

Opinnäytetyö (AMK)

Tieto- ja viestintäteknikka

2021

Harald Lindroos

MUSIIKIN SUORATOISTOPALVELUN LUONTI

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tieto- ja viestintäteknikka

2021 | 38 sivua

Harald Lindroos

MUSIIKIN SUORATOISTOPALVELUN LUONTI

Opinnäytetyö käsittelee yleisellä tasolla musiikkipainotteisen suoratoistopalvelun kehittämisen edellyttämiä eri vaiheita.

Alussa aihetta aletaan käydä läpi avaamalla suoratoiston historiaa, josta päädytään perusaspekteihin kuten tekijänoikeuksien hankintaan musiikkikappaleita varten. Tämän jälkeen opinnäytetyö antaa selityksen äänikoodekeista ja suoratoistotekniikasta, jotka alustavat tulevaa teknistä toteutusta varten.

Loppupuolella näytetään esimerkki siitä, miten palvelu luodaan lähiverkon palvelimelle ja lopputuloksena on verkkosivu, jossa pystyy kuuntelemaan musiikkia. Viimeisinä asioina on tietoturvan ja hyödyllisten verkkosivujen kehittämistyökalujen käyminen lyhyesti läpi.

ASIASANAT:

suoratoisto, musiikki, tiedonsiirto, enkoodaus, tekijänoikeudet

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information and Communications Technology

2021 | 38 pages

Harald Lindroos

CREATING A MUSIC STREAMING SERVICE

The thesis goes through the various stages required to develop a music-oriented streaming service at a general level.

In the beginning, the thesis starts going through the topic by opening up the history of streaming, which ends up with basic aspects such as the acquisition of streaming rights for the copyrighted music tracks. The work then provides an explanation of audio codecs and streaming technology that initializes for the coming technical implementation.

Towards the end, an example will be shown of how to create a service on the server in a local area network (LAN) and the finish result is a webpage where music can be listened to. The last subjects go through security and useful web page development tools briefly.

KEYWORDS:

streaming, music, data transmission, encoding, copyright

SISÄLLYS

SISÄLLYS	4
KÄYTETYT LYHENTEET JA SANASTO	6
1 JOHDANTO	7
2 SUORATOISTON HISTORIA	8
2.1 Lähtölaukaus musiikin suoratoistolle	9
3 PALVELUN PERUSTAMINEN	12
3.1 Tekijänoikeudet	12
3.2 Musiikin hankinta	13
4 ÄÄNIKOODEKIT	15
5 SUORATOISTOTEKNIikka	17
5.1 Sisällönjakeluverkot	18
6 PALVELUN TEKNINEN TOTEUTUS	19
7 TIETOTURVA	29
8 VERKKOSIVUN TYÖKALUT	31
9 LOPUKSI	33
LÄHTEET	34

KUVAT

KUVA 1. KÄYTTÖLUPA MAKSULLISELLE SUORATOISTOPALVELULLE (TEOSTO 2021).	14
KUVA 2. KÄYTTÖLUPA MAINOSRAHOITTEISELLE SUORATOISTOPALVELULLE (TEOSTO 2021).	14
KUVA 3. YLEISIMMÄT ÄÄNIFORMAATIT SELITETTYNÄ (SCARROTT 2021, MUOKATTU).	16
KUVA 4. MPEG-DASH:N TOIMINTARAKENNE (ENCODING.COM 2021).	17
KUVA 5. SISÄLLÖNJAKELUVERKON TOIMINTA (GTMETRIX 2017).	18
KUVA 6. MÄÄRITELTY VIRTUALHOST-TIEDOSTO APACHE-PALVELIMELLA.	20
KUVA 7. PORTIN MÄÄRITTELY TIEDOSTOSSA PORTS.CONF.	21
KUVA 8. MEDIASOITTIMEN KOPIOINTI PALVELIMELLE.	22
KUVA 9. HTML-SIVU (OSA 1).	23
KUVA 10. HTML-SIVU (OSA 2).	24
KUVA 11. SIVUN NÄKYMÄ ENNEN MEDIASISÄLTÖÄ.	24
KUVA 12. SKRIPTIN RAKENNE.	25
KUVA 13. SKRIPTI AJOVAIHEESSA (OSA 1).	26
KUVA 14. SKRIPTI AJOVAIHEESSA (OSA 2).	27
KUVA 15. VALMIIN JA TOIMIVAN VERKKOSIVUN NÄKYMÄ.	28

KUVIOT

KUVIO 1. SUORATOISTOPALVELUIDEN TILAAJAT AJALTA 2019–2020 (MUSIC ALLY 2020, MUOKATTU). 10	
KUVIO 2. MUSIIKISTA SAADUT TULOT FYYSISISSÄ JA DIGITAALISSA MUODOISSA 40 VUODEN AJALTA VUOSILTA 1977–2017 (ROUTLEY 2018).	11
KUVIO 3. YLEISIMMÄT KÄYTETYT HAITTAOHJELMAT KYBERHYÖKKÄYKSISSÄ (STATISTA 2020, MUOKATTU).	29
KUVIO 4. VERKKOPALVELIMIEN ERI OHJELMIEN KÄYTTÖOSUUKSIA (NETCRAFT 2021).	31

KÄYTETYT LYHENTEET JA SANASTO

Apache	Palvelimessa toimiva ohjelma, joka hallitsee sinne tulevaa ja lähtevää HTTP-liikennettä.
FFmpeg	Ilmainen ja monipuolinen videon ja äänen prosessointityökalu, jolla voidaan esimerkiksi transkoodata mediasisältöä.
Metadata	Äänitiedostoissa mukana tuleva tieto artistista, albumista, kappaleesta, tyylilajista, kansikuvasta, ja kappaleen numerosta. (Cambridge Audio 2016)
MPEG-DASH	Moving Picture Experts Group-Dynamic Adaptive Streaming over HTTP: Monilta osin samankaltainen kuin HSL, mutta MPEG-DASHissa on mahdollista käyttää mitä tahansa enkoodaus standardia ja lukemattomissa laitteissa, sillä se on kansainvälinen standardi. Käyttää TCP-siirtoprotokollaa. (Cloudflare 2021)
TCP	Transmission Control Protocol: Käytetään yhdessä IP:n kanssa yhteyden ylläpitoon lähettäjän ja kohteen välillä sekä pakettien järjestyksen turvaamiseen. (Cloudflare 2021)

1 JOHDANTO

Tätä opinnäytetyötä lähdettiin tekemään, koska suoratoisto elää kulta-aikaansa ja valtaosa musiikin kulutuksesta tapahtuu Internetin välityksen kautta muun muassa striimaten. Työssä halutaan tuoda esille asioita, joiden avulla musiikin suoratoistopalvelua on luontevaa lähteä rakentamaan. Siinä käytetään verkkosivujen artikkeleita ja tutkimuksia tiedonpohjana. Kokonaisuus on jaettu eri aihealueisiin, joista edetään seuraavaan loogisessa järjestyksessä. Aiheista esitetään analysoitua ja koostettua tietoa sekä havainnollistavia kuvia ja kuvioita.

Kaikki mediatieto, jota kuljetetaan Internetin välityksellä on paketoitu. Tieto on järkevää pilkkoa pieniksi osiksi, koska tällöin ne voidaan toimittaa monia eri reittejä pitkin määränpäähän, sovittaa erilaisiin siirtonopeuksiin ja se mahdollistaa kesken jääneiden tai hukattujen pakettien uusintalähetyksen. Katseltaessa tai kuunneltaessa suoratoistoa mediasoitin tulkitsee pakettivirran videoksi ja ääneksi ja esittää näin sen katseltavassa ja kuunneltavassa muodossa katsojalle. (Cloudflare 2021)

Kun puhutaan suoratoistosta (engl. streaming), sillä tarkoitetaan, että tietoa ladataan pakettien muodossa reaaliaikaisesti palvelimelta päätelaitteelle esimerkiksi tietokoneelle Internetin välityksellä (Kuvio 1). Tämä mahdollistaa suoratoiston, jossa Internetin käyttäjä voi suoraan aloittaa toistamaan haluamaansa musiikkia nettisivulta, esimerkiksi YouTubesta tai Spotifysta. Musiikkikappaleen ei tarvitse olla kokonaan latautunut voidakseen olla toistettava. Päätelaitteeseen ei paikallisesti tallenneta muuta kuin tilapäinen puskuriosuus (engl. buffer), joka korvautuu uudella, kun musiikin kuuntelua jatketaan. Puskuroinnilla mediasoitin lataa muutaman sekunnin kestoisen osuuden mediaa etukäteen. Näin vältetään satunnaiset pysähdykset, jos latausnopeus vaihtelee tai Internet-yhteys on heikko. (Cloudflare 2021)

2 SUORATOISTON HISTORIA

Suoratoistoa aloitettiin kehittämään 90-luvulla. Alkuvuosina haasteellista oli hitaat Internet-yhteydet, sillä käytössä oli modeemi, jolla saatiin vain 56 Kbps:n latausnopeus. Progressive Networks -yritystä pidetään ensimmäisenä, joka aloitti suoratoiston kehityskulun. Myöhemmin yritys muutti nimensä RealNetworksiksi. RealPlayer - mediasoitin voi olla monille tuttu vuosituhannen alusta. Toinen suuri ja ainoa kilpailija siihen aikaan oli Microsoft, joka myöhemmin sai valtasijan suoratoistokilpailussa sen Windows Media teknologioiden saavutuksista. (Zambelli 2013)

Syyskuun viidentenä päivänä vuonna 1995 striimattua suoraa radiolähetystä pidetään yhtenä merkittävimpänä ensimmäisenä suoratoistotapahtumana Internetissä, missä kaksi amerikkalaista pesäpallojoukkuetta Mariners ja Yankees ottelivat toisiaan vastaan. On ollut myös debattia, että The Rolling Stones -yhtyeen konsertti marraskuussa 1994 olisi ensimmäinen verkkostriimi, joka esitettiin Mbonen välityksellä. (Rayburn 2016)

