

Kia Mäkinen

Datan visualisointi avuksi tiedolla johtamiseen:

Power BI tuotekehityksen tukena

Insinööri (AMK)

tieto- ja viestintätekniikka

Kevät 2024



**KAMK • University
of Applied Sciences**

Tiivistelmä

Tekijä(t): Mäkinen Kia

Työn nimi: Datan visualisointi avuksi tiedolla johtamiseen: Power BI tuotekehityksen tukena

Tutkintonimike: Insinööri (AMK), tieto- ja viestintätekniikka

Asiasanat: Power BI, raportointi, visualisointi

Opinnäytetyön tavoitteena oli rakentaa ohjelmistokehityskäyttöä varten suuntautuva Power BI -kojelauta. Työn tilanteella yrityksellä oli kiinnostusta saada tietoa tuotteensa käytöstä, jotta olisi mahdollista suunnata tuotekehityksessä käytettäviä resursseja tehokkaammin. Ennen tätä opinnäytetyötä ei kerättyä dataa ollut hyödynnetty aktiivisesti.

Työn teoriaosuuksissa käytiin läpi yleisellä tasolla Power BI -sovellus, kojelaudat sekä työssä käytetty tietovirta. Myös visuaalisen suunnittelun ja sommittelun teoriaa käsiteltiin.

Toteutuksessa käytettiin olemassa olevaa tietokantaa, josta tietovirran kautta tuotiin tarvittavat taulukot Power BI -sovellukseen. Kehittäjien jatkuvan palautteen avulla saatiin selville, minkälaista tietoa pyrittiin tuotteesta saamaan. Sen pohjalta pystyttiin tutkimaan halutulle tiedolle selkeä ja visuaalisesti miellyttävä esitystapa. Toteutukseen liittyi vahvasti myös käytetyn datan todenmukaisuuden tarkistaminen, jotta tuloksista voitiin olla varmoja nyt ja myös tulevaisuudessa.

Insinöörityön lopputuloksena oli monisivuinen kojelauta, josta löytyy tärkeimmät tuotteen käyttöä esittävät kuvaajat. Kojelauta auttaa yritystä ja kehityksessä työskentelevää tiimiä näkemään ja ymmärtämään tarkemmin tuotteensa käyttöä. Sivutuotteena työstä kehkeytyi myös yrityksen visuaaliseen identiteettiin sopiva raporttimalli, jota on mahdollista käyttää tulevissa Power BI -raporteissa.

Jatkokehitysmahdollisuutena olisi muun muassa samantyyllisen kojelaudan kehitys erilliseksi yrityksen tarjoamaksi tuotteeksi, josta asiakkaat itse voisivat saada informaatiota omien käyttötapojensa pohjalta.

Abstract

Author(s): Mäkinen Kia

Title of the Publication: Visualizing Data for Strategic Insights: A Power BI Approach

Degree Title: Bachelor of Engineering, Information and Communication Technologies

Keywords: Power BI, reporting, visualisation

The objective of this thesis was to create a Power BI dashboard for software development purposes. The company ordering the work was interested in gaining more insight into the usage of their own product, so that it would be possible to direct the resources used in product development more efficiently. Before this thesis project, the collected data existed, but was not being used in an effective manner.

In the theoretical parts of the thesis, the Power BI application and dashboards were reviewed at a general level, as well as the dataflow used in the work. The theory of visual design and composition was also explained.

For the implementation an existing database was used, from which the required tables were brought to Power BI application through a dataflow. With the help of continuous feedback from the development team, it became possible to figure out what kind of information they were interested in acquiring about the product. Based on that, it was possible to research a clear and visually appealing way of presenting the data. In addition to this, a significant part of the implementation was to verify the validity of the data used, to ensure that the results were and would remain dependable.

The outcome of the thesis was a multi-page dashboard, from which graphs about the product's usage could be found. The dashboard assists the company and the development team in viewing and understanding their product's utilisation more clearly. As a side product the work also became a Power BI report template that fit the visual identity of the company, so that it may be used in future reports.

There are possibilities for further development, such as constructing a similar dashboard targeted towards the customers of the company, from which they could see information about their own usage of the application.

Sisällys

1	Johdanto	1
1.1	Yritys.....	1
1.2	Alkutilanne	1
1.3	Työn aihe	1
2	Kojelauta.....	3
2.1	Erlaisia kojelautoja	3
2.2	Työn visuaalinen suunnittelu	4
2.3	Power BI	5
3	Tietovirta	7
3.1	Tietojen päivitys ja ajantasaisuus.....	8
3.2	Taulukoiden keskinäiset suhteet.....	9
3.2.1	Ongelma 1: Kaksi yhteyttä yhdellä tunnisteella	10
3.2.2	Ongelma 2: Duplikaatteja datassa	13
3.3	Tietokannan ymmärrettävyys	18
4	Kojelaudan toteutus	19
4.1	Sivu 1: Kalibroinnit.....	19
4.2	Sivu 2: Instrumentit	23
4.3	Sivu 3: Menetelmät	26
4.4	Sivu 4: Roolipohjainen käytön hallinta	29
4.5	Sivu 5: Sekalaiset	33
4.5.1	Referenssit	33
4.5.2	Laitosrakenne.....	35
4.5.3	Muokattavat kentät	35
4.6	Lopullisen ulkoasun yhdenmukaistaminen	37
4.7	Käytetyn datan luotettavuus.....	39
5	Pohdinta	41

Lähteet

Symboliluettelo

Backend	Nettisivun taustalla palvelimella pyörivä ohjelmakoodi, joka huolehtii sivuston syvemmästä logiikasta ja tietojen hallinnasta. Palvelee frontendille tietoa sen sitä pyytäessä.
Blob	<i>(Binary Large Object)</i> Datakokoelma, jonka rakennetta ei tarvitse määrittää. Tyypillisiä käyttökohteita ovat kuvien, videoiden, ja audion tallennus.
DAX	<i>(Data Analysis Expressions)</i> Kaava- ja kyselykieli, joka on käytössä Microsoftin datankäsittelytuotteissa. DAX:illa voidaan noutaa dataa taulukoista sekä laskea ja palauttaa arvoja.
Enum	<i>(Enumerated type, suom. jäsenelty arvo)</i> Muuttujatyyppi, joka on ryhmä valmiiksi määriteltyjä vakioita.
Frontend	Nettisivun näkyvä osuus: käyttöliittymä, jonka kautta käyttäjä voi liikkua sivustolla ja joka määrittelee käyttäjälle näytettävät elementit.
GUID	<i>(Globally Unique Identifier)</i> 128 bittiä pitkä ainutkertainen tunnistus, jonka monistuminen on erittäin epätodennäköistä. Tunnetaan myös lyhenteenä UUID <i>(Universally Unique Identifier)</i> .
JSON	<i>(JavaScript Object Notation)</i> Helposti luettava tekstipohjainen tiedostomuoto, jota käytetään esimerkiksi tiedon siirtämiseen palvelimen ja nettisivun välillä.
Kalibrointi	Kalibroinnilla tarkoitetaan mittalaitteen antaman arvon vertaamista johonkin tarkempaan mittaan. Tuloksen pohjalta saadaan kalibroivan laitteen virhe, jonka perusteella se voidaan virittää takaisin oikeaan tilaan.
Kojelauta	Datan visualisoinnissa käytetty esitystapa, jossa tärkein informaatio on koottu selkeäksi kokonaisuudeksi. Usein käyttäjä myös pystyy olemaan vuorovaikutuksessa esitetyn tiedon kanssa, esimerkiksi käyttämällä suodattimia.
Power Query	Microsoftin datan muokaus- ja valmistelumoottori. Käytössä mm. Excelissä ja Power BI:ssä.

RBAC	<i>(Role Based Access Control, suom. Roolipohjainen käytön hallinta)</i> Käytönhallinnan menetelmä, jossa luodaan rooleja tietyillä oikeuksilla. Käyttäjät voidaan määrätä tiettyyn rooliryhmään, josta tämä saa annetut oikeudet.
SaaS	<i>(Software as a Service, suom. ohjelmistopalvelu)</i> Pilvipalveluna toimiva ohjelmisto, joka tarjotaan internetin kautta.
Scrum	Projektinhallinnan työkalu, joka hyvin toteutettuna mahdollistaa nopean reagoinnin kehityksessä tapahtuviin muutoksiin.
Tietovirta	Microsoftin tarjoama tapa, jolla pystytään hakemaan valituista tietolähteistä haluttu data. Haettua tietoa voi muokata tarpeen mukaan ennen sen käyttämistä muissa sovelluksissa.
Tuoteomistaja	Vastuussa tuotteen kehityksestä ja siitä, että tehtävät tehdään oikeassa järjestyksessä.

1 Johdanto

Tiedon visualisointi on muuttunut tiedonalaksi taiteen sijaan. Nykyajan informaatiotulvan keskellä tietoa on helppo kerätä lähes kaikesta, mutta sitä on vaikeampaa hyödyntää. Monilla yrityksillä voi olla jo tietokantoja, joihin on kerätty suuria määriä informaatiota, mutta joita ei käytetä. Syynä voi olla yksinkertaisesti datan hyödyntämisen tietotaidon puuttuminen. Kaiken kokoisten yritysten tulisikin vakavasti harkita tapoja, joilla jo olemassa olevaa tietoa voitaisiin esittää ja hyödyntää esimerkiksi päätösten suunnittelussa ja toteutuksessa. [1, luku 2 Gaining Data Literacy with Power BI.]

1.1 Yritys

Toimeksiannon tilaaja on eräs suomalainen teknologia-alan yritys, joka toimii myös muualla maailmalla oman alansa huipulla. Yritys suunnittelee, tuottaa ja huoltaa kalibrointilaitteita ja -ratkaisuja teollisuuden tarpeisiin. Heidän tuotteisiinsa kuuluu myös kalibroinninhallintaohjelmistoja, jotka toimivat SaaS-toimintaperiaatteella. Kuitenkaan pilvipalvelun käyttäjiltä kerättyä tietoa ei ole pystytty vielä käyttämään tehokkaasti.

1.2 Alkutilanne

Yrityksellä oli tarve laajentaa dataosaamistaan ja hyödyntää olemassa olevia tietokantoja, sillä opinnäytetyön aloitushetkellä niitä käytettiin todella vähän: ainoastaan myynnin ja markkinoinnin tulosten mittaamiseen. Heillä oli kiinnostusta tietää esimerkiksi, mitä kalibraattorimalleja käytetään eniten heidän hallintaohjelmistonsa kanssa, kuinka useasti kalibroinnit tehdään ja millaisia laitteita näillä kalibroidaan. Data tästä oli jo olemassa, mutta se täytyi saada muutettua helposti luettavaan muotoon.

1.3 Työn aihe

Insinööriyön aiheeksi muodostui yrityksen sisäiseen käyttöön tarkoitetun kojelaudan luominen jo olemassa olevasta tuotteesta kerätystä käyttödatasta. Työn tarkoituksena olisi myös toimia

pohjana myöhemmille samanlaisille kojelautaprojekteille niin toteutuksen, mutta myös visuaalisen ulkonäön ja suunnittelun puolesta. Tärkeää oli pitää yrityksen yleinen tuotokuva mielessä, jotta lopputulos olisi yhtenäinen muiden tuotteiden kanssa.

Työtä olisi voinut myös laajentaa samankaltaiseen kojelautaan, joka olisi kohdennettu kalibrointihallintaohjelmiston asiakkaille. Se olisi sisältänyt paljon samanlaisia kaavioita, mutta siihen käytetty data olisi tullut ainoastaan sitä katselevalta asiakkaalta. Ideaa ja sen mahdollista toteutustapaa tutkittiin, mutta tultiin lopputulokseen, ettei sitä olisi helppoa tai edes ajankäytöllisesti järkevää toteuttaa tehdyn työn rinnalla.

Opinnäytetyössä etsittiin ratkaisua näihin kysymyksiin: Miten kerättyä tietoa olisi järkevää ja esittää, niin että se on myös selkeää ja helposti luettavaa? Millainen kojelaudan ulkoasu pitäisi olla, jotta se olisi myös yhdenmukainen työn tilanteen yrityksen tuotekuvan kanssa? Miten saadaan varmistettua näytetyn tiedon luotettavuus ja mitä datasta ylipäättään haluttaisiin tietää?

Seuraavassa luvussa kerrotaan hieman teoriaa kojelautoista, visuaalisesta suunnittelusta sekä Power BI:stä sovelluksena. Kolmannessa luvussa esitetään, miten data on hankittu, ja miten se saatiin käyttöön Power BI:ssä. Aiheeksi otetaan myös, kuinka dataa piti muokata ja millaisia haasteita siinä tuli vastaan. Neljännessä luvussa avataan enemmän kojelaudan visuaalista esitystapaa sekä käydään läpi koko kojelauta. Viidennessä ja viimeisessä luvussa käydään läpi saavutetut tulokset ja mahdolliset jatkokehitysideoita tulevaisuudelle sekä pohdintaa opinnäytetyöstä.

2 Kojelauta

Kojelauta on raportti, jonka kanssa käyttäjän on mahdollista olla jollain tavalla vuorovaikutuksessa. Ne mahdollistavat yrityksen sisäisen suorituskyvyn tarkkailemisen, joiden seuraamisessa on yleisiä esimerkiksi perinteiset kaaviot, mutta myös keskeiset suorituskykyindikaattorit. [2.]

Keskeiset suorituskykyindikaattorit (KPI) ovat tapa mitata toimintaa ja sen tasoa verrattuna yrityksen omiin asettamiin tavoitteisiin sekä niiden mahdollisuutta onnistumiseen [3, s. 5]. Seuratut arvot voivat olla esimerkiksi myyntilukuja ja sivuston kävijämääriä. KPI:n pääpiirteinä on myös helppolukuisuus, eli kaikille täytyy olla selvä, mitä ne mittaavat. Tärkeää on myös olla tietoinen, miten niihin pystytään vaikuttamaan yritystasolla. [3, s. 6.] Opinnäytetyössä ei kuitenkaan käytetty keskeisiä suorituskykyindikaattoreita lainkaan, mutta tämä johtui sopivien käyttökohteiden puutteesta.

Käytetty data voi olla joko reaaliaikaista, tai sitä voidaan päivittää tunneittain, päivittäin, viikoittain tai jopa kuukausittain. Kuitenkin suurimpaan osaan kojelautoista riittää päivittäinen tietojen päivitys. [4, s. 19.]

Vuorovaikutteisuus kojelautoissa tarkoittaa esimerkiksi kuvaajien suodattamista ja porausmahdollisuuden hyödyntämistä, mitkä antavat käyttäjälle tehokkaan tavan tutkia annettua tietoa [5, s. 4]. Porautuminen tarkoittaa esitettyyn kuvaajaan tarkemmin tutustumista, joka antaa yksityiskohtaisempaa tietoa valitusta datasta [6, s. 252]. Esimerkiksi myynnin vuositasoa esittäjässä kuvaajassa voidaan porautua hierarkiassa alaspäin ja tutkia lukuja kuukausi- tai päivätasolla.

2.1 Erilaisia kojelautoja

Kojelautatyyppejä on monia erilaisia. Eniten käytettyjä ovat toiminnalliset, analyttiset sekä strategiset kojelaudat. Toiminnalliset kojelaudat ovat tarkoitettu mahdollistamaan nopea toiminta näytetyn tiedon perusteella, joten käytetty data on yleensä reaaliaikaista. [4, s. 7.] Tällaisia ovat esimerkiksi projektien seurantaan luodut kojelaudat, jossa päivittäistä edistymistä voidaan tarkkailla.

Yrityksen menestymistä ja tavoitteiden saavutusta seurataan strategisilla kojelautoilla. Niistä usein voidaan verrata edellisiä tuloksia nykyhetkeen, ja ne vastaavat yritysten tarpeeseen tietää,

mihin suuntaan esimerkiksi tuloskehitys on menossa [6, s. 250–251]. Strategisissa kojelaudoissa on laajasti käytössä keskeiset suorituskykyindikaattorit, jotka antavat viitteitä asetetuista tavoitteista ja niiden saavuttamisesta [2].

Analyttisessä kojelaudassa on samoja piirteitä kuin aiemmin mainituissa. Se on eniten käytössä yrityksen johtoasemassa olevilla henkilöillä, joiden täytyy päästä porautumaan näytettyyn dataan. Kojelaudassa täytyy olla mahdollisuus löytää tietoa kuvaajien ja lukujen kehityssuunnasta, mikä auttaa päätösten teossa. [2.]

Opinnäytetyön toteutuksessa keskityttiin saavuttamaan tasapaino analyttisen ja operatiivisen kojelautatyyppien välillä. Koska kalibrointi harvoin on joka päivä suoritettava operaatio, ei myöskään täydellistä päivittäistä dataa ollut tarve esittää. Epäsäännöllisyyden vuoksi erilaisten keskeisten suorituskykyindikaattoreiden käyttöön ei myöskään löytynyt hyvää käyttökohdetta. Suurimasta osasta tietokannan datasta puuttui myös aikaleima, joten täytyi tyytyä käyttämään kuvaajia, jotka esittävät vain nykyhetken tilanteen.

2.2 Työn visuaalinen suunnittelu

Kojelaudan selkeys oli yksi tavoitteista. Visuaalinen kokonaisuus täytyi olla loogisesti järjestelty, eikä kaavioita tai muuta tekstiä saanut olla liikaa samalla näkymällä. Onkin parempi luoda tarvittaessa kojelaudalle enemmän sivuja, kuin tehdä yhdestä näkymästä liian monella elementillä täytetty. Näytöllä oleva tila on rajattu ja liika informaatio yhdellä ruudulla aiheuttaa vain käyttäjän ylikuormitusta ja vaikeuttaa päätöksien tekemistä. [7, s. 57.]

