Tampereen yliopisto

Lassi Helminen

TIEDONSIIRTOPROSESSIN TEHOSTAMINEN ETL-TYÖKALUN AVULLA

Diplomityö
Johtamisen ja talouden tiedekunta
Maija Lampu
Pasi Hellsten
Toukokuu 2024

TIIVISTELMÄ

Lassi Helminen: Tiedonsiirtoprosessin tehostaminen ETL-työkalun avulla
Diplomityö
Tampereen yliopisto
Tietojohtaminen
Toukokuu 2024

ETL-prosessi on yksi esimerkki tiedonsiirtoprosessista, joka mahdollistaa järjestelmien ja tietovarastojen välisen tiedon siirtämisen. ETL-prosessi kuvaa datan siirtymistä yhden tai useamman järjestelmän ja tietovaraston välillä. ETL-termi muodostuu englannin kielen sanoista *extract* (poimia), *transform* (muokata) ja *load* (ladata), kuvaillen datan käsittelyä prosessin eri vaiheissa. Hyvin suunniteltu ETL-prosessi mahdollistaa tietovarastojen oikeaoppisen hyödyntämisen, missä dataa kerätään, jalostetaan ja yhdistetään sekä siirretään edelleen järjestelmien ja tietovarastojen välillä. ETL-prosessi täydentää dataa yhdistelemällä eri järjestelmistä saatavaa dataa ja lisää sen arvoa jalostamalla raakadataa tiedoksi, joka on helposti luettavassa muodossa loppukäyttäjille liiketoiminnan ratkaisuihin ja päätöksentekoprosesseihin.

Kohdeorganisaation nykyinen ETL-prosessin tiedonsiirtomenetelmä hyödyntää manuaalisesti ohjelmoituja koodeja, jotka ovat vaikealukuisia ja niiden ymmärtäminen vie paljon aikaa. Tutkimuksen tavoitteena oli selvittää mahdollisia vaihtoehtoisia ratkaisuja kohdeorganisaation WMS-järjestelmän ja tietovaraston väliseen ETL-prosessitoteutukseen, johon etsittiin korvaavia ratkaisuja nykyiselle manuaalisesti ohjelmoidulle tiedonsiirtoprosessille. Tutkimuksessa hyödynnettiin kvalitatiivisen tutkimuksen lainalaisuuksia, jossa kirjallisuuskatsauksen ja dokumenttianalyysin avulla luotiin tapaustutkimuksen teoreettinen viitekehys, jossa syvennyttiin tutkittavaan tapaukseen sen luomassa kontekstissa. Tapaustutkimuksen työpajakeskusteluissa kartoitettiin ETL-prosessiin sijoittuvan ratkaisun tarpeita asiantuntijoiden kanssa, ja tunnistettiin ETL-työkalulta vaadittavat ominaisuudet kohdeorganisaation näkökulmasta.

Tutkimuksessa löydettiin mahdollisia kohdeorganisaation toimintaympäristöön ja käyttötarkoituksiin soveltuvia liiketoimintatavoitteen täyttäviä ratkaisuja, joita esitettiin vastauksina tutkimuskysymyksiin. Ratkaisuilta edellytettäviä ominaisuuksia ja muita yksityiskohtia vertaillessa huomattiin niiden tunnusomaiset erityispiirteet, jotka toivat esiin ratkaisuiden yksilölliset eroavaisuudet.

Avainsanat: ETL, ETL-prosessi, ETL-työkalut, tietovarasto, ohjelmistotuotanto

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

ABSTRACT

Lassi Helminen: Enhancing data transfer process with ETL tool
Master's thesis
Tampere University
Information and Knowledge Management
May 2024

The ETL process is an example of a data transfer process that enables data movement between systems and data warehouses. The ETL process describes the movement of data between one or more systems and data warehouses. The term ETL is derived from the English words extract, transform, and load, describing the handling of data at various stages of the process. A well-designed ETL process enables the proper utilization of data warehouses, where data is collected, refined, combined, and transferred between systems and data warehouses. The ETL process enhances data by combining data from different systems and adding value to it by transforming raw data into information that is easily understandable for end-users in business solutions and decision-making processes.

The current ETL process data transfer method in the target organization utilizes manually programmed codes that are difficult to read and understanding them takes a lot of time. The aim of the study was to explore potential alternative solutions for the ETL process implementation between the organization's WMS system and data warehouse, seeking alternative solutions to the current manually programmed data transfer process. The study utilized the principles of qualitative research, where a theoretical framework for the case study was created through literature review and document analysis, delving into the researched case within its created context. Through workshop discussions in the case study, the needs for a solution within the ETL process were identified with experts, and the required features of an ETL tool were identified from the organization's perspective.

The study identified potential solutions suitable for the organization's operating environment and purposes, fulfilling the business objective, which were presented as responses to the research question. When comparing the required features and other details of the solutions, the distinctive characteristics of each solution were noted, highlighting the individual differences between them.

Keywords: ETL, ETL process, ETL tools, data warehouse, software engineering

The originality of this thesis has been checked using the Turnitin Originality Check service.

ALKUSANAT

Diplomityö toteutettiin toimeksiantona Leanware Oy:lle, jolle haluan osoittaa suuret kiitokset tästä mahdollisuudesta, erityisesti diplomityön ohjaajille Pekka Saarelaiselle ja Mikko Vilppulalle sekä kaikille työkavereille, jotka tarjosivat asiantuntevia kommentteja työhön liittyen.

Erityiskiitokset diplomityön ohjaajalle Maija Lampulle nopeista reagoinneista kysymyksiin sekä rakentavasta palautteesta työhön liittyen. Ajatustenvaihto loi uskoa kirjoitusprosessiin ja työn valmistumiseen. Kiitos myös työn toiselle tarkastajalle Pasi Hellstenille.

Kiitos myös kavereilleni, jotka ovat olleet mukana opiskeluvuosieni aikana ja tarjonneet vastapainoa opiskelulle. Suurimmat kiitokset perheelleni. Vuosien mittaan tarjottu sanoinkuvaamattoman tärkeä tuki, kannustus ja luotto antoivat voimia vaikeilla hetkillä ja auttoivat kohti valmistumista.

Nyt on aika suunnata katseet tulevaan. Kiitos.

Tampereella, 4.5.2024

Lassi Helminen

SISÄLLYSLUETTELO

1. JOHDANTO ... 1
1.1 Tutkimuksen tausta ja tutkimusongelma ... 1
1.2 Tutkimuksen tavoitteet ja tutkimuskysymykset ... 7
1.3 Tutkimuksen rakenne ... 8
2. TUTKIMUSMENETELMÄT ... 10
2.1 Tutkimusasetelma ... 11
2.2 Tutkimuksen viitekehys ... 12
2.2.1 Tutkimusfilosofia ... 13
2.2.2 Teoreettinen lähestymistapa ... 13
2.2.3 Metodologinen valinta, tutkimusstrategia ja aikahorisontti ... 14
2.2.4 Tiedonkeruumenetelmät ... 16
2.3 Tutkimuksen rajoitteet ... 21
3. ETL-PROSESSI OHJELMISTOTUOTANNOSSA ... 22
3.1 Ohjelmistotuotanto ... 22
3.1.1 Ohjelmistokehitys ... 23
3.1.2 Kokonaisarkkitehtuuri ... 26
3.1.3 Tuotearkkitehtuuri ... 29
3.1.4 Ohjelmistoarkkitehtuuri ... 29
3.1.5 Kerrosarkkitehtuuri ... 31
3.2 ETL-prosessi ... 35
3.2.1 Extract ... 36
3.2.2 Transform ... 37
3.2.3 Load ... 38
3.3 ETL-prosessien kehitys ... 39
3.4 ETL-työkalut ... 41
3.5 Tietovarastot ETL-prosessissa ... 43
3.6 Tietovarastoarkkitehtuuri ... 46
3.7 Tietojärjestelmät ... 48
3.8 Ohjelmointirajapinnat - API ... 50
3.9 Integraatiot yleisesti ... 51
3.9.1 Tietojärjestelmien väliset integraatiot ... 52
3.9.2 Integraatioarkkitehtuuri ... 53
4. TUTKIMUKSEN KOHDEORGANISAATIO ... 56
4.1 Leanware Oy ... 56
4.2 LeanwareWMS-järjestelmä ... 57
4.3 ETL-työkalun valintakriteerit ... 59
5. TUTKIMUKSEN LÖYDÖKSET JA NIIDEN TARKASTELUT ... 65
6. YHTEENVETO JA PÄÄTELMÄT ... 74
6.1 Tutkimuksen arviointi ... 75

6.2 Jatkotutkimusmahdollisuudet 77
LÄHTEET 79

LYHENTEET JA MERKINNÄT

ERP	engl. Enterprise Resource Planning, toiminnanohjausjärjestelmä
ETL	engl. Extract - Transform - Load, datan käsittely- ja siirtomenetelmä
iPaaS	engl. Integration as a Service, integraatioalustapalvelu
MES	engl. Manufacturing Execution System, tuotannonohjausjärjestelmä
SAAS	engl. Software as a Service, ohjelmistopalvelu
SCM	engl. Supply Chain Management System, toimitusketjun hallintajärjestelmä
WMS	engl. Warehouse Management System, varastonhallintajärjestelmä

1. JOHDANTO

Johdannossa käsitellään diplomityön taustat ja diplomityön kohdeorganisaation tarpeet, miksi kyseistä tutkimusongelmaa tutkitaan sekä esitellään tutkimusongelma tarkemmin. Lisäksi johdannossa kuvataan diplomityöhön liittyvät tutkimuskysymykset, havainnollistetaan tutkimuksen tavoitteet sekä esitellään tutkimuksen rakenne.

1.1 Tutkimuksen tausta ja tutkimusongelma

Martin & Feathers (2009) toteavat, kuinka toimivan, mutta huonosti kirjoitetun, koodin ymmärtäminen vie lukemattomia tunteja aikaa ja hukkaa organisaation resursseja turhaan. Pienetkin koodimuutokset johtavat rikkoviin muutoksiin muissa osissa koodia. Yksinkertaisistakin muutoksista kasvaa kompleksisia kokonaisuuksia, jotka kasaantuvat muodostaen uusia monimutkaisia ja monirakenteisia kokonaisuuksia, jotka lisäävät vain koodin monimutkaisuutta ja vaikealukuisuutta. (Martin & Feathers 2009)

Tästä lähtökohdasta lähdettiin kehittämään diplomityön aihetta, jonka avulla pystyttäisiin ottamaan kantaan tähän, niin kirjallisuudessa kuin kohdeorganisaatiossakin, tunnistettuun ongelmaan; kuinka pystyttäisiin tehostamaan datansiirtoprosesseja tietojärjestelmästä tietovarastoon sekä pääsemään eroon työtuntien hukkaamisesta huonosti kirjoitetun koodin ymmärtämiseen. Vuosituhannen vaihteessa kehitetty tietovarastointiprosessi voidaan nähdä yhtenä tärkeimmistä kehitysaskeleista tietojärjestelmäkentän kehitykseen liittyen (Wixom & Watson 2001), mikä on tärkeässä roolissa tämän tutkimuksen viitekehyksessä, kun selvitetään mahdollisia vaihtoehtoisia ratkaisuja datansiirtoprosessiin. Khan et al. (2024) toteavatkin, kuinka 42 prosenttia organisaatioista kamppailevat dataintegroinnin parissa tietovarastointiin liittyvissä ETL-prosesseissa (Extract - Transform - Load -prosessi) ja Xavier & Moreira (2013) ovat havainneet, kuinka tietovarastoprojektien ETL-prosessiosuus kuluttaa suurimman osan resursseista, joita tarvitaan rakentaa tietovarasto.

Datansiirtoprosessia voidaan kuvata ETL-prosessina, jossa dataa siirretään järjestelmien ja tietovarastojen välillä. ETL-termi muodostuu englanninkielisistä sanoista *extract* (poimia), *transform* (muuntaa) ja *load* (ladata), mitkä kuvaavat datan käsittelyä prosessin eri vaiheissa. ETL-prosessi mahdollistaa tietovarastojen tehokkaan käytön, jossa dataa kerätään, siivotaan, yhdistellään, jäsennellään ja jaetaan järjestelmien sekä tietovarastojen välillä. (Kimball & Caserta 2011)

Yksinkertaistettuna ETL-prosessi pyrkii ottamaan raakadataa eri järjestelmistä ja siirtämään sen tietovarastoihin. Data luetaan järjestelmistä, sen laatua tarkkaillaan ja tarpeen mukaan muunnetaan, muokataan ja puhdistetaan. Prosessissa dataa yhdenmukaistetaan ja yhdistellään ja lopulta data ladataan tietovarastoon. ETL-prosessien operaatioiden avulla dataa tuodaan helposti saatavaksi tietovarastoon esitysvalmiissa muodossa päätöksentekoon sitä tarvitseville käyttäjille. (Kimball & Caserta 2011)

Toisaalta ETL-prosessi käsittää enemmän kuin pelkän datan siirtämisen lähdejärjestelmästä tietovarastoon. Se poistaa datavirheet ja korjaa puuttuvat tiedot, se muuntaa raakadataa tiedoksi, yhdistää useiden lähteiden datan tiedoksi ja jäsentelee ne loppukäyttäjille hyödynnettäväksi liiketoiminnan ratkaisuissa ja niissä tukevissa päätöksentekoprosesseissa. ETL-prosessi on enemmän kuin pelkkä tiedonkulkuprosessi, se muuttaa useiden lähteiden dataa tiedoksi, jota organisaatio pystyy hyödyntää tietoperustaisessa päätöksenteossa. ETL-prosessi käsittelee tietoa arvokkaana resurssina ja pyrkii luomaan tiedosta merkityksellistä arvoa organisaation toiminnalle. (Xavier & Moreira 2013)

Tämä diplomityö tehtiin toimeksiantona Leanware Oy:lle, jossa tutkittiin mahdollisia työkaluja, menetelmiä ja ratkaisuja, jolla pystyttäisiin korvata geneeriset koodit lähdekoodissa kustomoidulla ratkaisulla ETL-prosessissa dataa siirrettäessä WMS-tietojärjestelmästä (Warehouse Management System) tietovarastoon. Datan määrän kasvaessa, datan lukemis- ja siirtämisprosessien on sopeuduttava tähän muutokseen ja niitä on kehitettävä entistä tehokkaammiksi ja helppokäyttöisiksi. Tämä edellyttää uusia ja paranneltuja tekniikoita, menetelmiä ja käytäntöjä tietojen siirtämisen mahdollistamiseksi.

Wixom & Watson (2001) toteavat omassa tutkimuksessaan, kuinka tietojärjestelmä- ja tietovarastoprojektin onnistumisiin liittyvät samankaltaiset osatekijät. Heidän tutkimuksensa käsittelee tietovarastointiprojektin onnistumisen tekijöitä, josta selviää, kuinka järjestelmän ja datan laatu olivat viime kädessä suurimmat vaikutustekijät tietovarastointiprojektin onnistumisessa. Näiden onnistumisen tekijöiden taustalla vaikuttaa organisaation, kyseisen projektin ja teknisen kyvykkyyden osatekijöiden muodostama kokonaisuus, mitkä mahdollistavat omalta osaltaan datan laadun ja järjestelmän toiminnan tietovarastointiprosessiin liittyen. (Wixom & Watson 2001)

Mufioz et al. (2011) tunnistavat ETL-prosessien tarkoituksenmukaisen suunnittelun olevan päätekijä tietovarastointiprojektien onnistumisiin liittyen. Hiltunen (2022) toteaa omassa tutkimuksessaan, kuinka ETL-prosessin kehittämistä on mahdollista toteuttaa

ETL-työkaluilla tai ohjelmointikielillä koodaamalla ja huomauttaa ETL-työkalujen toiminnoista, joista se vastaa; ETL-työkalulle on ensiarvoisen tärkeää sen sopivuus ja kyvykkyys suorittaa ETL-prosessin vaiheet. Hiltunen (2022) viittaa omassa tutkimuksessaan Patel & Patel (2020) teokseen ja toteaa, kuinka aikaisemmin ETL-prosessien pääasiallisen kehitystyön toteutustapa oli ohjelmointi, mutta ajan mittaan ETL-työkalujen kehittyessä, niistä on tullut vakiintunut osa ETL-prosessien kehitystyötä. Goldfedder (2020) toteaa, että järjestelmää, joka voi noutaa tai vastaanottaa (*extract*) dataa lähdejärjestelmästä, toteuttaa datamuunnoksia datalle (*transform*) erilaisten käsittelyvaiheiden kautta ja lopulta siirtää (*load*) nämä tiedot määrättyyn paikkaan, voidaan kutsua ETL-työkaluksi.

Kirjallisuudesta löytyy useita selvitys- ja tutkimustöitä ETL-prosesseista ja tarkemmin ETL-prosesseissa käytettävistä työkaluista, joissa työkalujen eroja analysoidaan ja niiden tunnusomaisia piirteitä verrataan ja tarkastellaan tiettyjen tutkimusvalintojen kautta. Esimerkiksi Khan et al. (2024) tutkivat aihetta kokonaisvaltaisesti, ETL-prosessissa käytettävien menetelmien, resurssien ja keskeisten standardien kautta ja pyrkivät tarjoamaan perusteellisen käsityksen ETL-prosessin merkityksestä tietovarastoinnin kontekstissa.

Kohdeorganisaation nykyinen menetelmä, jossa tiedonsiirto tapahtuu geneeristen koodien avulla, ei ole tehokas ratkaisu, kun käytettävät koodit käyttävät vanhoja toimintatapoja, ovat vaikealukuisia ja niiden ymmärtäminen vie paljon aikaa. Tällainen malli ei ole tehokas, kun halutaan mahdollistaa nopea ja kustannustehokas tiedonsiirto-operaatio, joka tarjoaa asiakaskohtaista, asiakkaiden tarpeisiin mukautuvaa räätälöintiä. Hovi et al. (2009, s. 53) toteavat, kuinka tietovarastoinnin ETL-prosessi voidaan toteuttaa käsin ohjelmoimalla tai ETL-työkaluilla. He tunnistavat perinteisen manuaalisen ohjelmoinnin ongelmakohdat tietovarastoinnin ETL-vaiheessa, esimerkiksi ohjelmoinnin vaatimat työmäärät, ohjelmoijien eri tyylit, josta seuraa koodin vaikealukuisuutta sekä koodin kehitys- ja ylläpitotyöt ovat henkilöriippuvaisia, aiheuttavat haasteita ETL-prosessissa. Myös Zode (2007) on havainnut saman ja lisää, kuinka ohjelmoimalla kirjoitetut ETL-prosessin vaiheiden koodit ovat pitkiä ja usein vaikeasti dokumentoitavissa, kun koodisto koostuu yleensä monesta eri ohjelmointikielestä, mikä lisää haastavuutta entisestään. Näille esitetyille ongelmille molemmat tarjoavat yhdeksi ratkaisuksi ETL-työkaluja, jotka ovat suunniteltu lähdejärjestelmän ja tietovaraston väliseen ETL-prosessin toteuttamiseen.

Mali & Bojewar (2015) toteavat, kuinka markkinoilla on tarjolla useita erilaisia ETL-työkaluja asiakkaiden tarpeisiin ja huomauttavat, kuinka ETL-työkalun valintaprosessia

pitäisi ohjata organisaation liiketoimintavaatimukset, jotka määrittelevät loppujen lopuksi ETL-työkalun valinnan.

ETL-prosessien datansiirtoa voidaan siis toteuttaa ohjelmoimalla tai ETL-prosesseihin spesifioiduilla työkaluilla, jotka ovat optimoitu ETL-prosessien datan poimimiseen, sen käsittelyyn ja siirtoon. Vaihtoehtoisten ratkaisuvaihtoehtojen selvityksessä on tarpeen erotella käsin ohjelmoitujen ja ETL-työkaluilla toteutettujen ETL-prosessien vahvuuksia ja heikkouksia. Käsin ohjelmoitujen ETL-prosessien ja ETL-työkalupohjaisten ETL-prosessien vahvuuksien ja heikkouksien eroja voidaan tarkastella esimerkiksi Zoden (2007) näkemyksien pohjalta.

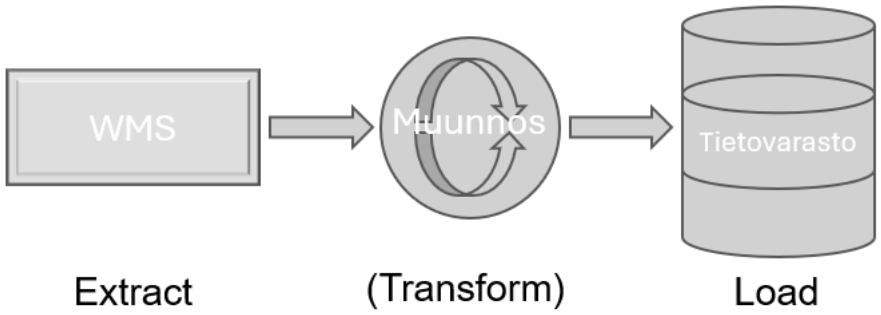
Zode (2007) näkee, että käsin ohjelmoitujen ETL-prosessien vahvuuksien liittyvän koodiin hallittavuuteen, sen itsemääräämisoikeuteen ja sen joustavuuteen. Kun itse tekee, saa juuri sellaista, mitä haluaakin ilman ETL-työkalujen mahdollisia rajoitteita. Myös ETL-putkien testaus on manuaalisesti helpompaa, koska siinä pystyy hyödyntämään ohjelmistotuotannosta tuttuja automatisoituja yksikkötestaustyökaluja.

Toisaalta käsin ohjelmoitujen ETL-prosessien heikkouksina Zode (2007) on havainnut niiden muutostyöt, versionhallinnan ja dokumentoinnin, metatietotaulukoiden erilliset manuaaliset ylläpitotyöt sekä prosessien suorituskyvyn. ETL-prosessien jatkuvat kehitys- ja muutostyöt ovat kokonaan kehittäjien vastuulla, jotka vastaavat koodin dokumentoinnista ja kommentoinnista. Prosessien toimintalogiikan ymmärtäminen kietoutuu vahvasti koodin dokumentointiin sekä sen kommentointiin. Metatietotaulukoiden ylläpito täytyy toteuttaa manuaalisesti erikseen, ETL-prosessin muutokset täytyy tehdä aina myös metatietotaulukoihin. Huomionarvoista on myös, että käsin ohjelmoitujen ETL-prosessien suorituskyky ja suoritusnopeus on todennäköisesti heikompaa kuin modernien ETL-työkalujen. (Zode 2007)

ETL-työkalujen vahvuuksina Zode (2007) listaa niiden toiminnot, jotka ovat kehitetty ETL-prosessin lainalaisuuksien mukaan, ETL-työkalut selviävät monimutkaisimmistakin datamuunnoksista nopeammalla suoritusnopeudella, mikä mahdollistaa prosessien paremman suorituskyvyn. ETL-työkalut tarjoavat graafisen käyttöliittymän, joka visualisoi prosessin toimintoja, prosessiajoja ja virhetilanteita. Toiminnot ovat integroituja ja automatisoituja, joita voi esimerkiksi aikatauluttaa, seurata, yhdistää ja muuttaa helposti työkalun avulla. ETL-työkalut mahdollistavat datan, komponenttien ja prosessiajojen rinnakkaisen käsittelyn sekä ne tukevat datan lukemista erilaisista tietolähteistä ja tietovarastojen erityylisiä konsepteja. ETL-työkalut tarjoavat myös versionhallinnan tuen, ne auttavat kehittäjää ylläpitämään metatietoja ja lähdekoodin versioita ilman alkuperäisen koodin päällekirjoittamista sekä tukevat dokumentointia. (Zode 2007)

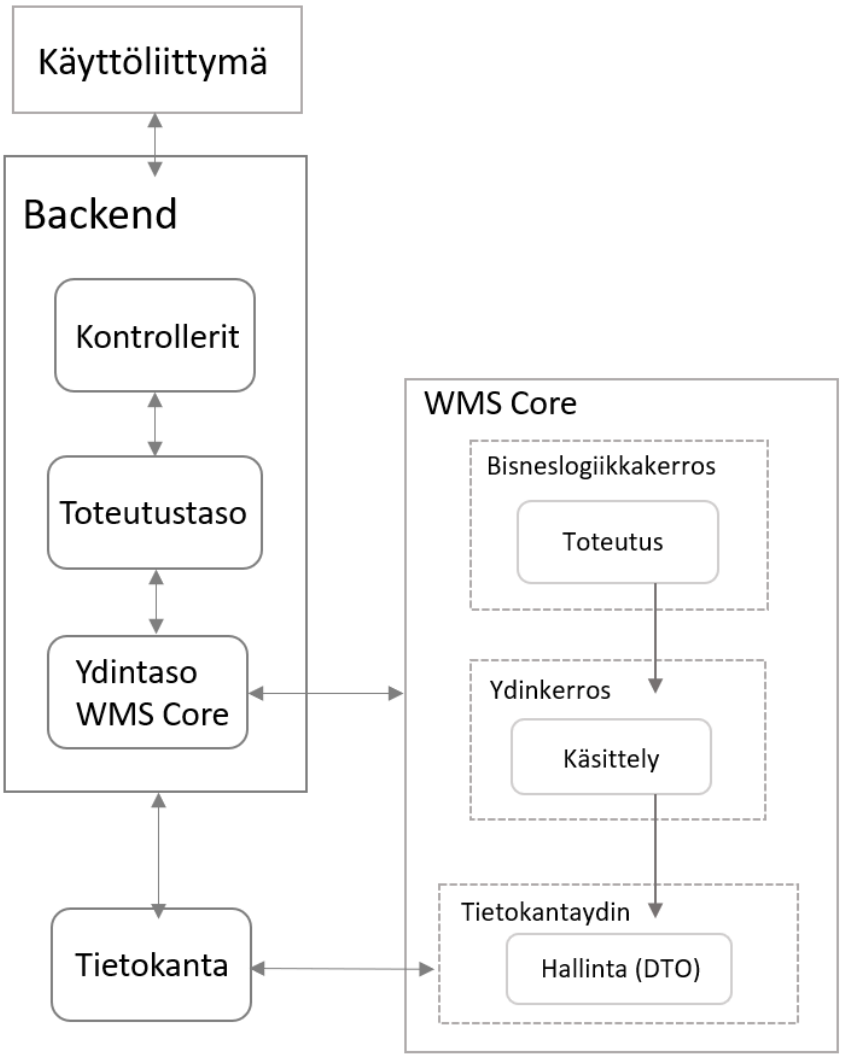
Vastapainoksi ETL-työkalujen heikkouksina Zode (2007) huomioi työkalujen yksilölliset toimintatavat ja niiden toimintalogiikan tuntemisen; vaatii työkaluosaamista ennen kuin pystyy hyödyntää työkaluja niiden täydellä potentiaalilla. Myös joidenkin ETL-työkalujen reaaliaikainen tuki sovelluksille sekä lähdetietokannoille tai lähdejärjestelmille rajoittaa niiden toimintaa. Lisäksi useista lähdetietokannoista tai lähdejärjestelmistä ladattavien datamuunnoksien käsitteleminen suurilla tietomäärillä saattaa muodostaa pullonkauloja ETL-prosessissa. (Zode 2007)

Diplomityössä pyrittiin siis etsimään vaihtoehtoisia ratkaisuja ja toimenpiteitä, joilla pystyttäisiin väistämään Martin & Feathers (2009), Hovi et al. (2009, s. 53) sekä Zode (2007) esiin nostamia huomioita ETL-prosessin manuaaliseen ohjelmointiin liittyvistä haasteista ja näin tehostamaan kohdeorganisaation toimintaa ETL-prosessiin liittyen. Kuva 1 havainnollistaa ETL-prosessia yleisellä, karkealla tasolla, jossa nuolilla kuvataan datan virtausta WMS-tietojärjestelmästä tietovarastoon.



Kuva 1. Yksinkertaistettu ETL-prosessi

Tutkimuksen konteksti voidaan kuvata tarkemmin (kuva 2) WMS-järjestelmän arkkitehtuurin kautta, jossa esitellään tärkeimmät rakennuskomponentit, joiden kautta WMS-järjestelmä muodostuu. Kuvassa 2 on korostettu järjestelmän ydintasoa, johon tutkimuksen aihealue viittaa järjestelmän arkkitehtuurissa. Kuvasta 2 huomataan, kuinka tutkimuskonteksti keskittyy järjestelmän backend-toteutukseen, joka vastaa järjestelmän ydintoimintalogiikasta ja määrittelee järjestelmän toiminnallisuudet sekä prosessoi järjestelmän taustatoimintoja ja hallinnoi tietoja. Ydintason WMS Core -palvelu käsittää eri käyttötapausten toteutukset ja niiden käsittelyt sekä tietokannan hallinnan. Tutkimuskysymykset kohdistuvat järjestelmän ydintasolle, jossa määritellään järjestelmän tiedonsiirtoon liittyvät toimintaperiaatteet sekä muut toiminnot.



Kuva 2. Tutkimuskonteksti (mukaillen Leanware 2024a)

1.2 Tutkimuksen tavoitteet ja tutkimuskysymykset

Diplomityön tavoitteena oli löytää mahdollisia, Leanwaren tarpeisiin ja tavoitteisiin sopivia ratkaisuja, joilla pystytään siirtämään asiakaskohtaista dataa tietovarastoihin ilman, että vanhaan, geneeriseen, legacy-koodiin tarvitsee koskea. Feathersin (2005) mukaan IT-toimiala määrittelee legacy-koodin vaikeasti hallinnoitavaksi koodiksi, johon on hankala tehdä muutoksia ja jonka ymmärrettävyys on usein vaikeaa. Hussain et al. (2017) esittelevät legacy-järjestelmä -käsitteen, joka koostuu legacy-koodista ja jonka järjestelmä on suunniteltu ennen kuin modernit tekniikat ja menetelmät olivat kehitetty. Legacy-järjestelmien vanha suunnitteluarkkitehtuuri vaikeuttaa järjestelmän ymmärtämistä, monimutkaistaa järjestelmän ylläpitoa ja vaikuttaa negatiivisesti järjestelmän suorituskykyyn. (Hussain et al. 2017)

Näiden legacy-termien määritelmien kautta lähdettiin etsimään vaihtoehtoisia ratkaisuja ja menetelmiä, joilla pystyttäisiin väistämään tätä IT-toimialalla tunnettua ongelmaa legacy-käsitteisiin liittyvistä haasteista. Tutkimus tarkastelee kirjallisen teorian ja tutkimusaineistojen kautta mahdollisia vaihtoehtoja, joilla pystyisi väistämään legacy-koodin aiheuttamia, tunnistettuja ongelmakohtia kohdeorganisaation toiminnassa. Tutkimus toteutetaan kohdeorganisaation kontekstissa, heidän toimintaympäristössänsä, jotta saavutetaan paras tulos juuri kohdeorganisaation toimintaan ja käyttötapauksiin liittyen.

Tutkimuskysymykset ohjaavat tutkimusta ja muodostavat lähtökohdat tutkimuksen toteutukseen. Tutkimuskysymyksestä selviää tutkimuksen tavoite, mitä tutkimuksella pyritään selvittämään. (Eriksson & Koistinen 2005) Tutkimuksen päätutkimuskysymys on muodostettu kohdeorganisaation tunnistettujen ongelmien ja tutkimuksen tavoitteiden kautta. Päätutkimuskysymys on muotoa:

- Onko olemassa soveltuvia liiketoimintatavoitteen täyttäviä ratkaisuja legacy-koodille, joilla voidaan siirtää dataa WMS-tietojärjestelmästä tietovarastoon?

Alatutkimuskysymyksen avulla tarkastellaan kohdeorganisaatiossa tunnistettuja valintakriteereitä ja niiden soveltumista ETL-työkalujen ominaisuuksiin. Alatutkimuskysymys on muotoa:

- Onko olemassa kohdeorganisaation valintakriteereihin soveltuvia ETL-työkaluja?

Tutkimuskysymyksillä haetaan mahdollisia soveltuvia ratkaisuja kohdeorganisaation tiedonsiirtoprosessiin, jolla pystytään korvaamaan nykyinen tiedonsiirtomenetelmä uudella, kokonaisvaltaisesti tehokkaammalla ratkaisulla, joka täyttää kohdeorganisaation tunnistamat tarpeet ja vaatimukset.

Hirsjärvi et al. (2008) kuvaavat tutkimuksen tarkoitusta neljän piirteen perusteella. Tutkimuksen piirteet voidaan sanoittaa kartoittavana, selittävänä, kuvailevana ja ennustavana. Piirteet taustoittavat tutkimuksen tarkoitusperää ja sen pyrkimystä. Nämä piirteet muotoutuvat tutkimuskysymyksien perusteella, jotka ohjaavat tutkimusta ja sen luonteen kehittymistä. (Hirsjärvi et al. 2008)

Tämän tutkimuksen tarkoitus on etsiä ja tarkastella vaihtoehtoisia ratkaisuja tunnistettuun ongelmaan ETL-prosessissa, jonka perusteella tutkimuksen tarkoitusta voidaan pitää kartoittavana selvityksenä, jossa selvitetään mahdollisia, korvaavia ratkaisuja nykyiseen toimintatapaan. Kartoittavaa selvitystä tehdään kohdeorganisaation näkökulmasta ja löydettyjä uusia menetelmiä peilataan kohdeorganisaation vaatimuksiin, käyttötarpeisiin, toimintatapoihin ja liiketoimintatavoitteisiin. Tutkimuksessa esiinnousseita ratkaisuja pyritään integroimaan kohdeorganisaation toimintaympäristöön mahdollisuuksien mukaan.

1.3 Tutkimuksen rakenne

Tutkimuksen ensimmäisessä luvussa esiteltiin tutkimuksen tausta ja lähtökohdat sekä tutkimusongelma, tutkimuskysymykset ja tavoitteet. Tämän jälkeen toisessa luvussa syvennytään tutkimuksen tutkimusmenetelmiin ja niiden lainalaisuuksiin sekä esitellään tutkimuksen ominaispiirteet ja siihen liittyvät tiedonkeruumenetelmät, mistä muodostuu tutkimuksen selkäranka. Tutkimuksen viitekehys esitellään Saunders et al. (2019) luoman mallin pohjalta, jonka avulla pureudutaan tutkimuksen kuuteen osatekijään.

