

Tiina Mankonen

**ARKKITEHTUURIN DOKUMENTOINTI KEHITTÄJÄN  
JA YLLÄPIDON NÄKÖKULMASTA - CASE: KANTA-  
PALVELUT**



JYVÄSKYLÄN YLIOPISTO  
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA  
2022

## TIIVISTELMÄ

Mankonen, Tiina Katriina

Arkkitehtuurin dokumentointi kehittäjän ja ylläpidon näkökulmasta - Case: Kanta-palvelut

Jyväskylä: Jyväskylän yliopisto, 2022, 74 s.

Tietojärjestelmätiede, pro gradu -tutkimus

Ohjaaja: Halttunen, Veikko

Tietojärjestelmäkehityksessä arkkitehtuurin dokumentoinnilla on suuri merkitys. Merkitys korostuu kehityksen eri vaiheissa, niin uuden työntekijän perehdytyksessä kuin ylläpidossa tuotannon ongelman selvittelyssä. Alan kirjallisuudessa dokumentaation merkitys korostuu erityisesti, kun dokumentaatio ei ole ajan tasalla tai helposti löydettävissä ja sidosryhmien välisessä kommunikaatiossa. Myös ketterä kehitys antaa oman näkökulman dokumentaation merkitykselle. Tällä pro gradu -tutkimuksella pyrittiin etsimään vastauksia siihen, mikä on Kanta-palveluiden arkkitehtuurin dokumentaation nykytila, millaiselle arkkitehtuurin dokumentaatiolle on tarvetta ja miten arkkitehtuuria tulisi mallintaa, jotta se voisi palvella mahdollisimman laajaa joukkoa henkilöstöstä yli organisaatorajojen. Tutkimuksen empiirinen osuus toteutettiin teemahaastatteluilla, johon osallistui 13 asiantuntijaa Kanta-palveluista. Tutkimuksen motivaationa oli tutkijan oma mielenkiinto arkkitehtuurin dokumentaatioon sekä tarve kehittää dokumentaatiota Kanta-palveluissa. Tutkimustulokset analysoitiin sisällönanalyysillä. Teemahaastatteluiden perusteella havaittiin, että dokumentaatio on kirjavaa ja ne ovat hajautuneet moneen eri paikkaan. Tämä johtaa siihen, että dokumentaatiot eivät ole ajan tasalla tai helposti löydettävissä. Puutteellisesta dokumentaatiosta on enemmän haittaa kuin hyötyä. Dokumentaation merkitys myös korostui sidosryhmien välisessä kommunikaatiossa ja yhteisten mallien puutteena. Haasteet johtavat siihen, että dokumentaatiota ei tuoteta kirjoitettuun muotoon riittävällä tavalla tai dokumentaatio jopa sivuutetaan ja tehdään, kun siihen on aikaa. Haasteet voidaan voittaa sillä, että dokumentaatio on osa jatkuvaa kehittämistä ja siihen on varattu riittävästi aikaa. Kehityksessä täytyy ottaa selkeitä askeleita dokumentaation selkeyttämiseen kehityksen alkuvaiheesta asti. Tähän voi auttaa myös työpajojen järjestäminen aiheen tiimoilta. Ohjeistusta täytyy päivittää ja työkaluja ottaa käyttöön aktiivisemmin.

Asiasanat: ohjelmistoarkkitehtuuri, arkkitehtuurin dokumentointi, kommunikaatio

## ABSTRACT

Mankonen, Tiina Katriina

Architecture documentation from a developer and maintenance perspective –

Case: Kanta-services

Jyväskylä: University of Jyväskylä, 2022, 74 pp.

Information Systems, Master's Thesis

Supervisor: Halttunen, Veikko

In information system development, the documentation of architecture is of the great importance. The importance is emphasized at different stages of development, both in the orientation of a new employee and in the maintenance when dealing with the issue from the production. In the literature, the importance of documentation is especially emphasized when the documentation is not up-to-date or easy to find and in communication between stakeholders. Agile development also gives its own perspective on the importance of documentation. The aim of this master's study was to find answers to the current state of the architecture documentation for Kanta services, what kind of architectural documentation is needed and how the architecture should be modeled in order to serve the widest possible number of personnel across organizational boundaries. The empirical part of the study was carried out through thematic interviews with 13 experts from Kanta services. The motivation for the research was the researcher's own interest in the documentation of architecture and the need to develop documentation in Kanta services. The study results were analyzed by content analysis. Based on the thematic interviews, it was found that the documentation is diverse and scattered in many different places. This results in the documentation not being up-to-date or easy to find. Incomplete documentation does more harm than good. The importance of documentation was also emphasized in the communication between stakeholders and in the absence of common models. Challenges result in the documentation not being produced in a written form adequately, or the documentation is even ignored and done when time allows. Challenges can be overcome by ensuring that documentation is part of continuous development and that sufficient time is set aside for it. Clear steps must be taken early on in the development to clarify the documentation. Organizing workshops on the topic can also help with this. The instructions need to be updated and the tools must be taken into use more actively.

Keywords: software architecture, architecture documentation, communication

## KUVIOT

KUVIO 1	Zachman kokonaisarkkitehtuuriviitekehys .....	14
KUVIO 2	TOGAF ADM-metodi .....	15
KUVIO 3	JHS 179 Arkkitehtuurikuvausten viitekehys.....	17
KUVIO 4	ArchiMate viitekehysten ydin .....	23
KUVIO 5	BPMN ydin ja kerros rakenne .....	24
KUVIO 6	UML kaaviotyypit .....	25
KUVIO 7	Kanta-arkkitehtuuri .....	31
KUVIO 8	Dokumentaation tarpeet .....	59
KUVIO 9	Dokumentoinnin polku kehitystiimissä .....	63

## TAULUKOT

TAULUKKO 1	Haastateltavien taustatietoja .....	38
TAULUKKO 2	Haastattelun tiedot .....	39
TAULUKKO 3	Dokumentoinnin parantamisen SWOT-analyysi.....	62

# SISÄLLYS

TIIVISTELMÄ .....	2
ABSTRACT .....	3
KUVIOT .....	4
SISÄLLYS.....	5
1 JOHDANTO.....	7
1.1.1 Tutkimuksen tavoite ja saavutetut tulokset .....	8
1.1.2 Tutkimuksen tiedonhakumenetelmät ja rakenne.....	8
2 ARKKITEHTUURIT JA NIIDEN DOKUMENTOIMINEN .....	10
2.1 Arkkitehtuurit .....	10
2.1.1 Kokonaisarkkitehtuuri .....	10
2.1.2 Ohjelmistoarkkitehtuuri.....	11
2.2 Viitekehykset.....	13
2.2.1 Zachman:n viitekehys.....	13
2.2.2 TOGAF .....	14
2.2.3 JHS-179.....	15
2.2.4 Tiedonhallintalaki .....	17
2.3 Arkkitehtuurin dokumentointi.....	18
2.3.1 Dokumentaatio ketterässä kehityksessä .....	21
2.3.2 Dokumentaatio kommunikaation välineenä .....	21
2.3.3 ArchiMate.....	22
2.3.4 BPMN.....	23
2.3.5 UML .....	24
2.3.6 Yhteenvedo.....	25
3 ARKKITEHTUURIKUVAUSTEN DOKUMENTOINTI KANTA- PALVELUISSA.....	27
3.1 Kanta-palvelut lyhyesti.....	27
3.1.1 Omakanta-palvelu.....	28
3.1.2 Omatietovaranto.....	28
3.1.3 Reseptikeskus ja lääketietokanta.....	28
3.1.4 Potilastiedon arkisto .....	29
3.1.5 Kuva-aineistojen arkisto.....	29
3.1.6 Sosiaalihuollon asiakastiedon arkisto .....	29
3.2 Kanta-arkkitehtuuri.....	30
3.2.1 Arkkitehtuurin kehittäminen ja dokumentointi Kanta- palveluissa.....	31
3.2.2 Arkkitehtuuriperiaatteet .....	32

4	TUTKIMUSMENETELMÄ .....	34
4.1	Tutkimusote ja tiedonkeruumenetelmät .....	34
4.2	Haastateltavien valinta .....	36
4.2.1	Haastattelujen toteutus.....	36
4.2.2	Haastateltavien taustatiedot .....	36
4.3	Aineiston analysoinnin toteutus.....	38
4.3.1	Haastattelun litterointi .....	40
4.3.2	Aineiston luotettavuus .....	40
5	TUTKIMUKSEN TULOKSET .....	41
5.1	Arkkitehtuurin dokumentoinnin nykytila.....	41
5.1.1	Dokumentaation löydettävyys .....	43
5.1.2	Dokumentaation ajantasaisuus .....	45
5.1.3	Mallinnus- ja piirtotyökalut.....	47
5.2	Arkkitehtuurin dokumentoinnin tarve .....	48
5.2.1	Täydellinen dokumentaatio, onko sitä? .....	51
5.2.2	Yhteiset mallit .....	52
5.3	Arkkitehtuurin dokumentointi kommunikaation välineenä .....	53
6	JOHTOPÄÄTÖKSET JA POHDINTA.....	57
6.1	Tutkimustulosten pohdintaa.....	60
6.2	Tutkimuksen arviointi .....	63
6.3	Aiheita jatkotutkimukseen .....	64
7	YHTEENVETO .....	65
	LÄHTEET .....	66
	LIITE 1 SÄHKÖPOSTIKYSELY HAASTATELTAVILLE .....	70
	LIITE 2 TEEMAHAASTATTELU KYSYMYKSET .....	72

# 1 JOHDANTO

Kanta-palvelut on moniulotteinen palvelu, jossa tulee ottaa huomioon riittävällä tavalla tehty arkkitehtuurin dokumentointi. Tutkimuksessa selvitetään, millaista tiettyjen Kanta-palveluiden sisällä olevien palveluiden arkkitehtuurin dokumentointi tällä hetkellä on ja miten siitä mahdollisesti saadaan toimivampi.

Tarkoituksena on tarkastella ja vertailla kehittäjien ja ylläpidon näkemyksiä arkkitehtuurin dokumentoinnista kuudesta eri palvelusta. Nämä palvelut ovat Omakanta, Omatietovaranto, Potilastiedon arkisto, Kuva-aineistojen arkisto, Resepti-palvelu ja Sosiaalihuollon arkisto. On tärkeää saada tietoa siitä, mikä dokumentoinnin aito tarve on. Kohderyhmältä odotetaan, että saadaan selville heidän näkemyksiänsä ja tarpeita dokumentaatiolle. Näin ollen voidaan tehdä oikeanlaista dokumentointia, jotta näiden kahden tarpeet otetaan huomioon. Tämän jälkeen on mahdollista antaa kehitysehdotuksia dokumentoinnin kehittämiseksi, dokumentaation tehokkaammalle käytölle ja sitä myötä saada selkeämpi ja toimivampi dokumentaatiokokonaisuus.

Kanta-palveluiden monimutkaisen rakenteen vuoksi arkkitehtuurin dokumentointi tarvitsee kokonaisuudessaan uudenlaista tarkastelua. Kehityksen ja ylläpidon tueksi dokumentoinnin tulisi olla mahdollisimman selkeä kokonaisuus. Palveluiden kehittämisen erityispiirteisiin kuuluu yhteentoimivuus muiden tietojärjestelmien kanssa, yhteistyö ulkopuolisten sidosryhmien ja Kanta-palveluun sisältyvien palveluiden kanssa, tietoturva, tietosuojat sekä laki- ja asetussidonnaisuus.

Arkkitehtuurin dokumentointi on keskeinen osa tietojärjestelmien kehittämistä. On tärkeää, että arkkitehtien ja kehitystiimin välillä on selkeää ja riittävää dokumentaatiota. Tämän vuoksi aiheesta on tärkeää käsitellä tapausorganisaatiossa ja palvelussa.

Tutkimus on rajattu koskemaan vain osaa Kanta-palveluiden sisäisistä palveluista. Palveluita on yhteensä 13, joista otettiin mukaan 6. Tämä raja on sen vuoksi, ettei tutkimuksesta olisi tullut liian laaja.

Tutkimuksen keskeiset käsitteet

- Kokonaisarkkitehtuuri (engl. enterprise architecture)

- Ohjelmistoarkkitehtuuri (engl. software architecture)
- Arkkitehtuurin dokumentointi (engl. architecture documentation)
- Kommunikaatio (engl. communication)

### 1.1.1 Tutkimuksen tavoite ja saavutetut tulokset

Tämän tutkimuksen tavoitteena on selvittää arkkitehtuurin dokumentointia kehittäjän ja ylläpidon näkökulmista. Aihetta tarkastellaan kirjallisuuden ja empiirisen tutkimuksen avulla ja yritetään saada selville mikä arkkitehtuurin dokumentoinnissa on tarpeellista. Tavoitteena on myös selvittää millä tavalla dokumentointia olisi tehtävä, jotta siitä hyötyy mahdollisimman moni käyttäjäryhmä, ensisijaisesti kehittäjät ja ylläpito. Tähän tavoitteeseen esitetään seuraavat tutkimuskysymykset:

1. Mikä on arkkitehtuurin dokumentoinnin tarve kehittäjän ja ylläpidon näkökulmista?
2. Kehittäjän ja ylläpidon töiden erottaminen ja niihin liittyvä dokumentoinnin tarve?
3. Miten arkkitehtuuria tulisi mallintaa, jotta se voisi palvella mahdollisimman laajaa joukkoa henkilöstössä yli organisaatorajojen?

Teemahaastattelun perusteella havaittiin, että dokumentaatio on osittain puutteellista ja dokumentaatiot ovat hajautuneet moneen eri paikkaan. Tämä johtaa siihen, että dokumentaatiot eivät ole ajan tasalla tai helposti löydettävissä. Dokumentaation merkitys korostui sidosryhmien välisessä kommunikaatiossa ja yhteisiä malleja kaivataan työn helpottamiseksi. Haasteet johtavat siihen, että dokumentaatiota ei tuoteta riittävällä tavalla tai dokumentaatio jopa sivuutetaan ajanpuutteen vuoksi. Dokumentaation on oltava osa jatkuvaa kehittämistä ja sen työstämiselle on varattava työaika. Dokumentaation kehittäminen on aktiivisesti otettava työn alle ja kehitettävä siitä toimiva kokonaisuus.

### 1.1.2 Tutkimuksen tiedonhakumenetelmät ja rakenne

Tutkimukseen liittyvien artikkeleiden ja kirjallisuuden hakeminen tehtiin pääosin sähköisesti. Muutamia kirjallisuuden lähteitä on luettu fyysisestä kirjasta esimerkiksi arkkitehtuurin dokumentaatio ja tutkimusmenetelmät lukuun. Sähköisinä hakupalveluina käytettiin Jyväskylän yliopiston JykDok-palvelua ja sieltä löytyviä tietokantoja, ResearchGate-yhteisöä, Google Scholar -hakupalvelua, O'Reilly e-kirjapalvelua sekä Googlea yleisesti. Hakusanoina on käytetty esimerkiksi seuraavia: "Zachman framework", "enterprise framework", "TOGAF framework", "JHS-179", "JHS-suositukset", "software architecture documentation", "software architectural knowledge", "knowledge communication". Myös hakusanoilla löytyneiden artikkeleiden lähdeluettelosta on ammennettu uusia lähteitä ja näin pystytty siirtymään lähteestä toiseen.



Käytetyt lähteet on valikoitu niin, että ne ovat yhdistettävissä tutkimuksen aiheeseen. Sopivuutta on arvioitu otsikon, tiivistelmän, yhteenvedon ja lopulta koko lähteen sisällön perusteella. Lähdeluetteloon päätyi 53 lähdetä.

Lähteiden luotettavuutta on voitu arvioida sillä, missä artikkeli on julkaistu. Esimerkiksi jos julkaisu tai artikkeli löytyi yliopiston kanssa yhteistyötä tekevistä Emerald-palvelusta tai ResearchGate-yhteisöstä, lähde on arvioitu luotettavaksi. Tutkimusmenetelmiä koskevat kirjat on arvioitu luotettavaksi, koska ne löytyvät myös JykDok-palvelusta sekä ovat lainattavissa fyysisinä kirjoina yliopiston kirjastosta tai Jyväskylän kaupungin kirjastosta.

Tutkimus koostuu seitsemästä (7) pääluvusta: johdanto, arkkitehtuurit ja niiden dokumentointi, arkkitehtuurin dokumentointi Kanta-palveluissa, tutkimusmenetelmä, tutkimuksen tulokset, tutkimuksen johtopäätökset ja pohdinta sekä yhteenvedo. Johdannossa kerrotaan tutkimuksesta yleisesti, sisältäen tutkimuksen tavoitteita ja tutkimuskysymykset. Toisessa luvussa käsitellään teoriaan liittyviä arkkitehtuureja ja niiden dokumentaatiota. Kolmannessa luvussa kerrotaan lyhyesti mikä on Kanta-palvelut ja tarkastellaan tämänhetkistä arkkitehtuurin dokumentaation toteutumista Kanta-palveluissa. Neljännessä luvussa esitellään tutkimusmenetelmät. Viidennessä luvussa käydään läpi tutkimuksen tulokset. Kuudennessa luvussa kerrotaan tutkimuksen johtopäätökset ja pohdintaa. Tutkimus päättyy seitsemänten lukuun, jossa kerrataan tutkimuksen pääpiirteet yhteenvedoon.

## 2 ARKKITEHTUURIT JA NIIDEN DOKUMENTOIMINEN

Tässä luvussa avataan muun muassa seuraavia käsitteitä kuten kokonaisarkkitehtuuri, arkkitehtuurikuvaus ja dokumentointi. Kerrotaan mitä kokonaisarkkitehtuuri ja järjestelmäarkkitehtuuri yleisesti ottaen tarkoittavat, mitkä ovat niiden olennaisia rakenteita, viitekehukset ja mitä on arkkitehtuurinäkökulmat. Lisäksi kerrotaan mitä arkkitehtuurikuvauksen on tarkoitus kuvata sekä miten arkkitehtuurikuvauksia olisi tarkoitus dokumentoida. Lopuksi hieman avataan, millaisia tutkimuksia arkkitehtuurista on aiemmin tehty. Luvun tarkoituksena on, että lukijalle tulee käsitys kokonaisarkkitehtuurin ja arkkitehtuurin dokumentoinnin merkityksestä. Tämän luvun jälkeen lukijan on jatkossa toivottavasti helpompi ymmärtää tutkimustuloksia arkkitehtuurin dokumentoinnista.

### 2.1 Arkkitehtuurit

Arkkitehtuurin suunnittelu on tärkeää tietojärjestelmien kehittämisen yhteydessä. Varsinkin hyvin laajojen ja monimutkaisten tietojärjestelmien suunnittelu vaatii suunnitelmallisuutta ja hyvää arkkitehtuuria. Hyvin suunniteltu tietojärjestelmä on suunnitteluvaiheessa pilkottu pieniksi osiksi, näin määrittely, toteutus, testaus ja ylläpito ovat vaivattomampaa, kun pieniä toiminnallisuuksia voi kehittää omina yksikköinä. Arkkitehtuurin kautta saa kattavan ylätason näkymän kehitettävänä olevaan tietojärjestelmään ja arkkitehtuuri toimii myös kommunikaation välineenä eri sidosryhmien välillä. Tässä alaluvussa kerrotaan lyhyesti mitä kokonaisarkkitehtuuri ja ohjelmistoarkkitehtuurit tarkoittavat.

#### 2.1.1 Kokonaisarkkitehtuuri

Kokonaisarkkitehtuurin synty ja määritelmä on lähtöisin 1980-luvulta (Zachman, 1987). Kotusev (2018) mainitsee kokonaisarkkitehtuurin olevan kokonaiskuva organisaatiosta, jossa organisaatiota katsotaan liiketoiminnan ja tietojärjestelmien näkökulmista. Lankhorts (2017) määrittelee kokonaisarkkitehtuurin olevan johdonmukainen kokonaisuus periaatteita, menetelmiä ja malleja, joita käytetään organisaatorakenteen, liiketoimintaprosessien, tietojärjestelmien ja infrastruktuurin suunnitteluun ja toteuttamiseen. Kokonaisarkkitehtuuri vaikuttaa siis koko organisaatioon, liiketoiminnan perusrakenteisiin asti. Kokonaisarkkitehtuurin tärkein ominaisuus on se, että se tarjoaa kokonaisvaltaisen näkemyksen organisaatiosta, eli se kokoaa liiketoiminnan, IT:n ja siihen liittyvät olennaiset asiat yhteen (Lankhorst, 2017).

Kokonaisarkkitehtuurin viitekehukset (engl. framework) antavat selkeyttä organisaation IT-järjestelmään ja helpottavat kokonaisarkkitehtuurin suunnittelamista. Kansainvälisiä viitekehyksiä ovat esimerkiksi Zachman:n viitekehys ja

TOGAF sekä kotimainen JHS 179 -viitekehys ja tiedonhallintalaki. Tässä tutkimuksessa on keskitytty näihin aiemmin mainittuihin viitekehyksiin, koska ne ovat relevantteja kohdeorganisaatiota ajatellen. Viitekehukset esitellään tarkemmin luvussa 2.2.

Kokonaisarkkitehtuuriin liittyy vahvasti arkkitehtuurinäkökulmat, joita ovat tietoarkkitehtuuri (engl. information architecture), liiketoiminta- tai toiminta-arkkitehtuuri (engl. business architecture), tietojärjestelmäarkkitehtuuri (engl. systems architecture) ja sovellus- ja teknologia-arkkitehtuuri (engl. technology architecture). Tietoarkkitehtuurin avulla voi kuvata organisaation tuottamaa tietoa eli dataa tarkemmalla tasolla. Sillä voi kuvata mm. tiedon arvoketjua ja tietovarastoja (Pulkkinen, 2006). Liiketoiminta- tai toiminta-arkkitehtuurin periaatteena on kuvata organisaation rakenteita toiminnalliselta näkökulmalta, kuten liiketoimintaprosesseja, strategiaa ja visiota (Pulkkinen, 2006). Toiminnallisia rakenteita voi olla myös organisaation eri sidosryhmät ja palvelut. Tietojärjestelmäarkkitehtuurin tavoitteena on kuvata organisaation tietojärjestelmät. Näitä kuvauksia ovat muun muassa tietojärjestelmäkartta, sovellusmallit ja kehittäjien ohjeistus (Pulkkinen, 2006). Viimeisen eli sovellus- ja teknologia-arkkitehtuurin tavoitteena on kertoa mitä teknologioita toteutuksessa käytetään ja teknologian rakenteet eli millä teknologialla järjestelmä pitää tehdä. Se kuvaa teknologista infrastruktuuria, esimerkiksi alustoja, verkkoja sekä tiedonkulku (Pulkkinen, 2006). Fowler (2002) mainitsee, että tietojärjestelmässä on useita arkkitehtuureja ja näkemys siitä, mikä on arkkitehtonisesti merkittävää, voi muuttua järjestelmän elinkaaren aikana.

Tutkimuksen kannalta kokonaisarkkitehtuurin rinnalla on oleellista keskittyä tarkentamaan myös seuraavaa käsitettä eli ohjelmistoarkkitehtuuria, joka on lähtökohta tietojärjestelmän suunnittelulle ja siitä muodostuville malleille ja kuvauksille. Luvussa käsitellään myös ohjelmistoarkkitehdin roolia ja ohjelmistoarkkitehtuurin merkitystä tiedon välittämisessä ja kommunikaatiossa.

### 2.1.2 Ohjelmistoarkkitehtuuri

Kokonaisarkkitehtuuri kuvaa koko organisaation ketjun liiketoiminnan perusrakenteista lähtien tietojärjestelmien rakenteisiin saakka. Ohjelmistoarkkitehtuuri sen sijaan kuvaa yksittäisen ohjelmiston arkkitehtuuria. Ohjelmistoarkkitehtuuri on ns. tietojärjestelmän yleissuunnitelma, joka kertoo miten asioiden tulisi toimia. Richards ja Ford (2021) kertovat, että ohjelmistoarkkitehtuuri on käsitteenä vaikea, toimialalla on ollut vaikeuksia määritellä sitä. Ohjelmistoarkkitehtuuria kutsutaan järjestelmän suunnitelmaksi tai järjestelmän kehittämisen tiekartaksi (Richards & Ford, 2021). Koskimies ja Mikkonen (2005) mainitsevat myös, että ohjelmistoarkkitehtuurille on useita määritelmiä mutta tärkeintä on, että arkkitehtuuri kuvaa osien välisiä suhteita eikä vain järjestelmän osiin jakamista. Arkkitehtuuri koskee järjestelmän rakennetta sekä käyttäytymistä (Koskimies & Mikkonen, 2005). Richards ja Ford (2021) itse määrittelevät ohjelmistoarkkitehtuuri käsitteen koostuvan järjestelmän rakenteesta, yhdistettynä järjestelmän tukemiin arkkitehtuurin ominaisuuksiin, arkkitehtuuripäätöksiin ja

suunnitteluperiaatteisiin. Järjestelmän rakenne viittaa arkkitehtuurityyleihin, jolla järjestelmä on rakennettu. Arkkitehtuurin ominaisuudet määrittelevät järjestelmän onnistumiskriteerit, joita on esimerkiksi käytettävyys, skaalautuvuus, testattavuus ja luotettavuus. Arkkitehtuuripäätökset sen sijaan määrittelevät säännöt tietojärjestelmän rakentamiselle, ne muodostavat tietojärjestelmälle rajoitteita ja ohjaavat kehitystiimin toimintaa. Suunnitteluperiaate ei ole suoranainen sääntö vaan ohjeistus, jolla tietojärjestelmää tulisi suunnitella. Sillä voi ohjata noudattamaan esimerkiksi suositeltua menetelmää (Richards & Ford, 2021).

Richards ja Ford (2021) määrittelevät arkkitehtuurityylin (engl. architecture style) yleiseksi rakenteeksi käyttöliittymän ja taustan lähdekoodin ja kuinka lähdekoodi on vuorovaikutuksessa tietovaraston kanssa. Arkkitehtuurimallit (engl. architecture model) puolestaan ovat alemman tason suunnittelurakenteita, jotka helpottavat muodostamaan ratkaisuja arkkitehtuurityylin sisällä. Arkkitehtuurityylit auttavat arkkitehtejä tekemään oikeita valintoja tiettyä liiketoiminnallista ongelmaa varten (Richards & Ford, 2021). He ovat jaotelleet arkkitehtuurityylit kahteen eri päätyyppiin, näitä on monoliittiset ja hajautetut arkkitehtuurit.

#### Monoliittiset arkkitehtuurit:

- kerrosarkkitehtuuri (engl. layered architecture)
- tietovuoarkkitehtuuri (engl. pipeline architecture)
- mikroydin arkkitehtuuri (engl. mikrokernet architecture)

#### Hajautetut arkkitehtuurit:

- palvelupohjainen arkkitehtuuri (engl. service-based architecture)
- tapahtumalähtöinen arkkitehtuuri (engl. event-driven architecture)
- avaruus-pohjainen arkkitehtuuri (engl. space-based architecture)
- palvelukeskeinen arkkitehtuuri (engl. service-oriented architecture)
- mikropalveluarkkitehtuuri (engl. microservices architecture) (Richards ja Ford (2021)).

Monoliittinen kerrosarkkitehtuuri on yksi yleisimmistä arkkitehtuurityyleistä ja on ns. defacto standardi useimmille ohjelmistoarkkitehtuureille sen yksinkertaisuuden, tunnettavuuden ja alhaisten kustannusten vuoksi (Richards & Ford, 2021). Hajautetut arkkitehtuurityylit ovat tehokkaampia suorituskyvyltään, skaalautuvuudeltaan ja saatavuudeltaan kuin monoliittiset arkkitehtuurityylit (Richards & Ford, 2021).

