Niki Halm

Big Data -arkkitehtuurit julkipilvessä

Tietotekniikan pro gradu -tutkielma

1. heinäkuuta 2019

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Niki Halm

Yhteystiedot: halmnik1@gmail.com

Ohjaajat: Tommi Kärkkäinen & Toni Taipalus

Työn nimi: Big Data -arkkitehtuurit julkipilvessä

Title in English: Big Data architectures in public cloud

Työ: Pro gradu -tutkielma

Opintosuunta: Ohjelmistotekniikka

Sivumäärä: 59+0

Tiivistelmä: Tämä tutkielma keskittyy massadata-arkkitehtuureihin ja niiden toteuttamiseen julkipilvessä. Tutkielmassa käsitellään teoreettisena taustana massadata, pilvipalvelut ja massadata-arkkitehtuurit.

Tutkielman lopputuloksena on tutkia tapaustutkimuksen pohjalta Lambda- ja Kappa-arkkitehtuureja julkipilvessä ja vertailla niiden kustannuksia teoreettisella tasolla.

Avainsanat: Massadata, pilvipalvelut, massadata-arkkitehtuurit

Abstract: This thesis focuses on Big Data architectures and how to implement them to public cloud. Thesis cover Big Data, Cloud Services and Big Data architectures as theoretical background and focuses to case research Lambda- and Kappa-architectures in public cloud, based on comparing expenses at theoretical level.

Keywords: Big Data, Cloud Service, Big Data architectures

Termiluettelo

Big Data	(suomennettuna massadata) on suuri määrä nopeasti lisääntyvää monimuotoista dataa.
Lambda-arkkitehtuuri	on Nathan Marzin kehittämä Big Data -arkkitehtuuri, joka tarjoaa puitteet niin erä- kuin reaaliaikaiselle analytiikalle.
Kappa-arkkitehtuuri	on Jay Krepsin kehittelemä Big Data -arkkitehtuuri, joka on luotu Lambda-arkkitehtuurin pohjalta.
Internet of Things	(suomennettuna esineiden internet) on käsite, jolla tarkoitetaan kaikkia laitteita, jotka ovat yhteydessä internettiin ja tuottavat dataa sinne.
Data Lake	(suomennettuna tietoallas) on järjestelmä tai säilytyspaikka rakenteelliselle ja rakenteettomalle datalle raakamuotoisena.
Serverless	on palveliton palvelu, jolloin esimerkiksi infrastruktuuria ei tarvitse ylläpitää.

Kuviot

Taulukot

Sisältö

1 Johdanto

Yrityksille on liiketoiminnallisesti kannattavaa tuoda data lähelle liiketoiminnan päätöksentekoa. Monet yritykset ovat jo pitkään tehneet datan pohjalta päätöksiä, johon saatu data on tullut esimerkiksi toiminnanohjaus- ja asiakkuudenhallintajärjestelmistä. Tietomäärät ovat kasvaneet ja varsinkin yrityksille esineiden internet (Internet of Things, IoT) on kasvattanut ladattavan datan määrää siten, että voidaan joissakin tapauksissa puhua jo massadatasta (Big Data). Massadata on hieman epämääräinen käsitteenä, koska on vaikeaa määrittää, milloin datasta voidaan puhua massadatana. Esimerkiksi Doug Laney (2011), joka toimii Gartnerin analyytikkona, määritteli jo vuonna 2011 Massadatan perimmäisen rakenteen kolmella V:llä: volyymi (Volume), vauhti (Velocity) ja vaihtuvuus (Variety).

Nykyään dataa tuotetaan noin 2,5 eksatavua (yksi eksatavu on miljoona teratavua) päivässä ja datamäärän kasvu vain kiihtyy (Forbes 2018). Yritykset eivät kykene keräämään kaikkea raakadataa talteen vaan datasta pyritään tuottamaan informaatiota, joka kerätään sitten talteen. Informaatiota kerätessä datamassoja aggregoidaan yhteen, mutta informaatiomassatkin ovat hyvin suuria ja vaativat täten myös resursseja enemmän kuin perinteisemmät ratkaisut. Tällöin käytettävä arkkitehtuuri ja infrastruktuuri tulee suunnitella hyvin. Tällöin myös pilvipalvelut tulevat pohdittavaksi, koska niiden skaalautuvuus ja hinta-prosessointiteho-suhde on huomattavasti kilpailukykyisempi kuin perinteinen konesaliratkaisu.

Pilvipalveluihin siirryttäessä tietoturvallisuudesta huolehtiminen on otettava tarkemmin huomioon, koska data ei enää välttämättä ole säilössä omassa konesalissa, vaan käytettävä konesali saattaa sijaita toisessa maassa tai aivan toisella mantereella. Tällöin varsinkin henkilödatan osalta huomioitavaksi tulee myös GDPR-lainsäädäntö (General Data Protection Regulation), joka voi rajoittaa datansäilöntään tarkoitettuja konesalivaihtoehtoja.

Tutkielma toteutetaan tapaustutkimuksena, jonka kohteena on Lambda- ja Kappa-arkkitehtuuri ja tapauksina eri julkipilven tarjoajat. Tapaustutkimuksen tuloksia verrataan kirjallisuuskatsauksena.

1.1 Tutkimuksen tavoitteet ja rajaus

Tässä tutkielmassa käydään läpi massadata-arkkitehtuureja julkipilvessä. Tutkielman alussa käydään läpi massadatan teoriatausta, jossa selvitetään käsitteen tarkoitusta ja siihen liittyviä muita käsitteitä, kuten esineiden internet ja tietoallas (Data Lake). Teoriaosuus toteutetaan narratiivisena kirjallisuuskatsauksena.

Empiirisessä osiossa käsitellään tapaustutkimuksen keinoin parhaita käytäntöjä massadata-arkkitehtuureissa julkipilven palveluissa. Pilvialustojen tarjoajista on rajattu kolmeen eniten käytössä olevaan palveluntarjoajaan: Amazon Web Service, Microsoft Azure ja Google Cloud Platform. Arkkitehtuurin osalta tarkastelu keskittyy palveluiden kustannuksiin teoreettisella tasolla. Tapaustutkimuksen tuloksien vertailu tehdään kirjallisuuskatsauksena, jossa lähteenä toimii pilvipalveluiden tarjoajien hinnastot ja suorituskykytiedot.

1.2 Tutkimusongelma ja tutkimuskysymykset

Yritysten datamäärien kasvaessa on tärkeää suunnitella sen talteenottotapa ja mihin kaikki kerätty data säilötään. Datamassan kasvaessa myös sen prosessoimiseen vaadittavat resurssit kasvavat ja perinteisillä omilla konesaliratkaisuilla voivat palvelimien hinnat kohota suuriksi, varsinkin mikäli dataa ei ole tarvetta prosessoida ympärivuorokautisesti. Tällöin pilvipalveluiden tarjoamat hyödyt voivat kasvaa suuriksi, mutta tällöinkin tarvittava arkkitehtuuri kannattaa suunnitella hyvin. Tutkielma rakentuu näin tutkimuskysymyksen ympärille:

- Kuinka muodostaa nykyaikainen massadata-arkkitehtuuri pilvialustoilla?

Arkkitehtuuria suunnitellessa tulee ottaa monia asioita huomioon, niin tietoturvallisuuden kuin kustannustenkin osalta. Massadata-arkkitehtuuri noudattaa hyvin tyypillistä tietoarkkitehtuurin kaavaa, jossa data ladataan ja prosessoidaan eri datakerroksien läpi, mutta ei ole olemassa yhtä arkkitehtuuria, joka ratkaisisi kaikki eteen tulevat ongelmat. Tämän lisäksi eri pilvitarjoajilla on hieman toisistaan eroavat palvelut, joiden käyttötavat ja kustannukset rakentuvat erilaisesti. Tämä johtaa seuraavaan tutkimusongelmaan: ”Mitkä ominaisuudet tulee huomioida vertailtaessa pilvipalveluita massadata-arkkitehtuurissa?”

2 Teoriatausta

2.1 Narratiivinen kirjallisuuskatsaus

Kirjallisuuskatsaus on tutkimustekniikka, jonka avulla kootaan tutkimuksien tuloksien luoden perustaa uusille tutkimustuloksille. Salminen (2011) esittää, että kirjallisuuskatsauksen tavoitteena on kehittää olemassa olevaa teoriaa ja rakentaa uutta teoriaa. Sen avulla voidaan myös arvioida teoriaa ja rakentaa kokonaiskuva tietystä aihekokonaisuudesta.

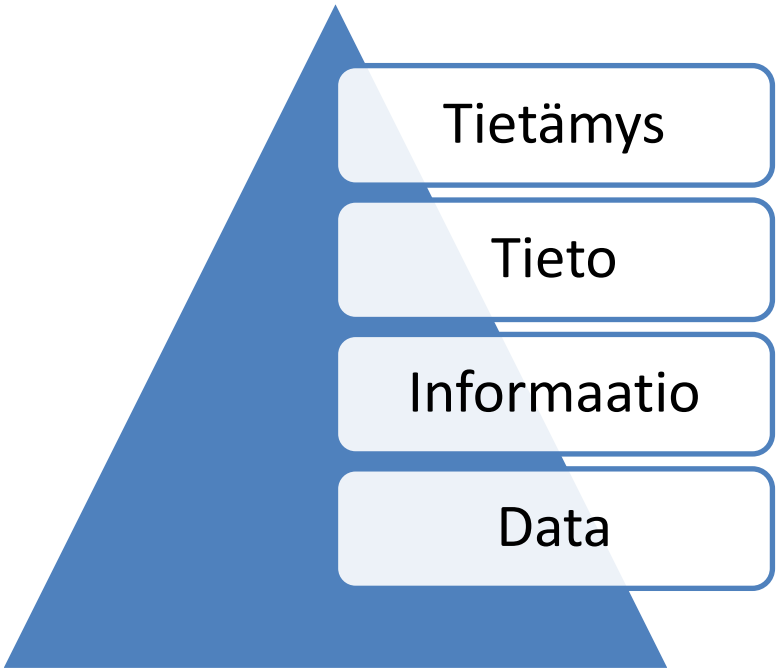
Kirjallisuuskatsauksia on eri tyyppisiä, joista yksi yleisemmin käytetty perustyyppinen tapa on kuvaileva kirjallisuuskatsaus. Se on yleiskatsaus, jossa käytetään laajoja aineistoja ilman rajaamista metodisilla säännöillä. Tutkimuksen aihe pystytään kuitenkin laajasti kuvaamaan ja luokittelemaan sen sisältämiä ominaisuuksia. Myös tutkimuskysymykset ovat väljempiä kuin systemaattisessa katsauksessa.

Kuvaileva kirjallisuuskatsaus jakautuu kahteen tyyppiin: narratiivinen ja integroiva katsaus. Näistä integroivalla on paljon yhtymäkohtia systemaattisen katsauksen kanssa. Narratiivinen katsaus on metodisesti kevyin kirjallisuuskatsauksen muoto, jonka avulla pystytään antamaan laaja kuva käsiteltävästä aiheesta. Narratiivisessa katsauksessa on kolme toteutustapaa: toimituksellinen, kommentoiva ja yleiskatsaus. Näistä kolmesta yleiskatsaus on laajin, kun taas toimituksellinen ja kommentoiva taas ovat huomattavasti suppeampia. Yleisesti puhuttaessa narratiivisesta kirjallisuuskatsauksesta, niin tarkoitetaan juuri yleiskatsausta toteutustapana. Yleiskatsauksen tarkoituksena on tiivistää aiemmin tehtyjä tutkimuksia eikä sen keräämä tutkimusaineisto käy läpi erityisen systemaattista seulontaa. Silti lopputuloksena on mahdollisuus päätyä johtopäätöksiin, jotka ovat kirjallisuuskatsauksen mukaisia.

2.2 Massadata

Datasta käytetään arkikielessä usein synonyymiä tieto, joka ei ole määritelmällisesti sama asia. Data on raaka-aine, josta voidaan saada informaatiota, jonka pohjalta muodostetaan tietoa. Tiedon myötä ymmärrys kasvaa ja lopputuloksena tieto muodostaa tietämystä. Massadata-työkalut painottuvat datan tallentamiseen, siirtämiseen ja muuntamiseen, minkä

pohjalta tämä muunnos datasta informaatioksi on mahdollinen. Informaation muuntaminen tiedoksi on esimerkiksi data-analytiikan prosessi. Näiden neljän suhdetta kutsutaan DIKW-pyramidiksi (Data-Information-Knowledge-Wisdom), joka esitelty kuviossa 1, ja jonka alkuperästä ei ole tarkkaa varmuutta (Wallace, 2007).



Kuvio 1. Data, informaatio, tieto ja tietämys.

Datasta on muodostunut tulevaisuuden öljy, jonka hyödyntäminen on edellytys parempaan menestymiseen (Salo 2014, 8).

2.2.1 Määritelmä

Vuoden 2005 tietämillä massadata käsite tuli tutuksi, mutta se nousi suureksi kiinnostuksen aiheeksi vuonna 2011. Sille, milloin massadata nimitystä alettiin käyttää, on mahdotonta tarkoin määrittää. (Salo 2014, 26)

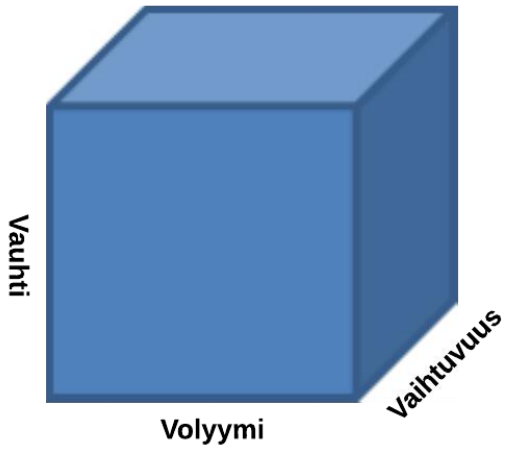
Yksinkertainen määritelmä massadatalle on suuri volyymi dataa, jota ei voida varastoida ja prosessoida käyttäen perinteisiä keinoja. Tämä data voi sisältää arvokasta informaatiota, jonka myötä se pitää prosessoida lyhyellä aikavälillä. Mikäli tätä ongelmaa lähestytään

perinteisin keinoin, niin on mahdotonta suorittaa tämä prosessointi annetussa aikavälissä, koska tallennus- ja prosessointikapasiteetti eivät ole riittäviä tämän tyyppiselle tehtävälle.