2000-luvulle tultaessa vuoden 2005 aikoihin uusi kilpailija nimeltään Macromedia alkoi saada hiljalleen markkinaosuutta suosiota saaneen ja kasvavan Flash Playerin takia. Se samalla söi Windows Median markkinaosuutta. Flash Player onnistui luomaan toimivan vuorovaikuteisuuden Web 2.0 tekniikalla ja median suoratoiston välillä. Koitti uusi aikakausi suoratoistossa, mutta edelleen jäi vanhoja ongelmia selvittämättä kuten kaistanleveys, skaalautuvuus ja kattavuus. Siihen aikaan valtava osa Internet-liikenteestä perustui HTTP-alustalle. (Zambelli 2013)

Sisällöntarjoajaverkkoja (engl. Content Delivery Network) käytettiin kasvavissa määrin tarjoamaan sisältöjä suurille yleisöille. Silloin käytettiin kankeaa protokollaa nimeltään UDP:tä, jonka käyttö kuormittui kysynnän kasvaessa, ja nojautui puhtaasti Internet-yhteyden nopeuteen. Merkittävä muutos tapahtui jälleen, kun yritys Move Networks oli kehittänyt HTTP:hen perustuvan adaptoituvan suoratoiston vuonna 2007. (Zambelli 2013) Se mahdollisti sen, että videoita ei tarvinnut puskuroida eli ladata etukäteen. Toisto voitiin aloittaa heti, koska mediasoitin mukautui verkon kaistanleveyteen valikoimalla sopivan bittinopeuden omaavan mediasegmentin eri laatuluokista suhteutettuna sen hetkiseen kaistanleveyteen. Kun verkossa kaistanleveys parani, voitiin siirtyä seuraavaan laatuluokkaan. Näitä laatuluokkia vuoroteltiin aina sen hetkisen verkon kaistanleveyden mukaan.

Tämän jälkeen Microsoft julkisti oman samaan tekniikkaan perustuvan ratkaisun vuonna 2008. Sen jälkeen samaisena vuonna Netflix seurasi perässä, ja vuotta myöhemmin Applella tuli oma versio, HLS (engl. HTTP Live Streaming). Adobe liittyi mukaan 2010. (Zambelli 2013)

Adaptoituvalla suoratoistolla saatiin aikaan, että monet suurenluokan tapahtumat voitiin esittää kuten Lontoon ja Vancouverin Olympialaiset, Wimbledon ja monia muita. (Zambelli 2013)

Maksulliselta puolelta löytyi ainakin Netflix ja Amazon omalla videosuoratoisto palvelullaan. Se johti siihen, että jokaisella yrityksellä oli oma suoratoistotekniikka, joka toimisi vain heidän toistopalveluissaan. Ryhdyttiin standardisoimaan kyseistä tekniikkaa 3GGP:n toimesta. Prosessiin osallistui yli 50 yritystä, joissa mukana suuria yhtiöitä, kuten Microsoft, Apple ja Netflix. Huhtikuussa 2012 uusi standardi oli julkistettu, joka sai nimeksi DASH (engl. Dynamic Adaptive Streaming Over HTTP). Yleisesti sitä kutsutaan nimellä MPEG-DASH. (Zambelli 2013)

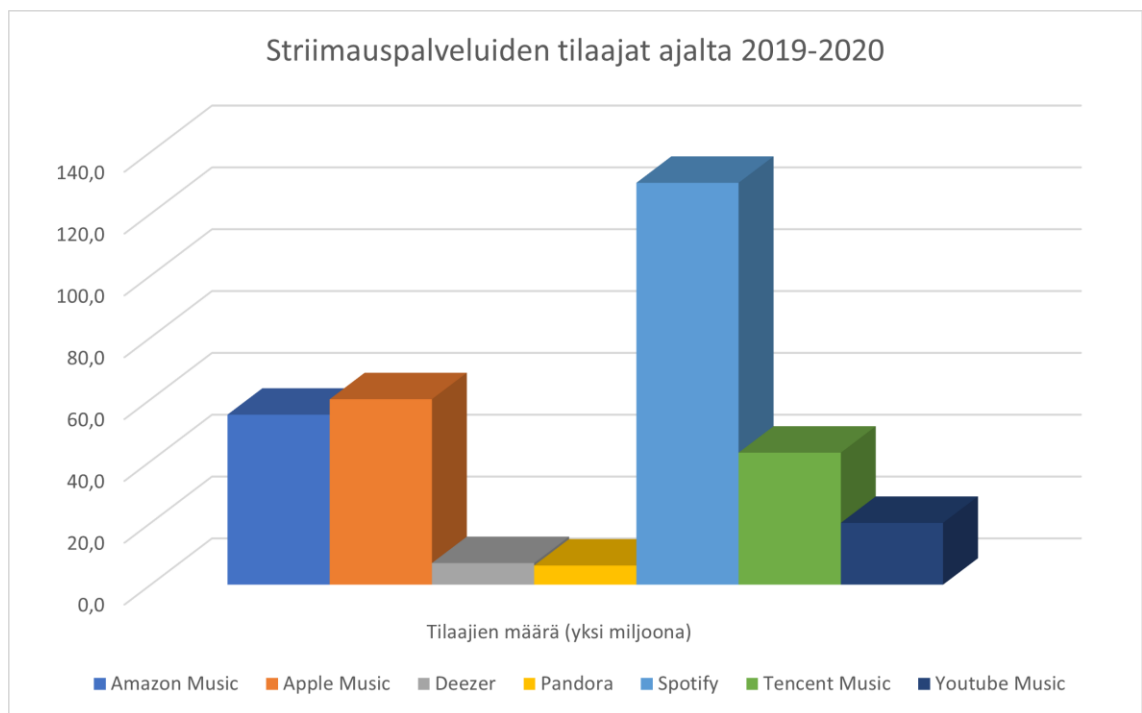
Aluksi sen käyttö oli epäselvää, joten osa yrityksistä, jotka olivat standardisoinnissa mukana, halusivat tuoda selkeämmät implementointi ohjeet. Siitä seurasi DASH264-tekniikan kehitys, joka mahdollisti videossa käytettävän H.264-koodekin tuen, tehokkaamman ääniformaatin HE-AAC v2 ja yleisen tuen sisällön suojaukselle (DRM). Palveluiden ylläpitäjien ei tarvinnut enää nojautua muiden yritysten omistamien sisällönsuojaus tekniikoiden käyttämiseen. (Zambelli 2013)

2.1 Lähtölaukaus musiikin suoratoistolle

Puhuttaessa musiikin suoratoistosta miten se sai alkunsa kiteytyy digitaalisen musiikin alkusysäykseen juuri ennen vuosituhannen vaihtumista. Yritys nimeltään Napster loi jotain aivan uutta, miten musiikkia voisi kuunnella sen tultua yleisön käyttöön toukokuussa 1999. Se perustui tiedostonjakelupalveluun, johon jokainen kytkeytyisi omalla tietokoneellaan, ja voisi ladata musiikkia muiden tietokoneelta. Napsterin omille palvelimille ei tallentunut musiikkitiedostoja. Parhaimmillaan palvelulla oli käyttäjiä 57 miljoonaa. Myöhemmin palvelu lopetettiin sen aikaisessa muodossaan tekijänoikeusrikkomusten vuoksi. (Lamont 2013)

Tämä avasi uuden markkinaraon, johon Spotify vastasi. Yrityksen ideana oli tarjota laaja musiikkikirjasto kaikille pienä maksua vastaan tai kuuntelemalla mainoksia kappaleiden välissä. Se tapahtui vuonna 2005. Kyseinen aika laittoi musiikin suoratoiston varsinaisesti käyntiin. Sen jälkeen on tullut muita musiikin suoratoistopalveluita esimerkiksi Applelta ja Amazonilta, mutta siitä huolimatta Spotify on pitänyt selvää etumatkaa muihin kilpailijoihinsa. (Lamont 2013)

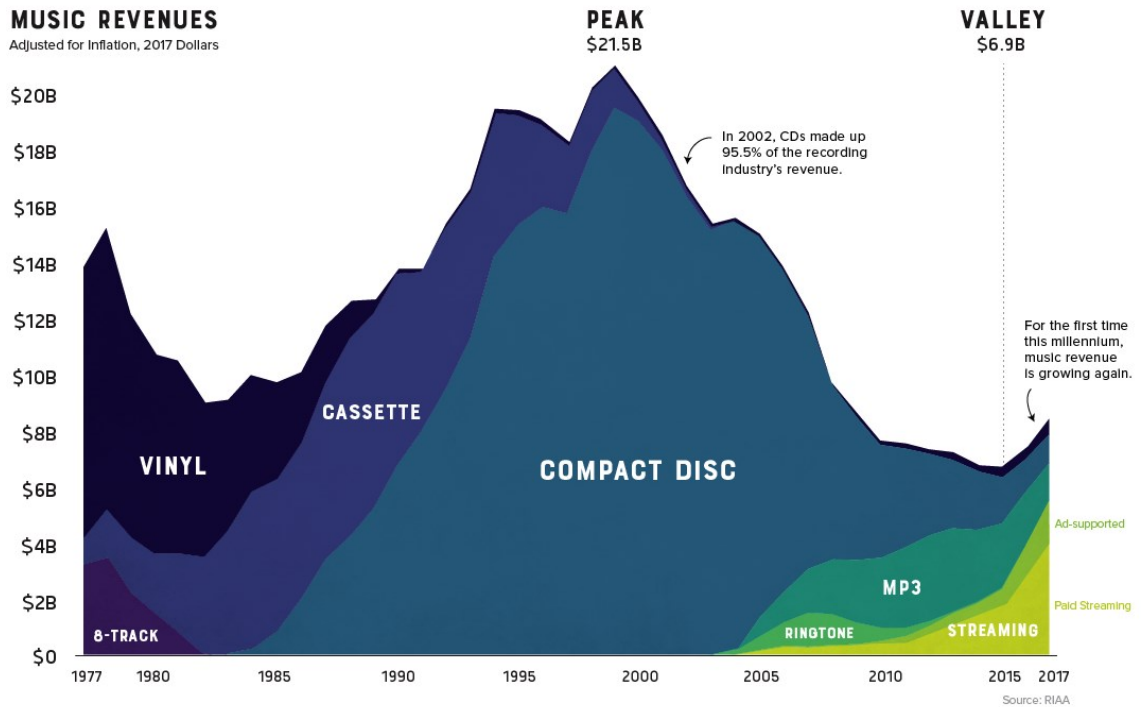
Kun tarkastellaan musiikin suoratoiston valta-aseman titteliä vuosina 2019–2020 on sitä pitänyt ylivoimaisesti hallussaan Spotify noin 140 miljoonalla tilaajamäärällään, jonka jälkeen toiseksi sijoittuu Apple Music noin puolet pienemmällä tilauskannalla, 70 miljoonaa (Kuvio 1).



