Vaihtoehtoisen kanta-asiakastunnistautumisen taustajärjestelmän toteutus Musti Groupille

Joni Juntto

Haaga-Helia ammattikorkeakoulu

Tradenomi

2023

Tiivistelmä

Tekijä(t)
Joni Juntto

Tutkinto
Tradenomi

Raportin/Opinnäytetyön nimi
Vaihtoehtoisen kanta-asiakastunnistautumisen taustajärjestelmän toteutus Musti Groupille

Sivu- ja liitesivumäärä
22 + 1

Tämän toiminnallisen opinnäytetyön tavoitteena oli suunnitella ja toteuttaa vaihtoehtoinen kanta-asiakkuuden tunnistautumismenetelmä Musti Groupin Suomen ja Ruotsin myymälöihin. Tässä opinnäytetyössä vaihtoehtoinen kanta-asiakkuuden tunnistautumismenetelmä tarkoittaa ajo- tai henkilökortin näyttämistä myymälän henkilökunnalle. Uusi kanta-asiakkuuden tunnistautumismenetelmä on vaihtoehtoinen, koska vanha menetelmä jää edelleen käyttöön uuden rinnalle. Toteutetussa menetelmässä asiakas näyttää ajo- tai henkilökorttinsa viivakoodin kassahenkilölle, joka lukee viivakoodin järjestelmään. Järjestelmä luo yksisuuntaisella salauksella henkilötunnuksesta salauksen, jolla asiakas voi jatkossa tunnistautua.

Opinnäytetyön aiheena on vain taustajärjestelmän kehitys. Uusi vaihtoehtoinen kanta-asiakkuuden tunnistautumismenetelmä on tarpeellinen, koska olemassa olevat menetelmät ovat hitaita ja niiden tietosuoja ei ole täydellinen. Tietosuojan ongelma nykyisessä menetelmässä on, että kanta-asiakkaat joutuvat sanomaan henkilötietojaan ääneen muiden mahdollisten asiakkaiden kuullen. Kehityksessä käytettiin Python ohjelmointikieltä ja Serverless Framework ohjelmistokehityskehystä.

Opinnäytetyö on onnistunut, kun kehitetty menetelmä on nopeampi, turvallisempi ja käyttäjäystävällisempi, kuin olemassa olevat menetelmät. Opinnäytetyölle asetetut tavoitteet täyttyivät ja menetelmä on jo käytössä kahdessa Musti Groupin myymäläketjuista, joista tullut palaute on ollut erittäin positiivista. Opinnäytetyö tehtiin kevään 2023 aikana.

Asiasanat
Kanta-asiakasjärjestelmät, tietosuoja, ohjelmointi, serverless

Sisällys

1 Johdanto

Musti Group on Pohjoismaiden johtava lemmikkieläinhoidon asiantuntija, joka toimii Suomessa, Ruotsissa ja Norjassa. Syyskuussa 2022 Musti Groupilla on ollut 335 myymälää ja 391,1 miljoonan euron liikevaihto päättyneellä tilikaudella. Musti Groupiin kuuluvat brändit ovat Musti, Musti ja Mirri, Peten koiratarvike, Arken Zoo, Djurmagazinet ja Vetzoo.

Opinnäytetyöni toteutan Musti Group Oy:n toimeksiannolla toiminnallisena opinnäytetyönä. Toimeksiantona yritykseltä opinnäytetyöhön on suunnitella ja toteuttaa uusi tapa tunnistautua kanta-asiakkaaksi Musti Groupin myymälöissä, joka on nopeampi, turvallisempi ja käyttäjäystävällisempi. Uudeksi tavaksi tunnistautua kanta-asiakkaaksi on valittu henkilö- tai ajokortin skannaaminen järjestelmään ja sen salaaminen käyttäen yksisuuntaista salausta. Salaus muodostetaan käyttäen SHA-512 algoritmiä ja taustajärjestelmää kehitetään Python ohjelmointikielellä. Uusi tapa tunnistautua kanta-asiakkaaksi kehitetään jo olemassa olevien tapojen rinnalle.

Tämä opinnäytetyö keskittyy Musti Groupin myymälöissä käytettävän kanta-asiakasjärjestelmän, taustajärjestelmän uuden ominaisuuden kehittämiseen. Opinnäytetyö on rajattu koskemaan vain taustajärjestelmän suunnittelua ja kehitystä. Lisäksi projektiin kuuluu käyttöliittymän kehitystä, joka on jätetty opinnäytetyön ulkopuolelle. Tämä opinnäytetyö käsittelee myös uuden kanta-asiakkuuden tunnistautumismenetelmän valintaan liittyviä tekijöitä ja päätöksiä, vaikka tehty päätös ei kuulu opinnäytetyön aiheisiin. Tässä opinnäytetyössä keskitytään vain Suomen ja Ruotsin myymälöissä tapahtuvaan kanta-asiakastunnistautumiseen.

Nykyisessä kanta-asiakkuuden tunnistautumismenetelmässä ongelmia aiheuttavat sen hitaus ja puutteellinen tietosuoja. Tällä hetkellä asiakkaat tunnistautuvat myymälöissä antamalla asiakasnumeron, sähköpostiosoitteen, puhelinnumeron tai Ruotsin tapauksessa henkilötunnuksen. Vanhan menetelmän hitaus johtuu muun muassa asiakkaiden useista eri sähköpostiosoitteista ja juuri sitä kanta-asiakkuuteen käytettyä on vaikea muistaa. Vaikka yksittäisen asiakkaan kannalta aika ei ole merkittävä, se kertyy nopeasti, sillä 80 % Musti Groupin asiakkaista on kanta-asiakkaita. Nykyinen tunnistautumistapa ei myöskään takaa kanta-asiakkaiden tietosuojan säilymistä, koska kanta-asiakkaat joutuvat sanomaan henkilötietojaan ääneen muiden mahdollisten asiakkaiden kuullen.

Opinnäytetyössä pyritään vastaamaan seuraaviin kysymyksiin: miten henkilötunnusta voidaan käyttää tunnistautumisessa tietoturvallisesti, miten henkilötunnuksen käyttö tunnistautumiseen toteutetaan taustajärjestelmän puolella ja miten valittuihin teknologioihin on päädytty. Projektin aikana opin ymmärtämään syvällisemmin SHA-algoritmeista, henkilötunnusten käytöstä ja henkilökohtaisten tietojen salaamisesta.

1.1 Käsitteet

DRY	Ohjelmointiin liittyvä periaate, joka tulee sanoista “Do Not Repeat Yourself”. DRY-periaate kehottaa ohjelmoijaa välttämään toisteisuutta koodissa
JSON	JavaScript Object Notation. Avoimen standardin tiedostomuoto.
GIT	Hajautettu versiohallintajärjestelmä, joka on suunniteltu helpottamaan ohjelmistokehityksen seurantaa ja yhteistyötä.
CRM	Customer Relationship Management. Asiakassuhteiden hallinta yrityksessä. Viittaa strategioihin, käytäntöihin, tekniikoihin ja työkaluihin, joita yritykset käyttävät asiakkaidensa kanssa tapahtuvan vuorovaikutuksen hallintaan ja analysointiin
CRM-järjestelmä	Auttaa yrityksiä ymmärtämään asiakkaidensa tarpeita paremmin ja tarjoamaan heille räätälöityjä palveluja ja tuotteita. Sisältää usein asiakastiedot, myynnin seurannan ja raportointityökaluja.

2 Kanta-asiakkuuden tunnistautumismenetelmät

Musti Groupin Kanta-asiakasohjelman tavoite on saada asiakas keskittämään ostoksensa ja luoda pitkäaikaisia asiakkuuksia. Kanta-asiakkaaksi tunnistautuminen halutaan pitää vaivattomana ja nopeana asiakkaille, jotta mahdollisimman moni asiakas liittyisi kanta-asiakkaaksi.