Massadataa voidaan tarkastella kahdesta käsitteellisestä näkökulmasta: yleisesti ja teknologiallisesti. Yleisesti tarkastellessa massadata voidaan havainnoida datamäärien kasvuna, tietorakenteiden monipuolistuminen ja datan muodostumisen jatkuva kiihtymisenä. Teknologialliselta kannalta se on nimitys teknologioille ja palveluille, joilla sitä pyritään hallitsemaan (Salo 2014, 8,32). Massadata ei ole siis yksittäinen teknologia, vaan yhdistelmä vanhaa ja uutta teknologiaa, jonka avulla voidaan hallita suuria määriä erityyppistä dataa oikealla nopeudella mahdollistaen esimerkiksi reaaliaikaiset analyysit ja päätöksenteot. massadata-käsite sisällyttää kaiken datan mukaan lukien rakenteellisen ja rakenteeton, mitkä voivat tulla mistä lähteestä tahansa esimerkiksi sähköpostit tai sosiaalinen media (Hurwitz, Nugent, Halper & Kaufman 2013, 16)

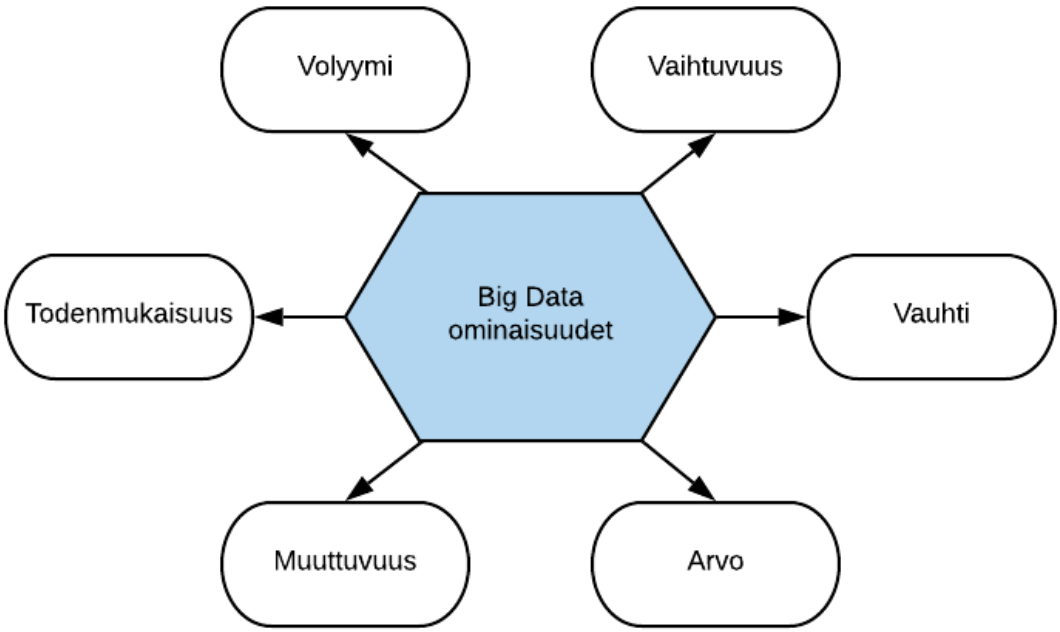
2.2.2 Ominaisuudet

Doug Laneyn (2001) julkaisema raportti käsittelee tulevaisuuden datamäärien ja niiden sisällön vaihtelevuuden kasvua. Raportissa hän käyttää kolmea V-kirjaimella alkanutta sanaa volume, variety ja velocity, jotka voidaan suomentaa myös V-alkuisiksi volyymi, vaihtelevuus ja vauhti. Nämä kolme V-kirjainta vakinaistuivat käytettäväksi vuoden 2001 jälkeen puhuttaessa massadatasta ja mitä paremmin data sopii näihin kolme ulottuvuuteen, jotka esitelty myös kuviossa 2, niin sitä selkeämmin se on massadataa.



Kuvio 2. Massadatan kolme ulottuvuutta (Salo 2014, 28)

Teknologian kehityksen ja tutkimuksien edetessä näiden kolmen massadataa kuvaavan V-kirjaimen lisäksi ollaan saatu kolme V-kirjainta lisää ja näin ollen nykyisellään massadataa voidaan kuvata kuuden V-kirjaimen avulla. Tämä määrä voi myöskin kasvaa vielä tulevaisuudessa. Nämä kolme muuta V-kirjainta ovat veracity, variability ja value, jotka voidaan suomentaa todenmukaisuus, muuttuvuus ja arvo. Kuvio 3 kuvastaa massadatan ominaisuuksia näiden kuuden osalta. (Akhtar 2018, 9).



Kuvio 3. Massadatan kuusi ulottuvuutta. (Akhtar 2018, 9).

Aikaisemmin yrityksen data muodostui vain työntekijöiden muodostamasta datasta. Teknologian käytön kasvaessa dataa generoituu myös yrityksen tuotantokoneista, sosiaalisesta mediasta ja muista internet-lähteistä. Kun puhutaan volyymista massadatan osalta, niin tarkoitetaan datamassaa, jota hyödyntävä järjestelmä ei voi keretä, tallentaa tai prosessoida käyttäen aikaisempia menetelmiä, jossa yksittäinen tietokone tai palvelin vastasi koko datan elinkaaresta. (Akhtar 2018, 10).

Vaihtelevuudella (Variety) tarkoitetaan tietotyyppien monipuolisuutta ja lukematonta määrää lähteitä, joista nämä voivat tulla. Teknologioiden ja eri tyyppisten sovellusten lisääntyessä myös eri tietorakenteet moninaistuvat. Karkeasti nämä tietotyypit voidaan jaotella rakenteelliseen ja rakenteettomaan. Sovellukset usein käyttävät rakenteellista dataa, joka on tallennettu relationaalisiin tietokannanhallintajärjestelmiin, mutta rakenteeton data on myös huomioitava esimerkiksi videoiden, valokuvien ja binäärimuotoisen tiedostosisältöjen osalta. (John & Misra 2017, 21). Vaihtuvuutta näiden mediatiedostojen osalta on myös kuvien ja videoiden pakkaustekniikat, sekä kvantisoinnin aste.

Vauhti (Velocity) kuvastaa millä nopeudella dataa muodostetaan tai kuinka nopeasti dataa tulee säilöttäväksi. Toinen dimensio vauhdista on aikaväli siitä, kun datasta saadaan ymmärrettävää siihen, kun sillä on vielä arvoa. Vanheneeko data menettäen arvonsa vai onko se pysyvästi arvokasta? (Akhtar 2018, 11).

Todenmukaisuus (Veracity) kuvastaa datan epävarmuutta, joka voi johtua huonosta datalaadusta tai häiriöistä datasta. Tällaisia laatuongelmia ovat esimerkiksi tyhjät datakentät (null values) ja sensorilaitteiden tuottamat virhemittaukset. Tällaisessa haasteena on varsinkin tietovirtadatan ja suurella nopeudella tulevan datan siivoaminen saadakseen epävarmuustekijät siitä pois. Näin ollen luottamus datan oikeellisuuteen vaikuttaa sen informaationalliseen arvoon ja päätöksiin, jotka tehdään sen pohjalta. (Akhtar 2018, 15).

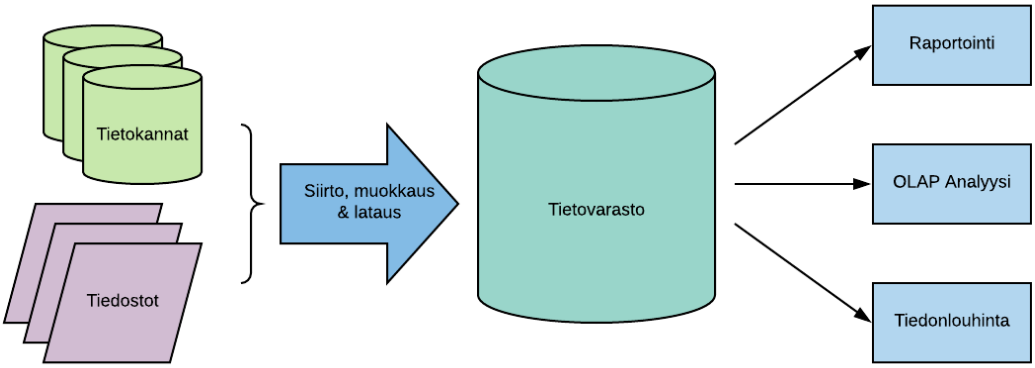
Muuttuvuus (Variablity) kuvastaa datan johdonmukaisuuden tai kaavamaisuuden puuttumista. Se eroaa vaihtelevuudesta siten, että sillä ei tarkoiteta tietorakenteellista muutosta vaan datan tarkoituksen ja ymmärtämisen muuttumista, jolla on suuri vaikutus analyyseihin ja kaavamaisuuden tunnistamiseen. (Akhtar 2018, 15). Toinen esimerkki muuttuvuudesta on concept drift-käsite, jota esintyy varsinkin ennustavassa analytiikassa ja koneoppimisessa, ja jolla tarkoitetaan syötedatan ja tavoitemuuttujan välisen suhteen muuttumista ajan saatossa (Gama, Zliobaite, Bifet, Pechenizkiy & Bouchachia, 2013).

Arvo (Value) on tärkein massadatan ominaisuus, vaikka samaa voidaan sanoa myös pienemmistä datoista. Se kuvastaa datan arvoa siltä osin, että milloin kyseinen data kannattaa varastoida ja käyttää rahaa infrastruktuuriin sen myötä. Yksi näkökulma arvolle on se, että tarvitsee varastoida suuri määrä dataa ennen kuin siitä voidaan saada arvokasta informaatiota irti. Aikaisemmin tällaisen datamassan säilöminen aiheutti suuria kustannuksia, mutta nykyään tallennustila on huomattavasti halvempaa, joka on mahdollistanut massadatan kasvun. (Akhtar 2018, 15).

2.2.3 Tarvittava infrastruktuuri

Perinteiset järjestelmät käyttävät eräajoja, jotka on ajastettu päivittäin, viikoittain tai kuukausittain hakemaan data eri palvelimille tai tietovarastoon. Tällä datalla on skeema ja se kategorisoituu rakenteelliseksi dataksi. Tämä data prosessoidaan ja analysoidaan saaden siitä

lopulta ulos hyödyllistä informaatiota. Tämän kaltainen rakenne on optimoitu raportoinnin ja analytiikan käyttöä varten ja on perusta Business Intelligence (BI) -järjestelmille, jonka arkkitehtuuria kuvataan kuviossa 4. Tämän lähestymistavan ongelma on kuitenkin viive, jolla data saadaan saataville päätöksentekoa varten. Datamäärän kasvaessa myös eri tyyppiset datarakenteiden määrä on kasvussa. Rakenteellisen datan määrä on pienempi kuin ei-rakenteellisen, jonka myötä kaikkea dataa ei voida hyödyntää perinteisiä keinoja käyttäen (Akhtar 2018, 17-18)



Kuvio 4. Business Intelligence -järjestelmäarkkitehtuuri (Akhtar 2018, 17)

Hajautetut järjestelmät ovat yksi osasyy mikä on mahdollistanut massadatan säilömisen ja prosessoinnin. Hajautetut järjestelmät eivät ole uusia teknologiakonsepti vaan se on ollut käytössä noin 50 vuotta. Alussa teknologiaa käytettiin skaalaamaan laskentatehtäviä ja täten ratkaisemaan komplekseja ongelmia ilman kustannusta suurista laskentajärjestelmistä. Hajautettujen järjestelmien myötä pilvilaskenta, virtualisointi ja massadatan -käsittelyt ovat mahdollisia. Hajautetut järjestelmät ovat teknologia, jonka myötä yksittäiset tietokoneet on mahdollista verkostoida yhdeksi ympäristöksi riippumatta tietokoneiden maantieteellisestä sijainnista. Nämä verkostoidun koneet tämän jälkeen työskentelevät yhdessä suorittaakseen työmääriä tai prosesseja. (Hurwitz ym., 2013, 37-38).

Hajautetun järjestelmistä saadaan muitakin hyötyjä kuin pelkkä kustannussäästö. Järjestelmien ollessa hajautettuna voidaan parantaa vikasietoisuutta, koska arkkitehtuuri ei ole enää

riippuvainen yhdestä tietokoneesta. Samoin laitteiston skaalautuvuus on helppoa, koska tähän verkostoon voidaan lisätä lisää tai vähentää tietokoneita tarvittaessa. Hajautettujen järjestelmien myötä suurien datamassojen käsittely on mahdollista niin tallennus- kuin laskentatehojen osalta. (Akhtar 2018, 19-21).

2.3 Pilvipalvelut

Salon (2014) mukaan yksi yleisin määritelmä pilvipalveluille on Yhdysvaltalaisen National Institute of Standards and Technologies (NIST) määritelmä:

> *Cloud Computing on toimintamalli, joka mahdollistaa pääsyn vapaasti konfiguroitaviin ja skaalautuviin tietotekniikkaresursseihin, jotka voidaan ottaa käyttöön tai poistaa käytöstä helposti ja nopeasti.*

Tämän määritelmän lisäksi NIST listaa viisi ominaispiirrettä pilvipalvelulle: itsepalvelullisuus, pääsy palveluihin eri päätelaitteilla, resurssien yhteiskäyttö, nopea joustavuus ja käytön tarkka mittaaminen. Pilvipalvelut eivät ole ainoastaan ulkoisen palveluntarjoajan tuottamia palveluita vaan yritykset voivat yllämainittujen ominaispiirteiden mukaisesti tuottaa pilvipalveluita. Tällöin puhutaan yksityisestä pilvestä ja NIST onkin listannut sen yhdeksi neljästä pilvipalveluiden käyttöönottotavoista: yksityinen pilvi, yhteisöllinen pilvi, julkinen pilvi ja hybridipilvi. Siinä missä yksityisessä pilvessä infrastruktuuri oli yhden organisaation käytössä, niin yhteisöllisessä pilvessä käyttö jakautuu useamman organisaation välille. Julkinen pilvi on taas palveluntarjoajien toimittama, jossa palvelun tarjoaja vastaa taustalla olevista laitteistoista. Hybridipilvi on yhdistelmä näistä aikaisemmin mainituista, missä osa arkkitehtuurista on yksityistä ja osa julkista. (Salo 2014, 93-95).

Pilvipalveluita voidaan tarkastella kategorioittain, joista yleisimmät ovat infrastuktuuri palveluna (Infrastucture as as Service, IaaS), sovellusalusta palveluna (Platform as a Service, PaaS), sovellus palveluna (Software as a Service, SaaS). (Hurwitz ym., 2013, 74).

Infrastruktuuri palveluna on yksi suoraviivaisimmista pilvipalveluista, jonka myötä toimitetaan laskentapalvelujen infrastruktuuri sisältäen laitteisto, verkko, tallennustila ja

datakeskustila vuokrausmallia käyttäen. Näistä palveluista maksetaan käyttömäärän ja -ajan mukaan. (Hurwitz ym., 2013, 74)

Sovellusalusta palveluna on mekanismi, joka yhdistelee infrastruktuuria palveluna ja yhdistelmäjoukkoa väliohjelmistopalveluista, ohjelmistokehityksestä ja käyttöönottopalveluista. Tämän myötä saadaan kehitys ja käyttöönotto tuotua yhteen luoden hallitumpi tapa rakentaa, ottaa käyttöön ja skaalata ohjelmistoja. (Hurwitz ym., 2013, 75)

Sovellus palveluna on bisnesohjelmisto, joka on kehitetty ja ylläpidetty sovellustarjoajan toimesta monitasoisessa mallissa (multitenant model). Monitasoisuus tarkoittaa tilannetta, jossa yksittäinen instanssi ohjelmistosta suoritetaan pilviympäristössä, mutta se palvelee useampaa asiakasorganisaatiota pitäen silti niiden datat erillään. (Hurwitz ym., 2013, 75)

Näiden edellä mainittujen lisäksi on olemassa uusi tietojenkäsittely muoto, palveliton tietojenkäsittely (Serverless computing), joka saa tällä hetkellä jalansijaa pilviympäristöistä. Tähän on syynä se mitä nimikin jo lupailee eli se tarjoaa tietojenkäsittelyä ilman ylläpitoja, rajattomalla joustavuudella ja hyvin pienellä kustannuksella. Tämä on askel edemmäs PaaS-ympäristöistä, jossa sovelluksen komponentit voidaan jakaa pienemmiksi osiksi ja ajaa pelkkinä funktioina pilvessä. Tämän kaltainen joustavuus mahdollistaa kustannusten muodostumisen pelkän käytön osalta ja rajattoman skaalautuvuuden tulevaisuuden varalle. (Kritikos & Skrzypek 2018).

