Tietovarastoinnin hyvät käytänteet ja niiden toteutuminen tietovarastoprojektissa.

Saku Junni

2020 Laurea

Laurea-ammattikorkeakoulu

Tietovarastoinnin hyvät käytänteet ja niiden toteutuminen tietovarastoprojektissa.

Saku Junni

Tietojenkäsittely

Opinnäytetyö

Joulukuu,2020

Laurea-ammattikorkeakoulu **Tiivistelmä**
Tietojenkäsittely
Tradenomi (AMK)

Saku Junni

Tietovarastoinnin hyvät käytänteet ja niiden toteutuminen tietovarastoprojektissa.

Vuosi 2020 Sivumäärä 413

Tämän opinnäytetyön tarkoituksena oli tutkia mitkä ovat yleisiä hyviä käytänteitä, joita on syytä ottaa huomioon tietovaraston suunnittelu- ja kehitystyössä ja kuinka ne ovat toteutuneet Vartas Oy:n asiakasprojektissa. Työstä saatuja johtopäätöksiä ja tuloksia voidaan hyödyntää projektin jatkokehityksessä ja vastaavanlaisissa tulevaisuuden tietovarastoprojekteissa.

Teoriaosuuden tarkoituksena oli kuvailla yleisellä tasolla tietovarastoinnin perusteita ja tutkia mitä yleisiä hyviä käytänteitä tietovarastojen suunnitteluun ja toteutukseen liittyy. Työ kartoittaa tietovaraston rakenteeseen, kustannuksiin, tehokkuuteen ja loppukäyttäjiin liittyviä käytänteitä, joita yleensä pidetään hyvän ja onnistuneen tietovarastoprojektin perustana.

Työn jälkimmäisessä osuudessa peilattiin näitä käytänteitä Vartas Oy:n asiakkaalle tuotetun tietovaraston suunnittelu- ja kehitysprosessiin. Työssä käytiin läpi projektin eri vaiheet ja tarkasteltiin suunnittelussa tehtyjä valintoja kuten tietovaraston tyyppiä ja valittuja toteutusvälineitä. Tällä tavoin saatiin laaja-alaisia tuloksia siitä mitkä hyvien käytänteiden osa-alueet projektissa toteutettiin onnistuneesti, missä olisi vielä huomioitavaa ja minkälaista kehitystä jo toteutettuihin tietovaraston osiin olisi mahdollista tehdä.

Asiasanat: Tietovarasto, Hyvät käytänteet, Data Warehouse, ETL-prosessit, Data

Laurea University of Applied Sciences **Abstract**
Degree Program in Business Information Technology
Bachelor's Degree
Saku Junni
Data Warehousing Best Practices and their Implementation in a Data Warehouse Project
Year 2020 Pages 413

The goal of this bachelor's thesis was to examine what are the typical best practices associated with data warehouse design and development and how they were implemented in a data warehouse project by Vartas Ltd. The results and findings can be used in further development of the data warehouse and future projects with similar goals.

The intention of the theoretical section was to describe the basics of data warehousing and study what are the typical best practices that are related to the design and development of a data warehouse. The section explored practices linked to architecture, costs, performance and end-users that are widely held necessary in a successful data warehouse project.

The latter section of the study aimed to reflect these practices to the design and development of a customer's data warehouse project. The study described the steps that were taken in the data warehouse design process and what choices were made concerning the type of the data warehouse and the tools used. This way a wider understanding was achieved on which sections were successful, what areas need further consideration and which already completed parts could be improved with the insights received.

Keywords: Data Warehousing, Best Practices, ETL-process, Data

Sisällys

1 Johdanto

Tämän opinnäytetyön aiheena on Vartas Oy:n asiakasprojektina toteutettu tietovarasto ja siihen linkittyvä tietovarastoinnin teoria. Tutkimuksessa käsitellään tietovaraston perusperiaatteita ja sen suunnitteluun ja toteutukseen liittyviä hyviä käytänteitä ja kuinka hyvin nämä toteutuivat projektissa.

Tutkimuksessa fokuksena oleva tietovarasto kehitettiin Vartas Oy:n toimesta asiakkaalle vuosien 2019 ja 2020 aikana. Koska projekti oli jo opinnäytetyötä tehdessä loppusuoralla, on tutkimuksen tulokset lähinnä retrospektiivinen katsaus jo tehtyyn työhön. Koska opinnäytetyön laatija oli itse mukana työn suunnittelussa ja toteutuksessa, tämä tutkielma on myös itsereflektio omasta työstä ja sen onnistumisesta.

Tietovarastoprojektin taustalla on yleensä tarve käsitellä yrityksen hallussa olevaa datamassaa, esimerkiksi hyödyntää sitä liiketoiminnan apuvälineenä tai seurata oman toiminnan kehittymistä. Jotta tähän tavoitteeseen päästään on jo tietovaraston suunnitteluvaiheessa otettava huomioon seikkoja, jotka palvelevat lopullista käyttötarkoitusta parhaalla mahdollisella tavalla. Monella projektin suunnittelussa tehdyllä valinnalla on kauaskantoisia vaikutuksia, joiden tunnistaminen jo suunnitteluvaiheessa voi säästää paljon resursseja ja kustannuksia tietovaraston elinkaaren aikana.

2 Tutkimusmenetelmät ja tavoitteet

Tutkielman tavoitteena on vastata kahteen tutkimuskysymykseen. Ensinnäkin minkälaisia ovat yleiset hyvät käytänteet liittyen tietovarastoiden suunnitteluun ja toteutukseen. Toisekseen kuinka hyvin näitä käytänteitä työn fokuksena olevassa tietovarastoprojektissa noudatettiin. Tutkimustarve siis muodostuu siitä, että halutaan tutkia projektissa tehtyjä ratkaisuja ja kuinka hyvin ne vastaavat alalla yleisinä pidettyjä käytänteitä ja onko suunnittelutyössä jätetty ottamatta huomioon jotain kriittisiä aspekteja.

Työn ensimmäisessä osiossa käydään läpi tietovarastoinnin perusperiaatteet ja mikä on tyypillisten tietovaraston rakenne. Tämän jälkeen käydään lähdekirjallisuuden kautta läpi sitä mitä yleensä pidetään tietovarastoinnin hyvinä käytänteinä. Työn jälkimmäisessä osiossa kuvaillaan kohteena olevan tietovarastoprojektin lähtökohdat, suunnittelu, toteutus ja käytetyt teknologiat. Tämän jälkeen peilataan teoriaosuudessa ilmenneitä hyviä käytäntöjä projektissa tehtyihin suunnitteluvalintoihin ja lopputuloksiin.

Työn tutkimusmenetelmä on kaksiosainen, alkupää on aineiston analyysia, jossa lähdekirjallisuudesta pyritään etsimään teoreettista näkökantaa sille, mikä on yleinen tapa mitata tai arvioida tämänkaltaisia projekteja ja toteutustapoja. Itse projektin osuus on taas Case-tutkimus, jossa tarkastellaan tehtyä työtä ja sen toteutusta. Työ rakentuu siis aineistoanalyysissä löydettyjen vastausten peilaamisesta Case-tutkimuksen kohteena olevan projektin toteutukseen. Case-tutkimus metodina soveltuu erityisen hyvin kehitysehdotusten tuottamiseen ja sen avulla saadaan mahdollisuus ymmärtää tutkimuksen kohdetta kokonaisvaltaisesti ja realistisessa toimintaympäristössä. (Ojasalo, Moilanen & Ritalahti 2005, 52). Projektin nykyisen tilan vertaaminen yleisesti hyvänä pidettyihin käytänteisiin mahdollistaakin erilaisten kehitysideoiden tuottamisen joita projektin loppuvaiheessa voidaan hyödyntää.

Tutkimuksen tavoitteena on löytää vastaus siihen mitkä projektin osa-alueista noudattivat tietovarastoinnin hyviä käytänteitä, mitkä asiat jäivät huomioimatta ja miten tutkimuksen havaintoja voitaisiin hyödyntää tietovarastoprojektin jatkokehityksessä.

Tutkimuksen eettisyys varmistetaan sillä, että asiakasorganisaation nimeä eikä sen henkilöstä ole mainittu tutkimuksessa ja kaikista projektin osa-alueista on kirjoitettu vain korkealla tasolla tunnistamisen välttämiseksi. Myöskään asiakkaan dataa ei esitellä missään muodossa muuta kuin kuvailemalla sen yleistä luonnetta.

3 Tietovarastoinnin perusteet

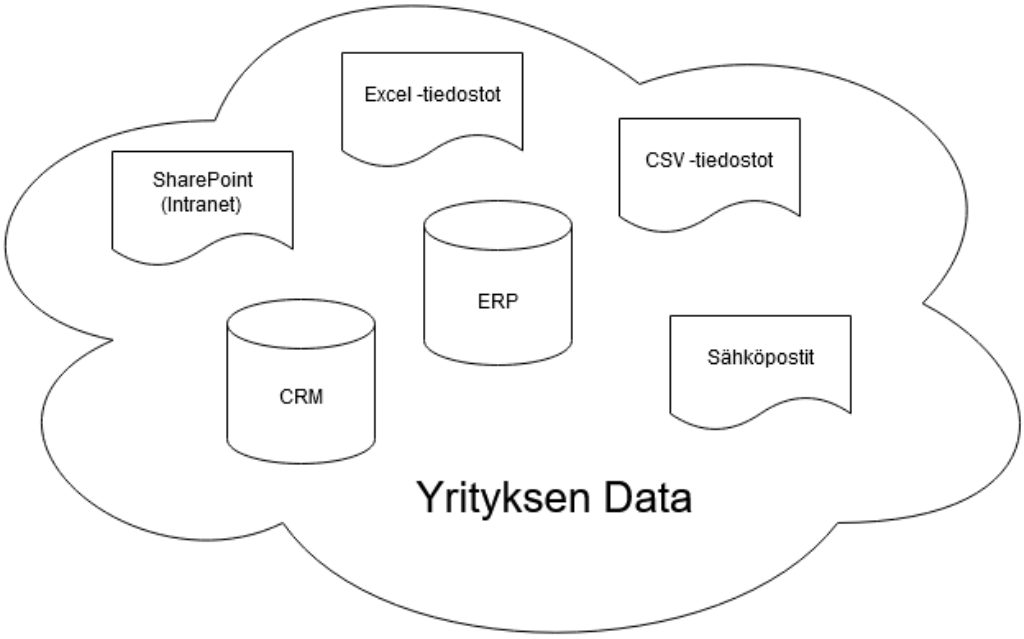
Ennen sukeltamista käytäntöihin on tässä osiossa tarkoituksena kartoittaa tietovarastoinnin peruskäsitteitä ja syitä miksi tietovarastoja yleensä päädytään rakentamaan. Ensin on tarkasteltava mistä tietovaraston peruskivi eli data muodostuu ja mikä on sen luonnollinen ilmenemismuoto. Sen jälkeen käydään läpi mitkä ovat tietovaraston tyypillisimmät rakennuspalikat ja kuinka niitä hyödynnetään.

3.1 Data ja tieto

Nykypäivänä yritysten on milteipä mahdotonta harjoittaa liiketoimintaansa ilman jonkinlaisia sähköisiä järjestelmiä, ja kaikkien näiden järjestelmien taustalla on aina jonkinlaista dataa. Listaus asiakkaista ja yhteistyökumppaneista, varastokirjanpito tai tuotetieto voivat olla esimerkkejä datasta, joita yrityksen järjestelmät keräävät ja joita hyödynnetään jokapäiväisessä liiketoiminnassa. Mitä enemmin yrityksen toiminnot ovat siirtyneet sähköiseen muotoon, sitä enemmin tätä liiketoiminnan dataa syntyy. Tällainen ns. Master Data on liiketoiminnalle kriittistä ja ilman sitä koko organisaation olisi mahdotonta toimia, joten sen saatavuus ja nopea hyödyntämin on elintärkeää (Väre 2019, 23). Tässä vaiheessa tuleekin yleensä eteen kysymys, miten tätä dataa voitaisiin parhaalla tavalla hyödyntää esimerkiksi liiketoiminnan suunnittelun tukena. Juuri datan hyötykäyttö voi olla se etulyöntiasema, jolla yritys pääsee

kilpailijoidensa edelle. Salo (2014, 8) kuvaakin datan tuomia etuja näin: ”Datasta on todellisuudessa tulossa tulevaisuuden öljy ja ne, joilla on sen hyödyntämiseen parhaat edellytykset tulevat menestymään parhaiten.”

Data itsessään ei ole itseisarvoisesti arvokasta, vaan se miten sitä käyttää. Monellakin yrityksellä todennäköisesti makaa palvelimillaan teratavuja dataa, jota ei hyödynnetä mitenkään joko siksi, ettei sen potentiaalia ole tiedostettu tai koska data on muodossa, jossa sitä eniten tarvitsevat eivät pysty sitä hyödyntämään (Kuvio 1). Ideaalitilanne olisi, että tämä data voitaisiin tarjoilla siinä muodossa, että sitä tarvitsevat pystyisivät pääsemään siihen käsiksi helposti ja käyttämättömän datan oikea potentiaali voitaisiin käsittää. Salo (2013, 26) kuvastaa datan ja tiedon suhdetta näin: ”Data on raaka-aine, josta voidaan louhia informaatiota ja tästä muodostaa tietoa. Tieto lisää ymmärrystä ja kumuloitunut tieto muodostaa tietämystä.” Tällainen tietämys voi olla kultaakin kalliimpaa esimerkiksi havaittaessa oman liiketoiminnan ongelmakohtia ja pullonkauloja. Tietämystä voidaan myös käyttää tulevaisuuden ennustamiseen tiettyjen trendien kautta, ja tällä tavalla ohjailla omaa liiketoimintaa haluttuun suuntaan (Salo 2013, 33).



Kuvio 1: Esimerkki yrityksen datan monimuotoisuudesta

Yrityksen olisi siis löydettävä datankäsittelyyn metodi, jolla päästäisiin pisteeseen, jossa yrityksen data on helposti saavutettavissa ja muunnettavissa oikeaksi tiedoksi ja tietämykseksi. Yksi vastaus ongelmaan on tietovarastointi, jossa yrityksen hajallaan olevaa tietoa pyritään

viemään yhteen keskitettyyn paikkaan ja muokkaamaan sellaiseen muotoon, josta sen hyötykäyttö on mahdollisimman vaivatonta.