Ohjelmistoarkkitehtuurin parissa työskentelee ohjelmistoarkkitehti. Richards ja Ford (2021) mainitsevat, että ohjelmistoarkkitehdin roolin määrittäminen on yhtä vaikeaa kuin ohjelmistoarkkitehtuurin määrittäminen. Ohjelmistoarkkitehdin roolin omaavalta henkilöltä odotetaan arkkitehtuuripäätösten tekemistä, arkkitehtuurin jatkuvaa analysointia, kykyä pysyä uusimpien

trendien mukana, varmistaa, että päätöksiä noudatetaan, monipuolista kokemusta, liiketoiminta-alueen tuntemusta, ihmissuhdetaitoja ja politiikan ymmärrystä (Richards & Ford, 2021). Organisaation sisällä on monta eri sidosryhmää, jotka ovat tekemisissä kokonaisarkkitehtuurin kanssa. Tämän vuoksi kommunikatio ja viestintä on tärkeää kaikkien eri sidosryhmien välillä.

## 2.2 Viitekehukset

### 2.2.1 Zachman:n viitekehys

Zachmanin viitekehys on kehitetty vuonna 1987 John Zachmanin toimesta. Kotusev (2016) mainitsee artikkelissaan, että monet tekijät kutsuvat John Zachmania kokonaisarkkitehtuurin "isäksi", joka on luonut kokonaisarkkitehtuuri käsitteen ja inspiroinut myöhemmin julkaistuja kokonaisarkkitehtuuri viitekehyyksiä ja menetelmiä.

Zachmanin viitekehys ei sisällä metodia, jolla arkkitehtuuria voidaan toteuttaa, vaan sillä on tarkoitus kuvata organisaation nykyinen tai haluama rakenne. Viitekehys tulee täydentää metodilla, jolla arkkitehtuuria voidaan toteuttaa. Viitekehys on näin ollen arkkitehtuurin perusta (Zachman International Enterprise Architecture, 2008).

Zachmanin viitekehys kuvataan tyypillisesti 6x6 matriisina (kuvio 1). Sarakkeet esittävät organisaatiolle seuraavia kysymyksiä kuten mitä, miten, missä, kuka, milloin ja miksi. Viitekehyyksen rivit esittävät organisaatiolle eri perspektiivejä, esimerkiksi suunnittelijan, liiketoimintajohdon (omistajan) tai arkkitehdin näkökulmasta. Näin ollen eri perspektiiveistä saadaan selville tarvittavia tietoja, joilla organisaatio voi toteuttaa määrittelyä ja kehittämistä (Zachman International Enterprise Architecture, 2008).

	Mitä?	Kuinka?	Missä?	Kuka?	Milloin?	Miksi?	
Suunnittelija							Laajuus (konteksti)
Omistaja							Arkkitehtuurimalli (käsitteellinen)
Arkkitehti							Systeemimalli (looginen)
Rakentaja							Teknologiamalli (fyysinen)
Ohjelmoija							Yksityiskohtaiset mallit
Käyttäjä							Toteutus/käyttöönotto
	Tieto	Toiminto	Verkko	Ihmiset	Aika	Motivaatio	

KUVIO 1 Zachman kokonaisarkkitehtuuriviitekehys (Zachman International Enterprise Architecture, 2008)

Lankhorst (2017) mainitsee Zachman:n viitekehysten eduiksi ymmärrettävyyden, koska viitekehys koskee koko organisaatiota. Viitekehys on määriteltävissä menetelmistä tai viitekehyksistä riippumatta. Kaikki ongelmat voi kartoittaa viitekehysten kautta, jotta tulee ymmärrys, mihin ne sopivat. Haittapuolena viitekehyksessä on suuri solujen määrä, eikä eri solujen välisiä suhteita ole tarkasti määritelty (Lankhorst, 2017).

## 2.2.2 TOGAF

TOGAF (The Open Group Architecture Framework) on The Open Groupin kehittämä ja ylläpitämä kokonaisarkkitehtuuriviitekehys, joka tarjoaa menetelmiä, työkaluja ja ohjeita arkkitehtuurin suunnitteluun. Kaikki organisaatiot, jotka haluavat kehittää kokonaisarkkitehtuuriaan, voivat sitä maksutta vapaasti käyttää. TOGAF:ista tämän tutkimuksen kirjoitushetkellä on julkaistu versio 9.2 (The Open Group, 2018) ja TOGAFia kutsutaan oletusarvoiseksi standardiksi sen suuren levinneisyyden vuoksi kokonaisarkkitehtuurin käytössä (Kotusev, 2016). TOGAF on suunniteltu tukemaan neljää (4) kokonaisarkkitehtuurin osa-aluetta, joita ovat liiketoiminta-, tieto-, järjestelmä- ja teknologia-arkkitehtuuri (The Open Group, 2018). TOGAF:n toimintakehys (engl. TOGAF Capability Framework) muodostuu neljästä pääelementistä:

- Arkkitehtuurin kehittämisen metodi ADM (engl. Architecture Development Method), joka on TOGAF:n ydin (kuviot 2).
- Arkkitehtuurin sisällön viitekehys (engl. Architecture Content Framework) tarjoaa rakenteellisen mallin kokonaisarkkitehtuurin sisällölle, joka mahdollistaa johdonmukaisen määrittelyn, jäsentämisen ja esittämisen.
- Yrityksen jatkumo (engl. Enterprise Continuum and Tools), tarjoaa arkkitehtuurien varaston (engl. repository), jonka avulla arkkitehti voi luoda malleja siitä, miksi ja miten arkkitehtuuri on suunniteltu. Jatkumo on myös tärkeä kommunikoinnin ja viestinnän väline organisaation sisällä ja sidosryhmien välillä.
- Arkkitehtuurin toimintakehys (engl. Architecture Capability Framework), jossa käsitellään organisaation rakenteita, prosesseja, rooleja, vastuita ja taitoja, jotta arkkitehtuuri voi toteutua tarvittavalla tavalla (The Open Group, 2018).



KUVIO 2 TOGAF ADM-metodi (The Open Group, 2018, 4.2.2 mukaan)

Yksi mainituista pääelementeistä on ADM (Architecture Development Method) -metodi TOGAFin ydin, jolla organisaatiosta voi toteuttaa suunnitellun arkkitehtuurikuvauksen organisaation kehittämiseksi ja hallitsemiseksi. ADM-metodi tarjoaa testatun ja toistettavan prosessin arkkitehtuurien suunnitteluun. Metodi sisältää arkkitehtuurikehityksen vision luomisen, arkkitehtuurin sisällön kehittämisen, siirtymän suunnittelun ja arkkitehtuurien täytäntöönpanon hallinnan. Kaikki metodin toiminnot toteutetaan iteratiivisessa syklissä, mutta metodi on kuitenkin joustava, joten organisaatiot voivat käyttää sitä tarvittavilta osin liiketoimintatavoitteiden ja mahdollisuuksien mukaisesti (The Open Group, 2018).

### 2.2.3 JHS-179

JHS-suositukset on kotimainen viitekehys. Viitekehystä on valmisteltu valtion ja kuntien yhteistyössä vuosina 1992–2019 ja ovat näin ollen valtion- ja kunnallishallintoa varten kehitetty. JHS-suosituksia valmisteli julkisen hallinnon tietohallinnon neuvottelukunnan (Juhtan) alainen JHS-jaosto. JHS-järjestelmä on nyt kuitenkin lakkautettu uuden tiedonhallintalain myötä 1.1.2020 mutta JHS-

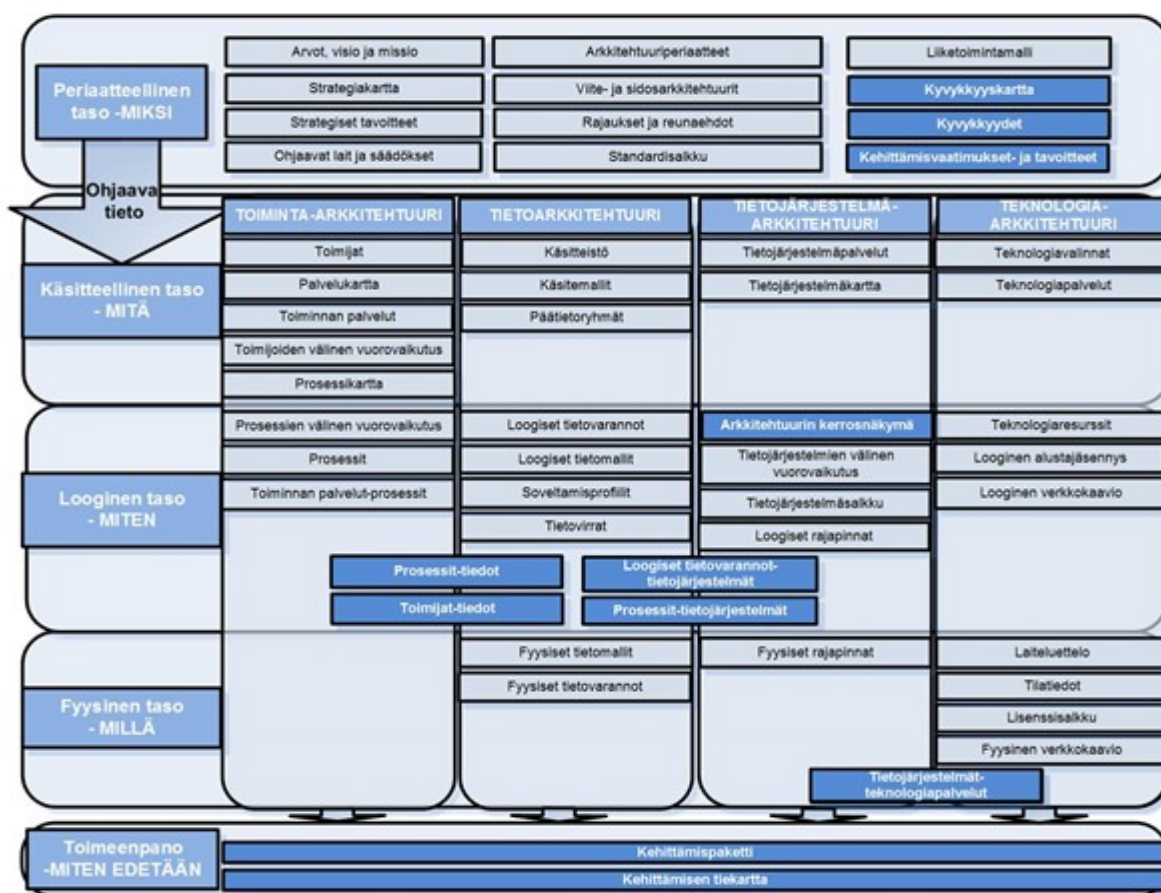
suosituksia voi edelleen käyttää siinä muodossa, jossa ne olivat 1.1.2020. JHS-suositukset eivät ole enää ajan tasalla, eikä tukea niiden käyttöön enää anneta, joten tämä pitää huomioida niitä käytettäessä. Uudesta tiedonhallintalaista kerrotaan seuraavassa luvussa lisää (Suomidigi, 2021a).

JHS-suosituksissa on kokonaisarkkitehtuurityössä käytettävä menetelmä, JHS 179, jolla voi määritellä julkisen hallinnon organisaation kokonaisarkkitehtuurin. Suositus antaa ohjeistusta arkkitehtuurin eri osa-alueiden kuvausten ja mallien laatimiseen. JHS 179 suositus perustuu TOGAF kokonaisarkkitehtuuriviitekehyksen versioon 9.1. (Suomidigi, 2021b). JHS 179:ssä on näin ollen TOGA-Fin mukaiset arkkitehtuurinäkökulmat, jotka mainittiin 2.2.2. TOGAF luvussa.

JHS 179 on kuvattu suosituksen verkkosivulla seuraavin sanoin:

*”Suosituksessa kuvataan yhtenäinen suunnittelumenetelmä sekä yhtenäiset kuvaus tavat ja -mallit julkisen hallinnon organisaatioiden kokonaisarkkitehtuurin kehittämiseen sen eri vaiheissa. Kokonaisarkkitehtuurityön tavoitteena on julkisen hallinnon organisaatioiden toiminnan ja palveluiden yhteentoimivuuden parantaminen. Kokonaisarkkitehtuurimenetelmä ei korvaa muita toiminnan kehittämisen suunnittelumenetelmiä. Kokonaisarkkitehtuuri on toiminnan, prosessien ja palvelujen, tietojen, tietojärjestelmien ja niiden tuottamien palvelujen muodostaman kokonaisuuden rakenne. Kokonaisarkkitehtuurimenetelmä on kokonaisarkkitehtuurityön järjestelmällinen työ- ja menettelytapa, jonka avulla tunnistetaan, jäsennetään, suunnitellaan ja kuvataan kokonaisuuden rakenneosat ja niiden riippuvuudet (Suomidigi, 2021c).”*





KUVIO 3 JHS 179 Arkkitehtuurikuvausten viitekehys (Suomidigi, 2021c)

JHS-179 suositukset antavat ohjeistusta strategian kuvaamiseen strategia kartan avulla, liiketoimintamalleja ja kyvykkyyksiä kokonaisarkkitehtuurin suunnitteluun. Suositusten avulla voi kuvata arkkitehtuurin nyky- ja tavoitetilaa. Suositus sisältää semanttisen yhteentoimivuuden menetelmäohjeen. Suositukset antavat ohjeistusta myös kokonaisarkkitehtuurin visuaalisten kuvauksien tuottamiseksi, integraatio ja rajapinta kuvauksen tekemiseen, virtualisointiin ja pilvipalveluihin teknologia-arkkitehtuurin suunnittelussa (Suomidigi, 2021c).

## 2.2.4 Tiedonhallintalaki

Tiedonhallintalaki (laki julkisen hallinnon tiedonhallinnasta 906/2019) on tullut voimaan 1.1.2020. Lain tarkoituksena on tiedonhallinnan järjestäminen julkisen hallinnon tiedonhallintayksiköissä. Tiedonhallintayksiköt muodostavat tiedonhallintamallin eli toiminnallisen kuvauksen organisaation toimintaympäristöstä, jonka tiedonhallintayksikön johto hyväksyy (Finlex, 2019).

Tiedonhallintalain 4§:n 1 momentin mukaan tiedonhallintayksiköitä ovat:

- valtion virastot ja laitokset;

- tuomioistuimet ja valitusasioita käsittelemään perustetut lautakunnat;
- eduskunnan virastot;
- valtion liikelaitokset:
- kunnat;
- kuntayhtymät;
- itsenäiset julkisoikeudelliset laitokset;
- yliopistolaissa tarkoitetut yliopistot sekä ammattikorkeakoululaissa tarkoitetut ammattikorkeakoulut (Finlex, 2019).

Laissa ei puhuta nimenomaan kokonaisarkkitehtuurista, vaan ainoastaan muodostettavasta dokumentaatiokokonaisuudesta, jossa tulee kuvata organisaation tiedonhallinta kokonaisuudessaan. Tiedonhallintalain 5§:n mukaan tietohallintamallin tulee sisältää tietoja ainakin toimintaprosesseista, tietovarannoista, tietoineistojen arkistoinnista, tietojärjestelmistä sekä tietoturvaluottamustoimenpiteistä. Nämä luetellut osat kuuluvat kokonaisarkkitehtuuriin ja tietohallintamallissa ne on eroteltu omiksi kokonaisuuksikseen lain vaatimusten mukaisesti. Tiedonhallintayksiköissä kokonaisarkkitehtuurityötä on jo pidemmän aikaa tehty, koska Laki julkisen hallinnon tietohallinnon ohjauksesta (634/2011) on jo edellyttänyt julkisen hallinnon toimijoiden kuvattavan kokonaisarkkitehtuuria ja noudatettavan laadittua julkisen hallinnon kokonaisarkkitehtuuria (Finlex, 2011).

Valtiovarainministeriön alaisuudessa toimivan tiedonhallintalautakunnan tehtävänä on edistää ja valvoa tiedonhallintalaissa säädettyjä menettelytapoja, vaatimusten toteutusta ja arvioida tiedonhallintolain noudattamista (Valtiovarainministeriö, 2021a). Tiedonhallintalautakunnan asettamat jaostot työstävät jatkossa uusia suosituksia tiedonhallintolain mukaisten vaatimusten perusteella. Suositukset ohjaavat tiedonhallintayksiköjä toteuttamaan tiedonhallintalain vaatimuksia ja hyviin käytäntöihin, eivätkä ole velvoittavia tai sitovia (Valtiovarainministeriö, 2021b).

## 2.3 Arkkitehtuurin dokumentointi

Arkkitehtuurin ja tietojärjestelmien suunnittelussa ja kuvauksien tuottamisessa tietoa luodaan paljon. Pyle (2020) toteaa, että dokumentaatiolla on merkittävä rooli ohjelmistojen kehitysympäristössä ja dokumentaation voi laajasti kattaa sisältävän kaikkea kirjoitettua tai kuvitettua viestintää, jossa kuvataan, miten ohjelmisto toimii, mitä se tekee ja miten sitä käytetään. Dokumentaatio voi sisältää arkkitehtonisia asiakirjoja, vaatimusmäärittelyä ja yksityiskohtaisia suunnitteluasiakirjoja (Pyle, 2020). Koskimies ja Mikkonen (2005) tarkentavat, että dokumentin muoto ja sisältö vaihtelee organisaation toimintatavoista sekä järjestelmän luonteesta ja laajuudesta riippuen. Hyvän dokumentaation periaatteita on Clements ym. (2003) mukaan:

1. Kirjoita lukijan näkökulmasta

2. Vältä toistoa
3. Vältä epäselvyyttä
4. Käytä organisaation standardia
5. Kirjoita perustelut
6. Pidä dokumentaatio riittävän ajan tasalla
7. Tarkista dokumentaation tarkoitus

Pyle (2020) mainitsee, että suurin ongelma, joka vaikuttaa ohjelmistojen ylläpitoon on ajantasaisen dokumentaation puute. Tieto voi sen vuoksi olla hyvin sekavaa ja oleellista tietoa voi jäädä dokumentoimatta. Koskimies ja Mikkonen (2005) painottavat, että on kyseessä sitten pieni tai laaja ohjelmistoprojekti, on tärkeää, että arkkitehtuurikuvaukset ovat selkeä osa dokumentaatiota ja helposti löydettävissä. On siis huomioitava, että kaikki oleellinen arkkitehtuuriin liittyvä tieto on dokumentoitu ja siitä on tehty riittävällä tasolla olevat dokumentaatiot.

Koskimies ja Mikkonen (2005) luettelevat arkkitehtuuridokumenttityyppejä olevan alustava-, järjestelmä-, alijärjestelmä-, tuoterunko ja tuotearkkitehtuuridokumentti sekä rajapintadokumentti. Alustava arkkitehtuuridokumentti kuvaa tärkeimpiä arkkitehtuuriratkaisuja. Dokumenttia käytetään esimerkiksi liiketoimintapäätöksien ja projektisuunnitelmien pohjana, jossa kuvataan konkreettinen arkkitehtuuri. Järjestelmäarkkitehtuuridokumentti kuvaa järjestelmän arkkitehtuurin ylempää tasoa eli järjestelmän sidoksia muihin järjestelmiin ja niiden vuorovaikutusta sekä on pohjana alijärjestelmien arkkitehtuurisuunnitteluun. Alijärjestelmäarkkitehtuuridokumenttia ei välttämättä käytetä pienissä ohjelmistoprojekteissa, koska niissä komponenttitason kuvauksia voidaan kuvata jo järjestelmäarkkitehtuuridokumentissa. Laajoissa järjestelmissä sen sijaan alijärjestelmäarkkitehtuuridokumentissa kuvataan alijärjestelmän arkkitehtuuri järjestelmäarkkitehtuurin reunaehtojen mukaisesti. Dokumentti kuvaa komponenttien ulkoisia rajapintoja ja niiden vuorovaikutusta. Dokumentti tukee myös toteutusta yksikkötestauksen suunnittelussa. Tuoterunkoarkkitehtuuridokumentti kuvaa sovelluskehittäjälle miten tuoterunkoarkkitehtuurin varianssia käytetään sovelluksen rakentamisessa. Tuotearkkitehtuuridokumentti on erillinen arkkitehtuuridokumentti tuoterungon pohjalle toteutetulle ohjelmistolle. Se kuvaa miten tuoterunkoarkkitehtuuria on sovellettu ja mitä arkkitehtuuriratkaisuja on tehty. Rajapintadokumentti täydentää kaikkia aikaisemmin mainittuja dokumentteja kuvaamalla esimerkiksi järjestelmän tarjoaman ohjelmointirajapinnan (engl. application programming interface, API) (Koskimies & Mikkonen, 2005).

Arkkitehtuuridokumentin rakenne voi vaihdella organisaatiosta riippuen. Yleensä rakenteeseen liittyvät Koskimiehen ja Mikkosen (2005) mukaan seuraavat kohdat:

- identifiointi
- konteksti
- vaatimukset
- ympäristö
- näkymät

- arkkitehtuuriviipaleet
- analyysi
- perustelut (Koskimies & Mikkonen, 2005).

Arkkitehtuurin dokumentoinnissa on tärkeää, että tietojärjestelmäkehitykseen liittyvää olennaista tietoa saadaan tuotettua kirjoitettuun muotoon riittävällä laajuudella. Liang ja Avgeriou (2009) mainitsevat, että tiedonhallinnassa tehdään usein ero kahden erityyppisen tiedon välillä, joita on implisiittinen ja eksplisiittinen tieto. Implisiittinen tieto, jota kutsutaan myös hiljaiseksi tiedoksi (engl. tacit knowledge) on ihmisen päässä olevaa tietoa. Ekplisiittinen tieto on dokumentoitua ja muodollista tietoa. Kruchten, Lago ja van Vliet (2006) argumentoivat, että suurin osa arkkitehtonisesta tiedosta on yleensä hiljaista tietoa. Hiljaista tietoa eli henkilön päässä olevaa, aikojen saatossa kerättyä tietoa, saattaa jäädä dokumentoimatta sen vuoksi, että tiedon ei oletettu olevat relevanttia dokumentaatiota luonti hetkellä tai keskusteluissa. Ilman riittävää dokumentaatiota tieto pysyy hiljaisena ja on turvauduttava muihin keinoihin hankkia puuttuva tieto (Dagenais ym., 2010). Hiljaisen tiedon vastakohta on avoin tieto (engl. explicit knowledge), joka tarkoittaa sitä tietoa, joka on tuotettu kirjoitettuun muotoon dokumentoinnin yhteydessä tai keskusteluissa. Arkkitehtuurin suunnittelun alkaessa on tärkeää, että kaikki mahdollinen tärkeä tieto saadaan dokumentoitua, jotta kehitystiimi saa tarvittavan materiaalin kehittämisen tueksi.

Kruchtenin, Lagon ja van Vlietin (2006) mukaan myöhemmässä vaiheessa jo tehtyjen ratkaisujen jäljittäminen on vaikeaa ja erityisesti kehityksen aikana hiljainen tieto eli dokumentaation puute voi koitua erittäin kalliiksi tai jopa mahdottomaksi. He ovat jaotelleet neljä (4) erityyppistä suunnittelupäätöstä ja niiden perustelut:

- Implisiittinen ja dokumentoimaton kuvaa sitä, että arkkitehti ei ole tietoinen päätöksestä tai päätös on ilmiselvä asia tai päätös, ns. ”kaikkihan sen tietää”, henkilön kokemukseen perustuen.
- Selkeä mutta dokumentoimaton tarkoittaa sitä, että arkkitehti tekee päätöksen mutta päätöstä ei ole dokumentoitu ja sen vuoksi tietoa ajaa saatossa.
- Explisiittinen ja tietoisesti dokumentoimaton tarkoittaa sitä, että päätöstä ei tuoda julki, jonka syitä voi olla organisaation sisäiset tai arkkitehdin henkilökohtaiset syyt esimerkiksi suojella asemaansa.
- Selkeä ja dokumentoitu, joka on suositelluin tapa toimia mutta todennäköisesti poikkeuksellista (Kruchten, Lago & van Vliet, 2006).

Pyle (2020) argumentoi, että silloin kun tekniset asiantuntijat kirjoittavat itse dokumentaatio, voi olla, että he eivät sen vuoksi näe aihetta laajemmasta perspektiivistä ja asioita voi jäädä dokumentoimatta tai he eivät kirjoita aiheesta kohderyhmälle parhaalla mahdollisella tavalla.

### 2.3.1 Dokumentaatio ketterässä kehityksessä

Ketterä kehitys ja sen käytännöt ovat luoneet muutoksia dokumentaation tekemiseen. Ketterän kehityksen julistuksessa, Agile Manifestossa (Agile Alliance, 2001) mainitaan seuraavasti: "Working software over comprehensive documentation". Julistus arvostaa "Mieluummin toimivaa ohjelmistoa kuin kattavaa dokumentaatiota", tämä julistus saattaa aiheuttaa kuitenkin väärinkäsityksiä. Julistuksen tarkoituksena ei ole, etteikö dokumentaatiota tehdä lainkaan vaan tarkoituksena on vähentää turhaa dokumentaatiota. Abrahamsson, Babar ja Kruchten (2010) mukaan, että ketterän kehityksen lähestymistavan pyrkimyksenä on tuottaa arvoa varhaisessa vaiheessa ja usein. Tästä syystä dokumentaation aikaa vievän olemuksen vuoksi se ei sovi yhteen ketterän kehityksen kanssa. Cakir, Cetin, Savasci ja Findik (2018) mainitsevat myös, että dokumentaatiota ei ketterässä kehityksessä välttämättä pidetä lainkaan tarpeellisena tai arvokkaana mutta järjestelmäkehityksen monimutkaisuus puolestaan tuo dokumentaatiolle oman paikkansa. Cakir, Cetin, Savasci ja Findik (2018) argumentoivat, että yleisesti käytössä oleva perinteinen arkkitehtuurin dokumentaatio on kattavaa ja yksityiskohtaista, jonka vuoksi dokumentaation ylläpito vie ketteriltä tiimeiltä liikaa aikaa. Dokumentaation pitäisi olla yksinkertaista ja ottaa huomioon ketterät periaatteet, joten perinteinen arkkitehtoninen malli ei palvele tällä hetkellä tarkoitustaan. Kuitenkin on tärkeää, että arkkitehtuuriin liittyvät asiat on dokumentoitava yleisen ymmärryksen vuoksi. Cakir, Cetin, Savasci ja Findik (2018) esittelevät artikkelissaan uuden kevyemmän arkkitehtonisen mallin, jolla ketterät tiimit pystyisivät tuottamaan arkkitehtuurin dokumentaatiota kattavammin. He toteavat myös, että vaikka ketterän kehityksen filosofia neuvoa vähentämään dokumentaatiota, yksinkertaisen ja kevyen ohjelmistoarkkitehtuurin dokumentaatio luo suuria etuja toimivalle tuotteelle (Cakir ym, 2018). Pyle (2020) toisaalta mainitsee ketterän kehityksen edistävän dokumentaation tekemistä koko kehityksen aikana ja kattavat dokumentaatiot auttaisivat selviytymään muutoksista nopeasti. Ketterässä kehityksessä kun dokumentaation työstäminen on mukana jokaisessa iteraatiossa (Pyle, 2020). Ketterässä kehityksessä, julistuksesta huolimatta, kannattaa dokumentaatiolle antaa arvoa. Tämä helpottaa esimerkiksi ylläpidollista työtä, kun ohjelmistosta on tarpeeksi riittävä dokumentaatio olemassa.