Pilven tarjoamat mahdollisuudet ja palvelut luovat lukemattoman määrän tapoja hyödyntää niitä massadatan tarpeisiin. Massadata tarvitsee toimiakseen kustannustehokkaasti hajautettuja järjestelmiä, joka on pilvialustojen perimmäinen arkkitehtuuri. (Hurwitz ym., 2013, 75)

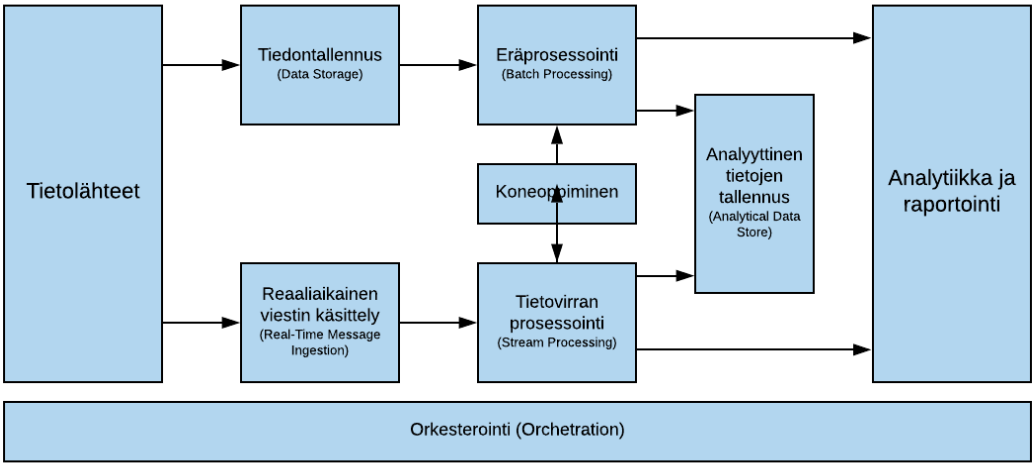
Jokaisella yrityksellä on erilaiset vaatimukset ja lähestymistavat mitä tulee massadatan hyödyntämiseen. Yritykset tekevät analyysit ja soveltuvuustutkimukset ennen suuria muutoksia, varsinkin teknologian puolella. Massadatan hyödyntämiseen on aikaisemmin ollut vain mahdollisuutena luoda ympäristö yrityksen omille palvelimille, mutta nykyään vaihtoehtoisena tapana on julkipilven hyödyntäminen. Moni iso palveluntarjoaja, kuten Microsoft, Google tai Amazon, tarjoaa nykyään suuren määrän valmiita palveluita yrityksen tarpeisiin. Valittaessa yksityisen- ja julkipilven väliltä on hyvä ottaa huomioon muutamia eroavaisuuksia. (Akhtar 2018, 21-22)

Kustannus on tärkeä osatekijä niin pienille kuin suuremmillekin yrityksille ja joskus ainoa päätökseen vaikuttava tekijä. Infrastruktuurin pystyttäminen massadatan käyttöä varten on suuri alkukuluerä, johon tarvitaan tehokkaat palvelimet ja verkkoasetukset. Julkipilven osalta tällaista suurta kuluerää aluksi ei tarvita päästäkseen hyödyntämään massadataa ja käyttöön otettavaa infrastruktuuria voidaan myöhemmässä vaiheessa skaalata helposti myös suuremmaksi ilman suuria kertakustannuksia. Samoin julkipilven palvelimet ja verkkoyhteydet eivät vaadi samanlaista ylläpitotyötä kuin omille palvelimille pystytettävät ratkaisut. (Akhtar 2018, 22)

Toisena suurena tekijänä on turvallisuus. Oman infrastruktuurin rakentaminen luo yrityksille tunteen paremmasta tietoturvasta. Se antaa yrityksille myös hallinnan kenellä on pääsy heidän dataansa, milloin sitä käytetään ja mihin tarkoitukseen. Toisaalta datan sijoittamisella pilveen on omat riskinsä ja suurimmat kysymysmerkit nousevat siitä, että missä dataa säilytetään julkipilven tarjoajan toimesta ja millä tiimeillä on pääsy siihen julkipilven puolelta. Näiden huolien vuoksi julkipilven tarjoajat ovat panostaneet tietoturvaan, että jokainen bittidataa on varmasti säilötty turvallisesti. Tähän apukeinona on käytetty erityyppisiä salausmekanismeja, jolloin varastettu data olisi käyttökelvotonta. Samoin datahävikkiä on pyritty välttämään luoden varmuuskopioita täysin eri palvelinsaleihin. Näillä keinoilla on julkipilven infrastruktuureista pyritty luomaan yhtä turvallisia kuin yritysten omat palvelinkokonaisuudet. (Akhtar 2018, 23)

2.4 Massadata-arkkitehtuurit

Massadata-arkkitehtuurit on suunniteltu hallitsemaan sellaiset datamassat, jotka ovat liian suuria tai kompleksisia perinteisille tietovarastointijärjestelmille. Dataa voi tulla syntyä todella suurella nopeudella, kuten aiemmin tutkimuksessa ollaan jo todettu ja tämän myötä sen lataus, varastointi ja prosessointi vaativat erilaista lähestymistapaa kuin perinteisissä eräajoissa. Samoin massadatasta puhuttaessa voivat eräajojen datamassat olla niin suuria, että niiden lataaminen vaatii rinnakkaisajoa ja hajautettuja prosesseja. Nämä ovat haasteita, jotka pyritään ratkaisemaan massadata-arkkitehtuurilla. (Microsoft Corporation 2018b). Kuviossa 5 on eritelty yleisellä tasolla massadata-arkkitehtuurissa tarvittaviin komponentteihin, mutta on huomioitava, että näitä jokaista ei aina tarvita arkkitehtuurista ratkaisua luotaessa.



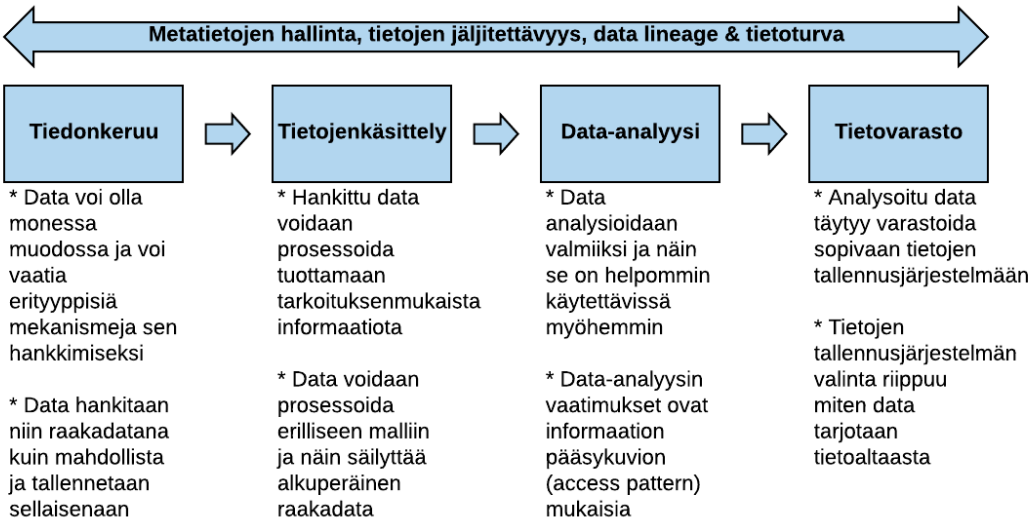
Kuvio 5. Massadata-arkkitehtuuri yleisesti (Microsoft Corporation 2018b)

2.4.1 Datan säilöminen

Datan säilömiseen pilveen on kaksi päävaihtoehtoa: tietoallas tai tietovarasto. Näillä kahdella on omat käyttötarkoituksensa ja ne enemmänkin täydentävät toisiaan kuin olisivat pelkästään täysin vaihtoehtoisia tapoja keskenään. (John & Misra 2017, 44).

2.4.1.1 Tietoallas

James Dixon (2010) esitteli termin tietoallas isompana kokonaisuutena datan säilömiselle verrattuna aikaisempiin datamartti-varastoihin (datamart), jossa dataa säilötään vain sillä hetkellä tarvittavan määrän verran. Tietoallas on konsepti, joka syntyi haasteista datan hallinnan, prosessoinnin ja varastoinnin kanssa. Ajansaatossa yritysten sovellukset ovat keränneet paljon dataa, jota ei välttämättä ole voitu hyödyntää sovellusten välillä ja jonka myötä data siiloutuu. Tietovarastoinnilla ollaan voitu vastata tähän haasteeseen rakenteellisen datan osalta, mutta rakenteeton data on tällöin jäänyt hyödyntämättä. Tietoaltaan avulla voidaan säilöä kaiken tyyppistä dataa ja tarjota se prosessoitavaksi tietoaltaan kautta. (John & Misra 2017, 39). Kuvio 6 kuvastaa tietoaltaan elinkaarta.

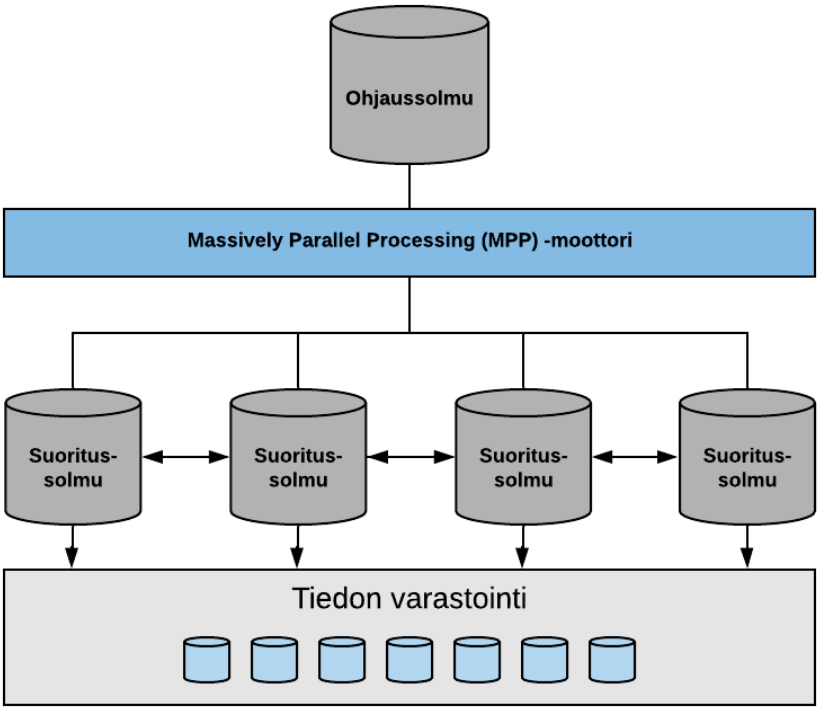


Kuvio 6. Tietoaltaan elinkaari (John & Misra 2017, 42)

2.4.1.2 Tietovarasto

Kirjaimellisesti, jos käsite tietovarasto puretaan auki, niin saadaan luonnollisesti kaksi käsitettä: tieto ja varasto. Tieto on faktaa tai koottua informaatiota johonkin liittyen ja varasto taas paikka tai laitos säilöä tavaroita. Tietovarasto kuvastaa siis hyvinkin sen tarkoitusta eli säilö dataa. Tarkoituksena tiedonsäilömisellä on hyödyntää dataa raportointia ja analyyseja varten, jonka myötä se on arkkitehtuurisesti suunniteltu juuri tätä käyttötarkoitusta varten. (Hammergren & Simon 2009, 12-13)

Massadatan osalta perinteinen tietovarastointiratkaisu ei välttämättä kykene prosessoimaan dataa riittävällä nopeudella. Tähän ongelmaan voidaan hyödyntää Massively Parallel Processing -arkkitehtuuria (MPP), joka perustuu kyselyn jakamiseen useampaan eri suoritussolmuun. Täten kysely voidaan jakaa osiksi ja ajaa nämä eri osat rinnakkain. MPP-arkkitehtuurissa on tiedon suorittaminen eriytetty tiedon varastoimisesta ja näin ollen suoritustasoa voidaan helposti skaalata tarpeen tullen. (Microsoft Corporation 2018a). Kuviossa 7 on esitelty MPP-arkkitehtuuri.



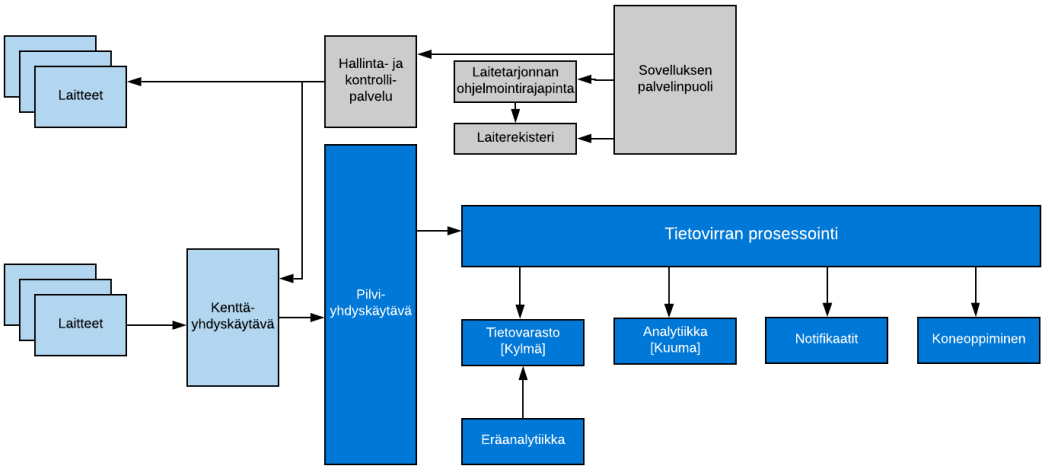
Kuvio 7. MPP-arkkitehtuuri

2.4.2 Tapahtumapohjainen arkkitehtuuri

Tapahtumapohjainen arkkitehtuuri on keskinäinen osa esimerkiksi esineiden internetiä, jossa data tulee sisään ja sen pohjalta tehdään toimenpiteitä automaattisesti. Esineiden internet on käytännössä mikä tahansa laite, joka on kytketty internettiin. Näiden laitteiden määrä kasvaa päivittäin ja samoin myös niistä saatavan datan määrä. Tällaisen datamassan kerääminen vaatii hyvin skaalautuvan ja tehokkaasti dataa käsittelevän arkkitehtuurin. (Microsoft Corporation 2018b).

Laitteet voivat lähettää tapahtumadataa suoraan prosessointijärjestelmälle tai sitten välissä saattaa olla jokin toinen laite, joka toimii yhdyskäytävänä. Tällainen asetelma voi olla esimerkiksi tehtaissa, jossa useammasta laitteesta kerätään dataa ja se lähetetään erillisen yhdyskäytävälaitteiston kautta. Laitteista tulevat tapahtumadatat menevät tietovirran prosessointiin (Stream Processing), josta dataa voidaan ladata varastoitavaksi tai suoraan

analytiikan käyttöön esimerkiksi koneoppimiseen. Tapahtumapohjaisessa arkkitehtuurissa datan pohjalta voidaan suoraan tehdä toimenpiteitä ja ohjata saadun informaation perusteella komentoja ja ohjauskäskyjä suoraan takaisin laitteille, jonka myötä arkkitehtuurista saadaan tapahtumapohjainen reagointi datan tarjoaman tiedon mukaisesti. (Microsoft Corporation 2018b). Kuviossa 8 esitelty tapahtumapohjaisen arkkitehtuurin hyödyntäminen IoT-ratkaisussa.



