

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Informaatikainstituut

IDU70LT
Priit Laht 132273

PLATVORMIST SÕLTUMATU VOOGEDASTUSSÜSTEEM
PARTNERVÕRGUS TELESARJADE NÄITEL

Magistritöö

Juhendaja: Enn Õunapuu
doktorikraad
dotsent

Tallinn 2016

Autorideklaratsioon

Deklareerin, et antud magistritöö on minu töö tulemus ning seda pole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

Autor: Priit Laht

04.05.2016

Platvormist sõltumatu voogedastussüsteem partnervõrgus telesarjade näitel

Annotatsioon

Käesoleva töö eesmärk on luua veebipõhine keskkond telesarjade voogedastuseks partnervõrgus. Loodav süsteem peab töötama erinevatel platvormidel lisatarkvara ja pistikprogramme paigaldamata. Lisaks on oluline, et suurem osa vajaminevatest andmetest pärineksid välistest andmeallikatest.

Püstitatud eesmärkide saavutamiseks teostatakse esmalt loodava lahenduse süsteemianaalüüs ning võrreldakse väliseid andmeallikaid vajaliku informatsiooni pärimiseks. Seejärel pannakse paika esialgne süsteemiarhitektuur ning arendatakse välja rakenduse aluseks olev andmebaas. Lõpetuseks antakse ülevaade loodud süsteemist, esinenud probleemidest ning võimalikest edasiarendustest.

Töö põhitulemus on platvormist sõltumatu telesarjade voogedatusteenuse artefakt. Pakutud kontseptsiooni ja domeenimudelit on hiljem võimalik aluseks võtta teiste sarnaste süsteemide loomisel. Lisaks võib teostatud väliste andmeallikate võrdlus lihtsustada telesarjade ja *torrent* informatsiooni liidestuse disainimist analoogsetes rakendustes.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 73 leheküljel, 5 peatükki, 22 joonist, 11 tabelit.

Platform Independent Streaming in Peer-to-Peer Network: A Case Study of TV Series

Abstract

The aim of this thesis is to develop a web-based environment that enables streaming TV series in peer-to-peer networks. Given system must work on different platforms without any additional software or plug-ins installed. In addition, it is important that most of the required information is queried from external sources.

To accomplish the stated objectives, simplified system analysis is conducted and external data sources are compared. Next, system architecture is established and its underlying database developed. Finally, a short overview of the created application is given and several possible improvements are proposed.

The main outcome of the thesis is a platform independent TV series streaming service artifact. The proposed concept and domain model can be used as a basis for creating other similar systems. In addition, the carried out data source comparison might simplify the interface design for communicating with external systems that offer TV series and torrent material.

The thesis is written in Estonian and contains 73 pages of text, 5 chapters, 22 figures, 11 tables.

Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> , rakendusliides
CGI	<i>Common Gateway Interface</i> , üldine lüüsiliides
CSS	<i>Cascading Style Sheets</i> , kaskaadlaadistik
FLAC	<i>Free Lossless Audio Codec</i> , tasuta kadudeta audiokodek
FTP	<i>File Transfer Protocol</i> , failiedastusprotokoll
HTML	<i>HyperText Markup Language</i> , hüpertexti märgistuskeel
HTTP	<i>HyperText Transfer Protocol</i> , hüpertexti edastusprotokoll
IEC	<i>International Electrotechnical Commission</i> , Rahvusvaheline Elektrotehnika Komisjon
IMDB	<i>Internet Movie Database</i> , kinematograafia teemaline andmebaas
IP	<i>Internet Protocol</i> , internetiprotokoll
ISO	<i>International Organization for Standardization</i> , Rahvusvaheline Standardiseerimisorganisatsioon
JSON	<i>JavaScript Object Notation</i> , andmevahetusformaad
KB	<i>Kilobyte</i> , kilobait
MP3	<i>MPEG-1 audio layer 3</i> , MPEG audiokiht 3
P2P	<i>Peer-to-Peer</i> , partnervõrk
REST	<i>Representational State Transfer</i> , klient-server suhtlusprotokoll
RTSP	<i>Real Time Streaming Protocol</i> , reaalaaja-striimingiprotokoll
SHA	<i>Secure Hashing Algorithm</i> , turvaline räsi algoritm
SSL	<i>Secure Sockets Layer</i> , turvasoklite kiht
TCP	<i>Transmission Control Protocol</i> , edastusohje protokoll
TMDB	<i>The Movie Database</i> , kinematograafia teemaline andmebaas
TVDB	<i>Television Database</i> , telesarjade andmebaas
UDP	<i>User Datagram Protocol</i> , kasutajadatagrammi protokoll
UML	<i>Unified Modeling Language</i> , unifitseeritud modelleerimiskeel
URL	<i>Uniform Resource Locator</i> , internetiaadress

