



OULUN YLIOPISTO  
UNIVERSITY of OULU

# **Verkkopelipalvelinten viestintäkäytäntöjen jälleenmallintaminen**

Oulun yliopisto  
Tietojenkäsittelytieteiden laitos  
LUK-Tutkielma  
Janne Määttä  
15.3.2021

## Abstrakti

Tässä paperissa tulen käsittelemään palvelinohjelman matkimista, eli emulointia, sekä protokollien jälleenmallintamista ilmiönä. Samalla käsittelen siihen liittyviä mahdollisia juridisia ongelmia käymällä läpi ennakkotapauksia maailmalta. Päätin tarkastella ilmiötä lähemmin massiivisten nettiroolipelien(MMORPG) näkökulmasta, sillä näiden pelien emuloinnin ympärille on kehittynyt yhteisöjä, sekä useita emulaattoreita niin suljettuina, kuin avoimen lähdekoodin ratkaisuinakin. Samoin juuri tämän kategorian peleihin liittyen löytyy esimerkkitapauksia tekijänoikeus kiistoista, joita tarkastelemalla pyrin tuomaan esille tekijöitä jotka mahdollisesti vaikuttavat yksityisten palvelimien toiminnan laillisuuteen immateriaalioikeuksien suhteen.

Tulen myös havainnollistamaan kuinka laajasta ilmiöstä on kyse kun puhutaan nk. Yksityisistä palvelimista. Samalla käyn läpi mahdollisia syitä miksi tällaisen palvelun tarjoaminen on kuitenkin ongelmallista ja usein lainsäädännön harmaalla alueella vaikka jälleen mallintamisen seurauksena tuotettu ohjelmisto olisikin täysin alkuperäinen teos. Pelin protokollan tutkimiseen voi olla muitakin syitä emulaattorin kehityksen lisäksi ja käsittelenkin lyhykäisyydessään myös peleissä huijaamista.

## Sisällysluettelo

Abstrakti.....	2
1 Johdanto.....	4
2 Palvelimen emulointi.....	5
2.1 Protokollien jälleenmallintaminen.....	6
2.2 Syitä palvelinpuolen toiminnallisuuden jäljittelyn taustalla.....	7
2.3 Verkkopelien emulaattoreiden historiaa.....	9
2.4 MMORPG Palvelin emulaattorit ja yhteisö.....	10
3 Protokollien jälleenmallintaminen muihin käyttötarkoituksiin.....	11
3.1 Asiakkaan toimintojen automatisointi.....	12
4 Emulointiin liittyviä mahdollisia juridisia ongelmia.....	13
4.1 Asiakasohjelman käyttö ja manipulointi.....	14
4.2 Immateriaalioikeuksien merkitys yritysten näkökulmasta.....	16
4.3 Esimerkitapauksia palvelin emulointiin liittyvistä oikeustaistoista suomessa ja muualla maailmalla.....	17
Yhteenveto.....	19
Lähteet.....	20

# 1 Johdanto

Internetin välityksellä pelattavien moninpelien suosio on kasvanut räjähdysmäisesti läpi 2000-luvun . Jo hyvin varhaisessa vaiheessa internetin kehityksessä usean pelaajan roolipelit keräsivät suosiota varsinkin opiskelijoiden keskuudessa. Varhaisin esimerkki tästä lie 80-luvulla kehitetty ”MUD” eli *Multi User Dungeon*, jota pelattiin tekstipohjaisen terminaalin välityksellä. Teknologian kehityksen myötä tätä konseptia vietiin eteenpäin ja jo 90-luvun alussa ilmestyivät ensimmäiset graafiset netissä pelattavat roolipelit. Vuosien saatossa internetin käyttäjien määrä kasvoi ja niin kasvoi myös pelaajakuntakin. Suositut pelit kuten Ultima Online, Tibia sekä EverQuest keräsivät ympärilleen satoja tuhansia pelaajia ansaiten näin nimen ”massiivinen moninpeli”. Kenties yksi merkittävimpiä hetkiä internet-roolipelien historiassa tähän mennessä oli 2010-luvun taitteessa kun suur-suosion saavuttanut *World of Warcraft* mittasi ennätysmäiset 12 miljoonaa pelaajaa ympäri maailman.

Merkittävistä suosioista johtuen oli sinänsä täysin odotettavissa että kuvioihin mukaan tulee jo aiemmin muista peleistä tuttu ilmiö; nimittäin piratismi. Moninpeleissä tosin ei riitä että käyttäjän hallussa on pelin asiakasohjelmisto. Tämän lisäksi pelin pitää myös yhdistää palvelimelle ja esimerkiksi edellä mainitun World of Warcraft pelin tapauksessa tilin käyttäminen pelin virallisilla palvelimilla itsessään edellyttää kuukausimaksua, joten pelaamiseen tarvitaan myös piratisoitu (tai usein emuloitu) palvelin ohjelmisto. Kuten piratismiin yleensä, myös palvelin-puolen toiminnallisuuden kopiointiin/matkimiseen liittyy potentiaalisia ongelmia, kuten tulen tässä paperissa osoittamaan. Erityisen mielenkiintoista tästä ongelmasta tekee se, ettei varsinkaan MMORPG peleihin ja niiden palvelinpuolen matkimiseen keskittyvää tutkimustietoa ole paljoa tarjolla siitä huolimatta että kyseessä on verrattaen laajalle levinnyt ilmiö.

Tämän kirjallisuuskatsauksen myötä pyrin kartoittamaan kuinka laajasta ilmiöstä on kyse sekä mitä ongelmia siihen liittyy lainsäädännön näkökulmasta. Samalla käyn lyhyesti läpi myös mitä palvelimen emuloinnilla ylipäänsä tarkoitetaan sekä siihen liittyen myös protokollien jälleenmallintamiseen liittyviä seikkoja avatakseni käsitteitä.

## 2 Palvelimen emulointi

Perinteisessä asiakas/palvelin ratkaisussa yksi tai useampi palvelin vastaa palvelupyyntöihin usealle asiakkaalle. Kommunikointi tapahtuu applikaatio-tason protokollan välityksellä, joka voi olla joko standardoitu avoin protokolla, tai yrityksen sisäinen suljettu protokolla. Emuloinnissa useimmiten kyseessä on määräyksiltään suljettu, mahdollisesti myös lähetyksen ajaksi salattu protokolla. Palvelimen sisäinen toiminta on asiakkaalle näkymätöntä, eikä sitä normaali tilanteessa päästä millään keinoin lähemmin tutkimaan, sillä asiakas ja palvelin-ohjelmisto suoritetaan fyysisesti eri laitteistolla. Palvelimen toiminnallisuus on siis asiakkaan näkökulmasta nk. ”musta laatikko”. Asiakasohjelma siis näkee näkee vain ulospäin menevän sekä sisäänpäin tulevan liikenteen.

Mustan laatikon toimintoja voidaan arvioida tutkimalla asiakkaan ja palvelimen välistä kommunikaatiota niin että pyritään liittämään viestit pelin sisäisiin toimintoihin. Käytännön esimerkkinä tästä jos käyttäjä lähetettää pelin sisällä chat-viesti kahden pelaajan välillä, voidaan olettaa että pelissä kirjoitettu teksti esiintyy palvelimelle lähetettävän viestin mukana. Vastaavasti kontekstia hyödyntämällä pystytään liittämään viestittely pelin sisäisiin toimintoihin ja sen myötä liikenteen ymmärtäminen helpottuu. Tarkkaa mallia palvelimella suoritettavasta ohjelmakoodista ei tällä menetelmällä kyetä selvittämään, mutta riittävän kattava käsitys viestittelystä on mahdollista saavuttaa. Tätä tietoa hyödyntäen voidaan kehittää oma palvelinohjelmisto joka toteuttaa protokollan vastaten asiakasohjelman pyyntöihin oikean, alkuperäisen palvelimen tavoin. Jos palvelimen ja asiakkaan välinen viestittely on protokollan määrittämisen mukaista, ei asiakkaalla ole pelkästään sen perusteella mahdollista päätellä käydäänkö keskustelua alkuperäisen vai jäljennetyn palvelimen kanssa.