Kuvio 1. Suoratoistopalveluiden tilaajat ajalta 2019–2020 (Music Ally 2020, muokattu).

2000-luvulle tultaessa MP3-soittimet alkoivat yleistymään, ja musiikin kuuntelu fyysisessä olomuodossa (CD, LP, C-kasetti) alkoi selvästi ottamaan alamäkeä alleen, kun Napster sysäsi musiikin kuuntelun tapahtuvaksi kappaleiden lataamisella Internetistä ihmisten tietokoneille kodista käsin. Myöhemmin suoratoiston otettua vahvan otteen, ja musiikin myynnin siirtyminen enemmän verkkosivuille digitaalisessa

muodossa, heikensi merkittävästi fyysisten olomuotojen menekkiä, joka voidaan selkeästi huomata kuviossa 2.



Kuvio 2. Musiikista saadut tulot fyysisissä ja digitaalisissa muodoissa 40 vuoden ajalta vuosilta 1977–2017 (Routley 2018).

3 PALVELUN PERUSTAMINEN

Kun suunnitellaan musiikin suoratoistopalvelun perustamista, on hyvä aluksi miettiä kuten kaikissa muissakin yrityksen perustamisen aloitusvaiheessa liiketoimintamalleja, joilla voidaan saada juuri oikea kaupallistamismalli omaan tarkoitukseen.

Tässä tuodaan ilmi kolme eri mallia, joita suositaan musiikin suoratoiston yrityksissä. Ne ovat mainosrahoitteinen, tilaukseen perustuva ja tavallinen kaupankäynti. Mainosrahoitteisessa mallissa mainostajat maksavat kuulijalle musiikin kuuntelun, mutta mainoksia esitetään kappaleiden välissä. Nykyään monilla sivulla on mahdollisuus tilata palveluun käyttöoikeus vähintään kuukaudeksi, ja palvelu hinnoitellaan kuukausittain maksettavalla summalla. Tällöin kuluttaja voi esimerkiksi kuunnella musiikkia ilman mainoksia ja paremmalla äänenlaadulla sekä saada muita etuja. Viimeinen ja tutuin muoto on kertamaksuun perustuva kaupankäynti. Se takaa kuluttajalle, että ostaessaan kappaleen tai albumin hän voi kuunnella sitä missä ja milloin vaan ilman mitään lisäkuluja, ja se jää hänelle omaksi. (Boricha 2020)

Erityisesti musiikin suoratoistopalveluissa voidaan ne vielä liiketoimintamallin lisäksi jaotella kahteen alustan käyttömalliin, joista ensimmäinen perustuu yhteisöön, jossa kaikilla on mahdollisuus ladata omaa musiikkia palveluun, joista yksi tunnettu on SoundCloud. Toisessa mallissa puhutaan brändistä, johon päästääkseen on suoratoistomaksu suoritettava, jotta omaa musiikkia saa siellä kuuluville. Näitä palveluja ovat Spotify, Tidal, Amazon Music jne. (Boricha 2020)

3.1 Tekijänoikeudet

Suurin osa musiikista, jota maailmassa kuunnellaan, on tekijänoikeuksilla suojattu. Se tarkoittaa, että artisti, joka on luonut kappaleen omaa yksinoikeuden käyttää sitä ja samalla omistaa sen. Näin ollen muilla ei ole lupaa käyttää sitä. Tekijänoikeus antaa suojan vain persoonalliselle ilmaisumuodolle, jolloin teoksen idea, tietosisältö ja teoria jäävät vapaan käyttöön. Tekijänoikeuden suoja ulottuu laissa kolmeen tahoon, jotka ovat teoksen tekijä, teoksen esittäjä ja teoksen rahoittamiseen sitoutunut yritys. Erilaiset henkilöt yritykset voivat ostaa teokseen ns. käyttöoikeuden, jonka maksun suuruus määräytyy sen käytön laajuuden mukaan. (Tekijänoikeus.fi 2021)

Kun artisti solmii levy-yhtiön kanssa sopimuksen ns. levytyssopimuksen, hän luovuttaa yhtiölle tekijänoikeuden tuotettaviin äänitteisiin, mutta saa niistä rojaltein itselleen. Rojalteissa tekijälle maksetaan teoksen eli äänitteen tuoma tuotto palkkana, jonka suuruus määritellään määräprosentteissa. (Keune, Tomaszewski 2011)

Tavalliselle kuulijalle, kun hän hankkii musiikkia fyysisessä (CD, LP tai kasetti) tai digitaalisessa muodossa hän ei varsinaisesti omista kyseistä musiikkia, vaan hänelle luovutetaan käyttöoikeus siihen. Ja se on tarkoitettu vain hänen yksityiseen käyttöönsä.

3.2 Musiikin hankinta

Hankittaessa musiikkia suoratoistopalveluun, on sen käyttöön hankittava lisenssi. Niitä on mahdollista saada erilaisista musiikin oikeuksien järjestöistä esimerkiksi BMI:stä. BMI on perustettu vuonna 1939, ja on suurin musiikin oikeuksien järjestö Yhdysvalloissa. Kyseinen organisaatio kerää lisenssimaksuja liiketoiminnoista, jotka käyttävät musiikkia lauluntekijöiden, säveltäjien ja musiikin kustantajien puolesta. Kerätyt lisenssimaksut välitetään sitten eteenpäin niille rojalteina, joiden teoksia on esitetty. (SoundMachine 2020)

Suomessa vaikuttaa musiikin tekijänoikeusjärjestö Teosto, jolla on tarjottavana erilaisiin käyttötarkoituksiin soveltuvia käyttöluvia eli lisenssejä. Sieltä löytyy esimerkiksi ”Musiikin suoratoistopalvelut, kuluttajamaksulliset”, jossa on määritelty, että palvelut koostuvat vain musiikista ja/tai musiikkivideoista sekä että se on kuluttajille maksullinen (Kuva 1). Toinen ”Musiikin suoratoistopalvelut, mainosrahoitteiset” on tehty palveluille, jotka tarjoavat suoratoistopalveluita, jotka ovat mainosrahoitteisia, ja joihin kuuluu vain suoratoisto eikä latauksia (Kuva 2). (Teosto 2021)

Kun on päätetty minkälaista liiketoimintamallia suoratoistopalvelulla halutaan toteuttaa, on sen jälkeen helppo valikoida oikea käyttöluva palvelun tarkoitusta mukailten.

Musiikin suoratoistopalvelut, kuluttajamaksulliset



- **Palveluille**, jotka tarjoavat kuluttajille **maksullisia musiikin suoratoistopalveluita (audio streaming)**
- Palveluiden sisältö **pelkästään musiikkia ja/tai musiikkivideoita**
- Antaa oikeuden tallentaa toimijan hallinnoimille palvelimille Teoston/NCB:n edustamia teoksia ja välittää niitä palvelun loppukäyttäjille niin, ettei niistä jää loppukäyttäjälle pysyvää kopiota.

[Lue lisää](#)

Kuva 1. Käyttöluva maksulliselle suoratoistopalvelulle (Teosto 2021).

Musiikin suoratoistopalvelut, mainosrahoitteiset



- **Palveluille**, jotka tarjoavat **mainosrahoitteisia musiikin suoratoistopalveluita**
- Antaa oikeuden tallentaa toimijan hallinnoimille palvelimille Teoston/NCB:n edustamia teoksia ja välittää niitä palvelun loppukäyttäjille
- Kattaa **vain striimauksen**, ei latauksia

[Lue lisää](#)

Kuva 2. Käyttöluva mainosrahoitteiselle suoratoistopalvelulle (Teosto 2021).

4 ÄÄNIKOODEKIT

Äänitiedostot rakentuvat kahdesta osasta, säiliöstä (engl. container) ja koodekista (engl. codec). Koodekin tehtävänä on tallentaa muokkaamaton mediasignaali binaariseen muotoon ja pakata se pienempään ja jakeluystävällisempään kokoon. Suurin osa koodekeista hävittää tietoa (engl. data) koodaamisen (encoding) yhteydessä. Näitä koodekkeja kutsutaan häviöllisiksi. Päinvastaisessa koodaamisessa (engl. decoding) mediasignaali muunnetaan takaisin binaarisesta muodosta alkuperäiseen signaaliin. Säiliö on määritelty tiedostomuoto, joka kuvaa kuinka erilaiset multimedian tietoinneokset (engl. multimedia data elements) ja metatiedot esiintyvät tiedostossa. Kun koodekki muunnetaan toiseen, puhutaan koodinvaihdosta (engl. transcoding). Tarkemmin siinä ensimmäiseksi dekodataan vanhasta koodekista lähtömediasignaali auki eli muokkaamattomaan muotoon, jonka jälkeen se päinvastaisesti koodataan uuteen koodekkiin, ja näin ollen jälleen binaariseen muotoon. (Trattnig 2019)

Kun päätetään mitä koodekkia käytetään suoratoistamiseen, on valikoitava palvelun käyttötarkoitukseen sopiva malli. Äänikoodekit voidaan karkeasti jakaa kolmeen eri luokkaan, häviöllisesti pakattuun (engl. lossy compression), häviöttömästi pakattuun (engl. lossless compression) tai häviöttömästi ei-pakattuun (engl. uncompressed). Häviöllisesti pakkaamalla saadaan äänitiedoston kokoa paljon pienemmäksi, mutta se menetetään äänenlaadussa, kun vastaavasti häviöttömästi ei-pakatussa äänitiedosto vastaa täydellistä digitaalista kopiota analogisesta äänitteestä. Jälkimmäistä pidetään kaikista täsmällisimpänä. (Lee 2019)

Eniten suoratoistopalveluissa hyödynnetään häviöllisiä koodekkeja, mutta nykyään on olemassa muutamia palveluita, joissa musiikkia voi kuunnella häviöttömällä laadulla esimerkiksi Tidalissa. Viimeksi mainitun kaltaisessa palvelussa halutaan tarjota kuulijalle HiFi-tasoinen kuuntelukokemus.