3.2 Tietovarasto

Hyvin yleisesti yritysten liiketoiminnan data sijaitsee sen ydinliiketoimintaan liittyvissä niin sanotuissa operatiivisissa järjestelmissä (Hovi, Ylinen & Koistinen 2001, 15). Operatiivisten järjestelmien taustalla olevat tietolähteet palvelevat ensisijaisesti järjestelmää itseään eivätkä ne aina sovellu muunlaiseen käyttötarkoitukseen. Yleisiä operatiivisten järjestelmien heikkouksia datan hyödynnettävyyden kannalta ovat:

- Data on hajautunut moneen eri tietolähteeseen ja sen kokoaminen yhteen tarpeeseen, esimerkiksi raportointiin on työlästä.
- Operatiiviset järjestelmät eivät tallenna historiatietoa, jota voitaisiin käyttää raportoinnissa trendien havainnointiin.
- Alkuperäistä lähdejärjestelmää ei haluta kuormittaa raskailla raportointihauilla.
- Tietolähteen rakenne on liian monimutkainen ja sopimaton tavallisen käyttäjän hyötykäyttöä varten.
- Samaa dataa on päällekkäisinä eri operatiivisissa järjestelmissä, joten ei ole niin kutsuttua ”yhtä totuutta” josta tietoa voidaan hakea.
- Data on sirpaloitunut erinäisiin tiedostoihin ja sähköposteihin manuaalisista ja automatisoiduista prosesseista johtuen.

(Hovi, Huotari, Lahdenmäki 2005, 16)

Näiden operatiivisten järjestelmien heikkouksien takia usein päädytäänkin siirtämään niiden sisältämä data uuteen paikkaan, jossa sen hyötykäyttö on helpompaa. Tällaista keskitettyä datasäilöä, joka yleensä palvelee erinäisiä käyttötarkoituksia esimerkiksi yrityksen raportoinnissa ja datahallinnassa kutsutaan Tietovarastoksi (Data Warehouse Concepts 2020).

Operatiivisten järjestelmien heikkouksia peilaten, tietovarastoinnissa yleensä pyritään paikkaamaan näiden datalähteiden puutteita ottamalla huomioon ainakin seuraavanlaisia seikkoja:

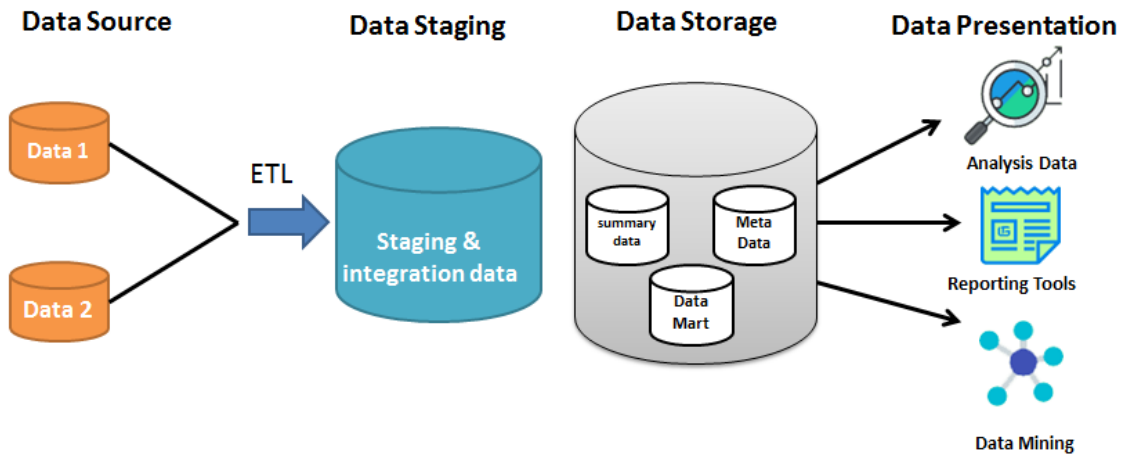
- Tietovarastossa olevasta datasta kerätään historiatietoa analysointia ja raportteja varten.
- Eri järjestelmistä tuleva data yhdenmukaistetaan, jotta eri lähteiden yhdisteleminen on helppoa.
- Osa tiedosta on jalostettu tai summattu valmiiksi helpompaa käyttöä varten.
- Rakenne ja tiedon muoto on muokattu loppukäyttäjiä ajatellen.

(Hovi, Huotari, Lahdenmäki 2005, 134)

Tällä tavalla varmistetaan, että eri järjestelmistä johdettu tieto on helposti saatavilla sellaisessa muodossa, että sen hyötykäyttö on vaivatonta. Yrityksen eri osastojen työntekijät saattavat olla hyvinkin kiinnostuneita tietyistä yrityksen datan osista, mutta heiltä saattaa puuttua teknistä osaamista tai käyttöoikeudet kysellä sitä nykyisistä tietolähteistä. Tietovarastossa tämä data voidaan räätälöidä juuri sellaiseksi, että siitä on helppo saada irti juuri sitä mitä halutaan.

3.3 ETL

Tietovaraston toimintaperiaatteisiin tärkeänä osana on tiedon hankinta ja siirto lähdejärjestelmistä itse tietovarastoon (Kuvio 2). Tämän yleensä hoitaa ETL (Extract, Transform, Load) -prosessi, jonka tehtävänä ei ole ainoastaan kuljettaa dataa tietolähteestä toiseen mutta usein myös muokata sitä erilaiseen muotoon. Yleensä lähdejärjestelmän data saattaa vaatia puhdistusta tai rikastamista jostain toisesta tietolähteestä ennen kuin se on kelvollista siirrettäväksi tietovarastoon. Jo pelkkä relaatiotietokannan taulujen ja kenttien uudelleennimeäminen voi selkeyttää haettua dataa ja tehdä siitä käyttökelpoisempaa loppukäyttäjälle. Yleensä lähdejärjestelmien data siirretään muokkaamattomana tietovaraston Staging-alueelle, josta se sitten jatkojalostetaan tietovaraston varsinaiseksi dataksi. Tällä tavoin lähdejärjestelmiä ei kuormiteta liiaksi ja tietovarastolla on mahdollisuus myös käyttää dataa uudestaan esimerkiksi myöhemmässä ETL-prosessissa.



Kuvio 2: Tietovaraston rakenne (Data Warehouse Architecture 2020)

Tietovaraston alkupäässä olevat datalähteet vaihtelevat mutta yleensä taustalla on ainakin yksi isompi operatiivinen tietolähde, jota yritys käyttää jokapäiväisessä liiketoiminnassaan. Tämä voi olla esimerkiksi toiminnanohjausjärjestelmän tai asiakastietojärjestelmän takana

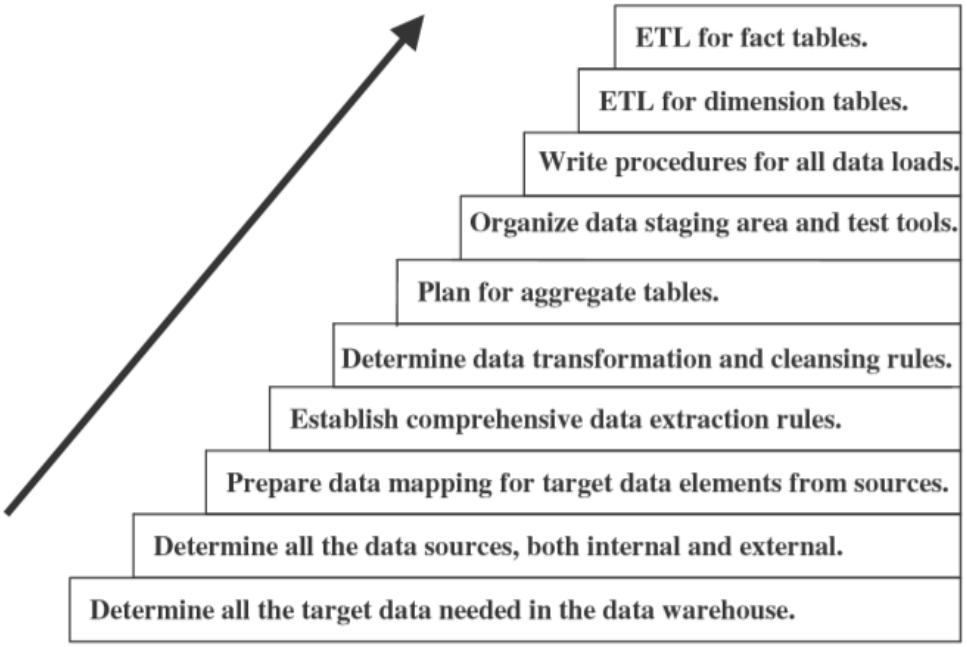
oleva tietokanta. Yleensä tämän lisäksi yrityksessä on myös pienempiä tietolähteitä, joiden sisältämä data on elintärkeää liiketoiminnan kannalta, tämä voi olla esimerkiksi verkkolevyillä säilytettävät Excel-tiedostot tai intranetissä ylläpidettävät tiedot. Tämä johtaakin siihen, että ETL-prosessien suunnittelu ja toteutus voi olla koko tietovarastoprojektin työläimpiä vaiheita, jokaiselle eri tietolähteelle saatetaan joutua rakentamaan omanlaisiaan siirtomekanismeja, jos valmiita ratkaisuja ei ole tarjolla. Yleisiä ongelmia joihin ETL-prosesseja kehittäessä voi törmätä:

- Tietojärjestelmät saattavat sijaita palvelimilla, jotka toimivat eri tekniikalla tai käyttöjärjestelmällä kuin tietovaraston järjestelmä.
- Järjestelmät saattavat olla vanhoja ja pyöriä vanhentuneilla tekniikoilla, joita ei enää uusissa järjestelmissä tueta.
- Sama data saattaa olla tallennettu kahteen eri järjestelmään eri tavalla, jolloin on päätettävä kumman järjestelmän data käyttökelpoisempaa.

(Ponniah 2001, 259)

Vaikkakin ETL-prosessien tärkein tehtävä on itse tiedon siirtäminen, tarvitsee siirrettävää dataa myös jollakin tavalla muokata ennen kuin se on valmista tietovarastoon tallennettavaksi. Matkalla lähdejärjestelmästä tietovarastoon data saattaa käydä läpi useamminkin muokkausvaiheen esimerkiksi:

- Roskarivien siivoaminen (puuttuvat tiedot, väärät tiedot, duplikaattidata)
- Datan uudelleenorganisointi, tiedon irrottaminen erillisiin tyyppitauluihin
- Summaus ja aggregointi.
- Tiedon anonymisointi (henkilötunnusten ja nimen poisto tai obfuskointi)



Kuvio 3: ETL-Prosessin vaiheet (Ponniah 2001, 261)

Kun data on siirretty ja muokattu haluttuun muotoon tietovaraston sisälle voidaan sitä helposti analysoida. Yleisesti tietovaraston datasta tehdään syvempää analyysia ja raportteja erillisillä siihen tarkoitetuilla sovelluksilla, ja dataa saatetaan myös jatkojalostaa eteenpäin esimerkiksi tapauksissa, joissa tietovarasto sisältää tietoja, joita voidaan hyödyntää esimerkiksi sovelluksissa, viestinnässä tai tekoälyratkaisujen kanssa.

4 Hyvät käytänteet

Tietovarastoprojektit eivät yleensä ole kevyitä ja helppoja toteuttaa, joten niiden suunnittelu vaatii paljon valintoja, joilla voi olla kauaskantoisia vaikutuksia. Onnistuneen projektin takaamiseksi onkin suunnittelutyössä tehtäviä ratkaisuja syytä tarkastella tietovarastoinnin yleisten hyvien käytänteiden kautta. Tähän osioon on valittu tutkittavaksi viisi osa-aluetta, jotka koskettavat jokaista tietovarastoprojektia:

- Tietovaraston tyyppi: Minkälainen tietovaraston rakenne sopii mihinkin käyttötarkoitukseen.
- Tiedon laatu: Missä muodossa ja millä tavalla dataa tietovarastoon tallennetaan.
- Loppukäyttäjät: Miten tietovaraston lopulliset käyttäjät huomioidaan suunnittelutyössä.

- Elinkaari: Minkälaisiin yllätyksiin ja muutoksiin on syytä varautua tietovaraston elinkaaren aikana.
- Teho ja Kustannukset: Mitkä vaikuttavat tietovaraston tehokkuuteen ja sen lopullisiin kustannuksiin.

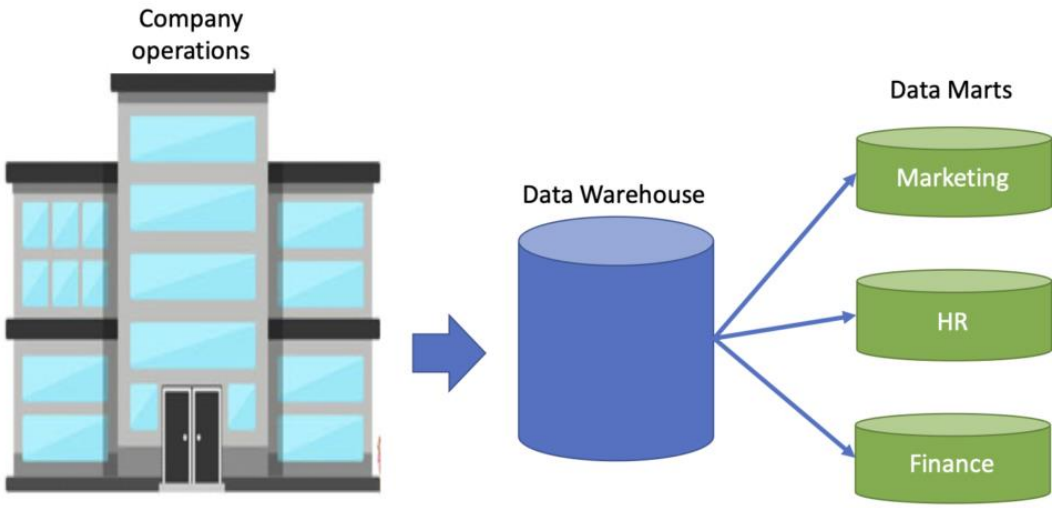
4.1 Tietovaraston tyyppi

Tietovarastoa suunnitellessa heti alkumetreillä on tehtävä päätös minkä tyyppistä tietovarastoa aletaan toteuttaa. Aiemmin pitkään vallalla ollut ja nykyään jo niin sanotusti perinteiseksi tietovarastoksi leimattu relaatiotietokantoihin perustuva malli, joka peilaa hyvin läheisesti lähdejärjestelmän muotoa. Perinteinen tietovarastomalli on yleensä toteutettu relaatiotietokantana, mutta siihen saatetaan liittää osioita, jotka eroavat hieman normaaleista operatiivisten järjestelmien kannoista. Perusdata saattaa olla tallennettu hyvinkin perinteisellä tavalla mutta sen päälle on ehkä rakennettu näkymiä tai erillisiä tauluja, jotka tarjoavat datan muodossa, joka palvelee paremmin raportointia ja analysointia, esimerkiksi historioivan datan tapauksessa on otettava huomioon, miten aikadimensio rakennetaan normaalin datan ympärille. Tietovaraston tietomalleissa onkin jo pitkään hyödynnetty niin s sanottua tähtimallia, jossa data jaetaan fakta- ja dimensiotauluihin (Hovi, Ylinen & Koistinen 2001, 94). Tämän suunnittelumenetelmän periaatteena onkin juuri helpottaa esimerkiksi aikadimension käyttöä raportoinnin yhteydessä.