Toinen luku käsittää myös tutkimusmenetelmien tarkemmat kuvaukset ja esittelee niiden luonteenomaisia piirteitä sekä esiin tullutta teoreettisesta kirjallisuudesta löydettyä tietoa tutkimuksen rajaamassa viitekehyksessä. Kirjallisen teorian merkitystä hyödynnetään tieteellisten käytäntöjen mukaisesti esitettynä, millä perustellaan käytettyjä valintoja ja menetelmiä, jotka on tunnistettu mahdollisiksi vaihtoehtoisiksi ratkaisumenetelmiksi. Lisäksi luvussa käsitellään tutkimuksen rajoittavat tekijät ja muut rajoitteet, jotka oli tunnistettu tutkimuksessa.

Kolmannessa luvussa käsitellään tutkimuksen aihepiiriin liittyvää kirjallista tietoa ja teoriaa, mitkä on tunnistettu esiteltyjen tiedonkeruumenetelmien avulla ja yhdistetään tutkimuksessa käsiteltäviä aiheita tutkittavan ongelman ympärille lähdeaineistoihin pohjautuen. Tutkimuksen viitekehyksessä tunnistetut aihepiirit esitellään tunnistettujen, tiedonkeruumenetelmillä haettujen akateemisten julkaisujen ja aineistojen kautta. Tutkimusongelman ympärille liittyviä aihepiirejä esitellään ylhäältä alaspäin - menetelmällä, jossa kuljetaan karkeilta tasoilta yksityiskohtaisemmille tasoille

tutkimusongelmaan paneutuen. Aluksi esitellään yleisesti ohjelmistotuotantoa ja -kehitystä sekä ohjelmistojen kehitystyötä ohjaavia arkkitehtuureja, jotka auttavat hahmottamaan ETL-prosessiin liittyviä lainalaisuuksia ylemmältä tasolta. Tämän jälkeen paneudutaan ETL-prosessiin sekä sen ominaispiirteisiin ja ETL-prosessiin liittyviin aihealueisiin. Lopuksi kuvataan datan ja tietojen jakamista ja niiden yhdistelyä järjestelmien välillä.

Neljäs luku pitää sisällään tutkimuksen kohdeorganisaation esittelyn sekä avaa sen WMS-järjestelmän toimintaa. Luvussa kuvataan myös, kuinka kohdeorganisaation tarpeet ja vaatimukset ETL-työkalulle selvitettiin, ja joista muodostettiin valintakriteerit vaihtoehtoisen ETL-työkalun ominaisuuksille.

Viidennessä luvussa analysoidaan tutkimusmenetelmien avulla tunnistettujen ja tutkimuksen raameihin sopivien ETL-työkalujen vaihtoehtoja. Kirjallisuudesta tunnistettuja vaihtoehtoja ja menetelmiä tulkitaan ja sopeutetaan kohdeorganisaation toimintaympäristöön. Luvussa etsitään mahdollisia ratkaisuja tutkimusongelmaan ja analysoidaan vaihtoehtoja ja niiden ominaisuuksia kohdeorganisaation näkökulmasta. Ratkaisuvaihtoehtojen ominaisuuksia tulkitaan ja vertaillaan kohdeorganisaation toimintaympäristöön ja käyttötarkoituksiin peilaten.

Viimeisessä luvussa vastataan tutkimuskysymyksiin, kootaan tutkimuksen havainnot yhteen ja pohditaan niiden merkitystä kohdeorganisaation näkökulmasta sekä arvioidaan tutkimuksen toteutusta ja pohditaan ratkaisuvaihtoehtojen vaikutuksia kohdeorganisaation toimintaan. Lopuksi tarkastellaan vielä tutkimuksen jatkokehityskohteita, jotka on tunnistettu tutkimuksen viitekehyksessä.

2. TUTKIMUSMENETELMÄT

Diplomityö toteutettiin tapaustutkimuksena, jossa keskeisinä elementteinä toimivat tutkimuskysymykset, tutkimusasetelma ja aineistojen analyysit. Tapaustutkimuksessa tarkastellaan yhtä tai useampaa ”tapausta”, jotka voivat olla yksilöitä, ryhmiä, ohjelmia, prosesseja tai ilmiöitä. (Eriksson & Koistinen 2005) Tämän tutkimuksen tapauksessa tarkasteltiin kohdeorganisaation ETL-prosessia, jota pyritään parantamaan, tehostamaan ja optimoimaan vaihtoehtoisilla ratkaisuilla, joilla mahdollistetaan dynaamisempi datansiirtoprosessi LeanwareWMS-tietojärjestelmästä tietovarastoon. Tutkimus perustuu yhteen tapaukseen, jossa tapaukseen ehdotettuja ratkaisuja analysoitiin ja vertailtiin kohdeorganisaation tarpeiden, vaatimuksien, toimintatapojen ja käyttötarkoituksien perusteella.

Tapaustutkimus on yleisesti tunnistettu sopivaksi tutkimusmenetelmäksi ohjelmistotuotantoon, koska sen nähdään paneutuvan senaikaisiin, ohjelmistotuotannon kontekstin ilmiöihin ohjelmistotuotannon luontaisessa omassa tutkimuskehyksessään. (Runeson & Höst 2009) Leanware toimii ohjelmistotalona, joten tämän määritelmän perusteella tapaustutkimus on sopiva tutkimusstrategia tämän tutkimuksen viitekehyksessä sekä ETL-prosessi liittyy vahvasti ohjelmistotuotannon tietojärjestelmien kehitykseen, käyttöön sekä tiedonsiirtoprosesseihin. Ohjelmistotuotanto ja ETL-prosessi ovat keskeisessä osassa modernien tietojärjestelmien kehitystä ja toimintaa, ja niiden tehokas yhdistäminen parantaa järjestelmän toimivuutta ja tuo arvoa liiketoiminnalle.

Tapaustutkimus pyrkii selittämään prosessien yhteisiä ominaisuuksia ja yhtenäisiä malleja sekä kehittämään uusia lähestymistapoja ja ideoita teoreettiseen aineistoon perustuvan järjestelmällisen vertailun avulla. (Eriksson & Koistinen 2005) ETL-prosessia ja siihen liittyviä aihepiirejä tarkasteltiin kirjallisen teoria-aineiston kautta ja dokumenttianalyysin avulla tunnistettiin vaihtoehtoisia ratkaisuja ja menetelmiä, joiden ominaisuuksien soveltuvuutta analysoitiin ja tarkasteltiin kohdeorganisaation tarpeisiin, vaatimuksiin, toimintaympäristöön ja käyttötapauksiin järjestettyjen työpajakeskusteluiden avulla.

Tapaustutkimus syventää käsitystä tutkittavasta ilmiöstä, tutkittavana olevasta tapauksesta sen omassa kontekstissaan. Tapaustutkimuksessa hyödynnetään aiempaa teoreettista aineistoa, jota hyödynnetään tiedonkeruun tukena tapauksen aihepiireihin liittyen ja jonka pohjalta analysoidaan vaihtoehtoja tutkittavaan tapaukseen. Edellä

kuvatut toimenpiteet ohjaavat ehdotuksien ja päätelmien kehitystä, joita tarjotaan ratkaisuina käsiteltävään tapaukseen. (Yin 2018)

Yinin (2018) näkemyksien pohjalta on luotu tämän tapaustutkimuksen johdonmukainen tutkimussuunnitelma, joka pyrkii syventämään käsitystä Leanwaren WMS-tietojärjestelmän ETL-prosessista ja siihen liittyvistä elementeistä ja toiminnoista.

Tutkimusmetodologiset valinnat kuvataan Saunders et al. (2019) kehittämän sipulimallin avulla. Sipulimalli kuvaa tutkimuksen metodologisia valintavaihtoehtoja, jotka muodostavat yksilölliset piirteet tutkimukselle. Sipulimalli koostuu kuudesta kerroksesta, joista jokaisella kerroksella on oma tarkoituksensa ja yhdessä nämä kerrokset luovat tutkimusrungon, johon tämän diplomityön tutkimusperäiset valinnat pohjautuvat. Tutkimuksen ominaispiirteet ovat muodostettu sipulimallin tarjoaman viitekehyksen ympärille.

2.1 Tutkimusasetelma

Vuori (n.d.) toteaa tutkimusasetelman muodostuvan kokonaisuudesta, jonka tutkimusongelma, käytettävät aineistot ja niiden analyysimenetelmät muodostavat. Suunnitelmallisesti rakennetulla tutkimusasetelmalla pyritään löytämään vastauksia muodostettuun tutkimusongelmaan. Jokaisessa tutkimuksessa on omanlaisensa tutkimusasetelma, mikä erottaa esimerkiksi saman aihepiirin tutkimukset toisistaan. (Vuori n.d.) Myös Saunders et al. (2019) huomioi tutkimusasetelman keskeiset piirteet. Tutkimusasetelma sisältää tutkimuskysymykset, joihin etsitään vastauksia, tutkimuksen tavoitteet, jotka seuraavat tutkimusongelman määrittelystä sekä käytettävät aineistot ja niiden lähteet, jotka edesauttavat tiedon keräämistä ja niiden analysointia. Tutkimusasetelma on alustava suunnitelma kokonaiskuvasta, jonka ympärille tutkimus rakentuu.

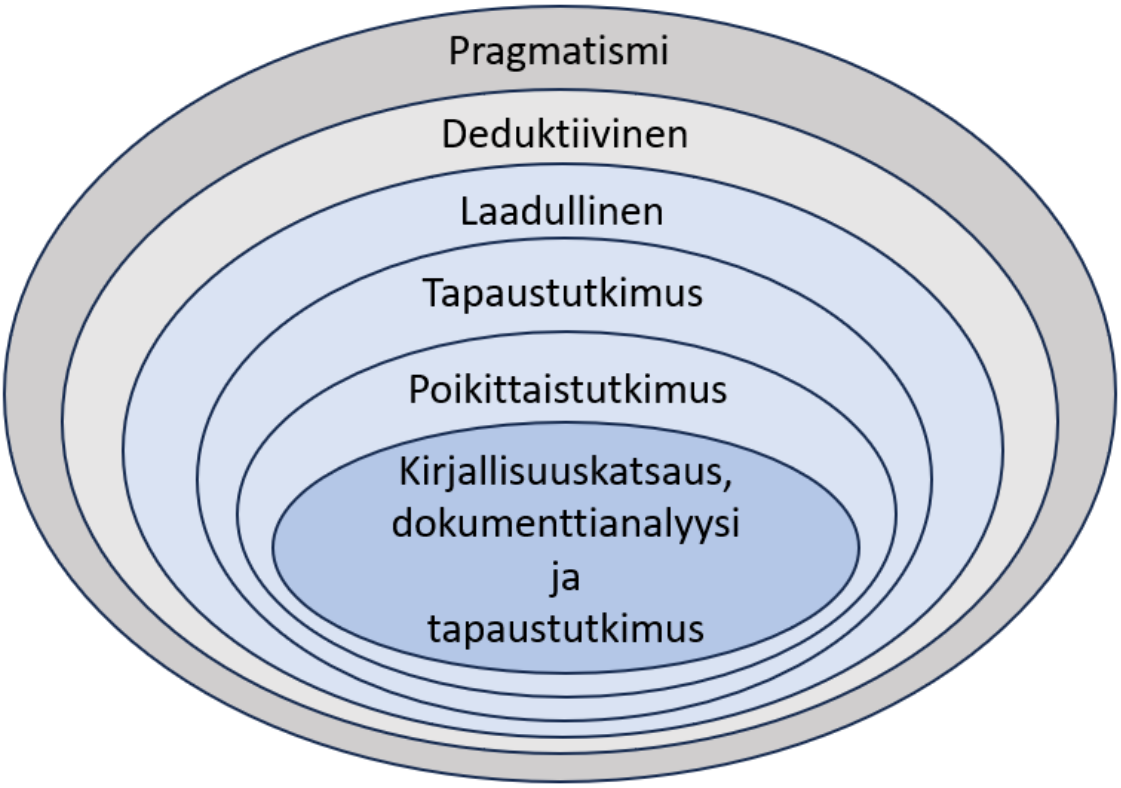
Tutkimusasetelma luo tutkimukselle viitekehyksen, jota tutkimus seuraa. Tutkimusasetelman valinnoilla pyritään etsimään vastauksia valittuihin tutkimuskysymyksiin, jotka määrittelevät ja ohjaavat tutkimusta. Tutkimusasetelma luo kokonaiskuvan menetelmistä, joita tutkimus hyödyntää pyrkiessään vastaamaan tutkimuskysymyksiin. (Saunders et al. 2019)

Tutkimusasetelma muodostaa tutkimuksen selkärangan, jonka ympärille tutkimuksen valinnat kietoutuvat. Tutkimusasetelmaa esitellään luvussa 2.2 Saunders et al. (2019) sipulimallin avulla. Sipulimalli pilkkoo tutkimusasetelman kuuteen yleiseen tekijään, jotka muodostavat tutkimuksen yksilölliset piirteet. Saunders et al. (2019) ovat kehittäneet sipulimallin liiketoiminnan ja johtamisen aloille, mikä on erityisesti kohdennettu

diplomityötä tekeville opiskelijoille. Selittävänä tutkimusasetelman viitekehyksenä sipulimalli sopii tämän tutkimuksen aihepiirikokonaisuuteen, jossa käsitellään liiketoiminnan ja johtamisen aihealueisiin vahvasti linkittyvää ETL-prosessia ja sen aihealueita.

2.2 Tutkimuksen viitekehys

Tutkimuksen viitekehystä esitellään Saunders et al. (2019) kehittämän sipulimallin avulla. Sipulimalli koostuu kuudesta kerroksesta, jotka luovat yhdessä raamit tutkimuksen toteutukselle. Uloimmat kerrokset, tutkimusfilosofia ja teoreettinen lähestymistapa edesauttavat tutkimuksen suorittamista, niiden valintaa ohjaa pitkälti tutkimuskysymykset. Tutkimuksen metodologiset valinnat, tutkimusstrategia ja aikahorisontti pohjautuvat kahteen uloimpaan kerrokseen, jotka ohjaavat niiden valinnassa. Sipulin ydin muodostuu aikaisempien valintojen perusteella uloimmasta kerroksesta sisimpään. Ydin koostuu tiedonkeruutekniikoista ja tiedon analyysimenetelmistä, joiden avulla analysoidaan kerättyä tietoa. Sipulimallin kuusi kerrosta on kuvattu alla olevassa kuvassa 3, johon on yhdistetty tässä tutkimuksessa käytettävät tekniikat ja valinnat kyseisillä sipulin kerroksien tasoilla.



Kuva 3. Sipulimalli tutkimukseen liitetyillä valinnoilla (mukaillen Saunders et al. 2019)

2.2.1 Tutkimusfilosofia

Tutkimusfilosofia kuvaa uskomuksia ja olettamuksia, joita tutkimuksen tiedon keräämiseen liittyy. Valittu tutkimusfilosofia tukee tutkimuksessa käytettäviä valintoja, menetelmiä ja strategioita, jotka muodostavat kokonaisvaltaisen ja yhtenäisen, vankan selkänojan tutkimuksen toteutukselle. (Saunders et al. 2019) Tutkimuksessa päädyttiin pragmatismiin, jossa sen käytännönläheinen lähestymistapa tutkimusongelmaan yhdistettynä spesifiseen ongelmanratkaisupotentiaaliin olivat avaintekijöitä tutkimusfilosofian valinnassa.

Pragmatismi yhdistää faktat ja subjektiiviset näkemykset sekä tiedon ja kokemukset käyttäen hyväksi menetelmiä, joissa yhdistyvät teoriat ja ideat sekä konseptit ja tutkimuslöydöt, joilla etsitään käytännön ratkaisuja tiettyyn ongelmaan. (Saunders et al. 2019) Pragmatismi etsii käytännönläheisiä ratkaisuja tutkittavaan ongelmaan ja pyrkii tarjoamaan käytännöllisiä toimenpiteitä ongelman ratkaisuun, jossa korostuu ratkaisujen hyödyllisyys ja soveltuvuus konkreettisissa tutkimustilanteissa.

Pragmatismin käytännönläheistä suhdetta tiedon hyödyntämisessä oli mahdollista hyödyntää kirjallisuuskatsauksessa esitettyjen teorioiden ja faktatiedon pohjalta kuin myös dokumenttianalyysin kautta muodostettujen lähdeaineistojen pohjalta, kun etsittiin ratkaisuja ja vaihtoehtoja käytännöntoteutukseen kohdeorganisaation toimintatapoihin ja vaatimuksiin liittyen. Pragmatismin valintaan vaikutti myös siihen liittyvien subjektiivisten näkemyksien sekä kokemusperäisen tiedon hyödyntäminen ratkaisuvaihtoehtojen analysoinnissa.

2.2.2 Teoreettinen lähestymistapa

Tutkimuksen teoreettinen lähestymistapa kuvaa, kuinka tutkimus selittää tutkimuksessa löydettyjä päätelmiä ja tuloksia. Deduktiivisessa lähestymistavassa kirjallisuudessa esiintyvien menetelmien ja teorioiden avulla suunnitellaan tutkimusstrategiat, joilla testataan akateemisen kirjallisuuden havaintoja aiheista. (Saunders et al. 2019) Myös Greener (2011) ja Tuomi (2007) toteavat omissa teoksissaan, kuinka deduktiivinen lähestymistapa testaa jo olemassa olevaa teoriaa käytännössä. Deduktiivinen lähestymistapa sopi tähän tutkimukseen, kun haluttiin löytää vaihtoehtoisia, jo olemassa olevia ratkaisuja, joita pystyi teoreettisen kirjallisuuden kautta tukemaan ja analysoimaan kohdeorganisaation toimintaan pohjautuen. Tutkimuksessa hyödynnettiin teorialähtöistä analyysia, jossa jo olemassa olevia työkaluja, teorioita, malleja ja menetelmiä analysoitiin ja tarkasteltiin kohdeorganisaation vaatimuksiin, toimintakenttään ja käyttötapauksiin mukauttaen. Analyysit perustuivat tunnistettuun aikaisempaan tietoon, jota etsittiin ja tarkasteltiin kirjallisuuskatsauksen ja dokumenttianalyysin avulla ja jonka

löydöt liitettiin tutkimuksen työpajakeskusteluihin, joissa aikaisemman tiedon perusteella tunnistettiin vaadittavat ominaisuustarpeet tiedonsiirtoratkaisulle kohdeorganisaation toimintaympäristöön ja käyttötarkoituksiin.

Deduktiivisessa lähestymistavassa tehdään yleispäteviä päätelmiä teoriaan pohjautuvista olettamuksista, joiden voidaan olettaa olevan analyyttisesti tosia. Yleispätevänä sääntönä päättelylle voidaan pitää loogista johdonmukaisuutta. (Ketokivi & Mantere 2010) Tässä tutkimuksessa haluttiin selvittää jo olemassa olevia ratkaisuja ja vaihtoehtoja, joilla tutkimusongelma pystyttäisiin ratkaista ja arvioitiin ratkaisuvaihtoehtojen soveltuvuutta kohdeorganisaation toimintaan ja käyttötapauksiin liitettynä. Kirjallisuuskatsauksen teoria yhdistettynä dokumenttianalyysin aineistoihin ja niistä löydettyihin ratkaisuvaihtoehtoihin mahdollisti tiedon analysoinnin ja tarkastelun, jota hyödynnettiin työpajakeskusteluissa, joissa tunnistettiin kohdeorganisaation tarpeet ja vaatimukset ratkaisulle. Tiedon sopivuutta peilattiin kohdeorganisaation vaatimuksiin ja toimintaan heidän toimintaympäristössään. Ratkaisuvaihtoehtojen mahdollisia toiminnallisia toimintatapojamuutoksia ETL-prosessiin tarkasteltiin tutkimuksen viitekehyksessä, jonka perusteella luotiin päätelmät ja johtopäätökset havainnoista sekä arvioitiin tutkimuksen onnistuneisuutta.

2.2.3 Metodologinen valinta, tutkimusstrategia ja aikahorisontti

Sipulimallin kahden uloimman kerroksen, tutkimusfilosofian ja teoreettisen lähestymistavan, valinnat ohjaavat seuraavan kolmen kerroksen valintaa. Seuraavat kerrokset kuvaavat tutkimussuunnitelman muodostumista, jossa tutkimuskysymyksistä muodostuu tutkimuksen ydin. Tutkimussuunnitelman voidaan olettaa kuvaavan, kuinka tutkimuskysymyksiin aiotaan vastata. Se koostuu kolmesta menetelmästä: metodologiset valinnat, tutkimusstrategia ja aikahorisontti. Nämä menetelmät ja niiden valinnat ohjaavat tutkimuksen toteutusta. (Saunders et al. 2019)

Laadullinen tutkimus nähdään usein tulkitsevana tutkimuksena, jossa tutkija tekee subjektiivisia havaintoja tutkittavasta aiheesta. (Saunders et al. 2019) Saunders et al. (2019) Yinin (2018) mukaan deduktiivinen lähestymistapa on yksi mahdollinen teoreettinen lähestymistapa laadullisissa tutkimuksissa, joissa testataan olemassa olevaa teoriaa laadullisin menetelmin. Tähän pohjautuen tämä tutkimus seuraa kirjallisuudesta löytyvää teoriaa ja aineistoista löytyneitä vaihtoehtoisia menetelmiä, mikä luo yhtenäisen ja uskottavan suunnitelman toteuttaa tutkimusta. Tutkimuksen tarkoitus on löytää mahdollisia kirjallisuudessa ja aineistoissa esiintyviä menetelmiä ja vaihtoehtoja, joilla pystytään etsimään tutkimusongelmaan vastauksia. Tutkimuksen tutkiva luonne selkeyttää näkemyksiä ja ymmärrystä tutkittavasta ongelmasta etsimällä

vastauksia muun muassa tieteellisestä kirjallisuudesta ja niihin liittyvistä aineistoista. (Saunders et al. 2019)

Tutkimusstrategia kuvaa menetelmällisten ratkaisujen kokonaisuutta, joita tutkimukseen liittyy. Tutkimusstrategian valintaa ohjaa tutkimusongelma ja siihen liittyvät tutkimuskysymykset sekä tutkimuksen tarkoitus, mitä tutkimuksella pyritään saavuttaa. (Hirsjärvi et al. 2008) Laadulliseen tutkimukseen liitetty yksi tunnistettu tutkimusstrategia on tapaustutkimus (Saunders et al. 2019), ja tutkimuksen kartoittavalla tarkoituspiirteellä huomataan olevan myös yhteys tapaustutkimukseen. (Hirsjärvi et al. 2008) Valittu tutkimusstrategia edistää tutkimuksen johdonmukaisuutta ja auttaa pääsemään tutkimuksen tavoitteisiin eli löytämään vastaukset tutkimuskysymyksiin. (Saunders et al. 2019) Esitetyt tutkimusstrategiaan liitetyt havainnot tukevat ja vahvistavat tämän tutkimuksen strategian valintaa.

Tietojärjestelmään, tietovarastointiin ja ETL-prosessiin liittyviä aihepiirejä tarkasteltiin teoreettisen kirjallisuuden kautta, mikä tarjosi selkänojaa dokumenttianalyysiin ja siinä vaihtoehtojen tarkasteluun sekä työpajakeskusteluihin ja siinä ominaisuustarpeiden tunnistamiseen. Kirjallisen teorian, aineistojen ja työpajakeskusteluiden kautta muodostettiin päätelmät ja ehdotukset tapaukseen liittyen sekä arvioitiin tutkimuksen onnistumista neljän kirjallisuudessa tunnistetun laadullisen tutkimuksen arviointikriteerin perusteella. Analysointi ja päätelmät muodostettiin kohdeorganisaation näkökulmasta heidän toimintaympäristön organisaatiokulttuurin toimintatapoihin peilaten sekä ETL-prosessin käyttötapauksien ja -tarpeiden perusteella.

Tutkimuksessa selvitettiin mahdollisia vaihtoehtoisia ratkaisuja kohdeorganisaation nykyisille menetelmille, joilla siirretään dataa tietojärjestelmästä tietovarastoon. Tutkimuksessa käytettävää aineistoa kerättiin akateemisista lähteistä löytyvistä tieteellisistä julkaisuista ja tutkimuksista sekä muista kirjallisista aineistoista, joiden pohjalta tunnistettiin kohdeorganisaation vaatimat ominaisuustarpeet ratkaisulle. Löydöksien ja kohdeorganisaation tunnistettujen tarpeiden pohjalta toteutettiin vertailevaa analyysia olemassa olevista menetelmistä, teknologioista ja työkaluista. Löydöksien soveltuvuutta ja tarkoituksenmukaista tutkimusongelman ratkaisua tarkasteltiin kohdeorganisaation näkökulmasta heidän toimintaympäristöönsä ja toimintatapoihin peilaten.

Kirjallisuudessa (esimerkiksi Hovi 2009) kuin myös alan tutkimus- ja selvitystöissä (esimerkiksi Khan et al. 2024) esitellään ETL-työkaluja ja -teknologioita, joita hyödynnetään ETL-prosessien tiedonsiirrossa. Esiteltyjä työkaluja, teknologioita ja menetelmiä tarkastellaan tietystä näkökulmasta tai keskeisten vaatimuksien ja tarpeiden

kautta, jotka luovat aineistoille yksilölliset, tunnusomaiset piirteet. Tässä tutkimuksessa ETL-työkalujen tarpeiden selvitys ja ETL-työkalujen vertailu tehdään kohdeorganisaation tapauksen viitekehyksessä, joka luo yksilöllisen lähestymisen aiheeseen ja sen tuloksien analysointiin.

Tutkimuksen aikahorisontti kuvaa tutkimuksen aikaikkunaa, johon tutkimus sijoittuu. Tutkimuksessa käytettävä poikittaistutkimus kuvaa tietyn ajanhetken tapahtumia. (Saunders et al. 2019) Tämä tutkimus käsittelee kohdeorganisaation nykytilaa, vuonna 2024 käytössä olevia toimintatapoja ja -ympäristöjä, poikittaistutkimuksella, vaikkakin kirjallisuuden ja aineistojen kautta tutkimuksen aihepiireihin liittyvää historiatietoa tullaan käsittelemään pidemmältäkin aikaväliltä tähän päivään.

2.2.4 Tiedonkeruumenetelmät

Tutkimuksen tiedonkeruu perustui kirjallisuuskatsaukseen, dokumenttianalyysiin ja työpajakeskusteluihin. Tutkimuksen tiedonkeruu perustui kirjallisuuskatsaukseen, jossa etsittiin teoreettista tietoa lähdekirjallisuudesta. Teoreettista aineistoa kerättiin tutkimuksen viitekehyksien sisällä tunnistetuista aihepiireistä, mistä muodostui tutkimuskehys vaihtoehtoisten toimintatapojen ja toimintamenetelmien pohjustukseen dokumenttianalyysia varten. Hirsjärvi et al. (2008) toteavat, kuinka kirjallisuuskatsaus toimii tutkimuksen perustana ja ohjaa sen suuntaa, johon tutkimuksella pitäisi edetä. Kirjallisuuskatsaus auttaa hahmottamaan aihepiirin keskeisiä käsitteitä ja tunnistamaan teoriaa valittujen viitekehyksien ympäriltä, jolla voidaan tarkentaa tutkimuskysymyksiä kirjalliseen teoriaan pohjautuen. Kirjallisuuskatsauksessa tulkitaan kokonaisvaltaisesti kirjallisuutta, joka liittyy tutkimuskysymyksen ja tutkimusongelman aihepiiriin. Kirjallisuuskatsauksessa haetaan vastauksia tutkimuskysymykseen, jota tarkastellaan jo olemassa olevan kirjallisuuden kautta. (Aveyard 2019) Kirjallisuuskatsaus auttaa hahmottamaan tutkimuksen aihepiiriin liittyviä näkemyksiä, auttaa syventämään tutkimuksen viitekehyksen aihepiirikokonaisuuksia sekä auttaa rakentamaan kokonaiskuvaa tutkimuksen viitekehykseen rajatusta asiakokonaisuudesta.

Kirjallisuuskatsaus toteutettiin hyödyntäen Tampereen yliopiston kirjaston Andor-palvelua, josta löytyy kattava valikoima tieteellisiä julkaisuja sekä muita kirjallisia aineistoja. Toinen tiedonhakuun käytetty palvelu oli Google Scholar -tietokanta, joka koostuu pelkästään tieteellisistä julkaisuista. (Hirsjärvi et al. 2008) Kirjallisuuskatsauksen tiedonhaussa huomioitiin tutkimusongelman ja tutkimuskysymyksien ympärille sijoittuvat aihepiirit, jotka muodostavat teoreettisen pohjan vaihtoehtoisten menetelmien tutkimiselle tutkimuksen rajaamassa viitekehyksessä sekä kohdeorganisaation asettamassa toimintaympäristössä. Aihepiireistä etsittiin lähdeaineistoja sekä suomeksi

että englanniksi. Kirjallisuuskatsauksessa käytetyt hakusanat ja niiden tulokset ovat esitetty alla olevassa taulukossa (taulukko 1).

Taulukko 1. Hakusanat ja tulokset

Hakusana	Andor	Google Scholar
ETL*	24 837 tulosta	347 000 tulosta
ETL prosessi	2 tulosta	1390 tulosta
ETL process	5326 tulosta	165 000 tulosta
ETL työkalut	1 tulosta	512 tulosta
ETL tools	2762 tulosta	91 900 tulosta
ETL tools comparison	114 tulosta	47 100 tulosta
ETL OR ohjelmistotuotanto	25 348 tulosta	46 600 tulosta
ETL process and software engineering	344 tulosta	41 500 tulosta

Tutkimuksen toisena tiedonkeruumenetelmänä hyödynnettiin dokumenttianalyysia. Bowen (2009) määrittelee dokumenttianalyysin systemaattiseksi toimintamalliksi, joka pohjautuu dokumenttien, painettujen että sähköisten, sekä muiden materiaalien tarkasteluun tai arviointiin ja lisää, kuinka dokumenttianalyysi edellyttää, että aineistoja tarkastellaan ja tulkitaan merkityksien löytämiseksi, ymmärryksen lisäämiseksi ja empiirisen tiedon kehittämiseksi kuten muissakin laadullisen tutkimuksen analyysimenetelmissä. Dokumenttianalyysin avulla pyritään ymmärtämään tiettyä ilmiötä, kulttuuria, tai ilmiöiden välisiä suhteita asiakirjojen sisällön kautta. (Bowen 2009)

Dokumenttianalyysin aineistot tarjoavat tietoa tutkimuksen kontekstista, joka tutkimukseen liittyy. Aineistot tarjoavat taustatietoa sekä historiallista näkemystä asiayhteyteen kuuluvista julkaisuista, jotka toimivat lisätietoina ja avartavat näkemyksiä, mikä auttaa ymmärtämään tutkittavien asioiden historiallisia juuria ja ilmiöitä. (Bowen 2009) Aineistot tarjoavat lisätietoja tutkimusaineistoon, joissa johdetut tiedot ja näkökulmat voivat olla arvokkaita lisäyksiä tutkimuksessa hyödynnettävään tietopohjaan.

Dokumenttianalyysi on tiedonkeruumenetelmänä hyödyllinen, kun tutkitaan historiaa, kulttuuria, yhteiskunnallisia ilmiöitä tai organisaatioiden toimintaa. Lisäksi

dokumenttianalyysi voi olla hyödyllinen lisätietojen hankkimisessa ja tukena muille tutkimusmenetelmille, kuten haastatteluille tai kyselyille. (Bowen 2009)

Tässä tutkimuksessa dokumenttianalyysia hyödynnettiin osana tapaustutkimusta ja siinä hyödynnettyjä työpajakeskusteluita, joissa määriteltiin kohdeorganisaation toimintaa ja tarkemmin sen WMS-järjestelmän ETL-prosessia ja sen luonteenomaisia piirteitä sekä siihen liittyviä toiminnallisuuksia, mitkä asettivat vaatimukset vaihtoehtoisten ETL-työkalujen selvittämistyölle. Bowen (2009) toteaa, kuinka dokumenttianalyysi soveltuu erityisen hyvin laadullisiin tapaustutkimuksiin, jotka tuottavat yksityiskohtaisia kuvauksia yksittäisestä ilmiöstä, tapahtumasta, organisaatiosta tai ohjelmasta. Tämän tutkimuksen tapauksessa pyrittiin selvittämään ETL-prosessiin vaihtoehtoisia tiedonsiirtomenetelmiä pohjautuen kohdeorganisaation tunnistettuihin tarpeisiin ja vaatimuksiin, jotka tunnistettiin ja määriteltiin työpajakeskusteluissa. Dokumenttianalyysin avulla tarkasteltiin aineistojen sisältöä ja asiayhteyttä tämän tutkimuksen ongelmaan liittyen, ja löydöksiä hyödynnettiin tämän tapaustutkimuksen yksilöllisessä tutkimusasetelman luomassa tutkimuskehyksessä. Aineistojen löydöksiä tarkasteltiin ja analysoitiin sekä peilattiin niiden merkityksiä tutkimuskehyksessä, joka oli muodostettu yksilöllisen tapaustutkimuksen tutkimusasetelman pohjalta.

Tapaustutkimuksen aineistoanalyysissa hyödynnettiin tieteellisiä julkaisuja ja tutkimuksia, jotka oli tunnistettu dokumenttianalyysissa ja joiden pohjalta lopulta muodostettiin tutkimussuunnitelma tutkimuskysymyksiin vastaamiseen. Tutkimukseen liittyvien aihepiirien aikaisempia julkaisuja ja tutkimuksia etsittiin Tampereen yliopiston kirjaston Andor-palvelusta sekä Google Scholar -tietokannasta. Aineistojen etsinnässä hyödynnettiin hakusanoja, jotka nähtiin linkittyvän tutkimuksen keskeisimpien aihepiirien käsitteiden ympärille ja liittyvän vahvasti tutkimusongelmaan. Hakusanoina hyödynnettiin niin suomen- kuin englanninkielisiä termejä, mikä lisäsi hyödynnettävien aineistojen määrää (taulukko 2).

Taulukko 2. Hakusanat ja tulokset

Hakusana	Andor	Google Scholar
ETL	23 940 tulosta	355 000 tulosta
ETL prosessi	2 tulosta	1390 tulosta
ETL process	5326 tulosta	165 000 tulosta
ETL työkalut	1 tulosta	512 tulosta
ETL tools	2762 tulosta	91 900 tulosta

ETL tools comparison	114 tulosta	47 100 tulosta

Tutkimuksen aineistoanalyysi suoritettiin dokumenttianalyysina. Bowen (2009) toteaa, kuinka dokumenttianalyysissa aineistoja tarkastellaan ja tulkitaan merkityksien löytämiseksi, ymmärryksen lisäämiseksi ja empiirisen tiedon kehittämiseksi. Dokumenttianalyysissa aineistojen avulla pyritään varmentaa havainnot sekä vahvistetaan näkemyksiä muita lähteitä hyödyntäen.