Palvelinohjelman jäljennöstä kutsun tässä yhteydessä palvelin emulaattoriksi. Emuloinnin tavoitteena on yleisimmin alkuperäisen laitteiston tai ohjelmiston toiminnallisuuden matkiminen mahdollisimman tarkasti. Kenties hieman emuloinnin perinteisestä määräyksestä poiketen palvelin emulaattori ei välttämättä pohjautu alkuperäiseen palvelinohjelmistoon missään määrin. Itseasiassa alkuperäiseen ohjelmakoodiin saati siitä tuotettuun binääriin ei välttämättä vaadita mitään näkyvyyttä. Varsinainen toteutus itsessään voi poiketa alkuperäisestä täysin, mutta esimerkiksi tässä tapauksessa viestittelyyn käytettävän protokollan täytyy noudattaa alkuperäisiä määräyksiä (lähes) sellaisenaan jotta yhteensopivuus asiakkaan ja palvelimen välillä on mahdollista. Palvelin emulointiin protokolla toteutetaan yleensä jälleenmallintamisen pohjalta (Debeauvais, Nardi, 2010). Protokollan uudelleen määrittäminen ei ole mahdollista ilman että myös asiakasohjelma mukautetaan ymmärtämään sitä. Toteutuksesta riippuen sen sijaan voi olla mahdollista vaikuttaa hieman tilanteisiin missä mitään viestiä käytetään, sekä raja-arvojen puitteissa myös viestien sisältöön. Pelaajan näkökulmasta kyse on edelleen samasta pelistä, sillä pelaajan rajapintana toimiva asiakasohjelma ei välttämättä ole muuttunut alkuperäisestä missään määrin. Ei tosin ole ennenkuulumatonta etteikö myös asiakasohjelmaa voisi muuttaa. Tämä avaa uusia potentiaalisia ongelmia immateriaalioikeuksien näkökulmasta, sillä juuri asiakasohjelma on kopiosuojattu tuote.

## 2.1 Protokollien jälleensallintaminen

Useiden maiden lainsäädännöissä on huomioitu tilanteita missä jälleensallintaminen itsessään on laillista tietyissä tilanteissa, kuten esimerkiksi yhteensopivuuden nimissä. Suomessa 2018 voimaan astunut EU direktiivien mukainen liikesalaisuuslaki (595/2018) määrittelee tuotteen tai esineen havainnoinnin, tutkimisen sekä purkamisen sallituksi mikäli tuote on asetettu yleisön saataville tai muutoin laillisesti tutkivan tahon hallussa (Finlex, 2018). Yleisesti ohjelmistojen jälleensallintamiseen itseensä liittyy kuitenkin myös potentiaalisia juridisia ongelmia, mutta ne jäävät hieman aihepiiriin ulkopuolelle. Tämän paperin puitteissa en myöskään syvenny erityisen pitkälle teknisiin menetelmiin, mutta jotta voidaan paremmin ymmärtää mistä on kyse, käyn lyhesti läpi mitä tarkoitetaan protokollien jälleensallintamisella ja miten se käytännössä toteutuu.

Käytännössä kommunikointia voidaan tutkia joko pelkkää verkkoliikennettä seuraamalla, tai asiakasohjelmaa tutkimalla joko suorituksen aikaisesti, tai pelkkää binääritiedostoa tutkimalla. Eli joko dynaamista tai staattista analyysiä hyödyntäen (Duchene et al, 2018). Jälleensallintamisen laillisuuteen voi mahdollisesti vaikuttaa sekin seurataanko liikennettä passiivisesti vai joudutaanko tutkimaan myös asiakasohjelmiston toiminnallisuutta. Ohjelmisto itsessään on tekijänoikeuksien alainen teos, joten paikallisesta lainsäädännöstä riippuen on hyvin mahdollista että sen tutkiminen on ongelmallisempaa kuin vain verkkoliikenteen seuraaminen. Binääri analyysiin sekä verkkoliikenteen analysointiin löytyy lukuisia eri menetelmiä sekä työkaluja, mutta niihin en tässä paperissa tule sen enempää keskittymään. Yhteistä eri menetelmille on kuitenkin että dokumentoimattoman protokollan täydellinen ymmärtäminen on vaativa ja aikaa vievä prosessi. Toki vaativuus vaihtelee tapauskohtaisesti riippuen protokollan kompleksisuudesta sekä mahdollisista käytetyistä salausmenetelmistä. Lähtökohta kuitenkin varsinkin MMORPG-pelien tapauksissa on että protokolla on hyvin erikoistettu ja kooltaan optimoitu minimoimaan kuormaa. Tämä tarkoittaa sitä että protokolla itse tuskin sisältää mitään selkokieliä kuvauksia sisällöstään jättäen lähes kaiken tiedon arvailun ja tulkinnan varaan. Vastakohtana tähän ovat monisanaiset, selkokielliset ihmisen luettavissa olevat protokollat, kuten usein esim. JSON ja XML -pohjaiset protokollat ovat. Sen lisäksi että suljetun protokollan analysointi jo yksistään on aikaa vievä prosessi, on siitä yhä reilusti matkaa toimivaan emulaattoriin. Yksinkertainen toteutus liene onnistuu lyhyessäkin ajassa, mutta koska kyseessä on massiivinen moninpeli, on vaatimukset palvelimelle myös sen mukaiset. Palvelun tulee skaalautua muutamasta samanaikaisesta käyttäjästä potentiaalisesti muutamaan tuhanteen. Ottaen huomioon että useimmiten emulaattoreita kehittävät tahot ovat itsenäisiä henkilöitä tai ryhmittymiä vailla rahoitusta tai muitakaan yrityksillä käytöissä olevia resursseja, on aiheellista olettaa että emulaattorin kehitykseen kuluu huomattavasti enemmän aikaa kuin alkuperäisen, virallisen palvelinohjelmiston tuotantoon kului.

Esimerkkinä protokollien jälleensallintamisen pohjalta toteutetuista palvelin emulaattoreista mainittakoon avoimen lähdekoodin toteutukset *l2jserver* sekä *ArcEmu*. Kyseiset emulaattorit ovat NCSofthin kehittämälle Lineage 2- sekä Blizzardin World Of Warcraft -peleille. Molemmat pelit ovat suosittuja koostuen miljoonista käyttäjistä virallisilla palvelimilla. Yhteistä näille on myöskin että molempien emulaattorit ovat käyneet läpi vuosien kehityksen avoimen lähdekoodin projekteina, joten on reilua olettaa että pelien kehittäjät ovat tietoisia aiheesta. Toki emulaattorin kehittäminen ja käyttö ovat kaksi eri asiaa.

## 2.2 Syitä palvelinpuolen toiminnallisuuden jäljittelyn taustalla

Emulaattorin kehitys voi olla hyvin monimutkainen ja pitkä prosessi. Varsinkin jos kehityksen ohella joudutaan jälleenmallintamaan koko protokolla. Tämän lisäksi siihen liittyy vielä mahdollisia juridisia ongelmia, joten miksi siis kukaan ryhtyisi siihen? Seurattuani emulaattori yhteisöä tunnistin muutamia syitä. Yksi, kenties jokseenkin arvattava motivaatio toiminnan taustalla on rahallisen hyödyn tavoittelu. Emulaattorin kehitys sekä palvelun ylläpito on kovaa työtä, eikä kehittäjillä keskimäärin ole tukena yritystä maksamassa palkkaa kaikesta vaivasta. Kuinka sitten toiminta voidaan tuotteistaa? Vastaus löytyy palvelun tarjoamisesta.

Suosituimmat MMORPG pelit keräävät ympärilleen jopa miljoonien pelaajien yhteistön (Staff, 2006) ja tuottavat näin ollen huomattavat määrät tuloja pelin kehittäjille ja palveluntarjoajan lisenssin haltijoille. Samoin suosittu peli voi olla tuottava myös yksityiselle, kilpailevaa palvelua ylläpitävälle taholle. Omaa ohjelmistoa kehittämällä välttyt tarpeelta vuokrata oletettavasti jokseenkin hintavaa palvelinohjelmiston lisenssiä pelin kehittäjältä, etkä samoin myöskään maksa osinkoa tuloista. Yleisin tapa tehdä tuloja yksityisellä pelipalvelimella tuntuisi olevan mikromaksujen kerääminen. Usein yksityisen palvelun ylläpito käyttää tässä yhteydessä terminologiaa joka enemmän viittaa lahjoituksiin. Kenties tällä pyritään peittämään toiminnan voittoa tavoitteleva luonne. Todellisuudessa kuitenkin tällöin kyseessä on perinteisimmin pelimaailmasta tuttu *mikrotransaktio*. Pientä maksua vastaan pelaaja saa pelin sisällä jotain haluamaansa, joten on aihetta sanoa että välttämättä kyse ei ole ihan perinteisestä lahjoituksesta. Todennäköisimmin transaktiot pyritään naamioimaan lahjoitukseksi juridisista syistä johtuen. Mikrotransaktioiden tai lahjoitusten lisäksi palvelun voi tuotteistaa myös keräämällä kuukausimaksua pelaajilta samoin kuin useat pelien lisensoidut palveluntarjoajat tekevät. Tällöin houkutellakseen pelaajia yksityisen palveluntarjoajan tulee tarjota jotain etua jotta palvelu olisi kilpailukykyinen pelin virallisiin palvelimiin verrattuna. Kilpailukykyä voi tavoitella esimerkiksi halvemmalla kuukausimaksulla, tai pelin sisäisillä eduilla, kuten nopeammalla progressiolla hahmon kehityksessä.

