

**SAVONIA**

ammattikorkeakoulu

OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO  
TEKNIIKAN JA LIIKENTEEN ALA

# YRITYKSEN TOIMINNAN SEURAN- TANÄKYMÄ

Datan käyttö teollisuudessa

TEKIJÄ    Jonna Räsänen

Koulutusala Tekniikan ja liikenteen ala	
Tutkinto-ohjelma Tietotekniikan tutkinto-ohjelma	
Työn tekijä Jonna Räsänen	
Työn nimi Yrityksen toiminnan seurantanäkymä, Datan käyttö teollisuudessa	
Päiväys 4.5.2023	Sivumäärä/Liitteet 30/0
Toimeksiantaja/Yhteistyökumppani Lapinlahden Koneistus Oy	
<p>Tiivistelmä</p> <p>Opinnäytetyön tarkoituksena oli tutustua, miten dataa on mahdollista käyttää yrityksen toiminnan parantamiseksi ja tämän pohjalta kehittää yrityksen datasta analyyttisiä työpöytiä yrityksen toiminnan seurantaan. Opinnäytetyön aihe oli suoraa jatkoa Tietotekniikkaprojekti -kurssilla jo yritykselle tehdystä analytiikkanäkymästä. Työn tavoitteena opinnäytetyön tekijän osalta oli syventyä syvemmälle yrityksen dataan ja kehittää analytiikan näkymiä eteenpäin. Yrityksen tavoitteena oli saada lisää tietoa yrityksen toiminnasta helposti tulkittavassa muodossa.</p> <p>Projekti toteutettiin toukokuun 2022 ja joulukuun 2022 välisenä aikana. Pääsääntöisenä toteutustekniikkana toimi Grafana -visualisointityökalu, jolla itse analytiikkanäkymä on toteutettu. Tämän lisäksi datan hakeminen tapahtui C#-koodilla tehdyllä sovelluksella, joka hakee Azuren DataFactorysta yrityksen dataa ja siirtää sen SQL-tietokantaan. Yrityksen johtoportaan työntekijät olivat tiiviisti mukana näkymän suunnittelussa ja heidän mielipiteensä olivat suuressa roolissa työn etenemisessä.</p>	
Avainsanat Grafana, analytiikka, data, visualisointi, kaaviot	

Field of Study Technology, Communication and Transport	
Degree Programme Degree Programme in Information Technology	
Author Jonna Räsänen	
Title of Thesis Performance Analytics Dashboard, Data Usage of an Industrial Company	
Date 4.5.2023	Pages/Appendices 30/0
Client Organisation /Partner Lapinlahden Koneistus Oy	
<p><b>Abstract</b></p> <p>The purpose of the thesis was to get to know the data of the company and develop analytic dashboards for analysing the company's performance. The theme of the thesis was direct continuation from Tietotekniikkaprojekti -course where an analytics dashboard had already been made for the company. The goal of the project for the student was to gain a deeper knowledge of handling the company's data and to develop the analytics dashboards further. The goal of the company was to get more knowledge about the performance of the company in an easily read format.</p> <p>Project was executed between May and December of 2022. The main technique used was visualisation tool called Grafana, which was used for the creation of the analytics dashboard. C# programming was used to search for the data from the Azure DataFactory and for transferring the data to an SQL database. The heads of the target company were a highly involved in the planning of the project and their ideas and opinions had a big role in the progress of the project.</p>	
<b>Keywords</b> Grafana, analytics, data, visualisation, graphs	

## SISÄLTÖ

1	JOHDANTO .....	5
2	KÄSITTEET .....	6
3	DATAN HYÖDYNTÄMINEN TEOLLISUUDESSA .....	8
3.1	Tietovarastointi .....	8
3.2	Tiedon jalostaminen .....	9
3.3	Jalostetun datan hyödyntäminen .....	11
3.4	Miksi Business Intelligence? .....	12
4	PROJEKTIN TARKOITUS .....	13
4.1	Lähtötilanne .....	13
4.2	Toteutuksen periaate .....	13
4.3	Toteutusaikataulu .....	14
5	TEKNINEN TOTEUTUS .....	15
5.1	Grafana .....	15
5.2	Azure .....	15
5.2.1	DataFactory .....	16
5.2.2	Blob Storage .....	17
5.3	Tietokantamigraatio -sovellus .....	19
5.4	Projektin arkkitehtuuri .....	21
5.5	Muut käytetyt työkalut .....	22
6	TOIMINNAN ANALYTIIKAN SOVELLUS.....	23
6.1	Datan keräys ja käsittely .....	23
6.2	Näkymät.....	24
6.3	Grafanan visuaalinen rakenne .....	25
6.4	Huomautuksia.....	28
6.5	Jatkokehitys .....	28
7	YHTEENVETO JA POHDINTA .....	29

## 1 JOHDANTO

Tämän opinnäytetyön tavoitteena on tutustua, miten datan hyödyntäminen teollisuudessa vaikuttaa yrityksen toimintaan. Tähän teoreettiseen pohdintaan perustuen projektin tavoitteena on luoda Grafana-pohjainen analytiikkanäkymä, joka auttaa yrityksen työntekijöitä ja työnjohtajia yrityksen toiminnan seuraamisessa.

Työn tilaaja on perheyhtiö Lapinlahden Koneistus Oy, lapinlahtelainen vuonna 2005 perustettu metalliteollisuuden koneistamo, joka luo osia ja osakokonaisuuksia kone- ja laitevalmistajille. Yrityksen asiakkaisiin kuuluu yrityksiä, kuten Ponsse Oyj, sekä Normet Oy. Yritys kuvailee itseään avoimena ja inhimillisenä sekä työhyvinvointia kannattavana. Heidän yritysmodellinsa kannattaa yksittäisten yritysten erityistoiveiden toteuttamista ja asiakastyytyväisyys on yrityksen arvojen kärkiasemassa. Yrityksen alkumatka vastaa hyvin monen vastaavan yrityksen alkujuuria, sillä yritys on syntynyt irrottautulla Lametal Oy nimisestä yrityksestä. Nykyään yritys työllistää 17 henkilöä. (Lapinlahden Koneistus, ei pvm.)

Opinnäytetyötä varten yrityksen toiveena oli saada helposti luettavassa muodossa esille heidän osia valmistavien laitteiden toimintaan liittyvää dataa. Työn suunnitelma lähti alun perin Tietotekniikka-projekti-kurssista. Kurssin aikana kyseiselle yritykselle loimme ryhmäprojektina myös tässä projektissa käytettyjen työkalujen avulla ensimmäisen yrityksen haluaman näkymän. Työhön sisältyi näkymä yrityksen päivän aikana toteutuneesta työstä, edellisen päivän työ, sekä päivän jättämä tunteina. Opinnäytetyö on suoraa jatkokehitystä ryhmäprojektin tuotokselle ja se käsittelee pääsääntöisesti samaa dataa, mutta eri näkökulmista, kuin ryhmäprojekti.

## 2 KÄSITTEET

**.NET:** Microsoftin kehittämä ohjelmistokomponenttikirjasto.

**API:** Application Programming Interface, ohjelmointirajapinta, jolla voi hakea tietoja kolmansien osapuolten sivustoilta määriteltyjen menetelmien mukaisesti.

**Analytiikkanäkymä:** Näkymä, joka esittää analyysin käyttäen hyväksi erilaisia kaavioita.

**Anturi:** mittalaitteen osa, joka lähettää dataa, joka on riippuvaista sen ympäristön muutoksista.

**Avoin lähdekoodi:** Ohjelmiston kehitysmenetelmä, jossa käyttäjien on mahdollista lukea, käyttää, muokata, sekä jaella ohjelman lähdekoodia.

**Binääri:** 2-järjestelmä, lukujärjestelmä, jonka kantaluku on kaksi.

**Business Intelligence:** Liiketoimintatiedon hallinta, eli tiedon hankinta, tallennus ja analysointi.

**Code First:** Koodaustyyli, jossa datan rakenne määritellään ensin sovelluksessa, ennen tietokannan luomista.

**Data-analytiikka:** Prosessi, jossa analysoidaan, kerätään, järjestetään ja tallennetaan liiketoiminnan kannalta tärkeää dataa.

**Data:** Digitaalisessa muodossa olevaa, koneluettavaa tietoa.

**Dataintegraatio:** Tiedon muuntaminen ja kuljettaminen tietojärjestelmästä toiseen.

**Dynaaminen:** Mukautuva, muokattavissa oleva käyttäjän mieltymysten ja kiinnostuksen perusteella.

**ERP-järjestelmä:** Toiminnanohjausjärjestelmä, jonka tehtävänä on yrityksen toiminnan ja resursien hallinta.

**ETL-prosessi:** Prosessi datan siirtämiselle, muokkaamiselle ja lataamiselle.

**Entity Framework:** .NETin objekti-relaatiokartoituskehys.

**Formaatti:** Muoto, muotoilu.

**Graafi:** Kaavio, kuvaaja.

**IaaS:** Infrastructure as a Service, eli palvelimien ja palvelinsalien ulkoistaminen, niiden ostaminen ulkoiselta taholta.

**Kehittämismalli:** Suunnitelma projektin kehitysprosessista.

**Ketjutus:** Prosessien käynnistäminen jonottain peräkkäin tai rinnakkain.

**Migraatio:** Prosessi, jossa tieto siirretään tietojärjestelmästä toiseen.

**NoSQL:** Not only SQL, perinteisestä relaatiotietokantajärjestelmästä poikkeava tietorakenne. Tietokannan rakenne ei seuraa kiinteästi määrättyä taulukkoskeemaa, vaan rakenne on skaalautuva.

**Objekti-relaatiokartoituskehys:** Kehys, joka kartoittaa objektien välisen relaation, eli yhteyden.

**Objekti:** Eli olio, ohjelmiston rakenteellinen perusyksikkö.

**PaaS:** Platform as a Service, eli alustan ulkoistaminen palveluna. Kehitysalusta, joka on ostettu ulkopuoliselta taholta.

**Pilvipalvelu:** Tietoteknisten palveluiden tuottaminen ja toimittaminen internetin välityksellä. Yleensä käsittää tietoteknisen infrastruktuurin vuokraamista pilvipalvelun tarjoalta.

**Raakadata:** Jalostamaton tieto, jota ei ole käsitelty millään tavalla sen hetkistä tarvetta varten.

**Reaaliaikainen:** Tosi aikainen, välitön, ajan tasalla oleva.

**SQL-tietokanta:** Structured Query Language -tietokanta, eli tietokanta, jonka rakenne perustuu standardoituun kyselykieleeseen.

**Tietovarastointi:** Hajallaan eri järjestelmissä olevan datan poimintaa ja yhdistelemistä yhteen tietokantaan raportointia, analytiikkaa ja muuta käyttöä varten.

**Trigger:** Laukaisin, jokin tapahtuma, joka käynnistää prosessin.

**Virtuaalikone:** Ohjelmallisesti toteutettu tietokone.

**Visma Nova:** Visman kehittämä toiminnanohjausjärjestelmä.

**Visualisointi:** Jonkin tiedon tai asian havainnollistaminen visuaalisesti.

### 3 DATAN HYÖDYNTÄMINEN TEOLLISUUDESSA

Nykyään ja tulevaisuudessa dataa hyödynnetään teollisuudessa aina vain kasvavissa määrin. Tämä johtuu pääosin datan loputtomasti kasvavasta määrästä, sekä siitä, että sen käsittelystä tulee uusien teknologioiden avulla jatkuvasti helpompaa. Datan käsittelystä ja sen esittämisestä eri tavoin on tullut tärkeä ominaisuus minkä tahansa yrityksen liiketoimintasuunnittelussa, ja tulevaisuudessa datan tärkeys on aina vain suuremmassa roolissa.