4.2 Datamart

Joskus kokonainen tietovarasto voi olla liian raskas liiketoiminnan tarpeisiin nähden tai jo olemassa olevasta tietovarastosta halutaan irrottaa pienempi osa-alue tietyn liiketoimintayksikön käyttöön. Tällaista tietovarastoa pienempää ja vain tietyn ryhmän dataa sisältävää tuotetta kutsutaan datamartiksi. Kokonaisesta tietovarastosta voidaan esimerkiksi irrottaa oma erillinen datamart markkinointiosaston datalla, jota tämä yksikkö voi sitten käyttää liiketoimintansa hyväksi (Coronel & Morris 2017, 610). Datamartin sisältämä data voi myös olla jo valmiiksi aggregoitua haluttuun raportointimuotoon eikä enää samassa muodossa kuin sen taustalla olevassa tietovarastossa. Jos yrityksen datamäärä on pieni ja tiedon käyttäjäjoukko rajattu saatetaan joskus datamart liittää myös suoraan lähdejärjestelmiin ja käyttää sitä

tietovaraston korvikkeena (Data Warehouse Concepts 2020).



Kuvio 4: Mahdolliset datamartin käyttökohteet (Haddou, M. 2020)

Datamart voi myös soveltua yrityksille ns. harjoittelutietovarastoksi, joka toimii prototyyppinä jonkun liiketoiminta-alueen osalta. Jos tämä pienempi kokonaisuus saadaan toteutettua ilman ongelmia, siirrytään varsinaisen tietovaraston kehitystyöhön.

4.3 Tiedon laatu

Tietovaraston suunnitteluvaiheessa on tärkeää hahmotella missä muodossa lopullinen tieto sinne tallennetaan. Tietomallissa on otettava huomioon eri lähteistä yhdistetyn datan ominaisuudet ja kuinka ne saadaan muotoon, joka palvelisi parhaiten tiedon loppukäyttäjää. ETL-prosessien logiikalla voidaan sisään tuotua dataa muokata vapaasti ja esimerkiksi suodattaa datasta pois hyödyttömät rivit, joita ei tietovarastossa voida käyttää. Onkin yleistä, että tässä vaiheessa tietovarastoprojekteja huomataan lähdejärjestelmän datan laadussa olevan ongelmia, joka saattaa johtaa merkittävään kehitystyöhön tai jopa yrityksen toimintatapojen muutoksiin. Tiedon laadulla on myös rooli tietovaraston käyttöönotossa, sillä Törmäsen (1999) mukaan ”jos tiedon laatu ja ylläpito on huonoa, niin tietovaraston käyttöaste ja hyödynnettävyys laskee samassa suhteessa” (38).

Alkuperäisen tietovarastointiajattelun neljä keskeisintä teesiä dataa ajatellen ovat, että tietovaraston pitäisi olla: Integroitu (Integrated), asiaperustainen (subject-oriented), aikasidonnainen (time-variant) ja muuttumaton (non-volatile) (Törmänen 2017, 9). Tietovaraston tapauksessa integroitu tarkoittaa, että tietovaraston sisältämä data on yhtenäisesti tallennettu sovittuun muotoon, joka myötäilee eri datalähteiden ja osastojen mallia. Samanlaista dataa saatetaan tallentaa monella eri tavalle eri tietolähteissä, joten on tärkeää, että tämä yhtenäistäminen suunnitellaan niin että tiedon löytäminen tietovarastosta on vaivatonta.

Asiaperustaisuuden tavoitteena on tallentaa tietovarastoon data sellaisessa muodossa, jossa sitä on helpoin käyttää ja jolla se palvelee loppukäyttäjien tavoitteita. Koska tietovaraston ei tarvitse palvella mitään operatiivisen järjestelmän tallennustarpeita voidaan siellä oleva data muokata täysin lukemista varten ja näin sen rakenne voidaan suunnitella palvelemaan jotain tiettyä aihealuetta. Tällä tavoin esimerkiksi yrityksen eri osastoille voidaan tarjota dataa heidän tarvitsemaltaan näkökannalta muokattuna sellaiseen muotoon, josta vastauksia on helpoin löytää.

Tämä operatiivisista järjestelmistä riippumaton tietovaraston toiminta mahdollistaa myös aikasidonnaisuuden, jolloin datasta voidaan luoda erilaisia aikasarjoja ja historiatietoa. Operatiivisen järjestelmän datan tilasta voidaan ottaa jokaöisiä tallenteita, joista sitten aggregoidaan laajempia raportteja esimerkiksi viikko-, kuukausi- tai vuositasolla. Tähän ajatukseen yhdistyy myös neljäs ajatus muuttumattomuudesta: tietovarastoon tuotu data ei koskaan häviä eikä sitä olisi tarkoitus myöskään poistaa. Tällä tavalla operatiivisesta järjestelmästä käytön myötä katoava tieto olisi aina saatavilla tietovaraston sisältä. (Coronel & Morris 2017, 607)

Liiketoiminnan kannalta tärkeintä tietoa on tietenkin operatiivisista järjestelmistä saatu tieto mutta tietovaraston sisällössä on otettava huomioon myös data, joka syntyy itse tietovarastoinnin sivutuotteena. Tätä dataa kutsutaan metatiedoksi ja se on niin sanottua ”tietoa tiedosta” (Hovi, Hervonen & Koistinen 2009, 43). Metatiedon tarkoituksena on antaa käyttäjille lisäinformaatiota tiedon luonteesta ja alkuperästä. Metatieto voi koostua liiketoiminnallisesta lisätiedosta, joka kertoo esimerkiksi mitä menetelmiä käyttäen tieto on kerätty ja minkä liiketoiminta-alueen dataa tieto on. Metatieto voi olla myös täysin teknistä tietoa esimerkiksi siitä mistä lähdejärjestelmistä kyseinen tieto on peräisin tai milloin tieto on tietovarastoon siirretty tai milloin sitä on ETL-prosessien toimesta muokattu. Metatieto saattaa myös kertoa kuka tiedon omistaa ja kenellä on oikeudet sen käyttöön. (Hovi, Hervonen & Koistinen 2009, 43). Teknisen metadatan hyöty on ensisijaisesti tietovaraston kehittäjille, sillä yleensä tämä data on kriittistä silloin kun lähdejärjestelmissä tapahtuu odottamattomia muutoksia tai tietovaraston toiminnassa havaitaan ongelmia, joiden syitä on syytä jäljittää jo lähdejärjestelmätasolla.

4.4 Loppukäyttäjät

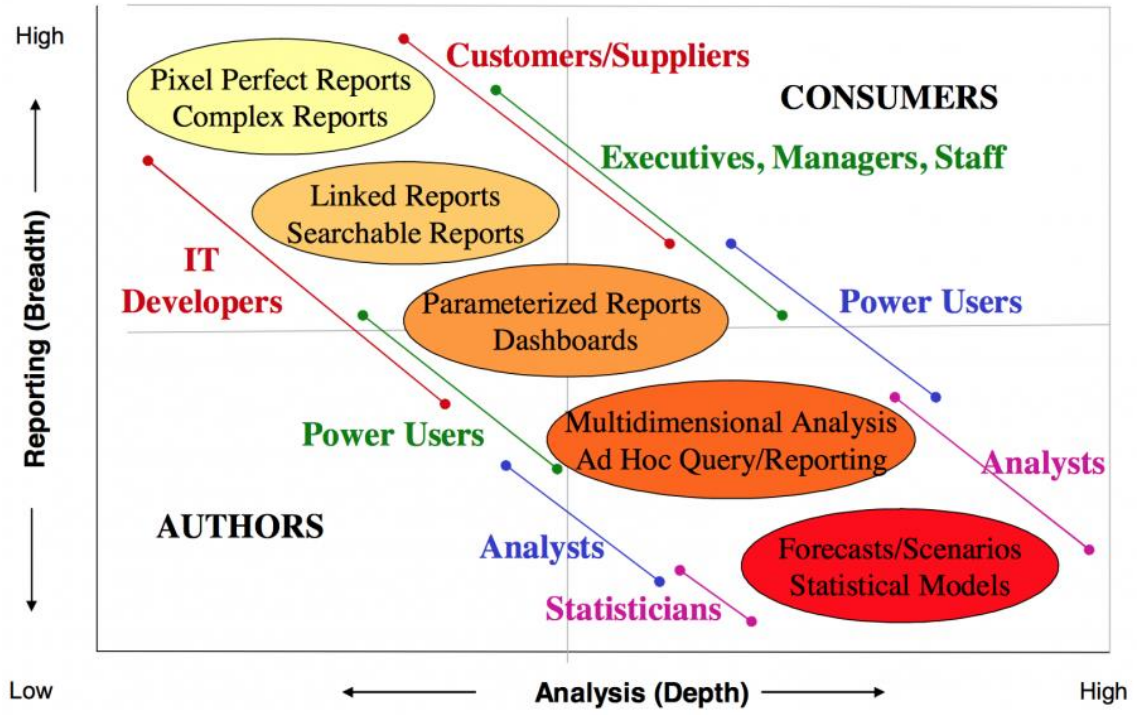
Tietovaraston olemassaolo yrityksessä ei pitäisi olla sen itseisarvo vaan koko projektin pitäisi lähteä tietystä tarpeesta. Tästä syystä on hyvä olla tietoinen loppukäyttäjien tarpeista ja siitä, miten tietovarastoa tullaan yrityksessä hyödyntämään. Loppukäyttäjät tarvitsevat erilaisia kombinaatioita dataa yrityksen eri toiminta-alueilta ja heidän datahakutaitonsa saattavat vaihdella huomattavasti. Onkin tärkeää jo projektin alkuvaiheessa olla perillä siitä mihin tarkoitukseen tietovarastoa käytetään ja minkälaisia vaatimuksia sen käyttäjillä nyt ja

tulevaisuudessa on (Silvers 2012, 6). Ajattelun onkin syytä lähteä siitä mitä tietovarastolla halutaan saavuttaa ja minkälaisia käyttäjiä sillä halutaan palvella, koska: ”Tietovarastointi tarvitsee käyttäjiä pysyäkseen hengissä.” (Törmänen 1999, 28)

Tietovaraston datan käyttäjät voidaan jakaa erilaisiin ryhmiin:

- Sisällönkatsojat: Nämä käyttäjät kuluttavat valmiita raportteja ja yleensä ovat tottuneet katsomaan staattisia taulukoita tai graafeja, mutta mahdollisesti käyttävät myös joitakin interaktiivisia raportteja, joissa dataa voidaan suodattaa.
- Datalöytöretkeilijät: Käyttäjät, jotka haluavat sukeltaa hieman pintaa syvemmälle ja löytää yhteyksiä kahden erilaisen data-alueen välillä. Käyttäjät yleensä käyttävät valmiita raportteja mutta odottavat niiltä enemmin muokattavuutta ja ominaisuuksia, joilla asioiden yhteyden toisiinsa tulevat esiin.
- Sisällöntuottajat: Nämä käyttäjät luovat tietovaraston datasta uusia raportteja ja visualisointeja. Näillä käyttäjillä on suora pääsy tietovaraston datakerrokseen mutta usein jonkun metadatatason lävitse (esimerkiksi näkymien tai taulujen, joissa dataa on koostettu käytettävämpään muotoon). He käyttävät datan käsittelyyn yleensä jotain valmiita työkaluja kuten raportointisovelluksia.
- Hakuekspertit: Tämä ryhmä koostuu ohjelmoijista ja data-analytiikoista jotka käyttävät datahakuihin yleensä suoraa ohjelmakoodia ja skriptejä. He yleensä hakevat datansa suoraan tietovaraston perustauluista. Taustalla voi olla tarve esimerkiksi luoda trendikäyriä ja ennusteita käyttäen isoja datamassoja.

(OpenText 2014)



Kuvio 5: Erilaiset loppukäyttäjät (OpenText 2014)

Olisi hyvä tunnistaa nämä ryhmät yrityksen sisällä ja jo tietovarastoa suunnitellessa kuulla heidän mielipiteitään siitä, mitä he tietovarastolta haluavat. Tällä tavalla suunnittelu ja toteutustyö ohjautuu kohti oikeaa haluttua päämäärää. Fon Silvers kiteyttää asian jotakuinkin näin: tietovaraston suunnittelussa tärkeää on se miltä suunnalta itse varaston suunnittelua lähdetään toteuttamaan. Jos ajatuksena on vain tehdä tietovarasto, on tuloksena vain tietovarasto. Jos taas lähdetään siitä ajatuksesta, että tietovaraston tarkoituksena on pystyä generoimaan esimerkiksi tämän päivän myyntiraportti, aletaan tietovarastoa suunnitella sellaisilla ratkaisuilla ja maaleilla, jotka johtavat siihen, että sillä pystytään tuottamaan tämän päivän myyntiraportti. Ja kun tämä maali on saavutettu, on helppo lähteä jalostamaan toiminnallisuutta tästä eteenpäin esimerkiksi tuottamalla dataa, jolla voidaan generoida edellisen päivän myyntiraportti. (Silvers 2012, 8).

