



Verkkomainonnan tiedonkeruu- järjestelmän suunnittelu ja to- teutus

Emil Kantaneva

OPINNÄYTETYÖ
Huhtikuu 2024

Tietotekniikka
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikka
Ohjelmistotekniikka

KANTANEVA, EMIL:

Verkkomainonnan tiedonkeruujärjestelmän suunnittelu ja toteutus

Opinnäytetyö 45 sivua, joista liitteitä 1 sivu
Huhtikuu 2024

Opinnäytetyössä toteutettiin järjestelmä datan keräämiseksi ja analysoimiseksi mainonnanjakelukoodin suorituskyvystä Alma Median verkkopalveluissa erilaisilla käyttöjärjestelmä- ja selainyhdistelmillä. Tarkoituksena oli puuttua analytiikan aukkoon, joka vaikuttaa käyttäjäkokemukseen, mainonnan tehokkuuteen sekä mainostuloihin.

Projekti keskittyy datan keräämiseen mainonnanjakelukoodista, tavoitteenaan kerätä dataa, joka auttaa optimoimaan mainoskoodin suorituskykyä. Keskeisiä alueita ovat mainonnan jakelun parantaminen ja laajan analytiikan tarjoaminen, jotka ovat keskeisiä käyttäjäsitoutumisen ja mainosten tehokkuuden lisäämisessä. Työn tavoitteena oli syventää ymmärrystä mainonnanjakelukoodin käyttäytymisestä, tarjoten pohjan dataan perustuville päätöksille digitaalisen mainonnan tulosten parantamiseksi ja tulon maksimoimiseksi paremman mainonnan suorituskyvyn ja käyttäjäsitoutumisen kautta.

Lopputuloksena oli järjestelmä, joka pystyy keräämään tehokkaasti dataa erilaisista mainonnan suoritukseen liittyvistä parametreista ja esittämään tämän tiedon selkeästi ja ymmärrettävästi. Järjestelmä tarjoaa monipuolisia analyysejä ja raportteja mainoskoodin suorituskyvystä eri käyttöjärjestelmillä ja selaimilla, mikä mahdollistaa nopean puuttumisen virhetilanteisiin ja parannukset mainonnan jakeluun.

Avainsanat: verkkoanalytiikka, digitaalinen mainonta, tiedonkeruu, suorituskykymittarit

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in ICT Engineering
Software Engineering

KANTANEVA, EMIL:

The Design and Implementation of Online Ad Data Collection System

Bachelor's thesis 45 pages, appendices 1 page
April 2024

This thesis implemented and explored the process of creating a system to collect and analyze data on advertising distribution code across Alma Media's online services on various operating system and browser combinations addressing the analytics gap that affects user experience, ad effectiveness and ad revenue.

The project focused on collecting data from ad code with the aim of optimizing ad code performance. The key areas of focus included improving load times and ad presentation which are essential to improving user engagement and ad effectiveness. The aim was to deepen understanding of the behavior of the ad delivery code providing a basis for data-driven decisions to improve digital advertising performance and maximize revenue through improved ad performance and user engagement.

The thesis resulted in a system designed to effectively collect and clearly present various advertising performance metrics. The system provides rich analytics and reports on the performance of the advertising code across different operating systems and browsers allowing for rapid issue intervention and improvements to ad delivery.

Key words: web analytics, digital advertising, data collection, performance metrics

SISÄLLYS

1. JOHDANTO	6
2. DATAN HYÖDYNTÄMINEN LIIKETOIMINNASSA	7
2.1 Datan rooli CRISP-DM mallissa	7
2.2 Liiketoiminnan ymmärtäminen	9
2.3 Dataymmärtäminen.....	9
2.4 Datavalmistelu	10
2.5 Datan visualisointi	11
2.6 Yksityisyys, tietosuoja ja datan etiikka	12
3. VERKKOMAINONTA.....	14
3.1 Ohjelmallinen mainonta	15
3.2 Mittarit ja seuranta	15
4. VERKKOANALYTIikka.....	17
5. VERKKOMAINONNAN TIEDONKERUUJÄRJESTELMÄ	20
5.1 Tekninen tausta	21
5.2 Analytiikan tarpeet	23
5.3 Mainoskutsujen datan kerääminen	24
5.3.0 Kerättävä data	24
5.3.1 Kerääminen erilaisista arkkitehtuureista.....	27
5.4 Virheiden kerääminen	32
5.4.0 Kerättävä data	32
5.4.1 Keräysprosessi.....	33
5.5 Tapahtumien lähettäminen	36
5.6 Käyttäjän yksityisyys ja suostumus	37
6. Kerätyn datan tulokset	38
7. POHDINTA	42
LÄHTEET	43
LIITTEET.....	45

LYHENTEET JA TERMIT

CRISP-DM	Cross-Industry Standard Process for Data Mining. prosessimalli, jota käytetään data-analyysin ja tiedonlouhinnan projekteissa.
HTML	HyperText Markup Language. Merkintäkieli, joka on suunniteltu verkkosivujen luomiseen.
API	Application Programming Interface. Ohjelmointirajapinta, joka mahdollistaa eri ohjelmistojen välillä tapahtuvan tiedonsiirron.
JSON	JavaScript Object Notation. Kevyt datanvaihtoformaatti, joka on helppolukuinen sekä ihmisille että koneille
HTTP	Hypertext Transfer Protocol. protokolla, joka mahdollistaa tiedonsiirron web-selainten ja palvelimien välillä.
HTTPS	Hypertext Transfer Protocol Secure. Salattu versio HTTP:stä, joka on turvallisempi luottamuksellisten tietojen siirtämiseen.

1. JOHDANTO

Alma Median verkkomainonnan jakelussa havaittiin tarpeelliseksi syventää olemassa olevan mainoskoodin suorituskyvyn seuranta eri alustoilla ja selaimilla. Aiemmin toteutetut analytiikkamenetelmät eivät tarjonneet riittävän syvällistä tietoa, minkä vuoksi kehittämällä uusia tiedonkeruumenetelmiä ja laajentamalla datan keräystä oli mahdollista parantaa toiminnan tehokkuutta. Tämän myötä nousi esille kysymys siitä, kuinka mainoskoodin seuranta voidaan vahvistaa ja miten kerättyä tietoa voidaan hyödyntää parhaiten sen optimoinnissa. Opinnäytetyössä asetetaan tavoitteeksi kehittää järjestelmä, joka mahdollistaa aikaisempaa perusteellisempaa datankeruuta mainoskoodista ja tarjoaa alustan sen suorituskyvyn analysointiin ja sitä kautta tehostamiseen.

Mainoskoodin toiminta vaikuttaa olennaisesti palveluiden latausaikoihin ja mainosten näyttämiseen, mikä on suoraan yhteydessä käyttäjäkokemukseen ja mainostuloihin. Opinnäytetyö keskittyy luomaan perustan datan keruulle ja välitykselle suoraan mainoskoodista, minkä pohjalta on mahdollista tehdä jatkotutkimuksia ja kehittää mainonnan teknistä toteutusta. Projektin ulkopuolelle jää datan syvällisempi käsittely ja analysointi, mutta työssä käsitellään kootun datan ensihavaintoja ja tehdään alustavia johtopäätöksiä, jotka ohjaavat tulevia toimenpiteitä ja kehityssuunnitelmia. Työssä kuitenkin esitellään lopputuloksia esimerkkien kautta, miten kehitetty järjestelmä pystyy tuottamaan merkityksellistä tietoa päätöksentekoprosessien tueksi, ohjaamaan kehitystoimia ja edistämään ylläpitollisia toimia.

2. DATAN HYÖDYNTÄMINEN LIKETOIMINNASSA

Jokaista datahanketta tulisi lähestyä liiketoiminnan ongelman kautta, ymmärtäen että datan analysoinnin tarkoituksena on konkreettisten liiketoiminnallisten haasteiden tukeminen ja ratkaiseminen. (Holz N, 2023).

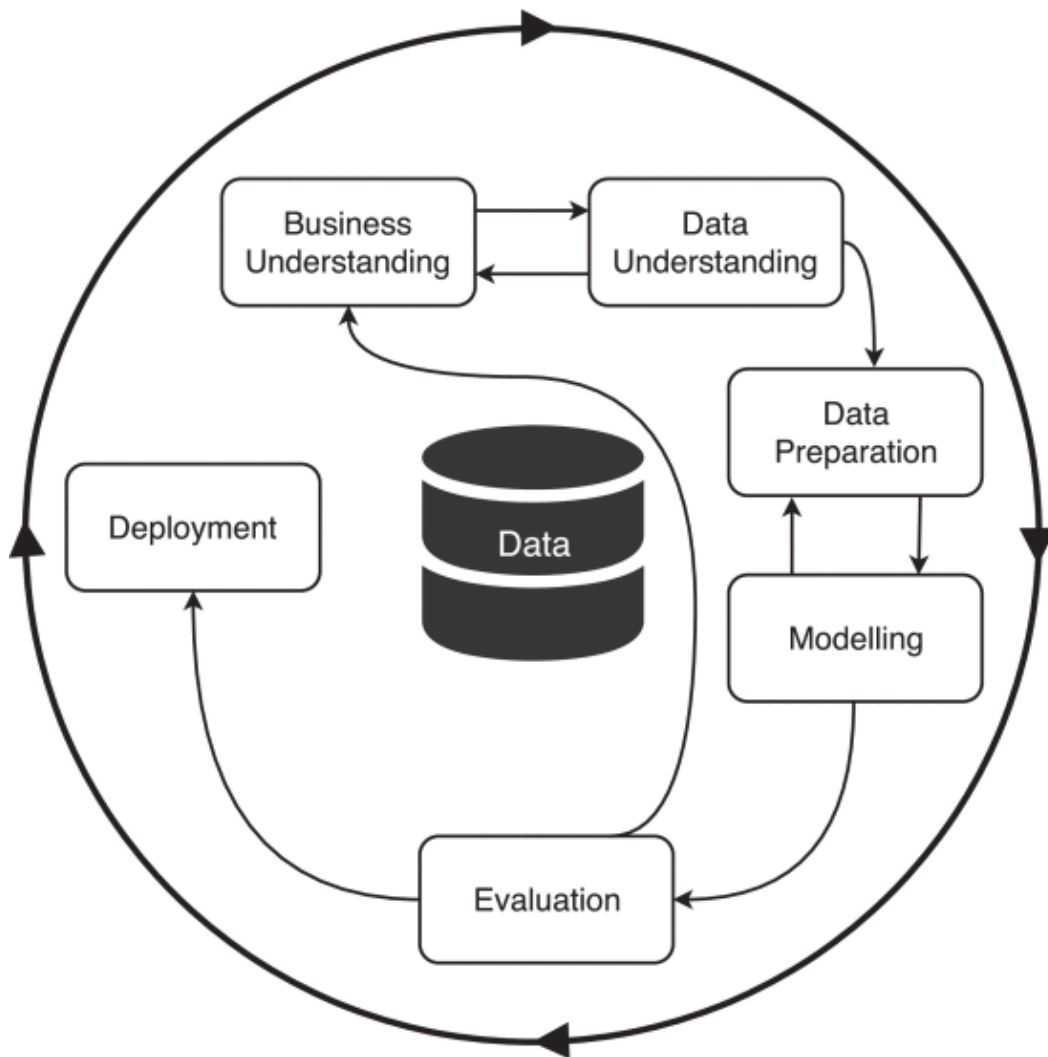
Tämän periaatteen valossa on ensiarvoisen tärkeää tunnistaa, miten datan kerääminen järjestelmästä, ja sen jatkokäsittely voivat ohjata ja tukea liiketoiminnan tavoitteita. CRISP-DM-mallin soveltaminen tässä kontekstissa tarkoittaa, että datan keräämisen suunnittelu ja toteutus integroidaan osaksi liiketoimintaongelman ratkaisuprosessia, varmistaen, että kerättävä data on merkityksellistä, laadukasta ja vastaa asetettuihin analyysin tarpeisiin.

2.1 Datan rooli CRISP-DM mallissa

Tarkastellaan datan vaatimuksia CRISP-DM prosessin vaiheiden kautta. CRISP-DM on kuusivaiheinen prosessi, jota käytetään mm. erilaisissa data-analyysin sekä datalouhinnan projekteissa.

Prosessin vaiheet ovat seuraavat:

1. Liiketoiminnan ymmärtäminen: Tarpeiden ja projektin tavoitteiden ymmärtäminen sekä määrittäminen.
2. Dataymmärtäminen: Tarvittavien datalähteiden tunnistaminen ja soveltuvuuden arviointi.
3. Datavalmistelu: Datan muokkaaminen tarkoitukseen sopivaan muotoon.
4. Mallinnus: Datan mallintaminen erilaisten algoritmien avulla ongelman ratkaisemiseksi.
5. Arviointi: Mallien luotettavuuden ja käyttökelpoisuuden testaaminen
6. Toteutus: Käyttöön otettu ratkaisu, joka ratkaisee alkuperäisen ongelman ja palvelee alkuperäistä tarvetta.



Kuvio 1. CRISP-DM prosessi datahankkeita varten. (IEEE transactions on knowledge and data engineering, 2019).

Vaikka nämä vaiheet esitetään aikajärjestyksessä, eri vaiheiden välillä hypitään yleensä paljon. Merkittävä osa data-analytiikasta perustuu kokemukseen ja parhaita käytäntöjä noudattavaan kokeiluun; ei ole yhtä taikakeinoa, joka toimisi jokaisessa projektissa. Riippuen ratkaistavasta ongelmasta prosessi voi olla melko iteratiivinen ja aikaa vievä. (Delen, 2020, kappale 3).