### 2.3.2 Dokumentaatio kommunikaation välineenä

Arkkitehtuurityö on kommunikointia ja viestintää arkkitehdin ja sidosryhmien välillä. Kommunikaatio tapahtuu usein esityksissä ja työpajoissa, joissa arkkitehtuuria ja siihen liittyviä päätöksiä esitellään. Schwittek & Eicker (2010) mainitsevat, että myös arkkitehtuurin dokumentaatio nähdään arkkitehtonisen tiedon kommunikaatiotapana. Lankhorst (2017) toteaa myös, että arkkitehtuurimallit, näkymät, esitykset sekä analyysit auttavat kuromaans umpeen ns. "viestintäkuilua arkkitehtien ja sidosryhmien välillä.

Dokumentaatio on tärkeä osa kommunikaatiota ja vuorovaikutusta. Tietojärjestelmäkehityksessä kommunikaatiolla on merkityksensä esimerkiksi

sidosryhmien välillä tapahtuvassa tiedonkulussa, jotta kaikki sidosryhmät ymmärtävät dokumentaatiossa olevan informaation. Pyle (2020) toteaa, että tietojärjestelmien kehittämiseen osallistuvat henkilöt mukaan lukien organisaationjohtajat, projektipäälliköt ja kehittäjät lukevat dokumentaatiota, joten tehokas dokumentaatio tukee kehitystä, pitää laadun korkeana ja mahdollistaa myös projektien helpomman siirtämisen tiimistä toiseen. Erilaisilla toiminta-alueilla ja taustoilla olevat sidosryhmät, esimerkiksi liiketoiminta ja tietojärjestelmäkehitys, on erilaiset ajatusmaailmat ja puhuvat eri kieltä, joka vaikeuttaa viestintää entisestään. Kommunikaatio vaatii hyviä viestintätaitoja, jotta voi saavuttaa yhteisen ymmärryksen arkkitehtuurista kaikkien sidosryhmien välillä (Schwittek & Eicker, 2010).

Muuttuvat tietojärjestelmät ja tästä syntyvät tietojärjestelmävaatimukset tarvitsevat päivittämistä. Sidosryhmillä voi olla erilaisia tarpeita, joihin dokumentaation on vastattava ja kommunikoitava riittävällä laajuudella kyseistä kohderyhmää ajatellen. Kommunikaatio ja koordinointi ovat tärkeitä menestystekijöitä näissä muutoksista (Abraham, 2013). Lankhorst (2017) toteaa, että viestinnän helpottamiseksi arkkitehtuuri tulisi olla mahdollista visualisoida mahdollisimman tarkasti tiettyä sidosryhmää ajatellen. Erityisen tärkeää on saada aikaan onnistunutta viestintää sidosryhmille eri arkkitehtuurien kuvaamisen välillä (Lankhorst, 2017).

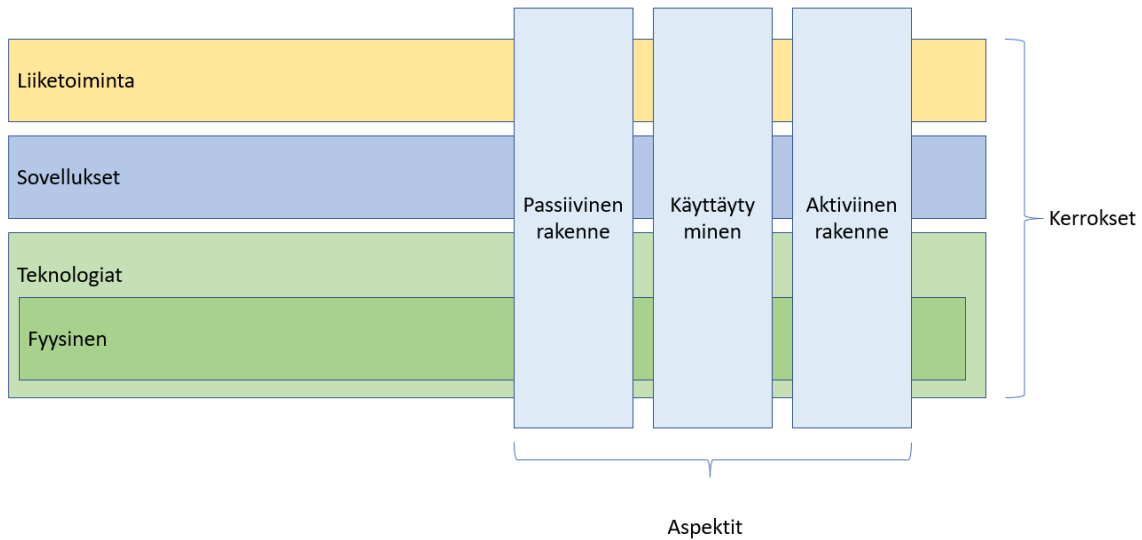
Dokumentaatio edesauttaa myös uusien työntekijöiden perehtymistä ja tutustumista uuteen toimintaympäristöön. Perehtymisvaiheessa erityyppiset dokumentaatiot ovat työntekijän näkökulmasta oleellista, jotta hän saa muodostettua tarvittavan yleiskuvan järjestelmästä, jonka parissa tulee työskentelemään. Ilman tarvittavaa ymmärrystä järjestelmästä, voi integroituminen uuteen toimintaympäristöön olla vaikeampaa (Dagenais ym., 2010). Kuten olemme todenneet, dokumentaation tarkoitus on moninainen. Lankhorst (2017) mainitsee, että loppujen lopuksi dokumentaation tai mallin on tarkoitus toimia ymmärtämisen apuna ja kommunikaation välineenä.

Tietojärjestelmäkehityksessä erilaisten mallinnus- ja piirtotyökalujen käytöllä on merkitystä, jotta arkkitehtuurin mallintaminen, dokumentointi ja kommunikointi on mahdollisimman selkeää ja ymmärrettävää niin organisaation sisäisten kuin ulkoisten sidosryhmien välillä. Tietojärjestelmäkehityksessä kehityksen kohteena olevaa järjestelmää kuvataan mallien avulla. Arkkitehtuuria voidaan mallintaa mallinnuskielillä, joita ovat yleisesti käytössä olevat formaalit kielet ArchiMate ja BPMN (Business Process Model and Notation) ja puoliformaali kieli UML (The Unified Modeling Language).

### 2.3.3 ArchiMate

ArchiMate on laajasti käytetty ja monipuolinen mallinnuskieli, jolla kuvataan liiketoimintakerrosta eli ylätasoa kuvauksia. ArchiMate viitekehityksen ydin (engl. core) rakentuu kolmesta eri aspektista ja kolmesta eri kerroksesta, jotka ovat havainnollistettu alla olevassa kuviossa 4 (OMG, 2012–2019). Horisontaalisia kerroksia ovat liiketoiminta, sovellukset ja teknologiat. Teknologiat kerrokseen

sisältyy myös fyysinen kerros. Vertikaalisesti esitettävät aspektit ovat passiivinen rakenne, käyttäytyminen eli toiminta ja aktiivinen rakenne (OMG, 2012–2019). Hosiaislouma (2021) selventää, että ArchiMaten avulla pystyy erottamaan toiminnan ja rakenteen, joka helpottaa organisaation kuvaamista oleellisesti. Viitekehyksen rakenne näin ollen mahdollistaa organisaation kuvaamisen monesta eri näkökulmasta.



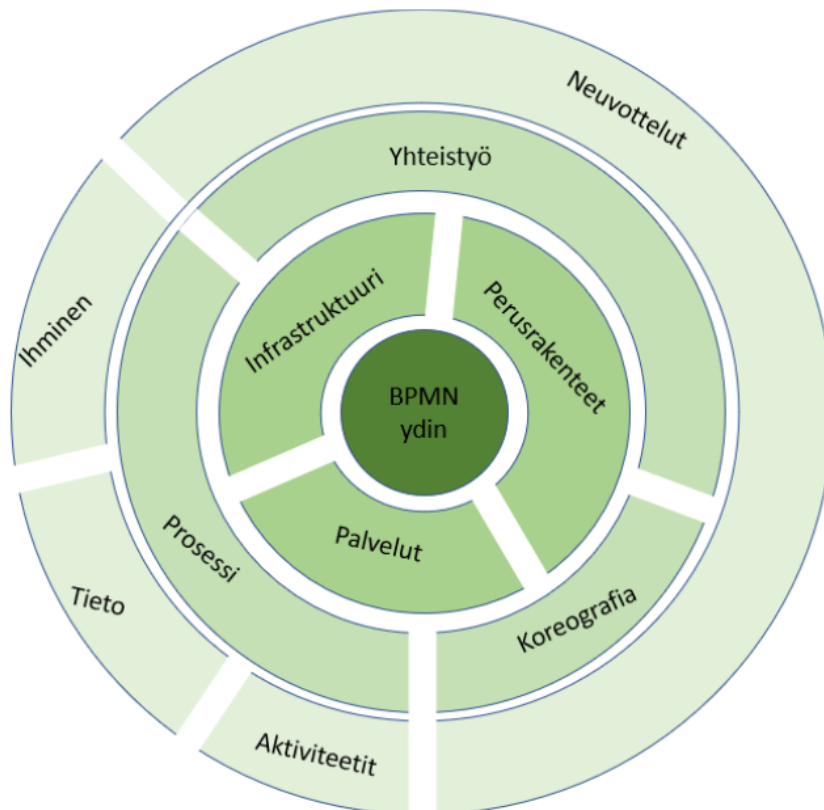
KUVIO 4 ArchiMate viitekehyksen ydin (OMG, 2012-2019, 3.4 mukaan)

ArchiMaten peruselementit ovat aiemmin mainituista kerroksista ja aspekteista muodostuva elementtijoukko. Elementtejä on yhteensä noin 60 kappaletta mutta peruselementit, joita on 15 kappaletta, ovat tarpeeksi riittäviä useampiin mallinnus tarpeisiin. Liiketoimintaan liittyviä peruselementtejä on rajapinta (Business Interface), toimija (Business Actor), liiketoiminnan palvelu (Business Service), liiketoiminnan prosessi (Business Process) ja käsite (engl. Business Object). Sovelluskerroksen peruselementtejä on sovellusrajapinta (Application Interface), sovelluspalvelu (Application Service), sovellus (Application Component), sovellusprosessi (Application Process) sekä tieto (Data Object). Teknologiakerroksen peruselementeiksi mainitaan teknologiarajapinta (Technology Interface), teknologiapalvelu (Technology Service), solmu (Node), teknologiaprosessi (Technology Process) sekä Artefakti (Artifact) (Hosiaislouma, 2021). ArchiMatella voi tehdä erilaisia kaaviotyyppisiä, kuten tavoitenäkymiä (Vision View Diagram), kerroksnäkyymiä (Layered View Diagram), integraationäkymiä (Integration View Diagram), sekä käsitelmalleja, sijoittelukaavioita (Deployment Diagram). Sijoittelukaavioilla kuvataan järjestelmän ohjelmistojen ja palvelinten välisiä suhteita (Hosiaislouma, 2017).

### 2.3.4 BPMN

BPMN (Business Process Model and Notation) mallinnuskieli tarjoaa mahdollisuuden tehdä graafisia kuvauksia liiketoimintaprosessien kuvaamiseen. BPMN

mallinnustyökalua kutsutaan organisaatioiden prosessikuvauksien defacto standardiksi eli on hyvin yleisesti käytetty työkalu. Työkalun tavoitteena on tukea vakiomuotoisia ja ymmärrettäviä kuvauksia mutta myös monimutkaisia prosessikuvauksia teknisille käyttäjille. Työkalulla voi tehdä teknisiä prosessikuvauksia, uimaratoja ja yksinkertaisia liiketoimintaprosessikuvauksia. BPMN-kaaviot voi kääntää suoraan ohjelmiston komponenteiksi (OMG, 2013). BPMN perustuu siihen, että yksinkertaisen peruselementin päälle voi laajentaa monta eri kerrosta, kuten kuvio 5 tätä havainnollistaa.



KUVIO 5 BPMN ydin ja kerros rakenne (OMG, 2013, 8.1 mukaan)

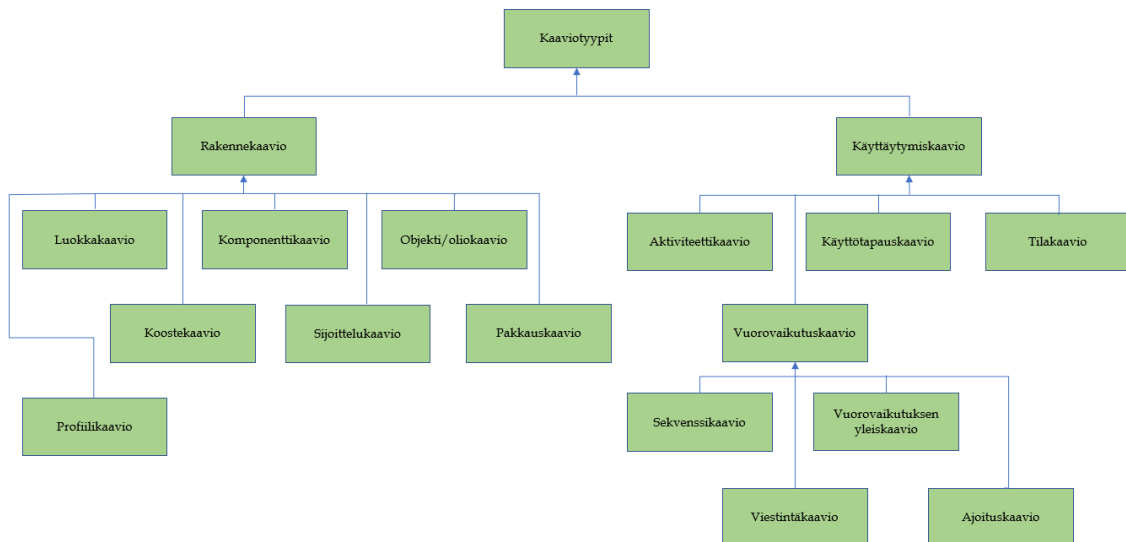
BPMN:n ydinelementtejä ovat infrastrukturi (engl. Infrastructure), perusrakenteet (engl. Common Elements) ja palvelut (engl. Services). Infrastrukturi sisältää kaksi elementtiä, joita käytetään abstrakteissa syntaksimalleissa ja kaaviomalleissa. Perusrakenteet sisältävät rakenteita BPMN mallintamiselle. Palvelut sen sijaan sisältää palveluiden, rajapintojen ja toimintojen mallintamiseen tarvittavat rakenteet. (OMG, 2013).

### 2.3.5 UML

UML (The Unified Modeling Language) on graafinen mallinnuskieli ja tällä hetkellä UML:n uusin versio on 2.5.1. (OMG, 2017). UML:n tarjoaa tietojärjestelmäarkkitehdeille ja ohjelmistokehittäjille työkaluja, joilla pystyy analysoimaan, määrittelemään, visualisoimaan malleja ja kuvauksia tietojärjestelmän



suunnittelua ja toteutusta varten (OMG, 2017). UML sisältää 14 erilaista kaaviotyyppiä. Nämä kaaviotyypit on jaettu kolmeen (3) eri kategoriaan, joista rakenne- ja käyttäytymiskaaviot ovat pääkaavioita ja vuorovaikutuskaaviot sisältyvät käyttäytymiskaavioiden kategoriaan. Kaaviotyypit esitetään kuviossa 6. Rakennekaavioita on seitsemän (7), käyttäytymiskaavioita (3) ja vuorovaikutuskaavioita neljä (4). Rakennekaavioita ovat luokkakaavio (Class Diagram), objekti/oliokaavio (Object Diagram), komponenttikaavio (Component Diagram), koostekaavio (Composite Structure Diagram), pakkauskaavio (Package Diagram), käyttöönotto/sijoittelukaavio (Deployment Diagram) ja profiilikaavio (Profile Diagram). Käyttäytymiskaavioita ovat käyttötapauskaavio (Use Case Diagram), aktiviteettikaavio (Activity Diagram) ja tilakaavio (State Machine Diagram). Vuorovaikutuskaaviot ovat sekvenssikaavio (Sequence Diagram), viestintäkaavio (Communication Diagram), ajoituskaavio (Timing Diagram) ja vuorovaikutuksen yleiskaavio (Interaction Overview Diagram) (OMG, 2017).



KUVIO 6 UML kaaviotyypit (OMG, 2017, s. 685 mukaan)

Rakennekaaviot kuvaavat tietojärjestelmän staattista rakennetta eli ajasta riippumattomia elementtejä. Käyttäytymis- ja vuorovaikutuskaaviot kuvaavat tietojärjestelmän dynaamisen eli ajasta riippuvan rakenteen. Kaavioiden avulla voi kuvata menetelmiä, yhteistyötä ja toimintaa (OMG, 2017).

### 2.3.6 Yhteenveto

BPMN ja UML ovat OMG:n kehittämiä mallinnustyökaluja, jotka täydentävät toinen toistaan. BPMN keskittyy määrittelemään enemmän liiketoimintaprosessien suunnittelua ja analysointia, kun taas UML soveltuu parhaiten tietojärjestelmän suunnitteluun ja analysointiin. Vaikka UML on suuntautunut enemmän tietojärjestelmien suunnitteluun, sillä voi soveltaen mallintaa myös liiketoimintaprosesseja (OMG, 2017).

ArchiMaten ja UML:n eroavaisuudet ovat siinä, että ArchiMate sisältää kattavat ja monipuoliset standardinotaatiot arkkitehtuurikuvausten mallintamiselle ja UML on erinomainen työnkulkujen kuvaamiseen. Näiden mallinnuskielien avulla tietyn tietojärjestelmän arkkitehtuuri voidaan mallintaa, jolloin siitä on hyötyä koko järjestelmäkehitykselle.

Mallinnus- ja piirtotyökaluilla tuotetut arkkitehtuurimallit ja -kuvaukset mahdollistavat arkkitehtuurin suunnittelua, tietojärjestelmän onnistumisen ja ne toimivat erinomaisesti ymmärryksen ja tiedonkulun välineenä sidosryhmien välillä. Tuotetut kuvaukset toimivat myös tietojärjestelmän määrittelyn, toteutuksen ja testauksen tukena.

### 3 ARKKITEHTUURIKUVAUSTEN DOKUMENTOINTI KANTA-PALVELUISSA

Tässä luvussa perehdytään Kanta-palveluihin lyhyesti, sivutaan hieman Sote-arkkitehtuuria ja kerrotaan tarkemmin Kanta-palveluiden arkkitehtuurista. Käydään läpi myös sitä, miten tällä hetkellä Kanta-palveluissa toteutetaan arkkitehtuurin dokumentointia. Tämä luku toimii pohjustuksena tutkimuksen empiiriselle osuudelle, jossa tutkitaan arkkitehtuurin dokumentointia kehittäjän ja ylläpidon näkökulmasta Kanta-palveluissa.

#### 3.1 Kanta-palvelut lyhyesti

Kanta-palvelut on otettu käyttöön vuodesta 2010 lähtien. Käyttöönotto on tapahtunut vaiheittain niin, että esimerkiksi reseptejä on aloitettu määräämään sähköisesti vuoden 2017 alusta lähtien. Lainsäädännön kehittymisen ja asiakkaiden tarpeiden myötä Kanta-palvelut kehittyvät monipuoliseksi palveluksi. Kanta-palveluita käyttävät kaikki Suomen kansalaiset, terveydenhuolto, sosiaalihuolto ja apteekit. Kanta-palveluihin kuuluu Omakanta, Resepti-palvelu, Lääketietokanta, Potilastiedon arkisto, vanhojen potilastietojen arkistointi, Sosiaalihuollon asiakastiedon arkisto, terveydenhuollon todistusten välitys, Kelain ja Kanta-asiakastestipalvelu (Kanta, 2021a).

Kanta-palveluiden tulee noudattaa lainsäädäntöä:

- Laki sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä (Asiakastietolaki)
- Laki sähköisestä lääkemääräyksestä (Reseptilaki) (Kanta, 2021b).

Kanta-palvelut tehdään yhteistyössä Sosiaali- ja terveysministeriön (STM), Kelan, Terveyden ja hyvinvointilaitoksen (THL), Sosiaali- ja terveystieteiden tutkimuskeskuksen (STM), Digitaalisen ja väestötietoviraston (DVV, ent. VRK) sekä apteekkien, sote-toimijoiden ja järjestelmätoimittajien kanssa. STM vastaa Kanta-palveluiden koordinoinnista ja lainsäädännöstä, Kela vastaa Omakannan rakentamisesta ja ylläpidosta sekä Resepti-palvelusta, Lääketietokannan ja Potilastiedon arkiston ylläpidosta (Kanta, 2021c). Seuraavaksi kerrotaan aiemmin mainituista Kanta-palveluihin kuuluvista palveluista tarkemmin. Alla olevat palvelut ovat sellaisia palveluita, joista tähän tutkimukseen on haastateltu asiantuntijoita. Aivan jokaisesta Kanta-palveluihin kuuluvasta palvelusta ei tämän vuoksi ole kerrottu.

### 3.1.1 Omakanta-palvelu

Omakanta-palvelu on kansalaisen käyttöliittymä, jossa kansalainen näkee omat terveys- ja hyvinvointitiedot, reseptit, EU:n koronatodistuksen sekä voi nähdä mitkä apteekit tai terveydenhuollon yksiköt ovat tietoja käsitelleet Kanta-palveluiden kautta. Omakannassa voi pyytää reseptin uusimista, tallentaa hoito- ja elinluovutustahdon sekä antaa suostumuksen tai kiellon omien tietojen luovuttamiseen (Kanta, 2021d).

Kansalaisella tulee olla suomalainen henkilötunnus ja tunnistautumisvälineet, joilla Omakantaan voi kirjautua. Tunnistautumisvälineitä ovat verkkopankkitunnukset, mobiilivarmenne ja sähköinen henkilökortti. Omakantaan voi tunnistautua myös Digi- ja väestötietoviraston (DVV) myöntämällä organisaatiokortilla tai sosiaali- ja terveydenhuollon ammattikortilla (Kanta, 2021d).

Omakanta on valtakunnallinen palvelu, jossa näkyy julkisen, yksityisen ja työterveyshuollon potilas- ja reseptitiedot (Kanta, 2021d). Kaikki julkisen terveydenhuollon palvelunantajat ja suurin osa yksityisen terveydenhuollon palvelunantajista ovat jo liittyneet Kanta-palveluihin (Kanta, 2021g). Omakanta on kansalaisen ikkuna kaikkiin hänestä terveydenhuollossa kirjattuihin tietoihin. Näin ollen terveydenhuolto vastaa tietojen kirjaamisesta sekä niiden korjaamisesta (Kanta, 2021d).

### 3.1.2 Omatietovaranto

Omakanta-palveluun kansalainen voi Omatietovarannon kautta tallentaa omia hyvinvointitietojaan, esimerkiksi verensokerin ja painon tai Omaolon-palvelun kautta tehdyn oirearviokyselyn. Hyvinvointitietoja voi tallentaa oman laitteen kuten älypuhelimien, internetissä toimivan sovelluksen tai muun internetissä toimivan ohjelmiston avulla. Tallennettuja tietoja voi katsella ja poistaa Omakannassa. Omakannassa kansalainen näkee myös mille kaikille sovelluksille hän on antanut käyttöoikeudet. Näitä käyttöoikeuksia voi myös muokata ja poistaa niin halutessaan.

Hyvinvointitiedot eivät tällä hetkellä ole terveydenhuollon tavoiteltavissa, vaan ainoastaan Omakannan käyttäjät eli kansalaiset voivat nähdä ne itse. Tulevaisuudessa lainsäädäntö tulee muuttumaan asiakastietolain myötä ja näin ollen mahdollistaa tietojen luovuttamisen sosiaali- ja terveydenhuollon ammattilaisille (Kanta, 2021e).

### 3.1.3 Reseptikeskus ja lääketietokanta

Kaikki reseptit tallennetaan keskitetysti Kanta-palvelujen tietokantaan eli Reseptikeskukseen. Kansalainen voi tarkastella omia reseptejään Omakanta-palvelun kautta. Omakannasta näkee mitä reseptejä on määrätty, mihin asti resepti on voimassa ja paljonko reseptissä on lääkettä jäljellä. Omakannasta voi myös pyytää reseptin uusimisen. Apteekit tarkistavat vastaanotolla kansalaiselle määrättyt reseptit reseptikeskuksesta, jonka jälkeen kansalainen saa lääkkeensä. Myös

ulkomailta Suomen kansalaisen on mahdollista ostaa lääkkeitä, kun kansalainen on antanut Omakanta-palvelussa suostumuksen toimittaa reseptitietoja toisen Euroopan maan apteekissa (Kanta, 2021f).

Kaikilla Suomen terveydenhuollon yksiköillä ja apteekeilla on mahdollisuus käyttää Lääketietokantaan perustuvia tietoja. Näin ollen he saavat Lääketietokannasta lääkkeen määräämisen ja toimittamisen kannalta olennaiset tiedot lääkkeestä ja esimerkiksi keskenään vaihtokelpoisista lääkevalmisteista ja korvattavista perusvoiteista (Kanta, 2021f).

### **3.1.4 Potilastiedon arkisto**

Kanta-palvelujen Potilastiedon arkisto on palvelu, johon kaikki julkisen terveydenhuollon palvelunantajat ovat jo liittyneet, osa valtiollisista organisaatioista sekä yli 1900 (1926 kpl 29.4.2022 tilanne) yksityisen terveydenhuollon palvelunantajaa. Potilastiedon arkistoon arkistoituvat hoidon yhteydessä syntyneet sähköiset potilastiedot. Nämä potilastiedot näkyvät kansalaiselle Omakanta-palvelun kautta (Kanta, 2021g). Tarkemmin kerrottaen, kun kansalainen on asioinut terveydenhuollon yksikössä, kyseinen terveydenhuollon yksikkö kirjaa potilastietoja tietojärjestelmäänsä ja tämän jälkeen ne arkistoituvat Potilastiedon arkistoon. Potilastiedon arkisto myös välittää terveystietoja eri terveydenhuollon toimijoiden välillä, jos kansalainen on antanut siihen luovutusluvan (Kanta, 2021h), joko terveydenhuollon yksikössä tai Omakanta-palvelussa.

### **3.1.5 Kuva-aineistojen arkisto**

Kuva-aineistojen arkisto on Kanta-palveluihin kuuluva palvelu, joka koostaa ja arkistoi EKG- ja kuvantamistutkimuksista syntyneet kuvamateriaalit ja kuvantamisen asiakirjat (Kanta, 2022a). Kuva-aineistojen arkisto on yksi Potilastiedon arkiston lisäpalvelu, jonka palvelunantaja voi laajentaa käyttöönsä (Kanta, 2022b). Terveydenhuollon ammattilainen voi hakea EKG- ja kuvantamistutkimustietoja potilastietojärjestelmän kautta, kansalainen ei voi nähdä kuvamateriaalia Omakannassa (Kanta, 2022a).