Eräitä yleisimpiä nettisivuilla käytettyjä sommittelutapoja ovat niin kutsutut Z- ja F-muodot, joiden viivoja pitkin katse liikkuu esitetyllä sivulla [8]. Katsoja ei heti lue kaikkea esitettyä materiaalia, vaan selaa ensin näkymän läpi, sillä nykyajan internetissä se on paljon tehokkaampaa kuin kaiken lukeminen. [9, s. 14–15.] Kojelaudassa sovellettiin F-muotoa, jossa katse liikkuu ensin näkymän yläreunassa vasemmalta oikealle, siirtyy takaisin vasempaan reunaan ja jatkaa oikealle hieman alemmaksi.

Sivun tärkeimmät asiat täytyy sommitella näkymän yläosaan, ja kaikki vähemmän tärkeät elementit mahdollisimman alas. F-muotoa seurattaessa katse käy useasti sivun vasemmassa laidassa, joten ylimääräinen infoteksti ja tärkeimmät numerot oli loogista asettaa sinne. Isoimmat ja tärkeimmät kaaviot kannatti heti laittaa sivupaneelin viereen, jolloin maksimoitiin niiden vaikutus. F-muoto

toimii lähinnä puolustusmekanismina sivulla, joissa on paljon tekstiä [9, s. 54], mutta sitä voitiin käyttää hyväksi työn suunnittelussa.

Kuvaajien laatimisessa täytyi miettiä parasta esitystapaa datalle. Tasapaino oli pidettävä eri kuvaajatyyppeiden välillä, ettei jokaisella sivulla ole pelkkiä pylväs- tai piirakkadiagrammeja. Pyöreitä donitsi- ja piirakkakaavioita yleisesti kannattaa välttää [10, s. 397], tai niitä on suotavaa käyttää vain kuvaajissa, joissa kategorioiden määrä on alhainen. Liian moneen osaan jakautunutta kuvaajaa on vaikea lukea, eikä määrien vertailu ole enää selkeää [1, luku 6 Visualizing Data]. Kojelaudassa käytetyt donitsikaaviot kuitenkin sisälsivät vähän kategorioita, joten ongelmia ei syntynyt.

Käyttäjälle tarjottujen näkymien täytyy olla samaan aikaan mielenkiintoisia ja noudattaa jonkinlaista yhtenäistä kaavaa, mutta ei kuitenkaan liian monotonisesti. Liian tylsä esitystapa ei saa käyttäjää kiinnostumaan koko kojelaudasta, kun taas erikoiset kuvaajat näyttävät ensin vaikuttavilta, mutta vaikeuttavat tiedon ymmärtämistä. [10, s. 351.] Parhaimman käyttäjäkokemuksen saisi liittämällä kuvaajat jotenkin käyttäjään [10, s. 341], mutta kalibrointia kuvaavien tilastojen kanssa tämä on haastavaa. Kokonaisuutta tutkiessa oikeanlainen värienkäyttö on myös tärkeää, sillä se edistää kuvaajien ymmärtämistä tehokkaasti [5, s. 10].

Värit olivat haastavia saada sopimaan kokonaiskuvaan, etenkin niin, että niihin olisi jäänyt tarpeeksi kontrastia. Teeman täytyi sopia myös yrityksen omaan värimaisemaan ja värit piti olla valittuna ennalta määrätystä paletista, joka kuitenkin tarjosi hyvin rajallisen määrän väriskaalaa käytettäväksi. Kunnollisia käytössä olevia värejä oli paletissa kahdeksan kappaletta, mutta määrä oli riittävä. Kahdeksan väriä on ymmärrettävyyden ylärajalla [10, s. 370].

Sopivan kontrastin löytyminen oli erityisen vaikeaa kuvaajissa, joissa näytettäviä elementtejä löytyi enemmän kuin kahdeksan. Väriteema täytyi pitää sopivana, mutta samaan aikaan liian samanaisten värien käyttöä piti välttää. Työssä otettiin myös värisokeus huomioon, joten vihreän värimailman kanssa punaisia värejä ei käytetty lainkaan. Värisokeudessa ongelma ei kuitenkaan yleensä ole kirkkaiden tai tummien värien käytössä, vaan itse värisävy, joten esimerkiksi normaalisti paletista löytyvä kirkas vihreä olisi sopinut tumman punaisen kanssa. [10, s. 391, 393.]

2.3 Power BI

Työ tehtiin Microsoftin Power BI -ohjelmistolla, joka on tarkoitettu tiedon visualisointiin ja esittämiseen. Se on pilvipohjainen SaaS-palvelu, jonka vahvuuksiin kuuluu edullinen hinta sekä helppo

käytettävyys [5, s. 9]. Power BI on intuitiivinen ja se on helppo oppia, mutta kokonaisuus on vaikea hallita.

Power BI:stä on olemassa kolme eri versiota: työpöytä- ja älypuhelinversiot sekä pilvipalvelu Power BI Service, jota voi käyttää selaimen kautta.

Työpöytäversio on jokaisesta vaihtoehdosta käytetyin raporttien luomiseen ja datan kanssa työskentelyyn [5, s. 8]. Sovelluksessa käytettävän datan voi hakea sadoista eri lähteistä, ja se tarjoaa käyttöympäristön suuren datamäärän visualisointiin. Datan päivittämisen on myös mahdollista automatisoida, jolloin luodut raportit ovat aina ajan tasalla. [11.]

Selaimen kautta on myös mahdollista luoda raportteja Power BI Service -portaalissa, mutta sen toimintakyky on vielä työpöytäversiota huonompi, ja se sopii lähinnä jo olemassa olevien raporttien selaamiseen ja tutkimiseen [12]. Se kuitenkin on pääasiallinen paikka, josta raportteja jaetaan ja avataan.

Power BI Mobile on puhelimelle ladattava sovellus, joka on toimintaperiaatteeltaan selainversion kaltainen. Työpöytäversiossa voi kuitenkin rakentaa halutulle raportille lisäksi myös älypuhelisten näytöille sopivan sommittelun, jotta kaavioihin pääsee helposti käsiksi mistä tahansa. Sovellukseen voi myös asettaa ilmoituksia esimerkiksi päivittyneestä datasta. [13.]

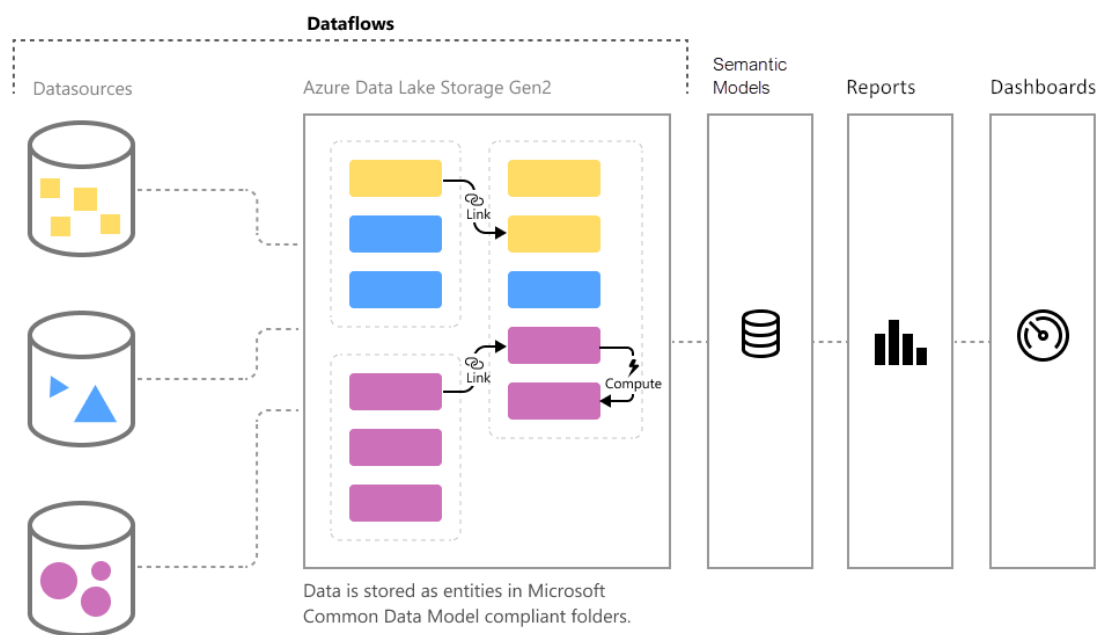
Käyttöliittymä muistuttaa paljon muita Microsoftin Office-tuotteita, ja sovelluksen yhteensopivuus myös näiden sovellusten kanssa on korkea. Power BI -kojelautoja ja -raportteja voi helposti käyttää muiden Microsoftin tuotteiden, esimerkiksi SharePointin, Excelin, PowerPointin ja Teamsin kanssa, ja käyttöliittymä sopii myös älypuheliinkäyttöön. [1, luku 2 Gaining Data Literacy with Power BI.]

Power BI:n käytössä auttaa Microsoftin Excel-ohjelmiston hallitseminen, sillä monet Excelistä tutut toiminnot pätevät myös Power BI:n puolella. Esimerkiksi Power Pivot ja Power Query ovat työkaluja, jotka ovat siirtyneet lähes samanlaisina Excelistä Power BI:hin [1, luku 2 Gaining Data Literacy with Power BI]. Myös hyppy Power BI:n DAX-kaavakieleen on lähes vaivatonta, jos hallitsee Excelissä erilaisten funktioiden käytön [1, luku 5 Leveraging DAX].

3 Tietovirta

Ensimmäisenä datan käyttöä varten piti päästä itse tietokantaan käsiksi. Työtä varten saatiin käyttöoikeus tietokannasta hakevaan niin kutsuttuun tietovirtaan, joka tässä merkityksessä on Microsoftin tuote ja toimii myös sulavasti heidän muiden tuotteidensa kanssa. [14.]

Kuva 1 esittää tietovirtojen käyttötapaa. Tietovirrat ovat erinomainen tapa jakaa samaa dataa moneen eri paikkaan. Se voi yhdistellä monta tietokantaa sekä tehdä muokkauksia sarakkeisiin ja niiden välisiin yhteyksiin, ennen kaiken kokoamista yhdeksi eteenpäin annettavaksi virraksi. Luotua virtaa voidaan sen jälkeen käyttää monessakin eri Power BI -raportissa, ilman uusien tietovirtojen luomista. Näin on mahdollista saada tietokannasta niin sanotusti ”yksi totuus” käyttöön, eikä ole tarvetta hallinnoida montaa eri konfiguraatiota yhtä aikaa, jolloin mahdollisten virheiden määrä kasvaa. [14.]



Kuva 1. Tietovirtojen käyttö Azuressa [14]

Tietovirrat eivät oletuksena päivity samaan aikaan kuin tietokanta, josta taulukot haetaan, mutta virralle on mahdollisuus asettaa automaattinen päivitys. On olemassa kahta eri päivitystyyppiä: täysi ja inkrementaalinen. Opinnäytetyössä käytettiin päivitysmuotona ainoastaan täyttä päivitystä.

Täysi päivitys on prosessina hitaampi. Se tyhjentää olemassa olevan kannan täysin ja hakee kaikki tiedot uudelleen. Se toimii erinomaisesti pienempien tietokantojen kanssa, jolloin niiden uudelleenhakuaika ei ole pitkä. [15.]

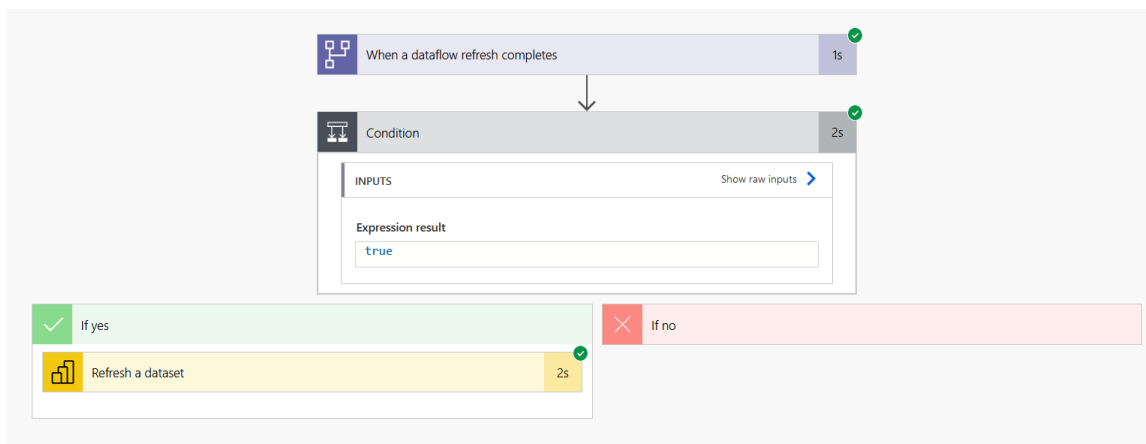
Inkrementaalinen tietovirran päivitys on tarjolla vain Power BI Premium -käyttäjille [15], ja antaa huomattavaa etua isompien tietokantojen kanssa. Inkrementaalinen päivitys lisää taulukkoihin vain rivit, jotka ovat ilmestyneet viimeisimmän päivityksen jälkeen, joten hakujen prosessointiajat pysyvät matalina. [16.] Inkrementaalisisessa päivityksessä kuitenkin tarvitaan jonkinlainen aika-leima. Paras on esimerkiksi rivin luontipäivä, jota tietovirta vertaa edellisen päivitystapahtuman päivänmäärään. [17.]

3.1 Tietojen päivitys ja ajantasaisuus

Työssä käytetty tietovirta päivittyy kerran päivässä aamuyöllä noin kello neljän aikaan, joten täydellistä reaaliaikadataa ei ollut mahdollista saada. Power BI:n työpöytäversiossa työssä käytetyn datan uudelleenlataus on tehtävä manuaalisesti, mutta pilvipalveluversiossa on mahdollista asettaa automaattinen tietokannan päivitys. Päivitys asetettiin tapahtumaan päivittäin kello 5:00, jolloin tietovirta on varmasti ehtinyt päivittyä.

Toinen varteenotettava vaihtoehto olisi ollut hyödyntää Microsoftin Power Automate -työkalua, jossa voisi hyödyntää tietovirran päivitysaikoja. Power Automatella on mahdollista automatisoida toistuvia tehtäviä huomattavan helposti, ja tähänkin käytetty niin kutsuttu ”virta”, eli työkulku olisi ollut yksinkertainen.

Power Automate olisi saanut tiedon tietovirran päivityksen valmistumisesta, mikä olisi sitten käynnistänyt kojelaudassa käytetyn tietokokonaisuuden uudelleenlataamisen. Kuva 2 esittää tämän työkulun toteutuksen Power Automatessa. Jos työkulun suorittamisessa olisi ilmennyt ongelmia, olisi Power Automate lähettänyt sähköposti-ilmoituksen päivityksen epäonnistumisesta. Yksinkertaisuuden vuoksi tehtiin kuitenkin päätös pitää päivitysasetukset lähellä kojelaudaa, eli asetuksia ei myöhemmin tarvitse muuttaa monesta eri sovelluksesta.

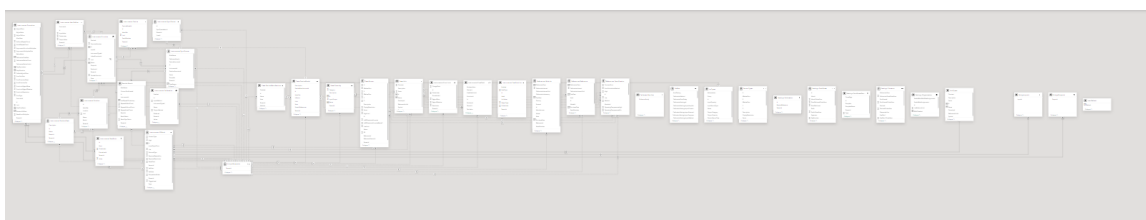


Kuva 2. Työkulku Power Automatessa

Tietovirran käyttöönotto Power BI:ssä oli yksinkertaista; täytyi valita haluttu data ja sen taulukot, jolloin ohjelma hoiti lopun. Sen jälkeen jäi tehtäväksi huolehtia, että taulukoiden väliset suhteet olivat oikein, ja että rivien sisältö oli ymmärrettävässä muodossa.