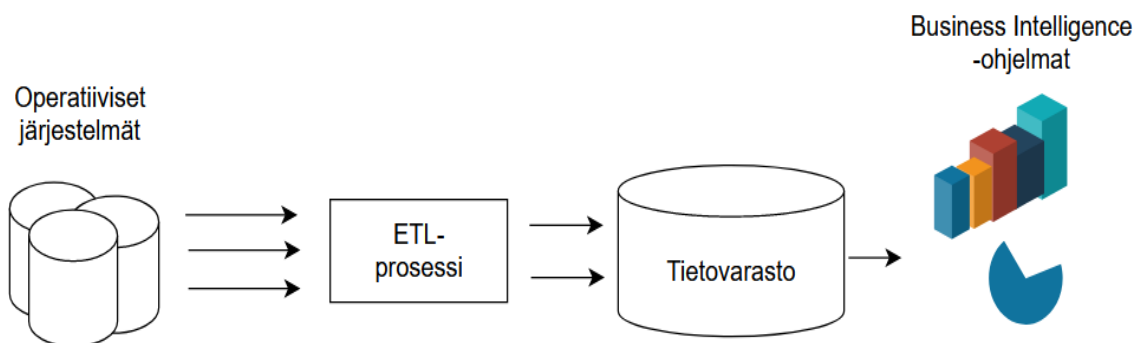
#### 3.1 Tietovarastointi

Datan hyödyntämisessä on useita hankaluuksia ja haasteita, joista ensimmäinen on tietovarastointi. Jatkuvasti kasvavat tietomassat on saatava tallennettua jonnekin, mutta jos tietokantasuunnittelu ei tehdä kunnolla, datan laatu on heikkoa, eikä kerätystä datasta ole tällöin hyötyä. Yleisellä tasolla tarkasteltuna data on yleensä joko sirpaloitu omiin siiloihinsa tai yhdistetty, eli integroitu yhtenäiseen kokonaisuuteen, jolloin hyvin laajasta datakokonaisuudesta saadaan hyvinkin täsmällistä ja helppokäyttöisempää tietoa, jota on tästä eteenpäin paljon yksinkertaisempaa jatkojalostaa halutunlaiseksi analytiikaksi. Yleisesti ottaen tätä kutsutaan Business Intelligenceksi, lyhennettynä BI. (Hovi;Hervonen;Koistinen, 2009, s. XI) Suomennettuna Business Intelligence tarkoittaa liiketoimintatiedon hallintaa.

Tietoa on mahdollista tallentaa esimerkiksi tietokantoihin, datan säilytykseen erikoistuneiden yritysten valmisohjelmistoihin, tai ERP-järjestelmiin. ERP-järjestelmä (Enterprise Resource Planning), eli toiminnanohjausjärjestelmä tarkoittaa yrityksen toimintaan ja resurssien suunnitteluun kehitettyä tietojärjestelmää. Kaikki pohjautuu siihen, että datan laadusta on pidettävä hyvää huolta ja se on pidettävä laadukkaana, jotta siitä olisi kilpailuhyötyä yritykselle. Tämä tarkoittaa, että datakokonaisuus on suunniteltava hyvin, mutta sen lisäksi sen laatua on jatkuvasti kehitettävä, sillä dataa tulee aina lisää, uusista lähteistä, sekä uusiin käyttötarkoituksiin. Hyvälaatuinen data on yritykselle arvokasta omaisuutta. (Hovi;Hervonen;Koistinen, 2009, s. 6) Sen ansiosta yrityksen on mahdollista saada huomattava kilpailuetu verrattuna yritykseen, jossa ei käytetä data-analytiikkaa hyväksi, sillä data-analytiikan avulla on mahdollista ennustaa muutoksia yrityksen toimintaan liittyen huomattavasti etujassa verrattuna, jos dataa ei analysoidaisi. Hyvälaatuisella datalla tarkoitetaan mahdollisimman tarkasti luotua ja toimivaa dataa, jossa ei ole aukkoja tai epämääräisyyksiä. Vastaavasti huonolaatuinen data on sekavaa, sisältäen hajanaisesti täytettyjä kenttiä, joiden toimintaan ei ole selkeää ohjeistusta.

### 3.2 Tiedon jalostaminen

Seuraavana mahdollisena ongelmana on tiedon jalostaminen. Data voi olla hyvinkin monessa eri muodossa eri tietolähteissä. Tiedon jalostaminen tarkoittaa erinäköisten, samaa tietoa tarkoittavien datojen yhtenäistämistä.



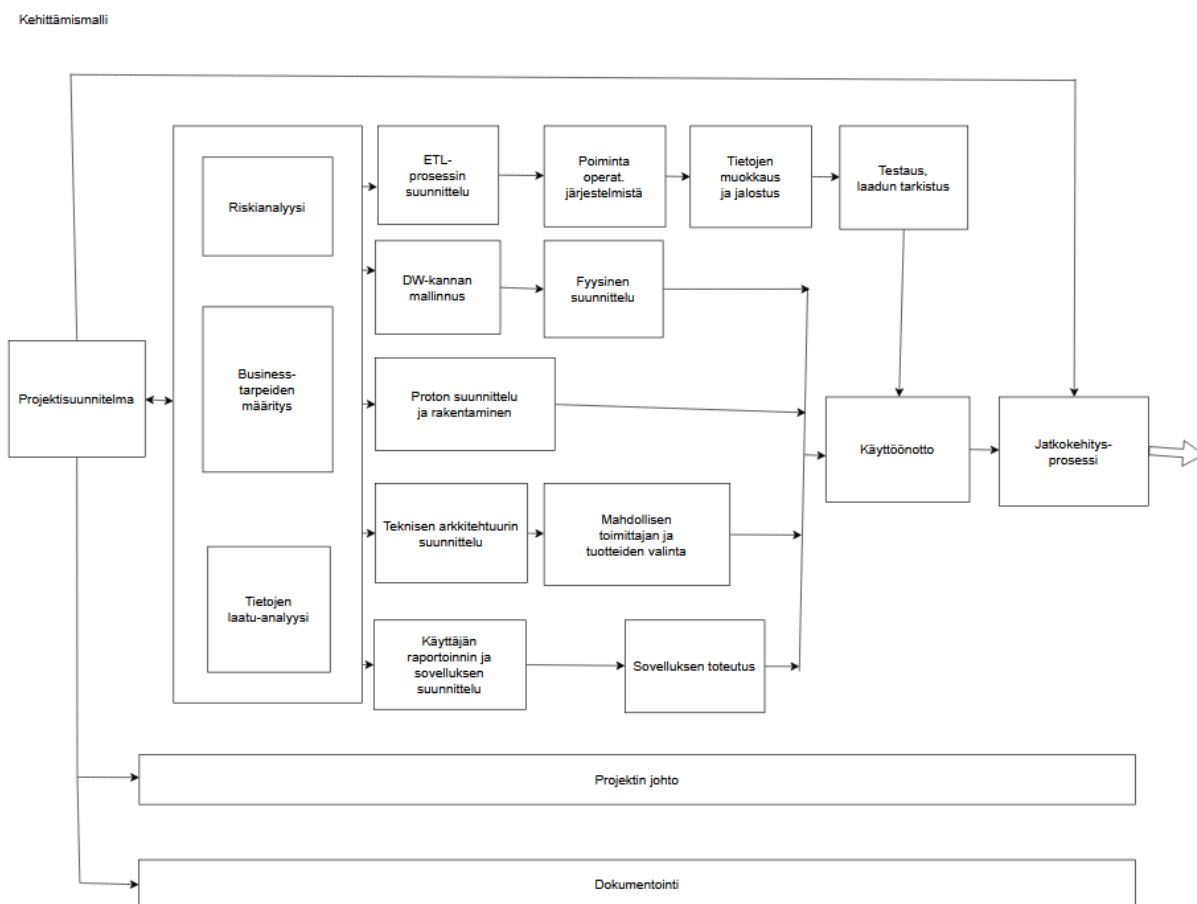
KUVA 1. Tietovarastojen hyödyntäminen Business Intelligencen kanssa. Tietovarastotietokantojen käyttäminen tukee Business Intelligence käyttöä. (Hovi;Hervonen;Koistinen, 2009, s. 14.)

Kuten kuvassa 1 esitetään, Business Intelligencessä on yleisesti tapana yhdistää useat erilliset tietojärjestelmät yhdeksi tietovarastoksi. Tämä tapahtuu jalostamalla dataa yhtenäiseksi. Tietovarastot ja Business Intelligence -teoksessa (Hovi;Hervonen;Koistinen, 2009) hyvänä datan yhdistämisen ja yhtenäistämisen esimerkkinä esitellään sukupuolien esittäminen yhdessä tietolähteessä arvoilla "M" ja "N" ja toisessa "1" ja "0", jolloin näistä on pääteltävä yhteinen muoto dataa hyödyntäessä. Toinen hyvin yleinen tapaus on päivämäärien esittäminen eri muodoissa, jolloin niiden vertailua varten kaikki päivämäärät on saatava samaan muotoon. Varsinkin vertaillessa eri tiedon säilytysmuotojen välillä päivämäärien talletusta, on niissä havaittavissa hyvinkin suuria eroja. Esimerkiksi yhdessä tietovarastossa päivämäärä saattaa olla merkittynä tarkan kellonajan kanssa, kun taas toisessa varastossa kellonaikaa ei ole. Aika voi olla myös täysin eri formaatissa, tai toisessa varastossa ei ole otettu huomioon aikavyöhykkeitä ja toisessa on.

Tietovarastoa rakentaessa on suunniteltava sen käyttöä monesta näkökulmasta. Näitä näkökulmia ovat tietovaraston laajuus, käyttäjien tarpeet, sekä mitkä tiedot poimitaan tietovarastoon. Tietovaraston laajuudella tarkoitetaan käytännössä mikä on tietovaraston kohdealue, eli millaista tiedonkäsittelyä varten tietovarasto tehdään. Käyttäjien tarpeet ovat lopputuotoksen käyttäjien näkemys siitä, millaista tietoa käyttäjät itse tarvitsevat. Kuitenkin yleensä tietovarastoon lisätään enemmän tietoja, kuin mitä käyttäjät määrittelevät tarpeikseen. tai tietovarastosta tehdään helposti muokattava. Tämä johtuu siitä, että on mahdotonta täysin tietää, millaiset tarpeet tiedoille on, ellei tietova-

raston tarkoitus ole hyvin yksinkertainen ja vain yhteen hyvin tarkkaan tarkoitukseen. Tällaiset tietovarastot ovat kuitenkin hyvin harvinaisia ja epätodennäköisiä. (Hovi;Hervonen;Koistinen, 2009, s. 14.)