Kuvio 8. Tapahtumapohjainen arkkitehtuuri IoT-ratkaisussa (Microsoft Corporation 2018b)

2.4.3 Lambda-arkkitehtuuri

Nathan Marz ideoi termin Lambda-arkkitehtuuri kuvaamaan geneeristä mallia dataprosessointiin, joka on skaalautuva ja vikasietoinen. Hänen kokemuksensa massadatan käsittelystä ja siihen liittyvistä teknologioista on peräisin hänen työstään BackTypen ja Twitterin parissa. Lambda-arkkitehtuurissa ei ole teknologia sidonnaisuutta, vaan se antaa käytännön ja teorian periaatteet käsitellä ja kerätä massadataa. Se on hyvin geneerinen malli, jonka pohjana on yleiset vaatimukset massadatan käytöstä. Arkkitehtuuri mahdollistaa yhdistelmään historiadataa reaaliaikaisendatan kanssa. Lambdan perimmäisenä ajatuksena on kyky hallita ja prosessoida suuria määriä dataa hyödyntäen kahta komponenttikokonaisuutta: eräkerros (Batch Layer) ja nopeuskerros (Speed Layer). Tämän lisäksi Lambda-arkkitehtuurissa on

kolmaskin kokonaisuus: palvelukerros (Serving Layer), jonka tehtävä on nimensä mukaisesti tarjota prosessoitu data hyödynnettäväksi. Lambda-arkkitehtuurista tuli merkittävä ratkaisu massadatan hyödyntämiseen reaaliaikaiseen analytiikkaan ja yritysten digitaaliseen transformaatioon. Se mistä Lambda-arkkitehtuuri saa nimensä, ei ole täysin varmaa. (John & Misra 2017,67-69)

Marz ja Warren (2015) kuvaavat Lambda-arkkitehtuurin pääajatuksen olevan siis rakentaa massadata-järjestelmä näistä aiemmin mainitusta kolmesta kerroksesta. Tausta tälle ajatukselle lähtee yhtälöstä:

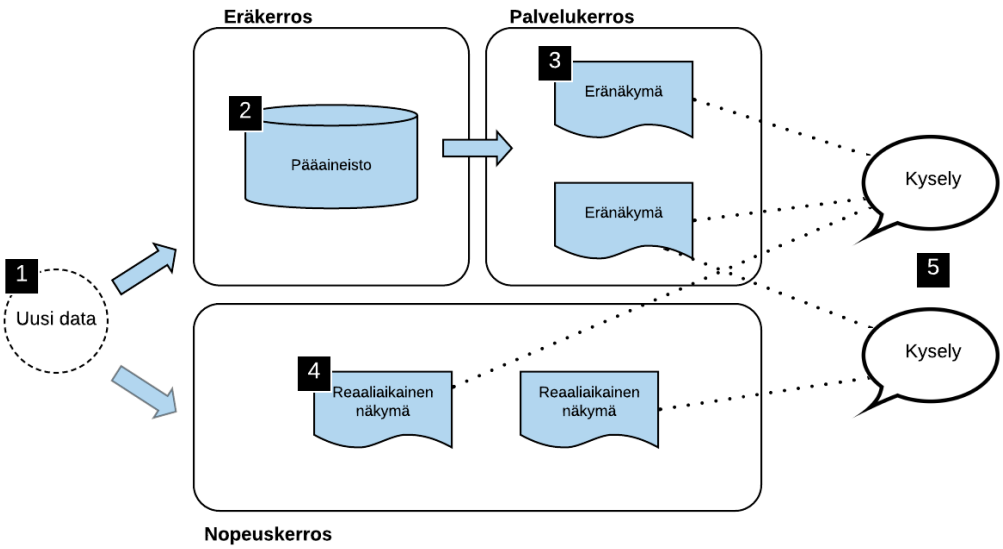
kysely = funktio(kaikki data)

Halutaan siis saada vastaus johonkin kysymykseen koko datasta, mutta tällaisen yhtälön toteuttaminen massadatalla lennosta veisi jatkuvalla käytöllä paljon prosessointiresursseja, jonka myötä sen kustannus tulisi kalliiksi. Sanomattakin on siis selvää, että tällaiset jatkuvassa käytössä olevat kyselyt halutaan esi-prosessoida valmiiksi, jonka myötä lennosta tehtävä kysely luetaankin tällaisesta esiprosessoidusta datasta. Samanlaista yhtälöajatusta käyttäen siis käytetäänkin kaiken datan sijasta esiprosessoitua dataa, joka tässä nimetty eränäkymä (batch view):

eränäkymä = funktio (kaikki data)

kysely = funktio (eränäkymä)

Hyödyntäen esiprosessoitua tietoaineistoa saadaan vastaus kysymykseen nopeammin, koska ei enää tarvitse prosessoida koko dataa joka kerta uudelleen, kun kysely tehdään. Kuitenkaan prosessoimalla koko datamassaa valmiiksi eränäkymiksi on haasteena vastata reaaliaikaisiin analyysihaasteisiin, koska eränäkymien prosessointi voi kestää kauankin riippuen datamassan suuruudesta. Näin ollen on oltava myös keino hyödyntää reaaliaikaista dataa, joten tätä varten rakennetaan nopeuskerros. Tämän kerroksen tehtävä on tarjota data nopeasti saataville, joten prosessointi kohdistuu ainoastaan uuteen dataan, eikä koko datahistoriaan kuten eräkerroksessa.



Kuvio 9. Lambda-arkkitehtuuri (Hausenblas & Bijnens 2017)

Kuviossa 9 on kuvattuna ylätasolla Lambda-arkkitehtuuri ja tähän kuvioon liittyen on Hausenblas ja Bijnens (2017) koonnut listatut arkkitehtuurin vaatimukset:

1. Kaikki data, joka tulee järjestelmään, lähetetään sekä erä- kuin nopeuskerrokselle prosessoitavaksi.
2. Eräkerroksella on kaksi funktiota: hallita pääainestoa (master dataset) ja esiprosessoida eränäkymät.
3. Palvelukerros indeksoi eränäkymät, jotta niitä voidaan kysellä alhaisella viivetasolla, ad-hoc-tavalla.
4. Nopeuskerros kompensoi korkeaa viivettä päivityksissä eräkerroksessa ja käsittelee ainoastaan tuoreinta dataa.
5. Mihin tahansa tulevaan kyselyyn saadaan vastaus yhdistelemällä erä- ja reaaliaikaisia näkymiä.

Eräkerros pitää sisällään alkuperäisen kopion datasta, jonka pohjalta prosessoidaan aikaisemmassa esimerkissä olleet eränäkymät. Tämä alkuperäinen kopio, pääaineisto, voi olla hyvinkin suuri määrä dataa. Eräkerros tulee olla kykenevä toteuttamaan kaksi asiaa: tallentamaan muuttumaton alkuperäisdata, jonka määrä kasvaa jatkuvasti, ja suorittamaan

sattumanvaraisia funktioita tästä pääaineistosta. Tämän kaltainen prosessointi on parasta toteuttaa käyttäen eräprosessoinnin järjestelmiä. Hadoop on hyvä esimerkki tällaisesta järjestelmästä. (Marz & Warren 2015, 16).

Palvelukerros on seuraava askel eräkerroksesta ja sen tarkoitus on tarjota eräkerroksen tuottamat eränäkymät käytettäväksi hajautettuun tietokantaan. Kun uudempi versio eränäkymästä tulee saataville, Palvelukerros automaattisesti vaihtaa tuon käytettäväksi kyselylähteeksi. Palvelukerros tukee eräpäivityksiä ja satunnaisia kyselyitä, mutta huomioitavaa on, ettei sen tarvitse tukea satunnaisia kirjoituksia (write). Tämä huomio on tärkeä, koska nämä satunnaiskirjoittamiset aiheuttavat suurimmat kompleksisuudet tietokannoissa. Toisin sanoen ilman satunnaiskirjoituksia, näistä palvelukerroksesta tulee samalla yksinkertaisempia. (Marz & Warren 2015, 17).

Palvelukerros päivittyy vasta, kun eräkerros on päivittänyt tiedot, joten sillä välin käytössä on niin sanotusti vanhentunutta dataa. Tähän ongelmaan nopeuskerros on kehitetty, se tarjoaa datan reaaliaikaisesti käytettäväksi. Isona erona eräkerrokseen on siis, että nopeuskerros tarjoaa vain viimeisimmän datan, kun eräkerros tarkastelee koko datamassaa. Saavuttaakseen pienimmän mahdollisen latenssin nopeuskerros ei tarkastele koko uutta dataa kerralla, vaan päivittää reaaliaikaiset näkymät inkrementaalisesti, kun uutta dataa tulee saataville. (Marz & Warren 2015, 18-19). Tämä voidaan esittää yhtälönä:

reaaliaikainen näkymä = funktio(reaaliaikainen näkymä, uusi data)

Lambda-arkkitehtuurista voidaan tehdä yhteenveto näiden kolmen funktion myötä:

eränäkymä = funktio(koko data)

reaaliaikainen näkymä = funktio(reaaliaikainen näkymä, uusi data)

kysely = funktio (eränäkymä, reaaliaikainen näkymä)

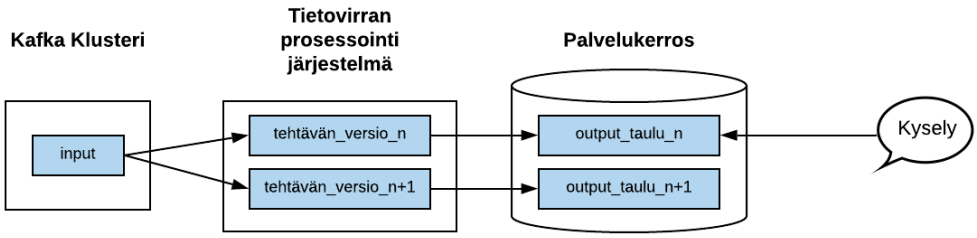
2.4.4 Kappa-arkkitehtuuri

Kappa-arkkitehtuuri on Jay Krepsin esittelemä vaihtoehto Lambda-arkkitehtuurille. Kappa-arkkitehtuuri on hyvin samankaltainen kuin Lambda-arkkitehtuuri, mutta huomattavasti

yksinkertaisempi, koska eräkerros on kokonaan poistettu. Pääajatuksena on välttää joutumasta prosessoimaan koko datamassaa uudelleen, vaan hoitaa datan prosessointi täysin reaaliaikaisena. Suurin etu tällä lähestymistavalla on, ettei dataa tarvitse prosessoida kahteen kertaa eri kerroksissa kuten Lambda-arkkitehtuurissa. (John & Misra 2017,87)

Kappa-arkkitehtuurissa siis tietovirran prosessoinnilla hoidetaan myös datahistorian käsittelyn. Tämä onnistuu kasvattamalla rinnakkaisajojen määrää ja samalla prosessikokonaisuudella onnistuu käsittelemään myös historiadatan prosessoinnit. Kreps (2014) määrittelee tämän uudelleenprosessoinnin suoraan tietovirran prosessoinnissa seuraavasti:

1. Käytä järjestelmää, joka pystyy säilyttämään täyden lokin datasta, joka halutaan kyetä uudelleenprosessoimaan ja joka mahdollistaa useita tilaajia (subscribers). Tällainen on esimerkiksi Apache Kafka, johon voidaan määrittää datan säilöntäaika järjestelmässä.
2. Kun halutaan uudelleen prosessoida dataa, niin käynnistetään toinen instanssi tietovirran prosessoinnissa ja syöttää tämän datan uuteen tauluun.
3. Kun tämä toinen instanssi on saanut prosessoinnin valmiiksi, vaihdetaan applikaatio lukemaan tätä uutta taulua.
4. Pysäytetään vanha versio prosessoinnista ja poistetaan vanha taulu.



Kuvio 10. Kappa-arkkitehtuuri (Kreps 2014)

3 Tapaustutkimus

Tutkimusosa toteutetaan tapaustutkimuksena, jossa tutkittavina tapauksina ovat Lambda- ja Kappa-arkkitehtuurin hyödyntäminen massadatalle kolmessa julkipilvessä: Amazon, Microsoft ja Google. Nämä kolme valikoituivat tutkittaviksi alustoiksi, koska ne ovat tätä tutkielmaa tehdessä kolme johtavaa pilvitarjoajaa (Statista 2019).

3.1 Tutkimusmetodi

Tapaustutkimus on laadullinen tutkimus, jossa keskeisenä on tutkittavat tapaukset, johon perustuu määritellyt tutkimuskysymykset, tutkimusasetelmat ja aineistoanalyysit. Tapaustutkimukselle on haastava antaa yhtä yleispätevää tai kattavaa määritelmää, koska tapaustutkimus on enemmän lähestymistapa kuin aineistojen analysointi- tai koontimenetelmä. Useilla tieteenaloilla tapaustutkimusta hyödynnetään erilaisista lähtökohdista ja eri tavoittein. Yhdenmukaista tapaustutkimuksissa on, että niissä tarkastellaan yhtä tai useampaa tapausta. Näiden tapausten määrittely, analysointi ja ratkaiseminen on keskeinen tavoite tapaustutkimuksessa, mutta ei ole itsestäänselvyys, kuinka tutkittavat tapaukset valitaan, rajataan tai perustellaan. (Eriksson & Koistinen 2005, 1-4)

Tapaustutkimuksessa tulee vastaan useampia valintoja niin tutkittavan tapauksen kuin lähestymistavan myötä. Valinnat eivät ole lineaarisia, vaan ne muodostuvat tutkimuksen edetessä. Tavoitteena on saada tieteellisesti pätevä ja perusteltu kokonaisuus, joka ilmentää tutkittavaa tapausta. Erityistä tapaustutkimuksessa on, että tutkittavaa tapausta pyritään ymmärtämään kokonaisuutena, jossa voi olla useampia näkökulmia ja prosesseja. (Laine, Bamberg & Jokinen 2007, 41-42)

Valittua tapausta tai tapauksia voidaan tarkastella monista näkökulmista, joten on tärkeää valita mistä näkökulmasta tutkija aikoo aihettaan katsoa. Liian laajan aiheen valinta johtaa, että tutkimuksessa ei saavuteta riittävää syvyyttä. Tutkijan on myös pohdittava, onko koko tapausta perusteltua ylipäätään tutkia kokonaisuudessaan vai olisiko perustellumpaa valita vain jokin osa tutkimuksen kohteeksi. Rajaamalla tutkimusta saavutetaan yhtenäisempi

kokonaisuus, samalla erottaen mikä on olennaista ja mikä ei tutkimukselle. (Laine, Bamberg & Jokinen 2007, 57-58)

Olennainen lähestymistapa tapaustutkimukseen on tapausten vertailu. Useamman tapauksen valinta ja vertailu voi auttaa löytämään tutkimuskysymyksiä, joita ei yksittäistä tapausta tutkimalla huomata. Vertaileva menetelmä on laajimmillaan mikä tahansa tutkimustekniikka, joka pyrkii selittämään vaihtelua. (Laine, Bamberg & Jokinen 2007, 74)

3.2 Vertailtavat arkkitehtuurit

Reaaliaikainen data tarjoaa mahdollisuuksia saada datasta irti informaatiota välittömästi, jolloin dataan reagoiminen päätöksen teossa tai ennusteissa on heti toteutettavissa. Yhdistämällä tämä vielä aiemmin kerättyyn dataan saadaan luotua vielä kattavampia analyyseja datan pohjalta ja näin esimerkiksi saadaan parempia ennustemalleja suoraan käytettäväksi. Reaaliaikaisuus on kuitenkin iso haaste massadatan osalta, koska perinteisillä tavoilla dataa ei ole mahdollista prosessoida riittävän nopeasti ilman erittäin suuria prosessointikapasiteetteja.

Tapaustutkimuksen osalta verrataan kahta arkkitehtuuria, joissa on mahdollisuus hyödyntää reaaliaikaista analyysia eräpohjaisten analyysien kanssa: Lambda- ja Kappa-arkkitehtuurit. Näiden kahden arkkitehtuurin pohjalta muodostetaan arkkitehtuurit kolmen julkipilven tarjoajan palveluista: Amazon, Google ja Microsoft.

Valittavissa teknologioissa arkkitehtuureihin painotetaan enemmän kustannustehokkuuteen, jolloin vertaillaan datan säilytyksestä ja prosessoinnista syntyviä kustannuksia näiden kahden arkkitehtuurin välillä. Valittavissa palveluissa käytetään hinnaston ja valikoiman puolesta Pohjois-Euroopan valikoimaa.

3.3 Käytettävien pilvipalveluiden valinta

Arkkitehtuureissa käytetyt pilviteknologiat ja -palvelut on valittu pilvipalveluntarjoajien suositusten mukaisesti. Valinnoissa on painottunut massadatan vaatimukset volyymin, vaihtuvuuden ja vauhdin mukaisesti, jolloin palveluiden on kyettävä tallentamaan ja

prosessoimaan eri tyyppistä dataa. Myöskin massadataa säilöttäessä on huomioitava tallennustilan kustannukset, jotka varsinkin NoSQL-tietokannoilla voivat kustannukset nousta suuriksikin, varsinkin jos palvelu käyttää SSD-muisteja (Solid-state drive).

Suositukset pilvipalveluiden käytöstä on katsottu tutkittavien pilvipalveluiden tarjoajien omilta referenssisivuilta.

