapahtuneeseen havainnointiin vuosien 2014-2017 ajalta.

Tyypillisellä tutkielman rajaukseen sopivalla yrityksellä on työntekijöitä muutamasta kymmenestä pariin sataan. Liikevaihtoa on miljoonasta kymmeniin miljooniin. Tärkeimpinä havaintokohteina ovat *Geniem Oy* ja *Solita Oy*. Yritykset poikkeavat toisistaan koon suhteen, mutta pohjimmiltaan liiketoiminta on samanlaista. Varsinaisia yrityksiä tärkeämpää tutkielmassa on niiden toimintatavat, asiakkuudet ja ympäristö, jossa arkkitehtuuria luodaan. Yleistettäviä kohtia on kuitenkin helpompi löytää samankaltaisille yrityksille.

Käydään seuraavaksi läpi, millaisessa kontekstissa tapausyrityksessä rakennetaan ohjelmistoja jaotellen hieman ohjelmistoyritysten joukkoa. Yrityksellä on tietty *asiakasprojekteja* toteuttava liiketoimintamalli, joka määritellään seuraavassa aliluvussa. Mallilla on vaikutus tutkielmassa esitettävän ratkaisun arvioitaviin ominaisuuksiin.

2.2 Asiakasprojektimalli

Yrityksen liiketoimintamalli on toteuttaa asiakasprojekteja. Tähän perustuvasta toiminnasta tulee omat vaikutuksensa suunniteltavaan arkkitehtuuriin, mihin perehdytään vielä tässä luvussa. Liiketoiminnassa toteutetaan tietojärjestelmiä tilaustyönä; asiakas ostaa yritykseltä ohjelmistokehittäjien aikaa ja käyttää sitä liiketoimintansa kehittämiseen. Uusien järjestelmien kehittämisen ohella vanhoja järjestelmiä ylläpidetään ja jatkokehitetään. Liiketoimintaan kuuluu myös palvelimien ja muiden ajoympäristöjen hallinta ja ylläpito.

Tutkielman kontekstissa *ohjelmistokehitysprojektilla* (lyhyesti *projektilla*) tarkoitetaan väliaikaista pyrkimystä, johon ryhdytään tietyn tuotteen, palvelun tai tuloksen saamiseksi [35], kuten Nokes ja Kelly ovat projektien määritelmiä koostaneet vuonna 2007. Projekteihin liittyy tunnettu alkamis- ja loppumisajankohta, ja yleisesti käsitteeseen liittyy vahvasti rajoitteiden tasapainottelu. Pyritään tiettyyn tulokseen ajan, hinnan ja resurssien puitteissa. Käytännössä projekteihin mukaan voidaan laskea ylläpito ja jatkotyöt varsinaisen projektin loputtua, jolloin määritelmää hieman venytetään. Tutkielman esimerkkiyrityksen projekteissa käytetään mahdollisimman ketteriä ohjelmistotuotantotapoja, vaikka eri asiakkaiden ja projektien välillä on eroja.

Lähtökohtaisesti asiakasprojekteja on kahdenlaisia: kokonaan uusia asiakkuuksia ja aiempien asiakkaiden uusia tilauksia. Vanhojen ja uusien asiakkaiden kohdalla on ero siinä, että jo olemassa olevat järjestelmät vaikuttavat arkkitehtuuriin. Mikäli projekti on vain vanhan järjestelmän laajentamista, suurimmat arkkitehtuurikysymykset ovat jo ratkaistu

(tai joudutaan tekemään suuria muutoksia). Tällöin taustajärjestelmät olisivat jo olemassa (kuten mahdollinen tietosuojapalvelu). Uusilla asiakkuuksillakin on usein jo olemassa olevia järjestelmiä, mikä osin yhdistää luokkien erottelua. Käytetään tästä havainnosta nimeä *(O1: jako asiakkuuksien välillä)*. Ominaisuuden mukaan esimerkiksi arkkitehtuurin tietomalli ei voi olla sidottu yhteen henkilötietorekisteriin.

Vanhojen järjestelmien jatkokehityksen lisäksi asiakkaan tarpeet voivat muutenkin vaikuttaa teknologia- ja alustavalintoihin. Tällöin aiheutuu kustannuksia esimerkiksi, jos yrityksen suunniteltu tietosuoja-arkkitehtuuri on näiden vaatimusten kanssa ristiriidassa. Otetaan tämä ominaisuus tarkasteluun nimellä *(O2: teknologiasidonnaisuus)*. Teoriassa uusia järjestelmiä voidaan tuottaa sopivimmalla teknologialla, mutta monesti asiakkaan vaatimukset ja teknologian myyvyys rajaavat tätä. Myyvyydellä tarkoitan sitä, että tarjouskilpailun voi voittaa kahdesta eri ratkaisusta asiakkaan paremmaksi kokema, vaikkei teknisiin syihin perustuvaa eroa olisi. Toisin sanoen arkkitehtuuria arvioidessa on tiedostettava tiettyyn teknologiaan sitoutumisen hinta.

Toinen liiketoiminnallinen projektin ominaisuus on projektin koko. Tätä mitataan projektikohtaisesti sekä käytettävissä olevan rahan, että ajan suhteen. Molemmat vaatimukset asettavat rajoitteita projektin arkkitehtuurille. Nostetaan projektin koosta tutkielmassa arvioitaviksi ominaisuuksiksi *(O3: projektin koon vaihtelu)*. Kooltaan työn puitteissa olevat projektit vaihtelevat pienistä (muutaman tuhannen euron) melko suuriin (satojatuhansia euroja vuodessa); aivan pienimpiä eikä maailmanlaajuisen kokoisia projekteja ei pidetä työn fokuksessa. Ajallisesti projektit voivat kestää noin kuukaudesta moneen vuoteen. Pienimissä projekteissa – varsinkin sellaisissa, jotka eivät ole lisätyötä jo olemassaoleville asiakkuuksille – tulevat liiketoiminnalliset seikat todella suureen rooliin suunnittelussa. Suurimmat järjestelmät puolestaan vaativat eri mittakaavassa olevan arkkitehtuurin hajautuksineen suurten datamäärien käsittelyyn ja globaalisti levitettyjen järjestelmien vuoksi. Tähänkään ei ole tutkimuksen puitteissa resursseja ryhtyä, mutta työn nostaminen suurempaan skaalaan on yksi mahdollinen jatkon aihe. Rajauksen ääripäihin sijoittuviin projekteihin voidaan työn tuloksista soveltaa ajatuksia, mutta niiden omat haasteensa muuttaisivat työn laajuutta merkittävästi.

Asiakasprojektimallissa tehtävä liiketoiminta tuo omia ominaisuuksiaan harkittavaksi ohjelmistoarkkitehtuurin kannalta. Kukin asiakas omistaa heille luodut järjestelmät, ja asiakkaat vaihtuvat. Kuitenkin asiakkaille voidaan kehittää monta eri järjestelmää, joilla on yhteisiä toiminnallisuuksia. Tästä seuraa jaottelu, jossa on harkittava ratkaisuja kahdesta eri kontekstista: oman yrityksen ja asiakkaan. Luonnollisesti voidaan olettaa olevan kestävää liiketoimintaa tehdä hyviä ratkaisuja asiakkaan kannalta, mutta kysymyksissä on myös oman yrityksen näkökulma. Parhaassa tapauksessa voidaan suunnitella arkkitehtuureja, jotka ovat sekä tehokkaimpia oman yrityksen kannalta, että palvelevat asiakkaan

tarpeita. Hyvin tuotteistetut ylätason arkkitehtuuriratkaisut voivat myös olla myyntivaltti uusien asiakkuuksien hankinnassa.

Jaottelusta ilmenee kaksi arkkitehtuurin tarkastelun tasoa, joissa alemmalla tarkastellaan yksittäisen asiakkaan hallussa olevia järjestelmiä, ja ylemmällä metatasolla kokonaiskuvaa kaikkien (nykyisten ja tulevien) asiakkuuksien ylitse. Merkitään tätä ominaisuutta nimellä *(O4: projektien ulkopuoliset investoinnit)*. Yrityksen projektien ulkopuolella tapahtuva kehitys pienentää tietyn yksittäisen projektin kustannuksia. Kehitykseen käytetty hinta täytyy luonnollisesti kattaa projektien tasolla, mutta toiston väheneminen ja synergia voisi johtaa myös projektien budjetin paranemiseen yrityskehityksellä.

On huomioimisen arvoista, että asiakasprojektimallissa asiakkaalle rakennettujen järjestelmien (lähdekoodien) tekijänoikeudet luovutetaan usein asiakkaalle. Tällaisissa tilanteissa heille kuuluu kaikki, mitä luodaan ajalla, josta asiakasta veloitetaan. Tekijänoikeuskysymykset ja lisenssit ovat toki sopimuksellisia seikkoja, jotka ovat neuvoteltavissa, ja siksi eivät tärkeimmässä fokuksessa työn puitteissa. Asiakkuuksien välillä yhteisten komponenttien käyttö ja rakentaminen vaatii siis myös strategiaa yrityksen tasolla. *(O4: projektien ulkopuoliset investoinnit)* on tässä roolissa siten, että kehitys, jota ei tehdä asiakkaan nimissä, on helpompaa ottaa muihin projekteihin käyttöön. Yksi vaihtoehto on tehdä avoimen lähdekoodin kehys, jota sovelletaan asiakasprojekteihin.

Meillä on nyt kuva yrityksen toimintamallista ja olemme esitelleet asiakasprojektin käsitteen. Yrityksen ja asiakasprojektien ominaisuuksista nostimme esiin *(O1: jaon asiakkuuksien välillä)*, *(O2: teknologiasidonnaisuuden)*, *(O3: projektin koon vaihtelun)*, sekä *(O4: projektien ulkopuoliset investoinnit)*. Kaikki toimivat laadullisina kriteereinä, kun arvioimme arkkitehtuuriratkaisuja. Liiketoiminnalliset seikat eivät kuitenkaan johda kovin konkreettisiin rajauksiin, joten seuraavaksi tarkastelemme teknistä puolta.

2.3 Tekninen ympäristö

Tässä alaluvussa perehdytään tarkemmin työhön valitun yrityksen tekniseen ympäristöön, sen arkkitehtuuriin ja prosesseihin. *Teknisellä ympäristöllä* tutkielmassa tarkoitetaan projektien puhtaan ohjelmistoarkkitehtuurin lisäksi kaikkea yrityksen ohjelmistotuotantoon liittyvää seikkaa. Osion tarkoitus on selventää, millaiseen kontekstiin työssä suunnitellaan ratkaisua tietosuojalle. Käydään käytännön tasolla läpi tapausyrityksen projektien jakautumista asiakkuuksien ja järjestelmien välillä, ja arkkitehtuurikuvauksia valituista esimerkkiprojekteista. Tarkasteltavista käsitteistä kerätään ominaisuuksia, jotka käyvät pohjaksi työssä suunniteltavalle arkkitehtuurille. Jotta tähän päästään, pitää myös määritellä joitain arkkitehtuurin konsepteja.

Nykyinen yrityksen tekninen ympäristö sisältää monia eri järjestelmiä eri alustoilla, mikä on otettava huomioon tietosuoja-arkkitehtuurissa. Yksittäisen projektin tuotoksena

syntyy useampi palvelu, jotka tyyppillisesti sijaitsevat fyysisesti eri palvelimilla. Tuotantopalvelimien lisäksi sama järjestelmä on kopioituna *staging*-ympäristöksi. Nimityksellä tarkoitetaan tuotantoa vastaavaa sovellusympäristöä, jolla voidaan tehdä viimeiset testit ennen uutta julkaisua. Tarkastellaan ympäristöjä tarkemmin edellä. Lisäksi usein on erillisiä kehityspalvelimia. Yleisölle avoimet verkkopalvelut käyttävät lisäksi välimuistipalvelimia ja sovelluspalvelimetkin voivat olla hajautettuja. Tietokannat ovat usein omilla palvelimillaan. Vähintään staging-ympäristössä tarvitsisi olla käytössä tuotantoa vastaava data. Tämä vaatimus arkkitehtuurille voi aiheuttaa ristiriitaa tietosuojan kannalta ja on ratkaistava tutkielmassa. Nimetään se *(O5: staging-ympäristön vaatimukset)*.

Tietosuoja on huomioitu aiemmissa projekteissa vaihtelevasti. Vaikka vanhat järjestelmät noudattavatkin nykyistä lainsäädäntöä, yhtenäistä ratkaisua tietosuojalle ei ole. Henkilötietoja on tallennettu järjestelmiin ja niiden käsittelyä on sisäänrakennettu niihin. Käytössä on monenlaisia eri teknologioita ja alustoja. Jo olemassa olevien järjestelmien tulee kuitenkin ottaa huomioon uusi tietosuoja-asetus. Useimpiin järjestelmiin asetus tulee vaatimaan jotain muutoksia, ja niiden minimoiminen on yksi tärkeä kriteeri. Tutkielmassa suunniteltavassa arkkitehtuurissa on siis otettava huomioon aiempien järjestelmien vaikutukset ja se, miten uusi ratkaisu otetaan käyttöön vanhoihin projekteihin. Kullakin yrityksellä on kuitenkin äärellinen määrä aiempia projekteja, ja voi olla vaihtoehto suunnitella niihin räätälöidyt muutokset yleisen tietosuoja-asetuksen myötä. Mitä vähemmän muutoksia aiempiin järjestelmiin tullaan tarvitsemaan, sitä edullisemmaksi käyttöönotto tulee. Toisaalta uuden arkkitehtuurin kehittäminen maksaa. Tästä tulee yksi arvioitava kriteeri uudelle suunnitelmalle: tasapaino uuden rakentamisen kustannuksien ja vanhojen järjestelmien muutostöiden välillä. Merkitään tätä ominaisuudella *(O6: vanhojen järjestelmien päivittäminen)*.

Ylemmällä tasolla oleva arkkitehtuurisuunnittelu koostuu *infrastruktuurista* ja uudelleenkäytettävistä *moduuleista*. Infrastruktuurilla tarkoitetaan työssä ohjelmistoja ja järjestelmiä, jotka edesauttavat asiakasprojektien toteuttamista. Nämä ovat siis aiemmin mainitulla ylemmällä tasolla arkkitehtuurin suunnittelussa, koska infrastruktuurijärjestelmiä voidaan hyödyntää eri projektien ja asiakkuuksien välillä. Moduulit puolestaan ovat kokonaisia palveluita, komponentteja, tai kirjastoja, joita on tarkoitus uudelleenkäyttää sellaisenaan. Moduulien ja infrastruktuurin väli ei ole täsmällinen, vaan tutkielman kannalta suurin ero on se, millä arkkitehtuurin tasolla niitä tarkastellaan. Moduulien käyttäminen ja määrittely on yksittäisen projektin tarpeisiin yksittäisen projektin tasolla, mutta sen nostaminen yrityksen laajuiseksi on strateginen päätös.

Kun tutkielmassa arvioidaan infrastruktuurien tai moduulien laatua, viitataan ominaisuuteen *(O7: uudelleenkäytettävyys)* tarkasteltaessa sitä, kuinka suuri työ tiettyä ratkaisua on soveltaa useaan eri asiakasprojektiin. Ominaisuuden tavoite on päästä mahdollisimman helpolla aina uuden sovelluskohteen tapauksessa. Skaalaa on molempiin suuntiin:

huono ratkaisu vaatisi koko tietosuoja-asetuksen uudelleenkäsittelyn joka kerta, kun taas täydellinen ratkaisu ei vaatisi lainkaan työtä. Realistinen vaihtoehto on arkkitehtuuri, jossa on tehty pohjatyötä henkilötietojen käsittelyn suhteen, mutta joka vaatii projektikohtaisia kustomointeja.

Parhaimmassa tapauksessa infrastruktuuriohjelmistojen kehittäminen on yritykselle investointi, joka maksaa itsensä takaisin nopeampana kehityksenä asiakasprojekteille. Infrastruktuuriohjelmistoiksi lasketaan kaikki käyttöjärjestelmistä sähköpostiohjelmiin, mutta tutkielman kannalta tärkempiä ovat ohjelmistokehittäjille tehdyt *DevOps*-järjestelmät [3]. Näillä muodostetaan esimerkiksi automatisoituja toimitus- ja asennusputkia. Clements ja Northrop käsittelevät kokonaisia ohjelmistotuotantolinjoja (Engl. *sofware product lines*) kirjassaan [6] 2002, joiden tarkoitus on rakentaa perinteisiä tuotantolinjoja vastaavia putkia ohjelmistotuotantoon. Linjan elementit ovat yrityksen voimavaroja, joita uudelleenkäyttämällä saadaan nopeampi *julkaisunopeus*, parempi laatu ja tuotteliaisuus. Infrastruktuurilla voi ratkaista tietosuojakysymyksiä vain osittain, koska suora liiketoimintalogiikka ei kuulu niiden piiriin. Ratkaistavia alueita voisi olla massadatan siirrot, muunnokset ja varmistukset. Infrastruktuuriohjelmistot voivat hoitaa myös loppukäyttäjien (henkilötietorekisterissä olevien) suuntaan kommunikoinnin ja tietosuojapyyntöjen hallinnan. Näihin palataan myöhemmissä luvuissa.

Moduulien tarkoitus on olla uudelleenkäytettäviä eri projektien välillä mahdollisimman pienellä työllä. Moduuli ratkaisee tietyn ongelman ja tarjoaa ratkaisua palveluna muulle arkkitehtuurille. Tämä voi tapahtua kaikkilla arkkitehtuurin tasoilla: moduuleita on kirjastoista itsenäisin palveluihin. Esimerkkeinä moduuleista voidaan esitellä kokonainen dokumentinhallintajärjestelmä, joka toimii verkkopalveluna muiden komponenttien käytettävissä, tai kirjasto joka tarjoaa tiedostojen käsittelylle rajapinnan. *Yhden vastuun periaatteen* (Martin [34]) mukaan parhaat moduulit ratkaisevat tasan yhden ongelman. Tällöin niillä on vain yksi syy muuttua, ja moduulista riippuvat järjestelmät eivät joudu ottamaan käyttöönsä ylimääräisiä riippuvuuksia asioihin, joita ne eivät välttämättä tarvitse. Moduuli voisi työn puitteissa esimerkiksi tarjota henkilötietojen käsittelyä palveluna.

Ohjelmistokehykset (Engl. *software framework*) sijoittuvat infrastruktuurin ja moduulien välille. Riehle määritteli vuonna 2000 [42] ohjelmistokehyksen olevan uudelleenkäytettävä suunnitelma implementaation kanssa: suunnitelma mallintaa sovelluksen alan, kun taas sen implementaatio määrittää sen, miten mallia käytetään. Ohjelmistokehyksien päälle rakennetaan projektien järjestelmiä, ja niillä käsitellään varsinaista liiketoimintalogiikkaa. Tämän vuoksi ne eivät sovi infrastruktuurin joukkoon. Toisaalta ohjelmistokehyksiä käyttävät toteutukset eivät ole täysin uudelleenkäytettäviä vaan vaativat kustomointia pohjan päälle, suuremmassa määrin kuin moduulit yleisesti. Ohjelmistokehysten valinta

ja luominen ovat siis suuria arkkitehtuurisia päätöksiä, mutta sellaiseen voisi suunnitella valmiiksi tietosuojaa.

2.3.1 Toimintakäytännöt

Tärkeässä osassa tietosuojaa arvioidessa ovat organisaation toimintakäytännöt. Tässä aliluvussa pyritään saamaan kuva siitä, miten tapausyrityksessä toimitaan ohjelmiston kehityksessä, järjestelmien käyttötuessa, ja erityisesti henkilötietoja käsitellessä. Myöhemmin voimme verrata näitä toimintamalleja siihen, mitä yleinen tietosuoja-asetus tulee vaatimaan.

Projekti alkaa siitä, kun sille valitaan projektitiimi. Tämä on projektin koosta riippuen noin kolmesta kymmeneen ihmistä, vaihtelevin työpanoksin ja roolein. Tiimi tekee projektin palvelumuotoilusta ohjelmiontiin, kommunikoiden työn aikana asiakkaan ja muiden sidosryhmien kanssa. Kehitys on tyypillisesti iteratiivista ja inkrementaalista, noudattaen ketteriä periaatteita. Erilaisia toimituksia tehdään usein projektin myötä. Tällaista prosessia, jossa tiimit on koostuvat monialaisista asiantuntijoista, ja jotka tuottavat ohjelmistoja *Lean*-periaatteiden mukaan kuvaavat Gothelf ja Seiden kirjassaan 2013 [15].

Ohjelmistokehityksen lisäksi tietosuojan kannalta tärkeitä osia toimintakäytännöistä ovat ylläpito ja palvelinympäristöjen hallinta. Yksinkertaistaen, voidaan kuvitella projektin ohjelmistokehityksen tuloksena syntyvän jokin artefakti: kutsutaan tätä *ohjelmaksi*. Ohjelmaa ajetaan tietyssä paikassa – *palvelimella*. Ohjelma käsittelee dataa, joka pysyväistallennetaan *tietokantaan*. Käytännössä asiat ovat monisyisempiä, mutta näiden kolmen komponentin pääsynhallintaa, suhteita ja datavirtoja tarkastelemalla saamme pohjan arvioida tietosuojaa. Kutsutaan tätä kolmikkoa tässä kontekstissa nimellä *ympäristö*.

Ympäristön kokonaisuus vielä tavanomaisesti kopioidaan vähintään kolmelle eri toisistaan käsitteellisesti erotellulle tasolle: kehitys-, staging ja tuotantoympäristöksi. Kehitysympäristöön voidaan jatkuvan integroinnin (Engl. *Continuous Integration*) periaatteen [9] mukaisesti tehdä muutoksia vapaasti ja usein. Tällöin tiimi voi toteuttaa eri ominaisuuksia rinnan, kun usein yhdistettävät kehityshaarat eivät pääse eroamaan paljoa toisistaan. Kehitysympäristössä ei pidetä henkilötietoja, vaan käytettävänä datana voidaan käyttää esimerkiksi generoitua testidataa tai vain käsin lisättyjä esimerkkirivejä.

Iteraatioiden edetessä projektitiimi ja asiakas näkevät kehitysympäristössä olevan (pidemmälle inkrementoidun) järjestelmän olevan otollisessa vaiheessa. Tällöin jäädytetään kehitysympäristön ohjelma *julkaisuksi* ja viedään se staging-ympäristöön. Staging-ympäristö on vakaampi muutoksien suhteen, ja siellä asiakas voi tehdä hyväksyntätestauksen ja projektitiimi laadunvarmistusta ennen tuotantoon vientiä. Tietokannan on hyvä vastata tuotantodataa, jotta uuden version toimivuuden vahvistaminen on varmempaa. Tämä on huomioitava tietosuojaa tarkastellessa.

Tuotantoympäristö on kunkin ohjelman version viimeinen vaihe, ja palveluissa se johon loppukäyttäjillä on pääsy. Kun päivitys on todettu tuotantokelpoiseksi, voidaan se ylentää staging-ympäristöstä tuotantoon. Tämä tarkoittaa ohjelman päivittämistä, mahdollisia palvelimen alustaohjelmien päivityksiä, tietokannan skeemamuutoksia ja mahdollisia päivityksiä dataan. Jokainen vaihe pitää tehdä hallitusti ja varmuudella. Automaatiolla voidaan usein tehdä mekaaniset osat ylläpitotyöstä, mutta jotkin tilanteet vaativat manuaalista operointia.

Tyypillisesti projektitiimistä kaikki tarpeelliset henkilöt saavat oikeudet hallita tietokantoja ja ajoympäristöjä, jollei projektin tietoturvaluokitus sitä erikseen muuta. Kehitysvaiheessa järjestelmän muutokset eivät ole vielä kriittisiä, mutta julkaisun jälkeen tuotantodataa alkaa kertyä ja sen hallinta on tärkeää. Pääsynhallinnan ja datan operoinnin tarkkuus korostuu siis projektin edetessä, ja kriteerit usein tiukentuvat jatkokehitys- ja ylläpitovaiheessa. Eri ympäristöjen välillä on myös eroja: kehitysympäristölle voi tehdä muutoksia vapaammin, kun taas esimerkiksi muutokset tuotannon tietokantaan tulee harkita kunnolla. Henkilötietojen osalta sama pätee myös.

Järjestelmän kehityskaaren loppupuolella kuvaan astuvat tuki- ja ylläpitotoimet jatkokehityksen lisäksi. Tapausyrityksellä on tukirinki, josta kulloinkin tietty henkilö on päivystäjä. Päivystäjä vastaa asiakkaiden pyyntöihin järjestelmää koskien, eli tuki on eri tasolla, kuin suora tuki loppukäyttäjille. Myös monitorointien hälytykset vikatilanteista ovat päivystäjän vastuulla. Tukitehtävät edellyttävät siis päivystäjältä pääsyn palvelimille ja ohjelmaan. Tukipyyntöjen lisäksi on muitakin tarpeita tehdä ylläpitotoimia tuotantopalvelimella, kuten palvelimen alustaohjelmien hallintaa. Tukihenkilöt eivät välttämättä ole alkuperäisestä kehitystiimistä, mistä johtuen he konsultoivat vaikeissa kysymyksissä asiasta perillä olevilta. Kun tutkielmassa myöhemmin tarkastellaan tietosuojaratkaisujen toimimista tukitoiminnan kannalta, viitataan ominaisuuteen *(O8: tukitoiminta ja palvelimien ylläpito)*.

Ohjelmistokehityksen toimintatavat tutkielman yrityksen kontekstissa on nyt esitelty. Tarkastelimme verkkosovellusjärjestelmää kolmen näkökulman kannalta: ohjelman, tietokannan ja palvelimen. Kehitystavasta esiteltiin kolme eri ympäristöä: kehitys-, staging- ja tuotantoympäristöt. Näitä koskee omat toimintatapansa, jotka aliluvussa käytiin läpi.

2.3.2 Esimerkkiprojekti

Otetaan tapaukseksi tarkempaan tarkasteluun muuan verkkolehtiprojekti. Referenssi on tarpeellinen selventämään, millaisessa ympäristössä tutkielmassa toimitaan. Asiakkaan liiketoiminta ei ole tässä oleellisessa osassa, eikä muutenkaan yhden tietyn projektin arkkitehtuuriin sitoutuminen. Pyritäänkin saamaan kuva yleisellä tasolla, millaisista verkkosovelluksista tutkielman rajauksessa on kyse. Tarkastellaan arkkitehtuuria erityisesti henkilötietojen näkökulmasta.

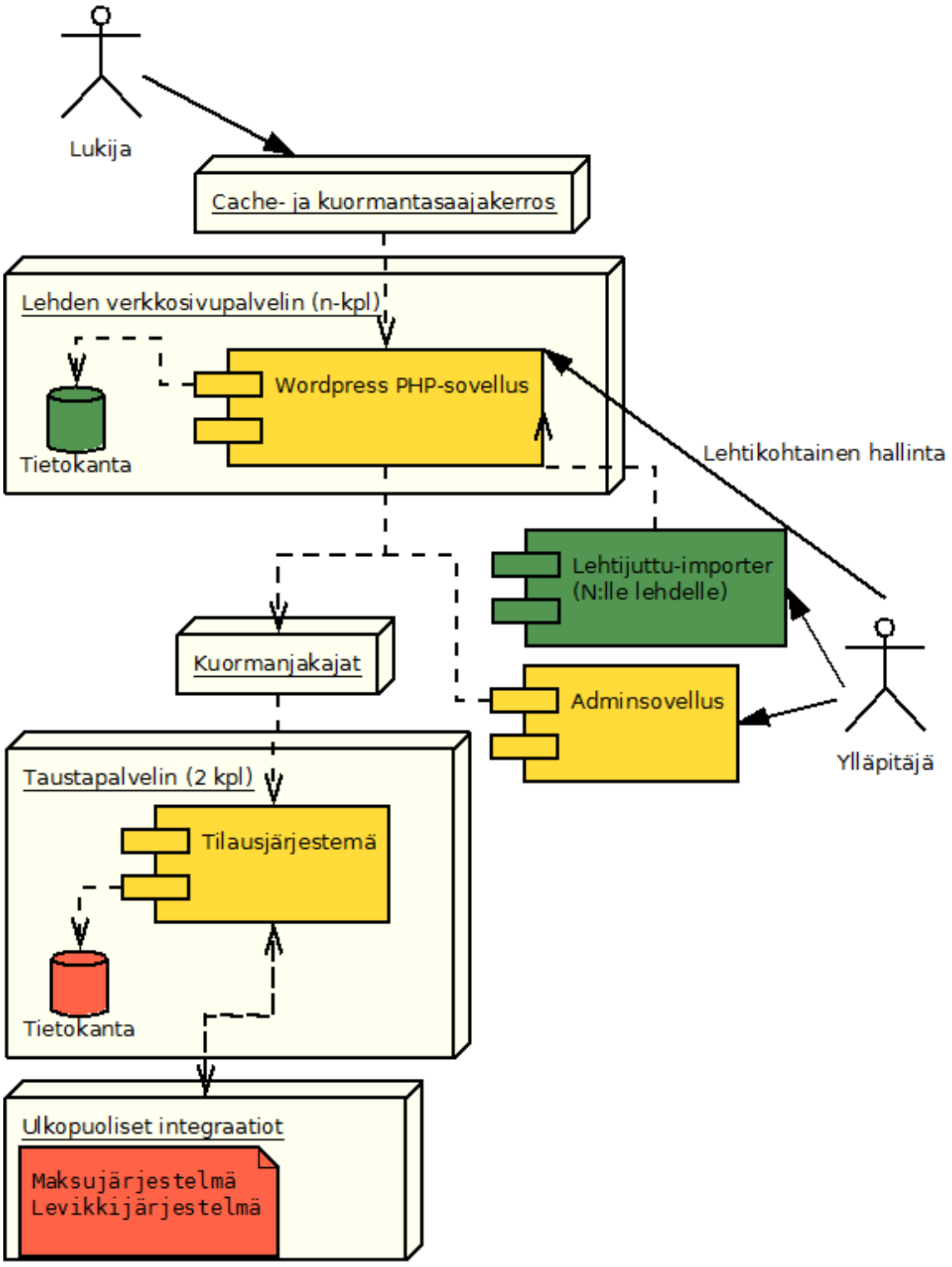
Valittu esimerkkiprojekti on lehtien digitaalisten versioiden hanke. Projektissa luotiin verkkopalvelu kullekin lehdelle ja taustajärjestelmät tämän mahdollistamiseksi. Käyttäjille näkyvä osa on paikallislehtien verkkosivut. Sivujen ulkoasu ja rakenne tulee olla muokattavissa, ja niihin tulee voida julkaista sisältöä. Tilauksia, käyttäjätilejä ja myyntiä hallitaan taustajärjestelmästä. Järjestelmässä on integraatioita paperilehden levikkiin ja maksujärjestelmään.

Edellä olevassa kuvassa *(Kuva 1)* on kaavio järjestelmän korkealla tasolla olevasta arkkitehtuurista. Huomioitavaa on järjestelmän jaottelu pienemmiksi palveluiksi ja jako eri palvelimien välillä. Käydään seuraavaksi läpi arkkitehtuurin komponentit.

Kullekin lehdelle on oma verkkosivupalvelimensa, joissa on *Wordpress* [2] PHP-sovellus. Wordpress on sisällönhallintajärjestelmä (Engl. *Content Management System, CMS*), joka vastaa tarpeeseen jokaisen lehden verkkosivun muokattavuudesta. Loppukäyttäjät (ylläpitäjät) voivat hallita sivuja itse. Koska lehtiä on useita, ne tarvitsevat omat palvelimensa suorituskyvyn vuoksi. Kukin käyttää omaa tietokantaansa, jossa pidetään vain sivujen rakenne ja sisältö. Käyttäjämäärien vuoksi verkkosivut tallennetaan välimuistiin, jota hallitaan lehden palvelimelta.

Verkkolehtien sovellukset kutsuvat taustapalvelimilla olevaa tilausjärjestelmää. Käyttäjätilit ja lehtitilaukset ovat osittain rinnakkaisia kaikkien lehtien välillä, joten on eriyttää ne omaksi palvelukseen. Tilausjärjestelmään tallennetaan käyttäjätilit, tilaukset ja myytävät tuotteet, ja niiden liiketoimintalogiikka on siellä. Tilausjärjestelmä myös hoitaa integraatiot ulkopuolisiin palveluihin. Ylläpitäjäkäyttäjille on oma sovellus, josta tilausjärjestelmää hallitaan. Tietokannat ovat taustajärjestelmässä niin kutsutussa *master–slave* -tyyppisessä suhteessa, jolloin lukuteho on suuri ja datan yhtenäisyys säilyy. Järjestelmä itse on Java-verkkosovellus, joka tarjoaa rajapinnan.

Henkilötietoja käsitellään järjestelmässä lehtitilausten vuoksi. Kuvassa olevien komponenttien väritys kuvaa niiden suhdetta henkilötietoihin: vihreät eivät sisällä henkilötietoja, keltaiset käsittelevät niitä ja punaiset varastoivat. Wordpress-sovelluksilla ei ole itsellään henkilötietoja tallessa, mutta ne käyttävät taustajärjestelmän tarjoamia tietoja. Henkilötietoja välitetään myös integraatioiden ylitse levikkijärjestelmälle. Huomataan tästä mahdollinen tarkastelukulma tietosuojalle: arkkitehtuurinen ominaisuus *(O9: henkilötietojen levinneisyys)*. Tällä tarkoitetaan sitä, monessako paikassa järjestelmässä käsitellään ja tallennetaan henkilötietoja. Tapausprojektissa henkilötiedot ovat rajoittuneet kahden tietokannan (yhden loogisen moduulin) sisään. Jos esimerkiksi käyttäjien osoitetiedot tallennettaisiin verkkosivupalvelimille, olisi arkkitehtuurin henkilötietojen levinneisyys suurempi kuin nykyratkaisussa. Hypoteesi on, että pienempi levinneisyys on parempi ratkaisu tietosuojan kannalta; esimerkiksi niin on vähemmän mahdollisesti murtauduttavia paikkoja, jotka vaarantaisivat henkilötietojen tietoturvan.



Kuva 1: Verkkolehtiprojektin arkkitehtuuri. Värit kuvaavat henkilötietojen käsittelyä.

2.4 Yhteenveto

Olemme nyt käyneet läpi eri kulmista tutkielman ympäristöä. Kerätään seuraavaksi yhteen luvussa tutkielman ympäristöstä tulkitut ominaisuudet, joiden mukaan tutkielman tietosuoja-arkkitehtuuria arvioidaan. Voimme jaotella arkkitehtuuriset ominaisuudet laatukriteereiksi ja ehdottomiksi vaatimuksiksi. Laatukriteerit ovat joustavampia ja niiden välillä voidaan tehdä kompromisseja ja vaihtokauppoja. Ehdottomia vaatimuksia ei tapausyrityksen kannalta ole montaa, tietosuoja-asetuksen lakiteksti taas määrittää näitä. Palaamme laatuattribuutteihin myöhemmässä luvussa, kun analysoimme arkkitehtuuria.

Liiketoiminnalliset kriteerit ovat motivaatio työlle, mutta yrityksen kannalta tärkeintä on saada luvussa esiteltyyn tekniseen ympäristöön sopiva ratkaisu yleisen tietosuoja-asetuksen vaatimuksille. Tämä on tutkielmassa suunniteltavan arkkitehtuurin hyväksymiskriteeri. Tässä luvussa etsittiin tätä ympäristöä kuvaavia ominaisuuksia, joihin peilataan sekä tietosuoja-asetusta että ehdotettavia ratkaisuja. Näitä erityiseen tarkasteluun otettavia ominaisuuksia löydettiin kahdeksan, ja ne on listattu edellä olevassa taulukossa *(Taulukko 1)*. Luetellaan seuraavaksi yhteenveto ominaisuuksista.

(O1: jako asiakkuuksien välillä) kuvaa sitä, miten arkkitehuurin tulee ottaa huomioon se, että eri projektit ovat eri asiakkuuksilla. Asiakkuuksilla ei ole yhteistä henkilötietorekisteriä, tai edes verkkoyhteyttä palvelimien välillä. *(O2: teknologiasidonnaisuus)* on huomiotava, koska asiakkaat voivat esittää vaatimuksia tiettyjen teknologioiden ja alustajärjestelmien suhteen. *(O3: projektin koon vaihtelu)* viittaa siihen, että ratkaisun tulee huomioida toimiminen eri kokoisten projektien välillä. Jossain määrin teknologiaan sitoudutaan kaikissa käytännön ratkaisuissa, mutta riippuuvuuden vahvuus on arvioitavana. *(O4: projektien ulkopuoliset investoinnit)* on vaihtokauppa, että paljonko tietosuojaratkaisusta tarvitsee kehittää projektien piirissä ja paljonko yleisenä kehityksenä.

Taulukko 1: Erityisesti tarkasteltavat järjestelmän ominaisuudet.

Ominaisuudet	
O1	jako asiakkuuksien välillä
O2	teknologiasidonnaisuus
O3	projektin koon vaihtelu
O4	projektien ulkopuoliset investoinnit
O5	staging-ympäristön vaatimukset
O6	vanhojen järjestelmien päivittäminen
O7	uudelleenkäytettävyys
O8	tukitoiminta ja palvelimien ylläpito
O9	henkilötietojen levinneisyys

(O5: staging-ympäristön vaatimukset) syntyi siitä, että toimintatavan mukaan perustetun staging-ympäristön vastaavuus tuotantoympäristön kanssa on maksimoitava, mistä seuraa haasteita tietosuojakysymyksille. *(O7: uudelleenkäytettävyys)* on *(O3:n)* ja *(O4:n)* kanssa läheinen, eli miten paljon työtä vaaditaan tutkielman ratkaisun käyttämiseksi uudelleen kontekstista toiseen. Vanhojen järjestelmien on sovittava uuteen ratkaisuun, ja muutosten tulee olla mahdollisimman pieniä. Vaihtoehtoisesti tietyille järjestelmille voidaan kustomoida tietosuoja, mutta tämän tulee olla kustannustehokasta. Käytännössä kustomoidut projektikohtaiset ratkaisut eivät ole kestäviä.

(O8: tukitoiminta ja palvelimien ylläpito) on toinen yrityksen toimintatavoista noussut ominaisuus, eli henkilötietojen käsittelyssä tulee ratkaista, miten järjestelmää ylläpidetään. *(O9: henkilötietojen levinneisyys)* on arkkitehtuurinen ominaisuus, jolla arvioidaan monessako paikassa järjestelmän sisällä on henkilötietoja. Oletus on, että pienempi levinneisyys helpottaa tietojen käsittelyä oikein.