Yleensä tietovarasto tehdään laajempaan käyttöön, kuten esimerkiksi varastonhallinnan ylläpitoa varten. Käyttäjän tarpeet täydentävät kysymystä siitä, mitkä tiedot poimitaan tietovarastoon ja mistä syystä. Tätä on mietittävä tietyllä tasapainolla, sillä vaikka alkuperäisessä tilanteessa päätellään, että jotain tiettyä riviä tietokannasta ei tarvita, voi se olla hankala lisätä jälkikäteen, kun kyseistä riviä tarvitaankin jostain syystä. Tällöin tietokannan suunnittelijan on tehtävä päätös, tehdäänkö tietokannasta mahdollisimman yksinkertainen ja suppea versio, joka on nopea luoda, vai lisätäänkö tietokantaan varalta kaikki aiheeseen liittyvät tiedot, jolloin kanta on raskaampi, mutta sitä ei tarvitse muokata jälkikäteen tarpeiden muuttuessa. (Hovi;Hervonen;Koistinen, 2009, s. 32.) Tämän vuoksi tietokannan suunnittelija luo yleensä jonkinlaisen suunnitelman tietovaraston luomisprosessista. Suunnitelma yleensä seuraa kuvan 2 kaltaista tietovaraston ja BI-ratkaisun kehittämismallia.



KUVA 2. Tietovaraston ja BI-ratkaisun kehittämismalli. Kuvassa esitetään, miten uuden tietovaraston ja BI-ratkaisun kehittäminen suunnitellaan ja toteutetaan (Hovi;Hervonen;Koistinen, 2009, s.131.)

### 3.3 Jalostetun datan hyödyntäminen

Kun tietokanta on huolellisesti käsitelty haluttuun muotoon, voidaan sitä hyödyntää Business Intelligence -ratkaisuisissa. BI-ratkaisujen avulla on mahdollista päästä käsiksi toimintaa kuvaavaan informaatioon. Tässä vaiheessa datalle on mahdollista tehdä tarkempia rajauksia, sekä suunnitella, millaisessa muodossa data halutaan näyttää. Business Intelligencestä on tullut iso osa käytännössä jo-kaista yritystä kerätyn datan kasvaessa. Koska nykyaikana dataa kerääntyy kaikkialta, paljon myös passiivisesti, ilman tietojen syöttämistä, Business Intelligencen hyväksikäyttö on aina vain tärkeämpää. (Hovi;Hervonen;Koistinen, 2009, s.74.) Hyvin suuri osa nykyaikaisen yrityksen datasta syntyy erilaisista antureista, jotka lähettävät passiivisesti jatkuvaa dataa yritykselle. Tällainen tieto ei ole helposti luettavissa, ellei sitä muokattaisi haluttuun muotoon jonkinlaisella BI-työkalulla. Myös käsin-tehty data, kuten esimerkiksi listaukset tilauksista voivat olla hyvin hidasta ja hankalaa luettavaa, jos niistä ei pystyisi tekemään yhteenvetoa tai jonkinlaista muuta näkymää, josta kaiken tarpeellisen näkee nopealla vilkaisulla.

BI-ratkaisujen hyödyntämisestä on useita merkittäviä hyötyjä. Datan avulla on mahdollista seurata, miten erilaiset päätökset vaikuttavat yrityksen toimintaan ja näin vertaamalla mahdollisuuksia päätöksenteko on helpompaa. Datan tarkastelua voi tehdä erilaisilla määrättyillä aikaväleillä, kuten päivittäin, viikoittain, kuukausittain tai kvartaalitasolla. Joissain tapauksissa datan tarkastelua voi tehdä lähes reaaliaikaisesti, jolloin on mahdollista seurata päivän tapahtumia. Tämä ei kuitenkaan ole yleistä muuten kuin yksittäisissä, hyvin kriittisissä tilanteissa. (Hovi;Hervonen;Koistinen, 2009, s.76.)

Tarkastellessa dataa pitkällä aikajaksolla, on mahdollista nähdä yrityksen keskiarvoinen toiminta, sekä onko yrityksen kehitys nousevaa, laskevaa tai tasaista. Näin on mahdollista nähdä jo reilusti etukäteen mahdolliset muutokset, joita yrityksen toimintaan voisi tehdä, jotta yritys jatkaa kasvuaan ja kehittymistään. Lyhyempien tarkastelujaksojen avulla on mahdollista nähdä tarkemmin pienem-mät muutokset, sekä yllättävämmät tilanteet, joilla voi olla suurikin vaikutus tulevaisuuteen riippuen, huomataanko ne ajoissa vai ei. Tämän vuoksi on erityisen tärkeää esittää sama data useassa eri muodossa, useasta eri näkökulmasta. Vertailua kannattaa tehdä mahdollisimman monella tavalla saadakseen mahdollisimman kattavan kuvan yrityksen toiminnasta, verraten sekä lyhyitä, että pitkiä aikavälejä.

### 3.4 Miksi Business Intelligence?

BI-ratkaisujen tavoitteita ovat esimerkiksi päätöksentekokyvyn nopeuttaminen ja parantaminen, käyttäjien tietotarpeisiin vastaaminen, organisaation strategian ja tavoitteiden tukeminen, käyttäjien tiedonhaun omatoimisuuden parantaminen, sekä kustannuksien vähennys ja toimintatehokkuuden kehittäminen (Hovi;Hervonen;Koistinen, 2009, s.80). Näiden toteuttamista varten ratkaisujen on oltava analyyttisiä, kvantitatiivisia, liiketoimintalähtöisiä, kaikkia organisaation osia koskettavia, pitkälle jalostettua sekä faktapohjaista. Käytännössä siis ratkaisujen on perustuttava tosiasioihin ja oltava jalostettu helposti ymmärrettävään ja jokaisen käyttäjän helposti havaittavaan muotoon. Esitettävän datan on myös oltava hyödyllistä yritykselle tai muuten käyttökelpoista. Yritykselle ei ole hyödyllistä, jos heillä on nähtävillä ulkonäöltään vaikuttava analytiikanäkymä, joka ei kuitenkaan kerro yritykselle mitään. Tai toisaalta myöskään näkymä, jossa on paljon tärkeää tietoa, mutta se on liian hankalasti luettavassa muodossa, tai muulla tavoin sekava tai vaikeasti ymmärrettävissä. Kuitenkin kaikista heikoin tapa esittää dataa on näkymä, josta käyttäjä voi ymmärtää asian täysin eri tavalla, kuin mikä on totuus. Tällainen näkymä syntyy esimerkiksi silloin, jos on graafi, jossa on hyvin epätavallinen asteikko.

BI-ratkaisun suunnittelu ja kehittäminen yrityksen tarkoituksiin on pitkä ja useita vaiheita sisältävä prosessi, joka tarvitsee useiden työntekijöiden mielipiteitä ja työpanosta. Kuitenkin ratkaisun luominen tuo pitkäkantoista hyötyä yritykselle, jonka vuoksi se on yritykselle tärkeää. Vaikka yleisesti ratkaisu luodaan jollain jo olemassa olevalla ohjelmistolla, vaatii sen luominen kuitenkin hyvää ohjelmiston osaamista sekä tietämystä yrityksen toiminnasta ja tarpeista. BI-työkalut muuttuvat jatkuvasti monimuotoisemmiksi, jokaiseen käyttötartpeeseen sopeutuvammiksi sekä helppokäyttöisemmiksi. Usein yritykset palkkaavat ulkopuolisen yrityksen tekemään BI-ratkaisun yritykselle projekti-työnä, sillä yleensä ulkopuolisella, analytiikkaan keskittyvällä yrityksellä on parempi ymmärrys ja osaaminen liittyen datan käyttötapoihin ja Business Intelligencen tekniseen pystyvyyteen. Yleensä ainoastaan isommat yritykset, jotka käsittelevät runsaasti dataa, palkkaavat oman Business Intelligence -asiantuntijan, joka toimii yrityksessä kokoaikaisesti suunnittelemassa ja luomassa uusia keinoja käyttää yrityksen dataa hyväksi.

Yllä kerrotun Business Intelligencen teorian mukaan tämän opinnäytetyön aikana tehty projekti on Business Intelligence -projekti, sillä projektin aikana yrityksen keräämä data tallennetaan ja analysoidaan hyödynnettävään muotoon käyttäen hyväksi Business Intelligence -työkaluja.

## 4 PROJEKTIN TARKOITUS

Jo opinnäytetyöprojektia ennen yritys antoi tehtäväksi luoda yrityksen datasta näkymä, joka auttaisi heitä ymmärtämään ja visualisoimaan paremmin, miltä heidän yrityksensä toiminta vaikuttaa. Yrityksellä oli ollut jo tätä ennen vastaava näkymä, joka oli tehty Power BI -sovelluksella. Kuitenkin tämä versio sovelluksesta ei toiminut heidän toiveiden mukaan, ja loppujen lopuksi se lakkasi toimimasta täysin. Tästä yritykselle syntyi halu testata, miten opiskelijat mahdollisesti toteuttaisivat samaisen sovelluksen. Tästä syntyi Grafana-pohjainen analytiikkanäkymä, jonka alkuperäinen tavoite oli näyttää tiedot, mitä vanha näkymä jo näytti yritykselle, mutta tällä kertaa oikeasti toimivana ja kaikin puolin puhtaana toteutuksena.

Alkuperäisen projektin aikana projektista oli tavoitteena saada ainoastaan konseptitasoinen prototyyppi, josta olisi mahdollista jatkokehittää yrityksen tarpeisiin soveltuva, helposti laajennettavissa oleva sovelluskokonaisuus. Kuitenkin jo alkuperäisen projektin aikana projektin toteutus edistyi paljon pidemmälle, kuin mitä tavoitteena oli, joten tästä syntyi erinomainen pohja jatkokehitykselle. Projektin myötä syntyi myös paljon jatkokehitysideoita, kuten mitä muuta dataa analytiikkanäkymässä voisi mahdollisesti esittää, sekä miten siitä voisi saada vielä toimivamman ja dynaamisemman. Näistä ajatuksista syntyi kattava kokoelma ideoita, joista kertyi tarpeeksi materiaalia jatkaa projektia opinnäytetyönä.

### 4.1 Lähtötilanne

Projektin tavoitteiksi ajoitettiin yrityksen dynaamisen toimintahistorian esittäminen, joka esittää pääsääntöisesti jo edellisen projektin aikana tehtyjä tietoja, mutta kehitettynä dynaamiseen muotoon, jolloin esitetyn datan aikaväliä on mahdollista muokata ja esittää käyttäjän haluamalla tavalla. Tähän liittyi myös mahdollisuus nähdä data päivittäisellä tasolla, eli aikajanaa siitä, kuinka laitteet toimivat peräkkäisinä päivinä. Tämän lisäksi toisena tavoitteena oli havaita yrityksen laatuun liittyvien tavoitteiden toteutumista. Yritykselle on tärkeää, että heidän tuottamansa osat ovat laadukkaita ja viallisten tuotteiden määrä on mahdollisimman vähäinen, joten laadunseurantaa varten yritys halusi nähdä, kuinka suuri prosentti valmistetuista tuotteista oli onnistuneita.