Käydään seuraavaksi läpi prosessin vaiheet 1–3 jotka ovat liiketoiminnan ymmärtäminen, dataymmärtäminen, datavalmistelu. Muut vaiheet ovat rajattu pois, koska työssä halutaan keskittyä datan keräykseen ja analytiikan mahdollistamista datalähteen näkökulmasta.

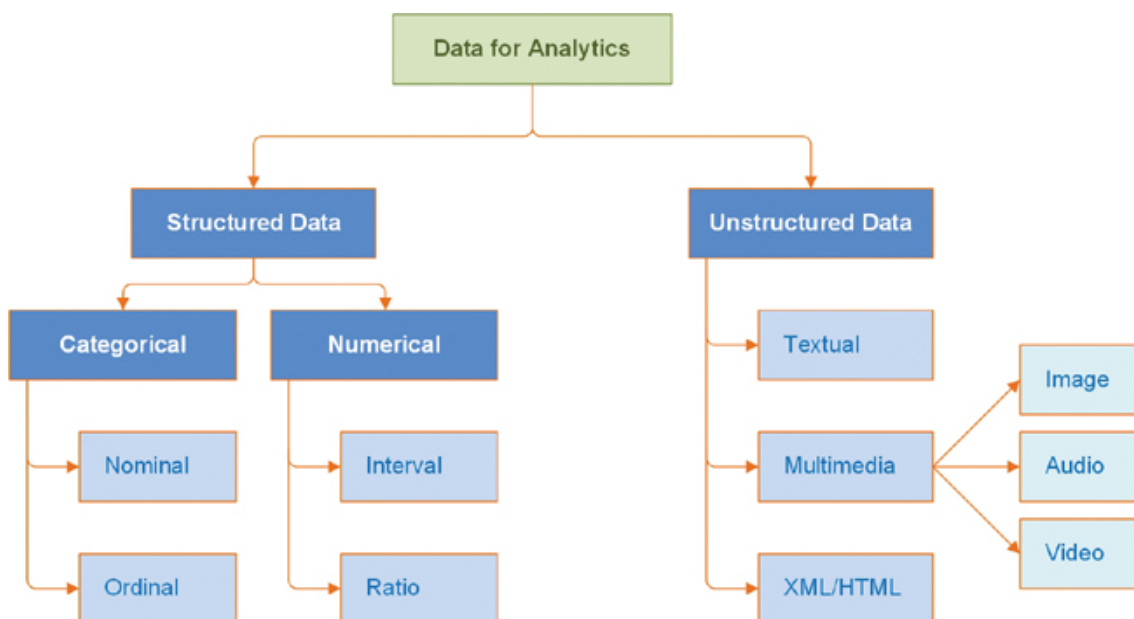
2.2 Liiketoiminnan ymmärtäminen

Datalouhinnan projektin menestys nojaa vahvasti liiketoiminnan tarpeiden syvälliseen ymmärtämiseen ja selkeiden liiketoimintatavoitteiden määrittelyyn. Tämä prosessi sisältää projektisuunnitelman laatimisen, joka kattaa tiedonkeruun, analyysin ja raportoinnin. Esimerkiksi vähittäiskaupan alueella tämä voi tarkoittaa kannattavien asiakassegmenttien tunnistamista ja täsmällistä markkinointia. Ruokakaupoissa tämä voi ilmetä tuotteiden yhdistämisenä, jotka usein ostetaan yhdessä, edistään näin parempaa sijoittelua myymälässä ja kohdennetumpia markkinointikampanjoita. Tällainen lähestymistapa edellyttää tarkkaa liiketoiminnan tavoitteiden määrittelyä ja niiden johdonmukaista seuranta datalouhinnan koko prosessin ajan. (Delen, 2020, kappale 3)

2.3 Dataymmärtäminen

Datalla itsellään tarkoitetaan erilaisia joukkoja numeroita, kirjaimia, tekstiä, ääntä tai muita muuttujia, joita kerätään kokemusten, havaintojen tai kokeiden tuloksena. Sitä voidaan abstraktoida eri kategorioihin, jonka pääkategoriat ovat strukturoitu ja strukturoimaton data. Dataa itsessään pidetään usein alimpana abstraktiotasona, josta tietoa johdetaan. Strukturoitu data on dataa, jota käytetään datalouhinnan projekteissa. (Delen, 2020, kappale 4)

Analytiikkaa varten kerättävä data on strukturoitua dataa. Se voi olla kvantitatiivista tai kvalitatiivista. Kvantitatiivinen eli numeraalinen data voi olla joko diskreettiä tai jatkuvaa. Kvalitatiivinen data on jaettu kahteen osaan: on nominaalista dataa ja ordinaalista dataa. Nominaalinen data koostuu äärellisistä, järjestämättömistä arvoista. Tällaisia arvoja olisivat esimerkiksi siviilisäädyt, kuten naimaton, naimisissa, leski ja sinkku. Arvot edustavat kategorioita tai luokkia, eikä niillä ole luonnollista järjestystä. Ordinaalinen data taas puolestaan voidaan asettaa järjestykseen. Tällaista dataa kuvaisi esimerkiksi luottoluokitus, joka voi olla erinomainen, hyvä tai huono. Alla olevassa kuviossa 2 esitellään datan jakautuminen pääkategorioista eri alakategorioihin. (Delen, 2020, kappale 4)



Kuvio 2. Datan taksonomia tiedonlouhinnassa. (Delen, 2020, kappale 4)

CRISP-DM-prosessin toisessa vaiheessa keskitytään siihen, kuinka luodaan yhteys liiketoimintaongelman ja ongelman ratkaisemiseen käytettävän datan välille. On tärkeää tunnistaa olennainen data saatavilla olevista lähteistä, jotta voidaan vastata liiketoimintaongelman asettamiin kysymyksiin. Datan tunnistamisessa ja valinnassa on otettava huomioon useita keskeisiä seikkoja, jotta voidaan löytää kaikkein olennaisimmat tiedot. Esimerkiksi vähittäiskaupan data-analyysiprojektissa tavoitteena voi olla selvittää naisten kulutustottumuksia ostaessaan kausivaatteita perustuen heidän demografisiinsa, luottokorttitransaktioihinsa ja sosioekonomisiin ominaisuuksiinsa. (Delen, 2020, kappale 3)

2.4 Datavalmistelu

Datavalmistelu eli datan esikäsittely on keskeinen osa CRISP-DM-prosessia. Tässä vaiheessa keskitytään raakadatan jalostamiseen analyysivalmiiksi. Datavalmistelu on aikaa vievää, ja usein se muodostaa jopa 80 % koko projektin työmäärästä. Käsittelyyn käytetty suuri aika johtuu todellisen maailman datan ominaisuuksista. Tyypillisesti kerätty data on epätäydellistä, meluista ja epä johdonmukaista, joka tekee hyvän analyysin tuottamisesta hankalaa. (Delen, 2020, kappale 3)

Keskeisiä vaiheita datan käsittelyssä ovat:

- **Datan puhdistus:** Tavoitteena on virheellisten ja tarpeettomien tietojen oikaisu sekä datan laadun kokonaisvaltainen kohentaminen. Poikkeavat arvot voivat merkittävästi vääristää analyysin tuloksia ja johtaa virheellisiin johtopäätöksiin. Arvot voidaan joko korjata sopivalla tilastollisella menetelmällä tai poistaa analyysistä kokonaan. Samoin duplikaattitiedot poistetaan välttämään päällekkäisyyksien aiheuttamaa vääristymää. Puuttuvia arvoja voidaan poistaa tai täydentää aikaisempaan dataan perustuvalla keskiarvolla tai muilla tilastollisilla menetelmillä, joiden avulla pyritään säilyttämään datan eheys ja parantamaan sen analysoitavuutta.
- **Data Transformation:** Erilaisten datatyypin muokkaamista eri lähteistä yhtenäiseen muotoon, jolla helpotetaan analyysiä.

Tehokas datavalmistelu luo perustan oivaltavalle ja tarkalle data-analyysille. Varmistamalla datan laadun ja johdonmukaisuuden, analyytikot voivat rakentaa merkityksellisiä malleja ja tarjota tietoon perustuvia analyyskejä. Asianmukaisen datavalmistelun laiminlyöminen voi johtaa harhaanjohtaviin johtopäätöksiin, mikä korostaa sen tärkeyttä kaikessa data-analytiikassa. (Delen, 2020, kappale 3)

2.5 Datan visualisointi

Datan visualisointi tarkoittaa datan, kuten numeroiden tai informaation kuvaamista visuaalisessa muodossa. Nykyaikana datan visualisointi nähdään monilla aloilla visuaalisen viestinnän modernina vastineena. Sen päätavoitteena on monimutkaisen tiedon yksinkertaistaminen ja sen esittäminen tavalla, joka on helposti ymmärrettävissä myös aiheeseen perehtymättömälle yleisölle. Merkityksellisten visualisointien luominen on myös haastavaa. Visualisoinnin tulee tarjota selkeä ja ymmärrettävä kuva datasta, mutta monimutkaisuus ja suuri koko voivat tehdä tästä vaikeaa. Tietojen asianmukainen järjestäminen ja tarkoituksenmukainen visualisointi ovat avainasemassa tehokkaan visualisoinnin aikaansaamiseksi. (Miller, 2017, kappale 1)

Tietoja voidaan visualisoida monin eri tavoin, riippuen tiedon luonteesta ja siitä, mitä halutaan esittää. IBM:n verkkosivuilla (Types of data visualizations, IBM, n.d) kuvataan useita yleisesti käytettyjä datan visualisointimenetelmiä, joita ovat:

- **Taulukot:** Tapa esittää yksityiskohtaista dataa, mutta suurien datamäärien kohdalla voi olla vaikeaselkoinen.
- **Piirakka- ja pylväsdiagrammit:** Havainnollistavat osuuksia kokonaisuudesta, helppo lukea ja ymmärtää.
- **Viiva- ja aluekaaviot:** Kuvaavat muutoksia ajan myötä, sopivat trendien seuraamiseen.
- **Histogrammit:** Näyttävät datan jakautumisen, auttavat tunnistamaan poikkeavuuksia.
- **Hajontakaaviot ja kuplakaaviot:** Paljastavat suhteet kahden tai useamman muuttujan välillä, kuplakaaviot lisäävät kolmannen ulottuvuuden (kuplan koko).
- **Lämpökartat:** Värien intensiteetillä ilmaistaan arvojen vaihteluita, hyödyllisiä sijaintidatan esittämisessä.
- **Puukaaviot:** Esittävät hierarkista dataa, sopivat kategorioiden suhteellisten osuuksien vertailuun.

2.6 Yksityisyys, tietosuojaja ja datan etiikka

Euroopan unionin yleisen tietosuojasetuksen (GDPR) mukaan henkilötietojen suhteen on noudatettava periaatteita, jotka varmistavat lainmukaisen, asianmukaisen ja läpinäkyvän käsittelyn rekisteröidyn kannalta. Henkilötietoja on kerättävä nimenomaisia ja laillisia tarkoituksia varten, ja niiden käsittelyn on oltava näiden tarkoitusten kanssa yhteensopivaa. Tietojen minimointi, täsmällisyys ja säilytyksen rajoittaminen ovat keskeisiä periaatteita, joiden avulla pyritään turvaamaan henkilötietojen asianmukainen turvallisuus. (Digiturvamalli.fi, GDPR:n periaatteet, n.d)

Tietosuojalaki puolestaan täsmentää ja täydentää GDPR:n soveltamista Suomessa. Laki määrittelee tarkemmin, missä tilanteissa henkilötietoja voidaan käsitellä ja mitkä ovat rekisteröidyn oikeudet. Esimerkiksi tietosuojalaki sisältää säännöksiä erityisten henkilötietoryhmien käsittelystä, kuten terveyteen liittyvistä

tiedoista, ja määrittelee tietosuojavaltuutetun tehtävät ja toimivaltuudet. Lain tavoitteena on varmistaa henkilötietojen suojaaminen ja oikeuksien toteutuminen tietoyhteiskunnassa. (Finlex.fi, Tietosuojalaki)

GDPR ja kansallinen tietosuojalaki tarjoavat ohjeistukset henkilötietojen käsittelylle, korostaen tarvetta noudattaa lainmukaisuutta, läpinäkyvyyttä ja tietojen minimointia. Näiden säännösten mukainen toiminta edellyttää organisaatioilta selkeitä käytäntöjä datan keräämiseen, käsittelyyn ja säilyttämiseen, varmistaen samalla, että käyttäjän oikeudet toteutuvat. Henkilötietoja tulisi kerätä vain määritettyjä, laillisia tarkoituksia varten ja käsittelyn rajoittua näiden tarkoitusten mukaisesti. Tietosuojan periaatteiden soveltaminen päivittäisessä toiminnassa edistää yksilöiden luottamusta siihen, että heidän tietojensa käsitellään oikeudenmukaisesti ja turvallisesti.