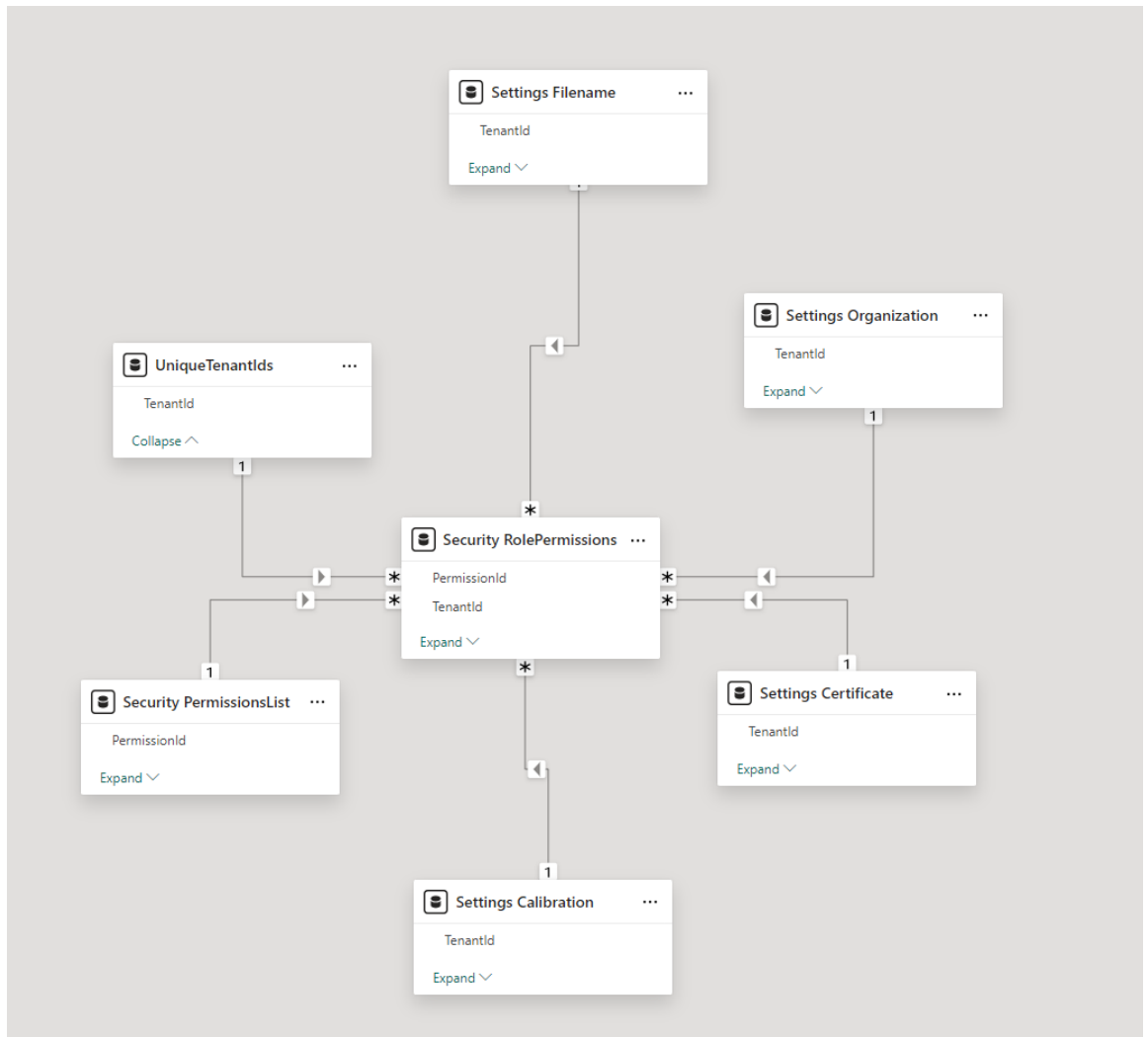
3.2 Taulukoiden keskinäiset suhteet

Power BI osaa automaattisesti päätellä helpoimmat yhteydet taulukoiden välillä ja luoda niille suhteet. Se ei kuitenkaan toimi aivan täysin luotettavasti, vaan saattaa jättää joitain yhteyksiä luomatta. Automaattinen yhdistämisoperaatio saattaa luoda erittäin pitkän rivin yhteyksiä, sillä tietokannassa oli yksi taulukko listaamassa kaikki yksittäiset organisaatiot, kuten Kuva 3 esittää.



Kuva 3. Kaikki Power BI:n löytämät yhteydet taulukoiden välillä

Kuvassa 4 on esimerkki, miten taulukoita voidaan yhdistää selkeästi. Kuvassa luodaan yhteyksiä asetuksia määrittävien taulukoiden välille ja suurin osa yhteyksien välillä käytetyistä tunnistuksista on organisaation uniikki tunnistus. Poikkeuksena on esimerkiksi "PermissionList", josta saadaan jonkin käyttöluvan tunnistus ja sen kautta myös sen nimi.



Kuva 4. Esimerkki yhteyksien muodostamisesta

3.2.1 Ongelma 1: Kaksi yhteyttä yhdellä tunnisteella

Taulukoiden välisissä suhteissa ainoat ongelmat ilmenivät, kun Taulukko A ja Taulukko B täytyi saada yhdistettyä datan käyttämiseksi. Taulukko A sisältää sarakkeet kahdelle eri tunnisteelle: InputId ja OutputId. Taulukko B tietää ainoastaan tunnisteiden, joka yhdistettäisiin Taulukko A:n jaoteltuihin tunnisteisiin. Power BI varoitti tämän tuottavan virheitä, joten ongelmaa täytyi yrittää ratkaista eri tavalla.

Taulukko A otti yhteydet vastaan, mutta ilmoitti ongelmista vasta yhteyksien muututtua aktiivisiksi. Yhteyksien taas täytyi olla aktiiviset, jotta taulukoita olisi voinut suodattaa halutulla tavalla.

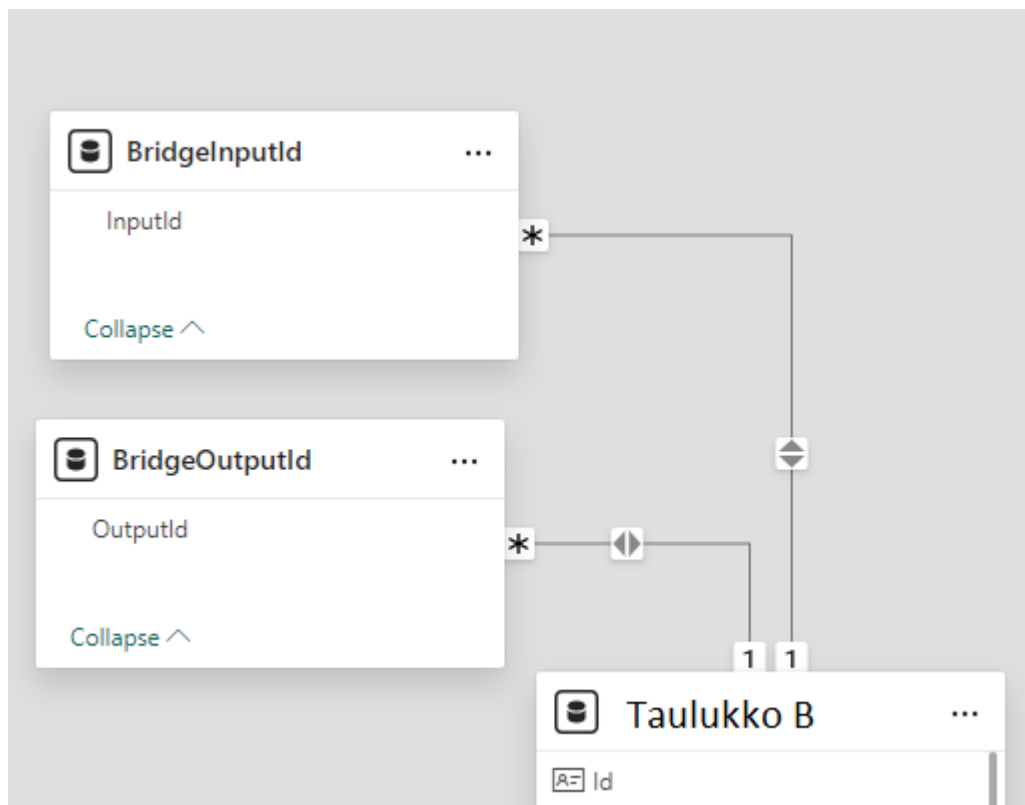
Ratkaisu löytyi eräänlaisen sillan luomisella näiden taulukoiden välille. Taulukosta A otettiin kaikki InputId:t ja luotiin niistä erillinen "BridgeInputId"-taulukko:

```
BridgeInputId = VALUES('Instruments Function'[InputId])
```

Sama toistettiin myös "BridgeOutputId"-taulukolle:

```
BridgeOutputId = VALUES('Instruments Function'[OutputId])
```

Kuvassa 5 nähdään, kuinka siltana toimivista taulukoista tuli Power BI:n silmissä erilliset taulukot, eikä ohjelman puolesta ongelmia syntynyt. Ainut jatko-ongelma oli, että näiden tunnisteiden kautta ei pystynyt enää suodattamaan Taulukko A:n kautta mitään.



Kuva 5. Taulukon yhdistäminen siltojen kautta

Jotta olisi ollut mahdollista suodattaa dataa InputId:n ja OutputId:n kautta, täytyi olla olemassa vain yksi taulukko. Sen Power BI pystyisi yhdistämään taulukkoon, jonka kautta sivun muu suodatus tapahtuisi.

Luotiin kokonaan uusi taulukko "BridgeTable" toimimaan siltana. Ohjelmakoodi 1 sisältää suodatimen tuleville sarakkeille, eli otetaan Taulukko A:sta ainoastaan rivit, joissa DeviceId ei ole tyhjä. Kappaleessa 4.7 Käytetyn datan luotettavuus kerrotaan tarkemmin, miksi näin tehdään.

```

BridgeTable =
VAR FilteredFunction =
    FILTER (
        'Function',
        NOT(ISBLANK('Taulukko A'[DeviceId]))
    )

```

Ohjelmakoodi 1

Seuraavaksi Ohjelmakoodissa 2 valitaan taulukolle yhteensä kolme saraketta alkuperäisestä taulukosta: Input ja OutputId sekä FunctionId. Viimeisellä mahdollistetaan uuden taulukon yhdistäminen haluttuun taulukkoon myöhemmin. Ohjelmakoodista 1 saatu "FilteredFunction"-muuttuja toimii suodattimena, jolloin saadaan valittua vain halutut rivit.

```

VAR BridgeTable =
    UNION(
        SELECTCOLUMNS(
            FilteredFunction,
            "BridgeId", 'Taulukko A'[InputId],
            "FunctionId", 'Taulukko A'[Id]
        ),
        SELECTCOLUMNS(
            FilteredFunction,
            "BridgeId", 'Taulukko A'[OutputId],
            "FunctionId", 'Taulukko A'[Id]
        )
    )

```

Ohjelmakoodi 2

Ohjelmakoodissa 3 palautetaan luodut sarakkeet uuteen taulukkoon, mutta lisätään sarakkeena myös tunniste UnitId. Tunniste viittaa halutun tulon tai lähdön toiminnon käyttämään yksikköön, jonka taulukon kautta saadaan myös tietoon QuantityId, eli suure. Tunniste etsitään suoraan Taulukko B:stä, joten siltataulukkoa ei tarvitse erikseen yhdistää kyseiseen taulukkoon, jolloin välteään mahdolliset aiemmin vaivanneet ongelmat.

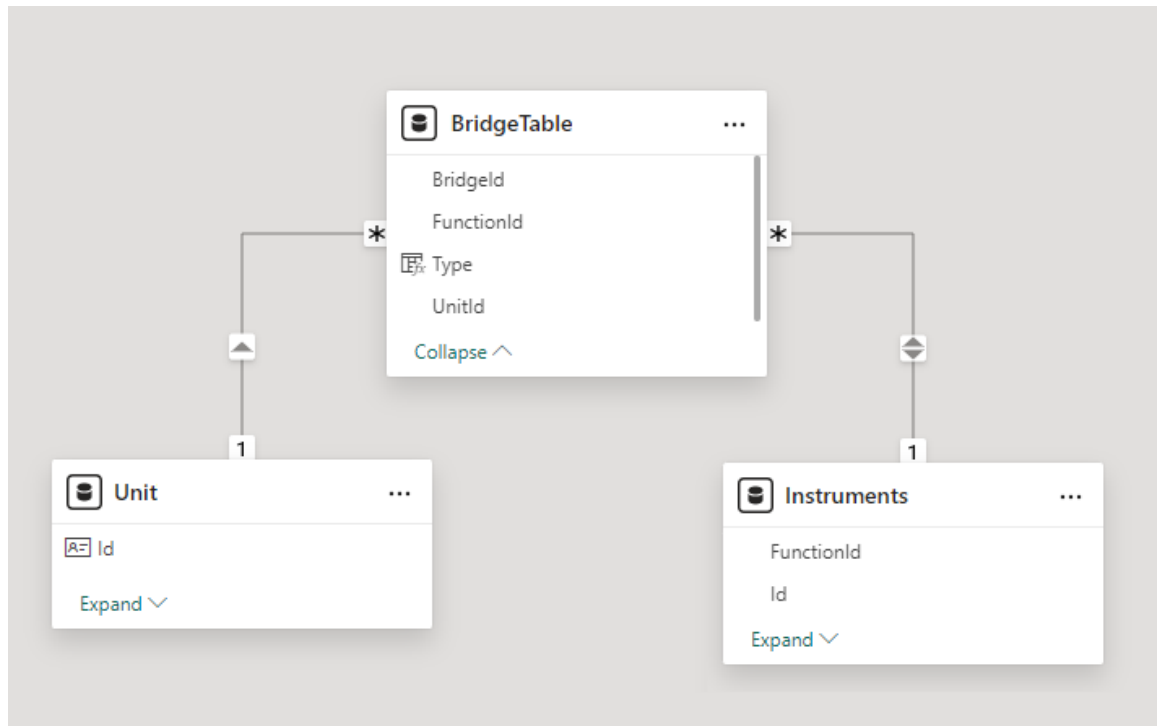
```

RETURN
    ADDCOLUMNS(
        BridgeTable,
        "UnitId",
        LOOKUPVALUE(
            'Taulukko B'[UnitId],
            'Taulukko B'[ID],
            [BridgeId]
        )
    )

```

Ohjelmakoodi 3

Tämän muutoksen kautta alkuperäiset BridgeInputId- ja BridgeOutputId-taulukot voitiin poistaa kokonaan, ja Power BI:n mallinäkyvässä pystyttiin nyt yhdistämään uuden taulukon FunctionId haluttuun taulukkoon (Kuva 6).



Kuva 6. Taulukoiden yhdistäminen toimi lopulta kiertoreittejä pitkin

3.2.2 Ongelma 2: Duplikaatteja datassa

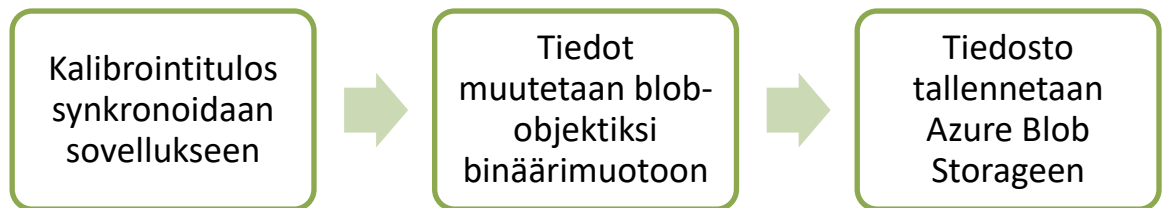
Työn aloittamisesta saakka vaikutti siltä, ettei kalibrointituloksia tallentavaa taulukkoa pystynyt yhdistämään mittaritaulukkoon erään kolmannen taulukon kautta. Mikään ei selkeästi viitannut siihen, ettei niitä pystyisi yhdistämään, mutta Power BI ei kyennyt löytämään yhteyttä kyseisten taulukoiden välille. Yleisin virheilmoitus oli monen saman tunnisteiden olemassaolo jossain sarakkeessa, mikä tarkoitti sitä, että tunniste ei ole uniikki ja sitä ei pitäisi käyttää tietokannassa avaimena. Tämän seurauksena todettiin, ettei näitä taulukoita voi yhdistää yksinkertaisesti, vaan reitti täytyi löytää muualta.

Eräs idea oli hakea Azure Blob Storagesta kaikkien tuloksien blob-tiedostot. Blob, eli ”binary large object” on tiedostotyyppi, joka sopii hyvin suurien tiedostomäärien jakeluun [18]. Sen vahvuutena ovat myös tallennusmuodot, jossa tiedosto ei ole rakenteeltaan ennalta määritelty, joita ovat esimerkiksi kuva- ja videotiedostot. Tämän kaltaiset tiedostot ovat yleensä niin kutsutussa

”block blob” -varastointimuodossa, johon mahtuu jopa 50 000 lohkoa. Muita varastomuotoja ovat niin kutsutut ”append”- ja ”page”-blobit, mutta ne eivät olleet relevantteja työssä. [19, s. 10–13]

Tässä tapauksessa jokaisen blobin uniikki tunniste oli tallennettu tulostaulukon sarakkeeseen, jonka kautta olisi Azure Blob Storagesta löydetty haluttu tiedosto. Itse blobissa taas olisi ollut instrumentin tunniste, jota olisi voinut Power BI:n puolella käyttää yhdistämään tulos- ja instrumenttitaulukot.

Kuva 7 esittää, miten tulos saadaan Azure Blob Storageen. Sen sijaan, että tulokset olisi tallennettu mahdollisesti moneen isoon SQL-taulukkoon, on ollut kannattavampaa muuttaa tiedot blobiksi binäärimuotoon. Tällöin ne voidaan tarvittaessa hakea Azure Blob Storagesta erikseen ja purkaa takaisin luettavaan JSON-muotoon, jonka frontend pystyy sitten tarjoamaan loppukäyttäjälle.



Kuva 7. Kalibrointitulosten kulku Azure Blob Storageen

Tarvittavien tietojen reitti voidaan nähdä kuvasta 8. Ensin Azure Blob Storageen täytyi saada yhteys ja ladata mahdollisesti kaikki tarvittavat blobit. Ladatut tiedostot purettiin Python-skriptillä, joka kääntää binaaritiedostot JSON-tiedostomuotoon. Lopuksi JSON-tiedostoista olisi voinut ottaa ainoastaan työssä tarvittut instrumenttien ja tulosten tunnisteet ja luoda niistä uuden taulukon toimimaan haluttuna siltana.



Kuva 8. Blob-tiedostojen muuttaminen Power BI:hin sopivaksi

Tiedostot olivat Protocol Buffer -kielellä serialisoitu, joka mahdollistaa tehokkaan tavan käsitellä jäsenneiltyä dataa. Protobufin yhtenä isona vahvuutena on tuki monille ohjelmointikielille, jolloin käytettyä dataa pystytään käyttämään helposti erilaisten sovellusten välillä [20]. Binääritiedostot ovat myös tehokkaammin pakattuja kuin esimerkiksi tekstipohjaiset JSON- tai XML-tiedostot, jolloin vaikutukset latausnopeuksiin ovat pienemmät [21].

Ennen koodin kirjoittamista täytyi muuttaa jäsenneilty data proto-tiedostoksi. Proto-tiedostot sisältävät niin kutsutun viestin, joka jäsentee datan sen sisällä. Viesti taas määrittää jokaiselle halutulle kentälle datatyyppin ja nimen sekä järjestysnumeron, joka toimii tunnisteena. Syntaksi on lähellä C++ tai Javaa. [22].