Kaikessa yksinkertaisuudessaan projektin taustana on, että Lapinlahden Koneistus oli yritys, jolla on paljon raakadataa, jota kertyy suuria määriä päivittäin, mutta ei keinoa käyttää sitä hyväksi millään tavalla. Kuten jo aikaisemmissa kappaleissa on kerrottu, datan käyttäminen ja jalostaminen on yrityksen toiminnan ja kehittymisen kannalta väistämätöntä, joten tarve analytiikkanäytölle oli suuri ja todellinen. Yleisellä tasolla tällaisten projektien tavoitteena on auttaa yritystä ymmärtämään heidän toimintaansa sekä edistämään sitä datan avulla.

### 4.2 Toteutuksen periaate

Työkaluksi projektin analytiikkanäkymälle valikoitui Grafana-analytiikkasovellus sen monipuolisen toimivuuden, muokattavuuden sekä puhtaan ja ammattimaisen ulkomuodon takia. Toimintaa analysoimalla sekä yrityksen johtajat, että muut työntekijät pystyvät kiinnittämään paremmin huomiota yrityksen toimintaan. Analytiikkaa luodessa on kiinnitetty eniten huomiota yrityksen tarpeisiin nähdä helposti jokaisen heidän laitteensa toimintatehokkuus ja -historia. Luomalla erilaisia näkymiä laitteiden toiminnasta on yrityksen mahdollista seurata kehitystä pitkällä aikajanelalla, huomata virheitä toiminnassa sekä tehdä toimintaan liittyviä päätöksiä.

Analytiikkanäkymän käyttäjiä on eri syistä sekä yrityksen johto, että kuka tahansa työntekijä, tosin pääpainotteena tämän projektin aikana on ollut toiminnan seuranta yrityksen johdon näkökulmasta. Näkymästä on hyötyä kenelle tahansa työntekijälle, sillä kuka tahansa voi seurata työn etenemistä ja tämän mukaisesti muuttaa omaa panostaan.

Sovellus on myös hyvin helposti päivitettävissä, sillä jos yrityksen laitteistossa tapahtuu muutoksia, voi uuden laitteen vain lisätä laitelistaan tai poistetun laitteen poistaa listasta, jonka myötä laite päivittyy automaattisesti näkymään. Myös esimerkiksi kapasiteettimuutokset tapahtuvat automaattisesti ja uusien robottien lisääminen tapahtuu lisäämällä muutama rivi SQL-koodia, joka on määritelty toteutusta varten tehdyssä käyttöohjeessa. Näin ollen näkymän näyttämät tiedot eivät vanhene, vaikka yrityksen toimintaan tulisikin muutoksia.

#### 4.3 Toteutusaikataulu

Työn tekeminen alkoi 11. toukokuuta 2022 vierailulla yrityksen tiloihin, sekä yhteisellä palaverilla, jonka tavoitteena oli suunnitella pääsääntöisesti, minkälaiset tavoitteet opinnäytetyön toteuttamisella on. Toukokuu kokonaisuudessaan koostui pääsääntöisesti suunnittelun ja oikeiden tietolähteiden hakemisesta. Projektin pääsääntöinen toteuttaminen alkoi elokuussa ja päättyi joulukuussa. Näkymä on ollut pilotoitavana koko suorituksen ajan, siitä lähtien kun ensimmäinen näkymä valmistui. Tämän aikataulun sisällä on myös ollut useamman kerran palavereita, joissa yrityksen kanssa on yhdessä suunniteltu projektin etenemistä, sekä tarkkailtu ja keskusteltu näkymän toimivuutta.

## 5 TEKNINEN TOTEUTUS

Toiminnan seurantanäkymä on toteutus, joka esittää yrityksen tuotannosta syntynyttä dataa kaavioiden avulla käyttäen hyväksi Grafana-analytiikkatyökalua. Grafana on jatkuvasti käynnissä palvelimella, joka on asennettuna yrityksen Azureen luodulle virtuaalikoneelle. Datat alkuperä on Visma Nova ERP -toiminnanohjausjärjestelmä, jonka tiedot yritys on tallentanut Azuren DataFactoryyn. Sieltä rajattu määrä tarvittavaa dataa on siirretty projektin aikana SQL-tietokantaan käsittelyn helpottamista varten. Myös SQL-tietokanta sijaitsee Azuressä. Data on siirretty DataFactorysta tietokantaan käyttäen hyväksi Entity Framework -migraatioihin pohjautuvaa tietokantamigraatio-sovellusta. Projektissa on siis käytetty hyväksi hyvin laajasti erilaista osaamista, mukaan lukien erilaisten tietovarastojen toiminnallisuuden tarkastelua, niiden välisen keskustelun luominen käyttäen hyväksi perinteisiä ohjelmointitaitoja sekä pilvipalveluiden tarjoamia toimintoja, SQL-kyselyiden luominen ja muokkaaminen muotoon, millä datasta saa selville halutut tiedot, sekä lopulta tietojen visualisointi miellyttävällä ja helppolukuisella tavalla.

### 5.1 Grafana

Grafanan sivuston (Grafana, ei pvm.) mukaan Grafana on avoimen lähdekoodin datan visualisointisovellus. Sen avulla on mahdollista tehdä kyselyjä, visualisoida, asettaa hälytyksiä sekä tutkia dataa erilaisten mittareiden ja kuvaajien avulla. Grafanassa on lähes loputtomasti erilaisia tapoja käyttää erilaisia kuvaajia ja niitä voi muokata hyvinkin laajasti. Projektin aikana suurimmassa käytössä olivat statistiikkamittarit, jotka näyttävät suoraan erilaisten laskutoimituksien lukuja numeroina tai prosentteina. Tämän lisäksi projektissa on käytetty mittareita, jotka kuvastavat ajanjakson tuloksia tavoitteeseen, sekä aikajanoja, joilla voi vertailla jokaisen päivän tuloksia.

Projektin toteutuksessa Grafana-sovellus asennettiin yrityksen Azure -pilvipalvelussa sijaitsevalle virtuaalikoneelle, jossa sovellus on jatkuvasti käynnissä. Tämän ansiosta sovelluksessa ei ole suunniteltuja käyttökatkoksia ja sitä voi tarkastella milloin tahansa. Grafanan näkymistä olisi mahdollista saada jopa lähes reaaliaikaisia näin halutessa, mutta päivitysväliksi valittiin tunnin välein, jotta tietokannan ei tarvitse päivittyä reaaliaikaisesti. Näkymien toteutuksesta jatketaan enemmän osiossa 6. Toiminnan analytiikan sovellus.

### 5.2 Azure

Kohdeyrityksellä oli jo valmiina Microsoftin Azure-pilvipalveluympäristö, joka oli luotu valmiiksi, koska heillä oli jo olemassa yksi versio tämän projektin aiheeseen liittyen, jonka he halusivat luoda uudelleen. Tätä pilvipalveluympäristöä käytettiin hyväksi projektin aikana, sillä aloitushetkellä ympäristöllä ei ollut mitään aktiivista toiminnallisuutta.

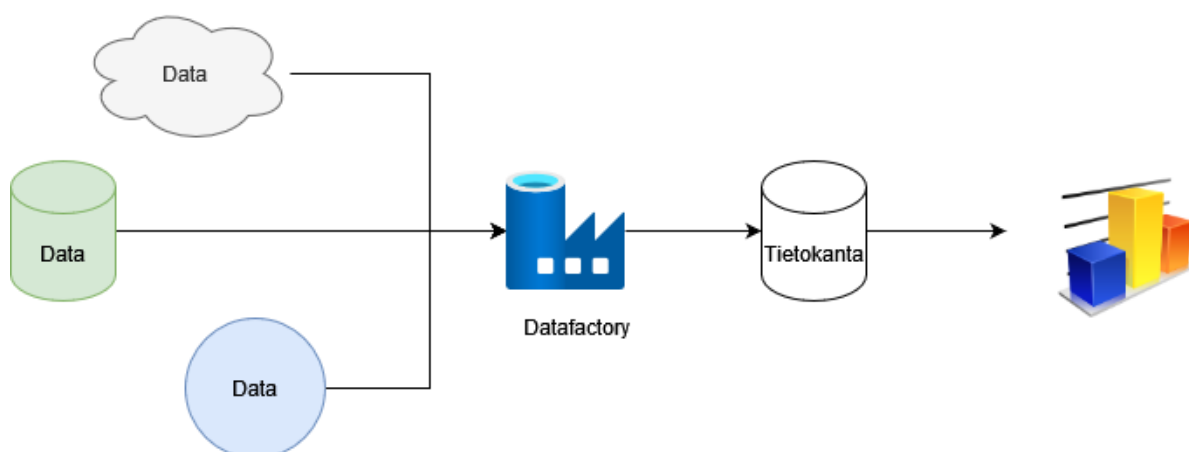
Yleisellä tasolla Azuren tarkoituksena on olla pilvialusta palveluiden ja sovelluksien rakentamiselle, julkaisulle sekä ylläpidolle, joka toimii PaaS-mallilla (Platform as a service, suomennettuna alusta palveluna) tai IaaS-mallilla (Infrastructure as a Service, suomennettuna infrastruktuuri palveluna)

(Microsoft, 2023). Projektin aikana näitä kahta mallia on yhdistelty, sillä osa käytetyistä ominaisuuksista on Azuren omia, osa itse luotuja ja Azureen tallennettuja.

Projektissa Azure on pääsääntöinen alusta, jonne kaikki projektin osat ovat siirretty. Itse infonäyttö, eli Grafanan palvelin pyörii Azuren virtuaalikoneella, kaikki kirjoitettu koodi on tallennettu Azuren portaalin versionhallintaan, tietokantamigraatio käyttää hyväkseen Azuren BlobStoragea sekä Azuren ajoitustoimintoa. Myös alkuperäinen data sijaitsee Azuren DataFactoryssa.

Jo näistä tämän projektin käyttötarpeista näkee, että Azure on hyvin monimuotoinen, laaja ja muokautuva kokonaisuus palveluita ja ominaisuuksia. Yksinkertaisuudessaan tämä tarkoittaa, että oli kohdeyrityksen käyttötarpeet ja kokoluokka minkälainen tahansa, Azuren palveluista löytyy yritykselle soveltuva ratkaisupaketti, tai yksittäisiä sovelluksia, joita yritys voi käyttää hyväkseen. Azurella on myös huomattava etu moniin muihin vastaavanlaisiin kokonaisuuksiin, sillä se on Microsoftin omistama, jolloin Microsoftin muut sovellukset ja järjestelmät on yhdisteltävissä Azuren palveluihin.

### 5.2.1 DataFactory



Kuva 3: Yksi DataFactoryn päätarkoituksista on yhdistää useampi datanlähde yhdeksi, jotta sitä voi jatkojalostaa.

