LAB University of Applied Sciences

Punnitusdatan käsittely ja visualisointi

LAB-ammattikorkeakoulu
Insinööri (AMK) Tieto- ja viestintätekniikka
2022
Anni Karja

Tiivistelmä

Tekijä(t)	Julkaisun laji	Valmistumisaika
Anni Karja	Opinnäytetyö, AMK	2022
	Sivumäärä 32	

Työn nimi

Punnitusdatan käsittely ja visualisointi

Tutkinto ja koulutusala

Insinööri (AMK) Tieto- ja viestintätekniikka

Toimeksiantajaorganisaatio

Lahti Precision Oy

Tiivistelmä

Opinnäytetyössä oli tavoitteena kartoittaa konseptitasolla, miten ja millaisilla resursseilla voitaisiin toteuttaa punnitusdatan visualisointia. Visualisoitu punnitusdata halutaan ottaa käyttöön työn toimeksiantajan digitaalisessa punnituspalvelu mScalesissa. Tavoitteena oli löytää vaihtoehtoja ja valmiita ratkaisuja toteuttaa kokonaisvaltainen ominaisuus, joka kattaa datan käsittelyn, varastoinnin ja visualisoinnin. Ominaisuuden haluttiin olevan helposti ylläpidettävä ja automatisoitava.

Työssä tarkasteltiin yleisluontoisesti dataa ja siihen liittyviä prosessointimahdollisuuksia. Datan käsittelyyn liittyviä ETL- ja ELT-prosesseja ja visualisointiin suunnattuja työkaluja vertailtiin ja tuloksien perusteella valittiin datan käsittelyyn AWS Glue DataBrew ja visualisointiin Chart.js-kirjasto. Olemassa olevaan dataa tutkittiin, ja siitä löydettiin kolme visualisoitavaa ilmiötä.

Työkalujen soveltuvuutta toteutukseen testattiin luomalla yksinkertainen malliratkaisu visualisoinnista. Työkalujen suoriutumista ja soveltuvuutta ominaisuuden luontiin arvioitiin annettujen kriteerien perusteella.

Ominaisuuden toteutuksen testaamisen tuloksena oli toimiva ja selkeä visualisointikomponentti. Työn tuloksien perusteella voitiin todentaa, että valitut työkalut ja ratkaisut soveltuivat halutusti ominaisuuden kehittämiseen. Seuraavaksi visualisoinnin ominaisuutta voidaan arvioida yrityksen sisäisesti ja määritellä seuraavat vaiheet sen kehityksessä.

Asiasanat

ETL, AWS Glue DataBrew, datan visualisointi, Chart.js

Abstract

Author(s)	Type of Publication	Published
Anni Karja	Thesis, UAS	2022
	Number of Pages 32	

Title of Publication

Procession and visualization of weighing data

Degree, Field of Study

Bachelor of Engineering, Information Technology

Organisation of the client

Lahti Precision Oy

Abstract

The aim of this thesis was to examine possibilities and resources needed to implement a data visualization feature for the client organisation's digital weighing service mScales. The feature was expected to be a comprehensive solution including data processing, storing, and visualizing. Tools to be chosen for this feature were required to ease management and automatization.

Different ETL and ELT processes were compared as well as multiple JavaScript visualization libraries. Based on the results of the comparisons AWS Glue DataBrew and Chart.js were selected to support data processing and visualization.

Chosen tools were evaluated by implementing a simple visualization component and a data processing pipeline. To support the testing phase, three different phenomena were chosen to be visualized. Visualization component was used to determine how well the selected tools served the purpose of this thesis.

The implemented component and data processing pipe was functional and supported the given criteria for the feature. Based on the results the selected tools were suited for the needs of the development and they could be used in developing the feature further. Results of this thesis can be used to define the next steps for future development with the intent of taking this feature to production use.

Keywords

ETL, AWS Glue DataBrew, data visualization, Chart.js

Sisällys

1 Johdanto ... 1
2 mScales ... 2
3 Data ... 4
3.1 Määritelmä ... 4
3.2 Big data ... 5
3.3 Datan käsittely ... 6
3.3.1 Käsittelyprosessit ... 6
3.3.2 ETL- ja ELT-työkalujen tarkastelu ... 8
3.4 Datan visualisointi ... 11
3.4.1 Visualisointi ja informaatiomuotoilu ... 11
3.4.2 Visualisoinnin työkalut ... 13
4 Suunnitteluvaihe ... 15
4.1 Visualisoinnin suunnittelu ... 15
4.2 Työkalun valinta ... 17
4.3 Datan yksinkertaistaminen ... 20
5 Toteutusvaihe ... 24
5.1 Ominaisuuden luonti ... 24
5.2 Käyttöönotto ... 27
6 Yhteenveto ... 29
Lähteet ... 30

1 Johdanto

Datalla on tänä päivänä hyvin merkittävä rooli. Se on oleellinen osa webkehitystä sekä verkkosivuja ja -sovelluksia, mutta sen vaikutusvalta ulottuu monialaisesti myös muihin toimintoihin, esimerkiksi liiketoimintaan. Datan avulla voidaan tehdä valistuneempia päätöksiä, parantaa suunnitelmallisuutta, tehostaa toimintaa ja tunnistaa ongelmakohtia paremmin.

Tämän opinnäytetyön toimeksiantajana on Lahti Precision Oy, joka on punnituksen ja annostelun kokonaisratkaisuihin ja -palveluihin erikoistunut teollisuuden yritys. Lahtelainen yritys toimii yhteistyössä eri teollisuuden alojen kanssa, joissa punnitseminen on osa toimintaprosesseja. Lahti Precision Oy toimii niin Pohjoismaissa kuin myös kansainvälisesti. Yrityksen palveluihin kuuluu digitaalinen ja pilvipohjainen punnituspalvelu mScales, johon tämä opinnäytetyö kohdistuu. Lahti Precision Oy:llä on pitkä historia, ja sen entinen emoyhtiö perustettiin jo vuonna 1908. (Lahti Precision Oy.) Vuonna 2021 Lahti Precision Oy:n liikevaihto oli noin 12,9 miljoonaa, ja se työllisti yhteensä 79 ihmistä. (Kauppalehti 2022).

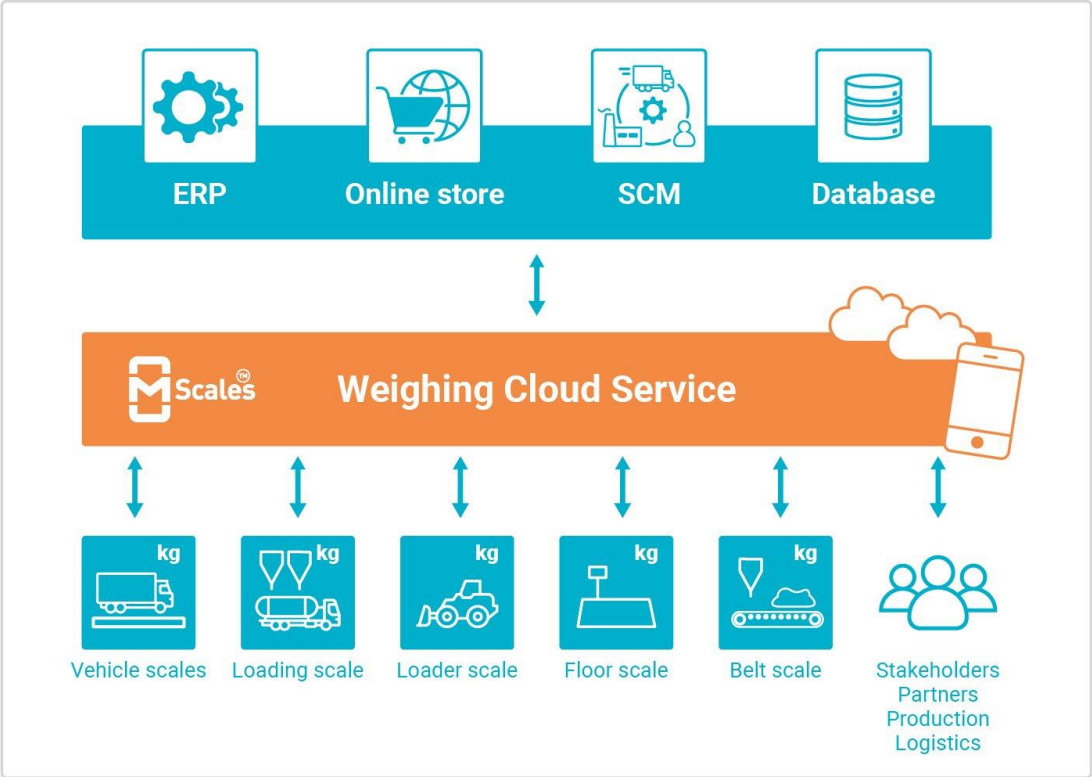
Opinnäytetyön tavoitteena on konseptitasolla suunnitella punnitusdatan visualisointia punnituspalvelu mScalesissa. Ominaisuuden taustalla on tarve tutkia, millaisin resurssein ja miten ominaisuus voitaisiin toteuttaa mahdollisimman ylläpidettävästi ja kevyesti. Toteutusta varten halutaan kartoittaa valmiita ratkaisuja ja työkaluja toteuttaa ominaisuus ja todentaa näiden toimivuutta testaamalla sitä visualisointien toteutuksessa. Ominaisuuden halutaan pitävän sisällään datan käsittelyn visualisointiin soveltuvaksi, datan varastoinnin sekä visualisoinnin.

Datan käsittelyä varten varten halutaan tutkia eri vaihtoehtoja ottaa käyttöön erillinen tietovarasto tai dataputki. Valitun ratkaisun kautta dataa halutaan käsitellä ja yksinkertaistaa visualisointia varten erillään muista tietovarastoista. Datan visualisointia varten täytyy määrittää, millaisia ilmiöitä tarkalleen halutaan esittää visualisoituna. Ilmiöiden tunnistamisen kautta voidaan tunnistaa vaadittava data ja valita kehitysympäristöön sopiva työkalu visualisointia varten.

Tämän työn teoriaosuudessa perehdytään tarkemmin punnituspalvelu mScalesiin sekä dataan, sen ominaisuuksiin ja merkitykseen ja siihen, miksi data ylipäätänsä on tärkeää. Datan käsittelyn prosesseja sekä datan visualisointia ohjaavia periaatteita tutkitaan ja vertaillaan halutun ominaisuuden näkökulmasta. Opinnäytetyön käytännön osuus on jaettu suunnittelu- ja toteutusvaiheeseen, joista ensimmäisessä keskitytään visualisointiin, työkalujen valintaan ja datan yksinkertaistamiseen. Toteutusvaiheessa otetaan käyttöön valitut työkalut sekä luodaan datan visualisoinnista jatkokehitettävä, komponenttipohjainen ratkaisu.

2 mScales

mScales on Lahti Precision Oy:n tuottama digitaalinen punnituspalvelu. Pilvipohjainen palvelu on tarkoitettu modernisoimaan ja yhtenäistämään punnitusjärjestelmiä, sillä se yhdistää liiketoimintajärjestelmät, vaa'at ja punnitustiedot yhdeksi kokonaisuudeksi. mScalesin avulla punnituksiin ja materiaalitapahtumiin liittyvä data on lähes reaaliaikaisesti liiketoimintajärjestelmien ja muiden raportointityökalujen käytettävissä. Dataa voidaan jakaa kaikille sidosryhmille ja muille punnitusprosessiin osallistuville osapuolille helposti. Kuvassa 1 on esitetty esimerkkejä osapuolista, joiden kanssa mScales voi olla vuorovaikutuksessa. mScalesin käyttöliittymän ja rajapintojen kautta kommunikointi onnistuu monien eri kohteiden kanssa. (mScales a.)



Kuva 1. Digitaalinen punnituspalvelu mScales (mScales a)

Punnituspalvelu voidaan ottaa käyttöön vaakatyypistä ja -toimittajasta riippumatta. mScales sopii käytettäväksi monilla teollisuuden toimialoilla, joissa punnitseminen on osa toimintaprosesseja. Palvelu on mukautettavissa asiakkaiden tarpeiden mukaisesti, sillä siihen on tarjolla erilaisia lisämoduuleja. Lisämoduulit laajentavat punnituspalvelun toiminnallisuutta yhdistämällä eri toimintoja, esimerkiksi viranomaisraportointia, yhden järjestelmän alle. (mScales b.)

mScalesin käyttöönotolla luvataan asiakkaille huomattavia taloudellisia hyötyjä. Perinteisesti punnitusprosessiin liittyy paljon manuaalista työtä ja erilaisia ylläpidettäviä oheislaitteita. Yritysten sisällä eri toimipisteiden tavat voivat erota keskenään huomattavasti, ja punnitustietojen käsittely voi olla epäjärjestelmällistä. mScalesin käytön tavoitteena on automatisoida ja yhtenäistää prosesseja sekä vähentää manuaalista työkuormaa. mScales pyrkii tehostamaan kaikkea punnitsemiseen liittyvää tiedonhallintaa. (mScales a.)

mScales tarjoaa työkaluja materiaalitapahtumien ja punnitusten raportointiin ja tarkasteluun, mutta osa näihin liittyvästä datasta voisi olla paremmin hyödynnettävissä ja ymmärrettävissä visualisoituna. Liiketoimintajärjestelmät ja raportointityökalut tarjoavat omat ratkaisunsa tähän ongelmaan, mutta visuaalisen datan esittäminen olisi käytännöllistä myös suoraan mScalesin käyttöliittymässä. Visualisoidusta datasta voisivat hyötyä kaikki mScalesin käyttäjät, niin asiakkaat kuin myös Lahti Precision Oy:n omat työntekijät, kuten asiakaspalvelu tai myynnin asiantuntijat.