On myös hyvä ajatella siltä kannalta, että vaikka kaikkia näytä datan kuluttajia ei yrityksestä tällä hetkellä löytyisikään, voi tilanne tulevaisuudessa olla toinen. On siis varauduttava siihen, että datan käyttö tulee laajenemaan tai keksitäänkin uusia tapoja hyödyntää olemassa olevaa dataa. Jokaisen tason loppukäyttäjät olisi siis syytä ottaa huomioon, vaikka sen tason käyttäjiä ei tällä hetkellä olisi vielä olemassa. On todennäköistä, että datan hyötykäyttö ja siihen kohdistetut kyselyt tulevat lisääntymään ja monimutkaistumaan mitä enemmin yrityksen dataa tulee käyttäjille tarjolle. Tulevaisuudessa modernin tietovaraston pitäisikin pystyä vastaamaan kaikkiin käyttäjän esittämiin kysymyksiin, palauttaa tuloksia nopeasti ja olla vielä

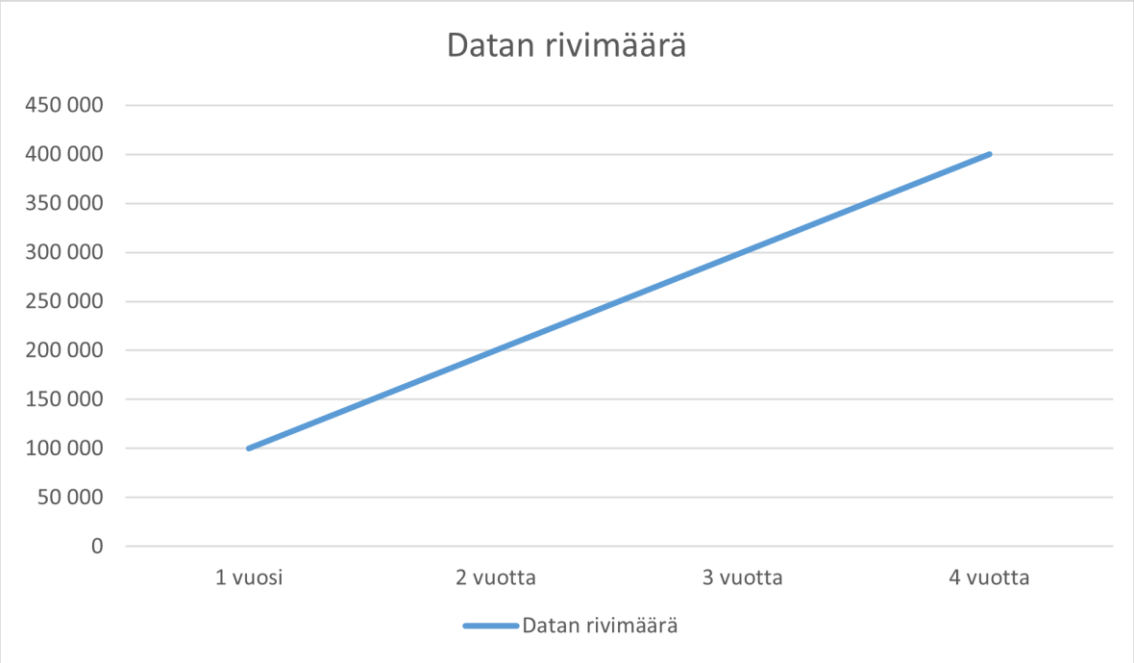
kaiken lisäksi kustannustehokas, jotta se voisi palvella kaikkia sen käyttäjiä (Mohanty, Jagadeesh & Srivatsa 2013, 96)

4.5 Elinkaari

Tietovarastoprojektia suunniteltaessa on syytä myös ottaa huomioon, miten tietovarasto palvelee yritystä myös tulevaisuudessa. Muutokset lähdejärjestelmiin voivat esimerkiksi aiheuttaa kauaskantoisia muutoksia datan laatuun ja jopa pahimmassa tapauksessa tehdä koko tietovaraston käyttökelvottamaksi. Myös datan määrä ja laatu voivat ajan saatossa muuttua esimerkiksi yrityskauppojen seurauksena.

Yksi selvin huomioonotettava seikka tietovarastoa suunniteltaessa on datan määrän kasvu sen eliniän aikana. Yleensä lähdejärjestelmät tuottavat liiketoiminnan seurauksena koko ajan huomattaviakin määriä uutta dataa, jota tietovarasto imee itseensä joka päivityskerralla. Tähän on siis syytä varautua sillä joka yö kasvava datamassa voi joko täyttää tietovaraston taustalla olevan tallennuskapasiteetin tai vaihtoehtoisesti venyttää datan siirtoaikaa ja näin aiheuttaa ongelmia tietovaraston toimintalogiikassa. Datan lisäksi saattavat kasvaa myös tietovaraston käyttäjämäärät, joka voi johtaa itse tietovaraston käytön hidastumiseen. (Linsteedt & Olschimmke 2016. 6)

Varsinkin historioivissa tietovarastoissa datamäärän kasvu vuositasolla voi olla hyvinkin suurta. Onkin siis erityisen tärkeää suunnitella tietovaraston taustainfra kestämään suuretkin datamassat tai vaihtoehtoisesti luoda loogisia sääntöjä, joiden perusteella vanhaa tarpeetonta dataa voidaan alkaa tietovarasto tuhoamaan. On turha säilöä tietovarastossa dataa, jota kukaan ei käytä ja jolle ei ole liiketoiminnalle enää minkäänlaista merkitystä. Siivotusta datasta voidaan kuitenkin ennen poistamista ottaa talteen vielä jotain koostettua tietoa josta voi olla hyötyä jossain tulevaisuuden käyttötarkoituksessa.



Kuvio 6: Datan määrän kasvu tietovaraston elinkaaren aikana (tutkimuksen tekijän havainnekuva).

Pilviteknologian aikakautena datan ja käyttäjien määrä ei sinänsä tuota suuriakaan ongelmia tietovarastoprojekteissa, sillä suuret pilvipalvelut yleensä tarjoavat saumatonta skaalautuvuutta järjestelmiinsä. Jos järjestelmän tila tai teho alkaa loppua, on helppo vain kääntää ohjausportaalista lisää tehoa, tämä tosin saattaa vaikuttaa tietovaraston käyttökustannuksiin hyvinkin dramaattisesti. Pilvipalvelujen valmis infra mahdollistaa myös vanhan datan arkistoinnin automatisoinnin valmiilla työvälineillä jotka voivat siirtää sitä paikkaan jossa sitä on kustannustehokkaampi säilyttää.

4.6 Teho ja kustannukset

Tietovarastoa suunnitellessa ja sen kehitystyössä yksi tärkeä kriteeri sen toiminnalle on tehokkuus, miten hyvin lopullinen tuotos selviää siltä vaadituista ominaisuuksista. Helpointa on havainnoida tietovaraston teknistä suorituskykyä, esimerkiksi miten nopeasti se päivittää itsensä tai kauanko tiedon hakeminen tietovarastosta kestää.

Jos tietovarasto päivittyy esimerkiksi kerran yössä ja sen sisältämää dataa tarvitaan raportointiin yrityksen normaalien työaikojen puitteissa, on sen syytä suoriutua öisestä päivitysprosessistaan tietyssä aikaikkunassa. Tietovaraston päivitysajojen venyessä saattavat myös lähdejärjestelmiin kohdistuvat ETL-datahakuajot haitata normaalia liiketoimintaa hidastamalla järjestelmien toimintaa. Olisi siis ensisijaisen tärkeää suunnitella päivitys tehokkaaksi ja ajastaa se niin että jopa pahimmissa tapauksissa siitä selvitään aikaikkunan puitteissa.

Loppukäyttäjien näkökulmasta on tärkeää, että tiedonhaku järjestelmästä on nopeaa ja vaivatonta. Osana tätä ovat tietovaraston tietokannan resurssit ja tapa, jolla tieto on varastoon järjestetty. Pilvitietokantojen tapauksessa resurssien skaalaus ylöspäin tilanteessa, jossa tehoa tarvitaan lisää, on yleensä helppoa mutta nostaa tietovaraston käyttökustannuksia. Onkin järkevämpää ensin etsiä muita keinoja tietovaraston käyttötehon parantamiseen, kuten tietovaraston tietokannan rakenteiden optimoiminen. SQL-pohjaisen relaatiotietokannan tapauksessa tähän on muutama erilainen vaihtoehto. Kun käyttäjää hakee dataa tietovarastosta, voidaan prosessin vaiheet jakaa neljään eri osaan:

1. Käyttäjä (täi hänen käyttämänsä sovellus) luo kyselyn, jolla haluaa dataa hakea.
2. Kysely lähetään ohjelmalle, joka hakee dataa tietovarastosta.
3. Kysely suoritetaan tietovarastossa.
4. Kyselyn tulokset palautetaan käyttäjälle.

(Coronel & Morris 2017, 516)

Kohta yksi voi käsittää suoran SQL-lausekkeen tai sitten jonkun raportointivälineen luoman kyselykutsun. Käyttäjän omat kyselylausekkeet ovat tietenkin osittain riippuvaisia siitä, miten hyvin käyttäjä ymmärtää hakemaansa dataa ja osittain myös hänen omista teknisistä taidoistaan. Tehoa on mahdollista saada lisää kouluttamalla tietovaraston käyttäjiä tekemään tehokkaampia hakulauseita tai toimittamalla heille valmiita hakuaihioita, joista he voivat johtaa haluamansa hakulauseet omilla kriteereillään. Jos kyseessä on kolmannen osapuolen sovellus, voi olla hyvin vaikeaa vaikuttaa suoraan sen tekemiin kyselyihin ja muokata niitä tehokkaammiksi. Jo valmista tietokantarakennetta voidaan myös virittää luomalla kantaan indeksejä tai siirtämällä tiettyjä yleisiä hakutuloksia pysyvästi palvelimen keskusmuistiin, josta ne voidaan hake nopeasti tarvittaessa (Hovi, Huotari, Lahdenmäki 2005, 134).

Yleensä vaiheet kaksi ja kolme riippuvat hyvin paljon sovelluksista, joita käytetään tietovaraston tiedon prosessointiin, joten tietovaraston toteutuksessa niihin on vaikea teknisesti vaikuttaa. Jos välineet ovat jo yrityksessä vakiintuneita ja niiden käyttöön liittyy paljon osaamispääomaa, on niitä myös vaikea vaihtaa paremmin optimoituihin pelkästään tietovaraston tarpeita varten. Yleensä näiden sovellusten heikkouksien ja toimintaperiaatteiden kanssa on vain pyrittävä elämään ja yritettävä tehostaa tietovaraston hakuprosessia muilta osa-alueilta. Tiedon prosessoinnin nopeuteen tietovarastossa voidaan kuitenkin vaikuttaa suunnittelemalla tietovaraston rakenne tietynlaisia hakekriteerejä varten.

Tietovarastoprojektissa lopulliset kustannukset voidaan jakaa kahteen ryhmään: kertakustannukset tietovaraston suunnittelusta ja kehityksestä (development costs) ja juoksevat kustannukset tietovaraston ylläpidosta sen valmistuttua (running costs). Kehitysvaiheessa tehdyt valinnat vaikuttavat niin ikään itse kehityskustannuksiin mutta huomattavissa määrin myös juokseviin kustannuksiin. Tietyn komponentin tai tekniikan valinta voi kehitystyössä säästää

paljonkin aikaa ja rahaa mutta saattaa sisältää ylläpitokustannuksia, jotka yhteenlaskettuna koko tietovaraston elinkaarelle voivat kohota hyvinkin suuriksi. Muita mainittavia Tietovarastoon liittyviä kustannustekijöitä ovat:

- Datan säilytyskustannukset
- Huonon laadun hinta
- Huonon suunnittelun hinta
- Muuttuvien liiketoimintavaatimusten hinta.

(Linsteedt & Olschimmke 2016. 10)

Säilytyskustannukset ovat näistä ryhmistä selvästi helpoiten ennakoitava ja budjetoitava osa-alue. Hinta datan säilytykselle pilvessä voidaan laskea selvästi laskureiden avulla ja budjetoida sen perusteella. Datan kasvukäyrä ei saata kuitenkaan olla ainakaan tietovaraston alkuvaiheessa täysin selvillä, joten yllätyksiä datan säilytyskustannuksissa voi elinkaaren aikana ilmetä. Huonon laadun ja suunnittelun hinta sen sijaan selviävät tietovaraston elinkaarin aikana hyvin nopeasti: kuinka paljon tunteja joudutaan käyttämään tietovaraston ongelmatilanteisiin ja näistä ongelmista johtuviin muutostöihin. Huono laatu ja suunnittelu kulkevatkin yleensä käsi kädessä ja säästöt suunnittelutyössä saattavat kostautua myöhemmin laadun kärsiessä.

Muuttuvien liiketoimintavaatimusten hinta on sidottuna paljon tietovaraston elinkaaren suunnitteluun. Datamäärän ja käyttäjien kasvua voidaan ennakoida mutta vaatimuksia datan ja raportoinnin suhteen ei niinkään. Saattaa olla, että joku osa-alue yrityksen datasta joka suunnitteluvaiheessa julistettiin tarpeettomaksi tietovaraston siirtämisen kannalta saattaakin olla liiketoimintakriittistä tulevaisuudessa.

5 Tietovarastoprojektin kuvaus

Tässä osiossa esitellään tutkimuksen kohteena oleva Vartas Oy:n toteuttama projekti, jossa asiakkaalle suunniteltiin ja kehitettiin tietovarastoratkaisu. Osiossa käydään läpi lähtötilanne ja esitellään tekniikka ja ratkaisut joihin toteutusta suunnitellessa päädyttiin.

5.1 Asiakkaan tarpeet

Asiakasyritys projektissa oli keskisuuri yritys, jonka data keskittyi suuremmalta osin jäsenyystietoon ja siihen liittyvään dataan. Asiakkaan suurin ongelma nykyisen ratkaisunsa kanssa oli raportoinnin ongelmallisuus ja datan yleinen vaikeakäyttöisyys. Varsinaisia lähdejärjestelmiä oli käytössä vain yksi mutta se sisälsi monia eri data-alueita, joissa oli esimerkiksi sisarjärjestöjen dataa, jota asiakas haluaisi tarkastella oman datansa ohessa. Aiemmin kaikenlainen

raportointi oli toteutettu SQL-kyselyillä ja Excel-tiedostoilla ja käytänteet hakukriteereille saattoivat vaihdella henkilöstä ja ajasta riippuen. Tarpeena oli yhtenäistää raportointia niin että se saataisiin automatisoitua ja ennen kaikkea sellaiseksi että jokainen raportti olisi haettu samoilla kriteereillä, jotta vertailu esimerkiksi eri vuosien välillä olisi helppoa ja luotettavaa.

Asiakkaan tarpeena oli siis keskitetty tietovarasto, josta datan haku ja raporttien laatiminen olisi vaivatonta. Osana projektia oli myös perusraportoinnin toteutus asiakkaalle mutta tarkoituksena oli myös tehdä tietovaraston rakenteesta sellainen, että yrityksen sisäinen henkilöstö pystyisi myös itse luomaan raportteja tietovaraston tarjoamista näkymistä. Tämän takia lopullinen ratkaisu nojasi monenlaisiin eri tapoihin tarjota dataa ulos tietovarastosta, osa näkymistä oli valmista dataa, jonka päälle oli helppo rakentaa raportteja ja osa taas antoi mahdollisuuden sukeltaa pintaa syvemmälle ja mahdollisti tarkastelun laajemmalla näkökannalla.

5.2 Tietovaraston ja datan rakenne

Kuten jo aiemmassa osiossa mainittiin, asiakkaalla on käytössä yksi lähdejärjestelmä joka tässä tapauksessa, on fyysinen Microsoftin SQL Server -palvelin, joka sisältää asiakkaan omia tietokantoja sekä sisarjärjestöjen tietokannat. Asiakkaan jäsenyystietoa sisältävä operatiivinen kanta toimi tietovaraston päätietolähteenä, jonka perusteella tietomalli luotiin. Sisarjärjestöjen tietokannat toimivan emoyhtiön tavoin saman sovelluksen operatiivisena kantana, joten niiden data oli helppo integroida osaksi tietovaraston päätietolähteen tietoja.