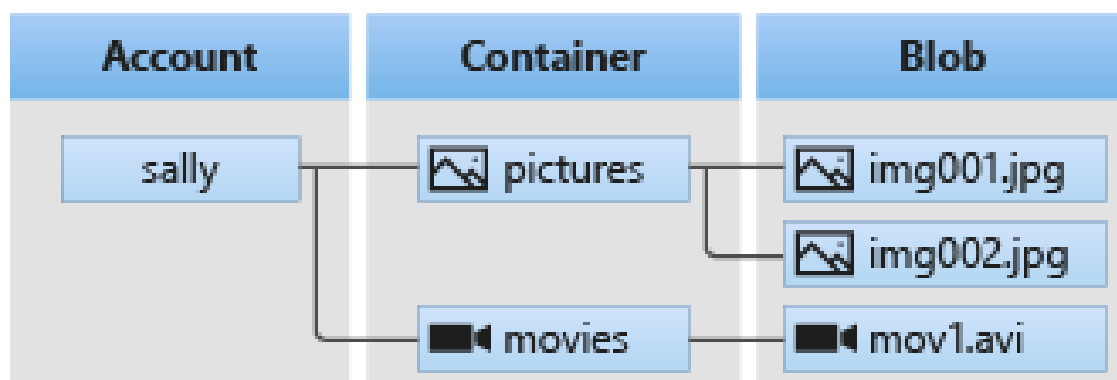
DataFactory on Azuren hallittu pilvipalvelu, joka on rakennettu ETL- (Datan siirtäminen, muokkaus, ja lataaminen), ELT- (Datan purkaminen, lataaminen ja muuntaminen) ja dataintegraatio- projekteja varten. Sen tarkoituksena on käsitellä laajasta raa'an datan massasta käyttökelpoista materiaalia, minkä voi hyödyntää yrityksen liiketoimintanäkemyksien luomiseen.(Microsoft, 2023)

Kuvan 3 mukaisesti yksi DataFactoryn toiminnallisuuksista on luoda ja ajoittaa tietopohjaisia työnkuluja, eli pipelinejä, joiden avulla se käsittelee irtonaista dataa useista eri datavarastoista. Pipeline on looginen työnkulku, joka suorittaa sille määriteltyä työyksikköä. Yksi työyksikkö voi sisältää useanlaisia tehtäviä, kuten tiedonkeräämistä Azure blobista ja sen käsittelyä. Pipeline on mahdollista käynnistää joko ajoitetusti, tai se pystyy odottamaan jotain tiettyä määriteltyä tapahtumaa, joka toimii viestinä pipelineen käynnistymiselle. Pipelineja käyttämällä on mahdollista hallita dataa ja aktiviteetteja kokonaisuutena. Aktiviteetit ovat pipelineen prosessinkulun askeleita. Lisäksi aktiviteetteja on

mahdollista ketjuttaa, jotta ne käynnistyvät peräkkäin tai rinnakkain. Näin datan käsittely on tehokkaampaa ja toiminnaltaan varmempaa. Käsittelyn jälkeen DataFactoryn avulla on mahdollista luoda datasta analytiikkaa, tai käsitellyn datan voi lähettää ulkopuoliseen analytiikkasovellukseen. (Microsoft, 2023)

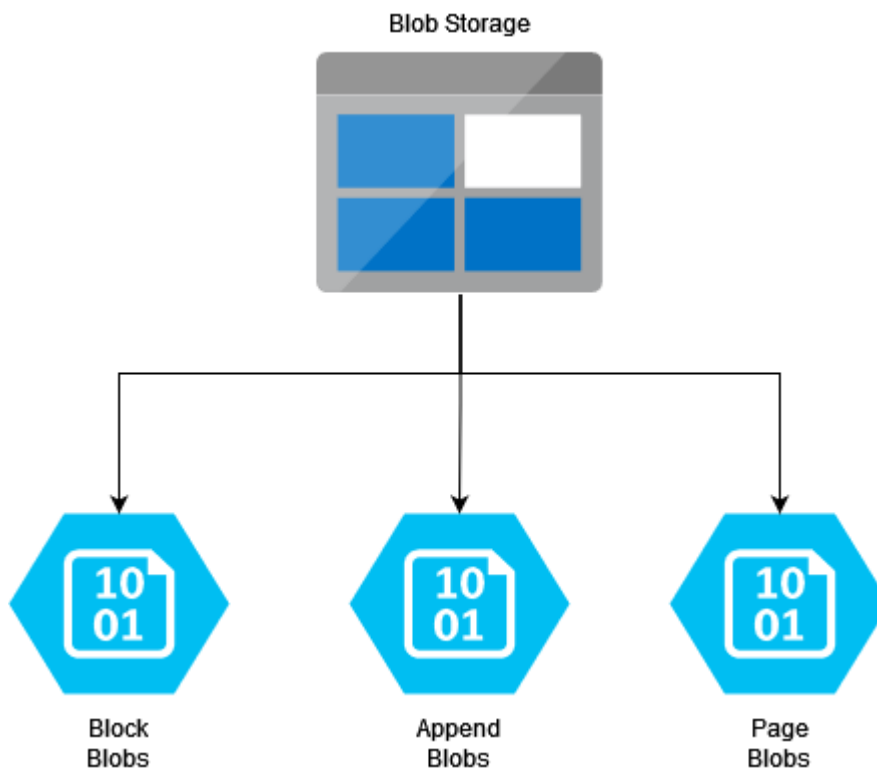
### 5.2.2 Blob Storage

Blob Storage on Azure'n pilvitalennusratkaisu, joka on optimoitu tallentamaan valtavia määriä jäsentämätöntä dataa, eli dataa, joka ei noudata tiettyä tietomallia tai määritelmää. Käytännössä kyseessä on siis NoSQL-tietokanta.



Kuva 4: Blob Storageen tallennettavien resurssien relaatio (Microsoft, 2023).

Blob Storagea on mahdollista käyttää esimerkiksi kuvien ja dokumenttien lähettämiseen suoraan selaimelle, tiedostojen tallentamiseen, videoiden ja äänen lähettämiseen, lokitiedostojen kirjoittamiseen, datan varmuuskopioimiseen sekä datan tallentamiseen analytiikkaa varten. (Microsoft, 2023) Kuva 4 esittää, miten Blob Storage käyttäytyy NoSQL-kantana tallentaessaan strukturoimatonta dataa, kuten kuvia ja videoita. Data tallennetaan blobeihin binäärimuodossa, jolloin datan sisällöllä ei ole rakenteen puolesta merkitystä.



Kuva 5: Blob Storagessa sijaitseva data koostuu erityyppisistä Blobeista, eli binäärisistä suurista objekteista.

Data tallennetaan Blob Storageen blobeina, eli binäärisinä suurina objekteina. Binäärisyyden ansiosta blobit voivat esittää millaista dataa tahansa. Blob Storage käyttää kolmea erityyppistä blobia, jotka ovat esiteltyinä kuvassa 5 sekä taulukossa 1. Pääsääntöisesti eri blob-tyypit määrittelevät, millaiseen tarkoitukseen ne ovat suunniteltu ja millä tavalla niihin liittyvä datansiirto ja -käsittely tapahtuu:

Tyyppi	Toiminto
<b>Append Blob</b>	Standardi lohkorakenteinen blob-tyyppi. Käytetään suurimmassa osassa tapauksista. Optimoitu liiteoperaatioihin sekä lokien keräämistä varten.
<b>Block Blob</b>	Lohkorakenteinen blob, jota käytetään pääsääntöisesti tekstin tai binääritiedostojen säilytykseen tai suurien tiedostojen tehokkaaseen siirtämiseen.
<b>Page Blob</b>	Rakentuu suurikokoisista sivuista. Suunniteltu toistuville, satunnaisille luku- ja päivitysoperaatioille.

TAULUKKO 1. Erilaiset Blob-tyypit ja niiden toiminnot (Microsoft, 2023).

Datan lähettämiseen eteenpäin Blob Storagesta projektissa on käytetty Azure Blob Storage triggerit ja Azuren funktioita. Triggereitä on useampia ja ne on käytettävissä sen hetkisen käyttötarpeen mukaisesti. (Microsoft, 2023) Taulukossa 2 on esitelty mahdolliset trigger-tyypit:

Tyyppi	Toiminto
<b>Blob Storage</b>	Kyselee muutoksia ja tapahtumia Blob Storagesta
<b>Blob Storage (tapahtumapohjainen)</b>	Käynnistyy tapahtumatilauksen tapahtuessa
<b>Queue Storage (jono)</b>	Käynnistyy, kun Blob siirtyy jonoon
<b>Event Grid (tapahtuma)</b>	Pystyy käynnistymään myös varaston ulkopuoleisista tapahtumista.

TAULUKKO 2: Blob Storagen triggerit ja niiden käynnistystavat (Microsoft, 2023).

### 5.3 Tietokantamigraatio -sovellus

Nimensä mukaisesti tietokantamigraatio -sovelluksen tarkoituksena oli siirtää yrityksen laitteiden data Azure DataFactorysta SQL-tietokantaan. Kyseessä on .NET-pohjainen Entity Framework -migraatio -sovellus, joka käyttää hyväkseen Azuren Blob Storagea. Sovellus on ohjelmoitu C# -ohjelmointikielellä dotNet -ohjelmistokomponenttikirjaston pohjalle.

Käytännössä sovelluksen tarve syntyi halutun datan löytyttyä Visma Novan ERP:stä, sillä se täytyi saada jollain tavalla Grafanan hyväksymään muotoon. Tähän syntyi välikädeksi tietokantamigraatio -sovellus. Haluttu data kirjoitettiin Code First -menetelmällä, luomalla ensin Visma Novan API-dokumenttaation avulla luokka datan rakenteesta, jonka jälkeen haluttu data oli mahdollista yhdistää haluttuihin kenttiin. Tämän jälkeen rakenteen pohjalta oli mahdollista rakentaa ajastettu funktio, joka lähettää päivitettyä dataa tunnin välein tietokantaan.

```

[FunctionName("SalesOrders")]
0 references
public static void SalesOrders([TimerTrigger("0 0 * * *")] TimerInfo myTimer, ILogger log) //tietokannan päivittäminen tasatunnein
{
    if (myTimer.IsPastDue)
    {
        log.LogInformation("Timer is running late!");
    }
    using (var context = new IakodbContext())
    {
        var orderNos = context.Works.Where(x => DateTime.Today.AddDays(-14) < x.EndDate).Select(t => t.OrderNumber).Distinct().ToList();

        foreach (var orderNo in orderNos)
        {
            if (orderNo.HasValue)
            {
                var soapQr = SOAP.SalesOrderSOAP(orderNo.Value);
                foreach (var so in soapQr)
                {
                    so.NetDate = so.NetDate < SqlDateTime.MinValue.Value ? SqlDateTime.MinValue.Value : so.NetDate;
                    so.CashDate = so.CashDate < SqlDateTime.MinValue.Value ? SqlDateTime.MinValue.Value : so.CashDate;
                    var oldso = context.SalesOrderInfos.Where(x => x.Number == so.Number).FirstOrDefault();
                    if (oldso == null)
                    {
                        context.Add(so);
                    }
                    else
                    {
                        context.Entry(oldso).CurrentValues.SetValues(so);
                        context.Entry(oldso).State = EntityState.Modified;
                    }
                }
            }
        }
        context.SaveChanges();
    }

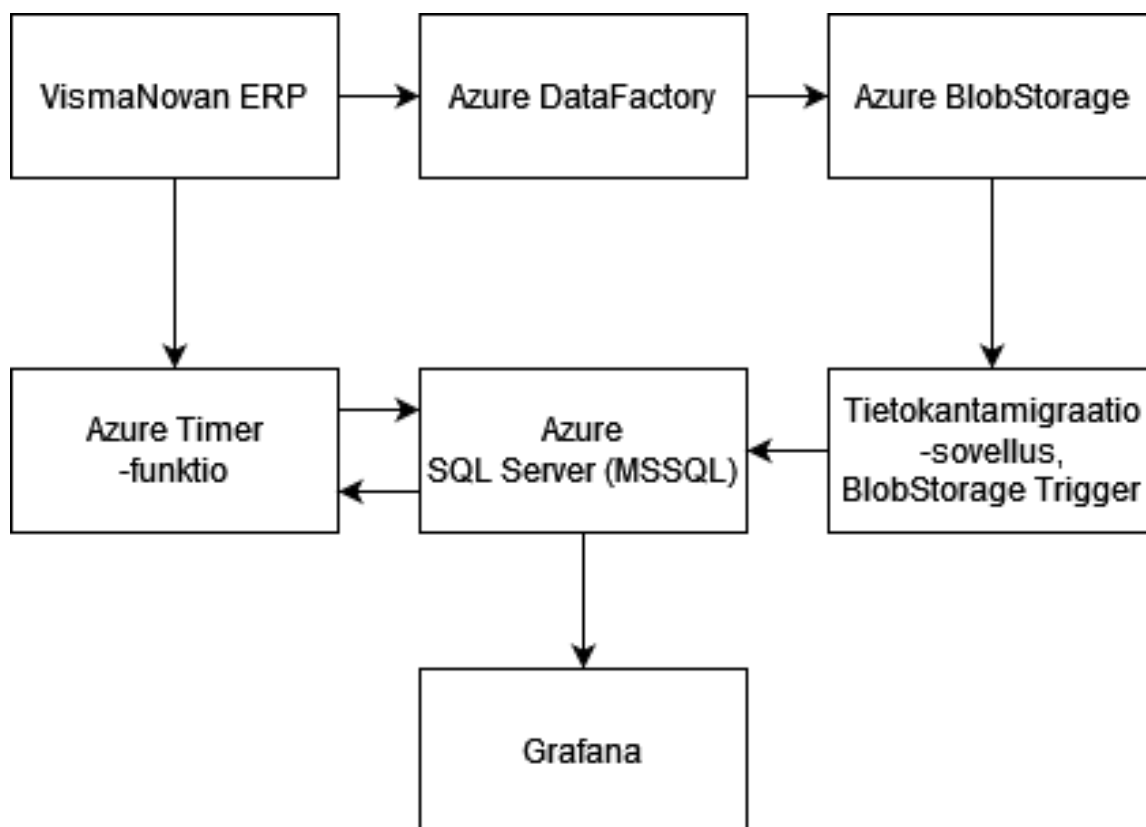
    log.LogInformation($"C# Timer trigger function executed at: {DateTime.Now}");
}
}