Valittujen ominaisuuksien lisäksi tässä luvussa on kokonaisuutena esitelty työn konteksti ja rajattu millaisessa ympäristössä tutkielman arkkitehtuuria tullaan käyttämään. Esimerkkiyrityksestä pitäisi nyt olla kuva. Sekä tärkeämmin, kuva siitä, millaisessa asetelmassa verkkojärjestelmät tulevat projekteissa olemaan. Luvussa esiteltiin asiakasprojekteja tekevä liiketoimintamalli ja millaisia haasteita tästä seuraa. Käytiin läpi yrityksen toimintatapoja, eri arkkitehtuurin tasoja yrityksen kannalta ja eri tason ratkaisumalleja näille. Näitä havainnollistettiin käyttämällä esimerkkiarkkitehturina esiteltyä verkkolehtijärjestelmää. Luvun aiheista eriytettiin luetellut ominaisuudet, joiden mukaan voidaan tulkita tutkielman arkkitehtuuria ja tietosuoja-asetusta. Tutkielman taustojen ja vaatimusten määrittely siis on aloitettu tekniseltä ja liiketoiminnan kannalta, mutta sitä jatketaan ja laajennetaan seuraavaksi vielä tietosuoja-asetuksen osalta.

Luku 3
Tietosuoja-asetus

Esitellään yleinen tietosuoja-asetus ja käydään läpi sen artiklat. Fokus teknisiin kohtiin, joista johdetaan vaatimusmäärittely arkkitehtuurille.

Tässä luvussa käsitellään yleinen tietosuoja-asetus (Engl. *General Data Protection Regulation, GDPR*) sisältöineen ja vaatimuksineen. Euroopan parlamentti ja neuvosto antoivat yleisen tietosuoja-asetuksen (asetus (EU) 2016/679) 27.4.2016. Tutkielman kannalta tärkeimpänä keskittymiskohteena ovat asetuksen tekniset vaatimukset, joita arvioidaan luvussa 2 esitellyn kontekstin ominaisuuksien näkökulmasta. Tavoitteena on käydä asetuksen kohdat läpi, mutta ei tyytyä ainoastaan referoimaan niitä.

Ensimmäisessä aliluvussa 3.1 käydään läpi asetuksen sisältö vastaavalla jaottelulla, kuin varsinaisen asetuksen luvut ja artiklat. Kutakin artiklaa arvioidaan tutkielman teknisen ympäristön kannalta, ja siten rajauksen ulkopuoliset kohdat jätetään pienemmälle huomiolle. Näin pysytään tutkielman ydinkysymyksessä kiinni. Tämän jälkeen aliluvussa 3.2 tarkastellaan tietosuoja-asetusta kokonaisuutena ja verrataan sitä työn teknisen ympäristön kokonaisuuteen. Tästä saadaan vaatimusmäärittely tutkielman arkkitehtuurille. Aliluku 3.3 kokoaa yhteen luvun tulokset.

Luvun tärkein viite on yleinen tietosuoja-asetus itse [40]. Täydeltä suomenkieliseltä nimeltään asetus on *Asetus luonnollisten henkilöiden suojelusta henkilötietojen käsittelyssä sekä näiden tietojen vapaasta liikkuvuudesta ja direktiivin 95/46/EY kumoamisesta (yleinen tietosuoja-asetus)*. Asetus on saatavilla verkossa jäsenmaiden omilla kielillä. Muuta tietosuoja-asetusta käsittelevää materiaalia ja tutkimusta arvioidaan luvussa 4.

3.1 Asetuksen sisältö

Yleinen tietosuoja-asetus koostuu kahdesta osasta. Ensimmäinen osa on komission perustelu ja tausta sille, miksi asetus on tehty. Tätä ei käydä erikseen läpi, vaan viitataan perusteluihin, kun niitä koskevia artikloita käsitellään. Toinen osa on varsinainen lakiteksti. Asetus on jaoteltu lukuihin, joissa on numeroituja artikloja. Artiklat koostuvat numeroiduista kohdista, jotka sisältävät vielä kirjaimin merkittyjä alikohtia. Kun tässä

tutkielmassa viitataan asetuksen artiklaan, käytetään notaatiota *Artikla A1(3f)* tai lyhyemmin *A1*. Numero viittaa artiklan numeroon, jota seuraa sulkeissa mahdollinen tarkennus viitattavaan kohtaan ja alikohtaan.

Käydään seuraavaksi läpi asetuksen sisältö sen luvuittain niin, että kutakin asetuksen lukua vastaa tutkielman aliluku. Asetuksen sisällön esittelyn jälkeen voimme tarkastella sitä tapausyrityksen kannalta ja tehdä vaatimusmäärittelyn.

3.1.1 Yleiset säännökset

Asetuksen luku 1, *yleiset säännökset*, koostuu neljästä ensimmäisestä artiklasta ja sisältää asetuksen kohteet ja tavoitteet, aineellisen ja alueellisen soveltamisalan, sekä termien määritelmiä. Lyhyesti, *Artikla A1* kertoo asetuksen tavoitteen olevan henkilötietojen ja niiden liikkuvuuden suojelu.

Artikla A2 määrittelee asetuksen aineellisen soveltamisalan. Tällä tarkoitetaan millaiseen tiedonkäsittelyyn asetus liittyy. Sovellusalaksi artikla määrittää henkilötietojen automaattisen käsittelyn ja henkilötietorekisteriin kuuluvien tietojen käsittelyn. Rajauksia tähän on, esimerkiksi henkilön henkilökohtainen toiminta ei kuulu alaan, mutta lähtökohtaisesti tutkielman tapausyrityksen toiminta kuuluu aina soveltamisalaan. *Artikla A3* rajaa asetuksen alueellisen soveltamisalan. Tämä on tiivistettynä kaikki EU:n alueella tapahtuva henkilötietojen käsittely ja kaikki eurooppalaisten henkilöiden tietojen käsittely. Tutkielman tapausyritys on pääsääntöisesti EU:n henkilötietojen kanssa tekemisissä, mutta *(O1: jaon asiakkuuksien välillä)* mukaan pitää olla avoin myös muuhun toimintaan.

Asetuksen luvun 1 lopuksi, *Artiklassa A4* määritellään asetuksessa käytettäviä termejä. Kiinnostavimmat tutkielman kannalta ovat *henkilötiedon*, *profiloinnin* ja *pseudonymisoinnin* määritelmät. Loppuja artiklan termeistä ei ole mielekästä esitellä tässä, niihin voidaan palata viitatessa.

Henkilötiedot ovat asetuksessa kaikki tiedot, joista voidaan tunnistaa luonnollinen henkilö: suoraan tai epäsuorasti, yhden tai useamman tekijän perusteella. Profilointi tarkoittaa *”mitä tahansa henkilötietojen automaattista käsittelyä, jossa henkilötietoja käyttämällä arvioidaan luonnollisen henkilön tiettyjä henkilökohtaisia ominaisuuksia”*. Pseudonymisoiminen on prosessi, jossa henkilötietoja muokataan siten, ettei niistä enää voi tunnistaa rekisteröityä *käyttämättä lisätietoja*. Tämä edellyttää mahdollisten lisätietojen erillään säilyttämisen ja yhdistämisen estämisen.

Henkilötietojen määritelmää tullaan tarvitsemaan yleisesti järjestelmiä suunnitellessa, mutta tekniset ratkaisut henkilötietojen käsittelyyn eivät sinänsä ota kantaa, mihin tiettyihin tietoihin niitä käytetään. Tutkielman tavoite ei kuitenkaan ole keinotella kysymyksellä siitä, mikä lasketaan henkilötiedoksi. Pseudonymisointi on käsitteenä tärkeän

oloinen tapausyrityksen ominaisuuden *(O5: staging-ympäristön vaatimukset)* kannalta. Palataan kuitenkin asiaan myöhemmin kohdassa, jossa asetus määrittelee säännöt pseudonymisoinnin käytölle.

3.1.2 Periaatteet

Asetuksen luvun 2 aihe on asetuksen *periaatteet*. Nämä tiivistävät tavoitteet asetuksen takana ja pohjustavat sen, mitä loput artiklat tarkentavat. Lienee perusteltavissa, että asetuksen periaatteet ovat sen tärkein osa; niitä seuraamalla teoriassa noudattaa asetusta. Loppuosassa asetusta kerrotaan, mitä periaatteiden noudattaminen käytännössä tarkoittaa. Tässä asetuksen luvussa on seitsemän artiklaa. *Artikla A5* määrittää *henkilötietoja koskevat periaatteet*, *Artikla A6 käsittelyn lainmukaisuuden* ja *Artikla A7 suostumuksen edellytykset*. *Artiklassa A8* määritetään tietoyhteiskunnan palveluihin liittyvään *lapsen suostumukseen sovellettavat ehdot*, *Artiklassa A9 erityisiä henkilötietoryhmiä koskevan käsittelyn*, *Artiklassa A10 rikostuomioihin ja rikkomuksiin liittyvien henkilötietojen käsittelyn*, ja *Artiklassa 11 käsittelyn, joka ei edellytä tunnistamista*. Käydään näiden sisältö seuraavaksi läpi.

Artiklan 5 määrittämät henkilötietojen käsittelyä koskevat periaatteet ovat asetuksen perusta. Henkilötietoja koskevat seuraavat vaatimukset: *A5(1a): lainmukaisuus, kohtuullisuus ja läpinäkyvyys*, *A5(1b): käyttötarkoitussidonnaisuus*, *A5(1c): tietojen minimointi*, *A5(1d): täsmällisyys*, *A5(1e): säilytyksen rajoittaminen* ja *A5(1f): eheys ja luottamuksellisuus*. Artiklan toinen kohta vielä määrää, että rekisterinpitäjän on voitava osoittaa periaatteiden noudattaminen – *(A5(2): osoitusvelvollisuus)*. Täsmällisyydellä tarkoitetaan sitä, että kerättyjen henkilötietojen on oltava oikeita. Säilytyksen rajoittaminen vaatii henkilötietoja säilytettävän vain niin kauan, kuin niiden keräämistarkoitus vaatii. Tiivistettynä, periaatteet täyttyvät, kun valideja henkilötietoja kerätään suostumuksella, tiettyyn tarkoitukseen, ja niitä kerätään mahdollisimman vähän. Henkilötietojen käsittelyssä niiden elinkaari on oltava hallinnassa ja käsittely tulee tapahtua tietoturvallisesti.

Seuraavaksi *Artiklassa A6* määritetään, milloin henkilötietojen käsittely on lainmukaista. Lainmukaisuus edellyttää artiklan kohtien mukaan sen, että tietoja käsitellään vain rekisteröidyn suostumuksella, tai rekisteröityä koskevan sopimuksen täytäntöönpanemiseksi, lakisääteisen velvoitteen tai elintärkeän edun vuoksi, julkisen vallan käyttämiseksi, tai rekisterinpitäjän oikeutettujen etujen toteuttamiseksi. Kohtia koskee tarkentavia ehtoja, ja *Artiklat A8-A10* määrittelevät vielä tiukemmat ehdot lasten ja erityisryhmien henkilötietojen käsittelyyn, sekä rajoittavat rikostuomioihin liittyvien tietojen käsittelyä. Voidaan olettaa tapausyrityksen käsittelyn täyttävän vähintään jonkin lainmukaisuuden ehdoista, tämän tutkielman puitteissa. Suostumuksen hallinta on kuitenkin otettava tarkasteltavaksi. *Artiklassa A7* määritelläänkin suostumuksen edellytykset. Näistä kohdassa 2 esitetään edellytys siitä, että suostumus kysytään selkeästi ja erillään muusta palvelun toimista. Kohdassa 3 kirjataan oikeus peruuttaa suostumus koska tahansa.

Asetuksen *Artikla A11* antaa rekisterinpitäjälle omanlaisensa vapauden: mikäli rekisterinpitäjän henkilötietojen käsittely onnistuu ilman rekisteröidyn tunnistamista, ei rekisterinpitäjän tarvitse säilyttää, hankkia tai käsitellä lisätietoja rekisteröidyn tunnistamiseksi vain asetuksen noudattamista varten. Tällainen henkilötieto koskee yhtä henkilöä, mutta tämän tiettyä henkilöllisyyttä ei siitä voi tunnistaa. Artiklan kohdan 2 mukaan, kyseisellä henkilöllä on oikeus tarjota puuttuvat tiedot hänen uudelleentunnistamisekseen, jotta hän voi jälleen käyttää asetuksen oikeuksiaan. Artiklan mukaiset tunnistuskyvyttömät tiedot ovat yksi vaihtoehto pseudonymisoinnissa, kun tutkielman arkkitehtuuriratkaisuja suunnitellaan.

3.1.3 Rekisteröidyn oikeudet

Seuraavasssa asetuksen luvussa määritellään ne oikeudet, joita rekisteröidyllä on. Luku kattaa artiklat 12-23, joissa oikeudet ja niiden rajoitukset kerrotaan yksityiskohtaisesti. Oikeudet ovat *läpinäkyvä informointi*, *tiedonsaanti rekisterinpitäjältä*, oikeudet *nähdä*, *oikaista* ja *poistaa henkilötiedot*, oikeus *käsittelyn rajoittamiseen*, *tulla informoiduksi*, oikeus *siirtää tiedot rekisterien välillä*, *vastustamisoikeus* ja oikeus *olla joutumatta automatisoidun päätöksen kohteeksi*. Oikeudet heijastuvat voimakkaasti vaatimuksiksi arkkitehtuurille. Esitellään seuraavaksi mainitut oikeudet ja tulkitaan niiden seurauksia tässä aliluvussa.

Artiklan A12 kahdeksassa kohdassa esitetään vaatimukset rekisterinpitäjälle siitä, miten rekisteröityjä tulee tiedottaa henkilötietojen käsittelyssä ja säännöt rekisteröidyn oikeuksien käyttöä varten. Rekisteröidylle toimitettava tiedot tulee toimittaa *ymmärrettävässä* muodossa ja ilman aiheetonta viivytystä. Rekisteröidyn pyynnöstä käyttää oikeuksiaan ei saa kieltäytyä ja toimien tulee olla maksuttomia. Perusteettomat tai kohtuuttomat pyynnöt eivät tosin ole oikeutettuja. Teknisempää määrittelyä artiklassa ei ole, joten kysymys on hyvästä viestinnästä rekisteröidyille. Kohdissa 7 ja 8 viitataan mahdollisuuteen saada standardoituja kuvakkeita viestinnän avuksi – tämä olisi palvelunkehittäjille hyvä asia, ominaisuuden *(O7: uudelleenkäytettävyys)* mukaisesti.

Artiklat A13 ja *A14* määrittävät tiedot, jotka rekisteröidylle tulee antaa henkilötietojen keräämisen yhteydessä. Toimitettavat tiedot sisältävät informaatiota siitä, miten ja miksi henkilötietoja käsitellään rekisterin puitteissa. Suurin osa näistä kohdista ovat hallinnollisia seikkoja. Erikseen mainittavia kohtia ovat henkilötietojen *säilytysaika* ja *säilytyksen päättymisen kriteerit*. Nämä ovat siis tiedostettava jo etukäteen, ja oltava valmiita kertomaan projektikohtaisesti. Muutenkin kaikki artiklan kohdat vaativat tietoja, jotka on valmisteltava henkilötietoja käsitteleviä järjestelmiä suunnitellessa. Kun näitä katsotaan tapausyrityksen ominaisuuksien *(O1: jako asiakkuuksien välillä)* ja *(O3: projektien koon vaihtelu)* kannalta, tuo vaikeutta se, että vaadittavat tiedot ovat hyvin käyttötarkoituskohtaisia. Aiemmin mainittujen kuvakkeiden lisäksi, lienee kannattavaa luoda valmiita pohjia näille tiedonvälityslomakkeille, joihin käyttötarkoituskohtaiset tiedot täytetään.

Artikla A15 kertoo rekisteröidyn oikeudesta saada pääsy tietoihinsa. Vaatimus on, että rekisteröity saa pyynnöstä vahvistuksen siitä, että käsitelläänkö häntä koskevia henkilötietoja ja pääsyn kyseisiin tietoihin. Lisäksi tulee antaa edellisessä kappaleessa käsitellyt tiedot henkilötietojen käsittelystä. Tämä oikeus on itsessään ihan selkeä vaatimus, mutta sen seuraaminen edellyttää henkilötietojen sijainnin järjestelmässä olevan hallinnassa. Hypoteesi on, että suurempi *(O9: henkilötietojen levinneisyys)* vaikeuttaisi tämän vaatimuksen toteuttamista.

Oikeuden henkilötietojen näkemisestä jatkoksi, *Artikla A16* määrittää oikeuden tietojen oikaisemiseen. Epätarkat ja virheelliset henkilötiedot on pyynnöstä oikaistava, ilman aiheetonta viivytystä. *Artikla A17* taas antaa rekisteröidylle oikeuden tietojen poistamiseen. Tämä tarkoittaa, että henkilötiedot on poistettava heti, kun on täytetty se tarkoitus, jota varten ne kerättiin. Rekisteröity voi peruuttaa suostumuksensa tietojen pitämiseen koska tahansa. Poistamiselle on joitain rajoituksia, mutta lähtökohtaisesti siihen tulee olla valmius. Näiden artiklojen vaatimukset vuorovaikuttavat *(O9:n)* kanssa kuten *A15*. Kaikki kolme vaativat myös sen, että henkilötiedot tallennetaan rakenteisesti – tai vähintään assosiaatiolla tunniste–muu data. Poistamisen vaatimus pitää myös huomioida varmuuskopioiden suhteen. On selvitettävä, onko viranomaisilla kantaa siihen, että pitääkö poistamispyynnön tullessa poistaa henkilötiedot kaikista varmuuskopioistakin. Vähintään vaatimuksesta seuraa se, että varmuuskopiosta palautettaessa pitää uudelleen poistaa jo poistetut tiedot. Tämä vaatii oman "poistopyyntölokinsa", jota pidetään tietokantojen rinnalla.

Artikla A18 antaa rekisteröidylle oikeuden rajoittaa henkilötietojensa käsittelyä. Rajoitustilanteessa henkilötietoja ei saa käsitellä, mutta ne pitää säilyttää rajoituksen ajan, vaikka muuten poisto olisi aiheellista. Rajoituksen voi tehdä, jos rekisteröity kiistää tietojensa paikkansapitävyyden, jos käsittely on lainvastaista, tai jos rekisterinpitäjä ei tarvitse tietoja enää, mutta rekisteröity tarvitsee niitä oikeudellisista syistä. Käsittely tulee myös rajoittaa, jos rekisteröity käyttää vastustamisoikeuttaan. Rajoitus-ominaisuus on kokonaisuudessaan oma itsenäinen vaatimuksensa, jota voidaan käsitellä erikseen. Mikäli järjestelmässä on suuri *(O9: henkilötietojen levinneisyys)*, tulee työlääksi tarkkailla kaikissa paikoissa mahdollisten rajoituksien voimassaoloa. Yksi vaihtoehto on hallita rajoituksia yhdessä paikassa, mutta tällöinkin on tehtävä aina kyselyitä rajoitustenhallintaan ennen muuta henkilötietojen käsittelyä. Helppo ratkaisu olisi tulkita kaikki rajoituspyynnöt henkilötietojen poistopyynnöksi, mutta artikla ei salli tätä.

Aiempia oikeuksia täydentää *Artikla A19*, joka määrittää rekisterinpitäjälle vielä velvollisuuden ilmoittaa eteenpäin kaikista oikaisuista, poistoista, tai rajoituksista kaikille tahoille, joille henkilötietoja on luovutettu. Luovutettujen tietojen vastaanottajista tulee myös antaa tieto rekisteröidylle pyydettäessä. Tästä ilmoitusvelvollisuudesta seuraa se, että kerran kerätyt henkilötiedot saa poistettua yhdellä pyynnöllä, vaikka niitä oltaisiin

levitetty eteenpäin moneenkin suuntaan. Luovutettua tietoa voi teoriassa luovuttaa edelleen montakin kertaa (rekisteröidyn hyväksynnällä). Syntyvää henkilötietojen luovutusten referenssiverkostoa pitää hallita vähintään kunkin yksittäisen rekisterin kohdalla niin, että rekisteri tietää omat luovutuskohteensa.

Artikla A20 antaa oikeuden siirtää tiedot järjestelmästä toiseen. Tämäkin on rekisteröidyn pyynnöstä tapahtuva toimenpide, jossa vähintään luovutetaan henkilötiedot *jäsennellyssä*, *yleisesti käytetyssä* ja *koneellisesti luettavassa* muodossa. Mikäli on teknisesti mahdollista, on siirto tehtävä suoraan rekisterinpitäjältä toiselle. Artiklasta ei vaikuttaisi seuraavan paljoa lisätyötä nykyisten vaatimusten lisäksi; jo oikeus tietojen katseluun vaatii mahdollisuuden kerätä kaikki tiedot. Tällöin ainoa ero olisi tietojen esitystapa. Siirtojen automatisointi tarvitsee enemmän määrittelyä *teknisesti mahdollisen* suhteen. Kaikki henkilötietojen siirrot ovat jotenkin teknisesti mahdollisia, mutta silti integraatioiden sopiminen kaikkien eri rekisterien välillä on kohtuuttoman suuri työ, jollei kehitetä jotain yleistä siirtostandardia.

Artikla A21 määrittää vastustamisoikeuden. Rekisteröity voi vastustaa henkilötietojensa käsittelyä vedoten erityiseen henkilökohtaiseen syyhyn. Mikäli rekisterinpitäjällä on huomattavan tärkeä (lainmukainen) syy jatkaa käsittelyä, tämä voi syrjäyttää vastustamisoikeuden. Suoramarkkinoinnin käsittelyä voi vastustaa aina. Kun organisaatio saa vastustamispyynnön, käytetään aiemmin mainittua tietojenkäsittelyn rajoittamista siihen asti, että selvitetään onko vastustus pätevä. Artiklan kohdassa 5 erikseen mainitaan, että vastustuksen voi tehdä teknisiä polkuja käyttäen kun mahdollista. Vastustamisoikeuden ja oikeuden tulla unohdetuksi (*A17*) välillä on paljon yhteistä. Ausloos selventää artiklojen tarkoituksia vuoden 2016 (epämuodollisessa) artikkelissaan [1] aiheesta. *A17* vaatii syyn tietojen poistamiselle. Yksi valideista syistä on *A21:n* vastustusoikeuden käyttö. Vastustamisoikeus koskee tiettyä tietojen prosessointia, kun taas poistaminen kaikkia tietoja. Rekisteröity joutuu siis tekemään sekä *A21*, että *A17* pyynnöt saadakseen tietonsa todellakin poistettua.

Artikla A22 koskee automatisoituja yksittäispäätöksiä. Artiklan kohdan 1 tarkka sanamuoto on seuraava: *"Rekisteröidyllä on oikeus olla joutumatta sellaisen päätöksen kohteeksi, joka perustuu pelkästään automaattiseen käsittelyyn, kuten profilointiin, ja jolla on häntä koskevia oikeusvaikutuksia tai joka vaikuttaa häneen vastaavalla tavalla merkittävästi"*. Tällä on omat vaatimuksensa ja sen voi kiertää nimenomaisella suostumuksella. Artiklan kysymys on siis sopimuksellinen, eikä arkkitehtuurin tarvitse ottaa siihen kantaa. Asetuksen luvun viimeinen artikla, *Artikla A23*, myöntää viranomaisille mahdollisuuden rajoittaa oikeuksien soveltamisalaa tietyissä tilanteissa.

Tässä aliluvussa esiteltiin tietosuoja-asetuksen rekisteröidyn oikeudet. Monessa artiklassa on yksityiskohtia ja lievennyksiä, joihin ei tässä paneuduttu. Tämän perustelu

on se, että tietojärjestelmien – ja siten arkkitehtuurin – tulee kuitenkin varautua kaikkiin tapauksiin. Tällöin kaikkiin vaatimuksiin pitää olla toiminnallisuus, vaikka sen käyttö olisi harvinaista, tai jonkin oikeuden noudattamisesta voidaan tapaus kerrallaan väitellä. Sivuuttamalla lakitekniset yksityiskohdat, saamme keskityttyä teknisten ongelmien ratkaisemiseen.

Monet osion vaatimuksista sisältävät oikeuden johonkin, mihin rekisteröity esittää pyynnön. Esimerkiksi käyttääkseen *A17* oikeuttaan tietojen poistamiseen, rekisteröidyn tulee jotenkin ottaa yhteys palveluntarjoajaan. Näiden pyyntöjen hallinta ja tapausyrityksen ominaisuuden *(O8: tukitoiminta ja palvelimien ylläpito)* huomioiminen on yksi tarkasteltava seikka rekisteröidyn oikeuksista kokonaisuutena. Palataan tähän ratkaisuja pohdittaessa.

3.1.4 Rekisterinpitäjän velvollisuudet

Tietosuoja-asetuksen luvussa 4 *"rekisterinpitäjä ja henkilötietojen käsittelijä"*, käydään läpi, mitä oikeuksia ja velvollisuuksia on rekisterinpitäjillä ja henkilötietojen käsittelijöillä. Luku kattaa artiklat 24-43 ja sisältää yleiset velvollisuudet, osion henkilötietojen turvallisuudesta, määrittelee tietosuojaa koskevan vaikutustenarvioinnin ja ennakkokuulemisen, ja esittelee tietosuojavastaavan roolin.

Rekisterinpitäjä ja henkilötietojen käsittelijä määriteltiin artiklassa *A4*. Rekisterinpitäjä on taho, joka määrittelee henkilötietojen käsittelyn tarkoitukset ja keinot. Henkilötietojen käsittelijä taas on taho, joka käsittelee henkilötietoja rekisterinpitäjän lukuun. Suurin vastuu henkilötietoja käsittelevistä järjestelmistä on siis rekisterinpitäjällä. Tapausyrityksen asiakasprojekteista valtaosassa rekisterinpitäjänä on asiakas, jolloin yritys toimii käsittelijänä *(O1: jako asiakkuuksien välillä)*. Käytännössä ohjelmistojen kehitystapa ei tule muuttumaan, eli tapausyrityksen asiantuntijat tekevät (vähintään ohjelmistoja koskevat) suunnittelupäätökset. Tämä jaottelu on myös tutkielman rajauksen lähellä, sillä nimenomaan henkilötietojen käsittelyn *tarkoituksiin* ei oteta kantaa. Toisaalta kaikki asetuksen rekisterinpitäjää koskevat tekniset vaatimukset ovat suoraan vaatimuksia tutkielman arkkitehtuurille, koska asiakkaat eivät huolisi epäpäteviä ratkaisuja.

Käydäänkin seuraavaksi luvun artiklat läpi. *Artikla A24* määrittelee rekisterinpitäjän vastuun. Artiklassa tämä on tiivistetty vastaavasti: *"Ottaen huomioon käsittelyn luonne, laajuus, asiayhteys ja tarkoitukset sekä luonnollisten henkilöiden oikeuksiin ja vapauksiin kohdistuvat, todennäköisyydeltään ja vakavuudeltaan vaihtelevat riskit rekisterinpitäjän on toteutettava tarvittavat tekniset ja organisatoriset toimenpiteet, joilla voidaan varmistaa ja osoittaa, että käsittelyssä noudatetaan tätä asetusta. Näitä toimenpiteitä on tarkistettava ja päivitettävä tarvittaessa."* Huomioidaan tästä, että asetus kuitenkin antaa rekisterinpitäjälle hieman tulkinnanvaraa riskienhallinnan suhteen. Toinen kohta on osoittamisvelvollisuus, eli on tehtävä toimenpiteet, joilla voidaan osoittaa asetusta noudatettavan.

Miten oikeuslaitos tulee tulkitsemaan näitä, on vielä epäselvää.

Artikla A25 vaatii rekisterinpitäjän noudattavan *sisäänrakennettua* ja *oletusarvoista* tietosuojaa. Artiklan mukaan rekisterinpitäjän on toteutettava asetuksessa aiemmin esiteltyjen tietosuojaperiaatteiden noudattamiseksi asianmukaiset tekniset ja organisatoriset toimenpiteet. Oletusarvoisesti saa käsitellä ainoastaan kunkin erityisen tarkoituksen kannalta tarpeellisia henkilötietoja. Erityishuomiota halutaan sille, etteivät henkilötiedot pääse leviämään rajoittamattoman henkilömäärän saataville. Jälleen artiklassa on tulkinnanvaraa sen verran, että rekisterinpitäjä saa huomioida asiaankuuluvat riskit. Tämän artiklan vaatimus on varsin yleisellä tasolla, mutta korostaa tutkielman tavoitteita.

Seuraavana *Artiklat A26* ja *A27* tarkentavat rekisterinpitäjän käsitettä. Yhteisrekisterinpitäjän käsite on käytössä, kun kaksi rekisterinpitäjää yhdessä määrittelevät henkilörekisterin käsittelyn tarkoitukset ja keinot. Tämä tilanne on mahdollinen *(O1)* asiakasprojektimallissa, riippuen siitä, miten henkilötietojen käsittelyn *tarkoitusten ja keinojen* määrittelyä tulkitaan. Tällöin asetus vaatii läpinäkyvästi tehdyn vastuunjaon ja roolituksen pitäjien välillä. *A27:ssa* vaadiitaan, että EU:n ulkopuolella sijaitsevat rekisterinpitäjät nimeävät edustajan EU:n sisältä. Edustajan tulee olla rekisterinpitäjien kontakti viranomaistahoihin. Tässäkin tehtävässä tapausyritys voi mahdollisesti toimia.

Artiklassa A28 esitetään vaatimukset henkilötietojen käsittelijälle. Suuri osa artiklan kohdista koskee sopimuksia rekisterinpitäjän ja käsittelijän välillä. Esimerkki vaatimuksista on salassapitovelvollisuus, lisäksi artiklassa on asetuksen muiden vaatimusten noudattamisen varmistamista. Henkilötietojen käsittelijä ei saa käyttää toisen käsittelijän palveluksia, jollei rekisterinpitäjä anna ennakkolupaa. Huomioitavaa on, että käsittelijällä on velvollisuus ilmoittaa välittömästi rekisterinpitäjälle, mikäli ohjeistus rikkoo tietosuojaasetusta tai muuta tietosuojasäännöstä. Käsittelijän on myös sitouduttava siihen, että käsittelee henkilötietoja *"ainoastaan rekisterinpitäjän antamien dokumentoitujen ohjeiden mukaisesti"*. Tämä vaatii monimutkaisemmissa tilanteissa sitä, että rekisterinpitäjä antaa ohjeistuksessa paljon vapauksia, tai muuten työ voisi hidastua. *Artikla A29* lisää kohdan, että kukaan rekisterinpitäjän tai käsittelijän alaisuudessa työskentelevä ei saa myöskään käsitellä henkilötietoja muuten kuin ohjeiden mukaisesti.

Rekisterinpitäjä velvoitetaan pitämään yllä ajantasaista *selostetta käsittelytoimista*. *Artikla A30* määrittää tämän sisällön, joka kattaa tiedot kyseessä olevan rekisterin henkilötietojen käsittelystä. Henkilötietojen käsittelijän on myös pidettävä oma selosteensa käsittelytoimista. Erikseen mainittava kohta selosteen sisällöstä on kuvaus teknisistä ja organisatorisista turvatoimista, joka on mahdollisuuksien mukaan kirjattava. Tämä on linjassa *A5(2):n* osoitusvelvollisuuden kanssa, ja edellyttää arkkitehtuurin olevan hallittu ja dokumentoitavissa. *Artikla A31* vielä täsmentää, että rekisterinpitäjän ja käsittelijän on tehtävä yhteistyötä valvontaviranomaisten kanssa.

Artikla A32 käsittelee rekisterinpitäjän velvollisuutta henkilötietojen *käsittelyn turvallisuudesta*. Rekisterinpitäjän pitää tehdä käsittelyä koskevien riskien vastaavan turvallisuustason varmistamisen vaatimat toimet. Tässä annetaan harkintakykyä käsittelyn ominaisuuksien ja toteuttamisen mahdollisuuksien kanssa, kunhan riskinhallinta on oikein. Kohdassa 1 luetellaan seuraavat vaadittavat toimenpiteet: *henkilötietojen pseudonymisointi ja salaus*, järjestelmien *luottamuksellisuus, eheys, käytettävyys* ja *vikasietoisuus*, *nopea palautuminen virhetilanteista* ja *menettely turvallisuuden arvioimiseksi*. Riskeistä korostetaan erityisesti henkilötietojen eheyttä ja tietoturvaa. Lisäksi pitää myös varmistaa, että kaikki rekisterinpitäjän alaiset käsittelevät tietoja vain ohjeiden mukaisesti. Artiklan vaatimukset ovat yleisestikin hyvän ohjelmistoarkkitehtuurin ominaisuuksia, joten näiden toteuttamisen ei pitäisi olla uusi asia. Se voi johtaa strategisiin seurauksiin, että näiden laatuominaisuuksien laiminlyömisellä on lainmukaiset seuraamukset. Esimerkiksi teknisen velan käyttö näiden suhteen ei tuotantojärjestelmässä välttämättä onnistu.

Artiklat A33 ja *A34* koskevat tietoturvaloukkauksista ilmoittamista. Tietoturvaloukkauksen sattuessa on toimitettava valvontaviranomaiselle ilmoitus ilman aiheetonta viivytystä. Ilmoitus on myös toimitettava rekisteröidyille, mikäli tapaus aiheuttaa korkean riskin rekisteröidyn oikeuksia ja vapauksia kohtaan. Tietoturvaloukkauksista on kartoitettava todennäköiset seuraukset ja ketä se koskee. On kuvattava toimenpiteet, joita on tehty haittavaikutusten lieventämiseksi. Kaikki loukkaukset ja niihin liittyvät seikat on dokumentoitava, mukaanlukien toteutetut korjaavat toimet. Rekisteröidyille ei tarvitse tehdä ilmoitusta, mikäli henkilötiedot ovat salattuja ja rekisterinpitäjä on tehnyt jatkotoimenpiteet, jotka varmistavat ettei tilanne toistu. Ei ole realistista odottaa, ettei tietoturvaloukkauksia koskaan tapahtuisi, vaikka tietoturvaan sijoitetaankin. Järjestelmän luonteen ja riskien mukaan toteutettu tietoturvaloukkausten hallintasuunnitelma [46] kuitenkin vähentää seurauksia. Artiklan vaatimus on siis yksi vaihe lisää normaaliin suunnitelmaan. Asetuksen myötä hallintasuunnitelman ja monitoroinnin merkitys korostuu, kun muiden vaikutusten lisäksi uhkana ovat asetuksen sanktiot.

Seuraavaksi asetus määrittelee *tietosuojaa koskevan vaikutustenarvioinnin Artiklassa 35*. Tällainen arviointi tulee tehdä, kun henkilötietojen käsittely aiheuttaa korkean riskin henkilötietojen turvalle. Erityisesti mainitaan uuden teknologian käyttöönotto. Vaikutustenarvioinnissa tulee kuvata käsittelytoimet, tarkoitukset, tarpeellisuus, riskit ja riskien hallinta. Mikäli käsittelytoimien riski jatkossa muuttuu, on arviointi tehtävä uudelleen. Miten korkea riski määritellään on tulkintakysymys, mutta otetaan arvionti huomioon. *Artikla A36* määrää vaikutustenarvioinnin lisäksi *ennakkokuulemisen*. Ennakkokuuleminen on tehtävä, jos vaikutustenarvioinnissa todetaan olevan korkea riski käsittelyssä ja riskiä ei jatkotoimilla pienennetä. Tällöin on kuultava valvontaviranomaisia ennen käsittelyn aloittamista ja otettava huomioon heidän ohjeistuksensa. Vaikutusten ja riskien arviointi on hyvin projektikohtaista. Arkkitehtuuriratkaisuilla voidaan helpottaa arviota, tai pienentää riskejä.

Artiklat A37, A38 ja *A39* esittelevät *tietosuojavastaavan* roolin ja aseman. Rekisterinpitäjän tulee nimittää tietosuojavastaava ja antaa tälle valtuudet hoitaa tehtävänsä. Tietosuojavastaavan rooli on hallinnollinen. Hänen tehtävänsä on vahtia, että tietosuoja-asetusta noudatetaan, neuvoa henkilöstöä ja toimia kontaktina valvontaviranomaisiin. Nämä eivät heijastune ohjelmistoarkkitehtuurin tasolle. Tietosuojavastaavalle voi olla tarve antaa oma roolinsa ja näkymänsä järjestelmiin, mikäli näin parhaaksi nähdään.

Asetuksen luvun loppuosa käsittelee standardeja ja sertifikaatteja. *Artikla A40* määrittelee säännöt sille, miten käytantöstandardeja luodaan ja julkaistaan. Hyväksyttyjä käytännesääntöjä seuraamaan voidaan luvittaa ulkopuolisia tahoja, jotka osoittavat kykynsä tehtävään. Standardoiduista toimintatavoista on se hyöty, että rekisterinpitäjien ei tarvise jäädä tulkintakysymysten varaan asetuksen noudattamisessa. Toisaalta niiden seuraamisesta tulee ylimääräistä vaivaa. Paremmin tulevia standardeja ei voi analysoida ennen kuin niitä alkaa ilmentyä. Standardeista voi tulla myös uusia vaatimuksia ohjelmistojen puolelle. *Artiklat A42* ja *A43* määrittelevät sertifioinnin sääntöjä. Asetus itse vain kannustaa jäsenvaltioita ottamaan käyttöön sertifiointimekanismeja asetuksen rooleista. Sertifikaattien hankkiminen voi olla hyvää liiketoiminnan kannalta, mutta niitä ei voi etukäteen kommentoida.

3.1.5 Loput asetuksesta

Olemme käsitelleet nyt tietosuoja-asetuksen periaatteet, rekisteröidyn oikeudet ja rekisterinpitäjän velvollisuudet. Nämä aiemmat asetuksen luvut sisältävät valtaosan asetuksen teknisistä vaatimuksista. Seuraavat asetuksen luvut käsittelevät henkilötietojen siirtoa ulkomaille, valvontaviranomaisia, oikeussuojakeinoja ja seuraamuksia, sekä erityistilanteita ja asetuksen täytäntöönpanoa. Käydään nämä kohdat kuitenkin lyhyesti läpi kattavuuden vuoksi ja katsotaan, onko niissä jotain tapausyrityksen kannalta erikseen huomioitavaa kohtaa. Tämän jälkeen pääsemme varsinaiseen vaatimusmäärittelyyn.

Asetuksen luvussa 5 määritellään säännökset henkilötietojen siirtämiseen EU:n ulkopuolisiin maihin tai kansainvälisille järjestöille. Komissio voi etukäteen päättää, että tietyt ulkopuoliset maat tai tahot täyttävät riittävän tietosuojan tason. Näihin maihin saa siirtää tietoja ilman eri lupaa. Muiden maiden tapauksessa tulee täyttää asetuksessa määritetyt suojatoimet. Tapausyritykselle EU:n ulkopuolella käytettävät tiedot on harvinainen tilanne, mutta hyvinkin mahdollinen. Yksittäisillä palvelimilla olevat järjestelmät on lähes aina EU:n sisällä. Mikäli palvelu on globaali, olisi sillä silti omat palvelimet kullakin mantereella. Tällöin kunkin maanosan henkilötietoja käsiteltäisiin sen sisällä. Tämä pitää kuitenkin huomoida esimerkiksi pilvipalveluita käytettäessä, joissa väärillä asetuksilla tehdyt asennukset voivat huomaamattakin sallia tietojen menemisen EU:n ulkopuolelle. Vaatimus on joka tapauksessa otettava huomioon ja voisi vaikuttaa jossain skenaariossa ohjelmistoarkkitehtuuriinkin.