Joissakin tapauksissa asiakkaalla oli dataa, jota ei ollut varsinaisesti tallennettu mihinkään vaan se oli niin sanottua yleistä tietoa, tällaisia oli esimerkiksi tietyt hierarkkiset linkit sisarjärjestöjen välillä. Koska tietovaraston toiminnalle tämä tieto oli kuitenkin tärkeää luotiin tälle datalle oma säilytyspaikka, johon sitä alettiin keräämään ja josta sitä voisi hyödyntää tietovaraston ETL-ajoissa. Tällaiseksi valikoitui Microsoftin SharePoint Datalistat, joihin eiteknisten käyttäjien oli helppo lisätä uutta tietoa. Tämä siis loi alkuperäisen tietolähteen rinnalle myös uuden tietolähteen.

Kolmas tiedon muoto, jonka hyödyntäminen osoittautui projektin edetessä tarpeelliseksi, oli vuosien saatossa kerätty vanha data, jota oli mahdotonta enää hankkia lähdejärjestelmän datan nykytilasta. Koska lähdejärjestelmä ei millään tavalla historioi dataansa on osa datasta saatavissa vain lähdejärjestelmän nykyhetkestä ja heti kun data muuttuu, on sitä mahdotonta enää saada takaisin. Tämän takia osa vanhoista tilastoista oli syytä myös hyödyntää tietovarastossa. Nämä jo aiemmin kerätyt datat piti siis myös liittää osaksi tietovarastoa. Vanha data oli monessa eri muodossa kuten Excel-taulukoissa ja PowerPoint-esityksissä. Asiakas kuitenkin käytti omia resurssejaan ja muokkasi henkilöstönsä avulla suuren osan näistä vanhoista tiedoista muotoon, jossa tietovaraston ETL-prosessien olisi se helpoin lukea.

Tietovaraston sisällä data jaettiin kolmeen eri alueeseen: Staging, Core ja BI. Staging-alueella on raakadataa, joka siirretään ETL-prosesseille suoraan lähdejärjestelmästä muokkausta ja uudelleentallennusta varten. Tämä alue tyhjennetään joka yö ennen uusien ETL-ajojen alkua ja siirretään kokonaisuudessaan asiakkaan järjestelmästä tietovarastoon. Core-alue edustaa varsinaista tietovaraston dataa ja on pysyvää tietoa, joka lisääntyy ja päivittyy joka yö. Nykyisellään Core-alueen data ei ole itsessään historioivaa vaan jokainen päivitys yliajaa vanhan rivin datan. Tähän ratkaisuun päädyttiin koska perusdatan ei katsottu olevan sinänsä historioimisen arvoista vaan datasta luotiin tiivistetty muoto BI-Tauluihin, joka kertoi ylemmällä tasolla, mikä oli datan tilanne kullakin hetkellä. Mahdollinen ongelma tässä on, jos tulevaisuudessa tulee tarpeita uusille näkökannoille, on niiden datakeräys aloitettava nollapisteestä koska sitä ei voida hakea Core-datan historiatiedoista.

5.3 Tietovaraston toimintaperiaate

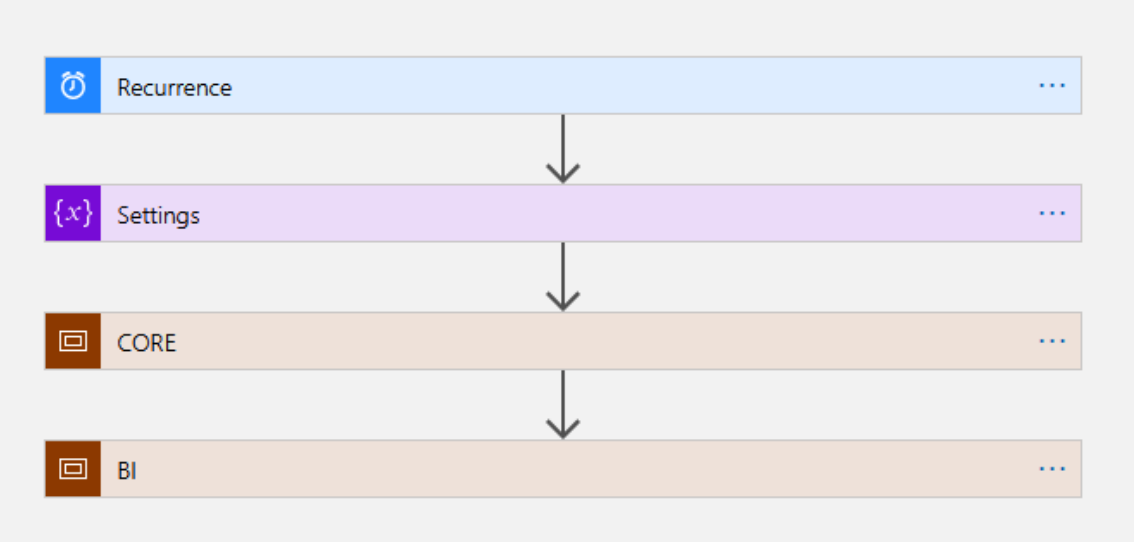
Tietovaraston tiedon päivittäminen perustuu ajastukseen, joka käynnistää ETL-prosessien ketjun haluttuna ajankohtana. Yö on operatiivisen järjestelmän käytön kannalta hiljaista aikaa, joten oli luonnollista ajoittaa tietovarasto päivittymään aamuyön hiljaisina tunteina. Asiakasyrityksen toiminta painottuu vahvasti normaaleihin virastoaikoihin, joten tietovaraston aikataulutuksen kriteerinä oli, että ETL-ajot olisivat valmiina ennen aamua, jolloin ensimmäiset työntekijät alkavat käyttää operatiivista järjestelmää (ja tulevaisuudessa myös tietovarastoa). Huomioon piti ottaa myös mahdolliset huolto- ja ylläpitoajot operatiivisen järjestelmän tietokantaan, joita sitä hallinnoiva toimittaja usein teki myös yön hiljaisina tunteina.

Tietovaraston datan alkuperä on siis asiakkaan oma operatiivisen järjestelmän tietokanta, josta se siirretään joka yö ETL-prosesseilla tietovaraston kannan Staging-alueelle. Tiedon muokkausta tapahtuu jo asiakkaan palvelimella sillä tietolähteessä toimivat hakuskriptit palauttavat vain osan datasta, kaikki taulujen sisältämä data ei ole tietovarastolle tarpeellista ja tällaisen ylimääräisen datan siirto palvelimelta toiselle aiheuttaisi vain ylimääräisiä siirtokuluja ja tuhlaisi resursseja. Monesti taulu joka lähdejärjestelmässä sisältää esimerkiksi 20 saraketta dataa kutistuukin siirrossa vain viisisarakkeiseksi tauluksi. Monessa lähdejärjestelmän taulussa on esimerkiksi sarakkeita, jotka ovat jäänteitä lähdesovelluksen vanhasta toiminnallisuudesta, joka on jo vuosia sitten poistettu käytöstä.

Seuraavaksi Staging-alueella sijaitseva data otetaan uudestaan käsittelyyn ja muokataan tietovaraston lopputaulujen skeemaan sopivaksi, ja joissain tilanteissa siihen myös yhdistetään toisista tietolähteistä saatua dataa. Tämän jälkeen tälle datalle suoritetaan ’delta detection’-mekaniikka, joka vertaa uutta dataa jo tietovarastossa olevaan dataan, tämä mahdollistaa vanhojen rivien päivityksen ja näin myös vältetään kaikkien rivien uudelleenkirjoitus tapauksessa, jossa ne jo löytyvät kannasta samassa muodossa. DD-mekaniikka skannaa Staging- ja Core-taulun sisällön ja vertaa tuloksissa molempien taulujen riveiltä löytyvää

tarkastussummakenttää, joka kertoo, onko rivin sisältö erilainen vai yhtenevä. Jos summa on sama, riviä ei päivitetä, jos taas erilainen päivitetään Core-taulussa oleva rivi Staging-taulun tiedoilla. Joihinkin tauluihin on myös rakennettu mekaniikka, joka mitätöi rivejä, jotka on poistettu lähdejärjestelmästä. Tietovarastosta ei koskaan poisteta rivejä, vaikka ne alkuperäisdatasta häviävätkin, ne vain passivoidaan muuttamalla niiden metakenttien ominaisuuksia. Voidaan ajatella, että tietovaraston taulut ovat tällä mekaniikalla ainakin osittain historioivia sillä vanhaa operatiivisesta järjestelmästä voidaan tutkia ainakin poistettujen rivien osalta.

Kun kaikki Core-alueen data on siirretty ja muokattu haluttuun muotoon alkaa tietovarastoinnin kolmas vaihe, jossa ETL-prosessit muokkaavat ja siirtävät Core-tauluissa olevaa dataa BI-tauluihin. Tämän vaiheen ETL-prosessit summaavat, suodattavat ja koostavat operatiivisen järjestelmän tietoa muotoon, jota loppukäyttäjän on helpompi käsitellä. Jäsenyysdatasta esimerkiksi lasketaan yöllisiä kokonaisjäsenmääriä eri ominaisuuksien näkökulmasta. Monet BI-alueen prosesseista myös yhdistelevät dataa useasta eri taulusta ja tekevät johtopäätöksiä näiden perusteella. ETL-prosessien lopullisessa logiikassa datan siirto Staging-alueelle ja sen jatkokehitys Core-alueen tauluihin tapahtuu osittain limittäin, joten molemmat molempien prosessit on sijoitettu ajokartassa Core-kattotermin alle. BI-prosesseja taas ei voida suorittaa ennen kuin koko Core-alueen data päivitetty data löytyy tietovaraston tauluissa, joten ne toteutetaan vasta kahden ensimmäisen vaiheen valmistuttua omassa BI-vaiheessaan (Kuvio 7).



Kuvio 7: Tietovaraston ETL-ajojen järjestys.

Kun tietovaraston koko päivitysprosessi on suoritettu onnistuneesti, on tietovaraston data valmista käytettäväksi raportointiin. Jos prosessissa ilmenee ajojen aikana ongelmia, pysähtyy sen suoritus siihen pisteeseen, jossa virhe ilmeni. Tällaisessa tapauksessa osa tiedosta jää päivittymättä ja ETL-prosessi on aloitettava uudestaan, jotta kaikki sen vaiheet saadaan

suoritettua. Yleensä tämä tapahtuu seuraavana yönä tai vähintään virastoaikojen ulkopuolella, jotta asiakkaan liiketoiminta ei häiriinny.

5.4 Välineet

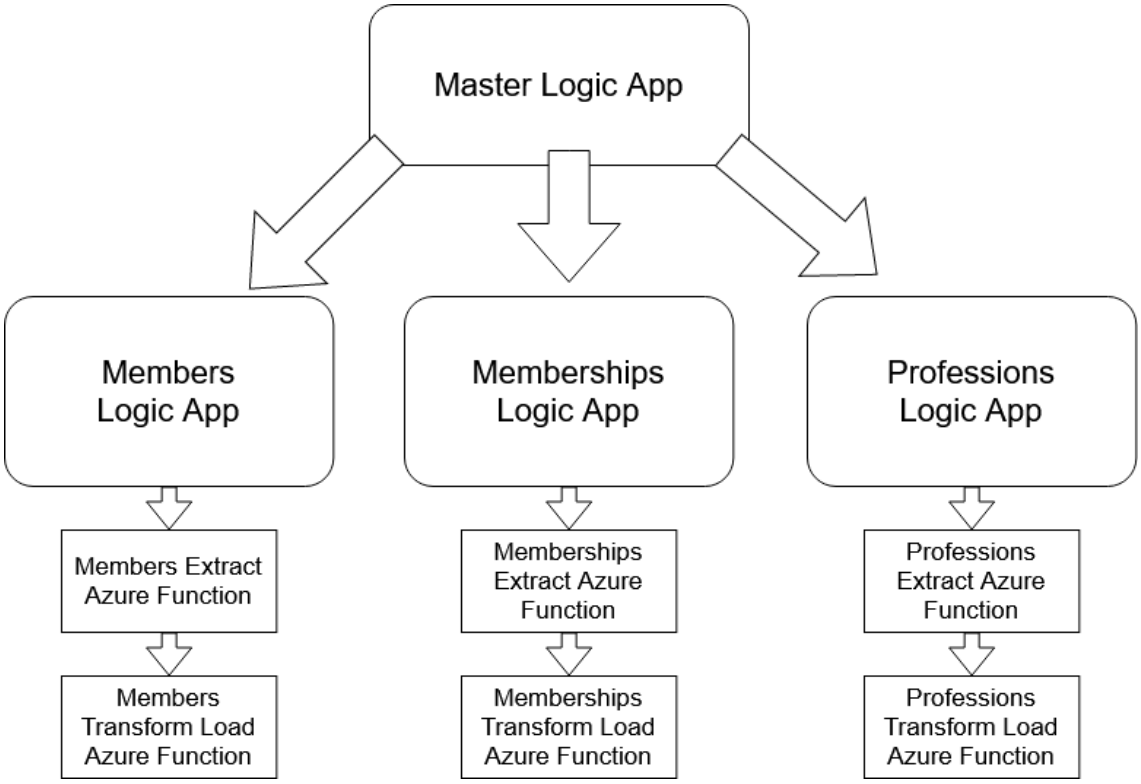
Yleisesti ottaen projektin lähtökohtana oli, että kaikki tietovaraston osat toteutetaan pilvipalveluina (pois lukien lähdejärjestelmä) ja mieluiten yhden palveluntarjoajan välineillä. Vartas Oy:n erikoisosaaminen painottuu Microsoftin Azure -pilvipalveluympäristöön ja sen tarjoamiin työvälineisiin, joten oli siis luonnollista, että koko tietovarasto toteutettiin näillä työkaluilla ja jo olemassa olevalla osaamisella.

5.5 ETL-prosessin komponentit

Itse tietovarasto on pilvessä sijaitseva Azure SQL Server tietokanta, joka on jaettu tuotanto- ja kehitysympäristöihin. Pilvessä sijaitseva SQL Server eroaa hyvin vähän lähdejärjestelmässä käytetystä paikallisesta (on-premises) tietokannasta toiminnallisuudessaan, joten tämä helpotti datan siirtoa lähdejärjestelmästä tietovarastoon huomattavan paljon. Samat SQL-hakulauseet toimivat molemmilla järjestelmillä ja tietokannan datatyypit ovat identtisiä. Suurena etuna oli myös, että kommunikaatio asiakkaan tietohallinnon työntekijöiden kanssa oli sujuvaa koska molemmilla oli käytössään samanlainen järjestelmä ja samat datahakujen työvälineet.