```

KUVA 6. Kuvattu koodi lähettää tasatunnein dataa DataFactorysta tietokantaan SOAP-kutsulla.

Ajoituksen käynnistys toimii kuvassa 6 esitetyllä Azuren timer triggerillä (ajastettu käynnistys). Ajastettua käynnistystä käytettäessä sovellus lähettää tunnin välein kyselyn, joka hakee kaikki viimeisimmän 24 tunnin aikana tulleet uudet tietueet. Saaduista tietueista sovellus siirtää SQL-tietokantaan ne, joita siellä ei vielä ollut. Kysely hakee viimeisimmät 24 tuntia estääkseen mahdollisesti virheen vuoksi pois jääneiden tietueiden puuttumisen tietokannasta. Tällöin lopputuloksena olevan näkymän toimintaan ei tule huomattavia katkoksia. Vaikka datan tallennus ei jostain syystä onnistuisi, näkymästä puuttuisi ainoastaan viimeisimmän tunnin data, joka ei edes päivätasolla ole kovinkaan huomattava virhe.

## 5.4 Projektin arkkitehtuuri



KUVA 7. Projektin arkkitehtuurikaavio.

Kuvassa 7 esitellään projektin arkkitehtuurikaavio, eli mistä projektissa käytettävä data on peräisin, ja mitä kautta se päätyy lopulliseen muotoonsa analytiikkanäkymään tarkasteltavaksi.

Yrityksen data kerääntyy alun perin VismaNovan ERP-järjestelmään. Järjestelmä määrittelee datan rakenteen, sekä käytettävissä olevat kentät. Käyttäen hyväksi Azuren Datafactorya yritys on ladanut sen pohjalle muodostuneen datan Azuren Blob Storageen.

Projektin toiminnallisuuden vuoksi data on siirretty SQL-palvelimelle. Tämä tapahtuu kahdensuuntaisesti, suoraan ERP-järjestelmästä Azure Timer -funktioiden avulla, sekä Blob Storageen kautta käyttäen BlobStorage Triggeriä. Käytetty funktio riippuu yksinkertaisesti siitä, mitä tietoja yritys on alun perin laittanut heidän Blob Storageen, puuttuvat tiedot oli saatava suoraan ERP-järjestelmästä. Suoraan ERP-järjestelmästä tulevan datan kohdalla sovelluksen on ensin tarkistettava, että onko kyseinen data jo tietokannassa, vai onko kyseessä uutta dataa.

Sekä Azure ajastimen käynnistys, että datan haku Blob Storagesta sijaitsevat tietokantamigraatio-sovelluksessa. Kun data on valmista siirrettäväksi, Tietokantamigraatio-sovellus käynnistää siirtoprosessin tietokantaan. Kun data on tietokannassa, on se välittömästi nähtävissä ja muokattavissa Grafanassa. Datan päivitys tapahtuu noin tunnin välein.

Viimeisimpänä data päätyy Grafanaan, jossa sen viimeinen jalostus tapahtuu sovelluksen käyttöliittymällä.

## 5.5 Muut käytetyt työkalut

Yllä mainittujen sovelluksien lisäksi projektin tekemisessä on käytetty hyväksi useita muita työkaluja. Koodin kirjoittamiseen on käytetty hyväksi Visual Studio Code -sovellusta. Visual Studio Code on avoimen lähdekoodin koodieditorisovellus, joka pystyy vastaamaan suurimpaan osaan mahdollisista koodaustarpeista (Visual Studio Code, ei pvm.). Toiminnan seurantanäytön tarpeet sovelluksen käytölle liittyivät pääsääntöisesti Azuren ja SQL-kannan välisiin toimintoihin, joissa käytettiin hyväksi .NET -tietokoneohjelmakehystä sekä sen osana Entity Framework -objekti-relaatiokartoituskehystä.

Tietokantatoteutuksessa työkaluna toimi HeidiSQL, joka on avoimen lähdekoodin SQL-hallintatyökalu. Vuonna 2002 luotu HeidiSQL on yksi suosituimmista MariaDB:n ja MySQL:n hallintatyökaluista. (HeidiSQL, ei pvm.) Ohjelmiston pääsääntöinen käyttötarkoitus projektin aikana oli luoda ja hallita SQL-tietokannan tauluja, sekä tarkkailla datan oikeellisuutta ennen sen visualisoimista Grafanaan.

## 6 TOIMINNAN ANALYTIIKAN SOVELLUS

Tässä kappaleessa käsitellään itse sovelluksen toteutusta. Kappaleessa käydään läpi, miten data siirtyy alkuperäisestä lähteestä käsiteltäväksi ja miten data on lopussa käsitelty. Lisäksi kappaleessa pohditaan projektin onnistumista kokonaisuudessaan, sekä mahdollisia jatkokehitysmahdollisuuksia.

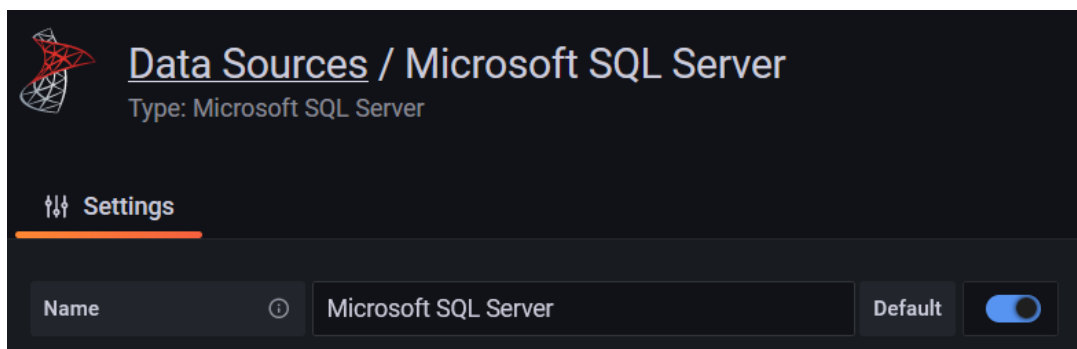
### 6.1 Datan keräys ja käsittely

Yrityksen datan alkuperäinen säilytyspaikka on Azuren DataFactoryssa, jonne se kerääntyy yrityksen käyttämistä laitteista niitä käytettäessä, sekä esimerkiksi työntekijöiden työmerkinnöistä sekä asiakkaiden tilauksista ja toimituksista. Osa datasta on syntynyt automaattisesti työtä tehdessä, ja osa on lisätty riippuen asiakkaiden tilauksiin liittyvistä prosesseista ja muista sopimuksista. Kuitenkin muissa tapauksissa data on erittäin tasalaatuista ja helposti käytettävissä ja prosessoitavissa.

Datan laadusta löytyy heikkous liittyen siihen, että sen pohjana on käytetty VismaNovan ERP -järjestelmää. Järjestelmä on luotu soveltuvaksi laajaan käyttöön usealle eri yritykselle, ja on aivan yrityksestä kiinni, kuinka montaa kenttää he käyttävät, ja millä tavalla. Tällöin joissain tilanteissa data on hyvin katkonaista, vapaamuotoista ja tämän vuoksi paikoittain heikkolaatuista.

Koska datan siirto ja käsittely DataFactoryn kautta Grafanaan havaittiin hankalaksi, siirrettiin ne SQL-tietokantaan käyttäen hyväksi tietokantamigraatiota Code First Entity Framework -tekniikalla.

Tietokantaan siirretty data on täysin identtistä DataFactoryssa sijaitsevaan dataan, mutta siitä on projektin näkökulmasta katsoen karsittu pois kaikki ylimääräiset kentät, jolloin ainoastaan projektin kannalta välttämättömät tiedot löytyvät kannasta. Jatkosuunnittelun kannalta olisi järkevää, jos tietokannassa olisi enemmän dataa, kuin kaikista välttämättömin, mutta projektin rajauksen perusteella on katsottu oleelliseksi, että tietokannasta löytyy vain tarpeellinen tieto. Kuitenkin tietokantamigraatiosovelluksen ansiosta tietokantaan on kovin yksinkertaista lisätä tarvittaessa lisää tietokenttiä, jos tälle on tarvetta jatkokehityksessä.



KUVA 8. Data on yhdistetty Grafanaan Microsoft SQL-serveriltä.