Tässä luvussa käsitellään Musti Groupin pohtimia kanta-asiakkuuden tunnistautumismenetelmiä, sekä nykyistä tapaa tunnistautua kanta-asiakkaaksi. Kappaleessa käsitellään myös uudeksi tunnistautumismenetelmäksi valitun menetelmän kriteereitä, sekä suunnittelua.

Vaikka Musti Group on valinnut kehitettäväksi tietyn kanta-asiakkuuden tunnistautumismenetelmän, on tärkeää tutkia myös muita vaihtoehtoisia menetelmiä ja niiden ominaisuuksia. Muutamia erilaisia kanta-asiakkuuden tunnistusmenetelmiä Musti Groupin myymälöihin voisivat olla esimerkiksi:

- QR-koodi kanta-asiakassovellukseen. Tällä tavalla jokainen asiakas kantaisi vaadittavaa tunnistautumismenetelmää aina mukanansa. Sovelluksen huonoja puolia on sen kehityksen ja ylläpidon kustannukset ja asiakkaiden ennakkoluulot sovelluksia kohtaan.
- Ajokortin tai henkilökortin skannaaminen. Henkilöllisyystodistus on yleensä asiakkailla mukana ja skannaaminen ei vie kauan aikaa asiakkaalta tai henkilökunnalta. Henkilötunnuksen käytössä on omat tietoturvariskinsä, jotka on otettava huomioon suunnitteluvaiheessa.
- Puhelinnumeron tai sähköpostin kertominen myymälässä. Tämä vaihtoehto on tämänhetkinen tunnistautumismenetelmä Musti Groupin myymälöissä. Sen huonoina puolina on esimerkiksi sen heikko tietoturva, jos asiakkaan tiedot kuulee joku muu asiakas (Musti Group 2023 a).

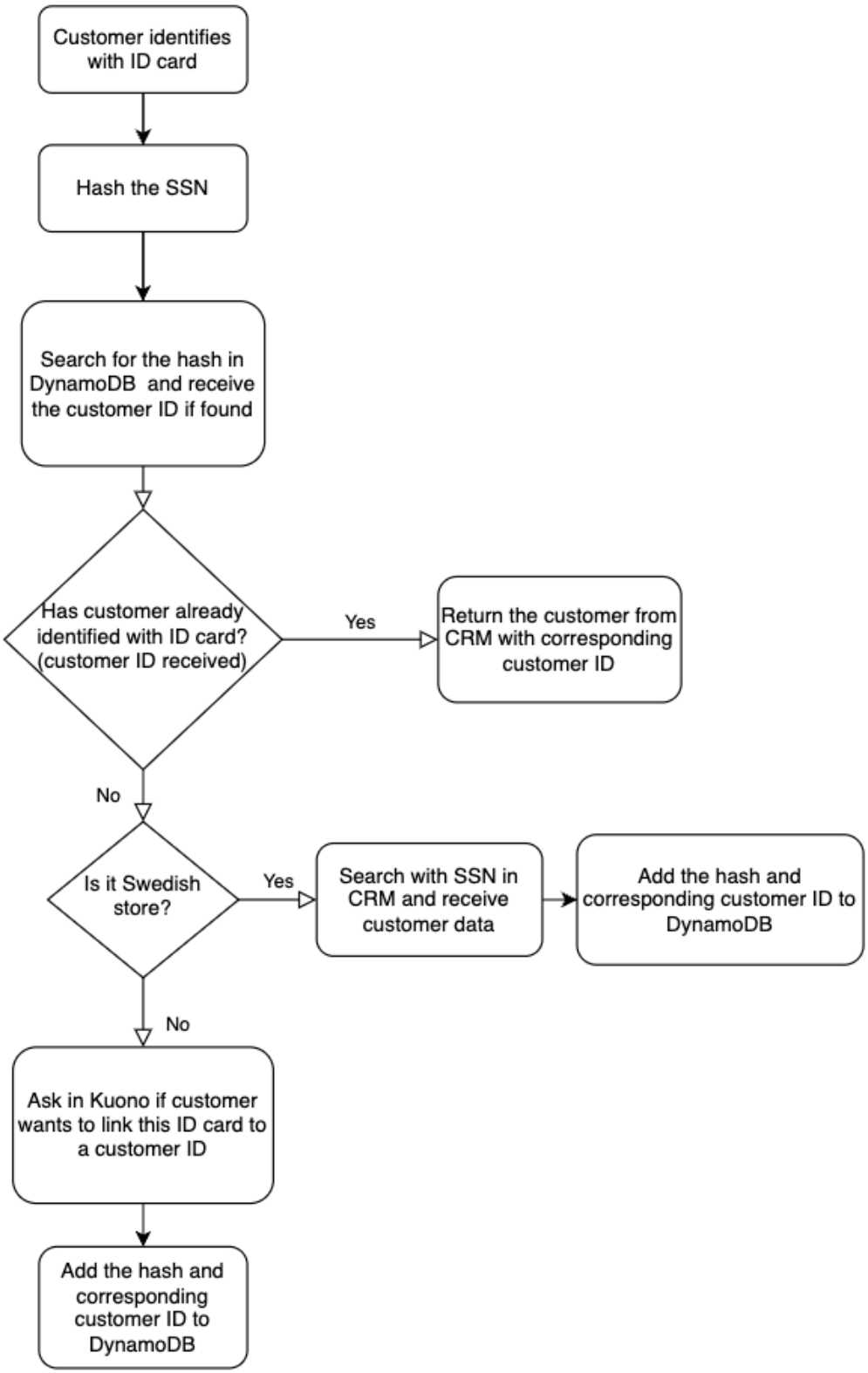
Musti Groupilta saadussa toimeksiannossa, sekä johdannossa esiteltyjen vaatimusten vuoksi uudeksi kanta-asiakkuuden tunnistautumismenetelmäksi valittiin ajo- tai henkilökortin skannaaminen kanta-asiakasjärjestelmään ja siitä saatavan henkilötunnuksen hajauttaminen käyttäen SHA algoritmia. Tämän menetelmän käyttöönotto on helpompaa verrattuna muihin menetelmiin, sillä se vaatii vain vanhan järjestelmän jatkokehitystä Tällä tavalla täysin uutta kehitystä ei vaadita ja olemassa olevaa runkoa voidaan hyödyntää. Vahvaa tukea tälle menetelmälle tulee myös siitä, että 98 % Ruotsalaisista vähittäiskaupan ketjuista on jo ottanut samankaltaisen järjestelmän käyttöönsä (Musti Group 2023 b). Joten asiakkaat Ruotsissa ovat jo tottuneita ottamaan oman henkilökorttinsa esille kassalle tultaessa. (Musti Group 2023 b)

2.1 Uusi tunnistautumismenetelmä

Musti Groupin Suomen ja Ruotsin myymälöihin vaihtoehtoiseksi tunnistautumismenetelmäksi on valittu ajo- tai henkilökortti, joista työntekijä skannaa kortista löytyvän viiva- tai QR-koodin kanta-asiakasjärjestelmään. Tämä tapa on jo käytössä Suomessa esimerkiksi Stadium -ketjulla, jonka nettisivuilla kerrotaan: ”Voit tunnistautua myymälässä henkilöllisyystodistuksella, jos olet liittänyt henkilötunnuksesi jäsenyydellesi.” (Stadium s.a.). Samankaltaisesti henkilötunnuksen salaus on toteutettu jo Suomessa Motonet ketjun liikkeissä. Motonetin nettisivuilla todetaan: ”Järjestelmään tallennettu asiakastunniste on salattu yksisuuntaisella salauksella, jonka vuoksi sitä ei voi avata takaisin henkilötunnukseksi.” (Motonet 2023). Projektia suunnitellessa huomioitiin myös vuonna 2023 käyttöön otettavat uudet henkilökortit ja niiden vaikutukset tunnistautumismenetelmän kehittämiseen. Näissä henkilökorteissa ei ole enää viivakoodia, vaan se on korvattu QR-koodilla (Poliisi 2023). Taustajärjestelmän vastaanottama data tulee olla sama, välittämättä siitä, luetaanko henkilötunnus vanhan- vai uudenmallisesta henkilökortista.