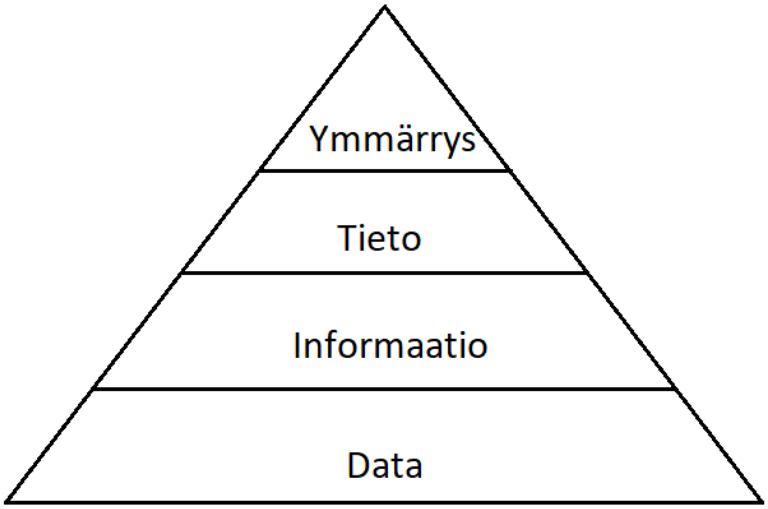
Visualisointia varten halutaan ottaa käyttöön työkalu, joka soveltuu JavaScript-teknologioihin perustuvaan ympäristöön. mScalesin kehityksessä käytetään Angular-kehystä, joten yhteensopivuus tämän kanssa olisi toivottavaa, mutta ei pakollista. Työkalulta toivotaan lisenssiä, joka mahdollistaisi sen käytön maksutta. Työkalulla halutaan havainnoida kaksitasoista tietoa, eli sen halutaan tukevan sellaisia visualisointikeinoja, joissa dataa voidaan mallintaa kahdella akselilla. Käytön suhteen työkalun halutaan sisältävän älykkäitä ominaisuuksia, kuten animaatioita, jo valmiiksi. Helppokäyttöisyys oli tavoiteltavaa, mutta ei ratkaiseva ominaisuus valinnassa.

Visualisoitavaa dataa halutaan käsitellä erillään mScalesin päätietokannasta, jotta visualisointiominaisuudesta saadaan varmasti suorituskykyinen. Datan käsittelyn ja varastoinnin halutaan olevan mahdollisimman automatisoitua ja ylläpidettävyydeltään yksinkertaista. Paljoa tiukkoja kriteerejä käsittelylle ja varastoinnille ei ole, mutta molempien halutaan tehostavan ominaisuuden kehitystä ja käyttöönottoa.

3 Data

3.1 Määritelmä

Datalla voidaan viitata toimialan mukaan eri asioihin. Data voi olla esimerkiksi kirjaimia tai numeroita, mittauksia tai havaintoja. Yleisellä tasolla datalla tarkoitetaan kuitenkin tosiasioihin perustuvaa joukkoa yksittäisistä arvoista, joilla ei itsessään ole suurta merkitystä tai selkeää kontekstia. Dataa voidaan jalostaa informaatioksi, ja siitä eteenpäin tiedoksi ja ymmärrykseksi, kuten kuvan 2 DIKW-mallissa (data, information, knowledge, wisdom pyramid) nähdään. Arkikielessä näitä mallin käsitteitä voidaan käyttää joskus toistensa synonyymeinä. (Ontotext.) Esimerkki datan eteenpäin jalostamisesta voisi olla kuormien punnitut painot, josta lasketaan päivän kuormien keskiarvo ja jonka perusteella tehdään päätöksiä tai kehitetään punnitustoimintaa.



Kuva 2. DIKW-pyramidi (mukailtu Data Literacy)

Datan ja sitä seuraavien välivaiheiden rajat ovat keskenään suhteellisia. Keskeistä on kuitenkin se, että tasolta toiselle siirryttäessä datasta löydetään uusia merkityksiä ja se saa lisää arvoa. Mitä enemmän dataa jalostetaan, sitä paremmin sitä pystyy hyödyntämään päätöksien teossa. (Ontotext.) Yhteyksien ja viittauksien löytäminen eri datajoukkojen välillä helpottuu, mitä enemmän dataa on käsitelty.

Yrityksille data on elintärkeää. Business intelligence, lyhennettynä BI ja suomennettuna liiketoimintatiedon hallinta, on tänä päivänä yhä merkittävämmässä asemassa organisaatioiden toiminnassa. Datan hallitseminen ja hyödyntäminen voivat olla menestystekijöitä, kun

kyse on strategisesta päätöksenteosta ja johtamisesta. Suunnitelmallinen liiketoimintatiedon hallinta voi antaa kilpailuetuja muihin alan yrityksiin nähden. (Lemonsoft.)

Liiketoimintatiedon hallintaan kuuluu perinteisesti tiedonkeruu, raportointi ja analysointi, jotka yhdessä tarjoavat työkaluja päätöksenteon, toiminnansuunnittelun ja johtamiseen tueksi (Lemonsoft). Tärkeää on, että data on helposti käsitettävissä ja ymmärrettävissä, jotta sitä voidaan vaivattomasti hyödyntää osapuolesta riippumatta. Yrityksissä monet roolit ja vastuualueet ovatkin yhä enemmän kytköksissä näihin liiketoimintatiedon hallinnan osa-alueisiin. Tietojohtaminen on nykypäiväinen johtamisen periaate, jossa pyritään tehostamaan käytössä olevan tiedon kokonaisvaltaista hyödyntämistä läpi organisaation. (Valtiovarainministeriö.)

3.2 Big data

Datan määrä on lisääntynyt rajusti viimeisen vuosikymmenen aikana, ja sen on arvioitu lisääntyvän seuraavien muutaman vuoden aikana vielä jyrkemmin (Statista 2021). Ilmiö on maailmanlaajuinen ja koskettaa jollain tapaa kaikkia vaiheita datan elämänkaarella. Datan määrään ja sen kasvuun on arvatenkin reagoitu: Big data käsitteenä on syntynyt jo ennen 2000-lukua, mutta sen suosion kasvu voidaan ajoittaa 2010-luvun vaihteeseen. (Salo 2013, 14)

Big datalla ei ole yksiselitteistä määritelmää, ja sillä voidaan viitata useampaan aiheeseen. Big datasta puhuttaessa voidaan viitata datan määrään ja sen lisääntymiseen ilmiönä tai vaihtoehtoisesti erilaisiin teknologioihin, palveluihin ja ratkaisuihin, joilla vastataan ilmiön vaikutuksiin. (Salo 2013, 10-12). Big data vastaa yritysten tarpeeseen hyödyntää ja hallita erilaisia datamassoja tarjoamalla erilaisia varastointi- ja analytiikkatyökaluja datan käsittelyyn. Tavoitteena on muokata ja jäsentää dataa helpommin ymmärrettäväksi ja käytettäväksi. Erilaisia ratkaisuja on tarjolla useita eivätkä ne kaikki sovi kaikkiin olosuhteisiin. Tärkeää on löytää omia edellytyksiä ja budjettia vastaava ratkaisu. (Salo 2013, 52 ja 70).

Salon (2013, 65) havaintojen mukaan big data ilmiönä asettaa datan käsittelylle erilaisia vaatimuksia ja rajoitteita. Perinteiset, relaatiomalliin perustuvat tietovarastointikeinot ovat toimivia keinoja, mutta eivät välttämättä skaalaudu, jos dataa on paljon. Big datan vaikutuksenkin takia suosiotaan ovat kasvattaneet NoSQL-tietokannat, jotka soveltuvat suurien datamassojen varastointiin hyvin. Nykypäivänä tiedon varastointiin liittyy erilaisia näkökulmia, mitä suunnittelutyössä on hyvä ottaa huomioon. Tiedon varastoinnin täytyy olla joustavaa, sillä data voi tulla eri lähteistä, olla eri muotoista tai eri tavoin käsiteltyä.

Big data ja sen tarjoamat vaihtoehdot eivät varsinaisesti ratkaise käsillä olevaa työtä. Ilmiötä voi kuitenkin pitää yläkäsitteenä tai ohjaavana tekijänä monille työssä hyödynnettäville

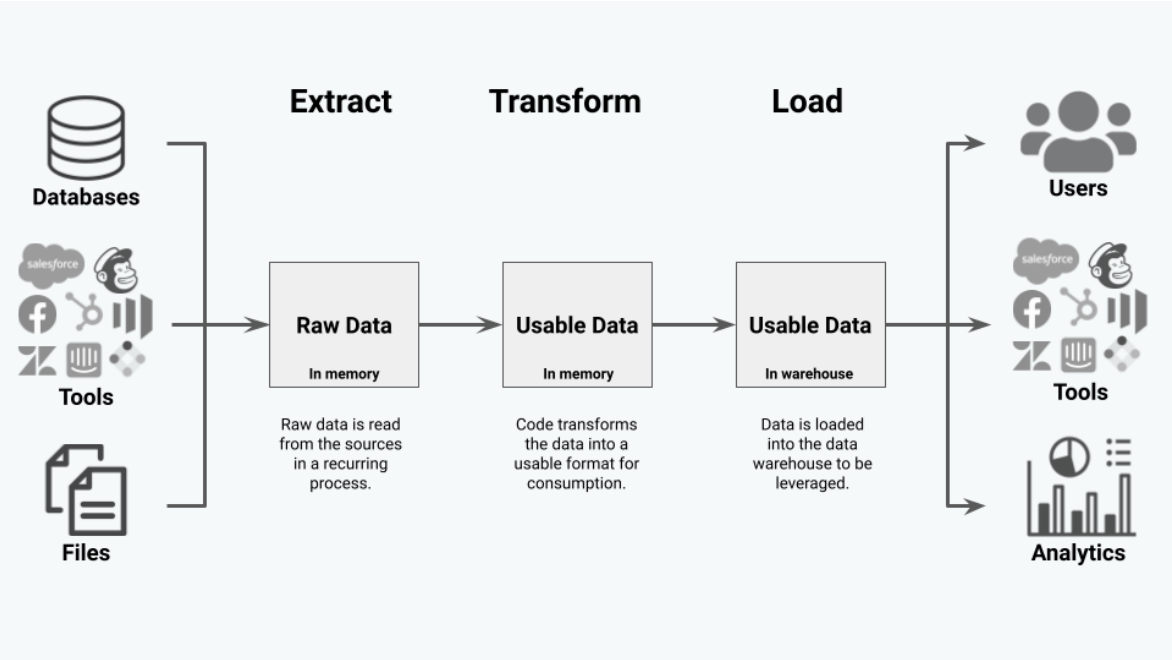
periaatteille ja työkaluille. Big datan tavoitteet ja pyrkimykset ovat ainakin osittain yhteneväiset tämän työn kanssa. Data on vain harvoin heti käyttövalmista etenkään dataa visualisoitaessa. On hyvä pitää mielessä, miten ja mihin kaikkeen valtava määrä dataa voi todella vaikuttaa.

3.3 Datan käsittely

Koska data ei itsessään ole kovin merkityksellistä, se vaatii yleensä jonkinlaista käsittelyä, jotta sitä voidaan helpommin hyödyntää. Käsittelystä puhutaan myös datan integrointina. Datan käsittely kokonaisuutena voidaan jakaa kolmeen keskeiseen osaan: datan keräämiseen, muokkaamiseen ja tallettamiseen. Myöhempiin vaiheisiin voi usein kuulua datan hallinta, analysointi tai visualisointi.

3.3.1 Käsittelyprosessit

ETL-prosessi (extract, transform ja load) ja samalla siihen liittyvien työkalujen historia ulottuu 1970-luvulle asti. ETL-prosessi on vaihtoehto datan käsittelyyn ja se on syntynyt tarpeesta yhdistää yritysten eri datalähteitä liiketoiminnallisesta näkökulmasta. Kuvassa 3 on esitelty datan käsittelyn vaiheet ETL-prosessin mukaisesti.