Datan keräämisen kasvava merkitys tuo mukanaan myös monia eettisiä haasteita, jotka pitävät sisällään keräysmenetelmät, käyttäjän suostumuksen, tietoturvan sekä lainsäädännön. Keskiössä on käyttäjien itsemääräämisoikeus ja kontrolli omiin tietoihinsa, mikä nostaa esiin kysymyksiä suostumuksen pyytämisestä ja henkilökohtaisten tietojen käyttötarkoituksista. Eettinen käyttö edellyttää tasapainon löytämistä yhteiskunnallisten hyötyjen ja yksilöiden oikeuksien välille, painottaen läpinäkyvyyttä, osallistumista ja vastuullisuutta datan käytössä. Tässä yhteydessä korostuu tarve tietoturvalliseen käsittelyyn ja datan eettisen käytön periaatteiden vahvistamiseen. (Bellatyne, A, 2018)

3. VERKKOMAINONTA

Verkkomainonta eli digimainonta on osa digitaalista markkinointia, joka hyödyntää internetin tarjoamia kanavia mainosviestien välittämiseen kohderyhmille. Tähän kategoriaan kuuluvat mainosmuodot kattavat laajan kirjon erilaisia strategioita ja kanavia, joista keskeisimpiä ovat bannerimainonta, videomainonta, sosiaalisen median mainonta, hakukonemainonta ja sisältömarkkinointi. Verkkomainonnan perusajatuksena on hyödyntää digitaalista alustaa viestin levittämiseen mahdollisimman tehokkaasti ja kohdennetusti, käyttäen hyväksi teknologiaa, joka mahdollistaa mainosviestien personoinnin ja kohdentamisen erityisille kohderyhmille.

- **Luokiteltu ilmoittelu:** Verkkosivustojen kategorioihin jaoteltu maksullinen ilmoittelu.
- **Display-mainonta:** Näkyvä graafinen mainonta, joka sisältää kuvia ja videoita, ja joka on tyypillistä verkkosivustoilla.
- **Rich Media:** Interaktiiviset ja toiminnalliset mainokset, jotka voivat sisältää liikkuvaa kuvaa ja ääntä.
- **Sisältömarkkinointi:** Hyödyllisen ja kiinnostavan sisällön tuottaminen ja jakaminen tarkoituksenaan houkuttaa ja sitouttaa kohdeyleisöä.
- **Videomainonta:** Mainonnan muoto, jossa käytetään videota viestin välittämiseen.
- **Hakusanamainonta:** Mainonta hakukoneissa käyttäjän käyttämien hakusanojen perusteella.
- **Mainonta sosiaalisessa mediassa:** Mainonta sosiaalisen median alustoilla kuten Facebook ja Instagram.

Nämä kaikki muodostavat verkkomainonnan kokonaisuuden, jossa yhdistyy teknologian mahdollistama personointi ja kohdentaminen. Digimainonta mahdollistaa viestien tehokkaan levittämisen digitaalisissa ympäristöissä, ja se hyödyntää kohderyhmien tietoja ja käyttäytymistä mainosviestien optimoimiseksi. (Jormalainen, Liikkanen, Morri, Ojala, Runila, Stenberg, Digimainonnan ABC, Alma Media Oy)

3.1 Ohjelmallinen mainonta

Ohjelmallinen mainonta tarkoittaa automatisoitua prosessia, jossa mainostilaa ostetaan reaaliaikaisesti huutokauppaperiaatteella. Tämä mahdollistaa mainostajille tarkemman ja tehokkaamman kohdennuksen kohdeyleisölleen, sillä mainokset voidaan suunnata erittäin tarkasti käyttäjien toiminnan ja mieltymysten perusteella. Sen ytimessä on algoritmien ja datan hyödyntäminen päätöksenteossa. Mainostilaa ei enää myydä ja osteta manuaalisesti, vaan käytetään alustoja ja teknologioita, jotka automatisoivat näitä prosesseja. Mainosnäyttöjen osto tapahtuu reaaliajassa käyttäjän selatessa verkkosivustoa, mikä mahdollistaa mainostajille mahdollisuuden tavoittaa yleisönsä oikeassa kontekstissa ja oikealla hetkellä. (Alaimo & Kallinikos, 2018)

Ohjelmallisen mainonnan ekosysteemi koostuu useista toimijoista, mukaan lukien mainostajat, julkaisijat, mainosverkostot, kysyntäpuolen alustat (DSP), tarjontapuolen alustat (SSP) ja datanhallintapalvelut (DMP). Näiden toimijoiden yhteistyö ja vuorovaikutus mahdollistavat dynaamisen järjestelmän, jossa mainoksia voidaan kohdentaa entistä tarkemmin ja tehokkaammin. (Alaimo & Kallinikos, 2018)

3.2 Mittarit ja seuranta

Digitaalisessa markkinoinnissa mainonnan seurannan ja mittaamisen tärkeys korostuu, kun halutaan varmistaa toimenpiteiden tehokkuus ja kustannustehokkuus. Keskeisimpiä mittareita ovat klikkausprosentti (CTR), joka kuvaa mainosten houkuttelevuutta ja yleisön reagointia, sekä konversioprosentti (CVR), joka kertoo, kuinka monta klikkausta johtaa toivottuun lopputulokseen, kuten ostokseen. Mainoskustannus per konversio (CPA) on myös olennainen mittari, joka osoittaa yksittäisen konversion hankinnan hinnan ja auttaa arvioimaan kustannustehokkuutta. Lisäksi, mittarit kuten cost-per-thousand (CPM), joka ilmoittaa tuhannen mainosnäytön hinnan, ja cost per click (CPC), joka osoittaa yksittäisen klikin hankintakulun. Mittarit antavat kattavaa tietoa mainoskampanjoiden taloudellisesta suorituskyvystä. Markkinoijien ja mainostajien on tärkeää ymmärtää

näiden mittareiden merkitys ja soveltaa niitä kampanjoidensa jatkuvaan kehittämiseen ja optimointiin, joka voi parantaa sijoitetun pääoman tuottoa (ROI) ja vahvistaa markkinointistrategiaa. (IAB Finland Ry, 2009)

4. VERKKOANALYTIikka

Verkkoanalytiikka tarkoittaa käyttäjien vuorovaikutusten tutkimista verkkosivustojen ja mobiilisovellusten kanssa keräämällä ja analysoimalla heidän käyttäytymisestään kerättävää dataa. Tiedonkeruun laajuus erottaa verkkoanalytiikan muista käyttäjätutkimusmenetelmistä. Verkkoanalytiikka kerää verkkosivujen käyttäjien toimista erilaista dataa, kuten mitä sivuja käyttäjä katselee, milloin ja missä järjestyksessä. Tämä käyttäjän toiminnan seuranta mahdollistaa verkkoanalytiikkatyökalujen luoda kertomuksen siitä, miten kukin käyttäjä on liikkunut verkkosivuston läpi. Toiminnasta kerätään myös erilaisia teknisiä yksityiskohtia, kuten tietoja käyttäjän selaimesta tai näytön resoluution tarkkuudesta. Oikean työkalun käytöllä tai integroimalla omaa seurantakoodia sivustolle voidaan tallentaa ja analysoida lähes mitä tahansa, mitä käyttäjä tekee verkkosivustolla ja tarvittaessa yhdistää nämä tiedot muihin tietoihin. Analyysityökalut ovat kehittyneet huomattavasti 2000-luvun puolivälistä lähtien ja ovat pääasiassa käytössä online-markkinoinnissa kampanjoiden tehokkuuden mittaamisessa, seuraten mittareita kuten vierailijoiden muuntamista suhteessa markkinointimenoihin. (Beasley, 2013, kappale 1)

Verkkoanalytiikka on merkittävästi parantanut online-markkinointia mahdollistamalla markkinoijien strategioiden tehokkuuden mittaamisen. Tavoitteena on seurata erilaisia mittareita, kuten konversioita. (Beasley, 2013, kappale 1) Google määrittelee konversion toiminnoksi, joka lasketaan, kun joku vuorovaikuttaa mainoksen (esimerkiksi klikkaamalla teksti-ilmoitusta tai katsomalla videoilmoitusta) ja sen jälkeen suorittaa toiminnon, jonka mainostaja on määritellyt arvokkaaksi liiketoiminnalleen. Tällainen toiminto voi olla esimerkiksi ostoksen tekeminen verkossa tai soitto mainostajalle. (Google Ads Help, n.d) Kerätty data tarjoaa selkeää näkemystä asiakashankinnan tehokkuudesta vertaamalla onnistuneita muunnoksia markkinointi-investointeihin. Verkkoanalytiikka on ratkaisevassa roolissa markkinointikulujen ja -strategioiden optimoinnissa, johtaen paremmin informoituihin päätöksiin ja korkeampaan sijoitetun pääoman tuottoon. (Beasley, 2013, kappale 1)

Tehokas verkkoanalytiikka edellyttää ensisijaisesti kattavaa verkkosivuston ymmärtämistä. Sivuston rakenteen, sisällön ja toimintojen tunteminen on välttämätön vaihe ennen analytiikkatyökalujen käyttöönottoa. Sivuston perusteellinen tuntemus muuttaa analytiikkatiedot pelkistä numeroista ja kaavioista merkitykselliseksi tiedoiksi, jotka kuvastavat sivuston todellista käyttöä ja käyttäjien vuorovaikutusta sen kanssa. (Beasley, 2013, kappale 2.)

Datan keräys alkaa kysymyksestä tai tietämyksen puutteesta, joka halutaan täyttää. Kysymys määrittelee toiminnan rajat ja auttaa ymmärtämään, kun tavoite on saavutettu tai milloin on aika lopettaa, jos tavoitetta ei päästä lähemmäksi. Verkkoanalytiikka ei itsessään anna vastauksia "miksi"-kysymyksiin. Sen sijaan se tarjoaa työkaluja "mitä"-kysymysten tutkimiseen. Käytännössä tämä tarkoittaa analytiikan hyödyntämistä käyttäjien nykyisen toiminnan havainnointiin. Tätä tietoa voidaan käyttää joko teorioiden testaamiseen tai lähtökohtana verkkosivuston syvällisemmälle arvioinnille muilla menetelmillä. (Beasley, 2013, kappale 2.)

OMTM (One Metric That Matters) on tapa analytiikan keräyksen määrittelylle ja rajaamiselle. Käytännössä OMTM auttaa keskittämään resurssit ja toimet kriittisiin KPI:hin (Key Performance Indicator), joka mahdollistaa nopean oppimisen, tehokkaamman päätöksenteon sekä selkeän suunnan yrityksen kehitykselle. OMTM:n avulla yritykset voivat välttää hajanaisuutta ja mitata sitä, mikä on todella tärkeää niiden kasvun tai kehityksen kannalta. Pienten testien tekeminen on helpompaa, mikä auttaa välttämään suurempia virheitä ja oppimaan jatkuvasti. Säännöllinen arviointi ja hienosäätö varmistavat, että kerättävä metriikka pysyy relevanttina ja hyödyllisenä organisaation tavoitteiden saavuttamiseksi. (Croll, A. & Yoskovitz, B. 2024, kappale 6)

Verkkoanalytiikan syvällisempi hyödyntäminen esimerkiksi digitaalisessa markkinoinnissa avaa uusia ovia asiakasymmärrykseen ja sivustokokemuksen optimointiin. Sen avulla voidaan tunnistaa käyttäjien käyttäytymismalleja ja ennustaa tulevia trendejä, mikä mahdollistaa kohdennetumpien ja henkilökohtaisempien markkinointistrategioiden kehittämisen. Kerätyn datan ja siitä tehdyn analyysin avulla voidaan parantaa asiakaskokemusta ja tehostaa markkinointibudjetin käyttöä. Analytiikan käyttö edellyttää jatkuvaa sitoutumista oppimiseen ja prosessien

kehittämiseen, joka korostaa koulutuksen ja teknologian integraation tärkeyttä yrityksissä. (Chaffey & Patron. 2012)

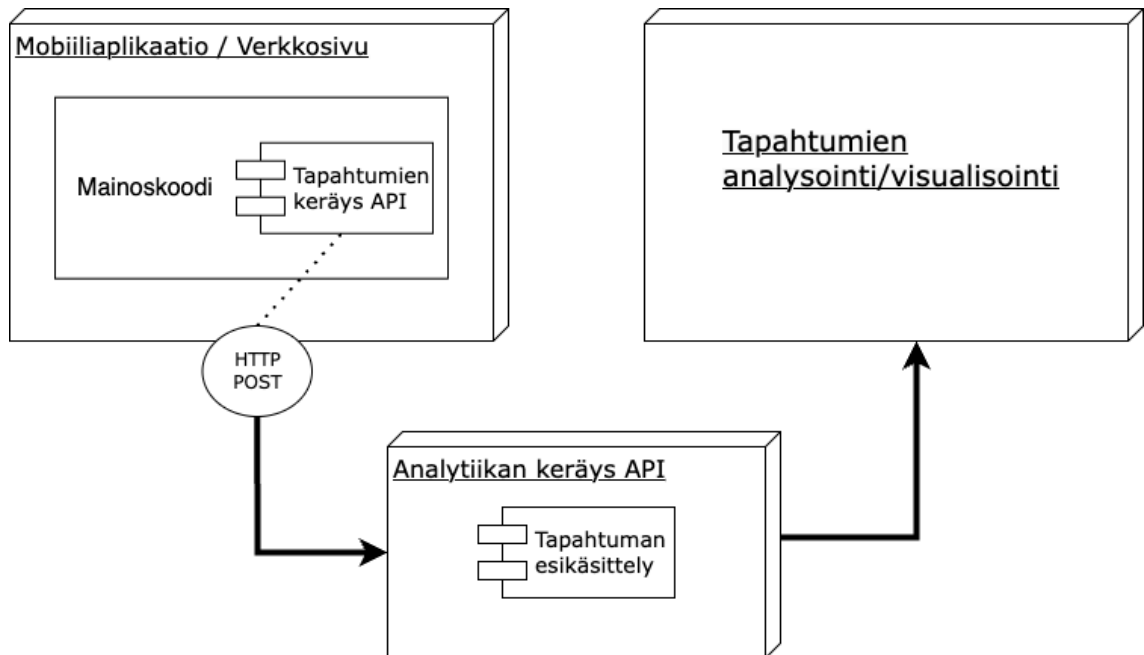
5. VERKKOMAINONNAN TIEDONKERUUJÄRJESTELMÄ

Opinnäytetyön tavoitteena on kehittää mainoskoodijakeluun ominaisuus, joka tarjoaa tehokkaan tavan seurata sen suoritusta erilaisilla selain- ja käyttöjärjestelmäyhdistelmillä Alma Median palveluverkoston verkkopalveluista kerätyn datan avulla. Toteutettavan keräyksen vastuulla on kolme tapahtumaa: Datan keräys, suodatus sekä lähetys analysoitavaksi. Kerättävä data koostuu kahdesta pääkomponentista: virheistä ja tapahtumista. Virheet voivat olla esimerkiksi teknisiä virheitä tai koodin suoritusongelmia, jotka vaikuttavat sovelluksen toimintaan. Tapahtumat voivat olla käyttäjän toimia tai muita merkittäviä mainonnan näyttämiseen liittyviä tapahtumia sovelluksen elinkaaren aikana.