### **3.1.6 Sosiaalihuollon asiakastiedon arkisto**

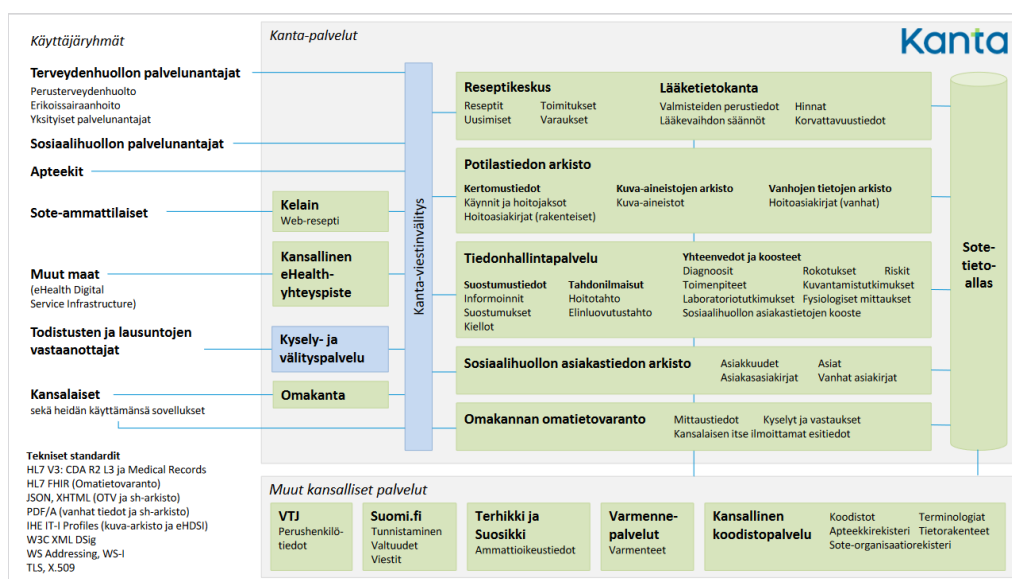
Sosiaalihuollon asiakastiedon arkisto toimii samalla periaatteella kuin Potilastiedon arkisto. Sosiaalihuollon asiakastiedon arkistoa käyttävä organisaatio (palvelunantaja) arkistoi potilastietoja Sosiaalihuollon arkistoon, josta ne tulevaisuudessa näkyvät kansalaiselle Omakanta-palvelun kautta. Vielä tätä ominaisuutta ei ole Omakannassa tarjolla. Sosiaalihuollon asiakastiedon arkistoa käyttää hieman yli 100 (102kpl 29.4.2022 tilanne) sosiaalihuollon organisaatiota (Kanta, 2021i).

## 3.2 Kanta-arkkitehtuuri

Sosiaali- ja terveydenhuollon kansallista kokonaisarkkitehtuuria kutsutaan nimellä Sote KA. Arkkitehtuurityön kehittämistä johtaa sosiaali- ja terveysministeriö. Työhön osallistuvat sosiaali- ja terveydenhuollon kehittämisen viranomais-toimijat, tiedonhallinnon palvelujen tuottajia sekä sosiaali- ja terveydenhuollon edustajia. Työssä laaditaan yhteisiä periaatteita, linjauksia ja tarkempia arkkitehtuurikuvauksia teemoista: asiakas- ja potilastietojen käsittely, asiointi ja omahoito, toisiokäyttö ja yksilöllistetty lääkehoito. Erityisesti asiakas- ja potilastietojen käsittelyn arkkitehtuurista vastaa Terveyden ja hyvinvoinnin laitos (THL) (Terveyden ja hyvinvoinnin laitos, 2022).

Sosiaali- ja terveydenhuollon tietojärjestelmäpalvelut tehdään ja otetaan käyttöön vaiheittain. Tällä vaiheistuksella kokonaisarkkitehtuurin tavoitteet tulee saavutettua. (Terveyden ja hyvinvoinnin laitos, 2022). Luvuissa 2.2.3 on tarkemmin kerrottu JHS-179 suosituksista ja luvussa 2.2.4 Tiedonhallintalaista, jotka ohjaavat Sosiaali- ja terveydenhuollon kokonaisarkkitehtuurityötä.

Kanta-palvelut ovat keskeisessä osassa sosiaali- ja terveydenhuollon kokonaisarkkitehtuuria. Sitä kehitetään yhteistyössä kaikkien sosiaali- ja terveydenhuollon toimijoiden kanssa. Kanta-arkkitehtuuri sen sijaan on kansallisten terveydenhuollon sähköisten palveluiden kokonaisuus ja sen hallintamalli (Kanta, 2021j). Kanta-arkkitehtuurin viitekehyksestä Kanta-palveluihin liittyvät järjestelmät löytävät teknisiä määrittämiä tietosisällöllisen yhteensopivuuden, tietoturvallisen tiedonsiirron ja avointen rajapintojen toteuttamiselle. Tietosisällöllinen määrittäminen voi olla esimerkiksi 'Alaikäisen tietojen luovuttaminen huoltajille terveydenhuollossa - toiminnallinen määrittely' ja sanomaliikenteen kuvauksena esimerkiksi 'Osapuolitiedot Kanta-sanomaliikenteessä' (Kanta, 2021j). Alla olevassa kuviossa 7 on kuvattu Kanta-arkkitehtuuri, jossa on esitelty kaikki Kanta-palvelut, niiden käyttäjäryhmät, tekniset standardit ja muut kansalliset palvelut, jotka ovat yhteydessä Kanta-palveluihin.



Kuten aikaisemmin jo mainittiin, Kanta-palvelut tehdään yhteistyöllä. Kanta-palvelujen verkkosivut tarjoavat ulkoisille järjestelmäkehittäjille ajankohtaista tietoa kehittämisen ja yhteistestauksen julkaisuaikatauluista, sertifiointin prosessista, vaatimuksista ja tietoturvasta sekä tarjoaa tietojärjestelmätoimittajille tukea (Kanta, 2021k).

Kanta-palveluissa noudatetaan kokonaisarkkitehtuurin periaatteita ja -linjauksia, näistä kerrotaan tarkemmin alaluvussa 3.4. Kokonaisarkkitehtuuripäätökset tehdään päätöksen aiheesta riippuen joko Kela-tasoisesti tai Kanta-tasoisesti. Kokonaisarkkitehtuuriasioiden prosessissa käydään päätettävä asia läpi, jossa tutkitaan esimerkiksi arkkitehtuurinäkökulma, johon päätös tulisi vaikuttamaan. Tämän jälkeen asia siirtyy asiaa käsittelevälle taholle, erimerkiksi teknologiaa käsittelevä asia siirtyy Teknologia CoE:n käsiteltäväksi.

### 3.2.1 Arkkitehtuurin kehittäminen ja dokumentointi Kanta-palveluissa

Tämän tutkimuksen yhtenä tarkoituksena on tunnistaa ne Kanta-palveluiden arkkitehtuurin dokumentoinnin prosessin vaiheet, jossa dokumentoinnilla on merkitystä. Näin osataan jatkossa kuvata ne tärkeät toiminnot, jotka edesauttavat järjestelmän suunnittelemista.

Arkkitehtuurien kehittämistyötä Kanta-palveluissa tehdään aktiivisesti, esimerkiksi suunnittelua ja ohjausta dokumentaation työstämiseen. Arkkitehtuurikuvauksien työstämiseen on käytössä Sparx Systemsin Enterprise Architect ja muita työvälineitä voi olla myös Microsoft Visio ja Confluencen Draw.io. Päämallinnuskielenä käytetään ArchiMatea, jolla kuvataan suurimmalta osalta vain liiketoimintakerrosta ja työväline on käytössä ainoastaan arkkitehteilla.

Kanta-palveluiden arkkitehtuurin dokumentoinnissa on ylätasoinen arkkitehtuuriin liittyvä kuvauksia, sekä teknisiä arkkitehtuurikuvauksia. Arkkitehtuurien kuvaamiselle on luotu Word-dokumenttipohja, jossa kuvataan yleisellä tasolla toteutettavan järjestelmän arkkitehtuuri. Tähän yleisen tason kuvauspohjaan on yhdistetty kaikki arkkitehtuurinäkökulmat eli toiminta-, tieto-, tietojärjestelmä-, sovellus- ja teknologia-arkkitehtuurikuvaukset. Arkkitehtuurinäkökulmista on mainittu myös teoriaosuuden alaluvussa 2.1.1. Kuvauspohjaan on lueteltu ne kohdat, joita järjestelmästä tulisi kuvata, soveltaen mitä järjestelmää sillä hetkellä kuvataan.

Toiminta-arkkitehtuurikuvaukseen tulee kuvata käyttäjät ja muut toimijat, joille järjestelmä on tarkoitettu, palvelukartta, jossa kuvataan järjestelmän tarjoamat tietojärjestelmäpalvelut sekä pääkäyttötapausten. Käyttötapausten muutoin ovat enemmän määrittelytyöhön liittyvää tekemistä, joten tässä vaiheessa on tarkoitus kuvata käyttötapausta hyvin yleisellä tasolla.

Tietojärjestelmäarkkitehtuurissa kuvataan johdanto järjestelmän tietoarkkitehtuuriin karkealla tasolla, kuvataan käsitteellinen taso, johon kuuluu käsitteet,

päätietyöryhmät tai -tietotyypit, sekä looginen ja fyysinen taso, johon liittyy näiden tietomallit ja järjestelmän käyttämät tallennusratkaisut.

Tietojärjestelmäarkkitehtuurissa kuvataan tietojärjestelmäkartta, joka on visuaalinen ja palvelukeskeinen kuvaus järjestelmäkokonaisuudesta. Kuvataan myös suunnittelun pääperiaatteet ja linjaukset, osajärjestelmien tarkoitukset, tarjottavat rajapinnat sekä käytettävät rajapinnat.

Sovellusarkkitehtuurissa kuvataan järjestelmän toteutustason pääasiat tukemaan ja yhtenäistämään toteutustyötä ja testausta. Teknologia-arkkitehtuurissa kuvataan mitä standardeja ja teknologioita järjestelmän suunnittelussa ja toteutuksessa käytetään.

Arkkitehtuurin dokumentointi on muutakin kuin arkkitehtuurikuvauksien dokumentointia. Vaatimusmäärittelyt ja käyttötapaukset ovat käytännössä myös arkkitehtuurin dokumentaatiota, mutta enemmän käytännön tasolla. Dokumentaatioiden lopullinen tarkoitus on informaation välittämistä eri palveluiden ja sidosryhmien välillä.

### 3.2.2 Arkkitehtuuriperiaatteet

Kanta-palvelujen arkkitehtuurisuunnittelussa noudatetaan yleisiä arkkitehtuuriperiaatteita ja -linjauksia. Linjaukset ovat osittain Koko kelaä koskevia strategian perusteella tehtyjä linjauksia mutta joitakin linjauksia joudutaan soveltamaan tilanteen mukaan Kanta-palveluihin soveltuviksi, esimerkiksi Kanta-palveluita koskevan lainsäädännön vuoksi. Näitä periaatteita ja linjauksia ovat:

- yleiset periaatteet
- toimintaan vaikuttavat periaatteet
- tietoon kohdistuvat periaatteet
- tietojärjestelmiin kohdistuvat periaatteet
- teknologiaan kohdistuvat periaatteet sekä
- tietoturvaan ja varautumiseen kohdistuvat periaatteet

Yleisiin periaatteisiin kuuluu noudattaa JHS-arkkitehtuuriperiaatteita, Kanta-arkkitehtuuriperiaatteita, linjauksia ja pätemisjärjestystä, voimassa olevaa Kanta-palveluihin liittyvää lainsäädäntöä sekä voimassa olevaa yleistä mutta myös Kanta-palveluihin ohjaavaa lainsäädäntöä. Yleisiin periaatteisiin on mahdollista tehdä projektikohtaisia soveltamisohjeita.

Toimintaan vaikuttavia periaatteissa Kelan tulee osallistua Kanta-lainsäädännön tulkintatyöhön lainsäädännön toimeenpanon näkökulmasta. Kelan tulee tarjota tukipalveluita, informaatiota ja ohjeita Kanta-palveluun liittyville tahoille sekä järjestelmätoimittajille. Teknologia-arkkitehtuurin omistajuus ja ylläpito ovat Kelalla sekä järjestelmätoimittajat vastaavat omien tuotteidensa testaamista asiakastestiympäristössä.

Tietoon kohdistuva periaate kertoo, että Kanta-palveluihin tallennetun tiedon laatu tulee varmistaa sekä tiedon tuottamisprosessin, staattisen tallennettavan tietosisällön sekä tallennetun tiedon hyödyntämisen näkökulmasta. Tämä



tarkoittaa sitä, että tallennettavan tiedon laatu tulee varmistaa eli varmistetaan yhtenäisten määritysten, luokitusten ja koodistojen käytöstä.

Tietojärjestelmiin kohdistuvat periaatteet tarkoittavat sitä, että Kanta-palvelujen tulee perustua teknisiltä rajapinnoiltaan yhteen toimivuuden ja avoimuuden periaatteille. Periaatteella pyritään myös yksinkertaiseen ja yhdenmuukaisiin ratkaisuihin, jotka ovat tehokkaita, ymmärrettäviä ja ylläpidettäviä. Tietojärjestelmäkehityksessä tulee noudattaa Kanta-viitearkkitehtuuria sekä tulee huomioida vaatimus nopeasta reagoinnista lainsäädännön muutoksiin.

Teknologiaan kohdistuvat periaate liittyy palveluväylään. Palveluväylä tarkoittaa sitä, että sisäinen arkkitehtuuri perustuu palveluväylään ja kaikki sanomapohjaiset yhteydet Kanta-palvelujen välillä tulisi keskitetysti hoitaa tämän väylän kautta.

Tietoturvaan ja varautumiseen liittyvät periaatteet korostavat sitä, että tietoturvaratkaisujen tulee lisätä ja ylläpitää tietoturvaa. Ratkaisuja ei tehdä vain siksi, että ne näyttävät hyviltä vaan niiden pitää olla hyödyllisiä. Päällekkäisiä tietoturvakerroksia ei tule rakentaa, vaan päällekkäisiä ratkaisuja voidaan jopa purkaa, koska ne eivät yleensä lisää tietoturvaa eivätkä ole kustannustehokkaita.

Tämän tutkimuksen kannalta olennainen linjaus on nimeltään 'Tietojärjestelmien dokumentointi'. Linjauksen mukaan tietojärjestelmien tulee olla dokumentoituna koko niiden elinkaaren ajan ja dokumentaation tulee mahdollistaa tietojärjestelmän kehittäminen, ylläpito ja hallittu tuotantokäyttö. Dokumentaatiota tulee ylläpitää, jotta se on ajan tasalla koko ajan. Ylläpito on myös otettava huomioon dokumentaation toteuttamisessa, dokumentaation on oltava riittäväällä tasolla kuvattu ja ylläpidon hyväksymä.

Seuraavassa luvussa esitellään tutkimuksen tutkimusmenetelmät, joka sisältää haastateltavien valintaa ja tulosten analysoinnin toteutusta.

## 4 TUTKIMUSMENETELMÄ

Tässä luvussa käydään läpi tutkimuksen toteutukselle olennaiset asiat, kuten käytetyt tutkimus- ja tiedonkeruumenetelmät ja kuinka tutkimustuloksia on analysoitu. Ensimmäisessä alaluvussa kerrotaan tutkimusote ja tiedonkeruumenetelmät perusteluineen. Toisessa alaluvussa esitellään teemahaastattelun toteutusta ja haastateltavien valintaa sekä haastattelun toteutusta. Kolmannessa alaluvussa käsitellään analysointimenetelmää ja neljännessä alaluvussa aineiston luottavuutta.

### 4.1 Tutkimusote ja tiedonkeruumenetelmät

Tutkimusotteena käytettiin laadullista tapaustutkimusta, jossa haastateltiin Kanta-palveluiden arkkitehtuurin parissa työskenteleviä asiantuntijoita, tarkemmin mainittuna, kehitystiimeissä työskenteleviä arkkitehtejä, kehittäjiä ja ylläpidon asiantuntijoita. Laadullinen tapaustutkimus sopii tähän tutkimukseen, koska tämän tyylinen tutkimus perustuu Puusan ja Juutin (2020) mukaan ihmisten subjektiivisten kokemusten ja näkemysten tarkasteluun ja tutkimuksessa on tarkasteltava niin, teorian, empirian kuin käytännön yhteyttä. Laadullisella tutkimuksella voidaan tutkia tiettyä ilmiötä tai tapahtumankulkua, kuten tämän tutkimuksen kohteena olevaa arkkitehtuurin dokumentointia, joka on tietty vaihe järjestelmäkehityksen prosessista (Laine, Bamberg, & Jokinen, 2007).

Tutkimuksen tiedonkeruu tapahtui haastatteluilla, koska tiedonkeruumenetelmänä haastattelu on joustava ja moniin lähtökohtiin sekä tarpeisiin sopiva menetelmä (Hirsjärvi & Hurme 2011). Haastatteluilla pyrittiin saamaan tietoa arkkitehtuurin dokumentoinnin nykytilasta, dokumentoinnin tarpeesta sekä tietoa haastateltavien kokemuksista ja näkemyksistä dokumentointiin liittyen. Haastattelu tiedonkeruumenetelmäksi valikoitui myös sen vuoksi, että haastattelussa on mahdollista muuttaa kysymysten järjestystä luontevasti haastattelun edetessä ja tarkentaa haastateltavan vastauksia, jos siihen on tarvetta. Hirsjärvi & Hurme (2011) kertovat haastattelun eduiksi vastauksien selventämisen ja syventämisen haastateltavan niin halutessaan. Muita haastattelun etuja he mainitsevan olevan esimerkiksi sen, että ihmisen on mahdollista tuoda vastauksissa esille asioita mahdollisimman vapaasti (Hirsjärvi & Hurme, 2011). Tämä maininta sopii myös tämän tutkimuksen haastatteluun, koska siinä halutaan tuoda esille haastateltavan omia näkemyksiä ja kokemuksia arkkitehtuurin dokumentoinnista kokonaisuudessaan.

Tiedonkeruumenetelmänä käytettävän haastattelun tarkentavana menetelmänä on teemahaastattelu. Teemahaastattelussa on ennalta valikoidut teemat, jonka mukaan haastattelussa keskustellaan. Hirsjärvi ja Hurme (2011) kutsuvat teemahaastattelua puolistrukturoitukseksi haastattelumenetelmäksi, jossa teema-alueet ja aihepiirit ovat kaikille samat. Haastatteluiden alussa kysymysten tarkka

muoto oli vielä avoin ja kysymysten järjestys muuttui haastattelun aikana. Hirsjärvi, Remes ja Sajavaara (2012) mainitsevatkin tämän olevan tyypillistä teema-haastattelussa. Tuomi & Sarajärvi (2018) mainitsevat haastattelun olevan joustavaa sen vuoksi, että tutkija voi esittää kysymykset siinä järjestyksessä, kun katsoo sen olevan aiheellista. Tämä antoi mahdollisuuden tutkijalle johdattaa haastattelua luonnollisesti ja tarkentaa vastauksia joustavasti haastattelun aikana.

Haastattelun teemojen pohdinta lähti liikkeelle aiheen keskeisestä kohteesta eli arkkitehtuurin dokumentoinnista. Siitä muodostettiin tutkijan mielestä tärkeimmät kysymykset, joita haluttiin saada selville haastateltavilta henkilöiltä. Hirsjärvi ja Hurme (2011) mainitsevat teoksessaan, että varsinaiset haastattelukysymykset kohdistuvat teema-alueisiin ja teema-alueet tulee tarkentaa kysymyksillä. Haastattelutilanteessa teema-alueuuttelo ohjaa keskustelua ja haastattelijaa muistilistamaisesti (Hirsjärvi & Hurme, 2011).

Ensimmäinen teema on haastateltavien taustatiedot, jotta voidaan saada tietoa haastateltavien taustoista. Toiseksi teemaksi valikoitui arkkitehtuurin dokumentoinnin tila. Tällä teemalla saa kerättyä tietoa nykytilasta, jota voi myöhemmin verrata haluttuun lopputulokseen tai kehitysehdotuksiin. Kolmantena teemana on arkkitehtuurin dokumentoinnin tarve. Tällä saa selville haastateltavan näkemyksen siihen, mitä hän haluaisi, jotta arkkitehtuurin dokumentointi olisi mahdollisimman hyvää ja kustannustehokasta. Neljäntenä teema oli kommunikaatio. Tämä teema valittiin sen vuoksi, että kommunikaatio ja vuorovaikutus on tärkeää, jotta arkkitehtuurin dokumentointi ymmärretään samalla tavalla monessa eri kohderyhmässä.

Teemahaastattelulla kerättiin Kanta-palveluiden kehittäjiltä ja ylläpidon parissa työskenteleviltä asiantuntijoilta näkemyksiä, joiden pohjalta pyrittiin luomaan kokonaiskuva arkkitehtuurin dokumentoinnin tilasta ja tarpeesta sekä kommunikaatiosta. Haastateltaville lähetettiin sähköpostitse alla olevat teemat, jotta he pystyivät varautumaan haastatteluun etukäteen.

#### Teema 1. Haastateltavien taustatiedot

- Taustatietoja haastateltavasta, kuten työtehtävä, mallintamisosaaminen jne.

#### Teema 2. Arkkitehtuurin dokumentoinnin nykytila

- Keskustellaan mikä on nykyisen arkkitehtuurin dokumentoinnin tila tällä hetkellä.

#### Teema 3. Arkkitehtuurin dokumentoinnin tarve

- Keskustellaan minkä tasoiselle dokumentoinnille on tarvetta, minkälaista dokumentaatiota tehdään ja mitä voisi tehdä toisin.

#### Teema 4. Kommunikaatio

- Keskustellaan dokumentaatiosta kommunikaation välineenä.

Näiden aiemmin mainittujen teemojen perusteella laadittiin haastattelurunko, jossa oli tarkempia kysymyksiä, jotta haastattelu sujuisi ilman turhia katkoksia.

## 4.2 Haastateltavien valinta

Haastateltavat henkilöt olivat Kanta-palveluiden eri kehitystiimeissä työskenteleviä arkkitehtejä, kehittäjiä ja ylläpidon asiantuntijoita. He toimivat tiimeissään oman tuotealueensa asiantuntijoina. Nämä henkilöt on valittu haastateltaviksi nimenomaan sen vuoksi, että tutkimuksen aihe on arkkitehtuurin dokumentointi kehittäjän ja ylläpidon näkökulmasta. Tämän tutkimuksen haastattelun kohdejoukko on kehittäjät ja ylläpidon asiantuntijat, joilta kysytään heidän mielipidettä arkkitehtuurin dokumentoinnista.

Tutkimuksen kohdejoukolle kysymykset on esitetty mahdollisimman samalla tavalla, jotta vastaukset saadaan standardoituun muotoon. Myös tällaisesta joukosta vastauksia voidaan etsiä mahdollisia yhtäläisyyksiä ja eroja (Hirsjärvi & Hurme 2011).

### 4.2.1 Haastattelujen toteutus

Tammikuussa 2022 valituille sovellusarkkitehdeille, sovelluskehittäjille ja ICT-palveluvastaaville lähetettiin sähköpostikysely, jossa kartoitettiin halukkuutta osallistua haastatteluun. Sähköpostikyselyn löytää liitteestä 1. Haastattelut toteutettiin käytännössä helmikuussa 2022. Haastattelut käytiin videoneuvottelun keinoin, käyttäen Zoom-videoneuvottelutyövälinettä. Videoneuvottelu oli tähän tarkoitukseen pakollista, koska haastateltavat asiantuntijat työskentelevät laajasti koko Suomen alueella ja tutkimuksen kirjoittamisen hetkellä maailmalla liikkui koronavirustauti (Covid-19), jonka vuoksi kaikki haastateltavat olivat etätyössä. Videoneuvottelu on sen vuoksi hyvin joustava ja kustannustehokas tapa toteuttaa haastattelut.

Haastattelut tallennettiin Zoom-videoneuvottelutyökalun omalla tallennusominaisuudella. Haastattelut tallentuivat heti haastattelun jälkeen haastateltavan omalle tietokoneelle. Ja jotta välttyttiin mahdolliselta äänityksen epäonnistumiselta, haastateltava myös kirjoitti vastaukset ylös Microsoft Word -työkalua käyttäen. Haastatteluja käytiin yhteensä 13 kappaletta ja niihin meni aikaa yhteensä 10 tuntia ja 20 minuuttia.

Jokainen haastattelu sisälsi neljä eri aihealuetta, jotka lähetettiin haastateltaville etukäteen sähköpostitse. Näin haastateltavat pystyivät alustavasti valmistautumaan tulevaan haastatteluun.

### 4.2.2 Haastateltavien taustatiedot

Haastattelut aloitettiin haastateltavan taustatietojen kartoittamisella. Tähän ei itse haastattelussa käytetty aikaa, vaan kysymykset lähetettiin haastateltaville

sähköpostitse haastattelun jälkeen. Tämä sen vuoksi, että yhteen haastatteluun per henkilö oli varattu aikaa noin 45–60 minuuttia. Taustatietokysymyksissä kysyttiin henkilön työtehtävää, työkokemusta Kelassa ja muualla, missä tiimissä henkilö työskentelee, tarkempaa työnkuvaa, toimiiko enemmän kehitys- vai ylläpitotehtävissä vai kummassakin. Lisäksi haastateltavilta kysyttiin, millaista osaamista heillä on tuottaa arkkitehtuurin liittyviä malleja ja kuvauksia.

Haastateltavista kuusi (6) henkilöä on sovelluskehittäjiä, näistä kahdella (2) toisena roolina on sovellusarkkitehdin rooli ja yhdellä (1) scrum masterin rooli. Neljällä (4) haastateltavalla on sovellusarkkitehdin rooli. Kaksi (2) haastateltavaa ovat ICT-palveluvastaavia, jotka työskentelevät tuotannon ylläpidossa ja yksi (1) testaaja, jolla on myös sovellusarkkitehdin rooli.

Haastateltavilla on työkokemusta sekä Kelassa että muualla. Keskimääräisesti he ovat työskennelleet Kelassa noin 8,3 vuotta, pisimpään 24 vuotta ja lyhimmän aikaa 8kk. Muualla he ovat työskennelleet keskimäärin 8,8 vuotta, josta pisimpään 18 vuotta ja lyhimmän aikaa 0 vuotta.

Yhtenä kysymyskokonaisuutena oli mikä on haastateltavan tarkempi työtehtävä ja toimiiko kehitys- vai ylläpitotehtävissä. Pelkästään kehitystehtäviä tekee viisi (5) henkilöä ja ylläpitotehtäviä neljä (4) henkilöä. Kehitys- ja ylläpitotehtävien yhdistelmää tekee neljä (4) henkilöä. Viimeisenä kysyttiin henkilön mallintamisosaamista. Haastateltavista 11 henkilöä mainitsee osaavansa arkkitehtuurin mallintamista ja kuvauksien tekemistä mutta kaksi (2) kertoo, ettei kokemusta mallintamisesta ole. Taulukossa 1 nämä on mainittu haastateltavan keskeisimmät taustatiedot.

TAULUKKO 1 Haastateltavien taustatietoja

Haastateltava	Työtehtävä	Kehitys/ylläpito	Mallintamisaami-nen
H1	Sovelluskehittäjä	Kehitys	Ei kokemusta
H2	Sovelluskehittäjä ja -arkkitehti	Kehitys	On
H3	Sovelluskehittäjä, scrum master	Kehitys ja ylläpito	On
H4	Sovellusarkkitehti	Kehitys ja ylläpito	On
H5	ICT-palveluvas-taava	Ylläpito	On
H6	Sovelluskehittäjä	Kehitys	Ei kokemusta
H7	Testaaja, sovel-lusarkkitehti	Ylläpito	On
H8	Sovelluskehittäjä	Kehitys	On
H9	Sovellusarkkitehti	Kehitys	On
H10	ICT-palveluvas-taava	Ylläpito	On
H11	Sovellusarkkitehti	Ylläpito	On
H12	Sovelluskehittäjä ja -arkkitehti	Kehitys ja ylläpito	On
H13	Sovellusarkkitehti	Kehitys ja ylläpito	On

Haastateltavat ovat työskennelleet IT-alalla keskimääräisesti noin 17,2 vuotta. Muilla paitsi kahdella (2) haastateltavalla on työkokemusta julkishallinnon ulkopuolelta, yksityissektorilta. Monet haastateltavista tekevät arkkitehtuurityötä muiden tehtävien ohessa, kuten sovelluskoodin kirjoittamisen tai ylläpito töiden ohessa. Haastateltavilta löytyy useita eri rooleja ja kokemusta erilaisista työtehtävistä IT-alalla.