Proto-tiedoston luonnin jälkeen piti se kääntää protocol buffer -kääntäjällä toiseen muotoon. Käännös tapahtui ajamalla seuraava komento tietokoneen komentoliittymässä:

```
protoc -I=$SRC_DIR --python_out=$DST_DIR $SRC_DIR/Result.proto
```

Komento loi "Result_pb2"-nimisen tiedoston samaan kansioon. Vasta tiedoston kääntämisen jälkeen sitä oli mahdollista käyttää binääritiedostojen käsittelemisessä.

Koodi yksinkertaiselle skriptille toteutettiin Python-kielellä. Se ajettiin komentoliittymässä, kuten tavallinen Python-kooditiedosto, ja otti argumentiksi käsiteltävän binääritiedoston nimen. Tiedosto luettiin ja jäsenneilty aiemmin luodun proto-tiedoston perusteella. Saatu tulos otettiin mukaan muuttujaan, josta myöhemmin koodi pystyi sitä käsittelemään uudelleen. Ohjelmakoodista 4 löytyy perustoiminnot tälle, minkä jälkeen siirrytään jäsentelemään halutut tiedot datasta.

```

# Reads the entire blob from a file and prints all
# the information inside
if len(sys.argv) != 2:
    print("Usage:", sys.argv[0], "BLOB_FILE") # The file must be specified
    sys.exit(-1)

# Use the compiled .proto file to map the result
Result = Result_pb2.Result()

# Read the existing file
with open(sys.argv[1], "rb") as f:
    Result.ParseFromString(f.read())

folderPath = "./"
outputPath = "./"

# Parse the blob data
ParseResult(Result)

```

Ohjelmakoodi 4

Ohjelmakoodi 5 on tärkein osa skriptiä. Se sisältää funktion, joka luo `resultEntry`-objektin. Objekti sisältää halutut tiedot, mutta siihen on lisätty varmuuden vuoksi ylimääräistä informaatiota, kuten käytetyn kalibraattorin malli ja versio. Tiedot olisi saanut noudettua tietokannasta suoraan, mutta tällä varmistettiin oikeiden tulosten saanti.

```

def ParseResult(Result):
    resultEntry = {
        "ResultId": Result.result_id,
        "InstrumentId": Result.instrument.instrument_id,
        "Calibrator": getCalibrator(Result.source_info.calibrator),
        "ClientVersion": Result.source_info.client_version,
        "CalibratorVersion": Result.source_info.calibrator_version,
    }

```

Ohjelmakoodi 5

Mukaan haluttiin kalibroitilaitteen malli ja tyyppi, jotka tulivat tiedostossa numeromuodossa. Niiden saamiseksi luettavaan muotoon kutsuttiin `getCalibrator`-funktiota, joka esitetään Ohjelmakoodi 6:ssa. Funktion sisällä on yksinkertainen `switch`-lause, joka palauttaa numeron perusteella kalibroitilaitteen nimen, joka lisätään `resultEntry`-objektiin.

```

# Get a readable version of the calibrator based on the value
def getCalibrator(calibrator):
    calibratorType = "undefined"
    match calibrator:
        case 1:
            calibratorType = "Product 1"
        case 2:
            calibratorType = "Product 2"
        case 3:
            calibratorType = "Product 3"
        case 4:
            calibratorType = "Product 4"
        case 5:
            calibratorType = "Product 5"
        case 6:
            calibratorType = "Product 6"
        case 7:
            calibratorType = "Product 7"
        case 8:
            calibratorType = "Product 8"
        case _:
            print("Something went wrong!")

    return calibratorType

```

Ohjelmakoodi 6

Kun käsiteltävä tiedosto on saatu kokonaan jäsenneltyä, voidaan kutsua `printToJson`-funktiota, joka löytyy koodista Ohjelmakoodi 7. Mukaan laitetaan luotu objekti, jota käytetään luomaan datasta JSON-tiedosto. Tässä versiossa tiedoston luominen kuitenkin jäi alkeelliseksi, eikä se erottele jo olemassa olevien tiedostojen välillä, vaan kirjoittaa suoraan minkä tahansa saman nimisen tiedoston päälle. Käytännössä skriptillä oli mahdollista prosessoida vain yksi objekti kerrallaan.

```

def printToJson(entry):
    json_object = json.dumps(entry, indent=4)

    # write to file
    with open("list.json", "w") as outfile:
        outfile.write(json_object)

```

Ohjelmakoodi 7

Työssä ei kuitenkaan ollut tarpeellista käyttää skriptiä tai Azure Blob Storagea. Tietokantaa tarkemmin tutkiessa selvisi, että tulostaulukon erääseen kolmanteen taulukkoon yhdistävässä tunnisteessa nimenomaan pitäisi olla uniikkeja arvoja, mutta Power BI silti valitti duplikaateista. Muokkaamalla käsiteltävää dataa oli helppo poistaa kaikki muut paitsi tunnisteet, joita oli enemmän kuin yksi. Kävi ilmi, että tietokannassa oli tallennettuna virheellistä dataa.

Tämä oli työn tilanteelle yrityksellekin uusi ongelma, ja heidän oli hyvä olla tietoinen sen olemassaolosta. Virhe ei ainakaan vielä ollut rikkonut mitään tietokannassa, mutta todennäköisesti duplikaatit eivät tulevaisuudessakaan tule aiheuttamaan erityisiä ongelmia. Duplikaattien löytymisen jälkeen osattiin jäljittää niiden alkuperä ja pystyttiin toteamaan monistuneiden tunnisteiden olemassaolon olevan lähinnä inhimillinen virhe, eikä virheellisen koodin syytä. Viikkojen ajan kestänyt ongelmanratkaisuprosessi ratkaistiin yhdellä muutoksella taulukkoon asetettuihin kyselyihin.

3.3 Tietokannan ymmärrettävyys

Tietovirran kautta saatu tietokanta oli juuri siinä muodossa, missä taustajärjestelmä sen käsittelee. Tämä tarkoittaa, että jotkin sarakkeet saattavat sisältää pelkkiä numeerisia arvoja tai GUID-merkkisarjoja, joista ei voi silmämääräisesti päätellä, mitä ne tarkoittavat. Tällaiset arvot täytyi saada luettavaan muotoon ennen niiden käyttämistä kaavioissa.

Tietokannan GUID-merkkisarjoihin ja muihin numeerisiin arvoihin ei ollut olemassa minkäänlaista dokumentaatiota, vaan luotettiin koodin kommentoivan itsensä. Tämä tarkoitti sitä, että haluttu ymmärrys piti etsiä manuaalisesti backend-koodista. Esimerkiksi etsittiin, mitä tarkoittaa numeroarvo "7" taulukossa tuloksen tilasta. Koodista sarakkeen nimeä hakemalla saattoi löytää määritellyn enumin, jonka seitsemäs arvo vastasi tekstijonoa "Fail" eli epäonnistunut.

Tämän löytyessä pystyttiin Power BI:ssä muokkaamaan sisäänrakennetulla Power Query -editorilla kaikki sarakkeen "7"-arvot vastaamaan tekstijonoa "Fail". Tämä täytyi toistaa jokaiselle enum- ja GUID-arvolle, jota oli tarkoitus käyttää näyttöön tulostuvana tekstinä. Esimerkiksi rooli-pohjaisen käyttöoikeuksien hallinnointiin vaadittavat oikeudet oli tallennettu tietokantaan GUID-arvona, ja niiden käytön seuraaminen oli yksi kojelautaan haluttu ominaisuus. Rivien käyttämistä varten täytyi uudelleennimetä jopa neljäkymmentä eri arvoa.

4 Kojelaudan toteutus

Kojelauta on jaettu yhteensä viidelle eri sivulle, joista jokainen käsittelee hieman eri tilastoja. Ensimmäinen sivu sisältää kaiken tärkeimmän jo ensisilmäyksellä, eli tilastot kalibroinneista, niiden tuloksista ja työkaluista, joilla tulokset ovat luotu.

Toinen sivu esittää instrumenttityypit hieman tarkemmin, ja kaavioita pystyy suodattamaan muun muassa instrumentin tyyppien mukaan. Samat kaaviot löytyvät myös ensimmäiseltä sivulta, mutta niille tarkoitetulla sivulla on mahdollista porautua niihin paremmin, esimerkiksi tutkimalla, minkälaisia arvoja ja suureita kalibroidaan.

Menetelmiä varten luotu sivu sisältää muutaman kaavion kalibroinnissa käytetyistä testipisteistä sekä kytkintyyppisten instrumenttien testikerroista. Kalibrointipisteisiin liittyvä kaavio oli suuri, joten kategorialle tehtiin täysin oma sivu.

RBAC-sivulla esitellään pelkästään roolipohjaisen käytön hallinnan toimintaan liittyviä käyttöluoja, jotka on selitetty tarkemmin myöhemmin.

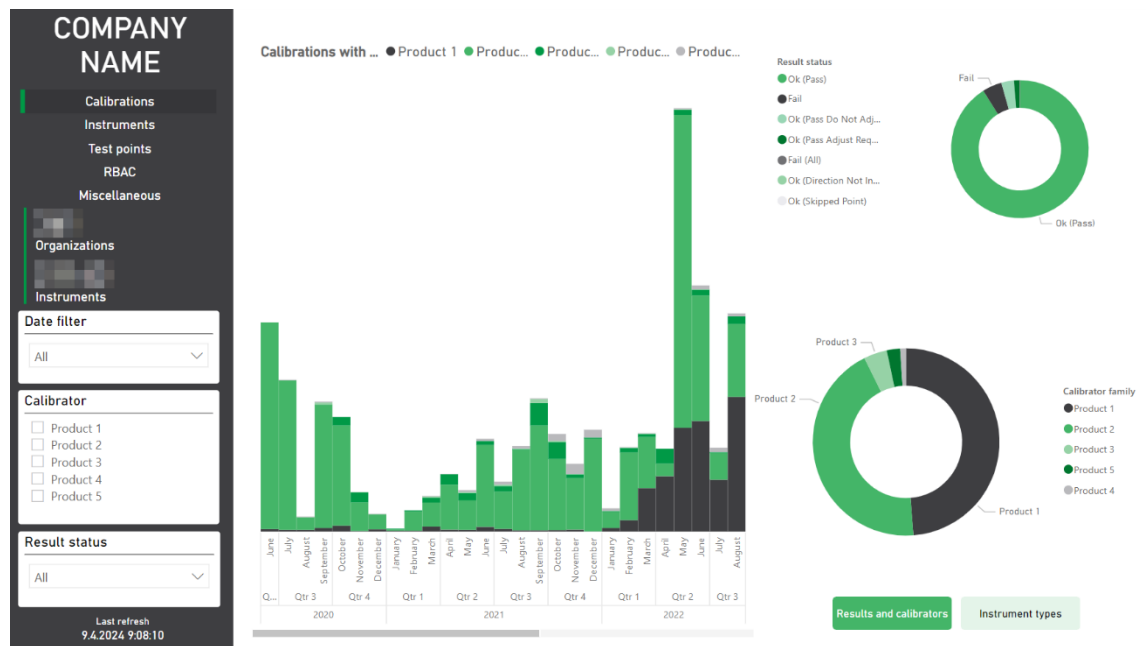
Viimeinen sivu sisältää sekalaisia kaavioita, jotka eivät sopineet muualle. Se käsittää muun muassa kaaviot kalibrointien jäljitettävyydestä, instrumenteissa käytettävistä vapaista kentistä sekä sovelluksessa löytyvän kansiorakennetoiminnon käytöstä.

Tekstin joukossa esiintyvistä kuvista on tarkoituksella poistettu kaikki esiintyvät luvut ja tuotteiden nimet, mutta ne ovat työn tilanneelle yritykselle luovutetussa kojelaudan versiossa mukana. Täten joitain elementtejä saattaa puuttua tässä esitetyistä versioista, ja jotkut kaaviot saattavat näyttää yksinkertaisemmilta, kuin mitä ne oikeasti ovat. Työ ei myöskään esitä tuotteen suoritus-tasoa tai tehokkuutta, vaan se on kehittäjille suunnattu yleiskatsaus tuotteen kokonaiskäytöstä.

4.1 Sivun 1: Kalibroinnit

Ensimmäinen sivu kalibrointitilastoista on koko kojelaudan etusivu (Kuva 9). Siinä esitetään kehittäjiä eniten kiinnostavat faktat: miten kalibrointikertoja on kerääntynyt tietyllä aikavälillä, mitä laitteita on käytetty kalibroinnista ja minkälaisia tuloksia niillä on saatu. Sivun reunasta löytyy navigaatio sekä organisaatioiden että instrumenttien määrä. Sivupaneelin suodattimet pätevät koko

sivulla, mutta jokaista aukeaman kuvaajaa voi erikseen porata syvemmälle nähdäkseen yksityiskohtia tarkemmin esimerkiksi tietyn kalibrointilaitteen kannalta.



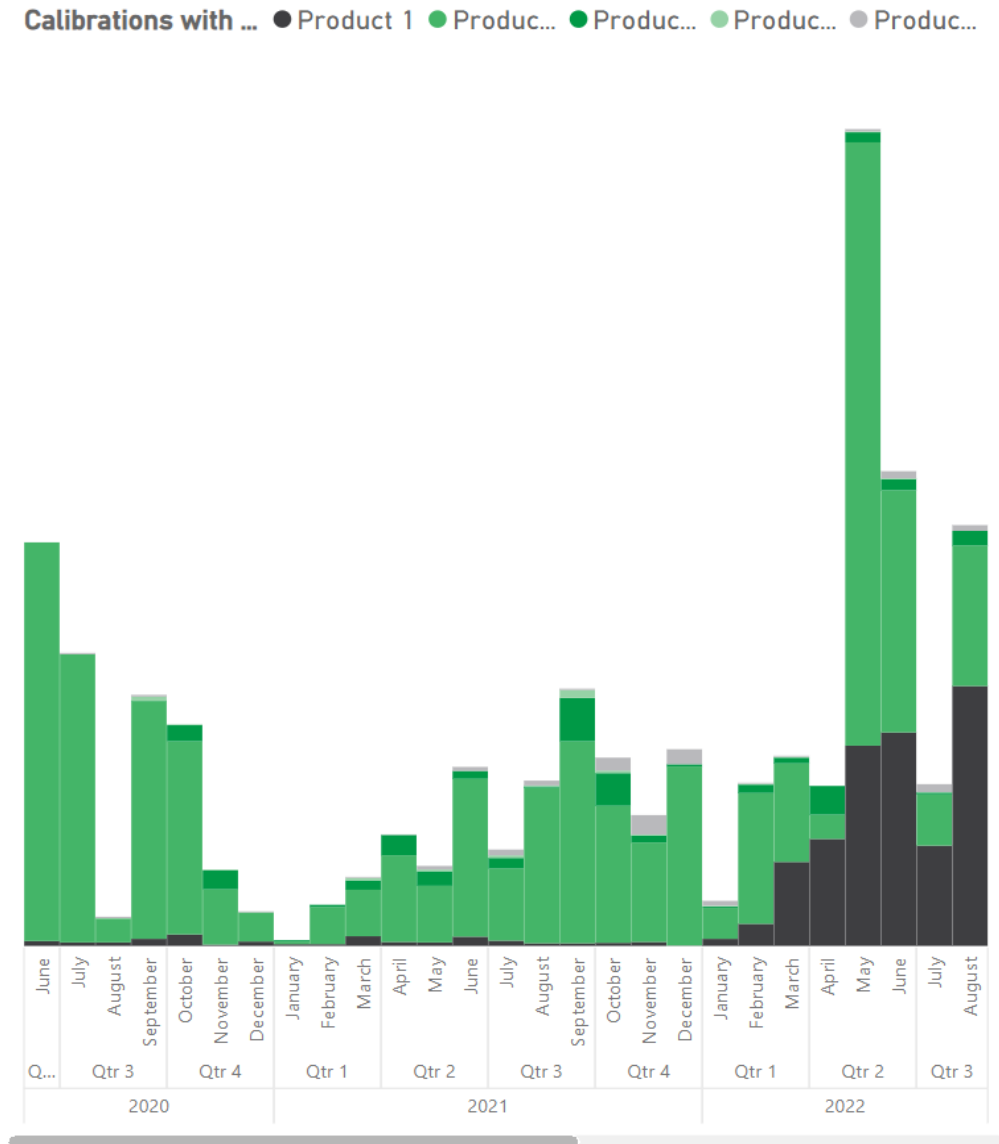
Kuva 9. Ensimmäinen sivu kokonaisuudessaan

Jokaisen sivun sivupaneelissa on myös päivämäärä ja aika, jolloin tietokanta kojelaudalle on viimeksi päivitetty. Normaalisissa tilanteissa tätä tietoa ei pysty helposti löytämään, eikä peruskäyttäjällä tähän pitäisikään olla oikeuksia. On kuitenkin tarkoituksena nähdä mahdollisimman tuoretta informaatiota, joten päivämäärä ja kellonaika on hyödyllistä olla esillä jossain. Ajanhetken tallennus on toteutettu yksinkertaisella taulukolla, joka ottaa sen hetkisen kellonajan päivityksen tekemältä palvelimelta tai paikalliselta laitteelta:

```
= DateTimeZone.UtcNow()
```

Valitettavasti kerran päivässä tapahtuvassa päivityksessä se ei kuitenkaan pysty ottamaan kantaa siihen, onko itse tietovirta päivitetty. Ongelman voisi ratkaista käynnistämällä taulukoiden uudelleenlataus tietovirran päivittäessä, kuten aikaisemmassa kappaleessa mainittiin vaihtoehtoiseksi ratkaisuksi.