Vaikka käytetään häviöllisesti pakattuja koodekkeja, ne ovat sen verran laadukkaita, että ihmisen korvalla on vaikea erottaa musiikin äänenlaadun hienoista laskua. (Tables 2020)

Suosituimpia äänentoistoon tarkoitettuja koodekkeja ovat MP3 ja AAC, jotka molemmat ovat häviöllisiä ääniformaatteja. Näistä MP3 on saavuttanut suurimman osuuden äänimarkkinoilla. Se esiteltiin vuonna 1991. Myöhemmin kehitettiin AAC, joka mahdollistaa

suuremman pakkaamisen paremmalla äänen toistotarkkuudella (engl. fidelity), jolloin samalla siirtonopeudella verrattuna MP3:een saadaan aikaisiksi parempaa äänenlaatua. Vaikka AAC ei ole saanut samaa jalansijaa kuin MP3, käytetään sitä esimerkiksi musiikkikappaleiden ostoon keskittyneissä verkkokaupoissa. (MDN Web Docs 2021) Kuvassa 3 on lueteltu ja selitetty yleisimmät formaatit.



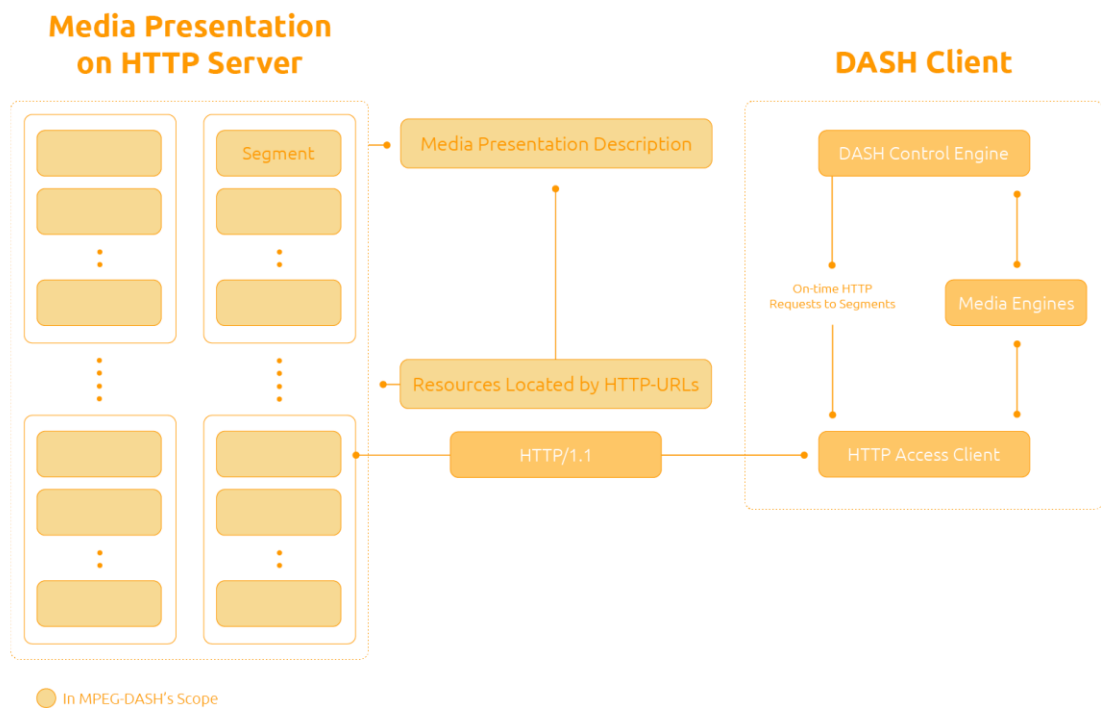
Kuva 3. Yleisimmät ääniformaatit selitettynä (Scarrott 2021, muokattu).

Kun kuuntelee artistin CD-albumia, on siellä kappaleet enkoodattu PCM-muotoon, joka on täysin häviöttömästi ei-pakattu ääniformaatti. PCM-formaatti edustaa digitaalista näytettä analogisista äänialloista. Sen toimintaa perustuu siihen, että äänialloista otetaan ja tallennetaan näyte tietyn väliajoin tai pulssein, josta PCM onkin saanut nimensä eli Pulse-Code Modulation. (Kyrnin 2020)

Musiikkitiedostossa voi usein nähdä mikä on näytteenottotaajuus ja bittisyvyys. Näytteenottotaajuus kuvaa, kuinka usein näyte luodaan eli esimerkiksi tyypillisessä 44 100 Hz:n taajuudessa analogista ääntä tallennetaan ottamalla 44 100 kappaletta näytteitä sekunnissa. Bittisyvyys kuvaa, kuinka monta bittiä on käytetty näytteen esitykseen. Yleisesti voidaan sanoa, mitä suurempi bittimäärä sitä tarkempi näyte eli parempi esitys analogisesta äänestä digitaalisessa muodossa. (Kyrnin 2020)

5 SUORATOISTOTEKNIikka

On olemassa erilaisia suoratoistoon tarkoitettuja protokolleja, jotka toimivat HTTP:llä. Tässä työssä käsitellään dynaamisesti adaptoituvaa suoratoistoprotokollaa nimeltään MPEG-DASH. Sen toiminta perustuu siihen, että DASH-asiakasohjelma (engl. DASH Client) hakee HTTP-palvelimelta MPD-tilun (engl. Media Presentation Description), joka esittelee sisällön, sen vaihtoehtoiset koodatut mediaversiot, niiden URL-osoitteet ja muita protokollaan liittyviä muita piirteitä. Taulun lisäksi palvelimelta löytyy tallennettu sisältö segmentteinä, jotka sisältävät varsinaiset median tietovirrat kimpaleina (engl. chunks) joko yksittäisenä tiedostona tai lukuisina tiedostoina (Kuva 4). (Encoding.com 2021)



Kuva 4. MPEG-DASH:n toimintarakente (Encoding.com 2021).

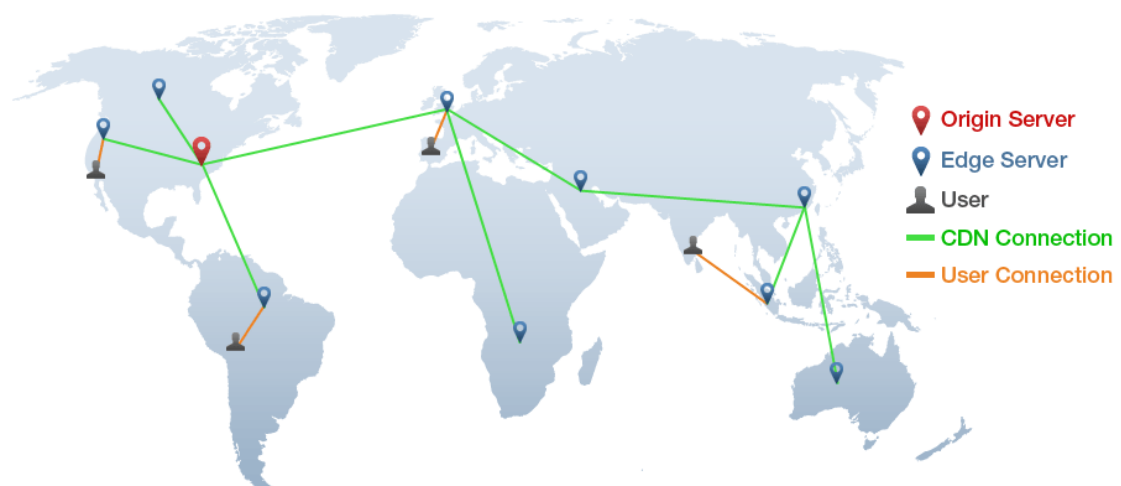
Aloittaakseen sisällön toiston DASH-asiakasohjelma käy poimimassa MPD-tilun palvelimelta, joka toimitetaan HTTP:n välityksellä. Myös muita siirtotapoja voidaan käyttää kuten esimerkiksi sähköpostia ja USB-muistia. Asiakasohjelma jäsentää taulun, jolloin se oppii ymmärtämään ohjelman ajoituksen, saatavilla olevan mediasisällön, mediatyypit, pienimmän ja suurimman tarvittavan kaistanleveyden (engl. bandwidth) ja vaadittavat DRM-oikeudet sekä paljon muuta tietoa. Kyseisillä tiedoilla asiakasohjelma

valitsee sopivan koodatun vaihtoehdon, jonka jälkeen suoratoisto voi alkaa puskuroinnilla (engl. buffering). Se tapahtuu hakemalla segmenttejä HTTP GET-pyyntöillä palvelimelta. (Encoding.com 2021)

DASH-asiakasohjelma jatkaa jälkeisten segmenttien hakemista alussa tapahtuneen puskuroinnin jälkeen, jolloin se valitsee bittinopeuden vertailemalla segmentin latausnopeutta. Asiakasohjelma seuraa jatkuvasti nopeuden huojuntaa, ja mukautuu mittauksiensa perusteella hakemalla eri vaihtoehtoisia segmenttejä, joissa käytetään joko pienempää tai suurempaa siirtonopeutta. Tällä tavoin ylläpidetään sopiva laatu suhteutettuna verkon kaistanleveyteen. (Encoding.com 2021)

5.1 Sisällönjakeluverkot

Varmistaakseen palvelun sujuvan suoratoiston ympäri maapalloa on audiostriimi ohjattava sisällönjakeluverkkoihin (CDN), jotka ovat eri puolille maapalloa hajautettuja palvelimia, jotka toimivat yhdessä tarjotakseen nopean Internet-sisällön jakelun koko maailmalle (Kuva 5). Varsinkin kun palvelun kappaleita suoratoistetaan tuhansia kertoja samanaikaisesti, olisi vain yhden paikallisen palvelimen käyttäminen aivan mahdoton työtaakka, koska se ei millään pystyisi prosessoimaan ja ohjaamaan liikennettä ajoissa kaikille käyttäjille. (Cloudflare 2021)



Kuva 5. Sisällönjakeluverkon toiminta (GTmetrix 2017).