Uusi tunnistautumismenetelmä poistaa tarpeen uusien fyysisten kanta-asiakaskorttien luomiseen ja teettämiseen, mikä vähentää materiaalikulutusta. Valittu tunnistautumismenetelmä on myös asiakkaille vaivattomampi, koska heidän ei tarvitse muistaa pitää mukanaan erillistä korttia vain Musti Groupin myymälöissä asiointia varten.

Opinnäytetyön määrittelyvaiheessa laadittua vaatimusluetteloa hyödynnettiin kehityksen suunnittelussa päätösten tekemiseen. Suunnitteluvaiheen tuotteena tässä projektissa on vuokaavio, joka kuvaa tilannetta, jossa asiakas tunnistautuu näyttämällä henkilöllisyystodistusta myymälätyöntekijälle. Suunnittelu vaihe tehtiin yhdessä Musti Groupin työntekijöiden kanssa.



Kuva 1 New customer identification flow in stores (Musti Group 2023)

3 Henkilötunnuksen salaaminen

Tässä luvussa käsitellään henkilötunnuksen käyttöä yksilöivän tunnisteen luomisessa Musti Groupin taustajärjestelmässä. Kappaleessa esitellään SHA-algoritmin käyttöä henkilötunnuksen salauksessa, sekä selvitetään henkilötunnuksen perusrakenne ja mahdolliset hajautusfunktion käyttöön liittyvät ongelmat.

SHA on lyhenne sanoista secure hashing algorithm, ja nämä funktiot kuuluvat kryptograafisiin tiivistefunktioihin. Kaikki SHA algoritmit ovat iteratiivisia, yksisuuntaisia hajautusfunktioita, jotka ottavat vastaan viestin, jonka pituus voi olla mitä tahansa, ja tuottavat siitä kiinteän pituisen merkkijonon, jota kutsutaan hajautusarvoksi, hajautteeksi, tiivisteeksi tai hashiksi. Tämä hajautusarvo edustaa alkuperäistä viestiä ja sen pituus riippuu käytetystä algoritmista. Tiivistefunktioita käytetään yleisesti kryptografian sovelluksissa, jotka liittyvät datan eheyteen ja aitouteen. (National Institute of Standards and Technology, 2015.)

SHA tiivistefunktiot ovat turvallisia, koska laskennallisesti on haastavaa löytää sellaista viestiä, joka vastaa annettua hajautusarvoa, tai löytää 2 erilaista viestiä, jotka tuottaisivat saman hajautusarvon. Jokaisella tiivistefunktiolla on kaksi vaihetta: esikäsittely ja tiivistyslaskenta. Esikäsittelyssä viestiä täydennetään, jaetaan lohkoihin ja alustetaan arvot tiivistyslaskentaa varten. Tiivistyslaskennassa muodostetaan viestiaikataulu täydennetystä viestistä ja luodaan toistuvasti sarja tiivistearvoja käyttäen funktioita, vakioita ja viestin operaatioita. Lopullinen arvo on haluttu hajautusarvo. (National Institute of Standards and Technology, 2015)

Tiivistefunktiot eroavat toisistaan pääasiassa lopputiivisteen turvallisuudessa. Esimerkiksi NIST, eli National Institute of Standards and Technology, ei enää suosittele SHA-1 funktion käyttämistä salaukseen sen tarjoaman heikon suojan vuoksi. Suosituksesta huolimatta, Stevens, M. Bursztein, E. Karpman, P. Albertini, A. Markov, Y (2017) kertoo SHA-1 algoritmia käytettävän silti edelleen useissa eri kohteissa, kuten GIT versionhallintajärjestelmässä. Taulukko 1 esittää näiden tiivistefunktioiden perusominaisuudet.

Taulukko 1. Eri SHA algoritmien ominaisuuksia (mukaillen National Institute of Standards and Technology 2015)

Algoritmi	Viestin koko bitteinä	Lohkon koko bitteinä	Sanan koko bitteinä	Tiivistelmän koko bitteinä
SHA-1	< 2^64	512	32	160
SHA-224	< 2^64	512	32	224
SHA-256	< 2^64	512	32	256
SHA-384	< 2^128	1024	64	384
SHA-512	< 2^128	1024	64	512
SHA-512/224	< 2^128	1024	64	224
SHA-512/256	< 2^128	1024	64	256

3.1 Henkilötunnus

Henkilötunnus on 1960-luvulla Suomessa käyttöön otettu yksilöimiskeino, jonka tarkoituksena on yksilöidä Suomen kansalaiset nimeä tarkemmin. Tunnus koostuu syntymäajasta ja yksilönumerosta, jotka paljastavat henkilön iän ja sukupuolen. Henkilötunnus on henkilökohtainen eikä esimerkiksi kahdella Suomen kansalaisella voi olla samaa tunnusta. Henkilötunnusta käytetään yleisesti eri viranomaisten tai yksityisen sektorin kanssa toimiessa, jos tarvitaan varmuus tietojen rekisteröimisestä juuri oikealle henkilölle. Lukuun ottamatta tiettyjä poikkeuksia henkilötunnusta ei voi vaihtaa. Poikkeuksia voivat olla syntymäajan tai sukupuolen virheellisyys tai jos terveys tai turvallisuus ovat uhattuna. Lisäksi henkilötunnuksen toistuva väärinkäyttö voi johtaa tunnuksen vaihtoon. (Digi- ja väestötietovirasto 2023.)

Ruotsissa henkilöiden tunnistamiseen käytetään henkilötunnusta ja koordinaationumeroita. Henkilötunnus annetaan henkilölle, joka on rekisteröity Ruotsin väestörekisteriin ja koordinaationumero annetaan henkilölle, joka ei ole Ruotsin väestörekisterissä. (Skatteverket s.a.)

Henkilötunnus Ruotsissa on pysyvä tunniste, joka pysyy samana koko elämän ajan. Tunnusta käytetään tunnistautumisessa kommunikoidessa viranomaisten tai yritysten kanssa. Koordinaationumerot otettiin käyttöön Ruotsissa vuonna 2000 ja ne auttavat viranomaisia tunnistamaan henkilöitä, jotka eivät ole Ruotsin väestörekisterissä. Koordinaationumero on yksilöllinen ja pysyvä tunniste. Sitä voidaan käyttää kommunikoinnissa Ruotsin viranomaisten, terveyspalveluiden ja pankkien kanssa. Koordinaationumero voidaan merkitä passiiviseksi, jos sitä ei käytetä viiteen vuoteen. Passiivinen numero voidaan tarvittaessa aktivoida uudelleen. Koordinaationumero annetaan henkilölle Ruotsin veroviranomaisen toimesta tilanteessa, jossa henkilöllä ei ole Ruotsissa rekisteröityä

osoitetta. Näitä tapauksia ovat esimerkiksi työsuhde tai opintojen aloitus Ruotsissa. Kuitenkin, jos henkilö haluaa oleskella Ruotsissa yli vuoden, on hänen ilmoitettava asiasta Ruotsin veroviranomaiselle, jolloin henkilölle annetaan henkilötunnus. (Skatteverket s.a.)