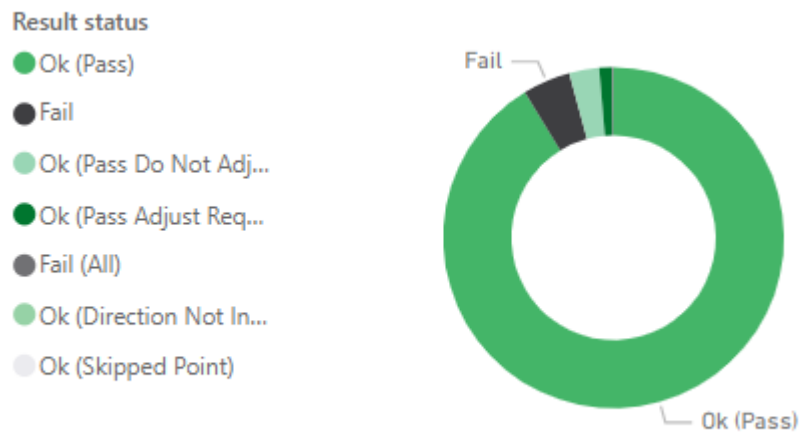
Sivulla ensimmäisenä ja tärkeimpänä kaaviona (Kuva 10) on kalibrointien määrä ja laite, jolla kalibrointityö on suoritettu. Jo nopealla vilkaisulla saa kokonaiskuvan määrästä, ja sivupaneelista löytyy suodattimia, joilla voidaan tarkentaa tiedon esitystapaa. Esimerkiksi voidaan tarkistaa, onko tuote 1:n kalibrointimäärät nousseet viime vuodesta, kuukaudesta tai jopa viikosta.



Kuva 10. Kuvaaja toteutuneista kalibroinneista

Kaavioon sopii pylväskaavio, sillä se esittää käyttäjälle nopeasti kokonaiskuvan tietyn ajanjakson kalibroinneista. Pylväiden avulla on myös helppo verrata lukumäärien kehitystä. [6, s. 254, 255.] Eri väreillä merkatut pylväiden alueet esittävät jokaisella laitteella tehdyt kalibroinnit, jotka kertovat tuotteiden käytöstä suhteessa kokonaisuuteen.

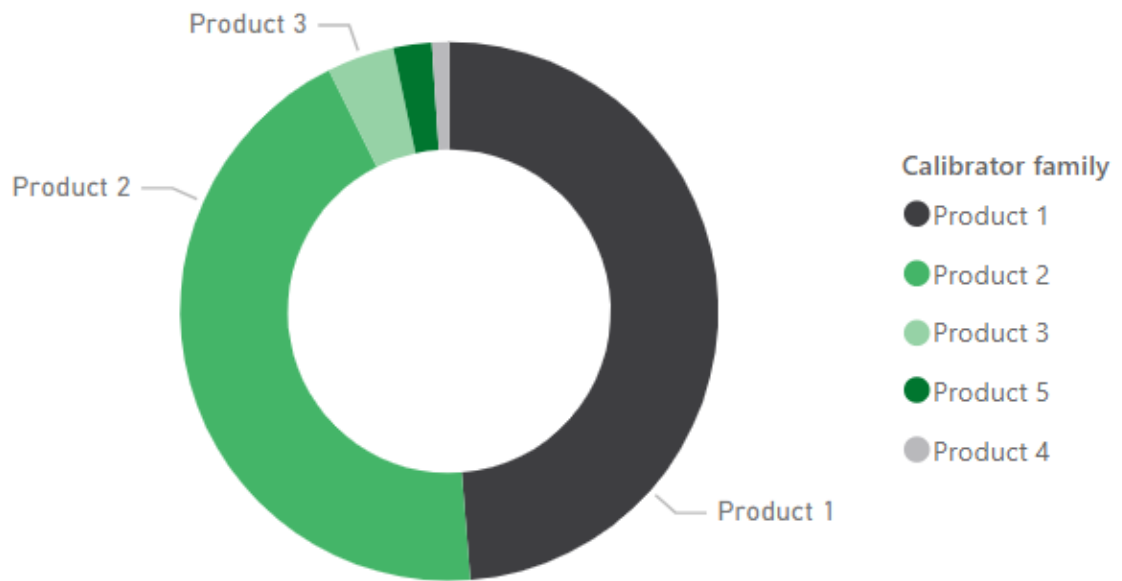
Tuloksia esittävä kaavio (Kuva 11) on donitsikaavio, jossa väri vaihtoehtojen suunnittelu oli haastavaa. Tuloksia oli tietokannassa ja backendistä löytyvissä enumeissa monen tyyppisiä, mutta vain muutamaa niistä oli yleisesti käytössä. Vihreä yleisesti tarkoittaa onnistunutta arvoa ja punainen epäonnistunutta, mutta punaista väriä vältettiin. Kaikki läpi menneet arvot merkattiin jollain vihreän sävyllä ja epäonnistuneet mustalla. Tärkeimmät ja tietokannassa eniten ilmeneville arvoille annettiin selkeimmät värit, ja lopuille jätettiin haaleimmat sävyt.



Kuva 11. Donitsikuvaaja tuloksista

Tuloksia suodattamalla on mahdollista nähdä nopeasti, minkälaisia tuloksia eri kalibrointilaitteet tuottavat. Esimerkiksi voidaan suodattaa sivua tuote 4:lla, jolloin nähdään sen tuottavan erityisen paljon epäonnistuneita kalibrointeja esimerkiksi verrattuna tuote 1:een.

Sivulta näkee (Kuva 12) myös yleisen jakautumisen laitteissa, joilla kalibrointi on luotu ja synkronoitu tietokantaan. Suodattimia käyttämällä voi nähdä laitteiden käytön prosenttiarvoja verrattuna toiseen valittuun kalibrointilaitteeseen, mikä on mahdollisesti hyödyllistä tietoa kyseisten tuotteiden kehittämisessä. Miksi esimerkiksi tuotteet 1 ja 2 ovat paljon suosittumia kuin tuote 3? Miten tuote 3:sta voitaisiin parantaa? Myynti- ja markkinointiosastot saavat kojelaudasta myös hyötyä, jolloin voidaan miettiä, miten pystyttäisiin kohdistamaan laitteiden myyntiä parhaiten. Esimerkiksi jonkin tuotteen ostajalle voisi olla kannattavaa mainostaa yrityksen pilvipalvelutuetta enemmän.

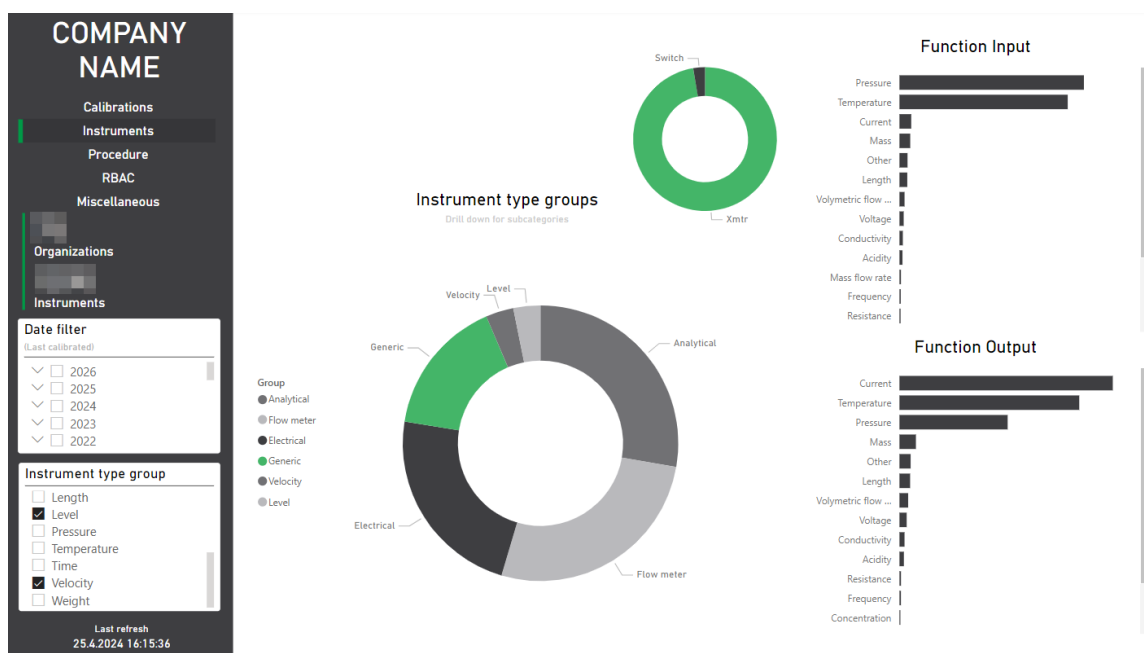


Kuva 12. Kalibrointilaitteiden jakautuminen

Kuvassa (Kuva 9) oikean puolen donitsikuvaajien alla sijaitsee sivun toinen navigaatio. Power BI:n kirjanmerkkien avulla on mahdollista luoda yhdelle sivulle elementti, jossa kaavioita voidaan vaihtaa toiseen napin painalluksella. Tässä navigaation yllä olevat kaaviot vaihdetaan kaavioiden instrumenttien tyypeistä (Kuva 12), jolloin niiden kautta on myös mahdollista suodattaa kalibrointitaulukkoa (Kuva 10). Kirjanmerkinavigaatio on lähinnä käyttäjille, jotka tarkastelevat kojelaudaa omilta päätelaitteiltaan. Se ei kuitenkaan toimi esimerkiksi kojelaudassa, joka on upotettu diaesitykseen, sillä mahdollisuus vuorovaikutukseen kojelaudan kanssa katoaa.

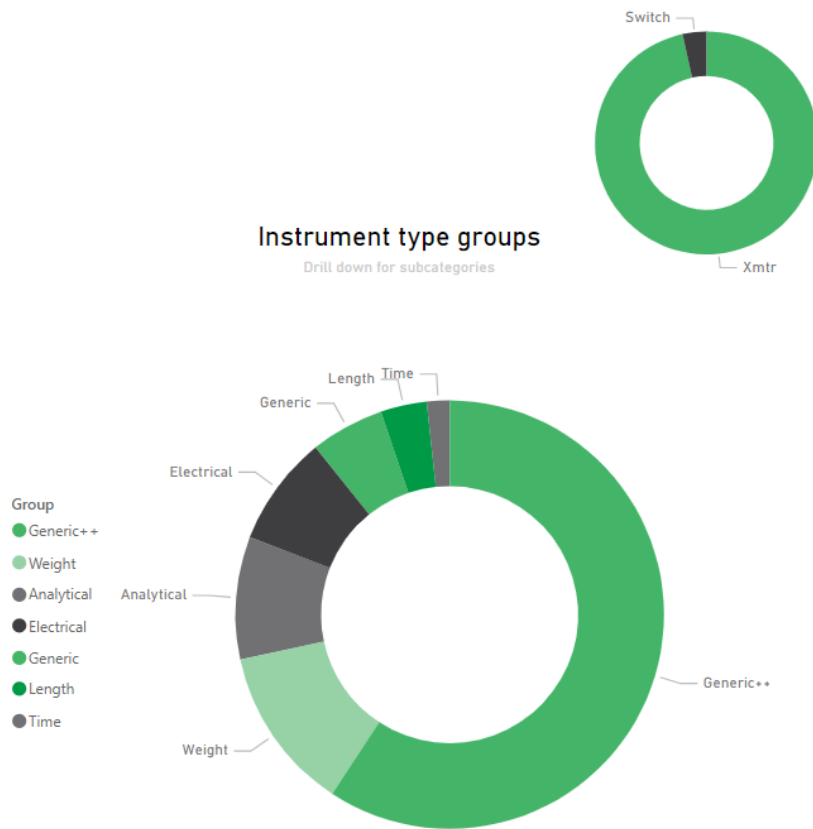
4.2 Sivun 2: Instrumentit

Instrumenteista erikseen kertova sivu (Kuva 13) ei juurikaan tarjoa uutta tietoa toimintojen tulon ja lähdön suureita lukuun ottamatta. Instrumenttityyppiryhmien kaaviot (Kuva 14) löytyvät jo ensimmäiseltä sivulta, ja tässä ne ovat tarjottu vain suurempana. Isomman koon ja sivupaneelien suodattimien kautta instrumentteja on mahdollisuus tutkia tarkemmin ja esimerkiksi porautua eri instrumenttiryhmien tarkempiin tyypeihin. Kuva 15 näyttää version kaaviosta, jossa on porauduttu painetyyppiin.

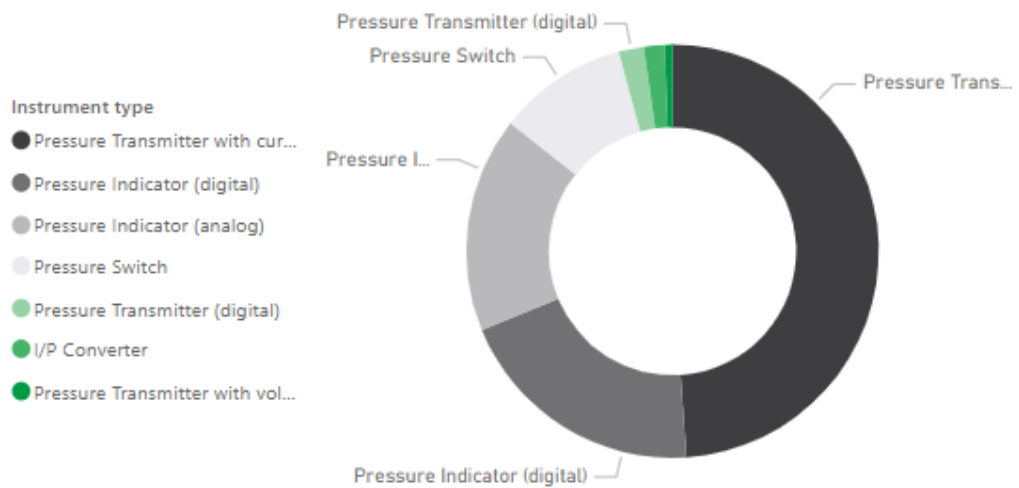


Kuva 13. Toinen sivu kokonaisuudessaan

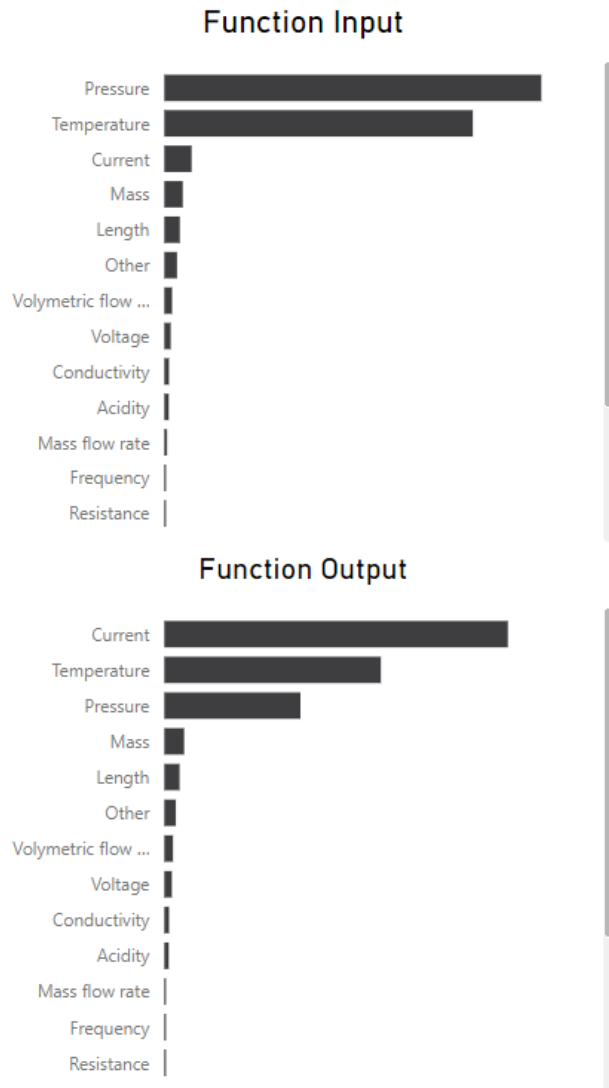
Instrumenttikaavioiden laitto omalle sivulleen piti rakenteen selkeänä. Se tarjoaa myös hyvän mahdollisuuden luoda instrumenttien toiminnon tulojen ja lähtöjen suureisiin liittyvän kaavion (Kuva 16), josta nähdään, miten eri suureita yleisesti käytetään. Esimerkiksi instrumentti voi olla tyyppiltänsä lämpötiläohitin, joka viittaa tulon suureen olevan lämpö ja lähdön suureena olevan virta. Käytössä on myös yleisiä instrumentteja, jotka eivät suoraan kerro, minkälaisista tulosta ja lähtöä toiminnossaan käyttävät. Niiden yhdistelmä on joku muu kuin valmiiksi määritellyt tyypit, eikä ohjelma täten tunnista sitä. Järjestelmä kuitenkin tukee myös tällaisten laitteiden luonnin tietokantaan, eikä aiheuta erityisiä ongelmia.