Seuraavaksi asetuksen luku 6 käsittelee valvontaviranomaisten roolia. Kukin jäsenvaltio nimeää valvontaviranomaisen, jotka toimivat itsenäisesti asetuksen noudattamista vahtien. Valvontaviranomaiset kuitenkin tekevät yhteistyötä, jotta asetusta noudatetaan mahdollisimman yhtenevästi EU:n sisällä. Jäsenvaltioiden tulee antaa valvontaviranomaiselle riittävä valta ja resurssit, ja valvontaviranomaisen tulee olla riippumaton ulkopuolisesta vaikutuksesta. Asetuksen luvussa 7 käsitellään valvontaviranomaisten yhteistyötä ja määritetään tietosuojaneuvosto. Näillä on määrättyjä prosesseja, kuten avunanto ja yhteiset operaatiot. Tietosuojaneuvosto antaa lausuntonsa asiasta, kun valvontaviranomainen on toteuttamassa toimenpiteen. Neuvosto koostuu valvontaviranomaisten johtajista ja EU:n komission edustajista. Sen tehtävä on valvoa asetuksen yhtenäistyyttä. Rekisterinpitäjien – ja tapausyrityksen – kannalta valvontaviranomaiset vaikuttavat toimintaan siten, että järjestelmien ja tilojen on oltava valmiita mahdollisille auditoinneille.

Asetuksen luku 8 määrittää rekisteröityjen oikeussuojakeinot ja seuraamukset. Oikeussuojakeinot ovat oikeuksia rekisteröidyille, mutta ne ovat rekisteröidyn ja valvontaviranomaisen välisiä. Täten vaikutuksia rekisterinpitäjien järjestelmiin ei ole. Oikeudet ovat tehdä *valitus*, *kanne valvontaviranomaista vastaan* ja *kanne rekisterinpitäjää vastaan*. Mikäli henkilölle aiheutuu vahinkoa asetuksen rikkomisen johdosta, on hänellä oikeus saada korvaus rekisreinpitäjältä tai käsittelijältä. *Artikla A83* määrää hallinnollisten sakkojen suuruuden ja määrämisen edellytykset. Sakko asetuksen periaatteiden tai rekisteröityjen oikeuksien rikkomisesta voi maksimissaan olla 20 miljoonaa euroa, tai 4% yrityksen vuotuisesta liikevaihdosta (suurempi vaihtoehdoista). Seuraamukset eivät itsessään ole uusi vaatimus järjestelmien suunnittelulle, mutta ne taas motivoivat suuretkin yritykset noudattamaan asetusta. Jää vielä nähtäväksi, kuinka kevyin perustein ja kuinka suuria sakkoja yrityksille tullaan määräämään. Riski suurelle sakolle tekee tietosuojan suunnittelemisesta ohjelmistoarkkitehtuuriin liiketoiminnallisesti järkevää.

Asetuksen lopuksi luku 9 määrittelee erityistilanteita, joihin jäsenvaltiot voivat itse määrittää säännöt. Esimerkkinä kansallisten henkilötunnuksien käsittelyn määrittää jäsenvaltiot. Luvut 10 ja 11 antavat delegoitua säännösvaltaa komissiolle, kumoavat vanhan tietosuojadirektiivin ja asettavat yleisen tietosuoja-asetuksen voimaantulon.

Näin asetus on siis käyty läpi ja sen kohdat esitelty. Yksityiskohtaisempia tietoja tarvittaessa, esimerkiksi tiettyyn artiklaan liittyvää toimintoa suunnitellessa, on viitattava varsinaiseen lakitekstiin. Olemme kuitenkin valmiit vaatimusmäärittelyyn.

3.2 Vaatimusmäärittely

Luvussa 3.1 käsiteltiin yleinen tietosuoja-asetus läpi sellaisenaan. Seuraavaksi voimme tuottaa näistä tutkielman arkkitehtuuria koskevan vaatimusmäärittelyn. Tutkielman kannalta tärkeät artiklat ovat esitelty luvussa 3.1 ja nyt viitattavissa. Ne toimivat raakadatana, joista saadaan formaalimmat vaatimukset. Vaatimukset ovat artiklojen lisäksi kytköksissä

tapausyrityksen ominaisuuksiin, jotka taas esiteltiin luvussa 2. Vaatimusmäärittelyn tavoitteena onkin kytkeä nämä ominaisuudet artiklojen lakiin ja siten tiivistää nämä erilliset konseptit yhdeksi käsiteltäväksi joukoksi. Vaatimuksia vasten voimme myöhemmin arvioida tutkielman arkkitehtuuria ja siihen suunniteltuja ratkaisuja.

Vaatimusmäärittely (Engl. *Requirements Engineering, RE*) on perinteinen ohjelmistokehityksen dokumentti, jossa kuvataan suunniteltavan järjestelmän toiminnallisuutta, laajuutta, rajoituksia ja tiettyjä vaatimuksia. IEEE on tehnyt vaatimusmäärittelystä kansainvälisen standardin ISO/IEC/IEEE 29148:2011 [11]. Dokumentti toimii viitteenä arvioille järjestelmästä ja referenssinä, jonka mukaan järjestelmä kehitetään. Tämä strategia ei sovi tapausyrityksen ketterän ohjelmistokehityksen tapaan – iteratiivinen ja inkrementaalinen kehittäminen ei ole mielekästä, jos järjestelmän vaatimukset ovat etukäteen lukittu. Toisaalta yleisen tietosuoja-asetuksen esittämät vaatimukset eivät ole neuvoteltavissa, joten niiden osalta voidaan suunnitella ratkaisuja ennalta.

Ketterän ohjelmistokehityksen ja vaatimusmäärittelyn suhdetta ovat tutkineet esimerkiksi Paetsch et al. artikkelissaan [38] vuonna 2003. Haasteena nähdään juuri ketterien menetelmien keskittyminen keskustelevaan yhteistyöhön ja vaatimusmäärittelyn keskittyminen dokumentointiin. Yhtäläisyyksiä lähestymistapojen välillä kuitenkin on ja molempien strategioiden tavoite on sama. Perinteisessä RE-mallissa kaikki vaatimukset pyritään keräämään etukäteen, kun taas ketterässä tarpeen mukaan.

Kun molempia RE-strategioita tarkastellaan taas tämän tutkielman kontekstissa, huomataan seuraavat seikat. Asetuksen vaatimuksien löytämiseksi ei tarvitse tehdä (enempää) työtä, ne eivät muutu, eikä vaatimuksia tarvitse priorisoida. Näillä perusteluilla vaatimusmäärittely on mielekästä tehdä etukäteen ja arkkitehtuuria on mielekästä suunnitella asetuksen kannalta sopivaksi. Huomioidaan kuitenkin, että asetus jättää monessa kohdassa tulkintavaraa, kuten käyttää useassa paikassa termiä ”asianmukaiset toimenpiteet”. Asetusta voidaan myöhemmin tarkentaa ja EU voi esitellä sitovia standarditoimintatapoja henkilötietojen käsittelyn suhteen.

IEEE:n standardin mukaista vaatimusmäärittelydokumenttia ei kuitenkaan ole tutkielman parametrit huomioiden mielekästä tehdä. Työssä tehtävä vaatimusmäärittely ei ole kattava määrittely minkään *tietyn* järjestelmän toteutuksesta. Tehtävä määrittely on siis uudelleenkäytettävä osajoukko kunkin järjestemän kokonaisvaatimuksista. Tapausyrityksen kehitystyö ei käytä perinteistä vaatimusmäärittelyä, joten asetuksen vaatimuksien sovittaminen siihen muottiin ei ole mielekästä. Keskitytään asetuksen vaatimusten listauksessa käyttötapauksiin ja teknisiin vaatimuksiin. Tutkielmassa käytetään määritellyistä vaatimuksista notaatiota *(V1: nimi)*, jossa numero on vaatimuksen esittelyjärjestyksen numero ja nimi vaatimusta lyhyesti kuvaava termi.

Asetuksen vaatimukset jakautuvat kolmeen luokkaan: teknisiin, organisatorisiin ja juridisiin vaatimuksiin. Tekniset vaatimukset ovat tutkielman tärkeimmässä osassa. Ne sisältävät sellaiset vaatimukset, joihin voidaan vastata järjestelmä oikein suunnittelemalla. Organisatoriset vaatimukset koskevat rekisteriä pitävän yrityksen rooleja ja toimintatapoja, kuten tietosuojavastaavan määrittäminen. Niihin lasketaan tässä myös rekisterejä koskevat dokumentointivelvoitteet. Juridiset vaatimukset taas ovat sellaisia, jotka eivät riipu teknisistä ratkaisuista ja ovat usein rekisterikohtaisia. Esimerkiksi henkilötietojen käsittelyn tarkoitukset. Luokittelu ei ole tarkasti rajattu ja vaatimuksilla on useita mäkökulmia. Tutkielmassa tehtävä vaatimusmäärittely onkin *arkkitehtuurin* vaatimusmäärittely. Tällöin muita vaatimuksia tarkastellaan siitä kannalta, miltä ne voivat vaikuttaa järjestelmien arkkitehtuuriin.

3.2.1 Tekniset vaatimukset

Asetuksen vaatimuksia tarkastellessa on huomioitava sen periaatteista *osoitusvelvollisuus*, joka on peräisin artiklasta *A5(2)*. Tämä ei liity yhteen tiettyyn tekniseen kohtaan, joten liitetään osoitusvelvollisuus attribuutiksi jokaiseen alla numeroituun vaatimukseen. Järjestelmän kehittäjien on siis voitava osoittaa kunkin vaatimuksen täyttäminen. Tämä tarkoittaa dokumentointia ja valmiutta vastata kysymyksiin mistä tahansa järjestelmän aspektista.

Tekniset vaatimukset kuvataan alla. Tiivistetään ne lisäksi *käyttäjätarinoiksi* [7], kun se on relevanttia vaatimuksen kannalta. Käyttäjätarinoiden arvo on se, että ketteriä ohjelmistokehitysmenetelmiä käyttävät tiimit voivat ottaa ne arviointiin ja suunnitteluun sellaisenaan. Tällöin vaatimusmäärittelystä on hyötyä, vaikkei seuraisi tutkielmassa tehtävää arkkitehtuuria.

(V1: järjestelmän tietosuoja) on ensimmäinen vaatimus, joka nostetaan esiin. Se perustuu artikloihin *A25* tietosuojasta ja *A32* käsittelyn turvallisuudesta. Vaatimus on vaikeampi ratkaista kuin selkeät käyttäjätarinat, lisäksi se liittyy kaikkiin järjestelmän komponentteihin. Vaatimus rakentuu alakohdista, jotka luetellaan seuraavaksi. Alakohdat johtuvat tietyn artiklan tietystä kohdasta. *(V1.1: asianmukaiset toimet tietosuojalle)* johtuu kohdasta *A25(1)*. Järjestelmällä tulee olla seuraavat ominaisuudet *A32(1):n* mukaan: *(V1.2: luottamuksellisuus)*, *(V1.3: eheys)*, *(V1.4: käytettävyys)* ja *(V1.5: vikasietoisuus)*. Lisäksi artiklassa vaaditaan *(V1.6: nopea palautuskyky)* ja *(V1.7: tietosuojan testausmenettely)*. Erityisesti arkkitehtuurissa huomioitavia teknisiä seikkoja ovat tiedonsiirto, varastointi ja pääsynhallinta.

(V2: tietojen minimointi) on artiklan *A5(1c)* samannimistä periaatetta vastaava vaatimus. Järjestelmässä tulee kerätä kutakin tarkoitusta varten minimaaliset henkilötiedot ja kunkin käsittelyn laajuus tulee olla mahdollisimman pieni. Artikla *A25(1)* tarkentaa

vaatimusta seuraavasti: *"oletusarvoisesti käsitellään vain käsittelyn kunkin erityisen tarkoituksen kannalta tarpeellisia henkilötietoja"*. Vaatimus on osin juridinen, koska käsittelyn tarpeeseen ei voi ottaa teknisellä tasolla kantaa. Hajautetussa arkkitehtuurissa on voitava määrittää kutakin tarkoitusta varten oma projektionsa koko henkilödatasta, jotta käsittely rajoittuu minimijoukkoon.

(V3: suostumuksen hallinta) liittyy järjestelmiin, jotka käsittelevät henkilötietoja rekisteröidyn suostumuksen mukaan (*A6* mukaisesti). Suostumuksen hallinnassa on kolme teknistä kohtaa: a) suostumuksen kerääminen, b) suostumuksen osoittaminen ja c) vanhemman suostumus. Artikla *A7* erittelee suostumuksen edellytykset, koskien suostumuksen käyttöliittymää, tietomallia ja suostumuksen perumista. *(V3.1):n* kerääminen on vaatimus käyttöliittymille siitä, että suostumuksen kysyminen on *"esitettävä selvästi erillään muista asioista helposti ymmärrettävässä ja saatavilla olevassa muodossa selkeällä ja yksinkertaisella kielellä"*. Kysymisen ohessa tulee tarjota *A13* mukaiset tiedot käsittelystä. Suostumuksen peruuttaminen tulee olla yhtä helppoa kuin antaminen.

(V3.2) on suostumuksen hallinnan tietomallia koskeva osa. Osoitusvelvollisuuden vuoksi on voitava osoittaa käyttäjän antaneen suostumuksensa. Siipä tallennetaan ajankohta, jona kukin käyttäjä on suostumuksen antanut. Digitaalisia allekirjoituksia voidaan hyödyntää tässä, jos halutaan aukoton osoittaminen. *(V3.3)* listataan vielä erikseen, koska lasten[1] henkilötietojen keräämiseen tulee saada vanhempien suostumus. Artikla *A8* vaatii *kohtuulliset toimenpiteet* vanhemman tunnistamiseksi. Kokonaisuudessaan *(V3)* täyttäminen vaatii kaikkien näiden kohtien ratkaisemisen.

(V4: henkilötietojen jäljitettävyys) syntyy artiklan *A30*, artiklan *A5.2*, sekä muiden rekisteröityjen oikeuksien johdosta. Osoitusvelvollisuuden vuoksi on voitava näyttää jälkikäteen, että asetusta on noudatettu henkilötietojen käsittelyssä. Voidaksemme osoittaa, että rekisteröidyn oikeuspyyntöjä on käsitelty oikein ja ajoissa, on niiden toteuttaminen lokitettava. Rekisteröidyllä on oikeus nähdä, mille tahoille henkilötiedot on luovutettu tai julkistettu. Oikaisu- ja poistamispyynnöt tulee välittää eteenpäin, mikäli tietoja on luovutettu. (Luovutuksen itsensä tarpeellisuus ja laillisuus ovat tapauskohtaisia kysymyksiä.) Erikseen asetuksessa mainitaan siirrot EU:n ulkopuolelle. Tarvitsemme siis lokin kaikista erillisistä henkilötiedon käsittelytapahtumista, oikeuspyyntöjen toteuttamisista ja luovutuksista. Kustakin lokivaatimuksesta merkitään oma alakohta *(V4)*:lle, jotta niitä voidaan paremmin tarkastella osina. Vaatimus sisältää siis seuraavat alakohdat: *(V4.1: loki käsittelytapahtumista)*, *(V4.2: loki oikeuspyyntöjen toteuttamisista)*, *(V4.3: loki siirroista EU:n ulkopuolelle)*, sekä *(V4.4: loki tietojen luovutuksista)*.

(V5: rekisteröidyn pääsy tietoihin) täyttää artiklojen *A15* ja *A20* velvoitteet. Rekisteröidyllä on oltava pääsy kaikkiin henkilötietoihin, joita hänestä on kerätty. *(V4):n* mukaan tallennetut luovutuskohteet tulee myös ilmoittaa. *A20:n* siirron järjestelmien välillä

[1] Alle 13-16 vuotias, jäsenmaat tarkentavat lainsäädännössä.

voi toteuttaa minimissään sillä, että henkilötiedot on ladattavissa jäsennellyssä koneluettavassa muodossa. Molemmat velvoitteet ratkeavat liki samalla työllä: henkilötietojen tulee olla muodossa, jossa ne ovat noudettavissa. Vaatimus sisältää tavan, jolla rekisteröity voi esittää pyyntönsä. Vaatimus jakautuu siis seuraaviin alakohtiin: *(V5.1: rekisteröidyn tietojen katselu)*, *(V5.2: katselupyynnön käyttöliittymä)* ja *(V5.3: pääsy tietoihin koneluettavassa muodossa)*. *(V5.1)* ja *(V5.2)* kattavat artiklan *A15*, josta johtuen tietojen katselu sisältää myös artiklan mukaiset tiedot rekisteröidyn oikeuksista. *(V5.3)* vastaa *A20* velvoitteeseen toistaiseksi. Mikäli *A20(2):ssa* esitetyt tekiset mahdollisuudet aukeavat esimerkiksi yleisten standardien myötä, vaatimusta on muutettava.

(V6: rekisteröidyn tietojen oikaisu) kattaa artiklan *A16* velvoitteen. Rekisteröidylle tulee tarjota mahdollisuus korjata ja päivittää henkilötietonsa. Oikaisupyynnön tulee propagoitua kaikille tahoille, joille henkilötietoja on mahdollisesti luovutettu. Vaatimus sisältää myös tavan, jolla rekisteröity voi esittää pyyntönsä. Oikaistut tiedot tulee huomioida myös varmuuskopioiden kanssa, ettei oikaisua palauteta takaisin vanhaan muotoonsa. Vaatimus sisältää siis kaksi kohtaa: *(V6.1: mahdollisuus päivittää rekisteröidyn tiedot)* ja *(V6.2: oikaisupyyntöjen käyttöliitymä)*.

(V7: rekisteröidyn tietojen poistaminen) vastaa artiklan *A17* oikeutta tulla unohdetuksi. Henkilötiedot pitää tallentaa muodossa, jossa kaikki yhteen rekisteröityyn liittyvät tiedot saadaan poistettua. Poistamispyynön pitää myös propagoitua *(V4):n* tahoille. Vaatimus sisältää myös tavan, jolla rekisteröity voi esittää pyyntönsä. Pyyntöön vastaamisen ei tarvitse olla automaattista ja pyyntö voidaan kyseenalaistaa ennen täyttämistä. Toinen puoli henkilötietojen poistamisesta on käsittelyn päättyminen. Juridinen vaatimus artiklasta *A5(1e)*, ettei henkilötietoja saa säilöä kauempaa kuin tarpeellista, on mahdollistettava teknisesti. Poistettavat tiedot tulee huomoida myös varmuuskopioiden kanssa, samoin kuin *(V6):ssa*. Jaotellaan tämäkin vaatimus kahteen vastaavaan alivaatimukseen: *(V7.1: tietojen poistamismahdollisuus)* ja *(V7.2: poistamispyyntöjen käyttöliittymä)*.

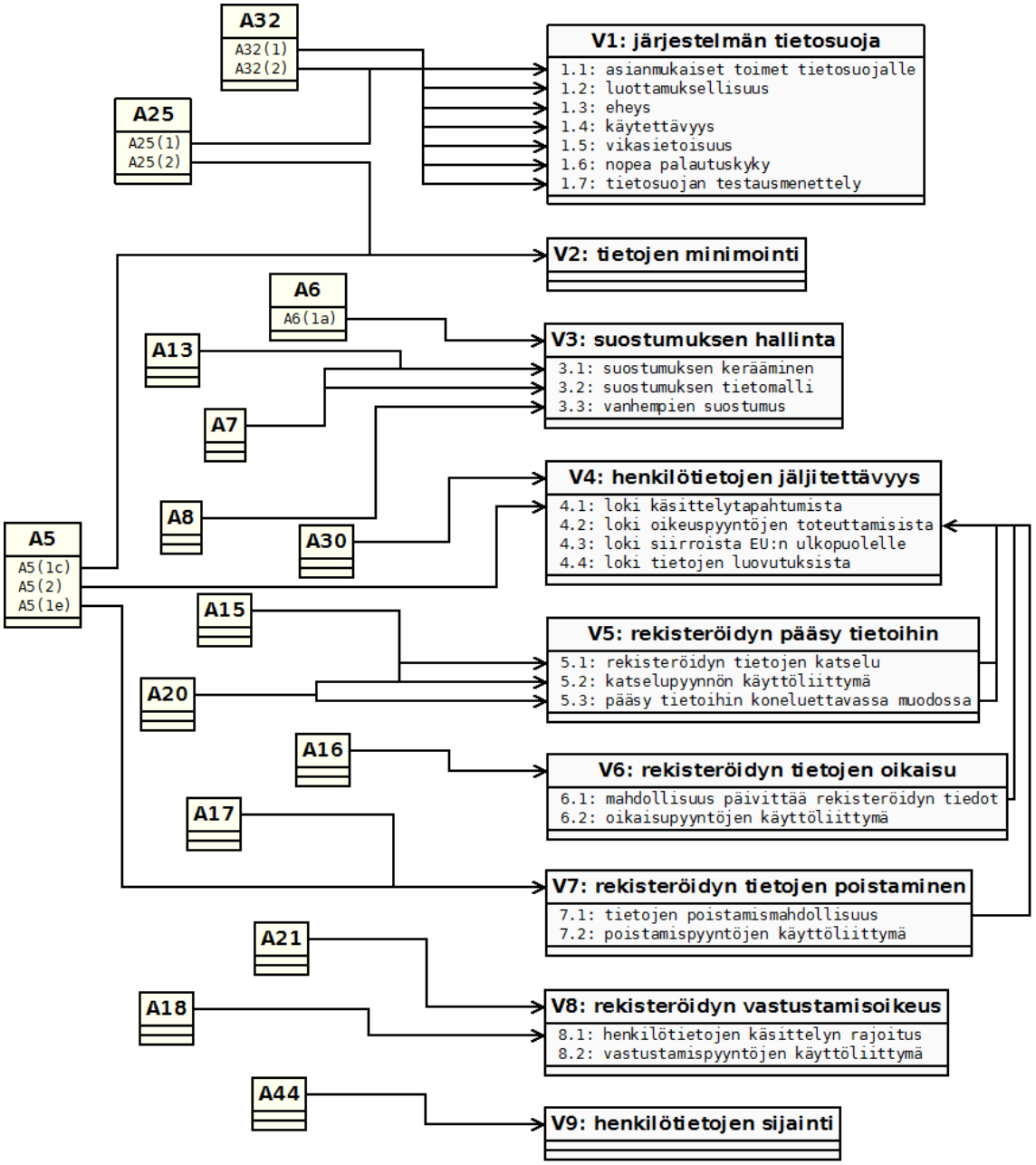
(V8: rekisteröidyn vastustamisoikeus) on määritelty artiklassa *A21*. Rekisterinpitäjän on tarjottava mahdollisuus esittää vastustamispyyntö. Mikäli pyyntö tulkitaan validiksi, on järjestelmässä oltava mahdollisuus asettaa henkilötieto rajoitetuksi. Rajoitettua henkilötietoa ei voi käsitellä, muokata tai poistaa. Rajoituksen käsittelylogiikka on määritelty artiklassa *A18*. Tämäkin vaatimus koostuu kahdesta alivaatimuksesta. Tarvitaan sekä tietomallissa tapa käsitellä rajoitukset *(V8.1: henkilötietojen käsittelyn rajoitus)*, että mahdollisuus esittää vastustuspyyntö *(V8.2: vastustamispyyntöjen käyttöliittymä)*.

(V9: henkilötietojen sijainti) vaatii huomioimaan sen, missä maassa henkilötiedot sijaitsevat. Artiklan *A44* ja asetuksen viidennen luvun mukaan datan siirtäminen EU:n ulkopuolelle tuo lisäedellytyksiä. Juridisten seikkojen lisäksi tästä seuraa tekninen vaatimus, että järjestelmää suunniteltaessa on tiedettävä missä data fyysisesti sijaitsee.

Käyttäjätarinat tiivistävät asetuksen rekisteröidylle ilmenevät vaatimukset kehityksen suunnittelussa auttavaan muotoon. Merkitään käyttäjätarinoita tutkielmassa lyhenteellä *KT* ja numerolla.

Käyttäjätarinat:
KT1: Rekisteröitynä voin antaa ja peruuttaa suostumukseni tietojeni käsittelylle (V3a).
KT2: Rekisteröidyn vanhempana voin antaa ja peruuttaa suostumukseni lapseni tietojen käsittelylle (V3c).
KT3: Rekisteröitynä voin ladata henkilötietoni ja nähdä mille tahoille ne on luovutettu (V5).
KT4: Rekisteröitynä voin oikaista henkilötietoni, jotta käsittely tapahtuu oikealla datalla (V6).
KT5: Rekisteröitynä voin poistaa henkilötietoni, jotta ne eivät jää rekisterinpitäjälle (V7).
KT6: Rekisteröitynä voin vastustaa tietojeni käsittelyä, jotta lainvastaista käsittelyä ei tapahdu (V8).

Kukin määritelty vaatimus perustuu tiettyihin asetuksen artikloihin, kuten edellä on kuvattu. Luvussa määritellyt vaatimukset ovat tiivistettynä kuvassa *(Kuva 2)*, edellä. Tästä ilmenee erityisesti mistä asetuksen osasta kukin vaatimus on johdettu. Kuvasta saadaan myös kokonaiskuva tehdystä määrittelystä.



KUVA 2: Yleisen tietosuoja-asetuksen artiklojen linkitys luvun vaatimusmäärittelyyn.

3.3 Vaatimukset tapausyrityksen kannalta

Edellisessä osiossa määriteltiin tietosuoja-asetuksen tuomat tekniset vaatimukset arkkitehtuurille. Tarkastellaan näitä vaatimuksia seuraavaksi tutkielman tapausyrityksen ja erityisesti luvun 2 ominaisuuksien näkökulmasta.

(V1: järjestelmän tietosuoja) on laajin asetuksen vaatimuksista ja se on huomioitava arkkitehtuurissa monella tasolla. Tämä ilmenee myös tapausyrityksen ominaisuuksien kannalta katsottaessa: monissa ominaisuuksissa on tietosuojaan liittyvä osa. *(O1: jaon asiakkuuksien välillä)* kuvaama asiakasprojektimalli on siitä työläs tietosuojan kannalta,

että jokaiselle asiakkuudelle on perusteltava tietosuojan tärkeys ja tietoturvaratkaisujen hinta.

(V1) näkyy ominaisuuden *(O3: projektin koon vaihtelun)* kannalta siinä, että pienimmissäkin projekteissa on panostettava tietosuojaan laajasti. *(O7: uudelleenkäytettävyys)* on *(O3)* johdosta tärkeässä osassa tässä. Vaatimus ei kuitenkaan ole yksittäinen toteutettava palvelu ja siksi se on haastava myös uudelleenkäytettävyyden näkökulmasta. *(V1):een* liittyvät arkkitehtuuriratkaisut sisältävät myös tietoliikennettä, infrastruktuuria ja toteutuskäytäntöjä.

(O8: tukitoimintaan ja palvelimien ylläpitoon) vaikuttavimpia vaatimuksista on myös *(V1)*. Palvelimien pääsynhallinta on toteutettava uskottavasti, samoin tietoturvan ajantasalla pitäminen. Nämä eivät kuitenkaan ole erityisesti henkilötietojen aiheuttamia kohtia, vaan yleisesti hyvän ylläpidon ominaisuuksia. *(O9: henkilötietojen levinneisyys)* hankaloittaa *(O8):n* toteuttamista *(V1):n* mukaisesti siten, että turvattavia alueita on enemmän.

(V2: tietojen minimointi) on toinen haastava vaatimus arkkitehtuurin kannalta. On projektikohtaista, mitä *minimaalinen henkilötietojen määrä* tarkoittaa kussakin kontekstissa. *(O9: henkilötietojen levinneisyys)* on osin ristiriidassa vaatimuksen kanssa, koska on voitava osoittaa kaikkialla käytettävän mahdollisimman vähän henkilötietoja. Luonnollisesti joissain tapauksissa laajasti levinneet henkilötiedot ovat tarpeellisia järjestelmän kannalta, mutta tällöinkin joka paikassa on käsiteltävä mahdollisimman vähäisiä tietoja. Ominaisuudesta seuraa tarve siis aiemmin mainituille projektioille osatietoihin yksittäisestä henkilötiedosta.

(V2):sta seuraa myös haaste *(O5: staging-ympäristön vaatimuksien)* kannalta. On voitava perustella, miksi henkilötiedot olisivat tarpeellisia testiympäristöissä. Perustelemisen helpottamiseksi voidaan käyttää esimerkiksi pseudonymisointia. Asetuksen mukaan täysin anonymisoidut (entiset) henkilötiedot eivät ole henkilötietoja, mutta anonymisoinnin osoittaminen on myöskin haastavaa. Anonymisointi tarkoittaisi, että henkilötietoja ei voitaisi mitään yhdistävää lisätietoa käyttämälläkään enää linkittämään rekisteröityyn.

(V3: suostumuksen hallinta) on suoraviivaisempi vaatimus kuin aiemmat. Vaatimus kuuluu samaan joukkoon vaatimuksien *(V5: rekisteröidyn pääsy tietoihin)*, *(V6: rekisteröidyn tietojen oikaisu)*, *(V7: rekisteröidyn tietojen poistaminen)* ja *(V8: rekisteröidyn vastustamisoikeus)* kanssa. Kullakin näistä on omat erityispiirteensä, mutta yhtenäistä on rekisteröidylle tarjottava palvelu. Tapausyrityksen ominaisuuksista tästä korostuu *(O7: uudelleenkäytettävyys)*, jonka kannalta näiden vaatimusten ratkaisua pitäisi voida mahdollisimman hyvin käyttää eri projekteissa. *(O4: projektien ulkopuoliset investoinnit):n* kannalta on ratkaistava strategia, jolla uudelleenkäytettävät ratkaisut voidaan kehittää ilman kohtuutonta sijoitusta. Sama käyttöliittymä voinee ratkaista kaikki tietosuoja-asetukseen

liittyvät käyttöliittymälliset vaatimukset, joten vaatimuksia ei yleisesti kannata tarkastella eristettynä yksittäin.

(V4: henkilötietojen jäljitettävyys) on pohjimmiltaan yksinkertainen vaatimus muodostaa loki henkilötietojen käsittelytoimista. Suuri *(O9: henkilötietojen levinneisyys)* vaikeuttaa tätä. Erityisesti vanhojen järjestelmien saattaminen asetuksen mukaiseksi on vaikeaa, kun henkilötietojen kaikki käsittelypaikat eivät ole välttämättä tiedossa. Itse lokin ylläpito on oma eriytettävä kohtansa, jolla voidaan saada *(O7)* mukaista etua projektien välillä. *(O8: tukitoiminta ja palvelimien ylläpito)* on lokitusvaatimusten kannalta erikseen huomioitava. Lokivaatimukset tuntuvat koko arkkitehtuurin läpi, mikä vaatii toimivan ratkaisun.

(V5: rekisteröidyn pääsy tietoihin), *(V6: rekisteröidyn tietojen oikaisu)* ja *(V7: rekisteröidyn tietojen poistaminen)* ovat luonteeltaan samankaltaisia. Näissä kaikissa on korostuu tarve saada kaikki rekisteröidyn tiedot erikseen järjestelmästä hallintaan. Relaatiotietomallista tulisi saada koko yhtä rekisteröityä koskeva graafi relaatioita. *(O9)* taas vaikeuttaa tätä, kun esimerkiksi eri palvelimilla oleviin henkilötietoihin kaikkiin tulee päästä kiinni. Ratkaisussa näihin on otettava huomioon jälleen *(O7):n* näkökulma. Haastavaa siltä kannalta on se, että tallennettavat henkilötiedot ovat usein eri järjestelmissä erilaiset – tietomallit ovat sovelluskohtaisia.

(V6) ja *(V7)* aiheuttavat myös aiemmin mainitun tarpeen muutoslokista. Arkkitehtuuria suunnitellessa voidaan arvioida onko *(V4):n* kattava käsittelytoimien loki riittävä kattamaan myös tämän käyttötarkoituksen. Muutos- ja poistoloki on vaatimus siksi, että yleisesti tehtävät varmuuskopiot järjestelmistä sisältävät tulevaisuudessa muuttuvat tiedot. Mikäli varmuuskopio palautetaan, on muutoslokia tarkkailtava ja tehtävä vastaavat muutokset palautettuun dataan.

(V8: rekisteröidyn vastustamisoikeus) on läheinen *(V3: suostumuksen hallinnan)* kanssa. Haettaessa henkilötietoja on tarkastettava sekä suostumuksen olemassaolo, että rajoituksen puuttuminen. Rajoitus estää lisäksi tietojen poistamisen.

(V9: henkilötietojen sijainti) liittyy eniten ominaisuuksiin *(O8: tukitoiminta ja palvelimien ylläpito)*, sekä *(O5: stating-ympäristön ominaisuudet)*. Palvelimien ja muun järjestelmien infrastruktuurin sunnittelussa tulee huomioida missä maassa palvelut sijaitsevat. Testiympäristöt ovat myös saman vaatimuksen piirissä, mikäli niissä henkilötietoja käsitellään. Muut ohjelmistokehityksessä käytettävät palvelut, kuten keskusteluohjelmat tai projektien dokumentaatiojärjestelmät voivat myös sijaita EU:n ulkopuolella. Näistä ongelmista vältytään lähinnä ohjeistuksella kehittäjille: henkilötietoja ei tule käsitellä huolimattomasti. *(O2: teknologiariippuvaisuus)* liittyy vaatimukseen siten, että erityisesti jo olemassaolevat järjestelmät voivat olla riippuvaisia pilvipalveluntarjoajista. Useimmilla pilvipalveluilla on valittavissa EU:n sisällä olevat palvelimet.

Ominaisuuden *(O6: vanhojen järjestelmien päivittäminen)* kannalta, tärkeintä on kattaa vanhoissa järjestelmissä pakolliset vaatimukset. Sellaisissa projekteissa, jotka ovat käytössä, mutta eivät aktiivisessa kehityksessä, voidaan tehdä kompromisseja arkkitehtuurin laadun kanssa. Näissä tapauksissa lain täyttäminen on tärkein kriteeri. Luonnollisesti helpommin päivitettävät ratkaisut ovat parempia kuin vaikeasti, tämän ominaisuuden kannalta.

3.4 Yhteenveto

Luvussa on nyt käsitelty yleinen tietosuoja-asetus lähdemateriaalina ohjelmistoarkkitehtuurille. Asetus käytiin läpi kokonaisuudessaan ja siinä esitetyistä vaatimuksista tehtiin ketterä vaatimusmäärittely. Vaatimusmäärittelystä tiivistettiin vielä käyttäjätarinat. Tutkielman arkkitehtuuri perustuu näihin vaatimuksiin vastaamiseen, luvussa 2 esitellyn tapausyrityksen ominaisuuksien kannalta.

Aliluvussa 3.1 esiteltiin asetuksen sisältö referoituna ja pohdittiin sen vaikutuksia. Asetus jaoteltiin yleisiin säännöksiin, periaatteisiin, rekisteröidyn oikeuksiin, rekisterinpitäjän velvollisuuksiin, ja loppuun asetuksesta. Tutkielman kannalta tärkeimmät artiklat saivat suurimman huomion, muut kohdat esiteltiin pintapuolisesti.

Vaatimusmäärittelyssä tavoite oli eritellä tarkalleen, mitä konkreettisia teknisiä vaatimuksia asetuksen tekstimassasta syntyy. Vaatimusmäärittelyn kohde on tapausyrityksen osaamisalan kaltaisten verkkosovellusprojektien ohjelmistoarkkitehtuuri. Vaatimusmäärittely tehtiin aliluvussa 3.2. Siinä asetuksesta johdetut vaatimukset ovat vielä listattuna alla taulukossa *(Taulukko 2)*.

Taulukko 2: Yleisestä tietosuoja-asetuksesta johdetut tekniset vaatimukset.

Vaatimukset	
V1	järjestelmän tietosuoja
V2	tietojen minimointi
V3	suostumuksen hallinta
V4	henkilötietojen jäljitettävyys
V5	rekisteröidyn pääsy tietoihin
V6	rekisteröidyn tietojen oikaisu
V7	rekisteröidyn tietojen poistaminen
V8	rekisteröidyn vastustamisoikeus
V9	henkilötietojen sijainti

Aliluvussa 3.3 käytiin läpi määriteltyjä vaatimuksia tapausyrityksen kannalta. Näistä löydettiin erilaisia haasteita, joita tutkielman tavoitteena on ratkaista. Tietosuoja-asetuksen esittelyn ja sen vaatimusten määrittelyn jälkeen voidaan edetä esittelemään tutkielman

arkkitehtuuri. Tätä ennen seuraavassa luvussa käydään läpi aiempaa tutkimusta tietosuojaa koskien ja selvitetään saadaanko näistä uusia ajatuksia ratkaisuille.

Luku 4
Liittyvä tutkimus

Tutkielman aiheeseen liittyviä julkaisuja. Käsitellään tarkimmin ohjelmistoarkkitehtuuria koskevaa tutkimusta.

Yleinen tietosuoja-asetus on mielenkiintoinen sekä oikeustieteelliseltä kannalta, organisaatioiden johtamisen kannalta, että tekniseltä kannalta. Yrityksille kaikki nämä ovat arvoikkaita lähestymistapoja, ja tieteelle kaikissa on tutkittavaa. Jotain asetuksen osa-alueita koskevia julkaisuja on lukuisia. Myös yksityisyritykset perustavat palveluita asetuksen vaatimusten ratkaisemiseksi (esimerkiksi [37]).

Julkaisuja on myös asetuksen vaikutuksista eri tieteenaloihin: sekä ohjelmistoalalla, että muilla (esimerkiksi Hordern 2016 [20], Goodman ja Flaxman 2016 [14]). Lisäksi on yleisemmällä tasolla olevia tietosuoja-asetuksen vaatimusten tarkasteluja, kuten artikkeli, jonka Tikkinen-Piri et al. julkaisivat 2017 [45]. Tutkielman vaatimusmäärittelyssä tultiin samankaltaisiin lopputuloksiin.

Tässä luvussa käsitellään erityisesti ohjelmistoarkkitehtuurin kannalta mielenkiintoisia julkaisuja, joista pyritään saamaan hyötyä luvun 5 työssä. Aliluvussa 4.1 esitellään privacyTracker-ohjelmistokehys ja aliluvussa 4.2 eritellään anonyymiyden tasoja. Näitä rinnastetaan tutkielman vaatimuksiin, sekä tapausyrityksen ominaisuuksiin. Aliluvussa 4.3 esitellään *Privacy As A Service* -kehys tietosuojan hallintaan.

4.1 privacyTracker

Gjermundrød et al. julkaisivat vuonna 2016 artikkelin *privacyTracker: A Privacy-by-Design GDPR-Compliant Framework with Verifiable Data Traceability Controls* [13]. Artikkelissa esitellään ohjelmistokehys, *privacyTracker*, jonka avulla tietojärjestelmiin saataisiin sisäänrakennettua tietosuojaa. Kehyksen luvataan olevan yleisen tietosuoja-asetuksen mukainen. Tässä aliluvussa tutustutaan privacyTrackeriin ja tutkitaan saadaanko siitä jotain käyttöön tutkielman ratkaisuun.