3.4 Tutkimuksen tapaukset

Tutkimuksen kohteena on siis Lambda- ja Kappa-arkkitehtuurin luominen kolmelle eniten käytössä olevalle eri julkipilven tarjoajalle ja millaisia palveluita heiltä löytyy arkkitehtuurin toteuttamiseksi.

3.4.1 Amazon Web Services

Amazon Web Services (AWS) käsitteen esittelivät Chris Pinkham ja Benjamin Black Amazonin perustajalle Jeff Bezosille vuonna 2003. AWS:n visiona oli toimia Amazonin infrastruktuurina, joka olisi täysin standardoitu, täysin automatisoitu ja nojautuisi laajasti web-palveluihin. (Black, 2009).

Amazon aloitti siis toimintansa kirjojen verkkokauppana, josta kehittyi ajansaatossa johtava pilvilaskennan tarjoaja. AWS virallisesti julkistettiin maaliskuussa 2006, jolloin AWS tarjosi Simple Storage Serviceä (S3) ensimmäisenä julkisena palvelunaan. Tämä palvelu tarjoaa konseptin objektitallennukselle verkon yli, jolloin kuka tahansa voi esimerkiksi tallentaa kuvia, tiedostoja tai mediasisältöä helposti saataville. S3-palvelun jälkeen myös muita pilvipalveluita aloitettiin lisäämään AWS:n tarjontaan. (Golden 2013,10-11).

AWS on turvallinen pilvipalvelualusta, joka tarjoaa pilvilaskentatehoa, tietovarastointia, sisällön toimitusta (content delivery) ja muita toiminnallisuuksia auttamaan bisneksiä skaalautumaan ja kasvamaan. Miljoonat asiakkaat hyödyntävät AWS:n tuotteita ja ratkaisuja vipuvartena rakentamaan kehittyneitä sovelluksia kasvattamaan joustavuutta, skaalautuvuutta ja luotettavuutta. AWS tarjoaa yli 50 palvelua valmiina käyttöön otettavaksi muutamalla hiiren painalluksella ilman etukäteismaksuja. Näin ollen suuret, pk- ja start-up- yritykset

voivat hyödyntää palveluita vastaamaan nopeasti muuttuviin bisnestarpeisiin. (Amazon Web Services, Inc, 2019a).

Tätä tutkielmaa tehdessä AWS tarjoaa pilvipalveluitaan 61 saatavuusalueella (Availability Zone), jotka jakautuvat 20 maantieteelliselle alueelle. AWS on myös tällä hetkellä tuomassa 15 saatavuusaluetta ja viisi maantieteellistä aluetta lisää. (Amazon Web Services, Inc, 2019a).

3.4.2 Google Cloud Platform

Google Cloud Platformin (GCP) perusta julkaistiin esikatseluun vuonna 2008, kun Google julkaisi Google App Engine -palvelun, joka tarjoaa kehitystyökalut suorittaa web-sovelluksia Googlen infrastruktuurissa. (McDonald 2008).

GCP on rakennettu samalle maailmanluokan infrastruktuurille, jonka Google on suunnitellut, koonnut ja käyttänyt yrityksen tuotteissa esimerkiksi Google-haku, joka toimittaa miljardeja hakutuloksia millisekunneissa. massadatan osalta Google on kehittänyt useita innovaatioita, kuten Google File System (GFS) vuonna 2002 ja MapReducen vuonna 2004, joita esimerkiksi avoimen lähdekoodin projekti Apache Hadoop on hyödyntänyt. (Krishnan & Gonzalez, 2015, 7-9).

Tätä tutkielmaa kirjoittaessa GCP tarjoaa palveluitaan 58 alueella (zone), jotka ovat jakautuneet 19 maantieteelliseen alueeseen. (Google LLC, 2019a).

3.4.3 Microsoft Azure

Microsoft Azure, josta aiemmin käytettiin nimeä Windows Azure, julkistettiin ensimmäisen kerran 2008 ja se oli silloin saatavilla yhteisön teknilliseen esikatseluun (Comminity Technical Preview, CTP). Azure tuli kaupallisesti saataville 2010, jonka jälkeen sen palvelut ja ominaisuudet ovat jatkaneet jatkuvaa kasvua. Windows Azure uudelleen nimettiin johtuen, että sen brändiä haluttiin tuoda pois sidoksista Windowsin käyttöjärjestelmien, tietokantojen ja sovellusalustojen kanssa, koska Azure esimerkiksi pystyi tukea myös Linux-

käyttöjärjestelmiä virtuaalisissa koneissa, Oraclea tietokannoissa ja Node.js:ää verkkosivujen kehityksessä. (Webber-Cross 2014, 11)

Microsoft Azure on alituisesti laajeneva kokoelma pilvipalveluita, joiden avulla organisaatiot voivat vastata bisneksen luomiin haasteisiin. Azuressa voi luoda, hallita ja ottaa käyttöön sovelluksia massiivisessa julkiverkossa käyttäen Microsoftin aikaisempia työkaluja ja viitekehyksiä. Azuren palvelut tätä tutkielmaa tehdessä tarjolla 54:llä eri seudulla (region), joka on enemmän kuin millään muulla julkipilven tarjoajalla. Fortune 500 -yrityksistä 95% käyttää Azuren pilvipalveluita bisneksessään. (Microsoft 2019a).

Microsoft määrittää neljä pääsyytä miksi Azure on oikea valinta: produktiivisuus, hybridisyys, älykkyys ja luotettavuus. Produktiivinen hyöty on, että Azure vähentää markkinointisyklejä tuottaen ominaisuuksia nopeammin yli 100:n end-to-end-palvelun avulla. Hybridisyydellä tarkoitetaan kykyä kehittää ja ottaa käyttöön palveluita paikasta riippumatta, sekä hyödyntää pilvipalveluita myös on-premises-käyttöympäristöissä hyödyntäen Azure Stackia. Azuren älykkyys taas tulee palveluista esimerkiksi koneoppimis- ja tekoälypalveluiden ympärillä, jotka Azure tarjoaa suoraan valmiina tuotteina. Luotettavuus näkyy Azuressa siinä, että sen on valinnut pilvitarjoajaksi niin startup-yritykset, hallitukset ja 95 prosenttia Fortune 500 -yrityksistä. (Microsoft Corporation, 2019). Kuviossa 11 esitellään päätösvirtaa Microsoft Azuren valitsemista.

Kuvio 11.Päätösvirta Microsoft Azuren valitsemisesta pilvitarjoajaksi (Webber-Cross 2014, 15)

3.5 Lambda-arkkitehtuuri

Lambda-arkkitehtuurin vaatimukset olivat:

1. Kaikki data, joka tulee järjestelmään, lähetetään sekä erä- kuin nopeuskerrokselle prosessoitavaksi.
2. Eräkerroksella on kaksi funktiota: hallita pääainestoa (master dataset) ja esiprosessoida eränäkymät.

3. Palvelukerros indeksoi eränäkymät, jotta niitä voidaan kysellä alhaisella viivetasolla, ad-hoc-tavalla.
4. Nopeuskerros kompensoi korkeaa viivettä päivityksissä eräkerroksella ja käsittelee ainoastaan tuoreinta dataa.
5. Mihin tahansa tulevaan kyselyyn saadaan vastaus yhdistelemällä erä- ja reaaliaikaisia näkymiä.

Ensimmäisen vaatimuksen täyttämiseksi tulee valita palvelu, jolla esimerkiksi tietovirtadata voidaan helposti lukea sisään ja jakaa kahdeksi putkeksi. Amazonin palveluista tällainen on Amazon Kinesis Data Firehose, jolla voidaan lukea tietovirtadataa suoraan datasäilöön ja tarjota samalla data reaaliaikaseen käyttöön. Amazon Kinesis Data Firehose on palveliton, jolloin sen käyttöönotto on hyvin yksinkertaista ja ylläpitäminen maksutonta. Maksu koostuu pelkästään käytöstä. (Amazon Web Services, Inc 2019b).

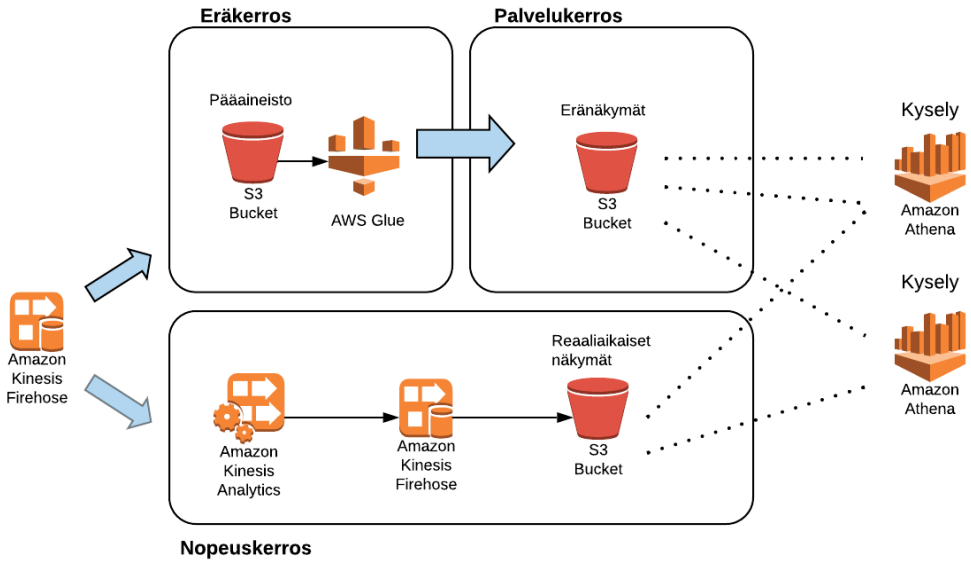
Toiseen vaatimukseen tarvitaan kahta palvelua: yksi datan säilömiseen ja toinen datan prosessointiin, mikäli tätä ei voida toteuttaa samassa palvelussa. Näin ollen datan säilömiseen käytetään Amazon S3:sta, joka tulee sanoista Simple Storage Service. Amazon S3 on objektivarastopalvelu, joka tarjoaa skaalautuvuutta, tietoturvallisuutta tiedostoille sekä suorituskykyä. Amazon S3 tarjoaa helppokäyttöiset hallintatoiminnot, joilla voidaan hallita objektidataa ja sen käytön valvontaa. (Amazon Web Services, Inc 2019c). Datan prosessointi sen sijaan suoritetaan AWS Glue -palvelulla, joka on Amazonin ETL-työkalu (Extract, Transform & Load). AWS Glue on myös palveliton palvelu, jolloin kustannukset syntyvät pelkästään palvelun käytöstä. AWS Glue perustuu Apache Sparkiin, joka toimii siinä pohjana. (Amazon Web Services, Inc 2019d).

Kolmas vaatimus on tuoda edellisen vaiheen datat käytettäväksi palvelukerrokseen. Datan säilömiseen Amazonin palveluista hyödynnetään samaa palvelua kuin raakadatan säilömiseen eli Amazon S3sta. Datan tarjoaminen S3sta toteutetaan Amazon Athenalla, joka on interaktiivinen kyselypalvelu. Athenalla voidaan kysellä dataa S3sta hyödyntäen SQL-kyselyitä. Athena on myös palveliton palvelu. (Amazon Web Services, Inc 2019e).

Neljäs vaatimus käsittelee nopeuskerroksen tarjoamaa dataa, jonka tulee olla latenssiltaan hyvin pientä. Tähän kuuluu myös reaaliaikaisten analyysien laskeminen. Amazonin

palveluista reaaliaikaiseen analysiin on järkevintä käyttää Amazon Kinesis Data Analytics, jolla voidaan tehdä prosessointia suoraan tietovirtadatalle. Samalla tavalla kuin Firehose, Data Analytics on palveliton palvelu. (Amazon Web Services, Inc 2019f). Prosessoitu data siirretään tämän jälkeen Firehosella Amazon S3 -objektivarastoon, johon muodostuvat näin reaaliaikaiset näkymät. Tätä dataa voidaan lukea samanlailla kuin erädataa eli hyödyntäen Athenaa datan hakemisessa.

Viides vaatimus on kyky yhdistellä erä- ja reaaliaikaisia näkymiä. Amazonin osalta tämä tehdään Athenalla, kuten aikaisemmissa vaatimuksissa käytiin läpi.



Kuvio 12. Amazon Web Services: Lambda-arkkitehtuuri

Kuviossa 12 esitellyn Amazonin teknologioilla rakennetun Lambda-arkkitehtuurissa käytetyt teknologiat:

- Amazon Athena
- Amazon Kinesis Analytics
- Amazon Kinesis Firehose
- Amazon S3
- AWS Glue

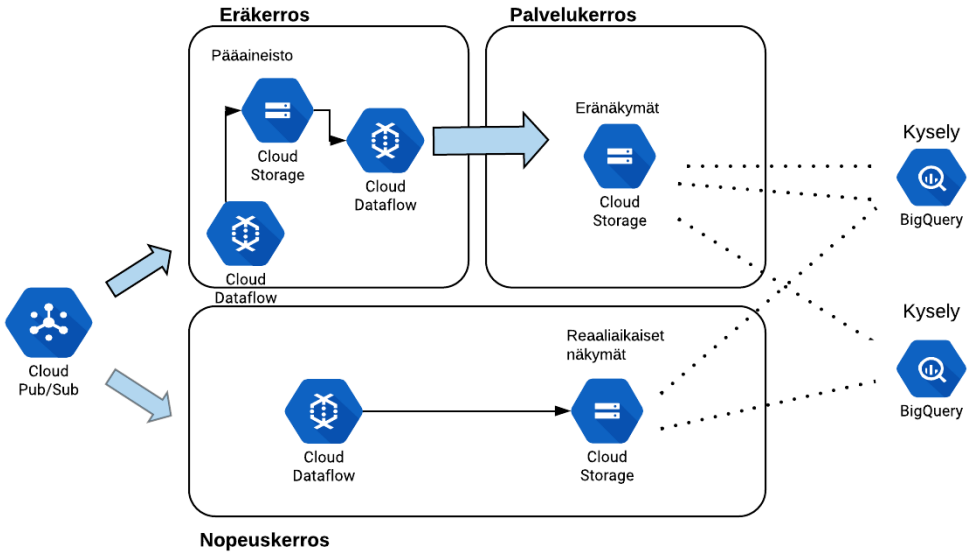
Ensimmäisen vaatimuksen osalta Google Cloud Platformissa datan jakaminen näihin kahteen putkeen tehdään Cloud Pub/Sub -palvelu, joka on skaalautuva tietovirtadatan käsittelijä, jonka myötä data saadaan tarjottavaksi muille palveluille. (Google LLC 2019b). Tästä palvelusta data on vielä siirrettävä näihin kahteen putkeen, johon hyödynnetään Cloud Dataflowta, jolla voidaan käsitellä niin tietovirta- kuin erädataa. Cloud Dataflow hyödyntää Apache Beam -palvelua prosessointimoottorina. (Google LLC 2019c).

Toiseen vaatimukseen Googlen palveluista master-aineiston säilömiseen käytetään Cloud Storagea, joka on hyvin samankaltainen kuin Amazonin S3. Se tarjoaa korkeatasoisen objektivaraston, josta löytyy lisäksi erittäin kattava ohjelmistorajapinta. (Google LLC 2019d). Datan prosessointiin hyödynnetään samaa palvelua kuin aikaisemmassa vaatimuksessa eli Cloud Dataflowta, jolla data siirretään säilöttäväksi palvelukerroksen käytettäväksi.

Kolmannen vaatimuksen täyttämiseksi eränäkymät tallennetaan Cloud Storageen, josta niitä voidaan hyödyntää.

Neljänteen vaatimukseen Googlen osalta nopeuskerroksen toteutus tehdään hyödyntäen Cloud Dataflown tietovirtadatan käsittelyä ja data tallennetaan Cloud Storage-palveluun reaaliaikaisiksi näkymiksi.

Viidennen vaatimuksen osalta Googlen palveluista hyödynnetään Googlen palvelua BigQuerya, jolla data voidaan lukea suoraan Cloud Storagesta ja joka on myös palveliton palvelu. (Google LLC 2019f).