3.2 Hajautusarvon käyttö yksilöivänä tunnisteena

Introduction to the hash function as a personal data pseudonymisation technique - raportissaan Agencia española de protección de datos (AEPD) ja European Data Protection Supervisor (EDPS) havainnoivat ongelmaa, jossa eri syötetyllä arvolla tulee sama hajautustulos. Heidän esimerkissään annetaan hajautusarvo jokaiselle julkaistulle kirjalle, joita on noin 130 miljoonaa. Tämä havainnollistava esimerkki vastaa hyvin henkilötunnuksen käyttöä yksilöivän hajautuksen luomiseen, koska kuten kirjojen nimet, myös henkilötunnukset noudattavat tiettyä kaavaa. Kirjan nimet eivät ole satunnaisia merkkijonoja, vaan noudattavat tietyn kielen kielioppia. Tässä esimerkissä ei siis tarvitse ottaa huomioon satunnaisia merkkijonoja laskettaessa mahdollisuutta, jossa kahdella annetulla arvolla tulee sama hajautustulos. Henkilötunnuksen kohdalla voidaan siis poistaa laskuista kaikki sellaiset alkuarvot, jotka eivät noudata henkilötunnuksen perusrakennetta. Henkilötunnus noudattaa tilastokeskuksen mukaan muotoa PPKKVVXNNNT, jossa ensimmäiset 6 merkkiä kuvaavat syntymäaikaa, X vuosisataa, jolloin henkilö on syntynyt, NNN juoksevaa numeroa väliltä 002-899 (naisilla parillinen ja miehillä pariton) ja T laskennallisesti määritettävää tarkistusmerkkiä. Esimerkiksi 000000Å9999 on arvo, joka ei noudata henkilötunnuksen muotoa. Tämän takia sitä ei tarvitse ottaa huomioon laskettaessa mahdollisuutta kuulua samaan alkukuvaan oikeaa henkilötunnuksen muotoa noudattavan arvon kanssa. Kuten raportissakin pääteltiin, tässäkin tapauksessa on syytä olettaa, että jokainen henkilötunnus tuottaa eri hajautusarvon.

Vaikka henkilötunnuksen muotoilu tuo hyviä puolia mukanaan, se myös helpottaa hajautusarvon palauttamista alkuperäiseen muotoonsa. Tämä riski syntyy samasta syystä, kuin hyöty, eli henkilötunnuksen muodon säännöistä. Jos hyökkääjä tietää, että tietokannassa on säilytettynä henkilötunnuksesta jalostettu hajautusarvo, hän voi helposti kohdistaa väsytyshyökkäyksen koskemaan vain mahdollisia henkilötunnuksia. Henkilötunnuksen kohdalla tietojoukko on siis erittäin järjestyksellistä. Järjestyksen tai epäjärjestyksen määrää kutsutaan entropiaksi, mitä suurempi entropia, sitä enemmän informaatiota tietojoukko sisältää. Käytettäessä hajautusta identifioivana tunnisteena on siis pidettävä mielessä, että mitä pienempi entropia, sen epätodennäköisempää on sama hajautusarvo kahdella eri syötteellä, mutta sitä helpompi on päätellä alkuperäinen arvo hajautusarvosta.

Vaikka henkilötunnuksen saisikin pääteltyä hajautusarvosta, Musti Groupille tehty taustajärjestelmä pitää sisällään lisätyn tietoturvakerroksen käyttämällä hajautusarvoa vain pseudoyksilöivänä tunnisteena. Henkilötunnuksen hajautusarvosta ja asiakkaan asiakastunnisteesta luodaan tietokantaan hajautustaulukko, jonka avulla taustajärjestelmä voi hakea hajautusarvoa vastaavan

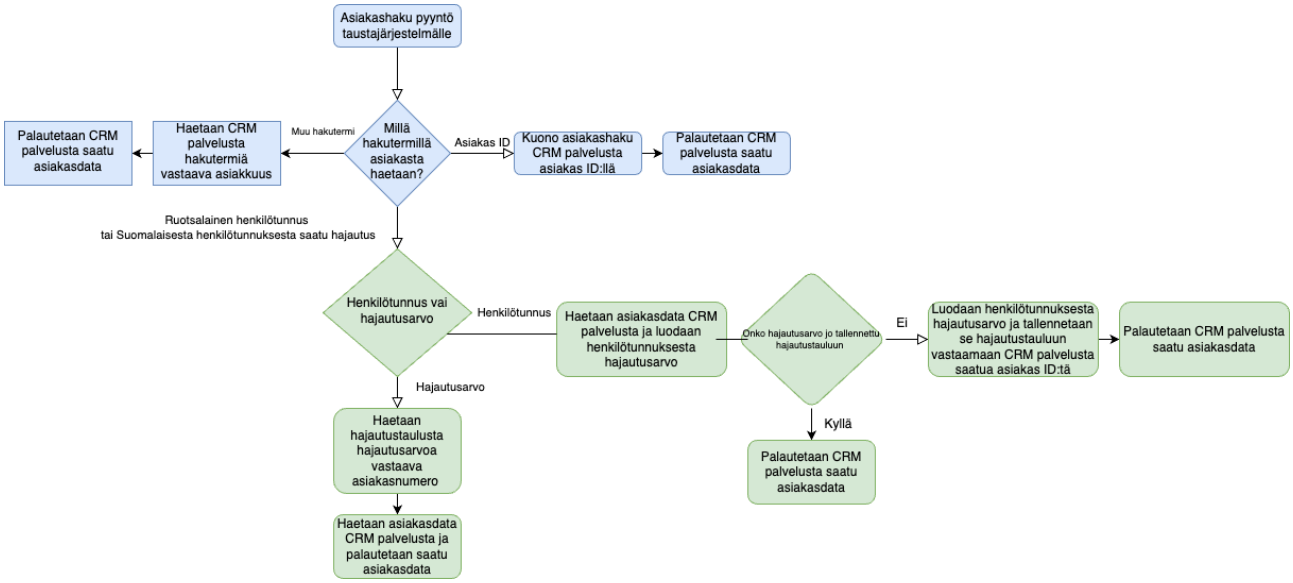
asiakastunnisteen ja vasta sillä haetaan CRM-järjestelmästä asiakkaan henkilökohtaisia tietoja (Kuva 1). Vaikka hyökkääjä saisi tietoonsa alkuperäisen arvon, millä hajautusarvo on luotu, hän saisi tietoonsa vain numeron, jota ei voi yhdistää asiakkaaseen. Näin hyökkääjän saama arvo on sama kuin vain arvaamalla saatu satunnainen henkilötunnus.

3.3 Salauksen luominen

Salauksen luominen voidaan suorittaa joko käyttöliittymässä tai taustajärjestelmässä. Musti Groupin toteutuksessa salaus päädyttiin suomalaisten kanta-asiakkaiden osalta suorittamaan jo käyttöliittymän puolella, koska verkkoliikenteeseen ei haluttu viedä suomalaisten asiakkaiden henkilötunnuksia. Kuitenkin ruotsalaiset henkilötunnukset lähetetään edelleen http pyynnön rungossa, jolloin salaus suoritetaan vasta taustajärjestelmän puolella. Tämä loi haasteen valita oikea tapa luoda Pythonilla SHA-512 salaus. Lopulta taustajärjestelmässä päädyttiin käyttämään hashlib kirjastoa, koska se on sisäänrakennettu Python ohjelmointikieleen. Hashlib kirjastoa hyväksikäyttäen Ruotsalaisesta henkilötunnuksesta luodaan hajautusarvo koodirivillä hashed_value = hashlib.sha512(value.encode('utf-8')).hexdigest(). Tämä toteutus osoittautui toimivaksi, kun testattiin, tuleeko samasta henkilötunnuksesta sama arvo käyttäen taustajärjestelmän ja käyttöliittymän toteutusta.

4 Asiakashakufunktion kehittäminen

Tässä luvussa käsitellään olemassa olevan asiakashakufunktion jatkokehitystä. Funktion tarkoituksena on löytää käyttöliittymästä haettu asiakas CRM palvelusta. Jatkokehityksessä keskitytään lisäämään ominaisuus, jolla saadaan jo olemassa olevalla Python-funktiolla haettua kanta-asiakkuuden tiedot myös hajautusarvolla. Tähän tavoitteeseen päästään hakemalla hajautusarvoa vastaava asiakastunnus DynamoDB tietokannasta ja saadulla asiakastunnuksella haetaan CRM-järjestelmästä asiakasdata. Jatkokehityksen tueksi luvussa 2.3 esiteltyä vuokaaviota on jalostettu vastaamaan järjestelmän tarpeita. Vuokaaviossa Kuva 2 on esitelty olemassa oleva toiminnallisuus sinisellä pohjavärillä ja kehitettävät toiminnallisuudet vihreällä värillä.