Artikkelissa peräänkuulutetaan vastauksia seuraaviin henkilötietodataa koskeviin vaatimuksiin: 1) henkilötiedon hankinta-ajan ja kerääjätietojen tarkasti asettaminen, 2) mahdollisuus toimittaa luettelo kaikista alkuperäisen datan hallitsijoista, 3) määrittää muutokset alkuperäiseen datan, 4) määrittää datan tarkkuus ja oikeellisuus, ja 5) mahdollisuus

hallita datan elinkaarta. PrivacyTracker lähestyy näitä tarjoamalla teknisen ratkaisun ja siksi jakaa tavoitteen tämän tutkielman kanssa. Esitelty ratkaisu pääasiassa vastaa vaatimuksiin *(V4: henkilötietojen jäljitettävyys)* ja *(V5: rekisteröidyn pääsy tietoihin)* omalla komponentillaan.

PrivacyTrackerin lähtökohta on muodostaa yksittäinen komponentti, jossa järjestelmässä käsitellään henkilötietoja. Tämä on alustavasti linjassa luvussa 2 määritellyn tutkielman ympäristön ominaisuuden *(O9: henkilötietojen levinneisyys)* kanssa. Kehys koostuu omaksi entiteetikseen määritellystä asiakastietueesta (*Customer Record*), ja kolmesta niitä käsittelevästä komponentista: tiedonkeruu-, jakelu- ja jäljitettävyysmoduulista. Kehykseen sisällytetään myös näiden toimintaa hallitseva hallintamoduuli, mutta sitä ei artikkelissa määritelty tarkemmin. Tarkastellaan seuraavaksi syvemmin kehyksen ehdottaman asiakastietueen rakennetta ja sen vaikutuksia tietosuoja-arkkitehtuuriin.

Yksittäinen henkilötieto on moninkertaisesti linkitetty lista asiakastietueita. Asiakastietueet ovat varsinaiset tiedon varasto, ja kokonaisuus syntyy niiden metadatan linkeistä. Kehyksen logiikka keskittyykin asiakastietueen metadatan hallintaan. Metadata on joukko pakollisia attribuutteja, joilla tietosuojan vaatimuksiin saadaan vaikutettua. Tietuella on lisäksi varsinainen hyötykuorma, johon kehystä käyttävä järjestelmä tallentaa tiedot. Metadata-attribuutit jakautuvat kolmeen luokkaan: tietueen tunnisteisiin, datan jäljitettävyyteen ja kryptografisiin hallintatietoihin. Käydään nämä attribuutit seuraavaksi läpi. Ne luetellaan myös taulukossa *(Taulukko 3)*.

Tietueen tunnistetiedot ovat seuraavat: *URI (Unique Resource Indentifier)* identifioi tietueen ja on uniikki kaikkien kehystä käyttävien yritysten kesken. *Sähköposti* yksilöi henkilötietojen omistajan, tosin artikkelissa ehdotetaan tilalle myös sähköistä allekirjoitusta. Voidaankin kyseenalaistaa, onko sähköposti lainkaan oikea tapa yksilöidä henkilöitä, mutta se on sivuseikka kehyksen meriittien kannalta. Tietueella on *Luomisaika (globaali) ja Luomisaika (lokaali)*. Nämä vastaavat itse tietueen tallentamisaikaa ja koko henkilötiedon globaalia tallentamisaikaa. *Vanhenemisaika* on ajankohta jonka jälkeen tietueen arvoja ei pidetä enää validina.

Jäljitettävyysattribuutit ovat linkkejä asiakastietuiden välillä. Jokaisella henkilötiedon tietueella on viite ensimmäiseen tietueeseen, joka oli tämän henkilötiedon alkuperäinen tallennustapahtuma. Kutsutaan tätä *juurireferenssiksi*. Toinen linkki on viite taaksepäin listalla siihen asiakastietueeseen, jonka tiedoista nykyinen tietue luotiin datan siirrossa. Nimetään se *referenssiksi taakse*. Tietueella on vielä lista viitteitä eteenpäin kaikkiin niihin tietueisiin, jotka ovat luotu nykyisen tietueen pohjalta. Kutsutaan tätä *referensseiksi eteenpäin*.

Taulukko 3: privacyTrackerin asiakastietueen metatiedot.

Luokka	Attribuutti
Tunnistetiedot	URI
	Sähköposti
	Luomisaika (globaali)
	Luomisaika (lokaali)
	Vanhenemisaika
Jäljitettävyys	Juurireferenssi
	Referenssi taakse
	Referenssit eteenpäin
Kryptografiset	Kopio lähdetietueesta
	Allekirjoitus

Kryptografiset attribuutit sisältävät *kopion lähdetietueesta* ja *allekirjoituksen*. *Kopio lähdetietueesta* on täydellinen ja muuttumaton kopio alkuperäisestä tietueesta, johon referenssi taakse viittaa. Tällä saadaan tallennettua lähtötila, johon tietueeseeen tehtyjä muutoksia voidaan verrata. Kopio sisältää koko alkuperäisen viestin, jotta viitatun tietueen tilasta jää luomishetken tilanne talteen. Näin muutokset eivät heijastu historiatietoon. *Allekirjoitus* sisältää tarkistekoodin koko tietueesta lukuunottamatta *kopiota lähdetietueesta*. Tarkistekoodi allekirjoitetaan tietuetta muodostavan organisaation avaimella.

Tarkastellaan tätä asiakastietueen rakennetta tutkielman arkkitehtuurin kannalta. Henkilötietojen irrottaminen omaksi entiteetikseen muusta sovelluksesta on hyvältä vaikuttava ratkaisu. Moduulina toimiva komponentti on hyvä ominaisuuden *(O7: uudelleenkäytettävyys)* kannalta. Asiakastietueen rakenne ja hallinta keskittyy tiedon siirtämiseen eri yritysten (henkilötietorekisterien) välillä. Tähän liittyvät tietueen jäljitettävyystiedot. Lähestymistapa on validi, mutta se ei tuo parannuksia tiedon käsittelyyn yksittäisen henkilötietorekisterin sisällä. Tietosuoja-asetuksen *A20* antaa oikeuden tietojen siirtoon, mutta pakottaa vain luovuttamaan tiedot jäsennellyssä muodossa. Automaattinen siirto järjestelmien välillä on *A20(2)* mukaan oikeus vain, jos se on teknisesti mahdollista. Se, miten *teknisesti mahdollinen* määritellään viranomaisten toimesta jää vielä avoimeksi. Parhaimmillaan kehys toimisi, kun monet organisaatiot ottaisivat sen käyttöön ja henkilötiedon linkitetty lista jakautuisi asiakastietueilla eri järjestelmien yli. Käytännössä tämä lienee liikaa pyydetty.

Aliluvussa esitelty privacyTracker käsittelee työn aihetta, mutta suoraan sitä ei sellaisenaan oteta käyttöön. Moduuli vastaisi vaatimuksiin *(V4: henkilötietojen jäljitettävyys)* ja *(V5: rekisteröidyn pääsy tietoihin)*, mutta suuri osa kehyksen tavoitteesta käsittelee sivuavaa asiaa. Arvokkainta kehyksessä tutkielman kannalta on esitelty tapa suunnitella kryptografista jäljitettävyyttä arkkitehtuuriin ja validointia modulaariselle lähestymistavalle

tietosuojan hallintaan. Palataan myöhemmässä luvussa tutkielman työtä määriteltäessä siihen, että saadaanko näitä sovellettua työn arkkitehtuuriin. Artikkeli myös antaa kannustusta, että tekniselle tutkimukselle yleisen tietosuoja-asetuksen kysymyksissä on markkina.

4.2 Anonyymiyden tasot

Hintze esittää vuoden 2016 artikkelissaan [17] eri anonymisoinnin tasoja auttamaan henkilötietojen käsittelyyn yleisen tietosuoja-asetuksen mukaisesti. Henkilötietolakien noudattaminen helpottuu tunnistamalla, että datan anonyymiys on spektrumi ja käyttämällä sen eri muotoja oikeaoppisesti. Vahvemmin anonymisoidut henkilötiedot ovat luonnollisesti pienempiriskisiä, kuin heikommin anonymisoidut.

Tutkielman arkkitehtuuriratkaisuissa eri anonymisoinnin tasot voivat olla hyödyksi, joten artikkelin tulokset ovat kiinnostavia. Esitellään seuraavaksi, miten Hintze on spektrumia eritellyt. De-identifiontitekniikat ovat yksi tapa ratkaista tietosuojavaatimuksia, tai ainakin helpottaa teknisten vaatimusten toteuttamista.

Ennen yleistä tietosuoja-asetusta, nykytilan konsepti henkilötietojen anonymisoinnista on mustavalkoinen. Data joko on henkilötietoa, tai ei ole. Anonymisoinnin on oltava peruuttamaton ja sen on tuhottava linkki henkilötietoon. Hinze näkee tämän haitallisena, koska se ei kannusta organisaatioita tekemään mitään välimuotoisia anonymisointeja tilanteissa, joissa täyttä anonymiteettiä ei voida saada. Lopputuloksena henkilötiedot ovat muodossa, jossa niistä aiheutuu suurempi riski.

Yleisessä tietosuoja-asetuksessa taas esitellään enemmän laajuutta anonymisoinnin käsitteisiin. Henkilötiedon ja anonymisoidun tiedon lisäksi tunnistetaan pseudonymisoitu tieto. Pseudonymisointi tuotiin esille luvussa 3. Pseudonymisoitu tieto tarkoittaa henkilötietoa, josta ei voida tunnistaa henkilöä tuomatta lisätietoa kontekstiksi. Neljänneksi käsitteeksi, asetuksen artiklassa *A11* tuodaan vielä yksi anonymisoinnin taso, jota Hintze nimittää *A11-anonymisoiduksi*. A11-anonymisoitu data on henkilötieto, josta rekisterinpitäjällä ei ole mahdollisuutta tietoja yhdistelemälläkään tunnistaa henkilöä. Tämä antaa rekisterinpitäjälle vapauksia asetuksen vaatimuksista, jotka tarjoavat rekisteröidylle oikeuksia. Rekisteröidyn on toimitettava tarvittavat lisätiedot, joilla hänet taas voi tunnistaa, mikäli hän haluaa käyttää oikeuksiaan.

Artikkelissa jaotellaan tietosuoja-asetuksen tuntemat henkilötietojen anonyymiyden tasot seuraavasti: 1) *tunnistettu*, 2) *tunnistettavissa oleva*, 3) *A11-anonymisoitu*, ja 4) *anonyymi*. Tunnistettu data on sellaista, josta luonnollinen henkilö on välittömästi tunnistettavissa. Tunnistettavissa olevaa dataa ei yksinään voida linkittää henkilöön, mutta on tiedossa systemaattinen tapa tehdä linkitys. *A11-anonymisoitu* vaatii rekisterin ulkopuolista lisätietoa tunnistamiseksi. Anonyymiä dataa ei voida mitenkään linkittää henkilöön.

Alla oleva taulukko *(Taulukko 4)* on Hintzen jaottelu tasoista järjestyksessä. Taulukosta ilmenevät kunkin tason ominaisuudet. Tiedon säilyttämisen riski pienenee jokaisella tasolla, mutta ainoastaan täysin anonyymi tieto on tietosuoja-asetuksen henkilötiedon määritelmän ulkopuolella.

TAULUKKO 4: Anonyymiyden tasot Hintzen mukaan. [17]

	Tunnistettu	Tunnistettavissa oleva	A11-anonymisoitu	Anonyymi
Välitön linkki henkilöön	Kyllä	Ei	Ei	Ei
Tiedetty tapa tunnistaa	Kyllä	Kyllä	Ei	Ei
Liittyy tiettyyn henkilöön	Kyllä	Kyllä	Kyllä	Ei

Hintze esittää oman lakitekstin tulkintoihin perustuvan kantansa anonymisoinnin tasojen käytöstä tietosuoja-asetuksen vaatimuksien ratkaisuun. Eri anonymisoinnin tasoilla voitaisiin vastata asetuksen "pehmeisiin", tulkinnanvaraisiin vaatimuksiin. Vielä ei tiedetä, miten asetuksen tekstiä tullaan käytännössä tulkitsemaan viranomaisten toimesta, mutta Hintzen kanta on ainakin perusteltavissa. Käydään seuraavaksi läpi henkilötietojen käsittelyn osa-alueet, joihin Hintze esittää anonymisoinnin auttavan. Ne ovat käsittelyn lainmukaisuus, ilmoitukset, datan säilytys, tietoturva, ja rekisteröidyn oikeuksien käyttö.

Käsittelyn lainmukaisuus on, *A6* mukaisesti, joko rekisteröidyn suostumuksen mukaista tai rekisterinpitäjän oikeutettujen etujen mukaista. Hintze ehdottaa, että oikeutettujen etujen mukaista henkilötietojen käsittelyä tulisi tulkita suotuisasti rekisterinpitäjän kannalta silloin, kun käytetään *A11-anonymisoitua* dataa. Argumentti on se, että anonymisoinnin tason kasvaessa suostumuksesta riippuvuuden tulisi olla pienempää. Artikla tukee tätä osittain, mainitsemalla pseudonymisoinnin olemassaolon ehtona uutta tarkoitusta varten tehtävälle käsittelylle.

Datan säilytyksestä alkuperäisen käsittelyn jälkeen ei asetus anna täysin selvää kuvaa, Hintzen mukaan. *A5* sallii henkilötietojen säilyttämisen *"yleisen edun mukaisia arkistointitarkoituksia taikka tieteellisiä tai historiallisia tutkimustarkoituksia tai tilastollisia tarkoituksia varten"*. Tämä liittyy tutkielmassa käsiteltävään vaatimukseen *(V2: tietojen minimointi)*. Neljännen tason täysi anonymisointi on aina vaihtoehto, kun käsittelyn on päätyttävä. Tällöin jäljelläolevan datan käsittely ei olisi enää henkilötiedon käsittelyä. Hintze esittää, että myös tason *A11-anonymisoitu* käyttö voisi olla mahdollisuus, tekstin tulkinnasta riippuen. Hän kyseenalaistaa myös *käsittelyn tarpeellisuuden* määritelmää.

Tietoturvan kannalta kaikki anonymisoinnin käyttö on selvästi parempi strategiana, kuin sen tekemättä jättäminen. Vaatimukset *(V1: järjestelmän tietosuoja)* ja *(V2)* edellyttävät asianmukaiset toimet henkilötietojen leviämisen riskinhallintaan ja sen määrän minimointiin. Riskit ovat vahvasti sidoksissa datan luonteeseen – edes jollain anonymisoinnin tasolla oleva data on vähemmän vaarallista menettää. Anonymisointitekniikat siis tekevät muista tietoturvatoimista vähemmän kriittisiä.

Rekisteröidyn oikeuksiin anonymisointi vaikuttaa suoraan asetuksen tekstin mukaan. Vaatimuksissa *(V5: rekisteröidyn pääsy tietoihin)*, *(V6: rekisteröidyn tietojen oikaisu)*, *(V7: rekisteröidyn tietojen poistaminen)* ja *(V8: rekisteröidyn vastustamisoikeus)* kuvatut rekisteröidylle tarjottavat palvelut eivät koske *A11-anonymisoitua* dataa. Tämä voi helpottaa käytännön ongelmia näiden vaatimusten ratkaiussa. Esimerkiksi, tietyssä osassa järjestelmää voidaan saada taso *A11-anonymisoitu* datalle, jolloin se jää em. rekisteröidyn oikeuksien ulkopuolelle.

Artikkelissa esitellyt seikat on tärkeää huomioida arkkitehtuuria suunnitellessa. Hintzen esittelemä perpektiivi, jossa selvitetään anonymisoinnin mahdollisuus kaikkiin ratkaistaviin kohtiin, on avuksi ongelmien kanssa. Yleisiä linjauksia työn suhteen ei tässä tehdä, mutta palataan asiaan luvussa 5 yksittäisien ratkaisujen parissa. Tutkielman tapausyrityksen kohdalla, anonymisoinnin tasojen vertailu on erityisen relevanttia ominaisuuksien *(O5: staging-ympäristön vaatimukset)* ja *(O8: tukitoiminta ja palvelimien ylläpito)* kannalta.

4.3 Privacy As A Service

Su et al. julkaisivat artikkelissaan 2016 [43] arkkitehtuurisia ratkaisuja terveysdatan hallintaan. Heidän esittämä mallinsa on nimi *yksityisyys palveluna* (Engl. *Privacy As A Service*). Kyseinen artikkeli käsittelee henkilödatan hallintaa eri palveluiden välillä. Arkkitehtuuri on ”ihmiskeskeinen” ja se pyrkii myös huomioimaan tietosuoja-asetuksen vaatimukset. Tutkielman kannalta kiinnostavat osat ovat heidän esittämänsä ratkaisut, joita voidaan käyttää yleiseen tietosuojan hallintaan. Artikkeli esittelee myös terveysdataan erikoistuneita ratkaisuja, mutta sivuutetaan ne tämän tutkielman aiheen ulkopuolisina.

Artikkeli viittaa konseptiin *MyData*-operaattorista [41], jossa henkilö hallitsee oman henkilödatansa siirtoja verkossa. Operaattoriin rekisteröidään datalähteitä ja -nieluja, jotka ovat itsenäisiä palveluitaan. MyData-arkkitehtuuri määrittää kommunikointitavat näiden välille ja suostumuksen hallinnan siirtojen tekemiseen. Käyttäjä voi vapaasti liittää palveluita omaan tiliinsä. Tämän jälkeen hän voi valita, mitkä datanielut saavat käyttää mitäkin datalähdettä.

Suostumuksen välittäminen datalähteen ja nielun yhdistämiseen tapahtuu *OAuth 2*:sta [16] johdetulla protokollalla. Lopputulos on turvallinen suostumuksen myöntäminen ja eri palveluiden käytön helppous. Yhteisen rajapinnan toteuttavat, avoimen lähdekoodin

MyData-operaattorit ovat helposti vaihdettavissa keskenään, mikä on tietosuoja-asetuksen hengen mukaista.

Luvussa 5.5 esiteltävä ratkaisu suostumuksen hallinnalle on hyvin samankaltainen artikkelin arkkitehtuurin kanssa. Su et al. julkaisivat kuitenkin pitkälle kehitetyn arkkitehtuurin reilusti ennen tutkielman kirjoittamista, mikä on tutkielman työn kannalta harmillista. Kehitetyt ratkaisut olivat kuitenkin alunperin itsenäisiä, mutta myönnän puutteen taustatutkimuksessa. Koska tutkielman ratkaisu on kuvattu tarkasti luvussa 5.5, ei tässä vielä voida tehdä tarkempaa vertailua eri näiden välillä. Todetaan kuitenkin, että artikkelista saatiin vahva validointi tutkielman ratkaisutavan suhteen.

Luku 5
Tutkielman arkkitehtuuri

Tutkielmassa suunniteltu ohjelmistoarkkitehtuuri, siinä tehdyt ratkaisut ja keskeisten kysymysten tarkastelu. Esitetään vastauksia aiemmin tehtyyn tietosuoja-asetuksen vaatimusmäärittelyyn.

Tässä luvussa esitetään tutkielman ratkaisuehdotus ohjelmistoarkkitehtuurista, joka vastaisi yleisen tietosuoja-asetuksen vaatimuksiin. Luku alkaa arkkitehtuurin yleiskuvasta, jossa esitellään suunniteltu palvelukeskeinen ratkaisu. Arkkitehtuuri koostuu moduuleista ja kirjastoista, jotka kukin vastaavat tiettyihin vaatimuksiin. Kokonaisuutena ne muodostavat kattavan ratkaisun. Moduulit käydään erikseen läpi luvun aliluvuissa. Tutkielman puitteissa arkkitehtuurin tietyistä osista on tehty erimerkkitoteutukset, joiden keskeisiä osia luvussa tarkastellaan.

Luvun tavoitteena on esitellä perusteltu ratkaisu luvun 3 vaatimusmäärittelyyn. Luvussa 6 taas analysoidaan tätä arkkitehtuuria ja pyritään validoimaan tässä luvussa esitettävä argumentti. Luvuissa 2 ja 3 esitetyt premissit oletetaan tässä luvussa oikeiksi. Näin tutkielman kulku sitoo tapausyrityksen ominaisuudet tietosuoja-asetuksen vaatimuksiin. Lopulta saamme luottamusta sille, että tämän luvun arkkitehtuuria soveltamalla tapausyritys (ja sen kaltaiset yritykset) voivat tehokkaasti ratkaista tietosuoja-asetuksen tuomat tekniset haasteet.

Tämä luku pyrkii esittämään vastauksen kahteen väitteeseen. Esiteltävä arkkitehtuuri 1.) kattaa luvun 3 vaatimusmäärittelyn ja 2.) on paras ratkaisu tapausyrityksen arkkitehtuuriksi. Väitteistä ensimmäinen on helpompi käsitellä, olettaen vaatimusmäärittelyn validiksi. Toinen on käytännössä mahdotonta todistaa ei-triviaaleille järjestelmille. Siksi lievennetään sitä seuraavasti: tavoite on *kattaa* tapausyrityksen vaatimukset ja tehdä kaikki ratkaisevat valinnat *perustellusti*.

5.1 Arkkitehtuurin yleiskuva

Seuraavaksi käydään läpi tutkielman arkkitehtuuri korkealla tasolla. Tämän jälkeen esitellään yksittäiset komponentit tarkemmin. Aliluvun tavoite on esitellä arkkitehtuurin lisäksi valinnat, joiden mukaan siihen päädyttiin. Valinnat perustuvat tapausyrityksen

ominaisuuksiin (luku 2) ja tietosuoja-asetuksen vaatimuksiin (luku 3). Aliluvun kulku käy arkkitehtuurin muodostamisen läpi ja lopulta esitetään valmis ratkaisu.

Aloitetaan kysymyksestä: *mitä tässä ollaan suunnittelemassa?* Tutkielman tavoite oli ratkaista verkkopalvelun arkkitehtuuri tapausyrityksen tyypilliselle asiakasprojektille. Kyse ei ole kuitenkaan tietystä *tapausprojektista*, jonka rakenteeseen voitaisiin täysin sitoutua. Myöskään tavoite ei ole tehdä yleismaailmallista ratkaisua kaikkiin verkkopalveluihin – tämä ei olisi realistista. Viitataan tapausyrityksen ominaisuuksiin *(O1: jako asiakkuuksien välillä)*, *(O2: teknologiasidonnaisuus)* ja *(O7: uudelleenkäytettävyys)*. Arkkitehtuurin tulee olla vahva näiden kannalta, mutta tasapainossa *(O4: projektien ulkopuoliset investoinnit)*:n ja ratkaisun kompleksisuuden kanssa.

Ominaisuuksilla *(O1)* ja *(O7)* on yksi erityinen ulottuvuus toistensa kanssa, joka ilmenee arkkitehtuuria suunnitellessa. *(O1)* tavallaan laajentaa uudelleenkäytettävyyden skaalaa. Pienimmillään ohjelmistojen uudelleenkäyttö olisi vain yksittäisen ohjelmakoodin osan kutsumista monesta paikasta. Tästä korkemmalle tasolle mennessä, sama moduuli voidaan ottaa käyttöön useassa projektissa. Korkein taso on käyttää yhtä palvelua (palvelun instanssia) kaikkien projektien kesken. Nyt tietosuoja-asetuksen vaatimuksia ratkaistaessa on tarkastelemisen arvoista, että miten eri ratkaisun osat sopivat tähän. Henkilötietorekisterin rajat ovat vahvat: tämän vuoksi on haasteellista tuoda suuria osia tälle korkeimmalle uudelleenkäytettävyyden tasolle.

(O2: teknologiasidonnaisuus) on olennainen ominaisuus huomioida arkkitehtuurin suunnittelun alkuvaiheessa. Tämä on myös tiukasti sidoksissa *(O7)*:n kanssa: mitä tiukemmin ratkaisut ovat teknologiaan sidottuja, sitä vähemmän vaihtoehtoja niitä uudelleen käytettäessä on. Kuitenkin kaikki implementaatiot ovat sidoksissa teknologiaan, jolla ne on toteutettu. Tutkielman arkkitehtuurin kannalta tärkeintä onkin se, miten tietosuoja-asetusken ratkaisut sitovat projektien omaa, projektikohtaista toteutusta. Esimerkiksi mikäli kaikki vaatimukset voitaisiin ratkaista *Javakirjastolla*, ei tätä voitaisi käytännöllisesti hyödyntää muissa projekteissa. Hyvä ratkaisu ominaisuuden kannalta olisi taas erilliset palvelut, joita millä tahansa teknologialla toteutetut projektit voivat käyttää esimerkiksi HTTP:n kautta.

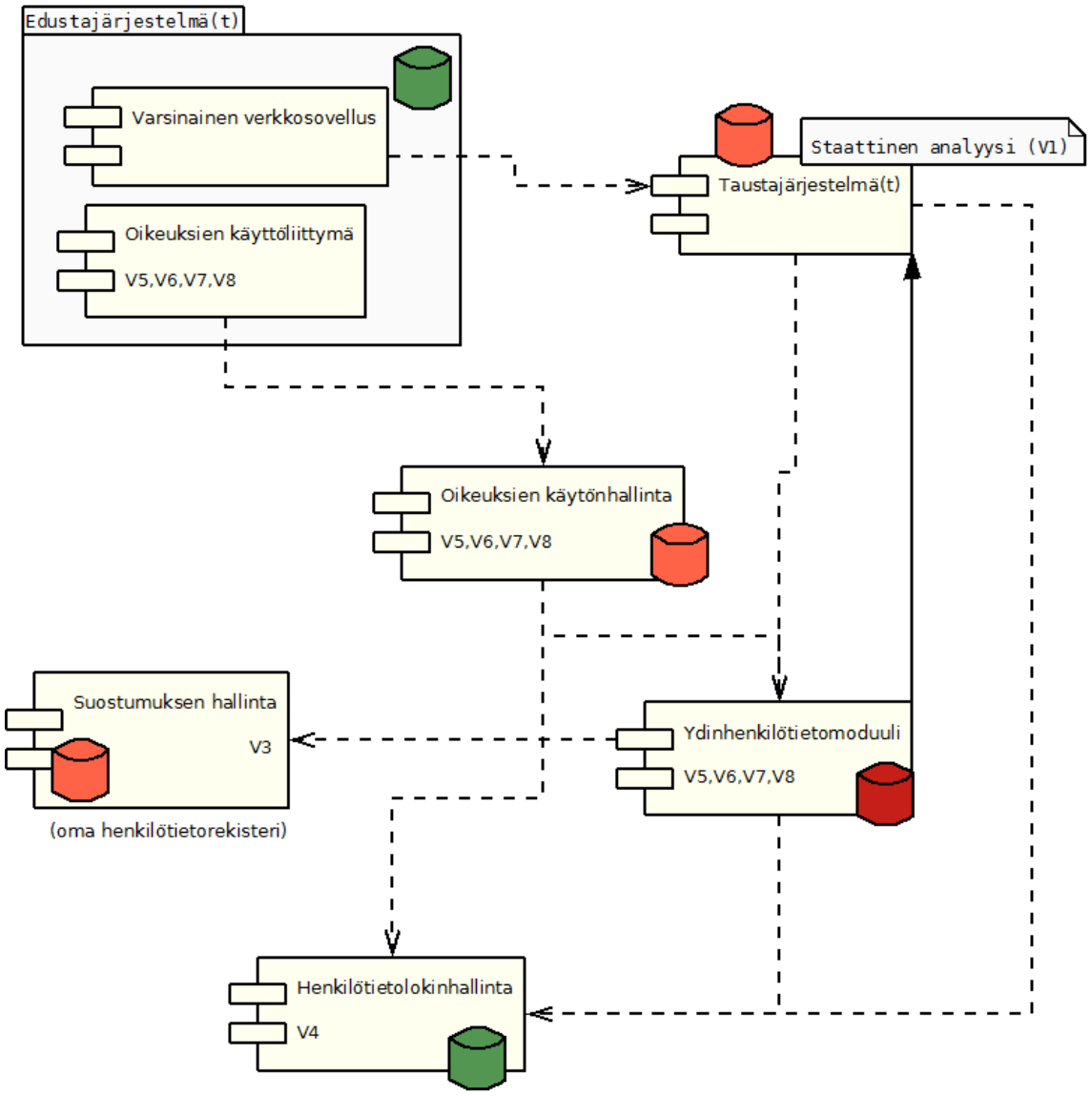
Sekä uudelleenkäytettävyyden, että teknologiasidonnaisuuden kannalta, arkkitehtuuri voidaan jakaa useaan komponenttiin. Tällöin huonosti uudelleenkäytettävät osat ja erittäin teknologiakohtaiset osat haittaavat vain omaa komponenttiaan. Muut komponentit tällöin eivät rajoitu näiden tasolle. Suuri komponenttien määrä aiheuttaa haastetta ominaisuuden *(O8: tukitoiminta ja palvelimien ylläpito)* kannalta ja niiden kompleksisuus vaikeuttaa eri järjestelmien koordinointia. Tämä on jälleen tasapaino, josta optimaalisen vaihtokaupan löytäminen ei ole yksinkertaista. Hyvä komponenttijako on sellainen, että niiden välillä on vähän riippuvuuksia (Engl. *low coupling*), mutta komponenttien sisäinen koheesio on suuri (Engl. *high cohesion*) [29].

Määritellään seuraavaksi hieman pohjaa arkkitehtuurille. Pelkät tietosuojaratkaisut eivät ole mielekkäitä, jollei niillä ole kontekstia. Luvussa 2 esiteltiin tapausyrityksen esimerkkiprojekti. Tutkielmassa suunniteltava arkkitehtuuri ei ole tarkoitettu lukituksi tälle, mutta otetaan esimerkkiprojektista joitain vähimmäisvaatimuksia. Tutkielman arkkitehtuuri kuvaa verkkopalvelua, jota käyttää jokin loppukäyttäjäryhmä. Muun palvelun toiminnan lisäksi loppukäyttäjiltä kerätään erinäisiä henkilötietoja. Loppukäyttäjien lisäksi järjestelmää käyttävät ylläpitäjät, jotka tekevät asiakaspalvelua ja sisällön syöttämistä. Arkkitehtuuri koostuu vähintään edustajärjestelmistä (1-N kpl) ja taustajärjestelmistä (1-N kpl). Henkilötietoja pitää olla mahdollista käsitellä vähintään kaikissa taustajärjestelmissä.

Näiden lähtökohtien perusteelta, tarkastellaan sitten tietosuoja-asetuksen vaatimuksia (luvusta 3). Vaatimuksia voidaan ryhmitellä hieman, täten pyrkiä kattaaman vaatimuksia tehokkaammin. Rekisteröidyn eri oikeuksien *(V5, V6, V7, V8)* käyttäminen johtaa kahteen selvään kokonaisuuteen. Ensimmäiseksi oikeuksien käyttöpyynnöt tulee mahdollistaa ja käsitellä. Nämä voidaan tehdä irrallaan muusta järjestelmästä, mutta riippuvat luonnollisesti järjestelmän rakenteesta. Oikeuksien toteuttaminen taas vaatii koko järjestelmän tietomallin mahdollistavan kunkin vaatimuksen toiminnallisuuden. Muut omat kokonaisuutensa ovat jäljitettävyysloki *(V4)* ja suostumuksen hallinta *(V3)*. Vaatimukset *(V1: järjestelmän tietosuoja)* ja *(V2: tietojen minimointi)* ovat laajoja ja koskevat koko järjestelmää.

(V9: henkilötietojen sijainti) on uniikki vaatimusten joukossa siksi, että sen voi ratkaista vain linjaamalla ettei järjestelmän tietoja viedä EU:n ulkopuolelle. Tällöin sen noudattaminen ei ole enää arkkitehtuurinen kysymys, vaan vaatii huolellisuutta toteutuksessa. *A45:n* perusteella EU:n komissio voi vielä laajentaa sallituiksi henkilötietojen säilytyssijainneiksi muita maita, jotka se tulkitsee riittävän turvallisiksi. Mikäli siirroille on tarvetta, on se huomioitava myös henkilötietolokin osalta.

Tavoitteena on vastata kaikkiin jäljellä oleviin vaatimuksiin jollain tavalla arkkitehtuurin tasolla. Käydään seuraavaksi läpi kukin tutkielmassa ehdotettavista ratkaisuista eri vaatimuksiin. Näihin jokaiseen syvennytään tarkemmalle tasolle myöhemmin tässä luvussa, omissa aliluvuissaan. Arkkitehtuurin kokonaiskuva ilmenee edellä olevasta kuvasta *(Kuva 3)*. Kuvassa ovat tapausyrityksen tyypillisen projektin edustajärjestelmät ja taustajärjestelmät. Seuraavat moduulit tukevat niitä, tietosuoja-asetuksen vaatimuksiin vastaten: *oikeuksien käyttöliittymä*, *oikeuksien käytönhallinta*, *suostumuksen hallinta*, *ydinhenkilötietomoduuli* ja *henkilötietolokinhallinta*. Kuvassa on eritelty kunkin komponentin riippuvuudet toisistaan. Lisäksi kuhunkin kuuluvan tietokannan väri on joko punainen (tallennettu henkilötietoja), tai vihreä (ei henkilötietoja). Kriittisimpiä henkilötietoja on korostettu tummemmalla sävyllä.



Kuva 3: Tutkielman arkkitehtuurin yleiskuva.

Vaatimus *(V4: henkilötietojen jäljitettävyys)* on hyvä aloituskohta, koska sen sisältö ei riipu muiden vaatimusten ratkaisusta. Pohjimmiltaan riittää, että järjestelmä pitää lokia valituista henkilötietojen käsittelytapahtumista. Kun huomioidaan *(O9: henkilötietojen levinneisyys)*, on selvää, että lokin hallinta ei ole kuitenkaan täysin triviaali ongelma. Viranomaisten selvityspyyntöjen ja projektien oman seurannan vuoksi lokiin pääsy olisi hyvä olla keskitetty. Tällöin ongelman ydin on kerätä useasta järjestelmän osasta tulevat lokitiedot määrätyssä muodossa yhteen palveluun. Palvelun on tuettava useasta lähteestä keräämistä ja se voi tarjota lokin analysoimiseen eri näkymiä ja tilastoja. Tavoitteena on, ettei lokiin itseensä tallenneta henkilötietoja, ainoastaan tiedot yksilöiviä järjestelmän sisäisiä avaimia. Keskitettyjä lokinhallintaratkaisuja on olemassa, myös avoimen lähdekoodin materiaalina. Esimerkiksi vaihtoehtona on usean eri lokihallintatuotteen yhdistelmä *ELK-pino (Elastic, Logstash ja Kibana)* [22]. Tutkielman työksi jää näiden valinta ja soveltaminen henkilötietojen jäljitettävyyden kontekstissa.

Rekisteröidyn oikeuksien käyttöpyyntöjen hallintajärjestelmä on toinen oma kokonaisuutensa. Tämä koostuu kahdesta moduulista: rekisteröidyn käyttöliittymästä ja pyyntöjen hallintamoduulista. Jako on relevantti uudelleenkäytettävyyden ja ylläpitotyön määrään liittyen. Käyttöliittymä, josta rekisteröity voi tehdä tietosuoja-asetuksen oikeuttamia pyyntöjään, on sulautettava muuhun kunkin projektin käyttöliittymään. Siitä on siis tarpeen tehdä tyylillisesti kustomoitava, jotta se sopii kuhunkin sivuston ulkoasuun. Ratkaisussa käyttöliittymän logiikka on sama kaikilla instansseilla, ja ne konfiguroidaan tekemään pyynnöt keskitettyyn pyyntöjenkäsittelyjärjestelmään. Käyttöliittymäosaan ei tallenneta mitään tietoja, se on ainoastaan alusta kutsua oikeuksien käytönhallintaa.

Pyyntöjen taustajärjestelmä on ylläpitäjien hallinnoima. Sen instanssit voivat olla joko henkilötietorekisteri-, tai asiakaskohtaisia tarpeen mukaan. Tämä keskittää ylläpitäjien työn yhteen paikkaan. Hallintajärjestelmän olemassaolon perustelu on se, ettei kaikkia rekisteröidyn oikeuksia ole kiistattomasti ja välittömästi toteutettava. Ylläpitäjä voi arvioida pyyntöjä niitä käsitellessään. Hallintajärjestelmä myös enkapsuloi logiikan siitä *miten* pyynnöt rekisterin laajuisesti toteutetaan; tällöin jokaisen käyttöliittymän ei tarvitse tietää siitä. Järjestelmään ei tallenneta muuta henkilötietoa, kuin oikeuksien käyttöpyyntöjen mahdollinen sisältö. Muuten tallennetaan ainoastaan viittauksia henkilötietoon.

Rekisteröidyn oikeuksien käytönhallintajärjestelmän lisäksi tutkielma esittää erillisen *ydinhenkilötietomoduulin*. Raja näiden kahden komponentin välillä on vastuissa: pyyntöjen käsittelyjärjestelmän tehtävä on kommunikoida rekisterinpitäjän ja rekisteröidyn välillä, ja välittää näitä pyyntöjä ydinmoduulille. Lopulta ei ole suurta eroa, ovatko nämä kokonaan omat järjestelmänsä, vai erilliset osat samaa järjestelmää. Ydinhenkilötietomoduulin tulee ehdottomasti kuitenkin olla rekisterikohtainen.

Ydinhenkilötietomoduulin tulee ottaa huomioon tapausyrityksen vaatimus, että taustajärjestelmiä on voitava olla usea kappale. Moduuli kuvataan tarkemmin omassa aliluvussaan (5.4). Tavoite on kuitenkin se, että välittömästi henkilöön viittaavat henkilötiedot (nimi, henkilötunnus, yms.) tallennetaan ainoastaan ydinmoduuliin. Tällä pyritään turvaamaan yleisesti koko järjestelmän tietosuojaa, *(V1: järjestelmän tietosuoja)* ja *(V2: tietojen minimointi)* mukaisesti. Perustelu on, että esimerkiksi rekisteröidyn nimen tallentaminen moneen paikkaan olisi huonompi ratkaisu näiden vaatimusten kannalta. Muihin – sovelluskohtaisiin – taustajärjestelmiin tallennetaan vain viittauksia ydinhenkilötietueeseen. Sovelluskohtainen data voi myös olla henkilötietoa, mutta tämä olisi sitä vain välillisesti. Vaatimukset *(V5, V6, V7, V8)* kuitenkin riippuvat koko järjestelmän kattavasta henkilötietomallista. Ydinhenkilötietomoduuli onkin kontrolleri, jonka on voitava välittää näiden toteuttaminen kaikkialle, minne ne liittyvät.