Tässä tapaustutkimuksessa etsittiin mahdollisia vaihtoehtoisia ratkaisumenetelmiä kohdeorganisaation nykyiselle ETL-prosessin tiedonsiirtomenetelmälle, jolla pystytään siirtämään dataa tietojärjestelmästä tietovarastoon. ETL-prosesseissa tiedonsiirtoon käytettävien menetelmien, teknologioiden ja olemassa olevien ETL-työkalujen etsinnässä hyödynnettiin Bowen (2009) huomioita, joiden pohjalta hyödynnettiin aiheesta ja aihepiirikokonaisuudesta löytyviä aineistoja ja tieteellisiä julkaisuja. Löytyneitä menetelmiä, teknologioita ja ETL-työkaluja tarkasteltiin kohdeorganisaation näkökulmasta heidän toimintaansa peilaten. Vaihtoehtojen tarkastelu tehtiin kohdeorganisaation toimintaympäristön vaatimien liiketoimintatapojen täyttävien ehtojen perusteella, jotka asettivat raamit vertailevalle analyysille.

Löytyneiden aineistojen sisältöjä analysoitiin ja arvioitiin niiden sisältämien teemojen, käsitteiden ja niissä käsiteltyjen trendien mukaan, jotka huomioitiin tutkimusongelman ja tutkimuskysymyksien tarkastelujen kautta. Aineistoista tunnistettiin yleisiä teemoja, joista tärkeimmiksi nousivat ETL-työkalut ja niiden ominaisuudet sekä ETL-työkalujen vahvuudet ja heikkoudet. Aineistojen käsiteltävät käsitteet linkittyivät vahvasti ETL-käsitteen ympärille, jossa keskityttiin ETL-työkalu-käsitteen merkitykseen ja sen ymmärtämiseen. Aineistoista huomioituja trendejä pyrittiin ottamaan huomioon julkaisujen ajankohtaan liittyen ja tarkasteltiin, kuinka ajan kuluessa ETL-työkalujen julkaisut ovat muuttuneet ja onko tässä tunnistettavissa muutosta.

Tutkimuksen vertaileva selvitysosuus toteutettiin tapaustutkimuksena, jossa etsittiin vaihtoehtoisia ratkaisuja kohdeorganisaation ETL-prosessin nykyiselle tiedonsiirtomenetelmälle. Tapaustutkimuksen alkuun koottiin aihepiirin lähdekirjallisuudessa mainitut ETL-työkalut, jotka oli tunnistettu dokumenttianalyysin avulla mahdollisiksi vaihtoehdoiksi tiedonsiirtoon ETL-prosesseissa. Tämän jälkeen tunnistettujen vaihtoehtojen listaa lähdettiin tarkastelemaan kohdeorganisaation tarpeiden, vaatimuksien ja käyttötarkoituksen perusteella, jonka avulla saatiin karsittua sopimattomat ratkaisuvaihtoehdot pois. Kohdeorganisaation tarpeet, vaatimukset ja käyttötarkoitukset tunnistettiin pidetyistä työpajakeskusteluista. Karsitun listan ETL-

työkalujen ominaisuuksia lähdettiin analysoimaan kohdeorganisaation näkökulmasta, jossa painotettiin kohdeorganisaation käyttötarpeita ja -tapauksia, johon ETL-työkalun tulee taipua ja johon se tuo lisäarvoa.

Alustavien tarkastelujen jälkeen otettiin kolme ETL-työkalua jatkotarkasteluun, jossa ETL-työkaluille luotiin vertailutaulukko, johon koottiin olennaiset ominaisuudet, jotka oli tunnistettu työpajakeskusteluissa WMS-järjestelmän ETL-prosessin ETL-työkalun vaatimuksiksi. Taulukkoon kootut ominaisuudet kirjattiin kunkin työkalun kohdalla, jonka pohjalta muodostettiin lopullisen vertailevan analyysin huomiot ja päätelmät ratkaisujen sopivuudesta kohdeorganisaation käyttöön ja toimintaan.

Löytyneistä aineistoista ja tieteellisistä julkaisuista luotiin kattava lista mahdollisista vaihtoehdoista ETL-prosesseissa käytettävistä työkaluista, teknologioista ja muista menetelmistä. Tämän kattavan, suodattamattoman listan vaihtoehtojen ominaisuuksia käytiin läpi kohdeorganisaatiossa pidetyissä työpajakeskusteluissa, joissa saatiin muotoiltua tarpeet ja vaatimukset, joita ratkaisuvaihtoehdolta vaadittiin. Keskusteluiden ja viestinnän perusteella tehtiin suodatettu lista varteenotettavista vaihtoehdoista kohdeorganisaation toimintaympäristöön ja käyttötapauksiin, joita lähdettiin analysoimaan ja tarkastelemaan ratkaisulta vaadittavien ominaisuuksien kautta.

Vaatimukset täyttävät ratkaisuvaihtoehdot ja niiltä vaaditut ominaisuudet listattiin taulukkoon, jossa niitä arvioitiin ratkaisuvaihtoehdoilta vaadittujen ominaisuuksien pohjalta. Listauksen perusteella vaihtoehdoille tehtiin vertailevaa analyysia vaatimuksiin peilaten, ja tämän pohjalta ratkaisuvaihtoehdot saatiin arvioitua. Listauksen vaihtoehtoja esitettiin vastauksiksi tutkimuskysymyksiin.

Työpajakeskusteluissa tunnistettiin WMS-järjestelmän ETL-prosessin ETL-työkalun vaatimukset, joita ratkaisuvaihtoehdolta odotettiin. Työpajakeskusteluissa muodostettiin ETL-aihepiirialueen asiantuntijoiden kanssa keskeiset ominaisuustarpeet, jotka nähtiin tärkeimpinä tarpeina ratkaisulle. Tutkimuksen vertaileva selvitysosuus toteutettiin työpajakeskusteluiden pohjalta, jossa etsittiin vaihtoehtoisia ratkaisuja kohdeorganisaation ETL-prosessin nykyiselle tiedonsiirtomenetelmälle. Dokumenttianalyysin avulla saatiin koottua aihepiirin lähdekirjallisuudessa mainitut ETL-työkalut, jotka oli yleisesti tunnistettu mahdollisiksi vaihtoehdoiksi tiedonsiirtoon ETL-prosesseissa. Tämän jälkeen tunnistettujen vaihtoehtojen listaa lähdettiin tarkastelemaan kohdeorganisaation työpajakeskusteluissa tunnistettujen tarpeiden, vaatimuksien ja käyttötarkoituksen perusteella, jonka avulla saatiin karsittua sopimattomat ratkaisuvaihtoehdot pois. Karsitun listan ETL-työkalujen ominaisuuksia lähdettiin analysoimaan kohdeorganisaation näkökulmasta, jossa painotettiin

kohdeorganisaation työpajakeskusteluissa tunnistettuja ominaisuusvaatimuksia sekä käyttötarpeita ja -tapauksia, johon ETL-työkalun tulee taipua ja johon se tuo lisäarvoa.

2.3 Tutkimuksen rajoitteet

Tutkimus toteutettiin tapaustutkimuksena kohdeorganisaation toimintaympäristöön ja toimintaan kohdentaen. Tutkimuksen selvitystyö tehtiin analysoimalla kirjallisia julkaisuja sekä erilaisia selvitys- ja tutkimustöitä. Tutkimuksen vertaileva analysointi perustui kerättyyn tietoon ja työpajakeskusteluissa tunnistettuihin ominaisuuskriteereihin, jossa ETL-työkalun tarpeita tarkasteltiin kohdeympäristön näkökulmasta heidän vaatimien arviointikriteerien pohjalta ja heidän käyttötapauksissansa esiintyvien tarvittavien ominaisuuksien kautta.

Tutkimuksen ratkaisuvaihtoehtojen vertailevat analyysit toteutettiin kohdeorganisaation toimintaan peilaten heidän näkökulmastaan, jossa huomioitiin myös tutkijan ja kollegoiden tulkintoja käyttötapauksista ja niihin tarvittavan ratkaisun ominaisuuksista, joita vaihtoehtoiselta ratkaisulta vaaditaan. Ratkaisuvaihtoehtojen analysointi perustui julkaisuihin, joita etsittiin ja tarkasteltiin dokumenttianalyysin avulla alan aihepiirien parista. Tunnistettuja ratkaisumenetelmiä ei testattu todellisessa käytössä oikealla datalla kohdeorganisaation toimintaympäristössä, joten ratkaisuvaihtoehdoille ei saatu konkreettisia tuloksia ominaisuuksien sopivuudesta kohdeorganisaation toimintaan erilaisille käyttötapauksille.

3. ETL-PROSESSI OHJELMISTOTUOTANNOSSA

Tässä luvussa paneudutaan tutkimuksen aihepiiriin liittyvään kirjalliseen teoriaan, mikä auttaa ymmärtämään tutkimusongelmaa syvemmällä tasolla teorian kautta ja antaa tapaustutkimukselle teoreettisen selkänojan, joka auttaa syventämään ymmärrystä tutkittavasta tapauksesta. Taustateoriaa esitellään kirjallisuuskatsauksen avulla, johon tutustutaan tutkimusongelmaan liittyvien, tunnistettujen aihealueiden kautta, mikä tukee ja rakentaa pohjaa vaihtoehtoisten menetelmien selvittelyyn kirjallisen teorian pohjalta. Teoreettista taustaa esitellään ensin korkealta tasolta, jossa alustetaan ohjelmistotuotannon kautta ETL-prosessia, ja aihepiiri kerrallaan paneudutaan tarkemmin tutkimuksen aihealueen ytimeen – ETL-prosessiin ja sen lainalaisuuksiin.

3.1 Ohjelmistotuotanto

Ohjelmistotuotanto tai ohjelmistotekniikka (engl. *software engineering*) on nimensä mukaisesti ohjelmistojen tuotantoon keskittyvä prosessi, jossa järjestelmällisten vaiheiden kautta kehitetään ohjelmistoja. Ohjelmistotuotanto on kokonaisvaltainen prosessi, mikä kattaa ohjelmiston tuotantoon tarvittavat tunnistetut vaiheet. Se käsittää ohjelmiston tuotantoprosessin määrittelyn ja suunnittelun, toteutuksen ja kehityksen sekä tarvittavat testaukset. Ohjelmistotuotantoon sisältyy myös ohjelmiston käyttöönotto ja ylläpito sekä dokumentointi ja tuotteenhallinta. (Haikala & Märijärvi 2006)

Ohjelmistotuotanto tarjoaa ohjelmiston kehitykseen tarvittavat mallit ja viitekehykset sekä kattaa myös ohjelmiston kehityksen jälkeisen ajan ylläpidon ja päivitysten näkökulmasta. Ohjelmiston elinkaarenhallinta edellyttää virheiden korjaamista, päivitysten ajamista, uusien ominaisuuksien lisäämistä sekä muita muutostöitä esimerkiksi tietoturvallisuuteen liittyen. Ohjelmistotuotanto yhdistää ohjelmiston kehitysprosessit ja valmistamisen osaksi organisaation liiketoimintaa tarjoten liiketoiminnallisen näkökulman ohjelmistojen valmistukseen ja kehitykseen. (Haikala & Märijärvi 2006)

Ohjelmistotuotanto voidaan jakaa osa-alueisiin, joista yksi tunnistettu alue kuvaa ohjelmistokehitystä, jossa keskitytään nimensä mukaisesti ohjelmiston kehitystyöhön liittyviin menetelmiin ja toimintoihin. (Haikala & Märijärvi 2006) Ohjelmistokehitystyöhön liittyviä lainalaisuuksia ja osakokonaisuuksia tarkastellaan tarkemmin seuraavassa alaluvussa 3.1.1.

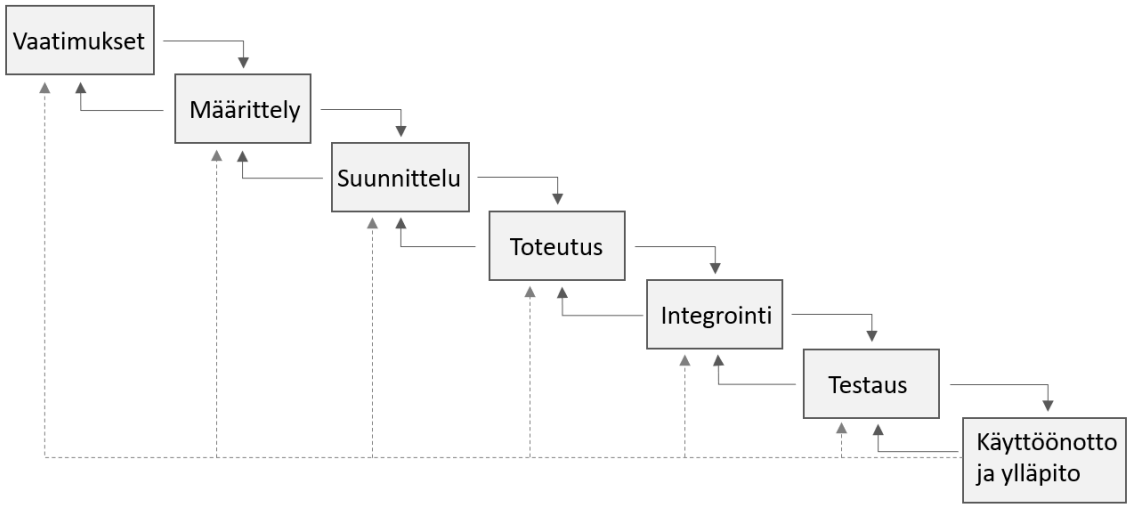
3.1.1 Ohjelmistokehitys

Ohjelmistokehitys perustuu ohjelmistoarkkitehtuuriin (kts. luku 3.1.4), joka ohjaa ohjelmiston kehitystyöprosessia. (Koskimies & Mikkonen 2005) Ohjelmistokehitys on prosessi, jossa tiettyyn ongelmaan tai tiettyyn tarpeeseen kehitetään tietokonepohjainen ohjelmisto. Ennen ohjelmiston kehitystyötä ongelmaa kartoitetaan, jotta saadaan tarkempi käsitys, mitä ohjelmistolta oikeasti halutaan. Selvityksen vaatimuksien pohjalta ongelmaa aletaan mallintamaan ja määrittelemään ohjelmistolle näkyvyysaluetta, ratkaisun laajuutta. Ohjelmiston suunnittelu perustuu ohjelmistoarkkitehtuuriin, jossa kuvataan ohjelmiston toiminnallisuuksien ja ratkaisujen toteutukset tarkemmalla tasolla. Lopulta ohjelmisto toteutetaan arkkitehtuurisuunnitelman perusteella kattavasti testaillen ennen käyttöönottoa. Ohjelmistokehitysprosessi käsittää myös ohjelmiston julkaisun jälkeisen ajan pitäen sisällään ohjelmiston ylläpitotyöt, päivitystyöt sekä tarvittavat muutostyöt uusien, esiinnousseiden tarpeiden pohjalta. (Dooley 2017)

Myös Haikala & Märijärvi (2006) tunnistavat ohjelmistokehitysprosessista erottuvat osaprosessit, jotka he kuvaavat määrittely-, suunnittelu-, ohjelmointi-, testaus-, käyttöönotto- ja ylläpitoprosesseiksi. Näiden osaprosessien toimintoja tuetaan ja vahvistetaan erilaisten tukitoimintojen avulla, jotka liittyvät esimerkiksi laadunvarmistukseen, tuotteenhallintaan, vaatimustenhallintaan, riskienhallintaan ja dokumentointiin. (Haikala & Märijärvi 2006)

Narang & Mittal (2022) tunnistavat ohjelmistokehityksen elinkaariprosessista viisi päävaihetta. Prosessi alkaa vaatimusten keräämisellä ja ymmärryksellä, jota seuraa vaatimusten suunnittelu. Suunnitteluvaiheessa etsitään useita vaihtoehtoisia ratkaisuehdotuksia, minkä jälkeen lähdetään toteuttamaan lopullisen suunnitelman koodaamista. Tuotteen testausvaiheessa varmistetaan virheetön tuote, joka testauksen jälkeen otetaan käyttöön tuotantoympäristössä ja tuote liitetään osaksi ylläpitoprosessia. (Narang & Mittal 2022)

Prosessimallit jakavat ohjelmiston kehitystyötä tunnistettujen vaiheiden ja osaprosessien kautta erillisiin työvaiheisiin. Prosessimallia voidaan havainnollistaa vaihejakomallilla, joka kuvaa ohjelmiston kehitysprosessin työvaiheita (Haikala & Märijärvi 2006). Kuva 4 esittää yhtä tunnistettua vaihejakomallista kehitettyä versiota – vesiputousmallia. Roycen (1970) kehittämä vesiputousmalli on yleisesti tunnistettu alan kirjallisuudessa (muun muassa Narang & Mittal 2022; Dooley 2017; Haikala & Märijärvi 2006) ja se on edelleen laajassa käytössä ohjelmistokehitysprojekteissa. (Narang & Mittal 2022)



Kuva 4. Ohjelmistokehitysprosessi - vesiputousmalli (mukaillen Royce 1970)

Ohjelmiston kehitystyö alkaa asiakasvaatimuksien määrittelyllä. Määrittelyvaiheessa muunnetaan asiakasvaatimukset ohjelmistovaatimuksiksi; määritellään ratkaistava ongelma tai tarve, josta seuraa ohjelmistovaatimukset toteuttavalle ohjelmistolle. Toiminnallisten ominaisuuksien lisäksi määritellään projektin yleiset ehdot, muun muassa projektin sisältö, budjetti sekä muut reunaehdot projektille. Määrittelyvaiheen jälkeen projektille on olemassa dokumentoitu toiminnallinen määrittely, jossa kuvataan ohjelmiston toiminnot, ei-toiminnalliset vaatimukset sekä rajoitukset. Määrittelyvaihe havainnollistaa vastauksen kysymykseen: ”mitä järjestelmä tekee?”. (Haikala & Märijärvi 2006)

Määriteltyjen ohjelmistovaatimuksien toteutus kuvataan suunnitteluvaiheessa, jossa määritellään järjestelmän tekniset vaatimukset. Arkkitehtuurisella järjestelmämallilla kuvataan, miten määritellyillä ohjelmistovaatimuksilla järjestelmä toteutetaan mahdollisimman tarkasti täyttäen asetetut vaatimukset. Suunnitteluvaiheesta voidaan erottaa arkkitehtuurinen suunnittelu, jossa järjestelmä jaetaan toisistaan riippumattomiin itsenäisiin osiin, moduuleihin. Tarkemmin näitä tunnistettuja moduuleita kuvataan moduulisuunnitteluvaiheessa, jossa suunnitellaan moduulien sisäinen rakenne ja niiden käyttämä toiminnallisuus. Suunnitteluvaiheessa selvitetään vastaus kysymykseen ”miten järjestelmä suorittaa tehtävänsä?”. (Haikala & Märijärvi 2006)

Toteutusvaiheessa ohjelmoidaan ohjelmistoa laadittujen suunnitelmien pohjalta. Ohjelmointivaiheessa varmistetaan myös yksittäisten moduulien toiminta yksikkötestauksella, jotta havaitaan näiden loogisten kokonaisuuksien mahdollinen toimimattomuus jo ohjelmointivaiheessa. Järjestelmän kokonaisvaltainen testaus suoritetaan testausvaiheessa. (Dooley 2017; Haikala & Märijärvi 2006)

Järjestelmän osien yhteen liittäminen tehdään integrointivaiheessa, jossa yhdistetään itsenäiset järjestelmän osat, moduulit, kokonaiseksi yhteen toimivaksi järjestelmäksi. (Haikala & Märijärvi 2006)

Testausvaiheessa suoritetaan järjestelmätestaus, jossa varmistetaan ohjelmiston kokonaisvaltainen toiminen määrittely- ja suunnitteludokumenttien pohjalta. Testausvaiheessa testataan vielä yksittäisten moduulien toimivuus, niiden integrointi ja yhteistoiminta sekä koko järjestelmän toiminta ja suorituskyky. (Haikala & Märijärvi 2006)

Käyttöönottovaiheessa valmis ohjelmisto siirretään tuotantokäyttöön (Narang & Mittal 2022) ja ylläpitovaiheessa selvitetään asiakkaan ongelmia, korjataan virheitä, muutetaan ohjelmiston toimintaperiaatteita vaatimuksien muuttuessa sekä lisätään uusia ominaisuuksia uusien asiakastarpeiden mukaan. (Haikala & Märijärvi 2006)

Jokaiseen vaiheeseen sisältyy laadunvarmistustoimenpiteitä, kuten tarkastuksia, katselmuksia ja testausta, joiden tavoitteena on havaita ja korjata virheet järjestelmästä mahdollisimman varhaisessa vaiheessa. Tällä tavalla vältytään isoista, epämiellyttävistä yllätyksistä sekä pyritään vähentää kehitysprojektin kokonaiskustannuksia. (Haikala & Märijärvi 2006)

Prosessimallit ohjaavat ohjelmistoprojektin etenemistä sekä määrittää, missä järjestyksessä ohjelmiston eri kehitysvaiheet toteutetaan. Vesiputousmallia voidaan pitää ohjelmistokehitystoiminnan taustamallina, mutta nykyaikana sen heikkouksia on myös kritisoitu. (Narang & Mittal 2022; Dooley 2017; Haikala & Märijärvi 2006) Vesiputousmallissa alun vaatimuksien määrittely ohjaa vahvasti prosessia ja yleensä vaatimuksia on mahdotonta määritellä täydellisesti ennen projektia tai vaatimukset muuttuvat projektin aikana. Myös vesiputousmallin laaja dokumentointiin liittyvä työmäärä on huomioitu tiukkana rajoitteena mallille. (Narang & Mittal 2022; Dooley 2017; Haikala & Märijärvi 2006).

Näiden huomioiden pohjalta on kehitetty ketteriä ohjelmistokehitysmenetelmiä, jotka huomioivat asiakkaan vaatimuksien muuttumisen ohjelmistokehitysprojektin edetessä ja niiden tarkennukset ohjelmiston kehitysprosessin aikana. Tunnettuja ja yleisesti käytössä olevia ketteriä ohjelmistokehitysmenetelmiä ovat muun muassa Scrum-, Kanban- ja XP-menetelmät. (Narang & Mittal 2022; Dooley 2017; Mathura et al. 2015) Viimeisin ketterä ohjelmistokehitysmenetelmänä tunnistettu DevOps-toimintamalli tuo jatkuvuutta ohjelmistokehitykseen yhdistäen kehityksen ja käytännön toiminnot joustavasti yhteen. Siinä yhdistyvät jatkuvat integroinnit ja toimitusmenetelmät, jotka mahdollistavat palveluiden ja ohjelmistojen nopean toimituksen. (Krief 2022)

Ohjelmistokehitystyön lainalaisuuksia hyödynnetään ETL-prosessien kehityksessä (kts. luku 3.3), jossa on myös tunnistettu ohjelmistotuotannon periaatteet ja hyödynnetään sen tarjoamia malleja ja viitekehyksiä kokonaisvaltaiseen ETL-prosessien kehitykseen. Tietojenkäsittelyn näkökulmasta ohjelmistokehitys ja ETL-prosessit kulkevat rinnakkain, sillä molemmat ovat olennaisia osia tietojärjestelmien kokonaisvaltaisessa kehitystyössä.

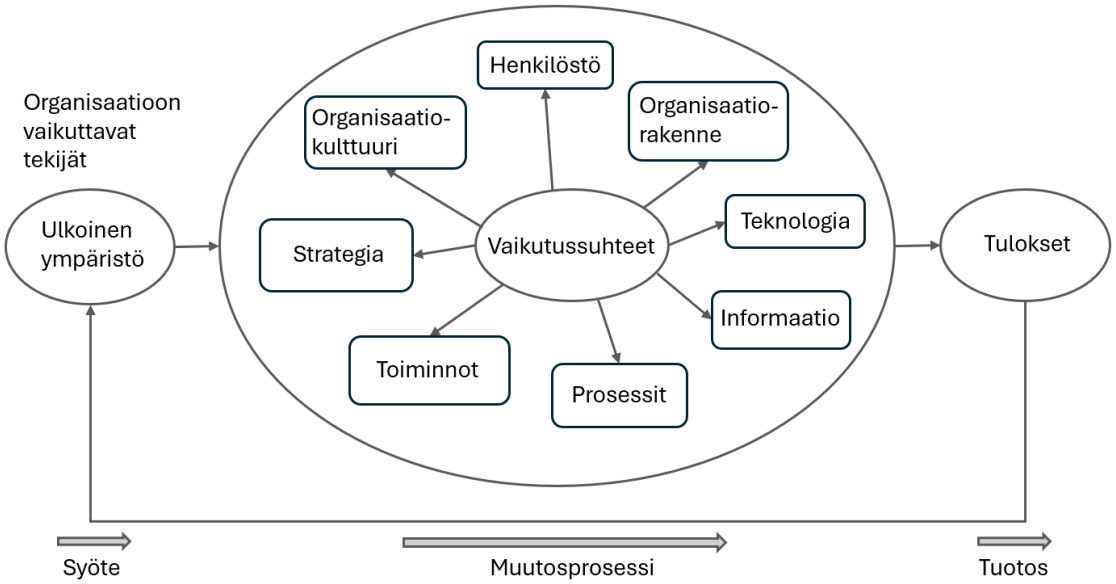
3.1.2 Kokonaisarkkitehtuuri

Arkkitehtuuri havainnollistaa jonkun asian kokonaisuutta ja kuvaa sen osien rakennetta ja osien välisiä suhteita. Referenssiarkkitehtuuri keskittyy jonkun spesifisen asian rakenteen suunnitteluun ja kehittämiseen, mikä auttaa ymmärtämään tätä asiaa syvällisemmin, tarkemmalla tasolla. Referenssiarkkitehtuuri auttaa ymmärtämään jo olemassa olevaa, luotua rakennetta sekä helpottaa uuden ominaisuuden kehityksessä, kun tunnistetaan käytettyjen menetelmien ja toimintojen merkitys uutta luodessa. (Ariyachandra & Watson 2010) Referenssiarkkitehtuurin määritelmää hyödyntäen voidaan kuvata organisaation kokonaisarkkitehtuuria, jonka avulla saadaan käsitys organisaation rakenneosista ja niiden välisistä suhteista sekä kokonaisarkkitehtuurin tarjoamasta viitekehyksestä, jota organisaatiot hyödyntävät omassa toiminnassaan.

Kokonaisarkkitehtuuri tarjoaa kokonaisvaltaisen yleiskuvan organisaation keskeisistä toiminnoista. Kokonaisarkkitehtuuri vastaa yleisellä tasolla primitiivisiin kysymyksiin organisaation toiminnasta sekä toisaalta kuvaa konkreettisesti näiden toimintojen toteutusta. (Doomen 2023) Kokonaisarkkitehtuurin tavoitteena on parantaa organisaation toimintaa tarjoamalla kokonaisvaltainen ja yhtenäinen kuva organisaation strategian, liiketoiminnan, tiedon ja teknologian muodostamasta kokonaisuudesta. (Bernard 2005)

Kaisler et al. (2005) toteavat organisaation kokonaisarkkitehtuurin kuvaavan liiketoiminta- ja hallintoprosessien sekä tietotekniikan välisiä suhteita. Siinä esitellään nykyisten ja tulevaisuudessa toivottujen suhteiden kokonaisvaltaiset kuvaukset ja dokumentoinnit. Kokonaisarkkitehtuuri on organisaatiolaajuinen läpikuvaus, jossa organisaation toimintaa hallinnoidaan liiketoiminnan, tiedonhallinnan ja tietotekniikan muodostaman kokonaisuuden avulla. (Kaisler et al. 2005) Myös Bernard (2005) näkee kokonaisarkkitehtuurin kokonaisvaltaisena tarkasteluna organisaation toiminnasta, joka esittelee organisaation nykytilaa ja tulevaisuuden tavoitteita. Kokonaisarkkitehtuuri sisältää organisaation strategian, liiketoiminnan ja teknologian näkökulmat organisaation suunnitelmiin ja päätöksiin liittyen tarjoten kolmikantaisen, holistisen yleisnäkymän asetettujen tavoitteisiin pääsemisen tueksi. (Bernard 2005)

Rood (1994) määrittelee kokonaisarkkitehtuurin abstraktiksi viitekehykseksi, joka kuvaa organisaation ensisijaiset rakenneosat ja niiden väliset suhteet sekä organisaation keskeiset komponentit ja niiden keskinäiset suhteet. Kokonaisarkkitehtuurissa huomioidaan myös organisaation toimintaan vaikuttavat ulkoiset tekijät ja niiden vaikuttavat muutostekijät. Kokonaisarkkitehtuuri pyrkii havainnollistamaan näiden rakenneosien ja komponenttien muodostamia kokonaisuuksia sekä niiden vuorovaikutuksia. Kokonaisarkkitehtuuri tarjoaa rakenteellisen kehyksen organisaation eri toiminnoista ja pyrkii yhdistämään organisaation eri tasot, jotta mahdollistetaan kaikkien toimintojen optimaalinen yhteistyö. (Rood 1994) Kuvassa 5 esitellään organisaation kokonaisarkkitehtuurin keskeiset ominaispiirteet ja toimintaa tukevat elementit, jotka mahdollistavat organisaation eri toimintojen keskinäisen yhteistyön.



Kuva 5. Kokonaisarkkitehtuurin viitekehys (mukaillen Rood 1994)

Kuvaa 5 kokonaisarkkitehtuurin viitekehyksestä voidaan ajatella ETL-prosessina (kts. luku 3.2), jossa syöte toimii raakadatana, jota muokataan ja käsitellään tarpeiden mukaan muutosprosessissa, ja josta saadaan haluttu tuotos tuloksena. Kokonaisarkkitehtuuriin linkittyvät kokonaisuudet, esimerkiksi teknologia, informaatio, prosessit ja organisaation strategia ovat kaikki keskeisiä elementtejä ETL-prosessissa ja siihen vaikuttavia tekijöitä. Tietojenhallinta on osa kokonaisarkkitehtuuria ja ETL-prosessit tarjoavat siihen omat tunnusomaiset toimintonsa.

Kokonaisarkkitehtuurista voidaan erottaa keskeisinä, erillisinä arkkitehtuurityyleinä liiketoiminta-, teknologia-, järjestelmä- ja tietoarkkitehtuurit, jotka sisältävät oman toimintakenttänsä mukaiset toiminnot ja vastaavat niiden suunnittelusta ja hallinnasta. Liiketoiminta-arkkitehtuuri sisältää liiketoimintaprosesseja tukevat prosessit, menetelmät, organisaation ihmiset ja rakenteet sekä organisaatiokulttuurin. Teknologia-

arkkitehtuuri hallinnoi organisaation komponentteja, laitteistoja, ohjelmistoja, pilvipalveluita ja verkkojärjestelmiä. Teknologia-arkkitehtuuri auttaa hahmottamaan organisaation teknologiset resurssit ja ohjaa järjestelmien ja ohjelmistojen kehittämistä, ylläpitoa ja strategian mukaista yhtenäistä käyttöä liiketoimintaa tukien. (Bernard 2005; Kaisler et al. 2005; Rood 1994) Teknologia-arkkitehtuuri käsittää organisaation teknisen infrastruktuurin, sen suunnittelun ja hallinnoinnin. Organisaation tietotekninen infrastruktuuri muodostaa jalustan kehitettävälle teknologialle, mikä mahdollistaa tietojärjestelmien ja ohjelmistojen suunnittelun ja toteutuksen. Tietotekninen infrastruktuuri muodostaa perustan organisaation teknologiselle kehittymiselle. (Laudon & Laudon 2007)

Järjestelmäarkkitehtuuri käsittää organisaation palveluita tukevat järjestelmät, niiden väliset linkitykset ja integraatiot organisaation jokaisella tasolla. Se avaa järjestelmien keskeisiä toimintoja ja selkeyttää järjestelmien välisiä integraatioita, joita kuvataan tarkemmin integraatioarkkitehtuurin avulla (kts. luku 3.9.2). Tietoarkkitehtuuri kuvaa organisaation tietovarastot ja tietomallit sekä sen avulla hallinnoidaan organisaation tietoturvapolitiikkaa määräystenmukaisesti. Tietomallit auttavat tietojärjestelmien sisältävien tietojen esityksissä avaten tietojen virtauksia järjestelmissä, tietojen välisiä suhteita ja niiden rakenteita. Tietoarkkitehtuuri käsittää organisaation hallussa olevan tiedon ja varmistaa sen eheyden organisaation säännösten mukaiseksi yhtenäiseksi kokonaisuudeksi yhteisten toimintamallien avulla ja auttaa ymmärtämään, mitä tietoja henkilöstö tarvitsee suorittaakseen työtehtäviä tai mitä tietoja liiketoimintaprosessit tarvitsevat toimiakseen tarkoituksenmukaisesti. (Bernard 2005; Kaisler et al. 2005; Rood 1994) Tietoarkkitehtuurin avulla valjastetaan organisaation arvokas tietopääoma jäsennellyksi ja hallittavaksi kokonaisuudeksi, mikä on hyödynnettävissä organisaation jokaisella tasolla. (Hovi et al. 2009, s. 66)

Kokonaisarkkitehtuuria voidaan kuvata strategisena lähestymistapana, jatkuvasti kehittyvänä prosessina, organisaation kokonaisvaltaiseen toimintaan, joka reagoi muuttuviin tarpeisiin. Sen päämääränä on ymmärtää, suunnitella ja hallinnoida organisaation monimutkaisia rakenteita, kuten liiketoimintaprosesseja, tietojärjestelmiä, teknologiainfrastruktuuria sekä henkilöstöä ja organisaation kulttuuria. Kokonaisarkkitehtuurin avulla pyritään luomaan yhtenäinen näkemys organisaation toiminnoista, resursseista ja tavoitteista, jossa yhdistyy liiketoiminnan, teknologian ja tietoresurssien luomat mahdollisuudet päätöksentekoon. (Bernard 2005; Kaisler et al. 2005; Rood 1994)

Kokonaisarkkitehtuurin yksi tehtävistä on siis avata tietojärjestelmien suunnittelua ja niiden hallinnointia, kuvata niiden toimintoja ja niiden välisiä suhteita.