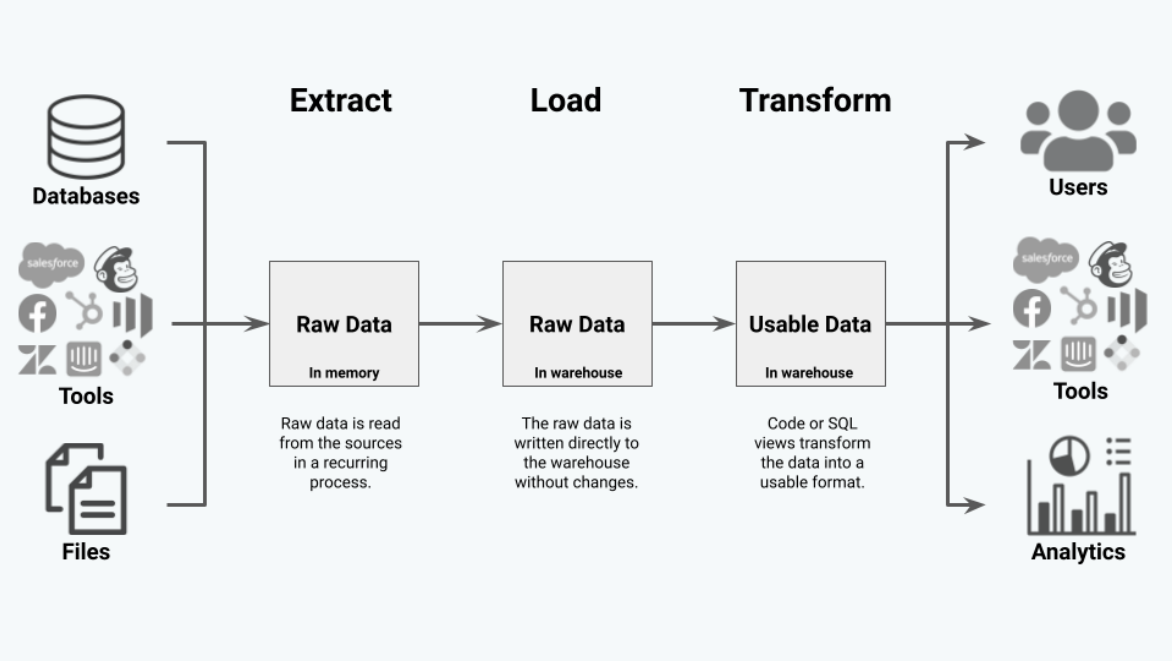


Kuva 3. ETL-prosessi (Leonard 2021)

ETL-prosessin toiminta etenee ennalta määriteltyjen vaiheiden kautta. Prosessi alkaa datan keräämisellä ja yhteen kokoamisella mahdollisesti useistakin erilaisista tietolähteistä, jonka jälkeen dataa muokataan. Muokkausvaiheeseen voi kuulua erilaisia vaiheita, kuten datan

putsausta, yhdistelyä tai erottelua. (Leonard 2021). Esimerkkejä muokkauskeinoista ovat sarakkeiden ja arvojen lisääminen tai poistaminen ja arvojen muuntaminen. Viimeisessä vaiheessa data varastoidaan niin, että se on helposti ja kattavasti eri kohteiden, käyttäjien ja työkalujen käytettävissä. Kohdevarastotyyppejä on erilaisia: perinteisiä vaihtoehtoja ovat yritystietovarastot (data warehouse) tai aihekohtaisesti rajatut datamartit (data mart). Big datan myötä nykyaikainen valinta varastoksi on tietoallas (data lake), joka soveltuu erityisesti suurille, käsittelemättömille datamassoille. Se ei kuitenkaan sellaisenaan aina sovellu korvaaman yritystietovarastoja vaan pikemminkin niiden rinnalle. Kaikki ETL-työkalut eivät aina edes tue tietoaltaita. (SAS).

ETL-prosessin rinnalle on muodostunut ELT-prosessi (extract, load ja transform). ELT-prosessi on hyvin samankaltainen kuin ETL-prosessi, ja se muodostuu samoista käsittelyvaiheista, kuten kuvassa 4 on esitelty, mutta käsittelyn vaiheet ovat prosesseissa eri järjestyksessä.



Kuva 4. ELT-prosessi (Leonard 2021)

ELT-prosessissa data kerätään ensin, jonka jälkeen se ETL-prosessista poiketen viedään kohdevarastoon. Datan muuntaminen tapahtuu vasta, kun dataa haetaan käyttöä varten. Muunnos tapahtuu käyttökohteen mukaan oikeanlaiseksi, sillä tarvittava datan muoto voi erota käyttäjän tai työkalun välillä. Prosessin sanotaan olevan tehokkaampi, koska datan muuntaminen on tuotu osaksi varastointia, olennaisesti poistaen itsenäisen muuntamisvaiheen kokonaan. (Talend b.)

Eroja prosessien välillä on myös muita vaiheiden järjestyksen lisäksi. Keskeinen ero prosessien välillä on varastoitavan datan määrä. ETL-prosessissa määrä usein laskee muuntamisen myötä ennen sen varastointia, mutta ELT-prosessissa säilytettävä data voi olla muodoltaan hyvin raakaa tai vähän käsiteltyä, ja siksi sen määrä on yleensä huomattavasti suurempi. (Talend b.) ELT-prosessit kestävät ajallisesti lähtökohtaisesti vähemmän ja tarvitsevat vähemmän hallintaa muuntamisvaiheen siirtyessä käytön yhteyteen

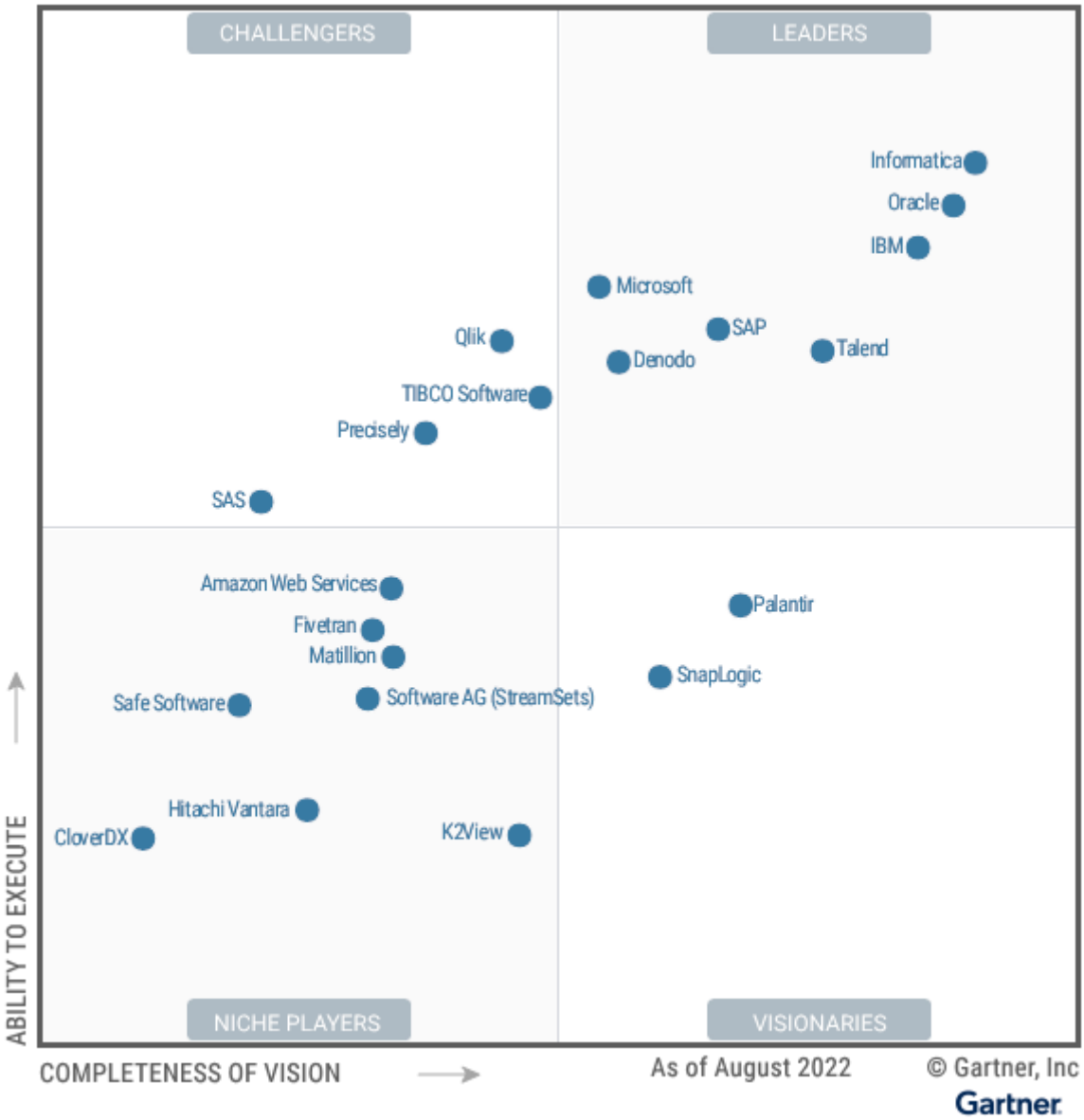
Oli valittu prosessi tai kohdevarasto mikä tahansa, on se kokonaisuutena hyvinkin mukautuva. Suunnittelussa on monia yksityiskohtia, jotka on hyvä ottaa huomioon: Kuinka paljon dataa ja datalähteitä on, ja millaisia loppukäyttäjien ja -kohteiden tarpeet ovat. Liiketoiminnalliset tavoitteet yleensä ohjaavat prosessin käyttöönottoa. (Talend b.)

ETL- ja ELT-prosesseja vertaillaan luonnollisesti paljon toisiinsa. ELT-prosessi on kahdesta nykyaikaisempi ja joissain määrin ehkä jopa kehitetty paikkaamaan epäkohtia, joita ETL-prosessissa voi olla. Molemmilla on silti paikkansa datan käsittelyn kentällä, sillä niiden tuottama hyöty voi erota keskenään. On tärkeää tunnistaa, mitä halutaan saavuttaa ja millä keinoin, kun prosessia ollaan ottamassa käyttöön. (Talend c.)

3.3.2 ETL- ja ELT-työkalujen tarkastelu

ETL- ja ELT-työkalujen suhteen vaihtoehtoja on tarjolla todella paljon. Valittavana on niin avoimen lähdekoodin ratkaisuja kuin myös kaupallisia versioita. Ominaisuuksiltaan ja käyttötapauksiltaan työkalut voivat erota toisistaan merkittävästi, mihin kannattaa kiinnittää tarkasti huomiota prosessia otettaessa käyttöön. Osa työkaluista voi olla kohdistettu vain tietyn vaiheen suorittamiseen, ja osa vain tietynlaisia lähde- tai kohdevarastoja varten. Jotkut työkalut soveltuvat vain tietyn tyyppiseen datan muokkaukseen, ja osa taas tarjoaa parempaa tukea tiettyä käyttövaihetta kuten analysointia varten.

Gartner tekee vuosittain eri liiketoiminnallisista ja teknologisista aihealueista ja trendeistä raportteja, joissa se jakaa alan toimijoita neljään eri kategoriaan. Nämä kategoriat ovat johtajat, visiönäärit, markkinarakoihin keskittyvät tai kapea-alaiset ja haastajat (leaders, visionaries, niche players ja challengers). Raportit ovat tarkoitettu antamaan näkemys markkinoiden nykytilanteista ja eri toimijoiden vahvuuksista ja heikkouksista ajanhetkellä. (Gartner.) Kuvassa 5 on Gartnerin näkemys (2022b) datan käsittelyn työkalujen vertautumisesta toisiinsa nähden Magic Quadrant -raportissa. Raportissa on vertailtu yli 20 eri työkalua.



Kuva 5. Magic Quadrant -tulkinta datan käsittelyn työkaluista vuonna 2022 (Gartner 2022b)

Gartner on tarkastellut datan käsittelyn eri toimijoita ja heidän työkalujaan, ja arvoinut näitä työkaluja erilaisten kriteerien ja ominaisuuksien näkökulmasta, jotka Gartnerin mukaan ovat oleellisia niin työkalun käytölle kuin myös menestymiselle tulevaisuudessa. Tällaisia kriteerejä ovat esimerkiksi laaja tuki erilaisille lähdevarastoille ja monimutkaisille datan muokkauksille. Magic Quadrant -raportissa käsitellyistä vaihtoehdoista osa on suurien yritysten markkinoimia ratkaisuja, mutta kiitettävä osuus on myös pienempien ja tuntemattomampien toimijoiden työkaluja. (Gartner 2022b).

Talend on yritys, joka on keskittynyt tarjoamaan datan käsittelyyn työkaluja ja palveluita. Yritys onkin pärjännyt Gartnerin Magic Quadrant -raporteissa hyvin jo useiden vuosien ajan. Talend tarjoaa datan käsittelyyn kahta, joustavaa ja mukautuvaa työkalua, Data Fabric ja Stitch. Data Fabric on näistä kahdesta kokonaisvaltaisempi alustaratkaisu datan käsittelyyn

ja hyödyntämiseen. Työkalulla on mahdollista käsitellä datan koko elinkaarta. Työkalun vahvuuksia mainostetaan olevan erityisesti datan eheyden ja laadun huomiointi, kattava tuki eri datalähteille sekä datan helppo siirtäminen muihin sovelluksiin rajapintojen avulla. (Talend a).