Suostumuksen hallinnan *(V3)* tutkielma ehdottaa olevan kokonaan oma itsenäinen järjestelmänsä. Parhaimmillaan rekisteröidyn suostumusta, sekä alaikäisten tapauksessa

vanhempien suostumusta, voisi hallita yhden järjestelmän kautta kaikkien henkilötietorekisterien yli. Tämä ratkaisu voi olla käytännössä liian vaativa, mutta vaihtoehtoa tarkastellaan myöhemmässä kohdassa (luvussa 5.5). Täysin keskitetty suostumuksen hallinta olisi ennen kaikkea käyttäjäystävällistä, mutta siinä voisi myös olla mahdollisuus liiketoiminnan laajentamiseen. Moduulin on huomioitava suostumuksen hallinnan lisäksi käyttöliittymät suostumuksen keräämiselle ja sen perumiselle. Samoin kuin oikeuksien käyttöpyyntöjen kanssa, niiden on voitava sulautua kunkin verkkopalvelun käyttöliittymään vähintään ulkonäöllisesti.

Vaatimukset *(V1: järjestelmän tietosuoja)* ja *(V2: tietojen minimointi)* ovat haastavimmat ratkaista. *(V1):n* alakohdat ovat laatuattribuutteja ja *(V2):ta* voi tulkita samalla tavalla. Asetuksen teksti antaa vapauksia muodossa: *"Ottaen huomioon uusimman tekniikan ja toteuttamiskustannukset sekä käsittelyn luonteen, laajuuden, asiayhteyden ja tarkoitukset sekä käsittelyn aiheuttamat todennäköisyydeltään ja vakavuudeltaan vaihtelevat riskit"*. Tällöin vaatimuksena on voitava perustella, ettei järjestelmä riko näitä ominaisuuksia. Tietojen minimointi on osittain tautologia; mikäli henkilötietoa tarvitaan tietyssä paikassa järjestelmää, sitä ei tarvitse poistaa. Tulee siis vain varmistaa, ettei selvästi turhaa dataa käytetä tai tallenneta redundantisti.

Tutkielmassa esitetään *(V1):n* ja *(V2):n* turvaamiseksi kaksi konkreettista välinettä. Tapausyrityksen ominaisuuden *(O5: staging-ympäristön vaatimukset)* täyttäminen voisi olla näiden vaatimusten kannalta kyseenalaista. Ominaisuuden mukaan tarkoituksena on käyttää tuotantodataa vastaavaa dataa myös testikäytössä järjestelmää kehittäessä. Voitaisiin toki yrittää perustella, että tämä on välttämätöntä ja siksi täyttää *(V2):n* sellaisenaan. Asetuksen henkeä noudattaen esitellään kuitenkin *pseudonymisointiprosessi*, jonka tuloksena tuotantodata muutetaan vähintään *A11-anonymisoiduksi*. Taso *A11-anonymisoitu* esiteltiin luvussa 4. Siinä henkilötieto on tilassa, josta sitä ei voi tunnistaa ilman *ulkopuolista* lisätietoa.

Toinen vaatimuksien täyttämistä helpottava väline on työkalu lähdekooditason tietosuojan edistämiseksi. Moduulista esitellään tutkielmassa konsepti. Toteutus *Java*-kielelle tehdään tutkielmaan liittyvänä työnä. Moduuli pyrkii staattisen lähdekoodin analyysin [33] keinoin vahtimaan, etteivät ohjelmointivirheet vaarantaisi tietosuojaa. Koodissa voidaan annotoida henkilötietoja sisältävät ja käsittelevät luokat. Tällöin henkilötietoa sisältävän luokan mahdollisesta väärinkäytöstä voidaan varoittaa ohjelmoijalle välittömästi. Validaattorin varsinainen sisältö ei ole arkkitehtuurikysymys, mutta tällaisten työkalujen valitseminen on.

Näistä osista siis koostuu tutkielman ehdotus asetuksen täyttävästä arkkitehtuurista. Kokonaisuutena strategia on eritellä vaatimuksien vastaukset omiksi järjestelmiksiin kohdista, joista se on kannattavaa. Jaottelun seurauksena henkilötiedot jakautuvat vielä useampaan paikkaan kuin vain alkuperäisiin taustajärjestelmiin, mutta väitämme tämän

olevan kannattava vaihtokauppa. Seuraavaksi käydään läpi moduuleittain niiden sisältö ja kommunikaatio muiden moduulien kesken.

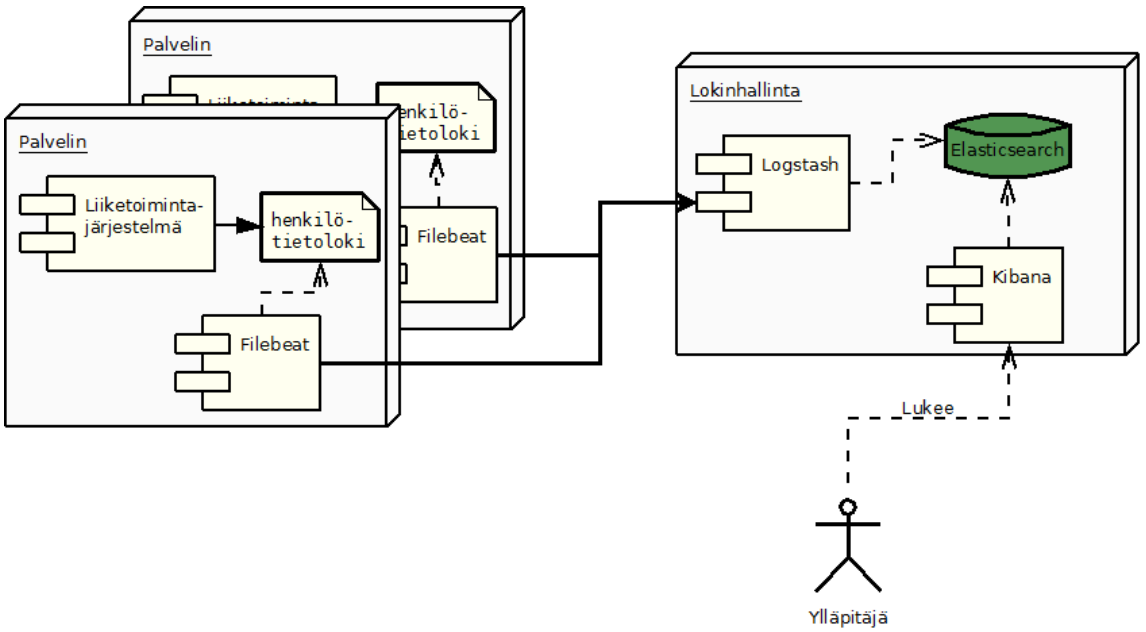
5.2 Henkilötietoloki

Aloitetaan arkkitehtuurin osien käsittely henkilötietolokista. Henkilötietoloki on useasta komponentista rakentuva moduuli, jonka tehtävä on ratkaista tietosuoja-asetuksen vaatimus *(V4: henkilötietojen jäljitettävyys)*. Moduulin tekninen puoli on helpoiten suunniteltu tutkielman ratkaisuista, koska lokienhallinta on varsin yleinen ongelma ja siihen on ratkaisuja vapaassa levityksessä. Tarkastellaan tässä luvussa sitä, miten henkilötietoloki sopii arkkitehtuuriin moduulina ja määritetään lokin itsensä sisältö. Ratkaisun tavoitteena on *keskittää* ja *yhtenäistää* lokit kaikista palvelun järjestemistä. Lisäetuna valmiit lokinseurantatyökalut antavat mahdollisuuden analysoida lokia ja tehdä hälytyksiä vikatilanteissa.

Valitaan tutkielman lokienhallintajärjestelmäksi aiemmin mainittu *ELK-pino (Elastic, Logstash ja Kibana)* [22]. Valinta ei ole vahvan mielipiteen takana, mutta yhdistelmä vaikuttaa hyvältä. Paras puoli on se, että pino koostuu aktiivisesti ylläpidetyistä avoimien lähdekoodin komponenteista. Esitellään lokinhallinnan rakenne tässä luvussa. Korkealla tasolla, eduksi osoittautuu toisistaan löyhästi riippuvien komponenttien vapaa konfigurointi ja tarvittaessa vaihtaminen.

Käydään ensin läpi yleisesti miten lokinhallintakomponentit rakentuvat kokonaisuudeksi. Edellä olevassa kuvassa *(Kuva 4)* on esitelty lokinhallintajärjestelmä kaaviona. *Palvelimet* kuvassa ovat mitä tahansa arkkitehtuurin henkilötietoja käsitteleviä moduuleja. Ne kirjoittavat henkilötietolokin tekstitiedostoon, kukin omassa ympäristössään. Täten koko lokinhallinta on piilotettu järjestelmien kannalta ainoastaan vaatimukseksi kirjoittaa loki levylle – huolehtimatta siitä, miten sitä myöhemmin käsitellään. Määritellään lokin sisältö vielä myöhemmin tässä luvussa.

Tiedostoiksi kirjoittuvia lokitiedostoja tarkkailee kullekin palvelimelle asennettava ohjelmisto Filebeat [24]. Lukiessaan uudet rivit, se toimittaa lokin sanomana lokienhallintapalvelimella kuuntelevalle Logstash-palvelulle [26]. Logstash jalostaa ja yhtenäistää lokitietoja, mikäli se on tarpeellista. Lopulta Logstash tallentaa kaikki saapuvat lokivirrat Elasticsearch-tietokantaan [23]. Elasticsearch ylläpitää tallennetusta datasta hakuindeksiä lähes reaaliajassa. Moduulin viimeinen komponentti on visualisointityökalu Kibana [25], jolla ylläpitäjät voivat seurata lokitapahtumia valituilla näkymillä.



KUVA 4: Henkilötietolokin kerääminen ja esittäminen.

Valitun lokinhallinta-arkkitehtuurin etu on koota hajautetun, monista järjestelmistä koostuvan palvelun kaikki henkilötietolokit samaan paikkaan. Huomoitavaa on se, ettei lokissa itsessään tule olemaan *henkilötietoja*, joten erillisen palvelun pystyttäminen ei vaaranna tietosuojaa. Logstash pystyy keräämään monenlaisia lokeja samanaikaisesti ja Kibana pystyy jaottelemaan lokien visualisoinnit. Tästä seuraa myös mahdollisuus kerätä henkilötietolokin lisäksi järjestelmien tekniset lokit yhteen paikkaan – saavuttaen samat hyödyt, kuin henkilötietolokinkin keskittämisestä. Tällöin työstä arkkitehtuurin toteuttamiseksi saadaan myös teknistä edistystä, vaikka sitä alunperin tehdään tietosuoja-asetuksen vuoksi.

Pohditaan seuraaavaksi: *mitä* oikeastaan pitäisi lokittaa? Lokitettavien tapahtumien lisäksi on määritettävä lokin tietosisältö. Erilaisten lokitapahtumien kategoriat muodostavat vaatimuksen *(V4)* alakohdat: *1) käsittelytapahtuma*, *2) oikeuspyyntö*, *3) siirto EU:n ulkopuolle* ja *4) tiedonluovutus*. Järjestelmiin on toteutettava lokitus kaikkiin kohtiin, jossa näitä tapahtuu. Mikäli seurannan vuoksi on tarve eritellä kategoroita tarkemmin, voidaan näiden lisäksi ottaa käyttöön alakategorioita tapauskohtaisesti.

Käsittelytapahtuma on tietosuoja-asetuksen osoitusvelvollisuudesta johtuva lokitapahtumatyyppi. Lokitetaan tätä tyyppiä käyttäen sellaiset tapahtumat, joissa henkilötietoja on muutettu järjestelmässä. Tällaisia olisi tapahtumat, kuten *”käyttäjä teki tilauksen”*, tai *”ylläpitäjä lisäsi käyttäjälle tilauksen”*. Näistäkin esimerkeistä huomataan, että eri tapahtumilla on eri asteinen tarve lokitukselle. Automaattisten päätösten teko on myös erikseen huomioitava seikka. Tapahtumista tarvitaan siis tieto niiden luojasta. Lokin olemassaolo on vähintään perustelu sille, että henkilötietoja on käsitelty asetuksen mukaisesti.

Oikeuspyynnöt vastaavat seuraavassa luvussa määriteltäviä sanomia, joita lähetetään kun rekisteröity tekee pyynnön käyttää jotain oikeuttaan. Pyynnöt ovat täysin tietosuoja-asetuksen alaisia, joten on myös voitava osoittaa niiden asianmukainen käsittely. Oikeuspyyntöjen käsittely vaatii tarkempaa lokia kuin normaali järjestelmän toiminta; tietosuoja-asetus erityisesti mainitsee, että ne on toteutettava *"ilman aiheetonta viivytystä"*.

Lokitetaan kaikki oikeuspyyntöjen käsittelyn vaiheet, jotta voimme perustella ettei aiheetonta viivytystä ole tapahtunut. Vaiheet ovat seuraavat: pyynnön *saapuminen*, pyynnön *käsittely*, pyynnön *toteuttaminen* ja *vastaus* pyyntöön. Prosessi kuvataan tarkemmin seuraavassa luvussa. Vaiheiden lokittamisella voidaan esimerkiksi osoittaa, että poistamispyyntöön on reagoitu viivytyksettä, mutta poistamisen tarpeellisuuden selvittäminen on yhä kesken. (Poistamista ei ole pakollista tehdä, jos käsittelijällä on esimerkiksi lakisääteinen velvoite säilyttää tiedot.)

Siirto EU:n ulkopuolelle on lokityyppi sille, jos palvelun arkkitehtuuri vaatii välttämättä siirtoja unionin alueen ulkopuolelle. Aiemmassa luvussa määritettiin ratkaisuksi tähän kysymykseen yksinkertaisesti se, ettei tietoja siirretä. Mikäli tämä olisi kyseessä olevan yrityksen vaatimus, on kaikista siirroista tehtävä lokimerkintä.

Tietojen luovutus kolmannelle osapuolelle on usein tarpeellista, kun järjestelmien välille tehdään integraatioita. Tietosuoja-asetuksen mukaisesti luovutettavat tiedot minimoidaan, mutta muuten se on hyväksyttyä kunhan rekisteriselosteessa on asiasta tiedotettu. Luovutuksesta on kuitenkin jätettävä lokitieto, koska niiden asianmukainen käsittely on myös voitava osoittaa.

Nyt olemme määritelleet lokitettavat tapahtumat. Seuraavaksi käydään läpi lokin tietosisältö. Henkilötietojen päätyminen lokiin olisi haitaksi, koska silloin lokiin pätisi kaikki asetuksen haasteet henkilötietojen käsittelylle. Tehdään siis linjaus, että lokissa viitataan henkilöihin ainoastaan järjestelmän sisäisten tunnisteiden kautta.

Lokiriviin tarvitaan vähintään seuraavat arvot: *henkilön tunniste*, tapahtuman *luojan tunniste*, *aikaleima*, *tapahtuma* ja *lokittava järjestelmä*. Lisäksi voidaan jättää avoimeksi järjestelmäkohtaisia laajennuksia, mikäli joillain järjestelmillä on tarve lisätiedoille – *ELK* tukee varsin vapaata tietoformaattia. Logstash voi myös lisätä erittelytiedon siitä, että kyseessä on *henkilötietoloki*, mikäli samaan lokinhallintaan tallennetaan myös muita lokitietoja. Mikäli monta eri henkilötietorekisteriä käyttää samaa keskitettyä lokinhallintaa, voi Logstash lisätä myös kentän *henkilötietorekisteri* lokiriveille.

Luvussa on nyt katettu arkkitehtuurin tuki vaatimukselle *(V4: henkilötietojen jäljitettävyys)*. Suuri osa ehdotetuista ratkaisuista perustuu osoitusvelvollisuuteen. Ehdotettu lokinhallinta kattaa sekä järjestelmän lokitietojen käsittelylle ja varastoinnille, että arkkitehtuurinlaajuiset käytännöt henkilötietolokin muodostamisesta. Keskitetty lokitus on uudelleenkäytettävä yksittäisten järjestelmien välillä (yhden rekisterin piirissä), mutta myös

mahdollisesti koko yrityksen eri rekisterien välillä.

5.3 Oikeuksien käytönhallinta

Tässä luvussa kuvataan *oikeuksien käytönhallinnan* osuus tutkielman arkkitehtuurista. Tietosuoja-asetuksessa määrätyt oikeudet rekisteröidyille edellyttävät, että rekisteröity esittää vaatimuksen oikeuden käytöstä. Osuus ei itsenäisesti kata mitään tietosuoja-asetuksen vaatimuksia, mutta se on oleellinen osa vaatimuksien *(V5: rekisteröidyn pääsy tietoihin)*, *(V6: rekisteröidyn tietojen oikaisu)*, *(V7: rekisteröidyn tietojen poistaminen)* ja *(V8: rekisteröidyn vastustamisoikeus)* ratkaisua.

Oikeuksien käyttöä kuitenkin on tarpeen valvoa ja mahdollisesti rajoittaa. Tähän tehtävään tarvitaan arkkitehtuurin tasolla järjestelmä ylläpitäjille. Toisella puolella rekisteröidylle tulee taas tarjota mahdollisuus esittää näitä pyyntöjä. Pyyntöjen esittämisen toiminnallisuus on sama kaikkien eri projektien kesken, joten on tehokasta kehittää uudelleenkäytettävä ratkaisu tälle. Toisaalta kaikki edustatoiminnot on voitava tehdä kunkin projektin ulkoasun mukaiseksi. Luvussa esitellään nämä kaksi komponenttia: oikeuksien käytönhallinta ja oikeuksien käyttöliittymä.

Tämän oikeuksien käyttöpyyntöjen välitysjärjestelmän tarpeellisuus voidaan kyseenalaistaakin. Tietosuoja-asetus ei suoranaisesti vaadi pyyntöjen käsittelylle erillistä ratkaisua. Lähdetään siitä, että tässä kuvatut roolit ovat kuitenkin tarpeellisia: käyttäjille on kerrottava minne pyynnöt voi esittää, ja pyynnöt on käsiteltävä. Kevyimmillään tämän voisi esimerkiksi ratkaista sillä, että käyttöliittymän osuus on ilmoitus: *"Tietosuoja-asioissa ottakaa yhteyttä gdpr@yritys.com"*. Tällöin oikeuksien käytönhallinta olisi sähköpostilaatikko, jota ylläpitäjä voi tarkkailla. Tämä luultavasti riittäisi vain pienimmille palveluntarjoajille.

Käsiteltävien oikeuksien käyttöpyyntöjen määrä kuitenkin kasvaa kahdessa akselissa: ylläpidettävien projektien koon ja lukumäärän mukaan. Eri oikeuksien pyynnöt ovat erilaisia ja tarvitsevat oman käsittelytapansa. Käsittelyä nopeuttaakin se, että pyynnöt ovat määrätyssä formaatissa, mikä tukee erillistä käyttöliittymäkomponenttia. Täysin manuaalisesti ylläpidettävä järjestelmä vaatii myös silti *henkilötietolokin*, jonka luotettavuus on huonompi kuin automaattisen lokituksen.

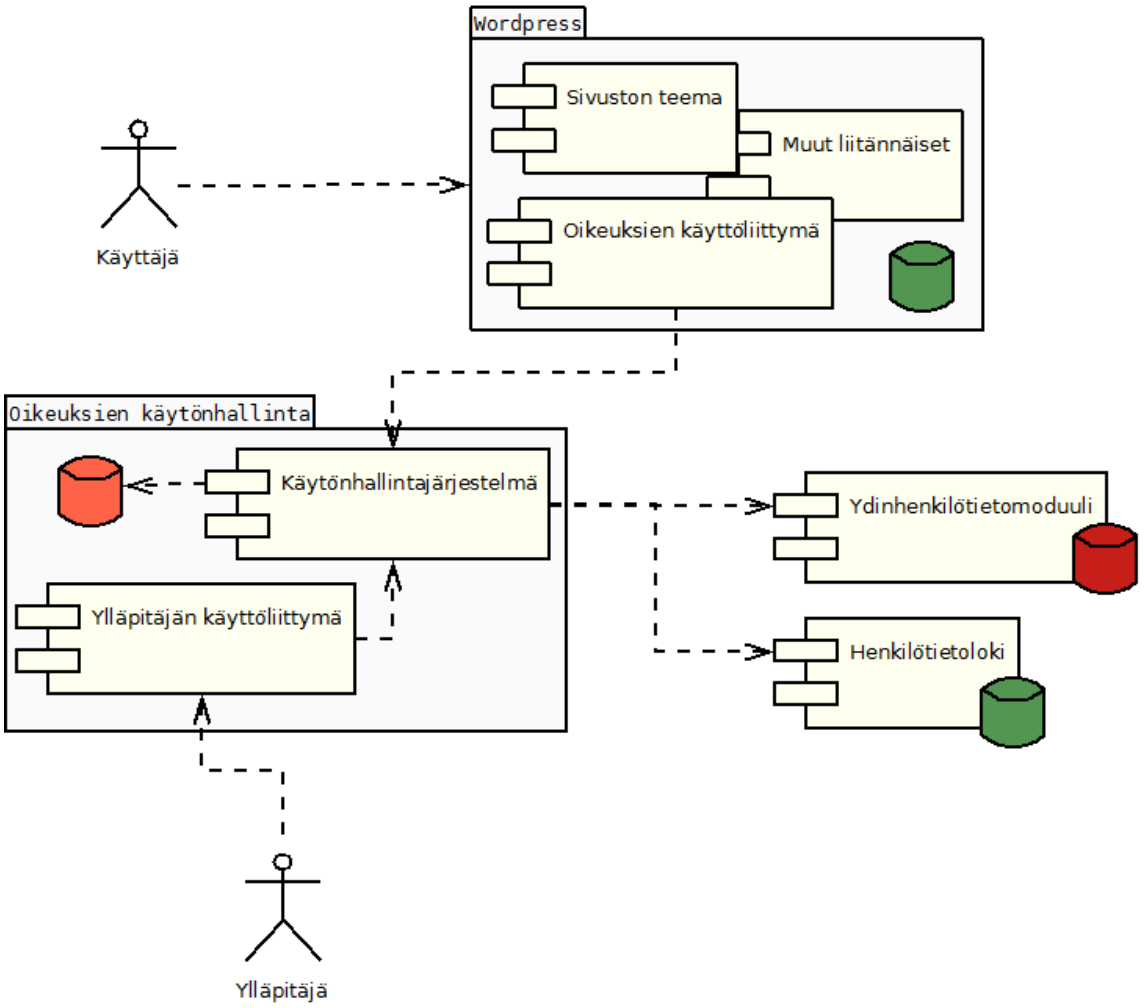
Toinen kyseenalaistus, jonka voisi edellisen jälkeen tehdä, on hallintajärjestelmän tarve. Mikäli oikeuksien käyttöpyynnöt lähetetään automaattisesti, ne voitaisiin toteuttaa automaattisesti ilman viivettä. Ehdotetussa arkkitehtuurissa pyyntöjen varsinaisen toteutuksen tekee kuitenkin *ydinhenkilötietomoduuli*, joten oikeuksien käytönhallinta vain hidastaa pyyntöjen toteuttamista. Tähän vastaan sillä, että kyseessä on kuitenkin *asiakaspalvelu*, jonka täysin automatisointi on haastavaa. Hyvin toteutettu palvelu oikeuksien käytössä voikin olla *etu* liiketoiminnalle. On myös kuviteltavissa, etteivät kaikki pyynnöt

sovi määrämuotoisille lomakkeille, tai ne vaativat tarkempaa selvittelyä ja kommunikointia. Kolmas peruste on se, että käyttäjien tietoja voidaan menettää huolimattomuuden tai ilkivallan seurauksena. Esimerkiksi oikaisupyyntö voisi olla viallista dataa, joka ylikirjoittaisi oikean. Automatisointia oikeuksien käytölle on silti hyvä harkita, esimerkiksi *(V5: rekisteröidyn pääsy tietoihin)* tuskin tarvitsee ylläpitäjää.

Näiden kommenttien jälkeen, esitellään seuraavaksi itse komponenttien sisältö. Lähdetään tilanteesta, jossa sekä käyttöliittymälle, että taustajärjestelmälle on tarve. Käyttöliittymäkomponentin suunnittelu johtaa strategisiin kysymyksiin, jotka tulee ratkaista kontekstikohtaisesti. Vertaillaan käyttöliittymä- ja taustakomponenttia tapausyrityksen ominaisuuksien *(O2: teknologiasidonnaisuus)* ja *(O7: uudelleenkäytettävyys)* kannalta. Teknologiasidonnaisuus ilmenee molemmissa naiivisti ajatellen samoin: kumpikin on sidottu teknologiaan, jolla ne toteutetaan. Käyttöliittymä on kuitenkin upotettava jollekin verkkosivulle, josta käyttäjä siihen pääsee. Taustajärjestelmä taas voi olla oma itsenäinen kokonaisuutensa. Tästä johtuen käyttöliittymäkomponentin teknologiasidonnaisuus on raskaampi; kymmenen sovellusta voisi käyttää taustajärjestelmää ilman muutoksia, mutta käyttöliittymä on kunkin palvelun käyttöliittymätekniikkaan sidottu. Uudelleenkäytettävyys on käyttöliittymäkomponentissa toiseltakin kannalta heikompi. Kukin verkkopalvelu on omanlainen ulkoasultaan, joten komponentin tulee voida olla kustomoitavissa.

Yleismaailmallisesti oikeaa käyttöliittymäratkaisua ei ole työssä tarkoituskaan keksiä, joten voidaan käsitellä tuota kohtaa tapasyrityksen kannalta. Yleinen strategia olisi tuottaa tarvittava määrä oikeuksien käyttöliittymämoduuleja, kaikille kunkin yrityksen tarvitsemille teknologioille. Yritys, joka käyttää hyvin monia käyttöliittymäteknologioita, voisi tyytyä esimerkiksi *Javascript*-kirjastoon, johon on toteutettu ainoastaan palvelun kutsumisen logiikka. Sitä voisi laajentaa kuhunkin projektiin, tuoden näkymäkerroksen projektikohtaisesti. Tutkielman tapausyrityksen kannalta hyödyllisin käyttöliittymäkomponentti on *Wordpress*-liitännäinen (Engl. *plugin*). Käytetään tätä esimerkkinä ratkaisussa.

Tarkastellaan seuraavaksi oikeuksien käytönhallintaa kokonaisuutena. Edellä olevassa kuvassa *(Kuva 5)* esitetään osat, joista vaatimuksen ratkaisu koostuu. Pyyntöjen käsittelyssä on kaksi roolia: rekisteröity ja ylläpitäjä. Rekisteröity on verkkosovellusta (esimerkkitapauksessa *Wordpress*-sivusto) kutsuva käyttäjä. Ylläpitäjä on palveluntarjoajan edustaja, joka valitsee toteutetaanko pyynnöt ja kommunikoi rekisteröidyn kanssa. Tärkeimmät komponentit oikeuksien käytön kannalta on eritelty kuvaan tarkemmin.



Kuva 5: Oikeuksien käytönhallintakomponenttien riippuvuudet.

Halutessaan käyttää tietosuoja-asetuksen oikeuksiaan, käyttäjä navigoi *oikeuksen käyttöliittymä* -komponentin tarjoamalle näkymälle, jossa on sopiva lomake kullekin oikeudelle. Voimme olettaa käyttäjän olevan kirjautunut (tai kirjautuvan tässä vaiheessa) sisään järjestelmään. Lomakkeelle käyttäjä täyttää tarvittavat tiedot, jonka jälkeen hän voi lähettää lomakkeen *käytönhallintajärjestelmään*. Esimerkiksi käyttäessään oikeuttaan oikaista tietonsa, on pyyntöön syötettävä muuttuneet tiedot. Oikeuksien käyttöliittymä myös listaa käyttäjän aiemmat pyynnöt tiloineen, joten käyttäjä voi seurata pyyntönsä toteutumista siitä.

Käyttöliittymä välittää pyynnön taustajärjestelmälle, jonne se tallennetaan. Käytönhallintajärjestelmän tehtävä on säilöä oikeuksien käyttöpyynnöt, niiden tilat, ja kommunikoida ydinhenkilötietomoduulin kanssa pyyntöjen toteuttaminen. Taustajärjestelmään voi saapua monen eri edustajärjestelmän pyynnöt, mutta se kattaa aina yhden henkilötietorekisterin. Ylläpitäjä tarkkailee saapuneita oikeuksien käyttöpyyntöjä. Hän voi hyväksyä ne, kysyä lisätietoja, tai kiistää pyynnön. Pyyntöjen hyväksyminen voidaan myös automatisoida pyyntötyyppikohtaisesti, joten esimerkiksi *(V5: rekisteröidyn pääsy tietoihin)* ei

kuormita ylläpitoa. Kaikista pyynnön tapahtumista jää merkinta *henkilötietolokiin*, jotta voidaan kattaa asetuksen noudattamisen *osoittamisvelvollisuus*.

Tarkastellaan komponenttien teknistä toteutusta. Oikeuksien käyttöliittymä on kirjasto (*Wordpress*-liitännäinen esimerkkitapauksessa) ja käytönhallintakomponetti on oma järjestelmänsä. Käyttöliittymän *(O7: uudelleenkäytettävyys)* tulee siitä, että liitännäinen sisältää ainoastaan minimikokoisen *GUI*-kerroksen. Loput tyylit määrittyvät sivuston omista tyyleistä (teemasta). Verkkosivujen tapauksessa *CSS*-tyylit kaskadoituvat käyttäjän selaimessa sivupohjasta, johon oikeuksien käyttöliittymä on upotettu. Kirjaston tehtäväksi jää lisätä lomakkeet sivulle, hallita selaimessa tapahtuvaa tiedon asynkronista lataamista, ja pyyntöjen lähettäminen taustajärjestelmälle.

Taustajärjestelmä voi olla tavallinen käytäntöjen mukainen verkkosovellus. Se on itsenäinen ja käyttää omaa tietokantaansa, joten koko järjestelmä voi hyvin olla omalla palvelimellaan. Tämä piilottaa järjestelmän teknologiariippuuvuden muulta arkkitehtuurilta: kommunikaatio käydään *HTTPS*:n yli. Ylläpitäjiä varten taustajärjestelmällä on oma käyttöliittymänsä. Sen tavoite on olla mahdollisimman kevyt, mutta tarjota tarpeelliset näkymät oikeuspyyntöjen käsittelylle. Oikeuksien hallintajärjestelmän tulee osata kytkeytyä ydinhenkilötietomoduuliin. Jako toimii niin, että ydinmoduulissa on logiikka siitä, *miten* eri oikeuksien käyttö pannaan käytäntöön. Oikeuksien hallintajärjestelmä hallitsee *miksi* ja *koska*.

Tässä on esitelty prosessi ja arkkitehtuuri prosessia noudattavalle järjestelmälle. Tarkimmin eriteltynä tutkielman vaatimuksista komponentit kattavat *(V5.2)*:n, *(V5.3)*:n, *(V6.2)*:n, *(V7.2)*:n ja *(V8.2)*:n. Oikeuksien käytönhallinta ei itsessään ratkaise vielä kokonaisia vaatimuksia, mutta esitetty prosessi on välttämätön. Kuten luvussa pohdittiin, toteutuksen yksityiskohdat eivät ole välttämättömiä asetuksen noudattamiseksi. Esitellyt järjestelmät ovat esimerkkejä uudelleenkäytettävyyden ja skaalautuvuuden saamiseksi tavalla, joka toteuttaa vaatimukset.

5.4 Ydinhenkilötiedot

Ydinhenkilötietomoduuli on nimensä mukaisesti tutkielman arkkitehtuurin ydin. Se osaa käsitellä tietosuojan kannalta riskialteimmat henkilötiedot ja kontrolloi muita järjestelmiä. Moduulin ensimmäinen tehtävä on vähentää arkkitehtuurin *(O9: henkilötietojen levinnäisyyttä)*. Toinen tehtävä on toimia rekisteröidyn oikeuksien käyttöä koskevien vaatimusten ohjaajana. Ydinhenkilötietomoduuli myös säilöö tietoa henkilötietoon liittyvästä suostumuksesta; tarkemmin suostumus esitellään seuraavassa aliluvussa.

Tarkastellaan arkkitehtuurin kokonaiskuvaa. Taustajärjestelmiä voi yhden palvelun piirissä olla useita kappaleita. Näihin tulee voida tallentaa henkilötietoja tapausyrityksen tarpeiden mukaan. Edustapalvelimilla ei ole henkilötietoja esitetyssä arkkitehtuurissa.

Tätä taustajärjestelmien joukkoa on tarpeellista hallita ja rajoittaa. *(V2: tietojen minimointi)* sisältää jo sen, että henkilötietojen redundantti tallennus ei olisi helposti perusteltavissa.

Tähän tarpeeseen tutkielma esittää *ydinhenkilötietomoduulia*. Koko arkkitehtuurin sisällä käsiteltävät henkilötiedot voidaan jakaa kahteen ryhmään: *ydinhenkilötietoihin* ja henkilöön *liittyviin* tietoihin. Nämä rinnastuvat suoraan luvussa 4 esiteltyihin Hintzen anonyymiyden tasoihin. Ydinhenkilötiedot ovat *tunnistettuja*, ja henkilöön liittyvät tiedot ovat *A11-anonyymejä*. A11-anonyymi tieto erottui tunnistetusta henkilötiedosta siten, että rekisterillä ei ole tiedossa tapaa yhdistää pseudonymisoituja tietoja henkilöön. A11-tason tiedot voi kuitenkin yhdistää henkilöön ulkopuolisella *lisätiedolla*.

Kun tarkastellaan joukkoa henkilötietoja, niiden anonyymiyden taso on vähiten anonyymi tieto koko joukon sisällä olevista yksittäisistä tiedoista ja tietojen yhdistelmistä. Voimme siis jakaa joukon osiin niin, että tunnistetut tiedot ovat erillään A11-anonyymeistä. Koko arkkitehtuurin laajuinen anonyymiyden taso ei nouse, mutta kun yksittäisessä järjestelmässä on vain A11-anonyymiä tietoa, on tämä selkeä parannus järjestelmän tietosuojalle. Tällöin koko arkkitehtuurin laajuinen henkilötieto keskittyy, ja *(O9: henkilötietojen levinäisyys)* on pienempi.

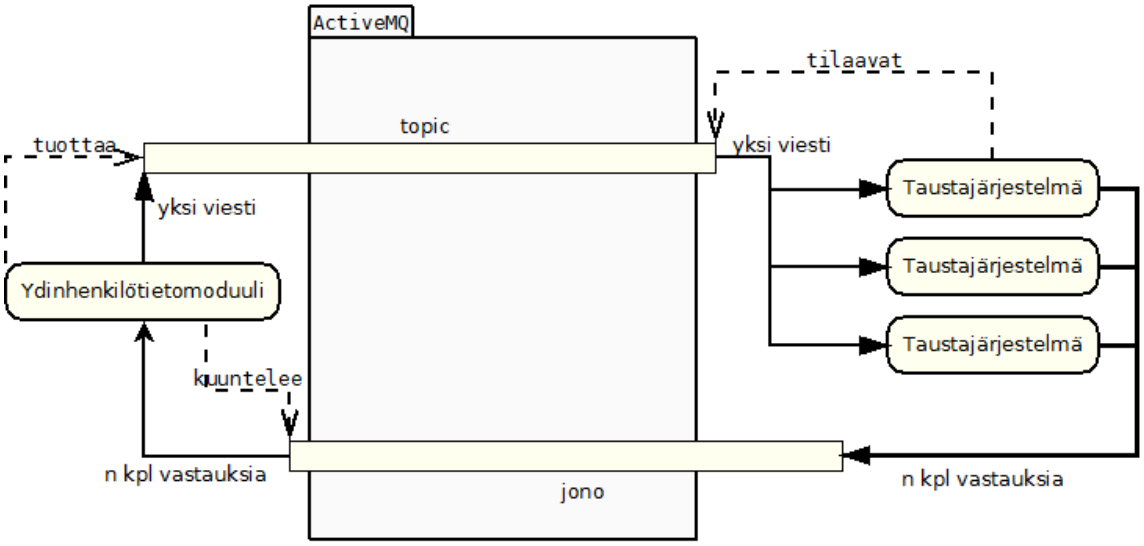
Ydinhenkilötietomoduuli siis tallentaa kaikki sellaiset tiedot, joista voidaan *välittömästi* tunnistaa rekisteröity. Muut taustajärjestelmät tällöin tallentavat vain viitaukset ydinhenkilötietoon, mutta voivat kuitenkin sisältää henkilöön välillisesti liittyvää dataa. Luonnollisesti kaikki järjestelmät saattavat tarvita ydinhenkilötietoja käsittelyyn. Ydinhenkilötietomoduulin onkin tarjottava rajapinta muille järjestelmille.

Toinen ratkaistava seikka oli rekisteröidyn oikeuksien ohjaaminen ja täytäntöönpaneminen. Oikeuksienhallintamoduuli välittää toteutettavat oikeuksien käyttöpyynnöt *(V5, V6, V7 ja V8)* ydinhenkilötietomoduulille. Moduulin tehtäväksi jää ydinhenkilötietojen muokkaaminen pyynnön mukaiseksi ja pyynnön välittäminen muille taustajärjestelmille. Tämä pitäisi voida tehdä niin, että riippuvuus ydinhenkilötietomoduulista muihin järjestelmiin minimoitaisiin ja ettei skaalautuvuuta rajoiteta pahasti. Haastava osuus on sanomien välittäminen luotettavasti ja yleiskäyttöisen skeeman määrittäminen.

Esitetään siis taustajärjestelmien ja ydinhenkilötietomoduulin välille rakennettavan viestinvälitysarkkitehtuuri. Tapauksessa monta järjestelmää kuuntelee yhdestä lähteestä tulevia sanomia. Tähän sopiva suunnittelumalli on *publish–subscribe* (julkaise–tilaa) [18]. Mallin haittapuoli on se, että oikeuksien käyttöpyyntöjen on mentävä perille luotettavasti. Siksi on perustettava myös kuittausmekanismi. Toinen osa viestien välityksestä on mekanismi *kerätä* henkilötiedot kaikista muistajärjestelmistä *(V5, V6)*.

Viestien välitys on yleinen arkkitehtuurihaaste. Siihen on jo olemassa ratkaisuja, myös avoimen lähdekoodin vaihtoehtoja. Tutkielman arkkitehtuuri ei itsessään sitoudu tiettyyn

viestinvälitysohjelmistoon, mutta suunnitelmat esittävät *Apache ActiveMQ*:n [12] kaltaisen palvelun käyttöä. Ydinhenkilötietomoduulin oheen pystytetään siis viestinvälityspalvelin. Ydinhenkilötietomoduulin suuntaan rajapinta on kaksi viestijonoa: sisään- ja ulosmenevät. Ulosmenevä julkaisee viestit *topic*-tyyppiseen kanavaan, jota muut taustajärjestelmät tilaavat. Näillä on siten myös kaksi rajapintaa viestinvälityksen suuntaan: tilattava topic ja viestijono, joka välittää sinne asetetut viestit ydinhenkilötietomoduulille. Edellä olevassa kuvassa (Kuva 6) visualisoidaan ehdotettu viestinvälitysarkkitehtuuri.