Kokonaisarkkitehtuuri auttaa hahmottamaan tietojärjestelmien tietovirtoja sekä avaa niiden hallintaa kuin myös tietojen laatua ja tarvittavia tietomalleja. ETL-prosessit ovat olennainen osa tietojärjestelmiä ja niiden tietojenkäsittelyä (kts. luku 3.2) ja ETL-prosessit linkittyvät kokonaisarkkitehtuurin avulla havainnollistettujen kokonaisuuksien myötä osaksi tietojärjestelmien suunnittelua, kehitystä ja käyttöä.

3.1.3 Tuotearkkitehtuuri

Tuotearkkitehtuuri mallintaa tuotteen käsitteellistä olemusta, jossa tuotteen toiminnalliset komponentit järjestetään fyysisiksi rakenteiksi ja kuvaa näiden rakenteiden vuorovaikutusta keskenään. Tuotearkkitehtuuri ryhmittelee tuotteen funktionaaliset toiminnot ja kuvaa ryhmitellyt toiminnot fyysisinä komponentteina, jotka on järjestelty toiminnallisuuksien mukaan. Tuotearkkitehtuuri määrittelee ja kuvaa näiden fyysisten komponenttien vuorovaikutussuhteita, esimerkiksi komponenttien välistä viestintää, ja havainnollistaa koko tuotteen toimintaa, kuinka tuotteen eri osat liittyvät toisiinsa tuotteen toiminnan mahdollistamiseksi. Tuotearkkitehtuuri määrittelee tuotteen käyttöympäristön ja siihen liittyvät vuorovaikutussuhteet. (Ulrich 1995)

Fowler (2003) esittelee teoksessaan tuotearkkitehtuuria ja sen ominaispiirteitä ja kuvaa, kuinka tuotearkkitehtuurin avulla voidaan havainnollistaa tuotekehitysprosessia, mikä helpottaa tuotekehitysprosessin suunnittelemista ja tuotteiden hallintaa. Fowler (2003) tunnistaa tuotearkkitehtuurin ja ETL-prosessien välisen yhteyden tietojärjestelmien suunnittelussa ja kehityksessä, tarkemmin tietojärjestelmien taustalla hyödynnettävien tietolähteiden kautta. Tietolähteiden logiikka liittyy kommunikointiin muiden järjestelmien kanssa, ja useimmissa sovelluksissa tietolähteen logiikkana hyödynnetään tietokantaa, joka on pääasiassa vastuussa pysyvien tietojen tallentamisesta. Tuotearkkitehtuuri määrittelee järjestelmän rakenteen ja komponenttien väliset suhteet, kun taas ETL-prosessit ovat vastuussa tietojen keräämisestä, muokkaamisesta ja latauksesta. Tuotearkkitehtuuri tarjoaa puitteet, jotka mahdollistavat ETL-prosessin toteuttamisen osana arkkitehtuuria. (Fowler 2003)

Tuotearkkitehtuurin määritelmää hyödyntäen ja kohdistaen tutkimuksen viitekehyksessä kohdeorganisaation tuotteeseen, voidaan kuvata WMS-järjestelmää vielä tarkemmalla tasolla. Seuraavaksi jatketaan tuotteen, tässä tapauksessa ohjelmiston, kokonaisuuden hahmottamista ohjelmistoarkkitehtuurikuvauksen mukaan.

3.1.4 Ohjelmistoarkkitehtuuri

Ohjelmistojen osia, niiden toimintoja, suhteita ja kehittymistä kuvataan ohjelmistoarkkitehtuurin viitekehyksillä. (Koskimies & Mikkonen 2005)

Ohjelmistojen rakenteellisia komponentteja, niiden toiminnallisuuksia, keskinäisiä suhteita ja kehittymistä tarkastellaan ohjelmistoarkkitehtuurin viitekehyksien avulla. (Koskimies ja Mikkonen 2005) Ohjelmistoarkkitehtuuri käsittää ohjelmistojen rakenteet, komponentit ja niiden väliset suhteet sekä sisältää organisaation sovelluksien, käyttöjärjestelmien ja muiden ohjelmistotyökalujen väliset integraatiot. Ohjelmistoarkkitehtuuri avaa järjestelmien välisiä integraatioita organisaatiotason läpileikkauksina ja selkeyttää organisaation käytössä olevien järjestelmien välisiä linkityksiä.

Järjestelmän ohjelmistoarkkitehtuuri kuvaa järjestelmän rakenteita, joista järjestelmä koostuu. Rakenteita voidaan kuvata elementteinä, jotka ovat vuorovaikutuksissa keskenään ja joiden toiminta on riippuvaista toisistaan. Järjestelmän rakenteet käsittävät ohjelmistoelementtejä ja niiden välisiä suhteita sekä selittää niiden ominaisuuksia. Ohjelmistoarkkitehtuuri pyrkii siis kuvaamaan järjestelmän toiminnan mahdollistavat rakenteet ja komponentit. Ohjelmistoarkkitehtuuri voidaan jakaa kolmeen ryhmään, jotka ohjaavat arkkitehtuurin suunnittelua, dokumentointia sekä tarkastelua. (Bass et al. 2022)

Bass et al. (2022) kuvaavat nämä ohjelmistoarkkitehtuurin kolme ryhmärakennetta komponentti- ja niiden liitäntärakenteina, moduulirakenteina sekä näiden kahden ryhmärakenteen kautta muodostuvana jakorakenteena, joka jakaa ohjelmiston sen käyttötarkoituksien mukaan. Ohjelmiston komponentti- ja liitäntärakenteet kuvaavat, miten elementit ovat vuorovaikutuksessa toistensa kanssa ajosuorituksen aikana mahdollistaen järjestelmän toimintojen suorittamisen, moduulirakenteet jakavat järjestelmän toteutusyksiköiksi (moduuleiksi), jotka hahmottavat, kuinka järjestelmä rakentuu koodina tai tietoyksiköinä, jotka on kehitettävä, ja ohjelmiston allokointi-rakenne määrittää ja jakaa järjestelmän ohjelmistorakenteisiin ja ei-ohjelmistorakenteisiin, mikä auttaa määrittämään, miten ohjelmistokomponentit tai moduulit liittyvät fyysisiin, organisatorisiin tai toiminnallisiin osiin kokonaisjärjestelmässä. (Bass et al. 2022)

Ohjelmistoarkkitehtuurilla kuvataan ja perustellaan ohjelmiston ratkaisujen toteutukset, määritellään säännöt ja periaatteet sekä luodaan rajat, joiden puitteissa järjestelmää kehitetään ja ylläpidetään. Tämän kaltaiset säännökset mahdollistavat järjestelmän yhtenäisen toiminnan, auttaa hahmottamaan sen toimintaperiaatteita sekä tukee järjestelmän pitkäaikaista toimintaa. (Koskimies & Mikkonen 2005)

Ohjelmistoarkkitehtuurin avulla kuvataan ohjelmiston keskeiset ratkaisut, joita voidaan tarkastella useasta eri näkökulmasta niiden toimintoihin liittyen. Ohjelmistoarkkitehtuurin avulla voidaan pilkkoa ohjelmiston moduuleita osiin *dekompositio*-periaatteen mukaan, joka jakaa moduulit pieniksi osakokonaisuuksiksi, mikä auttaa ymmärtämään

ohjelmistoyksiköiden toimintaa ja määrittää moduulit myöhempää suunnittelua ja mahdollista toteutusta varten. (Bass et al. 2022) *Aspektiperustainen ohjelmointi* ryhmittelee ohjelmiston rakenteita erilaisten aspektien, rakenteisiin liittyvien ominaisuuksien mukaan ryhmiin, minkä avulla pyritään vähentämään ohjelmiston koodin toistoa ja helpottamaan ohjelmiston ylläpidettävyyttä. (Koskimies & Mikkonen 2005)

Ohjelmistoarkkitehtuuri tarjoaa viitekehyksen, jossa määritellään, miten ohjelmistojärjestelmä on rakennettu, miten sen eri osat ovat järjestelty ja miten ne toimivat yhdessä. (Bass et al. 2022) ETL-prosessit ovat tärkeä osa tietojärjestelmien toimintaa; ne käsittelevät tiedon keräämistä, muokkaamista ja sen siirtämistä eri tietolähteistä tietovarastoon tai muuhun kohdejärjestelmään. (Kimball & Caserta 2011) Ohjelmistoarkkitehtuuri tarjoaa pohjan, jonka avulla tietojärjestelmien ETL-prosessit voidaan toteuttaa tehokkaasti skaalautuen organisaation toimintaan. Se määrittelee esimerkiksi, miten tietokannat ja muut tietolähteet integroidaan osaksi järjestelmän toimintaa, miten tietoja käsitellään ja muunnetaan prosessoinnin aikana sekä miten lopulliset tiedot tallennetaan ja jaetaan muille järjestelmän osille. (Bass et al. 2022; Kimball & Caserta 2011)

Yksi esimerkki kirjallisuudessa (esimerkiksi Bass et al. 2022; Dooley 2017; Richards 2015; Mikkonen & Koskimies 2005) tunnistetuista ohjelmistoarkkitehtuurin arkkitehtuurimalleista on kerrosarkkitehtuuri, joka jakaa järjestelmän toiminnot ja vastuut tasoihin kerroksien mukaan. Kerrosarkkitehtuurimalli auttaa määrittelemään ohjelmiston perusominaisuuksia kerroksittain kuvattuna ja havainnollistaa ohjelmiston käyttäytymistä syvemmällä tasolla kuin ohjelmistoarkkitehtuuri. (Richards 2015) LeanwareWMS-järjestelmän toiminta pohjautuu kerrosarkkitehtuurimallin lainalaisuuksiin ja tunnusomaisiin piirteisiin.

3.1.5 Kerrosarkkitehtuuri

Arkkitehtuurimallit auttavat määrittelemään sovelluksen perusominaisuudet ja selittävät niiden käyttäytymistä. Arkkitehtuurimallin ominaisuuksien, vahvuuksien ja heikkouksien tunteminen mahdollistaa oikean tyylin valinnan, joka soveltuu sovelluksen toimintaan suunnitellulla tavalla. (Richards 2015)

Kerrosarkkitehtuurimalli on tunnetuin arkkitehtuurimalli. Kerrosarkkitehtuurimalli soveltuu perinteisiin IT- ja organisaatiorakenteisiin, joita löytyy useimmissa organisaatioissa, mikä tekee siitä luonnollisen valinnan useimpiin liiketoiminta-sovelluskehitysprojekteihin. (Richards 2015) Kerrosarkkitehtuuri on arkkitehtuurimalli, jota käytetään järjestelmän rakenteiden hahmottamisen ja ryhmittelyn apuna. Kerrosarkkitehtuuri jakaa järjestelmän eri kerroksiin, joilla on oma vastuualueensa

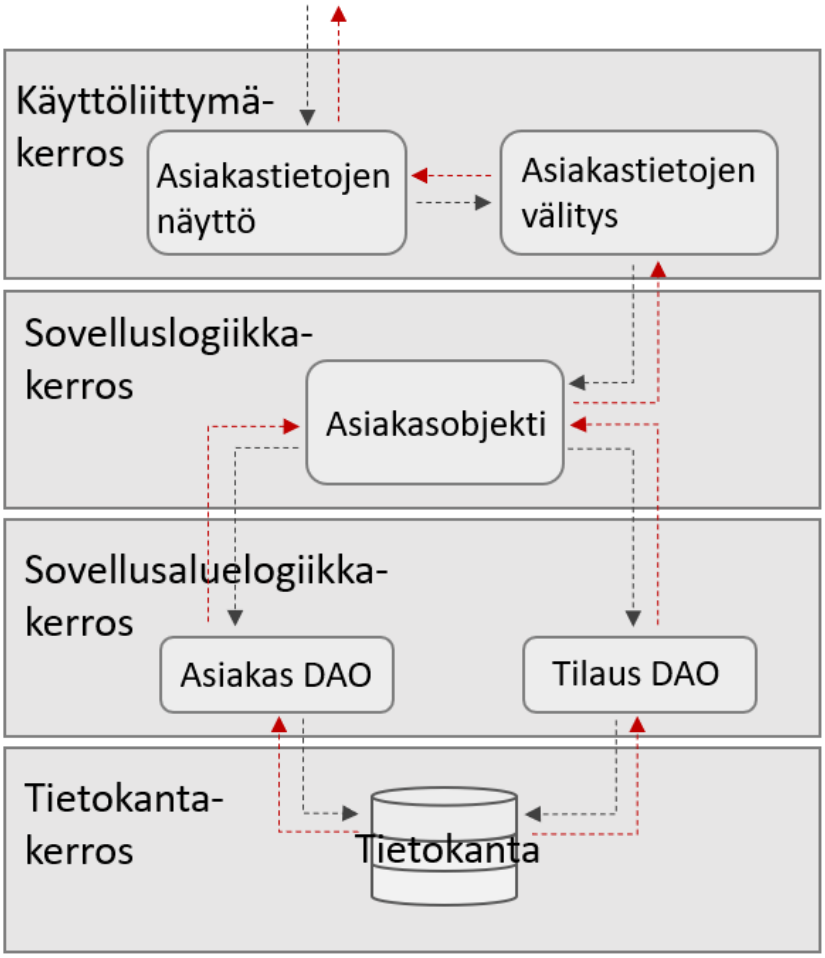
järjestelmän toimintaan liittyen. Järjestelmän kerrokset ovat vuorovaikutuksessa keskenään, missä järjestelmien alempi kerros tarjoaa ylemmälle kerrokselle palveluita ja komponentteja *abstrahointi*-periaatteen mukaan. Ohjelmistokehityksessä *abstrahointi*-periaate viittaa monimutkaisten asioiden ja käsitteiden yksinkertaistamiseen, jossa järjestelmän kerroksien toiminnot on jaettu abstraktiotasojen mukaan, jotka nousevat käyttäjätasoa kohden näyttäen käyttäjälle olennaiset tiedot ja toiminnot graafisessa käyttöliittymässä. (Koskimies & Mikkonen 2005)

Kerrosarkkitehtuurimallissa komponentit järjestetään kerroksiin, joista kullakin kerroksella on tietty rooli sovelluksessa (esimerkiksi esityslogiikka tai sovelluslogiikka). Yleisessä kerrosarkkitehtuurin esitysmallissa on neljä standardi kerrosta, jotka on esitetty kuvassa 6. Kerrosarkkitehtuurimallissa jokaisella kerroksella on oma roolinsa ja omat vastuunsa järjestelmän toimintaan ja toimintoihin liittyen. Esimerkiksi esityslogiikkakerroksen tehtävänä on käsitellä kaikki käyttöliittymään ja selainviestintään liittyvät toiminnot ja esitellä tietoja käyttäjäystävällisessä muodossa näytöllä. (Richards 2015) Bass et al. (2022) näkee kerrosarkkitehtuurityylin osana ohjelmistoarkkitehtuurin moduulirakennekuvausta, jossa jokainen kerros on abstrakti "virtuaalikone", joka tarjoaa yhtenäisen joukon palveluita käyttöliittymän kautta.

Kuvassa 2 esitetty LeanwareWMS-järjestelmän kerrosarkkitehtuuri perustuu kerrosarkkitehtuurin perusajatusmalliin, jossa järjestelmä koostuu kerroksista, joita hyödynnetään järjestelmän toiminnassa. Kerroksien välissä on rajapintoja, jotka kerrokset tarjoavat ja joita kerrokset vaativat toimiakseen. Rajapinnat mahdollistavat kerroksien välisen kommunikoinnin sovitulla tavalla ilman, että kerrokset tuntevat toistensa toimintalogiikkaa. Kuvassa 2 esitetyssä graafisessa muodossa kuvatusta WMS-järjestelmän kerrosarkkitehtuurimallista huomataan sen kerroksien samankaltaisuus Koskimies & Mikkonen (2005) näkemyksiin kerrosarkkitehtuurin graafisen esityksen kerroksista (kuva 6).

Kuva 6. Kerrosarkkitehtuurin graafinen esitys (mukaillen Koskimies & Mikkonen 2005)

Kuva 7 havainnollistaa kerrosarkkitehtuurin toimintaperiaatetta esimerkkitapauksessa, jossa mustilla nuolilla kuvataan käyttäjän tekemää palvelupyyntöä tietokantaan asiakastietojen hakemiseksi. Tässä esimerkkitapauksessa asiakastiedot sisältävät myös tilaustiedot, ja punaisilla nuolilla kuvataan vastausta tähän palvelupyyntöön.



Kuva 7. Kerrosarkkitehtuurin toimintaperiaate (mukaillen Richards 2015)

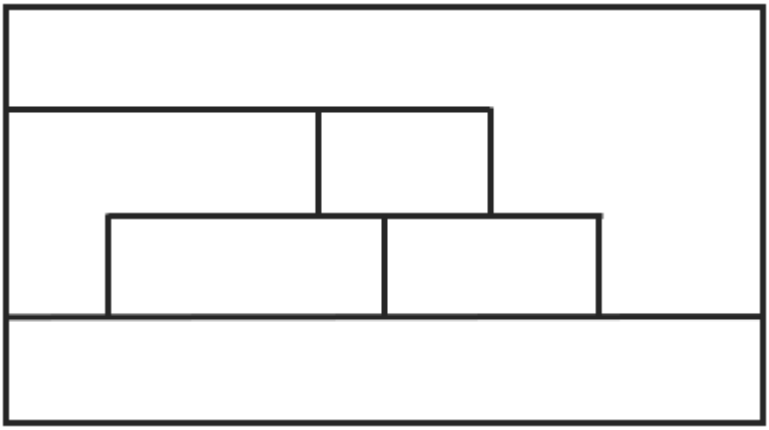
Käyttöliittymäkerroksen asiakastietojen näyttö -moduuli vastaa palvelupyynnön hyväksymisestä ja asiakastietojen näyttämisestä käyttäjälle, mikä on tämän moduulin rooli järjestelmän toiminnassa. Asiakastietojen näyttö -moduuli ei tiedä missä tiedot ovat, miten ne haetaan tai kuinka monta tietokantataulua tietokantakysely yhdistää tietojen saamiseksi, kun palvelupyynnön esittämät tiedot haetaan tietokannasta. Tämänkaltainen toimintatapakuvaus esittää konkreettisesti kerrosarkkitehtuurimallin omien kerroksien vastuualueiden toimintatapaa. Kun asiakastietojen näyttö -moduuli vastaanottaa pyynnön saada tietyn henkilön asiakastietoja, se välittää pyynnön edelleen asiakastietojen välitys -moduulille. Tämä moduuli on vastuussa, että sovelluslogiikkakerroksen moduulit pystyvät käsittelemään kyseisen palvelupyynnön. Asiakastietojen välitys -moduuli myös vastaa, että tarvittavaan moduuliin saa yhteyden ja tietää, mitä tietoja se tarvitsee, jotta se voi suorittaa käyttäjän palvelupyynnön. Sovelluslogiikkakerroksen asiakasobjekti-moduuli on vastuussa kaikkien käyttäjän palvelupyynnön edellyttämien tietojen yhdistämisestä, tässä esimerkkitapauksessa asiakastietojen ja tilaustietojen yhdistämisestä. (Richards 2015)

Asiakasobjekti-moduuli kutsuu asiakkaan Data Access Object -moduulia (DAO) sovellusaluelogiikkakerroksessa saadakseen asiakastiedot sekä myös tilauksen Data Access Object -moduulia tilaustietojen saamiseksi. Nämä Data Access Object -moduulit suorittavat SQL-lauseita hakeakseen palvelupyyntöä vastaavat tiedot ja välittääkseen ne takaisin asiakasobjektille sovelluslogiikkakerrokseen. Toisin sanoen nämä DAO:t ovat yksinkertaisia objekteja, jotka toimivat välikäsinä tietokannan ja sovelluslogiikan välillä. Niiden tehtävänä on säilyttää ja siirtää tietoa tietokannasta sovelluslogiikalle ja päinvastoin, samalla mahdollistaen tiedon tallentamisen tietokantaan. (Richards 2015)

Kun asiakasobjekti vastaanottaa pyydetyt tiedot DAO-moduuleilta, se kokoaa ne yhdeksi kokonaisuudeksi ja välittää tiedot takaisin asiakastietojen välitys -moduulille, joka välittää tiedot asiakastietojen näyttö -moduulille, joka esittää ne graafisessa, visuaalisessa muodossa käyttöliittymän kautta käyttäjälle. (Richards 2015)

Kerrosarkkitehtuurin periaatteena on hyödyntää tietyn tason komponenttia tai palvelua käyttäen hyväksi alemman kerroksen komponentteja tai palveluita. Tästä yleisestä periaatteesta joudutaan kuitenkin useasti poikkeamaan, jolloin palvelukutsu voi siirtyä alemmasta kerroksesta ylempään (engl. *hierarchy breach*) tai palvelukutsu voi ohittaa kerroksia ylhäältä alaspäin kerroksien välissä kulkiessaan (engl. *bridging*). Mahdollisissa kerrosten ohittamisissa haetaan tehokkuusparannuksia tai kun alempi kerros ei pysty suoraan välittämään haluttua palvelua. Palvelukutsun siirtymistä alemmasta kerroksesta ylempään tulisi välttää, varsinkin, jos alempi kerros tulee riippuvaiseksi ylemmästä.

(Koskimies & Mikkonen 2005) Kuvassa 8 on havainnollistettu toisen ja kolmannen kerroksen mahdollinen ohittaminen graafisesti esitettynä.



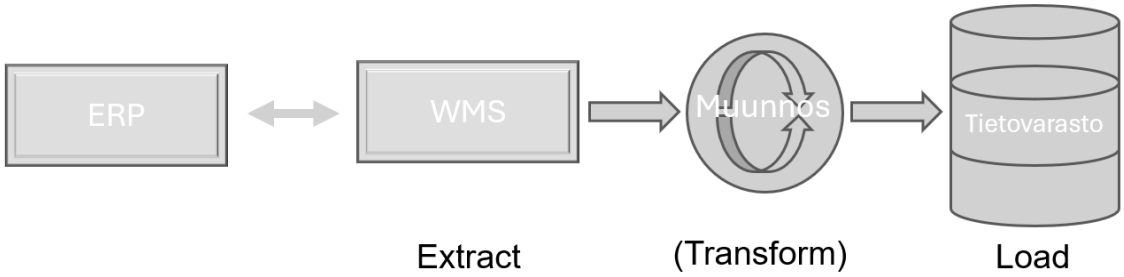
Kuva 8. Kerrosarkkitehtuurin graafinen esitys kerrosten ohittamisesta (mukaillen Koskimies & Mikkonen 2005)

3.2 ETL-prosessi

ETL-prosessi on ratkaisevassa asemassa, kun halutaan siirtää ja käsitellä dataa organisaation sisällä. ETL-prosessi on monimutkainen tiedonsiirtoprosessi, jossa yhdistyy tietovarastointi ja sen hyödyntäminen osana ETL-prosessia. ETL-prosessi on tärkeä aspekti organisaation tietojenhallinnassa, kun halutaan välittää tietoa ja rikastaa raakadataa päätöksenteon tueksi sekä organisaation toiminnan ymmärtämiseksi. (Khan et al. 2024) ETL-menetelmä kehitettiin datan integrointi- ja latausmenetelmäksi tietojen analyyseja ja laskentoja varten. Tietovarastojen yleistyessä, ETL-prosesseista muodostui pääasiallinen toteutusmenetelmä tietojenkäsittelyprosesseihin tietovarastotoiminnoissa. (Khan et al. 2024)

ETL-prosessi mahdollistaa datan siirtämisen järjestelmien ja tietovarastojen välillä. ETL-prosessi kuvaa datan siirtymistä yhden tai useamman järjestelmän ja tietovaraston välillä (kuva 9). ETL-termi muodostuu englannin kielen sanoista *extract* (poimia), *transform* (muokata) ja *load* (ladata), kuvaillen datan käsittelyä prosessin eri vaiheissa. Hyvin suunniteltu ETL-prosessi mahdollistaa tietovarastojen oikeaoppisen hyödyntämisen, missä dataa kerätään, jalostetaan, yhdistetään ja yhdistellään sekä siirretään edelleen järjestelmien ja tietovarastojen välillä. ETL-prosessi täydentää dataa yhdistelemällä eri järjestelmistä saatavaa dataa ja lisää sen arvoa jalostamalla raakadataa tiedoksi, joka on helposti luettavassa muodossa. (Kimball & Caserta 2011) Jos datalähteestä haettu

data ei vastaa kohteen tarvitsemaa dataa, haettu väliaikainen data täytyy muokata vastaamaan kohdejärjestelmän odottamaa muotoa. (Goldfedder 2020)



Kuva 9. Esimerkkikuvaus ETL-prosessista

ETL-prosessi käsittelee dataa liiketoimintasääntöjen mukaisesti täyttääkseen asetettuja liiketoimintavaatimuksia, kuten kuukausittaista raportointia tai monimutkaisempia analytiikkaprosesseja parantaakseen kohdejärjestelmien toimintaa tai loppukäyttäjäkokemuksia. (Khan et al. 2024)

Xavier & Moreira (2013) toteavat, kuinka ETL-prosessi on enemmän kuin pelkästään menetelmä datan siirtämiseen lähdejärjestelmästä tietovarastoon. ETL-prosessi poistaa virheet ja korjaa puuttuvat tiedot sekä yhdistää useista lähteistä peräisin olevaa dataa, jolloin datamallit voidaan analysoida yhdessä. ETL-prosessi tähtää datan rikastamiseen käytettäväksi loppukäyttäjille heidän ehdoillaan. ETL-prosessi käsittää siis muutakin kuin datan virtauksen; ETL-prosessi korjaa datavirheet ja muuntaa raakadatan tiedoksi, jota voidaan hyödyntää organisaation liiketoimintaprosesseissa ja päätöksenteossa. (Xavier & Moreira 2013)

ETL-prosessi jalostaa dataa ja tuo sen käyttäjille ymmärrettävässä, luettavassa muodossa valmiina hyödynnettäväksi. ETL-prosessia hyödynnetään organisaation toiminnan ja palveluiden kehittämisessä ja sen avulla voidaan soveltaa tiedolla johtamisen perusperiaatteita, jossa tiedosta luodaan arvoa, kun tarjolla olevaa tietoa hyödynnetään oikeaoppisesti ja liiketoiminnan päätökset perustuvat totuudenmukaiseen tilannekuvaan niin operatiivisilla kuin strategisilla tasoilla. (Laihonen et al. 2013)

3.2.1 Extract

ETL-prosessin ensimmäisessä vaiheessa dataa poimitaan (*extract*) prosessiin erilaisista lähteistä. Kuvassa 9 kuvataan esimerkkilähteinä ERP-järjestelmä (Enterprise Resource Planning) sekä WMS-järjestelmä, joista dataa kerätään prosessiin. Datan keräys -

vaiheessa muodostetaan yhteys lähdejärjestelmiin ja luetaan niiden dataa. Datan hankinta ajoitetaan lähdejärjestelmistä ja niiden mahdolliset taustaprosessit katkaistaan hetkellisesti, jotta mahdollistetaan datan luku. Lähdejärjestelmistä haettua dataa käsitellään ja järjestellään, jotta se on valmis seuraavaan vaiheeseen prosessia. (Kimball & Caserta 2011)

Ensimmäisessä vaiheessa eli tiedon keräämisessä, data haetaan lähdejärjestelmistä - pilvestä, dokumenteista, tietokannoista – liitäntöjen avulla, jotka mahdollistavat yhdistymisen moniin erilaisiin lähteisiin. Liitäntöjä voidaan kuvata eräänlaisina kääntäjinä, jotka mahdollistavat lähdejärjestelmän datan lukemisen, joka yleensä on yhteensopimatonta kohdejärjestelmän kanssa. ETL-työkaluihin yleensä sisältyy mukaan omia liittimiään kaupallisten ohjelmistojen kanssa, jotka kuuluvat ETL-työkalujen kokonaispakettiin. (Goldfedder 2020)

Kun yhteys lähdejärjestelmään on muodostettu, pyydetyt tiedot haetaan ja ne tallennetaan väliaikaisesti välivarastointialueelle. ETL-työkalut tarjoavat tämän tyylistä sisäänrakennettua väliaikaista tallennusaluetta, jota ne hallitsevat ja ylläpitävät itse. (Goldfedder 2020) Goldfedder (2020) huomauttaa erikseen välivarastointialueen tietoturvallisuudesta ja tarpeellisista suojauksista, kun käsitellään arkaluontoisia henkilötietoja, joiden suojaaminen on tarpeen väliaikaisen välivarastointiprosessin aikana.

Datankeruuvaiheessa varmistetaan datan aitoperäisyys, että tiedot täsmäävät lähteestä poimittujen ja väliaikaisesti tallennettujen tietojen välillä. Tämä on erityisen tärkeää ETL-prosessin ensimmäisessä vaiheessa, jotta ei menetetä poimittuja tietoja tai jouduta korjaamaan virheellistä dataa prosessin seuraavissa vaiheissa, jossa se on haastavampaa. (Goldfedder 2020)

3.2.2 Transform

Prosessin seuraavassa vaiheessa (*transform*) hankittua dataa muokataan, muunnellaan, tarpeen mukaan yhdistellään ja yhdenmukaistetaan, jos dataa on luettu useammasta lähdejärjestelmästä sekä tarkistetaan muokatun datan rakenne, laatu ja sen sisältävät tietuet, jotta dataa voidaan käyttää haluttuun tarkoitukseen. Dataa muunnellaan, jotta se on yhdenmukaista ja sopivassa muodossa kohdejärjestelmään tallentamista varten. Datan muunnokset tehdään asetettujen sääntöjen ja tavoitteiden rajoissa, jotka ohjaavat datan valmistelua seuraavaan vaiheeseen. (Kimball & Caserta 2011)

Datamuunnokset toteutetaan lähdejärjestelmästä haettuun dataan. ETL-työkaluilla pystytään hoitamaan datamuunnoksia, jotka toteuttavat ohjelmoituja toimintoja datan

transformaatioon. Dataa, jota ei tarvitse käsitellä datan muunnosvaiheessa, kutsutaan suoraksi datasiirroksi tai läpimenodataksi. (Goldfedder 2020)

Datamuunnoksien datan eheyden varmistuksessa tarkastellaan datan tarkkuutta, sen virheettömyyttä, täydellisyyttä, yhdenmukaisuutta sekä sen kokonaisuuden johdonmukaisuutta. Esimerkiksi varmistetaan datan tarkkuus, voidaanko jäljittää muunnettu data takaisin lähteeseen tai varmistetaan datan täydellisyys, onko kohdejärjestelmän tyhjät tai NULL-kentät tarvittaessa täytetty valideilla arvoilla tai varmistetaan datan yhdenmukaisuus, onko sovellettu samat säännöt kaikille tietyntyyppisille kentille tai samanlaisille arvoille. (Goldfedder 2020)

Datamuunnoksien vahvistus varmistaa, että datamuutokset ladataan oikein kohdejärjestelmään. Datan validointi kattaa kaikki datalle tehdyt muunnokset. Datan validointivaihe liittyy enemmän datan käsittelyjen vahvistamiseen kuin datan laadun varmistamiseen, vaikkakin datan validointi ja datan eheyden varmistus kulkevat pitkälti käsi kädessä. (Goldfedder 2020)

3.2.3 Load

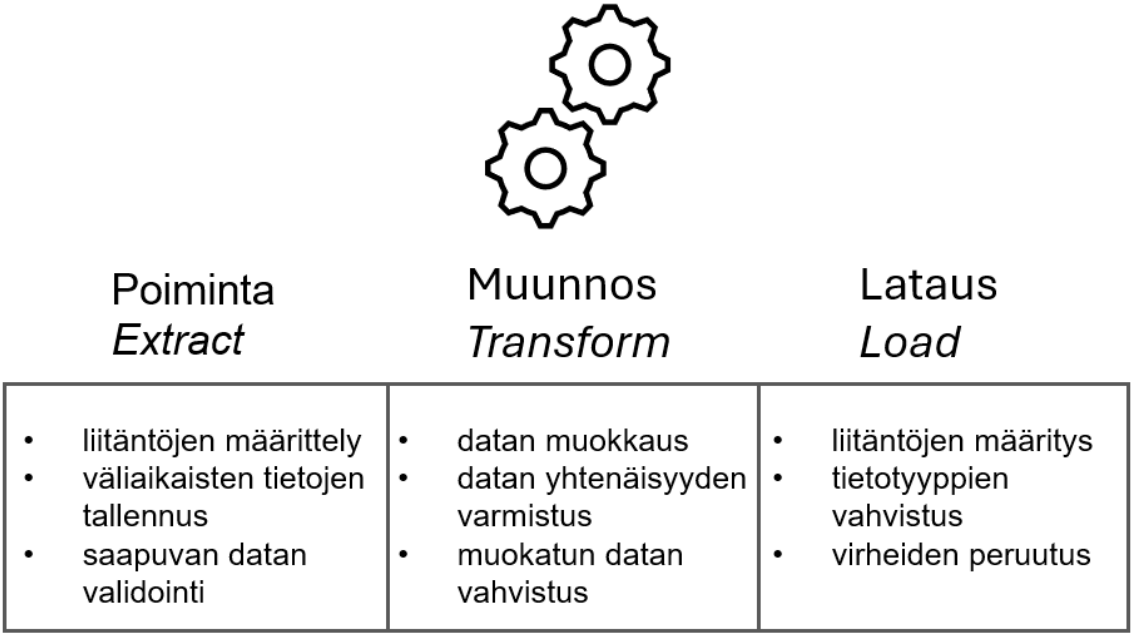
Prosessin viimeinen vaihe (*load*) sisältää haetun ja valmistellun datan tallentamisen kohdetietokantaan tai tietovarastoon, jossa se on helposti saatavana. Koko prosessin ajan huolehditaan datan eheydestä, jotta dataa voidaan hyödyntää haluttuihin käyttötarkoituksiin ja se on sopivassa tallennusmuodossa haluttuun kohteeseen. Lataus-vaiheen jälkeen data on käyttövalmista, käyttötarkoitukseen sopivassa muodossa valmiina käytettäväksi. (Kimball & Caserta 2011)

ETL-prosessin viimeisessä vaiheessa muunneltu data ladataan lähdejärjestelmästä kohdejärjestelmään. ETL-työkalut tarjoavat niiden valmiiden toimintojen hyödyntämisen tarjoamalla komponentteja yhteyksien luomiseen kohdejärjestelmään, lataustyyppien määrittämiseen sekä virhetilanteiden hallintaan. (Goldfedder 2020)

Toisin kuin ETL-prosessin poimintavaiheessa, jossa dataa haettiin lähdejärjestelmästä, latausvaiheessa yhteyksiä hyödynnetään lisäämällä tai päivittämällä tietoa kohdejärjestelmään. Latausvaiheessa on huomioitava vaiheen kesto; ladattava data voi muodostua useasta eri lähteestä, jolloin latausvaiheen aikataulutus täytyy huomioida prosessin ajanhallinnassa. (Goldfedder 2020)

ETL-prosessin latausvaiheessa lataustyypit kuvaavat ladattavien tietojen kokonaisuuksia; onko kyseessä kertalataus kaikelle haetulle historiatiedolle, inkrementaalinen tietojen lisäys vai toistuva tietojen lisäys ja päivitys jo olemassa oleviin tietoihin jatkuvana latauksena. Virheiden peruuttamiseen tai aiemman version

palautukseen (*roll back*) ETL-työkalut tarjoavat ominaisuuksia, joiden avulla mahdollistetaan virheiden peruutustoiminnot sekä niiden seuraaminen kuin myös prosessiajojen kirjaustapahtumien seuraaminen. (Goldfedder 2020) Alla olevassa kuvassa (kuva 10) esitetään ETL-prosessin vaiheet ja niiden toiminnot, jotka sisältyvät myös sisäänrakennettuina ETL-työkalujen funktiotoimintoihin.