Talendin toinen työkalu, Stitch, on Data Fabric -työkalua perinteisempi ETL-ratkaisu. Työkalu on kooditon vaihtoehto datan käsittelyyn. Stitch tukee laajasti Data Fabric -työkalun tavoin eri datalähteitä, -kohteita ja analysointityökaluja. Työkalu tarjoaa laajasti ominaisuuksia prosessin parantamiseksi. (Stitch a).

IBM on Gartnerin mukaan johtava toimija alalla ja yritys tarjoaa useita eri työkaluja datan elinkaarta ja käsittelyä varten (Gartner 2022b). IBM DataStage on yrityksen tarjoama kokonaisvaltainen ratkaisu ETL- ja ELT-prosessien toteuttamista varten. Työkalu sisältää monia ominaisuuksia prosessin käyttöön ja optimointiin. Työkalu tukee hyvin laajaa valikoimaa eri tietolähteitä. DataStage-työkalun voi ottaa käyttöön paikallisesti omalla palvelimellaan tai pilvipalveluna. (IBM.)

AWS Glue on Amazonin tarjoama ETL-työkalu, osana yrityksen Amazon Web Services - eli AWS-työkalujen ja palveluiden valikoimaa. Gartnerin raportissa Amazon ja sen Glue-työkalu ovat uusi, vasta tänä vuonna raporttiin sisällytetty kapea-alainen toimija datan käsittelyn kentällä. (Gartner 2022.) Glue tarjoaa samankaltaisen valikoiman ominaisuuksia kuin jo edellä mainitut työkalut. Työkalu kuitenkin tukee pitkälti vain AWS-lähtöisiä datalähteitä ja -kohteita, ja näin ollen se soveltuu erityisesti sellaisiin ympäristöihin, joissa on jo valmiiksi käytössä muita AWS-palveluita. (AWS a.)

Glue-työkaluun kuuluvat myös Glue DataBrew ja Glue Data Studio, jotka mahdollistavat prosessin käytön ilman koodia. Molemmat ominaisuudet on tuotu osaksi Glue-työkalua helpottamaan sen käyttökokemusta visuaalisten käyttöliittymien kautta. (AWS a.) Glue DataBrew ja sen käyttöä on esitelty kuvassa 6. Työkalu tarjoaa datan muokkaamiseen ja arviointiin keinoja.

Kuva 6. Glue DataBrew ja esimerkki sen toiminnasta (AWS a)

Glue DataBrew on apuväline ETL-prosessin toteuttamiseen, ja sen pääpainona on tehdä datan muokkaamisesta ja käsittelystä helppoa. Glue DataBrew käyttää Glue-työkalun tavoin samankaltaisia keinoja yhdistää datalähteet itseensä, jonka jälkeen yhdistettyä dataa voi muokata ja muuntaa sisäänrakennetuilla funktioilla ja muunnoksilla. Datan eheyttä voidaan myös tutkia. Työkalulla voidaan rakentaa datan muuntamisesta automatisoituja ja ajastettuja suorituksia, jossa datan muuntamiseen käytetään ennalta määriteltyjä reseptejä. (AWS b.)

3.4 Datan visualisointi

Datan keräämisen, käsittelyn ja varastoinnin jälkeen tulee usein vaihe, jolloin sitä hyödynnetään ja jalostetaan eteenpäin, datasta informaatioksi ja tiedoksi. Datan visualisointi on erittäin tehokas apuväline tässä vaiheessa. Visualisoinnin tavoitteena on auttaa lukijaa ymmärtämään esitettävää dataa paremmin sekä käsittämään ilmiöitä datan taustalla helpommin. Visualisointikeinoja on useita: Dataa voidaan tuoda paremmin ymmärrettäväksi esimerkiksi erilaisten kaavioiden, diagrammien, taulukoiden, karttojen tai tunnuslukujen avulla.

3.4.1 Visualisointi ja informaatiomuotoilu

Datan visualisointia ohjaavat periaatteet perustuvat informaatiomuotoilun teoriaan. Informaatiomuotoilun päämääränä on esittää ja välittää tietoa mahdollisimman selkeästi ja helppolukuisesti. Käsitteenä se on peräisin 1970-luvun Yhdysvalloista. (Koponen ym. 2016, 19-20).

Koposen ym. (2016, 20-23) mukaan informaatiomuotoilu voidaan usein jakaa kahteen eri kategoriaan, infografiikoihin ja datan visualisointiin. Infografiikat ovat luonteeltaan selittäviä (explanatory) ja visualisointi kartoittavaa (exploratory). Kategoriat voidaan hahmottaa niin, että infografiikat kertovat tarinan, ja visualisointi auttaa lukijaa löytämään oman näkemyksensä tarinasta. Ero kategorioiden välillä ei kuitenkaan ole yksiselitteinen ja käsitteitä saatetaan käyttää myös ristiin ja myös eri yhteyksissä.

Informaatiomuotoilu ja graafinen suunnittelu ovat hyvin samansukuisia, mutta ne eroavat toisistaan kuitenkin huomattavasti. Graafinen suunnittelun päämääränä on kuvallinen viestintä, vetovoimainen ja taiteellinen ilmaisu, mikä kiteytyy usein tuotemerkkien esilletuomiseen. Informaatiomuotoilu keskittyy päinvastoin pelkästään tiedon tehokkaaseen välittämiseen. (Koponen ym. 2016, 20). Tosielämän esimerkki graafisen suunnittelun tuloksesta voisi olla elintarvikkeen pakkauksen ulkoasu ja informaatiomuotoilusta kaupungin julkisen liikenteen kartta. Molemmissa on kuitenkin monia yhdistäviä piirteitä erityisesti muotoilun teoriassa.

Visualisointi pohjautuu pitkälti numeraaliseen dataan, ja sen taustalla on tiedon vertailu keskenään. Arvojen merkitys on usein helpompaa hahmottaa, kun sitä vertaillaan johonkin toiseen. Visualisoinnilla voidaan pääasiassa kuvata viittä eri relaatiota aineistoissa: lukumäärää tai suuruutta, järjestystä, kategorioita, aikaa tai sijaintia. Lopputulokset syntyvät pitkälti näiden edellä mainittujen relaatioiden yhdistelmästä. (Koponen ym. 2016, 26).

Datan visualisoinnissa on paikkansa myös mittasuhteille, väreille, fonteille ja muulle muotoilun teorialle. Molemmat teokset Koposelta ym. (2016 ja 2019) ovat täynnä periaatteita ja ohjeita, joita noudattaa tai soveltaa visualisointia suunniteltaessa ja tehtäessä. Nykypäiväisissä, moderneissa datan visualisoinneissa on huomioitu myös se, miten käyttäjä voi olla vuorovaikutuksessa visualisointien kanssa. Älylaitteet tuovat lisää arvoa ja mahdollisuuksia käsittää visualisointeja. Vaihtoehtoja on paljon, ja niillä voidaan parantaa datan visualisointien ymmärrettävyyttä ja saavutettavuutta suuresti.

Visualisointi ei missään nimessä ole aina yksinkertaista. Erilaiset tietoaineistot tuovat omat haasteensa suunnittelulle, eivätkä kaikki visualisointikeinot sovellu kaikkiin käyttötapauksiin. Onnistuneessa ja hyödyllisessä datan visualisoinnissa on osattu valita dataa parhaiten kuvaava visualisoinnin keino, joka tukee päämäärää tehdä datasta helpommin omaksuttavaa. Huonosti valittu visualisoinnin keino voi jopa hämmentää lukijaa ja viedä huomiota täysin vastakkaiseen suuntaan. Riski sortua virheellisen tiedon välittämiseen on hyvä tiedostaa. (Tableau). Datan visualisointi on tehokas keino, mutta se ei ratkaise kaikkea. Jossain tapauksissa datan visualisointi voi olla jopa järkevämpää jättää kokonaan tekemättä.

Suunnittelussa on tärkeää tutustua käsillä olevaan dataan perusteellisesti. Aineistoissa voi olla useampia ilmiöitä, joita voidaan selkeyttää visualisoimalla. Ilmiöt voivat vaatia tuekseen erilaisia toimenpiteitä, eikä niiden sekoittaminen keskenään ole aina hyödyllistä. (Koponen ym. 2016, 19) On oleellista löytää datasta oikea näkökulma, jonka avulla voidaan esimerkiksi nopeuttaa päätöksentekoa tai toistuvien kaavojen ja uusien ominaisuuksien tunnistamista.

3.4.2 Visualisoinnin työkalut

Dataa voidaan visualisoida lukemattomilla eri tavoilla ja työkaluilla. Yksinkertainen ja perinteinen tapa on mallintaa asioita paperilla ja kynällä, ja se onkin tehokasta visualisoinnin suunnittelussa. Alan erikoistuneet tietokoneohjelmat- ja työkalut tarjoavat tänä päivänä kuitenkin monia etuja dataa analysoitaessa ja visualisoitaessa. (Gartner 2022a.)

Monet visualisointityökalut ja -mahdollisuudet ovat yhdistetty liiketoimintatiedon hallintatyökaluihin, jolloin visualisointi on helposti ja nopeasti hyödynnettävissä osana datan käsittelyn kokonaisprosessia. Hallintatyökaluihin on usein myös sisäänrakennettu analysointiin tarvittavia ominaisuuksia, ja usein analysoinnin rooli on työkaluissa visualisointia suurempi. (Gartner 2022a.)

Tunnettuja ja suuria nimiä datan analysoinnin ja visualisoinnin työkaluissa ovat esimerkiksi Microsoftin Power BI, Googlen Data Studio, Talend sekä Tableau, kuten Gartnerin (2022a) analytiikka- ja liiketoimintatiedon hallinta-alustojen Magic Quadrant-raportista selviää. Edellä mainitut sekä muut raportissa esitellyt alustat ovat suunnattu hieman erilaisia käyttötarkoituksia ja käyttökokemuksia varten. Työkalut voivat olla kohdennettu tietynlaisen datan prosessointia varten, tai ne voivat tarjota erilaisia valmiita ratkaisuja datan analysointiin ja visualisointiin.

Datan visualisointiin löytyy myös keinoja liiketoimintatiedon hallinnan ulkopuolelta. Webkehityksessä on tarjolla monia eri kirjastoja datan visualisointiin, ohjelmointikielestä ja ohjelmistoviitekehyksestä riippumatta. Avoimen lähdekoodin yhteisövetoisia työkaluja ovat esimerkiksi D3.js, Chart.js ApexCharts.

D3.js on avoimen lähdekoodin JavaScript-kirjasto, joka tarjoaa tehokkaita rajapintoja visualisoida dataa verkkoselaimissa. Kirjasto pyrkii tarjoamaan vahvan ja suoraviivaisen perustan datan visualisointia varten hyödyntäen HTML SVG -elementtiä. D3.js ei lähtökohtaisesti tarjoa valmiita ratkaisuja kaikkia yksityiskohtia ja tilanteita varten, vaan pikemminkin laajan valikoiman rajapintoja datan visualisointiin eri olosuhteissa. D3.js markkinoi itseään monikäyttöiseksi, joustavaksi sekä hyvin yhteensopivaksi perinteisten HTML- ja CSS-elementtien kanssa. (D3.)

Chart.js on toinen tunnettu JavaScript-kirjasto datan visualisointiin. Avoimen lähdekoodin työkalu mainostaa olevansa yksinkertainen vaihtoehto datan visualisointiin. Chart.js käyttää visualisointien pohjana HTML Canvas -elementtiä. Työkalulla voi lähtökohtaisesti visualisoida kahdeksaa perinteisempää kaaviota. Chart.js tarjoaa näiden kaavioiden yhteydessä laajan valikoiman käytettäviä ominaisuuksia, kuten animaatioita ja vuorovaikutusmahdollisuuksia. (Chart.js 2022.) Työkaluun on liitettävissä useita eri lisäosia, joilla laajentaa sen käytettävyyttä.

ApexCharts on tuoreempi vaihtoehto datan visualisointiin verkkoselaimissa, mutta samalla vähemmän tunnetumpi. Avoimen lähdekoodin, JavaScript-pohjaisen kirjaston toiminta perustuu HTML SVG -elementtiin. ApexCharts tukee useampaa kaaviomallia, ja perinteisten mallien lisäksi löytyy myös erityisempiä vaihtoehtoja. Työkaluun sisältyy useita erilaisia vuorovaikutteisia ja dynaamisia ominaisuuksia. (ApexCharts.) Ominaisuuksiltaan ApexCharts ja Chart.js ovat hyvin samankaltaiset.

4 Suunnitteluvaihe

4.1 Visualisoinnin suunnittelu

Tämän opinnäytetyön päätavoitteena oli konseptitasolla suunnitella ja toteuttaa ominaisuus, jonka avulla punnituspalvelu mScalesin käyttäjät voisivat nähdä punnitusdataansa visualisoituna. Ominaisuuden taustalla oli pyrkimys jatkojalostaa olemassa olevaa dataa ja sitä kautta tuoda esille palvelun käyttäjille datasta löytyviä merkityksiä ja ilmiöitä. Työn käytännön osuus alkoi visualisointien suunnittelulla.