Kuvio 13. Google Cloud Platform: Lambda-arkkitehtuuri

Kuviossa 13 esitellään Googlen palveluilla rakennetun Lambda-arkkitehtuurin käytetyt teknologiat:

- Cloud BigQuery
- Cloud Dataflow
- Cloud Pub/Sub
- Cloud Storage

Microsoft Azuren palveluista ensimmäisen vaatimuksen osalta valitaan Azure Event Hub, jolla voidaan lukea reaaliaikaista tietovirtadataa suoraan sisään ja jakaa se erä- ja nopeuskerrokselle. Event Hub -kerää stiimidatan tarjottavaksi samalla tapaa kuin Google Pub/Sub ja Event Hub on myös palveliton palvelu. Event Hubilla voidaan tallentaa data suoraan eräkerrokseen käyttäen kaappausominaisuutta (capture) ja nopeuskerros voi lukea datan suoraan täältä käsiteltäväksi. (Microsoft Corporation 2019b).

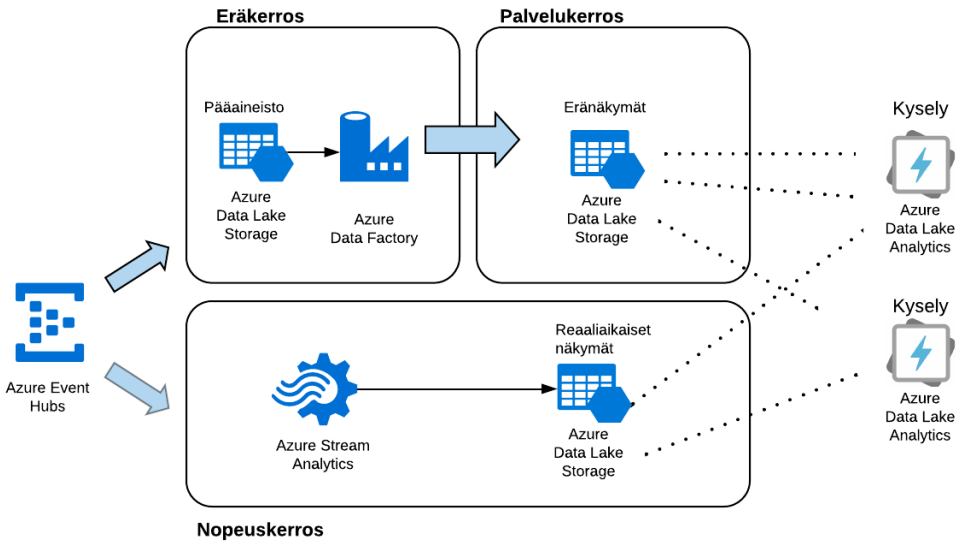
Toiseen vaatimukseen Microsoftin palveluista hyödynnetään myös objektivarastoa nimeltään Azure Data Lake Storage. Tästä palvelusta hyödynnetään uudempaa versiota Gen2, joka on rakennettu Azure Blob Storagen pohjalta, kun aikaisempi versio perustui Hadoop-

distribuutioon. Tämän myötä datan säilömisen hinnoittelua on saatu pienemmäksi. Erona Azure Blob Storageen Data Lake Storage säilöö datan hierarkisesti, jolloin sen hyödyntäminen analyyttisiin tarpeisiin on tehokkaampaa. (Microsoft Corporation 2019c). Datan prosessointi eräpohjaisiin näkymiin tehdään Azure Data Factorylla, joka on koodivapaa (code-free) datan integrointipalvelu. Tämä tarkoittaa, että Data Factorystä löytyy suoraan graafinen käyttöliittymä, jolla voidaan luoda tarvittavat lataukset tuottamaan palvelukerros eränäkymät. (Microsoft Corporation 2019d).

Kolmannessa vaatimuksessa Microsoftin osalta Data Factoryn tuottamat eränäkymät tallennetaan takaisin Azure Data Lake Storageen, josta niitä voidaan kysellä hyödyntäen Azure Data Lake Analytics -palvelua, joka on samankaltainen kuin Amazonin Athena, ja jolla data voidaan lukea suoraan Data Lake Storagesta käytettäväksi. (Microsoft Corporation 2019e).

Neljänteen vaatimukseen Microsoftin palveluista hyödynnetään Azure Stream Analytics -palvelua, jolla data luetaan Event Hubista ja prosessoidaan reaaliaikaisiksi näkymiksi Azure Data Lake Storageen, josta se on luettavissa Azure Data Lake Analyticsillä. Azure Stream Analytics on siis tietovirtadatan prosessointiin ja analysointiin tarkoitettu palveliton palvelu, jolla voidaan käsitellä dataa reaaliaikaisesti. (Microsoft Corporation 2019f).

Viidennen vaatimuksen osalta Microsoftin palveluista Azure Data Lake Analytics yhdistää erä- ja reaaliaikaset datat keskenään.



Kuvio 14. Microsoft Azure: Lambda-arkkitehtuuri

Kuviossa 14 esitellään Microsoftin palveluilla rakennetussa Lambda-arkkitehtuurissa käytetyt teknologiat:

- Azure Data Factory
- Azure Data Lake Analytics
- Azure Data Lake Storage
- Azure Event Hub
- Azure Stream Analytics

3.6 Kappa-arkkitehtuuri

Kappa-arkkitehtuurin vaatimukset:

1. Käytä järjestelmää, joka pystyy säilyttämään täyden lokin datasta, joka halutaan kyetä uudelleenprosessoimaan ja joka mahdollistaa useita tilaajia (subscribers). Tällainen on esimerkiksi Apache Kafka, johon voidaan määrittää datan säilöntäaika järjestelmässä.

2. Kun halutaan uudelleen prosessoida dataa, niin käynnistetään toinen instanssi tietovirran prosessoinnissa ja syöttää tämän datan uuteen tauluun.
3. Kun tämä toinen instanssi on saanut prosessoinnin valmiiksi, vaihdetaan applikaatio lukemaan tätä uutta taulua.
4. Pysäytetään vanha versio prosessoinnista ja poistetaan vanha taulu.

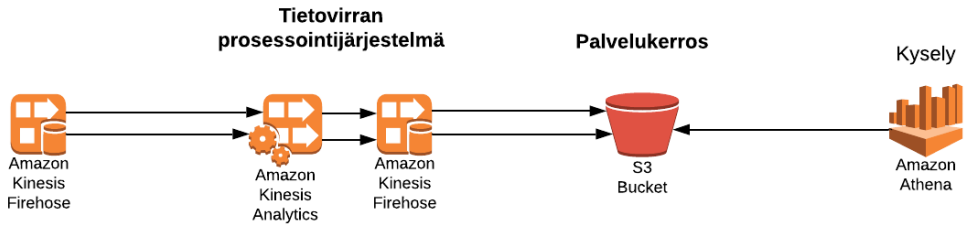
Ensimmäisen vaatimuksen osalta hyödynnetään samoja palveluita kuin Lambda-arkkitehtuurissa: Amazonilta Kinesis, Googlelta Cloud Pub/Sub ja Microsoftilta Event Hub. Jokainen näistä palveluista voidaan määritellä säilyttämään dataa palvelussa ja jokainen perustuu useamman tilaajan hallintaan. Datan prosessoinnissa hyödynnetään myös samoja palveluita kuin Lambdan nopeuskerroksessa eli Amazonilta Kinesis Analytics, Googlelta Cloud Dataflow ja Microsoftilta Stream Analytics.

Toiseen vaatimukseen nämä prosessointipalvelut pystyvät vastaamaan myös, koska jokainen näistä on skaalautuva palvelu, jossa voidaan käynnistää prosesseja rinnakkain. Datan säilömisessä hyödynnetään samoja palveluita kuin Lambda-arkkitehtuurissa: Amazonin osalta data säilötään Amazon S3seen, Googlen osalta hyödynnetään Cloud Storagea ja Microsoftin palveluista käytetään Azure Data Lake Storagea.

Kolmannen vaatimuksen taulun vaihdot ja neljännen vaatimukset vanhan version poistaminen voidaan hoitaa hyödyntäen objektivarastojen ominaisuuksia objektien uudelleen nimeämisestä ja poistamisesta. Samoin vanhat prosessointitehtävät voidaan automatisoidusti siivota pois prosessointipalveluista.

Kuviossa 15 esitellään Amazonin palveluilla rakennettu Kappa-arkkitehtuuri, jossa on siis käytetty seuraavia palveluita:

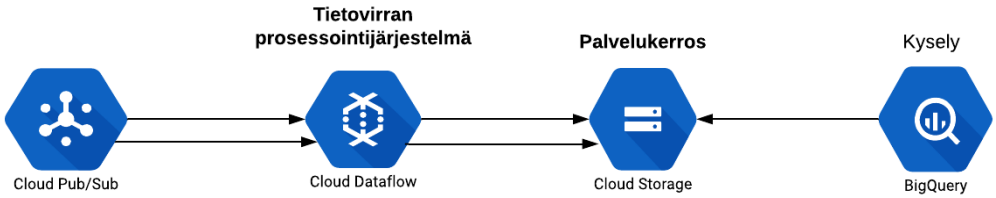
- Amazon Athena
- Amazon Kinesis Analytics
- Amazon Kinesis Firehose
- Amazon S3



Kuvio 15. Amazon Web Services: Kappa-arkkitehtuuri

Kuviossa 16 on esitelty Googlen palveluilla rakennettu Kappa-arkkitehtuuri, jossa käytetty seuraavia palveluita:

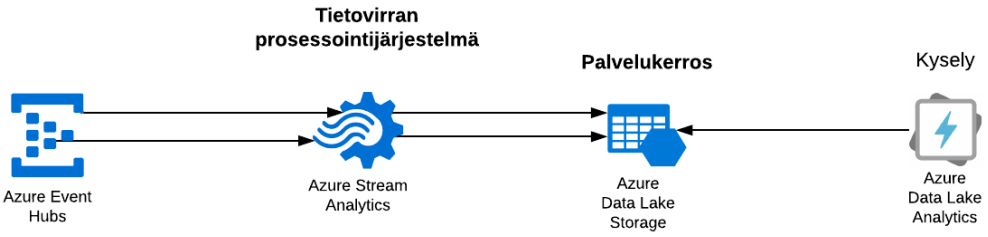
- Cloud BigQuery
- Cloud Dataflow
- Cloud Pub/Sub
- Cloud Storage



Kuvio 16. Google Cloud Platform: Kappa-arkkitehtuuri

Kuviossa 17 on esitelty Microsoftin Azure-palveluilla muodostettu Kappa-arkkitehtuuri, jossa käytetty seuraavia palveluita:

- Azure Data Lake Analytics
- Azure Data Lake Storage
- Azure Event Hub
- Azure Stream Analytics



Kuvio 17. Microsoft Azure: Kappa-arkkitehtuuri

4 Tulokset

Tässä osiossa käsitellään tapaustutkimuksen tulokset tarkastellen eri arkkitehtuurien kustannuksia eri pilvitarjoajilla. Kustannuksien tarkastellussa käytetään tutkimusmenetelmänä kirjallisuuskatsausta, johon teoria kerätään palveluntarjoajien verkkosivuilta, ja vertailussa keskitytään datamääriin eikä toistokertoihin. Tarkastelussa käydään läpi datan säilömisen ja datan prosessoinnin kustannuksia. Pilvipalveluiden hinnasto perustuu tätä tutkielmaa tehtäessä vallitsevaan hintatasoon Pohjois-Euroopan alueella. Amazonin palvelimet Pohjois-Euroopassa sijaitsevat Ruotsissa, Googlella Suomessa ja Microsoftilla Irlannissa.

4.1 Lambda-arkkitehtuurin kustannukset

4.1.1 Eräkerros

Amazonin tapauksessa data tulee eräkerrokseen Kinesis Firehosen kautta ja data tallennetaan Amazon S3 -objektivarastoon. Eräfunktiot on toteutettu AWS Glue -palvelulla. Amazon S3sta on eri vaihtoehtoja, mutta tutkielmassa tarkastellaan S3 Standard -vaihtoehtoa. Datavarastoinnin kustannuksien lisäksi S3 veloittaa pieniä kustannuksia datan lukemista ja tallentamisesta, jotka perustuvat tehtyjen kutsujen (request) määriin, jotka laskutetaan tuhansien toistojen erissä, mutta näitä ei huomioida tässä tutkielmassa. Taulukossa 1 esitellään Firehosen hinnasto datan käsittelylle ja taulukossa 2 Amazon S3 -hinnasto tiedonvarastoinnille.

Dataa käsitelty	**Dollaria / GB**
Ensimmäiset 500 TB / kuukausi	$ 0.031
Seuraavat 1.5 PB / kuukausi	$ 0.027
Seuraavat 3 PB / kuukausi	$ 0.022
Yli 5 PB / kuukausi	Amazonin kanssa sovittavissa

Taulukko 1. Amazon Kinesis Firehose -hinnasto (Amazon Web Services, Inc 2019h)

Datamäärä	**Dollaria / Kuukausi**
Ensimmäiset 50 TB / kuukausi	$ 0.023 / GB
Seuraavat 450 TB / kuukausi	$ 0.022 / GB
Yli 500 TB / kuukausi	$ 0.021 / GB

Taulukko 2. Amazon S3 -hinnasto (Amazon Web Services, Inc 2019i)

Datan prosessoinnin hinnasto eränäkymiin AWS Glueta käyttäen on sekuntipohjainen ja perustuu tietoprosessointiyksikköä (Data Processing Unit, DPU) on käytössä. Yksi tällainen prosessointiyksikkö sisältää 4 vCPU:ta ja 16 gigaa RAM-muistia, ja maksaa 0,44 dollaria tunnilta. (Amazon Web Services, Inc 2019j)

Googlen pilvialustalle tehdyssä arkkitehtuurissa data tulee sisään Cloud Pub/Sub -palvelusta, josta se luetaan käyttäen Cloud Dataflow:ta ja tallennetaan Cloud Storageen. Prosessointi Cloud Storagesta prosessointiin hyödynnetään Cloud Dataflowta. Cloud Pub/Sub hinnasto perustuu datamääriin, joita palvelulla käsitellään, ja hinnasto on kuvattuna taulukossa 3.

Kuukausittainen datavolyymi	**Dollaria / TiB**
Ensimmäiset 10 GB	$ 0.00
Seuraavat 50 TB	$ 60
Seuraavat 100 TB	$ 50
Yli 150 TB	$ 40

Taulukko 3. Google Cloud Pub/Sub -hinnasto (Google LLC 2019b)

Datan säilömisen hinnoittelu Google Cloud Storagessa riippuu valittavasta säilytystavasta, samalla tapaa kuin Amazon S3sessa. Näistä säilytystavoista hyödynnetään maantieteellistä säilytystä (Regional Storage), jossa data on säilötty yhteen maantieteelliseen alueeseen.

Hinta on tälle 0,02 dollaria per gigatavu, josta ensimmäiset viisi gigatavua ovat ilmaiset. (Google LLC 2019d)

Datan lataaminen Cloud Storageen ja sieltä sen prosessointi tehdään molemmat Cloud Dataflowta käyttäen. Cloud Dataflown hinnastosta löytyy eri hinnat erädatalle ja tietovirtadatalle. Luettaessa data sisään hyödynnetään tietovirtadatan käsittelyä ja eränäkymien luonnissa eräkäsittelyä. Taulukossa 4 esitelty Cloud Dataflow -palvelun hinnasto.

Cloud Dataflow **Prosessointityyppi**	**vCPU** **Dollaria / tunti**	**RAM-muisti** **Dollaria / GB / tunti**
Erä (Oletus: 1 vCPU, 3.75 GB)	$ 0.0616	$ 0.0042684
Tietovirta (Oletus: 4 vCPU, 15 GB)	$ 0.0759	$ 0. 0042684

Taulukko 4. Cloud Dataflow -hinnasto (Google LLC 2019c)

Microsoftin palveluista käytetään Event Hubia lukemaan data sisään, josta se tallennetaan Azure Data Lake Storageen. Data Lake Storagesta prosessointi suoritetaan Data Factoryllä. Event Hubista on valittavissa eri tasoisia palveluita, mutta tässä tutkimuksessa on valittu käytettäväksi Standard-tasoa, jolloin käytössä on Capture-ominaisuus ja Apache Kafka. (Microsoft Corporation 2019h). Taulukossa 5 kuvattuna Event Hub -palvelun hinnasto.