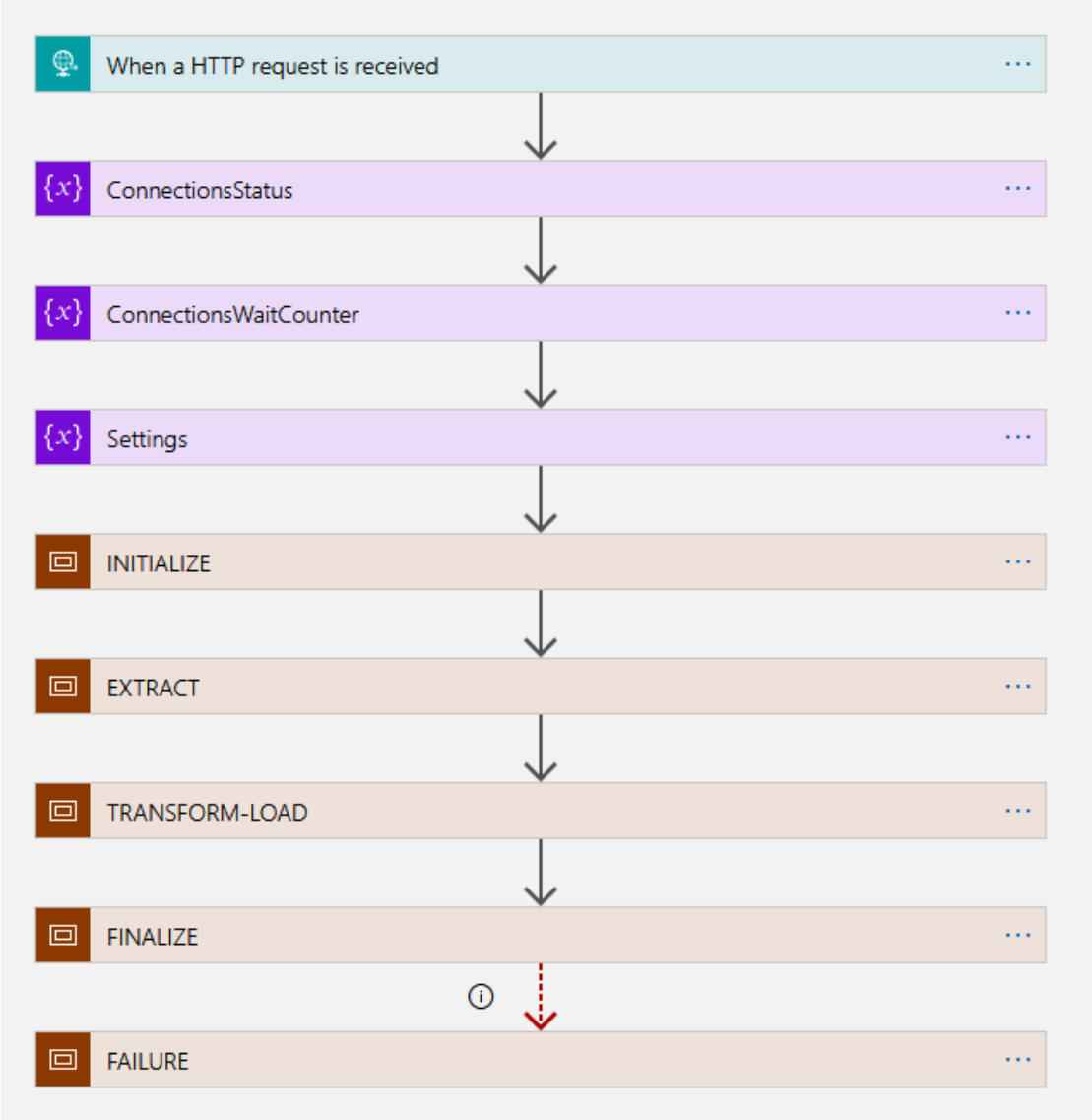
Itse tiedonsiirto päätettiin toteuttaa kerrosteisesti käyttäen Azuren perustyökaluja. Ylimmällä tasolla toimivat Azure Logic Apps -sovellukset, jotka käynnistyivät ajastetusti öisin suorittamaan ETL-ajoja. Logic Appit ovat Azuressa pyöriviä Flow-kaaviomaisia yksinkertaisia sovelluksia, jotka suorittavat yksinkertaisella logiikalla erinäisiä toimintoja. Logic Appien vahvuus on niiden nopea kehitys ja integrointi Azuren muihin työkaluihin. Ne myös kuvaavat prosessia visuaalisesti, jolloin niiden esittely asiakkaalle on helppoa.

Ylimpänä toimintoketjussa toimii Logic App -isäntäsovellus, joka on ajastettu käynnistymään keskiyöllä ja jonka tehtävänä käynnistää alatason ETL -prosesseja tietyssä järjestyksessä ja valvoa niiden tilaa ja sitä suoritettiinko prosessi loppuun onnistuneesti. Alatason Logic App -sovellukset vastaavat aina yhtä ETL-prosessia ja Core-datan tapauksessa vastaavat myös aina yhtä tietovaraston taulua.



Kuvio 8: ETL-prosessien rakenne Azuressa

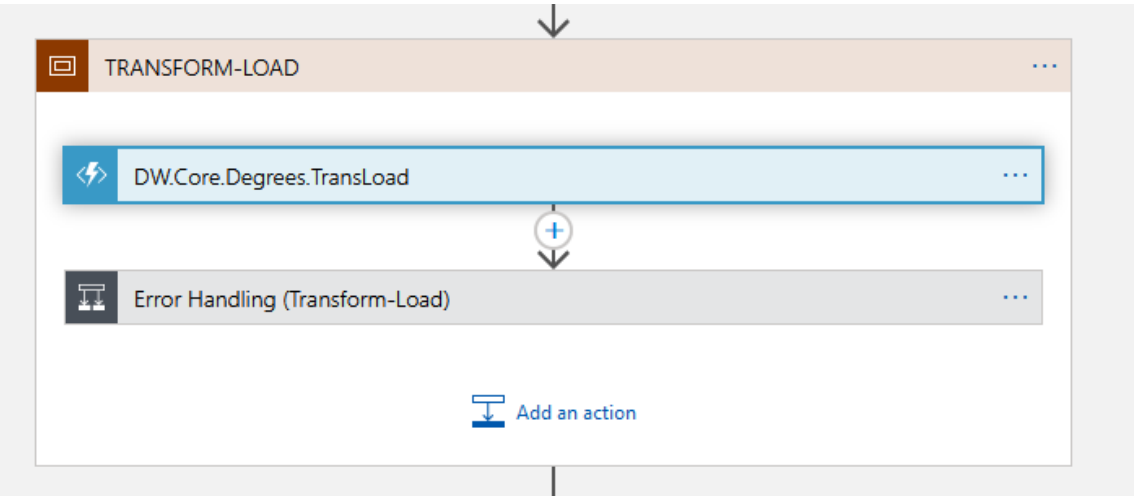
ETL-prosessin toimintaperiaate seuraa neljää vaihetta: Initialize, Extract, Transform Load ja Finalize. Initialize vaihe tyhjentää kulloisenkin ETL-prosessin Staging-taulun, jotta uusi data voidaan tuoda lähdejärjestelmästä. Extract käynnistää tiedonsiirtoprosessin Staging-tauluun, Transform-Load taas prosessin, joka muokkaa ja sitten siirtää datan lopulliseen Core-tauluun. Finalize vaiheen tehtävänä on kirjoittaa lokiviesti siitä, että prosessi onnistui ja lähettää tieto ylätason hallintasovellukselle prosessin valmistumisesta. Tämän lisäksi ETL-prosesseihin on sisäänrakennettu logiikkaa erinäisiä virhetilanteita varten, esimerkiksi jos tietokantayhteys Azuren ja asiakkaan tietokannan välillä on tilapäisesti katkennut.



Kuvio 9: ETL-prosessin Logic App sovellusrunko

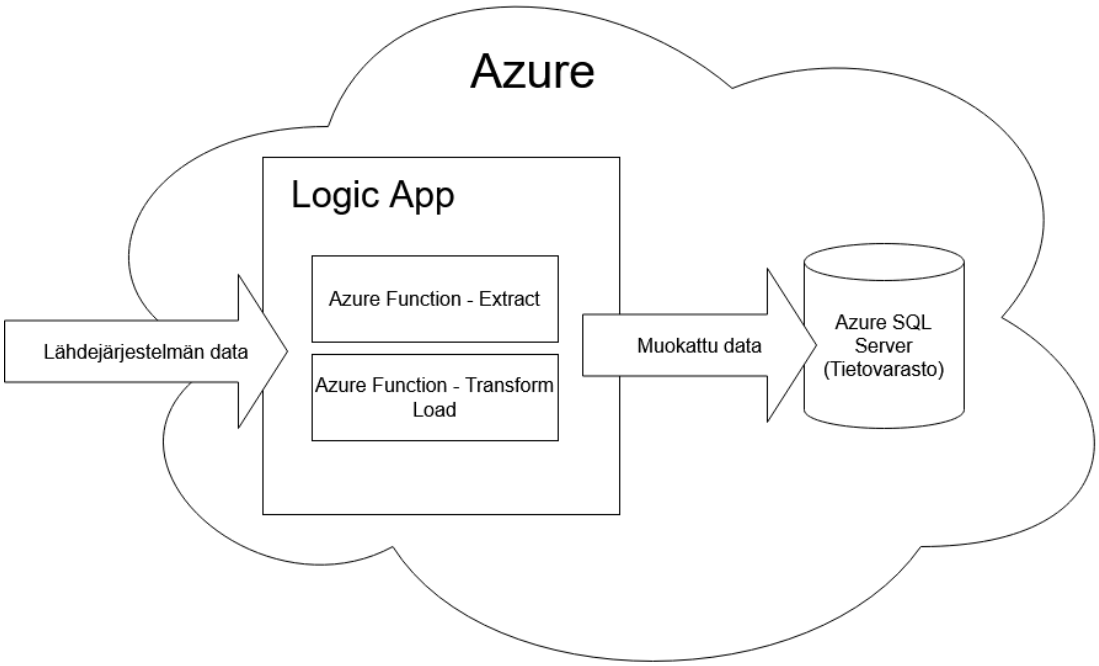
Itse siirto ja datamuokkaus tapahtuu Azure Functions -sovelluksilla, jota ETL-prosessin Logic Appit kutsuvat sisällään. Azure Functions -sovellukset ovat serverless-tekniikalla toimivia itsenäisiä mikropalveluita, jotka suorittavat C#-koodia, jolla ETL-prosessin logiikka on ohjelmoitu. Jokaiselle ETL-prosessille on luotu kaksi funktiosovellusta, yksi Staging-vaihetta ja toinen Transform-Load vaihetta varten. Yleisesti Transform ja Load vaiheet on eroteltu erillisiksi prosesseikseen, mutta tietovaraston kehitystyössä ilmeni, että nykyisten Core-prosessien datalle tekemät muokkaukset ovat niin yksinkertaisia, että ne voidaan suorittaa samaan aikaan latausvaiheen kanssa. Tällä tavalla kaksi vaihetta yhdistämällä projektin funktiosovellusten määrä pysyy puolet pienempänä ja näistä syistä nämä kaksi vaihetta päätettiin yhdistää ja

nimetä Transform-Load-vaiheeksi.



Kuvio 10: Funktiosovelluksen kutsu alatason sovelluksessa.

Staging-vaiheen funktiosovellukset kutsuvat lähdejärjestelmässä olevaa Stored Procedure pakettia, joka sisältää SQL-hakulauseita ja logiikan, jolla data palautetaan lähdejärjestelmästä. Tämän jälkeen funktiosovellus käsittelee dataa halutulla tavalla ja tallentaa sen tietovaraston Staging-tauluun käyttäen SQL-komentoja. Koska usea lähdejärjestelmän taulu saattoi sisältää satoja tuhansia rivejä dataa, oli funktiosovellustasolla otettava käyttöön eräajomenettely, jossa dataa siirrettiin osissa lähdejärjestelmästä tietovarastoon. Jo 100 000 rivin siirtäminen kerralla kuormitti sovellusta sekä tietovaraston SQL-serveriä niin rankasti että koko ETL-prosessi saattoi kaatua muistin loppumiseen. Pienen testailun jälkeen osoittautui että 25 000 riviä kerralla oli ajallisesti optimaalisin määrä eräajolle. Kun data oli siirretty Staging-tauluihin hoitaa Transform-Load vaiheen funktiosovellus datan muokkauksen ja Core-tauluihin lataamisen. Myös Transform-Load vaihe hyödyntää eräajoa ja käsittelee dataa 25 000 rivin erissä.



Kuvio 11: Kokonaisarkkitehtuuri

Joissakin tapauksissa jo ylempi Logic App -taso hoiti datan lukemisen lähteestä ja kirjoittamisen tietovaraston kantaan. Tämä siitä syystä, että yksinkertaisten luku- ja kirjoitusoperaatioiden tapauksessa oli turhaa toteuttaa koko prosessia liian monimutkaisilla vaiheilla. Tätä hyödynnettiin esimerkiksi tilanteissa, joissa asiakkaan operatiivisista järjestelmistä ei löytynyt tiettyä dataa, joka oli kuitenkin kriittinen osa liiketoimintaa. Näissä tilanteissa asiakas täytti tiedot manuaalisesti SharePoint -datalistoihin, joista ETL-prosessi kävi ne ajon aikana poimimassa.

6 Hyvien käytänteiden toteutuminen

Tutkimuksen tässä osiossa hyödynnetään teoriaosuudessa tutkittuja tietovaraston hyviä käytänteitä ja tutkitaan niitten kautta kriittisesti tietovarastoprojektissa tehtyjä ratkaisuja. Osiossa käydään läpi jokainen aiemmin mainittu hyvien käytänteiden kategoria ja tarkastellaan sitä vastaavaa tietovarastoprojektin osa-aluetta. Näin saadaan selkeä kuva siitä missä osa-alueissa suunnittelutyössä on onnistuttu, epäonnistuttu ja mihin on vielä mahdollista jatkokehityksessä vaikuttaa.

6.1 Tietovaraston tyyppi

Kuten aiemmin mainittu, tietovaraston tyypiksi valittiin perinteinen relaatiopohjainen tietovarasto. Yksi syy tähän oli se, että relaatiotietokanta on sopiva kompromissi laajoihin käyttötarpeisiin ja asiakasorganisaatioiden osaamiseen. Tähtimallilla toteutettu tietovarasto sopii paremmin raskaampaan analysointiin ja data-analyytikoiden työvälineille, jotka eivät olleet tämän tietovaraston ensivaiheen vaatimuksissa. BI-alueen toteutus jätettiin kuitenkin vielä projektin alkusuunnittelussa avoimeksi, jolloin oli mahdollista toteuttaa osa koostetusta datasta tähtimallin mukaan. Yrityksen henkilöstön kokemus relaatiomallisista perinteisistä tietovarastoista myös auttoi tiedostomaan valinnan heikkoudet ja puutteet ja varautua niihin ennalta.