Käyttäjille näytettävien kaavioiden haluttiin olevan tässä kohtaa ominaisuuden kehitystä yksinkertaisia, mutta silti hyödyllisiä. Näin ominaisuuden tarpeellisuutta ja kehityssuuntaa voidaan arvioida jo työn varhaisissakin vaiheissa. Suunnitteluprosessi alkoi tarkastelemalla ja perehtymällä olemassa olevaan dataan. mScalesin toimintaan liittyy keskeisesti punnitukset ja näihin liittyvä oheisdata.

Suunnittelun tukena käytettiin mScalesin asiakaspalvelun ja myynnin asiantuntijoiden ideoita ja ajatuksia siitä, millaisista visualisoinneista voisivat hyötyä palvelun käyttäjät. Tiimin ideoiden, datan tarkastelun sekä tunnettujen visualisointityökalujen esimerkkien ja valmiiden mallien tutkimisen avulla päädyttiin kolmeen eri visualisointiin:

- Esitetään vuorokauden aikana tehdyt punnitukset kellonajan mukaan.
- Esitetään kuormien lastausajat.
- Esitetään operaattori- ja mobiilikäyttöliittymällä tehtyjen punnitusten osuudet.

Näiden kolmen visualisoinnin yhteydessä haluttiin myös näyttää tietyn aikavälin aikana tehtyjen materiaalitapahtumien kokonaismäärä. Kyseinen tunnusluku sitoo visualisointeja yhteen ja voi helpottaa niiden asiayhteyden hahmottamista.

Visualisointien suhteen päädyttiin esittämään päivän aikana tehdyt punnitukset pystysuorana pylväskaaviona, kuormien lastausajat vaakasuorana pylväskaaviona ja käyttöliittymien osuudet punnituksista ympyräkaaviona tai vastaavana. Kaikkia kolmea testattiin myös mallintaa muina kaavioina niin käsin kuin myös työkalun valinnan yhteydessä. Esitettävät ilmiöt vaikuttivat selkeimmin ja helpoimmin hahmotettavilta valituissa muodoissa. Valitut visualisointikeinot tuntuivat luonnollisilta, mutta kokeilun myötä myös hyödyllisimmiltä. Lähtökohdat muotoilla valituista kaavioista halutunlaiset ja ilmiöitä tukevat olivat suotuisat, mutta muutokset alkuperäisiin suunnitelmiin onnistuisivat, jos kehityksen aikana ilmenisi muita vaikuttavia tekijöitä.

Visualisointien määrittelyn jälkeen käsillä olevaa dataa tuli tarkastella uudesta näkökulmasta, sillä kaikki kolme visualisointia ja tunnusluku vaativat hieman erilaista dataa tuekseen. Aikasarja- tai sitä muistuttava data ovat yleinen datan muoto visualisoinnissa, sillä sen käyttäminen on usein johdonmukaisinta. Valituissa visualisoinneissa haluttiin pääsääntöisesti näyttää eri asioista yhteenlaskettuja arvoja, mutta taustalla olemassa oleva data on perinteisen relaatiomallin mukaisesti rivitettyä. Esimerkiksi kellonajan mukaan esitettävät punnitusrivit tulisi laskea yhteen jokaiselta tunnilta. Kuvassa 7 on hahmoteltu, miltä visualisoinnissa käytettävä data voisi näyttää yhteenlasketuilla arvoilla. Data olisi hyvin yleisluontoista tällaisessa muodossa.

```
{
    "timestamp": "2022-10-01T12:00:00.000Z",
    "totalAmount": 15,
    "operator": 7,
    "mobile": 8
}
```

Kuva 7. Esimerkki visualisointidatasta

Yhteenlasketut arvot esimerkiksi jokaiselta tunnilta soveltuisivat lähtökohtaisesti hyvin visualisointiin. Kuormien lastausaikojen esittäminen ei kuitenkaan suoraan onnistu yhteenlasketuilla summilla. Lastausajat tarvitsisivat taustalle rivikohtaista tietoa. Vaihtoehtoisesti lastausaikoja voitaisiin esittää keskiarvoina. Näistä rivikohtainen tarkkuus vaikutti kuitenkin hyödyllisemmältä ja dataa paremmin kuvaavalta vaihtoehdolta. Lastausajoissa voi olla suurtakin vaihtelua, ja siksi keskiarvon laskeminen ei välttämättä antaisi todenmukaista katsausta ilmiöön. Tiedon oikeellisuuden ja lastausajoissa olevien mahdollisten poikkeamien tunnistaminen on helpompaa ja selkeämpää rivikohtaisen tiedon kanssa.

Datan hyödyntäminen rivikohtaisesti mahdollistaisi myöhemmin kehityksessä visualisointien vuorovaikutuksien toteutettavuuden sekä paremman laajennettavuuden. Jos datarivien yhteyteen liittää yksilöivää tietoa (kuva 8), voidaan sen avulla visualisoitua dataa liittää paremmin mScalesin käyttöliittymän muihin osiin, mikä voisi kiinnostaa käyttäjää visualisoinnin tarkastelun jälkeen.

```
{
    "id": 1,
    "timestamp": "2022-10-01T12:00:00.000Z",
    "operator": true,
    "loadingTime": 60
}
```

Kuva 8. Esimerkki rivikohtaisesta visualisointidatasta

Lisäämällä esimerkiksi punnituksen datariville, voi käyttäjä helpommin päästä käsiksi kyseiseen punnitukseen liittyvään muuhun dataan, jota ei ole mukana visualisoinnissa. Visualisoitavan datan liittäminen yhteen muuhun käyttöliittymässä saatavilla olevaan dataan muodostaa dynaamisemman kokonaisuuden käyttäjän näkökulmasta.

Oheistiedon liittäminen osaksi datarivejä voi vähentää tai jopa kokonaan poistaa tarpeen hakea minkäänlaista dataa päätietokannasta visualisointiominaisuutta varten. Jos käyttöliittymässä halutaan suodattaa visualisoitavaa dataa esimerkiksi toimipisteen perusteella, täytyisi toimipistetiedot hakea päätietokannasta erikseen. Vaihtoehtoisesti toimipistetiedot voidaan tarpeellisin osin liittää mukaan visualisoitaviin datariveihin, ja käyttää niitä hyödyksi suodatuksessa.

Suunnittelussa oli tarpeen panna merkille myös ajan ja aikaleimojen käyttö datassa ja visualisoinnissa. Aikavälit ja -leimat vaikuttavat käytettävän datan muotoiluun kuin myös visualisointiin ja sen vuorovaikutusmahdollisuuksiin. Valitut visualisoinnit eivät niinkään aseta erikseen rajoituksia aikavälien käytölle, ja ne voisi todennäköisesti saada toimimaan lyhyemmillä ja pidemmillä aikaväleillä.

Yhdenmukaisuuden näkökulmasta valittiin ainoaksi aikavälivaihtoehdoksi tässä kohtaa tarkastella ja visualisoida dataa päiväkohtaisesti. Aikavälin pidentäminen viikoksi tai lyhentäminen tunniksi teknisesti onnistuisi varmasti, mutta valittujen visualisoitavien ilmiöiden suhteen se ei välttämättä olisi niin yksinkertaista. Päivän mittainen aikaväli antaa täsmällisen kuvan datasta, kun taas viikon tai tunnin mittainen väli voisi mahdollisesti antaa datasta epätarkemman vaikutelman tässä muodossa. Valittuja visualisointeja hieman muuttamalla tai vaihtamalla kokonaan toisenlaiseen voisi olla suotavaa muiden aikavälivaihtoehtojen harkinnan suhteen. Esimerkiksi viikon aikana tehtyjen punnitusten näyttäminen tunneittain muuttuisi helposti hankalasti tulkittavaksi ja vaikealukuiseksi. Jos punnituksia vaihtoehtoisesti ryhmittelisi vuorokauden tai päivänaikojen, kuten aamun ja illan, mukaan, voisi visualisointi olla helpommin ymmärrettävissä.

4.2 Työkalun valinta

Kantava tekijä koko opinnäytetyön aikana oli toteuttaa tarvittavia asioita valmiiden ratkaisujen avulla. Datan visualisointia varten haluttiin ottaa käyttöön erillinen työkalu (kirjasto tai kehys), joka täyttäisi tietyt kriteerit. Työkalun haluttiin ennen kaikkea soveltuvan JavaScript-pohjaiseen ympäristöön ilman ylimääräisiä sovittimia, joita esimerkiksi toisen ohjelmointikielen käytön takia voitaisiin tarvita. mScalesin toteutuksessa on hyödynnetty Angular-kehystä, joten työkalun kehysintegraatio tai TypeScript-yhteensopivuus olisi toivottavaa,

mutta ei kuitenkaan pakollista. Lisenssin haluttiin ensisijaisesti olevan ilmainen, mutta maksulliset vaihtoehdot voisivat olla neuvoteltavissa kehityksen jatkuessa myöhemmin.

Työkalun haluttiin soveltuvan kaksitasoiselle tiedolle ja mahdollistavan älykkäiden toimintojen toteutus ja käyttö. Esimerkiksi työkaluvihjeet (tooltips) tai kaavioiden animointi olivat haluttuja ominaisuuksia. Työkalun käytön haluttiin olevan edes jollain tasolla helppokäyttöistä ja nopeuttavan kehitystä. Vertailtaviksi työkaluiksi valittiin D3.js, Chart.js ja ApexCharts, joiden ominaisuuksia vertailtiin keskenään annettuihin kriteereihin (taulukko 1). Työkalut valikoituivat vertailtaviksi tarkastelemalla suosittuja vaihtoehtoja toteuttaa datan visualisointia webkehityksessä, sekä työkaluja, joista löytyi integraatio Angular-kehyksen kanssa. Kaikkiin työkaluihin on rakennettu mahdollisuus toteuttaa pylväs- ja ympyräkaavioita, ja työkaluilla oli selkeä dokumentaatio näistä kaavioista.

Työkalu	ApexCharts	Chart.js	D3.js
JavaScript	Kyllä, Angular-integraatio saatavilla	Kyllä, Angular-integraatio saatavilla	Kyllä
Lisenssi	MIT-lisenssi (ilmainen)	MIT-lisenssi (ilmainen)	ISC-lisenssi (ilmainen)
Soveltuvuus kaksitasoiselle tiedolle	Kyllä	Kyllä	Kyllä, myös muut ulottuvuudet
Älykkäät ominaisuudet	Kyllä	Kyllä	Kyllä
Helppokäyttöisyys	Paljon valmiita rajapintoja ja ratkaisuja	Paljon valmiita rajapintoja ja ratkaisuja	Vaikeampi ottaa käyttöön ja oppia

Taulukko 1. Datan visualisoinnin työkalut ja niiden ominaisuudet

Työkalujen ominaisuuksien vertailussa kaikki kolme vaihtoehtoa pärjäsivät hyvin keskenään ja täyttivät annetut kriteerit. D3.js jäi vertailussa kuitenkin kahdesta muusta vaihtoehdosta hieman jälkeen. Työkalu on erinomainen, mutta ehkä liian edistynyt tässä kohtaa kehitystä. D3.js tarjoaa muihin työkaluihin nähden enemmän keinoja toteuttaa monimutkaisempia visualisointeja ja käsitellä visualisoitavaa dataa laajemmin. D3.js suorittaa vain vähän toiminnallisuuksia automaattisesti, eli kehityksessä täytyy käydä läpi ja toteuttaa visualisoinnin vaiheita perusteellisempi. Tämän takia työkalu ei välttämättä ole helppokäyttöisin ja siksi heikompi valinta, jos ominaisuuden toteutukseen halutaan löytää mahdollisimman valmiita ratkaisuja.

Chart.js ja ApexCharts vertautuivat ominaisuuksiltaan hyvin samankaltaisesti, joten oli aiheellista tarkastella työkalujen muita ominaisuuksia. Työkalut perustuvat eri HTML-elementtien käyttöön. Chart.js käyttää HTML canvas -elementtiä taustalla ja ApexCharts SVG-

elementtiä. Molemmat ovat laajasti tuettuja eri selaimissa, myös vanhemmissa versioissa. Canvas-elementin käyttöä ei kuitenkaan suositella sen heikon saavutettavuuden ja responsiivisuuden takia (MDN Web Docs 2022). Chart.js tarjoaa kuitenkin sisäänrakennettuja keinoja korjata ja parantaa näitä kahta epäkohtaa.

Canvas- ja SVG-elementtien käytön suhteen SVG on hieman helpompi ohjelmoinnin näkökulmasta. SVG-elementtien sisään piirretyt kuviot ovat kaikki itsenäisiä elementtejä, jolloin niistä on helpompi etsiä ja korjata virheitä, jos visualisointi ei toimi halutusti. SVG-elementin käyttö voi kuitenkin asettaa haasteita silloin, kun visualisoitavaa dataa on paljon. Datan määrä on yhteydessä piirrettävien elementtien määrään, ja se voi vaikuttaa suorituskykyyn haitallisesti. Canvas piirretään usein vain kerran, jolloin se on suorituskyvyltään hieman edullisempi käyttää suuremman datamäärän kanssa.