6 PALVELUN TEKNINEN TOTEUTUS

Palvelun perustaminen aloitetaan rakentamalla HTTP-palvelin. Esimerkissä on päätetty käyttää tietokonetta, johon on asennettu Ubuntu Server 20.04 LTS käyttöjärjestelmä. Asennus ja konfigurointi tarvitsee pääkäyttäjän (engl. root) oikeudet sekä perustietämyksen Linuxista, koska komentoja ei käydä tässä yksityiskohtaisesti läpi. Käytetyt komennot on esitelty kursivoituna.

Esimerkissä keskitytään tekniseen toteutukseen ei niinkään mediatiedostojen prosessointiin ja HTML-sivun ulkoasuun. Tiedostojen prosessointi käydään läpi ainoastaan kuvien avulla. Tätä palvelua ei viedä Internetiin saakka, vaan se toimii ainoastaan kiinteistön lähiverkossa ilman sisällönsuojausta (DRM) eikä sen vuoksi ole soveltuva käytettäväksi julkisessa verkossa. Jos palvelu haluttaisiin viedä verkkoon MPEG-DASH tukee CENC-suojausta (engl. Common Encryption). Suojauksen tekoa ei tulla esittelemään tässä esimerkissä, koska se on haluttu pitää mahdollisimman yksinkertaisena. Porttina ei käytetä HTTP:n vakioporttia 80 vaan 8024.

Ensimmäiseksi otetaan SSH-yhteys palvelimelle avaamalla asiakasohjelman (engl. client) tietokoneen komentorivi ja kirjoittamalla komento:

```
$sudo username@ip-address
```

Kun yhteys on saatu muodostettua palvelimelle, asennetaan tarvittavat ohjelmat muun muassa Apache HTTP-verkkopalvelin ja videon ja äänen prosessointityökalu FFmpeg komendoilla:

```
$sudo apt install apache2
```

```
$sudo apt install ffmpeg
```

Seuraavaksi luodaan kolme kansiota ja muutetaan niiden omistaja ja oikeudet:

```
$sudo mkdir /var/www/musapalvelu
```

```
$sudo mkdir -p /var/www/musastatic/dash
```

```
$sudo mkdir /var/www/musacontent
```

```
$sudo chown -R $USER:$USER /var/www/musapalvelu
```

```
$sudo chown -R $USER:$USER /var/www/musastatic
```

```
$sudo chown -R $USER:$USER /var/www/musacontent
```

```
$sudo chmod -R 755 /var/www/musapalvelu
```

```
$sudo chmod -R 755 /var/www/musastatic
```

```
$sudo chmod -R 755 /var/www/musacontent
```

Musapalvelu sisältää HTML-sivun, musastatic JavaScript-tiedoston, jossa on DASH-mediasoitin (dash.all.min.js), ja musacontent eri kappaleille luodut hakemistot, joiden sisältä löytyy kappaleen segmentoidut äänitiedostot ja MPD-taulu.

Apache2:n konfigurointi aloitetaan luomalla uusi VirtualHost-tiedosto komennolla:

```
$sudo vim /etc/apache2/sites-available/musapalvelu.conf
```

Ja valmiiksi tehtynä se on kuvan 6 mukainen.

```
<VirtualHost *:8024>
  ServerName musapalvelu
  DocumentRoot /var/www/musapalvelu

  Alias /musastatic/ /var/www/musastatic/
  Alias /musacontent/ /var/www/musacontent/

  <directory /var/www/musapalvelu>
    AllowOverride all
    Require all granted
    Options FollowSymlinks
  </directory>

  <directory /var/www/musastatic>
    AllowOverride all
    Require all granted
    Options FollowSymlinks
  </directory>

  <directory /var/www/musacontent>
    AllowOverride all
    Require all granted
    Options FollowSymlinks
  </directory>

  CustomLog /var/log/apache2/musapalvelu-access.log common
  ErrorLog /var/log/apache2/musapalvelu-error.log

</VirtualHost>
```

Kuva 6. Määritelty VirtualHost-tiedosto Apache-palvelimella.


```
$sudo ufw allow 8024
```

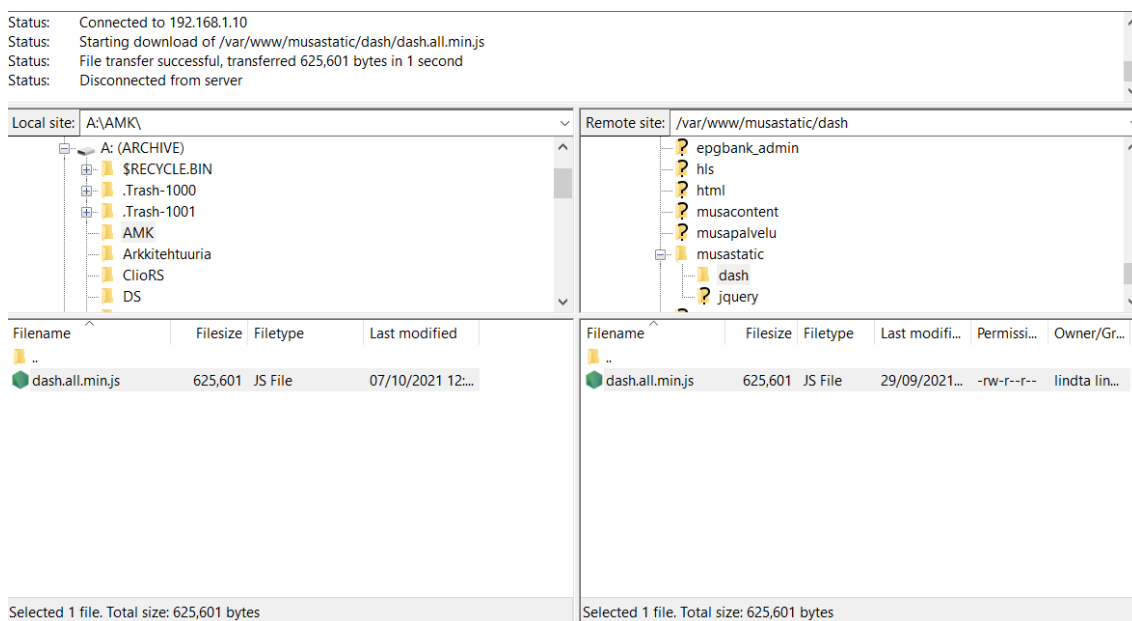
Palvelin on tämän jälkeen toimintakykyinen, mutta sieltä vielä puuttuu itse sisältö.

Seuraavaksi tarvitaan mediasoitin, joka osaa toistaa MPEG-DASHia, koska selaimet eivät itse osaa toistaa sitä. Tässä käytetään dash.js soitinta, joka tullaan upottamaan HTML-sivulle. Soitin voidaan ladata sivulta: <http://cdn.dashjs.org/latest/dash.all.min.js>

Ladattu tiedosto pitää siirtää palvelimelle musastatic hakemistoon.

Tätä varten voidaan käyttää erilaisia tiedonsiirtoon tarkoitettuja ohjelmia kuten esimerkiksi FileZillaa.

FileZillalla otetaan yhteys palvelimeen ja kopioidaan tiedosto hakemistoon `/var/www/musastatic/dash/` (Kuva 8).



Kuva 8. Mediasoitin kopiointi palvelimelle.

Nyt luodaan HTML-sivu alla olevien kuvien 9 ja 10 mukaisesti. Sen teko on käytetty koodin käsittelyyn suunnattua Visual Studio Code -ohjelmaa. Jokaiselle kappaleelle on luotu oma sektio `div`-merkinnällä ja audio-elementille oma uniikki `id`, jolloin musiikkikappaleet saadaan eriteltä omiin soittimiinsa. Soittimet tarvitsevat JavaScript-alustuksen.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="utf-8">
5  <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  <meta name="viewport" content="width=device-width, initial-scale=1">
7  <meta name="description" content="">
8  <meta name="author" content="">
9  <title>Musapalvelu</title>
10 </head>
11 <html>
12 <body>
13   <div>
14     <h1>Halo Helsinki!</h1>
15     <audio id="audioPlayer1" controls="true"></audio>
16   </div>
17
18   <div>
19     <h1>J. Karjalainen</h1>
20     <audio id="audioPlayer2" controls="true"></audio>
21   </div>
22
23   <div>
24     <h1>Dire Straits</h1>
25     <audio id="audioPlayer3" controls="true"></audio>
26   </div>
27
28   <script type="text/javascript" src="/musastatic/jquery/jquery-3.4.1.min.js"></script>
29   <script type="text/javascript" src="/musastatic/dash/dash.all.min.js"></script>
30
31   <script>
32     function audioPlayer1() {
33       var audio,
34         player,
35         url = "/musacontent/kappale1/kappale1.mpd";
36
37       audio = document.querySelector("#audioPlayer1");
38       player = dashjs.MediaPlayer().create();
39       player.initialize(audio, url, true);
40     }
41   </script>
42
43   <script>
44     function audioPlayer2() {
45       var audio,
46         player,
47         url = "/musacontent/kappale2/kappale2.mpd";
48
49       audio = document.querySelector("#audioPlayer2");
50       player = dashjs.MediaPlayer().create();
51       player.initialize(audio, url, true);
52     }
53   </script>

```

Kuva 9. HTML-sivu (Osa 1).

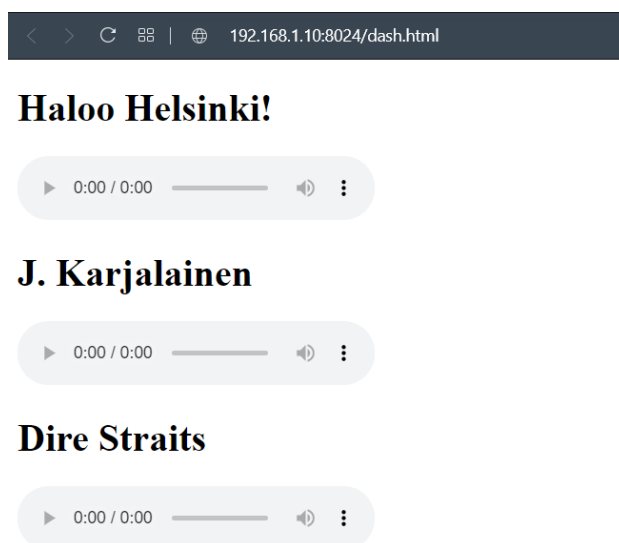
```