### 4.3 Aineiston analysoinnin toteutus

Tutkimuksen tiedonkeruumenetelmänä oli teemahaastattelu ja sanatarkkaan tutkimusaineiston litterointiin ei ollut tarvetta. Haastattelun litteroinnista on kerrottu tarkemmin seuraavassa luvussa 4.3.1. Tutkimusaineistoa analysoitiin sisällönanalyysillä. Sisällönanalyysi auttaa tiivistämään, selkeyttämään ja tekemään selkeän kokonaisuuden tutkimusaineistosta (Juuti & Puusa, 2020). Analyysin näkökulmana on teoriasidonnaisuus, joka voi olla sekä teoria- että aineistolähtöistä

analyysiä (Juuti & Puusa, 2020). Juuti ja Puusa (2020) mainitsevat, että aineiston analyysin kautta voidaan kuvailla ja ymmärtää tutkimuksen kohdetta eli tutkimuksen ilmiötä.

Tutkimuksessa analysoitavaa haastattelumateriaalia oli 13 haastattelun verran, jota tuntimääräisesti 10 tuntia ja 26 minuuttia. Haastattelujen määrä ja tuntimäärä eivät kohtaa sen vuoksi, että osa haastatteluista ei kestänyt täyttä 60 minuuttia, vaan esimerkiksi 45 minuuttia. Analysoitavaa materiaalia on kuitenkin hyvin runsaasti, keskimääräisesti 6 sivua per haastattelu. Taulukko 2 tarkentaa haastattelun kestoja sekä litteroidun materiaalin sivumäärää.

TAULUKKO 2 Haastattelun tiedot

Haastateltava	Haastattelun kesto	Litteroidun materiaalin sivumäärä
H1	50 min	6
H2	57 min	6
H3	40 min	4
H4	65 min	8
H5	55 min	6
H6	45 min	4
H7	42 min	5
H8	40 min	5
H9	45 min	6
H10	40 min	6
H11	37 min	5
H12	52 min	8
H13	48 min	6
<b>Yhteensä</b>	616 min	75

Haastattelujen aikana tutkija keräsi vastauksia ja kommentteja osittain valmiiksi ja pystyi aloittamaan analysoinnin jo haastatteluiden aikana. Jokaisen haastattelun jälkeen tutkija luki kerätyt muistiinpanot heti läpi ja sai näin ensivaikutelman juuri käydystä haastattelusta. Tutkija myös litteroi haastattelu aineiston mahdollisimman pikaisesti haastatteluiden jälkeen, jotta haastattelut olivat vielä tuoreessa muistissa. Puusa ja Juuti (2020) mainitsevat, että analyysin ensimmäinen vaihe kannattaa suorittaa aineistolähtöisesti. Tällöin tutkijan tulee lukea aineistoa mahdollisimman avoimin mielin niin, että mikään teoreettinen näkökulma ei rajoittaisi tai vaikuttaisi aineiston sisällön tulkintaa.

Varsinaisesti tutkimusaineiston analysointi aloitettiin käymällä jokainen haastattelu ja siihen liittyvät kysymykset läpi useaan kertaan. Tutkimusaineistoon sisältyi haastatteluista syntyneitä taltioita ja muistiinpanoja. Samalla tutkija teki tekstinkäsittelyohjelman toiminnolla merkintöjä mielenkiintoisiin kohtiin, jotka olivat tutkimuksen kannalta relevantteja. Tämän jälkeen aineistosta värikoodauksen avulla koostettiin samankaltaisuuksia. Puusan ja Juutin (2020) mukaan koodaaminen on yksinkertaisimmillaan yhtenäisten merkitysten tai sanojen tunnistamista. Värikoodauksen jälkeen haastateltavien kommentteja teemoiteltiin. Tällä tavalla vastauksista saatiin esille useimmin toistuvia asioita, joista

lopulta muodostui suurempia käsiteltäviä kokonaisuuksia. Hirsjärvi ja Hurme (2011) mainitsevatkin, että analyysivaiheessa teemoittelulla tarkastellaan aineistosta nousevia asioita, jotka ovat yhteisiä usealle haastateltavalle.

#### 4.3.1 Haastattelun litterointi

Haastattelumateriaalin analysoinnissa ei tässä tutkimuksessa lähdetty aivan sanatarkkaan litterointiin, koska se ei ollut tähän tutkimukseen relevanttia. Haastatteluista oli tarkoitus poimia sellaisia asioita, jotka olivat oleellisia tutkimuksen tulosten kannalta. Kuten aiemmassa kappaleessa jo mainittiin, haastatteluiden jälkeen aineisto litteroitiin mahdollisimman pian, jotta tutkijalla oli haastattelussa käydyt asiat tuoreena mielessä. Tässä tutkimuksessa hyvin kattavaa tai suurinta mahdollista litteroinnin tasoa ei ollut tarpeellista käyttää, koska tarkoituksena ei ollut tehdä keskusteluanalyysiä. Keskusteluanalyysin Ayaß (2015) mainitsee olevan tarkkaa kielen käyttöä ja siinä tulee käyttää hyvin tiukkoja litterointiin liittyviä käytänteitä ja menetelmiä. Tässä tutkimuksessa kaikilla hengitysäänillä tai muilla vuorovaikutukseen vaikuttavilla tekijöillä ei ollut merkitystä tutkimuksen tulosten kanssa. Tämän tutkimuksen litteroinnissa haastattelut kirjoitettiin ylös suunnilleen sanasta sanaan, jos keskustelun osa haluttiin tuoda tutkimuksessa esiin kirjallisessa muodossa. Tämän tasoisen litteroinnin hyötynä oli se, että haastattelusta ei tullut rajattu myöskään liikaa tulosten kannalta hyödyllistä aineistoa pois.

#### 4.3.2 Aineiston luotettavuus

Haastattelumateriaalin luotettavuus voitiin mahdollistaa hyvin suunnitelluilla haastattelun teemoilla, teemoja tarkentavilla ja syventävillä haastattelukysymyksillä sekä toimivalla videoneuvotteluvälineellä.

Videoneuvotteluvälineen luotettavuus ja toimivuus testattiin muutaman kerran ennen haastattelua. Tutkija testasi nauhoitteiden tallentamista ja ääneen kuulumista. Kesken haastattelun tarkistettiin, että tallennus on edelleen käynnissä. Haastattelumateriaalin kirjoittaminen aloitettiin jo haastattelun aikana tutkijan toimesta, hän kirjoitti vastauksia omalle koneelleen. Haastattelurungon läpikäymisen jälkeen haastattelijä kävi kysymykset vielä kerran läpi ja totesi, että kaikkiin kysymyksiin on vastattu. Haastateltavalta myös kysyttiin lupa ottaa häneen yhteyttä, jos jokin seikka jäi epäselväksi. Muutamalle haastateltavalle kysymys ”Millaista dokumentaatiota pitäisi tehdä, että se olisi täydellistä?”, tuotti hankaluuksia ja sen muotoa tuli hieman tarkentaa seuraaviin haastatteluihin. Kysymyksen merkitystä ei varsinaisesti muutettu vaan tarkennettiin, että ”Millaista dokumentaatiota haastateltavan mielestä pitäisi tehdä, että se olisi täydellistä ja ymmärrettävää?”.

Yleisesti haastattelun laatua voidaan Hirsjärven ja Hurmeen (2017) mukaan parantaa litteroimalla aineisto mahdollisimman pian haastattelun jälkeen. Tässä tutkimuksessa haastattelut litteroitiin heti haastattelun jälkeen, jotta kaikki mahdolliset tulkinnat ja mieleen tulleet asiat olivat tuoreena tutkijan mielessä.



## 5 TUTKIMUKSEN TULOKSET

Tässä luvussa käsitellään tutkimuksen tuloksia Kanta-palveluiden arkkitehtuurin dokumentoinnin nykytilasta, tarpeesta ja kommunikaatiosta kehittäjän ja yläpidon asiantuntijoiden näkökulmasta. Luvun tarkoituksena on esitellä tutkimuksessa löydettyjä havaintoja ja tuloksia sekä löytää vastauksia siihen, miten arkkitehtuurin dokumentoinnin kokonaisuus toimii tällä hetkellä.

Haastateltavat oli valittu Kanta-palveluiden eri palveluista, jotta saadaan mahdollisimman laaja kokonaiskuva dokumentaation tekemisestä, eikä vastaukset näin ollen keskittyisi liikaa vain muutamaaan palveluun.

Aivan haastatteluiden alkuvaiheessa tuli ilmi, että mitä arkkitehtuurin dokumentaatiolla oikeastaan tarkoitetaan. Tarkoitetaanko sillä vain arkkitehtuurinäkökuulmissa mainittuja tieto-, toiminta-, järjestelmä- sekä sovellus- ja teknologia-arkkitehtuuria vai kuuluuko esimerkiksi määrittelydokumentaatio osaksi arkkitehtuurin dokumentaatiota. Yhdessä haastateltavien kanssa päädyimme siihen olettamukseen, että käsittelemme dokumentaatiota kokonaisuudessaan, johon kuuluu määrittelydokumentaatio sekä pienet yksityiskohtaiset dokumentaatiot.

### 5.1 Arkkitehtuurin dokumentoinnin nykytila

Haastattelun yhtenä teemana oli käsitellä arkkitehtuurin dokumentoinnin nykytilaa Kanta-palvelussa. Teemaan liittyvillä kysymyksillä oli tarkoitus saada tarkempaa kuvaa siitä, millaista dokumentaatiota tehdään, kuvataanko nykytilaa, tavoitetilaa vai molempia. Entä onko dokumentaatiot helposti löydettävissä tai ajan tasalla. Tärkeäksi kysymykseksi otettiin mukaan myös se, että onko dokumentaatiota kuvattu riittävällä tasolla. Jääkö kenties jotain tarpeellista kokonaan kuvaamatta, joka on olennaista kokonaisuuden ja kehittämisen kannalta tai mahdollisesti estää kehittämistyötä merkittävällä tavalla. Kysyttiin myös, miten arkkitehtuuri- ja mallintamispäätöksiä ratkaistaan, sekä millaista dokumentaatiota olisi hyvä tehdä, että se olisi täydellistä.

Kysymykseen mikä on nykyisen arkkitehtuurin dokumentoinnin tila tällä hetkellä ja millaista dokumentaatiota tällä hetkellä tehdään, sai monen kirjavana vastausta. Haastateltavat kertoivat hyvin yksityiskohtaisesti, millaista sovellusarkkitehtuuria heidän palvelussaan tehdään. Teknisiä järjestelmäarkkitehtuurikuvauksia löytyy jokaisesta palvelusta, näitä kuvauksia ovat esimerkiksi sijoittelukaaviot, sekvenssikaaviot sekä komponenttikaaviot.

Meillä [palvelussa] on hyvin eri kerroksia, pienempiä osapalikoita, niiden välisiä riippuvuuksia. Kuvataan niitä, miten ne on missäkin kerroksissa, mikä kutsuu mitäkin. Ollaan tehty sijoittelukaaviota, missä on kuvattu eri komponenttien välisiä riippuvuuksia eli karkeasti sitä mitä meillä on ja mitä uutta me tehdään ja mitä eri

komponentit kutsuu ja mitkä niiden riippuvuudet on. Meillä on arkkitehtuurikuvauksia, järjestelmäkuvauksia, tulevaisuuden tavoitetilakuvauksia, sovellusarkkitehtuuri. (H1)

Haastateltavat mainitsivat, että kaikenlaiselle dokumentaatiolle on paikkansa. Dokumentaatiota tehdään kooditasolla saakka eli ohjelmakoodiin kehittäjät kirjoittavat dokumentaatiota ja web-sovellusten osalta löytyy rajapintakuvauksia. Myös asennusohjeita ja versiosaatteet dokumentoidaan ja nämä mielletään osaksi kokonaisprosessia.

Ihan kooditasolla javadocia tehdään ja sitten järjestelmästä nämä tekniset arkkitehtuurikuvaukset. Sitten ihan Web -sovellusten osalta löytyy rajapintakuvauksia. Nää ei sitten ihan ole arkkitehtuurikuvauksiin niinkään liity, kuten asennusohjeet ja versiosaatteet dokumentoidaan, että ne on osana tätä kokonaisprosessia. Ja määrittelyt ja käyttötapaukset on sitten tottakai kaiken toteutuksen pohjana, että saadaan sitä arkkitehtuuridokumentointia aikaiseksi. Aika laaja-alaisesti dokumentoidaan, joka näkökulmasta pyritään dokumentoimaan. Kyllä nää arkkitehtuurikuvaukset, on niin kuin ihan sovellus, jokaisesta sovelluksesta tehdään, sovellustasolla tehdään ja sitten tietokannat kuvataan rakenteeltaan. (H13)

Osassa palveluita on hieman edistyneempää vai voisiko sanoa täydellisempää dokumentointia. Palvelusta löytyy kaikki arkkitehtuurinäkökulmiin liittyvät arkkitehtuurikuvaukset. Sovellusarkkitehdit H9 ja H11 kertoivat, että dokumentaation tila on heidän mielestään hyvällä tasolla ja dokumentaatiota on kehitetty paljon, jotta dokumentaatio on saatu riittävälle tasolle.

Sovellusarkkitehtuurin osalta meillä on toiminta-arkkitehtuurin kuvaus, tietoarkkitehtuurinkuvaus, järjestelmäarkkitehtuurinkuvaus, teknologia- ja sovellusarkkitehtuurikuvat, jotka ovat yhdistetty samaan dokumenttiin. (H9)

Meidän arkkitehtuuridokumentaatio on tehty semmoisella tasolla, että meillä on segmentoitu se muutamaani eri arkkitehtuuritasoon, että kuin tarkkoja ne tiedot on. Löytyy semmoinen nollataso, että tässä se meidän palvelu X ja yhteydet muihin palveluihin, se on sen kuvaus. Sen jälkeen on tarkempia, oman palvelun sisällä olevat komponentit ja miten ne asioi toisensa kanssa. Sit mennään vielä tarkempiin, mitkä on komponenttien sisäisiä toimintoja, että mitä ne tekee. Sit on rajapintadokumentaatiota, että minkälaisia rajapintoja on olemassa, mitä ne vastaanottaa ja mitä ne vastaa, erilaisia virhetilanteita, ja mitkä on pakollisia tietoja mitä tehdään. Lisäksi voi olla semmoisia, kulkukaavioita, että miten esimerkiksi käyttötapaukset toimii palveluiden sisällä, esimerkiksi kun henkilö tekee pyynnön jonnekin, miten se liikkuu se tieto, miten se käyttäjän pyyntö liikkuu palvelusta toiseen ja miten pyydetään ja miten vastetaan ja mikä lopputulos, mihin päästään. Sen tarkoitus on enemmän antaa hahmotusta sellaisille toiminnolle, missä kehittäjä tai kukaan muu projektitiimin jäsen ei välttämättä osaa hahmottaa muuten miten se toimii. (H12)

Arkkitehtuurin kuvaaminen on tarvelähtöistä eli kuvataan nykytilaa, tavoitetilaa tai molempia, tilanteen mukaan. Sovelluskehittäjät H3, H6 ja H8 sekä ICT-palveluvastaava H10 kertoivat, että heidän palvelussaan kuvataan lähinnä vain nykytilaa mutta tunnistivat myös tavoitetilan merkityksen.

Ainakin nykytilaa. Toki sekin on hyvä, se tavoitetilakin. Mutta ainakin nykytila kuvataan. (H6)

Nykytilan perusteella kuvataan, kun tehdään parannuksia ja kun tarvetta on, niin nykytilaa päivitetään. (H10)

Sovellusarkkitehdit H2, H7, H12 ja H13 mainitsivat, että heidän palvelussaan kuvataan molempia eli niin nyky- kuin tavoitetilaakin. Nykytilaa kuvataan silloin tarpeen mukaan, esimerkiksi päivitysten vuoksi. Tavoitetilaa kuvataan järjestelmän uudistamisessa ja tulevaisuuden suunnitelmien kuvaamisessa.

Sekin on tapauskohtaista, tällä hetkellä ei ole sellaista tavallaan tehtävää päällä, että meillä olisi tarve kuvata arkkitehtuurin nykytilaa. Nyt kun me lähdetään tekemään ja uudistamaan järjestelmää, siihen liittyen tehdään tavoitearkkitehtuurin kuvaustyötä. Sellaiselle työlle ei ole käytännössä edes aikaa, että lähdettäis järjestelmällisesti kuvaamaan ainakaan suuremmissa määrin nykyistä arkkitehtuuria. Ehkä se tulee sitä kautta, että dokumentaatiota tällä hetkelläkin yritetään kehittää. (H2)

Tämänhetkistä tilaa ja tulevaisuuden näkymiä, että mikä on suunnitelma. (H7)

Molempia, nykytilaa voidaan kuvata aikaisemman tekemisen mukaan ja sitten tavoitetilaa käytännössä on se, kun tulee uusi toiminnallisuus ja silloin se pyritään dokumentoimaan mahdollisimman tarkasti, että miten se toimii. (H12)

Molempia kuvataan, nykytila tottakai kuvataan ja sitten tietysti kun dokumentaatiota ylläpidetään, vanhoihin pitää aina palata ja pitää tehdä sinne niitä päivityksiä, jos huomataan, että jossain vaiheessa on jotain jäänyt puuttumaan. Mutta kyllä uuden kehittämisessä aina kuvataan myös sitä tavoitetilaa että, mihin me pyritään, se pitää olla selkeä. (H13)

Tavoitetilan kuvaaminen sai kaksi vastausta sovellusarkkitehdeiltä H9 ja H11. Näissä vastauksissa painotettiin sitä, että heidän palvelunsa ovat sellaiset, missä ei ole tarvetta nykytilan kuvaamiselle, koska palvelu on niin uusi, ettei nykytilan kuvaamiselle ole tarvetta.

Lähinnä tavoitetilaa tässä vaiheessa, ei ole ollut mitään palvelua, josta olisi lähdetty. (H9)

Tavoitetilaa järjestelmäarkkitehtuurissa on tarkoitus kuvata. (H11)

### 5.1.1 Dokumentaation löydettävyys

Vastausten perusteella tuotetut dokumentaatiot ovat ja eivät ole helposti löydettävissä. Viisi haastateltavaa vastasi, että heidän dokumentaationsa ovat löydettävissä Confluencen työtilasta helposti. Tämän perusteella näiden palveluiden tiimit ovat ottaneet aikaa työstää dokumentaatiot helposti löydettävään muotoon.

On, meillä ne on Confluencessa, ne on ihan tuossa missä on määrittelykuvaukset, siellä on oma kohta arkkitehtuurikuvauksille. Olemassa olevat on löydettävissä, päivittäisessä työssä en niitä käytä, mutta ovat hyviä. (H1)

Confluencessa kaikki on tällä hetkellä. Sovellusarkkitehtuurin osalta sillä on yksittäinen paikka missä se kaikki dokumentaatio löytyy per sovellus ja löytyy sitten niitä isompia kuvauksia, mitkä kuvaa ei niinkään yhtä sovellusta vaan koko palvelua. (H12)

No siihen pyritään kyllä. Kaikkien etuhan se on, että ne on sillein kootusti tietyssä paikassa, että niihin sitten tiimiläisellä on pääsy niihin kun niitä tarvitsee. Ettei vaikeuta työtä sen takia, että dokumentaatio on jossain mihin on vaikea päästä. (H13)

Suurin osa vastaajaa kertoi, että materiaalien löytämisessä on haasteita. Dokumentaatio saattaa olla liian hajautettua, sitä voi olla edelleen verkkolevyllä, Sine-tissä tai jossakin mallinnustyökalussa, jota ei enää aktiivisesti käytetä tai sen käyttöä on rajoitettu vain tietyille henkilöille. Tämä vaikeuttaa osittain materiaalien löytämistä. Yksi kertoi, että vaikka kaikki materiaali olisikin jo viety Confluenceen, siellä on monia työtiloja, jonka vuoksi ei materiaaleja on vaikea löytää.

Tällä hetkellä olemassa oleva dokumentaatio, kyllä siinä haasteita on, että löydät kaiken olemassa olevan dokumentaation, sitä on niin monessa paikkaa. Ei ole sellaista keskitettyä paikkaa missä ne on, missä vaikkapa arkkitehtuuria olisi kuvattu. Ne on vähän missä sattuu, siis meillä on Confluencessa valtaosa ja sitten Confluencessa on se ongelma, että se on kuitenkin meidän tapauksessa aika, no työtiloja on useita, Confluencen rakenne ei ole tällä hetkellä siinä kunnossa, että asioita löytäisi helposti. Siellä on kaikennäköisiä, osa on verkkolevyillä, osa versionhallinnassa. (H2)

Helposti löydettävissä sellaiselle henkilölle, joka tietää mitä hakea. Jos puhutaan, että pitäisi yhtäkkiä Confluencesta hakea, ei noin vaan löydy mutta ehkä tiimin sisällä ne löytyy. (H3)

Tässä on ollut vähän haasteita, sen takia, että historia on se, että niitä on ollut monessa paikassa. Oli niitä sparx, joka poistui ja tuli QPR ja kävi taas toisinpäin. Sitten oli verkkolevyille dumpattu asioita, sitten oli Sine-tissä jossakin niitä juttuja. Aikaisemmassa oli Jazz käytössä, oli myöskin sieltä. Nyt tullut JIRA + Confluence kombinaatio. (H4)

Voi olla vaikeata, et aika usein joutuu kysymään mistä dokumentit löytyvät. (H6)

Enpä usko. Palvelukuvauksiahan on verkkolevyllä. Osittain on Confluencessa ja osittain muualla, on epäselvää mitkä on missäkin. (H8)

Osa löytyy ja osa ei. Ei ne siellä kohtaa dokumentaatiota ole, mistä niitä intuitiivisesti niitä olisi etsinyt. (H10)

Muutamista vastauksista tulee selville se, että työtä materiaalien löytämiselle tehdään aktiivisesti. Tavoitteena on, että materiaalit vähitellen saadaan tuota kootusti Confluenceen palvelun omaan työtilaan.

Tavoite on, että me ollaan pyritty askel kerrallaan, enemmän tai vähemmän tehokkaasti keskittämään niitä sinne Confluenceen enemmän. (H4)

Näkisin kyllä, että tähän satsataan, että löytyisi paremmin. (H6)

### 5.1.2 Dokumentaation ajantasaisuus

Edellisessä kohdassa todettiin, että dokumentaatiot eivät välttämättä ole helposti löydettävissä, koska dokumentaatiot saattavat olla liian moneen eri paikkaan hajautettua. Seuraavaksi yritettiin löytää vastausta kysymykseen, ovatko kuvaukset ja aineistot ajan tasalla. Entä korjataanko tehtyjä suunnitelmia ja jos korjataan, minkälaiset korjataan. Myöhemmin pienetkin korjaamatta jätetyt dokumentaatiot voivat olla suuritöisiä ja aikaa vievää. Vastausta haetaan myös sille mikä on peruste jättää korjaamatta kyseisiä dokumentaatioita. Vaikka kehityksessä todetaisiin, että dokumentaation korjaaminen ei olisi taloudellisesti tai resurssien puitteissa kannattavaa, voi olla, että resurssit on kaikesta huolimatta kaivettava esiin ja dokumentaatio tehtävä.

Yleisesti vastauksien perusteella on havaittavissa, että dokumentaatiot eivät ole tarpeeksi ajan tasalla. Tarkentaen sellaiset dokumentaatiot, joita on useassa eri paikassa. Tämä näkemys kumpusi siitä, että dokumentaatiota voi olla esimerkiksi Confluencen työtilassa ja verkkolevyllä. Monessa eri paikassa olevissa dokumentaatioissa saattaa olla duplikaatteja, joten dokumentaatioiden päivittäminen on sen vuoksi hankalaa.

Useamman haastateltavan mielestä kuitenkin Confluencen työtilassa olevat dokumentit ovat paremmin ajan tasalla kuin muualla sijaitsevat dokumentaatiot.

Siinä voisi olla terävöittämissä paikka, että kun luodaan uusia ja niin poistettaisiin niitä vanhoja, ei ihan joka tapauksessa ole näin käynyt. (H10)

Ylläpidon näkökulmasta dokumentaatiot yleisesti ottaen eivät ole ajantasaisia. Ylläpidossa työskentelevät asiantuntijat tarvitsevat ajan tasaista dokumentaatiota ylläpidon työn luonteen vuoksi. Ylläpidolle on tärkeää, että esimerkiksi palvelinten riippuvuudet on kuvattu selkeästi.

Ei ole ajan tasalla. Toivoisin, että niitä joku ylläpitäisi aktiivisemmin. (H5)

Päävastuu ylemmän tason arkkitehtuurin dokumentaation päivittämisestä on sovellusarkkitehdillä. Sovellusarkkitehti H7 mainitsi, että ei ole kuitenkaan väärin, etteikö kukaan muu niitä voisi päivittää, jos huomaa puutteita. Tämä onkin tärkeää, että kaikki, jotka huomaavat puutteita dokumentaatioissa, päivittävät dokumentin tai kertoo puutteista muille, jolloin yhdessä tiimin tai arkkitehdin kanssa päivitetään tarvittavat muutokset.

Tiimillä pyritään päivittämään, päävastuu sovellusarkkitehdeillä mutta ei kukaan kiellä, etteikö kukaan muu voisi päivittää. (H7)

Loppujen lopuksi voidaan myös ajatella, että dokumentaatio ei koskaan voi olla ajan tasalla, koska kehitystehtäviä on paljon, varsinkin isossa palvelussa. Kuten

sovellusarkkitehti H13 mainitsi, että ikinä se ei tule olemaan täydellistä. Tutkimuksen tekijän mielestä täydellisyyteen on kuitenkin tavoiteltava, jotta dokumentaatio voi tulla edes lähelle täydellistä.

Dokumentaation toteuttaminen voi olla haastavaa, koska dokumentaatiota työstävällä henkilöllä ei välttämättä ole tietoa mikä on riittävä taso kuvauksessa tai kuvauksen tekijä on itse hyvin kokenut asiantuntija ja kirjoittaa liian suppeita dokumentaatioita. Näin ollen kokemattomampi asiantuntija tai uusi henkilö ei välttämättä saa dokumentaatiosta kaikkea irti. Osittain pitäisi myös sidosryhmät osallistaa osaksi dokumentaation tekemistä, jotta palvelun ulkopuoliset henkilöt osaisivat ottaa kantaa siihen, mikä on riittävää dokumentaatiota.

Neljä haastateltavaa kertoi, että kuvauksia ei ole dokumentoitu riittävällä tasolla. Kun palvelua aloitetaan ylläpitämään ja tarvittava dokumentaatio on puutteellista, voi ylläpidolla mennä turhaa aikaa järjestelmän dokumentaation etsimiseen.

Detaljitason kuvia saattaa jäädä pois. (H5)

Kyllä periaatteessa sovelluskehityksen näkökulmasta jää epäselväksi joitakin asioita. (H8)

Puutteita on esimerkiksi käytännön toimintaohjeissa, mitä oon sitten itselle tehnyt ylläpitoon. Maku asia tietysti se on, että millä tasolla ohjeet pitäisi olla. (H10)

Mun henkilökohtainen näkemys on, että voisi paremminkin kuvata tällä hetkellä. (H11)

Vastauksista käy ilmi myös se, että asioita on osittain dokumentoitu riittävästi mutta silti puutteita on esimerkiksi yksittäisissä ylläpidon tapauksissa. Ylläpitoa ja ongelmanratkaisua ajatellen olisi hyvä, että tarkempaa dokumentaatiota löytyisi riittävän helposti ja nopeasti.