KUVA 6: Henkilötietosanomien viestinvälitysarkkitehtuuri.

Viestikanavat ovat yksinkertaisempi osa oikeuksien käyttöpyyntöjen toteutusta. Seuraavaksi määriteltävänä ovat prosessin kulku, sekä varsinaiset sanomat. Kutakin tässä täytettävää rekisteröidyn oikeutta kohden on oma sanomansa: *(V5: rekisteröidyn pääsy tietoihin)*:n täyttää *Hakusanoma*, *(V6: rekisteröidyn tietojen oikaisu)*:n *Oikaisusanoma* ja *(V7: rekisteröidyn tietojen poistaminen)*:n *Poistosanoma*. Kutakin vastaa myös omat vastaussanomat.

Yleisellä tasolla oikeuden toteuttamisen prosessi on yksinkertainen, tosin yhdellä huomiolla. Ydinhenkilötietomoduuli lähettää sanoman, kun sitä koskeva oikeudenkäyttöpyyntö on valittu toteutettavaksi. Muodostettu sanoma on yksi kolmesta aiemmin määritetystä tyypistä. Sanoma luetaan esimerkiksi *JMS*:n *(Java Message Service)* välityksellä topic-julkaisuun. Viestinvälityspalvelin välittää saapuvan sanoman kaikille topicin tilaaville järjestelmille. Nämä toteuttavat pyynnön: joko poistavat valitun henkilötiedon, päivittävät sen tietokantaan, tai noutavat kaikki liittyvät tiedot. Tämän valmistuttua kukin taustajärjestelmä muodostaa oman vastaussanomansa ja lähettää sen vastausjonoon. Vastauksia kerätessä ydinhenkilötietomoduuli pitää lukua ja lopulta esittää virheilmoituksen, jos vastauksia ei tule yhtä montaa mitä taustajärjestelmiä on.

Oikeuksien täytäntöönpanon prosessi on siten ehdotetussa arkkitehtuurissa täysin asynkroninen. Tämän haittapuoli on se, ettei pyynnön toteutusta ole mielekästä odottaa käyttäjän toimesta. Pyyntöjen toteuttamista ennen on jo kuitenkin ylläpitäjän hyväksymisvaihe, kuten kuvattu edellisessä luvussa. Asynkronisuus myös helpottaa suuren viestimäärän käsittelyä, mikä voi olla hyödyksi ehkä suurimmilla yrityksillä.

Esitellään seuraavaksi sanomien rakenteen määrittely. Arkkitehtuurin kannalta varsinaisen tiedostoformaatin valinnalla ei ole suurta vaikutusta; samat periaatteet toimivat *JSON*:in, *XML*:n, tai vaikka serialisoitavien Java-olioiden kanssa. Kaikilla eri sanomatyypeillä (haku, oikaisu, poisto ja vastaukset) on yhteiset perustiedot. Nämä ovat seuraavat: *UUID*, *viite-UUID*, *lähetysaika*, *tyyppi*, *henkilötiedon tunniste* ja *lähettävä järjestelmä*. Viite-UUID on vastaussanomien kenttä, jolla vastauksen voi yhdistää alkuperäiseen sanomaan. Henkilötiedon tunniste yksilöi tietyn rekisteröidyn kaikki tiedot koko arkkitehtuurin piirissä.

Vastaussanomillä on yleinen kenttä *status*-koodi. Käytetään *HTTP*:n koodeja, jollei syytä muihin ole. Yksinkertaiset sanomat ovat hakusanoma, poistosanoma, poistosanoman vastaus, sekä oikaisusanoman vastaus. Näihin kaikkiin riittää henkilötiedon tunniste ja status-koodi. Haastavammat sanomat ovat ne, joissa tulee kuljettaa varsinaista henkilötietoa. Lisätään molempiin sanomiin oma *hyötykuorma*-kenttänsä. Tämän sisältöllä ei ole merkitystä viestinvälityksessä, mutta riippuvuus on oikeuksien käyttöliittymän ja taustajärjestelmien välinen. Ei ole myöskään tarkoituksenmukaista, että ydinhenkilötietomoduuli tietäisi tasutajärjestelmien tietomallin, tai taustajärjestelmät toisensa.

Hakusanoman vastauksen ja oikaisusanoman hyötykuormat ovat toisistaan riippuvaisia. Kukin taustajärjestelmä määrittää oman sisältönsä hakusanoman vastaukseen, mutta sen on oltava käyttöliittymän tulkittavissa esitystä ja oikaisua varten. Oikaisusanoman hyötykuorma sisältää (järjestelmäkohtaisesti) muutettavien avain–arvo-parien uudet arvot. Avain–arvo-parien laajennukseksi voidaan määrittää käyttöliittymän ja taustajärjestelmien välille tyypittävä *DSL (Domain Specific Language)*. Tämän määritteleminen rajataan tutkielman ulkopuolelle mahdolliseksi jatkotyöksi. Periaate on kuitenkin se, että avain–arvo-pariin voidaan lisätä halutuille tietotyypeille lisämääreitä, joita käyttöliittymä osaa tulkita. Esimerkiksi pari {`a: b`} voidaan laajentaa muotoon {`a: {arvo: b, pituus: 50}`}. Tällöin käyttöliittymä, joka osaa tätä tulkita, voi estää oikaisun pidemmäksi kuin taustajärjestelmän odottama arvo.

Oikaisusanoma rakentuu siitä, kun rekisteröidylle esitetään hakusanomien tulokset. Käyttäjä tekee muutoksia, jotka lähetetään oikaisusanomana viestinvälitykseen. Koska ydinhenkilötietomoduuli julkaisee sanoman kaikille taustajärjestelmille, on sen hyötykuormassa tarpeen eritellä kullekin kuuluvat tiedot *lähettävä järjestelmä*-kentän mukaan. Hyötykuormaan sisällytetään ainoastaan ne avaimet, joiden arvo on muuttunut. Taustajärjestelmä odottaa siis tiedon tulevan samassa muodosa, kuin missä se hakuvastauksessa lähti.

Esimerkit kaikista tässä määritellyistä sanomista ovat tutkielman liitteenä. Ne voivat havainnollistaa tietojen asettelua sanomien kulun prosessisa, mutta jätettiin tästä pois tilan säästämiseksi.

Viestinvälityksen lisäksi toinen ratkaistava seikka on ydinhenkilötietomoduulin tarjoama rajapinta muille järjestelmille. Ydinhenkilötietomoduulin on voitava säilöä ennalta määriteltyjen yleisten ydinhenkilötietojen (esimerkiksi nimi, hetu, tai osoite) lisäksi taustajärjestelmäkohtaisia ydinhenkilötietoja. Lienee mahdotonta määrittää etukäteen yhteinen ydinhenkilötietomalli, joka kattaisi kaikki käyttötapaukset. Siispä taustajärjestelmät voivat tallentaa omia ydinhenkilötietojaan moduuliin oman mallinsa mukaisesti.

Taustajärjestelmät ovat ydinhenkilötietomoduulin asiakkaita. Moduuli tarjoaa *REST*-tyyppisen rajapinnan, jolla tietoja luetaan ja päivitetään. Rajapinnan on myös tarjottava mahdollisuus tehdä hakuja.

Ydinhenkilötiedon juuri on uniikki tunniste. Tämä on sama tunniste, johon viitataan viesteissä taustajärjestelmille. Yksittäinen ydinhenkilötieto on verrattavissa luvussa 4.1 esiteltyyn asiakastietueehen. Muuten tietomalli rakentuu dokumenttityyppisesti; juurena olevalla tiedolla on attribuutteja, jotka sisältävät tietoa avain–arvo -mallilla. Arvoilla taas on omat attribuuttinsa, jotka sisältävä omia käsitteitään. Täten juuresta attribuutteja seuraamalla syntyy rakenne, joka kattaa kaikki yhtä henkilöä koskevat ydinhenkilötiedot. Tietomallia voi kuvata esimerkiksi *JSON*:illa. Dokumenttityyppisiä tietokantoja on useita [50], eikä tutkielman arkkitehtuuri ota kantaa tietyn tietokantaohjelmiston valintaan.

Määritetään ydinhenkilötiedon skeema seuraavasti. Juuritietueen attribuuteissa hyödynnetään Gjermundrød et al. esittämiä tunnistetietoja. Kentät ovat: id, luomisaika (globaali), luomisaika (lokaali) ja vanhenemisaika. Näiden lisäksi lisätään tieto *vastustamisen (V8)* aloitusajankohdasta. Näiden lisäksi tietueella ovat varsinaiset henkilötiedot. Attribuutin *yleiset tiedot* alle asetetaan tavanomaisia ydinhenkilötietoja, joille voidaan nähdä käyttöä useammassa taustajärjestelmässä. Tällä vähennetään redundanssia ja ylläpidon määrää. Juuritietueen attribuutin *laajennukset* alle asetetaan taustajärjestelmäkohtaiset ydinhenkilötiedot, kukin järjestelmän yksilöivän tunnisteen alle. Näiden skeemaa ei määritellä, vaan ne ovat taustajärjestelmän itse dynaamisesti asetettavissa.

Huomioidaan myös se, että poistosanoman käsittelyssä ei poisteta *koko* ydinhenkilötietuetta. Jätetään tunniste, luomisajat ja lisätään *poistettu*-lippu. Näille on tarve myöhemmin, jos on osoitettava, että asetusta on noudatettu. Henkilötietolokin viittaukset ovat silloin yhä kohdistettavissa tietoon.

Edellä olevassa kuvassa *(Kuva 7)* on esimerkki ydinhenkilötiedoista. Kuvasta voidaan havaita sitä, miten tiedot asettuvat dokumenttitietomalliin. Tämän etu relaatiomalliin on se, ettei skeemaa tarvitse määrittää etukäteen. Luonnollisesti tietojen käyttäjällä on oltava ajatus tiedon rakenteesta, mutta taustajärjestelmäkohtaiset laajennukset eivät vaikuta

ydinhenkilötietomoduuliin, eivätkä muihin järjestelmiin. Esimerkin tapauksessa tietueella on muutama yleisiä ydinhenkilötietoja ja yksi laajennus taustajärjestelmältä ”SERVICE-1”.


```
{
  "id": "39a369f3-33c8-4eb8-90ff-d94b40c05b4e",
  "luomisaika_globaali": "2017-10-05T20:12:43.511Z",
  "luomisaika_lokaali": "2017-10-05T20:12:43.511Z",
  "vanhenemisaika": "2020-10-05T20:12:43.511Z",
  "vastustaminen": null,
  "yleiset": {
    "hetu": "090100A706C",
    "etunimi": "Maija",
    "sukunimi": "Meikäläinen",
    "osoitteet": []
  },
  "laajennukset": {
    "SERVICE-1": {
      "rekisteritunnus": "ABC-123"
    }
  }
}
```


Kuva 7: Havainnollistus mahdollisesta ydinhenkilötiedon rakenteesta.

Viestinvälityksen ja ydinhenkilötietojen hallinnan jälkeen moduulin rakenne on määritelty. Tästä huomattiin riippuvuus tietomallin osalta edellisen luvun oikeuksien hallintamoduulin käyttöliittymään. Oikeuksien hallinta on kuitenkin itsenäinen konseptinsa, mutta tiukan riippuvuuden vuoksi moduulit voisivat olla yhteisessä järjestelmässä. Tällöin kommunikaatio ydinhenkilötietomoduulin ja oikeuksien hallinan välillä on yksinkertaisempaa. Huomioiden sen, että oikeuksienkäyttöpyynnöt itsessään sisältävät ydinhenkilötietoja (oikaisut), jaettu tietokanta kuulostaa hyvältä riskienhallinnan vuoksi.

Arvioidaan mallin toteuttamista. Oikeuksien käyttöpyyntöjen toteuttaminen on tehtävä tietosuoja-asetuksen myötä välttämättä. Ydinhenkilötietomoduulin viestinvälitysmekanismi ei sellaisenaan ole pakollinen, mutta jokin tapa saada oikeuspyyntö taustajärjestelmille on oltava. Ydinhenkilötietojen keskitetty säilytys ei ole *vaatimus*, vaan laatuasia. Suuren olemassaolevan palvelun uudistaminen kokonaan tutkielman arkkitehtuurin malliin voi olla liian työlästä. Tällöin mahdollista olisi jättää ydinhenkilötietojen keskitys pois ja toteuttaa ainoastaan viestinvälitys. Tämä olisi kompromissi tietosuojan osalta, mutta asetus salliikin *toteuttamiskustannusten* huomioon ottamisen *(V1)*:n osalta. Kuitenkin on tarpeen sitoa kaikki eri järjestelmien yhtä henkilöä koskevat tiedot tiettyyn tunnisteeseen, jotta oikeuksien käyttöpyynnöt voidaan yhdistää oikeaan tietoon.

Luvussa esitetty moduuli on kriittinen osa tutkielman arkkitehtuuria vaatimusten toteuttamiseksi. Se ratkaisee täytäntöönpanemisen vaatimuksille *(V5: rekisteröidyn pääsy tietoihin)*, *(V6: rekisteröidyn tietojen oikaisu)*, *(V7: rekisteröidyn tietojen poistaminen)* ja

(V8: rekisteröidyn vastustamisoikeus). Lisäksi mahdollisesti muuten redundanttien henkilötietojen kerääminen yhteen paikkaan on etu vaatimuksen *(V2: tietojen minimointi)* kannalta.

Ratkaisussa riskialtteimmat henkilötiedot, *tunnistetut* henkilötiedot, eristyvät yhteen moduuliin koko arkkitehtuurin laajuudelta. Tämä on merkittävää tapausyrityksen kaltaisissa tilanteissa, joissa on tarve käsitellä henkilötietoja monessa erillisessä järjestelmässä. Ominaisuuden *(O9: henkilötietojen levinneisyys)* väheneminen voidaan nähdä suoraan parannuksena tietosuoja-asetuksen vaatimuksen *(V1: järjestelmän tietosuoja)* noudattamisessa.

5.5 Suostumuksen hallinta

Tietosuoja-asetuksen vaatimus *(V3: suostumuksen hallinta)* tulee jotenkin ottaa huomioon useimmissa verkkopalveluissa. Suostumus henkilötietojen käsittelyyn on kysyttävä rekisteröidyltä selkeästi erillään ja ymmärrettävästi. Käytännössä tämä on tehtävä palveluun rekisteröitymisen yhteydessä. Suostumuksen peruuttaminen on oltava mahdollista yhtä helposti, kuin sen antaminen. Erillinen haaste suostumuksen hallinnassa on alaikäisten suostumus. Jotta lapsen henkilötietoja voidaan käsitellä, tulee kysyä suostumus vanhemmalta.

Arkkitehtuurin on vastattava suostumuksen hallinnan teknisiin kysymyksiin. Kuten rekisteröidyn oikeuksienhallinnassa, tästäkin vaatimuksesta erottuvat käyttöliittymä ja taustakäsittely. Suostumuksen tila (ja sen muutokset) on tallennettava ja tarkastettava ennen tietojen käsittelyä. Rekisteröidyn on voitava hallita omaa suostumustaan, ja sen päätyttyä tulee tehdä *(V7):n* kaltainen tietojen poistaminen.

Liiketoiminnallisesti lienee suotavaa jaotella suostumusta henkilötietojen käsittelyyn tarkemmin, kuin yksittäisenä totuusarvomuuttujana. Tämä on jälleen selkeästi tapauskohtainen seikka, mutta monessa tilanteessa rekisteröidyn kieltäytyminen vain osasta käsittelyä on toivotumpaa kuin koko palvelusta poistuminen. Tällöin *suostumus* olisi tiettyä henkilöä koskeva henkilötietorekisterikohtainen käsite, joka sisältää useita (ennalta määritettyjä) suostumuskategorioita. Sosiaalisen median palvelu voisi esimerkiksi tarjota suostumuskategorioina ”julkaisujen esittämisen” ja ”markkinoinnin kohdentamisen”. Tällöin käyttäjä voisi kieltäytyä ainoastaan markkinoinnin kohdentamisesta, sallien palvelun kuitenkin pitää käyttäjän. Toisissa tapauksissa suostumus on rekisterinpitäjän kannalta parempi pitää ”kaikki tai ei mitään”-tyyppisenä, joten uudelleenkäytettävyyden kannalta näiden on oltava konfiguroitavissa. Vaatimusten monimuotoisuus johtuu ominaisuudesta *(O1: jako asiakkaiden välillä)*; eri asiakkailla on eri liiketoimintatarpeet.

Suostumuksen kerääminen on prosessi, jossa ennalta tuntematon henkilö myöntyy

rekisteröidyksi. Lopputuloksena voimme tallentaa liitoksen *johonkin* henkilötietoon ja aiemmin määriteltyyn *suostumukseen*. Haastavinta tässä on se, että voimme varmistua suostumuksen antajan olevan henkilötietojen omistaja (tai tämän vanhempi). Suostumuksen keräämisen jälkeen suostumuksen tilan ja henkilötietojen muuttaminen on yksinkertaisempaa.

Henkilötietojen kerääminen tallentamalla rekisteröidyn ilmoittamia tietoja on väistämättä riskialtista. Verkkopalveluiden on siis perustuttava jonkinasteiseen luottamukseen. Tietosuoja-asetuksen teksti ei anna tässä helpotuksia, paitsi alaikäisen suostumuksen tapauksessa. Artikla 8:n kohta 2 on seuraava: *"Rekisterinpitäjän on toteutettava kohtuulliset toimenpiteet tarkistaakseen tällaisissa tapauksissa, että lapsen vanhempainvastuunkantaja on antanut suostumuksen tai valtuutuksen, käytettävissä oleva teknologia huomioon ottaen."* Mikä *kohtuulliset toimenpiteet* tarkoitavat jää vielä nähtäväksi ennakkotapauksien mukaan. Aikuisen rekisteröityessä voidaan pyytää häntä vakuuttamaan annettujen tietojen pitävän paikkansa, ja olemaan valmis reagoimaan mahdollisiin väärinkäytöksiin. Mikäli identiteettivarkaus tapahtuu, syyllinen on kuitenkin varas itse.

Tutkielmassa esitetään suostumuksen hallintaan *idealistinen* ratkaisu: rekisterienvälinen suostumuksenhallinta. Käyttäjille tai rekisterinpitäjille tulisi tästä etuja vasta laajan adoption jälkeen, mikä on enemmän vaadittu kuin käytännössä voidaan odottaa. Esitettävä malli on kuitenkin mahdollista nostaa esimerkiksi valtakunnallisen standardin tasolle, jolloin se tarjoaisi helpon vastauksen yksittäisen rekisterin perustajalle. Kommentoidaan kuitenkin myös käytännöllisempää ratkaisua suostumuksenhallintaan luvun lopulla.

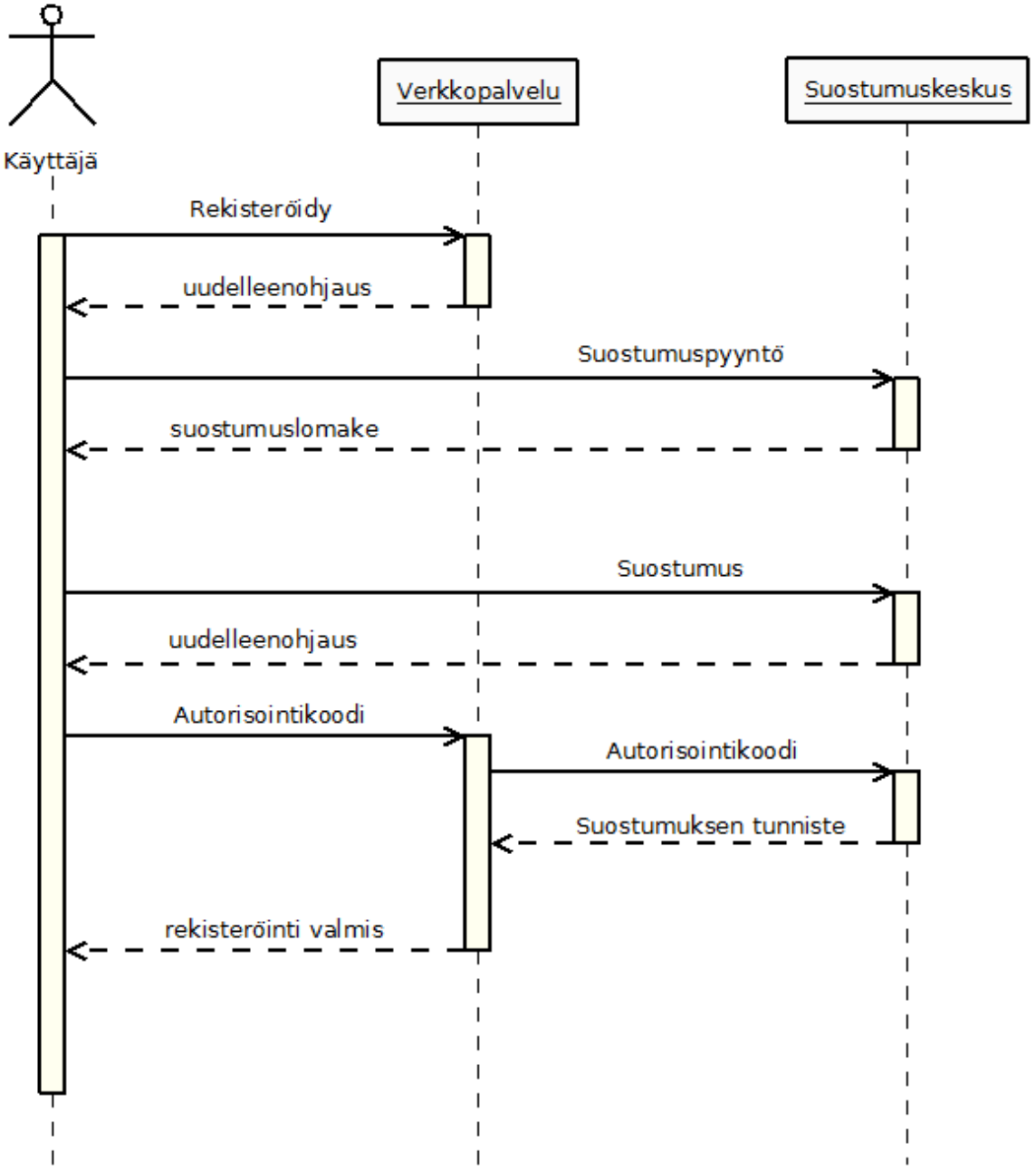
Tavoitteena on yksi paikka, jossa on tiedossa rekisteröityjen henkilöllisyys vahvalla tunnistautumisella. Järjestelmä on kaikkien rekisterien välinen suostumuskeskus. Tästä syntyy henkilöllisyyden ja suostumuksen välille luottamusketju: suostumusketju luottaa vahvaan tunnistautumiseen ja muut rekisterit luottavat suostumuskeskukseen. Olisi kohtuutonta esittää, että yksittäinen verkkopalvelu edellyttäisi käyttäjiltä vahvaa tunnistautumista. Se häviäisi kilpailussa, jolleivat muut sivut edellyttäisi tätä myös. Keskitetyn suostumuksen hallinnan avulla tämä ratkeaisi. Käyttäjän tarvitsisi tunnistautua vahvasti kerran suostumuskeskukseen rekisteröityessä, jonka jälkeen kaikki rekisterit voivat nojata siihen. Keskus ratkaisee myös vanhempien suostumuksen keräämisen mallintamalla vanhemmuussuhteen henkilöiden välillä.

Esitellään suostumuksen kerääminen tutkielman ratkaisussa. Kerääminen tapahtuu OAuth 2:n [16] kaltaisella prosessilla, jota on sovellettu aihealueen tarpeisiin. Prosessissa on kolme tahoa: käyttäjä, verkkopalvelu (rekisterinpitäjä) ja suostumuskeskus. Suostumuksen kerääminen edellyttää, että verkkopalvelu on tallennettu asiakkaaksi suostumuskeskukseen. Asiakkaaksi rekisteröinnin yhteydessä suostumuskeskukseen tallennetaan käyttäjälle näytettävät tiedot ja palvelun suostumuskategoriat. Suostumuskeskus antaa verkkopalvelulle oman *salaisen avaimen* ja *tunnisteen*. Prosessi on helppokäyttöisempi,

mikäli käyttäjä on ennalta rekisteröitynyt ja vahvasti tunnistautunut suostumuskeskukseen. **Huomautus:** tutkielman ratkaisu on erittäin samankaltainen luvussa 4.3 esitellyn kanssa (viite [43]). Molemmat ovat kuitenkin itsenäisesti kehiteltyjä. Tutkitaan eroavaisuuksia tämän luvun lopussa.

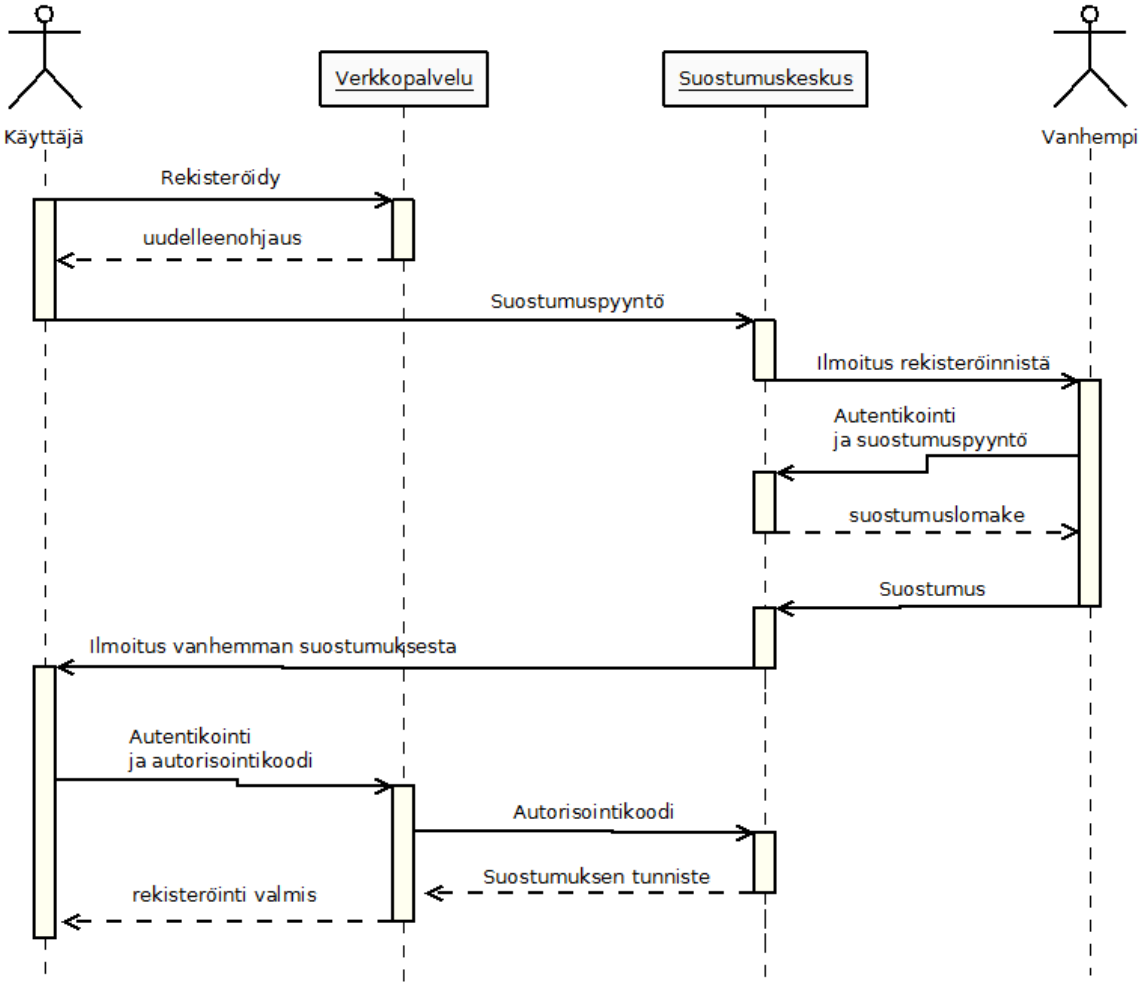
Suostumuksen keräämisen kulku on seuraava. Käyttäjä aloittaa prosessin kun hän valitsee rekisteröityä verkkopalveluun. Verkkopalvelu tietää tilanteen vaativan henkilötietojen käsittelyä ja siten suostumuksen. Verkkopalvelu lähettää käyttäjälle uudelleenohjausvastauksen suostumuskeskuksen sivulle. Vastauksessa annetaan parametrina *paluuosoite* ja palvelun tunniste. Käyttäjä lataa suostumuskeskukselta lomakkeen, jossa suostumus henkilötietojen käsittelyyn pyydetään. Mikäli käyttäjä ei olisi jo ennalta rekisteröitynyt keskukseen, hän tekisi sen ennen suostumuslomakkeen saamista.

Lomake lähetetään täytettynä suostumuskeskukselle, joka tallentaa suostumuksen järjestelmäänsä. Käyttäjälle palautetaan verkkopalvelun uudelleenohjausosoite, jolle on lisätty parametrina *autorisointikoodi*. Ladatessaan rekisteröinnin valmistumissivun, verkkopalvelu validoi autorisointikoodin sopimuskeskukselta. Tähän annetaan mukaan verkkopalvelun salainen avain. Vastauksena annetaan *suostumuksen tunniste*, johon verkkopalvelu voi viitata myöhemmin. Siten suostumus on kerätty, ja käyttäjän henkilöllisyys on vahvasti tiedossa, eikä verkkopalvelu itse sitä joutunut selvittämään. Edellä olevassa kuvassa *(Kuva 8)* on prosessin sekvenssikaavio.



KUVA 8: Suostumuksen keräämisen kulku.

Toinen skenaario suostumuksen keräämisessä on vanhemman suostumuksen kerääminen, kun rekisteröityvä henkilö on alaikäinen. Tässä lähtökohtana on suostumuskeskukseen rekisteröidyt vanhempi ja lapsi, joiden perhesuhde on tiedossa. Prosessi ei ole yhtä käyttäjäystävällinen, koska se ei voi edetä yhdessä vaiheessa. Ei voida odottaa vanhemman olevan valmiina odottamassa, kun käyttäjä (lapsi) päättää rekisteröityä palveluun. Käyttäjälle tulee siis tauko verkkopalvelun käytössä vanhemman suostumuksen keräämisen ajaksi. Suostumuksen keräämisen kulku on muuten samankaltainen kun aikuisen tapauksessa. Prosessi on kuvattu edellä olevassa kuvassa *(Kuva 9)*.



Kuva 9: Vanhemman suostumuksen keräämisen kulku.

Vanhemman suostumuksen kerääminen alkaa jälleen käyttäjän (lapsi) pyynnöstä rekisteröityä verkkopalveluun. Verkkopalvelu tunnistaa henkilötietojen käsittelyn tarpeen ja uudelleen ohjaa käyttäjän suostumuskeskuksen sivulle. On huomion arvoista, ettei verkkopalvelun tarvitse tietää rekisteröityvän ikää, tai vanhemman suostumuksen tarpeen tilaa. Kun käyttäjä kirjautu suostumuskeskukseen, on siellä tiedossa tarve vanhemman suostumukselle. Tällöin, poiketen kuvan *(Kuva 8)*:n prosessista, ei suostumuslomaketta palvellakaan käyttäjälle. Käyttäjän palvelun käyttö jää odottamaan vanhemman suostumusta, jolle taas lähetetään esimerkiksi *push notification* -tyyppinen ilmoitus.

Lopulta vanhempi myöntää suostumuksen, jolloin suostumuskeskus lähettää käyttäjälle ilmoituksen. Mukana tulevat alkuperäisen suostumuspyynnön uudelleenohjausosoite ja autorisointikoodi. Tietojen avulla käyttäjä voi lopulta saattaa rekisteröinnin valmiiksi ja käyttää palvelua. Autorisointikoodin tuominen verkkopalvelulle on tässäkin verkkopalvelun kannalta läpinäkyvää. Kokonaisuutena verkkopalvelun tarvitsee toteuttaa ainoastaan yksi suostumuksenkeräämisprosessi, jota molemmat skenaariot noudattavat. Suostumuskeskus siten nostaa taakan verkkopalveluilta.

Suostumuksen hallinnassa on keräämisen lisäksi kaksi käyttötapausta: *suostumuksen validointi* ja *suostumuksen peruminen*. Validointi tarkoittaa tapaa tarkastaa suostumuksen voimassaolo, eli verkkopalvelun on voitava kysyä suostumuskeskukselta onko tietty suostumuksen tunniste yhä voimassa. Suostumuksen perumisen tietosuoja-asetus velvoittaa olevan yhtä helppoa, kuin sen antamisen.

Kirjautumisen validointi on myös huomioitu OAuth 2 -protokollassa ja se vaikuttaisi toimivan myös suostumuksen tapauksessa. Suostumuksen keräämisen loppuvaiheessa verkkopalvelu toimittaa käyttäjän autorisointikoodin suostumuskeskukselle. Tämän vastauksena tulee suostumuksen tunniste, mutta voimme palauttaa lisäksi *voimassaoloajan*. Suostumuskeskuksen on tarjottava rajapinta, jolla verkkopalvelut voivat uusia vanhentuneen suostumuksen tunnisteen – tämän keskus luonnollisesti palauttaa ainoastaan, jos suostumus on yhä voimassa. Sopiva validointiväli on valittavissa oleva asia, joka voi olla esimerkiksi pienempi alaikäisille.

Suostumuksen peruminen voi tapahtua kahdella tavalla: verkkopalvelusta ja suoraan suostumuskeskuksesta. Verkkopalvelusta initioitava suostumuksen peruminen noudattaa samankaltaista prosessia kuin suostumuksen kerääminen. Tietosuoja-asetus velvoittaa sen olevan helppoa, joten verkkopalvelun on tarjottava palvelu tähän. Suostumuskeskuksesta taas suostumuksen on voitava perua vähintään vanhemman, jota ei voida edellyttää käyttävän verkkopalvelua. Mikäli verkkopalveluille annetun suostumuksen tunnisteen voimassaolo on riittävän lyhyt, ei perumisessa tarvita viestiä verkkopalvelun suuntaan. Verkkopalvelusta alkavaan perumiseen on yksinkertaisempaa lisätä paluuosoite, jonka kautta peruminen tulee verkkopalveluunkin voimaan välittömästi.

Tässä on kokonaisuudessaan tutkielman esitys suostumuksen hallinnasta. Kuten on kuvattu, suostumuskeskus poistaa verkkopalveluiden toteutuksesta vaikeat osat suostumukseen liittyvien vaatimusten täyttämisestä. Verkkopalveluiden ei tarvitse vahvasti tunnistaa käyttäjiä, eikä tietää heidän ikäänsä. Toteuttamalla luvussa esitetyn mallin, verkkopalvelut kattavat tietosuoja-asetuksen vaatimuksen *(V3: suostumuksen hallinta)*.

Haittapuolena tälle suostumuksenhallintamallille on se, että suostumuskeskuksesta saatavat edut materialisoituvat vasta, kun se saavuttaa laajan käytön. Suostumuskeskuksen ei välttämättä ole pakko olla uusi palvelu: se voitaisiin toki tehdä laajennuksena olemassaoleviin laajassa käytössä oleviin palveluihin. Samankaltaisesta toiminnasta on ennakkotapausta: esimerkiksi *Facebook* [27] ja *Google* [28] tarjoavat kirjautumismahdollisuutta muille sovelluksille. Nämä eivät nykyisellään kata tätä tapausta, mutta ne voisivat kehittyä suostumuskeskuksiksi.

Mikäli esitetty suostumuskeskusmalli ei ole vaihtoehto, on arkkitehtuuriin mahdollista upottaa suostumuksen hallinta verkkopalvelukohtaisesti. Luvun 5.3 oikeuksien käytönhallinta

voi tallentaa taustajärjestelmäänsä myös suostumustiedot ja tarjota käyttöliittymän suostumuksen myöntämiseen/poistamiseen. Tällöin henkilöllisyyden varmentaminen perustuu luottamukseen käyttäjää kohtan, ja vanhemman suostumusta ei voida täysin validoida.

Kuten huomioitu aiemmin, luvussa esitetty ratkaisu perustuu samaan ajatukseen OAuth 2:n käytöstä, kuin luvussa 4.3 esitelty artikkeli. Tutkielman ratkaisu yksinkertaistaa tuota sillä, että tavoite on saada suostumus vain alkuperäiselle tiedonkeruulle. Luvun suostumuskeskus on hyvin samankaltainen kuin MyData-operaattori. Mikäli MyData-arkkitehtuuri nousee suosioon, ovat operaattorit itsessään hyviä vaihtoehtoja suostumuskeskukseksi. Tällöin suostumuksen hallinta tiedonsiirtoihin palveluiden välillä on lisähyöty.

Vanhemman suostumuksen kerääminen on alue, johon Su et al. eivät ota kantaa. Tämän laajennuksen lisääminen MyData-operaattorin palvelun rekisteröintiin ei pitäisi olla vaikeaa, samankaltaisena prosessina kuin tässä luvussa on kuvattu. Tutkielman suostumuskeskus vaatisi MyData-operaattorin: olevan luotettava taho, vahvasti tunnistavan käyttäjän ja tietävän vanhemmuussuhteet. Nykyinen MyData-käyttäjätilin spesifikaatio [44] ei sisällä vanhemmuussuhteita toisiin tileihin.

Kokonaisuutena voidaan todeta, että OAuth 2:n käyttö suostumuksen keräämiseen ei ole ennennäkemätöntä. Esitetään kuitenkin, että se on sopiva osa tutkielman arkkitehtuuriin. Onko suostumuskeskuksen tuottaja lopulta MyData-operaattori, vai eri taho, on arkkitehtuurin kannalta yksityiskohta.

5.6 Staattinen lähdekoodin analyysi

Tietosuoja-asetuksen vaatimus *(V1: järjestelmän tietosuoja)* edellyttää muun muassa *asianmukaisia toimia tietosuojalle*, *luottamuksellisuutta* ja *testausmenettelyä tietosuojalle*. Tietosuojan laatu ei ole binäärinen ominaisuus, vaan sitä voidaan *parantaa* eri keinoin. Yksi näistä on järjestelmien lähdekoodin staattinen analyysi [33].

Staattisen analyysin käyttöä on tutkittu koodin laadun parantamiseksi (esimerkiksi Dixon [8]) ja verkkopalveluiden tietoturvan välineeksi (esimerkiksi Huang et al. [21]). Holvitie ja Leppänen tutkivat staattista analyysia teknisen velan leviämisessä ja dokumentoinnissa [19]. Tässä tutkielmassa taas ehdotettaan staattisen analyysin keinojen käyttämistä henkilötietojen hallintaan ohjelmien sisällä.