Tarkoituksena on tarjota tuotteiden kehittäjille, ylläpitäjille, sekä liiketoiminnan ohjausryhmille työkalut datan keräämiseen sekä seurantaan, joka auttaa heitä ymmärtämään paremmin palveluiden mainonnan teknistä suorituskykyä, käyttäjien käyttötapoja ja liiketoiminnan suoritusindikaattoreita. Tämän avulla työkalujen käyttäjät voivat tehdä tietoon perustuvia päätöksiä sovelluksen parantamiseksi, optimoimiseksi ja liiketoiminnan strategioiden kehittämiseksi.

Datan keräystä voidaan konfiguroida halutuilla parametreilla eri tilanteiden mukaan, mikä mahdollistaa joustavan ja kohdennetun datan keruun. Esimerkiksi jos tietyssä palvelussa halutaan seurata tietynlaista käyttäjätoimintaa tai virhetilannetta, voidaan asettaa järjestelmä keräämään juuri näitä tietoja. Räätelöintimahdollisuus tekee järjestelmästä tehokkaan työkalun sovellusten kehitykseen ja ylläpitoon laajassa verkostossa, jossa on monipuolisesti eri kokoisia ja eri käyttäjämääriltään vaihtelevia palveluita.

Alla olevassa kuviossa (Kuvio 3.) esitellään arkkitehtuurikaavio mainoskoodin suoritusta seuraavalle järjestelmälle. Kaavio kuvaa, kuinka verkkosivu integroituu mainoskoodin kanssa ja kuinka tapahtumien keräys API kerää tapahtumadataa. Tämän jälkeen data siirretään HTTP POST -pyynnön avulla datan vastaanottopisteeseen (data endpoint), jossa tapahtumien käsittely, tallennus ja visualisointi suoritetaan. Kaavio auttaa hahmottamaan eri komponenttien välisiä suhteita sekä datan kulkua järjestelmässä.



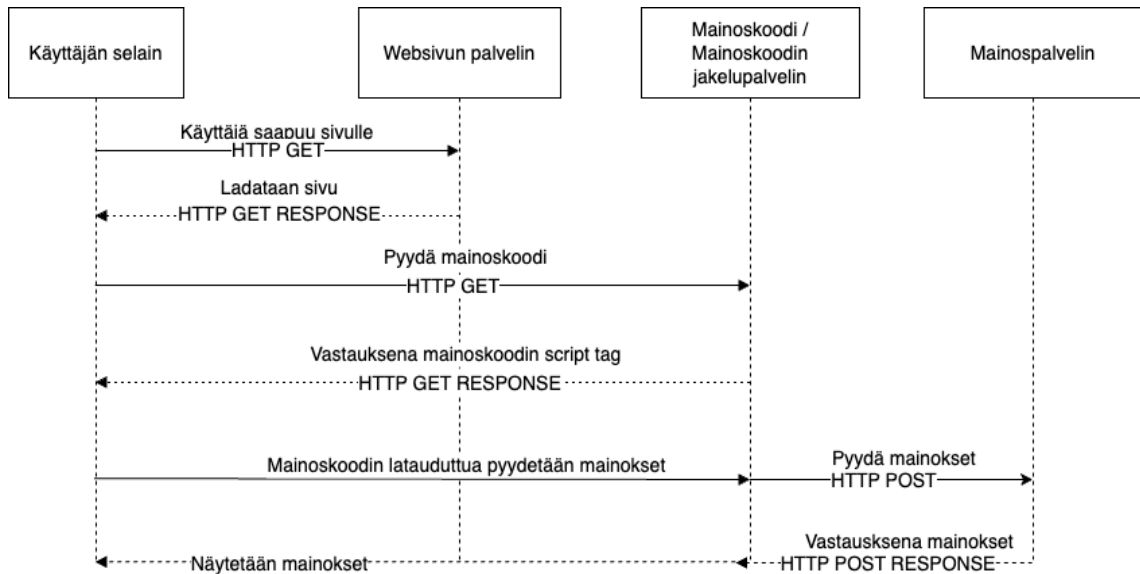
Kuvio 3. Datan keräys sekä lähetys palvelusta

Kerätty tapahtumadata käy läpi käsittelyn, jossa eritellään ja aggregoidaan dataa niin, että se muuttuu hyödylliseksi tiedoksi. Visualisointivaiheessa prosessoitu data esitetään visuaalisessa muodossa, kuten kaaviona tai graafina, joka helpottaa datan ymmärtämistä ja tulkitsemista. Visualisoinnin avulla havainnollistetaan datan trendejä, kuvioita ja poikkeavuuksia, mikä tekee datajoukoista helpommin tulkittavia ja mahdollistaa paremman päätöksenteon niiden pohjalta.

5.1 Tekninen tausta

Tässä kappaleessa käsitellään lyhyesti verkkomainonnan teknistä toimintaa korkealla tasolla. Tarkoituksena on antaa yleiskuva verkkomainonnan toiminnasta sekä taustatietoa, joka auttaa ymmärtämään sen toimintaa. Ymmärrys tästä teknologiasta on tärkeää, sillä opinnäytetyö pohjautuu tästä järjestelmästä kerättävään tietoon.

Tutustumme tekniseen toteutukseen sekvenssikaavion (Kuvio 4.) kautta, joka kuvaa käyttäjän saapumista verkkosivulle. Kaavio valottaa korkealla tasolla selaimen ja palvelimen välisen viestinnän eri vaiheita, alkaen ensimmäisestä HTTP GET -pyynnöstä ja päättyen mainosten näyttämiseen loppukäyttäjän selaimessa.



Kuvio 4. Käyttäjän ja palvelimen välinen viestintä mainoskoodin lataamista ja näyttämistä varten.

Kun käyttäjä saapuu verkkosivulle, selaimen ja palvelimen välinen viestintä alkaa välittömästi. Selain lähettää palvelimelle HTTP GET -pyynnön, jolla se hakee sivuston sisältöä. Palvelin vastaa tähän pyyntöön lähettämällä usein HTML-tiedoston, CSS-tyylitiedostot ja JavaScript-koodin sekä mahdolliset muut tarvittavat tiedostot HTTP-vastauksena. Nämä tiedostot muodostavat sivuston ja mahdollistavat toiminnallisuuden. HTTP-vastaus voi sisältää monenlaisia elementtejä, kuten kuvia, tyylitiedostoja ja skriptejä. Selain voi lähettää selauksen aikana lisää HTTP-pyyntöjä palvelimelle lisäresurssien hakemiseksi, kuten muita HTML-sivuja, JavaScript kirjastoja ja skriptejä, analytiikkatyökaluja ja muita taustalla tapahtuvia toimintoja, jotta se voi täydentää sivun kokonaisuuden ja varmistaa sen toimivuuden.

Yksi haetuista skripteistä on mainonnan jakelukoodi, joka mahdollistaa mainosten näyttämisen verkkosivulla. Mainoskoodi haetaan sisällönjakeluverkosta (CDN), joka sitten ladataan käyttäjän selaimeen. Kun mainoskoodi on ladattu, se käyttää sivulta saatuja tietoja pyytääkseen mainospalvelimelta mainoksia. Tällaisia tietoja voivat olla esimerkiksi yleisökohortit, sivun konteksti ja selaintiedot. Yleisökohortit ovat käyttäjäryhmiä, jotka sisältävät tietoa käyttäjän demografisista tiedoista, kuten ikäryhmästä, sijainnista, tulo- ja koulutustasosta, kiinnostuksen kohteista ja käyttäytymisestä. Sivun konteksti viittaa sivun otsikkoon, sisältöön ja

sen avainsanoihin. Selaintiedot sisältävät tietoja käyttäjän selaimesta ja laitteesta, kuten käyttöjärjestelmästä ja selainversiosta. Lopuksi mainospalvelin toimittaa käyttäjälle mainoskoodin pyytämät mainokset, jotka mainoskoodi sitten näyttää selaimessa.

5.2 Analytiikan tarpeet

Tiedonkeruujärjestelmän tarpeena on tuottaa dataa kahden keskeisen komponentin toiminnasta: mainoskutsut ja niihin liittyvä data sekä mainonnan jakelukoodin kohtaamat virheet.

Mainoskutsuja tarkasteltaessa haluttiin kerätä sellaista dataa, jolla voidaan ymmärtää, miten mainonnanjakelukoodi toimii eri ympäristöissä, ja tunnistaa ongelmat, jotka vaikuttavat mainosten tehokkaaseen toimittamiseen erilaisissa selain-, käyttöjärjestelmä- sekä sovellusyhdistelmässä. Mainoskutsuista kerätään useita aikaleimoja, joiden avulla saadaan tarkempi kuva mainonnanjakelukoodin toiminnasta eri ympäristöissä. Nämä aikaleimat auttavat tunnistamaan mahdollisia pulonkaloja tai ongelmia, jotka voivat vaikuttaa mainosten tehokkaaseen toimittamiseen.

Mainoskoodin virheiden ymmärtäminen eri käyttöympäristöissä on tärkeää. Ongelmaa lähestyttiin monitahoisesti, tarkastellen laajasti koodin suoritusta ja sen mahdollisia virhetilanteita ja niiden syitä. Kattava virheiden seuranta ja analyysi auttaa tunnistamaan ja korjaamaan suorituskykyongelmia, jotka voivat vaikuttaa laajasti mainonnan kokonaistoimintaan, mainosten latautumisaikoihin, näkyvyyteen, ja lopulta käyttäjäkokemukseen.

Mainoskutsujen ja virhetietojen mukana haluttiin myös tietoa käyttäjän laitteesta, ja verkosta yleistä dataa, joka tarjoaisi laajempaa kontekstia mainonnanjakelun ja sivuston toiminnan ymmärtämiseen. Tämä sisälsi yleisiä tietoja selaimesta kuten käyttöjärjestelmän, selainversion, laitetyypin, sivustolle saapumisen lähteen, sivuston osoitteen ja sivun nimen tai otsikon. Lisäksi pyrittiin keräämään tietoa verkko-olosuhteista, kuten verkon nopeudesta ja käytetystä yhteydestä (esim. WiFi tai mobiilidata). Tällainen data auttaa hahmottamaan, miten mainokset ja sivut toimivat erilaisissa ympäristöissä ja käyttötilanteissa, mikä puolestaan voi

ohjata kehitystyötä ja optimointia entistä paremman käyttäjäkokemuksen saavuttamiseksi. Kerätty data on esiteltyä alla taulukossa 1.

TAULUKKO 1. Selaimesta kerättävää dataa

Atribuutti	Selite
height	Selaimen näytön korkeus pikseleissä
width	Selaimen näytön leveys pikseleissä
deviceType	Laite, jolla sivua tarkastellaan
connectionType	Verkkoyhteyden tyyppi, jos saatavilla
userAgent	Selaimen käyttäjäagentti, joka sisältää tietoja selaimesta ja käyttöjärjestelmästä
locationHost	Sivun verkko-osoitteen isäntänimi
pageTitle	Verkkosivun otsikko
referrer	URL-osoite, josta käyttäjä tuli nykyiselle sivulle.

5.3 Mainoskutsujen datan kerääminen

Mainoskutsu on tapahtuma, jossa verkkosivu lähettää pyynnön mainospalvelimelle saadakseen näytettäväksi sopivan mainoksen. Tämä pyyntö lähettää mukanaan erilaisia parametreja, jotka kuvaavat, millaista mainosta ollaan hakemassa. Parametrit voivat sisältää tietoa käyttäjän laitteesta, selaimesta, sijainnista, sekä mahdollisesti käyttäjän kiinnostuksen kohteista, aiemmasta käyttäytymisestä sekä halutun mainoksen koosta ja tyypistä.

5.3.0 Kerättävä data

Kun mainospalvelin vastaa HTTP-pyyntöä kautta, vastaus on JSON (JavaScript Object Notation) -muodossa, joka mahdollistaa tietorakenteiden, kuten mainoksen metatietojen, formaatin ja hinnoittelutietojen siirtämisen palvelimelta selaimen. JSON-vastauksen sisältämä tieto voidaan sitten purkaa ja käsitellä

verkkosivulla tai sovelluksessa ja lopulta näyttää sen avulla mainoksen sille varatussa tilassa. Vastaus sisältää tietoa, jota voidaan kerätä ja analysoida mainonnan tehokkuuden parantamiseksi. Kerättävä tieto on esiteltyä alla taulukossa 2.