Kyllä täydentämisen varaa varmasti on. Mitä olen katsonut esimerkiksi muualta, rajapintojen osoitteet tuotannossa ja kaikkea tämmöisiä. Joskus kaippaa niitä, jotka olisi hyvä löytää dokumentaatiostakin, ettei tarvitse käydä katsomassa palvelimelta mistä se oikeasti kutsuu. Siltä osin varmasti on täydentämisen varaa. (H3)

Aina voi sanoa, että jotain jää dokumentoimatta, koska siinä aina tulee se kysymys, että millä tasolla niitä dokumentoi, mitä alemmas menee, sitä enemmän tulee pieniä dokumentaatioita, mikä käytännössä on vaan hankalempaa, ja siinä vaiheessa voi hyödyntää jotain muuta dokumentaatiota. (H12)

Kaksi haastateltavaa mainitsivat toisaalta sen, että pelkästä dokumentaation ilosta ei kannata dokumentoida jotain pientä tiettyä yksityiskohtaa tai kyseenalaistavat mihin dokumentaatiota tarvitaan. Puutteita dokumentaatioissa voi paikata tarpeen tullen ja selventää myöhemmin joko täydentämällä dokumentaatiota tai keskustelemalla.

Oma suhtautuminen dokumentaatioon on hieman kiero, etten oikein pidä siitä, että asioita dokumentoidaan vain dokumentaation ilosta. (H7)

Välttämättä dokumentaation olemassaolon takia ei lähdetä tekemään dokumentaatiota. (H12)

Osa haastateltavista kertoi, että kuvaukset on dokumentoitu riittävällä tasolla.

En ainakaan valita. (H6)

Mun mielestä ne on hyvällä tasolla, tietysti siitä voi keskustella, että mikä on riittävä taso. En mä siellä ole nähnyt arkkitehtuurin mielessä pahemmin puutteellisuuksia. Eikä tuota kehittäjätäkään ole asiasta mitään sanoneet, että sieltä jotain puuttuisi. (H9)

Tavoite on, että sitä on kuvattu riittävästi. Niinkun tossa sanoin, niin meillä hyvin laajalla skaalalla löytyy sitä dokumentaatiota. Itse tietysti toivon, että jos joku kokee, että sitä ei ole dokumentoitu riittävästi, niin nostaa sen asian esille, että mitä niinkun puuttuu. (H13)

Yleisesti arkkitehtuuripäätökset teknologioihin liittyvissä asioissa päätetään Kanta-arkkitehtuurista vastaavien kesken. Muussa tapauksessa palvelut voivat tilannekohtaisesti palveluiden sisällä tarkastella millaisesta asiasta on kysymys ja ehdottaa uutta linjaa. Arkkitehtuuripäätöksien prosessi vastauksien perusteella menee seuraavasti:

- Palvelun arkkitehti selvittää vaihtoehdot palvelun kannalta
- Hakee mahdollisesti vertaistukea ratkaisuarkkitehdilta
- Hakee virallisen päätöksen

Meillä oma toteutustiimi pystyy ratkaisemaan varmaan aika isosti, oma porukka pystyy tekemään tiettyjä linjanvetoja. (H4)

Sanotaan, että pienemmät sovellusarkkitehtuuriratkaisut tehdään tiimin kesken, siihen osallistuu toteuttajat ja tarvittaessa muuta porukkaa. Isommat linjaukset käydään kanta-tasolla, kanta-arkkitehtien kanssa. (H11)

### 5.1.3 Mallinnus- ja piirtotyökalut

Vastauksista käy ilmi, että mallintamiseen liittyvät päätökset tehdään suurimmalta osin palveluissa ja tiimeissä itse. Tällä hetkellä tiimit itse päättävät millä työkalulla mallinnetaan tai ainakaan tietoa linjauksesta ei ole.

Tällä hetkellä tiimi itse päättää millä tavalla asioita mallinnetaan ja käytännössä kuvausten laatija itse päättää millä tavalla sen mallintaa. (H2)

Meillä on tiettyjä suosituksia tai mitä muutkin tiimit tekee tai mitä Kantapuolella on tehty ja millä on kuvattu mutta varsinaiset päätökset pystytään tekemään palvelussa ja tiimeissä itse, se mikä palvelee eniten meitä. (H4)

Ihan omalla tyylillä tehdään, vapaat kädet on tehdä. (H5)

Sovellusarkkitehti H13 oli erimieltä linjauksesta, hänen mielestään työvälinevaatimukset ovat Kanta-tason linjauksia.

Työvälinevaatimukset ovat Kanta-tason linjauksia, että mitkä on sallittuja tässä mallinnuksessa, niin mennään niiden linjausten mukaan. (H13)

Vastauksien perusteella mallinnustyökaluna eniten käytetään Confluencen Drow.io-työkalua mutta myös Microsoftin Power Point on ollut käytössä. Jälkimmäiseksi mainitusta Power Point-työkalusta monet haastateltavat mainitsivat, että työkalua tulisi oikeasti käyttää vain esittämiseen, ei mallintamiseen. Power Point mallintamistyökaluna on myös haastava esimerkiksi, kun kuvia siirtää toiseen työkaluun ja/tai Power Point-tiedosto on tallennettuna verkkolevyllä, joka ei ole kaikkien saavutettavissa ja muokattavissa.

Confluencen oma Drow io, paljon käytetään varmasti vielä power pointia. Drow io:lla pystyy piirtämään vaikka UML:llä. (H2)

Varmaan millä osaa tehdä sen, mikä on helpoin tehdä. Itse käytän power pointia. Confluencessa olisi väline tehdä, Drow io, mutta itse en ole sitä opetellut käyttämään. Mä piirtelen power pointilla, niitä mitä tarvitsee. Pitää vaan opetella käyttämään sitä, mä tiedän sen. (H5)

Laidasta laitaa käytetään, se on vapaavalintainen tapa, Drow io:ta ja vaihtelevilla välineillä. Mitään varsinaista arkkitehtuurivälinettä ei ole, esim. Archimatea. (H8)

Itse olen tehnyt drow io:lla, tukea löytyy. Confluence on helppo väline, koska sinne ei tarvitse ladata mitään vaan suoraan siellä tehdä tarvittavat. (H12)

Eniten tehdään Confluencessa, se on toimiva ratkaisu, koska kuvia pystyy muokkaamaan palvelussa, kuvat tulee heti näkymään siellä sivulla, kuka tahansa tiimiläinen voi tulla muokkaamaan tai kommentoimaan niitä helposti. (H13)

Aiemmin on ollut käytössä QPR ja Sparx Systemsin Enterprise Architect, joita ei tällä hetkellä monikaan käytä. Näitä käytetään enemmän kokonaisarkkitehtuuritasolla, joten ne eivät ole sen vuoksi palveluiden suuressa suosiossa. Muutama haastateltava kertoi käyttävänsä Microsoftin Power Pointia, mutta se on käytössä ainoastaan esittelytarkoituksessa ja yhdellä ICT-palveluvastaavalla, joka tunnisti haastattelun aikana, että hänen on siitä luovuttava. Vastauksien perusteella Atlassianin Confluence on suurimmalla osalla käytössä ja se on helppo työkalu mallintamiseen.

## 5.2 Arkkitehtuurin dokumentoinnin tarve

Seuraavaksi käsittelemme arkkitehtuurin dokumentoinnin tarvetta Kanta-palveluissa. Teeman kysymyksillä pyritään vastaamaan kysymyksiin mikä on arkkitehtuurin dokumentoinnin tarve, minkä tasoiselle arkkitehtuurin



dokumentoinnille on tarvetta sekä mistä tarve yleisesti ottaen koostuu. Yritetään löytää vastauksia myös siihen, että tarvitseeko kehittäjän ja ylläpidon töiden vuoksi tehdä erilaista dokumentaatiota ja mikä on heidän näkökulmansa dokumentoinnin tarpeeseen.

Dokumentaation tarve tulee esiin uusien henkilöiden perehdyttämisen kautta sekä henkilöstövaihdoksissa palveluiden tai tiimien kesken. Perehdytyksessä tai henkilöstövaihdoksesta johtuvan uuden henkilön saavuttua tiimiin, uudelle henkilölle annetaan mahdollisimman paljon tietoa palvelusta.

Uusille ihmisille on tottakai tärkeää, että niitä esitellään ja pystyy perehtymään sitä kautta. Ja sitten taas henkilöstön vaihdosten kautta on tärkeää, että järjestelmä on hyvin kuvattu ja ettei ole sellain, että joku sen ties tai jonkun päässä se oli, mutta nyt sitä ei enää ole. (H1)

Olen huomannut aika selkeästi, nyt kun tullut uutta ylläpito henkilöitä Kantapuolelle, perehdytyksiä kun pitää, siinä huomaa tietyt puutteet dokumentaatioissa, että se on yksi oleellinen asia. Asiat mitkä saattaa ollut pitkään talossa, tietää miten asiat toimii vaikkei sitä ole mihinkään dokumentoitu, hankalahan se on ihan uutena ottaa tuollaisia haltuun, että mistä löytyisi joku kuvauksin. (H4)

Dokumentaation tarvetta esiintyy myös viestinnässä. Dokumentaatio kommunikation välineenä on olennainen osa, jotta asioista saa muodostettua yhteisen kuvan, jota moni eri lukija ymmärtää. Tätä aihetta käsittelemme lisää luvussa 5.3.

Itse näkisin, että arkkitehtuurin dokumentaatiota käytetään viestinnän välineenä, että puhutaan tai asioista on helpompi kommunikoida sitä kautta. Ja saadaan muodostettua yhteinen kuva asioista. (H2)

Kehittämisen ja suunnitteluvaiheessa dokumentaatiota tarvitaan siihen, että ymmärretään mitä ollaan tekemässä mutta tämä pätee myös ylläpidon vaiheeseen. Dokumentaatiosta pitäisi näkyä teknisiä tietoja kuten sitä, miten järjestelmät toimivat ja mitkä niiden yhteydet toisiinsa on. Tämä tarve tunnistettiin varsinkin ylläpidollista työtä tekevien vastauksista. Mutta samankaltaiset dokumentaatiot ovat tärkeitä myös kehittäjille.

Kyllä mun mielestä kaikelle dokumenteille ja arkkitehtuurikuvauksille, oli se sitten vaikka määrittelydokumentti tai muu. Tietysti mikä se siinä hetkellä tai työn kannalta on tärkeä dokumentti. (H1)

Sieltä pitäisi näkyä kaikki meidän liityntäjärjestelmät ja niiden rajapinnat, osoitteineen, siihen liitettynä miten omakannan oma arkkitehtuuri, esim. mikä palvelu kutsuu mitäkin. Siinä pääsisi linkillä esim. arkiston rajapinta, siitä olisi linkki seuraavaan dokumentaatioon, johon se olisi jatkumo, että sieltä pääsisi eteenpäin. Olisi hyvä tavoitetila. Tavallaan et näkisi kuvatyypinen, että sieltä pystyy helposti hahmottamaan nämä eri palvelut ja niiden riippuvuudet. (H3)

Erilaisen dokumentaation tarpeeseen kehittäjän ja ylläpidon välillä ei osattu ottaa kantaa. Vastauksista huokuu se, että yleisesti ottaen perusdokumentaatio kuvaisi järjestelmää vastauksien perusteella tarpeeksi, jotta ei ole tarvetta olla

erilaista dokumentaatiota kehittäjälle ja ylläpidolle. Haastateltavat pohtivat paljon onko näillä rooleilla mitään eroa dokumentaatiota ajatellen. Mutta toisaalta vastauksista tulee esiin myös se, että ylläpidon näkökulmasta tärkeimpiä dokumentaatioita voisi olla järjestelmäarkkitehtuurin kuvaukset, joissa kuvataan palvelimet, ajoympäristöt ja rajapinnat. Kehittäjän näkökulmasta tärkeimpiä kuvauksia ovat sovellus- ja toimintoarkkitehtuurin kuvaukset, joissa on enemmän teknologiaa, ja miten järjestelmän tulee toimia. Molemmat näkökulmat tarvitsevat ristiin toisille tärkeitä dokumentaatioita, kuten niitä kuvauksia, joissa on osoitteita, portteja, rajapintoja. Loppujen lopuksi sama dokumentaatio kattaisi molempien tarpeita.

Käytännössä mä en tiedä miten ne pitäisi olla erilaista. Jos meillä olisi erikseen ylläpitiimi esim, niin sitten saattaisi olla tarvetta erilaiselle dokumentaatiolle. Nyt kun on näin, että kehitetään ja ylläpidetään sitä, en näe siinä erilaista tarvetta, että olisi erikseen ylläpidon dokumentaatiota. (H3)

Suurin osa haastateltavista kokee, että dokumentaatio on tärkeä osa heidän työtään. Dokumentaation tärkeys korostuu jokapäiväisessä työssä. Varsinkin kehittäjät tarvitsevat vaatimusmäärittelyyn liittyvää dokumentaatiota, jotta järjestelmästä tulee vaatimusten mukainen. Tärkeys tulee esiin myös työtehtävään perehtymisvaiheessa, jolloin dokumentaatiosta saa tarvittavan tiedon lähteä kehittämään järjestelmää.

Ehdottoman tärkeänä pidän omassa työssäni, ne on ihan, mä itse ajattelen arkkitehtuurikuvaukset on teknisen tason ja määrittelyn dokumentaatiota käytän joka päivä, se on mun työn perusta. (H1)

On ne tärkeitä omassa työssä, ensinnäkin ilman niitä ei välttämättä voi tehdä töitä eikä tiedä mikä on haluttu lopputulos. Tärkeää on dokumentoida koska vuoden jälkeen ei muisteta mitä on tehty ja dokumentaatiosta voi katsoa miten se on tehty. (H6)

Ylläpidossa työskentelevät asiantuntijat mainitsevat, että ongelmanselvitystilanteissa dokumentaatio on tärkeää, jotta ongelmat ratkeisivat nopeasti ja tietoa ei tarvitse itse lähteä etsimään esimerkiksi konfiguraatioista.

Kyllä ne on tosi tärkeitä, se nopeuttaa hirveästi tätä meidän ongelmanselvitystä, mitä nopeammin saadaan ongelmat selviämään, sitä tyytyväisempiä asiakkaat ovat. (H10)

No mulle ne on hyvin tärkeitä, koska taas sitten niinku, harvemmin käyn sovellustasolla, en tällä hetkellä aktiivisesti koodaan mutta aika usein kuitenkin sinne koodiin menemään ja se helpottaa kun on ne kuvaukset ympärillä, että miten tää nyt menikään, ettei tarvi joka kerta kehittäjiä kiusaamaan jokaisella asialla. aika paljon joutuu käymään myös sitä keskustelua muiden arkkitehtien kanssa, jolloin se yleiskuvaus, että sen pystyy tarkistamaan, että miten se täällä olikaan. Mun työssä se on tärkeä. (H11)

Kyllä ne ainakin jos omaa työtä mietin, kun joutuu tekemään ihan tuota kehitystäkin, niin siitä näkökulmasta nämä on ihan ehdottoman tärkeitä ja sen kokonaisuuden ymmärtäminen. Jos siihen asiaan joutuu palaamaan myöhemmin, niin niistä on tosi

paljon hyötyä, että asiat on tullut dokumentoitua joskus. asiaa on niin paljon, että ei sitä kukaan voi muistaa. (H13)

Kaksi haastateltavaa mainitsivat, että he tuntevat oman palvelunsa niin hyvin, että he eivät pidä dokumentaatiota kovin tärkeänä omassa työssään. Toki he mainitsivat, että olisi tärkeää käydä lukemassa dokumentaatiota useammin, jotta ne pysyisivät ajan tasalla.

Ajoittain tärkeinä pidän, että joskus niitä tarvii taas, saattaa mennä pitkiä ajanjaksoja, että askartelee pieniä kokonaisuuksia mutta heti kun käsitellään sellaista kokonaisuutta mikä vaikuttaa useampaan palveluun, silloin niitä joutuu arkkitehtuuriin liittyviä asioita tutkimaan. (H8)

Kaikki haastateltavat olivat sitä mieltä, että dokumentaation työstäminen on tärkeää ja hyödyllistä sekä suunnittelun alussa, että pitkin toteuttamista. Mitä ylemmän tason arkkitehtuurikuvauksesta on kyse, sen tärkeämpää on suunnitella toteutettavan järjestelmän tai järjestelmän osakokonaisuuden kuvaukset alkuvaiheessa. Suunnittelun päälinjat on suunniteltava valmiiksi, jotta alemman tason kuvauksia pystyy työstämään.

Aluksi arkkitehtuuri on lyötävä lukkoon alussa, mutta itse toteutusdokumentaatio elää sen mukana, kun sitä tehdään. (H5)

Jos sinne kovin suuria muutoksia joudutaan tekemään kehitystyön edetessä tai loppuvaiheessa, se tarkoittaa, että sinne on jouduttu isoja muutoksia tekemään. Asiat olisi voitu suunnitella paremminkin alkuvaiheessa. (H2)

### 5.2.1 Täydellinen dokumentaatio, onko sitä?

Haastateltavilta kysyttiin vastauksia siihen, millaista dokumentaatiota pitäisi tehdä, että se olisi heidän mielestään täydellistä. Vastaajista muutama kertoi, että kysymys on paha, koska täydelliseksi dokumentaatiota tai dokumentaation tekemistä ei koskaan saada. Toiseksi asiaksi he mainitsivat, että milloin katsotaan dokumentaation olevan täydellistä ja riittävää sekä myös se seikka tuli ilmi, että dokumentaation ei tarvitse olla täydellistä mutta sen tarvitsee olla riittävää, kuten sovellusarkkitehti H9 kertoi.

Sen tarvii olla riittävää, että jos puhutaan arkkitehtuuridokumentaatiosta, että asiaan pääsee kiinni ja sen pääpiirteissään tulee ymmärretyksi mitä se järjestelmäkokonaisuus on. Jos ajatellaan, että sen pitäisi olla täydellistä, sen pitäisi olla kopio lopullisesta toteutuksesta, se on liikaa vaadittu, koska meidän pitää pitää myös sitä lähdekoodia myöskin yhtenä dokumenttina. (H9)

Täydellisemmäksi dokumentaatiota voi tehdä esimerkiksi kirjoittamalla dokumentaation suoraan ohjelmakoodiin. Koodauksen käytäntöjä joudutaan muuttamaan niin, että valitaan muuttujien ja metodien nimet hyvin, jotta ne kuvaavat sitä, mitä koodin on tarkoitus tehdä. Dokumentaation teko ohjelmakoodiin on

tärkeää myös siksi, jos se ei ole muuten selkeästi ymmärrettävää koodin rakenteesta johtuen.

Näyttää siltä, että sitä [koodin kommentointia] tehdään liian vähän, mutta ei sitä välttämättä tarvitse paljoa tehdä, jos se koodi on itsedokumentoivaa. (H9)

Yritetään, että se koodi tukee sitä tietyllä tasolla loppupäätä määrittelyiden kanssa. (H11)

Sovellusarkkitehti H12 kertoi, että pyrittäisiin tekemään erilaisia työkaluja, jotka automatisoivat tiettyjä yksityiskohtaisia asioita. Näitä automatisoinnin kohteita voisi olla esimerkiksi kommenttipohjan (esimerkiksi javakielessä käytetyn javadocin) ja rajapintojen generointia suoraan, työkalu kertoisi suoraan minkälaisia tietotyyppejä otetaan vastaan ja millaisia täytyy palauttaa. Työkalun perusteella pystyisi kertomaan myös mitkä ovat pakollisia ja mitkä täytyy olla tiettyä formaattia. Tällainen työkalu helpottaisi ylläpidollista työtä huomattavasti.

Noin puolet haastateltavista kertoi, että heille ei tule mieleen mitään erityisiä tarpeita tällä hetkellä. Sen sijaan toinen puolikas haastateltavista mainitsivat, että ohjelmakoodin joukkoon tulisi dokumentoida enemmän, kuten jo aikaisemmin tuli ilmi. Toivottiin myös samanlaisia malleja ja samoja työkaluja arkkitehdeille, tietomallien parempaa kuvausta ja päivittämistä ja ettei olisi liikaa mustia aukkoja dokumentaatiossa, jotta kaikki tarvittava tieto olisi dokumentoitu.

Ei olisi semmoisia mustia aukkoja liikaa, että joku tieto on ollut jollakin henkilöllä sähköpostilaatikossa mutta hän on lähtenyt jo kaksi vuotta sitten. (H4)

Kyllä ne tarpeet liittyy tietoihin mitä palvelusovellusten välillä liikkuu, et siellä menee sanomaliikennettä ja sit siellä on niitä rajapintoja, miten niitä validoidaan tosiaan ja tämmöistä. (H8)

## 5.2.2 Yhteiset mallit

Viimeisenä tässä teemassa käsitellään Kanta-palvelujen yhteisiä malleja. Millaisia yhteisiä malleja käytetään tai ei käytetä dokumentaation työstämisessä ja mitä mieltä haastateltavat ovat ajatuksesta yhteisten mallien käytöstä, jos niitä ei nyt käytetä.

Vastauksien perusteella yhteisiä malleja ei tiedosteta olevan tai niitä ei ainakaan käytetä. Yleistä ohjeistusta yhteisten mallien käyttöön ei vastauksien perusteella tunnu olevan, joten Confluencessa olevan dokumentaation sivustorakenteiden luomiseen otetaan mallia esimerkiksi toisten palveluiden sivustorakenteista. Yhdestä vastauksesta käy ilmi, että hän ei näe tarvetta yhteisille malleille, koska "talon" sisäisesti sillä ei olisi merkitystä.

Jos sellaisia on, niin äkkiä pois ne. Jos asia ei mene talon ulkopuolelle, että nähdään vaivaa, että talon sisäisesti pitäisi olla tietyllä pohjalla. (H7)

Pintapuolisesti on omia tapoja tehdään tiimeissä, miten niitä tehdään ja millaisella rakenteella ne dokumentaatiot siellä on. (H10)

Ei taida olla oikein ohjeistusta, että tätä tulee käyttää. Mä sanoisin, että ei ole sinänsä ole yhteistä mallia. (H12)

Vaikka yhteisiä malleja ei tiedosteta tällä hetkellä olevan, tiedostetaan se, että niiden olemassaolosta voisi kuitenkin olla hyötyä ja malleja olisi toisaalta hyvä olla.

Kyllä sillä varmaan tietty etu olisi, että jollakin tasolla olisi sovittu se, että mitä arkkitehtuurikuvauksia tulisi aina olemassa, kyllä se yhteisenmukaisuutta toisi tekemiseen. Että se palvelisi sitä asiaa, kun uutta aloitetaan tekemään, tiedettäisiin suoraan suurin piirtein, että tällaiset viralliset dokumentit meidän pitäisi tästä tehdä, se palvelisi sitä vaihetta ja myös sitä vaihetta kun ihmiset liikkuu paikasta toiseen, pystyisi varmistamaan että täältä tällaiset asiat on kuvattu tällä tavalla. En tiedä saattaahan siinä jotain olla ongelmakohtiakin tulla vastaan, kun pyritään semmoiseen, että jokainen järjestelmä olisi jollakin tasolla edes kuvattu yhtenevästi. (H2)

Eipä niitä käytetä. Voisi tuoda avun meidän palveluvastaavien tasolle, että sieltä näkisi himppu tarkemmin niitä. Miten ne siellä kuvataan ja ylläpidetään, kun ne pitäisi olla automaattisia tuollaisessa konfiguraationhallintavälineessä. (H5)

Ehkä tuo puoli voisi olla sellainen, jossa olisi kehitettävää, että kun viime kesänä näihin asioihin perehdyin, yritin löytää sellaisia malleja tai pohjia, ei niitä oikein ollut helppo löytää. Osittain sen takia että silloin oli QPR käytössä, oli paljon dokumentaatiota, että käytetään sitä, mutta sitten osottautui, ettei niitä käytetä missään, eikä tulla käyttämäänäkään. Mut sellaisia malleja olisi hyvä olla. (H9)

Sovellusarkkitehti H13 kertoi, että Kanta suosituksiin kuuluu JHS179 mukainen dokumentaation teko, jossa mallipohja on tehty. Tätä pohjaa käytetään yleisemmän ja korkeamman tason kuvausten työstämisessä. JHS179 kuvauksessa ei mennä pieniin spesifeihin yksityiskohtiin. Hän myös mainitsee, ettei varmasti olisi huonoa, että mallipohja olisi enemmän.

Kanta suosituksiin kuului JHS179 mukainen dokumentaatio. Mallipohja on olemassa ja kerrottu, mitä kohtia siellä arkkitehtuuridokumentaatiossa pitäisi olla. Tätä pohjaa on hyödynnetty ison kuvan dokumentaatiossa, ihan sellainen korkean tason kuvaus. Siinä ei mennä sovellus spesifeihin kohtiin. Mun mielestä tuo JHS179 pohjalta tehty, siinä on hyvät raamit, siinä kerrotaan, että mitä pitää olla tämän tason dokumentaatiossa, siitä tykkäsin, se kyllä auttaa omaa tekemistä, kun sen tason dokkari pitää tehdä. Ja se että se on palveluiden kesken yhdenmukainen. Pienempiä kuvauksia kun on niin, tekijällä on pikkasen oma vapaus minkälaisen saa tehdä, ettei sitä ole liikaa rajoitettu. Ei varmaan huono olisi sekään, että olisi enemmän mallipohjia. (H13)

### 5.3 Arkkitehtuurin dokumentointi kommunikaation välineenä

Tämän viimeisen teeman tarkoituksena on tuoda vastauksia siihen, miten arkkitehtuuria tulisi mallintaa, jotta se voisi palvella mahdollisimman laajaa joukkoa

henkilöstöstä yli organisaatorajojen. Vastauksista yritetään tuoda esiin tietoa myös siitä, kuinka moni ymmärtäisi arkkitehtuurin dokumentaatiota niin, että niistä saadaan täysi hyöty irti.

Luvussa 5.2.1 jo todettiin viestinnällinen näkökulma dokumentaation tarpeelle. Dokumentaatio on olennaisessa osassa, jotta asioista saa muodostettua yhteisen kuvan. Ylivoimaisesti tämän teeman vastauksista tulee esiin se, että moni kaipaisi yhteisiä malleja dokumentaation tuottamiseen. Yhteisten mallien kautta olisi helppo ohjata dokumentaation tekemistä samaan suuntaan eri palveluiden ja tiimien kesken. Tämä tarkoittaisi sitä, että esimerkiksi tiimien välinen henkilöstön liikkuvuus olisi sujuvampaa ja eri tiimien työtiloista löytäisi samalla tavalla tuotetut dokumentaatiot helposti ja vaivatta. Tätä käsiteltiin myös luvussa 5.2.5.

Conflua pitäisi käyttää, sillä tavalla se on kuitenkin se meidän väline, jota käytetään, ei Power Point. Power Point on vain esittelyyn, mutta ei dokumentaation kirjoittamiseen. (H3)

Ainakin se, että missä ne on dokumentoitu tai kuvattu että ne olisivat myös helposti saavutettavissa, ihan välinetasolla. (H4)

Toinen esimerkki yhteisten mallien hyödyllisyydestä on se, että eri palveluissa ja tiimeissä työskentelevät löytävät omaan palveluunsa sidoksissa olevan palvelun dokumentaatiot nopeasti, jolloin työn tekeminen helpottuu.

Eri tiimien välillä on kiva keskustella, tietää että ne on Potilastiedon arkiston mutta kun on samantyyppisesti ja samanlaista dokumentaatiota. (H1)

Kolmas esimerkki yhteisistä malleista vielä se, että dokumentaatiot on rakennettu työtilaan samanlaisella rakenteella.