Jälkikäteen ajateltuna esimerkiksi jokin helpompi tapa historioida perusdataa olisi voinut olla tietyissä tietovaraston data-alueissa hyödyllinen. Nyt historioiva data tallennetaan vain BI-alueille aggregoituna datana ja itse Core-alueen dataa ei historioida lainkaan. Tulevaisuudessa jos Core-datan historioimiselle tulisi tarvetta olisi se huomattavasti työläämpi toteuttaa nykyistä mallia käyttäen.

6.2 Datamart

Tietovaraston tietomallissa ei ole varsinaisesti huomioitu erilaisia data-alueita, joita erityyppiset loppukäyttäjät saattaisivat omissa liiketoimintayksiköissään käyttää, sillä asiakkaan datamassa ei ole laadultaan huomattavan laveaa, ja sen sisältämä tieto linkittyy suurelta osin samaan jäsenyysdataan. Tietomallissa eri alueiden data on kuitenkin jaoteltu muutamaan eri skeemaan:

- Education: jäsenten koulutustiedot ja yleinen tieto oppilaitoksista.
- Organization: Yrityksen sisäinen hierarkia.
- Member: Varsinainen jäsendata.

Varsinainen käyttötarkoituskohtainen jaottelu tapahtuu vasta datan siirtyessä Core-alueelta BI-alueelle. BI-alueen taulut ja näkymät on rakennettu tiettyjä käyttäjätarpeita ja yksiköitä varten ja edustavat tässä tapauksessa datamart-tyylistä jaottelua. Tämänhetkisellä datamäärällä ja sen luonteella olisi ylitseampuvaa lähteä rakentamaan pienempiä datamart kantoja palvelemaan erityistarpeita ja itse tietovaraston BI-näkymät ajavat asiansa tarpeeksi hyvin. Tulevaisuudessa datalähteiden ja määrän lisääntyessä pienempien datamart-kantojen perustaminen esimerkiksi jäsenmaksudatalle voi olla harkinnanvarainen asia.

6.3 Tiedon laatu

Tietovaraston datan laadukkuuden ja tiedon eheyden varmistamiseksi projektin alussa tehtiin syvä kartoitus asiakkaan päädatalähteen tietomallista, jonka pohjalta suunniteltiin tietovaraston kannan rakenne. Koska tässä projektissa operatiivisia lähdejärjestelmiä oli alkulähtökohdassa vain yksi, oli työ sinänsä helppo tehdä ja monet tietovaraston tauluista peilasivat lähdejärjestelmän taulurakennetta sellaisenaan. Tietovarastossa kuitenkin pyrittiin siirtämään näistä tauluista vain tieto, joka oli oikeasti raportoinnille tärkeää. Usein taulu, jossa saattoi olla lähdejärjestelmässä lähes 30 saraketta, typistyikin tietovarastossa 5 sarakkeen tauluksi. Joskus sarakkeita putosi pois myös siksi että niiden sisältämä data ei ollut tarpeeksi hyvin ylläpidettyä tai eheää että sitä voisi tietovarastossa hyödyntää. Asiakkaan oman IT-osaston työntekijä loi tietovaraston käyttämät SQL-hakuskriptit sen pohjalta mitä oli palavereissa datatarpeista sovittu. Kehitystyö oli kuitenkin iteratiivista ja uusia tarpeita esimerkiksi tietyille taulujen kentille ilmeni projektin edetessä. Osaa näistä tarpeista pystyttiin kartoittamaan asiakkaan vanhojen Excel-raporttien pohjalta, mutta osassa tapauksissa uusia näkökulmia syntyi tietovaraston kehitysvaiheessa.

Monissa tietovarastoinnin perusteoksissa mainittu skenaario, jossa tietovarasto paljastaa yllättäviä heikkouksia ja huonoja toimintatapoja lähdejärjestelmissä toteutui myös projektin aikana muutamaan otteeseen. Tietovaraston dataan tehdyt testikyselyt esimerkiksi paljastivat, että joissain liiketoiminnalle kriittisissä tauluissa oli esimerkiksi päivämääräkenttien kanssa ongelmia, jotka olisivat voineet vääristää raportoinnin lukuja. Kun nämä dataongelmat oli tietovaraston päässä todennettu ja dokumentoitu, pystyi asiakas tekemään datalle korjaustoimenpiteitä jo lähdejärjestelmän päässä. Näin siis tietovarasto toi liiketoimintahyötyjä asiakkaalle ennen varsinaista valmistumistaan.

Metatiedon osalta tietovaraston rakenne suunniteltiin niin että jokainen tietovaraston rivi sisälsi itsessään kenttiä, joissa oli kriittistä tietoa tietovaraston toiminnasta. Metatietokentät sisälsivät tietoa seuraavista asioista:

- Mistä järjestelmästä data oli tietovarastoon haettu
- Minkä organisaation dataa rivi edusti
- Mikä oli datan senhetkinen tila. Oliko se valmista käytettäväksi vai tarvitsiko se täydennystä jostain toisesta tietolähteestä.
- Luonti ja muokkausaika tietovarastossa.
- Luonti ja muokkausaika alkuperäisjärjestelmässä.
- Alkuperäisen järjestelmän tunniste, jolla pystyttiin jäljittämään alkuperäistietolähteen data.

Näistä metakentistä raportteja ajatellen ehkä tärkein oli organisaatiota edustava metakenttä, joka kertoi, oliko data haettu asiakkaan päätietokannasta vai jostakin sisarjärjestöjen kannasta. Tämän metakentän avulla raportointia voitiin rajata vain emojärjestön tai sisarjärjestön tasolle. Teknisten metatietojen olemassaolo huomattiin projektin aikana hyvin tärkeäksi seikaksi, sillä niiden avulla pystyttiin jäljittämään ongelmia datan kanssa. Esimerkiksi kun tietyt hakulauseet tuottivat tuloksia, jotka eivät vaikuttaneet täysin paikkansapitäviltä oli helppo ottaa metakenttien kautta alkuperäisen järjestelmän viittaustiedot ja ajaa sama kysely operatiivisessa järjestelmässä ja verrata lopputuloksia toisiinsa. Tällä tavoin oli mahdollista varmentaa, oliko ongelma alkuperäisen datan syytä vai oliko tietovaraston ETL-prosessit konfiguroitu jotenkin väärin.

Jokainen ETL-ajo generoi itselleen uniikin ID-numeron, joka myös kirjattiin datarivien metatietokenttiin. Tällä tavalla pystyttiin lokitaulujen avulla selvittämään tarkemmin, milloin mikäkin ETL-prosessi muokkasi tiettyä tietovaraston datariviä. Näin pystyttiin esimerkiksi hakemaan taulukohtaisesti tietyn yön ETL-ajojen muokkaamat rivit taulukohtaisesti ja tarkastelemaan minkälaisia muutoksia öinen ajo oli riveihin aiheuttanut.

6.4 Loppukäyttäjät

Jotta tietovarasto vastaisi mahdollisimman laajalla skaalalla asiakkaan tarpeisiin, lähdettiin suunnitteluprosessia tekemään käyttäjien tarvepohjalta. Ensimmäisenä referenssinä pidettiin asiakkaan nykyistä raportointia, joka pyrittiin ensimmäisenä modernisoimaan Excel-tiedostosta visuaalisemmaksi esitykseksi. Vanhojen raporttipohjien perusteella vedettiin ensimmäiset johtopäätökset siitä mikä oli ensisijainen tiedon tarve ja mistä kulmalta tietovaraston rakennetta lähdettiin lähestymään. Tässä vaiheessa siis keskityttiin vastaamaan jo olemassa olevaan tarpeeseen, joka oli tiedostettu. Tämän pohjalta luotiin suunnitelmat ensimmäisistä BI-näkymistä, joilla pystyttiin hakemaan data näihin jo ennalta määritettyihin raportointitarpeisiin.

Seuraavana askeleena oli tarkoitus kartoittaa raportoinnin ja datankäytön laajentamista myös sellaisiin osa-alueisiin, joista ennen raportteja ei ollut tehty. Asiakas hoiti tämän itsenäisesti järjestämällä erilaisia työpajoja eri liiketoiminta-alueiden kesken ja toteuttamalla kyselyitä työntekijöilleen liittyen dataan ja raportointitarpeisiin. Näiden kyselyiden, haastatteluiden ja työpajojen tulokset toimitettiin meille hyvin dokumentoituina ja niiden perusteella aloimme suunnitella tietovaraston eri osa-alueiden laajentamista ja raportointitarpeiden kartoitusta. Näiden dokumenttien pohjalta nousikin paljon ajatuksia dataan liittyen ja saimme mielestäni kattavan kuvan loppukäyttäjien tarpeista. Joukossa oli myös vaatimuksia, jotka olisivat olleet hyvin työläitä toteuttaa nykyisillä resursseilla ja puhtaasti mahdottomiakin pyyntöjä mutta noin 80 % oli toteutettavissa tavalla tai toisella. Ilmeni myös tarpeita integroida mukaan ulkoisia lisäjärjestelmiä ja yhdistää niiden dataa tietovarastossa olevaan dataan, nämä

lisäjärjestelmät päätettiin kuitenkin toteuttaa vasta toisessa vaiheessa sen jälkeen, kun tietovaraston perusdata oli saatu kuntoon ja perusraportointi toimimaan.

Kyselyiden ja työpajojen perusteella paljastui monen eri tasoista loppukäyttäjää. Oli paljon niitä, jotka halusivat vain tietyn raportin valmiina käteen tietyillä rajausvaihtoehdoilla, mutta mukana oli myös sellaisia, jotka halusivat päästä tutkimaan dataa myös hieman pintaa syvemmältä. Oli myös havaittavissa teknisesti harjaantuneempia, jotka olisivat jo aiemmin halunneet tehdä ad hoc -analyysia operatiivisen järjestelmän dataan, mutta heillä ei vain ollut siihen realistista mahdollisuutta.

Näiden käyttäjätarpeiden pohjalta päätös oli luoda BI-näkymiä monella eri tasolla. Osa näkymistä palvelisi suoraan vain yhtä valmisraporttia, jonka käyttäjä saisi päivitettynä käyttöönsä. Toisella tasolla taas olisi rajatumpia näkymiä, joissa olisi dataa laajemmalla mittakaavalla, jolla käyttäjä pystyisi luomaan omia raportteja tietyn data-alueen rajoissa. Kolmantena tasona olisi näkymät, jotka palauttaisivat lähestulkoon tietovaraston Core-alueen tasoista dataa siivottuna turhista metatiedoista. Tämä näkymä olisi varattu niille, jotka todella tietävät mitä tekevät ja haluavat, mutta joilla ei ole suoraa osaamista tehdä SQL-hakuja tietovaraston kantaan sellaisenaan.

6.5 Elinkaari

Projektin tietovaraston elinkaaren suunnittelussa tärkeimpänä lähtökohtana oli se miten uusien tietolähteiden lisääminen onnistuisi tulevaisuudessa. Asiakkaan päätietolähteen lisäksi oli tarkoitus, että tulevaisuudessa tietovarastoon liitettäisiin dataa sisäisistä ja ulkoisista vapaan tiedon tietolähteistä. Oli siis syytä suunnitella tietovaraston rakenne sillä tavalla, että uusien datalähteiden käyttöönotto olisi ollut mahdollisimman vaivatonta ja nopeaa.

Tällä hetkellä tietovaraston toiminta on hyvin riippuvainen päälähdejärjestelmästä ja koska pohjana toimii relaatiotietokanta vaikuttavat lähdejärjestelmän muutokset paljon myös tietovaraston toimintaperiaatteeseen. Jos lähdejärjestelmään luodaan esimerkiksi uusi kenttä, ei se suoraan vaikuta tietovaraston toimintaperiaatteeseen koska lähdejärjestelmän pään Stored Procedure-hakulausekkeet hakevat vain määritellyt kentät lähdejärjestelmän taulusta eikä tämä sinänsä aiheuta ongelmia nykyiselle ETL-prosessille. Jos tämä uusi kenttä kuitenkin haluttaisiin lisätä tietovarastoon, jouduttaisiin muutoksia tekemään niin tietovaraston tietomalliin kuin ETL-prosesseihin kooditasolla. Myös jos koko lähdejärjestelmä päädyttäisiin vaihtamaan kokoaan uuteen, saattaisi se vaatia tietovaraston ETL-prosesseihin suuria muutoksia.

Datan määrän suhteen pilvialustalla toimiva tietokanta vastaa tällä hetkellä hyvin sille asetettuja tavoitteita. Tässäkin tilanteessa, jos tallennustila alkaa käymään vähiin voidaan tietokannan kokoa saumattomasti kasvattaa, mutta tallennustilan kasvaessa myös kulut kasvavat sen mukaisesti. Tietyt tietovaraston taulut kasvavat tasaista vauhtia joka yö ja historioivat BI-

näkymät keräväät aggregoitua dataa omiin tauluihinsa, joten tallennuskapasiteetin tarve kyllä kasvaa koko ajan mutta ei mitenkään räjähdysmäisesti. Tästä syystä ei datan säilytykselle ei ole asetettu mitään takarajaa poistolle tai siirtämiselle johonkin muuhun tallennustilaan. Suurin ongelma kasvavan datamäärän kanssa todennäköisesti tulee tauluissa, jotka sisältävät satoja tuhansia rivejä, sillä on jo nyt havaittavissa, että ETL-prosessien käyttämä ’Delta detection’ – hakulauseke hidastuu sitä mukaa mitä isompia tauluja se joutuu käsittelemään, joka puolestaan hidastaa koko tietovaraston päivitysajoja.

Elinkaaren suurimpia muutostarpeita tulevat osaksi olemaan myös varmaan uudet tavat hakea dataa sitä mukaa kun nykyiset työntekijät pääsevät sinuiksi tietovaraston kanssa. Kun he huomaavat miten vaivatonta tietyn datan saaminen varastosta on, alkavat he haluta uusia näkymiä, joita hyödyntää liiketoiminnan suunnittelussa. Uskon että nykyiselle mallilla tällaiselle ajattelulle on valettu hyvä kivajalka, josta kehitystä on helppo jatkaa.

6.6 Teho ja kustannukset

Kuten jo aiemmin mainittiin moni tietovaraston tehokkuuden ongelmakohta, on helposti ratkaistavissa taustalla pyörivien pilviresurssien tehon nostolla, joka kulkee käsi kädessä tietovaraston kokonaiskustannusten kanssa. Tässä projektissa suurimmat tietovaraston käyttökulut koostuvat ETL-ajoista ja tiedon säilömisestä. ETL-ajojen kanssa emme projektissa törmänneet suurempiin ongelmiin tehon kanssa, ajoja ohjaavat Azure Functions -sovellukset suoriutuivat tehtävästään hyvin jo melkeinpä matalimmalle tasolle asetetuilla resursseilla. Yleisin ongelma ei ollut itse ETL-prosessissa vaan taustalla olevassa tietokannassa.