TAULUKKO 2. Mainoskutsuista kerättävää dataa

Atribuutti	Kuvaus
UUID	Tapahtuman uniikki tunniste
adSection	Kuvaa, missä osassa verkkosivua tai sovellusta mainos näytetään
requestIndex	Kuvaa pyynnön järjestysnumeron mainoskutsujen sarjassa
adSlots	Lista mainospaikoista, joihin mainokset on tarkoitus sijoittaa
placementId	Mainospaikan yksilöllinen tunniste mainospalvelimella
sizes	Mainospaikan sallimat koot
state	Mainospaikan tila, kuten loaded (ladattu) tai available (valmiina näytettäväksi)
cpm	Cost Per Mille (CPM), eli hinta tuhannelta näytöltä
impressionSource	Tiedon lähde mainoksen näyttökerasta
timestamp	Tapahtuman aikaleima

AdSection kertoo, missä osassa verkkosivua tai sovellusta mainos näytetään, antaen käsityksen mainoksen näkyvyydestä ja mahdollisesta vuorovaikutuksesta käyttäjän kanssa. Tämä tieto auttaa ymmärtämään, mitkä sivuston osat houkuttelevat eniten huomiota ja missä mainokset saavuttavat parhaan suorituskyvyn.

RequestIndex ja adSlots tarjoavat näkymän mainoskutsujen dynamiikkaan ja mainospaikkojen hallintaan. Tieto siitä, kuinka mones pyyntö tietyssä sarjassa on, sekä missä mainokset sijoitetaan, auttaa analysoimaan mainospaikkojen tehokkuutta. Erityisen arvokasta on miten tieto voi avata kylmä- ja kuormalatausten

eroja, jotka ovat keskeisiä mainosten latautumisenopeuden ja sitä kautta käyttäjäkokemuksen kannalta.

PlacementId ja sizes ovat keskeisiä mainospaikan määrittelyssä ja sen sallimissa ko'issa, jotka ovat välttämättömiä mainosten visuaalisen sovituksen kannalta. Tämä auttaa varmistamaan, että mainokset näyttävät hyvältä ja toimivat tehokkaasti erilaisissa näyttöympäristöissä.

State, cpm ja impressionSource ovat myös tärkeitä metriikoita, jotka tarjoavat tietoa mainospaikan tilasta, mainonnan kustannustehokkuudesta ja mainoksen näyttökertojen lähteestä. Tiedot ovat olennaisia mainospaikkojen suorituskyvyn seurannassa ja optimoinnissa, auttaen mainostajia ymmärtämään, mitkä strategiat toimivat parhaiten ja missä on parantamisen varaa. State kertoo mainospaikan tilasta, esimerkiksi onko mainos ladattu ja näytetty onnistuneesti. Tämän tiedon avulla voidaan tunnistaa teknisiä ongelmia liittyen tiettyihin paikkoihin, jotka vaikuttavat mainoksen näkyvyyteen ja käyttäjäkokemukseen.

ImpressionSource tarjoaa tietoa siitä, mistä mainoksen näyttökerrat ovat peräisin, mikä on keskeistä ymmärtää mainonnan monimuotoisuutta ja kohdennusten tehokkuutta. Mainoksen näyttökerrat voivat tulla erilaisista lähteistä, kuten suoramyytyinä tai ohjelmallisen mainonnan ostoalustoilta.

Tämän datan lisäksi kerättiin mainoskutsudataan yhdistettävää yleistä aikaleimatietoa. Näiden tietojen tarkoituksena on kerätä tietoa siitä, kuinka mainoskutsut ajoittuvat suhteessa sivun latausprosessiin ja muihin tapahtumiin. Analysoimalla näitä aikaleimoja voidaan ymmärtää paremmin mainosten latautumisen vaikutusta käyttäjäkokemukseen, sekä tunnistaa mahdollisia pullonkauloja sivun suorituskyvyssä. Tämän datan avulla voidaan myös arvioida, onko mainoskoodin käytössä, latauksessa tai mainoskutsujen ajoituksessa parantamisen varaa, jotta sivun kokonaislatausaikaa saataisiin lyhennettyä ja käyttäjäkokemusta parannettua. Datan analyysin tulosten perusteella voidaan tehdä suosituksia eri palveluille teknisten muutosten tekemiseksi, jotka parantavat sivuston toimivuutta sekä mainosten tehokkuutta. Kerättävä data esitelty alla taulukossa 3.

TAULUKKO 3. Kerättävä aikaleimadata

Aikaleiman nimi	Selite
navigationStart	Sivun navigoinnin aloitus.
stackFetchStart	Mainoskoodiskriptin haun aloitus.
stackFetchEnd	Mainoskoodiskriptin haun loppuminen.
domContentLoaded	DOM:n latauksen valmistuminen (Document Object Model).
stackLoadStart	Mainoskoodi scriptin latauksen aloitus.
stackLoadEnd	Mainoskoodi scriptin latauksen loppuminen.

5.3.1 Kerääminen erilaisista arkkitehtuureista

Mainoskutsudatan keräämisen haasteena oli saada koko mainoskutsu viimeisessä mahdollisessa tilassa talteen. Mm. mainoksen state (tila) saattoi muuttua vielä käyttäjän selatessa sivua. Esimerkiksi tilan muuttuminen available-tilasta loaded-tilaan on kriittinen hetki, joka vaikuttaa merkittävästi yksittäisen mainoksen suorituskykyyn. Tällainen tieto muutoksesta haluttiin saada mukaan lähetettävään dataan.

Myös verkoston palveluiden erilaiset arkkitehtuurit asettivat tiettyjä haasteita, erityisesti kun verrataan single-page applicationia (SPA) ja perinteisiä monisivuisia verkkosivustoja. SPA-rakenteet tarjoavat dynaamisen käyttäjäkokemuksen, missä sivu ladataan ja päivitetään ilman sivun täydellistä uudelleenlatausta. Tämä lähestymistapa tekee mainoskutsujen seuraamisesta ja datan keräämisestä monimutkaisempaa, sillä mainoksen tilan hallinta ja seuranta tulee suorittaa dynaamisessa ja jatkuvasti muuttuvassa ympäristössä.

Toisaalta perinteisissä monisivuisissa sovelluksissa jokainen sivun lataus tai siirtyminen uudelle sivulle tarjoaa selkeän pisteen mainoskutsujen suorittamiselle ja datan keräämiselle. Tässä ympäristössä mainoskutsun datan tilan seuranta on suoraviivaisempaa, sillä jokainen sivunlataus luo selkeän pisteen, jossa lähetetään uusi mainoskutsu ja edellisestä kerätty data voidaan ottaa talteen.

Tästä syystä oli tärkeää, että mainoskutsujen seuraamiseen ja datan keräämiseen käytettävä ohjelmisto on suunniteltu siten, että se toimii tehokkaasti kaikenlaisilla sivustoilla, olipa kyseessä sitten SPA-sivusto tai perinteinen monisivuinen verkkosivu.

Päädyttiin lopputulokseen, missä kahta erilaista tapahtumaa seuraamalla voitaisiin saada tarvittava data mahdollisimman täydellisenä kaikilta sivuarkkitehtureilta:

1. Käyttäjä sulkee sivun
2. Käyttäjä tekee uuden mainoskutsun

Näkymän vaihtamisen seurantaan päädyttiin käyttämään selaimen eventlistenereitä, jotka mahdollistavat selaimen erilaisten tapahtumien kuuntelun. Eventlistenarit, eli tapahtumankuuntelijat mahdollistavat reagoinnin erilaisiin käyttäjän toimiin, kuten hiiren klikkauksiin, näppäimistön painalluksiin tai sivun latautumiseen. Kun kuuntelija on lisätty, se odottaa määriteltyä tapahtumaa ja suorittaa sille määritellyn callback-funktion, kun tapahtuma ilmenee. Niitä voidaan luoda oman tarpeen mukaan mille tahansa tapahtumalle, ja niille voidaan lähettää tapahtumia `dispatchEvent` metodin avulla.

Joillekin tapahtumille, kuten DOM (Document Object Model) muutoksille selain tarjoaa valmiin tapahtuman lähetyksen, mutta kehittäjän tulee lisätä kuuntelijat manuaalisesti haluttuihin elementteihin, funktioihin tai dokumenttiin valitsemiinsa paikkoihin. (Document Object Model, MDN, n.d)

Esimerkiksi `beforeunload`-tapahtuma aktivoituu ennen kuin käyttäjä poistuu sivulta, mahdollistaen varoitusten näyttämisen esimerkiksi tallentamattomista muutoksista. `unload`-tapahtuma puolestaan ilmoittaa käyttäjän sivulta poistumisesta, soveltuen resurssien vapauttamiseen tai datan lähettämiseen viime hetkellä. `visibilitychange`-tapahtuma seuraa dokumentin näkyvyyden muutoksia, kuten välilehden vaihtoa tai ikkunan minimointia, mahdollistaen resurssien hallinnan ja toimintojen keskeytyksen, kun sivu ei ole käytössä. (MDN, Window – Events, n.d)

Näistä tapahtumista luotettavimmaksi osoittautui visibilitychange –tapahtuma. Toisin kuin beforeunload ja unload -tapahtumat, jotka keskittyvät käyttäjän sivulta poistumiseen, visibilitychange mahdollistaa jatkuvan seurannan siitä, onko käyttäjä aktiivisesti sivulla vai ei. Mahdollisuus kerätä dataa selaimen sulkemisen lisäksi myös välilehden vaihdoksessa sekä ikkunan minimoitaessa koettiin arvokkaaksi, sillä käyttäjä ei välttämättä enää palaa tällaiseen istuntoon.

Myös unload ja beforeunload tapahtumien luotettavuus on osoittautunut heikoksi, erityisesti mobiililaitteilla, joissa selain ei välttämättä aktivoi näitä tapahtumia lainkaan. Esimerkiksi käyttäjä saattaa vaihtaa toiseen sovellukseen sivustolla ollessaan ja sulkea myöhemmin selainsovelluksen puhelimen sovellushallinnan kautta, jolloin mainitut tapahtumat eivät laukea. (MDN, Navigator – sendBeacon, n.d)

Alla kuvatussa koodissa (Kuva 1) lisätään eventListener visibilitychange-tapahtumaan tarkastetaan, onko visibilityState arvoltaan hidden ja kutsutaan tämän perusteella sendPayload -metodia. Metodi hoitaa lopullisen datan muodostamisen ja lähettämisen eteenpäin.

```
private addOnCloseListener = (): void => {
  window.addEventListener("visibilitychange", () => {
    if (document.visibilityState === "hidden") {
      this.sendPayload();
    }
  });
};
```

Kuva 1. Esimerkki TypeScript-koodista, joka lisää visibilitychange-tapahtuman kuuntelijan dokumenttiin.

Tällä tavalla kerätessä dataa törmätään kuitenkin ongelmaan duplikaattidatan kanssa. Kun sovellukset keräävät jatkuvasti dataa käyttäjän toiminnoista, kuten sivun näkyvyyden muutoksista, on riski, että samaa tietoa lähetetään useita kertoja. Tämä ei ainoastaan kuormita turhaan verkkoa ja palvelimia, mutta myös vääristää analytiikan tuloksia, sillä duplikaattidata voi antaa virheellisen kuvan käyttäjän toiminnasta ja sitoutumisesta.

Ratkaisuksi duplikaattidatan ongelmaan tarvittiin järjestelmä, joka tunnistaa ja estää saman tiedon toistuvan lähettämisen ilman, että tukeudutaan pelkästään aikaleimoihin tai UUID-tunnisteisiin. Aikaleimat eivät tarjoa tarpeeksi tarkkuutta, koska ne voivat olla identtisiä tai hyvin lähellä toisiaan erityisesti sivun näkyvyystilan muutosten yhteydessä. UUID-tunnisteiden käyttö yksilöllisenä erottelijana ei myöskään ole ideaali, sillä tulevaisuudessa saattaa olla tarve yhdistää sama UUID useisiin tapahtumiin analytiikan syventämiseksi. Siksi tarvitaan monipuolisempi lähestymistapa, joka sallii datan tehokkaan yhdistämisen ja erottelun, ottaen huomioon sekä nykyiset että tulevaisuuden tarpeet, ilman että se rajoittaa datan uudelleenkäyttöä tai analytiikan laajuutta.

Ongelman ratkaisemiseksi kehitettiin `LimitedSet` -luokka, jolla toteutettiin tapa hallita rajoitettua joukkoa uniikkeja arvoja. Tämä lähestymistapa mahdollistaa tehokkaan datan hallinnan sovelluksissa, joissa on tarpeen rajoittaa muistin käyttöä tai säilyttää vain viimeisimmät tiedot tietyn koon rajoissa. Räätelöimällä perinteistä `Set`-rakennetta, `LimitedSet` tuo mukanaan automaattisen vanhimpien tietojen poistomekanismin, kun maksimikapasiteetti ylittyy, varmistaen näin, että kokoelma pysyy määritellyn koon rajoissa.

```
class LimitedSet<T> extends Set<T> {
    maxSize: number;

    constructor(maxSize: number) {
        super();
        this.maxSize = maxSize;
    }

    add(value: T) {
        if (this.size ≥ this.maxSize) {
            this.delete(this.values().next().value);
        }
        super.add(value);
        return this;
    }
}
```

Kuva 2. `LimitedSet` -luokan toteutus.