Kuva 14. Instrumenttityypiryhmät



Kuva 15. Instrumenttityypit



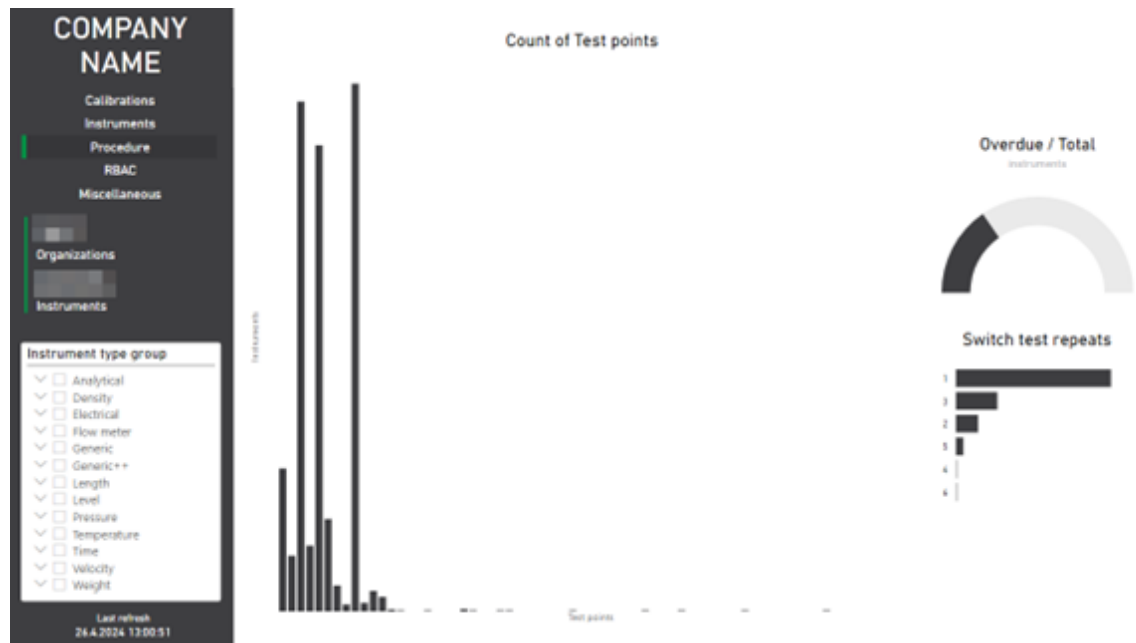
Kuva 16. Tulon ja lähdön suuret

Näiden yleisten instrumenttien tulojen ja lähtöjen tunnistaminen on hyödyllistä, sillä se antaa tuotekehitystiimeille mahdollisuuden tietää, minkälaisia tulojen ja lähdön arvoja kannattaisi käyttää ohjelmistotestauksessa. Suodattimien avulla voi esimerkiksi valita ainoastaan yleistyyppejä olevat instrumentit ja tutkia minkälaisia suureita käytetään. Tulon ja lähdön suureisiin voi myös poutua syvemmälle, jolloin on mahdollista nähdä tarkemmin, minkälaisia yksiköitä on käytössä.

4.3 Sivun 3: Menetelmät

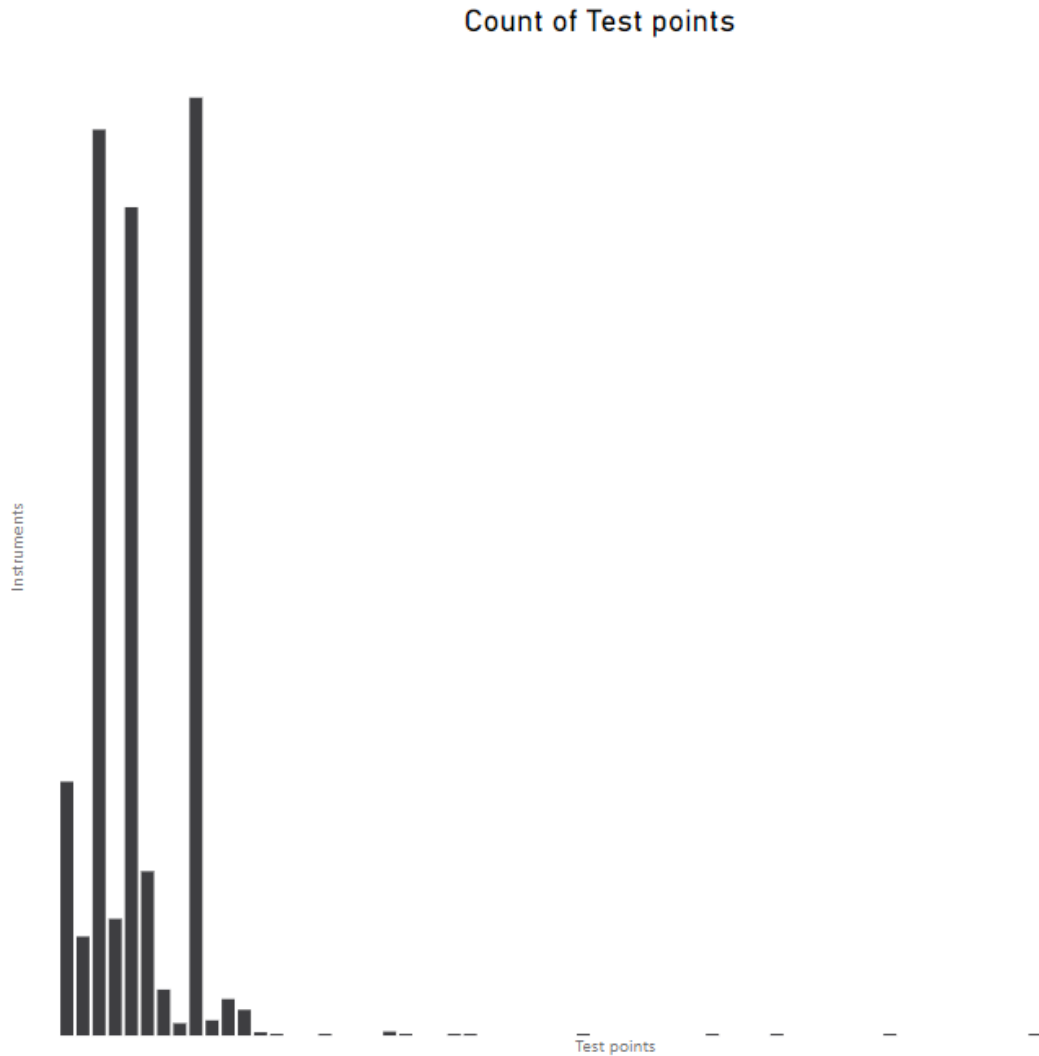
Sivulla (Kuva 17) esitetyt menetelmät tarkoittavat kalibroinnissa käytettyjä määriteltyjä menetelmiä. Esimerkiksi kalibrointipisteiden asettaminen on yksi useimmiten asetetuista menetelmistä.

Kalibrointipisteitä on yleisesti joko kolme, viisi, kuusi tai yksitoista, mutta määrän voi mukauttaa myös haluamaansa arvoon. Kalibroinnin aikana testataan arvot jokaisen pisteen kohdalla, joten mitä enemmän testipisteitä, sitä tarkempi kalibrointi on.



Kuva 17. Kolmas sivu kokonaisuudessaan

Kuvaajasta (Kuva 18Kuva 17) nähdään, että suurin osa käyttäjistä käyttää valmiiksi määriteltyjä testipisteitä kolme, viisi ja yksitoista, mutta muitakin arvoja on käytössä. Pylväistä näkee myös ääripäät, jotka ovat kehittäjille kiinnostavia. Minkälainen instrumentti on kyseessä, joka tarvitsee enemmän kuin kuusikymmentä kalibrointipistettä? Kojelaudan käyttäjällä on mahdollisuus valita kuvaajasta haluamansa pylväs ja siirtyä edelliselle sivulle tutkimaan suodatettuja instrumentti-datasta kertovia kuvaajia.

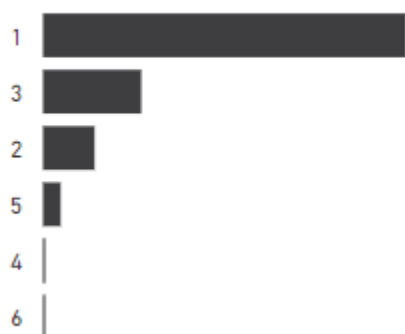


Kuva 18. Kuvaaja testipisteistä

Testipisteiden kuvaajaksi harkittiin ensin kuplakaaviota, jossa vaaka-akselilla olisi instrumenttien lukumäärä ja pystyakselilla testipisteiden määrä. Aluksi kuplakaaviot vaikuttavat näyttävältä esitystavalta [10, s. 406], mutta myöhemmin myös työssä huomattiin sen olevan liian vaikeasti ymmärrettävä. Kuplakaavio harvoin toimii hyvin esittämään tietoa ja usein perinteinen pylväskaavio toimiikin paremmin [10, s. 407].

Kuva 19 esittää kytkimen toiminnasta. Kytkin instrumenttina on erilainen tyyppillisestä instrumentista. Esimerkiksi virtakytkin instrumenttityyppinä ei käytä testipisteitä kalibroinnissa, vaan käytössä on toistojen määrä. Kuvaajasta nähdään, että suurin osa kalibroinneista tehdään vain kerran ja yli neljän toistokerran määrä on melko pieni. Kuutta toistokertaa enempää ei löydy.

Switch test repeats



Kuva 19. Kuvaaja kytkininstrumenttien testitoistoista

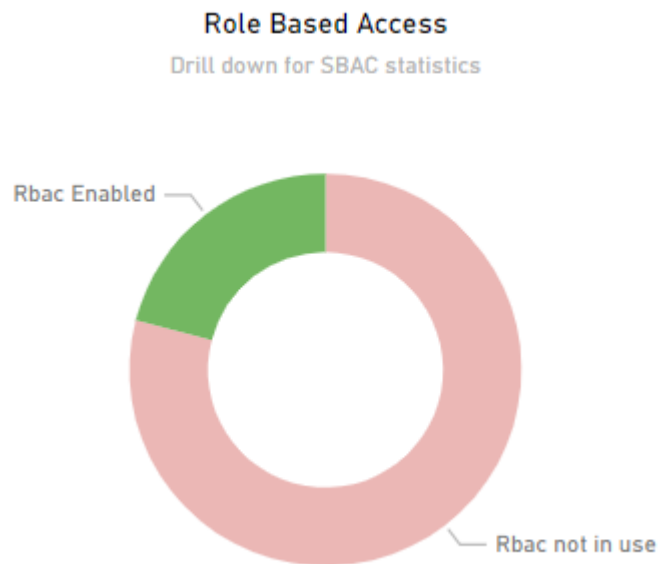
4.4 Sivu 4: Roolipohjainen käytön hallinta

Työn ollessa noin puolessa välissä, julkaistiin RBAC uutena toimintona tuotteen sivustolle. Omaisuuden käyttödatan näkeminen kiinnosti tuoteomistajaa. Lisätty toiminto oli pitkään kehityksessä ja muutti sovelluksen rakennetta kokonaisvaltaisesti. Oli ymmärrettävää, että kehittäjät halusivat tietää, oliko toiminto otettu käyttöön asiakkaiden organisaatioissa. Neljännellä sivulla (Kuva 20) esitetään tietoa toiminnon käytöstä.



Kuva 20. Neljäs sivu kokonaisuudessaan

Ensimmäisessä versiossa oli yksi donitsikuvaaja (Kuva 21), jossa näytettiin kokonaismäärä organisaatioista, jossa RBAC oli joko käytössä tai ei. Käyttäjä olisi voinut käyttää porautumistoimintoa selataksaan syvemmälle dataa, mutta palautteen perusteella kävi ilmi, etteivät normaalit käyttäjät ymmärtäneet taikka osanneet käyttää Power BI:tä tarpeeksi sulavasti. Käyttäjäystävällisyyden vuoksi kuvaajat jaettiin kahdeksi eri kuvaajiksi pelkän yhden sijaan (Kuva 22).



Kuva 21. Ensimmäinen versio RBAC-kaaviosta

Lopullisessa versiossa kuvaajia on kaksi, yksi rooliperustaiselle ja toinen laitosperustaiselle käytön hallinnalle. Näin käyttäjien ei tarvitse tietää mahdollisuudesta porautua syvemmälle dataan ja kokonaiskatsauksen saa ensisilmäyksellä.

Kuva 22 tuo esille huomattavan eroavaisuuden roolipohjaisten käytön hallinnassa verrattuna laitoskohtaiseen pääsynhallintaan. Kuvaajassa käytetyssä datassa oli alun perin vain taulukko, josta käyttöoikeuksiin ja asetuksiin liittyvä data oli peräisin, mutta taulukosta ei löytynyt jokaista luotua organisaatiota. Tällöin kuvaajakaan ei näyttänyt todellista suhdetta asetuksen käytön kannalta. Kokonaismäärä organisaatioista oli kuitenkin laskettu toiseen taulukkoon etsimällä kaikki uniikit tunnisteet muista taulukoista, mistä listan pystyi lisäämään Power Queryn avulla organisaatioasetukset-taulukkoon. Näin saatiin luotettavampi kaavio.



Kuva 22. RBAC-kaavioiden lopullinen versio

Sekään ei kuitenkaan esittänyt koko tilannetta totuudenmukaisesti. Ongelmana oli kaikki ei toiminnassa olevat organisaatiot, joita oli noin arviolta tuhat kappaletta. Ne oli joskus luotu tietokantaan, mutta kyseisiä organisaatioita ei ollut koskaan käytetty sivuston päätarkoitukseen. Täten ei ollut järkevää ottaa niitä kuvaajiin mukaan, vaikka taulukossa esiintyivätkin.

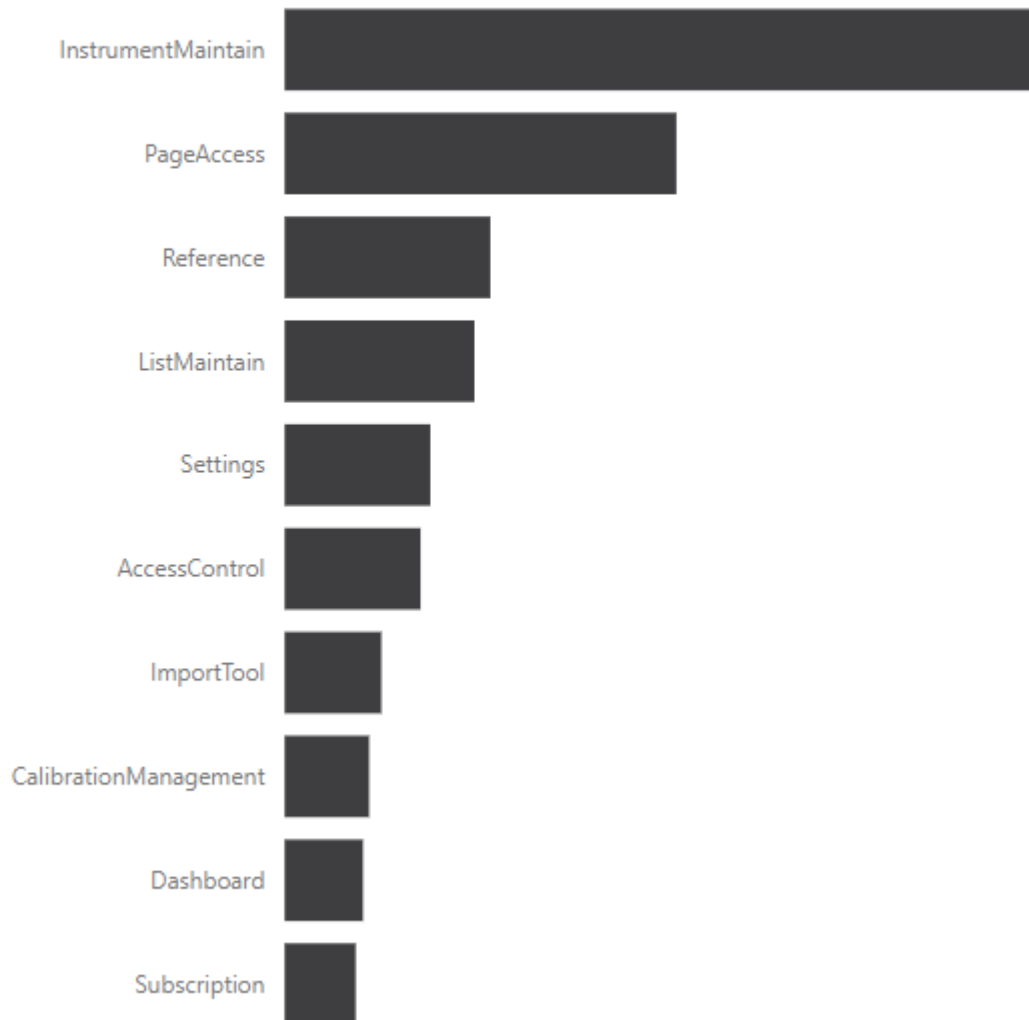
Paras vaihtoehto tähän olisi ollut suodattaa kokonaan pois organisaatiot, joilla ei tällä hetkellä ole tai ei ikinä ole ollutkaan aktiivista tilausta. Käytössä olevalla datalla tämä ei kuitenkaan ollut mahdollista, sillä organisaatioiden täytyi pysyä anonymisoituna. Sopivaksi keskitieksi valittiin tilastojen suodatus sovelluksen käytön perusteella. Jos organisaatio oli luonut yhdenkin tuloksen eli tuotetta oli käytetty edes kerran oikeaan tarkoitukseen, se voitiin laskea mukaan kaavioihin.

Roolipohjaiselle käytön hallinnassa luotuihin rooliryhmille oli mahdollista antaa käyttöoikeuksia. Näin sovelluksen toimintoja voidaan rajata eri käyttäjäryhmille, eikä esimerkiksi peruskäyttäjä pääse poistamaan instrumentteja tai muuttamaan organisaation asetuksia. Asetusten käyttö kiinnosti kehittäjätiimiä ja tuoteomistajaa, joten kaavion (Kuva 23) lisäystä pyydettiin.

Kaaviossa hyödynnettiin Power BI:n porautumistoimintoa, sillä kaikki neljäkymmentä asetusta erikseen olisivat luoneet erittäin vaikeasti luettavan kaavion. Yksittäiset asetukset eivät olleet niin tärkeitä näyttää raporttia lukevalle, ja otsikon alla on vihje porautumisesta tarkempaa informaatiota hakevalle.