Varmaan yhteiset mallit on yksi ja se että ihan se puhtaasti mihin se dokumentaatio on tehty. Tarkoitan dokumentaation sijoituspaikkaa, että onko se mallinnusvälineessä vai onko confluencen sivulla. (H2)

Jos pysyy suhteellisen kevyenä, niin miksi ei, sit ne olis samanlailla dokumentoitu, keskeisimmät asiat olis kirjattuna ylös, että ne mitkä kiinnostaa. En tiedä siitä ymmärtämistarpeesta, osaisi etsiä nopeammin ja tietäisi missä kohtaa se suunnilleen lukee. (H7)

Tuossa varmaan yhteiset mallit voivat olla yksi avain siihen asiaan, niiden lukeminen olisi helpompaa, kun eri puolella tehdään samoja ratkaisuja ja samalta näyttäviä ratkaisuja. (H9)

Jos se on jollain tavalla yhteismitallista, samanmuotoista, rakenne olisi samanlainen, jos vaikka vaihtaisi palvelusta toiseen, suurin piirtein sen rakenteen perusteella jo löytäisi kaikki tarvittavat esimerkiksi ympäristökuvauksen, datamallin, toimintaohjeet jne. Se ainakin nopeuttaisi sellaista sisäistä siirtymää. Pohjarakenne voisi olla samanlainen, detaljit tietysti muuttuu mutta pohja olisi samanlainen. (H10)

No varmaan sitten niitä arkkitehtuuridokumentteja ja kuvauksia pitäisi käydä aina välillä yhdessä läpi, että miten mennään, pääosin ne pohjautuu aikalailla samanlaisiin ratkaisuihin. Jos nyt puhutaan dokumentaation tasosta, yhteisiä läpikäynnin ja sovi-taan yhteisiä pelisääntöjä, sellaista ei mun mielestä ei ole, ainakaan aktiivisesti ei ole, ainakaan sellaista ryhmää ei ole mikä näitä kävisi läpi. Se voisi olla hyödyllinen doku-mentaation kannalta enemmän, että ne enemmän vastaisi toisiaan. (H11)

Toisaalta yhteisiä malleja Kanta-palveluissa on yritetty ottaa käyttöön aikaisem-min mutta se ei ole tuottanut haluttua lopputulosta.

Vähän sitä QPR ite oon käyttänyt, en tiedä kuinka paljon sitä Kanta-puolella on käy-tetty. Mulla jäi sellainen mielikuva, että kuinka hyvin se oli saavutettavissa tai löydet-tävissä tai käytettävissä sellaisten tahojen toimesta ketkä ei itse mitään piirtele tai tee. Jos se on sellainen sovellus tai ei ole käyttöoikeuksia, se ei palvele tiedonjakamista. (H4)

Kyllä on olemassa työkaluja kuten archimate, mutta kuinka yleistajuisia ne kuvaukset sitten voivat olla. (H8)

Onhan siihen pyrkimystä, että on nämä mallinnustyökalut. Archimatea voi käyttää, se on ihan hyvä ja sillä voi piirtää niin monella tavalla mutta täällä ne ohjeistukset eivät ota kovin tarkkaa siihen kantaa, miten asioita pitää loppujen lopuksi esittää. Se on jo hyvä, että on tietyt notaatiot mitä käytetään, kun ne oppisi kunnolla ja kaikki ne tuntisi. Ei kovin montaa notaatiota. UML ja Archimate käytännössä on ne mitä käytetään no-taatioiden tekemisessä. Joku kolmas on myös, jota olen nähnyt käytettävän. kun viime kesänä näihin asioihin perehdyin, yritin löytää sellaisia malleja tai pohjia, ei niitä oi-kein ollut helppo löytää. Osittain sen takia että silloin oli QPR käytössä, oli paljon do-kumentaatiota, että käytetään sitä, mutta sitten osoittautui, ettei niitä käytetä missään, eikä tulla käyttämään. On kyllä ollut conflun tai arkkitehtuurin rakenteesta on joku malli olemassa, missä on kuvattu mitä toimintoarkkitehtuurissa halutaan kuvata jne. mutta se ehkä on, no joo, se tuntuu aika yleisluontoiselta, ettei se loppuviimein ohjannut sitä tekemistä. Parantamisen varaa varmasti on. (H9)

Tossa pari vuotta sitten sitä yritettiin, että olisi mallipohjia, niitä yritettiin. Ehkä vähän lähempänä sitä järjestelmädokumentaatiotasoa, yritettiin yhtenäistää mutta siitä on muutama vuosi kun sellainen harjoitus on käyty läpi. Varsinkin kun vaihdetaan väli-neestä toiseen, että lähdetkö vain siitä ilosta päivittämään, että saat sen uuteen työka-luun. Suurin osa alkuperäisistä kuvista on EA kuvina, ne orginaalit, lisenssejä löytyy, että pääsee niihin alkuperäisiin mutta niitä on paljon kopsattu ja tuotu conflun puo-lle. Tarvittaessa ne kuvat on piirretty uusiksi, jos niihin tarvitsee lisätä jotain. (H11)

Useimmista vastauksista käy ilmi myös se, että dokumentaatiota tehdessä tulee ottaa kohderyhmä huomioon eli kenelle dokumentaatiota tehdään. Kun kohde-ryhmä on otettu huomioon, kirkastuisi myös se ajatus, että mitä pitää dokumen-toida. Jokaisella eri kohderyhmällä voi olla erilaisia tarpeita, esimerkiksi liiketoi-minnan tarpeet voivat olla erilaisia kuin ylläpidon tarpeet. Liiketoiminnalle riit-tänee ylemmän tason arkkitehtuurin dokumentointi, kun taas ylläpidolle on tär-keitä tarkemman tason dokumentaatiot järjestelmästä. Yhteismitallista doku-mentaatiota tai yhtä tiettyä kuvausta arkkitehtuurista ei pysty niputtamaan yh-teen dokumentaatioon vaan on otettava huomioon eri kohderyhmät.

Varmasti on eri kohderyhmällä oma tarpeensa, minne pitää tehdä omia kuvia. Esim. liiketoimintapuolta ei kauheasti kiinnosta käytännön toteutus, ei järjestelmäarkkitehtuuri tai varsinkaan sovellusarkkitehtuurin sekvenssikaaviot, sellaiset asiat siinä tietysti on. Kyllä uskoisin että ne yhteiset tarpeet, vaikka toimintoarkkitehtuuriin liittyvät ja tietoarkkitehtuuriin liittyvät asiat ovat liiketoiminnan, määrittelijän ja toteuttajan näkökulmasta tarpeellisia. Kyllä niistä pitäisi pystyä tekemään yhdessä ymmärrettäviä kuvia, jos ne eivät ole yhdessä ymmärrettäviä, ajattelen, että voi tulla jonkinlaista käppiä. Se ei ole hyvä, että liiketoiminta ymmärtää asian eri lailla kuin toteuttaja. Pitäisi olla sellaisia, että ei tehdä erikseen kuvia. Mahdollisimman ymmärrettäviä. Laatikossa on joku hyperlinkki tai vastaava toiseen kuvaan, niitä kuvia voi linkittää sillä tavalla. Liiketoiminnalle yksinkertainen kuva, toteutukselle ihan laajuudella toinen kuva, jossa on enemmän detaljeja. (H9)

Siinä on varmaankin se, että täytyy lähteä liikkeelle hyvin korkean tason dokumentaatioilla, kertoa siitä itse palvelusta, sen relaatioista muihin isoihin palveluihin, eikä välttämättä lähteä yksityiskohtiin, lähteä sieltä tasolta ja pikkuhiljaa rikkoa pienemmiksi paloiksi, että se ymmärrys pysyy siitä. Sen takia arkkitehtuuritasoja on monia, ja sen takia että pystytään esittämään se eri tahoille, että he pystyvät vielä sen ymmärtämään. Heidän ei tarvitse tietää tai he eivät ole kiinnostuneita yksityiskohtaisimmista asioista. (H12)

Tässä korostuu taas se, että pitää miettiä se kohdeyleisö, et ei voi saman tason kuvausta esitellä johtoryhmälle mitä esittelisit omalle kehitystiimille. Että millä tasolla niistä asioista kerrotaan. Mutta jos menee liian korkealle tasolle, esittelet liian tarkkaa kuvaa, johtajat eivät ymmärrä, jos sovellustasolla ruvetaan käymään. Ja tiimille ei ole hyötyä, jos liian korkeantason kuvausta näyttää. Korostuu, että dokumentaatio pitää olla hyvin eritasoille ja pitää miettiä oma kohderyhmä kenelle sitä esittää. Yhdellä kuvalla on vaikea kertoa ihan kaikille kaikkea siitä. Ehdottomasti ymmärrettävien kuvien kautta se kertoo todella paljon siitä, hyvillä kuvilla palvelaan mahdollisimman suurta kohderyhmää. (H13)

Myös sidosryhmien ottaminen suunnitteluun mukaan nousee vastauksista esiin. Se helpottaisi dokumentaation työstämistä sillä, että saadaan sidosryhmät arvioimaan dokumentaation tasoa ja mikä tieto heille olisi optimaalista.

Miten siitä lähtisi sitten liikkeelle, pitäisi osallistaa sidosryhmät dokumentaation luotiin ja arviointiin sitten matkan varrella ketterästi. Mikä se työkalu onkin, niin pitäisi osallistaa sidosryhmät arvioimaan, onko tämä hyvä tapa, jotta kaikki saadaan tästä jotain irti. (H8)



## 6 JOHTOPÄÄTÖKSET JA POHDINTA

Tässä luvussa käsitellään tutkimuksen johtopäätökset ja niiden pohdintaa. Näiden johtopäätösten ja pohdintojen kautta saadaan vastauksia tutkimuskysymyksiin:

1. Mikä on arkkitehtuurin dokumentoinnin nykytila kehittäjän ja ylläpidon näkökulmista?
2. Mikä on arkkitehtuurin dokumentoinnin tarve kehittäjän ja ylläpidon näkökulmista?
3. Miten arkkitehtuuria tulisi mallintaa, jotta se voisi palvella mahdollisimman laajaa joukkoa henkilöstössä yli organisaatorajojen?

Ensimmäisessä kysymyskokonaisuudessa käsiteltiin arkkitehtuurin dokumentoinnin nykytilaa kehittäjän ja ylläpidon näkökulmista ja siihen liittyen kysyttiin tarkennuksena, muun muassa seuraavia asioita: kuvataanko nykytilaa, tavoitetilaa vai molempia, onko dokumentaatiot helposti löydettävissä, ovatko dokumentaatiot ajan tasalla ja riittävällä tasolla kuvattuja eli miten kokonaisuus tällä hetkellä toimii. Tähän voi todeta, että arkkitehtuurin dokumentointi eri palveluiden kesken on kirjavaa ja dokumentaatiot ovat hajautuneet moneen eri paikkaan. Dokumentointi eri palveluiden kesken on kirjavaa sen vuoksi, että palvelut ovat hyvin eri kokoisia ja toiminnallisuudeltaan erilaisia sekä tarkkaa ohjeistusta ei dokumentaation tekemiseen ole. Teknisiä arkkitehtuurikuvauksia ja muita arkkitehtuuriin liittyviä materiaaleja löytyy jokaisesta palvelusta mutta niiden määrä ja laatu vaihtelevat. Dokumentaatiota tehdään niin ohjelmakooditasolla, kuin kaikkien arkkitehtuurinäkökulmien kautta.

Arkkitehtuurin kuvaaminen eri palveluissa riippuu sen tämänhetkisestä tilasta ja toiminnallisuudesta, onko palvelu jo niin sanotusti ”vanha” vai ”uusi”. Jo useita vuosia kehityksessä olleessa palvelussa kuvataan niin nyky- kuin tavoitetilaa ja dokumentaatiota on paljon mutta se saattaa olla myös hyvin sekavaa. Dokumentaation hajautuneisuuden vuoksi dokumentaatiota on vaikea tarvittaessa löytää. Dokumentaatiota saattaa olla monessa eri paikassa ja mahdollisesti myös sellaisessa paikassa mistä sitä ei loogisesti etsisi. Teoriassa on mainittu Koskimiehen ja Mikkosen (2005) toimesta, että on kyseessä sitten pieni tai laaja ohjelmistoprojekti, selkeät ja helposti löydettävät dokumentaatiot ovat tärkeä osa kokonaisuutta. Dokumentaatioiden löytämisen vaikeus tosin vaihteli haastatteluvasta riippuen. Siihen vaikutti se, kuinka useasti henkilön täytyy etsiä arkkitehtuuriin liittyviä dokumentaatiota ja työskenteleekö hän niiden parissa usein.

Dokumentaatioiden hajautuneisuus johtaa siihen, että dokumentaatiot eivät ole ajan tasalla. Kehittäjän ja ylläpidon näkökulmasta katsottuna dokumentaatiot tulisi olla ajan tasalla ja riittävällä tasolla tuotettuja jo sen vuoksi, että toteuttaminen olisi helpompaa ja tuotannon tehtävät ja ongelmanratkaisut sujuisivat ilman suurempaa keskeytystä. Toteutuksessa ja tuotannon ylläpidossa on tärkeää, että dokumentaatiosta on tehty riittävällä tarkkuudella tehty

dokumentaatiot ja ne ovat myös ajan tasalla olevia. Dokumentaation riittävä tarkkuustaso on dokumentaation lukijan mielipide. Toiset pitävät dokumentaatiota tarpeeksi tarkkana ja toiset eivät. Tämän vuoksi olisi tärkeää, että dokumentaatiosta on tehty ohjeistusta, jotta jokaiseen palveluun saadaan tietyllä tarkkuustasolla oleva dokumentaatio.

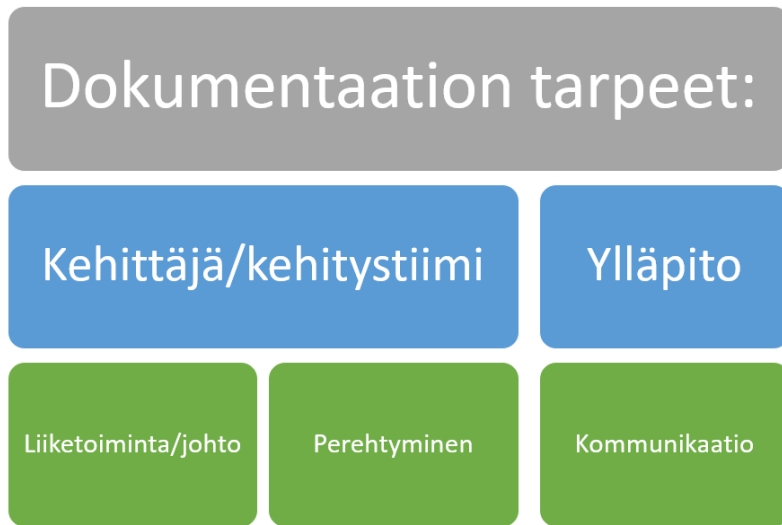
Luvussa 3.3.1 käsiteltiin Kanta-palveluiden linjausta 'Tietojärjestelmien dokumentointi. Linjauksen mukaan tietojärjestelmien tulee olla dokumentoituna koko niiden elinkaaren ajan ja dokumentaation tulee mahdollistaa tietojärjestelmän kehittäminen, ylläpito ja hallittu tuotantokäyttö. Dokumentaatiota tulee ylläpitää, jotta se on ajan tasalla koko ajan. Ylläpito on myös otettava huomioon dokumentaation toteuttamisessa, dokumentaation on oltava riittävällä tasolla kuvattu ja ylläpidon hyväksymä. Tuloksien perusteella linjaus toteutuu osittain.

Kanta-palvelu tasoisesti ei ole tiettyä pakottavaa linjausta millä mallintamis- tai piirityökalulla arkkitehtuurin kuvauksia tulisi mallintaa. Tällä hetkellä kuvauksia tehdään osittain Confluencen Draw.io-työkalulla tai muulla hyväksi havaitulla työkalulla. Kanta-palvelut on aikaisemmin yrittänyt ottaa käyttöön monia eri mallinnustyökaluja ja yrittänyt luoda yhtenäistä arkkitehtuurin dokumentointia ja kuvauksien työstämistä. Prosessin tulee jatkua ja dokumentointia tulee työstää yhtenäisemmäksi, jotta kokonaisuus tulisi eheämmäksi ja kaikille nopeasti omaksuttavaksi.

Toinen kysymyskokonaisuus käsitteli arkkitehtuurin dokumentaation tarvetta. Tarkentavina kysymyksinä kysyttiin dokumentaation tärkeyttä, onko täydellistä dokumentaatiota olemassa, millaisia yhteisiä malleja tällä hetkellä on ja käytetäänkö niitä. Dokumentaation tarve tulee vahvasti esiin uusien henkilöiden perehdyttämisen ja tiimien välisten henkilövaihdosten kautta. On tärkeää, että perehdytyksessä annetaan mahdollisimman paljon tarvittavaa tietoa, jotta perehtyminen lähtee vauhtiin ja henkilö pystyy omaksumaan tarvittavat tiedot nopeasti. Tarvetta esiintyy myös kommunikaation kautta. Dokumentaation avulla tarpeellinen tieto siirtyy kuulijalle tai dokumentaation lukijalle. Dokumentaatio on tärkeää eri sidosryhmien välisessä kommunikaatiossa. Erilaisten dokumentaatioiden tarpeellisuutta kehittäjien ja ylläpidon välillä ei varsinaisesti tullut esille. Tämä johtuu siitä, että samat dokumentaatiot hyödyttävät niin kehittäjää kuin ylläpidon parissa työskentelevää asiantuntijaa. Kehittäjät ja ylläpidon asiantuntijat kokevat dokumentaatioiden olevan hyvin tärkeitä heidän työssään. Dokumentaatioiden avulla he pystyvät keskeytyksessä työstämään omaa aluettaan palvelun kehittämisessä. Täydellistä dokumentaatiota ei tule koskaan olemaan mutta siihen tulee pyrkiä. Hyvä dokumentaatio on tarpeellista, riittävää, ajantasaista ja helposti löydettävissä. Täydellisemmäksi sitä voi saada myös kirjoittamalla dokumentaatiota suoraan ohjelmakoodiin ja tekemällä sellaisia työkaluja, jotka automatisoivat joitakin dokumentaatioon liittyviä alueita.

Kanta-tasoisesti on yhteisiä malleja olemassa mutta niitä ei kuitenkaan koettu käytettävän. Tämä johtuu siitä, että niiden olemassaoloa ei useinkaan tiedostettu. Tästä oletettavasti johtuu dokumentaation tason erilaisuus ja erilaiset käytännöt dokumentaatiota tehdessä. Yhteiset mallit ja ohjeistus niiden käytöstä

toisivat samankaltaisuutta dokumentaatioiden työstämiseen ja olemassaoloon. Kuviossa 8 on esitelty dokumentaatioita tarvitsevia kohderyhmiä.



KUVIO 8 Dokumentaation tarpeet

Dokumentaatioita Kanta-palveluissa tarvitsee niin kehittäjät, ylläpito kuin koko kehitystiimi. Tarpeita on myös liiketoiminnalla, organisaation johdolla sekä uusiin työtehtäviin tutustuvilla henkilöillä. Kommunikaatio sidosryhmien välillä tarvitsee myös dokumentaatiota.

Viimeinen kokonaisuus käsitteli arkkitehtuurin dokumentaatiota kommunikaation välineenä ja miten dokumentaatiosta voi saada kaikki hyödyt irti. Viestinnällistä ja kommunikatiivista tarvetta on sivuttu myös arkkitehtuurin dokumentoinnin tarpeen -osion tutkimustuloksissa ja johtopäätöksissä. Dokumentaatio kommunikaation välineenä on tärkeä osa tietojärjestelmäkehitystä eri sidosryhmien välillä. Avain tähän voi olla yhteiset mallit, joita tulisi käyttää Kanta-palveluissa. Yhteisten mallien kautta olisi helppo ohjeistaa miten arkkitehtuurin dokumentaatiota tulisi työstää ja millaisia dokumentaatioita tarvitaan ja kaivataan. Yhteiset mallit helpottavat jo aiemminkin mainittua uusien henkilöiden perehtymistä ja henkilöstövaihtuvuutta, jos dokumentaatiot ovat niin sanotusti samalla kaavalla tehtyjä. Yhteiset mallit helpottavat myös toisen palvelun dokumentaatioiden ymmärrettävyyttä, jolloin oma työtehtävä helpottuu ja tietyt dokumentaatiot ovat helposti löydettävissä. Dokumentaatiota tehdessä tulee ensisijaisesti ottaa kohderyhmä huomioon eli kenelle dokumentaatiota tekee, kuka on dokumentaation lukija. On erilaisia kohderyhmiä, kuten liiketoiminta tai sovelluskehittäjä, joille samasta aiheesta on työstettävä esimerkiksi erilainen esitys tai arkkitehtuurikuvaus. Viestintä ja kommunikaatio paranee myös osallistamalla sidosryhmiä kehitykseen, esimerkiksi arvioimaan mikä on tarpeellista dokumentaatiota tai riittävällä tasolla tehtyä dokumentaatiota.

## 6.1 Tutkimustulosten pohdintaa

Tässä luvussa käsitellään vielä kerran läpi tutkimuksesta nousseet kohdat. Arkkitehtuurin dokumentaation hallinta ei ole välttämätöntä mutta se on suotavaa. Vastauksien perusteella suurimmaksi huolenaiheeksi tutkimuksen tekijän mielestä nousee se, että dokumentaatiot ovat hajautuneet moneen eri paikkaan, tällä hetkellä dokumentaatiota voi olla esimerkiksi:

- Sinetissä
- Confluencessa
- verkkolevyllä
- mallinnus- tai piirtotyövälineellä

Hajautuneisuuden vuoksi dokumentaatiota muodostuu turhiin paikkoihin, ja dokumentaatiot eivät välttämättä ole ajan tasaisia. Dokumentaatioiden dokumentaatiot eli dokumentaatioiden duplikaatit tulisi saada poistettua tai ainakin minimoitua tällaisten olemassaolo, jotta päivittämisen tarvetta ei tulisi niin paljoa kuin mitä sitä tällä hetkellä on. Koska nykysuuntaus Kanta-palveluissa on, että dokumentaatiot ovat ja dokumentaatioita tulee työstää Confluencen työtilassa, tulee tätä ohjeistusta kertoa eteenpäin palveluille.

Dokumentaation ajantasaisuus ongelmaa voisi lähteä ratkaisemaan sillä, että dokumentaation päivittämiseen varataan riittävästi aikaa. Tämä voi tosin olla hankalaa, koska usein kehittäminen on hyvin hektistä, jolloin dokumentaation päivittämiseen ei jää riittävästi aikaa, kun pitää jo siirtyä tekemään toista kehitystehtävää. Dokumentaation pitäminen ajan tasalla on hyvin tärkeää myös auditointeja ajatellen. Jos dokumentaatiot eivät ole ajan tasalla, se vaatii myöhemmin paljon ylimääräistä työtä. Hektisyys tosin taitaa olla ketterän kehityksen yleinen ongelma mutta siihen tulisi puuttua. Tarkempaa teknistä dokumentaatiota voi olla itse ohjelmakoodissa ja/tai sovellusarkkitehtuurissa sekä teknisissä vaatimuksissa.

Mitkä ovat sitten syitä olla dokumentoimatta? Yksi iso ongelma olla dokumentoimatta tuntuisi olevan siinä, että tingitään sellaisista töistä, jotka ovat työläitä ja tämän vuoksi ongelman korjaamiseen puututaan liian myöhään. Esimerkki tästä liian myöhään puuttumiseen on jo aikaisemmin mainittu auditointi ja sitä varten dokumentaatioiden päivittäminen, vaikka sen pitäisi olla jatkuvaa tekemistä. Vähäinen dokumentaatio vaikuttaa myös ylläpidon töihin ja uusien työntekijöiden perehtymiseen.

Dokumentaatio on tärkeää niin toteutuksen alussa kuin pitkin toteuttamista. Pitkin toteuttamista kuvaukset muuttavat muotoaan, kun kehityksen aikana tulee täsmennyksiä, muutoksia tai korjauksia. Olisi tärkeää, että suunnittelun alussa tärkeimmät kuvaukset on kuvattu ja suunniteltu paremmin, jotta myöhemmän vaiheen muutoksia ei tulisi liian paljon. Myöhemmässä vaiheessa tehdyt muutokset tarkoittavat sitä, että järjestelmään on jouduttu tekemään isoja muutoksia, josta tulee myös mahdollisesti isoja kustannuksia (Boehm, 1984). On

järkevää ja taloudellista ajattelua, että havaitaan ja korjataan mahdolliset virheet suunnittelun alussa.

Dokumentaation merkitys korostui sidosryhmien välisessä kommunikaatiossa. Arkkitehtuurin dokumentointi tulisi palvella sidosryhmiä ja olla niiden tukena. Rajapinnoissa tulee olla toimivaa dokumentointia, joka on tuotettu riittäväällä tasolla sidosryhmää ajatellen.

Tietojärjestelmien kasvaessa ja monimutkaistuessa, on arkkitehtuurin entistä parempi hallinta avain asemassa. Mikropalveluita ajatellen on myös hyvin oleellista, että dokumentointia aloitetaan suunnittelemaan varhaisessa vaiheessa, että mikä meillä toimii ja mikä ei. Mikropalveluissa tulee olemaan avoimia rajapintoja ja se aiheuttaa myös muutoksia dokumentaation työstämiseen.

Dokumentaation hallinta on tärkeää ja sitä voi tehdä joko kehitystyön aikana tai sen jälkeen. Mutta dokumentaatiota täytyisi tehdä. Voidaanko ylläpidon vastuulle jättää muistuttaminen dokumentaation tärkeydestä tai puuttuvista kohdista. Jos aika ei riitä ja dokumentaatiota pitäisi tehdä, punnitaan kahta asiaa vastaan. Pitäisi etsiä perustetta sille, miksi voidaan jättää dokumentaation tekemättä ja mitä siitä seuraa. Voiko olla taloudellista jättää dokumentointia tekemättä? Tutkijan mielestä tähän on vain kaivettava resurssit, jotta dokumentaatio saadaan kuntoon ja näin ollen työt muuttuvat tehokkaammiksi.

Yleisesti ottaen vastauksista voi päätellä, että dokumentaation tulee olla niin kattavaa, selkeää ja helposti ymmärrettävää, että uusi henkilö tai pitkään projektista poissa ollut henkilö saa kokonaisuudesta kiinni mahdollisimman pian ja pystyy työskentelemään tehokkaasti lyhyen perehtymisen perusteella. Dokumentaation tulee olla laajaa ja yksinkertaista mutta riittävää, ei yhteen kuvaan puristettua tietoa, järjestelmä ja kokonaisuus tulee olla kuvattu eri tasoilla ja eri kohderyhmille. Seuraavanlaista toimintoa toivottiin, että jostain asiasta saisi lisätietoa "laatikkoa" napsauttamalla. Kuva kertoo paljon enemmän kuin pitkä tarinanomainen teksti dokumentaationa ja lisätietona saatavia tarkempia tietoja olisi hyvä saada näkyviin, jos niitä tarvitsee. Jokaisella dokumentaatiolla pitää olla käyttötarkoitus ja dokumentaation pitää olla helposti löydettävissä.