Välttämättä emulaattorin kehityksellä tai käytöllä ei kuitenkaan aina ole kyse voiton tavoittelusta. Pelaajien kannalta monipelit ovat ongelmallisia siinä mielessä että kun pelin kehittäjä päättää katkaista tuen pelille sen joko vanhennuttua uuden tuotteen myötä tai sen kannattamattomuuden vuoksi, jäävät pelin pelaajat vailla palvelinta millä pelata. Tällaisessa tilanteessa ainoa keino pelin elvyttämiseen voi olla lähteä tutkimaan pelin kommunikointiprotokollaa jotta voidaan toteuttaa epävirallinen palvelin. Esimerkiksi nk. ”*Return of Reckoning*” on yhteisön kehittämä sekä ylläpitämä palvelin ”*Warhammer Online: Age of Reckoning*” -pelille. Pelin virallinen palvelin toimi vuodesta 2008 lähtien vuoteen 2013 saakka. Kun virallisen palvelun sulkemisesta ilmoitettiin, yhteisön jäsenet alkoivat toteuttamaan omaa emulaattoria pitääkseen pelin elossa. Emulaattorin kehitys alkoi suljettuna, mutta julkaistiin sittemmin avoimena lähdekoodina (Return of Reckoning, n.d).

Uteliaisuus myös voi olla riittävä motivaatio lähteä kehittämään emulaattoria; teknisesti kykenevä pelaaja haluaa ehkä vain tutkia pelimaailmaa syvemmin vailla virallisen palvelimen asettamia rajoitteita. Omalla palvelimellaan pelaaja luonnollisesti itse määrää säännöt ja vastaa sisällön saatavuudesta. Uteliaisuus voi olla myös syy pelata epävirallisella palvelimella. Emuloidun palvelimen ominaisuudet voivat poiketa virallisesta monin tavoin. Esimerkiksi korkeimman tason saavuttaminen pelissä onnistuu ehkä huomattavasti nopeampaa, tai kenties virallinen palvelin vaatii kuukausimaksua ja jokin epävirallinen palvelin tarjoaakin jotain vastaavaa, mutta ilmaista palvelua. Syitä on monia, mutta usein yksityisillä palvelimilla pelataan jotta päästään kokeemaan jotain mikä ei virallisella ole käyttäjän tavoitettavissa.

World of Warcraft pelin tiimoilta oli huomattava myös yksi mielenkiintoinen ilmiö: osa pelin yhteisöstä halusi palata takaisin sellaiseen pelimaailmaan mitä pelin virallinen versio alkuaikoina tarjosi. Myöhemmin pelimaailma laajentui ja pelattavuus muuttui tavoin joka ei miellyttänyt kaikkia yhteisön jäseniä. Viralliset palvelimet eivät kuitenkaan mahdollistaneet paluuta entiseen, joten sittemmin kysynnän myötä alkoi syntyä yksityisiä palvelimia jotka käyttivät pelin vanhempaa asiakasohjelmaa ja pyrkivät matkimaan pelin alkuperäistä pelattavuutta mahdollisimman tarkkaan. Yksi tällainen palvelin oli *Nostalrius* joka vuosina 2015-2016 kerkesi kerätä jopa lähes miljoona rekisteröitynyttä käyttäjää ympärilleen (Johnson, 2019). Tiedettävästi kyseessä oli yksi suosituimmista yksityisistä palvelimista. Palvelimen toiminta lakkautettiin kun pelin kehittäjä Blizzard vaati palvelun sulkemista vedoten tekijänoikeuksiin. Palvelun sulkeminen aiheutti yhteisön keskuudessa laajalti närää, sillä useat pelaajat jäivät yhä toivomaan mahdollisuutta pelata peliä sen alkuperäisessä muodossa. Kenties tämä riitti vakuuttamaan siitä että kysyntää on riittävästi, sillä vuonna 2017 pelin kehittäjä Blizzard ilmoitti avaavansa virallisia palvelimia jotka tarjoaisivat kysyntää vastaavaa pelattavuutta. Vuonna 2019 palvelimet avattiin nimeä *World Of Warcraft: Classic* kantaen ja yhteisön keskuudessa se oli suuri menestys keräten avauksen yhteydessä satoja tuhansia pelaajia.

Akateeminen kiinnostus tulee huomioida yhtenä motivaationa. Tutkimalla ammattilaisten kehittämää peliä ja sen käytössä olevia teknologioita voidaan muodostaa hyvä käsitys alalla tarvittavasta tietoudesta. Kehittämällä omaa emulaattoria taas voidaan saada käytännön kokemusta näistä menetelmistä ja työkaluista. Massiivisissa moninpeleissä etenkin tulee huomioida oman ohjelmiston kehityksen hyöty oppimisessa niiden monimutkaisuuden vuoksi. Kyseisen kategorian peleissä palvelinten tulee nimittäin nimensä mukaisesti kyetä palvelemaan potentiaalisesti massiivista määrää samanaikaisia käyttäjiä. Tällaisen ohjelman kehittäminen ei ole täysin triviaalia ja vaatii kattavan teknisen tietotaidon kartuttamista. Emulaattoreiden ympärille kehittyneissä yhteisöissä käydäänkin aktiivista keskustelua myös teknisestä toteutuksesta, työkaluista sekä menetelmistä. Halua oppia vaaditut taidot selvästi löytyy olipa sen taustalla motivaatio mikä tahansa.



## 2.3 Verkkopelien emulaattoreiden historiaa

Vuonna 1997 julkaistu Ultima Online on yksi kaikkien aikojen tunnetuimpia sekä yksi ensimmäisistä massiivisista nettiroolipeleistä. Samalla se on tunnetusti myöskin ensimmäinen lajinsa peli jolle kehitettiin palvelin emulaattori. Ultima Offline eXperiment -nimisen emulaattorin kehitys alkoi jo pelin beta -vaiheen aikoihin ennen varsinaista julkaisua. Emulaattorin alkuperäinen tarkoitus oli mahdollistaa pelaajille paikallisen ”offline” serverin pyörittämisen testausmielessä. Emulaattorin aikaisemmat versiot olivat suljetun lähdekoodin projekteja, mutta myöhemmin kehitys siirtyi avoimen lähdekoodin piiriin GNU GPL lisenssin alaisena. Tätä seurasi kymmeniä muita emulaattoreita joista suurin osa on sittemmin lakkautettu. UOX sen sijaan avoimena jatkoi olemassaoloaan ja näki yhä kehitystä vuosia julkaisun jälkeenkin(UOX3, 2020).

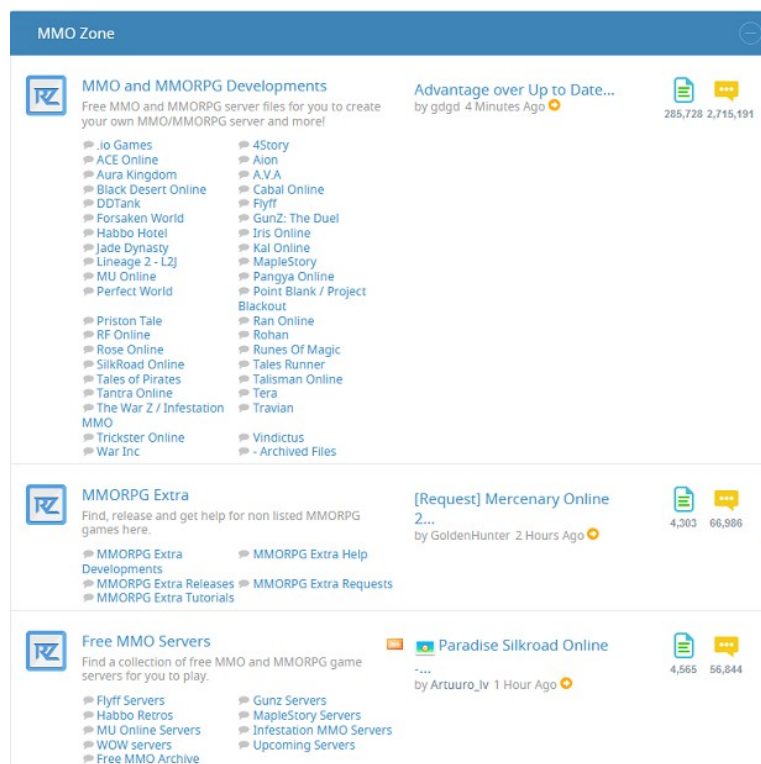
Lineage 2 on menestynyt Korealainen peli jatkona pelisarjan ensimmäiselle osalle. L2jserver on avoimen lähdekoodin emulaattori, jonka kehitys alkoi jo pelin beta -vaiheessa vuonna 2003. Sen ensimmäinen versio julkaistiin yleisön saataville vuonna 2004 pelin julkaisun jälkeen. Viimeisimmän tarkastelun myötä projekti on yhä aktiivisessa kehityksessä vuonna 2020 n. 16-vuotta alkuperäisen julkaisun jälkeen. Kehittäjiä ohjeistetaan välttämään pelin datan levittämistä palvelimen lähdekoodin yhteydessä. Syy tähän lie piilee kopiosuojatun materiaalin piratisoinnin välttämiseksi.