Kuva 10. ETL-työkaluihin sisältyvät funktiotoiminnot ETL-prosessin vaiheista (mukaillen Goldfedder 2020)

3.3 ETL-prosessien kehitys

ETL-prosessin kehitystä voidaan lähestyä kokonaisvaltaisen ohjelmistokehitysprosessin kautta. Yksi ETL-prosessin tunnistettu kehitysmalli seuraa ohjelmistokehityksen vesiputousmallia (kuva 4), jossa kuvataan ohjelmiston kehitysvaiheet alkuvaatimuksista lopullisen tuotteen ylläpitovaiheeseen. Tämänkaltainen prosessimalli on tunnistettu myös ETL-prosessin kehityksessä, mikä tarjoaa ETL-prosessin kehitystyölle yhden viitekehyksen, jonka kautta kehitystyötä voidaan lähestyä. (Bassil 2012; Gurung et al. 2020)

Yleisesti ETL-prosessin kehityksen vaiheet voidaan kuvata seuraavien tunnistettujen kehitysvaiheiden kautta. Analyysivaihe: vaatimustenkeruu, jotta ymmärretään tehtäväkokonaisuus. Suunnitteluvaihe: suunnitellaan ratkaisu määriteltyjen ehtojen pohjalta. Toteutusvaihe: suunnitellaan ja toteutetaan prosessin ohjelmointi. Testausvaihe: testataan prosessin toiminta vaatimuksia vasten. Käyttöönotto ja ylläpito: otetaan prosessi käyttöön ja ylläpidetään sen toimintaa. (Mishra & Dubey 2013)

Analyysivaiheessa kuvataan prosessin käyttäytymisestä, jossa määritellään prosessin toiminnalliset, että ei-toiminnalliset vaatimukset. Yleensä toiminnalliset vaatimukset määritellään käyttötapausten avulla, jotka kuvaavat käyttäjien vuorovaikutusta prosessin kanssa. Ne sisältävät vaatimuksia prosessista kuten sen tarkoitus, laajuus, näkökulma, toiminnot, prosessin sekä käyttäjän ominaisuudet, toiminnallisuuksien määrittelyt, käyttöliittymävaatimukset ja tietokantavaatimukset. Ei-toiminnalliset vaatimukset viittaavat erilaisiin kriteereihin, rajoituksiin ja vaatimuksiin, jotka on asetettu prosessin suunnitteluun ja toimintaan tiettyjen käyttäytymisten sijaan. Ei-toiminnallisiin vaatimuksiin sisältyy prosessin ominaisuuksia kuten luotettavuus, skaalautuvuus, testattavuus, saatavuus, ylläpidettävyys, suorituskyky ja laatuvaatimukset. (Bassil 2012)

Suunnitteluvaiheessa suunnitellaan ja ratkaistaan toteutukseen liittyviä ongelmia ja määritellään ratkaisun suunnitelma, joka kattaa kokonaisvaltaisesti prosessiin liittyvät entiteetit. (Bassil 2012)

Toteutusvaihe tarkoittaa liiketoiminta- ja suunnittelumääritysten toteuttamista ETL-prosessiksi ohjelmoinnin avulla. Toteutusvaiheessa kirjoitetaan varsinaista koodia, joka käännetään toimivaksi kokonaisuudeksi. (Bassil 2012)

Testausvaiheessa varmistetaan ja vahvistetaan, että prosessi täyttää alkuperäiset vaatimukset ja määrittelyt sekä suorittaa sille osoitetut tehtävänsä. Lisäksi testausvaiheessa suoritetaan virheiden etsintöjä, jossa toimintavirheet ja prosessin ajovirheet havaitaan ja korjataan sekä hienosäädetään prosessia niiden pohjalta. (Bassil 2012)

Käyttöönotto- ja ylläpitovaiheessa prosessia tarpeen mukaan muokataan prosessiajojen hienosäätämiseksi, virheiden korjaamiseksi sekä suorituskyvyn ja laadun parantamiseksi. Käyttöönotto- ja ylläpitovaiheessa voidaan lisäksi huomioida muita toimintoja, kuten eri ympäristöjen käyttöönottoja, sovittaa prosessia uusiin käyttäjävaatimuksiin tai parantaa prosessin luotettavaa toimintaa. (Bassil 2012)

El Akkaoui et al. (2011) esittävät omassa raportissaan toisen ETL-prosessin kehitysmallin, jossa lähestytään ETL-prosessin kehittämistä mallivetoisen kehitysmallin avulla. Mallivetoisen kehitysmallin pääasiallisena tavoitteena on automatisoida ETL-prosessin kehitystyöt, jotta kehitystuottavuutta voidaan parantaa, kehitysaikaa ja -kustannuksia vähentää sekä sen avulla pyritään saavuttaa parempilaatuinen ratkaisukokonaisuus ja pyritään mahdollistaa prosessin joustava käyttö. Mallivetoisen kehitysmallin tarjoamien viitekehyksien ympärille pyritään automatisoimaan prosessin kehitysprosessit, jotka lopulta johtavat konkreettiseen toteutukseen eli prosessin suorittavaan lähdekoodiin. (Turajlic et al. 2014)

Mallivetoista lähestymistapaa hyödynnetään myös ohjelmistokehityksessä, jossa laajoja malleja luodaan ennen lähdekoodin kirjoittamista. Mallivetoinen kehitysmalli pyrkii kattamaan koko ETL-kehitysprosessin, mukaan lukien ETL-toimittajakohtaisten koodien automaattisen generoinnin useille alustoille, minkä vuoksi malli soveltuu niin ETL-toimittajakohtaisiin ETL-työkaluilla toteutettuihin ETL-prosesseihin kuin käsin ohjelmoituihin ETL-prosesseihin. (El Akkaoui et al. 2011)

3.4 ETL-työkalut

ETL-työkalut perustuvat ETL-prosessiin, sen kehittämiseen ja ylläpitoon. ETL-työkalut auttavat organisaatioita hyödyntämään dataa ja mahdollistavat sen käytön. Ne auttavat pääsemään dataan käsiksi, muokkaamaan sitä käyttökelpoiseksi tiedoksi ja siirtämään tiedon haluttuihin tietojärjestelmiin. (Qaiser et al. 2023) ETL-työkaluja käytetään yksinkertaistamaan datanhallintaa ja vähentämään tarvittavaa työpanosta ETL-prosessin kehittämisessä ja ylläpidossa. ETL-työkalut ovat suunniteltu säästämään aikaa ja rahaa poistamalla tarve manuaaliselta käsin ohjelmoinnilta, kun uutta tietovarastoa ja ETL-prosessia kehitetään. (Mali & Bojewar 2015) ETL-työkalut tarjoavat tukea ja valmiita funktiotoimintoja ETL-prosessin vaiheisiin ja niiden tyypillisiin käyttötapauksiin. ETL-työkalut on varustettu visuaalisella käyttöliittymällä, joka auttaa havainnollistamaan tietovirtauksien suunnittelua ja datamuunnoksia sekä helpottaa virhetilanteiden selvitystyötä. (Hovi et al. 2009, s. 54)

ETL-työkalut on suunniteltu vähentämään ETL-prosessissa tarvittavaa manuaalista ohjelmointityötä tiedon siirtämisessä lähdejärjestelmistä tietovarastoon. ETL-työkalut tarjoavat käyttäjälle graafisen käyttöliittymän sekä mahdollistavat datan hakemisen ja yhdistelyn erilaisista tietolähteistä. ETL-työkaluissa on sisäänrakennetut toiminnallisuusominaisuudet tiedon hakemiseen, sen yhdistämiseen, tietotyyppimuunnoksiin ja tiedon muokkaamiseen haluttuun muotoon sekä sen siirtämiseen kohteeseen (kuva 10). (Hovi et al. 2009, s. 61) ETL-työkalujen avulla pystytään suorittamaan ETL-prosessin tunnusomaiset datan käsittelyyn, muokkaamiseen ja siirtämiseen liittyvät toiminnot sisäänrakennetuilla funktiotoiminnoilla.

ETL-työkalut mahdollistavat ETL-prosessin ajon joko ajastetusti, reaaliaikaisesti tapahtumien mukaan tai jatkuva lataus -tyyppisenä tiedon siirtämisenä. ETL-työkalut tarjoavat prosessien ylläpitoon datan laadun profilointia ja siihen liittyvää raportointia, kattavaa virheiden monitorointia, raportointia ja niiden hallintaa sekä tukea muutoksien hallinnalle ja versiohallinnalle. Lisäksi ETL-työkaluissa on toimintoja, jotka tarjoavat pääsyn prosessiajojen kirjauksiin, tarjoaa niiden lokituksen ja tilastoinnin sekä tukee prosessien testaamista ja kehittämistä. (Hovi et al. 2009, s. 61)

Goldfedder (2020) esittelee omassa teoksessaan standardinmukaisia teknisiä komponentteja, jotka hän on tunnistanut ETL-työkaluille. Goldfedder (2020) listaa yleisiä komponentteja, jotka muodostavat ETL-työkalun ominaisuudet ja ohjaavat niiden kehitystä ja käyttöä. Lähteestä-kohteeseen -malli, toimintalogiikka, liiketoimintasäännöt, virheiden hallinta, versiohallinta, prosessiajojen aikataulutus ja tapahtumien kirjaus ovat huomioituja käsitteitä, jotka ETL-työkaluihin ja niiden käyttöön sisältyy.

ETL-työkalun lähteestä-kohteeseen -malli kuvaa ohjeita, jotka määrittävät, miten data muunnetaan lähdejärjestelmästä oikeaan rakenteeseen, muotoon ja sisältöön, jota kohdejärjestelmä vaatii. Toimintalogiikka kuvaa ETL-työkalun lähdekoodia, joka määrittää työkalun toiminnot; datan poiminnan, muuntamisen ja lataamisen lähteestä kohdejärjestelmään. Dataintegraatioprosessin toiminnallisuuksia hoitavan lähdekoodin konfigurointi perustuu kokonaan valittuun ETL-työkaluun ja sen ominaispiirteisiin. (Goldfedder 2020)

Organisaation liiketoimintasäännöt ja -kriteerit ohjaavat datan muuntamista lähteestä kohdejärjestelmään ja luo ehdot ja vaatimukset datan käsittelyprosesseille. Organisaation liiketoimintavaatimukset ja -päätökset ohjaavat liiketoimintasääntöjä; liiketoimintavaatimukset ovat asioita, joiden on täytyttävä, jotta liiketoimintasäännöt voidaan ottaa käyttöön. (Goldfedder 2020)

ETL-työkalun virheiden ja poikkeuksien hallinta kuvaa virheilmoitusten ja vastaavien poikkeuksien määrittämistä, jotka tulisi suunnitella sisäänrakennettujen ohjelmistokonfiguraatioiden ja lähdekoodin avulla. ETL-työkalun versiohallinta kuvaa muutoksien tallentamista, joihin voidaan myöhemmin tarpeen mukaan palata ja palauttaa tiettyjä versioita takaisin käyttöön. (Goldfedder 2020)

ETL-työkalun avulla voidaan aikatauluttaa ETL-prosessien ajoja. Esimerkiksi prosessi voidaan asettaa automaattiseen suoritukseen toistettavasti eli se suoritetaan tiettynä päivänä tiettynä aikana määritellyin väliajoin. ETL-työkalun tarjoama lokituksien kirjaus kuvaa ETL-prosessien suorituskertojen ylös kirjaamista. ETL-työkalut tarjoavat lokitustoiminnon, joka kirjaa yksittäiset sisääntulotietueet ja lähtevät tietueet sekä suoritettujen työtehtävien ja ajosuorituksien keston, päivämäärän ja nimen sekä kirjaa mahdolliset virhetilanteet. (Goldfedder 2020)

Zode (2007) tarkastelee ETL-työkalujen ominaisuuksien kehittymistä omassa raportissaan ja listaa niiden vahvuuksia ja heikkouksia. Zode (2007) näkee ETL-työkalujen vahvuuksina funktiotoiminnot, jotka on suunniteltu vastaamaan ETL-prosessin vaatimuksia; ETL-työkalujen avulla monimutkaistenkin prosessiajojen suoritusnopeus pystytään optimoimaan, mikä parantaa prosessien suorituskykyä. ETL-

työkalut tarjoavat graafisen käyttöliittymän, joka helpottaa prosessien visualisointia, niiden hallintaa ja ylläpitoa sekä tarjoaa visuaalisen esityksen tietojen siirtymisistä ja niiden muunnoksista. ETL-työkalun toiminnot ovat integroituja ja automatisoituja, mikä mahdollistaa niiden aikatauluttamisen, helpon muokkaamisen ja muutoksien hallinnan. ETL-työkalut mahdollistavat datan ja prosessiajojen rinnakkaisen käsittelyn sekä ne tukevat erilaisten tietolähteiden lukemista ja tietovarastojen erityylisiä konsepteja. Lisäksi ETL-työkalut tarjoavat versionhallinnan tuen ja auttavat metatietojen ylläpidossa ja niiden dokumentoinnissa. (Zode 2007)

Toisaalta Zode (2007) mainitsee ETL-työkalujen heikkouksina niiden yksilölliset toimintatavat ja toimintalogiikan, jotka edellyttävät käyttäjältä ETL-prosessiosaamista ja ETL-työkalujen tuntemista. Lisäksi joidenkin ETL-työkalujen rajoitettu reaaliaikainen tuki sekä mahdollisten pullonkaulojen muodostuminen suurien datamassojen käsittelyssä voi aiheuttaa ongelmia ETL-prosesseissa. Zode (2007) toteaa myös, että ETL-työkalujen ja loppukäyttäjän työkalujen välinen metatietotasojen integrointi saattaa aiheuttaa rajoituksia ETL-työkalujen toiminnassa.

3.5 Tietovarastot ETL-prosessissa

Kimball et al. (2008) mukaan tietovarastointi kuvaa laajaa prosessia, jossa organisaatio hankkii dataa eri lähteistä ja muuntaa sen hyödylliseksi tiedoksi, jota voidaan käyttää hyödyksi liiketoiminnan päätöksenteossa. Tietovarastoinnin prosessissa käsitellään eri tietolähteistä koottua tietoa ja pyritään luomaan siitä koko organisaatiota kuvaava tai tiettyä osa-aluetta havainnollistava tilannekuva. Tietovarastointi mahdollistaa suurien tietomäärien tallennuksen kooten tiedot yhteen paikkaan saatavaksi. (Kimball et al. 2008) Tietovarastointi käsittää teknologiat, konseptit, menetelmät ja suunnittelumallit, jotka mahdollistavat tiedon hyödyntämisen osana tietoperustaista päätöksentekoa. (Inmon 2005) Organisaation tietoarkkitehtuuri auttaa hahmottamaan tietovarastoinnin tarpeet ja tehtäväkokonaisuudet. Tietoarkkitehtuurin avulla kuvataan sekä lähdejärjestelmien että tietovarastojen tiedot, mikä edistää tietojen siirrettävyyttä ja niiden yhteensopivuutta. (Hovi et al. 2009, s. 66)

ETL-prosessi syöttää tarvittavat tiedot tietovarastoihin ja toimii tiedonsiirtoprosessin kulmakivenä. Datalähteet, joista dataa siirretään ja jotka päätyvät tietovarastoon, ovat tyypillisesti heterogeenisia, hajautettuja ja niiden data vaatii muokkausta. ETL-prosessien tulisi selviytyä tällaisista datan ominaisuuksista ja varmistaa datan laatu, jota tietovarasto vaatii, jotta saadaan yhdenmukaistettua datalähteiden data yhteen paikkaan oikeassa muodossa ladattuna. (El Akkaoui et al. 2011)

Tietovarastoinnin tavoitteena on uudelleen järjestää, liittää yhteen ja yhdistää sekä ylläpitää valtavia tietomääriä useista järjestelmistä ja muodostaa niistä kollektiivinen tietopohja liiketoiminnalle. ETL-prosessi on keskeisessä asemassa tietovarastoinnissa, jossa dataa integroidaan useista lähteistä yhteen järjestelmään. Tietovarastot toimivat yhtenäisenä tallennuspaikkana historia- ja reaaliaikaiselle datalle, jotka on siirretty monista eri lähteistä. Tietovarastot koostuvat pääasiassa historiallisesta datasta, joka johdetaan nykyisestä, reaaliaikaisesta datasta ja liiketoimintatiedoista, jotka on ladattu useista eri lähteistä. (Khan et al. 2024)

Tietovarastoinnissa hyödynnetään tietovarastoja ja niiden rakennetta kuvaavaa infrastruktuuria. Ariyachandra & Watson (2010) kuvaavat tietovarastoja datan säilytyspaikkoina, joita organisaatiot hyödyntävät tietoperustaisessa päätöksenteossa ja jotka mahdollistavat tiedon hyödyntämisen osana organisaation toimintaa. Tietoperustaisessa päätöksenteossa organisaation johdon liiketoiminnalliset päätökset ja ratkaisut pohjautuvat sen hetkiseen merkitykselliseen tietoon, mitä kulloinkin tilanteesta on saatavilla. (Laihonen et al. 2013) Ponniah (2010) tunnistaa tietovarastojen kolme tärkeintä tehtäväaluetta, jotka tietovarastoilla täytyy ratkaista; datan hankinta, datan tallennus ja säilytys sekä tiedon jakaminen.

Rainardi (2008) kuvaa tietovarastoa järjestelmänä, joka hakee ja yhdistää dataa lähdejärjestelmistä ja varastoi sitä vuosiksi historiadatana. Tietovarasto tarjoaa analyyttisiin prosesseihin, raportointiin ja tietoon perustuviin liiketoiminnallisiin päätöksiin alustan, joka edesauttaa organisaation päätöksentekoa ja mahdollistaa organisaation liiketoimintatiedon hallinnan. Rainardi (2008) kuvaakin, kuinka tietovarastointi nähdään yhä enemmän liiketoiminnan mahdollistavana toimintona, jota hyödynnetään datan louhimisessa, raportoinnin tukena sekä dataintegraatioissa. Myös Ponniah (2010) näkee organisaation tietovarastoinnin strategisen päätöksenteon mahdollistajana. Tietovarastot mahdollistavat tietojen hyödyntämisen liiketoiminnan tukena, ne tarjoavat tietoa ymmärrettävässä ja helposti luettavassa muodossa päätöksentekoon sekä auttavat liiketoiminnan kehittämisessä. Tarjolla olevalla tiedolla ohjataan organisaation operatiivisia, taktisia ja strategisia päätöksiä. (Ponniah 2010)

Rainardin (2008) näkemyksen mukaan yksinkertaisin muoto tietovarastoinnista voidaan kuvata lähdejärjestelmän ja tietovaraston välisellä linkityksellä, jossa dataa siirretään ETL-prosessilla (kuva 11). ETL-prosessin avulla mahdollistetaan tietojen välitys lähdejärjestelmästä tietovarastoon, jossa tiedot ovat helposti saatavilla organisaation tarvittaviin toimintoihin. (Rainardi 2008)

Kuva 11. Tietovarastointi-prosessi (mukaillen Rainardi 2008; Ponniah 2010)

Laihonen et al. (2013) näkemyksen mukaan tietokannat ovat datan säilytyspaikkoja, joiden takana on usein niihin yhdistetty tietovarasto. Tietovarastot käsittelevät tietokantojen dataa ja tarjoavat organisoidumman näkymän samaan tietomassaan. (Laihonen et al. 2013) Tietovarasto erottuu usein kokoelmasta linkitettyjä, tiettyjä aihekohtaisia, muuttuvia ja aikavaihtelevia tietokantoja. (Khan et al. 2024) Tietovarasto on strategisten tietojen arkisto, jossa yhdistyy monen eri lähteen tieto, jotka on kerätty pitkän ajan kuluessa. Perinteiset tietovarastotoiminnot koostuvat pääasiassa datan poimimisesta useista lähteistä (*extract*), data muuntamisesta tiedoksi tietovarastoon yhteensopivaan muotoon (*transform*) ja lopulta tiedon lataamisesta (*load*) tietovarastoon lisäanalyysejä varten. (Mukherjee & Kar 2017)

Näiden näkemyksien perusteella voidaan todeta molempien termien, tietokannan ja tietovaraston, nimiensä mukaisesti sisältävän tietoa, jota hyödynnetään organisaation toiminnan kehittämisessä. Tietokanta voidaan nähdä yleisenä esityksenä tietojen kuvaukselle ja hallinnalle, kun taas tietovarasto yhdistää ETL-prosessin avulla tietoja eri lähteistä tarjoten tähän prosessoituun organisaation tietoon järjestelmällisen, helppokäyttöisen näkymän käyttäjilleen. ETL-prosessit muodostavat tietovarastoinnin selkärangan, sillä ne toimittavat tietovarastolle tarvittavan yhtenäistetyn ja yhdistetyn datan erilaisista hajautetuista ja hajanaisista tietolähteistä sovittaen sen tietovarastoon yhtenäisenä tietona. (El Akkaoui et al. 2011)

Tietovarasto toimii integroitujen, aihekohtaisten tietokantojen kokoelmana, mikä on tarkoitettu tukemaan organisaation päätöksentekoprosessia tarjoamalla yksinoikeuden useisiin lähteisiin, jotka edesauttavat ja mahdollistavat päätöksentekoa. (Khan et al. 2024)

Yhdistämällä eri tietolähteitä tietovarasto tarjoaa kattavan ja tarkan näkymän organisaation operatiiviseen dataan, joka voidaan syntetisoida joukoksi strategisia mittareita, ja jotka voidaan analysoida tarpeiden mukaan. Tämän vuoksi tietovarasto

mahdollistaa nopeiden ja tehokkaiden liiketoimintaraporttien tuottamisen operatiivisista tietolähteistä. (El Akkaoui et al. 2011)

Tietovarastojen tavoitteena on tukea päätöksentekijöitä tekemään nopeampia ja parempia liiketoimintapäätöksiä liiketoimintaympäristöissä tarjoamalla heille tarvittavat työkalut liiketoimintadatan muuntamiseen strategiseksi tiedoksi. Jotta organisaatio voi olla kilpailukykyinen vaativissa ja jatkuvasti muuttuvissa ympäristöissä, täytyy tietovaraston tarjota alusta, johon liiketoimintadata voidaan helposti tallentaa ja jossa sitä voidaan käsitellä, analysoida ja lopulta muuntaa se strategiseksi tiedoksi. (Turajlic et al. 2014)

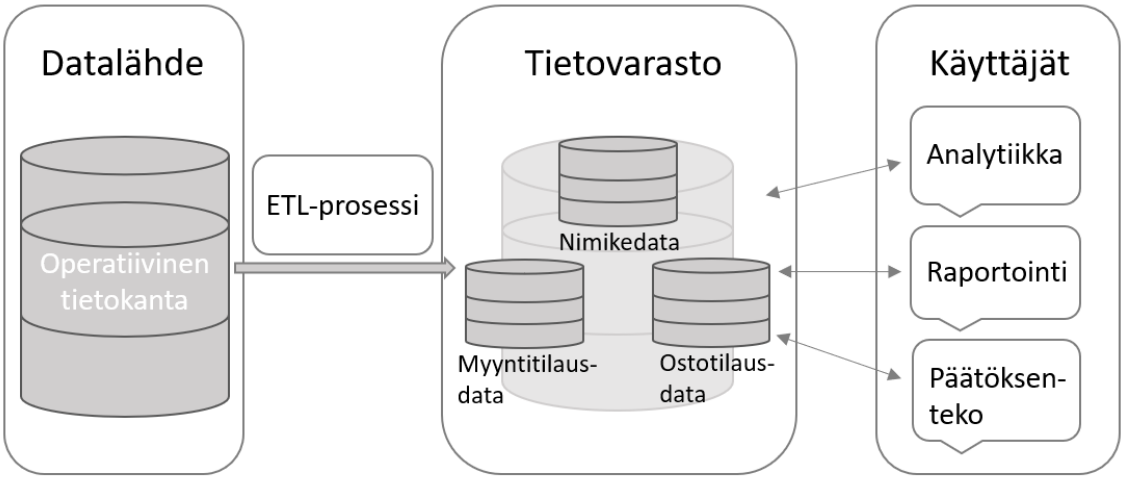
Tietovarastoarkkitehtuureilla voidaan kuvata vaihtoehtoisia suunnitteluperiaatteita ja ratkaisumenetelmiä, joilla ETL-prosessin kolmea tunnistettua vaihetta voidaan hyödyntää datasiirrossa tietovarastoihin. Tietovarastojen rakennetta ja toiminnallisuutta voidaan suunnitella menetelmien kautta, jotka kuvaavat tietovarastojen käyttöä ja ylläpitoa. Seuraavassa luvussa 3.2.7 esitellään tietovarastoarkkitehtuuri, jossa kuvataan arkkitehtuurin keskeisiä periaatteita ja toimintamenetelmiä tietovarastojen toteutukseen ja hallintaan liittyen.

3.6 Tietovarastoarkkitehtuuri

Jotta organisaatio voi analyyttisesti prosessoida tietoa tai tehdä tietoon perustuvia liiketoiminnallisia päätöksiä, täytyy organisaatiolla olla tietovarasto, jossa organisaation data on prosessoituna, tallennettuna ja helposti saatavilla. (Blazic et al. 2017) Tietovarastoarkkitehtuuri havainnollistaa tietovarastojen hyödyntämistä organisaation toiminnassa, jossa kuvataan tietojen säilytyspaikkoja osana tietovarastokokonaisuuksia. Tietovarastoarkkitehtuurissa kuvataan tietojen virtausta lähdejärjestelmistä käyttäjille sekä havainnollistetaan prosessissa mukana olevia fyysisiä järjestelmiä. Tietovarastoarkkitehtuuri voidaan nähdä osana organisaation tietoarkkitehtuuria, jossa käsitellään organisaation tietoresursseja kokonaisvaltaisesti. (Rainardi 2008)

Tietovarastoarkkitehtuurin kuvaamiseen on tarjolla useita erilaisia esityksiä (Blazic et al. 2017). Tässä tutkimuksessa kuvataan yhden kerroksen tietovarastoarkkitehtuuria, johon myös WMS-tietojärjestelmän tietovarastoinnin toimintaperiaatteet pohjautuvat. Yhden kerroksen tietovarastoarkkitehtuuria on kuvattu alla olevassa esimerkkikuvassa (kuva 12), johon on yhdistetty WMS-tietojärjestelmän toiminnallisuuksia. Kuvasta 12 huomataan, kuinka kuvan tietokanta-tietovarasto-esitys jäljittelee Laihonen et al. (2013) näkemyksiä tietokannan ja tietovaraston kuvauksista sekä kuinka ETL-prosessi siirtää operatiivisesta tietokannasta luetut tiedot suoraan tietovarastoon, jossa tiedot ovat

saatavilla käyttäjille tarvittaviin toimenpiteisiin, mikä tunnistettiin myös Rainardin (2008) ja Ponniahin (2010) kuvaamista tietovarastoinnin kuvauksista.



Kuva 12. Yhden kerroksen tietovarastoarkkitehtuuri (mukaillen Blazic et al. 2017)

Tietovarastoarkkitehtuurin avulla suunnitellaan ja järjestellään datan virtaukset datalähteistä tietovarastoon, jossa tieto on tarjolla käyttäjille, esimerkiksi päätöksenteon tai raportoinnin tueksi. Tietovarastoarkkitehtuuri tarjoaa kokonaisvaltaisen kuvauksen organisaation tietovarastoista, niiden suunnittelusta, kehityksestä sekä käyttöönotosta ja ylläpidosta. Tietovarastoarkkitehtuuri määrittää yleiset standardit, ulottuvuudet, suunnittelun ja tarvittavan tuen, jota tietovarastojen toiminta edellyttää toimiakseen organisaation toimintaympäristössä. (Ponniah 2010)

Ponniah (2010) esittelee tietovarastoarkkitehtuurin viitekehystä, joka tunnistaa neljä päärakenneosaa tietovarastointiin liittyen: datalähde, datan hankinta (ETL-prosessi), datan varastointi ja tiedon jakaminen. Myös Wojciechowski (2011) tunnistaa samat rakenneosat tietovarastoarkkitehtuuriin liittyen omassa tutkimuksessaan. Nämä rakenneosat ovat visuaalisesti havaittavissa myös kuvasta 12.

Dataa hankitaan datalähteistä, jotka ETL-prosessin avulla siirretään tietovarastoon. ETL-prosessi suorittaa datalle tarpeen mukaiset toimenpiteet, jotta tietovarastoon saadaan yhtenäistä ja eheetä dataa. Datan varastointi -vaihe tallettaa datan tietovarastoon ja varastoi sen tarkoituksenmukaisessa muodossa, jotta data saadaan muunnettua tiedoksi. Tietoa käytetään hyödyksi erilaisten toimintojen kautta, jotka mahdollistavat tiedon hyödyntämisen osana erikoistietokantoja, raportointia, graafisia käyttöliittymiä tai erilaisia suorituskykymittaus esityksiä. (Ponniah 2010)

Koskimies & Mikkonen (2005) näkevät tietovarastoarkkitehtuurin toimintaperiaatteen joukkona järjestelmiä, jotka käyttävät ja ylläpitävät yhteisesti tietovaraston tilaa ja käyttävät sen tarjoamia tietoja yhteisinä resursseina. Jaettua tietovarastoa käsittelevät järjestelmien komponentit, jotka tarjoavat tietoa palveluidensa kautta ja kommunikoivat keskenään tietovaraston kautta. (Koskimies & Mikkonen 2005)

Tietovarastoarkkitehtuuri mahdollistaa tietovarastojen suunnittelun organisaation tarpeisiin, missä yhdistyy konkreettisesti organisaation hallussa olevien tietoresurssien suunnitelmallinen hyödyntäminen sekä organisaation hyödyntämät toiminnot, menetelmät, palvelut ja teknologiset resurssit, jotka mahdollistavat tietovarastojen hyödyntämisen. (Ponniah 2010)

3.7 Tietojärjestelmät

Tietojärjestelmät ovat yksi esimerkki lähdejärjestelmästä, josta siirretään dataa ETL-prosessin avulla tietovarastoon tai toiseen tietojärjestelmään. Tietojärjestelmät ja ETL-prosessi ovat olennaisessa osassa organisaation kokonaisvaltaista tietovarastointia ja niiden välisiä vuorovaikutustoimintoja voidaan kuvata esimerkiksi tietovarastoarkkitehtuurin tarjoamien viitekehyksien avulla.