Arkkitehtuurin osa tästä on linjaus siitä, että staattista analyysiä käytetään kaikissa taustajärjestelmissä (ja luvussa esitellyissä moduuleissa). Tällöin vaatimuksen *V1* täyttymisen ollessa kyseenalaisena, voidaan perustella lähdekoodin täyttävän laatuvaatimukset analyysin avulla. Se ei suoranaisesti *todista* mitään, mutta on vakuuttavampaa kuin vain luottaminen koodin virheettömyyteen.

Tiedossa ei ole, että työkaluja tietosuojan staattiseen analyysiin olisi jo olemassa. Tässä luvussa määritellään sellaisen tarve ja vaatimukset. Tutkielmaan liittyvässä työssä toteutetaan *proof of concept* tällaisesta välineestä.

Tarve tietosuojaa validoivalle staattiselle anylyysille on ohjelmointivirheiden vähentäminen. Tietosuoja voi vaarantua, mikäli henkilötietoja pääsee epähuomiossa esimerkiksi lokitiedostoihin, tai muuten väärinkäytetään. Viimeaikainen tapaus ohjelmointivirheestä syntyneestä tietoturvariskistä oli, kun suositun käyttöjärjestelmän levyn dekryptauksen salasanadialogissa esiintyi vihjeen sijaan itse salasana [32] (uutisartikkeli). Tämä johtui yksinkertaisesti siitä, että data valui väärään kenttään ohjelman kulussa. Samanlaisten tapauksien voi helposti kuvitella johtavan tietosuojaloukkauksiin.

Tutkielman esittämän työkalun tarkoitus on kaksijakoinen: 1) *dokumentoida*, mitkä lähdekoodin luokat ovat henkilötietoja ja 2) *validoida*, ettei henkilötitoa sisältäviä luokkia käsitellä huomaamatta. Tämä on suunniteltu vahvan tyypityksen kannalta ja olettaa lähdekoodia voitavan annotoida. Esimerkki analysoi Javakoodia.

Yleinen verkkosovellusten tekniikka on *ORM (Object-relational mapping)*, jossa tiettyä tietokannan relaatiota esittää luokka, jonka oliot vastaavat sen rivejä. Järjestelmän sisällä tietoa siirretään tietoa kuvaavilla olioilla (*Data Transfer Object*). Näitä käyttäviin järjestelmiin on hyödyllistä dokumentoida henkilötietoja sisältävät luokat. Ohjelmiston kehittäminen on yksinkertaisempaa, kun henkilötiedot on merkitty selvästi – ei tarvitse tehdä harkita joka tapauksessa aina uudelleen.

Ehdotetaan tavaksi näiden luokkien merkintään annotaatiota `@PersonalData` (henkilötieto). Annotaatiolla voidaan dokumentoida sekä *ORM*-entiteetin luokka (esimerkiksi ”Henkilö”), että yksittäisen henkilötiedon arvoa kuvaava luokka (esimerkiksi ”Henkilötunnus”). Tämä itsessään riittäisi, jos tavoitteena olisi ainoastaan dokumentoida henkilötiedon sijaintia lähdekoodissa, millä toki voisi olla arvoa pidemmissä projekteissa. Annotointi kuitenkin mahdollistaa dokumentoinnin lisäksi seuraavan vaiheen, käsittelyn validoinnin Javan annotaatioprosessorilla.

Tavoitteena on luoda käännösaikainen mekanismi, joka varoittaa kehittäjää, jos hän on käyttänyt henkilötietoa sisältävää luokkaa huomaamattaan. Otetaan `@PersonalData`-annotaation pariksi käyttöön toinen: `@PersonalDataHandler` (henkilötiedon käsittelijä). Luokat tai metodit, joiden on tarkoitus käsitellä henkilötietoja, merkitään tällä annotaatiolla.

Nyt voidaan käännöksen yhteydessä tarkastus, että kaikki henkilötietoja käsittelevät operaatiot ovat kehittäjän tiedossa. Tarkastetaan paikat, joissa `@PersonalData`-annotoituja tyyppejä olevia olioita käsitellään. Tarkasteltavana ovat muuttujat, parametrit ja metodikutsut. Mikäli muuttuja/parametri on tyyppiä, joka on `@PersonalData`-annotoituja, on

muuttujan sisältävän metodin tai luokan oltava `@PersonalDataHandler`. Virheestä esitetään varoitus säännöstä poikkeavan arvon kohdalla (jos käytössä on *IDE*). Palaute tulee kehittäjälle siis nopealla syklillä ja korjaukset ovat vaivattomia.

Oletettavasti staattisen henkilötietojen suojaamisen edut tulevat esiin vasta suurempien projektien aikana, erityisesti sellaisten, joiden jatkokehitys kestää vuosia. Näissä koodin laatua parantavat arkkitehtuurivalinnat helpottavat kehittäjien vaihtoa ja annotoinnin tuoma (koodin mukana ajantasalla oleva) dokumentaatio lisäävät itseluottamusta muutoksiin. Vaatimuksen *(V1: järjestelmän tietosuoja)* toteutumisen osoittamista voi perustella esitellyllä työkalulla. Luonnollisesti annotaatioiden hyöty riippuu siitä, kuinka tunnollisesti kehittäjät niitä käyttävät. Henkilödatavirtojen dokumentointi annotoimalla on kuitenkin pieni vaiva tietosuojan edestä.

5.7 Pseudonymisointiprosessi

Tutkielman tapausyrityksen ominaisuus *(O5: staging-ympäristön vaatimukset)* kuvastaa toimintatapaa, jossa projektien jatkokehityksessä toteutettavat ominaisuudet käyvät *staging*-ympäristössä hyväksyntätestattavana ennen julkaisua. Ympäristöjä esiteltiin luvussa 2. *Staging*-ympäristön olisi oltava mahdollisimman tuotantoympäristöä vastaava, kuitenkin niin, että tietosuoja-asetuksen vaatimukset täyttyvät.

Eniten vaatimuksista konfliktissa *staging*-ympäristön kanssa on *(V2: tietojen minimointi)*. Tarkastellaan komponentteja, joista *ympäristö* koostuu luvun 2 määritelmän mukaan: ohjelma, ympäristö ja tietokanta. Henkilötietojen minimoinnin kannalta ohjelma ei ole relevantti – olettaen ettei lähdekoodissa ole henkilötietoja, mikä ei olisi ammattimaista. Ympäristössä henkilötietoja voi esiintyä väliaikaisesti (esim. muisti), mutta pohjapiirrustuksessa ympäristön pystyttämisestä ei ole henkilötietoja. Staging ympäristössä voi olla siis huoletta tuotantoa vastaavat ohjelma ja ympäristö. Tietokanta on kyseenalainen kohta, johon tutkielmassa perehdytään. Tässä luvussa esitetään *pseudonymisointiprosessi*, joka sopii kummankin tahon vaatimuksiin.

Pseudonymisointiprosessi on transformaatio, jossa syötteenä annetusta tuotantotietokannasta (kopiosta) muunnetaan riisuttu versio staging-ympäristöön asennettavaksi. Luvun päättelyssä käsitellään relaatiotietokantaa, mutta tuloksia voi soveltaa muihinkin tyyppeihin. Muunnoksen lopputuloksen raja-arvot ovat selvät: enimmillään tulos on vain skeema ilman dataa, vähimmillään tuotantotietokanta sellaisenaan.

Suhteutetaan näitä luvussa 4 esiteltyihin Hintzen anonymisoinnin tasoihin [17]. Niiden määrittelyjä käyttämällä luodaan argumentti sopivasta pseudonymisoinnin tasosta: *A11-anonymisoitu*.

Tuotantodata sellaisenaan on luonnollisesti tasoa *tunnistettu*. Tunnistus tulee joko *eksplisiittisesti* henkilöä koskevista arvoista kuten ”nimi”, tai *pyytämättä* mistä tahansa

käyttäjän vapaasti täyttämästä tekstiarvosta. Tasosta anonyymimmäksi pääsemiseksi on kaikki tällaiset arvot sensuroitava.

Seuraava taso, *tunnistettavissa oleva*, on ylitettävissä. Tuotantodatassa itsessään ei ole tähän luokkaan suoraan osuvaa dataa. Hintzen määritelmässä taso on itsessään anonyymiä dataa, jolla on kuitenkin *"tunnettu, systemaattinen tapa linkittää se henkilöön"*. Tähän tasoon voitasiin päätyä, jos pseudonymisointiprosessi esimerkiksi muuntaisi nimet tiedetyn algoritmin mukaan pseudonyymeiksi. (Vaikka käyttämällä Rot13-algoritmia kuhunkin tekstiarvoon).

A11-anonymisoitu data oli sellaista, johon ei rekisterinpitäjän tiedossa ole tapaa yhdistää henkilöön, mutta jonka voisi yhdistää lisätietoja saamalla (esimeriksi rekisteröidyltä). Tutkielman väite on, että pelkät datan *viitteet* ovat tasolla *A11-anonymisoitu*. Kuvitellaan skenaario, jossa kaikki ei-viiteavain-data poistetaan tietokannasta, jättäen ainoastaan joukoittain tunnisteita. Pelkkä rivin *id* ei voi sisältää henkilötietoa; se on tietokannan itsensä generoima arvo.

Koska tietokanta sisälsi aiemmin henkilötietoa, vastaa kukin viiteavaimista yhdistettävä graafi jotakin henkilöä. Suurinta osaa näistä ei pysty millään lisätiedolla yhdistämään takaisin tähän henkilöön, mutta reunatapauksissa se on mahdollista. Viiteavaimista ei välity muuta dataa kuin liitosten *lukumäärät*, mutta tämäkin saattaa yksilöidä henkilön, kun tiedetään kunkin relaation käyttötarkoitus.

Esimerkiksi, tietokannassa voisi olla tallennettuna henkilöt ja heidän autonsa. Dataarvojen poiston jälkeen relaatioista näkee yhä, että yhdellä henkilöistä olisi 91 autoa, joka sattuu pätemään tasan yhteen ihmiseen maailmassa. *Tasan yhdellä ihmisellä on 91 autoa* on tässä esimerkissä se lisätieto, jonka kautta koko graafi oli taas yhdistettävissä henkilöön. Viiteavaimet ovat siis tasoa *A11-anonymisoitu*.

Tämän päättelyn pitäessä paikkaansa, joudumme valitsemaan pseudonymisointiprosessin tavoitteen varsin vähistä vaihtoehdoista. Ainoasta anonyymistä vaihtoehdosta, pelkän skeeman kopioinnista, ei ole hyötyä testauksessa. Koskemattoman tuotantodatan käyttö voisi olla perusteltavissa *tarpeellisena*, mutta se ei ole asetuksen hengen mukaista. Harvassa testitapauksessa on hyötyä nähdä juuri todellisuutta vastaava nimi, kun pseudonyymi olisi vaihtoehto. Päädymme siis tavoittelemaan *A11-anonymisoitu*-tasoa.

Vapaiden tekstikenttien sensuroinnin jälkeen, kysymykseksi nousevat muut tietotyypit: esimerkiksi numeraaliset arvot, päivämäärät, tai enumeraatiot. Näiden käsittelyssä on käytettävä harkintakykyä – riippuu tallennetun datan luonteesta, voiko niistä tunnistaa henkilön ilman lisätietoa.

Haastavaa tästä tekee sen, ettei yksittäinen arvo ole välttämättä tunnistettavissa, mutta viiteavaimista johdetusta graafista ja muiden arvojen kanssa se voikin olla. Triviaalina esimerkkinä käyttäjän nimen ensimmäinen kirjain on enumeraatio. Tämä arvo liittyy henkilöön, joten se on henkilötieto. Arvosta ei voi tunnistaa henkilöä ilman lisätietoa, joten se on *A11-anonymisoitua*. Jos tallennamme eri enumeraatioihin kaikki kirjaimet henkilön nimestä, on tämä triviaalisti tunnistettu. Tämä todistaa ainoastaan sen, että pseudonymisointiprosessia on arvioitava koko tuloksen perusteella, ei yksittäisten muutosten.

Tutkielma jättää riskin arvioinnin toteuttajan vastuulle, mutta kyseenalaisissa tapauksissa sensurointia on suositeltava. Rajojen tarkentaminen *A11-anonymisoidun* osalta on mahdollisuus jatkotutkimukselle. Käytännössä suuri osa datasta ei nosta anonyymiyden tasoa *A11-anonymisoitua* ylemmäs. Sovellusten liiketoimintalogiikan kannalta nämä arvot ovat usein tärkeitä, joten niiden säilyttäminen auttaa testauksessa ja ongelmien ratkaisussa.

Tehdään prosessiin liittyen vielä yksi huomio. Viiteavaimien lisäksi taulujen rivien kesken on olemassa muita suhteita, joita olisi hyvä mahdollisuuksien mukaan säilyttää. Tällä tarkoitetaan enemmän datan muotoa ja sen tilastoituvuutta kokonaisuutena, kuin varsinaisia data-arvoja. Kaikkia suhteita ei voida säilyttää, jotta voimme pysyä *A11*-tasolla.

Otetaan vastaesimerkiksi *aakkosjärjestys*. Täydellisessä testiympäristössä järjestelmän aakkosjärjestetyt listaukset olisivat samat kuin tuotannossa. Aakkosjärjestyksen säilyttäminen eri taulujen eri rivien välillä vaatisi koko järjestettävän sarakkeen (nimen) säilyttämistä sellaisenaan. Tämä taas olisi tunnistava tieto.

Toisenlainen suhde taas on säilytettävissä: rivien keskinäiset yhtäläisyydet. Esimerkiksi jos kahdella käyttäjällä on sama sukunimi, on näiden rivien välillä ”näkymätön” viite. Tämä viite voi nousta esiin liiketoimintalogiikassa, vaikkei sitä tietokannassa ole viiteavaimena. Pseudonymisointiprosessi voi varmistaa, että kaksi samaa nimeä pseudonymisoidaan samaksi lopputulokseksi, tämä tosin vaatii oman tarkkuutensa.

Pseudonymisointiprosessin vaatimukset ovat nyt selvillä. Kysymys on enää, *millaseksi* kukin sensuroitava arvo muutetaan lopputulokseen? Tämä on oikeastaan triviaali ongelma, koska mitä tahansa merkistä ’x’ satunnaisiin kirjainyhdistelmiin on hyväksyttävissä. Huomiota on kiinnitettävä ainoastaan siihen, että arvot todella hyppäävät tason *tunnistettavissa oleva* yli tasolle *A11-anonymisoitu*. Ratkaisu tähän on johtaa pseudonyymi aina muunnettavan arvon rivin *tunnisteesta*. Arvosta itsestään johdettu pseudonyymi on riskialtis mahdollisuudelle palautukseen. Tunnisteesta johdetusta pseudonyymistä tätä tietoa ei ole olemasssa. Testiympäristön käyttökokemuksen parantamiseksi voidaan esimerkiksi ottaa pseudonyymit nimitiedot väestörekisterin nimilistasta tunnisteen mukaan.

Käydään varsinainen prosessi läpi seuraavaksi. Sen kulku on kuvattu pseudokoodina alla *(Algoritmit 1 ja 2)*. Prosessi on kustomoitava erikseen eri tietokantoja varten. Kutakin

tietokannan taulua kohden kootaan oma pseudonymisointikomentonsa, jossa määritetään sensuroitavat sarakkeet ja funktio, jolla kukin sensuroidaan.

Funktion (pseudokoodissa '*hashFunction*') tehtävä on palauttaa vähintään sopivaa tietotyyppiä oleva arvo rivin tunnisteen mukaan. Kosmeettisia muutoksia on mahdollista tehdä. Algoritmi 2 pitää jo huolen siitä, että samalle arvolle tulee sama pseudonyymi. Siksi toteutettavien hash-funktioiden olisi suotavaa palauttaa uniikki arvo jokaista uniikkia tunnistetta kohden.

Algorithm 1 Tietokannan pseudonymisointi

```
1: procedure PSEUDONYMIZE(db)
2:     for each table ∈ db do
3:         c = Map(value → pseudonym)
4:         UPDATE table SET(
                       fistName = pseudonym(fistName, id, c, nameHash),
                       ssn = pseudonym(ssn, id, c, ssnHash),
                       x = pseudonym(x, id, c, xTypeHash),
                       ...)
5:     end for          ▷ Kullekin taululle on luotava omanlaiset päivityskäskynsä.
6: end procedure
```

Algorithm 2 Pseudonyymin luonti

```
1: procedure PSEUDONYM(v, id, c, hashFunction)
2:     if c[v] != null then
3:         return c[v]                    ▷ Samanarvoisille sama pseudonyymi.
4:     end if
5:     ps = hashFunction(id)              ▷ Kutsuttava funktio riippuu arvon tyypistä.
6:     c[v] = ps
7:     return ps
8: end procedure
```

Tämän prosessin jälkeen koko tietokannan tulisi olla *A11-anonymisoitu*, jolloin se voidaan siirtää staging-ympäristöön. Validointi siitä, täyttääkö tietty tämän pseudonymisointiprosessin implementaatio tämän vaatimuksen, vaatii tarkastuksen. Tutkielman väite on se, että jokainen toteutus, joka kutsuu algoritmia 2 jokaisen taulun jokaiselle sarakkeelle, toteuttaa tämän.

Toimintasuositus on toteuttaa ensin tiukin mahdollinen tästä prosessista kullekin tietokannalle. Tämän jälkeen testaamisen käyttökokemusta voidaan alkaa parantaa sallimalla

enemmän *ei-vapaa-tekstikenttä* -tietoja läpi ja keksimällä luovempia hash-funktioita. Kenttien jättäminen pois algoritmista 1 tulee tehdä harkiten, mutta suuri osa datasta voinee lopulta olla sellaisenaan.

5.8 Yhteenveto

Olemme nyt käyneet läpi tutkielman ehdotuksen arkkitehtuurista, jonka tarkoitus on kattaa luvun 3 vaatimusmäärittely. Ratkaisut käyttävät hyväkseen luvussa 4 käsiteltyä aiempaa tutkimusta.

Luvussa määritettiin aluksi arkkitehtuuri kokonaisuutena, jonka jälkeen käytiin läpi yksittäiset komponentit tarkemmalla tasolla. Luvun komponenttien esittelyssä keskityttiin arkkitehtuurillisiin ratkaisuihin, yksityiskohdat jäävät työhön liittyvään käytännön osuuteen.

Kaiken kaikkiaan esiteltiin kuusi eri moduulia, joita palvelut voivat käyttää asetuksen vaatimuksien kattamiseksi. Moduulit ovat henkilötietoloki, oikeuksien käytönhallinta, ydinhenkilötietomoduuli, suostumuksen hallinta, staattinen lähdekoodin analyysi ja pseudonymisointiprosessi. Moduuleilla on osin riippuvuuksia keskenään, mutta teoriassa kukin on itsenäinen kokonaisuutensa. Kokonaisuutena, kaikkiin asetuksen vaatimuksiin on jotenkin otettu kantaa ehdotetussa arkkitehtuursssa.

Henkilötietoloki ratkaisee vaatimuksen *(V4: henkilötietojen jäljitettävyys)*. Moduuli koostuu lokinhallintajärjestelmästä ja ratkaisuista lokiuskäytäntöihin koko arkkitehtuurin osalta.

Oikeuksien käytönhallinta on vastaus tarpeeseen hallita rekisteröidyn pyyntöjä käyttää oikeuksiaan. Moduuli tarjoaa komponentit sekä pyyntöjen käyttöliittymälle, että niitä käsittelevällä taustajärjestelmälle.

Ydinhenkilötietomoduuli sisältää tavan järjestellä henkilötiedot arkkitehtuurin sisällä niin, että tietosuoja on hyvä. Moduulin viestinvälitysjärjestelmä taas hoitaa rekisteröidyn oikeuspyyntöjen toteuttamisen hajautetussa arkkitehtuurissa, jossa henkilötietoja varastoidaan monessa eri järjestelmässä.

Suostumuksen hallinta kattaa samannimisen vaatimuksen *(V3)*. Ehdotettu ratkaisu on linjassa aiemman tutkimuksen kanssa. Ulkoistettu suostumuskeskus piilottaa sitä käyttäviltä järjestelmiltä vaikeat osuudet suostumuksen hankinnasta. Lisäksi saavutetaan luottamus siitä, että rekisteröityvä henkilö on kuka hän esittää olevansa.

Staattinen analyysi ja pseudonymisointiprosessi ovat työkaluja, jotka tutkielma ehdottaa käyttöönotettavaksi henkilötietoja käsitteleviin järjestelmiin. Nämä ovat keinoja parantaa laatuvaatimusta *(V1: järjestelmän tietosuoja)*.

Luvussa on siis esitelty ratkaisut ja perustelut niille. Työ on onnistunut, mikäli tietosuoja-asetuksen vaatimusmäärittely vastaa asetusta, tutkielman arkkitehtuuri vastaa vaatimusmäärittelyä, ja kokonaisuus on järkevä ratkaisu. Tietosuoja-asetus on esitelty ja siitä on johdettu vaatimusmäärittely. Vaatimusmäärittelyä vastaava arkkitehtuuri on nyt esitelty. Tutkielman seuraavana tavoitteena on näyttää, kuinka tässä luvussa esitelty ratkaisu on validi ehdotus. On varmistettava, että arkkitehtuuri kattaa vaatimukset ja pystyy vastaamaan eri skenaaroihin. Tämä tehdään seuraavassa luvussa.

Luku 6
Ratkaisun validointi

Verrataan tutkielman ratkaisua vaatimuksiin, tapausyritykseen ja skenaarioihin. Pyritään validoimaan se, että arkkitehtuuri on suunniteltu oikein.

Edellisessä luvussa (luku 5) esiteltiin tutkielman arkkitehtuurin kokonaiskuva ja yksittäiset ratkaisut, joista se koostuu. Valinnat perusteltiin niiden esittelyn yhteydessä, mutta on tärkeää varmistaa, että arkkitehtuuri vastaa korkeammallakin tasolla vaatimuksiin.

Tämä luku koostuu kolmesta eri näkökulmasta, joista arkkitehtuuria tarkastellaan. Arkkitehtuurin on oltava sopiva tutkielman vaatimusmäärittelyn vaatimusten kannalta. Arkkitehtuurin tulee sopia tutkielman tapausyrityksen kontekstiin. Lisäksi ratkaisuja voidaan katsoa ulkopuolisesta näkökulmasta, arvioimalla miten arkkitehtuuri vastaisi eri tietosuojaskenaarioihin.

Tämän luvun eri osiot kattavat kukin yhden näistä näkökulmista. Aloitetaan tärkeimmästä, eli tietosuoja-asetuksen vaatimuksien täyttymisestä. Luvussa 3 tehtin vaatimusmäärittely. Vastaukseksi siinä johdettuihin yhdeksään vaatimukseen kehitettiin tutkielman arkkitehtuuri. Aliluvussa 6.1 peilataan arkkitehtuuria takaisin vaatimusmäärittelyyn.

Luvussa 2 määriteltyyn arkkitehtuurin kontekstiin, eli tutkielman tapausyritykseen, palataan aliluvun 6.2 pohdinnassa. Tavoite on arvioida arkkitehtuuria tapausyrityksen kannalta. Viimeisenä, aliluvussa 6.3, arvioidaan vielä ratkaisujen tietosuojaa realistisen esimerkkiskenaarion avulla.

Luvun jälkeen voimme olla luottavaisempia siitä, että tutkielman ratkaisu on validi. Luvun toinen tavoite on selvittää ehdotetun arkkitehtuurin vahvuuksia ja heikkouksia. Huomioitavaa on myös työn rajat: mihin siinä ei ole otettu kantaa? Tietosuoja-asetuksessa on esimerkiksi myös osia, jotka eivät ole teknisiä, mutta joihin on reagoitava. Teknisestä näkökulmasta ei voi arkkitehtuurin olevan *täysin kattava*, tai että se ratkaisisi kaikki tietosuojan kysymykset. Tärkeää on kuitenkin varmistua siitä, ettei arkkitehtuurissa ole myöskään selviä aukkoja.

6.1 Arkkitehtuuri tietosuoja-asetuksen kannalta

Tietosuoja-asetuksen vaatimuksiin ollaan tutkielman kulussa perehdytty monesti. Käsittelyn vaatimukset johdettiin luvun 3 vaatimusmäärittelyssä, jonka jälkeen niihin on viitattu vain sen tuloksina *(V1–9)*. Vaatimukset ovat kuitenkin pohjimmiltaan vain abstraktioita asetuksen lakitekstistä. Tutkielman ratkaisut taas perustuvat vaatimusmäärittelyn vaatimuksiin. Onkin syytä tarkastella päättelyketju taaksepäin ja tarkastella arkkitehtuuria kriittisesti.

Tässä osiossa käydään ensin läpi mihin vaatimuksiin kukin tutkielman ratkaisu ottaa kantaa. Kokonaiskuvan saamisen jälkeen tarkastellaan yksittäisiä komponentteja ja tutkitaan niiden rajoja.

Palautetaan mieliimme luvussa 3 esitelty kuva *(Kuva 2)*. Siinä olivat tiivistettynä vaatimusmäärittelyn vaatimukset ja tietosuoja-asetuksen artiklat, joihin vaatimukset perustuvat. Verrataan tätä edellä olevaan kuvaan *(Kuva 10)*. Siinä on koottu vaatimukset tutkielmassa ehdotettuine ratkaisuineen. Kuhunkin komponenttiin viittaa nuolilla ne vaatimukset, joihin moduuli erityisesti vaikuttaa. Käydään läpi kuvan kohdat seuraavaksi.

Tutkielman konkreettinen vastaus *(V1: järjestelmän tietosuojaan)* on staattinen analyysi, johon esiteltiin luvussa 5.6 henkilötietoannotaatioiden prosessori. Laaja vaatimus liittyy kaikkiin arkkitehtuurin osiin ja ympäristöjen ylläpitoon.

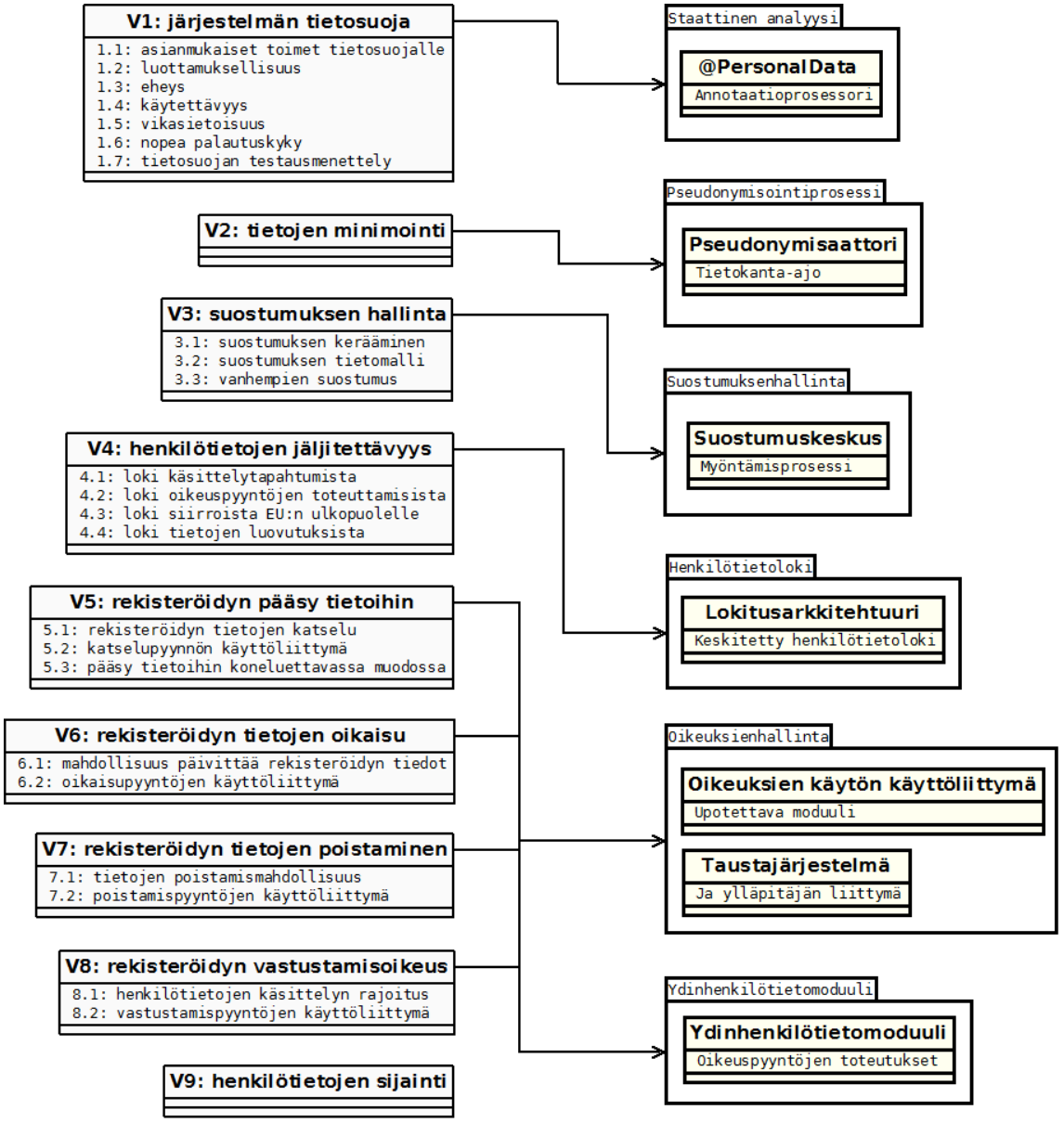
(V2: tietojen minimointi) on toinen laaja vaatimus. Tutkielman ehdotus minimoinnin edistämiseksi on pseudonymisointiprosessi. Tällä *staging*-ympäristön testaus onnistuu silti ilman pseudonymisoinnin menetystä. Minimointiin vaikuttaa jollain tasolla myös muut tutkielman moduulit, erityisesti ydinhenkilötietomoduuli.

(V3: suostumuksen hallinta) on tutkielman arkkitehtuurissa piilotettu täysin suostumuskeskusmoduulin taakse. Luonnollisesti muut moduulit käyttävät sen tarjoamaa palvelua.

Lokinhallintamoduuli taas kantaa vastuun *(V4: henkilötietojen jäljitettävyys)* lokien keskittämisestä. Lokirivien muodostaminen on muiden järjestelmien tehtävä.

Vaatimukset *(V5: rekisteröidyn pääsy tietoihin)*, *(V6: rekisteröidyn tietojen oikaisu)*, *(V7: rekisteröidyn tietojen poistaminen)* ja *(V8: rekisteröidyn vastustamisoikeus)* huomioidaan komponenteilla oikeuksienhallinta ja ydinhenkilötietomoduuli.

Vaatimukseen *(V9: henkilötietojen sijainti)* ei tutkielma tarjoa teknistä vastausta.



KUVA 10: Tutkielman komponenttien tärkeimmät ratkaisukohteet.

Kokonaiskuvan esittelyn jälkeen tarkastellaan seuraavaksi yksittäisiä vaatimuksia tarkemmin.

(V1: järjestelmän tietosuoja) on haastavin vaatimuksista arvioida täyttymisen kannalta. Sen alakohdat ovat kaikki laatuvaatimuksia. Suuri osa näistä on seikkoja, jotka on huomioitava palvelinympäristöjä perustettaessa ja ylläpidettäessä. Tämän tutkielman kaltaiset arvioinnit ovat hyväksi kohdan *(V1.7: tietosuojan testausmenettely)* kannalta. Luonnollisesti palvelukohtaista arviontia vaaditaan myös. Tutkielman ehdotus järjestelmien laadun parantamiseksi on staattisen analyysin työkalu. Vaikka oletettaisiin, että se toimii täydellisesti käyttötarkoitukseensa, on se silti hyvin pieni osa vaatimuksen alasta. Voidaan väittää myös, että ydinhenkilötietomoduuli ja lokitusarkkitehtuuri ovat hyväksi laatuvaatimusten kannalta, mutta kokonaisuutena tutkielman kannanotto vaatimukseen on

melko neutraali. Vaatimuksen osalta lain rikkominen vaatisi kuitenkin erityistä huolimattomuutta järjestelmän toteutuksessa – ei myöskään nähdä, että yksikään arkkitehtuurin komponentti olisi vaatimusta *vastaan*.

(V2: tietojen minimointi) vaikuttaa myös koko arkkitehtuurin laajuisesti. Tutkielman erillinen ehdotus vaatimuksen saavuttamiseksi on pseudonymisointiprosessi. Pseudonymisoinnin käyttö on välttämätöntä, jos testiympäristöön on saatava dataa. Tähän tutkielman ehdottama prosessi *täyttää* minimaalisen vaatimuksen. Testiympäristön data on kuitenkin pieni osa kaiken henkilötiedon minimoinnista. Kun tarkastellaan vain tuotannon arkkitehtuuria, tutkielman ratkaisuista ydinhenkilötietomoduuli edistää eniten vaatimuksen täyttämistä. Moduuliin voidaan keskittää eri taustajärjestelmien henkilötietoja, jolloin niiden levinneisyys koko arkkitehtuurin kesken pienenee. Mitkään tutkielman arkkitehtuurin moduuleista eivät käsittele *ylimääräisiä* henkilötietoja, joten vaatimuksen noudattaminen riippuu varsinaisten liiketoimintajärjestelmien datan minimoinnista. Ydinhenkilötietomoduulin käytön lisäksi ohjeistetaan kehittäjät käsittelemään mahdollisimman vähän dataa järjestelmiä rakentaessa.

(V3: suostumuksen hallinta) on arkkitehtuurissa ratkaisu varsin kattavasti. Suostumuskeskusmalli on sen asiakkaiden (projektiarkkitehtuurien) kannalta helppo tapa kattaa suostumuksen monimutkaiset seikat. Ydinhenkilötietomoduulin tulee siis vain kunnioittaa suostumuskeskuksen päätöksiä. Tämän ratkaisun heikkous on (jo luvussa 5 mainittu) suuri kynnys saada suostumuskeskus yleiseen käyttöön. Yleistymisen jälkeen eri projektien arkkitehtuurit hallitsisivat vaatimuksen yksinkertaisesti. Käytännössä tätä ei voida odottaa tapahtuvan nopeasti. Tutkielmassa esitetty vaihtoehtoratkaisu ei ole kovin pitkälle kehitetty; sama suostumuskeskus, mutta pienemmällä skaalalla, menettää edut keskitetystä rekisteröitymisestä. Suostumuskeskuksen kasvustrategian suunnittelu olisi mahdollisuus myöhemmälle työlle.

(V4: henkilötietojen jäljitettävyys) kattaa kaksi näkökulmaa: *mitä* on lokitettava ja *miten* lokitus hallitaan arkkitehtuurissa. Tutkielman ratkaisu yrittää vastata molempiin. Lokinhallintaluvussa määritettiin lokitapahtumat, sen sisältö ja esiteltiin lokit keskittävä moduuli. Lokitapahtumien määrittely on tietosuoja-asetuksen tulkinnassa konservatiivinen. Asetus ei suoranaisesti velvoita lokittamaan kaikkiea henkilötiedon käsittelytapahtumia. Osoitusvelvollisuus on kuitenkin helpoin toteuttaa näin. Esitelty lokinhallintamoduuli on oletettavasti validi, koska se on yleisessä käytössä teollisuudessa. Tutkielman ratkaisun heikkous voisi olla siinä, että tarkkojen lokituspaikkojen määrittely on jätetty täysin toteuttajien suunniteltavaksi. Mikään ei siis arkkitehtuurissa *takaa* sitä, että käsittely lokitetaan. Lokituksen helpottamiseksi voitaisiin kehittää joitain uusia ratkaisuja, mutta täysin vedenpitäväksi sitä ei voi tehdä. Pidetään vaatimusta kuitenkin ratkaistuna esitetyssä mallissa.

(V5: rekisteröidyn pääsy tietoihin) oli jaoteltu kolmeen osaan. Ne ovat rekisteröidyn tietojen katselun mahdollistaminen, katselupyynnön esittäminen ja pääsy tietoihin koneluettavassa muodossa. Koneluettavuus on triviaalia "tavallisen" tietojen katselun toteuttamisen jälkeen. Tutkielman arkkitehtuurissa kaikkien oikeuksien käyttöpyyntöjen esittäminen yhdistettiin samaksi palveluksi. Pyynnön toteuttamisen hoitaa ydinhenkilötietomoduuliin määritelty viestinvälitysarkkitehtuuri. Mikäli ratkaisu olisi vain tälle vaatimukselle, olisi se luultavasti ylimitoitettu. Huomioidaan kuitenkin synergia muiden rekisteröidyille tarjottavien palveluiden kanssa, mitkä vaativat enemmän arkkitehtuurilta.

(V6: rekisteröidyn tietojen oikaisu) ja ***(V7: rekisteröidyn tietojen poistaminen)*** ratkaistiin samalla tavalla kuin *(V5)*. Haastavinta näissä on jälleen varmistaa, että taustajärjestelmät, joille oikeuksien käyttöpyynnöt välitetään todellakin vastaavat niihin oikein. Realistisesti tätä ei voidakaan tietää varmaksi tuntematta järjestelmiä. Tutkielmassa esitelty ratkaisu tähän täyttää vaatimukset. Vaatimusmäärittelyssä mainittiin lisäksi *virheenkäsittelyn* huomiointi oikeuksien toteuttamisessa. Tähän ei luvun 5 ratkaisuissa otettu tarkemmin kantaa. Virheistä toipumisen ratkaisut olisivat yksi mahdollisuus jatkaa tutkielman työtä.

(V8: rekisteröidyn vastustamisoikeus) on myöskin ratkaistu arkkitehtuurissa oikeuksienhallinta- ja ydinhenkilötietomoduulien avulla. Vastustamisoikeuden käyttöä ei käsitelty kovin pitkällisesti määrittelyssä, mutta ratkaisu onkin yksinkertainen. Riittääkö se? Esitetyssä mallissa vastustamisen käyttö tallennetaan vain ydinhenkilötietomoduuliin tiedoksi. Tässä luotetaan siihen, että muut järjestelmät tarkastavat tilanteen ennen henkilötietojen käsittelyä. Palvelussa, jossa on tarve käsitellä suuria massoja tietoja, voisi tämä tulla pullonkaulaksi. Vastustamisen voimaantulosta pitäisi tällaisessa tilanteessa lähettää myös viesti ydinhenkilötietomoduulilta kaikille muille järjestelmille. Laajennus olisi kuitenkin pieni, kun viestinvälitys on jo suunniteltu.

(V9: henkilötietojen sijainti) sivuutettiin tutkielmassa sillä, ettei henkilötietoja käsitellä EU:n ulkopuolella. Keskimääräiselle tapausyrityksen projektin arkkitehtuurille tämä on paras vaihtoehto; *"ei mitään"* on aina elegantein ratkaisu verrattuna johonkin rakennelmaan. Vaatimus on kuitenkin tärkeää pitää tiedossa, kun uusia järjestelmiä suunnitellaan.