Permission groups

Drill down for individual permissions

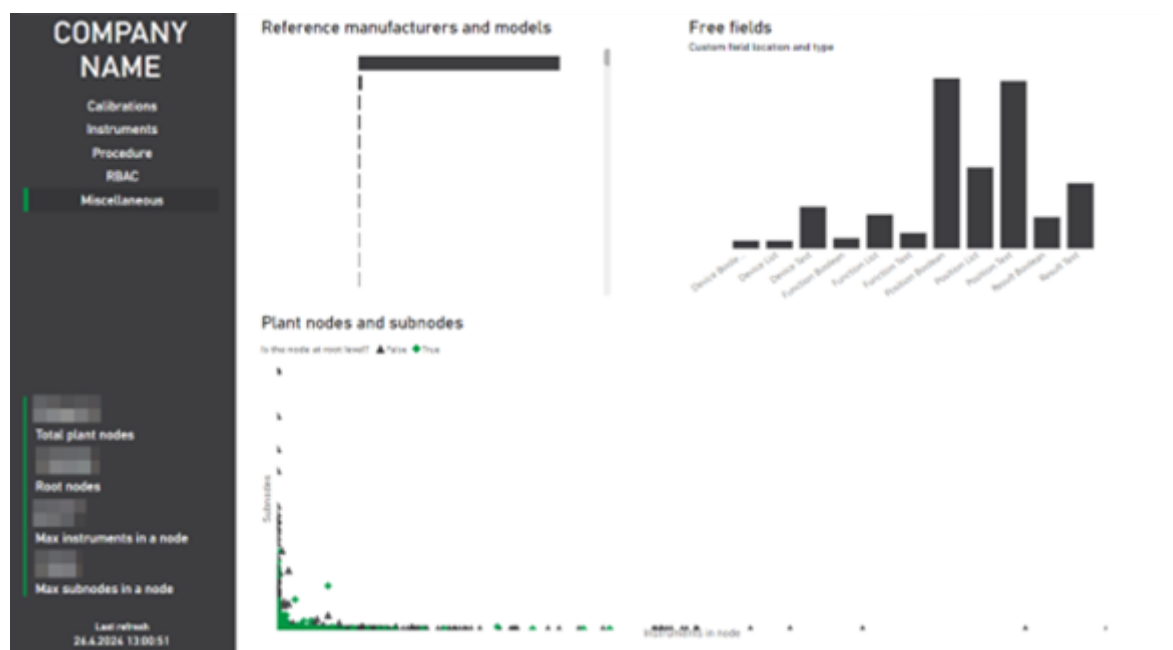


Kuva 23. Käyttöoikeudet ja niiden jakautuminen

Kaavion tyylinä olisi voinut olla myös pylväskaavio, joka olisi näyttänyt esteettisesti siistimmältä sivulla. Pylväskaavion isoin heikkous on kuitenkin X-akselilla olevien kategorioiden tekstien luetavuus [1, luku 6 Visualizing Data]. Visuaalisen selkeyden takia päädyttiin käyttämään palkkikaaviota, jossa tekstit löytyvät palkkien vasemmalta puolelta.

4.5 Sivun 5: Sekalaiset

Viimeinen sivu (Kuva 24) koostuu sekalaisesta joukosta kaavioita, jotka eivät oikein sopineet mihinkään muuhun kategoriaan. Tilastaan huolimatta olivat nämä toivottuja lisäyksiä ja lopulta sisältävätkin hyödyllistä informaatiota.



Kuva 24. Viides sivu kokonaisuudessaan

4.5.1 Referenssit

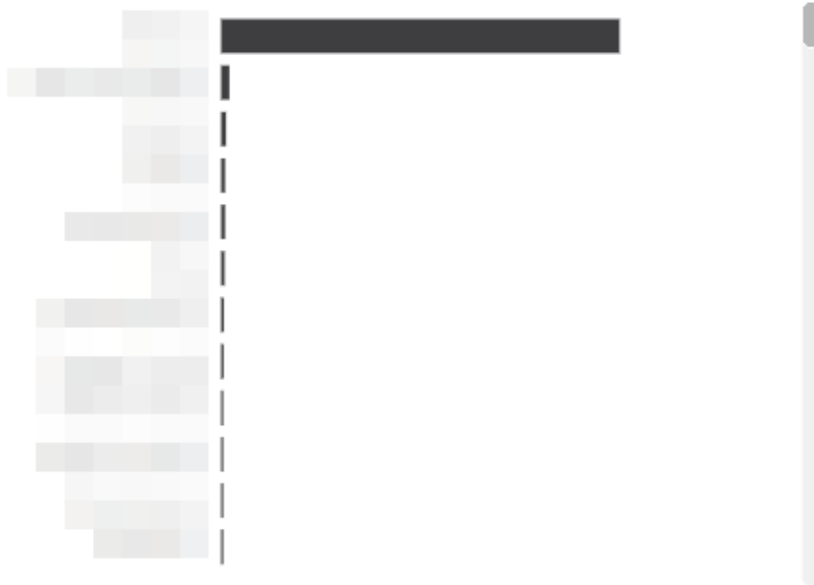
Ensimmäinen kaavio liittyy kalibroinnissa käytettyihin referensseihin. Referenssit liittyvät kalibrointien jäljitykseen, joka on tärkeä osa koko prosessia. Jäljitettävyys tarkoittaa sitä, että voidaan varmistaa mittausinstrumentin olleen kalibroitu laitteella, jonka tarkkuus on taas varmistettu vielä tarkemmilla laitteilla. [23.] Tätä tarkempaa laitetta kutsutaan referenssilaitteeksi [24].

Normaalia kalibrointilaitetta tarkemmat laitteet löytyvät yleensä sertifioiduista kalibrointilaboratorioista, jossa löytyy myös muu tarvittava hyvinkin erikoistunut välineistö, jota ei tavallisilta kalibrointia suorittavilta yrityksiltä löydy. Ketju jatkuu vielä tarkempiin kansainvälisiin kalibrointilaboratorioihin, jolloin jäljitettävyys on taattu, ja voidaan olla varmoja tulosten oikeellisuudesta. [25.]

Työn tilaaja valmistaa myös kalibrointilaitteita, ja heidän tuotteensa on referenssien valmistajissa ylimpänä. On kuitenkin yrityksen etuna tietää, mitä muita laitteita heidän sovelluksensa kanssa käytetään.

Kaavion (Kuva 25) perusnäkymässä näkyvät ainoastaan valmistajat. Tässäkin on porautumisvaihtoehto jokaiseen listattuun valmistajaan, jolloin näkymään saa esille tarkemmat tiedot käytetyistä malleissa. Tilastot kuitenkin kertovat vain sovellukseen luoduista referensseistä, eivätkä ota kantaa niiden käyttöasteeseen. Tulevaisuudessa tämä kuitenkin voisi olla kojelautaan lisäkehitysmahdollisuus.

Reference manufacturers and models

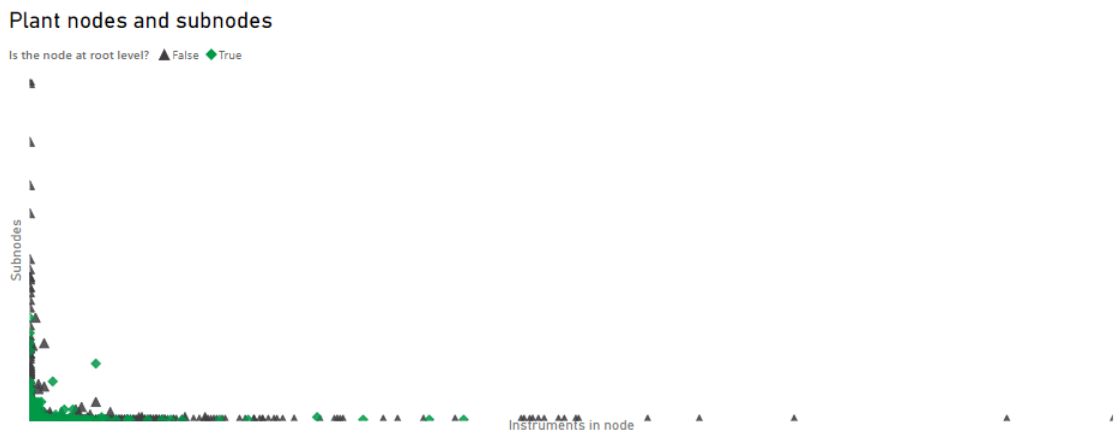


Kuva 25. Referenssit, valmistajat ja mallit

4.5.2 Laitosrakenne

Laitosrakenteen käyttöasteesta oli myös toivottu jonkinlaista kaaviota. Laitosrakenne tarkoittaa sovelluksessa olevaa lajittelutapaa, joka yksinkertaisuudessaan toimii samalla lailla kuin kansiorakenne. Jokaiseen kansioon eli tehtaaseen voi luoda instrumentteja ja alitehtaita, ja näiden maksimisyvyyttä ei ole määritelty. Yleisesti toiminto on käytössä määrittämään, missä tietyssä laitok- sessa jokin tietty instrumentti fyysisesti sijaitsee.

Jokaisella tehdastasokansiolla on oma tunnisteensa. Jos kansio oli alikansiotyyppinen, on sillä myös ylemmän tason tunniste tiedossa. Juurikansiossa sijaitsevista kansioista tämä tunniste on jätetty tyhjäksi. Kuvaaja (Kuva 26) esittää kansiorakenteesta alikansioiden määrän Y-akselilla ja instrumenttien määrän kansiota kohden X-akselilla.



Kuva 26. Kansiorakenne ja niiden käyttöaste

Tuoteomistaja kuitenkin oli kiinnostunut tietämään, kuinka monta instrumenttia on tehdastaso- kohden. Näin pystytään varmistamaan, että ohjelmiston testausvaiheessa käytetään mahdolli- simman samanlaista dataa kuin oikeassa käyttöympäristössä. Esimerkiksi kuvaajasta nähdään, kuinka monta instrumenttia on maksimissaan jollain asiakkaalla, tai montako alikansiota on yh- dessä kansiossa. Tietojen avulla voidaan saada tieto kansiorakenteen käytön ääripäistä.

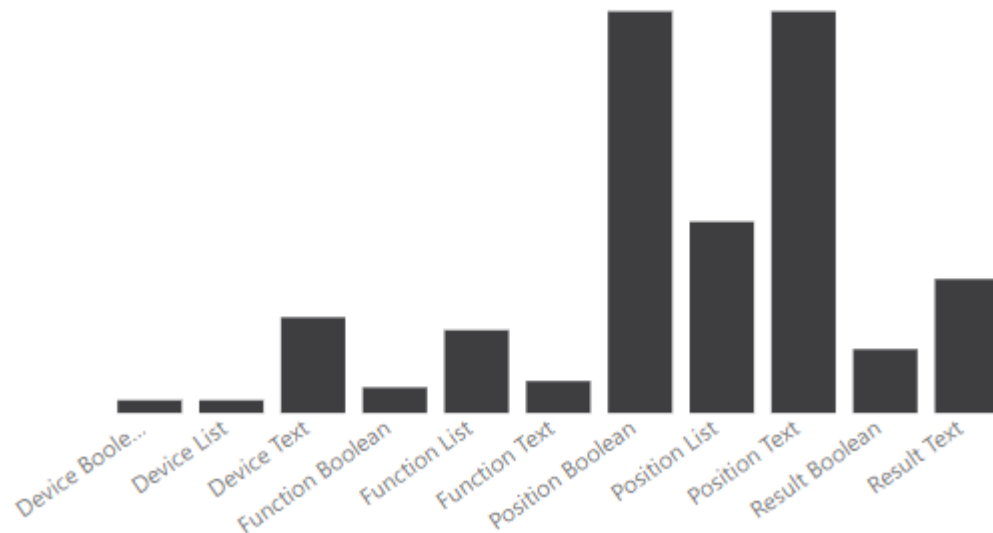
4.5.3 Muokattavat kentät

Viimeinen kaavio (Kuva 27) sivulta liittyy muokattaviin kenttiin, joilla käyttäjä voi antaa tarkempaa tietoa luotavasta instrumentista. Yleisesti toimintoa käytetään lisäämään esimerkiksi valintaruutu

merkitsemään, onko kyseinen instrumentti kriittinen vai ei. Kaaviosta kuitenkin näkee selkeän jakauman, jossa suurin osa lisätyistä mukautettavista kentistä keskittyy lähinnä position eli paikan ympärille.

Free fields

Custom field location



Kuva 27. Muokattavien kenttien jakautuminen

Kentät olivat tallennettuna joko tekstikentäksi tai totuusarvoksi. Jokainen listakenttä oli myös määritelty tekstikentäksi, joten ne piti jotenkin saada erotettua toisistaan, jotta saatiin tarkempi kaavio. Koska arvoja ei suoraan voinut helposti jakaa joko listaan tai tekstiksi, tapahtui tyyppien jako luomalla uusi sarake, joka ottaa huomioon sekä tyyppin että listan olemassaolon (Ohjelma-koodi 8).

```
ExpandedTypeColumn =
IF(
    'FreeFields'[FieldType] = "Text" && LEN('FreeFields'[ListValues]) > 0,
    "List",
    'FreeFields'[FieldType]
)
```

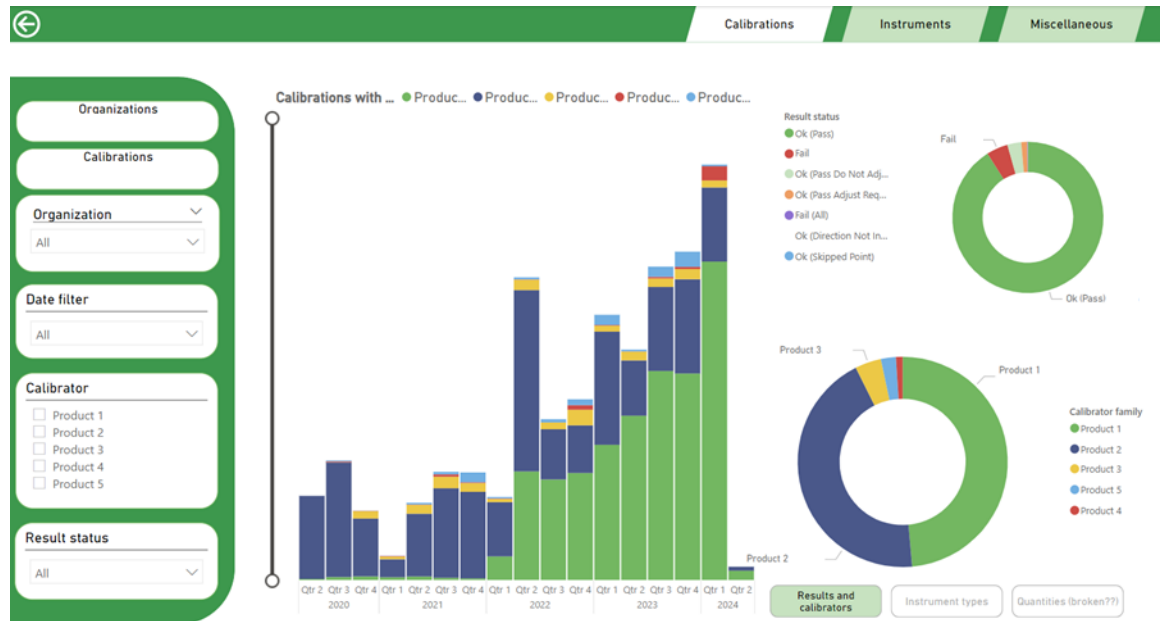
Ohjelmakoodi 8. Listakenttien erottaminen yleisistä tekstikentistä

Uudessa sarakkeessa tarkistetaan, onko kentän tyyppinä teksti ja onko listasarakkeeseen tallennettu mitään. Jos tyyppinä on teksti, mutta listasarake on tyhjä, tulee uuteen sarakkeeseen riville sama arvo kuin alkuperäisessä tyyppissä. Jos taas listasarakkeessa on jotain tallennettuna, riville tallennetaan tekstijono "List".

4.6 Lopullisen ulkoasun yhdenmukaistaminen

Kehitysvaiheessa kojelauta oli erilaisen näköinen, mutta tuli pyyntö sen yhdenmukaistamisesta yrityksen omaan ilmeeseen. Väripaletissa oli jo haettu haluttua teemaa, mutta asiakkaan mielestä sen tulisi näyttää samankaltaiselta kuin sivusto, jonka datasta työ oli tehty.

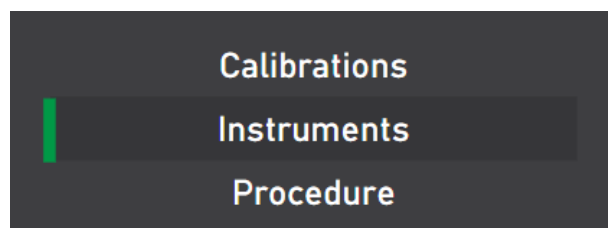
Raporttina alkuperäinen versiokin (Kuva 28) oli hyvä, mutta väripaletina oli käytetty Power BI:n valmiiksi tarjoamaa vaihtoehtoa. Se ei tuonut mielikuvaa, että kojelauta olisi räätälöity nimenomaan työn tilanneelle yritykselle. Vanhan version ulkoasusta myös puuttui yhdenmukainen suunnitelmallisuus. Työn tilannut yritys oli kuitenkin juuri päivittänyt suunnittelumallikirjastonsa, joten korjauksia varten ohjeita oli helposti saatavilla.



Kuva 28. Kojelaudan vanha versio

Kojelautaa varten luotiin kokonaan uusi väripaletti, jota jokainen komponentti myös oletuksena käyttäisi. Koko sivupaneeli laitettiin uusiksi ja pyöristetyt reunat muutettiin oikeaan tyyliin. Ylä-paneeli ja navigaatio siirrettiin myös sivupaneeliin, joten itse kaavioille jäi myös enemmän tilaa käyttöön, eikä rakenne ollut liian vilkas.

Aluksi uudemman version navigaatiossa ei ollut vihreää korostusta (Kuva 29), mutta sen lisääminen tuli yrityksen pyynnöstä. Se oli oikeastaan yksi merkittävimmistä yksityiskohdista, jotka yhdistivät kojelaudan yrityksen sivustoon entistä paremmin. Aktiivisen sivun korostus navigaatiossa ei kuitenkaan ollut yksinkertainen toimenpide.