Kuten kuvassa 8 on esitetty, Grafanalle on esiteltävä, mistä datanlähteestä data on peräisin, tässä tapauksessa lähteenä on käytetty Microsoft SQL -palvelinta. Tästä syystä data siirrettiin projektin alussa Azuren DataFactorysta SQL-palvelimelle. Grafanasta ei löytynyt selkeää tapaa välittää sille

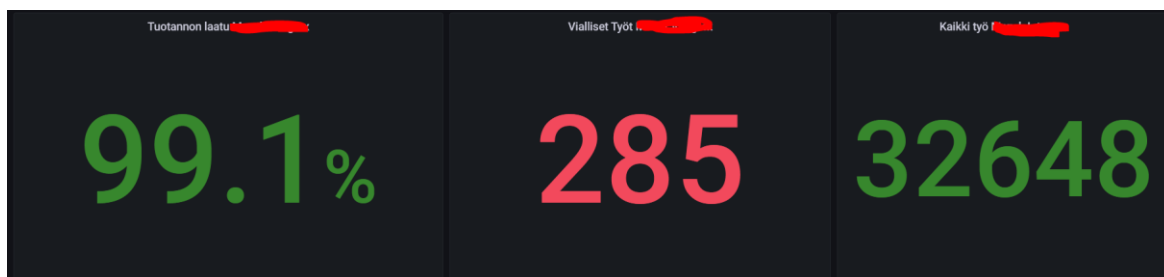
dataa DataFactorysta, joten parhaiten toimivaksi keinoksi käytettiin datan siirtämistä SQL-palvelimelle.

Käyttäen hyväksi SQL-tietokantaa, Grafanassa oli suora keino yhdistää data sovellukseen, jota kautta sitä pystyi käyttämään tietokantakyselyillä. Tietokantayhdistelyt on tehty paikallisesti Grafanan puolella suoraan, jolloin kyselyt ovat helposti sekä muokattavissa että laajennuskelpoisia tarvittaessa tulevaisuudessa. Yleisesti datan käsittely tapahtui ensin luomalla tietokantakysely, joka tuo lukuarvona tai aikajanana halutun muotoista tietoa. Tämän jälkeen Grafanan visualisoinnista pystyi tarkastelemaan, minkälainen graafi sopisi kyseisen tiedon esittämiseksi. Tässä vaiheessa tiedon rakennetta saattoi joutua hieman muokkaamaan, sillä jokaisella eri graafilla oli hieman erilainen tapa esittää tuloksen arvo. Joissain tilanteissa myös saattoi ensin olla ajatuksena millaisen graafin näkymään halusi esille, jonka jälkeen data täytyi saada näyttämään haluttua tietoa halutussa muodossa.

## 6.2 Näkymät

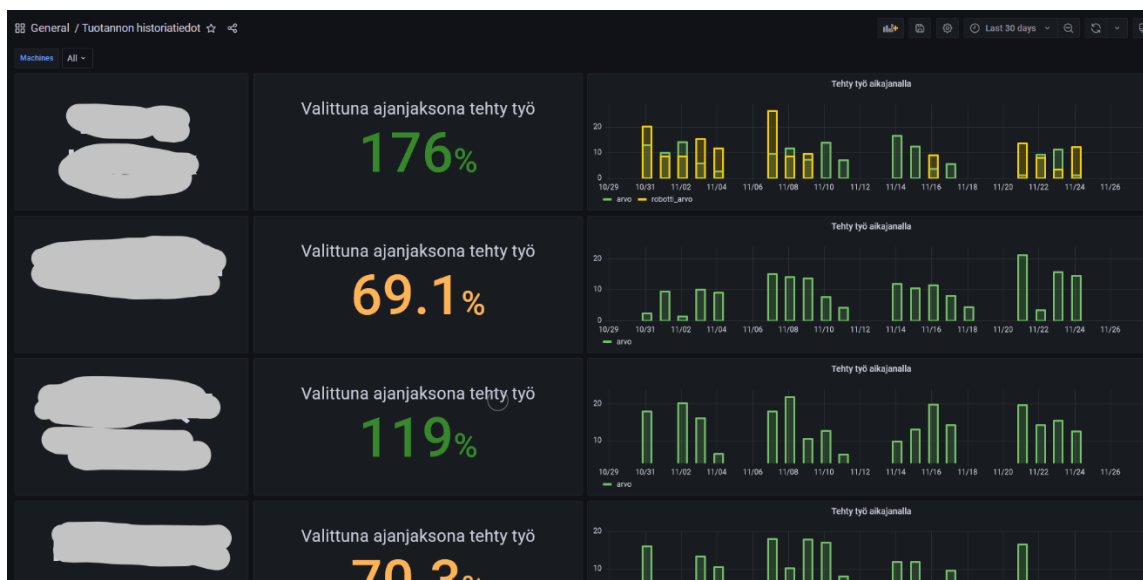
Opinnäytetyön aikana analytiikkasovellukseen on tehty kaksi uutta näkymää, jotka on nimetty "Toiminnan analytiikka" ja "Tuotannon historiatiedot". Seuraavaksi esitellään karkeasti näkymien käyttö-tarkoitukset.

"Tuotannon analytiikka" -näkyvän (KUVA 9) tavoitteena on esitellä toiminnan laatuun liittyviä tietoja käyttäjän valitsemalla ajanjaksolla. Näitä ovat (kuvassa 9 vasemmalta oikealle) Tuotannon laatu prosentteina (viallisten töiden määrä verrattuna kaikkeen tehtyyn työhön), viallisen työn määrä, kaiken tehdyn työn määrä sekä valmiiden tuotteiden toimitusvarmuus.



KUVA 9: Tuotannon analytiikkanäkymä Grafanassa. Laitteiden nimet piilotettu kohdeyrityksen yksityisyyden turvaamiseksi.

Tuotannon historiatiedoissa (KUVA 10) on nähtävissä historiatietoa jokaisen laitteen tuotannon määrästä valitulla aikajanelalla. Tehty työ on esitetty prosenttilukuna sekä aikajanana, joka esittää tehdyn työn määrän tunteina jokaisena aikajakson päivänä.



KUVA 10. Tuotannon historiatiedot -näkyvä Grafanassa. Vasemmanpuolimmaisista laatikoista on piilotettu laitteiden nimet kohdeyrityksen yksityisyyden turvaamiseksi.

Datan rakentamisessa esitettävään muotoon Grafanan puolella on käytetty SQL-kyselyitä. Näiden kyselyiden avulla tietokannasta saapuvasta muotoilemattomasta datasta syntyy analyyttikanäytössä nähtävät kuvaajat ja laskelmat. Kuvassa 11 on nähtävillä esimerkki SQL-kyselystä, joka muodostaa Tuotannon historiatiedot -näkyvässä (KUVA 10) nähtävän pylväsdiagrammin. Diagrammin luomiseen on käytetty kahta SQL-kyselyä, jotta kahden erivärisen palkin näyttäminen olisi mahdollista.

```

A (Microsoft SQL Server)
SELECT cast(phasestampings.StopTime as date),
sum(((workphase.Capacity/60)+(workphase.SetupDuration / w.OrderedAmount)) * phasestampings.Amount) as arvo,
machines.ResourceId as metric
FROM workphase
join machines on machines.ResourceId = workphase.MachineId
join phasestampings on phasestampings.WorkNumber = workphase.WorkNumber and phasestampings.PhaseNumber = workphase.PhaseNumber
JOIN works w ON w.WorkNumber = workphase.WorkNumber
WHERE
(cast(phasestampings.StopTime as date) BETWEEN $_timeFrom() AND $_timeTo())
AND machines.ResourceId IN ($Machines)
GROUP BY machines.ResourceId, cast(phasestampings.StopTime as date);

B (Microsoft SQL Server)
declare @robotti as varchar(50);
If ($Machines) = ''
set @robotti = 'ROBOTTI';

IF @robotti is not null
SELECT cast(phasestampings.StopTime as date),
sum(((workphase.Capacity/60)+(workphase.SetupDuration / w.OrderedAmount)) * phasestampings.Amount) as robotti_arvo,
machines.ResourceId as metric
FROM workphase

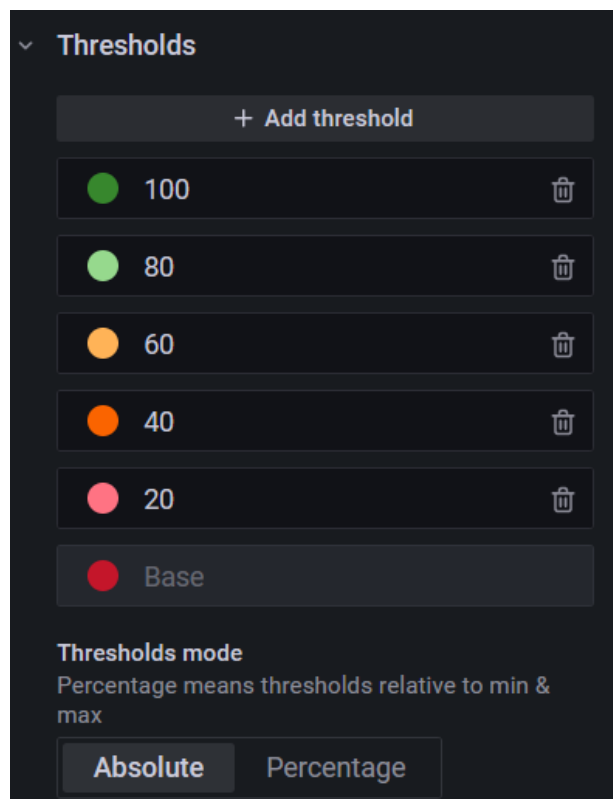
```

KUVA 11. Kuvassa 10 esiintyvä kaksivärinen pylväsdiagrammi on luotu yhdistämällä kaksi SQL-kyselyä yhteen kaavioon.

### 6.3 Grafanan visuaalinen rakenne

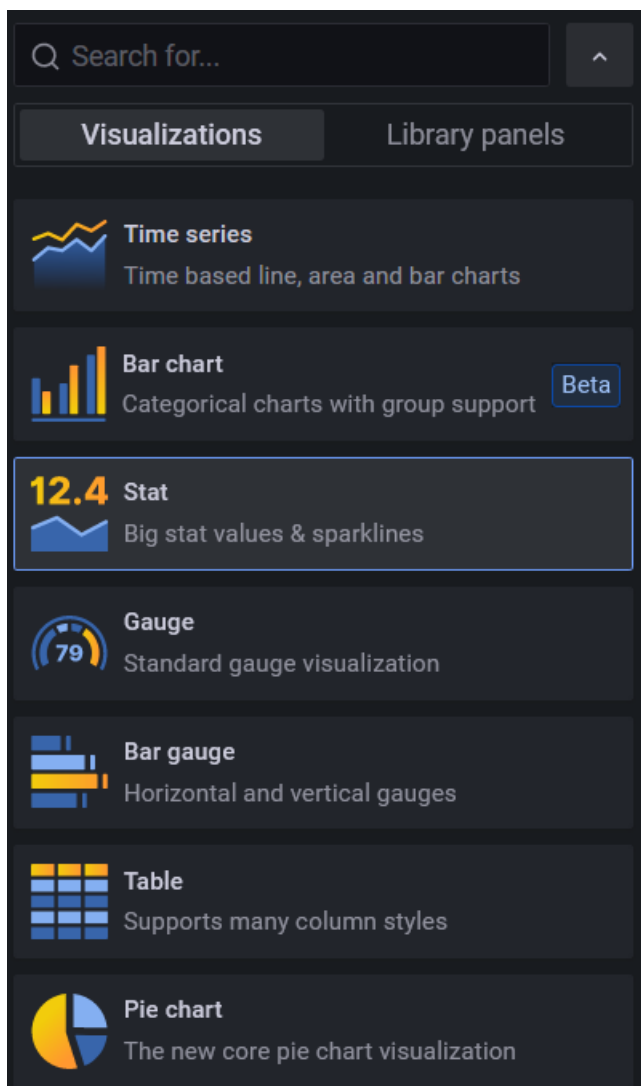
Suuri osuus Grafanan puolella tehdystä muokkaamisesta oli näkymän visuaalinen muokkaus. Kaikkiin analyyttikanäkymien numerotietoihin on luotu värityslogiikka, jonka perusteella esitetyn arvon väri muuttuu punaiseksi, keltaiseksi tai vihreäksi riippuen siitä, miten hyvä näytetty arvo on. Näkyvässä on muutenkin esitelty arvoja käyttäen hyväksi eri väriytyksiä, kuten aikajanänäkyvässä, jossa itse

laitteen arvot on esitetty vihreällä, ja laitteeseen liitetyn robotin arvot keltaisella. Tällaisilla visuaalisilla keinoilla on helppo esittää dataa ja muokata sitä helpommin luettavaksi.