Dokumentaation parantaminen olisi aloitettava ihan perustasolta. Confluenceen tulisi määritellä kansiorakenne, joka noudattaa samaa rakennetta jokaisessa palvelussa. Dokumenteille tulisi määritellä yhteinen nimeämiskäytäntö, jotta Confluenceen -työtilassa olevat dokumentit olisivat samanlaisilla nimetty ja näin ollen helposti löydettävissä. Tulisi laatia myös ohjeistus, millä tyyllillä dokumenttien tekemisen käytännöt pysyisivät yhtenäisinä. Voisiko myös dokumenttien löytämiseen olla apuna nimeämiskäytäntö ja sen perusteella myös Confluenceen hakutoiminnallisuus pystyisi löytämään käyttäjän hakemat dokumentit helposti.

Vaikka ketterässä kehityksessä dokumentaation työstäminen saattaa jäädä kiireiden vuoksi vähäisemmäksi, on dokumentaatio silti tärkeää ja kiirettä voi jollain tasolla myös voittaa. Dokumentaation hallinta tulisi ketterässäkin kehityksessä olla välttämätöntä, jotta välttyään suuremmilta ongelmilta, kuten aikataulumuutoksilta tai laadun heikkenemiseltä. Käytännön hyötyä ajatellen on tärkeää, että tekeminen olisi mahdollisimman sujuvaa ja kustannustehokasta.

Musta laatikko on nyt avattu ja voidaan lähteä etenemään parempaa dokumentaatiota kohti. Dokumentaatio on aina näkymättömästi tai näkyvästi läsnä.

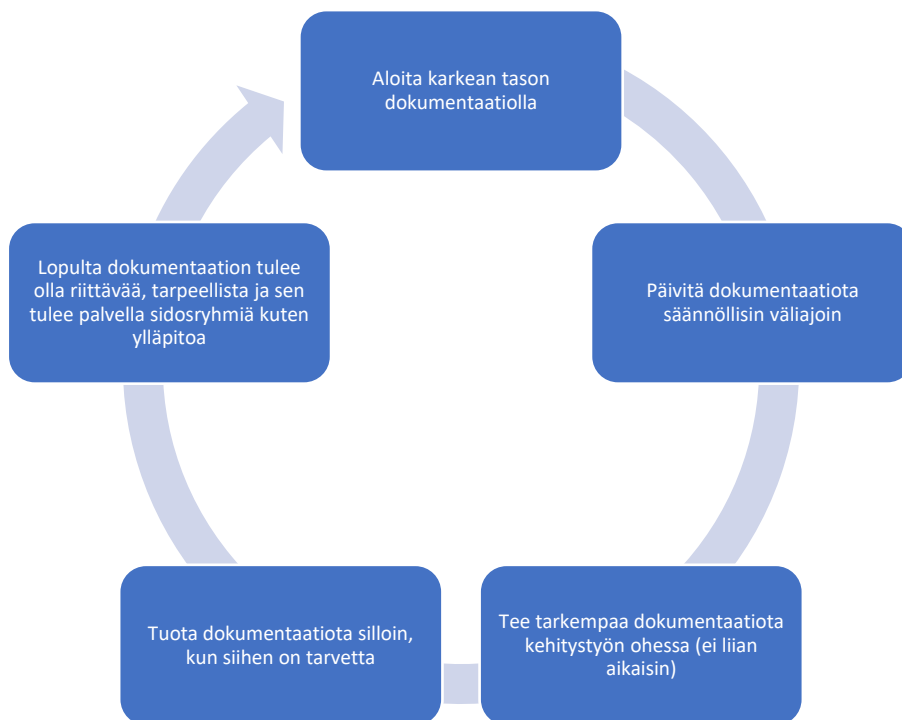
Taulukko 3 havainnollistaa SWOT-analyysin perusteella dokumentaation parantamiseen liittyviä vahvuuksia, heikkouksia, mahdollisuuksia ja uhkia.

TAULUKKO 3 Dokumentoinnin parantamisen SWOT-analyysi

Vahvuudet	Heikkoudet
Ohjeistus, missä dokumentaatio tulisi sijaita Toimivat työtilat työskennellä (esimerkiksi Confluence)	Dokumentoinnin taso Ketterä kehitys Ajanpuute
Mahdollisuudet	Uhat
Yhtenäiset toimintamallit Ymmärryksen lisääntyminen Kommunikaation lisääntyminen Nopea perehtyminen Nopea ongelmanselvitys	Palataan vanhaan tapaan työskennellä Ei oteta muutosta vastaan

Vahvuudeksi nousi esiin ohjeistus eli missä dokumentaatio tulisi sijaita sekä toimivat työtilat esimerkiksi Confluencen -työtilat. Dokumentaation looginen sijainti helpottaa dokumentaatioiden löytämistä ja toimivat Confluencen työtilat auttavat dokumenttien loogista sijoittelua sekä arkkitehtuurin liittyvien kuvauksien työstämistä. Heikkouksiksi voidaan mainita dokumentaation taso, ketterä kehitys sekä ajanpuute. Dokumentaation taso tarkoittaa sitä, että eri henkilöt kirjoittavat dokumentaatioita eri tyylillä ja siitä muodostuu tasoeroja. Ketterä kehitys sen sijaan saattaa aiheuttaa puutteita dokumentaation työstämisessä esimerkiksi kiireen vuoksi. Ajanpuute voi olla heikkous niin perinteisessä kuin ketterässä kehityksessä. Dokumentoinnin parantamisen mahdollisuuksiksi nousi yhtenäiset toimintamallit, ymmärryksen ja kommunikaation lisääntyminen, nopea perehtyminen ja nopea ongelmanselvitys. Yhtenäiset toimintamallit helpottavat muun muassa dokumentaation löytämistä ja dokumentoinnin työstämistä. Yhtenäisistä toimintamalleista seuraa ymmärryksen ja kommunikaation lisääntymistä. Nopea perehtyminen tarkoittaa perehtymistä niin kehitystiimien kesken kuin uusien henkilöiden perehtyminen. Nopea ongelmanselvitys tarkoittaa esimerkiksi ylläpidon työtä tuotannon ongelmien selvityksessä. Uhkaksi nousee kaksi asiaa, joita ovat seuraavat: vanha tapa työskennellä ja muutosvastarinta. Vaikka dokumentaatiota kehitetään parempaan suuntaan, saatetaan silti palata vanhaan tuttuun tapaan työskennellä, koska se on helpointa tapa. Muutoksia ei oteta vastaan, joka vaikeuttaa yhtenäisten toimintatapojen toteuttamisen.

Tutkimuksen tuloksena syntyi uusi lyhyehkö arkkitehtuurin dokumentaation kehittämisen prosessi, joka antaa mahdollisuuden dokumentoinnille myös ketterässä kehitystiimissä. Kuviossa 8 on kehämäisesti esitelty dokumentaation toteuttaminen kehitystiimissä.



KUVIO 9 Dokumentoinnin polku kehitystiimissä

Ensimmäisenä on aloitettava karkean tason dokumentaatio, jotta kehityksen pääpiirteet saadaan lueteltua. Toiseksi dokumentaatiota on päivitettävä säännöllisin väliajoin. Seuraavaksi olisi työstettävä tarkempaa dokumentaatiota kehitystyön ohessa. Neljäntenä kohtana on tuottaa dokumentaatiota tarpeeseen. Esimerkiksi jos kehittäjä tai ylläpitäjä tarvitsee jotain tiettyä dokumentaatiota, on dokumentaation puuttumiseen vastattava. Viimeisenä kohtana kehässä on mainittu, että lopulta dokumentaation tulee olla riittävää, tarpeellista ja sen tulee palvella kaikkia sidosryhmiä. Sidosryhmien dokumentaatiot voivat olla erilaisia tai eritasolla kuvattuja sidosryhmästä riippuen.

## 6.2 Tutkimuksen arviointi

Tutkimuksen aineistonkeruumenetelmänä oli haastattelu, joka oli tarkoituksenmukaista ja tilanteeseen sopiva tutkimuksen tutkimustulosten vuoksi. Aineistonkeruun voisi tehdä myös kyselytutkimuksella, jossa olisi mahdollisesti saatu tarkempaa dataa mutta ennen sitä tutkimuksen tekijän olisi pitänyt olla perehtyneempi aiheeseen, jotta olisi osannut valita oikeat ja tarkemmat kysymykset. Tämän tutkimuksen jälkeen voisi tehdä tarkemman tutkimuksen erilaisella menetelmällä, koska pohjatieto aiheeseen liittyen on nyt kerätty.

Haastateltavia tässä tutkimuksessa oli 13 henkilöä, joista osa oli sovelusarkkitehtejä, osa sovelluskehittäjiä tai ylläpidon asiantuntijoita. Tällä määrällä saavutettiin tutkimusaineiston kylläisyys mutta olisiko tärkeitä haastateltavia vielä ollut mahdollista saada osallistumaan tutkimukseen. Tämä pohdinta lähti

siitä, että sähköpostikyselyyn vastasi 13 henkilöä 34 henkilöstä. Tosin sähköpostikysely lähetettiin 2 kertaa, joten oletan, että tutkimukseen osallistujilla oli tietämystä arkkitehtuurin dokumentaatiosta. Ne henkilöt, jotka eivät vastanneet, eivät välttämättä olisi antaneet uutta tietoa tutkimustulosten kannalta.

Tutkimuksen tekijänä työskentelen myös Kanta-palveluissa Omakanta-palvelun määrittelijänä ja scrum masterina. Oma ennako-oletukseni arkkitehtuurin dokumentaatioon oli Omakannan näkökulmasta, mutta se ei haitannut tutkimustulosten johtopäätöksiä. Oma näkemykseni laajentui tutkimusta tehdessä koko Kanta-palveluihin kohdistuvaksi ja siitä on hyötyä niin Omakannan dokumentaatiota työstäessä kuin myös Kanta-palveluiden dokumentaatiota kehittäessä.

Aineiston erilainen käsittely voisi antaa toisenlaisen tutkimustuloksen. Tutkimustuloksia voisi luokitella tai teemoittaa tarkemmin ja etsiä eriävyyksiä ja ristiriitoja kohdistuen johonkin tiettyyn teoriasta nousevaan aiheeseen. Kyseessä oli Kanta-palveluihin kohdistuva tapaustutkimus, joten sen tuloksia ei välttämättä ole tarpeen yleistää muissa yhteyksissä. Tosin tutkimustulokset antoivat lisäinformaatiota esimerkiksi dokumentaation toteuttamiseen ketterässä kehityksessä. Tutkimustulokset toivat uutta näkökulmaa ja ymmärrystä dokumentaation merkitykselle ja tärkeydelle Kanta-palveluissa.

### 6.3 Aiheita jatkotutkimukseen

Tutkimuksesta on tarkoituksenmukaisesti rajattu pois tarkempi katsaus ketterään kehittämiseen, koska tässä tutkimuksessa oli tarpeeksi tutkittavaa siinä, että saatiin kerättyä aineistoa dokumentoinnin nykytilasta, tarpeesta ja kommunikaatiosta. Ketterässä kehitystiimissä tuotettava dokumentaatio ja miten ketterä kehitys vaikuttaa kehitystiimin toimintaan yleisesti, alkoi kiinnostamaan tutkijaa, joten jatkotutkimusta olisi erittäin hyvä tehdä. Tutkimuksesta voidaan nostaa seuraavaa analysointikierrosta varten näitä teemoja:

- Dokumentointi ketterässä kehityksessä
- Dokumentointi prosessin kehittäminen Kanta-palveluille



## 7 Yhteenveto

Tässä luvussa tutkimuksesta koostetaan yhteenveto. Tutkimus koostui teoriaosuudesta ja käytännön osuudesta. Johdanto kappaleessa käytiin läpi tutkimuksen tarkoitusta, tavoitteita ja tiedonhakumenetelmiä. Teoriaosuudessa kerrottiin lyhyesti kokonaisarkkitehtuurista ja ohjelmistoarkkitehtuurista, kokonaisarkkitehtuurin viitekehyksistä sekä arkkitehtuurin dokumentoinnista, niin kommunikaatiota kuin ketterää kehitystä sivuten, sekä kerrottiin millaisia mallinnus- ja piirtotyökaluja, on mahdollista käyttää. Lisäksi teoriaosuudessa kerrottiin lyhyesti Kanta-palveluista sekä Kanta-palveluihin liittyvästä kokonaisarkkitehtuurista ja Kanta-arkkitehtuurista. Mainittiin myös millaista arkkitehtuurin kehittäminen ja dokumentointi on Kanta-palveluissa.

Neljännessä luvussa käytiin läpi tutkimusmenetelmiä, joihin sisältyi tutkimusotteen, tiedonkeruumenetelmien perusteluita, haastateltavien valintaa, aineiston analysoinnin toteutusta ja luotettavuutta. Tutkimus toteutettiin laadullisena tapaustutkimuksena sen vuoksi, että haluttiin keskittyä arkkitehtuurin dokumentaation prosessiin ja sen ymmärtämiseen kehittäjien ja ylläpidon asiantuntijoiden kautta. Tutkimusaineisto kerättiin puolistrukturoidulla teemahaastattelulla. Haastateltavia oli yhteensä 13 henkilöä Omakanta-palvelusta, Omatietovarannosta, Resepti-palvelusta, Potilastiedon arkistosta, Sosiaalihuollon arkistosta ja Kuva-aineistojen arkistosta. Tutkimustulokset analysoitiin sisällönanalyysillä. Haastatteluaineisto käsiteltiin värikoodauksen ja teemoittelun avulla, jolla saatiin esiin samankaltaisuuksia ja eriäväisyyksiä.

Viidennessä luvussa käsiteltiin laajasti tutkimuksen tuloksia teemoittain. Teemoja oli arkkitehtuurin dokumentaation nykytila, arkkitehtuurin dokumentoinnin tarve ja arkkitehtuurin dokumentaatio kommunikaation välineenä. Näiden teemojen avulla saatiin tietoa arkkitehtuurin dokumentaation tilasta ja tarpeesta tällä hetkellä.

Kuudennessa luvussa tutkimustuloksista tehtiin johtopäätöksiä ja pohdintaa. Tällä hetkellä arkkitehtuurin dokumentointia kehitetään aktiivisesti ja on mahdollista, että dokumentointiin tulee hienoisia muutoksia. Arkkitehtuurin dokumentointiin Kanta-palveluissa kiinnitetään jatkossa enemmän huomiota. Dokumentaation kehittäminen antaa monia hyötyjä niin kehitystiimien kuin sidosryhmien kommunikaatiossa ja yhteistyössä. Dokumentaation kehittäminen palvelee myös perehtymistä ja ylläpidollista nopeaa ongelmanratkaisua.

## LÄHTEET

- Abraham, R. (2013). Enterprise architecture artifacts as boundary objects - A framework of properties. Proceedings of the 21st European Conference on Information Systems. University of St. Gallen. 1-12.
- Abrahamsson, P., Babar, M. A. & Kruchten, P. (2010). Agility and architecture: Can they coexist?. IEEE Software, vol. 27, no. 2, 16-22.
- Agile Alliance. (2001). Agile Manifesto. Haettu osoitteesta <http://agilemanifesto.org/>
- Ayaß, Ruth. (2015). Doing Data: The status of transcripts in Conversation Analysis. Discourse Studies. 17(5), 505-528.)
- Boehm, B. W. (1984). Software Engineering Economics. IEEE Transaction on Software Engineering, vol. se-10, no. 1, 4-21.
- Cakir, C., Cetin, F., Savasci, M. & Findik, O. (2018). Software Architecture Documentation in Agile. International Conference on Advanced Technologies, Computer Engineering and Science(ICATCES'18), May 11-13, 2018 Saranbolu, Turkey (1-3).
- Clements, P, Garlan, D., Little, R., Nord, R. & Stafford, J. (2003). Documenting software architectures: views and beyond. Proceedings of the 25th International Conference on Software Engineering. IEEE.
- Dagenais, B., Ossher, H., Bellamy, R. K. E., Robillard, M. P. & De Vries, J. P. (2010). Moving into a New Software Project Landscape. Teoksessa Processing of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1 (275-284). New York: ACM.
- Finlex. (2019). Laki julkisen hallinnon tiedonhallinnasta 906/2019. Haettu osoitteesta [Laki julkisen hallinnon tiedonhallinnasta 906/2019 - Säädökset alkuperäisinä - FINLEX®](#)
- Finlex. (2011). Laki julkisen hallinnon tietohallinnon ohjauksesta 634/2011. Haettu osoitteesta [Laki julkisen hallinnon tietohallinnon ohjauksesta 634/2011 - Säädökset alkuperäisinä - FINLEX®](#)
- Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley Professional. England.
- Hirsjärvi, S. & Hurme, H. (2011). Tutkimushaastattelu. Teemahaastattelun teoria ja käytäntö. Gaudeamus Oy.

- Hirsjärvi, S., Remes, P. & Sajavaara, P. (2012). *Tutki ja kirjoita*. (18. uud. painos). Helsinki: Tammi.
- Hosiaislouma. E. (2017). Kokonaisvaltainen kehittäminen. ArchiMate - elementtien osajoukko. Haettu 26.5.2022 osoitteesta ArchiMate - elementtien osajoukko - Kokonaisvaltainen kehittäminen (hosiaislouma.fi)
- Hosiaislouma. E. (2021). Kokonaisvaltainen kehittäminen. ArchiMateR peruselementit. Haettu 26.5.2022 osoitteesta ArchiMate peruselementit - Kokonaisvaltainen kehittäminen (hosiaislouma.fi)
- Kanta. (2021a). Mitä Kanta-palvelut ovat? Haettu 29.10.2021. <https://www.kanta.fi/mita-kanta-palvelut-ovat>
- Kanta. (2021b). Lainsäädäntö. Haettu 29.10.2021. <https://www.kanta.fi/lainsaadanto>
- Kanta. (2021c). Yhteistyökumppanit. Haettu 29.10.2021. <https://www.kanta.fi/yhteistyokumppanit>
- Kanta. (2021d). Omakanta. Haettu 1.11.2021. <https://www.kanta.fi/omakanta>
- Kanta. (2021e) Omatietovaranto. Haettu 1.11.2021. <https://www.kanta.fi/hyvinvointitiedot>
- Kanta. (2021g). Potilastiedon arkistoa käyttävät palvelunantajat. Haettu 2.12.2021. <https://www.kanta.fi/potilastiedon-arkistoa-kayttavat-yksikot>
- Kanta. (2021f) Reseptit. Haettu 1.11.2021. <https://www.kanta.fi/reseptit>
- Kanta. (2021h). Terveystiedot. Haettu 1.11.2021. <https://www.kanta.fi/terveystiedot>
- Kanta. (2021i). Sosiaalihuollon asiakastiedon arkistoa käyttävät organisaatiot. Haettu 29.4.2022. <https://www.kanta.fi/sosiaalihuollon-asiakastiedon-arkistoa-kayttavat-organisaatiot>
- Kanta. (2021j). Kanta-arkkitehtuuri. Haettu osoitteesta 29.10.2021. <https://www.kanta.fi/jarjestelmakehittajat/kanta-arkkitehtuuri>
- Kanta. (2021k). Järjestelmäkehittäjät. Haettu 29.5.2022 osoitteesta <https://www.kanta.fi/jarjestelmakehittajat>
- Kanta. (2022a). Kuva-aineistojen arkisto. Haettu 29.5.2022 osoitteesta <https://www.kanta.fi/ammattilaiset/kuva-aineistojen-arkisto>
- Kanta. (2022b). Potilastiedon arkisto. Haettu 29.5.2022 osoitteesta <https://www.kanta.fi/ammattilaiset/potilastiedon-arkisto>

- Koskimies, K & Mikkonen, T. (2005). Ohjelmistoarkkitehtuurit. Talentum Media Oy. Helsinki.
- Kotusev, S. (2016). The History of Enterprise Architecture: An Evidence-Based Review. *Journal of Enterprise Architecture - Volume. 12 (1)*, 29-37.
- Kotusev, S. (2018). Enterprise Architecture: A Reconceptualization Is Needed. *Pacific Asia Journal of the Association for Information Systems*, 10(4), Article 4.  
<https://journal.ecrc.nsysu.edu.tw/index.php/pajais/article/view/489>
- Kruchten, P., Lago, P. & van Vliet, H. (2006). Building Up and Reasoning About Architectural Knowledge. Vol. 4214. Berlin, Heidelberg: Springer Berlin Heidelberg, 43-58.
- Laine, M., Bamberg, J., & Jokinen, P. (2007). Tapaustutkimuksen taito. Gaudeamus Helsinki University Press.
- Lankhorst, M. (2017). Enterprise Architecture at Work: Modelling, Communication and Analysis. Berlin, Heidelberg, Germany: Springer.
- Liang, P. & Avgeriou, P. (2009). Tools and Technologies for Architecture Knowledge Management. Chapter 6. University of Groningen. The Netherlands.
- OMG. (2017). OMG® Unified Modeling Language (OMG UML®) Version 2.5.1. Haettu 26.5.2022 osoitteesta <https://www.omg.org/spec/UML/2.5.1>
- OMG. (2012-2019). ArchiMate® 3.1 Specification (OMG UML®). Haettu 26.5.2022 osoitteesta [https://pubs.opengroup.org/architecture/archimate3-doc/chap03.html#\\_Toc10045289](https://pubs.opengroup.org/architecture/archimate3-doc/chap03.html#_Toc10045289)
- OMG. (2013). Business Process Model and Notation™ (BPMN™). Graphical notations for business processes. Haettu 26.5.2022 osoitteesta <https://www.omg.org/bpmn/index.htm>
- Pulkkinen, M. (2006). Systemic management of architectural decisions in enterprise architecture planning. Four dimensions and three abstraction levels. *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on (8)*. IEEE.
- Puusa, A. & Juuti, P. (2020). Laadullisen tutkimuksen näkökulmat ja menetelmät. Gaudeamus Oy.
- Pyle, T. (2020). Guidelines for Creating Effective Software Design Documentation. Intercom, 22-26. [www.stc.org](http://www.stc.org).

- Richards, M. & Ford, N. (2021). Fundamentals of Software Architecture. Fourth Release. O'Reilly Media, Inc. United States of America.
- Schwittek, W. & Eicker, S. (2010). Communicating Architectural Knowledge: Requirements for Software Architecture Knowledge Management Tools. Springer-Verlag Berlin Heidelberg, 457-463.
- Suomidigi. (2021a). Julkishallinnon suositukset (lakkautetut). Haettu 26.10.2021. <https://www.suomidigi.fi/ohjeet-ja-tuki/jhs-suositukset-lakkautetut>
- Suomidigi. (2021b). Kokonaisarkkitehtuuri. Haettu 26.10.2021. <https://www.suomidigi.fi/ohjeet-ja-tuki/kokonaisarkkitehtuuri>
- Suomidigi. (2021c). JHS 179 Kokonaisarkkitehtuurin suunnittelu ja kehittäminen. Haettu 27.10.2021. <https://www.suomidigi.fi/ohjeet-ja-tuki/jhs-suositukset/jhs-179-kokonaisarkkitehtuurin-suunnittelu-ja-kehittaminen>
- Terveyden ja hyvinvoinnin laitos. (2022). Sote-arkkitehtuuti. Haettu 5.1.2022. <https://thl.fi/fi/web/tiedonhallinta-sosiaali-ja-terveysalalla/tiedonhallinnan-ohjaus/sote-kokonaisarkkitehtuuri>
- The Open Group. (2018). TOGAF® Standard, Version 9.2. Haettu 10.10.2021. <https://pubs.opengroup.org/architecture/togaf92-doc/arch/index.html>.
- Tuomi, J. & Sarajärvi, A. (2018). Laadullinen tutkimus ja sisällönanalyysi (Uudistettu laitos). Kustannusosakeyhtiö Tammi.
- Valtiovarainministeriö. (2021a). Tiedonhallintalautakunnan tehtävät. Haettu 13.11.2021. <https://vm.fi/tiedonhallintalautakunnan-tehtavat>
- Valtiovarainministeriö. (2021b). Tiedonhallintalautakunnan suositukset. Haettu 13.11.2021. <https://vm.fi/suosituks>
- Zachman International Enterprise Architecture. (2008). About the Zachman Framework. Haettu 24.10.2021. <https://www.zachman.com/about-the-zachman-framework>.
- Zachman, J. A. (1987). A framework for information systems architecture. IBM systems journal, 26(3), 276-292.

## LIITE 1 SÄHKÖPOSTIKYSELY HAASTATELTAVILLE

Hei!

Teen pro gradu -tutkimusta Jyväskylän yliopistossa tietojärjestelmätieteessä. Tutkimus käsittelee Kanta-palveluita aiheenaan ”Sote-arkkitehtuurin dokumentointi kehittäjän ja ylläpidon näkökulmasta”. Tiedustelen, voisitko osallistua haastatteluun aiemmin mainitusta aiheesta tammi-helmikuussa 2022. Haastattelu kestää noin 45–60 minuuttia. Kerään tutkimusta varten alan ammattilaisten näkemyksiä, joiden pohjalta pyrin luomaan kokonaiskuvan Kanta-palveluiden arkkitehtuurin dokumentoinnin tilasta. Laitan lisätietoa aiheesta ja kysymyksistä myöhemmin, kun olen saanut myönteisen vastauksen sinulta.

Kaikki vastaukset tulen käsittelemään anonymisti, joten haastatteluun osallistuneiden nimiä en tule julkaisemaan tutkimuksessa.

Pyydän vastausta mahdollisimman pian, kuitenkin viimeistään 20.1.2022.

Ystävällisin terveisin,

Tiina Mankonen

Koska olen tällä hetkellä äitiyslomalla Omakanta-palvelun määrittelystä ja palaan töihin 1.3.2022. Olethan yhteydessä minuun opintosähköpostini kautta xxxx tai puhelimitse xxxx.



## LIITE 2 TEEMAHAASTATTELU KYSYMYKSET

### Teemahaastattelu

#### Haastattelun kulku

##### 1. Haastateltavan taustatiedot

- Työtehtävä
- Työkokemus Kelalla + muualla
- Mikä on tarkempi työnkuva?
- Onko kehitys- vai ylläpitotehtävissä vai kumpaakin?
- Millaista osaamista henkilöllä on tuottaa arkkitehtuurikuvauksia?

##### 2. Arkkitehtuurin dokumentoinnin nykytila

- Mikä on nykyisen arkkitehtuurin dokumentoinnin tila tällä hetkellä?
- Millaista dokumentaatiota tällä hetkellä tehdään?
- Ovatko tuotetut materiaalit saatavilla/löydettävissä helposti?
- Ovatko aineistot ajan tasalla?
- Onko dokumentoitu riittävästi?
- Miten arkkitehtuuri- ja mallintamispäätökset tehdään?
- Millaista dokumentaatiota voitaisiin tehdä?
- Millaista arkkitehtuurin ja dokumentaation hallinta tällä hetkellä on?

##### 3. Arkkitehtuurin dokumentoinnin tarve

- Minkä tasoiselle arkkitehtuurin dokumentoinnille on tarve?
- Mistä tarve koostuu?
- Tarvitseeko kehittäjä erilaista dokumentaatiota kuin ylläpito?
- Miten tärkeänä dokumentaatiota pidetään omassa työssä?
- Onko dokumentointi tärkeämpää/hyödyllisempää suunnittelun alussa vai pitkin toteuttamista?
- Onko dokumentaatiota liikaa vai liian vähän?
- Millaista dokumentaatiota pitäisi tehdä, että se olisi täydellistä?
- Millaisia tarpeita olisi, jos niitä nyt ei käytetä?
- Yhteiset mallit palveluiden kesken, jos niitä ei nyt ole?

##### 4. Dokumentaatio

- Miten arkkitehtuuria tulisi mallintaa ja dokumentoida, jotta se voisi palvella mahdollisimman laajaa joukkoa henkilöstöstä yli organisaatiorajojen?
- Voisiko mallit palvella yli organisaatioyksiköiden kommunikaation välineenä?