Kuva 2 Vuokaavio tukemaan asiakashaun jatkokehitystä (Musti Group 2023)

4.1 Teknologiat

Tässä kappaleessa esitellään käytetyt teknologiat, joiden avulla asiakashakufunktion jatkokehitys toteutettiin. Kaikki luvussa esitellyt teknologiat ovat jo käytössä Musti Groupin taustajärjestelmissä. Taustajärjestelmässä teknologiavalintojen tulee mahdollistaa kestävien ja skaalautuvien ohjelmistojen kehittämisen. Teknologiavalinnoissa tulee ottaa myös huomioon kustannukset ja yhteensopivuus muiden teknologioiden kanssa. Pilvipalvelutarjoajana suositaan Amazon Web Servicesiä laskutuksen helpottamiseksi (Musti Group 2023 a).

4.1.1 AWS lambdat

AWS Lambda on Amazon Web Servicesin tarjoama palvelu, joka mahdollistaa koodin suorittamisen ilman palvelimien hallintaa tai allokointia. Lambdat ovat tapahtumavetoisia pilvifunktioita, joka tarkoittaa sitä, että Lambda palvelu suorittaa tarjotun funktion vain sitä tarvittaessa (Amazon Web Services. s.a a.). Musti Group käyttää kaikissa taustajärjestelmissään Serverless framework ohjelmistokehystä ja AWS palveluntarjoajaa. AWS palveluntarjoajaa käytettäessä kaikki tapahtumat Serverless Frameworkissa ovat Lambda funktioita.

Lambdoja käytettäessä funktiolle annetaan parametriksi ”context” ja ”event”. Event on Python sanakirja (dictionary), joka sisältää AWS Lambda funktiolle lähetetyt syötteet AWS API Gatewayn kautta. Tämä objekti voi sisältää erilaisia tietoja, kuten http-pyynnön parametreja. Context on AWS Lambdan objekti, joka tarjoaa tietoja ja metodeja, jotka liittyvät suoritettavaan funktioon. Tämä objekti sisältää muun muassa suoritukseen kuluneen ajan ja AWS:n suoritus id:n. (Serverless. s.a.)

4.1.2 Serverless framework

Serverless framework on avoimen lähdekoodin työkalu serverless ohjelmistojen kehittämiseen, paketointiin ja julkaisuun. Serverless framework hyödyntää suunnittelumallia, jossa ohjelmistoon tarvittavaa infrastruktuuria hallinnoi ulkopuolinen taho tarpeen mukaan. Tässä mallissa laskentatehoa ja muita vaadittavia resursseja käytetään vain tarpeesta, joka säästää kustannuksissa. Serverless saa nimensä konseptista, jossa ohjelmistolla ei ole sille omistettua palvelinta. (Medium 2021)

4.1.3 Serverless Frameworkin konfiguraatio

Serverless framework käyttää konfiguraatioiden asettamiseen serverless.yml tiedostoa, joka on kirjoitettu YAML kielellä. YAMLia käytetään yleisesti konfiguraatio tiedostojen luomiseen sen joustavuuden ja saavutettavuuden vuoksi (Red Hat. s.a.). Serverless.yml tiedosto kuvaa serverless sovelluksen funktiot, resurssit, lisäosat, ja muut tarvittavat konfiguraatiotiedot (Trendmicro. s.a.).

4.1.4 NoSQL tietokannat ja DynamoDB

NoSQL eli ei-relaatiotietokanta on tietokantasuunnittelun lähestymistapa. Tämä tapa mahdollistaa datan tallentamisen ja kyselyiden tekemisen perinteisten relaatiotietokantojen taulukkorakenteen sijaan yhteen tietorakenteeseen, kuten JSON-dokumenttiin. Koska tällainen ei-suhteellinen tietokannan suunnittelu ei vaadi skeemaa, se mahdollistaa nopean skaalautuvuuden suurten ja yleensä rakenteettomien tietojoukkojen hallintaan. (IBM. s.a.)

Amazon DynamoDB on täysin hallittu NoSQL-tietokantapalvelu, joka tarjoaa erittäin hyvän skaalautuvuuden. Täysin hallittu tietokantapalvelu tarkoittaa DynamoDB:n tapauksessa

tietokantapalvelua, joka hoitaa kaikki tietokannan hallintaan liittyvät tehtävät. Nämä tehtävät sisältävät esimerkiksi varmuuskopioinnin, laitteiston varaukset, asennukset, konfiguroinnit, ohjelmistopäivitykset ja skaalaukset. DynamoDB:llä voi luoda tietokantatauluja, jotka voivat tallentaa ja noutaa minkä tahansa määrän dataa ja palvella minkä tahansa tason pyyntöliikennettä. (Amazon Web Services. s.a b.)

4.2 Hajautustaulu

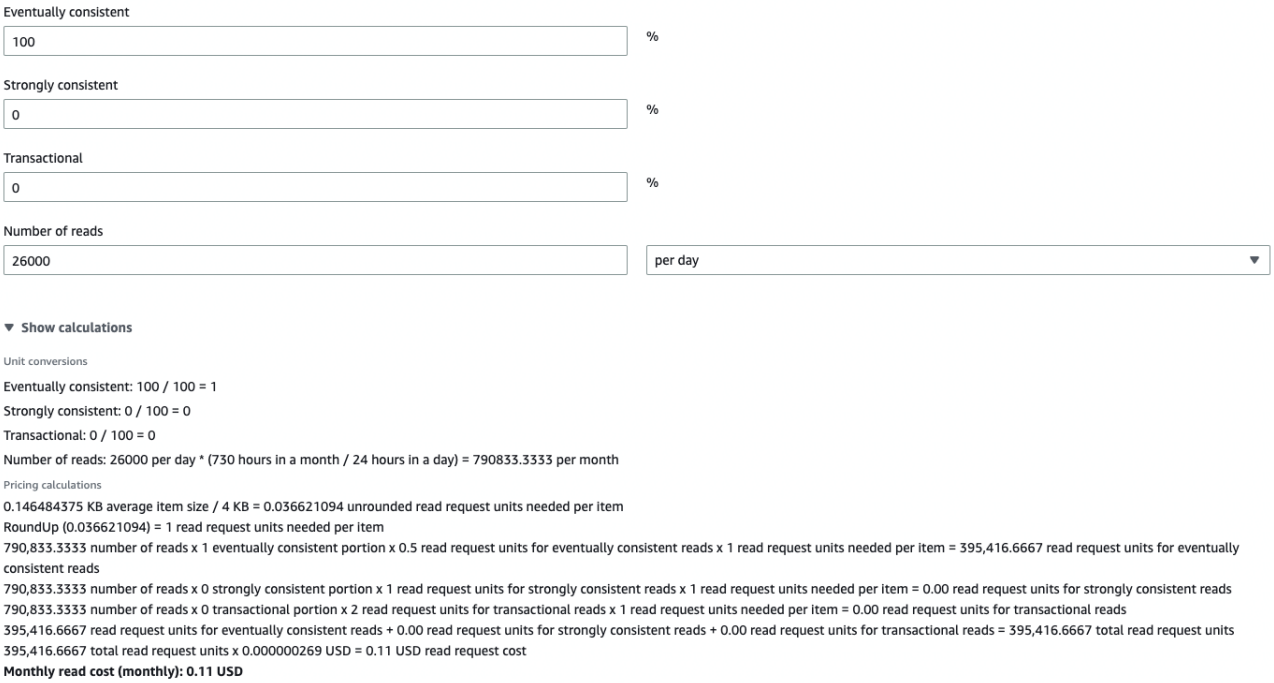
DynamoDB on valittu hajautusarvojen ja niitä vastaavan asiakasnumeron tallentamiseen, koska Musti Group käyttää Amazon Web Servicesin palveluita jo muissa toteutuksissa taustajärjestelmän puolella. Tässä toteutuksessa DynamoDB:tä tarvitaan tallentamaan asiakkaan henkilötunnuksesta luotu hajautus ja sitä vastaava asiakasnumero, jotta voidaan luoda yhteys hajautusarvon ja asiakkuuden välillä. DynamoDB:seen luodaan siis avain-arvo-pari, jonka avaimena toimii asiakkaan henkilötunnuksesta luotu hajautusarvo ja arvona toimii asiakkaan asiakastunnus.