Jo aiemmin paperissa mainittu Blizzard:n kehittämä World of Warcraft saavutti suuren kansainvälisen peliyhteisön, mikä ilmenee myös pelille myönnetystä Guinness World Record maininnasta suosituimmasta MMORPG pelistä. Pelin saavuttama suursuosio johti pitkään elinkaareen ja pelille onkin sittemmin julkaistu useita lisäosia pelimaailmaa laajentamaan. Arvatusi suuren yhteisön ja suosion myötä tuli myös kysyntää emulaattorille. Pelille kehitettiinkin ajan saatossa useita emulaattoreita ja niistä tiedetysti ensimmäinen, nimeltään Stormcraft. Joidenkin lähteiden mukaan sen kehitys alkoi jo ennen kuin Blizzard virallisesti edes ilmoitti pelin julkaisusta. Jälleenmallinnustyö tehtiin aiemmin julkisuuteen vuotaneen asiakasohjelmiston kehitysversion pohjalta ja pian pelaajilla olikin jo mahdollista astua sisään pelimaailmaan tutkiakseen ympäristöä.

Joissain tapauksissa pelin kehittäjä tiedostaa yksityisten palvelinten olemassaolon, mutta sallii toiminnan jatkua joko ottamatta asiaan kantaa, tai virallisesti vahvistaen ettei ole aikeissa puuttua tilanteeseen, kuten kävi EverQuest -pelin tapauksessa. *Project 1999* on voittoa tavoittelematon emulaattori, jonka ylläpito ilmoittivat vuonna 2015 foorumeillaan tehneensä kirjallisen sopimuksen pelin keittäjän kanssa toiminnan jatkamisesta ilman seuraamuksia (Rogean, 2015). Kyseessä on harvinaislaatuinen tapaus ja on syytä olettaa että esimerkiksi pelin ikä(julkaistiin 1999) sekä palvelun voittoa tavoittelematon luonne vaikuttivat sopimuksen syntyyn.

## 2.4 MMORPG Palvelin emulaattorit ja yhteisö

Massiivisten verkkopelien emulointi on sen verran yleinen ilmiö että sen ympärille on kehittynyt yhteisöjä jakamaan tietoa, ohjelmakoodia, vuotaneita pelipalvelimien virallisia ohjelmistoja sekä mainostamaan yksityisiä palvelimia eri peleille. Yksi tällainen yhteisö tunnetaan nimellä *RaGEZONE* ja kooltaan kyseessä on kenties suurin englanninkielinen yhteisö, jonka keskustelufoorumit keskittyvät lähes yksinomaan yksityisten pelipalvelimien aihepiiriin. Foorumeilla jaetaan emulaattoreita, pelien asiakasohjelmia, tietoa pelien sekä protokollien toiminnallisuuksista ja muuta vastaavaa. Yhteisön keskustelufoorumeilla on jopa satoja tuhansia rekisteröityneitä käyttäjiä ja sivusto on toiminut internetissä vuodesta 2001 lähtien. Sivusto mainostaa foorumeillaan itseään sanoin ”MMO Development Community” – eli massiivisten moninpelien kehitys yhteisö (RaGEZONE, 2020).



**Kuva 1.** kuvakaappaus *RaGEZONE* yhteisön keskustelufoorumeilta 2020

RaGEZONE foorumeilta löytyy hyvin pienellä vaivalla emulaattoreita kymmenille eri peleille, ohjeita kuinka asentaa ja käyttää emulaattoreita, ohjeita kuinka jälleenmallintaa pelien protokollia kuin myös kattava lista entisistä sekä yhä toiminnassa olevista yksityisistä palvelimista. Sivuston käyttäjien määrästä sekä aktiivisuudesta itsestään voidaan jo päätellä että MMORPG palvelinten emulointi on kenties jopa yllättävän laajalle levinnyt ilmiö.

Myös muitankin vastaavia yhteisöjä löytyy niin englanniksi kuin myös muillakin kielillä. Esimerkiksi *elitepvpers* niminen yhteisö on myös kerännyt sivuilleen valtavan määrän käyttäjiä. Sisällöltään kyseinen sivusto ei kuitenkaan keskity pelkästään massiivisiin moninpelihin tai niiden emulaattoreihin, vaan keskusteluaiheet vaihtelevat kaikentyyppisten pelien välillä.

### 3 Protokollien jälleenmallintaminen muihin käyttötarkoituksiin

Palvelimen toiminnallisuuden matkiminen ei ole ainut syy lähteä jälleenmallintamaan verkkopelin kommunikaatiota. Toinen yleinen tunnistamani syy tutkia asiakkaan sekä palvelimen välillä liikkuvia viestejä on peleissä huijaaminen. Asiasta löytyy hieman aiempaakin kirjallisuutta. Esimerkiksi Steven Webb ja Sieteng Soh (2007) kategorioivat protokolla tasolla toteutettuja huijausmenetelmiä. Huijaamisella tavoitellaan joko epäreilua etua muihin pelaajiin nähden, tai mahdollisesti myös rahallista hyötyä. Varsinkin em. massiivisissa moninpeleissä on usein toteutettu jonkinlainen pelin sisäinen mekanismi käydä kauppaa pelaajien välillä. Harvinaisuudet ovat haluttua tavaraa ja niistä ollaan usein valmiita maksamaan myös oikeaa rahaa. Näinpä jos huijaamalla onnistutaan haalimaan pelin sisäisiä esineitä, voidaan niitä kauppaamalla tavoitella taloudellista hyötyä.

Protokollan ymmärtäminen helpottaa mahdollisten tietoturvaongelmien tunnistamista ja hyödyntämistä pelissä. Pelin asiakasohjelma toteuttaa viestit alkuperäisten määritelmien mukaisesti ja pidättäytyy asetetuissa raja-arvoissa sekä viestittelee oletetussa järjestyksessä. Jos mahdollisia poikkeuksia ei ole huomioitu, tämä voi aiheuttaa palvelimen puolella määrittelemättömiä seurauksia ja pahimmassa tapauksessa paljastaa haavoittuvuuksia. Yksi esimerkki melko yleisestä haavoittuvuudesta on aritmetiikkaan liittyvä yli-/alivuoto missä kokonaisluku pyörähtää ympäri pieninpään tai suurimpaan mahdolliseen lukuun kun lukuavaruuden rajat ylitetään. Otetaan esimerkkinä kirjaamaton kokonaisluku, jonka pienin mahdollinen arvo on nolla. Jos nolasta vähennetään yksi, ollaankin yhtäkkiä suurimmassa mahdollisessa positiivisessa arvossa (HackersOnBoard, 2019, 04:30-06:00). Kuvitellaan vaikka että pelissä on virtuaali valuutta jolle on varattu kokonaisluku. Pelissä kauppaa käytäessä asiakas lähettää palvelimelle tiedon paljonko rahaa tulee vähentää. Normaalisti asiakas tarkistaa että jäljelle jäävä summa pysyy yli nollan, eikä todennäköisesti edes lähetä palvelimelle viestiä jos se alitetaan. Sen sijaan pelaajalle annetaan ilmoitus että rahat eivät riitä kaupan tekoon. Jos palvelin luottaa tällaisessa tilanteessa asiakkaan päädyn varmistukseen, on hyvin mahdollista että viestin lähettäminen palvelimelle vähentää kerrotun summan ilman tarkistuksia ja näin pelaaja on luonut itselleen suurimman mahdollisen summan pelin virtuaali valuuttaa. Joitain vuosia sitten julkisuuteen tuli haavoittuvuuksia metsästävä henkilö nimimerkillä ”Manfred”. Hän on pitänyt tietoturvakonferensseissa useampia puheita taustoistaan pelien parissa ja siitä kuinka hän useiden vuosien ajan elätti itseään hakeroimalla massiivisia moninpelejä (Franceschi-Bicchierai, 2017). Omien sanojensa mukaan hän pyrki pitämään matalaa profiilia ja toimimaan eettisesti niin ettei pilaisi pelikokemusta muille pelaajille. Samalla hän kuitenkin aktiivisesti etsi ja hyödynsi haavoittuvuuksia useissa peleissä saadakseen pelissä valuuttaa sekä harvinaisia esineitä myyntiä varten. Ottaen huomioon että joissain tapauksissa virtuaalisten esineiden arvo huutokaupassa voi nousta tuhansiin, tai jopa kymmeneen tuhansiin dollareihin, on helppo ymmärtää kuinka houkutteleva mahdollisuus näissä peleissä huijaaminen on.