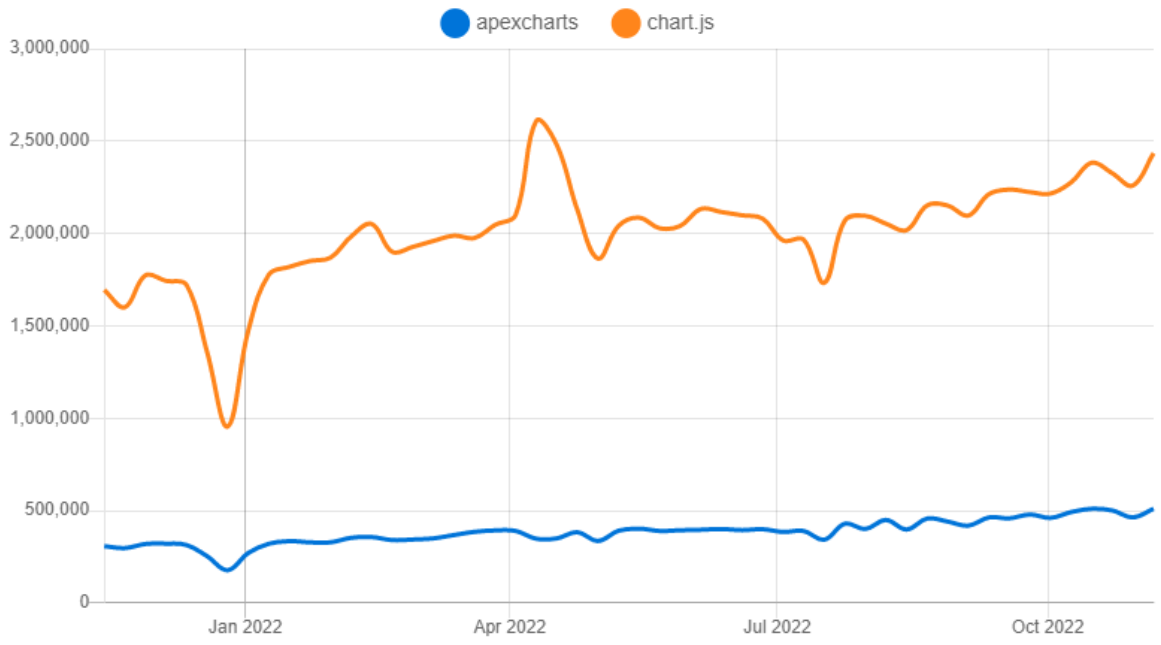
Työkalujen käyttämien HTML-elementtien perusteella valinta kallistui Chart.js:n puoleen. Datan määrä oli tässä kohtaa kehitystä hieman epävarmaa, joten oli perustellumpaa valita työkalu, joka soveltuu paremmin suuremman määrän käsittelyyn. Suorituskyvyn riskeeraaminen ei ollut missään nimessä suotavaa. Työkalulla on myös keinoja parantaa canvas-elementin käyttöön liittyviä haasteita.

Valintaa tehdessä tutkittiin myös työkalujen sopivuutta ottaa käyttöön erilaisissa projektissa. Kun työkaluja ja ohjelmistokirjastoja, eli paketteja, otetaan mukaan projektiin, on syytä tarkastella niitä ja niiden piirteitä tarkemmin. Huomioitavia ominaisuuksia ovat esimerkiksi ladattavan paketin suosio ja kehitykseen osallistuvien henkilöiden määrä, päivittymistahti sekä paketin koko. Paketin olisi suotavaa olla yhteensopiva projektissa jo olevien pakettien kanssa. (Aharoni 2022.)

Paketin suosio, kehittäjien määrä sekä säännöllinen päivitysten julkaisu viestivät paketin kehityksen jatkuvuudesta. Paketti, jolla on paljon käyttäjiä ja kehittäjiä mukana toiminnassaan menestyy ja vahvistuu todennäköisesti myös tulevaisuudessa. Tällaisilla paketeilla on paremmat valmiudet pysyä ajankohtaisina ja suosittuina, sekä kehittyä webkehityksen trendien, uusien teknologioiden ja standardien mukana. Paketin laatuun voi luottaa helpommin. (Aharoni 2022.)

Paketin koko on tärkeää huomioida valintaa tehtäessä. Projekteilla ja sovelluksilla on yleensä useampia paketteja ladattu tukemaan kehitystä ja toimintaa. Kaikki projektissa mukana olevat paketit lisäävät projektin kokonaiskokoa. Jos kokonaiskoko kasvaa liian suureksi, voi se vaikuttaa sovelluksen suorituskykyyn alentavasti. (Aharoni 2022.) Angular-kehystä käytettäessä on mahdollista määritellä moduulit ja moduulien käyttämät paketit latautumaan vasta, kun niitä tarvitaan. Pakettien valinnan suhteen eri näkökulmia ja keinoja on hyvä kartoittaa, sillä kaupallisissa sovelluksissa suorituskyvyllä on erityisen suuri merkitys.

Chart.js ja ApexChartsilla paketteina on merkittäviä eroja. Chart.js on kahdesta tunnetumpi, ja sitä myötä sillä on vakiintuneempi asema pakettimarkkinoilla sekä todennäköisemmin vakaampi toimintakyky työkaluna. Työkalulla on enemmän latauksia, kuten kuviossa 1 nähdään. Chart.js mainostaakin olevansa suosituin visualisointityökalu webkehityksessä latausmääriensä perusteella (Chart.js 2022). Päivitysten julkaisunopeudessa paketit sijoittuvat lähekkäin kuin myös tunnettujen koodiongelmien määrässä. Chart.js on pakettikooltaan kuitenkin huomattavasti pienempi kuin ApexCharts.



Kuvio 1. Työkalujen latausmäärät viimeisen vuoden aikana (Npm Trends 2022)

Lopulta kehitykseen valittiin Chart.js. Molemmat työkalut ovat ominaisuuksiltaan hyvin samanlaisia, ja ApexCharts joissain tilanteissa jopa kattavampi. Suorituskyvyltään ja pakettilaadultaan Chart.js oli kuitenkin huomattavasti perustellumpi valinta kaupalliseen projektiympäristöön.

4.3 Datan yksinkertaistaminen

Visualisoinnin suunnittelun ja työkalun valinnan jälkeen perehdyttiin datan käsittelyyn. Visualisoitavien kaavioiden tunnistaminen määritteli, mitä dataa tarkalleen tarvittiin niiden toteuttamiseen, ja työkalun valinta kohdisti tietynlaisia vaatimuksia siihen, millaisessa muodossa datan tulee olla.

Datan käsittelyä varten haluttiin ottaa käyttöön jokin erillinen tietovarasto tai käsittelyputki visualisoitavalle datalle. Visualisointiin tarvittava data voitaisiin tietenkin hakea suoraan päätietokannasta aina tarpeen mukaan, mutta se voisi mahdollisesti olla hyvin raskasta

suorituskyvyn kannalta. Tarvittava data on jakaantunut useampaan tietokantatauluun, ja haut, jotka kohdistuvat useampaan tauluun samanaikaisesti ja sisältävät monimutkaisia ehtoja tai suodatuksia, voivat olla erittäin raskaita. Datan visualisointi ominaisuutena haluttiin suunnitella niin, että se toimisi kestävästi myös jatkossa. Ominaisuuden jatkokehitys ja laajentaminen koskemaan muuta dataa voisi olla todella hankalaa, jos toiminta on heti alusta alkaen hieman epävakaata. Jatkokehityksessä halutaan keskittyä enimmäkseen olennaiseen, joten on ehdottoman tärkeää suunnitella toimintoa jo nyt kestävyyden ja jatkuvuuden kannalta.

Suorituskyvyn takaamisen näkökulmasta päädyttiin siihen, ettei visualisoitavan datan tarvitse olla reaaliaikaista. Reaaliaikainen data on saatavilla mScalesissa muualla, joten sen tuominen osaksi visualisointeja ei ollut välttämätöntä. Visualisoitava data voitaisiin prosessoida aina yön aikana, jolloin data edelliseltä päivältä olisi visualisointityökalun käytettävänä. Näin kuormittavat tietokantahaut saataisiin eristettyä kiireellisimmistä toimistoajoista, jolloin mScalesilla on tiedetysti eniten käyttäjiä. Suorituskyvyn kannalta tämä olisi myös kestävä valinta.

Yksiselitteinen ratkaisu oli ottaa päätietokannan tueksi toinen, pelkästään visualisointidatalle kohdistettu tietovarasto. Kerran yössä voitaisiin ajaa jokin ohjelma, joka hoitaisi datan haun päätietokannasta, muokkaamisen eli yksinkertaistamisen ja putsaamisen sekä siirtämisen tähän uuteen, erilliseen tietovarastoon. Ratkaisu olisi selkeä ja ehkä vaatimaton, mutta pohjimmiltaan toimiva.

Ylläpidettävyyden kannalta voisi olla haasteellista, jos ajettava ohjelma toteutettaisiin alusta alkaen itse. Lähde- ja kohdevarasto ovat molemmat alttiita muutoksille, jolloin myös ajettavaa ohjelmaa tulisi muokata aina muutosten mukaisesti. Visualisoitavien ilmiöiden lisääntyessä ohjelmakoodia tulisi lisätä tai korjata ottamaan huomioon nämä muutokset. Tietenkään kaikkiin muutoksiin ei pystytä varautumaan, mutta niihin valmistautuminen tai ehkäiseminen olisi suotavaa.

Liiketoimintatiedon hallinnassa perinteisesti käytetyt ETL- ja ELT-prosessit olivat potentiaalinen ratkaisu datan käsittelyn putken käyttöönotossa. Kumpikaan prosesseista eivät lähtökohtaisesti ole suunniteltu tämän kaltaisiin tilanteisiin, mutta prosessit täyttävät kehityksessä vaadittavat tavoitteet ja kriteerit hyvin. Prosessit ovat yleensä helposti muokattavissa, joten ETL- tai ELT-prosessin räätälöinti tämän työn tarpeisiin pitäisi olla mahdollista.

Dataputken kehitykseen voitaisiin siis ottaa käyttöön uusi ETL- tai ELT-prosessityökalu ja uusi tietovarasto käsiteltyä dataa varten. Prosessityökalun valinta edellytti ennen kaikkea sen yhteensopivuutta olemassa olevan lähdevaraston kanssa, joka tässä tapauksessa oli mScalesin päätietokanta. Muilta osin sen haluttiin edistävän käsittelyn automatisointia ja

ylläpidettävyyttä. Kohdetietovaraston suhteen oli enemmän joustovaraa. Tärkeitä kriteerejä kohdevaraston valinnassa olivat sen sujuva käytettävyys webkehitysympäristössä sekä sen skaalautuvuus ylläpidettävyyden, suorituskyvyn ja jatkokehityksen kannalta.

Koska mScalesin alustan ja kehitysympäristön toteuttamisessa käytetään hyödyksi AWS:n erilaisia palveluita jo ennestään, oli luontevaa perehtyä ensin kyseisen palveluntarjoajan ratkaisuihin. AWS:n tarjoaman ratkaisun valinta voisi todennäköisesti olla helpompaa, sillä sen käyttöönotossa voitaisiin hyödyntää mScalesin muiden kehittäjien kokemusta muiden AWS:n palveluiden käyttöönotosta.

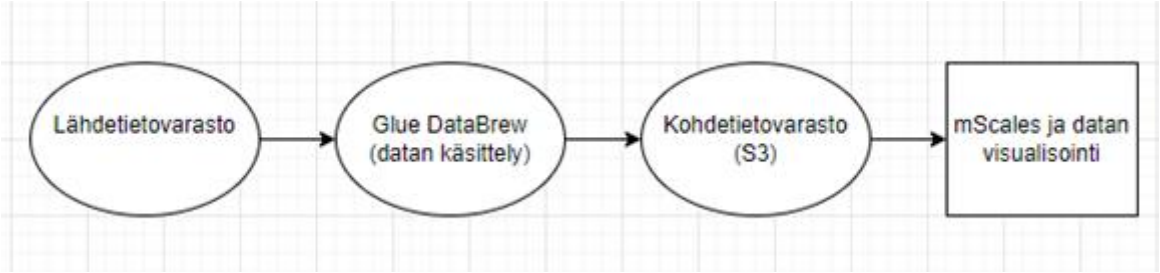
AWS tarjoaa ETL-prosessin toteuttamiseen Glue-työkalun, ja sen johdannaistyökalut Glue Data Catalog ja Glue DataBrew. Kaikilla kolmella palvelulla on yhdessä laaja tuki AWS:n eri natiiveja tietovarastoja varten, oli kyseessä sitten lähde- tai kohdevarasto. Vaihtoehdoista löytyi myös tarvittava tuki päätietokannalle, joten työkalua päädyttiin testaamaan tarkemmin.