DynamoDB sisältää ominaisuuden automaattiselle skaalaukselle. 2017 julkaistu ominaisuus varmistaa tietokantataulun olevan aina saatavilla. Automaattinen skaalaus monitoroi tietokantataulun käyttöä ja provisioi tarvittavan määrän resursseja tietokantataulun toimintaan (Amazon Web Services. 2017). Tämä ominaisuus varmistaa hajautustaulun saatavuuden myös silloin, kun asiakasmäärä on suhteellisesti suurempi, kuten Black Friday alennusten aikana Musti Groupilla.

Musti Groupin asiakasmikropalvelua kutsutaan kerran n. 0.7 sekunnin välein myymälöiden aukioloaikoina (Musti Group 2022). Tämän takia opinnäytetyössäni tutkin myös tietokannan kustannuksia yritykselle. Amazon Web Services tarjoaa kustannuslaskurin omilla sivuillansa, jota voidaan hyödyntää kustannuksien arviontiin. DynamoDB palvelua voidaan käyttää joko ”käytön mukaan” periaatteella tai provisioidulla kapasiteetilla. Koska Musti Groupin liikenne tietokanta tauluun ei ole myymälöiden aukiolojen vuoksi tasaista, parempi vaihtoehto on valita käytön mukaan skaalautuva palvelu. Datan säilyvyys on myös erittäin tärkeää Musti Groupille tässä tietokantataulussa, joten kustannusarvioon otetaan mukaan myös AWS:n tarjoama ”Backup and restore” ominaisuus, joka auttaa tietokantataulun varmuuskopioiden luomisessa ja palauttamisessa.

Keskimääräisen tietokanta rivin koko on laskennallisesti 150 tavua. Koska Musti Group on ilmoittanut tilinpäätöksessään vuonna 2022, että yhtiöllä oli tilikauden 2022 lopussa 1,5 miljoonaa kanta-asiakasta, voimme tehdä laskun suurimman rasituksen mukaan, eli jokainen kanta-asiakas käyttäisi uutta kanta-asiakkuuden tunnistustapaa. Kun kerrotaan 150 tavua 1,5 miljoonalla asiakkaalla, saadaan tulokseksi 225 000 000 tavua, joka vastaa 0,20955 Gigatavua. Musti Groupin antamalla arviolla asiakashauista sekunnissa, voimme arvioida asiakashakuoperaatioiden määrät päivässä muuttamalla sekunnit tunneiksi ja kertomalla se luku myymälöiden aukiolotunneilla.

Kun syötämme arvioidut luvut AWS:n kustannuslaskuriin, saamme arvion tietokannan kustannuksista. Kustannuslaskurin tulokset löytyvät kuvasta 3. Laskurin antama arvio olisi noin 0.11 USD kuukaudessa.



Kuva 3 Amazon Web Services kustannuslaskurin tulokset arvioiduille luvuille

4.2.1 Hajautustaulun luominen

Hajautustaulun luominen AWS konsolista ei ole riittävä, jos halutaan saada kaikki hyöty irti Serverless Frameworkin IAC eli Infrastructure As Code- periaatteesta. IAC varmistaa ympäristöjen helpon uudelleen pystyttämisen esimerkiksi vaihdettaessa datakeskusta, pitämällä kaiken tiedon infrastruktuurista projektin koodissa. Tämän takia serverless.yml tiedostoa, joka sisältää tiedon projektin infrasta, muokattiin luomaan uusi taulu DynamoDB palveluun. Kuva 4 antaa esimerkin siitä, miten ExampleTable niminen taulu luodaan YAML kielellä Serverless Frameworkissa. Kuvassa iamRoleStatements antaa oikeudet taulun käyttöön palvelulle, ja Resources alla olevat rivit luovat itse taulun.

```yaml
service: Example # Use the repository name here
frameworkVersion: '3'

provider:
  name: aws
  runtime: python3.9
  region: us-east-1

  iamRoleStatements:
    - Effect: Allow
      Action:
        - "dynamodb:GetItem"
        - "dynamodb:PutItem"
        - "dynamodb:Scan"
        - "dynamodb:DeleteItem"
      Resource:
        - ExampleTable.Arn
resources:
  Resources:
    ExampleTable:
      Type: AWS::DynamoDB::Table
      Properties:
        TableName: ExampleTable
        AttributeDefinitions:
          - AttributeName: ID
            AttributeType: S
        KeySchema:
          - AttributeName: ID
            KeyType: RANGE
        BillingMode: PAY_PER_REQUEST
        PointInTimeRecoverySpecification:
          PointInTimeRecoveryEnabled: true
```

Kuva 4 Serverless.yml DynamoDB taulun luominen koodilla

4.3 Uuden menetelmän integrointi asiakashakufunktioon

Jotta uusi ominaisuus saatiin lisättyä jo olemassa olevaan asiakashakuun, Musti Groupin taustajärjestelmässä jo olemassa olevaa Python-funktiota tuli muokata vastaamaan uuden vaihtoehtoisen kanta-asiakkuuden tunnistusmenetelmän vaatimuksia. Funktio suorittaa haun CRM palveluun hakuparametreina annetuiden tietojen avulla. Hakuparametreja voivat olla asiakastunnus, sähköposti, matkapuhelinnumero, tai Ruotsin myymälöille henkilötunnus. Musti Groupin vanhassa toteutuksessa Ruotsin myymälät pystyivät jo käyttämään henkilötunnusta asiakashaun parametrinä.

Jotta ominaisuus myös hajautusarvolla hakemiselle oli mahdollista kehittää, tuli ohjelmistologiikkaa muokata. Logiikka aikaisemmassa toteutuksessa on erottanut asiakastunnuksen ja ruotsalaisen henkilötunnuksen hakemisen muista hakutermeistä. Asiakastunnus ja henkilötunnus haku käyttävät eri päätepistettä CRM järjestelmän rajapinnoilta, jonka takia tämä tarkistus on jo ollut olemassa. Asiakastunnuksella haku haluttiin edelleen pitää erillään muista hauista, koska se on tehokkain tapa hakea asiakastietoja.

Kehitetyssä ohjelmistologiikassa tarkistetaan, onko hakuparametri hajautusarvo tai Ruotsalainen henkilötunnus. Jos hakuparametri on toinen näistä, kutsutaan tätä varten kehitettyä Python-funktiota. Kehitetty Python-funktio käsittelee datan kuvassa 2 esitetyn vuokaavion mukaisesti.

4.3.1 Hajautusarvon ja henkilötunnuksen käsittely

Kehitetty Python-funktio hakee annetulla asiakastiedolla asiakasdatan CRM järjestelmästä, hyväksi käyttäen DynamoDB-taulun avain-arvo-pareja. Tämän Python-funktion tulee ottaa vastaan, joko Ruotsalainen henkilötunnus tai Suomalaisesta henkilötunnuksesta luotu hajautusarvo. Jos kyseessä on ruotsalainen henkilötunnus, funktio luo siitä hajautusarvon käyttäen SHA-512 algoritmia. Tämän ansiosta kumpaakin arvoa voidaan käsitellä samalla tavalla funktion myöhemmissä vaiheissa. Tämä mahdollistaa ohjelmistokehityksen DRY -eli Do not Repeat Yourself -periaatteen noudattamisen. Kun funktiolla on käsiteltävänä hajautusarvo, halutaan tarkistaa, löytyykö hajautusarvoa vastaavaa asiakastunnusta DynamoDB:hen luodusta taulusta. Jos tämä funktio palauttaa customer_id muuttujan, hajautusarvoa vastaava asiakastunnus on löytynyt. Löytyneellä asiakastunnuksella voidaan hakea asiakasdata CRM palvelusta ja palauttaa se takaisin kutsuvaan käyttöliittymään. Jos hajautusarvoa vastaavaa asiakastunnusta ei kuitenkaan löydy tietokannasta,

tarkistetaan, onko company muuttujan arvo ruotsalainen yritys. Koska CRM palvelusta löytyy jo ruotsin asiakkaiden henkilötunnukset voimme tehdä CRM palveluun pyynnön, jolla saamme asiakasdatan. Asiakasdatan asiakastunnus ja henkilötunnuksesta luotu hajautus tallennetaan DynamoDB tietokantatauluun. Näin varmistetaan, että myös Ruotsin myymälöissä Musti Group voi siirtyä myöhemmin hajautusarvolla tehtävään hakuun, ilman suuria toimenpiteitä tai migraatio-ongelmia.