### 3.1 Asiakkaan toimintojen automatisointi

Protokollan jälleenmallintamista voi hyödyntää myös päinvastoin toteuttamaan asiakasohjelman toiminnallisuuden. Yleisin syy tällaiseen on pelin toimintojen automatisointi ja usean rinnakkaisen asiakkaan samanaikainen suoritus. Roolipeleissä hahmon kehitys, sekä resurssien kerääminen on usein työläs ja aikaa vievä prosessi. Niinpä etua tavoitellaan varsinaisen huijaamisen lisäksi myös pelin sisäisten toimintojen automatisoinnilla. Jos voidaan poistaa kuviosta pelin varsinainen asiakasohjelma, päästään samalla eroon kaikista mahdollisista sisäänrakennetuista huijauksen estoon tarkoitetuista menetelmistä, sekä poistetaan tarve raskaan graafisen ympäristön piirtämiselle. Ilman graafista käyttöliittymää on suorituskyvyn suhteen helpompi ajaa useampaa asiakasta rinnakkain yhdellä laitteella. Myös automatisoinnilla voidaan tavoitella taloudellista hyötyä jos resurssien keräys pelissä suuressa määrin on tuottavaa.

Useimmille kokeneille pelaajille ”kulta farmaus” on käsitteenä tuttu ilmiö. Tällä tarkoitetaan pelin sisäisen virtuaali valuutan haalimista myynti tarkoituksessa. Sitä voidaan suorittaa joko automatisoidusti, tai ihan perinteisellä työvoimalla. Tarkkoja tilastoja on vaikea kerätä, mutta joidenkin arvioiden mukaan ala työllistää satoja tuhansia ja ylivoimainen enemmistö näistä niin sanotuista kulta farmaajista sijaitsevat Kiinassa (Heeks, 2010). Julkisuuteen on tullut useampiakin Kiinassa sijaitsevia yrityksiä joiden työntekijät omistavat koko työpäivän suosittujen moninpelien pelaamiseen ja resurssien haalimiseen (Heeks, 2010). Halpa työvoima tässä tapauksessa tarkoittaa että tarve kehittää automatisointia ja etsiä haavoittuvuuksia palvelimen puolelta poistuu käytännössä kokonaan.

Asiakkaan toimintojen automatisointi voi tapahtua myös alkuperäisen asiakkaan välityksellä, eikä välttämättä tuolloin vaadi protokollan toteuttamista. Esimerkkinä tästä on vuosina 2005-2009 toiminut ”Glider” niminen ohjelmisto, joka automatisoi suosittun World Of Warcraft pelin toiminnallisuuksia niin että pelaaja pystyi kasvattamaan tasoaan pelissä tai keräämään resursseja ilman aktiivista osallistumista. Glider oli kaupallinen ohjelmisto ja sitä myytiin arviolta noin sata tuhatta kappaletta hintaan \$25 käyttäjää kohti. Pelin kehittäjä Blizzard katsoi että ohjelmisto loukkasi heidän käyttöehtoja ja haastoi Gliderin kehittäjän oikeuteen. Oikeus määräsi Michael Donnelyn maksamaan korvauksia Blizzardille immateriaalioikeuksien loukkauksesta, koska Glider kiersi pelin huijauksen esto järjestelmän. Ohjelmiston kehitys ja myynti tuli päätökseen 2009 tosin jutun käsittely oikeudessa jatkui vielä tämän jälkeenkin. Puolustuksen mukaan tekijänoikeuksia ei loukattu, sillä Gliderin käyttäjillä oli jo ennestään kopio pelistä ja Glider oli siitä täysin erillinen ohjelma. Oikeuden lopullisen päätöksen perusteella kuitenkin loukkaus tapahtui kun ohjelma kiersi pelin turvajärjestelmän mahdollistaen pääsyn myös pelin dynaamiseen sisältöön, joka ei ollut valmiiksi käyttäjän koneella asennettuna (Court Listener, 2010).

## 4 Emulointiin liittyviä mahdollisia juridisia ongelmia

Kuten aikaisemmin jo todettiin - varsinkin MMORPG peleissä liikkuu suuret määrät rahaa palvelun tarjoamisen ympärillä, ja kun rahasta on kyse, on kyse myös yrityksen intresseistä. Luonnollisesti yritykset haluavat varjella omia intressejään ja varmistaa että heidän kehittämälleen pelille ei ole kilpailevaa palvelun tarjoajaa. Jos tällaista kilpailua pääsee kuitenkin syntymään, on hyvin mahdollista että pelin immateriaalioikeudet omistava yritys pyrkii oikeusteitse eliminoimaan kilpailun. Tässä paperissa tuon myös esille esimerkkejä tapauksista jotka käsiteltiin oikeudessa. Ennen pitkää ja hintavaa oikeusprosessia tosin on mahdollista että yritys lähestyy yksityistä palvelinta pyörittävää tahoa ja pyytää palvelun sulkemista mahdollisen haasteen uhalla. Monissa tapauksissa se onkin riittävää, sillä pelko mahdollisesta oikeustaistosta ja sen häviämisen tuottamista sanktioista on tehokas väline. Esimerkiksi aiemmin mainitsemani *Nostalrius* palvelin koki juuri tämän kohtalon. Käytännössä on hyvin vaikea selvittää kuinka yleistä on että yksityisen palvelimen ylläpitoon ollaan yhteydessä ja palvelun sulkemista vaaditaan. Ilmiön laajuuteen verrattuna on melko harvinaista nähdä yksityisen palvelimen sulkemisen ylittävän uutisointi kynnyksen. Toisekseen palvelun sulkeuduttua syitä ei välttämättä aina edes tiedoteta pelaajien yhteisölle, joten aina ei ole selvää miksi palvelu sulkeutui.

Mitä enemmän käyttäjiä epävirallinen palvelu kerää, sitä todennäköisempää on että pelin kehittäjä kiinnostuu heidän toiminnastaan. Täysin ilmaiset ja voittoa tavoittelemattomat palvelimet eivät välttämättä ole kalliin oikeusprosessin arvoisia varsinkaan jos palvelua ylläpidetään toisesta maasta käsin. Toisekseen ei ole varmuutta pystytäänkö oikeudessa osoittamaan kyseessä olevan epäreilu kilpailu jos toiminta on voittoa tavoittelematonta. Tämä ei silti välttämättä tarkoita etteikö sellaista palveluakin vastaan voitaisi kerätä riittävää näyttöä haastetta varten. Vaikka emulaattorin lähdekoodi olisi täysin alkuperäistä eikä sellaisenaan riko mitään kopio-oikeuksia, on sen silti toteutettava pelin protokolla jotta yhteensopivuus asiakasohjelmiston kanssa voidaan saavuttaa. On myös mahdollista että yksityistä palvelinta pyöritetään julkisuuteen vuotaneella alkuperäisellä palvelin ohjelmistolla. Toki tuolloin ei enään ole kyse emulaattorista, enkä siihen lähde syventymään sen enempää, mutta mainitsemisen arvoinen huomio kuitenkin että on olemassa tapauksia missä alkuperäisen virallisen palvelimen ohjelmisto on päätynyt vapaaseen levitykseen. Emulaattoriin verrattuna siinä tapauksessa lie helpompi osoittaa että tekijänoikeuksia on loukattu ja ohjelmasta on tehty laitton kopio.

## 4.1 Asiakasohjelman käyttö ja manipulointi

Suurimmat, tai ainakin lainsäädännön kannalta selvimmät rikkeet vaikuttavat liittyvän asiakasohjelmiston muokkaamiseen ja jälleenlevitykseen, sekä kaupallisen palvelun ylläpitämiseen. Näitä ongelmia voi pyrkiä välttämään olemalla puuttumatta pelaajien käyttämään asiakasohjelmistoon ja tarjoamalla voittoa tavoittelematonta palvelua. Raskasta kuormaa kestävän palvelimen ylläpito ei kuitenkaan ole maksutonta ja toisaalta toisinaan toiminnalla halutaan myöskin kerätä tuottoa ja tästä johtuen yksityisten pelipalvelimien kesken on yleistynyt ilmiö missä ylläpito ottaa vastaan lahjoituksia ja niitä vastaan tarjoaa pelaajille esimerkiksi pelin sisäisiä esineitä. Näin voidaan yhä tehdä palvelulla voittoa ilman että kyseessä olisi teknisesti maksullinen/voittoa tavoitteleva palvelu. Se kestääkö tällainen lahjoituksiin pohjautuva voiton tavoittelu päivänvaloa oikeudessa onkin toki asia erikseen. Oleellista lie voidaanko lahjoitukset suoraan liittää tekijänoikeuksia loukkaavaan toimintaan.