Muuttamalla vastaanotettu tapahtuma merkkijonoksi ja tallentamalla se LimitedSet-luokan tietorakenteeseen voidaan varmistaa, että järjestelmä käsittelee jokaisen tapahtuman uniikisti ja estää saman tapahtuman käsittelyn useaan kertaan. Näin ennen lähetystä voidaan tutkia, onko kyseinen merkkijono jo tietorakenteessa. Jos se löytyy, tapahtuma tunnistetaan jo käsitellyksi ja sen lähetys perutaan.

Kun käsitellään vain yhdentyyppisiä tapahtumia, tietorakenteen koko voi olla yksi, sillä riittää että olemme säilöneet viimeisimmän lähetetyn tapahtuman. Kuitenkin jos tulevaisuudessa otetaan vastaan tapahtumia eri tietolähteistä, on tietorakenteen kokoa syytä kasvattaa vastaamaan tietolähteiden lisääntyvää määrää tai kehittää toinen tapa suodattamaan duplikaattitapahtumia.

Uuden mainoskutsun perusteella datan kerääminen oli huomattavasti helpompaa verrattuna käyttäjän toiminnan seurantaan kuin visibilitychange-tapahtuman kautta. Tässä tapauksessa riitti, että vanhat tiedot lähetetään siinä vaiheessa, kun uusi mainoskutsu tehdään.

```
private sendPayload = (): void => {
  const currentRequestIndex = this.requests.length - 1;
  if (currentRequestIndex ≥ 0) {
    const currentRequest = this.requests[currentRequestIndex];
    currentRequest.sendAnalyticsPayload(currentRequestIndex);
  }
};
```

Kuva 3. sendPayload –metodin toteutus.

Kun uusi mainoskutsu lähetetään, sendPayload-funktio aktivoituu ja tarkistaa, onko pyyntöjono taulukko (this.requests) ei-tyhjä. Jos jono sisältää pyyntöjä, viimeisimmän pyynnön tiedot lähetetään analytiikkapalveluun käyttämällä sendAnalyticsPayload-metodia, jolle annetaan viimeisimmän pyynnön indeksi argumenttina.

5.4 Virheiden kerääminen

Virheiden keruun suunnitteleminen aloitettiin määrittelemällä virheistä kerättävät attribuutit. Tarkoituksena oli varmistaa, että järjestelmä pystyy tunnistamaan, tallentamaan ja analysoimaan virhetiedot niiden oleellisilta osilta. Käyttämällä kerättyä dataa voidaan paitsi ymmärtää virheiden syyt ja seuraukset syvällisemmin, myös kehittää ratkaisuja niiden estämiseksi tulevaisuudessa.

5.4.0 Kerättävä data

Virheistä kerättävät attribuutit sisältävät muun muassa virhekoodin, virheviestin, vakavuustason, virheen kontekstin, virheen kutsupinon ja mahdolliset lisätiedot. Näiden attribuuttien avulla pystytään tunnistamaan, luokittelemaan ja tutkimaan erilaisia virheitä, mikä auttaa parantamaan ohjelmiston luotettavuutta ja suorituskykyä. Kerätyn datan avulla voidaan ennakoivasti havaita ja ehkäistä tulevia virhetilanteita, mikä edelleen parantaa ohjelmiston luotettavuutta, toimintaa ja käyttäjäkokemusta.

Alla taulukossa 4 esitellään tiedot, joita kerätään virhetilanteissa. Tarkastelemme virhekoodien, virheviestien, vakavuustasojen, kontekstin ja pinojäljitelmän keräämistä. Tiedot antavat tarvittavat välineet virheiden ymmärtämiseen ja korjaamiseen, sekä auttavat parantamaan ohjelmiston toimivuutta.

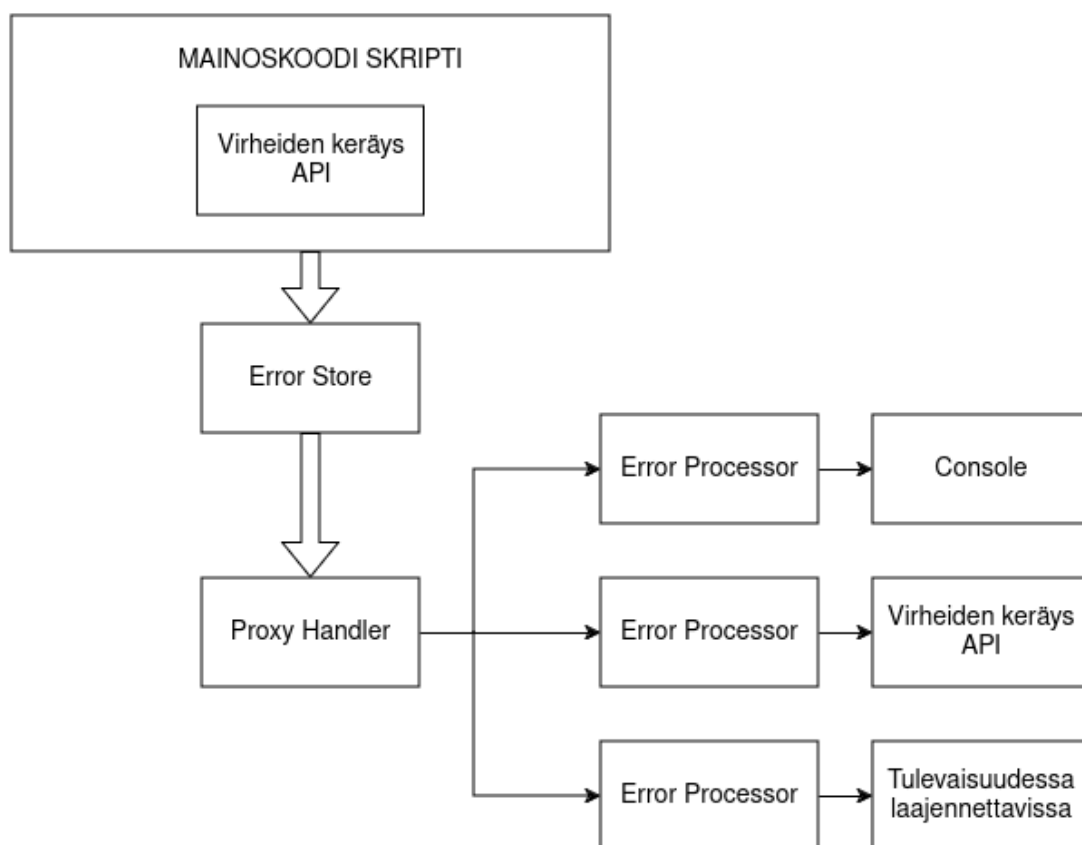
TAULUKKO 4. Virheistä kerättävää dataa

Avain	Kuvaus
code	Virhekoodi, joka identifioi virheen tyyppiä.
message	Virheviesti, joka selittää virheen luonteen tai syyn.
level	Virheen vakavuustaso. Mahdolliset tasot fatal, error tai warning.
scope	Virheen konteksti tai alue, jossa virhe tapahtui.
stack	Virheen kutsupino, joka kertoo reitin, miten koodissa edettiin virhetilanteeseen.
details	Mahdolliset lisätiedot virheestä, jotka eivät sovi muihin kenttiin.

Kerätyn datan avulla voidaan tunnistaa sekä korjata olemassa olevia virheitä ja sen lisäksi ennakoita, että estää potentiaalisia tulevia virhetilanteita, parantaen ohjelmiston luotettavuutta ja toimintaa.

5.4.1 Keräysprosessi

Virheidenkeruujärjestelmän toteutus aloitettiin suunnittelemalla arkkitehtuurikaavio, joka visualisoi järjestelmän pääkomponentit ja niiden väliset vuorovaikutukset. Tarkoituksena oli rakentaa mainoskoodijakeluun virheiden keräys API virhetilanteiden tallentamiseen. Tämä API toimii yhdyspintana, jonka kautta virhetiedot siirretään järjestelmän eri komponenteille analysointia, käsittelyä sekä lähetystä varten. API:n toiminta on kuvattu alla olevassa kuviossa 4.



Kuvio 4. Virheidenkeruujärjestelmän arkkitehtuurikaavio.

Virheidenkeräysprosessin implementoinnissa kohdattiin monenlaisia haasteita. Eräs keskeinen päätös oli, ettei selaimen `window.onerror` tapahtumaa voida käyttää suoraan. Tämä johtui siitä, että mainoskoodi suoritetaan samassa ikkunassa palvelun muun JavaScript koodin kanssa. `Window.onerror` olisi tallentanut kaikki sivustolla ilmenneet virheet. Tämä olisi johtanut tilanteeseen, jossa suurin osa kerätystä virhetiedosta olisi ollut epärelevanttia tai harhaanjohtavaa, sisältäen virheinformaatiota kolmansien osapuolten ja isäntäpalveluiden toiminnasta.

Haasteen ratkaisemiseksi käytiin läpi koko mainonnanjakelu koodin virhekäsittelymekanismit. Läpikäynnin aikana tehtiin katsaus olemassa olevien virhekäsittelyrutiineihin sekä suoritettiin näiden päivittäminen uusien, yllä kuvatun virheenkäsittelymekanismin mukaisesti. Tämän ansiosta voitiin varmistaa, että kerättiin vain sellaista virhetietoa, joka oli halutulla tavalla luokiteltu sekä suoraan yhteydessä mainoskoodin toimintaan. Uusissa virhekäsittelymekanismeissa keskityttiin erityisesti virheiden tunnistamiseen mahdollisimman varhaisessa vaiheessa ja ohjaamaan ne kuviossa 4 esiteltyyn prosessiin, jossa ne voitiin analysoida, käsitellä ja lähettää eteenpäin tehokkaasti. Tämän ansiosta oli mahdollista välttää

ei-toivotun virheinformaation kerääminen ja keskittyä vain niihin virhetilanteisiin, jotka todella vaikuttavat mainoskoodin suorituskykyyn ja toimintavarmuuteen.

Talteenotettujen virheiden käsittelyyn luotiin IErrorReporter-rajapinta (Kuviossa 4. virheidenkeräys API), joka mahdollistaa virheiden raportoinnin määriteltyjen attribuuttien mukaisesti. Rajapinnassa määritellään metodi virheen raportointiin, jonka keräämä data on esitetty aiemmin Taulukossa 3.

ErrorStoren rooli on varmistaa, että kaikki kerätyt virhetiedot tallennetaan luotettavasti, jotta ne voidaan myöhemmin prosessoida. Virheet voidaan Error Storen toteutuksen mukaan tallentaa esimerkiksi tietokantaan, lokitiedostoon tai selaimen muistiin. Tallennustavasta riippuen käyttäjä voi jälkikäteen tarkastella tallennettuja virheitä ja analysoida niitä tarkemmin, mikä auttaa parantamaan ohjelmiston luotettavuutta ja suorituskykyä. Lisäksi ErrorStoren avulla voidaan helposti integroida erilaisia tallennusmekanismeja riippuen käyttäjän tarpeista ja ympäristöstä, jolloin tulevaisuudessa tallennustapaa voidaan helposti vaihtaa.

Proxy Handler toimii välikätenä (proxy) ErrorStoren ja ErrorProcessorien välillä, mahdollistaen virhetietojen välittämisen ilman, että kumpikaan pää komponentista tarvitsee tietää toistensa sisäisestä toteutuksesta. Kun ErrorStore ja ErrorProcessorit ovat eristetty toisistaan, voidaan järjestelmään helposti lisätä tai poistaa ErrorProcessoreita ilman, että se vaikuttaa ErrorStoren toimintaan tai vaatii suuria muutoksia järjestelmän arkkitehtuuriin.

ErrorProcessorin tarkoituksena on olla konfiguroitavissa lopullisen käyttökohteen eli vastaanottavan rajapinnan (API:n) – vaatimusten mukaisesti. Useita eri ErrorProcessoreja voidaan asettaa toimimaan rinnakkain eri rajapintojen ja käyttötapojen mukaan. Yksittäinen ErrorProcessor voidaan asettaa käsittelemään ja luokittelemaan virheitä eri kriteerein. Esimerkiksi, jos virhetiedot lähetetään konsoliin, ne voidaan konfiguroida siirtymään suoraan ilman lisäkäsittelyä. Muissa tapauksissa, kuten kun virheitä tallennetaan tai lähetetään eteenpäin analytiikkatyökaluihin, virheet voidaan luokitella esimerkiksi vakavuuden, esiintymistiheyden ja muiden kerättyjen statistiikkojen perusteella.

```
export type errorStackOpts = {  
  stackFullFlag: boolean;  
  fullStackMsg: string;  
  maxErrorCount: number;  
  reportingThreshold: number;  
  errorLevelThreshold: keyof typeof AtsErrorLevel;  
  omitWarnings: boolean;  
};
```

Kuva 4. ErrorProcessorin konfiguraation tyyppitys.

Luokittelun ja käsittelyn mukauttaminen mahdollistaa, että vain merkitykselliset virheet lähetetään eteenpäin tai herättävät hälytyksiä. Tämä suodatus voi perustua esimerkiksi virheiden määrään (threshold) tai vakavuustasoon (errorLevelThreshold).

ErrorProcessorin suunnittelussa on huomioitu, että järjestelmän tehokkuuden ja resurssien hallinnan kannalta on mahdollista rajoittaa yksittäisestä istunnosta tulevien virheiden määrää. Tämä estää yksittäisen käyttäjän toiminnan aiheuttamasta liiallista kuormitusta virheidenkäsittelyjärjestelmälle ja varmistaa, että järjestelmä säilyy responsiivisena myös silloin, kun virheitä tapahtuu tiheään. Tätä tarkoitusta varten ErrorProcessor sisältää logiikan, joka seuraa virheiden määrää per istunto ja asettaa rajoituksia niiden käsittelylle. stackFullFlag-lippu on indikaattori siitä, että virhepieno on saavuttanut määritellyn koon (maxErrorCount), joka on konfiguroitavissa oleva raja-arvo.