5 Datan käsittely

CRUD, eli Create, Read, Update ja Delete on akronyymi, joka edustaa neljää perustoimintoa, joita suoritetaan tietojen käsittelyssä tietokannoissa (Mozilla Foundation. s.a.). Musti Groupin uuden kanta-asiakastunnistautumiseen käytettävään DynamoDB tauluun tarvitaan CRUD metodeista vain Create eli luonti, Read eli luku ja Delete eli poisto. Update jätetään kokonaan pois, koska tietokannassa olevaa dataa ei haluta päästä muokkaamaan. Näin datan käsittelyssä vältetään inhimilliset virheet datan käsittelyssä. Kuten mahdollisuus näppäillä väärin asiakasnumero, jolloin asiakkaan henkilötunnuksesta tehty hajautusarvo liitetään väärälle asiakkaalle. (Musti Group. 2023a)

Kaikki data mitä tietokantaan tallennetaan, voidaan pitää muuttumattomana datana. Asiakasnumero on aina sama yhdellä asiakkaalla Musti Groupin CRM järjestelmässä ja henkilötunnuksen muuttuminen on todella harvinaista. Näissä harvinaisissa tapauksissa, jossa asiakastiedot kuitenkin muuttuvat on suositeltavaa poistaa asiakkaaseen liittyvä data tietokannasta, jonka jälkeen lisätä kokonaan uusi tietokantamerkintä. Create ominaisuutta käytetään luomaan uusi avain-arvo-pari liitettäessä hajautusarvoa asiakastunnukseen ja Delete ominaisuutta käytetään tämän avain-arvo-parin poistamiseen.

5.1 Uuden datan lisääminen tietokantaan

Jotta uusi tieto voidaan lisätä tietokantaan, tuli sille luoda uusi Python-funktio Musti Groupin taustajärjestelmän asiakkaat mikropalveluun. Tätä metodia kutsutaan suoraan päätepisteestä, ja se kutsuu AWS lambda funktiota, joten sille annetaan parametreiksi event ja context. Funktio tarvitsee tehtäväänsä tietoja kuten liitettävän asiakkaan asiakasnumeron ja asiakkaan henkilötunnuksesta luodun hajautusarvon. Nämä arvot saadaan annettua funktiolle http-kutsun rungossa.

Kehitetyssä Python-funktiossa tarvittavien parametrien saamisen jälkeen, luodaan koodissa boto3:lla resurssi DynamoDB:stä ja annetaan resurssille tieto mitä taulua tarkistellaan. Resurssilla voidaan tarkastaa, onko annetulla hajautusarvolla jo olemassa olevaa dataa tietokannassa. Jos data on olemassa, palautetaan http vastaus, jossa kerrotaan datan olevan jo olemassa. Jos dataa ei löydy tietokannasta, voidaan sinne lisätä avain-arvo-pari. Kehitettyyn Python-funktioon lisättiin myös virheiden kirjaaminen AWS:n CloudWatch palveluun.

5.2 Datan poistaminen tietokannasta

Delete Python-funktion kehittäminen on erittäin samankaltaista, kuin Create Python-funktion. Suurin ero näillä funktiolla on DynamoDB tauluun tehtävä operaatio. Nimen mukaisesti Delete poistaa dataa tietokannasta. Kehitettävään funktioon ei kuitenkaan tarvitse Createn tavoin lisätä asiakasnumeroa parametreihin, koska asiakkaan henkilötunnuksesta luotu hajautusarvo toimii

osioavaimena. Osioavainta käyttäessä käytetään tauluresurssin delete_item funktiota koskemaan vain tiettyä dataa tietokannassa, antamalla Key arvoksi asiakkaan henkilötunnuksesta tehty hajautusarvo. Tämä poistaa tarpeen tehdä skannaus operaatiota koko taululle etsittäessä poistettavaa arvoa, joka säästää kustannuksissa käytettäessä DynamoDB tietokantaa.

5.3 Uusien päätepisteiden luominen

Asiakkaiden tietojen lukemisessa eli asiakashaussa käytetään olemassa olevaa rajapinnan päätepistettä, mutta uusien luonti- ja poistofunktioiden vaatimaa päätepistettä ei ole vielä olemassa. Koska Musti Group käyttää taustajärjestelmässään omaa ohjelmistokirjastoa päätepisteiden rakentamiseen, tuli varsinaisen ohjelmointilogiikan lisäksi muokata myös reitittimen ohjelmakoodia. Reitin on osa ohjelmakoodia, joka vastaa päätepisteiden paljastamisesta. Reitittimen tiedostoon on asetettu reitittimen konfiguraatio JSON-muodossa ja se sisältää kaikki asiakasmikropalvelun päätepisteet, sekä yleisiä asetuksia, kuten mikropalvelun nimi. Jotta uusi päätepiste saatiin asetettua reitittimeen, lisättiin JSON olioon polut kehitetyille luonti- ja poistofunktioille kuvan 5 mukaisesti.

```
"connect_id_to_hash" : {
    "path": "/customers/connect_id_to_hash",
    "methods": {"POST": connect_id_to_hash},
},
"delete_id_with_hash": {
    "path": "/customers/delete_id_with_hash",
    "methods": {"DELETE": delete_id_with_hash},
}
```

Kuva 5 Lisätyt rivit reititin tiedostoon

Uusien päätepisteiden toimivuus testattiin ensin Postman- työkalulla, jonka jälkeen käyttöliittymällä. Päätepisteet todettiin toimivaksi luomalla uusi avain-arvo-pari käyttäen connect_id_to_hash päätepistettä ja poistettiin sama avain-arvo-pari käyttäen delete_id_with_hash päätepistettä. Samalla testillä saatiin myös testattua asiakashakufunktion toimiminen hakemalla hajautusarvolla asiakasta, mikä onnistui.

6 Pohdinta

Opinnäyteyöni empiirinen osuus valmistui odotettua nopeammin. Valmistumistoive toimeksiantajan puolelta oli toukokuuhun mennessä, mutta empiirisen osuuden kehitys oli valmis jo huhtikuun alussa. Toimeksiantaja antoi tavoitteeksi kehittää kanta-asiakkuuden tunnistautumistapa, joka olisi nopeampi, turvallisempi ja käyttäjäystävällisempi. Nämä tavoitteet opinnäytetyössä täytettiin. Kehitetty menetelmä on nopeampi, koska asiakkaan kanta-asiakkuuden tunnistamiseen ei tarvita kuin yksi skannaus henkilö- tai ajokortista. Menetelmä on turvallisempi, koska asiakkaiden ei tarvitse enää kertoa omia henkilötietojaan ääneen. Menetelmä on myös käyttäjäystävällisempi, koska myymälähenkilökunnan ei tarvitse näppäillä pitkiä puhelinnumeroita tai asiakasnumeroita järjestelmään. Mielestäni opinnäytetyö on onnistunut, koska saavutin tavoitteet aikataulussa tai nopeammin.