54
55 <script>
56     function audioPlayer3() {
57         var audio,
58         player,
59         url = "/musaccontent/kappale3/kappale3.mpd";
60
61         audio = document.querySelector("#audioPlayer3");
62         player = dashjs.MediaPlayer().create();
63         player.initialize(audio, url, true);
64     }
65 </script>
66
67 <script>
68     document.addEventListener('DOMContentLoaded', function () {
69         audioPlayer1();
70         audioPlayer2();
71         audioPlayer3();
72     });
73 </script>
74 </body>
75 </html>
76

```

Kuva 10. HTML-sivu (Osa 2).

Valmis HTML-sivu kopioidaan FileZillan avulla HTTP-palvelimen kansioon /var/www/musapalvelu. Tämän jälkeen voidaan alustavasti kokeilla, että palvelin toimii oikein. Kuvassa 11 huomataan sivun sen hetkinen näkymä.



Kuva 11. Sivun näkymä ennen mediasisältöä.

Nyt voidaan aloittaa äänitiedostojen muuttaminen DASH-muotoon. Sitä varten on tehty helpottava skripti, jonka ajo luo MPD-taulun, ja kahden sekunnin mittaisia segmenttejä mediasisällöstä käyttäen FFmpegia. FFmpegia on käytetty, koska se tukee mediatiedoston en- ja dekodauksen monista eri formaateista.

Skriptiin on määritelty, että se luo neljää eri laatuversio varianttia alkuperäisestä tiedosta (48 kb/s, 96 kb/s, 128 kb/s ja 320 kb/s. Kuvassa 12 nähdään skriptin rakenne.

```

1 #!/bin/bash
2 FILENAME=$1
3 echo "Start processing"
4 echo "$FILENAME"
5 ffmpeg -i "$FILENAME" \
6 -map a:0 -c:a:0 aac -b:a:0 320k -ac 2 \
7 -map a:0 -c:a:1 aac -b:a:1 128k -ac 2 \
8 -map a:0 -c:a:2 aac -b:a:2 96k -ac 2 \
9 -map a:0 -c:a:3 aac -b:a:3 48k -ac 2 \
10 -f dash \
11 -init_seg_name init\${RepresentationID}\$.\$ext\$ \
12 -media_seg_name chunk\${RepresentationID}\$-\${Number%05d}\$.\$ext\$ \
13 -use_template 1 -use_timeline 1 \
14 -seg_duration 4 -adaptation_sets "id=0,streams=a" \
15 "${FILENAME%.*}".$mpd
16 echo "${FILENAME%.*}"
17 echo "Dash generated"

```

Kuva 12. Skriptin rakenne.

Skripti `gen.sh` ja alkuperäinen mediatiedosto kopioidaan omaan hakemistoon palvelimelle. Ennen skriptin käynnistystä on sille annettava ajo-oikeus, joka tapahtuu komennolla:

```
$chmod +x gen.sh
```

Segmenttien generointi ja transkoodaus alkaa, kun skripti käynnistetään:

```
$ ./gen.sh kappale1.mp3
```

Jokaiselle suoratoistettavalle kappaleelle tehdään sama prosessi. Kuvissa 13 ja 14 näkyy kuinka skripti luo segmenttejä.

```

drwxrwxr-x 2 hirmuinen hirmuinen 4096 loka 1 22:21 /
drwxrwxrwx 7 hirmuinen hirmuinen 4096 loka 1 22:22 /
-rwxrwxr-x 1 hirmuinen hirmuinen 529 syys 29 20:22 gen.sh*
-rw-rw-r-- 1 hirmuinen hirmuinen 4117313 syys 29 16:12 kappale1.mp3
hirmuinen@xander-HP-ZBook-15-G3:~/musapalvelu/kappale1$ ./gen.sh kappale1.mp3
Start processing
kappale1.mp3
ffmpeg version n4.3.1 Copyright (c) 2000-2020 the FFmpeg developers
  built with gcc 7 (Ubuntu 7.5.0-3ubuntu1~18.04)
  configuration: --prefix=/usr --disable-debug --disable-doc --disable-
-static --enable-cuda --enable-cuda-sdk --enable-cuvid --enable-libdrm --enable-
ffplay --enable-gnutls --enable-gpl --enable-libass --enable-libfdk-aac --enable-
-libfontconfig --enable-libfreetype --enable-libmp3lame --enable-libnpp --enable-
-libopencore_amrnb --enable-libopencore_amrwb --enable-libopus --enable-libpulse
--enable-sdl2 --enable-libspeex --enable-libtheora --enable-libtwolame --enable-
-libv4l2 --enable-libvorbis --enable-libvpx --enable-libx264 --enable-libx265 --en-
able-libxcb --enable-libxvid --enable-nonfree --enable-nvenc --enable-omx --en-
able-openssl --enable-opencl --enable-runtime-cpu-detect --enable-shared --enable-
vaapi --enable-vidua --enable-version3 --enable-xlib
  libavutil      56. 51.100 / 56. 51.100
  libavcodec     58. 91.100 / 58. 91.100
  libavformat    58. 45.100 / 58. 45.100
  libavdevice    58. 10.100 / 58. 10.100
  libavfilter     7. 85.100 / 7. 85.100
  libswscale     5.  7.100 / 5.  7.100
  libswresample  3.  7.100 / 3.  7.100
  libpostproc   55.  7.100 / 55.  7.100
Input #0, mp3, from 'kappale1.mp3':
  Metadata:
    artist       : RIFFINFO_IART
    comment      : RIFFINFO_ICMT
    date         : RIFFINFO_ICRD
    genre        : RIFFINFO_IGNR
    title        : RIFFINFO_INAM
    ISRC         : RIFFINFO_ISRC
    encoder      : Lavf58.45.100
  Duration: 00:04:17.31, start: 0.025057, bitrate: 128 kb/s
  Stream #0:0: Audio: mp3, 44100 Hz, stereo, fltp, 128 kb/s
  Metadata:
    encoder      : Lavc58.91
Multiple -ac options specified for stream 0, only the last option '-ac 2' will b
e used.
Multiple -ac options specified for stream 1, only the last option '-ac 2' will b
e used.
Multiple -ac options specified for stream 2, only the last option '-ac 2' will b
e used.

```

Kuva 13. Skripti ajovaiheessa (Osa 1).

```

[dash @ 0x55591443fb80] Opening 'chunk2-00061.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk3-00061.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk0-00062.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk1-00062.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk2-00062.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk3-00062.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk0-00063.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk1-00063.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk2-00063.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk3-00063.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk0-00064.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk1-00064.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk2-00064.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk3-00064.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk0-00065.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk1-00065.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk2-00065.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
[dash @ 0x55591443fb80] Opening 'chunk3-00065.m4s.tmp' for writing
[dash @ 0x55591443fb80] Opening 'kappale1.mpd.tmp' for writing
size=N/A time=00:04:17.30 bitrate=N/A speed=21.4x
video:0kB audio:18623kB subtitle:0kB other streams:0kB global headers:0kB muxing
overhead: unknown
[aac @ 0x5559144416c0] Qavg: 1446.918
[aac @ 0x55591444c300] Qavg: 350.728
[aac @ 0x55591444de00] Qavg: 348.399
[aac @ 0x55591444f900] Qavg: 329.866
kappale1
Dash generated
hirmuinen@xander-HP-ZBook-15-G3:~/musapalvelu/kappale1$

```

Kuva 14. Skripti ajovaiheessa (Osa 2).

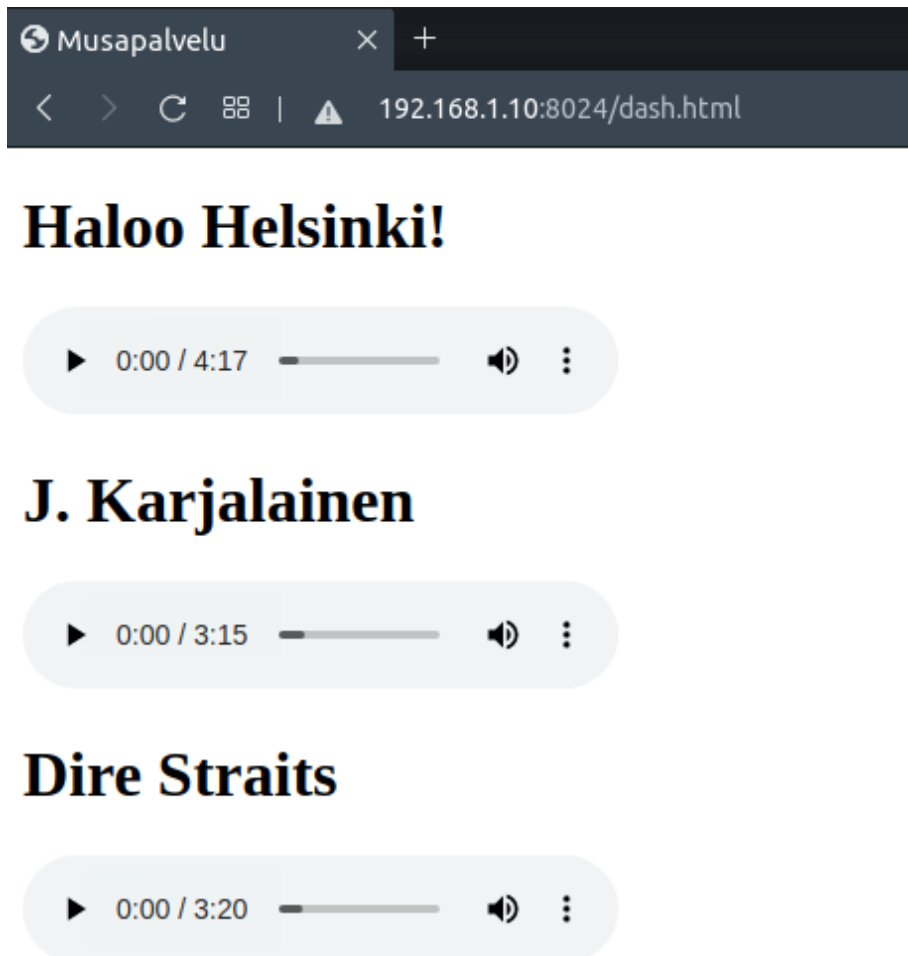
Kun kaikki on konfiguroitu oikein soittimien pitäisi toimia kuvan 15 mukaisesti. CSS-kielellä voi tarvittaessa muokata HTML-sivun tyyliä kattavasti oman makunsa mukaan.

Jos sivu ei jostakin syystä toimi, voidaan tutkia Apache2:n lokitiedostoja ja Ubuntu:n järjestelmälokietieto (engl. syslog), ja havaita mahdolliset virheet. Seuraavilla komennoilla päästään tutkimaan lokitietoja:

```
sudo tail -f /var/log/apache2/musapalvelu-error.log
```

```
sudo tail -f /var/log/apache2/error.log
```

```
sudo tail -f /var/log/syslog
```



Kuva 15. Valmiin ja toimivan verkkosivun näkymä.

7 TIETOTURVA

Perustettaessa palvelua, ja verkkosivua, jossa sen toiminta näkyy, on todella tärkeää ottaa huomioon, miten suojauskäytännöt on saatu toteutettua, sillä palveluun voidaan tehdä kyberhyökkäyksiä lukuisilla eri tavoilla (Kuvio 3). Koska suoratoisto tapahtuu Internetin välityksellä eri puolilla maailmaa, on välttämätöntä kiinnittää tietoturvaan huomiota, jotta voidaan välttää kriittiset tietovuodot. On varmistettava, että yhteys palvelimen ja päätelaitteen välillä on suojattu HTTPS:llä, ja että niiden välillä tapahtuvaa tiedonkulkua ei saataisi kaapattua. Varsinkin suoratoistossa, jossa kappaletta ladataan yhteydettömään tilaan (engl. offline mode), on huolehdittava, että sitä tietoa ei saada käytettyä jälkikäteen tai kopioitua/siirrettyä. Tämä mahdollistaisi sen, että kappaleita voitaisiin ladata palvelusta laittomasti ja pahimmassa tapauksessa laittaa kaikkien saataville ilmaiseksi Internetiin rikkoen tekijänoikeuksia ja aiheuttamalla tulonmenetyksiä palveluntarjoajalle sekä levy-yhtiöille ja artisteille.



Kuvio 3. Yleisimmät käytetyt haittaohjelmat kyberhyökkäyksissä (Statista 2020, muokattu).

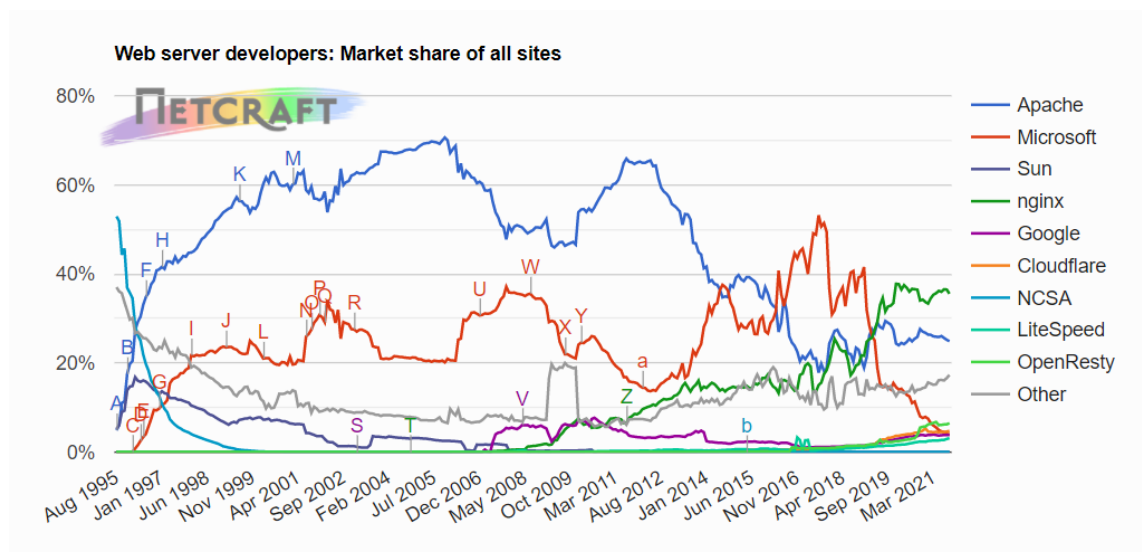
Kun puhutaan tietoturvasta, tarkoitetaan sillä hallinnollisia ja teknisiä toimia, ja tarkemmin määriteltynä erilaisten tietojen, järjestelmien, palveluiden sekä tietoliikenteen asianmukaista suojaamista. Päivittäisessä elämässämme, kun tietojamme käsitellään suullisesti, fyysisesti paperilla tai tietokoneella, kulkee tietoturva näissä asioissa mukana. Edellytyksenä pidetään, että kaikkien käyttäjien tietoturva toteutuu palveluiden käytössänsä. (Kyberturvallisuuskeskus 2020)

Tietoturva voidaan jakaa neljään lokerikkoon, joista se koostuu. Ne ovat tiedon luottamuksellisuus, eheys, käytettävyys ja todentaminen. Luottamuksellisuudella tarkoitetaan sitä, että tiedot ovat vain niiden käyttöön oikeutettujen saatavilla, tällöin evätään sivullisilta mahdollisuus muuttaa tai poistaa tietoja tai ylipäättänsä käsitellä niitä. Eheys merkitsee tietojen ja järjestelmien muuttamattomuutta. Niiden pitää olla luotettavia, oikeita, ja ajantasaisia. Ne eivät saa olla muutettavissa laitteisto- tai ohjelmistovikojen, luonnontapahtumien tai inhimillisen toiminnan seurauksena. Ainakin tietojen päivittämisellä ja säännöllisellä varmuuskopioinnilla voidaan vaikuttaa eheyden laatuun positiivisesti. (Helsingin yliopisto 2021)

Järjestelmien tietojen ja palveluiden ollessa käytävissä niihin oikeutettujen henkilöiden etukäteen määritellyssä vasteajassa puhutaan käytävyydestä. Lokerikon viimeisellä sektorilla olevalla todentamisella tai autentikoinnilla tarkoitetaan osapuolten muun muassa henkilön tai järjestelmän luotettavaa tunnistettavuutta. Salasanoja, muuttuvia avaintunnuksia ja sertifikaatteja käytetään ainakin todentamisen varmentamiseen, joita tarvitaan esimerkiksi verkkosivun kautta tehtävän maksun suorittamiseen. (Helsingin yliopisto 2021)

8 VERKKOSIVUN TYÖKALUT

Eniten käytetty verkkopalvelin ohjelmisto on Apache, joka on toiminnassa noin 25 %:ssa kaikista verkkopalvelimista maailmalla (Kuvio 4). Ohjelmisto on ilmainen ja perustuu vapaaseen lähdekoodiin. Se on laajasti käytetty siksi, koska se luotettava, nopea ja turvallinen. Lisäksi se on todella monipuolisesti muokattavissa eri ympäristöjen tarpeisiin käyttämällä moduuleita ja laajennuksia. Suosittu avoimeen lähdekoodiin perustuva sisällön hallintajärjestelmä WordPress, jolla voi luoda oman verkkosivun tai blogin, käyttää suurelta osin Apache-verkkopalvelin ohjelmistoa. (WPBeginner 2021)



Kuvio 4. Verkkopalvelimien eri ohjelmien käyttösuuksia (Netcraft 2021).

Bootstrap on kahden entisen Twitter-työntekijän luoma ilmainen verkkosovelluskehitystyökalu, joka on yksiselitteisesti valtava kokoelma käteviä, uudelleen käytettäviä koodin pätkiä on kirjoitettu HTML-, CSS- ja JavaScript-kielellä. Myös täysin reagoivien verkkosivujen tekeminen on nopeaa, koska Bootstrap on käyttöliittymän kehys (engl. frontend framework). Olennaisin hyöty on ohjelmoijien säästäminen runsaalta CSS-koodin kirjoittamiselta, jonka ansiosta verkkosivujen muotoiluun voidaan käyttää enemmän aikaa. (Ouellette 2016)

Django on verkkokehityskehys (web framework), joka mahdollistaa nopean kehityksen turvallisille ja helposti ylläpidettäville verkkosivustoille. Se perustuu Python-ohjelmointikielen. Django helpottaa verkkosivun kehittäjää säästämällä aikaa monissa asioissa, joita tulee vastaan verkkokehityksessä antaen käyttäjälleen jouhevan lähdön lähteä rakentamaan

omaa sovellusta käyttämättä liikaa aikaa perustuksiin. Yksi kulmakivistä on, että se on ilmainen ja avoimeen lähdekoodiin perustuva, aktiivinen ja kukoistava yhteisö, monipuolinen dokumentaatio, ja monia ilmaisen ja maksetun tuen vaihtoehtoja. Kun kehittää Djangoilla verkkosivustoa, tulee siitä valmis kokonaisuus, sillä kaikki tarvittavat osat löytyvät yhdestä paikasta. Näin ollen ne toimivat saumattomasti keskenään. (MDN Web Docs 2021)

9 LOPUKSI

Tässä työssä tutkittiin mahdollisuuksia, miten voidaan luoda verkkoon pohjautuva musiikin suoratoistopalvelu. Tavoitteena oli luoda selkeä ja hyödyllinen kokonaisuus, josta käy ilmi suoratoistopalvelun perustamiseen liittyvät olennaisimmat asiat. Aihetta pyrittiin avaamaan riittävästi ja kattavasti, jakamalla aiheen osa-alueet loogisesti eteenpäin etenevään kerrontaan.

Työ perustui konkreettisesti siihen, että tietoa haettiin eri artikkeleista ja tutkimuksista internetin nettisivuilta, ja tehtiin käytännössä toimiva järjestelmä. Ennen kyseistä työvaihetta tutkittavat osa-alueet nimettiin ja jaoteltiin loogiseen etenemisjärjestykseen, jonka jälkeen alkoi tarvittavan materiaalin tiedonkeruu ja analysointi. Lopulta haettu ja luettu tieto koostettiin järkevään muotoon. Teknistä toteutusta on selitetty tekstin lisäksi palvelimen konfiguroinnista otetuilla kuvakaappauksilla, jotka antavat selkeän kuvan siitä, mitä tapahtuu.

Työtä tehdessä tietoa löytyi kattavasti Internetistä aiheeseen liittyen, joten varsinaisia rajoituksia ei tullut vastaan, koska tehtyjä artikkeleita ja tutkimuksia työn eri aiheen osa-alueiden löytyi monipuolisesti.