Kuva 29. Navigaation tyyllitys

Power BI:n navigaatiot tukevat elementtien muokkaamista huomattavan joustavasti. On mahdollista muokata erikseen normaaleja, valittuja sekä painettuja valikon elementtejä. Lisäksi pysty-

tään muokkaamaan jopa navigointipainikkeita, joiden yllä hiiren osoitin on. Oli siis helppoa asettaa valitulle painikkeelle tummempi tausta. Reunaväriä napille pystyttiin lisäämään vain kokonaan sen ympärille tai ei ollenkaan.

Ratkaisuna oli luoda vihreästä palkista itse kuvatiedosto, joka pystyttiin asettelemaan sivulle. Jokaisella sivulla on oma kuvaelementtinsä, joka on siirretty oikeaan kohtaan navigaatiossa ja skaalattu oikean kokoseksi.

4.7 Käytetyn datan luotettavuus

Miten kehittäjä voi olla varma, että kojelaudan kaaviot näyttävät käsiteltävän datan oikein? Opin näytettä varten tehty työ rakennettiin ison tietokannan kanssa, joten datan luotettavuuden varmistaminen oli haastavaa.

Eräänä ideana kuitenkin oli luoda paikallisesti toimiva SQL Server -tietokanta, jossa olisi samat tiedot tallennettuna kuin sovelluksen kehityksessä käytettävässä tietokannassa. Näin käsiteltävä data olisi rajattu pieneen ennalta määritettyyn määrään, ja sen totuudenmukaisuutta olisi myös päässyt tutkimaan itse sovelluksen käyttöliittymän kautta paremmin visuaalisessa muodossa.

Ratkaisu ei kuitenkaan toiminut, kuten oli suunniteltu. Tietokannan yhdistämisessä Power BI:hin ilmeni monia ongelmia, jotka johtivat uusiin virheilmoituksiin. Seuraavana ideana oli paikallisen tietokannan vienti Excel-tiedostoon, mutta samat ongelmat ilmenivät myös sen kanssa. Tietokannassa oli todennäköisesti jotain ongelmaa, johon ei mistään löydetty apua.

Seuraavaksi vaihtoehdoksi muodostui suoraan käytetystä tietokannasta datan siivilöinti. Työssä käytetyssä tietokannassa oli myös yrityksen omien kehittäjien luomaa dataa, joten käytännössä oli mahdollista löytää ja suodattaa sieltä tietyn organisaation tallentama data käyttämällä organisaation anonyymiä tunnistetta. Organisaation omistajalla oli mahdollisuus saada oma tunnistensa tietoonsa, joten testiorganisaation kautta onnistui kannan rajaaminen tarpeeksi pieneksi, jotta datan luotettavuus voitiin todeta.

Rajatulla määrällä dataa oli helppo tarkistaa kaavioiden toimiminen. Tarkistuksessa löydettiin yksi kahden kaavion ryhmä, joka oli aiheuttanut epäilystä niiden toiminnasta jo pitkään ja jonka virheet tulivatkin ilmi varmistuksen aikana.

Testauksen aikana oli tiedossa, että kyseisellä organisaatiolla oli tallennettuna viisi kappaletta instrumentteja, joilla jokaisella oli oma toiminnon tulo ja lähtö. Näin oli mahdollista olettaa, että tuloja ja lähtöjä olisi ollut yhteensä viisi kappaletta kumpaakin, mutta odotusten vastaisesti kaaviot näyttivät kaksinkertaista arvoa (Kuva 30).



Kuva 30. Tulojen ja lähteiden kaaviot näyttivät virheellisesti kaksinkertaista arvoa

Saatiin selville, että jokaisen instrumentin luonnissa lisätään tietokantaan tulojen ja lähtöjen taulukkoon yhteensä kaksi eri riviä, jolla kummallakin on oma tunniste. Ensimmäisessä rivissä on PositionId ja toisessa DeviceId. Kummallakin rivillä on sarake "LinkedFunctionId", joka sisältää toisen instrumentissa olevan rivin tunnisteen, jolloin nämä rivit osoittavat ristiin. Tämän perusteella olisi voitu päätellä, että kuvaajaan saisi oikeat lukemat jakamalla määrän kahdella.

Yrityksen tuotteessa on kuitenkin mahdollisuus tehdä niin kutsuttuja vapaita laitteita, joilla ei ole tätä PositionIdtä ollenkaan. Silloin luodaan ainoastaan rivi, jossa DeviceId on olemassa, joten kuvaajaa varten ei tarvinnut välittää ollenkaan riveistä, joissa tunniste oli tyhjä. Käytetty taulukko suodatettiin aikaisemmin mainitussa Ohjelmakoodi 1:ssä.

5 Pohdinta

Koko opinnäytetyöprosessin aikana työn tilaajat olivat tyytyväisiä sekä edistymiseen että tuloksiin. Tuli myös tietoon, että kojelaudan antamaa hyötyä oli jo sovellettu jossain päätöksenteossa, mutta tiedossa ei ole mitä ja miten. Ulkoasuun oltiin myös tyytyväisiä ja samaa raporttipohjaa kenties voitaisiin käyttää muissakin yrityksen Power BI -raporteissa. Tämä osoittaa hyvin, miten työn visuaalisessa puolella onnistuttiin.

Kojelautaa mahdollisesti tullaan asettamaan tilanteen yrityksen toimistoon infonäytölle PowerPoint-esityksenä. Power BI:n kojelaudit kääntyvät helposti diaesitykseksi, jotka päivittyvät alkuperäisen tiedoston päivittyessä. Data saadaan pidettyä ajan tasalla, ja esitykseen tarvitsisi vain lisätä automaattinen dian vaihtuminen. Sitä varten olisi suotavaa muuttaa näkymien sommittelua paremmin diaesitykseen sopivaksi, esimerkiksi poistamalla sivupaneeli.

Kaavioiden muokkaamisen ja viimeisten korjausten aikana myös huomattiin, että erään kalibrointilaitteen käytössä ilmeni välillä merkittävä nousu, minkä jälkeen se laski takaisin matalampiin lukuihin. Laitteen tilastoja tutkittiin muutaman vuoden ajalta, jolloin huomattiin kyseisen piikin käyttömäärään tapahtuvan noin neljäntoista kuukauden välein, ja jokaisella kerralla tuplaantuvan. Olisi hyödyllistä tietää ja lopulta selittää, miksi näin tapahtuu, mutta se vaatisi lisää jatkokehitystä kojelautaan.

Projektissa käytetystä Power BI:stä ei ollut aikaisempaa kokemusta. Käyttöönotto oli kuitenkin helppoa ja työpöytäversio sovelluksesta hyvinkin monipuolinen. Power BI onnistui yllättämään joustavuudellaan, mutta myös yksinkertaisuudellaan.

Sovelluksessa oli kuitenkin omat puutteensa, ja pystyi epäonnistumaan eräissä todella alkeellisissa toiminnoissa. Yksi pyyntö oli laittaa katkoviivaa jakamaan kaksi osiota keskenään, mutta kehityksen aikana jouduttiin tarjoamaan ainoastaan kokonaista viivaa, sillä Power BI:ssä ei ollut mahdollista muokata viivaobjektin tyyliä katkonaiseksi. Kiertotietä olisi voinut kuitenkin käyttää, ja luoda seuraavanlainen viiva DAX-funktiolla:

```
DashLine = REPT("|" & UNICHAR(10), 20)
```

Funktio luo tekstiobjektin, joka toistaa haluttua merkkiä kaksikymmentä kertaa. Luodun tekstiobjektin olisi seuraavaksi voinut laittaa tyhjään kaavioon, jossa se piirtäisi pystysuuntaisen katkoviivan.

Suurin haaste projektissa oli erilaisten sovellusten ja työkalujen käyttö tilanteissa, joihin ei haku-koneesta löytynyt apua. Ainoastaan yrityksen omat tekijät tunsivat ja hallitsivat nämä toiminnot. Apua tarvittiin monta kertaa esimerkiksi tietokannan ja siihen liittyvien asioiden kanssa. Vastauksia kuitenkin kesti pitkään saada, sillä asiat parhaiten tuntevat henkilöt olivat usein liian kiireisiä vastataksaan kysymyksiin kokonaisvaltaisesti. Joskus vastaukset tulivat nopeastikin, mutta lyhyessä ja vaikeaselkoisessa muodossa. Kaikki ongelmat kuitenkin ratkaistiin tavalla tai toisella.

Jatkuvaa palautetta saatiin kojelaudan tekovaiheessa, mikä helpotti työn jatkamista huomattavasti. Opinnäytetyön virallinen asiakas, eli tässä tilanteessa työpaikalla oleva opinnäytetyön ohjaaja, oli erittäin aktiivisesti esittämässä työtä muulle yrityksen henkilöstölle, joten kerätyistä kehitysehdotuksista voidaan kiittää häntä.

Projektin ollessa noin puolivälissä kojelautaa esiteltiin ensimmäisen kerran kehitystiimille. Esitykseen osallistuneet saivat kysyä kysymyksiä, antaa parannusehdotuksia sekä kertoa mielipiteitänsä kojelaudasta. Esitystilaisuus sisälsi monta rakentavaa keskustelua, ja kojelautaan saatiin uusia ideoita, jotka paransivat työn laatua.

Palautetta saatiin esimerkiksi kaavioista ja datasta, jota tuoteomistajan ja tiimin mielestä olisi kiinnostavaa nähdä. Tämä oli tärkeää, sillä yrityksen kehitysosasto pystyi tietämään tarkemmin tilastot, jotka he halusivat nähdä. He myös tiesivät, millaista dataa tuote sisälsi ja miten tämä data toimii suhteessa muuhun dataan.

Työn edistymistä seurattiin viikoittain Scrum-tyylisesti: Kerran viikossa käytiin läpi, mitä oli tehty, mitä seuraavaksi tehdään ja onko tekemiselle jotain estettä. Tapaamiset olivat suuri apu tekemisen ylläpitämisessä, sillä se vähensi työn keskeytymisen mahdollisuuksia, jolloin jatkamisen kynnyks nousee.

Yhtenä mahdollisena opinnäytetyön laajennusmahdollisuutena olisi ollut samanlaisen kojelaudan luonti yritykselle asiakaskäyttöön. Käytännössä tämä olisi tarkoittanut tuotteen sivulle upotettuja kaavioita, jotka olisivat näyttäneet samanlaista dataa, mutta ainoastaan asiakkaan omasta organisaatiosta. Näin käyttäjä voisi seurata oman yrityksensä tilastoja ja saada mielenkiintoista dataa myös itselleen.

Työn aikana tätä mahdollisuutta tutkittiin hieman, mutta päätettiin tämän olevan liian laaja alkaa toteuttamaan opinnäytetyön rinnalla. Esimerkiksi pelkän Power BI -raportin olisi voinut upottaa

tuotteen sivustolle, mutta heti ensimmäiseksi ongelmaksi nousi datan hallinnointi ja sen turvallisuus. Entä pitääkö käyttäjällä olla myös tunnukset luotuna Power BI -ympäristöön käyttöä varten? Epäselväksi jäi myös, miten näytettävä data saataisiin suodatettua vain yhdelle organisaatiolle.

Koko kojelaudan lisääminen yrityksen tuotteen sivustolle oli kuitenkin helppoa ja hyvinkin mahdollista. Sellaisenaan nykyistä raporttia ei kuitenkaan kannattaisi laittaa, sillä sitä ei kehitetty sivustolle upottamista varten. Jatkokehitys kokonaisuutena vaatisi kuitenkin lisää asiaan perehtymistä.

Kojelaudalle voisi olla myös mahdollista lisätä tapoja selittää, miksi tiettyjä asioita tapahtuu kaavioissa. Se kuitenkin vaatisi enemmän kokemusta kojelautoista, mutta myös käytetystä datasta. Sitä varten olisi hyvä tietää tietokannasta arvot, jotka vaikuttavat toisiinsa sekä löytää yhteyksiä näiden välille. Tarinankerronnallinen datan esittäminen olisi lisätutkimusmahdollisuus, mutta sen toteuttaminen käytännössä on haastavaa.

Kokonaisuutena työssä onnistuttiin hyvin, vaikka jatkokehitysideoita vielä löytyisikin. Myös kojelautaa olisi aina mahdollista parannella. Saatiin myös ehdotuksia, että eräs toinenkin yrityksen kojelauta voitaisiin parannella visuaalisesti näyttävämpään muotoon. Kyseisestä kojelaudasta otettiin mallia, miten ei tehdä asioita, joten ehdotusta voitaisiin kuitenkin harkita.

Lähteet

- [1] Knight D, Pearson M, Schacht B. 3. päivitetty painos. Microsoft Power BI Quick Start Guide The ultimate beginner's guide to data modeling, visualization, digital storytelling, and more. Packt Publishing; 2022.
- [2] Calzon B. What Is A Data Dashboard? Definition, Meaning & Examples. [Internet]. [viitattu 25.4.2024]. Saatavilla: <https://www.datapine.com/blog/data-dashboards-definition-examples-templates/>
- [3] Warren J. Integrating KPIs into your company's strategy. [Internet]. [viitattu 3.5.2024]. Saatavilla: https://www.kwantyx.com/wp-content/uploads/AT_WP_KPI_EN.pdf
- [4] Eckerson WW. Deploying Dashboards and Scorecards. 2006. Saatavilla: <https://sophitech.mx/files/1113/7718/8544/TDWI-Best-Practices-Report-Deploying-Dashboards-and-Scorecards.pdf>
- [5] Rožman M, Gonçalves CT, José M, Gonçalves A, Campante MI. Developing Integrated Performance Dashboards Visualisations Using Power BI as a Platform. [Internet]. 2023. Saatavilla: <https://doi.org/10.3390/info14110614>
- [6] Pappas L, Whitman L. Riding the Technology Wave: Effective Dashboard Data Visualization. [Internet]. 2011. Saatavilla: https://doi.org/10.1007/978-3-642-21793-7_29
- [7] Losbichler H, Perkhofer L, Hofer P, Winzer J. Reporting 2.0-Interactive Visualizations and Dashboards for Big Data. [Internet]. 2020. Saatavilla: <https://accid.org/wp-content/uploads/2021/12/REPORT1.pdf>
- [8] Power BI-raportointi – tasapaino tekniikan ja muotoilun välillä. DB Pro Services. [Internet]. [viitattu 29.4.2024]. Saatavilla: <https://dbproservices.fi/power-bi/power-bi-raportointi/>
- [9] Pernice K, Whinton K, Nielsen J. How People Read on the Web: The Eyetracking Evidence. Nielsen Norman Group; 2014.
- [10] Wexler S, Shaffer J, Cotgreave A. The Big Book of Dashboards: Visualizing Your Data Using Real-World Business Scenarios. Wiley Publishing; 2017.

- [11] Power BI: An analytical view. Journal of Accountancy. [Internet]. [viitattu 27.4.2024]. Saatavilla: <https://www.journalofaccountancy.com/issues/2020/mar/microsoft-power-bi-data-excel.html>
- [12] Power BI - perusteet ja paljon enemmän. Meltlake. [Internet]. [viitattu 30.4.2024]. Saatavilla: <https://www.meltlake.com/blogi/power-bi-perusteet-ja-paljon-enemman>
- [13] Mobile. Microsoft Power BI. [Internet]. [viitattu 30.4.2024]. Saatavilla: <https://powerbi.microsoft.com/en-us/mobile/>
- [14] Introduction to dataflows and self-service data prep. Microsoft Learn. [Internet]. [viitattu 30.4.2024] Saatavilla: <https://learn.microsoft.com/en-us/power-bi/transform-model/dataflows/dataflows-introduction-self-service>
- [15] Understand and optimize dataflows refresh. Microsoft Learn. [Internet]. [viitattu 30.4.2024]. Saatavilla: <https://learn.microsoft.com/en-us/power-bi/transform-model/dataflows/dataflows-understand-optimize-refresh>
- [16] Power BI Incremental Refresh: The Ultimate Guide 2023. Alpha Serve. [Internet]. [viitattu 30.4.2024]. Saatavilla: <https://www.alphaservesp.com/blog/power-bi-incremental-refresh-the-ultimate-guide>
- [17] Power BI dataflows: a Deep Dive into Incremental Refresh & the Enhanced Compute Engine. Datalineo. [Internet]. [viitattu 30.4.2024] Saatavilla: <https://www.datalineo.com/post/powerbi-dataflows-deepdive-incremental-refresh-ece>
- [18] Introduction to Blob (object) Storage. Microsoft Learn. [Internet]. [viitattu 24.4.2024]. Saatavilla: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction>
- [19] Gorman BL. Developing Solutions for Microsoft Azure Certification Companion. Apress Berkeley, CA [Internet]. 2023. Saatavilla: <https://doi.org/10.1007/978-1-4842-9300-3>
- [20] Overview. Protocol Buffers Documentation. [Internet]. [viitattu 24.4.2024]. Saatavilla: <https://protobuf.dev/overview/>
- [21] What is Protobuf? Postman Blog. [Internet]. [viitattu 24.4.2024]. Saatavilla: <https://blog.postman.com/what-is-protobuf/>

- [22] Language Guide (proto 3). Protocol Buffers Documentation. [Internet]. [viitattu 24.4.2024]. Saatavilla: <https://protobuf.dev/programming-guides/proto3/>
- [23] Metrological Traceability in Calibration – Are you traceable? Beamex. [Internet]. [viitattu 24.4.2024]. Saatavilla: <https://blog.beamex.com/metrological-traceability-in-calibration-are-you-traceable>
- [24] Mitä on kalibrointi? Kalibroinnin määritelmä. Beamex. [Internet]. [viitattu 30.4.2024]. Saatavilla: <https://www.beamex.com/fi/tietopankki/mita-on-kalibrointi/>
- [25] Siekmann L. Requirements for Reference (Calibration) Laboratories in Laboratory Medicine. Clin Biochem Rev [Internet]. 2007. [viitattu 29.4.2024]. Saatavilla: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2282407/>