Tietojärjestelmät muodostuvat toisiinsa yhteydessä, vuorovaikutuksissa olevista komponenteista, jotka keräävät, käsittelevät, tallentavat ja jakavat organisaation ja sen ympäristön tietoa. Tietojärjestelmät mahdollistavat tiedon hyödyntämisen organisaation päätöksenteon tukena, sisäisissä toimintaprosesseissa ja toiminnan kehittämisessä. (Laudon & Laudon 2007)

Tietojärjestelmien toimintaa voidaan yleisesti kuvata kolmen pääaktiviteetin avulla. Tietojärjestelmä kerää dataa organisaation sisäisestä toiminnasta ja sen ulkoisesta ympäristöstä syötteeksi (*input*), jonka jälkeen syötedataa prosessoidaan, lajitellaan, ryhmitellään, järjestellään ja sille suoritetaan erilaisia toimenpiteitä, jotta se muuntuu helposti luettavaan muotoon (*processing*). Lopuksi tietojärjestelmä tarjoaa käsitellyn tiedon tuloksenaan sitä tarvitseville henkilöille tarvittaviin käyttötarkoituksiin (*output*). (Laudon & Laudon 2007) Tietojärjestelmät mahdollistavat tiedon hyödyntämisen organisaation toiminnassa muokaten datasta organisoitua tietoa. Tietojärjestelmät auttavat organisaatiota täyttämään tunnistettuja tietotarpeita keräämällä tietoa valituista toimintaprosesseista kohdistaen sen tunnistettuihin ongelmakohtiin. (Laihonen et al. 2013)

Tietojärjestelmiä voidaan jakaa toiminnallisten ominaisuuksien ja käyttötarkoitusten mukaan. (Laudon & Laudon 2007) Esimerkiksi varaston toimintaan liittyviä

käyttötarkoituskohteita voidaan parantaa Leanwaren WMS-tietojärjestelmän toiminnallisten ominaisuuksien avulla. Nämä toiminnot voivat liittyä esimerkiksi varaston toiminnan ohjaukseen, varastotasojen optimointiin, reaaliaikaisen tilannekuvan tarjoamiseen varaston toiminnasta tai ajantasaisen päivittäisjohtamisen mahdollistamiseen. Tietojärjestelmien toiminnalliset ominaisuudet voidaan kuvata liiketoimintaprosesseina, jotka viittaavat järjestelmään liittyviin toiminnallisiin tehtäviin, johon järjestelmää hyödynnetään. Nämä tietojärjestelmien liiketoimintaprosessit mahdollistavat ja tukevat organisaation kokonaisvaltaista liiketoimintaa kattaen oman toimintakenttänsä mukaisen osan organisaation strategisesta toiminnasta. (Laudon & Laudon 2007)

Rood (1994) toteaa, kuinka tietojärjestelmien kehitystä ohjaa organisaation kokonaisarkkitehtuuri tarjoamalla ymmärrystä organisaation hallussa olevasta tiedosta sekä mahdollistamalla organisaatiotasoisen läpileikkauksen tietojen hallinnalle. Kokonaisarkkitehtuuri toimii organisaation integroitavana tahona tietojen hallintaprosesseissa edesauttaen tietojen yhdistämistä tietojärjestelmien välillä ja auttaa tunnistamaan tietojärjestelmien kehittämistarpeet organisaation strategiseen toimintaan peilaten sekä ohjaa organisaation tietojärjestelmien kehittämissuuntaa tarjoamalla yhtenäisen terminologian ja käsitteistön tietoihin ja tietojärjestelmien liittyen. (Rood 1994)

Modulaarisuus jakaa tuotteen tai palvelun, tässä tapauksessa tietojärjestelmän, itsenäisiin komponentteihin (ohjelmistoyksiköiksi) tai osakokonaisuuksiin, jotka suorittavat omia tehtäviänsä ja tarjoavat palveluitaan rajapintojen kautta. Modulaarisuus mahdollistaa tietojärjestelmien räätälöinnin asiakkaiden yksilöllisiin toiveisiin ja uniikkeihin tarpeisiin. (Haikala & Märijärvi 2006) Tietojärjestelmien asiakaskohtainen räätälöinti tukee toimintaperiaatetta, jossa tietojärjestelmiä hyödynnetään huomioitujen haasteiden torjumisessa asiakkaan omassa, yksilöllisessä toimintaympäristössä, minkä avulla organisaatio pystyy käyttämään niiden tarjoamaa tietoa hyväksi omissa liiketoimintaprosesseissaan omalla toimintatavallaan. (Haikala & Märijärvi 2006)

Myös Laihonen et al. (2013) tiedostaa tietojärjestelmien modulaarisuuden. Tietojärjestelmien muuttaminen vastaamaan asiakkaan toiveita edellyttää järjestelmien räätälöintiä asiakastarpeisiin. Tietojärjestelmien modulaarisuus, moduulit asiakastarpeiden mukaan, ovat yksi osatekijä, mikä mahdollistaa tietojärjestelmien tietoon perustuvan hyödyntämisen osana organisaation toimintaa ja tietoon perustuvaa päätöksentekoa. (Laihonen et al. 2013)

Tietojärjestelmät tukevat organisaation tietoperustaista päätöksentekoa. Tietojärjestelmät ovat joukko toisiinsa liittyviä komponentteja, jotka keräävät,

prosessoivat, varastoivat ja jakavat organisaation ja siihen liittyvän ympäristön tietoa. Tietojärjestelmät mahdollistavat tiedon hyödyntämisen päätöksenteossa ja organisaation toiminnan kehittämisessä. (Laudon & Laudon 2007)

Tietojärjestelmät mahdollistavat tiedon hyödyntämisen organisaation toiminnassa ja niiden avulla organisaatio pyrkii täyttämään tunnistettuja tietotarpeita. Tietojärjestelmät auttavat tiedon keräämisessä valituista toimintaprosesseista tunnistettuihin tarpeisiin. (Laihonen et al. 2013)

3.8 Ohjelmointirajapinnat - API

Ohjelmointirajapinnat (engl. *Application Programming Interface, API*) tarjoavat katsauksen ohjelmiston tarjoamiin palveluihin ja tietoihin ennalta määriteltyjen resurssien kautta, esimerkiksi erilaisten toimintojen, tietokantaobjektien tai merkkijonojen kautta, jolloin muut ohjelmistot voivat käyttää kyseisen ohjelmiston tietoja ja palveluita. Ohjelmointirajapinnat ovat keskeisiä monille nykyaikaisille ohjelmistoarkkitehtuureille, koska ne tarjoavat korkean tason abstraktioita, jotka helpottavat ohjelmointitehtäviä, tukevat hajautettujen ja modulaaristen ohjelmistosovellusten suunnittelua sekä koodin uudelleenkäyttöä. (Meng et al. 2018) Ohjelmointirajapinta voidaan kuvata kanavana, jonka kautta sovellukset vaihtavat tietoja, toimintoja sekä muita resursseja keskenään. Ohjelmointirajapinta mahdollistaa tietojen, toimintojen ja muiden resurssien välittämisen eri ohjelmistojen ja sovellusten välillä niiden ymmärtämässä muodossa. (Moilanen et al. 2018) API on kommunikointitapa, jolla kaksi tietokonesovellusta voivat keskustella toistensa kanssa verkon kautta käyttämällä yhteistä kieltä, jota ne molemmat ymmärtävät. (Jacobsen et al. 2011)

Butterfield et al. (2016) kuvaavat ohjelmointirajapintoja määriteltynä joukkona toimintoja ja menettelyjä, jotka mahdollistavat ohjelman pääsyn sovelluksen sisäisiin toimintoihin. Tällaisten toimintojen käyttö mahdollistaa käyttäjien mukauttamisen sovellusta omiin tarpeisiinsa ja sen integroimisen räätälöityyn kehitysympäristöön. He kuvaavat ohjelmointirajapintaa sovelluksen sisäisenä tietokanavana, mikä määrittelee käytettävät ohjelmiston sisäiset rakenteet ja toiminnallisuudet, jotka ovat osa sen binääristä rajapintaa. Sisäinen rajapinta määrittelee esimerkiksi, miten eri ohjelmistokomponentit voivat kommunikoida keskenään binääriobjektien tasolla, käsittäen funktioiden kutsut ja niiden parametrit sekä tietorakenteet. (Butterfield et al. 2016)

API-rajapinta voi olla REST API -rajapinta (engl. *Representational State Transfer API*), jolloin se on käytössä internetissä riippumatta sovelluksen ohjelmointikielestä tai se voi olla tietyn ohjelmointikielen tai -kirjaston rajapinta. Tässä toteutustavassa hajautetut

järjestelmät pystyvät kommunikoimaan keskenään verkon yli. (Moilanen et al. 2018) REST API -rajapinta tarjoaa tehokkaan tavan integroida ja kommunikoida eri järjestelmien välillä. Sen ominaisuuksien kuten yksinkertaisuuden, laajennettavuuden ja joustavuuden vuoksi se on yleinen valinta suunnittelumalliksi internetissä API:en luomiseen. (Hou et al. 2016) API-rajapinnat mahdollistavat siis järjestelmien välisen integraation, jossa järjestelmät ovat vuorovaikutuksessa keskenään ja vaihtavat tietoja toistensa kanssa sekä kuvaa järjestelmän sisäisten toimintojen, rakenteiden ja tekniikoiden yhteen toimivuuden.

Ohjelmointirajapinnat ovat yksi esimerkki, jonka avulla ETL-prosessi pääsee käsiksi tietojärjestelmän dataan; ne mahdollistavat datan hakemisen tietojärjestelmästä ja sen siirtämisen takaisin tietojärjestelmään. Seuraavaksi keskitytään integraatioihin ja integraatiokäsitteeseen yleisellä tasolla sekä tarkastellaan järjestelmien välisiä integraatioita ja niiden ominaispiirteitä ETL-prosessin näkökulmasta.

3.9 Integraatiot yleisesti

Barrett et al. (1996) määrittelevät integraatiokäsitteen tietotekniikan toimintaympäristössä prosessiksi, jossa useita ohjelmistomoduuleja (ohjelmia, aliohjelmia jne.) saadaan toimimaan yhdessä. Tämä voi kattaa useita eri osa-alueita, kuten tietojen integraation, ohjelmistojen integraation ja järjestelmien integraation. Integraatiot mahdollistavat järjestelmien, sovellusten ja tietolähteiden yhdistämisen ja saumattoman toiminnan keskenään. (Barret et al. 1996)

Gulledge (2006) kuvaa integraatiokäsitettä omassa katselmuksessaan, jossa hän pyrkii selventämään integraatiokäsitteen tarkoitusperää. Gulledge (2006) viittaa integraatiokäsitteellä tiettyihin rajattuihin ja suljettuihin joukkoihin liiketoimintaprosesseja, joihin liittyvät tiedot käsitellään samassa ohjelmistosovelluksessa ja jossa tiedot jaetaan toisten ohjelmistojen kesken. Ohjelmistosovelluksen päivitykset yhdessä moduulissa tai komponentissa heijastuvat koko liiketoimintaprosessilogiikan toimintaan ilman monimutkaista ulkoista rajapintaa. Ohjelmistosovelluksen tiedot tallennetaan kerran ja ne jaetaan välittömästi kaikkien ohjelmistosovelluksen käytössä olevien liiketoimintaprosesseihin liittyvien ohjelmistojen kesken. (Gulledge 2006)

Voidaan siis todeta, että integraatiokäsite tarkoittaa yleisesti vuorovaikutusprosessia kahden tai useamman erillisen kokonaisuuden välillä, esimerkiksi ohjelmistojen, sovellusten tai laitteiden välillä. Integraatioprosessi luo kommunikointiyhteyden näiden kokonaisuuksien välille, mikä mahdollistaa niiden yhteistoiminnan ja viestiyhteyden.

Myös tietovarastoinnin ETL-prosessia voidaan tarkastella integroinnin näkökulmasta. Hovi et al. (2009) toteavat, kuinka ETL-prosessi voidaan kuvata integrointimenetelmänä sen käsittävien ominaispiirteiden mukaan (kts. luvut 3.2 ja 3.5).

Gartner (2024a) kuvaa integroitua järjestelmää liiketoimintaohjelmistona, joka liittää useita sovelluksia, muun muassa tekstinkäsittelyä, tietokannan hallintaa, taulukkolaskentaa, grafiikkaa ja viestintää, yhdeksi tuotteeksi, mikä mahdollistaa tietojen jakamisen kaikkien tai useimpien moduulien välillä järjestelmän toiminnassa. Integroitu järjestelmä on siis tietokoneohjelmisto, joka yhdistää useita toiminnallisuuksia ja sovelluksia yhdeksi kokonaisuudeksi tarkoittaen, että integroidussa ohjelmistossa on erilaisia toimintoja tai sovelluksia, jotka on yhdistetty yhdeksi käyttöliittymäksi tai sovellukseksi. (Gartner 2024a)

Toisaalta Gartner (2024b) näkee integraatiot palveluina, joilla tuotetaan yksityiskohtaisia suunnittelu- ja toteutuspalveluita, jotka yhdistävät sovellustoiminnallisuuden (kustomoitua ohjelmistoa tai standardinmukaista pakettiohjelmistoa) sekä tietoja toistensa kanssa joko olemassa olevaan tai suunniteltuun organisaation tietotekniikkainfrastruktuuriin. Integraatiopalveluiden erityisiin toimintoihin saattaa usein myös kuulua projektisuunnittelua, projektinhallintaa, sovellusohjelmointirajapintojen, verkkopalveluiden tai välitysohjelmistojärjestelmien yksityiskohtaista suunnittelua tai toteutusta. (Gartner 2024b)

3.9.1 Tietojärjestelmien väliset integraatiot

Gartner (2024c) kuvaa järjestelmien välistä integraatiota prosessina, jossa itsenäisesti suunnitellut järjestelmät saadaan toimimaan yhdessä. Järjestelmäintegraatioprosessin toimintaan liittyviä ominaisuuksia ja huomioon otettavia yksityiskohtia ovat muun muassa erillisissä järjestelmissä olevien tietojen yhdenmukaistaminen, erilaisten järjestelmien suorittamien toimintojen integroiminen yhteen toimiviksi ja -sopiviksi kokonaisuuksiksi sekä tarvittavien pääsyoikeuksien salliminen itsenäisesti suunniteltuihin järjestelmiin sekä niiden tietoihin ja toimintoihin, jotta ne näyttävät yhtenäiseltä käyttöliittymällä tai sovelluspalvelulla. Järjestelmien väliset integraatiot sisältävät erilaisia tekniikoita ja menetelmiä, joita hyödynnetään järjestelmien yhdistämiseksi ja saumattoman yhteistyön mahdollistamiseksi. Vaikka integraatioprosessin osapuolten väliset järjestelmät olisi toteutettu eri tekniikoilla ja tietomalleilla, integraatioprosessin tavoitteena on varmistaa, että tiedot välittyvät oikeassa muodossa oikeaan paikkaan halutulla tavalla. (Gartner 2024c)

Hohpe & Woolf (2003) toteavat organisaation integraatioprosessin käsittävän tehtävät, jotka mahdollistavat erillisten ohjelmistojen yhteistyön ja yhteen toimivuuden.

Integraatioprosessi käsittää organisaation sisäisten ohjelmistojen yhteen liittämisen sekä organisaation ulkopuolisten asiakkaiden ohjelmistoihin yhdistymisen. Integraatiolla saadaan yhdistettyä ohjelmistojen toiminnallisuuksia ja pystytään hyödyntämään niiden tarjoamaa yhdistettyä tietoa halutuilla tavoilla. (Hohpe & Woolf 2003) Giordano (2011) kuvaa järjestelmien välistä integraatiota mallina, jossa yhteen liitetään järjestelmien transaktiopohjaisia tietoja sekä räätälöidyissä että standardinmukaisissa järjestelmissä. Giordano (2011) näkee järjestelmien välisen integraation arkkitehtuurisena mallina, jonka tavoitteena on varmistaa eri järjestelmien välinen kommunikointi, tiedon jakaminen ja niiden välinen yhteen toimivuus asetettujen liiketoimintatavoitteiden saavuttamiseksi.

Tietojärjestelmien kehittämistä ei enää nähdä vain erillisinä itsenäisinä kokonaisuuksina, vaan niiden kehitystä ja kehittämistä ohjaa integrointi muihin järjestelmiin. Tietojärjestelmien integroinnin tarkoituksena haetaan muun muassa käyttäjäarvon lisäämistä ja ylläpitokustannusten vähentämistä. Tietojärjestelmien integraation suunnittelussa yksi lähestymistapa on ohjelmistoarkkitehtuuri (kts. luku 3.1.4), joka pyrkii kuvaamaan järjestelmien toiminnan mahdollistavia rakenteita ja komponentteja, jotka mahdollistavat integraation toteuttamisen. (Land & Crnkovic 2003)

Kahden tai useamman tietojärjestelmän välisiä integraatioita voidaan siis kuvata tietojen jakamisena kahden tai useamman sovelluksen välillä. Integraatioprosessi koostuu joukosta teknologioita, jotka mahdollistavat tietojen liikkumisen ja vaihdon eri sovellusten ja liiketoimintaprosessien välillä organisaation sisällä ja/tai organisaation ulkoisten toimijoiden välillä.

Näiden kuvauksen perusteella huomataan tietojärjestelmien välisten integraatioiden samankaltaisuus ETL-prosessin ominaispiirteiden kanssa. Kuten mainittu järjestelmien väliset integraatiot liittyvät tietojen siirtoon ja yhdistämiseen, ja ETL-prosessit ovat yksi tapa toteuttaa tietojen siirto- ja käsittelyprosesseja. Integraatiot pyrkivät mahdollistamaan eri järjestelmien välistä kommunikaatiota ja vuorovaikutusta, jossa ETL-prosessit voivat olla osallisina muodostaen yhteyksiä erilaisten järjestelmien välillä samalla mahdollistaen tietojen siirtämisen ja yhdenmukaistamisen. Järjestelmien välisillä integraatioilla pyritään varmistamaan, että eri järjestelmät toimivat yhteensopivalla tavalla keskenään, jossa ETL-prosessit mahdollistavat tietojen muuntamisen ja yhdistelyn. Tietojärjestelmien väliset integraatiot ja ETL-prosessit ovat keskeisissä osissa tietojärjestelmien välistä tiedonhallintaa ja -käsittelyä.

3.9.2 Integraatioarkkitehtuuri

ETL on suositeltava tietojen integraatioarkkitehtuurinen malli, kun ei-reaaliaikaisen, transaktiopohjaisen datan kertymisen ja sen tarvitsemisen välillä on viivettä. ETL-

prosessilla voidaan hallinnoida, milloin tapahtumat luodaan ja milloin niiden dataa tarvitaan. ETL on myös suositeltava lähestymistapa arkkitehtuurimalliksi, kun useita erilaisia tietomuotoja kerätään eri järjestelmistä, joiden tiedot vaativat muuntamista, yhdistämistä ja kokoamista yhteiseen tiedostomuotoon. (Giordano 2011)

Integraatioita voidaan kuvata integraatioarkkitehtuurin viitekehyksien avulla. Integraatioarkkitehtuuri tarjoaa yleisen viitekehyksen järjestelmien väliselle tiedonvaihdolle ja yhteensovittaa tietomuodot eri järjestelmien välillä. Integraatioarkkitehtuuri tarjoaa tietopalveluita, jotka on muodostettu vastaamaan organisaation liiketoimintatarpeita ja joita kehitetään näiden tarpeiden muuttuessa. Integraatioarkkitehtuurin viitekehys on avoin järjestelmien välinen tietovirtakuvaus, johon voidaan lisätä ja/tai poistaa viitekehyksessä kuvattujen järjestelmien osajärjestelmiä vaikuttamatta järjestelmän toimintalogiikkaan tai suorituskykyyn. (Mylopoulos et al. 1996) Mylopoulos et al. (1996) näkevät, että integraatioarkkitehtuurin kaksi ensisijaista suunnittelutavoitetta tulisi olla sen modulaarisuus ja kyky toimia erilaisissa integraatioiden käyttötapauksissa.

Jung et al. (2007) kuvaavat järjestelmien välistä integraatiota integraatioarkkitehtuurin kautta. Integraatioarkkitehtuuri auttaa järjestelmien rakenteen suunnittelussa ja sen avulla määritellään, miten erilaiset tietojärjestelmät kommunikoivat ja integroituvat toisiinsa. Se luo perustan tiedon ja toiminnallisuuksien jakamiselle organisaation eri järjestelmien välillä. Integraatioarkkitehtuuri on suunnittelukehys, joka mahdollistaa eri järjestelmien ja sovellusten yhteistoiminnan ja tiedonjakamisen. (Jung et al. 2007)

Lankhorst (2004) lähestyy organisaation integraatioarkkitehtuuria kokonaisarkkitehtuurin kautta. Organisaation kokonaisarkkitehtuuri on tärkeä viitekehysmalli organisaation kokonaisvaltaisten integraatioiden käsittelemiseksi. Se on johdonmukainen kokonaisuus organisaation toimintoja, periaatteita, menetelmiä ja malleja, jotka ovat käytössä organisaation organisaatiorakenteen, liiketoimintaprosessien, tietojärjestelmien ja infrastruktuurin suunnittelussa ja toteutuksessa. Koska kokonaisarkkitehtuurimalli ei kuitenkaan lähesty näitä osa-alueita riittävän integroidusti, integraationäkökulma edellä, tulee avuksi integraatioarkkitehtuurimalli. Integraatioarkkitehtuurimalli auttaa organisaation osa-alueita yhteen toimivaan kommunikointiin yhteisten, integroitujen mallien ja tekniikoiden avulla, jossa viestintä ja päätöksenteko on integroitu osa-alueiden välillä. Integraatioarkkitehtuuri auttaa olemassa olevien tekniikoiden yhdistämiseen ja integroimiseen sopivalle abstraktiotasolle organisaation toiminnassa sekä auttaa kuvaamaan organisaation toimintojen ja tasojen vuorovaikutussuhteita syvemmällä merkityksellä. Integraatioarkkitehtuuri pyrkii yhdistämään organisaation toiminnot, ohjaa

järjestelmien kommunikointikanavien luonneissa ja tietovirtojen määrittelyssä. (Lankhorst 2004)

Integraatioarkkitehtuuri siis määrittelee, miten eri järjestelmät ja sovellukset kommunikoivat ja vaihtavat tietoa keskenään. ETL-prosessi on yksi esimerkki tähän järjestelmien väliseen tiedonsiirtoprosessiin (Giordano 2011) sen tunnusomaisten piirteiden mahdollistamien toimintojen kautta. ETL-prosessin avulla voidaan toteuttaa tietojen poiminta, niiden muuntaminen ja yhdistely sekä siirtäminen eri järjestelmien välillä. Integraatioarkkitehtuurin avulla määriteltyjä tietovirtoja voidaan toteuttaa ETL-prosessien ominaisuuksia hyödyntäen käytännön tasolla sen sisältämiä toimintoja hyväksikäyttäen. Integraatioarkkitehtuuri kuvaa järjestelmien välisen integroidun tietojen hallintaprosessin, jossa yhdistyy ETL-prosessille tyypilliset tietojenkäsittely- ja tiedonhallintaprosessit.

4. TUTKIMUKSEN KOHDEORGANISAATIO

Tässä luvussa esitellään tutkimuksen kohdeorganisaatio, sen LeanwareWMS-tietojärjestelmä pintapuolisesti, jotta ymmärretään tutkimuksen tapausorganisaation toimintakenttää sekä niiden järjestelmän toiminnallisuutta. Luvussa kuvataan myös kohdeorganisaatiossa tunnistetut kriteerit vaihtoehtoiselle ratkaisulle, mikä rakentaa yhdessä teoreettisen kirjallisuuden ja aineistojen tarkastelun kanssa perustan vaihtoehtoisten menetelmien selvitykselle kohdeorganisaation ETL-prosessin toimintaan sopivista tiedonsiirtomenetelmistä.

4.1 Leanware Oy

Leanware Oy on tuotannon ja logistiikan asiantuntija- ja ohjelmistotalo, joka tarjoaa asiantuntijapalveluita sekä ohjelmistoratkaisuja kaupan, teollisuuden ja logistiikan tarpeisiin. Asiakkaiden operaattisia toimintoja ohjaavia ohjelmistoratkaisuja ovat yrityksen omat ohjelmistot: WMS-, MES- (Manufacturing Execution System) ja SCM-tietojärjestelmät (Supply Chain Management System). Lyhykäisyydessään kuvattuna WMS ohjaa varastonhallintaa ja optimoi varastotasoja, MES ohjaa tuotannon operaattisia toimintoja sekä digitalisoi tehtaan toimintaa ja SCM ohjaa tuotteiden toimitusketjua, jossa tuotteiden osto- ja hankintaprosessit ovat optimoitu älykkäillä ratkaisuilla. (Leanware 2024b)

Järjestelmät mahdollistavat tietoperusteisen toiminnan kehittämisen, jossa älykkäät järjestelmäratkaisut ohjaavat ja digitalisoivat asiakkaiden toimintaa. Tiedolla johtamiseen pohjautuvat tietojärjestelmät mahdollistavat tietoperustaisen arvonluonnin, jossa asiakkaiden päätökset perustuvat tietojärjestelmien visuaalisessa muodossa esitettyyn dataan. (Leanware 2024b) Leanware hyödyntää omassa strategiassaan tietojohtamisen perusaatetta, jossa tieto on merkittävässä roolissa organisaatioiden menestymisessä. (Laihonen et al. 2013)

Diplomityössä keskitytään Leanwaren WMS-tietojärjestelmään ja sen taustalla olevaan ETL-prosessiin, jotta voidaan rajata työn aihe diplomityöohjeiden lainalaisuuksien mukaan ilman, että sen viitekehys kasvaa liikaa. Diplomityön tavoitteena on kartoittaa mahdollisia vaihtoehtoisia menetelmiä WMS-tietojärjestelmän ETL-prosessiin ja siinä käytettäviin käsin kirjoitettuihin koodeihin. WMS-tietojärjestelmän ETL-prosessin koodit ovat vaikealukuisia ja niiden muutokset vaativat kokonaisvaltaista koko koodikokonaisuuden ymmärtämistä sekä WMS-järjestelmän toiminnan syvällistä

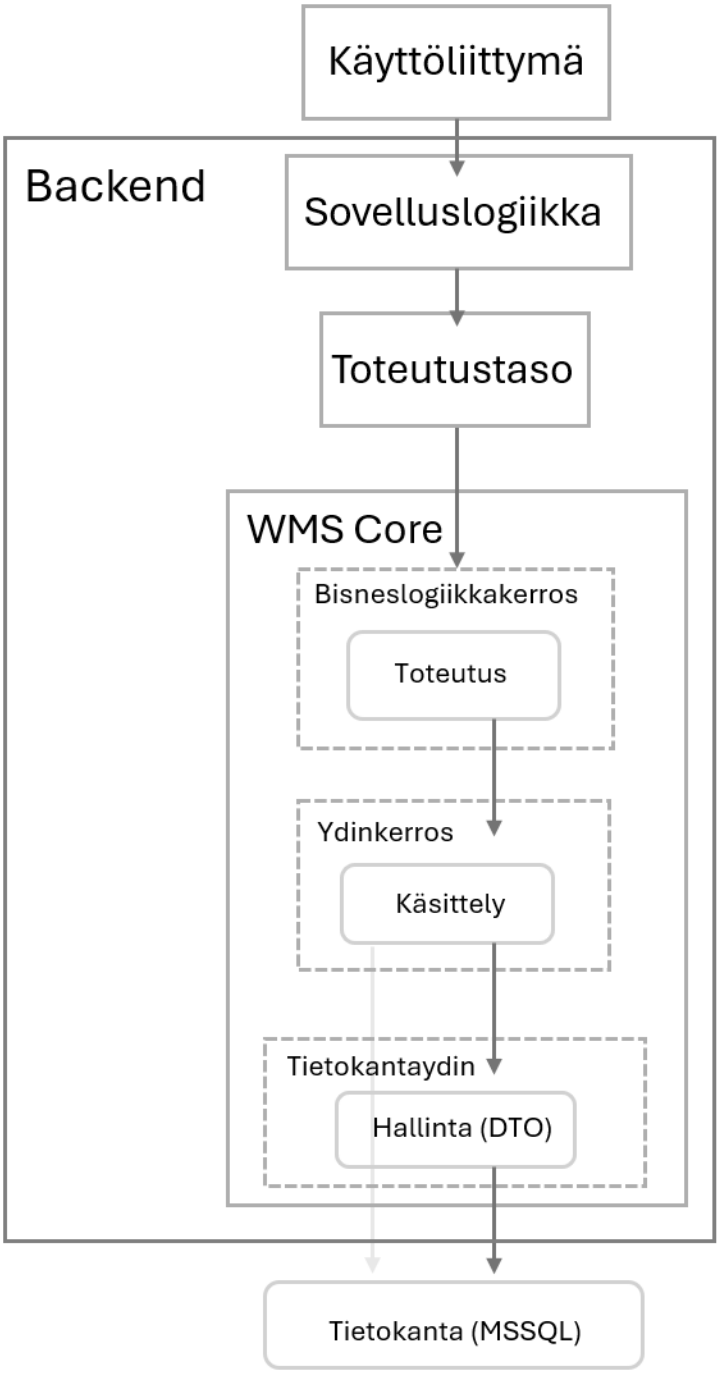
tuntemista. Työssä pyritään selvittää, kuinka pystyttäisiin tehostaa WMS-tietojärjestelmän ETL-prosessia sekä sen kehitys- ja ylläpitotyötä, jotta päästään eroon työtuntien hukkaamisesta vaikeasti kirjoitetun koodin ymmärtämiseen ja siihen kulutetusta hukka-ajasta. Kun datan määrä kasvaa ja palvelut kehittyvät, on tärkeää kehittää datan luku- ja siirtoprosesseja vastaamaan tähän muutokseen entistä tehokkaammalla, helppokäyttöisellä ja käyttäjäystävällisellä lähestymistavalla. Tämä vaatii uusien ja paranneltujen työkalujen, tekniikoiden, menetelmien ja käytäntöjen tutkimista ja käyttöönottoa, jotta voidaan mahdollistaa datan käsittelyprosessien kokonaisvaltainen tehostaminen.

WMS-tietojärjestelmän ETL-prosessi toimii esimerkkitapauksena, jonka pohjalta saatuja tuloksia ja esiin tulleita löydöksiä voidaan myöhemmin laajentaa muihin Leanwaren järjestelmien tiedonsiirron taustalla oleviin ETL-prosesseihin. Seuraavassa alaluvussa paneudutaan tarkemmin Leanwaren WMS-tietojärjestelmään ja sen ominaispiirteisiin.

4.2 LeanwareWMS-järjestelmä

LeanwareWMS-järjestelmä toimii kerrosarkkitehtuurin suunnitteluperiaatteen mukaan, jossa jokainen kerros tarjoaa edellistä kerrosta korkeamman abstraktiotason tietorakenteiden käsittelyyn. Useasta kerroksesta muodostuva luokkahierarkia jakautuu sen funktionaalisten toiminnallisuuksien mukaan. (Leanware 2024a) Kuten luvussa 3.1.5 todetaan, kerrosarkkitehtuurin perusajatuksena tietyn tason komponentti tai palvelu käyttää hyväksi alemman kerroksen komponentteja tai palveluita. Näin saadaan muodostettua järjestelmien rakentamista helpottava *dekompositio*-periaate, jossa kerroksia pilkotaan pienempiin, helpommin ymmärrettäviin osiin, mikä samalla jakaa järjestelmän toimintoja ryhmiin ryhmittelyperiaatteen mukaan.

Kuvassa 13 esitellään yksinkertaistetusti LeanwareWMS-järjestelmä kerrosarkkitehtuurin suunnitteluperiaatteen pohjalta, missä kuvataan järjestelmän kerrokset ja niiden toiminnallisuuksia. Kuvasta 13 huomataan, kuinka järjestelmän toiminnot ovat jaettu osiin niiden toiminnallisuuksien mukaan. Kuvassa 13 WMS Core -kerros on avattu vielä tarkemmin kolmeen kerrokseen sen toiminnallisen merkityksen vuoksi järjestelmän toiminnalle. WMS Core -kerrostaso voidaan nähdä järjestelmän sieluna, jossa tapahtuu järjestelmälle ominaisten käyttötapausten toteutukset ja niiden hallinta. Kerrosten yleisten toiminnallisuuksien ominaisuudet kuvattiin esimerkkitapauksen pohjalta luvussa 3.1.5.



Kuva 13. LeanwareWMS-järjestelmän kerrosarkkitehtuuri (mukaillen Leanware 2024a)

Kuvan 13 nuolet kuvaavat palvelukutsuja, joissa dataa siirretään järjestelmän kerroksien välillä päätyen järjestelmään liitettyyn tietokantaan. Kuvasta 13 huomataan, kuinka WMS Core -kerroksen sisällä Ydin-kerros voi ohittaa Tietokantaydin-kerroksen siirtyen suoraan Tietokanta-kerrokseen. Tämänkaltaista ohitusta kuvataan *bridging*-termillä, jossa hypätään kerroksien ylitse ylhäältä alaspäin.

Leanwaren tietojärjestelmät perustuvat yhteiseen tuotealustaan. Tuotealusta kuvaa kokonaisuutta, mikä koostuu erilaisista moduuleista ja parametreista, joita eri tuotteet

hyödyntävät perustanaan. Tuotealustan tarjoamat osakokonaisuudet ja rakenteet muodostavat perustan, jonka päälle tuotteita kehitetään ja ylläpidetään. Tuotealusta muodostaa pohjan, jota voidaan käyttää uudelleen tuoteperheen muissa tuotteissa. (Gao et al. 2009; Pakkanen et al. 2020)

Leanwaren tuoteperheen tietojärjestelmät pohjautuvat samaan perusideaan ja rakenteeseen (tuotealustaan), jotka on optimoitu tuotteiden omille toiminta-alueille erityisillä spesifikaatioilla, toiminnallisuuksilla eri käyttötarkoituksiin järjestelmien tarpeiden ja vaatimusten mukaisesti. Koskimies & Mikkonen (2005) tunnistaa saman tuoteperheen toteuttavan ohjelmiston tuoterunkona, joka mahdollistaa tuoteperheen yhteisen rakenteen ja toiminnallisuuden. Pulkkinen (2007) määrittää tuoteperheen joukoksi tuotteita, joilla on yhtenäisiä ominaisuuksia ja piirteitä. Tuoteperhe koostuu yhteisestä tuotealustasta ja tuoteperheen tuotteet jakavat yhteisiä komponentteja tai muita keskeisiä elementtejä.

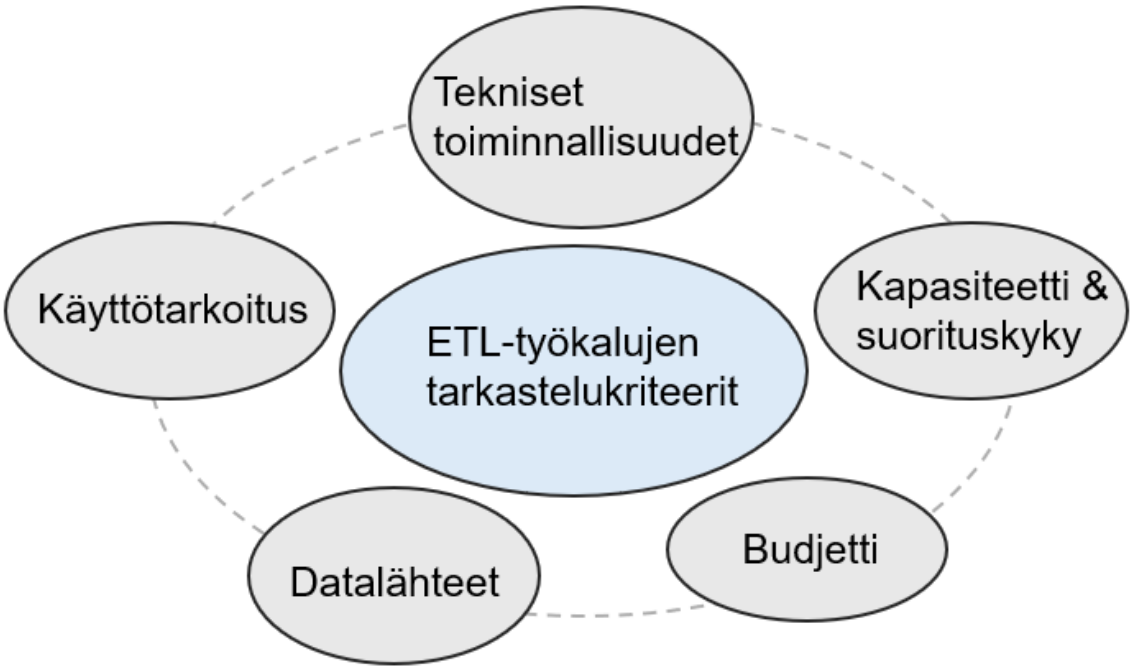
Tässä tutkimuksessa keskitytään Leanwaren tuoteperheen LeanwareWMS-tietojärjestelmään ja sen ETL-prosessiin, jota käytetään esimerkkitapauksena, johon tutkimusongelma pohjautuu ja johon tutkimuskysymykset kohdistetaan. Tämän esimerkkijärjestelmän ETL-prosessin toiminnan kautta muodostetut valintakriteerit ohjasivat kohdeorganisaation ETL-työkalun analyysia.