6.2 Arkkitehtuuri tapausyrityksen kannalta

Tarkastelimme tutkielman ratkaisua vaatimuksiin nähden. Tästä löydettiin kehitettäviä kohtia, mutta kokonaisuutena ei suuria ongelmia havaittu. Tehdään seuraavaksi katsaus esiteltyyn arkkitehtuuriin luvun 2 tapausyrityksen näkökulmasta. Tavoitteena on jatkaa kriittistä tarkastelua, löytäen arkkitehtuurin heikkouksia ja vahvuuksia. Edellinen luku vertasi ratkaisua vaatimuksiin, nyt arvioidaan sen *sopivuutta* tapausyritykselle. Tarkastellaan ensin kaikkia arkkitehtuurin moduuleja ja lopuksi kokonaisuutta.

Staattisen analyysin ratkaisun `@PersonalData`-annotaatioprosessori on vain kehitystä helpottava työkalu, jonka ei pitäisi tulla esteeksi tapausyrityksen projekteissa. Moduulin kannalta tärkeimmiksi tapasyrityksen ominaisuuksiksi erottuvat *(O2: teknologiasidonnaisuus)* ja *(O7: uudelleenkäytettävyys)*. Tutkielman toteutuksessa annotaatioprosessori on tehty Java-kielellä. Se on siis rajoittunut tiettyä teknologiaa käyttäviin järjestelmiin. Havaitaan siis, että staattisen analyysin uudelleenkäytettävyyden arvo on riippuvainen teknologiasidonnaisuudesta. Toisaalta, kerran toteutettu (avoimen lähdekoodin) työkalu on kaikkien samaa teknologiaa käyttävien saatavilla. Kysymys siitä, monenko teknologian staattisen analyysin työkaluihin sijoitetaan on liiketoiminnallinen päätös – tähän liittyy *(O4: projektien ulkopuoliset investoinnit)*.

Tutkielmassa esiteltyyn pseudonymisointiprosessiin liittyvät samat kysymykset. Teknologiasidonnaisuus tässä ei ole niin voimakas, koska algoritmi on sama ainakin kaikkien relaatiotietokantojen välillä. *(O5: staging-ympäristön vaatimukset)* on tutkielman ehdotuksessa ratkaistu, tietosuoja-asetusta kunnioittaen.

Suostumuksenhallinta voidaan nähdä tapausyrityksen kannalta kahdesta näkökulmasta: tuottajana tai kuluttajana. Mikäli jokin toinen taho perustaisi suostumuskeskusta vastaavan palvelun, olisi sen käyttö helppo vastaus vaatimukseen *(V3: suostumuksen hallinta)*. Toisaalta, mikäli suostumuskeskus nähtäisiin mahdollisuutena perustaa oma palvelu, voisi siitä saada uuden liiketoimintalinjan. Toimintatavasta huolimatta, malli on edullinen tapausyrityksen ominaisuuksien kannalta tarkasteltuna. Tutkielman ehdotuksen selvin heikkous on sen kynnys päästä alkuun, kuten aiemmin mainittu. Tietosuoja-asetus tulee voimaan nopeasti, joten jokin ratkaisu on oltava valmiina ennen sitä.

Lokitusarkkitehtuuri on neutraali tapausyrityksen kannalta. Monessa projektissa on jo jonkinlainen lokienhallinta valmiiksi olemassa. Tällöin henkilötietolokin lisääminen olisi vain vanhan laajentamista. Tutkielmassa ehdotettu teknologiapino ei ole mitenkään pakollinen vaatimusten kannalta. Mikäli projektilla tai yrityksellä on jo käytössä jokin muu lokinhallinta, sitä voidaan käyttää ihan yhtä hyvin.

Oikeuksienhallinnan ratkaisu on tärkeä tapausyrityksen ominaisuuksien *(O8: tukitoiminta ja palvelimien ylläpito)* ja *(O7: uudelleenkäytettävyys)* kannalta. Oikeuspyyntöjen käsitelly pitäisi saada keskitettyä ainakin yrityksen tasolla. Ei ole tehokasta, että ylläpitäjät

joutuvat seuraamaan erikseen jokaisen projektin oikeuspyyntöjen saapumista. Tässä on selvästi kohta parantaa vielä tutkielman arkkitehtuuria. Toisaalta, ei ole mahdollista tallentaa *eri* henkilötietorekisterien oikeuspyyntöjä yhteen järjestelmään. Realistinen mahdollisuus vähentää ylläpitäjien työtä olisi lähettää hälytyksiä saapuneista oikeuspyynnöistä heille. Tällöin varsinainen oikeuspyynnön sisältö ei lähtisi pois rekisterin piiristä ja ylläpitäjien ei tarvitsisi seurata yksittäisiä projekteja niin tarkasti. Vasta käytännössä tullaan näkemään, kuinka paljon oikeuksia tullaan käyttämään, joten niiden käsittely voi osottautua työlääksikin.

Ominaisuuden *(O6: vanhojen järjestelmien päivittäminen)* suhteen, erillisten moduulien käyttöönotto on helpompaa kuin suuret muutokset vanhoihin järjestelmiin itseensä. Suurin haaste tässä tulee olemaan ydinhenkilötietojen siirto omaan moduuliinsa. Sen kanssa voidaan tehdä jotain kompromisseja vanhojen järjestelmien osalta.

Arkkitehtuuri itsessään ei ota vahvasti kantaa varsinaisten palvelimien pystyttämiseen ja ylläpitoon. Ominaisuuden *(O8: tukitoiminta ja palvelimien ylläpito)* mukaan tämä on kuitenkin tapausyrityksen vastuulla. Oletus on, että ympäristöissä käytetään yleisesti hyväksi havaittuja käytäntöjä. Henkilötietoja sisältävät levyt olisi tärkeää olla kryptattuja, esimerkiksi. Uusia, erityisesti tietosuojaan liittyviä ratkaisuja, palvelinympäristöille voitaisiin kehittää.

Tapausyrityksen ominaisuus *(O3: projektin koon vaihtelu)* herättää mielenkiinnon siitä, että kuinka suuriin eri projekteihin ratkaisut sitten sopivat. Samalla voidaan pohtia sitä, miten ratkaisut yleistyisivät tapausyritystä pienemmille tai suuremmille organisaatioille.

Ydinhenkilötietomoduuli voi olla ylimitoitettu pienimpiin henkilötietorekistereihin. Paljon suurempiin järjestelmiin taas tutkielmassa esitettyä moduulia pitää varmasti soveltaa, mutta ydinhenkilötietojen keskittämisen malli itsessään on pätevä.

Tutkielmassa kuvatussa parhaassa tapauksessa suostumuskeskus on sopiva kaikenkokoisille henkilötietorekistereille. Suurimmat toimittajat voisivat haluta tehdä oman suostumuskeskuksensa vain heidän järjestelmiensä käyttöön.

Oikeuksien käytönhallinta on jossain määrin toteutettava asetuksen vaatimuksien vuoksi. Todella pienissä yrityksissä tätä voisi ulkoistaa käyttöliittymän osalta, mutta koska pyynnöt sisältävät henkilötietoja, on ne varastoitava rekisterin piirissä. PK-yritykset voisivat lisätä rekisteriselosteeseensa ilmoituksen siitä, että pyynnöt käsitellään kolmannen tahon puolesta, mutta sitä ei kannata välittömästi suositella ratkaisuksi.

Lokinhallinta voi skaalata sitä käyttävien yritysten koon mukaan. Lokin sisältö pysyy samana, mutta pienimmissä järjestelmissä riittänee *henkilötietolokitiedosto* koko lokitusarkkitehtuurin sijasta. Suurimmat toimijat taas saattavat haluta keskittää lokit useaan eri keskukseen.

`@PersonalData`-annotaatioprosessori toimii ja on hyödyksi kaikissa järjestelmissä, joihin sitä voi käyttää. Pienien projektien kompleksisuus voi olla pientä ja elinkaari lyhyempi. Näissä saavutettavat hyödyt voivat olla vähäisempiä, mutta prosessorin käyttökään ei ole vaivalloista.

Tutkielman pseudonymisointiprosessia voidaan käyttää kaikissa järjestelmissä, joissa tietokannat ovat kerralla manipuloitavissa. Pienet toimijat voivat siis käyttää samaa prosessia, mikäli he tarvitsevat vastaavaa testiympäristöä. Suurissa järjestelmissä koko tietokantaa kerralla käsittelevä prosessi pitää mukauttaa eri muotoon.

6.3 Esimerkkiskenaario

Nyt olemme arvioineet tutkielman ratkaisua tietosuoja-asetuksen vaatimuksien ja tapausyrityksen kannalta. Täydennetään seuraavaksi tätä tarkastelemalla sitä, miten arkkitehtuuri vastaisi ulkopuoliseen tietosuojaskenaarioon.

Karbaliotis julkaisi maaliskuussa 2017 hypoteettisen "painajaiskirjeen" [5], jonka hän esittää työkaluksi organisaation tietosuojan valmiuden arviontiin. Kirjeessä rekisteröity esittää oikeuksiensa käyttöpyynnön, joka esittää organisaatiolle pahinta uhkakuvaa tietosuoja-asetuksen puitteissa. Kirje on suunnattu organisatorisiin kysymyksiin, mutta on mielenkiintoista nähdä saadaanko siitä huomioita arkkitehtuurille.

Käydään seuraavaksi kirje läpi kohdittain (vapaasti suomennettuna). Arvioidaan kustakin vaaditusta asiasta, miten se huomioitaisiin tutkielman arkkitehtuurissa. Joitain kirjeessä esitettyjä vaatimuksia voidaan kiistää suurempina, kuin mihin tietosuoja-asetus rekisteröityjä oikeuttaa. Voidaan kuitenkin kuvitella vastaavia kysymyksiä tulevan myös tietosuojaviranomaisten toimesta.

> Kirjoitan teille roolissanne yrityksenne tietosuojavastaavana. Olen asiakkaanne ja – viimeaikaisten tapahtumien valossa – teen teille tämän pyynnön pääsystä henkilötietoihini, yleisen tietosuoja-asetuksen 15. artiklan mukaisesti.

Tutkielman arkkitehtuurilla rekisteröity olisi välttynyt kirjeen lähettämiseltä ja vastaanottaja paperin käsittelyltä. Pyyntö vastaa käyttäjätarinaa *KT4* ja vaatimusta *V5*.

> Tarpeelliset dokumentit henkilöllisyyteni varmistamiseksi ovat liitteenä. Mikäli tarvitsette lisätietoja, ottakaa yhteyttä osoitteeseeni yllä.

Suostumuskeskuksen kautta järjestelmällä on jo luottamus pyynnön esittäjän henkilöllisyyteen.

> Haluaisin tehdä tiedoksi, että odotan vastausta pyyntööni kuukauden sisällä, kuten 12. artikla määrää. Jos niin ei tapahdu, lähetän valituksen <sopivalle tietosuojaviranomaiselle>.

Arkkitehtuuri helpottaa tätä, koska henkilötiedot ovat automatisoidusti saatavilla ja pyynnöt nopeasti käsiteltävissä.

> Pyydän ilmoittamaan seuraavat tiedot:
>
> 1. Varmistuksen siitä, käsitelläänkö minun henkilötietojani. Jos käsitellään, kertokaa henkilötietokategoriat, joihin teillä olevat minua koskevat tiedot kuuluvat.
>
> a. Erityisesti, kertokaa mitä te tiedätte minusta järjestelmissänne. Olkoon ne tietokannoissa, sähköposteissa, tiedostoissa, tai muussa mediassa jota varastoitte.
>
> b. Lisäksi, ilmoittakaa missä valtioissa henkilötietojani säilytetään, tai mistä niihin voidaan päästä käsiksi. Mikäli käytätte pilvipalveluita tietojeni käsittelyyn tai varastoimiseen, sisällyttäkää maat, joissa palvelimet sijaitsevat (tai ovat sijainneet viimeisen 12 kuukauden aikana).
>
> c. Antakaa minulle kopio käsiteltävistä henkilötiedoistani, tai pääsy niihin.

Henkilötietojen lataaminen on automatisoitu tutkielman arkkitehtuurissa. Käsiteltävien tietojan kategoriat ovat rekisteriselosteessa kirjattavia asioita. Niihin ei ole otettu kantaa.

> 2. Antakaa yksityiskohtaisen selostus tietyistä tarkoituksista, joihin olette käyttäneet (tai käytätte) henkilötietojani.

Tämäkin on rekisteriselosteen asia ja projektikohtaista: ei voida antaa yleistä vastausta.

> 3. Antakaa lista kaikista kolmansista osapuolista, joille jakaneet henkilötietojani.
>
> a. Jos ette kykene tähän varmuudella, antakaa lista kaikista kolmansista osapuolista, joille olette saattaneet jakaa henkilötietojani.
>
> b. Kertokaa myös, missä kohdassa 1(b) luetelluista valtioista nämä kolmannet osapuolet varastoivat henkilötietojani, tai joista on pääsy niihin. Valaiskaa myös lailliset perusteet tietojeni siirtoon näihin valtioihin. Jos olette niitä käyttäneet, toimittakaa myös perusteet tähän sopivista varotoimenpiteistä.
>
> c. Lisäksi, haluaisin tietää mitä varotoimenpiteitä on käytetty mainittujen kolmansien osapuolien kanssa, henkilötietojeni heille siirtämisen suhteen.

Tutkielman arkkitehtuuri rajasi *(V9: henkilötietojen sijainti)* myötä tietojen käsittelyn EU:n sisälle. Tästä ei tule siis ongelmaa. Arkkitehtuurissa oletetaan kolmansien osapuolten integraatioita sisältävien taustajärjestelmien tallentavan kohteet.

> 4. Ilmoittakaa kauanko varastoitte henkilötietojani. Jos tämä aika perustuu henkilötietojen kategorioihin, ilmoittakaa kauanko kutakin kategoriaa säilytetään.

Tämä on taas palvelukohtainen seikka, joka on päätetty rekisteriselostetta luodessa.

> 5. Jos keräätte tietoja minusta muista lähteistä kuin itseltäni, toimittakaa kaikki lähdettä koskevat tiedot kuten 14. artikla määrittää.

Tutkielman arkkitehtuurissa ei ole huomioitu henkilötiedon *lähdettä*. Tämä on selkeä puute, minkä voi korjata lisäämällä kyseisen arvon ydinhenkilötietoon. Ilman tätä, henkilötietolokin tunnollinen täyttäminen riittäisi, mutta se ei ole yhtä luotettavaa.

> 6. Mikäli teette minua koskevia automatisoituja päätöksiä, mukaanlukien profilointia, 22. artiklan mukaisesti tai ei, toimittakaa minulle tälläisten päätösten logiikka, merkityksellisyys ja seuraukset.

Suostumusta kerätessä (suostumuskeskuksessa) ollaan eritelty eri käsittelyn muodot ja tarjotaan vaihtoehdot kieltäytyä osasta, mikäli mahdollista. Suostumuksen tiettyyn käsittelyyn voi myös perua sieltä.

> 7. Haluaisin tietää onko yrityksenne vahingossa luovuttanut henkilötietojani tai joutunut tietomurron kohteeksi menneisyydessä.
>
> a. Mikäli niin, ilmoittakaa seuraavat tiedot kaikista loukkauksista: i) yleinen kuvaus, ii) ajankohta, iii) ajankohta, jolloin tiedostitte loukkauksen, iv) loukkauksen aiheuttaja, v) luovutetut tiedot, vi) riskiarvio, vii) tehdyt vastatoimet, viii) yhteystiedot lisätietojen saamiseksi, ja ix) tietoa miten voin suojata itseäni haitalta.
>
> b. Jos ette voi tietää varmasti onko loukkausta tapahtunut, ilmoittakaa mitä riskejä pienentäviä toimia olette tehneet, kuten kryptausta, datan minimointia, anonymisointia, tai muita keinoja.

Tutkielman arkkitehtuuri ei erityisesti ratkaise tietomurtojen seurantaa. Nämä ovat toivottavasti niin harvinaisia, että erillistä järjestelmää ei tarvita. Mekanismi tiedottaa kaikkia rekisteröityjä voisi olla hyvä määritellä jatkossa.

> 8. Haluaisin tietää, mitä tietosuojaprosesseja ja käytäntöjä seuraatte henkilötietojeni suojaamiseksi. Noudatatteko esimerkiksi jotain tietosuojasertifikaattia? Lisäksi, käytäntönne liittyen seuraaviin kohtiin:
>
> a. Miten varmuuskopioitte tietoni, missä varmuuskopiot ovat varastoitu ja miten ne ovat suojattu? Mitä olette tehneet suojataksenne henkilötietojani häviöltä tai varkaudelta ja käytättekö kryptausta?
>
> b. Käytättekö teknologiaa, joka ilmottaisi kohtuullisen varmasti, jos tietomurto olisi tapahtunut? Esimerkiksi varashälyttimiä, palomuureja, pääsynhallintaa, auditointia, tai analyysityökaluja.

Tähän voisi vastata ainakin osin esittelemällä arkkitehtuuin tietosuojaratkaisuja. Palvelimien tietoturvaan ei olla tutkielmassa syvennytty, mutta sekin olisi relevanttia. Kuten kirjeen kyselyt vihjaavat, myös konesalien fyysinen turvallisuus on osa tietosuojaa. Tietomurtojen torjuminen ja havaitseminen on yksi jatkokehitysmahdollisuus.

> 9. Työntekijöihin ja alihankkijoihin liittyen, vastatkaa seuraaviin kysymyksiin:
>
> a. Miten valvotte, ettei työntekijänne tahallaan tai vahingossa luovuta henkilötietojani?
>
> b. Onko teillä ollut tapauksia, jossa työntekijä on irtisanottu tietosuojaloukkauksen vuoksi?
>
> c. Mitä toimenpiteitä olette tehneet varmistaaksenne, että työntekijänne käsittelevät tietojani yleisen tietosuoja-asetuksen mukaisesti?

Tutkielman arkkitehtuurin kontribuutiot tähän ovat henkilötietoloki ja oikeuksien käytönhallinta. Käsittelytapahtumat lokitetaan ja työntekijät ovat siitä tietoisia, mikä ehkäisee väärinkäytöksiä. Oikeuksien käytönhallinnan ylläpitäjien käyttöliittymä taas helpottaa työtä siten, että sen kautta toiminnot ovat rajoitettuja oikeaan suuntaan.

6.4 Yhteenveto

Tässä luvussa etsittiin tutkielmassa esitetyn ratkaisun heikkouksia ja vahvuuksia. Tällä pyrittiin löytämään kehitettäviä kohteita ja mahdollisuuksia jatkolle. Ensimmäisessä osassa arvioitiin arkkitehtuuria tietosuoja-asetuksen vaatimusten kannalta, toisessa osassa tapausyrityksen kannalta, ja kolmannessa esimerkkiskenaarion kautta.

Kokonaisuutena, tutkielman arkkitehtuuri havaittiin toimivaksi, vaikka siitä löydettiinkin vielä kehitettävää. Suuria esteitä ei löydetty, joten arkkitehtuurin voi väittää ratkaisevan tietosuoja-asetuksen vaatimukset tapausyritykselle sopivalla tavalla. Esitetyt ratkaisut vaikuttavat vahvoilta, varsinkin jos kehityskohteet toteutetaan.

Esimerkkiskenaarioissa arkkitehtuuria käyttävä organisaatio olisi paremmassa asemassa vastaamaan skenaarion kysymyksiin kuin varautumaton organisaatio.

Edellä olevassa taulukossa *(Taulukko 5)* on listattuna tutkielman arkkitehtuuriin jääneet (tiedossa olevat) kehityskohteet. Niitä voidaan käyttää pohjana, kun suunnitellaan miten arkkitehtuuri kannattaa ottaa käyttöön tai mitä aiheesta kannattaisi tutkia seuraavaksi. Selviä kehityskohteita löydettiin yhteensä kymmenen. Tietysti kaikkia muitakin osia työssä on mahdollista myös syventää.

Taulukko 5: Tutkielman arkkitehtuurista havaitut kehitysmahdollisuudet.

	Kehityskohde
1	Suostumuskeskuksen käyttöönotto
2	`@PersonalData` ja pseudonymisointi eri teknologioille
3	Virheistä toipumisen määrittely
4	Taustajärjestelmien tiedotus ”rajoituksen” alkamisesta
5	Teknologia helpommalle lokitukselle
6	Henkilötiedon lähde ydinhenkilötietoon
7	Pseudonymisoinnin tasojen rajojen tutkiminen
8	Oikeuksienhallinnan ylläpito monen projektin välillä
9	Ydinhenkilötietomoduulin ja oikeuksienhallinnan integraatio
10	Tietomurtojen ehkäisy ja havainnointi

Tutkielman työssä on ollut kyse arkkitehtuurista. Luonnollisesti tämä ei ota kantaa vielä toteutukseen. Työn ohessa tehty käytännön toteutus on osoittanut ainakin osan esitetyistä ratkaisuista olevan realistisia. Työn käyttöönotolle ei näy esteitä, mikä on tärkeää ottaen huomioon tietosuoja-asetuksen voimaantulon ajankohdan.

Luku 7
Johtopäätökset

Yhteenveto työstä ja saavutetut johtopäätökset. Arvoidaan suoriutumista ja tietosuoja-asetusta.

Tutkielmassa lähdettiin selvittämään yleistä tietosuoja-asetusta puhtaalta pöydältä, valitun ohjelmistoyrityksen näkökulmasta. Valitun kontekstin esittelyn jälkeen, käytiin lakiteksti läpi johtaen siitä teknisen vaatimusmäärittelyn. Näihin vaatimuksiin pyrittiin vastaamaan eri moduuleista koostuvalla arkkitehtuurilla. Kriittisen arvioinnin perusteella arkkitehtuuri vaikuttaa kattavan vaatimukset.

Luvussa 3 käsiteltiin tietosuoja-asetuksen lakiteksti. Tulkitsemalla asetusta systemaattisesti, lopulta johdettiin vaatimusmäärittely sen teknisistä osista. Havaittiin yhdeksän ylimmän tason vaatimusta: *V1) järjestelmän tietosuoja*, *V2) tietojen minimointi*, *V3) suostumuksen hallinta*, *V4) henkilötietojen jäljitettävyys*, *V5) rekisteröidyn päästy tietoihin*, *V6) rekisteröidyn tietojen oikaisu*, *V7) rekisteröidyn tietojen poistaminen*, *V8) rekisteröidyn vastustamisoikeus*, ja *V9) henkilötietojen sijainti*. Vaatimusmäärittely onnistui, eikä jälkikäteen havaittuja puutteita ole ilmennyt.

Vaatimuksien keräämisen jälkeen, työssä suunniteltiin ohjelmistoarkkitehtuuri vastaamaan niihin. Arkkitehtuuri perustuu luvun 3 tapausyrityksen toimintaympäristöön. Tutkielmassa esitelty arkkitehtuuri koostuu moduuleista, joilla pyritään piilottamaan asetuksen vaatimukset liiketoimintajärjestelmiltä.

Tutkielma esitteli kuusi moduulia eri vaatimusten ratkaisemiseksi. Ne ovat ydinhenkilötietomoduuli, suostumuskeskus, lokitusarkkitehtuuri, oikeuksien hallinta, pseudonymisointiprosessi ja `@PersonalData`-annotaatioprosessori.

Ydinhenkilötietomoduuli havaittiin tärkeäksi osaksi tietosuoja-arkkitehtuuria. Se pitää sisällään välittömästi henkilöön liittyvät tiedot ja tarjoaa niitä muille järjestelmille. Lisäksi moduulissa on tieto siitä, miten asetuksen oikeuksien käyttöpyynnöt toteutetaan.

Suostumuskeskus on malli, jolla ulkoistetaan rekisteröidyn suostumuksen hallinta arkkitehtuurista. Johtopäätöksenä tämä vaikuttaisi vahvalta. Sen suurimmaksi haasteeksi arvioitiin se, että hyödyt saavutetaan vasta laajan käyttöönoton jälkeen.

Lokitusvaatimuksien ratkaisuksi tutkielmassa ehdotettiin keskitettyä lokinhallintaa ja lokinvälitysarkkitehtuuria. Työssä määriteltiin myös tarpeelliset lokitapahtumat sisältöineen. Arvioinnin mukaan nämä kattavat asetuksen vaatimukset.

Tutkielmassa havaittiin tarve jonkinlaiselle palvelulle tehdä ja käsitellä eri rekisteröityjen oikeuksien käyttöpyyntöjä. Tähän vastaukseksi esiteltiin oikeuksien käytönhallinta -moduuli, joka kattaa käyttöliittymän ja taustajärjestelmän. Tästä arvioidaan saavan uudelleenkäytettävä komponentti kaikkiin projekteihin.

Varsinaisten arkkitehtuuriin sisällytettävien moduulien lisäksi, tutkielma esitteli kaksi työkalua tukemaan ohjelmistokehitystä. Pseudonymisointiprosessi tuli tapausyritykselle tarpeeseen testiympäristöön. Pseudonymisoimalla henkilötiedot niiden rakenne on sama, mutta sisältö noudattaa paremmin henkilötietojen minimointivaatimusta. Tutkielmassa arvioitiin eri anonyymiyden tasoja ja tultiin siihen tulokseen, että pseudonymisoinnin realistisin tavoite on tietosuoja-asetuksen artiklan 11 mukainen taso.

Toinen määritelty työkalu on kehityksen apuvälineenä käytettävä staattinen analyysi. `@PersonalData`-annotaatioita valvomalla saadaan luottamusta siitä, etteivät henkilötiedot joudu huolimattomuuden vuoksi vaaraan. Tämä on eduksi arvioidessa asetuksen laatuvaatimuksia tietosuojalle.

Tutkielman arkkitehtuuri suunniteltiin tapausyritykselle, mutta on myös hyödyllistä arvioida, että voidaanko tuloksia yleistää. Luvussa 6 arvioitiin kunkin moduulin yleistettävyyttä. Koko arkkitehtuuri ei ole sellaisenaan valmis huomattavasti pienempiin tai suurempiin järjestelmiin; moduulien todettiin yleistyvän melko hyvin, mutta ei ilman muutoksia. Vaatimusmäärittely pitää paikkansa kaiken kokoisille järjestelmille.

Näin tutkielman työn aikana kertyneen näkemyksen myötä on myös syytä arvioida yleistä tietosuoja-asetusta itseään. Kuten luvun 1 johdannossa todettiin, jonkinlaiselle lainsäädännölle tietosuojan osalta on ollut kysyntää. Vaatimusmäärittelyssä nähtiin, että asetus antaa rekisteröidyille merkittäviä oikeuksia. Jää nähtäväksi, lisääkö tämä yleisön luottamusta verkon käytössä.

Tietosuoja-asetus toi joukon vaatimuksia, mutta ratkaisuja se ei esitellyt. Tähän rinnalle oltaisiin kaivattu käytänteitä ja malliesimerkkejä. Niiden puute oli yksi syy, joka johti tarpeeseen tälle tutkielmalle. Tietosuoja-asetuksen tavoite on yhtenäistää tietosuojaa EU:n välillä. Lainsäädännön yhtenäistäminen on hyvä alku, mutta käytännön toteutuksille on tärkeää olla tasavertaiset perusteet. Asetuksen vakavasti ottava yritys ei saisi jäädä muiden jalkoihin. Lakitekstissä on kuitenkin tulkinnanvaraa, kuten tutkielmassakin ollaan paikoin todettu.

Entä voidaanko tutkielman jälkeen todeta tietosuoja-arkkitehtuurin olevan käsitelty kysymys? Ei voida. Työssä rakennettiin omaan kontekstiinsa sopiva ratkaisu tietosuoja-asetuksen vaatimuksille. Erilaisia arkkitehtuureja voitaisiin suunnitella ja vertailla. Eikä voida sanoa, että edes tutkielmassa valitulla linjalla oltaisiin tien päässä. Luvussa 6 eriteltiin tarkemmin havaittuja jatkokehitysmahdollisuuksia. Tutkielma kuitenkin saavutti tavoitteensa.

7.1 Kiitokset

Kiitän tutkielman ohjaamisesta Ville Leppästä ja Jukka Ruohosta, joiden neuvot olivat tärkeitä työn valmiiksi saamiseksi. Työn mahdollistivat ja tukea antoivat Geniem, Solita ja KS. Keskustelut Marko Lenkkerin kanssa herättivät kiinnostuksen aiheeseen. Tuuli Kankaala kannusti kirjoittamaan tutkielman.

Viittaukset

[1] Jef Ausloos. The Interaction between the Rights to Object and to Erasure in the GDPR, https://www.law.kuleuven.be/citip/blog/gdpr-update-the-interaction-between-the-right-to-object-and-the-right-to-erasure, Elokuu 2016.

[2] Automattic. Wordpress, https://wordpress.org/, Huhtikuu 2017.

[3] Len Bass, Ingo Weber, and Liming Zhu. *DevOps: A Software Architect's Perspective.* Addison-Wesley Professional, 2015.

[4] F. P. J. Brooks. No Silver Bullet – Essence and Accidents of Software Engineering. *Computer*, 20(4):10–19, 1987.

[5] C. Karbaliotis. The Nightmare Letter: A Subject Access Request under GDPR, https://www.linkedin.com/pulse/nightmare-letter-subject-access-request-under-gdpr-karbaliotis, Maaliskuu 2017.

[6] P. Clements and L. Northrop. *Software Product Lines: Practices and Patterns.* SEI series in software engineering. Addison-Wesley, 2002.

[7] M. Cohn. *User Stories Applied: For Agile Software Development.* Pearson Education, 2004.

[8] Mark Dixon. An objective measure of code quality. *Technical report, Energy Group, Beverly, Massachusetts*, 2008.

[9] P.M. Duvall, S. Matyas, and A. Glover. *Continuous Integration: Improving Software Quality and Reducing Risk.* Addison-Wesley Signature Series. Pearson Education, 2007.

[10] R. Folsom, R.B. Lake, and V.P. Nanda. *European Union Law After Maastricht:Practical Guide for Lawyers Outside the Common Market.* Springer Netherlands, 1996.

[11] Working Group for Life Cycle Processes. ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes –Requirements engineering. *ISO/IEC/IEEE 29148:2011(E)*, pages 1–94, Joulukuu 2011.

[12] The Apache Software Foundation. Apache ActiveMQ -integraatiopalvelin, http://activemq.apache.org/, Lokakuu 2017.

[13] Harald Gjermundrød, Ioanna Dionysiou, and Kyriakos Costa. privacyTracker: A Privacy-by-Design GDPR-Compliant Framework with Verifiable Data Traceability Controls. In *Current Trends in Web Engineering: ICWE 2016 International Workshops*, pages 3–15. Springer International Publishing, 2016.

[14] Bryce Goodman and Seth Flaxman. Eu regulations on algorithmic decision-making and a "right to explanation". In *ICML workshop on human interpretability in machine learning (WHI 2016), New York, NY. http://arxiv. org/abs/1606.08813 v1*, 2016.

[15] J. Gothelf and J. Seiden. *Lean UX: Applying Lean Principles to Improve User Experience*. Lean series. O'Reilly Media, Incorporated, 2013.

[16] Dick Hardt. The OAuth 2.0 authorization framework. *Internet Engineering Task Force (IETF)*, 2012.

[17] Mike Hintze. Viewing the GDPR through a De-Identification Lens: A Tool for Clarification and Compliance. *SSRN*, 2016.

[18] G. Hohpe and B. Woolf. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley Signature Series (Fowler). Pearson Education, 2012.

[19] Johannes Holvitie and Ville Leppänen. DebtFlag: Technical debt management with a development environment integrated tool. In *Proceedings of the 4th International Workshop on Managing Technical Debt*, pages 20–27. IEEE Press, 2013.

[20] V Hordern. The Final GDPR Text and What It Will Mean for Health Data. *Hldataprotection. com*, 20, 2016.

[21] Yao-Wen Huang, Fang Yu, Christian Hang, Chung-Hung Tsai, Der-Tsai Lee, and Sy-Yen Kuo. Securing web application code by static analysis and runtime protection. In *Proceedings of the 13th international conference on World Wide Web*, pages 40–52. ACM, 2004.

[22] Elastic Inc. Elastic stack, https://www.elastic.co/products, Elokuu 2017.

[23] Elastic Inc. Elasticsearch Reference, https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html, Lokakuu 2017.

[24] Elastic Inc. Filebeat Reference, https://www.elastic.co/guide/en/beats/filebeat/current/index.html, Lokakuu 2017.

[25] Elastic Inc. Kibana Reference, https://www.elastic.co/guide/en/kibana/current/index.html, Lokakuu 2017.

[26] Elastic Inc. Logstash Reference, https://www.elastic.co/guide/en/logstash/current/index.html, Lokakuu 2017.

[27] Facebook Inc. Facebook-kirjautumispalvelu, https://developers.facebook.com/docs/facebook-login/, Syyskuu 2017.

[28] Google Inc. Google-kirjautumispalvelu, https://developers.google.com/identity/, Syyskuu 2017.

[29] Rick Kazman, Gregory Abowd, Len Bass, and Paul Clements. Scenario-based analysis of software architecture. *IEEE software*, 13(6):47–55, 1996.

[30] Euroopan komissio. Special Eurobarometer 369 - Attitudes on Data Protection and Electronic Identity in the European Union, 2011.

[31] Euroopan komissio. Executive summary of the impact assessment, 2012.

[32] Mohit Kumar. Apple macOS High Sierra Bug Exposes Passwords of Encrypted APFS Volumes As Hint, https://thehackernews.com/2017/10/macos-high-sierra-apfs-password.html, Lokakuu 2017.

[33] P. Louridas. Static code analysis. *IEEE Software*, 2006.

[34] R.C. Martin. *Agile Software Development: Principles, Patterns, and Practices*. Alan Apt series. Pearson Education, 2003.

[35] S. Nokes and S. Kelly. *The Definitive Guide to Project Management: The Fast Track to Getting the Job Done on Time and on Budget*. Financial Times. Pearson Education. Prentice Hall Financial Times, 2007.

[36] Oikeusministeriö. Suomen perustuslaki, 2000.

[37] Geniem Oy. GDPR Start, https://www.geniem.com/gdpr-start/, Heinäkuu 2017.

[38] F. Paetsch, A. Eberlein, and F. Maurer. Requirements engineering and agile software development. In *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.*, pages 308–313, Kesäkuu 2003.

[39] Euroopan parlamentti ja neuvosto. Direktiivi 95/46/ey yksilöiden suojelusta henkilötietojen käsittelyssä ja näiden tietojen vapaasta liikkuvuudesta, 1995.

[40] Euroopan parlamentti ja neuvosto. Asetus luonnollisten henkilöiden suojelusta henkilötietojen käsittelyssä sekä näiden tietojen vapaasta liikkuvuudesta ja direktiivin 95/46/EY kumoamisesta (yleinen tietosuoja-asetus), 2016.

[41] Antti Poikola, Kai Kuikkaniemi, and Harri Honko. Mydata a nordic model for human-centered personal data management and processing. *Finnish Ministry of Transport and Communications*, 2015.

[42] Dirk Riehle. *Framework Design: A Role Modeling Approach*. Väitöskirja, ETH Zürich, 2000.

[43] X. Su, J. Hyysalo, M. Rautiainen, J. Riekki, J. Sauvola, A. I. Maarala, H. Hirvonsalo, P. Li, and H. Honko. Privacy as a service: Protecting the individual in healthcare data processing. *Computer*, 49(11):49–59, Marraskuu 2016.

[44] Helsinki Instutue Of Technology. MyData-spesifikaatio, https://github.com/HIIT/mydata-stack, Syyskuu 2017.

[45] Christina Tikkinen-Piri, Anna Rohunen, and Jouni Markkula. Eu general data protection regulation: Changes and implications for personal data collecting companies. *Computer Law & Security Review*, 2017.

[46] Inger Anne Tøndel, Maria B Line, and Martin Gilje Jaatun. Information security incident management: Current practice as reported in the literature. *Computers & Security*, 45:42–57, 2014.

[47] Euroopan unioni. Euroopan unionin perusoikeuskirja, 2000.

[48] B. Van Alsenoy, V. Verdoodt, R. Heyman, J. Ausloos, E. Wauters, and G. Acar. From social media service to advertising network - A critical analysis of Facebook's Revised Policies and Terms. *Belgian Privacy Commission*, 2015.

[49] A.F. Westin. *Privacy and Freedom*. Bodley Head, 1970.

[50] Wikipedia. Esimerkkejä dokumenttityyppisistä tietokannoista, https://en.wikipedia.org/wiki/Category:Document-oriented_databases, Syyskuu 2017.

LIITTEET

Luvun 5.4 sanomaesimerkkejä

Hakusanoma

```
{
  "uuid": "eb539cbe-c830-4857-82a1-93aebf7f9ede",
  "referencedMessage": null,
  "timestamp": "2017-10-05T20:12:43.511Z",
  "type": "GET",
  "person": "acb19020-9d97-46b6-96b7-5e5014ef8227",
  "sender": "CORE"
}

```

Hakusanoman vastaus

```
{
  "uuid": "9522c675-66ed-4055-963a-4a9a293935b6",
  "referencedMessage": "eb539cbe-c830-4857-82a1-93aebf7f9ede",
  "timestamp": "2017-10-05T20:12:43.511Z",
  "type": "GET",
  "person": "acb19020-9d97-46b6-96b7-5e5014ef8227",
  "sender": "SUBSERVICE",
  "status": 200,
  "payload": {
    "subscriptions": [
      {
        "id": 12345,
        "paperCode": "Lehti"
      }
    ],
    "accountName": {
      "value": "mattiMeikkalainen89",
      "type": "string",
      "maxLength": 50
    }
  }
}
```

Oikaisusanoma


```
{
  "uuid": "85c36578-2859-4c91-8bbe-e68028f1b930",
  "referencedMessage": null,
  "timestamp": "2017-10-05T20:12:43.511Z",
  "type": "POST",
  "person": "acb19020-9d97-46b6-96b7-5e5014ef8227",
  "sender": "CORE",
  "payload": {
    "SUBSERVICE": {
      "accountName": "mattiMeikalainen89"
    }
  }
}
```


Oikaisusanoman vastaus


```
{
  "uuid": "ce3d5e72-d531-42b2-9109-fde8fdc4c184",
  "referencedMessage": "85c36578-2859-4c91-8bbe-e68028f1b930",
  "timestamp": "2017-10-05T20:12:43.511Z",
  "type": "POST",
  "person": "acb19020-9d97-46b6-96b7-5e5014ef8227",
  "sender": "SUBSERVICE",
  "status": 200
}
```

Poistosanoma


```
{
  "uuid": "1b15818d-beb8-4f7a-943c-7f58e74c5e19",
  "referencedMessage": null,
  "timestamp": "2017-10-05T20:12:43.511Z",
  "type": "DELETE",
  "person": "acb19020-9d97-46b6-96b7-5e5014ef8227",
  "sender": "CORE"
}
```


Poistosanoman vastaus


```
{
  "uuid": "fe934ab6-654b-4295-bcb1-0d07fd0ae7af",
  "referencedMessage": "1b15818d-beb8-4f7a-943c-7f58e74c5e19",
  "timestamp": "2017-10-05T20:12:43.511Z",
  "type": "DELETE",
  "person": "acb19020-9d97-46b6-96b7-5e5014ef8227",
  "sender": "SUBSERVICE",
  "status": 200
}
```