Jos palvelimen toiminnallisuuden jäljittelyllä tavoitellaan palvelun tarjontaa, on pelin asiakasohjelma tuolloin väistämättä osa yhtälöä viimeistään loppukäyttäjän näkökulmasta. Pelin kehittäjän tai julkaisijan toimesta asiakasohjelma on tyypillisesti asetettu ottamaan yhteyttä tiettyihin ennalta määrättyihin julkaisijan hallussa oleviin palvelimiin. Jos kolmas osapuoli haluaa tarjota omaa palvelua ja tukea pelin asiakasohjelmalle, täytyy asiakkaan liikenne tuolloin pystyä ohjaamaan muihin palvelimiin. Liikenteen ohjaaminen ei välttämättä vaadi asiakasohjelmiston manipulointia, vaan se voidaan mahdollisesti toteuttaa myös verkkokerroksessa. Esimerkiksi domain-nimien uudelleenohjaus eri osoitteisiin on mahdollista toteuttaa ilman asiakasohjelman muutoksia. Loppukäyttäjän ja käytettävyyden kannalta kenties helpompaa olisi jos käyttäjän ei tarvitse tuntea reititykseen liittyviä käsitteitä ja menetelmiä. Tuolloin joudutaan todennäköisesti turvautumaan asiakasohjelman manipulointiin joko pysyvästi tai suorituksen ajaksi. Asiakasohjelmistot ovat kopiosuojattuja ja niiden jakelu sekä muuttaminen kolmannen osapuolen toimesta loukkaa mahdollisesti tekijänoikeuksia. Joissain tapauksissa pelin käyttöehdoissa suoraan kielletään pelin käyttö muilla kuin virallisilla palvelimilla. Samoin käyttöehdot voivat kieltää myös ohjelman muuttamisen esim. huijaus tarkoituksiin. Edellä jo mainitsinkin protokollien jälleenmallintamisen sellaisissa käyttötapauksissa kun tavoitteena on pelissä huijaaminen. Huijaaminen tosin voi vaatia sitä estämään luotujen toimenpiteiden kiertämistä, joten ei ole myöskään poikkeuksellista manipuloida pelin asiakasohjelmistoa myös siihen tarkoitukseen.

Verkkoliikenteen ohjaaminen yksityiselle palvelimelle, turvamenetelmien kierto saati pelissä huijaaminen ei ole ainut syy lähteä mukauttamaan asiakas ohjelmaa. Joissain tapauksissa yksityiset palvelimet tuovat peliin uutta sisältöä erottuakseen joko muista epävirallisista, tai virallisista palvelimista. Jotta nämä muutokset saataisiin loppukäyttäjän haltuun, pitää palvelun jakaa jossain ratkaisua joka muuttaa asiakasohjelman sisältöä, tai vaihtoehtoisesti suoraan tarjota muunneltua kopiota pelistä. Tämä altistaa jakajan selkeämmin mahdollisiin kiistoihin tekijänoikeuksien rikkomisesta sekä laittomien kopioiden tuottamisesta.

Siinä missä palvelimen puolella on mahdollista toteuttaa asiat eri tavoin kuin mitä alkuperäisessä ohjelmistossa, suurin osa itse loppukäyttäjän visuaalisesta kokemuksesta on ennalta määritelty pelin asiakasohjelmassa. Palvelin ei lähtökohtaisesti pysty puuttumaan siihen millainen pelimaailma käyttäjälle näytetään, miltä pelin hahmot näyttävät tai millaisia esineitä pelissä on mahdollista saada haltuun. Toistaiseksi yleisin keino vaikuttaa niihin on tuoda peliin päivityksiä, jotka muuttavat asiakasohjelman sisältöä ja toiminnallisuutta. Dynaaminen vaikuttaminen visuaaliseen sisältöön palvelimen toimesta vaatisi protokollalta tuen siirtää tietoa malleista, tekstuureista, äänistä ym. Teknisesti se on mahdollista ja paikoitellen myös jossain muodossa käytössä, mutta toteutuksen kannalta huomattavasti yksinkertaisempaa on tarjota pelissä lähinnä staattista sisältöä. Samalla se sisältö voidaan tuoda kopiosuojausten alaiseksi paketoimalla sen osaksi pelin asiakasohjelmistoa.

Vaikkakin pelin asiakasohjelman käyttöehtoihin saattaa sisältyä pykälä ohjelman käyttämisestä epävirallisilla palvelimilla, ei yksityisen palvelimen ylläpitäjän välttämättä tarvitse siitä välittää. Palveluntarjoajan ei nimittäin välttämättä tarvitse ottaa suoraan kantaa käytössä olevaan asiakasohjelmaan saati tarjota sille mitään mukautuksia. Tällöin vastuu siirtyy enemmän loppukäyttäjän puolelle, joka ohjelmaa asentaessa on käyttöehdot todennäköisesti hyväksynyt.

Jos palvelin emulaattori ei sisällä mitään asiakasohjelmasta peräisin olevaa dataa, vaan keskittyy pelkästään tarjoamaan yhteensopivuutta toteuttamalla protokollan mitä peli ymmärtää, voi tuolloin olla vaikea osoittaa että tekijänoikeuksia on loukattu. Tämä pätee varsinkin mikäli pelin kopion hankkiminen sekä mahdollisten muutosten tekeminen jätetään täysin loppukäyttäjän vastuulle. On kuitenkin hyvin tyypillistä yksityisille palvelimille tarjota käyttäjille jokin työkalu, tai suoraan muokattu kopio pelistä jotta sen käyttö olisi helpompaa. Esimerkiksi pelin käynnistämiseen saatetaan toteuttaa kolmannen osapuolen työkalu joka mahdollisesti vaikuttaa ajonaikaisesti vaikkapa siihen mihin palvelimeen peli ottaa yhteyttä. Tähän on olemassa eri menetelmiä jotka joko muuttavat muistin sisältöä tai jopa suoraan suoritettavaa ohjelmakoodia. Yksi laajalti käytetty menetelmä vaikuttaa ohjelman toiminnallisuuteen ajon aikana on niin sanottu "dll injektio" jota hyödyntämällä päästään käsiksi ohjelman muistiavaruuteen ja siihen pystytään vaikuttamaan ilman että ohjelmasta jouduttaisiin tuottamaan pysyvästi mukautettua kopiota(Alsari et al, 2012).

## 4.2 Immateriaalioikeuksien merkitys yritysten näkökulmasta

Kommunikointiprotokollista puhuttaessa on kyse immateriaalioikeuksista. Protokolla itsessään ei ole mikään konkreettinen tuote, vaan menetelmä. Periaatteessa kyseessä ei myöskään ole yrityksen varsinainen kaupallinen tuote. Pelejä kehittävän ja myyvän yrityksen tapauksessa varsinainen kaupallinen tuote on joko pelin asiakasohjelma, palvelinohjelmisto tai molemmat. Tästä huolimatta on nähtävissä että yritykset ovat toisinaan valmiita jopa oikeusteitse varmistamaan yksinoikeudet palvelun pyörittämiseen.

Miksi siis yritys, jonka tärkein myyntituote ei ole protokolla itse, vaan sitä hyödyntävä ohjelmisto, olisi kiinnostunut varjelemaan protokollan määritelmää yrityksen sisäisenä salaisuutena? Oletettavasti suurin vaikuttava tekijä on kilpailun eliminointi. Huolimatta siitä että varsinainen tuote on jotain muuta, on merkittävä osa massiivisia moninpelejä kehittävän yrityksen tuloista riippuvainen siitä että he ovat ainoa taho jolla on kyky tarjota palvelua asiakasohjelmistolle, tai vuokrata palvelinohjelmiston lisenssiä kolmannen osapuolen palveluntarjoajille.

Vaikeuttaakseen niin liikenteen manipulointia kuin myös tutkimistakin, peleille on melko yleistä että kommunikointiin käytettävä protokolla on salattu tai obfuskoitu jollain menetelmällä. Myös pelin asiakasohjelmistoon voidaan soveltaa salausmenetelmiä jotta sen tutkiminen vaikeutuisi (Joshi, 2008). Ei ole myöskään ennenkuulumatonta muuttamaan protokollan rakennetta ja/tai salausmenetelmää mitätöidäkseen ulkopuolisten tahojen mahdollisen aiemman jälleenmallinnustyön tai pakottaakseen niitä pysymään vanhentuneessa asiakasohjelmiston versiossa. Protokollan määritelmän vuotaminen tai tarkoituksellinen julkaisu avoimesti tarkoittaa mahdollisesti yksinoikeusaseman menettämistä palveluntarjoajana. Toki tässä on syytä huomioida että kommunikointimenetelmä yksin ei vielä takaa kilpailukykyä, vaan tämän lisäksi ulkopuolisen tahon tulisi myös kehittää oma ohjelmisto joka toteuttaa kyseistä protokollaa. Vielä sekään ei takaa tasavertaista kilpailukykyä alkuperäisen pelin julkaisijan kanssa, sillä asiakasohjelmisto on yhä pelitalon omaisuutta.