Ensimmäinen suunta toteuttaa dataputki oli ottaa prosessia varten Glue-työkalu, ja kohdevarastoksi AWS:n DynamoDB, joka on NoSQL-tietokanta. Valintaa perusteltiin NoSQL-tietokantojen sopivuudella käsitellä pienempää sekä suurempaa määrää dataa yhtä suorituskykyisesti. Relaatiotietokantojen suorituskyky ja skaalautuvuus ovat alttiita ongelmille erityisesti silloin, kun varastoitavaa dataa on paljon. Keinoja toiminnan optimointiin on useita, mutta nekin ovat lopulta rajallisia. Koska datan määrä ja muoto olivat suunnittelun aikana hieman epäselviä, oli luonnollista perehtyä vaihtoehtoihin, jotka skaalautuisivat paremmin myös suuremman datamassan suhteen.

NoSQL-tietokantojen suunnittelu vaatii usein hyvin erilaista lähestymistapaa verrattuna perinteisiin relaatiotietokantoihin. Esimerkiksi tietokantahakujen käyttötapauksia tulee usein pohtia tarkemmin, joten suunnittelutyöhön olisi täytynyt käyttää paljon aikaa ja resursseja. NoSQL-tietokantoihin voidaan kuitenkin tallettaa dataa vapaamuotoisemmin kuin relaatiotietokannoissa, joten visualisointien muuttuessa tai lisääntyessä datan tallettamisessa olisi silti ollut joustavuutta.

Glue-työkaluja testaamalla yksinkertaisimmalta vaihtoehdolta vaikutti lopulta Glue DataBrew. Työkaluun sisäänrakennetut valmiit muunnokset palvelivat tavoitteita parhaiten. Kaikki kolme työkalua olivat kokeilun tuloksena johdonmukaisia käyttää, ja ne tarjosivat keskenään hieman erilaisia toiminnallisuuksia datan käsittelyyn. Datan käsittely ilman koodia onnistui kaikilla kolmella työkalulla, kun taas koodin avulla määrittely onnistui rajallisemmin. Glue-työkaluja voisi joissain tilanteissa käyttää useampaa samaan aikaan datan käsittelyssä.

Glue DataBrew -työkalun käyttöön päätyminen selkeytti dataputken kehittämistä. Glue DataBrew tukee pelkästään AWS Simple Storage Service eli S3:n käyttöä kohdetietovarastona, joten siltä osin valinta oli helppo. S3 on pilvipohjainen tietovarasto objektimuotoiselle datalle, ja se on suunnattu monipuolisesti eri käyttötarkoituksia ja määrää dataa varten. Dataa varastoidaan käyttäjän hallitsemissa ämpäreissä (buckets). (Amazon b). Kuvassa 9 on esitelty lopulliset dataputken vaiheet ja eteneminen suhteessa toisiinsa. Visualisointidatan varastointiin ei lopulta otettu käyttöön tietokantaa ollenkaan.



Kuva 9. Suunniteltu prosessi datan käsittelyyn

Glue DataBrew -työkalu tarjoaa datan käsittelyyn yli 250 sisäänrakennettua valmista funktiota ja muunnosmetodia. Dataa voi muuntaa esimerkiksi poistamalla invalideja arvoja tai suodattamalla ehtojen perusteella. Yhdistämällä ja järjestelemällä funktioita voidaan luoda reseptejä, joita käytetään tehtävien (jobs) luonnin perustana. Tehtäviä ajamalla Glue DataBrew itsenäisesti hakee datan, muuntaa sitä reseptin perusteella ja tallettaa sen kohdevarastoon valitussa muodossa. Tehtäviä voi ajastaa ajamaan automaattisesti.

Chart.js tukee muutamaa eri tietotyyppiä datalähteenä, mutta laajimman tuen työkalu tarjoaa juuri objektimuotoiselle datalle. Tämän perusteella oli luonnollista päätyä tallettamaan Glue DataBrew -työkalulla muokattu data JSON Lines -muotoisena, sillä perinteistä JSON-muotoa työkalu ei tue. Glue DataBrew tukee myös muita tiedostomuotoja, mutta webkehityksen näkökulmasta JSON-muotoa muistuttava on yksiselitteinen valinta. JSON-dataa voidaan loogisesti muokata ja pilkkoa JavaScriptin valmisfunktioilla.

Rivikohtainen data ei muotoilematta sovi Chart.js:n käytettäväksi, joten sitä täytyy käsitellä vielä käyttöliittymän puolella visualisointityökalulle sopivaksi. Jokainen visualisoitava kaavio tarvitsee erikseen visualisoitavan datan, otsikot ja tunnisteet sekä muun oheistiedon. Muunnokset eivät kuitenkaan ole vaikeita.

5 Toteutusvaihe

5.1 Ominaisuuden luonti

Suunnitteluvaiheen tuloksena saatiin määriteltyä tarkemmat kriteerit työvälineiden valintaa varten, ja niiden avulla valittua välineet visualisointiominaisuutta varten. Toteutusvaiheessa työkalujen toimivuutta ja soveltuvuutta testattiin luomalla malliratkaisu datan visualisoinnista. Luotua komponenttia voidaan toteutuksen jälkeen arvioida, ja sen perusteella todentaa suunnittelutyön onnistuminen. Toteutusvaiheen tuloksena tehtyä mallia voidaan hyödyntää myös jatkokehityksen perustana.

AWS-resurssit, eli Glue DataBrew ja kohdevarasto S3, luotiin käyttäen hyödyksi alustatyökalu Terraformia ja AWS:n omaa hallintapaneelia. Terraformin avulla alustaan liittyviä resursseja voidaan määritellä ja muokata kooditiedostojen kautta, jolloin niitä voidaan käyttää uudelleen helpommin sekä tuoda ne osaksi versionhallintaa. (Terraform).

Glue DataBrew -työkalua käytettiin suoraan hallintapaneelista, sillä Terraformissa ei tällä hetkellä ole saatavilla tukea Glue DataBrew -resurssien määrittelyyn. Glue DataBrew -työkalulla luotiin ensin resepti, jonka perusteella työkalu hakee datan ja muokkaa sitä visualisointitarpeiden perusteella. Reseptissä esimerkiksi poistettiin turhia sarakkeita ja tyhjiä arvoja, muunnettiin aikaleimojen aikavyöhykkeitä ja normalisoitiin arvoja. Samassa reseptissä pystyttiin hakemaan ja muuntamaan data kaikista tarvittavista tietokantatauluista yhdeksi kokonaisuudeksi. Vain sarakkeet ja data, joita käytettiin jollain tavalla käyttöliittymässä, jätettiin osaksi käsiteltyä dataa.

Työkalua käytettiin tuotantodataa vastaavalla testidatalla ja tehtävät ajettiin pääasiassa manuaalisesti. Tehtävien ajastamista kokeiltiin myös onnistuneesti. Ajoajat olivat vain muutamia minuutteja. Glue DataBrew -työkalun tehtävien ajoa pystyttiin tarkentamaan niin, että käsitelty data voitin jakaa osiin datassa olevan sarakkeen mukaan. Ajon aikana syntynyt data jaettiin yrityksen id:n perusteella. S3-ämpärin kansiorakenne muodostui tämän osituksen perusteella. Osittava sarake vaikuttaa syntyvien tiedostojen nimeämiseen, joten päästäkseen kansion sisällä olevaan dataan tulee tietää yrityksen id ja ajetun tehtävän nimi.

S3-ämpäri ja sen ominaisuudet määriteltiin kokonaisuudessaan Terraformilla, kuin myös sen ja Glue DataBrew -työkalun käyttöön liittyvät oikeudet. AWS-palveluihin pääsy on lähtökohtaisesti aina estetty muilta palveluilta tai käyttäjiltä, ellei pääsyä ole erikseen annettu. Glue DataBrew -tehtäville annettiin S3-kohdeämpärille muokkausoikeus. Ämpärin sisältö jätettiin muiden AWS-palveluiden nähtäväksi.

Datan visualisointia varten luotiin uusi Docker-kontti mikropalveluarkkitehtuurin mukaisesti. Kontille määriteltiin tarvittavat oikeudet ja muuttujat, joita se tarvitsee ollakseen yhteydessä AWS:n palveluihin. Kontin käyttöönotto mahdollisti sen, että ominaisuus ja siihen liittyvä koodi saatiin tällä tavoin osittain eristettyä omaksi kokonaisuudeksi. Riski häiritä tai lamaannuttaa muita sovelluksen toimintoja vähentyi, ja ominaisuuden testaaminen olisi hieman yksinkertaisempaa kehityksen jatkuessa myöhemmin.

Perinteisesti datan hakeminen tapahtuisi kontin kautta. Tässä tapauksessa haku kuitenkin etenee niin, että kontti hakee ja palauttaa käyttöliittymälle ennalta määrätyn URL-osoitteen (presigned url), josta data voidaan ladata. URL-osoite haetaan kontissa käyttäen AWS S3:n valmiita ohjelmistokehitysrajapintoja. Data ladataan vasta käyttöliittymän puolella eikä haettua dataa tarvitse näin kierrättää useamman palvelimen kautta. Haun onnistuminen käyttöliittymän kautta vaatii sen, että S3-ämpärissä sallittiin datan hakeminen ulkopuolisesta lähteestä. Salliminen toteutettiin määrittelemällä ämpärille CORS-menettelypolitiikka.

Sallituksi lähteeksi määriteltiin paikallinen kehitysympäristö ja sallituiksi metodeiksi ainoastaan datan haku. Muut CORS-asetukset jätettiin määrittelemättä. Visualisointidataa ei siis pysty muokkaamaan tai poistamaan käyttöliittymän kautta lainkaan. Hakua varten tarvitaan myös tarkempaa tietoa kohdeämpäristä. Kohdeämpärin ja haettavan tiedoston nimi täytyy olla tiedossa.

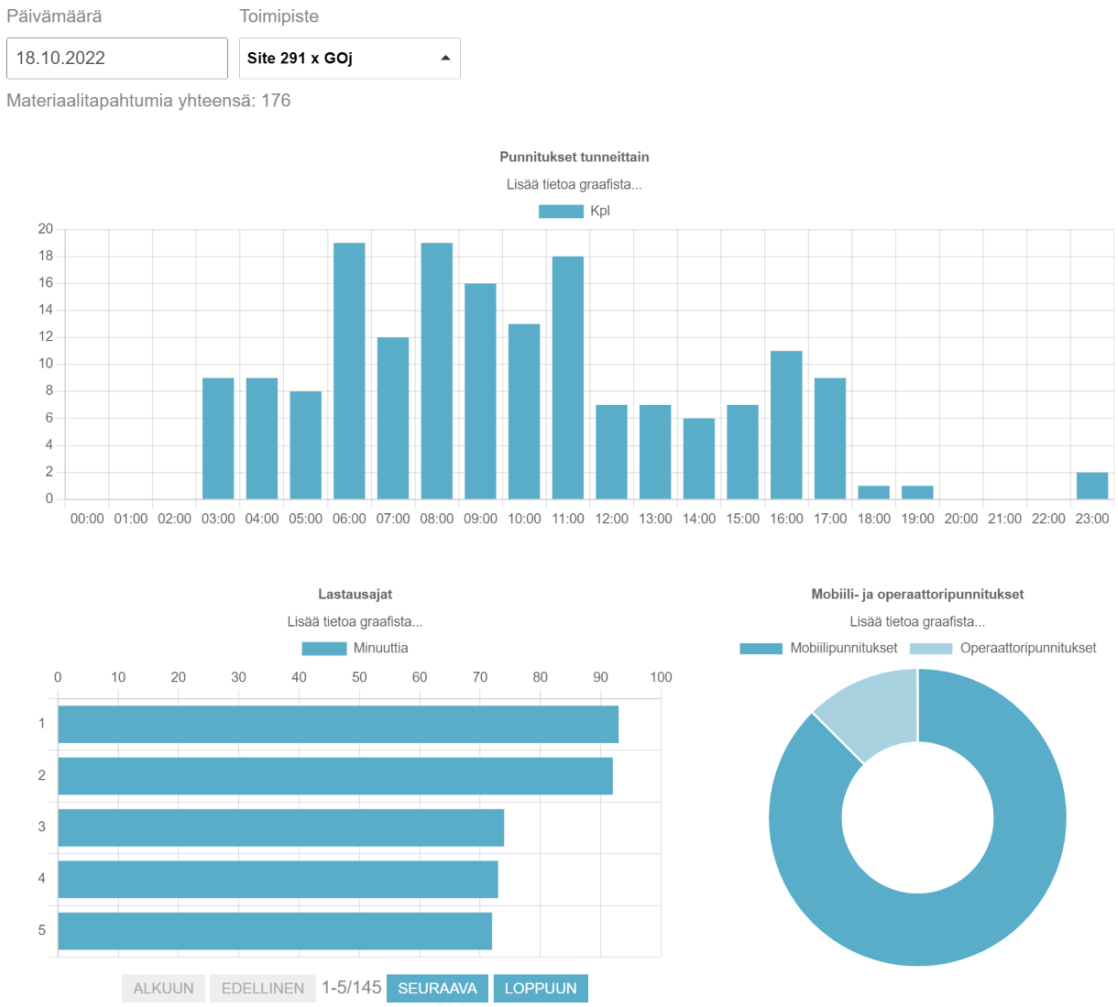
Datan haun ja vastaanoton jälkeen sitä täytyi käsitellä vielä Chart.js:n käyttöön sopivaksi, joten käsittelyä varten luotiin joukko rajapintoja, jotka näkyvät kuvassa 10. Rajapinnat luotiin ennen kaikkea suodattamaan jokaisen kaavion tarvitsema data, mutta myös lajittelemaan sitä. Suuruus- tai aikajärjestyksessä oleva data on helpompi havainnoida.

```
sortByTime(data: any[])

sortByLocation(data: any[])

sortByLoadingTime(data: any[])

filterByTime(data: any, startTime: string, endTime: string)

filterByLocation(data: any, locationId: number)

groupByHour(data: any)

calculateLoadingTimes(data: any)

calculateOperatorWeighingCodeRatio(data: any)
```

Kuva 10. Datan käsittelyn rajapinnat käyttöliittymässä

Ominaisuudesta haluttiin luoda jo toteutusta testatessa komponenttipohjainen ratkaisu. Komponentin toivottiin olevan mahdollisimman helppo siirtää tai ottaa käyttöön myös muualla käyttöliittymässä, joten komponentti toteutettiin mahdollisimman omavaraiseksi riippuvuuksien suhteen. Datan visualisoinnit luotiin niin, että kaikki kaaviot ovat näkyvillä yhtä aikaa. Komponenttiin lisättiin lomake, jonka avulla voidaan vaihtaa visualisoinnissa käytettävää dataa. Näytettävää visualisointia voi muuttaa päivän ja toimipisteen perusteella dynaamisesti, joihin käytettiin apuna kuvan 10 rajapintoja. Lomakkeen päivämäärävalitsin määriteltiin tässä kohtaa kehitystä tukemaan vain vuorokauden pituisia aikavälejä.