Suurimmat haasteet opinnäytetyössäni oli tietosuojaan ja tietoturvaan liittyvät kysymykset, sekä pitkään kehitetyn ohjelman jatkokehitys. Tietosuoja ja tietoturva olivat haasteellisia, koska kyseessä on henkilön henkilötunnus. Henkilötunnuksesta luodun hajautusarvon käyttö yksilöivänä tunnisteena ei ole laajasti dokumentoitu aihe. Pääsin opinnäytetyötäni varten haastattelemaan kiinnostavia henkilöitä, kuten Broman Groupin Sami Tuupaista (CIO), joka antoi minulle laajempaa kuvaa siitä mitä toteutuksessa tulee ottaa huomioon. Jo pitkään kehitetyn mikropalvelun jatkokehitys oli haastavaa, koska parhaat toimintatavat Serverless Frameworkin kehityksessä ovat muuttuneet suuresti muutamassa vuodessa. Kun kehitettävän mikropalvelun kehittäminen on alun perin aloitettu tämä ohjelmistokehityskehys on ollut erittäin nuori ja hakenut vielä identiteettiä. Mikropalvelua on kehittänyt useita eri henkilöitä, joista jokainen on jättänyt oman kädenjälkensä koodiin. Isoimpana erona on esimerkiksi Musti Groupin oma ohjelmakirjasto, joka on luotu vasta paljon kehityksen aloittamisen jälkeen, joten ohjelmakoodissa on useita eri ratkaisuja samaan ongelmaan.

Tuloksena opinnäyteyöstä saatiin toimiva vaihtoehtoinen kanta-asiakkuuden tunnistautumismenetelmä. Tämä menetelmä on opinnäytetyötäni viimeistellessä käytössä jo Ruotsin myymälöissä, joista tullut palaute on erittäin positiivista. Myymälähenkilökunta kertoo uuden menetelmän olevan nopeampi ja turvallisempi kuin edellinen. Tämän palautteen perusteella voin todeta opinnäytetyön olleen onnistunut suhteessa sille asetettuihin tavoitteisiin.

Projektia jatketaan opinnäytetyön rajausten ulkopuolella viemällä menetelmä käyttöön myös Musti Groupin Suomen myymälöihin. Pilotointi on jo aloitettu Peten koiratarvikkeen myymälöistä, jonka onnistumisen jälkeen jatketaan myös Musti ja Mirri myymälöihin. Tämä tehdään ensin Peten koiratarvikkeen myymälöihin, koska niitä on selkeästi vähemmän joten asiakkaat eivät käy niin useasti monessa eri Peten koiratarvikkeen myymälässä. Tämä estää hämmennystä asiakkaissa, jotka ovat pystyneet yhdessä myymälässä tunnistautumaan kortilla, mutta toisessa eivät. Projektin

pilotointia varten loin infografiikan myymälähenkilökuntaa varten, joka on opinnäytetyön liitteenä (Liite 1).

Opinnäytetyö prosessin aikana syntyi myös jatkokehitysideoita. Koska taustajärjestelmä ei välitä minkälaisesta datasta hajautusarvo on luotu, on mahdollista jatkossa käyttää kehitettyä ominaisuutta myös muita kuin henkilötunnusten hajautusarvoja asiakasdatan hakemiseen. Tämä data, josta hajautusarvo on luotu, voisi olla esimerkiksi asiakkaalle annettu QR-koodi. Toinen mahdollinen jatkokehitysidea olisi antaa mahdollisuus asiakkaalle liittää henkilökortti itsenäisesti verkkokauppaan kirjautumalla.

Olen itse oppinut opinnäytetyöni aikana itsenäistä projektinhallintaa, enemmän SHA-algoritmeista, henkilötunnuksesta ja henkilökohtaisten tietojen salaamisesta. Varsinkin itsenäinen projektinhallinta on auttanut minua jo nyt työelämässä, antamalla itsevarmuutta aikataulujen asettamiseen ja projektien suunnitteluun.

Lähteet

Agencia española de protección de datos (AEPD), European Data Protection Supervisor (EDPS). 2019. Introduction to the hash function as a personal data pseudonymisation technique. European data protection supervisor. Luettavissa: https://edps.europa.eu/sites/default/files/publication/19-10-30_aepd-edps_paper_hash_final_en.pdf Luettu: 7.3.2023

Amazon Web Services. 2017. New – Auto Scaling for Amazon DynamoDB. Luettavissa: https://aws.amazon.com/blogs/aws/new-auto-scaling-for-amazon-dynamodb/. Luettu: 13.4.2023.

Amazon Web Services. s.a a. What is AWS Lambda. Luettavissa: What is AWS Lambda? - AWS Lambda (amazon.com). Luettu: 22.3.2023

Amazon Web Services. s.a b. What is Amazon DynamoDB. Luettavissa: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html. Luettu: 12.3.2023

Digi- ja väestötietovirasto. Henkilötunnus. Luettu 20.2.2023. Luettavissa: https://dvv.fi/henkilotunnus.

IBM. s.a. What is a NoSQL database. Luettavissa: https://www.ibm.com/topics/nosql-databases. Luettu: 12.3.2023

Medium. 2021. Serverless Framework. Luettavissa: https://medium.com/kodeyoga/serverless-framework-a73f63ab603b. Luettu: 15.4.2023

Motonet. Tietosuojaseloste. Luettu 14.2.2023. Luettavissa: https://www.motonet.fi/fi/sivut/tietosuojaseloste

Mozilla Foundation. s.a. CRUD. Luettavissa: https://developer.mozilla.org/en-US/docs/Glossary/CRUD Luettu: 20.3.2023

Musti Group 2022. Musti Group Confluence Wiki. Musti MicroServices Cloud architecture. Luettu 21.3.2023.

Musti Group 2023 a. Musti Group Confluence Wiki. Customer identification methods in store. Luettu 28.2.2023.

Musti Group 2023 b. Musti Group Confluence Wiki. Pre-study for customer identification. Luettu 28.2.2023.

National Institute of Standards and Technology. 2015. FIPS PUB 180-4 Secure Hash. FIBS 180-4 Publication. Luettavissa: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf Luettu: 2.2.2023.

Poliisi. 2023. Passit ja Henkilökortit uudistuvat. Luettu 1.3.2023. Luettavissa: https://poliisi.fi/-/passit-ja-henkilokortit-uudistuvat

Red Hat. s.a. What is YAML. Luettavissa: https://www.redhat.com/en/topics/automation/what-is-yaml. Luettu: 21.3.2023

Serverless. s.a. AWS Lambda events. Luettavissa: https://www.serverless.com/framework/docs/providers/aws/guide/events. Luettu: 22.3.2023

Skatteverket. s.a. Personal identity numbers and coordination numbers. Luettavissa: https://www.skatteverket.se/servicelankar/otherlanguages/suomeksifinska/toimintamme.4.7856a2b411550b99fb7800086458.html. Luettu 4.4.2023.

Stadium. s.a. Kysymyksiä ja vastauksia. Luettu 14.2.2023 Luettavissa: https://www.stadium.fi/INTERSHOP/web/WFS/Stadium-FinlandB2C-Site/fi_FI/-/EUR/CC_ViewFAQ-Start

Stevens, M. Bursztein, E. Karpman, P. Albertini, A. Markov, Y. 2017. The first collision for full SHA-1. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Luettavissa: https://shattered.io/static/shattered.pdf Luettu: 2.2.2023

Tilastokeskus. s.a. Henkilötunnus. Luettavissa: https://www.stat.fi/meta/kas/hetu.html Luettu: 5.3.2023

Trendmicro. s.a. How to build a Serverless API with Lambda and Node.js. Luettavissa: https://www.trendmicro.com/es_mx/devops/22/c/how-to-build-a-serverless-api-with-lambda-and-node-js.html#:~:text=js%20files.-,The%20serverless.,hello%20world%E2%80%9D%20REST%20API%20function. Luettu: 22.3.2023

Liitteet

Liite 1. Myymälähenkilökunnan infografiikka