Alun perin pelkäsimme, että siirto asiakkaan serveriltä Azureen tulisi olemaan prosessin suurin pullonkaula, mutta loppujen lopuksi siirtonopeus ja kustannukset asiakkaan paikallispalvelimen ja Azuren välillä eivät aiheuttaneet missään vaiheessa minkäänlaisia ongelmia. Lukeminen asiakkaan päälähdejärjestelmästä oli salamannopeaa, todennäköisesti johtuen siitä, että operatiivisen järjestelmän taustakanta oli mitoitettu kovaa päivittäistä käyttöä varten. Suurimmaksi ongelmaksi alussa osoittautuikin Azuren SQL Server tietokanta.

Valitsimme tietokannan tehotasoksi Standard 2 jonka uskoimme riittävän hyvin tietovaraston tarpeisiin. Tämä taso toimikin hyvin n. sadantuhannen rivin tauluissa mutta taulut, joissa rivimäärä kasvoi miljooniin, aiheuttivat palvelimeen niin suurta kuormaa että monesti öiset ETL-ajot kestivät liian pitkään tai epäonnistuivat täysin. Standard 2 tasosta seuraava Standard 3 taso tuplaa palvelimen prosessoritehon ja tärkeimpänä muistin, mutta on myös tuplasti kalliimpi edelliseen tasoon verrattuna. S3 tasolle vaihto poisti kaikki kirjoitusongelmat ETL-prosesseista ja öiset ajot tapahtuivat ennätysajassa, vaikka kirjoitus- ja päivitysoperaatioita oli suuria määriä. Tämä kuitenkin herätti kysymyksen, mille tasolle saakka S3 palvelin tulee kestämään kasvavia kuormia. Jos datamäärät vuosien saatossa kasvavat, tuleeko jossain vaiheessa tarve nostaa palvelimen tehotasoa taas yhtä tasoa ylemmäs ja samalla tuplata

tiedonvarastoinnin kustannukset. Tähän olisi mahdollista tietenkin varautua generoimalla tauluihin testidataa ja tutkia miten hyvin ETL-prosessit suoriutuvat esimerkiksi 10 miljoonasta rivistä. Näin voitaisiin tehdä ainakin jonkinlainen ennuste siitä, milloin serverin tehoja olisi syytä nostaa ja näin varautua käyttökustannusten kasvuun.

Standard

	DTUS	INCLUDED STORAGE	MAX STORAGE	PRICE FOR DTUS AND INCLUDED STORAGE [1]
S0	10	250 GB	250 GB	~€12.4123/month
S1	20	250 GB	250 GB	~€24.8219/month
S2	50	250 GB	250 GB	~€62.0740/month
S3	100	250 GB	1 TB	~€124.1145/month
S4	200	250 GB	1 TB	~€248.2444/month
S6	400	250 GB	1 TB	~€496.4887/month
S7	800	250 GB	1 TB	~€992.9774/month
S9	1,600	250 GB	1 TB	~€1,985.9547/month
S12	3,000	250 GB	1 TB	~€3,723.6650/month

Kuvio 12: Azure SQL Server hinnoittelu (Microsoft 2020)

Tietenkin vaihtoehtona palvelimen tehon nostolle on hakujen ja tietokannan virittäminen esimerkiksi indekseillä. Tämän otimmekin jo huomioon tietovarastoa rakentaessa ja pyrimme luomaan indeksejä kaikille niille tauluille ja kentille, jotka liittyivät ETL-prosessien toimintaan. Indeksien toimintaperiaate kuitenkin aiheutti tilanteita, joissa datan lukemisessa saavutettu hyöty menetettiin kirjoitusnopeuden heikkenemisenä, jolloin hyöty jäi hyvin vähäiseksi. ETL-prosessien tapauksessa kuitenkin päädyimme siihen, että loppukäyttäjille lukunopeus on tärkeämpää kuin ETL-prosessin kirjoitusnopeus. ETL-prosessien hidastuminen kirjoitusnopeuden takia ei sinänsä ollut ongelmia koska prosessit suoriutuivat jo nyt tehtävistään hyvin nopeassa ajassa.

Kustannusten puolesta on tietenkin mahdollista, että suunnittelussa ja kehityksessä on tehty jotain sellaisia ratkaisuja, jotka vaikuttavat tuleviin kehityskustannuksiin. Esimerkiksi aiemmin mainituista lähdejärjestelmien vaihdoista tai uusista tietolähteistä voi syntyä ongelmatilanteita, jotka vaativat suuria kooditason muutoksia tietovarastoon. Tällaisia on tällä hetkellä vielä vaikea havainnoida ja niihin varautua, joten niiden kustannukset ovat vielä tietämättömissä.

7 Tulokset ja johtopäätökset

Näitä hyviä käytänteitä peilatessa näyttää hyvin paljon siltä, että useassa kategoriassa tietovarastoprojekti onnistui myötäilemään käytänteitä yllättävän hyvin. On kuitenkin otettava huomioon, että vaikka tietovarasto vastaa nykyisiä asiakkaan tarpeita, näin ei välttämättä ole tulevaisuudessa. Tulevaisuudessa eteen tulevat vaatimukset saattavat vielä paljastaa suunnittelussa ja toteutuksessa puutteita, joita ei vielä tässä projektin vaiheessa osattu ajatella. Esimerkiksi historioidun datan puute Core-datassa saattaa muodostua tulevaisuudessa ongelmaksi, jos halutaan tutkia tietoa, joka on datamuutosten takia kadonnut operatiivisista järjestelmistä, sillä tällä hetkellä tämä data katoaa myös tietovarastosta. Poikkeuksena ovat rivit, jotka poistetaan lähdejärjestelmässä, sillä tämän tiedon tietovarasto vain mitätöi. Jotta tähän voisi varautua paremmin, olisi BI-alueen historioivat taulut suunniteltava mahdollisimman laajalla skaalalla ja monesta näkökulmasta, että ne vastaisivat niitä tarpeita, joita tulevaisuuden käyttäjillä voisi dataa kohtaan olla.

Loppukäyttäjiä ajatellen tietovaraston toteutusta voidaan pitää onnistuneena, sillä projektia rakennettiin koko ajan asiakkaan tarpeita kunnioittaen. Asiakkaan teettämät kyselyt henkilöstölleen valaisivat sitä tarvetta mitä varten tietovarastoa alettiin alun perin rakentamaan. Varsinkin BI-näkymien suunnittelu ilman näitä tietoa olisi todennäköisesti johtanut kehittäjän omien johtopäätösten seurauksena vääränlaiseen näkökantaan asiakkaan dataan. Projekti on kuitenkin siinä vaiheessa, että yksikään loppukäyttäjä ei ole varsinaisesti päässyt dataa vielä hyödyntämään, joten totuus siitä vastaako lopullinen tuotos tarvetta oikeasti jää nähtäväksi tulevaisuudessa.

Tehojen ja kustannusten puolesta tietovarasto on tällä hetkellä hyvinkin kustannustehokas, kuukausikulujen jäädessä mataliksi. Mutta tässäkin on otettava huomioon datamäärien ja käyttäjien lisääntyminen. Itse ETL-prosessit suoriutuvat tällä hetkellä datan siirrosta ja muokkauksesta ilman suurempia viiveitä tai ongelmia, mutta datamäärän ja lähteiden lisääntyessä voi suorituskykyongelmia ilmetä. Tällöin jos mikään viritystoiminpide ei asiaan auta on ainoana vaihtoehtona nostaa taustalla olevien resurssien teholuokituksia mikä nostaa koko tietovaraston kuukausittaisia käyttökustannuksia huomattavasti. On siis paljon kiinni siitä, minkälaisia käyttöskenaarioita tietovarastolle tulevaisuudessa keksitään ja miten sitä vuosien saatossa laajennetaan.

Tietovaraston elinkaaren aikana siihen on todennäköisesti tehtävä myös paljon erinäisiä huolto ja päivitystoimenpiteitä, jotka myös vaikuttavat tulevaisuuden kustannuksiin. Suurimpia tarpeita aiheuttavat todennäköisesti asiakkaan operatiivisen järjestelmän muutokset, jotka vaikuttavat datan laatuun ja muotoon. Tällaisiin asioihin tietovarastossa on varauduttu lähinnä niin, että datan alkupää on jätetty asiakkaan itsensä ylläpidettäväksi. Tällä tavalla tietovaraston toiminta voidaan taata, vaikka datamuutoksia lähdejärjestelmässä

tapahtuisikin. Mutta jos nämä muutokset halutaan osaksi tietovaraston dataa, vaatii prosessi huomattavasti sovelluskehitystyötä, joita asiakas ei itse pysty toteuttamaan.

Yleisesti ottaen tässä vaiheessa projektia monet käsitellyt hyvä käytänteet näyttävän päällepäin toteutuneen mutta kauaskantoisemmat vaikutukset siintävät jossain tulevaisuudessa. Mutta uskon että hyvä pohjatyö tietovaraston toiminnalle on tehty ja toteutus tehty sillä tavalla, että tässä vaiheessa tehtyjä vääriä valintoja on vielä mahdollista korjata.

Tutkimuksen tulokset on listattu tiivistetysti oheiseen taulukkoon (Taulukko 1):

Osio	Onnistui	Ei vielä huomioitu	Kehitysehdotukset
Tietovaraston tyyppi	Tarpeeksi laaja-alainen, tarjoaa tietoa kaikille loppukäyttäjille.	Kaikki tulevat lisätietolähteet ja muut muuttuvat vaatimukset, Core-datan historiointi.	Tietyn datan historiointi, jos se on luonteeltaan sellaista, joka katoaa operatiivisesta järjestelmästä tiedon päivittyessä.
Tiedon laatu	Turha tieto karsittu pois alkuperäisestä datasta ennen tietovarastoon siirtoa, metatiedot ja jäljitettävyys kunnossa.	Taustajärjestelmissä tapahtuviin muutoksiin varautuminen, uusien tietolähteiden yhteensopivuus	Kehittää ETL-prosesseja niin että ne prosessoivat dataa tarkemmin ennen tietovarastoon siirtoa ja normalisoivat sen muotoon, joka mukautuu mahdollisten uusien tietolähteiden tarpeisiin.
Loppukäyttäjät	Nykyisten loppukäyttäjien tarpeita kartoitettu kyselyillä ja workshopeilla.	Tulevaisuuden loppukäyttäjien mahdollisten tarpeiden huomioiminen.	Tutkia syvemmin millä tavalla dataa yleensä hyödynnetään ja muokata BI-tauluja ja näkymiä vastaamaan näitä mahdollisia tarpeita.
Elinkaari	Pilviteknologia päälle rakennettu ratkaisu, jonka resursseja helppo skaalata tulevaisuuden tarpeiden mukaan.	Datamäärien kasvaminen ja uudet käyttötarkoitukset. Uusien kriittisten datalähteiden liittäminen tietovarastoon.	Katsaus seuraavalle viidelle vuodelle, miten liiketoiminta uskoo tietovaraston käytön ja tarpeiden kehittyvän.
Teho ja kustannukset	Tietokanta indeksoitu ja pilviteknologia mahdollistaa tehon skaalauksen tarvittaessa.	Datan määrän kasvaessa tietokannan tehotarpeen kasvu voi kasvattaa tietovaraston kustannuksia huomattavasti.	Tulevaisuuden datamäärien kartoitus ja tarkastus sille mikä tiedosta on oikeasti tarpeellista siirtää tietovarastoon. Laajennuskulujen huomiointi asiakkaan vuosibudjetissa.

Taulukko 1: Tutkimuksen tulokset

Lähteet

Painetut

Coronel, C. & Morris, Steven. 2017. Database Systems – Design, Implementation and Management. Boston: Cegage Learning.

Hovi, A., Hervonen, H. & Koistinen H. 2009. Tietovarastot ja Business Intelligence. Jyväskylä: Docendo.

Hovi, A., Huotari, J. & Lahdenmäki, T.2005. Tietokantojen suunnittelu & indeksointi. Jyväskylä: Docendo.

Hovi, A., Ylinen, J. & Koistinen, H. 2001. Tietovarastot liiketoiminnan tukena. Helsinki: Talentum Media

Krisnan, K. 2013. Data warehousing in the age of big data. Waltham: Morgan Kauffman.

Linsteedt, D. & Olschimmke, M. 2016. Building a Scalable Data Warehouse with Data Vault 2.0 .Waltham, MA: Morgan Kaufmann.

Mohanty, S., Jagadeesh, M. & Srivatsa, H. 2013. Big Data Imperatives – Enterprise Big Data Warehouse, BI Implementations and Analytics. New York: Apress.

Ojasalo, K., Moilanen, T. & Ritalahti, J. 2015. Kehittämistyön menetelmät – Uudenlaista osaamista liiketoimintaan. Helsinki: Sanoma Pro.

Ponniah, P. 2001. Data warehousing fundamentals a comprehensive guide for it professionals. New York: Wiley.

Salo, I. 2014. Big Data & Pilvipalvelut. Jyväskylä: Docendo.

Salo, I. 2013. Big Data – Tiedon vallankumous. Jyväskylä: Docendo.

Silvers, F. 2012. Data Warehouse Designs : Achieving ROI with Market Basket Analysis and Time Variance. Boca Raton: CRC Press.

Törmänen, A. 2017. Johdanto Tietovarastointiin. Törmänen.

Törmänen, A. 1999. Tietovarastointi – strategiasta toteutukseen. Helsinki: Suomen Atk-kustannus Oy.

Väre, T. 2019. Master Data. Liettua: Alma Talent.

Sähköiset

Aws.Amazon.Com. 2020. Data Warehouse Concepts. Viitattu 26.9.2020. https://aws.amazon.com/data-warehouse/

DWH Wiki. 2020. User Types. Viitattu 13.10.2020. http://en.dwhwiki.info/concepts/user-types

Educba.com. 2020. Data Warehouse Architecture. Viitattu 16.10.2020. https://www.educba.com/data-warehouse-architecture/

Haddou, M. 2020. Advantage and disadvantages of a data mart. Viitattu 26.9.2020 http://mbenhaddou.com/2020/01/16/advantages-and-disadvantages-of-a-data-mart/

Azure.Microsoft.com. 2020. Azure SQL Database Pricing. Viitattu 29.11.2020 https://azure.microsoft.com/en-us/pricing/details/sql-database/single/

OpenText, 2014. Understanding 4 Types of data users. Viitattu 22.10.2020. https://blogs.opentext.com/understanding-4-types-of-data-users/

Thareja, R. 2014. Managing traceability in data warehouse development projects. Viitattu 26.9.2020 http://www.ijcta.com/documents/volumes/vol5issue3/ijcta2014050332.pdf

Kuviot

Taulukot