Saaduilla tuloksilla voidaan esimerkiksi auttaa ihmisiä luomaan uusia omia musiikin ympärille toimivia ja tarkoitettuja suoratoistopalveluja, jotka voidaan soveltaa helposti oman yrityksen toimintaan nojaamatta muiden jo valmiina löytyvien palveluiden käyttöehtoihin ja mahdollisiin lisenssimaksuihin.

Kokonaisuutta voitaisiin kehittää tutkimalla vielä aihetta laajemmin ja syvällisemmin. Mahdollisesti myös uusien aihekokonaisuuksien mukaan ottaminen tähän opinnäytetyöhön on tutkimisen arvoinen asia.

LÄHTEET

Boricha, Mehul 2020: How to Create a Music Streaming Service/App like Spotify. Techrival -verkkosivu, joulukuu 2020. <https://techrrival.com/how-to-create-a-music-streaming-service/>. Viitattu 15.6.2021

Cloudflare 2021: What is streaming? | How video streaming works. Cloudflare -verkkosivu, 2021. <https://www.cloudflare.com/en-gb/learning/video/what-is-streaming/>. Viitattu 14.6.2021

Cloudflare 2021: How does live stream encoding work? | Video encoding. Cloudflare -verkkosivu, 2021. <https://www.cloudflare.com/en-gb/learning/video/live-stream-encoding/>. Viitattu 16.6.2021

Cloudflare 2021: What is a CDN? | How do CDNs work? Cloudflare -verkkosivu, 2021. <https://www.cloudflare.com/en-gb/learning/cdn/what-is-a-cdn/>. Viitattu 16.6.2021

Encoding.com 2021: MPEG-DASH – How Does it Work? Encoding.com -verkkosivu, 2021. <https://www.encoding.com/mpeg-dash/>. Viitattu 22.9.2021

GTmetrix 2017: Why use a Content Delivery Network (CDN)? GTmetrix -verkkosivu, helmikuu 2017. <https://gtmetrix.com/why-use-a-cdn.html>. Viitattu 10.10.2021

Helsingin yliopisto, 2021: Tietoturvan periaatteet. Helsingin yliopisto – Opiskelijan digitaidot -verkkosivu, 2021. <https://blogs.helsinki.fi/opiskelijan-digitaidot/4-tietoturva/4-1-tietoturvan-ja-tietosuojan-perusteet/tietoturvan-edellytykset/>. Viitattu 16.6.2021

ISO/IEC 23009-1:2019, 2019: Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats. ISO.org -verkkosivu, joulukuu 2019. <https://www.iso.org/standard/79329.html>. Viitattu 10.10.2021

Keune, Anna – Tomaszewski, Iris 2011: 6.1 Levytyssopimus. Aalto University Wiki -verkkosivu, joulukuu 2011. <https://wiki.aalto.fi/display/copyright/6.1++Levytyssopimus>. Viitattu 17.6.2021

Kyberturvallisuuskeskus, 2020: Tietoturva. Kyberturvallisuuskeskus – Liikenne- ja viestintävirasto – Traficom -verkkosivu, heinäkuu 2020. <https://www.kyberturvallisuuskeskus.fi/fi/toimintamme/saantely-ja-valvonta/tietoturva>.

Viitattu 16.6.2021

Kyrnin, Mark 2020: Computer Audio Basics: Standards and Digital Audio. Lifewire -verkkosivu, heinäkuu 2020. <https://www.lifewire.com/computer-audio-basics-831415>.

Viitattu 16.6.2021

Lamont, Tom 2013: Napster: the day the music was set free. The Guardian -verkkosivu, helmikuu 2013. <https://www.theguardian.com/music/2013/feb/24/napster-music-free-file-sharing>. Viitattu 14.6.2021

Lee, Joel 2019: The 10 Most Common Audio Formats: Which One Should You Use?. MUO – Make Use Of -verkkosivu, joulukuu 2019. <https://www.makeuseof.com/tag/audio-file-format-right-needs/>. Viitattu 16.6.2021

MDN Web Docs 2021: Web audio codec guide. MDN Web Docs – Mozilla -verkkosivu, toukokuu 2021. https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Audio_codecs. Viitattu 16.6.2021

MDN Web Docs 2021: Django introduction. MDN Web Docs – Mozilla -verkkosivu, toukokuu 2021. <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>. Viitattu 17.6.2021

Music Ally 2020: How many users do Spotify, Apple Music and other big music streaming services have? Music Ally -verkkosivu, heinäkuu 2020. <https://musically.com/2020/02/19/spotify-apple-how-many-users-big-music-streaming-services/>. Viitattu 16.6.2021

Netcraft 2021: September 2021 Web Server Survey. Netcraft -verkkosivu, syyskuu 2021. <https://news.netcraft.com/archives/category/web-server-survey/>. Viitattu 10.10.2021

Ouellette, Alexandre 2016: What is Bootstrap: A Beginner's Guide. CareerFoundry -verkkosivu, lokakuu 2016. <https://careerfoundry.com/en/blog/web-development/what-is-bootstrap-a-beginners-guide/>. Viitattu 17.6.2021

Rayburn, Dan 2016: The Early History Of The Streaming Media Industry and The Battle Between Microsoft & Real. Streaming Media Blog -verkkosivu, maaliskuu 2016. <https://www.streamingmediablog.com/2016/03/history-of-the-streaming-media-industry.html>. Viitattu 14.6.2021

Routley, Nick 2018: Visualizing 40 Years of Music Industry Sales. Visual Capitalist -verkkosivu, lokakuu 2018. <https://www.visualcapitalist.com/music-industry-sales/>. Viitattu 14.6.2021

Scarrott, Becky 2021: MP3, AAC, WAV, FLAC: all the audio file formats explained. What Hi-Fi?, huhtikuu 2021. <https://www.whathifi.com/advice/mp3-aac-wav-flac-all-the-audio-file-formats-explained>. Viitattu 16.6.2021

SoundMachine 2020: How much does a BMI license cost?. SoundMachine -verkkosivu, syyskuu 2020. <https://sound-machine.com/blog/2020/09/25/how-much-does-a-bmi-license-cost/>. Viitattu 17.6.2021

Statista 2020: Most commonly encountered types of malware used in industrial cyber attacks worldwide in 2019. Statista -verkkosivu, huhtikuu 2020. <https://www.statista.com/statistics/271037/distribution-of-most-common-malware-file-types/>. Viitattu 16.6.2021

Tables, Jason 2020: MP3 versus WAV: Can Anyone Tell the Difference?. B & H Foto & Electronics Corp -verkkosivu, joulukuu 2020. <https://www.bhphotovideo.com/explora/pro-audio/features/mp3-versus-wav-can-anyone-tell-the-difference>. Viitattu 16.6.2021

Tekijanoikeus.fi 2021: Mitä on tekijänoikeus?. Tekijanoikeus.fi -verkkosivu, 2021. <https://tekijanoikeus.fi/tekijanoikeus/>. Viitattu 15.6.2021

Teosto 2021: Musiikin käyttöluvut – Hanki lupa. Teosto -verkkosivu, 2021. <https://www.teosto.fi/musiikin-kayttoluvat/hanki-lupa/>. Viitattu 16.6.2021

Trattnig, Armin 2019: Fun with Container Formats – Part 1. Bitmovin -verkkosivu, kesäkuu 2019. <https://bitmovin.com/container-formats-fun-1/>. Viitattu 15.9.2021

WPBeginner 2021: What is: Apache. WPBeginner -verkkosivu, 2021. <https://www.wpbeginner.com/glossary/apache/>. Viitattu 17.6.2021

Zambelli, Alex 2013: A history of media streaming and the future of connected TV. The Guardian -verkkosivu, maaliskuu 2013. <https://www.theguardian.com/media-network/media-network-blog/2013/mar/01/history-streaming-future-connected-tv>. Viitattu 14.6.2021