5.5 Tapahtumien lähettäminen

Tapahtumien lähettäminen palvelimelle käyttäjän toimintojen seuraamiseksi on olennainen osa verkkoanalytiikkaa, erityisesti käyttäjän siirtyessä pois sivulta. Modernit web-teknologiat tarjoavat ratkaisuja, jotka mahdollistavat tiedon lähettämisen tehokkaasti taustalla käyttäjäkokemusta häiritsemättä. SendBeacon-metodi on yksi tällaisista teknologioista. Se on suunniteltu lähettämään pieniä määriä tietoja palvelimelle asynkronisesti, kun käyttäjä on poistumassa sivulta. Tämän metodin etuna on, että se mahdollistaa datan lähettämisen ilman, että se vaikuttaa negatiivisesti sivun purkautumiseen tai seuraavan sivun latautumiseen.

Toisaalta moderni Fetch API on toinen esimerkki tehokkaasta työkalusta datan lähettämiseen. Vaikka Fetch API on luotettava ja joustava tapa tehdä http-pyyntöjä, se ei ole aina optimaalinen ratkaisu sivulta poistumisen yhteydessä tapahtuvaan datan lähetykseen. Selaimet voivat keskeyttää asynkroniset pyynnöt sivun latautumisen tai poistumisen aikana, mikä voi johtaa siihen, että pyyntöjä ei suoriteta loppuun. Tästä syystä Fetch API:n käyttö ei välttämättä takaa datan luotettavaa siirtoa sivulta poistumisen yhteydessä samalla tavoin kuin SendBeacon-metodi.

SendBeacon-metodin etuna on sen yksinkertaisuus ja luotettavuus erityisesti web-analytiikan tarpeisiin. Käyttämällä SendBeaconia voitiin varmistaa, että analytiikkadata lähetetään palvelimelle ilman käyttäjäkokemuksen heikkenemistä. Se on suunniteltu toimimaan taustalla ja sen avulla voidaan vähentää tarvetta käyttää vanhentuneita tekniikoita, kuten synkronisia XMLHttpRequest-kutsuja, jotka voivat viivästyttää sivulta poistumista.

Haittapuolena metodin käytössä on se, että tietyn session maksimidatamäärä voi täytyä. Rajoitus voi aiheuttaa haasteita, jos kerättävä analytiikkadata on laajaa tai jos käyttäjä suorittaa pitkiä sessioita ilman sivun uudelleenlatauksia. Priorisoimalla tärkeintä dataa on mahdollista vähentää rajoitusten aiheuttamia haasteita.

5.6 Käyttäjän yksityisyys ja suostumus

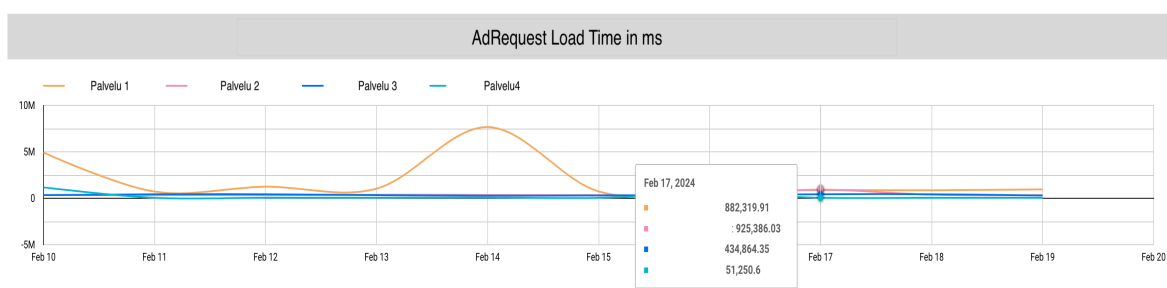
Projektin toteutuksessa käyttäjän yksityisyyden suojaaminen ja suostumuksen varmistaminen olivat keskeisiä periaatteita. Vaikka nämä mekanismit olivat jo valmiiksi toteutettuja, oli tärkeää osata hyödyntää olemassa olevia Consent Management Platform (CMP) -tarkistuksia asianmukaisesti. Näiden tarkistusten avulla varmistettiin, että käyttäjältä saatiin tarvittavat suostumukset ennen henkilökohtaisten tietojen keräämistä tai käsittelyä. CMP-tarkistukset mahdollistivat sen, että käyttäjien suostumukset eri tarkoituksiin voitiin tarkistaa luotettavasti, mikä puolestaan takasi, että tietojen kerääminen tapahtui aina käyttäjän selkeällä luvalla.

6. Kerätyn datan tulokset

Tässä kappaleessa esitellään tuloksia, jotka syntyivät palveluista kerätystä datasta. Vaikka datan visualisointi ja analyysi eivät olleet työn tekijän itsensä kehittämiä tai tekemiä, on niiden tuottamien tulosten esittely ja tulkinta projektissa tärkeää. Visualisointi auttoi esittämään kerätyn datan helposti ymmärrettävässä muodossa, kuten kaavioissa ja graafeissa, jotka paljastivat tärkeitä trendejä ja mahdollisia kehityskohteita.

Lisäksi on olennaista huomioida, että esitetty datamateriaali on koottu siten, että siitä on suodatettu pois kaikki mahdolliset liikesalaisuudet paljastavat tiedot, kuten tiettyihin palveluihin liittyvät nimet tai muut erityispiirteet. Tämä on tehty tarkoituksellisesti turvaamaan arkaluonteinen liiketoimintatieto, joka voisi olla arvokasta kilpailijoille tai muulle taholle. Tästä syystä esitellyt graafit ja analyysit on valittu siten, että ne auttavat lukijaa havainnoimaan, kuinka järjestelmä kykenee tuottamaan tietoa, joka auttaa päätöksenteossa, ohjaa kehitystyötä ja tukee ylläpitoa menettämättä yrityksen strategista etua.

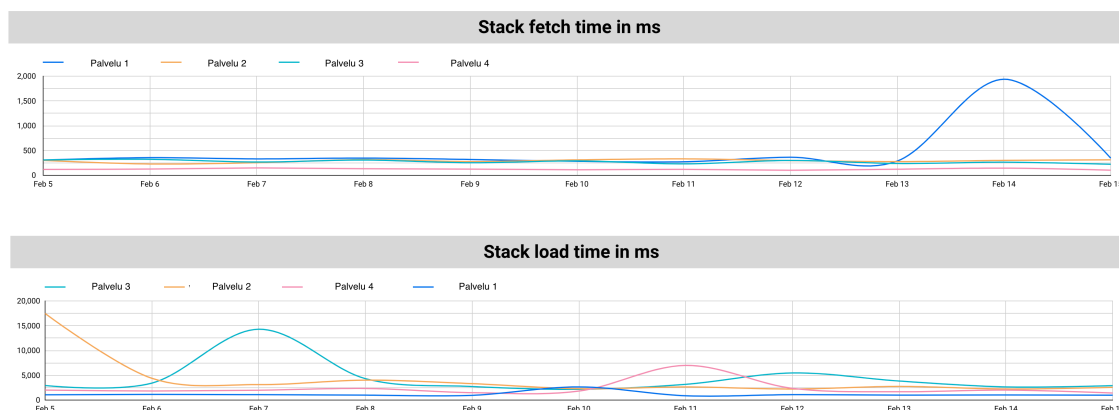
Alla kuviossa 5 havainnollistetaan neljän eri palvelun mainospyyntöjen latausajankoja millisekunnissa ajanjaksolla 10.–20. helmikuuta. Aikasarja-analyysin avulla seurataan kunkin palvelun suorituskykyä latausajkojen näkökulmasta. Palvelu 3 erottuu tasaisen nopeilla latausajoillaan, kun taas Palvelu 4 osoittaa pääsääntöisesti alhaisimpia latausajkoja, paitsi 17. helmikuuta tapahtuvan merkittävän piikin aikana. Tämä piikki on lyhytaikainen, ja palvelun latausaika palautuu nopeasti alemmalle tasolle. Palvelut 1 ja 2 näyttävät puolestaan saavuttavan samankaltaiset latausajat koko tarkastelujakson ajan.



Kuvio 5. Mainospyyntöjen kesto eri palveluissa.

Kuviossa 6. esitellään kahdessa kaaviossa neljän eri palvelun suorituskykyä hakea ja ladata tietoa. Ensimmäisessä kaaviossa esitellään mainoskoodin hakemiseen kulunutta aikaa. Kaikki palvelut pysyvät suhteellisen tasaisina ja alhaisina latausaikoina lukuun ottamatta Palvelu 1:stä, joka kokee merkittävän piikin latausajoissa 13. helmikuuta, nousten yli 2000 millisekuntiin.

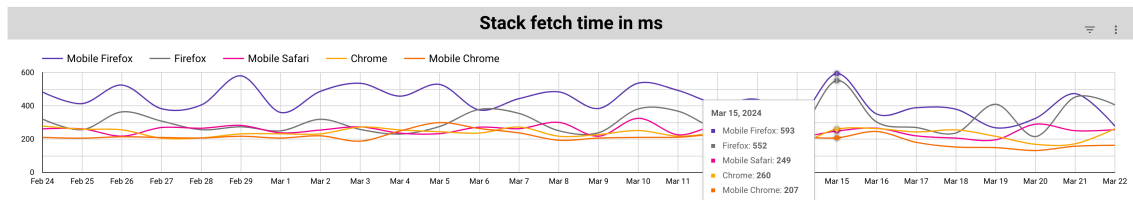
Toisessa kaaviossa, jossa esitellään mainoskoodin käynnistämiseen kuluva aikaa. Erot palveluiden välillä ovat selvemmin nähtävissä. Palvelu 3 aloittaa korkeilla latausajoilla, yli 15 000 millisekuntia 5. helmikuuta, mutta näyttää tasaantuvan ja laskevan nopeasti seuraavina päivinä. Palvelu 2 näyttää nousevan hitaasti ja saavuttavan huippunsa noin 10. helmikuuta, kun taas Palvelu 4:n latausajat vaihtelevat suhteellisen voimakkaasti koko tarkastelujakson ajan. Palvelu 1 pysyy suhteellisen matalana koko ajanjakson, mikä viittaa siihen, että sen suorituskyky on vakaa.



Kuvio 6. Palveluiden mainoskoodien haku- ja latausaikojen kuvaajat.

Näiden kaavioiden analyysi antaa ymmärtää, että eri palveluissa on vaihtelevia suorituskyvyn trendejä, jotka voivat johtua monista tekijöistä, kuten liikennemäärästä, palvelimen vastausajoista tai mahdollisista optimointiin liittyvistä toimenpiteistä. Palvelu 1:n äkillinen piikki erityisesti herättää kysymyksiä, jotka vaativat lisäselvitystä mahdollisten teknisten ongelmien tai yllättävien liikennepiikkien syistä.

Kuviossa 7 esitetään viiden eri selaimen – Mobile Firefox, Firefox, Mobile Safari, Chrome ja Mobile Chrome – mainoskoodin hakupyynnöiden latausaikoja millisekunneissa aikavälillä 24. helmikuuta – 22. maaliskuuta. Jokaisen selaimen latausajat on esitetty omalla värikoodatulla käyrällä.



Kuvio 7. Eri selainten mainoskoodin hakupyynnöiden latausaikoja.

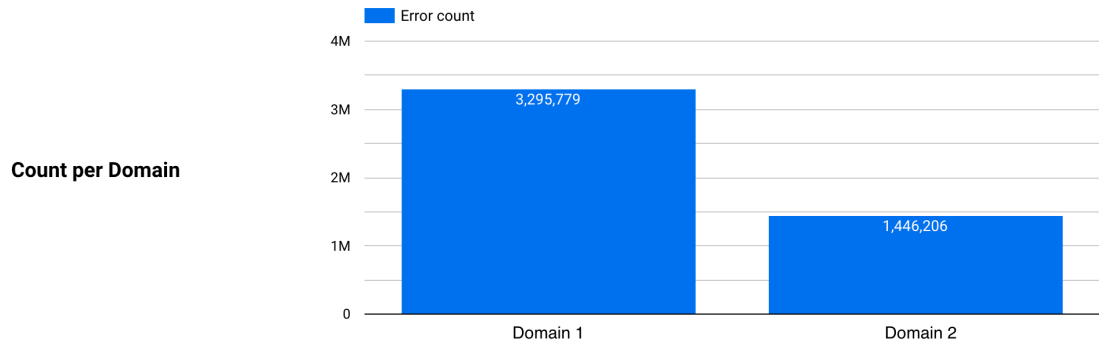
Käyrät näyttävät selaimen suorituskyvyn vaihteluita ajanjaksolla, ja huippukohtat osoittavat hetkiä, jolloin latausaika on ollut pisimmillään. Esimerkiksi 15. maaliskuuta Mobile Firefoxin latausaika on huipussaan noin 593 millisekunnissa, kun taas samana päivänä Mobile Safari ja Mobile Chrome ovat paljon nopeampia, latausaikojen ollessa vastaavasti noin 249 ja 207 millisekuntia.

Kaaviosta voi havaita, että Mobile Firefox ja Firefox näyttävät olevan hitaimpia lähes koko ajanjakson ajan, ja niiden suorituskyvyssä on havaittavissa selkeitä huippuja ja laaksoja. Mobiiliselaimista Mobile Chrome ja Mobile Safari näyttävät suoriutuvan tasaisemmin ja nopeammin kuin Firefoxin versiot.