Suorituskyky-yksikkö (1 MB/s sisään, 2 MB/s ulos)	$ 0.03 / tunti
Sisäänlukutapahtumat (Ingress Event)	$ 0.028 / miljoona tapahtumaa
Capture	$ 0.10 / tunti

Taulukko 5. Azure Event Hub -hinnasto (Microsoft Corporation 2019h)

Data säilömiseen Data Lake Storagesta löytyy samanlailla kuin Amazonilta ja Googleta eri vaihtoehtoja. Näistä palvelutasoista käytetään Hot-tasoa, joka vastaa samaa kuin mitä valittu

muilta tarjoajilta. Tiedonvarastoinnin hinnat muuttuvat sen mukaan paljon dataa tallennetaan ja hinnasto on kuvattuna taulukossa 6.

Datamäärä	**Dollaria / GB**
Ensimmäiset 50 TB / Kuukausi	$ 0.022
Seuraavat 450 TB / Kuukausi	$ 0.0211
Yli 500 TB / Kuukausi	$ 0.0202

Taulukko 6. Azure Data Lake Storage -hinnasto (Microsoft Corporation 2019i)

Datan prosessoinnin kustannukset näkymiin Microsoftin pilvipalvelussa käyttäen Data Factoryä koostuvat datan prosessoinnista ja ajettavista aktiviteeteista. Datan prosessointi maksaa jokaista dataintegraatioyksikköä (Data Integration Unit, DIU) kohden 0.25 dollaria tunnilta. Microsoft ei ole kuvannut tarkalleen paljonko virtuaalisia prosessoreita tai muistia on yksi dataintegraatioyksikkö. Aktiviteetit maksavat 0.005 dollaria tunnilta per aktiviteetti. (Microsoft Corporation 2019j).

Näiden pohjalta voidaan tehdä vertailuja datan lukemisesta eräkerrokseen, sen säilömisestä siellä ja prosessoimisesta eränäkymiin. Datan sisäänlukemisessa on vaikeaa verrata Azuren ratkaisua Googlen ja Amazonin vastaaviin, koska tuon kustannusrakenne eroaa täysin. Amazonin ja Googlen kustannukset taas ovat vertailukelpoisia, mutta Google on muuttamassa hinnoitteluaan kesäkuussa 2019. (Google LLC 2019b). Jo pelkästään taulukkoa katselemalla ja nopealla laskutoimituksella Google on huomattavasti kalliimpi kuin Amazon tältä osin. 50 teratavun datamassalla Amazonin kustannus on 1550 dollaria, kun Googlen vastaava on lähes 3000 dollaria eli tuplasti kalliimpi.

Datan säilömisen osalta vertailu on helpompaa, koska kustannukset pilvipalvelutarjoajin välillä rakentuvat samalla tavalla eli perustuen datamassaan.

	Dollaria / GB		
Datamäärä	**Amazon S3**	**Google Cloud Storage**	**Microsoft Azure Data Lake Storage**
<= 5GB / Kuukausi	$ 0.023	Ilmainen	$ 0.022
Ensimmäiset 50 TB / Kuukausi	$ 0.023	$ 0.02	$ 0.022
Seuraavat 450 TB / Kuukausi	$ 0.022	$ 0.02	$ 0.0211
Yli 500 TB / Kuukausi	$ 0.021	$ 0.02	$ 0.0202

Taulukko 7. Eräkerros: Datavarastoinnin kustannusten vertailu

Dataprosessoinnin suhteen Microsoftin vertaaminen Amazonin ja Googlen palveluihin on haastavaa, koska Microsoft ei ole ilmoittanut kuinka tehokas on yksi dataintegraatioyksikkö, kun sen sijaan Googlen hinnoittelu perustuu vCPU ja RAM -käyttöön ja Amazonin prosessointiyksikkö on 4 vCPUta ja 16 gigatavua RAM-muistia. Näin ollen Amazonin ja Googlen hinnoittelu on vertailtavissa käyttäen dataprosessointiyksikköä vertailunpohjana, mutta tätä ennen on laskettava yhden prosessointiyksikön hinta Googlen palveluilla:

Eräprosessointi: 4*0,0759+16*0,0042684 = $ 0,3146944

Tietovirran prosessointi: 4*0,0759+16*0,0042684 = $ 0,3718944

	Amazon Firehose	**Google Pub/Sub (Eräprosessointi)**	**Google Pub/Sub (Tietovirtaprosessointi)**
Prosessointiyksikkö (4 vCPU / 16 GB)	$ 0.44	$ 0.3146944	$ 0.3718944

Taulukko 8. Dataprosessoinnin hinnaston vertailu

4.1.2 Palvelukerros

Kun eräkerros on prosessoinut datan, niin se säilötään käytettäväksi. Amazonin osalta data säilötään takaisin S3seen, Microsoftin osalta takaisin Azure Data Lake Storageen ja Googlen arkkitehtuurissa takaisin Cloud Storageen.

Palvelukerroksen tiedon tallentamisen kustannus rakentuu samanlailla kuin eräkerroksessa, jonka hinnasto on vertailtu taulukossa 7. Erona kuitenkin, että tällä kerroksella ei tallenneta niin paljon dataa, koska tieto on jo aggregoitu ja yhdistelty valmiiksi, joten datamassa on pienempi kuin raakadatassa.

Tiedonhyödyntäminen palvelukerrokselta tehdään Amazonin palveluista Athenalla, jolla hinta on viisi dollaria per luettu teratavu (Amazon Web Services, Inc 2019k). Microsoftin palveluista data tarjotaan Data Lake Analyticsin kautta, jossa hinnoittelu on kaksi dollaria per analyysiyksikkö (Analytics Unit). Yksi analyysiyksikkö vastaa kahta CPU-ydintä ja kuutta gigatavua RAM-muistia (Microsoft Corporation 2019k). Googlen osalta taas data tarjoillaan BigQueryn kautta, jolla Euroopan alueen (EU Multi-region) palvelinsalien hyödyntäminen maksaa viisi dollaria teratavua kohden, josta ensimmäinen teratavu on ilmainen. Kuitenkin jos haluaa käyttää Pohjois-Euroopan konesalia datan prosessointiin Big Querya, niin hinta lähes kaksinkertaistuu 9,20 dollariin per teratavu (Google LLC. 2019f). Amazonin ja Googlen osalta hinta on sama, mikäli hyödyntää Googlen Euroopan alueella olevaa useamman konesalin kokonaisuutta, mutta jos haluaa prosessoinnin tapahtuvan vain Pohjois-Euroopassa, niin joutuu maksamaan lähes tuplahinnan. Amazonin ja Googlen datamäärän

prosessointiin perustuvaa laskutusta on vaikea verrata Microsoftin prosessointipalveluun, joka taas perustuu valittuihin analyysiyksikkö määriin.

4.1.3 Nopeuskerros

Nopeuskerroksen käytettäväksi data tulee samasta palvelusta kuin eräkerroksella. Amazonin osalta data siis tulee Kinesis Firehoselta, josta se syötetään Kinesis Analyticsin läpi, jossa tietovirtadatalle voidaan tehdä prosessointia. Analyticsistä data luetaan Kinesis Firehoselta Amazon S3seen, josta se on käytettävissä Amazon Athenalla. Kinesis Firehosen kustannukset käytiin läpi jo eräkerroksessa, mutta Analyticsin kustannus koostuu käytetyistä Kinesis prosessointiyksiköistä (Kinesis Processing Unit). Yksi prosessointiyksikkö käsittää yhden vCPU-ytimen sekä 4 gigatavua RAM-muistia ja sen tuntikustannus on 0,12 dollaria. (Amazon Web Services, Inc 2019l). Amazon S3 ja Athenan kustannukset käytiin myös läpi eräkerroksessa.

Googlen osalta nopeuskerros koostuu Cloud Dataflowsta, joka prosessoi tietovirtadatan, Cloud Data Storagesta, johon reaaliaikaiset näkymät tallennetaan, ja BigQuerystä, jolla data voidaan lukea Cloud Data Storagesta. Näiden palveluiden kustannukset käytiin läpi eräkerroksen kohdalla.

Microsoftin arkkitehtuuri sen sijaan hyödyntää Azure Stream Analyticsiä tietovirtadatan prosessoimisessa, josta data tallennetaan Azure Data Lake Storageen reaaliaikaisiksi näkymiksi, josta Azure Data Lake Analytics voi hyödyntää dataa. Stream Analyticsin kustannus määräytyy tietovirtayksikköjen (Streaming Unit) määrästä, jotka maksavat 0,12 dollaria per yksikkö. Nämä yksiköt sisältävät CPUn ja muistin, mutta Microsoft ei ole määritellyt paljonko näitä on käytössä yhdessä tietovirtayksikössä. (Microsoft Corporation 2019l).

Nopeuskerroksen prosessointia on vaikea vertailla palvelutarjoajin kesken, koska jokaisen palveluntarjoajan kustannukset perustuvat erityyppisiin tekijöihin. Amazonilla kustannus syntyy prosessointiyksiköistä, joiden ytimien ja muistin määrä on kerrottu, mutta Googlen Cloud Data Flown ja Microsoftin Stream Analyticsin tapauksessa on epäselvää mikä on taustalla oleva laitteisto.

4.2 Kappa-arkkitehtuurin kustannukset

Amazonin Kappa-arkkitehtuurissa data luetaan sisään hyödyntäen Kinesis Firehosea, josta data ladataan Kinesis Analyticsiin prosessoitavaksi ja lopulta Firehosella siirretään reaaliaikaiseksi näkymäksi Amazon S3seen samanlailla kuin Lambda-arkkitehtuurissa. Kinesiksen kustannukset käsiteltiin Lambda-arkkitehtuurin osiossa, mutta Kappa-arkkitehtuurissa on huomioitava, että dataa prosessoidaan enemmän Analyticsillä kuin Lambda-arkkitehtuurissa johtuen, ettei Kappa-arkkitehtuurissa ole eräkerrosta. Amazon S3sesta dataa voidaan lukea hyödyntäen Amazon Athenaa, jonka kustannus oli viisi dollaria jokaista prosessoitua teratavua kohti.

Googlen Kappa-arkkitehtuuri on samanlainen kuin Lambda-arkkitehtuurin nopeuskerros, johtuen ettei Googlen pilvipalveluissa ollut oikein muuta pilvipalvelua, jolla samankaltainen arkkitehtuuri olisi toteutettavissa. Tämän arkkitehtuurin kustannukset käsiteltiin Lambda-arkkitehtuurin nopeuskerros-osiossa, mutta tässäkin on huomioitava, että Cloud Dataflown tulee käsitellä enemmän dataa tietovirtaprosessoinnissa, jolloin halvempaa eräprosessointia ei voida käyttää. Tämä nostaa prosessoinnin hintaa Googlen arkkitehtuurissa.

Microsoftin Kappa-arkkitehtuurissa data ladataan sisään käyttäen Event Hub-palvelua, josta data siirretään prosessoitavaksi Stream Analytics -palvelulle ja joka tallentaa datan lopulta Azure Data Lake Storageen. Microsoftin arkkitehtuurinkin datan sisäänluku, tallentaminen ja prosessointi ovat samankaltaiset kuin Lambda-arkkitehtuurin nopeuskerroksessa ja näiden kustannukset on käsitelty kyseisessä osiossa aikaisemmin tässä tutkielmassa. Data voidaan hyödyntää myös samanlailla suoraan Data Lake Storagesta käyttäen Data Lake Analytics palvelua.

Kappa-arkkitehtuurin dataprosessoinnin vertailussa on sama ongelma kuin Lambda-arkkitehtuurissa, että pilvitarjoajilla on hyvin erilaiset kustannusmallit prosessointipalveluissaan. Sen sijaan datan säilyttämisen hintoja reaaliaikaisissa voidaan verrata Kappa-arkkitehtuurissa.

5 Pohdinta

Tässä tapaustutkimuksessa tutkittiin massadata-arkkitehtuureja julkipilvessä vertaamalla Lambda- ja Kappa-arkkitehtuuria Amazonin, Googlen ja Microsoftin pilvialustoilla. Tapaustutkimuksella pyrittiin löytämään vastaus tutkimuskysymykseen ”Kuinka muodostaa nykyaikainen massadata-arkkitehtuuri pilvialustoilla?”. Samalla tapaustutkimuksella etsittiin vastausta seuraavaan tutkimusongelmaan: ”Mitkä ominaisuudet tulee huomioida vertailtaessa pilvipalveluita massadata-arkkitehtuurissa?”

Tutkimuskysymykseen vastattiin tässä tutkimuksessa luomalla Lambda- ja Kappa-arkkitehtuurit julkipilveen. Näiden arkkitehtuurien pohjalta tutkittiin esiin nousseita tutkimusongelmia. Tapaustutkimuksessa käytiin läpi palvelut, jotka voidaan valita arkkitehtuuria luotaessa. Data-analyysin reaaliaikaisuuteen vastattiin hyödyntämällä Lambda-arkkitehtuurin nopeuskerrosta ja Kappa-arkkitehtuurin luonnetta käyttää pelkästään reaaliaikaisia näkymiä. Tapaustutkimuksen tuloksissa käytiin läpi sitä, millaisia kustannuksia syntyy käytettäessä näitä palveluita, vaikkakin joka kohdassa palvelut eivät olleet täysin vertailukelpoisia kustannusrakenteidensa vuoksi.

Arkkitehtuuriratkaisujen vertailun suurin haaste olikin juuri, että palveluiden kustannusrakenteet eivät olleet samankaltaisia ja joissain tapauksissa vaikeasti ymmärrettäviä siltä osin, että hinnoittelun perusteena oli vain yksiköt, joita ei voi verrata laitteisto-ominaisuuksiin.

5.1 Arkkitehtuurien yhtäläisyydet ja erot

Kappa-arkkitehtuuri on hyvin samankaltainen kuin Lambda-arkkitehtuuri, sillä erotuksella, että eräkerrosta ei ole, vaan datat käsitellään suoraan reaaliaikaisiin näkymiin ja tarvittaessa uudelleen prosessoidaan tietovirtadatan lukijalta. Eräkerroksen puuttuminen yksinkertaistaa Kappa-arkkitehtuuria ja erilaisia palveluita ei tarvita niin montaa. Kappa-arkkitehtuuri saatiin luotua jokaisella julkipilven tarjoajalla käyttämällä kolmea palvelua, kun Lambda-arkkitehtuurissa tarvittiin aina enemmän. Samoin Kappa-arkkitehtuurissa dataa ei tarvitse monistaa kahteen eri dataputkeen vaan prosessointi on suoraviivaisempaa.

5.2 Pilvitarjoajien yhtäläisyydet ja erot

Pilvitarjoajien osalta palvelutarjonta oli hyvin samankaltaista. Jokaiselta löytyy esimerkiksi samantyyppinen tietovarasto, mutta esimerkiksi objektivaraston hyödyntämiseen ei Googlelta löydy aivan samankaltaista palvelitonta palvelua kuin Amazonilta ja Microsoftilta, vaan tähän on käytettävä BigQuery-tietovarastoa, josta löytyy ominaisuus hyödyntää dataa suoraan objektivarastosta. Toinen arkkitehtuurinen eroavaisuus on Amazonin tietovirtadatan prosessoinnissa, jossa on hyödynnettävä kahta palvelua: yhtä datan prosessoimiseen ja toista datan siirtämiseen prosessointipalveluun ja sieltä pois.