Sen lisäksi että asiakkaan ja palvelimen välistä kommunikointia pyritään salaamaan ja säilyttämään suljettuna, on hyvin yleistä että myöskin asiakasohjelmiston käyttöehdoissa rajoitetaan ohjelman käyttöä vain virallisesti tunnustetuissa palveluissa. Toisinaan käyttöehtoihin liittyy myös jossain muodossa eksplisiittinen kielto tai rajoitus yrittää jälleenmallintaa tai manipuloida ohjelmistoa ja/tai sen käyttämää kommunikointiprotokollaa.

Vaikka on syytä olettaa että pelejä tai pelipalveluita tuottavan yrityksen perimmäiset intressit salata kommunikointi asiakkaan ja palvelimen välillä ovat puhtaasti taloudelliset, ei se siltikään suinkaan ole ainut mahdollinen selittävä tekijä. Yhteisön, eli maksavien asiakkaiden tyytyväisyys vaikuttaa suoraan myyntituloihin. Tyytyväisyys laskee sen myötä jos pelissä on yksittäisiä henkilöitä jotka saavuttavat epäreilun edun huijaamalla tavalla tai toisella. Tietoturva on näinollen epäilemättä myös merkittävä tekijä protokollan suljettuna pitämisessä. Kuten aimmin jo paperissa todettiin; haavoittuvuuksien löytäminen peleissä voi olla tuottavaa kaupankäyntiä. Huijaamalla voidaan kuitenkin mahdollisesti myös pilata pelin sisäinen ekonomia, joka puolestaan aiheuttaa tyytymättömyyttä muiden pelaajien kesken.



### 4.3 Esimerkkitapauksia palvelin emulointiin liittyvistä oikeustaistoista suomessa ja muualla maailmalla

Kuten aiemmin jo paperissani mainitsinkin, peliyhtiöiden liiketoiminnan kannalta immateriaalioikeuksien merkitys on suuri. Sen myötä on selvää että yritykset haluavat suojella tekijänoikeuksiaan ja joissain tapauksissa pyrkivät aktiivisesti estämään yksityisten palvelinten toimintaa. Havainnollistaakseni käytännössä emuloinnin juridisia ongelmia, käyn läpi joitain ennakkotapauksia mitä on käsitelty maailmalla oikeudessa. Pyrin samalla hieman tarkastelemaan lopputulokseen mahdollisesti vaikuttaneita tekijöitä.

Vuodesta 2000 lähtien Suomessa toiminut *Sulake* julkaisi suosituksen sosiaalisen nettipelin nimellä ”Habbo Hotel” vuonna 2000. Peli on nuorille suunnattu chat-peli ja nykyään saatavilla useilla eri kielillä suomen lisäksi. Yhteensä palvelulla on miljoonia käyttäjiä ympäri maailmaa. Vuodesta 2006 lähtien pelille on kehitetty ja myyty emulaattoria yksityisten palveluiden pyörittämiseen. Sulakkeen julkaisemassa tiedotteessa kyseisen emulaattorin kehittäjä oli ehtinyt myydä vuosien saatossa tuhansia lisenssejä kehittämänsä ohjelmistoon. Tämän lisäksi hänellä oli oma yhteisö joka keskittyi jakamaan tietoa kuinka pitää yllä omaa yksityistä palvelinta ja kuinka käyttää emulaattoria.

Sulake syytti emulaattorin kehittäjää – Aaron Marshallia yrityksen immateriaalioikeuksien loukkaamisesta sekä luvattomasta asiakasohjelmiston käytöstä. Näillä puitten yritys myös jätti kanteen helsingin tuomioistuimeen. Marshall ja Sulake saivat sittemmin sovittua jutun ja Marshall suostui lopettamaan emulaattorin kehityksen sekä poistamaan siihen liittyneet ylläpitämänsä palvelut (Sulake, 2013). Marshall joutui myös maksamaan Sulakkeelle salassa pidetyn summan korvauksia aiheutuneista vahingoista.

Suomen ulkopuolelta tapauksia löytyykin useampia. Esimerkki hieman suuremmasta tapauksesta oli oikeusjuttu *Blizzard vs Alyson Reeves - Scapegaming*(2009-2010). Juttu ratkaistiin lopulta *Blizzardin* eduksi ja syytetty määrättiin maksamaan korvauksia jopa \$88 miljoonaa (Justia, 2010). Syytteen mukaan *Scapegaming* pyöritti luvattomia *World of Warcraft* -pelin palvelimia ja keräsi käyttäjiltään maksuja. Syytteen mukaan *Scapegaming* ehti tehdä noin \$3 miljoonaa tuottoa pyörittämällä palvelimella. Tämä ei ollut ainoa tapaus missä *Blizzard* on ottanut kantaa vastaaviin palveluihin, mutta tämä yksittäinen tapaus demonstroi hyvin kuinka suurista rahoista on kyse isoille pelitaloille ja kuinka paljon tuottoa myös emulaattoreilla voidaan tehdä. Uhka mahdollisesta oikeustaistosta on usein riittävä pelote yksityisten palvelinten ylläpitäjille ja monissa tapauksissa jatkotoimenpiteitä ei vaadita.

Vuonna 2008 ”*OdinMS*” nimellä tunnettu yksityinen *MapleStory* palvelin ilmoitti sivuillaan että *Nexon* on viemässä heitä oikeuteen. Vuonna 2011 *OdinMS* palvelun ylläpitäjä määrättiin maksamaan \$750,000 korvauksia tekijänoikeuksien rikkomisesta(IPPractice, n.d). Toinen vastaava tapaus jossa kyseinen peli, sekä sen julkaisija *Nexon* oli mukana päättyi myös heidän eduksi. *UMaple* pyöritti yksityistä palvelinta ja keräsi lähes 20 tuhatta käyttäjää. Palvelun ylläpitäjä *Gurvinder Kumar* määrättiin korvaamaan *Nexonille* \$3.6 miljoonaa korvauksia *DMCA* pykälän nojalla(Casetext, 2012). Tapauksesta mielenkiintoisen tekee se että oikeudessa *Nexon* ei kyennyt osoittamaan riittäväällä näytöllä että *UMaple* ylläpito olisi tehnyt tuloja suoraan pelin palvelun pyörittämisellä. Tästä huolimatta oikeuden päätös lopulta totesi että *UMaple* rikkoi tekijänoikeuksia levittäessään ohjelmistoa jonka tarkoitus oli kiertää pelin asiakasohjelmiston suojauksia.

Vuonna 2009 Joymax LTD haastoi oikeuteen ECSRO nimellä pyörineen Silkroad Online pelipalvelimen omistajan Grzegorz Grycuk:n. Silkroad on vuonna 2006 alunperin Etelä-Koreassa julkaistu peli. Joymax vaati alunperin korvauksena \$500000 vedoten menetettyihin tuloihin sekä yksityisen palvelimen epäreilusti ansaittuihin tuloihin. Vuonna 2011 oikeuden lopullisen päätöksen perusteella Grzegorz määrättiin maksamaan korvauksia \$75000 sekä pysyvästi sulkemaan palvelu(Court Listener, 2009).

Alla olevassa taulukossa on yhteenveto oikeudessa käsitellyistä tapauksista. Ainakin tämän otannan perusteella voidaan jo todeta että Yhdysvalloissa DMCA lainsäädäntöä on mahdollista tulkita niin että yksityisten palvelimien pyörittäminen ei kaikissa tapauksissa ole sallittua. Joukkoon mahtui myös mukaan Kanadan tuomioistuimessa käsitelty tapaus, joten ei myöskään yksistään DMCA johda tähän päätökseen, mutta DMCA rajoittaa jälleenmallintamista melko tarkkaan rajattuihin tapauksiin (Johnson, 2019).