4.3 ETL-työkalun valintakriteerit

Työpajakeskustelut toimivat pääasiallisena kohdeorganisaation tarpeiden tunnistavana tekijänä, jossa luotiin valintakriteerit kohdeorganisaation tapaukseen sopivalle ETL-työkalulle. Työpajakeskusteluiden tavoitteena oli luoda ympäristö, jossa osallistujat pystyivät myötävaikuttaa omalla asiantuntijuudellaan vaadittavien ETL-työkalun ominaisuuksien vaatimuksista tuoden aiheeseen oman näkemyksensä. Työpajakeskusteluiden pääasiallinen tavoite oli kerätä monikantainen näkemys ETL-työkalun ominaisuuskriteereistä, minkä pohjalta luotiin lista ETL-työkalulta tarvittavista ominaisuuksista kohdeorganisaation näkökulmasta.

Kirjallisuudessa (Badiuzzaman Biplob et al. 2018; Hovi et al. 2009; Kherdekar & Metkewar 2016; Mali & Bojewar 2015; Qaiser et al. 2023; Xavier & Moreira 2013; Zode 2007) tunnistetut ETL-työkalujen yleiset ominaisuudet yhdistettiin luetteloksi, jonka avulla voitiin tarkastella ETL-työkalujen yleisiä ominaisuuksia ja validoida tarvittavat vaatimukset kohdeorganisaation tarpeisiin. Esimerkiksi Khan et al. (2024) esittävät omat näkemyksensä ETL-työkalujen yleisistä tarkastelukohteista, jotka ovat sovellettavissa jokaisen organisaation tarpeisiin ETL-työkaluihin liittyen (kuva 14), mutta lisää, kuinka

jokaisella organisaatiolla on oma liiketoimintastrategiansa ja organisaatiokulttuurinsa, mikä heijastuu yrityksen keräämässä ja arvostamassa datassa, johon työkalun pitäisi taipua. (Khan et al. 2024)



Kuva 14. ETL-työkalun yleiset tarkastelukriteerit (mukaillen Khan et al. 2024)

Kuvan 14 esitettyjen tarkastelukriteerien lisäksi ETL-prosessien monimutkaisuus, käsiteltävän datan eheys sekä vuorovaikutus lähde-dataplatformien kanssa, organisaation pilvipalveluiden toimittajakumppanuudet ja muut pilvipalveluyhteydet sekä organisaation sisäinen työkalun kehitysosaaminen ovat esimerkkejä tekijöistä, joita on tarpeen huomioida ETL-työkalua valitessa. (Khan et al. 2024)

Kirjallisuudessa (Badiuzzaman Biplob et al. 2018; Hovi et al. 2009; Kherdekar & Metkewar 2016; Mali & Bojewar 2015; Qaiser et al. 2023; Xavier & Moreira 2013; Zode 2007) tunnistetut yleiset ETL-työkalun ominaisuuskäsitteet jäsenneltiin erillisiksi kokonaisuuksiksi niiden symboloivien merkityksien mukaan ylläpito-, skaalautuvuus- ja tehokkuusominaisuuksiin sekä käyttöön ja toimintoihin kuin myös kustannuksiin liittyvien tarkastelukohtien mukaan:

Ylläpito-ominaisuudet: muutosten hallinnan tuki sekä tuki työryhmätyyppiselle prosessien kehittämiselle, prosessien testaustuki, prosessien laadun määrittely ja raportointi, virhetilanteet; niiden raportointi, tutkiminen, hälytys, jäljitys ja ilmoitukset, latausajojen lokitus ja tilastointi, prosessien muutosten versiointi ja versionhallinta, kirjausketju (*audit trail*), joka kertoo missä vaiheessa mikäkin data on lisätty tai päivitetty, muutosvaikutuksien hallinta, palvelutuki, prosessien uudelleenkäytettävyys ja muunnoslogiikka, prosessiarkkitehtuurin suunnittelun tukeminen, kehittäminen ja

datahallinnon tuki, virheenkorjaustoiminnot, käytettävyystuki sekä yleinen ETL-infrastruktuurin tukeminen.

Skaalautuvuusominaisuudet: erilaisiin lähdejärjestelmiin kytkeytymismahdollisuudet, erilaisten datamuotojen tukeminen, prosessien uudelleenkäytettävyys ja -hyödynnettävyys, tuki muista järjestelmistä siirretylle datalle ja datansiirto muihin järjestelmiin, palvelurajapintojen tuki, jotka tarjoavat helpot ja nopeat liitäntämahdollisuudet myös muihin järjestelmiin, automaatioratkaisut, prosessien yhdistettävyys, yhteystila ja alustaratkaisu sekä sovittimen tuki.

Tehokkuusominaisuudet: suorituskyky ja kapasiteetti sekä niiden muutosmahdollisuudet, tietokantaoperaatioiden suorittamismahdollisuudet, laskentatehon ja prosessien suoritusajan mahdolliset muutokset kokonaiskuormituksen mukaan sekä muut prosessien tehokkuuteen vaikuttavat tekijät.

Käyttöön ja toimintoihin liittyvät ominaisuudet: graafinen käyttöliittymä, tuki erilaisille tiedonsiirtomenetelmille, esimerkiksi SFTP-, FTP-, HTTP-, HTTPS-, SOAP-, REST-menetelmille, funktiotoiminnot (*Extract*, *Transform*, *Load*) ja niiden joustavuus tietojen käsittelemiseen, tietojen yhdistelemiseen eri lähteistä ja niiden lataamiseen, tuplarivien poistomahdollisuudet lähdetiedoista, tietomallinnus, datamuunnokset datan laadun profilointi, monitorointi ja hallinta sekä erälataustietojen ajastaminen ja prosessiajojen aikataulutus joko jatkuva lataus -tyyppisenä tai tapahtumien mukaan, reaaliaikainen tietojen integrointi ja analytiikkapalvelut, pilvipohjainen käyttö ja tuki, rinnakkaisten prosessien ajomahdollisuudet, metadatan hallinta, ratkaisun helppokäyttöisyys, käyttöönotto ja käyttöönoton helppous eri ympäristöjen välillä sekä moniroolisen tiimityön tukeminen, jossa ETL-työkalu edistää ongelmien ratkaisua ja ryhmien välistä kommunikointia, tiedon jakamista ja yhteistyötä.

Kustannuksiin liittyvät tarkastelukohdat: Aloitusmaksu, kuukausihinta, erilliset käyttömaksut sekä yleinen hinta-laatusuhde; toimintojen ja niistä saatavien hyötyjen suhde kustannuksiin, mahdollisten lisäpalveluiden hinta ja ETL-työkalun käyttöönottoon, käyttöön ja käyttöjärjestelmään liittyvät kustannukset kuin myös asiakaspalvelutuen kustannukset.

Työpajakeskusteluissa muodostettiin ETL-työkalun vaatimukset näihin yleisesti tunnistettuihin ominaisuuksiin peilaten, ja ne toimivat kohdeorganisaation kriteeristön muodostajana. Työpajakeskusteluissa muodostettiin WMS-järjestelmän ETL-prosessiin sopivalle menetelmälle olennaiset vaatimukset ETL-työkalujen yleisesti tunnistettujen ominaisuuksien pohjalta. Kirjallisuudessa esitettyjen ETL-työkalujen ominaisuusvaihtoehtojen pohjalta käytiin työpajakeskusteluita ja peilattiin tunnistettuja

ominaisuuksia kohdeorganisaation ETL-prosessiin ja siihen sopivaan ratkaisuun. Työpajakeskusteluiden tarkoitus oli tunnistaa ETL-työkalulta vaadittavia ominaisuuksia kohdeorganisaation käyttötapauksissa ja siinä toimimisessa.

Työpajakeskusteluihin ja siihen liittyvään viestintään osallistujat olivat ETL-prosessin syvällisesti tuntevia asiantuntijoita, muun muassa Chief Product Manager, WMS- ja SCM-tuotepäälliköt, WMS-arkkitehti, BI Developer ja Integration Manager, joiden ammattitaitoa hyödyntäen pystyttiin tunnistamaan ja rajaamaan vaadittavat ratkaisuvaihtoehdon ominaisuuskriteerit juuri kohdeorganisaation tarpeisiin.

Ominaisuuskriteerit tunnistettiin ETL-prosessin toimintojen ja niihin liittyvien tunnusomaisten piirteiden avulla, jossa erityisesti ETL-työkalun funktiotoiminnot ja niiden sopivuus Leanwaren toimintaan, useiden ympäristöjen tapauskohtainen huomioinen, prosessien uudelleenkäytettävyys, pilvipohjaisuus, palvelurajapintojen tuki kuin myös versiohallinnan tuki sekä prosessien, virhetilanteiden ja muutosten ylläpito ja hallinta nousivat työpajakeskusteluissa esiin. Myös työkaluun liittyvät maksut ja mahdolliset käyttökustannukset huomioitiin kriteerien valinnoissa ja olivat vahvasti esillä keskusteluissa.

Työpajakeskustelut sekä niihin liittyvä viestintä mahdollisti kokeneiden asiantuntijoiden mielipiteiden ja ideoiden keräämisen, joita pystyttiin yhdessä rikastamaan ja tarpeen mukaan kehittämään työpajojen avulla. Kommenttien ja ajatusvaihtoprosessien avulla saatiin luotua laajempia näkökulmia aiheeseen, mikä auttoi myös tunnistamaan vaihtoehtoisia vaatimuksia ratkaisuvaihtoehdolle. Työpajakeskustelut järjestettiin vuoden 2024 maaliskuun ja huhtikuun aikana ja ne pohjautuivat aikaisempiin, noin vuoden takaisiin keskusteluihin, mitkä muodostivat alustuksen ja kontekstin aiheeseen. Työpajakeskustelut pidettiin vapaamuotoisina *brainstorm*-keskusteluina, joissa keskityttiin aihepiirikokonaisuuteen ja sen muodostamaan viitekehykseen. Työpajakeskustelut pyrittiin pitämään joustavina, tehokkaina ja tiiviinä aihepiirikokonaisuuden ympärillä, missä keskityttiin osallistujien väliseen vuorovaikutukseen, eikä ohjattu keskusteluita sen tarkemmin. Tutkijan rooli pyrittiin pitämään vain keskusteluihin ja aiheeseen alustajana sekä työpajakeskusteluiden alustavan agendan selittäjänä ja vain tarpeen mukaan ohjattiin keskusteluita, muuten haastattelijan roolia keskusteluissa ei käytetty. Tällä pyrittiin vähentämään puolueellisuuden riskiä ja tutkijan omia subjektiivisia päätelmiä ja näkemyksiä aiheeseen liittyen.

Työpajakeskusteluiden pohjalta tunnistettiin ETL-työkalun vaadittavat ominaisuudet kohdeorganisaation käyttötapauksiin ja heidän toimintaympäristöönsä. Ylläpito-,

skaalautuvuus- ja tehokkuusominaisuudet sekä käyttöön, toimintoihin ja käyttökustannuksiin liittyvät yksityiskohdat nousivat olennaisimmiksi vaatimuksiksi, joita työkalulta edellytetään. Alla on listattu työkalulta vaadittavat ominaisuudet ja muut tarpeet kohdeorganisaation näkökulmasta:

- ETL-työkalun tulee mahdollistaa tietojen hakeminen operatiivisesta SQL Server-tietokannasta, suorittaa tarvittavat datamuunnokset, datan yhdistelyä ja loogista päättelyä sekä syöttää muokattu tieto toisessa muodossa SQL Server- tai PostGre-tietovarastokantaan.
- Asiakkuuksien ympäristöjen huomiointi; Azure-pilviasiakkaat vs. paikallisten ympäristöjen asiakkaat. Operatiivinen tietokanta ja tietovarastokanta sijaitsevat aina samassa ympäristössä, jolloin datansiirto pitää ottaa myös huomioon paikallisissa ympäristöissä.
- ETL-työkalu mahdollistaa tietovarastotietokannan tietokantarakenteiden ylläpidon ja hallinnan.
- ETL-työkalun pitää pystyä tukemaan tietomallin ja ETL-prosessin asiakaskohtaista versiointia – versiohallinnan ja muutostenhallinnan tuki sekä räätälöintien ja muiden asiakaskohtaisten muutoksien huomiointi ja mahdollistaminen.
- ETL-prosessin ETL-työkalun on kyettävä hallitusti ja automaattisesti päivittämään muutokset asiakkaille, esimerkiksi integraatioputken toimintojen ja tehtävien avulla. ETL-prosessien ajoväli ja aikataulutus pitää olla määritettävissä asiakaskohtaisesti.
- ETL-prosessin ETL-työkalun pitää tukea erilaisia asiakaskohtaisia skenaarioita tuotteiden ja räätälöintien suhteen. Esimerkkiskenaarioita: asiakkaalla on yksi tuote tai yksi tuote ja siihen liittyvät räätälöinnit tai monta tuotetta tai monta tuotetta räätälöintien kanssa.
- ETL-työkalussa pitää olla virhetilanteiden monitorointi niiden hallinnalle ja ylläpidolle sekä tietojen oikeellisuuden tarkastuksen ja validoinnin seurantamahdollisuudet.
- ETL-työkalun helppokäyttöisyys sekä käyttöönoton helppous ja prosessien uudelleenkäytettävyys uusien asiakkaiden luonnissa.
- ETL-työkalun hintakustannukset sekä niiden skaalautuvuus asiakasmäärien, asiakaskohtaisten räätälöintien ja asiakaskohtaisten ympäristöjen mukaan kuin myös muut käyttökustannukset.

- Kohdeorganisaation mahdollinen etukäteisosaaminen ETL-työkalusta ja sen toiminnoista.
- Rinnakkaisten ETL-prosessien ajomahdollisuudet.
- ETL-työkalun pilvipohjainen käyttömahdollisuus sekä ETL-työkalun tietoturvallisuuden huomiointi.
- Muut ETL-työkalun funktiotoiminnot sekä ETL-prosessin toiminnallisuuksia tukevat ominaisuudet, esimerkiksi tietojen muokkaus ja ylläpito, tietojen päivitys ja tarvittaessa korjaus ja yhdistely sekä tietojen kokonaisvaltainen hallinta ja monitorointi.

Lisäominaisuudet, jotka lisäävät ETL-työkalun valintamahdollisuutta ja nähdään etuna myös muiden järjestelmien näkökulmasta:

- ETL-työkalun mahdollistama tuki tulevaisuudessa kohdeorganisaation kaikkien tuotteiden yhteiselle tietovarannolle.
- ETL-työkalun tarjoama palvelurajapintojen tuki; helpot ja nopeat liitäntämahdollisuudet myös muihin järjestelmiin.
- ETL-työkalun suorituskyky ja kapasiteetti ja muut prosessien tehokkuutta lisäävät tekijät
- Suurten datamassojen hallinta, erilaisten datamuotojen käsittelyt sekä tietojen takaisinkirjoitus mahdollisuus tietovarastosta järjestelmään.
- Yleisesti SAAS-liiketoiminnan (Software as a Service -liiketoimintapalvelu) tukeminen. ETL-työkalu mahdollistaa prosessien ja niiden osien toistettavuuden, monistettavuuden, tuotteistettavuuden, uudelleenkäytön ja modulaarisuuden.

Näiden työpajakeskusteluiden pohjalta tunnistettujen ominaisuuksien pohjalta lähdettiin etsimään mahdollisia ratkaisuja kohdeorganisaation ETL-prosessiin ja siihen liittyviin toimintoihin.

5. TUTKIMUKSEN LÖYDÖKSET JA NIIDEN TARKASTELUT

Luvussa 4.3 esiteltiin yleisesti kirjallisuudessa tunnistettuja ETL-työkalun ominaisuuskäsitteitä, joiden pohjalta luotiin kohdeorganisaation valintakriteerit, jotka ohjasivat vaihtoehtoisen ratkaisun valinnassa. Tunnistetut ominaisuuskäsitteet jäsenneltiin erillisiksi kokonaisuuksiksi niiden symboloivien merkityksien mukaan. Ominaisuuskäsitteet jaoteltiin ylläpito-, skaalautuvuus- ja tehokkuusominaisuuksiin sekä käyttöön ja toimintoihin kuin myös kustannuksiin liittyvien tarkastelukohtien perusteella. Kirjallisuudessa tunnistettujen ETL-työkalujen ominaisuuksien perusteella pystyttiin luomaan kohdeorganisaation valintakriteerit, jotka ohjasivat vaihtoehtoisen ratkaisun analyysia.

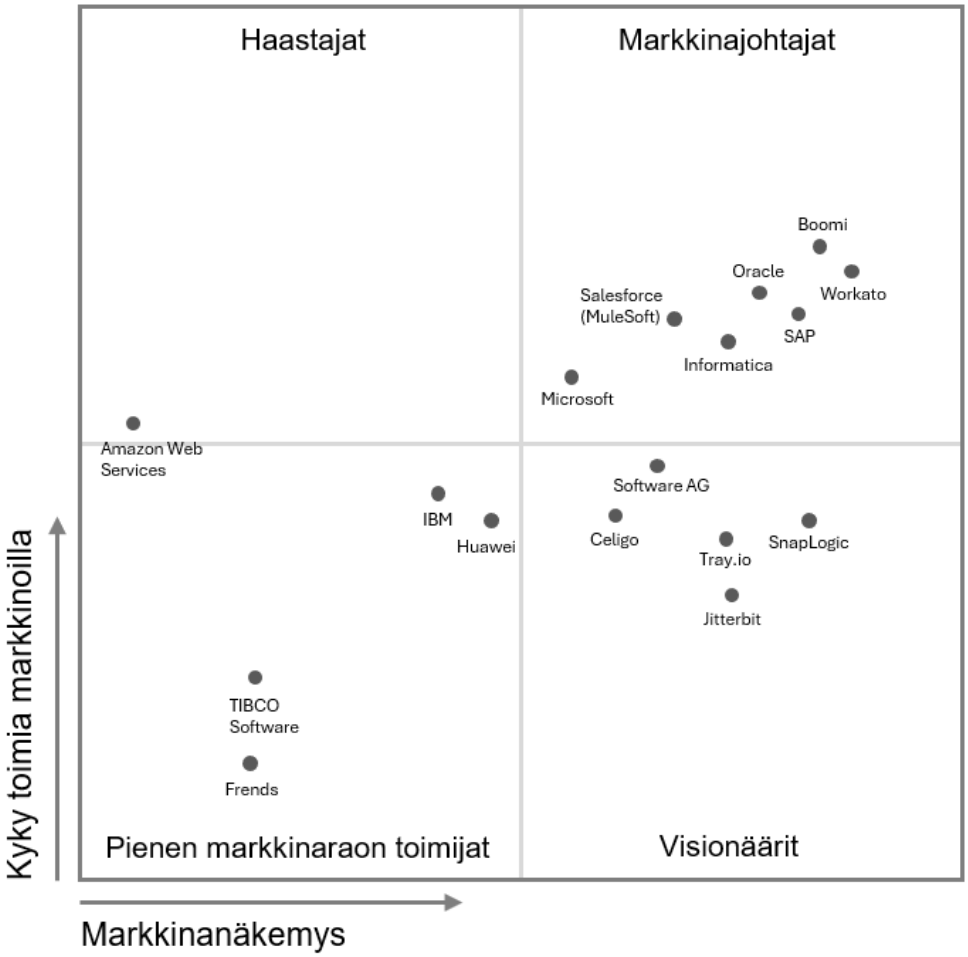
ETL-prosessin menetelmiä, teknologioita ja ETL-työkaluja voidaan ryhmitellä niiden ominaisuuksien tai niihin kohdistuvien vaatimusten kautta. Esimerkiksi Gartnerin (2024d) tekemässä Magic Quadrant™ for Integration Platform as a Service -raportissa ryhmitellään ETL-työkalujen toimittajia, joiden tarjoamat työkaluratkaisut he ovat nimenneet datan integrointityökaluiksi, niihin kohdistuvien käyttötapausten kautta. Hovi et al. (2009, s. 60) ovat huomanneet, että ETL-työkalujen toimittajat haluavat mielellään kutsua ETL-työkaluja integrointityökaluiksi korostaen niiden käytettävyyttä muissakin dataintegraatioratkaisuissa.

Magic Quadrant™ -raportti esittelee integraatioalustamarkkinoiden parhaimpia toimittajia ja järjestää ne heidän tekemien tulkintojensa mukaan. Raportti tarjoaa näkemyksiä palveluntarjoajien suhteellisista asemista markkinoilla sekä asettaa kehitysnäkymän integraatioalustamarkkinoiden kehitykselle. Raportin avulla ohjelmistokehitysyritykset pystyvät tarkastelemaan heidän toimintaansa sopivia iPaaS-ratkaisuja (Integration Platform as a Service), jotka tukevat sekä taktisia tavoitteita että strategista suuntaa.

Gartner (2024d) jakaa heidän tunnistamat ETL-työkalujen toimittajat neljän kategorian mukaan, jotka he ovat tunnistaneet ETL-prosesseihin liittyvillä markkinoilla: pienen markkinaraon toimijat, jotka keskittyvät pieneen asiakassegmenttiin, mutta toimivat siinä onnistuneesti; suurempien toimijoiden haastajat, jotka suoriutuvat hyvin tällä hetkellä, mutta joilta puuttuu näkemys markkinoiden tulevaisuuden suunnasta; tulevaisuuden visionäärit, jotka osoittavat ymmärrystä tulevaisuuden markkinoista tai jotka yrittävät itse vaikuttaa markkinoiden suuntaan; sekä markkinajohtajat, jotka toimivat hyvin ja ovat

hyvässä asemassa nykyhetken markkinatilanteessa ja ovat varautuneet mahdollisiin markkinoiden muutoksiin. (Gartner 2024d)

Kuva 15 esittää Gartnerin (2024d) tunnistamat ETL-työkalujen toimittajat graafisesti esitettynä, jossa toimittajat ovat sijoitettu kuvaajaan heidän näkemyksien ja arvioiden perusteella. Kuvasta 15 huomataan, kuinka ETL-työkalun suorituskyky ja ETL-työkalun toimittajan visio ohjaavat ETL-työkalun toimittajien sijoittumista kuvaajassa.



Kuva 15. Datan käsittely -työkalujen toimittajat (mukaillen Gartner 2024d)

Kuva 15 esittää Gartnerin (2024d) näkemyksen datan käsittely -työkalujen toimittajista, jotka he ovat arvioineet varteenotettavaksi ratkaisuvaihtoehdoiksi datan käsittelyprosesseihin. Datan käsittely -työkalujen toimittajien arviointiin he ovat luoneet oman arviointikriteeristönsä, joiden avulla muodostetut arviot ovat listattu erilaisten datan käsittely -työkalulta vaadittavien ominaisuuksien kautta peilaten niitä työkalujen toimittajien vahvuuksiin sekä tarkkaavaisuutta vaativiin ominaisuuksiin.

Qaiser et al. (2023) vertailevat omassa tutkimuksessaan ETL-työkalujen toiminnallisuuksia suuren datamassan käsittelyssä. Mali & Bojewar (2015) sekä Kherdekar & Metkewar (2016) lähestyvät omissa töissään ETL-työkalun valinnan

vaikutuksia tietovarastointiprosessiin ETL-työkalujen toiminnallisten ominaisuuksien kautta. Mukherjee & Kar (2017) tarkastelevat myös ETL-työkalun valinnan vaikutuksia tietovarastointiprosessiin, mutta tuovat aiheeseen datatieteen näkökulman, jossa selvitetään ETL-työkalujen markkina-arvoa ja niiden merkitystä käytännön käyttötilanteissa. Badiuzzaman Biplob et al. (2018) vertailevat ETL-työkalujen ominaisuuksia erilaisissa tilanteissa tietovarastointiprosessiin liittyen heidän kehittämässä tutkimusviitekehyksessään.

Pall & Khaira (2013) vertailevat omassa katselmuksessaan ETL-työkaluja, jotka he ovat tunnistaneet markkinajohtajiksi katselmuksen toteuttamishetkellä. Sreemathy et al. (2021) tunnistavat myös useita ETL-työkaluja, joiden pääpiirteet ja -ominaisuudet he esittelevät yleisellä tasolla omassa julkaisussaan. Thirumagal et al. (2014) lähestyvät ETL-työkaluja ja niiden toimittajia vahvuuksien ja heikkouksien näkökulmasta. He listaavat esitellyistä ETL-työkaluista ja niiden toimittajista tunnistetut ominaisuudet ja toiminnot niiden hyvien ja huonojen puolien mukaan.

Khan et al. (2024) tuovat omassa katsauksessaan ETL-työkalujen rinnalle myös muita ETL-prosessissa hyödynnettäviä menetelmiä ja tekniikoita, joiden ominaisuuksien vaikutuksia he tarkastelevat tietovarastointiprosessin kontekstissa. Hovi et al. (2009, s. 60) esittelevät omassa teoksessaan tietovarastoinnissa käytettäviä työkaluohjelmistoja ja tarkemmin ETL-työkaluja, jotka he ovat tunnistaneet tärkeimmiksi välineiksi tietovarastoinnin ETL-prosessissa. Myös Goldfedder (2020) tunnistaa useita markkinoilla olevia ETL-työkaluja ja esittelee niiden toimintakykyä vahvuuksien ja heikkouksien kautta.

Näiden esiteltyjen tutkimus- ja selvitystöiden sekä kirjallisten teoksien pohjalta luotiin kattava lista olemassa olevista ja tieteellisissä julkaisuissa tunnistetuista ETL- ja datan käsittely -työkaluista. Taulukko 3 esittää ETL-työkalujen toimittajaorganisaatiot ja niiden tarjoaman tuotteen ilman karsivaa suodatusta mahdollisista varteenotettavista vaihtoehdoista kohdeorganisaation toimintakäyttöön. Listattujen ETL-työkalujen lähteinä käytettiin aihepiiriä käsitteleviä tieteellisiä julkaisuja, jotka esiteltiin aiemmissa kappaleissa.

Taulukko 3. Tunnistetut ETL-työkaluvaihtoehdot

Toimittaja	ETL-työkalu
Ab Initio	Ab Initio
Agenty	Agenty
Amazon Web Services (AWS)	AWS Glue

Apify	Apify
CloverDx	CloverETL
DataJunction	DataJunction
Dell Boomi	Boomi
Fivetran	Fivetran
Google Cloud	Alooma
Google Cloud	Dataflow
Grepsr	Grepsr
Hevo	Hevo
Hitachi Vantara	Pentaho Data Integration
Hubdoc	Hubdoc
IBM	InfoSphere Information Server
Informatica	PowerCenter
Integrate.io	Integrate.io
Jaspersoft	Jaspersoft
Jitterbit	Jitterbit
Microsoft	Azure Data Factory
Microsoft	SQL Server Integration Services (SSIS)
MuleSoft	MuleSoft
Oracle	Oracle Data Integrator
Oracle	Oracle Warehouse Builder (OWB)
Qlik Stitch	Stitch
Qlik Talend	Talend Open Studio for Data Integration
SAP	SAP Data Services
SAS	SAS Data Integration Studio
Skyvia	Skyvia
StreamSets	StreamSets

Xtract.io	Xtract.io

Taulukossa 3 esitettyjen vaihtoehtojen sekä kuvassa 15 havainnollistettujen toimijoiden pohjalta lähdettiin luomaan listaa kohdeorganisaation toimintaympäristöön, käyttötarkoituksiin sekä liiketoimintaan sopivista ETL-työkaluista. Listaa haluttiin lisäksi täydentää integraatiopalvelualustaa – iPaaS:a tarjoavalla Frends:lla, joka on jo kohdeorganisaatiolla käytössä sisäisissä ja ulkoisissa integraatioissa sekä Microsoftin tarjoamalla Microsoft Fabric -työkalulla, joka tarjoaa pilvipohjaisia tiedonsiirto- ja datamuunnospalveluita ETL-prosesseihin.

Kohdeorganisaation käyttöön sopivista ETL-työkaluista muodostettiin lista, jossa otettiin huomioon työkalujen ominaisuudet sekä niiden käyttöön ja käyttöönottoon liittyvät huomioon otettavat yksityiskohdat kuin myös työkalujen soveltuvuus kohdeorganisaation toimintaympäristöön ja käyttötarkoituksiin. Lisäksi työkalun kustannuspuolen yksityiskohdat rajasivat vaihtoehtojen valintaa.

Vaadittujen ominaisuuksien ja yksityiskohtien perusteella kohdeorganisaation toimintaympäristön käyttötapauksiin perustuvaan vertailevaan tarkasteluun valittiin Microsoftin tarjoamat Microsoft Fabric sekä Microsoft Azure Data Factory -työkalut sekä integraatioalusta Frends (taulukko 4). Näissä valinnoissa vaikutti tarvittavien ominaisuuksien lisäksi olennaisesti kohdeorganisaation ja Microsoftin välinen yhteystyö sekä integraatioalusta Frends:n käyttö ja sen integraatioalustaosaaminen kohdeorganisaatiossa.

Taulukko 4. Vertailuun valitut ETL-työkalut

Toimittaja	ETL-työkalu
Frends	Frends iPaaS
Microsoft	Azure Data Factory
Microsoft	Microsoft Fabric

Frends toimittaa integraatioalustapalvelua, jonka niin sanottu low-code-ympäristö tarjoaa integraatiokokonaisuuden, joka mahdollistaa API-integraatioiden, API-kehityksen ja API-hallinnan kaikkiin low-code-integraatiotarpeisiin, mikä helpottaa ratkaisujen hallintaa. Alustan hajautettu integraatioarkkitehtuuri mahdollistaa datan ja järjestelmien välisen yhteyden paikallisesti on-premise-ratkaisuina tai julkisina pilvipalveluina tai näiden hybridipilvi-on-prem-versioina. Frends tarjoaa läpinäkyvyyttä ja jäljitettävyyttä

tiedonsiirtoprosesseihin ja virhetilanteisiin kokonaisvaltaisen monitorointi- ja hallintakäyttöliittymän avulla. Frends tarjoaa läpinäkyvän kustannusmallin, jossa kaikki integraatioalustapalvelut tarjotaan asiakkaalle riippumatta sen tilaustasosta, ja integraatioalusta skaalautuu tilaajan tarpeisiin ilman lisäkustannuksia. Frends:n iPaaS-ratkaisut on myös tunnustettu vuoden 2024 Gartnerin Magic Quadrant™ -tutkimuksessa (kuva 15). (Frends 2024)

Microsoftin Azure Data Factory on Azuren tarjoama ETL-pilvipalvelu palvelimettomaan tietojen integrointiin ja niiden muuntamiseen. Se tarjoaa koodittoman käyttöliittymän ETL-prosessien luomiseen, valvontaan ja niiden hallintaan. Azure Data Factoryn ETL-pilvipalvelun avulla voi suunnitella ja toteuttaa tiedonsiirtoprosesseja sekä muita dataintegraatioprosesseja, joissa yhdistyvät ETL-prosessien sisältämät lainalaisuudet raakadatan jalostuksesta tiedoksi. (Microsoft 2024a)

Microsoftin Microsoft Fabric on kattava analytiikkatyökalu organisaatioille, joka tarjoaa laajan palvelupaketin; esimerkiksi Data Lake -tallennustilaa, tietotekniikan ja tietojen integroinnin yhdistämisen, kaikki samassa paketissa. Microsoft Fabric tarjoaa palveluita datan siirtämisestä datatieteeseen, reaaliaikaiseen analytiikkaan ja liiketoimintatietojen analysointiin ja hallintaan. Microsoft Fabric yhdistää usean toimittajan palvelut yhteen pakettiin, mikä mahdollistaa integroidun, tuotevalikoimaltaan kattavan ja helppokäyttöisen palvelupaketin, joka yksinkertaistaa organisaation analytiikkatyökalut. Microsoft Fabric -alusta on rakennettu SaaS-palvelulle, mikä yksinkertaistaa integraatiomahdollisuudet helpoilla liitännöillä. (Microsoft 2024b)

Taulukko 5 esittää vertailevaan analyysiin valitut ETL-työkalut, jossa on listattu kohdeorganisaation tunnistamat ETL-työkalulta vaadittavat ominaisuudet ja muut tarpeet. Kohdeorganisaation ETL-työkalulle asettamia ominaisuuksia ja muita tarpeita tarkastellaan kyseisten työkalujen kohdalla, jossa Ominaisuus-sarake kuvaa, mitä ominaisuutta tarkastellaan ja rivillä oleva merkintä kuvaa, löytyykö kyseinen ominaisuus työkalusta. Jos rivillä kuvattu ominaisuus löytyy työkalusta, merkitään työkalun alle sarakkeeseen ”Kyllä” ja jos ominaisuus ei ole saatavilla, merkitään sarakkeeseen ”Ei”. Epäselvissä tapauksissa, joista ei ole varmaa tietoa, merkitään ”?”.