Eri selaimilla on erilaiset suorituskyvyt ja että ne voivat vaihdella ajasta ja ehkä myös olosuhteista riippuen. Tämän kaltaista aikasarja-analyysia voidaan käyttää selainten suorituskyvyn vertailuun ja mahdollisten optimointikohteiden etsimiseen.

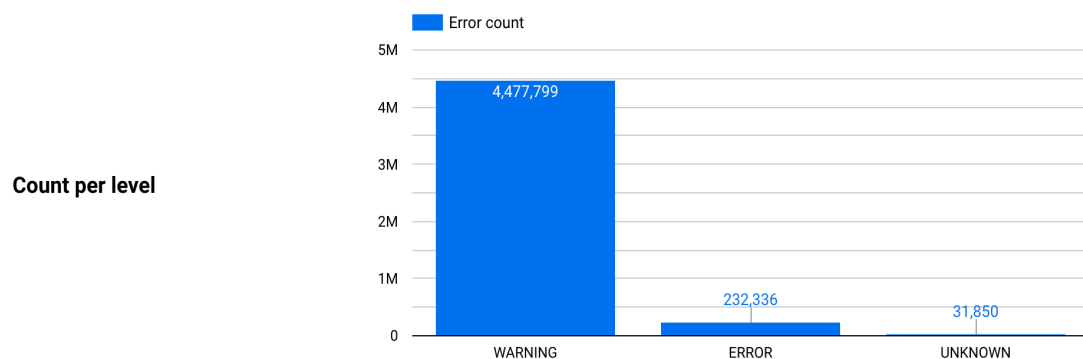
Kuviossa 8 esitetään kahden eri verkkotunnuksen mainoskoodissa kohdattujen virheiden määrät tietyllä aikajaksolla. Pylväsdiagrammi kuvaa virheiden määrää kummassakin verkkotunnuksessa. Palvelu 1 osoittaa huomattavasti suurempaa virhemäärää, yhteensä noin 3,3 miljoonaa virhettä, kun taas palvelu 2 raportoi merkittävästi vähemmän, noin 1,4 miljoonaa virhettä. Tämä ero virhemäärissä voi indikoida erilaisia teknisiä haasteita tai eriasteisia ongelmia verkkotunnusten ylläpidossa ja toimivuudessa. Kuvion tarkoitus on tarjota selkeä ja ytimekäs yleis-

katsaus näiden kahden verkkotunnuksen suorituskykyyn liittyen virheiden esiintymiseen. Tällaisessa vertailussa täytyy kuitenkin ottaa huomioon palveluiden mahdollinen eroavaisuus liikennemäärissä.



Kuvio 8. Virheiden määrä palvelua kohden.

Kuviossa 9 vertaillaan kolmea eri virhetasoa ja niiden esiintyvyyksiä: varoituksia, virheitä ja tuntemattomia virhetasoja. Kuvio osoittaa, että varoitusten määrä on ylivoimaisesti suurin, mikä viittaa siihen, että järjestelmässä on paljon tapahtumia, jotka eivät välttämättä aiheuta suoria toimintavirheitä, mutta jotka vaativat huomiota. Virheitä on huomattavasti vähemmän, joka osoittaa, että vakavampia ongelmia esiintyy vähemmän, mutta ne ovat silti merkittäviä. Tuntemattomien virheiden määrä on pienin, vain noin 32 000, mikä saattaa viitata siihen, että järjestelmän virheseuranta on pääosin kattava, mutta tietyt virheet jäävät luokittelematta.



Kuvio 9. Virheiden määrä luokkaa kohden.

7. POHDINTA

Opinnäytetyön päätarkoituksena oli kehittää järjestelmä, joka kerää ja analysoi dataa mainoskoodin suorituskyvystä erilaisilla käyttöjärjestelmällä ja selainyhdistelmillä. Projekti pyrki syventämään ymmärrystä mainonnanjakelukoodin käyttäytymisestä ja tarjoamaan dataa sen analysointiin. Tavoitteiden saavuttamista arvioitaessa voidaan todeta, että projektin myötä saatiin arvokasta tietoa mainonnan jakeluun liittyvistä latausajoista ja käyttäjäkokemuksen parantamiseksi, mikä on merkittävä edistysaskel verkkopalveluiden kehittämisessä. Puuttamalla löydettyihin ongelmiin voidaan parantaa käyttäjien tyytyväisyyttä, vähentää sivustolta poistumisen todennäköisyyttä ja lisätä sivuston käyttöaktiivisuutta ja potentiaalista tulovirtaa. Kehitetty järjestelmä tuo lisäarvoa yrityksen toimintaan, auttaen tunnistamaan ja poistamaan suorituskyvyn esteitä. Projektin tulokset korostavatkin datan keskeistä roolia liiketoiminnan kehityksessä ja päätöksenteossa.

Projektin aikana kohdatut haasteet, kuten suurten datamäärien hallinta ja analyysien luotettavuuden varmistaminen, toivat esille kehitetyn järjestelmän vahvuudet, kuten kyvyn reaaliaikaiseen seurantaan, mutta myös sen rajoitukset, kuten alustariippuvuus. Tämä nostaa esille tarpeen jatkokehitykselle ja järjestelmän monipuolistamiselle. Datankeräyksen toteutus verkostoon, jossa on kymmeniä eri kokoisia sekä eri tavoilla että arkkitehtuureilla toteutettuja palveluita toi oman haasteensa toimivien ratkaisuiden löytämiseen ja implementointiin. Erilaiset sivut ja niiden erityispiirteet vaativat joustavia ja muokattavia integraatiostrategioita, jotka mahdollistivat tiedonkeruun ja analyysin eri järjestelmien ja palveluiden välillä. Tämän saavuttamiseksi oli välttämätöntä kehittää dynaamisia tapoja konfiguroida keräystä, jolla mahdollistettiin sen mukautuminen erilaisten palveluiden käyttäjämääriin tai muihin erityispiirteisiin.

Jatkokehityskohteena tunnistettiin, että mainoskutsuista kerättyä dataa voidaan hyödyntää aiempaa monipuolisemmin keräämällä tietoa laajemmin mainoskoodin eri osa-alueilta. Tämä mahdollistaisi tarkemman ja kattavamman kuvan mainoskoodin toiminnasta. Esimerkiksi voitaisiin mitata aikaa, joka kuluu palvelussa sivulta toiselle siirtymisen ja ensimmäisen mainoksen näyttämisen välillä. Tämä antaisi arvokasta tietoa mainosten latautumisen nopeudesta ja siten vaikuttavuudesta käyttäjäkokemukseen.

LÄHTEET

Delen, Dursun. 2020. Predictive Analytics: Data Mining, Machine Learning and Data Science for Practitioners, 2nd Edition. Pearson FT Press.

Choi, B., Wallace, L., & Wang, Y. 2018. Big Data Analytics in Operations Management.

Miller, J.D. 2017. Big Data Visualization. Packt Publishing.

Holz, N. What is CRISP-DM? Viitattu 3.10.2023. <https://www.datascience-pm.com/crisp-dm-2/>

"Henkilötietojen käsittelyä koskevat periaatteet". Viitattu 30.3.2024. <https://fakta.digiturvamalli.fi/gdpr-asetus/5-henkilotietojen-kasittelya-koskevat-periaatteet>

Tietosuojalaki 30.3.2018/1050. Viitattu 30.3.2024. <https://www.finlex.fi/fi/laki/ajantasa/2018/20181050>

Beasley, M. 2013. Practical Web Analytics for User Experience: How Analytics Can Help You Understand Your Users. Elsevier Science & Technology.

Chaffey, D. & Patron, M. 2012. "From web analytics to digital marketing optimization: Increasing the commercial value of digital analytics." Journal of Direct, Data and Digital Marketing Practice. Volume 14, Issue 1, Macmillan Publishers.

Croll, A. & Yoskovitz, B. 2024. Lean Analytics. O'Reilly.

Google Ads Help. Viitattu 3.4.2024. <https://support.google.com/google-ads/answer/6365?hl=en>

"What is data visualization?". IBM. Viitattu 20.2.2024. <https://www.ibm.com/topics/data-visualization>

Alaimo, C. & Kallinikos, J. 2018. "Objects, Metrics and Practices: An Inquiry into the Programmatic Advertising Ecosystem." Viitattu 12.3.2024. https://link.springer.com/chapter/10.1007/978-3-030-04091-8_9

Jormalainen, P., Liikkanen, A., Morri, P., Ojala, S., Runila, S., Stenberg, J. "Digimainonnan ABC". Alma Media Oy. Viitattu 20.3.2024. https://www.etuovi.com/mediatieto/wp-content/uploads/Digimainonnan_ABC_.pdf

"Internet mainonnan sanasto". IAB Finland ry, 20.1.2009. Viitattu 12.3.2024. <https://www.iab.fi/media/pdf-tiedostot/standardit-ja-oppaat/verkkomainonnan-sanasto.pdf>

Konasani, V.R. & Kadre, S. 2015. Practical Business Analytics Using SAS. Apress.

MDN Web Docs. 2024. Document Object Model (DOM). Mozilla. Verkkosivu. Viitattu 19.2.2024. https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model

MDN Web Docs. 2024. Window. Mozilla. Verkkosivu. Viitattu 19.2.2024.
<https://developer.mozilla.org/en-US/docs/Web/API/Window/>

MDN Web Docs. 2024. Navigator.sendBeacon(). Mozilla. Verkkosivu. Viitattu 19.2.2024. [https://developer.mozilla.org/en-US/docs/Web/API/Navigator/send-Beacon](https://developer.mozilla.org/en-US/docs/Web/API/Navigator/sendBeacon)

LIITTEET

Liite 1. Datanäyte, aggregoitua mainoskutsuista kerättyä dataa.

date	domain	adsection	browser_name	reqstat	stack	avg_cpu_load_time	avg_ad_request_load_time	avg_keyword_load_time	avg_stack_fetch_time	avg_stack_load_time	avg_nav_start_dom_content_loaded_diff_ms	avg_nav_start_to_load_start	avg_nav_start_to_load_end	avg_adslot
2024-03-01	palvelu1	searchresultpage_mobile	Samsung Browser	cold	<semver.build.date_time>	6.64	2368.01	10270.53	286.23	2628.92	2359.4	6004.08	8643.09	7.0
2024-03-01	palvelu1	item-page_mobile_interstitial	Samsung Browser	hot	<semver.build.date_time>	5.74	812.76	1197.95	229.02	2021.41	2167.59	5085.34	7092.32	2.0
2024-03-01	palvelu1	front-page_mobile_interstitial	Samsung Browser	cold	<semver.build.date_time>	6.1	1718.59	1189.29	244.95	1988.06	1962.67	4435.78	6428.87	5.0
2024-03-01	palvelu2	front-page_desktop	Chrome	cold	<semver.build.date_time>	4.13	1251.59	1387.62	148.52	1571.03	2191.31	2570.16	4145.69	8.0
2024-03-01	palvelu1	searchresultpage_mobile	Chrome	cold	<semver.build.date_time>	3.55	959.31	601.05	247.9	873.05	1561.23	2722.75	3511.9	7.0
2024-03-01	palvelu1	searchresultpage_desktop	Mobile Safari	hot	<semver.build.date_time>	8.77	1668.19	522.44	554.62	1681.9	1613.76	4402.56	6049.04	11.0
2024-03-01	palvelu1	item-page_desktop_interstitial	Facebook	cold	<semver.build.date_time>	5.46	624.56	2818.73	279.04	3280.73	1892.19	3636.87	6917.6	2.0
2024-03-01	palvelu2	list-page_desktop	Opera	cold	<semver.build.date_time>	2.53	1535.54	492.66	30.41	638.17	2274.06	2449.08	3087.25	11.0
2024-03-01	palvelu1	front-page_mobile_interstitial	Chrome WebView	hot	<semver.build.date_time>	5.43	1000.59	786.96	397.24	1768.6	2147.7	4152.74	5915.4	5.0
2024-03-01	palvelu2	front-page_mobile	Samsung Browser	cold	<semver.build.date_time>	5.91	1472.73	1216.84	322.43	1622.41	2386.8	3297.95	4958.86	4.0

Liite 2. Datanäyte, aggregoitua virheistä kerättyä dataa.

device type	browser name	browser version	os name	domain	code	level	scope	count	date
web	Mobile Safari	16.0	iOS	palvelu3	ECONNABORTED	ERROR	Unknown	2	2024-03-01
web	Mobile Safari	16.5	iOS	palvelu2	ECONNABORTED	ERROR	Unknown	1	2024-03-01
web	Mobile Chrome	122.0.6261.62	iOS	palvelu3	ECONNABORTED	ERROR	Unknown	5	2024-03-01
web	Facebook	452.0.0.39.110	iOS	palvelu2	ATS TIMEOUT	WARNING	Unknown	2	2024-03-01
web	Mobile Chrome	122.0.6261.62	iOS	palvelu2	ATS TIMEOUT	WARNING	Unknown	2	2024-03-01
web	Mobile Chrome	122.0.6261.62	iOS	palvelu2	ECONNABORTED	UNKNOWN	Unknown	2	2024-03-01
web	Mobile Safari	16.0	iOS	palvelu3	ECONNABORTED	ERROR	Unknown	2	2024-03-01
web	Mobile Safari	16.5	iOS	palvelu2	ECONNABORTED	ERROR	Unknown	1	2024-03-01
web	Mobile Chrome	122.0.6261.62	iOS	palvelu3	ECONNABORTED	ERROR	Unknown	5	2024-03-0