KUVA 12. Grafanan arvojen värien säätely tapahtuu Thresholdien, eli kynnyсарvojen avulla.

Grafanalla on näkymien muotoilun suhteen omat toiminnallisuutensa ja sääntönsä, mutta se, miltä näkymä näyttää on helposti muokattavissa. Grafana tarjoaa esimerkiksi kuvassa 12 esiteltyyn Grafanan arvojen värien säätelyyn, jolla on mahdollista painottaa eri arvojen merkitystä.



KUVA 13. Grafanassa on käytettävänä useita kaaviopohjia, jolle analytiikkanäkymän voi rakentaa.

Grafanan visuaalinen toteutus riippuu myös hyvin paljon siitä, millaisia kaavioita toteutuksessa käytetään. Kuvassa 13 on muutama esimerkki, millaisia kaavioita Grafana tarjoaa. Projektin aikana analytiikassa on käytetty pylväskaaviota, статистиikkaa, sekä mittaria.

Projektin aikana näkymien rakenne muodostui pääsääntöisesti perustuen siihen, miten uusia laitteita on helpointa liittää sovellukseen. Tämän vuoksi päädyttiin ratkaisuun, jossa kaikki laitekohtainen data on yksinkertaisesti esitettyä yhdellä rivillä rinnakkain, jolloin uuden laitteen lisääminen on vain yksi rivi lisää näkymään, eikä tämän myötä näytettyjen tietojen rakenne pysty menemään mitenkään sekaisin, eikä sen näkymästä voi tulla hankalasti luettava.

Osana helppoa luettavuutta on myös, että data on eroteltu useaan eri näkymään. Projektin aikana tehdyt "Tuotannon historiatiedot" ja "Tuotannon analytiikka" -näkyvät esittävät hyvin erilaista tietoa, joten oli luonnollista luoda molemmista täysin omat näkymäsivut, jolloin yhdellä sivulla ei ole liikaa asiaa. Käyttäjän on nyt myös yksinkertaista löytää näkymän otsikon perusteella, minkälaista tietoa hän haluaa tarkastella, jolloin oikea tieto löytyy helposti, eikä sitä tarvitse etsiä suuren tietomassan keskeltä.

#### 6.4 Huomautuksia

Näkymien toteutuksessa on nähtävissä muutamia toimintaan liittyviä huomautuksia, jotka liittyvät enemmänkin yrityksen datan käyttötapaan, kuin näkymän toimivuuteen. Nämä ovat toisin sanoen tuotantoon liittyviä ominaisuuksia, joita ei olisi voinut huomata ilman näkymien olemassaoloa. Suurin käytössä huomattu asia on yhden tuotantolaitteen epämääräiseltä vaikuttanut data. Noin kerran viikossa kyseinen laite saattoi näyttää hyvin suuria tuotannon lukuja, esimerkiksi, että työtä olisi tehty 500% päivän tavoitteesta. Dataa tutkiessa kävi selville, että kyseinen epämääräisyys johtui työntekijöiden tavasta merkitä päivän työleimat järjestelmään. Jos työntekijä merkitsi robotin koko päivän ja yön yli tekemän työn kerralla, näkyi se suurempana tuloksena kuin kokonaiskapasiteetin perusteella pitäisi olla mahdollista. Muita datan epätasaisuuteen liittyviä syitä on myös hyvin näkymän avulla pystytty selittämään esimerkiksi laiterikoilla tai lomapäivillä.

#### 6.5 Jatkokehitys

Projekti on tehty osana DIVA-projektia, joten sen myötä projektiin on tullut useita mahdollisia jatkokehityksen aiheita. Nykyinen projekti on tehty yrityksen tämänhetkisten tarpeiden mukaan ja hyvin rajatulla tasolla.

Mahdolliset kehityksen kohteet ovat pääsääntöisesti keinoja, miten dataa pystyisi esittämään eri tavoin. Seuraavassa taulukossa on selitetty auki, mitä muita toimintoja yritys mahdollisesti haluaisi analytiikkanäkymään:

<b>Haluttu toiminto</b>	<b>Selite</b>
<b>PPM-luku</b>	Parts Per Million, kuinka monta kappaletta miljoonasta ei täytä laatuvaatimuksia.
<b>Nimikekohtainen läpimenoaika</b>	Yksittäisen tuotteen kokonaisprosessin läpimenoaika, kuinka kauan yksittäisen osan luomisella kestää. Jaoteltuna nimikkeittäin.
<b>Tuotannon vaihekohtainen tarkastelu</b>	Kattaa erilaisia tietoja tuotannon eri vaiheista, käytännössä dataa tuotteen tilauksen käsittelystä, valmistuksesta ja toimituksesta.
<b>Työvaiheiden ositus vaiheaikoihin</b>	Kuinka kauan yksittäisessä työvaiheessa kestää aikaa.
<b>Toimitusvarmuus</b>	Ajoissa toimitettujen, laatuvaatimukset täyttävien tilauksien määrä kaikista tilauksista.

TAULUKKO 3: Mahdolliset tulevaisuuden jatkokehityskohteet.

## 7 YHTEENVETO JA POHDINTA

Vaikka opinnäytetyö pohjautui vahvasti jo tietotekniikkaprojekti-kurssin aikana tehtyyn projektiin ja tämän myötä osa suunnittelusta ja pohjatyöstä oli jo tehty valmiiksi, opinnäytetyön aikana oli mahdollista tutustua tarkemmin data-analytiikan käyttötarkoituksiin, sekä siihen, mitä hyötyä siitä on yritykselle.

Kohdeyrityksen aktiivinen projektissa mukana oleminen oli hyvin suuri apu ja motivaattori näkymän toteutuksessa. Yrityksen näkemyksen ansiosta projektista oli mahdollista saada mahdollisimman heidän määritysten ja tarpeiden mukainen. Näkemystä näkymien sisältöön ja ulkonäköön ei tullut pelkästään projektin ohjaajalta, vaan myös muilta yrityksen työntekijöiltä, jolloin projektiin sai sisällön ja tarpeet useasta eri näkökulmasta. Selkeistä näkemyksistä ja toiveista huolimatta projektin toteutuksessa oli hyvin vapaat kädet.

Sovellukseen suunniteltiin alun perin hyvin paljon erilaisia ominaisuuksia yhdessä DIVA-projektin kanssa, mutta suunnittelun aikana kävi hyvin selväksi, että työn määrän lisääminen ei lisää arvoa opinnäytetyön toteutukselle, joten kokonaisuudesta karsittiin hyvin paljon pois.

Uusien näkymien tekeminen itsessään sujui pääosin hyvin, vaikka aikataulu jäikin hieman suunnittelusta jälkeen. Aikataulun lievä viivästyminen liittyi kuitenkin pikemmin muihin asioihin, kuin itse opinnäytetyön tekoprosessiin. Näiden muiden syiden vuoksi opinnäytetyö ei edennyt kesä-heinäkuussa. Kohdeyrityksen kanssa oli kuitenkin sovittu, että työn valmistumisella ei ole mitään kiirettä. Projektin aikana oli myös useita tekemistä hidastavia ongelmia, kuten vaikeudet saada data oikeassa muodossa Azuren Datafactorysta SQL-tietokantaan ja siitä eteenpäin Grafanaan. Tämä tiedon siirtämisen tapa on jotain, minkä tekisin todennäköisesti nykyään eri tavalla, mutta tämä tekotapa oli määritelty jo Tietotekniikkaprojekti-kurssin aikana enkä halunnut muuttaa jo toimivaksi varmistettua tapaa.

Kuitenkin kokonaisuudessaan opinnäytetyön sovellusosuus toimii kohdeyrityksen määrittelemällä tavalla. Sovelluksen tulevaisuus riippuu täysin kohdeyrityksen halusta muokata ja päivittää sovellusta, joka on täysin mahdollista. Sovelluksen tavoitteena on toimia päivittäisenä apuvälineenä yrityksen toiminnalle ja sen käyttö riippuu täysin heidän mielenkiinnostaan tutkia yrityksen toimintaa.

## LÄHTEET

Grafana (ei pvm.). Introduction to Grafana. Haettu 2. 10. 2022 osoitteesta Grafana: <https://grafana.com/docs/grafana/latest/introduction/>

HeidiSQL (ei pvm.). Etusivu. Haettu 22.3.2023 osoitteesta HeidiSQL: <https://www.heidisql.com>

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

Lapinlahden Koneistus (ei pvm). Etusivu. Haettu 2.10.2022 osoitteesta Lapinlahden Koneistus: <https://www.lapinlahdenkoneistus.fi/>

Microsoft (2023). Azure Blob storage trigger for Azure Functions. Haettu 1.5.2023 osoitteesta Microsoft: <https://learn.microsoft.com/en-us/azure/azure-functions/functions-bindings-storage-blob-trigger>

Microsoft (2023). Introduction to Azure Blob Storage. Haettu 1.5.2023 osoitteesta Microsoft: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction>

Microsoft (2023). Overview of Azure page blobs. Haettu 1.5.2023 osoitteesta Microsoft: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blob-pageblob-overview>

Microsoft (2023). Timer trigger for Azure Functions. Haettu 1.5.2023 osoitteesta Microsoft: <https://learn.microsoft.com/en-us/azure/azure-functions/functions-bindings-timer>

Microsoft (2023). What is Azure Data Factory? Haettu 1.5.2023 osoitteesta Microsoft: <https://learn.microsoft.com/en-us/azure/data-factory/introduction>

Microsoft (2023). What is Azure? Haettu 1.5.2023 osoitteesta Microsoft: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-azure/>

Valli, K.;& Ahlgren, S. (2013). Informaatiosta kilpailuetua teollisuusyrityksiin. Haettu 2. 10. 2022 osoitteesta [https://teknologiateollisuus.fi/sites/default/files/file\\_attachments/elinkeinopolitiikka\\_digitalisaatio\\_teollinen\\_internet\\_informaatiosta\\_kilpailuetua.pdf](https://teknologiateollisuus.fi/sites/default/files/file_attachments/elinkeinopolitiikka_digitalisaatio_teollinen_internet_informaatiosta_kilpailuetua.pdf)

Visual Studio Code (ei pvm.) Etusivu. Haettu 22.3.2023 osoitteesta Visual Studio Code: <https://code.visualstudio.com/>