5.3 Hinnoittelu

Tapaustutkimuksen hinnoittelun tarkastelussa syntyi eniten ongelmia tätä tutkielmaa tehdessä. Hinnastot ovat kyllä saatavilla helposti, mutta niiden vertailu ei joissain määrin ilman konkreettisia testauksia ole kovinkaan mahdollista. Hinnoittelussa Amazonilla tuntui olevan kaikkein läpinäkyvin hinnoittelu, kun palveluissa kerrottiin mitä tietyt prosessointiyksiköt tarkkaan ottaen tarkoittavat ytimien ja muistien osalta. Microsoftilla sen sijaan tuntui olevan näistä kolmesta kaikkein eniten prosessointiyksikköjä, joiden laitteistokuvausta ei lainkaan kerrottu. Tämä vaikeutti palveluiden vertailua keskenään, koska suoraan näiden prosessointiyksikköjen vertailu ei tarjoa koko totuutta. Huomioitavaa on myös, että laitteistokokonaisuuksien vertailu ei aina kerro kuitenkaan koko totuutta, koska palvelut ovat eri tyyppisiä luonteeltaan ja ne hyödyntävät prosessointikapasiteettia eritavoin.

Siinä missä prosessointiyksikköjä oli vaikea verrata, niin datan varastoinnin kustannuksia pystyttiin vertailemaan keskenään. Tämän osalta Googlella on halvin ratkaisu, vaikka sen hinta ei skaalaudu datamassan mukaan, kuten Amazonilla ja Microsoftilla.

5.4 Vaihtoehtoiset pilvipalvelut ja toteutukset

Tapaustutkimuksessa valitut palvelut ovat yhden kaltainen kokonaisuus luoda Lambda- ja Kappa-arkkitehtuuri. Loppujen lopuksi erilaisia kombinaatioita on lähemmäs rajaton määrä, miten nämä arkkitehtuurit voitaisiin koostaa. Jokaiselta näistä kolmesta julkipilven

tarjoajalta löytyy esimerkiksi Apachen avoimen lähdekoodin massadata-palvelukokonaisuudet, joita esimerkiksi Nathan Marz käyttää kirjassaan esimerkkinä Lambda-arkkitehtuurille.

Toinen selkeä arkkitehtuurinen vaihtoehto olisi hyödyntää relationaalisia tietovarastoja datan tarjoamiseen Lambdan palvelukerroksella ja Kappa-arkkitehtuurissa, varsinkin jos data on rakenteellisessa muodossa. Jokaiselta palveluntarjoajalta löytyy MPP-tietovarastot, joissa datan prosessointi ja lukeminen on nopeampaa kuin perinteisissä tietovarastoissa. Näistä jokaisesta löytyy myös mahdollisuus lukea dataa suoraan objektivarastosta käyttäen ulkoista taulua (external table), jolloin datan saaminen saataville on nopeampaa kuin perinteisen eräajon hyödyntäminen latauksissa. Relationaalisten tietovarastojen lisäksi myös NoSQL-kantojen käyttö olisi mahdollista datan säilömistä varten.

Yhtenä vaihtoehtona pilvipalveluita hyödyntäessä on käyttää useampaa pilvitarjoajaa samalla kertaa ja valita näin jokaiselta palveluntarjoajalta parhaat palvelut ratkaisemaan arkkitehtuuriset ja kustannukselliset ongelmat. Tämä ei kuitenkaan massadata-ratkaisussa välttämättä ole kustannuksellisesti paras ratkaisu, koska pilvipalvelun tarjoajat veloittavat verkkoliikenteestä, joka on ulospäin heidän pilvi-infrastruktuurista. Tämä ei ole kovin suuri kustannus per datamäärä, mutta massadatan suuren datamassan myötä se voi kasvaa huomattavasti. Toinen haittapuoli tässä on, että tällöin data liikkuu konesalista toiseen ja tästä syntyy myös viivettä dataan. Tämä on varsinkin reaaliaikaisia analyysiratkaisuja luotaessa otettava tarkasti huomioon. Lisäksi palvelujen hallinta monimutkaistuu, koska palvelut ovat tällöin jaoteltuna eri infrastruktuureihin.

6 Yhteenveto

Tässä tutkielmassa käsiteltiin massadata-arkkitehtuureja ja niiden muodostamista julkipilveen. Tutkielmassa käsiteltiin massadatan, pilvipalveluiden ja massadata-arkktehtuurien teoreettista taustaa narratiivisella kirjallisuuskatsauksella ja toteutettiin tapaustutkimus, jossa tutkittiin Lambda- ja Kappa-arkkitehtuurien luontia julkipilven palveluilla. Julkipilven tarjoajista tapaustutkimuksessa käytettiin tapauksina Amazonia, Googlea ja Microsoftia. Tapaustutkimuksen tuloksissa keskityttiin tutkimaan näiden luotujen arkkitehtuurin kustannuksia hyödyntäen palveluntarjoajien hinnastoja.

Nykyaikana datan jatkuva kasvu on tuonut haasteet sen säilömiselle ja prosessoimiselle, johon ei enää voida hyödyntää perinteisiä menetelmiä. Näin ollen on suunniteltava massadata-arkkitehtuureja, jotka pystyvät vastaaman tähän haasteeseen. Lambda- ja Kappa-arkkitehtuurit tarjoavat tähän hyvän lähtökohdan, josta löytyy mahdollisuus vastata erityyppisiin analyyttisiin tarpeisiin.

Lambda-arkkitehtuurin osalta tässä tutkielmassa käytiin läpi erä-, palvelu- ja nopeuskerros, joilla vastataan sekä erä- kuin reaaliaikaisiin analyysitarpeisiin. Tämän arkkitehtuurin pohjalta rakennettiin kolmen palvelun tarjoajan palveluista arkkitehtuurit, joita verrattiin keskenään ja lisäksi myös Kappa-arkkitehtuuriin, joka oli toinen tutkielman tapaustutkimuksessa tarkasteltu massadata-arkkitehtuuri. Kappa-arkkitehtuurin pohjalta luotiin myös pilvipalvelukokonaisuudet, joiden ominaisuuksia verrattiin keskenään ja Lambda-arkkitehtuuriin.

Tapaustutkimuksen tuloksissa vertailtiin arkkitehtuurien kustannuksia niin datan säilömisen kuin prosessoinnin osalta. Tässä osiossa oli haasteena, ettei palveluntarjoajien palvelut olleet täysin vertailukelpoisia, koska niiden kustannusrakenne oli täysin erilainen. Toinen ongelma oli epäselvät hinnastot, joissa kustannuksen pohjana oli yksikköjä, joilla ei ollut mitään laitteistomääritelmiä. Tämä teki vertailussa paikoittain mahdotonta, koska palveluita ei voitu vertailla eri yksikköjen takia.

Tapaustutkimuksen pohdinnoissa käytiin läpi tutkielman tutkimuskysymys ja -ongelmat ja tarkasteltiin, että niihin on löytynyt vastaus tutkielmasta. Lisäksi tarkasteltiin arkkitehtuurien

ja pilvipalveluiden eroja, sekä vaihtoehtoisia ratkaisumahdollisuuksia arkkitehtuurien toteuttamiseksi.

Lähteet

Akhtar, Syed. 2018. *Big Data Architect's Handbook.* UK, Birmingham: Packt Publishing. ISBN 978-1-78883-582-4.

Amazon Web Services, Inc. 2019a. *What is AWS.* Haettu osoitteesta: https://aws.amazon.com/what-is-aws/

Amazon Web Services, Inc. 2019b. *Amazon Kinesis Data Firehose*. Haettu osoitteesta: https://aws.amazon.com/kinesis/data-firehose/

Amazon Web Services, Inc. 2019c. *Amazon S3*. Haettu osoitteesta: https://aws.amazon.com/s3/

Amazon Web Services, Inc. 2019d. *AWS Glue*. Haettu osoitteesta: https://aws.amazon.com/glue/

Amazon Web Services, Inc. 2019e. *Amazon Athena*. Haettu osoitteesta: https://aws.amazon.com/athena/

Amazon Web Services, Inc. 2019f. *Amazon Kinesis Data Analytics*. Haettu osoitteesta: https://aws.amazon.com/kinesis/data-analytics/

Amazon Web Services, Inc. 2019g. *Amazon DynamoDB.* Haettu osoitteesta: https://aws.amazon.com/dynamodb/

Amazon Web Services, Inc. 2019h. *Amazon Kinesis Data Firehose Pricing*. Haettu osoitteesta: https://aws.amazon.com/kinesis/data-firehose/pricing/

Amazon Web Services, Inc. 2019i. *Amazon S3 Pricing*. Haettu osoitteesta: https://aws.amazon.com/s3/pricing/

Amazon Web Services, Inc. 2019j. *AWS Glue Pricing*. Haettu osoitteesta: https://aws.amazon.com/glue/pricing/

Amazon Web Services, Inc. 2019k. *Amazon Athena Pricing*. Haettu osoitteesta: https://aws.amazon.com/athena/pricing/

Amazon Web Services, Inc. 2019l. *Amazon Kinesis Data Analytics Pricing*. Haettu osoitteesta: https://aws.amazon.com/kinesis/data-analytics/pricing/

Amazon Web Services, Inc. 2019m. *Amazon DynamoDB: Pricing for On-Demand Capacity.* Haettu osoitteesta: https://aws.amazon.com/dynamodb/pricing/on-demand/

Amazon Web Services, Inc. 2017. *Unite Real-Time and Batch Analytics Using the Big Data Lambda Architecture, Without Servers!*. Haettu osoitteesta: https://aws.amazon.com/blogs/big-data/unite-real-time-and-batch-analytics-using-the-big-data-lambda-architecture-without-servers/

Black, B. 2009. *EC2 Origins.* Haettu osoitteesta: http://blog.b3k.us/2009/01/25/ec2-origins.html

Dixon, James. 2010. *Pentaho, Hadoop, and Data Lakes.* Haettu osoitteesta: https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/

Eriksson, Päivi ja Katri Koistinen. 2005. *Monenlainen tapaustutkimus*. Kerava: Savion Kirjanpaino Oy. ISBN 951-698-123-2.

Forbes. 2018. *How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read.* Haettu osoitteesta: https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read

Gama, Joao, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy, Abdelhamid Bouchachia. 2013. *A Survey on Concept Drift Adaptation*. ACM Comput. Surv. 1, 1, Article 1 (January 2013), 35 pages. Haettu osoitteesta: http://eprints.bournemouth.ac.uk/22491/1/ACM%20computing%20surveys.pdf

Golden, Bernard. 2013. *Amazon Web Services for Dummies.* USA, New Jersey: John Wiley & Sons, Inc. ISBN 978-1-118-65198-8.

Google LLC. 2019a. *Cloud locations*. Haettu osoitteesta: https://cloud.google.com/about/locations/

Google LLC. 2019b. *Cloud Pub/Sub*. Haettu osoitteesta: https://cloud.google.com/pubsub/

Google LLC. 2019c. *Cloud Dataflow*. Haettu osoitteesta: https://cloud.google.com/dataflow/

Google LLC. 2019d. *Cloud Storage.* Haettu osoitteesta: https://cloud.google.com/storage/

Google LLC. 2019e. *Cloud BigTable.* Haettu osoitteesta: https://cloud.google.com/bigtable/

Google LLC. 2019f. *Cloud BigQuery*. Haettu osoitteesta: https://cloud.google.com/bigquery/

Google LLC. 2019g. *Understanding Cloud Bigtable performance.* Haettu osoitteesta: https://cloud.google.com/bigtable/docs/performance

Hammergren, Thomas ja Alan Simon. 2009. *Data Warehousing for Dummies*. USA, New Jersey: Wiley Publishing, Inc. ISBN 978-0-470-40747-9

Hurwitz, Judith, Alan Nugent, Fern Halper ja Marcia Kaufman. 2013. *Big Data For Dummies.* USA, New Jersey: John Wiley & Sons, Inc. ISBN 978-1-118-64417-1.

John, Tomcy ja Pankaj Misra. 2017. *Data Lake for Enterprises.* UK, Birmingham: Packt Publishing. ISBN 978-1-78728-134-9.

Kreps, Jay. 2014. *Questioning the Lambda Architecture.* Haettu osoitteesta: https://www.oreilly.com/ideas/questioning-the-lambda-architecture

Krishnan, S.P.T. ja Jose Gonzales. 2015. *Building Your Next Big Thing with Google Cloud Platform.* USA, CA: Apress. ISBN 978-1-4842-1004-8

Kritikos, Kyriakos ja Paweł Skrzypek. 2018. "A Review of Serverless Frameworks." *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion).* doi: 10.1109/UCC-Companion.2018.00051

Laine, Markus, Jarkko Bamberg ja Pekka Jokinen. 2007. *Tapaustutkimuksen taito*. Helsinki: Gaudeamus. ISBN 978-952-495-032-9.

Laney, Doug. 2011. *Application Delivery Strategies*. Haettu osoitteesta: https://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf

McDonald, Paul. 2008. *Introducing Google App Engine + our new blog.* Haettu osoitteesta: http://googleappengine.blogspot.com/2008/04/introducing-google-app-engine-our-new.html

Microsoft Corporation. 2019a. *What is Azure.* Haettu osoitteesta: https://azure.microsoft.com/en-us/overview/what-is-azure/

Microsoft Corporation. 2019b. *Event Hubs.* Haettu osoitteesta: https://azure.microsoft.com/en-us/services/event-hubs/

Microsoft Corporation. 2019c. *Azure Data Lake Storage.* Haettu osoitteesta: https://azure.microsoft.com/en-us/services/storage/data-lake-storage/

Microsoft Corporation. 2019d. *Data Factory.* Haettu osoitteesta: https://azure.microsoft.com/en-us/services/data-factory/

Microsoft Corporation. 2019e. *Data Lake Analytics.* Haettu osoitteesta: https://azure.microsoft.com/en-us/services/data-lake-analytics/

Microsoft Corporation. 2019f. *Azure Cosmos DB Documentation.* Haettu osoitteesta: https://docs.microsoft.com/en-us/azure/cosmos-db/

Microsoft Corporation. 2019g. *Azure Stream Analytics.* Haettu osoitteesta: https://azure.microsoft.com/en-us/services/stream-analytics/

Microsoft Corporation. 2019h. *Event Hub Pricing.* Haettu osoitteesta: https://azure.microsoft.com/en-us/pricing/details/event-hubs/

Microsoft Corporation. 2019i. *Azure Data Lake Storage Gen2 Pricing.* Haettu osoitteesta: https://azure.microsoft.com/en-us/pricing/details/storage/data-lake/

Microsoft Corporation. 2019j. *Data Pipeline Pricing.* Haettu osoitteesta: https://azure.microsoft.com/en-us/pricing/details/data-factory/data-pipeline/

Microsoft Corporation. 2019k. *Data Lake Analytics Pricing.* Haettu osoitteesta: https://azure.microsoft.com/en-us/pricing/details/data-lake-analytics/

Microsoft Corporation. 2019l. *Data Lake Analytics Pricing.* Haettu osoitteesta: https://azure.microsoft.com/en-us/pricing/details/data-lake-analytics/

Microsoft Corporation. 2019m. *Cosmos DB Pricing.* Haettu osoitteesta: https://azure.microsoft.com/en-gb/pricing/details/cosmos-db/

Microsoft Corporation. 2018a. *Azure SQL Data Warehouse – Massively parallel processing (MPP) architecture*. Haettu osoitteesta: https://docs.microsoft.com/en-us/azure/sql-data-warehouse/massively-parallel-processing-mpp-architecture

Microsoft Corporation. 2018b. *Big data architectures*. Haettu osoitteesta: https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/

Salminen, Ari. 2011. *Mikä kirjallisuuskatsaus? Johdatus kirjallisuuskatsauksen tyyppeihin ja hallintotieteellisiin sovelluksiin*. Vaasan yliopiston Julkaisuja.

Salo, Immo. 2014. *Big Data & Pilvipalvelut.* Jyväskylä: Docendo Oy. ISBN 978-952-291-032-5.

Statista. 2019. *Current and planned usage of public cloud platform services running applications worldwide as of 2019*. Haettu osoitteesta: https://www.statista.com/statistics/511467/worldwide-survey-public-coud-services-running-application/

Wallace, Danny P. 2007. *Knowledge Management: Historical and Cross-Disciplinary Themes*, 1–14. UK, London: Libraries Unlimited. ISBN 978-1-59158-502-2.

Webber-Cross, Geoff. 2014. *Learning Microsoft Azure*. UK, Birmingham: Packt Publishing. ISBN 978-1-78217-337-3.