Taulukko 5. ETL-työkalujen vertaileva analyysi

Ominaisuus	Frends iPaaS	Azure Data Factory	Microsoft Fabric
Yhteyden muodostaminen SQL Serveriin	Kyllä	Kyllä	Kyllä

Kattavat funktiotoiminnot tiedon käsittely- ja siirtoprosesseihin	Kyllä	Kyllä	Kyllä
Datansiirto eri ympäristöihin	Kyllä	Kyllä	Kyllä
Tietovaraston ylläpito ja hallinta	Osittain kyllä	Osittain kyllä	Osittain kyllä
ETL-prosessien asiakaskohtainen versiohallinta	Kyllä	Ei suoranaisesti, ei ole sisäänrakennettu ominaisuus	Ei ole suunniteltu tähän tarkoitukseen
Automaattinen muutosten päivitys ja asiakaskohtaisten räätälöintien tukeminen	Kyllä	Kyllä	Ei suoranaisesti
ETL-prosessien ajosuoritusten asiakaskohtainen aikataulutus ja rinnakkaisten ETL-prosessien ajomahdollisuudet	Kyllä	Kyllä	Kyllä
Virhetilanteiden monitorointi, hallinta ja ylläpito. Tietojen tarkastuksen ja validoinnin seurantamahdollisuudet	Kyllä	Kyllä	Kyllä
Käyttöönoton helppous ja yleinen helppokäyttöisyys	Kyllä, lisäksi löytyy etukäteisosaamista	Kyllä, lisäksi löytyy etukäteisosaamista	Kyllä

	kohdeorganisaatiosta	kohdeorganisaatiosta	
Prosessien uudelleenkäytettävyys, niiden osien monistettavuus ja modulaarisuuden tukeminen	Kyllä	Kyllä	Kyllä
Hintakustannukset sekä niiden skaalautuvuus asiakasmäärien, asiakaskohtaisten räätälöintien ja asiakaskohtaisten ympäristöjen mukaan kuin myös muut käyttökustannukset	Kustannukset skaalautuvat asiakasmäärien ja asiakaskohtaisten ympäristöjen mukaan	Kustannukset skaalautuvat asiakasmäärien ja asiakaskohtaisten ympäristöjen mukaan	Kustannukset skaalautuvat asiakasmäärien ja asiakaskohtaisten ympäristöjen mukaan
Etukäteisosaaminen ETL-työkalusta	Kyllä	Kyllä	Ei
Pilvipohjainen käyttömahdollisuus sekä tietoturvallisuuden huomiointi	Kyllä	Kyllä	Kyllä
Tuki kohdeorganisaation kaikkien tuotteiden yhteiselle tietovarannolle	Kyllä	Kyllä	Kyllä
Palvelurajapintojen tuki	Kyllä	Kyllä	Kyllä
Riittävä suorituskyky ja kapasiteetti ja muut	Kyllä	Kyllä	Kyllä

tehokkuutta lisäävät tekijät			
Erilaisten datamuotojen käsittelyt sekä tietojen takaisinkirjoitus mahdollisuus tietovarastosta järjestelmään	Kyllä	Kyllä	Kyllä
SAAS-liiketoiminnan tukeminen	Kyllä	Kyllä	Kyllä

Taulukosta 5 huomataan, kuinka syvällisempään vertailuun valitut ETL-työkalut eivät varsinaisesti eroa ominaisuuksiltaan kohdeorganisaatiossa tunnistettujen ominaisuuksien kohdalta. Toisaalta työpajakeskusteluissa nousi vahvasti esiin ratkaisun kustannuspuoli, joka toimii viimeisenä, painottavana kriteerinä ratkaisuvaihtoehdon valinnalle. Hintakustannukset ja niihin liittyvät yksityiskohdat ohjaavat viime kädessä ETL-työkalun valinnassa, kun etsitään kustannustehokasta, kohdeorganisaation liiketoimintaan soveltuvaa vaihtoehtoa. Työpajakeskusteluiden perusteella ei ole kiveen hakattua hintaa, jonka ratkaisuvaihtoehto saa korkeintaan maksaa, vaan budjetti perustuu päätöshetkellä olevien tekijöiden summaan, jossa ratkaisun kokonaisuus ja sen tuoma lisäarvo vaikuttavat kustannuskysymyksiin.

6. YHTEENVETO JA PÄÄTELMÄT

Tietovarastointiin liittyy useita prosesseja, joista ETL-prosessi on tiedonsiirron pääprosessi. Sen tarkoituksena on säästää kustannuksia, aikaa ja tehostaa tiedonsiirtoprosessia lähdejärjestelmän ja tietovaraston välillä. ETL-prosesseissa käytetään erilaisia työkaluja ja tekniikoita, ja niiden sopiminen organisaation käyttötarkoituksiin ja toimintaympäristöön niiden omilla vahvuuksillaan ja heikkouksillaan on olennaisessa osassa niiden toiminnan kannalta. (Khan et al. 2024) Tässä tutkimuksessa tarkasteltiin kohdeorganisaation liiketoimintaan sopivia ETL-työkaluja; muodostettiin arviointikriteerit niiden analysoinneille ja peilattiin tunnistettuja ominaisuustarpeita kolmeen valittuun ETL-työkaluvaihtoehtoon ja niiden ominaisuuksiin.

Tutkimuksessa pyrittiin etsimään mahdollisia uusia ratkaisuja kohdeorganisaation tietovarastoinnin ETL-prosessin tiedonsiirtoprosessiin, jotka täyttäisivät niille asetetut toimintavaatimukset. Päätutkimuskysymys ”Onko olemassa soveltuvia liiketoimintatavoitteen täyttäviä ratkaisuja legacy-koodille, joilla voidaan siirtää dataa WMS-tietojärjestelmästä tietovarastoon?” ohjasi tutkimuksen suuntaa ja muodosti tutkimusasetelman tutkimukselle. Alatutkimuskysymyksen ”Onko olemassa kohdeorganisaation valintakriteereihin soveltuvia ETL-työkaluja?” avulla tutkittiin tunnistettuja ETL-työkaluja ja niiden soveltumista kohdeorganisaation tarpeisiin ja vaatimuksiin.

Ratkaisun soveltuminen liiketoimintatavoitteeseen määriteltiin kohdeorganisaation toimesta, joka kuvattiin vaatimuksien ja tarpeiden kautta, jotka ratkaisun tarvitsi täyttää ja jotka se pystyy suorittamaan toiminnoillaan. ETL-työkalulta vaaditut ylläpito-, skaalautuvuus- ja tehokkuusominaisuudet sekä käyttöönottoon ja käyttöön liittyvät kustannukset nähtiin olennaisimpina vaatimuksina, joita työkalulta edellytettiin. Nämä tarpeet esiteltiin tutkimuksen työpajakeskusteluissa tunnistettujen ETL-työkalujen valintakriteerien avulla.

Tutkimuskysymyksiin löydettiin kolme mahdollista ratkaisuvaihtoehtoa, jotka täyttivät kohdeorganisaation asettamat tarpeet ja suoriutuivat niille tunnistetuista vaatimuksista liiketoimintatavoitteisiin, toimintaympäristöön ja käyttötarkoituksiin liittyen. Tunnistettuja ratkaisuja esitetään vaihtoehtoisina ratkaisuina nykyiselle tiedonsiirtomenetelmälle, jossa jokainen ratkaisu täyttää niille asetetut tarpeet hyvällä tasolla, ja joiden ominaisuudet suoriutuvat niille määritetyistä vaatimuksista. Toisaalta hintakustannukset ja niihin liittyvät yksityiskohdat voivat rajata mahdollista ratkaisuvaihtoehtoa; kuinka

paljon ollaan valmiita sijoittamaan ratkaisuun ja sen ylläpitoon, kun on budjetti määritetty ja päätös tehty? Kaiken kaikkiaan ETL-työkalun valinta pitäisi pohjautua organisaation liiketoimintavaatimuksien mukaan, jotka määrittelevät viime kädessä oikean ratkaisun valinnan.

6.1 Tutkimuksen arviointi

Laadullista tutkimusta voidaan arvioida kirjallisuudessa esitettyjen arvosteluperusteiden kautta (muun muassa Tuomi & Sarajärvi 2018; Shenton 2004). Laadullista tutkimusta arvioitaessa sen uskottavuus, siirrettävyys, luotettavuus sekä vahvistettavuus ovat osatekijöitä (Tuomi & Sarajärvi 2018; Shenton 2004), joiden pohjalta voidaan tarkastella tutkimuksen onnistuneisuutta.

Tutkimuksen uskottavuus osoittaa tutkimuksen luotettavuutta sen tuloksien ja päätelmien pohjalta sekä kuinka tutkimus nähdään totuutena, onko se sovellettavissa ja kuinka tutkijan reflektiivisyys on otettu huomioon tutkimusta tehdessä. (Tuomi & Sarajärvi 2018) Tutkimuksen tutkimusasetelma ja viitekehys muodostuivat tapaustutkimuksen ympärille, jossa tiedonkeruumenetelminä hyödynnettiin kirjallisuuskatsausta ja dokumenttianalyysia sekä työpajakeskusteluita, joissa osallistujat olivat aihepiirin ja kohdeorganisaation toimintatavat syvällisesti tuntevia asiantuntijoita, joiden näkökulmat ja mielipiteet muodostivat kohdeorganisaation tarpeet ja vaatimukset vaihtoehtoiselle ratkaisumenetelmälle. Tutkijan objektiivisyyttä tarkastellessa voidaan todeta, että tutkija toimii kohdeorganisaation integraatiotiimissä, joten tutkijan reflektiivisyyttä tarkastellessa aihepiirin ennakkokäsitykset ja -uskomukset ovat ainakin alitajuisesti vaikuttaneet tutkimusprosessiin.

Tutkimuksen siirrettävyys kuvaa, kuinka tutkimuksen tulokset tai löydökset ovat siirrettävissä tutkimuskontekstin ulkopuoliseen vastaavaan kontekstiin. Tässä ehtona kuitenkin, että vastaavan kontekstin ympäristö täyttää samankaltaiset kriteerit. (Tuomi & Sarajärvi 2018) Tutkimuksen viitekehyksen dokumentointi sekä kirjallisuuskatsauksessa ja dokumenttianalyysissa hyödynnettävien menetelmien esittelyt ja niiden lähteet mahdollistavat tutkimuksen siirtämisen toiseen vastaavaan ympäristöön. Tutkimuksen siirrettävyyttä heikentää tutkimuksessa yhtenä tiedonkeruumenetelmänä hyödynnetyt työpajakeskustelut, jotka järjestettiin vapaamuotoisina *brainstorm*-keskusteluina, ja joissa muodostuneet tunnistetut tarpeet perustuivat osallistujien näkemyksiin, mitä ominaisuuksia kohdeorganisaation tiedonsiirtoprosessin uusi vaihtoehtoinen ratkaisu vaatii.

Tutkimuksen luotettavuus kuvaa tutkimuksen tutkimusprosessin toteutumista, kuinka tutkimus vastaa siihen, mitä sen tutkimusasetelmassa kuvataan. Tutkimuksen luotettavuus kulkee vahvasti tutkimuksen uskottavuuden rinnalla. (Tuomi & Sarajärvi 2018) Tutkimusprosessin perusteellinen esittely ja tutkimusasetelman kattava kuvaaminen edistävät tutkimuksen luotettavuutta ja tukevat tutkimuksen uskottavuutta, mikä kuvattiin tutkimuksen uskottavuutta tarkastellessa.

Tutkimuksen vahvistettavuus viittaa siihen, kuinka tutkimuksen tulokset ja löydökset, käytetyt aineistot ja julkaisut, tulkinnat sekä mahdolliset suositukset voidaan varmistaa ja vahvistaa ulkopuolisen henkilön toimesta. (Tuomi & Sarajärvi 2018) Tutkimuksen kattava lähdeaineiston kuvaaminen edesauttaa tutkimuksen vahvistettavuutta sekä tutkijan objektiivinen rooli tulkinnoissa ja löydöksissä. Toisaalta tutkimuksen kohdeorganisaation tarpeiden tunnistamisessa hyödynnettyjen työpajakeskusteluiden tulokset eivät ole ulkopuolisten henkilöiden saatavilla muuten kuin tutkimuksessa kuvattujen ominaisuustarpeiden kautta, mikä vaikeuttaa tutkimuksen varmistusta ulkopuolisen toimijan näkökulmasta ja täten heikentää tutkimuksen vahvistettavuutta.

Tutkimuksen selvitystyö tehtiin analysoimalla kirjallisuuskatsauksessa esiin tulleita aineistoja. Raporteista, julkaisuista ja artikkeleista sekä erilaisista selvitys- ja tutkimustöistä löytyneitä menetelmiä, joita voidaan hyödyntää ETL-prosessin datansiirrossa, tutkittiin ja arvioitiin niiden soveltuvuutta kohdeorganisaation toimintaan. Tutkimuksen vertaileva analysointi perustui hyödynnettävien raporttien, julkaisujen ja artikkelien sekä selvitys- ja tutkimustöiden tietoon kuin myös ratkaisuvaihtojen kotisivujen dataan, joita yhdistelemällä saatiin koottua tulokset yhteen taulukkoon, jossa ratkaisuvaihtoehtoja tarkasteltiin kohdeympäristön näkökulmasta heidän vaatimien arviointikriteerien pohjalta ja heidän käyttötapauksissansa esiintyvien tarvittavien ominaisuuksien kautta.

Ratkaisuvaihtoehtoja ei testattu oikeassa käytössä, erilaisissa käyttötapauksissa oikealla datalla, vaan ratkaisuvaihtoehtojen arviointi perustui julkaisuihin ja lähteisiin, jotka oli huomioitu kirjallisuuskatsauksessa sekä tapaustutkimuksen vertailevassa analysoinnissa. Ratkaisuvaihtoehdoilta tarvittavia erilaisia ominaisuuksia tarkasteltiin teorian pohjalta. Selvitystyön tulokset eivät kuvaa absoluuttista totuutta, mutta ne ovat suuntaa antavia teoria-aineiston pohjalta muodostettuja kohdeorganisaation tapaukseen yhdistettyjä tapauskohtaisia totuuksia, joita voidaan hyödyntää päätöksenteossa, jos kohdeorganisaatio päättää uudistaa WMS-tietojärjestelmän ETL-prosessissa käytettävän tiedonsiirtomenetelmän.

Tutkimuksen uutuusarvo arvioitaessa huomattiin, kuinka ennestään oli jo olemassa tutkimus- sekä selvitystöitä ETL-prosessien ETL-työkaluista, niitä hyödynnettävistä menetelmistä, teknologioista sekä erilaisista ratkaisuvaihtoehdoista ETL-prosesseihin, joten tutkimuksen uutuusarvoa mitattaessa, tämä tutkimus ei luo uutta tietoa ETL-prosessien datansiirtoratkaisuista. Toisaalta tämän tutkimuksen yksilöllinen ja uniikki tutkimusasetelma sekä tarkastelun kohteena ollut tapaus luo uutta tietoa yhdistellessä ja vertaillessa mahdollisia vaihtoehtoja tutkimuksen viitekehyksessä kohdeorganisaation tarpeisiin. Tutkimuksessa hyödynnettiin aikaisempien tutkimuksien löydöksiä ja selvityksiä, joita tarkasteltiin, analysoitiin ja vertailtiin kohdeorganisaation tarpeiden ja vaatimuksien sekä toimintaympäristön ja sen käyttötarkoituksien kautta.

Tutkimus toimi alustavana selvitystyönä mahdollisille ratkaisuille, joita voidaan lähteä tutkimaan ja testaamaan lisää tämän tutkimuksen pohjalta. Esimerkiksi konkreettisten käyttötapausten testaamiset oikealla datalla oikeassa toimintaympäristössä toisivat uuden näkökulman tutkimukseen. Testauksista esiinnoussut tieto mahdollistaisi uudenlaisen mahdollisuuden tarkastella ETL-työkalujen ominaisuuksien peilautuvuutta kohdeorganisaation toimintaympäristöön sekä antaisi ETL-työkalun käytöstä ja käyttöönotosta konkreettista tietoa.

6.2 Jatkotutkimusmahdollisuudet

Palmer (2024) huomioi omassa teoksessaan tulevaisuuden vääjäämättömät muutokset tietotekniikan alalla ja sen teknologisessa kehittymisessä. Palmer (2024) silti uskoo ETL-prosessin ja ETL-työkalujen toimivan vankkana kulmakivenä myös tulevaisuuden datan siirto- ja käsittelyprosesseissa ja perustelee oman näkemyksensä ETL-prosessin kokonaisvaltaisuuteen liittyen; ETL-prosessin kokonaisvaltaisuus datan käsittelyssä ja sen siirrossa mahdollistaa ETL-prosessien hyödyntämisen myös tulevaisuudessa, kun tarvitsee poimia dataa eri lähteistä, yhdistää, puhdistaa ja muuntaa dataa tiedoksi ja lopulta ladata tieto käyttökohteeseen käyttäjille hyödynnettäväksi päätöksentekoon. (Palmer 2024)

Palmerin (2024) tuoreita huomioita kompaten voidaan todeta, kuinka ETL-prosessi nähdään tulevaisuudessakin edelleen kokonaisvaltaisena tiedon käsittely- ja siirtoprosessina, joka suoriutuu sille asetetuista tehtävistä. ETL-työkalujen jatkuvasti kehittyessä niiden käyttö ETL-prosessien vaiheissa tullee tehostamaan kokonaisvaltaisesti koko prosessin toimintaa.

Khan et al. (2024) esittelevät tulevaisuuden tutkimussuuntia, jotka he näkevät ETL-aihepiireihin liittyvien tutkimuksien keskittyvän lähitulevaisuudessa. Esimerkiksi ETL-

automaation kehittäminen yhdessä koneoppimisen ja tekoälyn lainalaisuuksien kanssa sekä ETL-prosessien tietoturvaan ja yksityisyyteen liittyvien näkökohtien vahvempi huomioiminen prosessien kehityksessä ja hallinnoimisessa. He uskovat tulevaisuuden ETL-prosessien ja -työkalujen parhaiden käytäntöjen edistävän tiedonsiirtoprosessien kehittymistä, mutta tunnistavat tulevaisuuden ETL-prosessien skaalautuvuus- ja tehokkuushaasteet kasvavien datamäärien yhteydessä. (Khan et al. 2024) Nämä esitetyt tutkimussuunnat tukevat ETL-prosessien ja sen toimintojen jatkuvaa kehittämistä ja sopeuttamista tietovarastointiin. Kehitys mahdollistaa ETL-prosessien ja -työkalujen tehokkuuden ja tietoturvan parantamisen sekä ETL-prosessin joustavan hyödyntämisen muuttuvissa tietovarastointiprosessien ympäristöissä.

Yleisesti ETL-aihepiirin näkökulmasta katsottuna yhtenä näkemyksenä tulevaisuuden tutkimuksille voidaan esittää tekoälyn hyödyntämistä ETL-prosesseissa ja siinä hyödynnettävissä menetelmissä ja työkaluissa, jotka myös Khan et al. (2024) ovat huomioineet. Tekoälyn luomat mahdollisuudet prosessivaiheiden automatisoinnissa ja prosessien suoritusajojen ja suorituskyvyn optimoinneissa tai virheiden automaattisessa tunnistamisessa ja niiden korjaamisessa ovat esimerkkejä, joiden avulla tekoäly luo mahdollisuuksia ETL-prosessikentän tulevaisuuden kehittymiselle.

Kohdeorganisaation näkökulmasta tarkasteltuna voitaisiin tutkia kolmen tunnistetun vaihtoehtoisen ratkaisun konkreettista käyttöä todellisessa toiminnassa, jossa huomattaisiin, kuinka menetelmät oikeasti taipuvat kohdeorganisaation tarpeisiin konkretian tasolla. Toimintaympäristön ja käyttötarkoitusten simulointi mahdollistaisi uusien tutkimusasetelmien luomisen, joiden avulla saataisiin uusia näkemyksiä ratkaisujen soveltuvuudesta kohdeorganisaation käyttöön.

LÄHTEET

Alvesson, M. & Kärreman, D. (2001). Odd Couple: Making Sense of the Curious Concept of Knowledge Management. Journal of management studies. 38 (7), 995–1018.

Ariyachandra, T. & Watson, H. (2010). Key organizational factors in data warehouse architecture selection. DECISION SUPPORT SYSTEMS. 49 (2), 200–212.

Aveyard, H. (2019). Doing a literature review in health and social care : a practical guide. Fourth edition. London, England ; McGraw Hill Education/Open University Press.

Badiuzzaman Biplob, M., Sheraji, G. A. & Khan, S. I. (2018). Comparison of Different Extraction Transformation and Loading Tools for Data Warehousing. 2018 International Conference on Innovations in Science, Engineering and Technology (ICISET). 2018 IEEE. s. 262–267.

Barrett, D., Clarke, L., Tarr, P. & Wise, A. (1996). A framework for event-based software integration. ACM transactions on software engineering and methodology. 5 (4), 378–421.

Bass, L. Clements, P. & Kazman, R. (2022). Software Architecture in Practice, 4th Edition. Addison-Wesley Professional.

Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. arXiv.org.

Bernard, S. A. (2005). An introduction to enterprise architecture. 2nd ed. Bloomington (IN): AuthorHouse.

Blazic, G., Poscic, P. & Jaksic, D. (2017). Data warehouse architecture classification. 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). 2017 Croatian Society MIPRO. s. 1491–1495.

Bowen, G. A. (2009). Document Analysis as a Qualitative Research Method. Qualitative research journal. 9 (2), 27–40.

Butterfield, A., Ngondi, G. E. & Kerr, A. (2016). A dictionary of computer science. Seventh edition. Oxford, England: Oxford University Press.

Choo, C. W. (1997). Organizations as "Information-use Systems": A Process Model of Information Management. PrimaVera Working Paper 97-17, 1–18.

Dooley, J. F. (2017). Software Development, Design and Coding With Patterns, Debugging, Unit Testing, and Refactoring. 2nd ed. 2017. Berkeley, CA: Apress.

Doomen, P. (2023). Introduction to SparxSystems Enterprise Architect Documenting Enterprise Architecture in the Most Affordable Enterprise Architecture Suite. 1st ed. 2023. Berkeley, CA: Apress.

El Akkaoui, Z. Zimanyi, E. Mazon, J-N. & Trujillo, J. (2011). A model-driven framework for ETL process development. International Conference on Information and Knowledge Management, Proceedings. 2011 ACM. s. 45–52.

Eriksson, P. & Koistinen, K. (2005). Monenlainen tapaustutkimus. Helsinki: Kuluttajatutkimuskeskus.

Feathers, M. C. (2005). Working effectively with legacy code. 1st edition. Place of publication not identified: Prentice Hall Professional Technical Reference.

Finto (2024). Suomalainen asiasanasto- ja ontologiapalvelu. Tiedolla johtaminen. Verkkosivu. Saatavissa (viitattu 16.1.2024.): http://urn.fi/URN:NBN:fi:au:tt:t90.

Fowler, M. (2003). Patterns of enterprise application architecture. 1st edition. Boston: Addison-Wesley.

Frends. Frends Integration Platform. (2024). Verkkosivu. Saatavissa (viitattu 16.4.2024): https://frends.com/.

Gao, F., Xiao, G. & Simpson, T.W. (2009). Module-scale-based product platform planning. Research in engineering design. 20 (2), 129–141.

Gartner. Information Technology Glossary. Integrated Software. (2024a). Verkkosivu (viitattu 17.2.2024). Saatavissa: https://www.gartner.com/en/information-technology/glossary/integrated-software.

Gartner. Information Technology Glossary. Integration. (2024b). Verkkosivu (viitattu 17.02.2024). Saatavissa: https://www.gartner.com/en/information-technology/glossary/integration.

Gartner. Information Technology Glossary. Application Integration. (2024c). Verkkosivu. Saatavissa (viitattu 17.2.2024): https://www.gartner.com/en/information-technology/glossary/application-integration.

Gartner. Information Technology. Gartner Research. Gartner Magic Quadrant for Data Integration Tools. (2024d). Verkkosivu (viitattu 1.3.2024). Ei saatavissa ilman tilausta.

Giordano, A. (2011). Data integration blueprint and modeling techniques for a scalable and sustainable architecture. 1st edition. Upper Saddle River, N.J: IBM Press/Pearson.

Goldfedder, J. (2020). Building a Data Integration Team: Skills, Requirements, and Solutions for Designing Integrations. 1st ed. Berkeley, CA: Apress L. P.

Greener, I. (2011). Designing social research a guide for the bewildered. Los Angeles, [Calif.]; SAGE.

Gulledge, T. (2006). What is integration? Industrial Management & Data Systems. 106 (1), 5–20.

Gurung, G., Shah, R. & Jaiswal, D. P. (2020). Software Development Life Cycle Models-A Comparative Study. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 30–37.

Haikala, I. & Märijärvi, J. (2006). Ohjelmistotuotanto. 11. p. Helsinki: Talentum.

Hiltunen, T. (2022). Versionhallinnan kehittäminen Big Dataan liittyvässä ETL kehitys- ja ylläpitotyössä.

Hirsjärvi, S., Remes, P. & Sajavaara, P. (2008). Tutki ja kirjoita. 13.-14. p. Helsinki: Kirjayhtymä.

Hohpe, G. & Woolf, B. (2003). Enterprise integration patterns designing, building, and deploying messaging solutions. 1st edition. Boston: Addison-Wesley.

Hou, L., Zhao, S., Li, X., Chatzimisios, P. & Zheng, K. (2016). Design and implementation of application programming interface for Internet of things cloud: DESIGN, IMPLEMENTATION OF API FOR IOT CLOUD. International journal of network management. 27e1936-.

Hovi, A., Hervonen, H. & Koistinen, H. (2009). Tietovarastot ja Business Intelligence. Jyväskylä: WSOYpro.

Hussain, S. M. Bhatti, S.N. & Ur Rasool, M. F. (2017). Legacy system and ways of its evolution. 2017 International Conference on Communication Technologies (ComTech). 2017 IEEE. s. 56–59.

Inmon, W. H. (2005). Building the Data Warehouse. 4th ed. Hoboken: Wiley.

Jacobson, D., Brail, G. & Woods, D. (2011). APIs: a Strategy Guide. First edition. Farnham: O'Reilly.

Jung, J., Choi, I. & Song, M. (2007). An integration architecture for knowledge management systems and business process management systems. Computers in industry. 58 (1), 21–34.

Kaisler, S. H., Armour, F. & Valivullah, M. (2005). "Enterprise Architecting: Critical Problems," in Proceedings of the 38th Annual Hawaii International Conference on System Sciences. 2005 IEEE. s. 224b–224b.

Kallinen, T. & Kinnunen, T. Etnografia. Teoksessa Jaana Vuori (toim.) Laadullisen tutkimuksen verkkokäsikirja. Tampere: Yhteiskuntatieteellinen tietoarkisto (ylläpitäjä ja tuottaja). Saatavissa (viitattu 4.1.2024): https://www.fsd.tuni.fi/fi/palvelut/menetelmaopetus/kvali/tutkimusasetelma/tutkimusasetelman-rakentaminen/.

Ketokivi, M. & Mantere, S. (2010). TWO STRATEGIES FOR INDUCTIVE REASONING IN ORGANIZATIONAL RESEARCH. The Academy of Management review. 35 (2), 315–333.

Khan, B., Jan, S., Khan, W. & Chughtai, M. I. (2024). An Overview of ETL Techniques, Tools, Processes and Evaluations in Data Warehousing. Journal on Big Data (Henderson, Nev.). 6 (1), 1–20.

Kherdekar, V. A. & Metkewar, P. S. (2016). A technical comprehensive survey of ETL tools. International journal of applied engineering research. 11 (4), 2557–2559.

Kimball, R. & Caserta, J. (2011). The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. Wiley-Blackwell.

Kimball, R., Ross, M., Thornthwaite, W., Mundy, J. & Becker, B. (2008). The data warehouse lifecycle toolkit. Second edition. Indianapolis, Ind: Wiley.

Koskimies, K. & Mikkonen, T. (2005). Ohjelmistoarkkitehtuurit. Helsinki: Talentum.

Krief, M. (2022). Learning DevOps - Second Edition. Packt Publishing.

Laihonen, H., Hannula, M., Helander, N., Ilvonen, I., Jussila, J., Kukko, M., Kärkkäinen, H., Lönnqvist, A., Myllärniemi, J., Pekkola, S., Virtanen, P., Vuori, V. & Yliniemi, T. (2013). Tietojohtaminen. Tampereen teknillinen yliopisto, Tiedonhallinnan ja logistiikan laitos.

Land, R. & Crnkovic, I. (2003). Software systems integration and architectural analysis - a case study. IEEE International Conference on Software Maintenance, ICSM2003. 2003 IEEE. s. 338–347.

Lankhorst, M. M. (2004). Enterprise architecture modelling—the issue of integration. Advanced engineering informatics. 18 (4), 205–216.

Laudon, K. C. & Laudon, J. P. (2007). Management information systems : managing the digital firm. 10. ed. Upper Saddle River, NJ: Pearson/Prentice Hall.

Leanware. (2024a). Leanware - Sisäinen dokumentaatio. Viitattu 12.1.2024.

Leanware. (2024b). Leanware tuotteet – Vie varastoinnin ja tuotannon tehokkuus uudelle tasolle. Verkkosivu. Saatavissa (viitattu 12.1.2024): https://leanware.fi/tuotteet/.

Leanware. (2024c). Leanware - Sisäinen dokumentaatio. Viitattu 20.2.2024.

Mali, N. & Bojewar, S. (2015). A Survey of ETL Tools. International Journal of Computer Techniques. 2 (5), 20–26.

Matharu, G. S., Signh, H. Mishra, A & Upadhyay, P. (2015). Empirical Study of Agile Software Development Methodologies: A Comparative Analysis. Software engineering notes. 40 (1), 1–6.

Martin, R. C. & Feathers, M. C. (2009). Clean code : a handbook of agile software craftsmanship. Upper Saddle River, N.J: Prentice Hall.

Meng, M., Steinhardt, S. & Schubert, A. (2018). Application programming interface documentation: What do software developers want? Journal of technical writing and communication. 48 (3), 295–330.

Microsoft. (2024a). What is Azure Data Factory?. Verkkosivu. Saatavissa (viitattu 16.4.2024): https://learn.microsoft.com/en-us/azure/data-factory/introduction.

Microsoft. (2024b). What is Microsoft Fabric?. Verkkosivu. Saatavissa (viitattu 16.4.2024): https://learn.microsoft.com/en-us/fabric/get-started/microsoft-fabric-overview.

Mishra, A. & Dubey, D. (2013). A comparative study of different software development life cycle models in different scenarios. International Journal of Advance research in computer science and management studies, 1(5).

Moilanen, J., Niinioja, M., Seppänen, M. & Honkanen, M. (2018). API-talous 101. Helsinki: Alma Talent.

Mufioz, L., Mazon, J. & Trujillo, J. (2011). ETL Process Modeling Conceptual for Data Warehouses: A Systematic Mapping Study. Revista IEEE América Latina. 9 (3), 358–363.

Mukherjee, R. & Kar, P. (2017). A Comparative Review of Data Warehousing ETL Tools with New Trends and Industry Insight. 2017 IEEE 7th International Advance Computing Conference (IACC). 2017 IEEE. s. 943–948.

Mylopoulos, J., Gal, A., Kontogiannis, K. & Stanley, M. (1996). “A generic integration architecture for cooperative information systems,” in Proceedings - 1st IFCIS International Conference on Cooperative Information Systems, CoopIS 1996. 1996 IEEE. pp. 208–217.

Narang, P. & Mittal, P. (2022). Software Development Methodologies: Trending from Traditional to DOSE - An Empirical Study. 2022 IEEE Delhi Section Conference (DELCON). 2022 IEEE. s. 1–6.

Pakkanen, J., Lehtonen, T. & Juuti, T. (2020). Partitioning Types in Product Modularisation. 2020 Cambridge University Press.

Pall, A. S. & Khaira, J. S. (2013). A comparative Review of Extraction, Transformation and Loading Tools. Database Systems Journal. IV (2), 42–51.

Palmer, M. (2024). Understanding ETL. O’Reilly Media, Inc.

Patel, M. & Patel, D. (2020). Progressive Growth of ETL Tools: A Literature Review of Past to Equip Future. s. 389–398.

Ponniah, P. (2010). Data warehousing fundamentals for IT professionals. 2nd ed. Hoboken, N.J: John Wiley & Sons.

Pulkkinen, A. (2007). Product configuration in projecting company: the meeting of configurable product families and sales-delivery process. Tampere University of Technology.

Qaiser, A., Farooq, M. U., Nabeel Mustafa, S. M. & Abrar, N. (2023). Comparative Analysis of ETL Tools in Big Data Analytics. Pakistan Journal of Engineering & Technology. 6(1), 7–12.

Rainardi, Vincent. (2008). Building a Data Warehouse With Examples in SQL Server. 1st ed. 2008. Berkeley, CA: Apress.

Richards, M. (2015). Software architecture patterns : understanding common architecture patterns and when to use them. First edition. Sebastopol, CA: O’Reilly Media.

Royce, W. W. (1970). Managing the Development of Large Software Systems. In: Proceedings of the IEEE WESCON Conference (1970), 1-9.

Rood, M. A. (1994). ‘Enterprise architecture: definition, content, and utility’, in Proceedings of 3rd IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises. 1994 IEEE Comput. Soc. Press. s. 106–111.

Runeson, P. & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. Empirical software engineering : an international journal. 14 (2), 131–164.

Saunders, M. N. K., Thornhill, A. & Lewis, P. (2019). Research methods for business students. Eighth edition. Harlow: Pearson.

Shenton, A. K. (2004). Strategies for ensuring trustworthiness in qualitative research projects. Education for information. 22 (2), 63–75.

Sreemathy, J., Brindha, R., Selva Nagalakshmi, M., Suvekha, N., Karthick Ragul, N. & Praveennandha, M. (2021). Overview of ETL Tools and Talend-Data Integration. 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS). 2021. Piscataway: IEEE. s. 1650–1654.

Thirumagal, R., Suganthy, R., Mahima, S. & Kesavaraj, G. (2014). ETL Tools in Data Mining. Internal Journal of Research in Computer Applications and Robotics. 2 (1), 62–69.

Tuomi, J. (2007). Tutki ja lue : johdatus tieteellisen tekstin ymmärtämiseen. Helsinki: Tammi.

Tuomi, J. & Sarajärvi, A. (2018). Laadullinen tutkimus ja sisällönanalyysi. Uudistettu laitos. Helsinki: Tammi.

Turajlić, N., Petrović, M. and Vučković, M. (2014). Analysis of ETL process development approaches: some open issues. Proceedings of SYMORG, 14, s.45-51.

Ulrich, K. (1995). The role of product architecture in the manufacturing firm. Research policy. 24 (3), 419–440.

Wixom, B. H. & Watson, H. J. (2001). An Empirical Investigation of the Factors Affecting Data Warehousing Success. MIS quarterly. 25 (1), 17–41.

Wojciechowski, A. (2011). E-ETL: framework for managing evolving etl processes. Advances in Intelligent Systems and Computing. 2011 ACM. s. 59–66.

Xavier, C. & Moreira, F. (2013). Agile ETL. Procedia technology. 381–387.

Yin, R. K. (2018). Case study research and applications : design and methods. Sixth edition. Los Angeles: SAGE.

Zode, M. (2007). The Evolution of ETL – From Hand-coded ETL to Tool-based ETL. Cognizant Technology Solutions. 1–13.