Visualisoidut kaaviot luotiin Chart.js:n rajapintojen avulla käyttäen työkalun dokumentaatiota tukena. Rajapintojen ja työkalun Angular-integraation avulla kaavioiden määrittely, päivittäminen ja uudelleen piirtäminen onnistuivat johdonmukaisesti ja yksinkertaisesti. Kaavioita pystyttiin muokkaamaan yksittäin ja yhteisesti esimerkiksi ulkoasuun liittyvissä asioissa. Canvas-elementit, joiden sisällä kaaviot olivat yksittäin, saatiin toimimaan responsiivisesti ja näyttämään selkeiltä myös pienemmillä näytöillä. Älykkäitä ominaisuuksia testattiin lisäämällä kaikkiin kaavioihin datan lisäys- ja poistumisvaiheisiin valmiit animaatiot sekä työkaluvihjeet, joista selviää lisätietoa tietystä datapisteestä- tai joukosta kaaviossa. Kuvasta 11 näkee, miltä komponentti näytti toteutusvaiheen aikana. Ulkonäöllisiin yksityiskohtiin ei kiinnitetty paljoa huomiota testauksen aikana.



Kuva 11. Datan visualisoinnin komponentti

5.2 Käyttöönotto ja jatkokehitys

Testauksen aikana ei ehditty eikä edes tarvinnut ottaa huomioon kaikkia perinteisen websovelluksen komponentin tai ominaisuuden luontiin liittyviä osa-alueita. Toteutusvaiheen aikana kuitenkin havaittiin ja merkittiin ylös joitakin osa-alueita, joihin olisi kannattavaa kiinnittää vielä huomiota. Monia asioita tulisi vielä selvittää, ennen kuin ominaisuuden voisi viedä eteenpäin kohti tuotantoympäristöä.

Ominaisuuden ulkoasua tulee pohtia tarkemmin, jotta siitä saadaan täysin yhteensopiva mScalesin käyttöliittymän kanssa. Värimaailmaa ja sommittelua on hyvä miettiä myös ilmiön esiintuomisen parantamisen kannalta. Canvas-elementtiin liittyvät saavutettavuusvaikeudet täytyy pohtia tarkemmin ja hyödyntää Chart.js:n suosituksia käsitellä siihen liittyviä haasteita vielä lisää.

Visualisoitavaa dataa ja sen käsittelyä olisi aiheellista tarkastella vielä uudestaan. Datan käsittely Glue DataBrew -työkalulla kestää vain vähän aikaa, mutta sitä voi prosessin

valmistumisen jälkeen olla kokonsa puolesta hyvinkin paljon. Datan hakupyyntöjen käsittelyn nopeus ja suorituskyky käyttöliittymässä on hyvä tiedostaa nykyisellään, ja tutkia mahdollisuuksia, miten hakuja voisi optimoida. Glue DataBrew tukee JSON Lines -muotoista tallennusmuotoa, joka ei sellaisenaan ole heti JavaScript-ympäristöissä käytettävissä. Muunnos perinteiseen JSON-muotoon ei ole hankalaa, mutta suurella datamäärällä mahdollisesti tehokkuutta heikentävä. Muodonmuutoksen toteutus AWS:n palveluiden avulla valmiiksi ennen hakuja voisi olla potentiaalinen kehityskohde.

Datan käsittelyyn liittyvät toiminnot käyttöliittymässä olisi hyvä käydä läpi. Datan käsittely käyttöliittymässä vie aikaa ja voi vaikuttaa suorituskykyyn negatiivisesti. Osa tai mahdollisesti jopa kaikki käsittelytoiminnoista voitaisiin mahdollisesti siirtää AWS:n Lambda-funktioiden kautta suoritettaviksi. Funktioita voitaisiin kutsua esimerkiksi jokaisen Glue DataBrew -tehtävän ajon jälkeen tai vaihtoehtoisesti aina silloin, kun dataa haetaan S3-ämpäristä. Ämpärin asetuksista voidaan määritellä suoritettavia funktioita tiettyihin datanhakumetodeihin.

6 Yhteenveto

Opinnäytetyön tavoitteena oli konseptitasolla suunnitella ja testata toteuttaa ominaisuus punnitusdatan visualisoinnista. Työtä varten tarkasteltiin ja vertailtiin eri vaihtoehtoja ja työkaluja toteuttaa kokonaisvaltainen ominaisuus, johon liittyi keskeisesti datan käsittely, varastointi sekä itse visualisointi. Ominaisuuden haluttiin olevan kevyt ylläpitää ja mahdollisimman automatisoitu.

Visualisoitavaa dataa, haluttuja visualisoitavia ilmiöitä ja tarvittavia askeleita datan käsittelyn ja varastoinnin suhteen tarkasteltiin huolellisesti. Tarkastelun tuloksien perusteella valittiin työkaluja toteutuksen tueksi. Datan käsittelyyn ja varastointiin valikoitui AWS:n tarjoama ETL-prosessiin pohjautuva Glue DataBrew ja pilvipohjainen tietovarasto S3. Visualisointiin valittiin JavaScript-kirjasto Chart.js.

Tehdyn työn avulla haluttiin todentaa ja arvioida ominaisuuden toimivuutta. Toteutuksen testaamisen perusteella voidaan todeta, että valitut työkalut sopivat ominaisuuden implementointiin hyvin. Ominaisuus oli konseptitasolla onnistunut, ja työn aikana saatu tulos soveltuu jatkokehityksen perustaksi. Valitut työkalut tarjosivat yhdessä ratkaisuja luoda ylläpidettävä ja suorituskykyinen prosessi käsitellä, varastoida ja visualisoida dataa selkeiden rajapintojen kautta koko prosessin laajuisesti. Valmiiden ratkaisujen hyödyntäminen tehostivat ajan- ja resurssien käyttöä ja tarjosivat näkemyksiä siitä, miten valittuja työkaluja ja prosessia voisi hyödyntää myös muualla mScalesin kehityksessä.

Seuraava askel visualisointiominaisuuden kehityksessä on arvioida ja kerätä palautetta yrityksen sisäisesti, ja määritellä niiden mukaan seuraavat vaiheet. Annetun palautteen ja toteutuksen aikana ilmenneiden havaintojen perusteella kehitettävää riittää vielä varmasti. Visualisoitavia ilmiöitä on varmasti lisää, ja komponenttipohjaista ratkaisua voi kehittää dynaamisempaan suuntaan.

Lähteet

Aharoni, N. 2020. How to Choose the Right NPM Package for Your Project. Viitattu 26.9.2022. Saatavissa: https://betterprogramming.pub/how-to-choose-the-right-npm-package-for-your-project-c3d1cc25285e

AWS a. AWS Glue. Viitattu 24.10.2022. Saatavissa https://aws.amazon.com/glue/

AWS b. AWS Glue DataBrew. Viitattu 24.10.2022. Saatavissa https://aws.amazon.com/glue/features/databrew/

AWS b. AWS S3. Viitattu 17.11.2022. Saatavissa https://aws.amazon.com/s3/

ApexCharts. Features. Viitattu 9.11.2022. Saatavissa https://apexcharts.com/features/

Chart.js 2022. Viitattu 9.11.2022. Saatavissa https://www.chartjs.org/docs/latest

Data Literacy. An Overview of the DIKW Pyramid. Viitattu 28.9.2022. Saatavissa: https://dataliteracy.com/topic/an-overview-of-the-dikw-pyramid/

D3. Data-Driven Documents. Viitattu 6.11.2022. Saatavissa https://d3js.org/

Gartner. Gartner Magic Quadrant. Viitattu 24.10.2022. Saatavissa https://www.gartner.com/en/research/methodologies/magic-quadrants-research

Gartner. 2022a. Gartner Magic Quadrant for Analytics and Business Intelligence Platforms. Viitattu 9.11.2022. Saatavilla https://www.gartner.com/en/documents/4012759

Gartner. 2022b. Gartner Magic Quadrant for Data Integration Tools. Viitattu 24.10.2022. Saatavissa rajoitetusti https://www.gartner.com/en/documents/4017726

IBM. IBM DataStage. Viitattu 24.10.2022. Saatavissa https://www.ibm.com/products/datastage

Kauppalehti. 2022. Yrityshaku, Lahti Precision Oy. Viitattu 16.11.2022. Saatavilla https://www.kauppalehti.fi/yritykset/yritys/lahti+precision+oy/08984550

Koponen, J., Hildén, J. & Vapaasalo T. 2016. Tieto näkyväksi – informaatiomuotoilun perusteet. Helsinki: Aalto-yliopisto.

Koponen, J., Hildén, J. & Koivunen-Niemi, L. 2019. Data visualization handbook. Helsinki: Aalto-yliopisto.

Lahti Precision Oy. Yritys. Viitattu 16.11.2022. Saatavilla https://lahtiprecision.com/yritys/

Lemonsoft. Liiketoimintatiedon hallinta (BI). Viitattu 28.9.2022. Opas Saatavissa: https://www.lemonsoft.fi/lemonbi-liiketoimintatiedon-hallinta/

Leonard, B. 2021. ETL vs ELT Explained. Grouparoo. Viitattu 3.10.2022. Saatavissa https://www.grouparoo.com/blog/etl-vs-elt

Npm Trends. 2022. ApexCharts vs. Chart.js. Viitattu 16.11.2022. Saatavissa https://npmtrends.com/apexcharts-vs-chart.js

MDN Web Docs. 2022. The Graphics Canvas element. Viitattu 12.11.2022. Saatavissa https://developer.mozilla.org/en-US/docs/Web/HTML/Element/canvas#specifications

mScales a. Digitalisoi punnitusten materiaalivirrat. Viitattu 12.9.2022. Saatavissa https://www.mscales.com/fi/modernisointi-punnituksessa

mScales b. Toiminnalliset lisämoduulit. Viitattu 10.10.2022. Saatavissa https://www.mscales.com/fi/toiminnalliset-lis%C3%A4moduulit

Ontotext. What is the Data, Information, Knowledge, Wisdom (DIKW) Pyramid? Viitattu 28.9.2022. Saatavissa https://www.ontotext.com/knowledgehub/fundamentals/dikw-pyramid/

Salo, I. 2013. Big data - tiedon vallankumous. Jyväskylä: Docendo.

SAS. ETL – What it is and why it matters. Viitattu 19.10.2022. Saatavissa https://www.sas.com/en_us/insights/data-management/what-is-etl.html

Statista. 2021. Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025. Viitattu 17.10. Saatavissa rajoitetusti https://www.statista.com/statistics/871513/worldwide-data-created/

Stitch a. Viitattu 9.10.2022. Saatavissa https://www.stitchdata.com/

Tableau. What Is Data Visualization? Definition, Examples, And Learning Resources. Viitattu 9.11.2022. Saatavilla https://www.tableau.com/learn/articles/data-visualization#advantages-disadvantages

Talend a. Integrate data. Viitattu 9.10.2022. Saatavissa https://www.talend.com/products/integrate-data/

Talend b. What is ELT? Viitattu 9.10.2022. Saatavissa https://www.talend.com/resources/what-is-elt/

Talend c. What is ETL? Viitattu 9.10.2022. Saatavissa https://www.talend.com/resources/what-is-etl/

Terraform. What is Terraform? Viitattu 19.11.2022. Saatavissa https://developer.hashicorp.com/terraform/intro

Valtiovarainministeriö. Tietojohtaminen. Viitattu 15.10.2022. Saatavissa https://vm.fi/tietojohtaminen