Vuosi	Kuvaus	Maa	Päätös	Lainsäädäntö
2009	Blizzard vs Scapegaming	Yhdysvallat	Blizzard voitti. Häviäjälle jopa \$88m korvausvaatimukset	DMCA, 17 U.S.C. § 1201
2009	Joymax vs ECSRO	Yhdysvallat	Joymax voitti. Häviäjälle maksettavaksi \$75k korvaukset	28 U.S.C § 1338
2011	Nexon vs OdinMS	Kanada	Nexon voitti. Häviäjälle maksettavaksi \$750k korvaukset	Canadian Copyright Act
2012	Nexon vs UMaple	Yhdysvallat	Nexon voitti. Häviäjälle \$3.6m korvausvaatimukset	DMCA, 17 U.S.C. § 1201

**Taulu 1.** Oikeudessa käsitellyjä tapauksia

Kuten tästä voidaan päätellä, oikeuteen asti päätyvät tapaukset voivat tulla kalliiksi yksityisen palvelimen ylläpidolle. En ole tietoinen tapauksista missä oikeus olisi päättänyt tapauksen haastetun eduksi. Kenties tämä kertoo siitä että oikeuteen asti etenevät vain sellaiset tapaukset missä näyttö tekijänoikeuksien rikkomuksesta on erittäin vahvaa, tai vaihtoehtoisesti kenties se on merkki siitä että lainsäädäntö esimerkiksi Yhdysvalloissa ei ole erityisen suotuista palvelin emulaattoreiden käyttöön. Eurooppalaisissa tuomioistuimissa käsitellyjä tapauksia en tämän paperin tiimoilta onnistunut löytämään. On hyvin mahdollista että euroopan lainsäädäntö tekijänoikeuksien suhteen on hieman sallivampi esimerkiksi jälleenmallintamisen suhteen kuin mitä DMCA.

## Yhteenveto

Verkkopelaaminen yleistyy yhä ennestään olipa kyseessä konsoli, pc tai mobiili alusta. Nopean internet yhteyden saatavuus laajenee vuosi vuodelta ja kohdeyleisö peleille sen myötä. Tässä paperissa toin aiemmin esille että verkkopelien palvelinpuolen emulointi on ilmiönä lähes yhtä vanha kuin pelit itse. Protokollien sekä ohjelmien analysointiin ja jälleenmallintamiseen on kehitetty tehokkaita työkaluja, jotka ovat laajalti saatavilla asiasta kiinnostuneille. Internetistä löytyy ohjeistusta kuten myös yhteisöjä tukemaan emulaattoreiden kehitystä. Kysyntääkin yksityisille palvelimille tuntuisi löytyvän varsinkin sen myötä kun pelien viralliset palvelut aikanaan suljetaan.

Ilmiön laajuudesta huolimatta on selvää että täysin ongelmatonta palvelimen emulointi ei kuitenkaan ole. Tästä merkinä meillä on useita oikeudessa käsiteltyjä kiistoja tekijänoikeuksista kuten toin paperissani esille. Jälleenmallintaminen itsessään ei välttämättä ole rike riippuen paikallisesta lainsäädännöstä, eikä sen pohjalta kehitettyä palvelinohjelmistoa tule verrata alkuperäiseen, vaan kyseessä on erillinen teos. Palvelimella pelaaminen vaatii kuitenkin myös asiakasohjelman käyttöä, joka puolestaan on tekijänoikeuksien alainen teos. Sen myötä ei ole yksiselitteisen selvää etteikö jälleenmallintamisen pohjalta toteutettu emulaattori voisi loukata pelin kehittäjän immateriaalioikeuksia. Tilannetta voi ennestään mutkistaa sekin jos pelin viralliset palvelimet ovat yhä toiminnassa. Tällöin yksityiset palvelimet kilpailevat tekijänoikeuksien haltijaa vastaan aiheuttaen mahdollisesti taloudellista vahinkoa. Kuitenkin toin esille että on olemassa myös poikkeuksia, missä emulaattorin käyttö on sallittu pelin kehittäjien toimesta. On myös avoin kysymys siitä mikä on sallittua jos viralliset palvelimet ovat suljettu, tai jos pelin kehittänyt yritys lakkauttaa toimintansa. Jatkossa olisi mielenkiintoista myös kuulla pelialan toimijoilta miten heidän yrityksissä suhtaudutaan emulaattoreihin ja yksityisiin palvelimiin.

Tämän paperi oli katsaus aiempiin tapauksiin pohjustaen samalla myös mistä palvelimen emuloinnissa oikein onkaan kyse. En tämän tiimoilta pysty tarjoamaan vastauksia kiperiin kysymyksiin yksityisten palvelimien laillisuudesta, vaan tyydyn demonstroimaan että siihen ylipäänsä liittyy ongelmia. Pidän hyvin todennäköisenä että tämä ilmiö tulee jatkamaan eloa siinä missä verkkopelaaminenkin, joten olisi jatkon kannalta mielenkiintoista tutkia ongelmaa lisää varsinkin lainsäädännön näkökulmasta. Nähtäväksi jää huomioidaanko tulevaisuudessa lainsäädännön keinoin selvemmällä linjavedolla kommunikointiprotokollien jälleenmallintamisen laillisuudesta. Kenties lisää vastauksia saadaan sen myötä kun pelihistorioitsijat ympäri maailmaa pyrkivät taltioimaan vanhoja teoksia. Luonnollisesti ajan myötä yhä useampi vanha verkkopeli lakkauttaa toimintansa, eikä peliä tuolloin ole mahdollista pelata vaikka asiakasohjelma olisi yhä saatavilla.

## Lähteet

- Court Listener (2010). MDY INDUSTRIES, LLC v. Blizzard Entertainment.  
<https://www.courtlistener.com/opinion/181055/mdy-industries-llc-v-blizzard-entertainment/>
- Court Listener (2009). Joymax Co LTD v. Grzegorz Grycuk.  
<https://www.courtlistener.com/docket/4145342/joymax-co-ltd-v-grzegorz-grycuk/>
- Justia (2009). Blizzard Entertainment, Inc. v. Alyson Reeves et al.  
<https://dockets.justia.com/docket/california/cacdce/2:2009cv07621/456953>
- IPPractice (n.d). Nexon corporation v. Aleks Manov et al.  
<https://www.ippractice.ca/file-browser/?fileno=T-19-10>
- Casetext (2012). Nexon America Inc. v. Kumar. <https://casetext.com/case/nexon-america-inc-v-kumar>
- RaGEZONE – MMO Development community (2020). <http://forum.ragezone.com/>
- Return of Reckoning (n.d). <https://www.returnofreckoning.com/about>
- Johnson S (2019). Threats to Copyrighted Code: Bots, Mods and Reverse Engineering, 3Bus. Entrepreneurship & Tax L. Rev. 175.
- Thomas Debeauvais, Bonnie Nardi (2010). A qualitative study of Ragnarök Online Private servers: In-game sociological issues.
- Finlex (2018). Liikesalaisuuslaki. <https://www.finlex.fi/fi/laki/alkup/2018/20180595>
- J. Staff (2006). Guild Wars hits 2 million mark. <https://www.engadget.com/2006-06-19-guild-wars-hits-2-million-mark.html>
- Duchene J, Guernic C, Alata E, Nicomette V, Kaaniche M (2018), State of the art of network protocol reverse engineering tools. Journal of Computer Virology and Hacking Techniques 14.
- Rogean (2015). Project 1999 Daybreak, and Velious.  
<https://www.project1999.com/forums/showthread.php?t=191146>
- Franceschi-Bicchierai L (2017). For 20 Years This Man Has Survived Entirely by Hacking Online Games. [https://www.vice.com/en\\_us/article/59p7qd/this-man-has-survived-by-hacking-mmo-online-games](https://www.vice.com/en_us/article/59p7qd/this-man-has-survived-by-hacking-mmo-online-games)
- Heeks R (2010). Understanding “Gold Farming” and Real-Money Trading as the Intersection of Real and Virtual Economies. The Journal of Virtual Worlds Research.

Rahul J (2008). Cheating and Virtual Crimes in Massively Multiplayer Online Games.

Sulake (2013). Sulake shuts down a pirate Habbo software distributor.

<https://web.archive.org/web/20130820043510/https://www.sulake.com/press/releases/sulake-shuts-down-a-pirate-habbo-software-distributor/>

UOX3 (2020). Ultima Offline Experiment. <https://www.uox3.org/>

Alasiri A, Muteb Z, Dale L, Pavol Z, Ron R, Shafi A (2012). Comparative Analysis of Operational Malware Dynamic Link Library (DLL) Injection Live Response vs. Memory Image. International Conference on Computing, Communication and Informatics Management (ICCCSIM 2012)

Webb S, Soh S (2007). Cheating in networked computer games: a review. In Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts (DIMEA '07). Association for Computing Machinery, New York, NY, USA, 105–112.

HackersOnBoard. (2019, October 5). DEF CON 25 - Manfred - Twenty Years of MMORPG Hacking Better Graphics and Same Exploits [Video]. YouTube. [https://www.youtube.com/watch?v=ZAUf\\_ygqsDo](https://www.youtube.com/watch?v=ZAUf_ygqsDo)