Sisukord

Sissejuhatus	10
1 Ülevaade kontspetsioonidest ja tehnoloogiatest	12
1.1 BitTorrent protokoll	12
1.2 Voogedastus	15
1.3 WebRTC ja WebTorrent	18
2 Loodava lahenduse õiguspärasus	20
3 Lahenduse analüüs	24
3.1 Süsteemi eesmärgid	24
3.2 Pädevusalad	25
3.3 Põhiobjektid	25
3.4 Funktsionaalsed nõuded	26
3.5 Mittefunktsionaalsed nõuded	31
3.6 Ärireeglid	33
3.7 Kontseptuaalne andmemudel	34
4 Väliste andmeallikate võrdlus	40
4.1 Telesarjade informatsiooni allikad	40
4.1.1 The Movie Database	40
4.1.2 Trakt.tv	44
4.1.3 Muud liidesed	49
4.1.4 Valiku kriteeriumid ning sobivaima liidese valik	49
4.2 Torrenti otsingu rakendusliidesed	50
4.2.1 KickassTorrents	51
4.2.2 Strike	52
4.2.3 EZTV	52
4.2.4 Formaatide võrdlus ning sobivaima defineerimine	53
5 Lahenduse realisatsioon	55
5.1 Süsteemiarhitektuur	55
5.1.1 Tehniline ülevaade	56
5.1.2 Valitud platvormid ning nende alternatiivid	57
5.2 Andmebaasi disain	59
5.2.1 Disaini üldised põhimõtted	60

5.2.2	Andmebaasi struktuur	61
5.3	Ülevaade loodud lahendusest	62
5.4	Võimalikud edasiarendused	64
	Kokkuvõte	66
	Summary	67
	Kasutatud materjalid	69
	Lisa 1	70
	Lisa 2	73

Jooniste loetelu

1	<i>BitTorrent protokoll põhivoog</i>	12
2	<i>BitTorrent partnervõrk koos jälgijaga</i>	13
3	<i>Voogedastuse tsentsaliseeritud lahenduse mudel</i>	17
4	<i>Voogedastuse detsentsaliseeritud lahenduse mudel</i>	17
5	<i>WebTorrenti hübriidmudel</i>	19
6	<i>Kliendi ja administraatori pädevusala kasutusjuhtude diagramm</i>	27
7	<i>EVS-ISO/IEC 25010:2011 tarkvaratoote kvaliteedimudel</i>	31
8	<i>Süsteemi esialgne olemi-suhte diagramm (põhisüsteem)</i>	34
9	<i>Süsteemi esialgne olemi-suhte diagramm (audit)</i>	35
10	<i>The Movie Database populaarsete telesarjade vastuse klassidiagramm</i>	42
11	<i>The Movie Database telesarjade otsingu vastuse klassidiagramm</i>	43
12	<i>The Movie Database telesarja detailide vastuse klassidiagramm</i>	44
13	<i>Trakt.tv populaarsete telesarjade vastuse klassidiagramm</i>	47
14	<i>Trakt.tv telesarjade otsingu vastuse klassidiagramm</i>	48
15	<i>Trakt.tv telesarja detailide vastuse klassidiagramm</i>	49
16	<i>KickassTorrents torrent otsingu vastuse klassidiagramm</i>	51
17	<i>Strike torrent otsingu vastuse klassidiagramm</i>	52
18	<i>EZTV torrent otsingu vastuse klassidiagramm</i>	53
19	<i>Loodava lahenduse torrent liidestuse suhtlusformaadi klassidiagramm</i>	54
20	<i>Kolmekihilise süsteemiarhitektuuri mudel</i>	55
21	<i>Loodava lahenduse süsteemiarhitektuur</i>	56
22	<i>Loodava lahenduse andmebaasi diagramm</i>	61

Tabelite loetelu

1	<i>Olemitüüpide definitsioonid</i>	35
2	<i>Olemitüüpide atribuutide definitsioonid</i>	35
3	<i>The Movie Database rakendusliidese põhiparameetrid</i>	40
4	<i>The Movie Database populaarsete telesarjade sisendparameetrid</i>	42
5	<i>The Movie Database telesarjade otsingu sisendparameetrid</i>	43
6	<i>The Movie Database telesarja detailide sisendparameetrid</i>	44
7	<i>Trakt.tv rakendusliidese põhiparameetrid</i>	45
8	<i>Trakt.tv populaarsete telesarjade sisendparameetrid</i>	46
9	<i>Trakt.tv telesarjade otsingu sisendparameetrid</i>	47
10	<i>Trakt.tv telesarja detailide sisendparameetrid</i>	48
11	<i>Loodud lahenduse mahuanalüüsi tulemused</i>	62

Sissejuhatus

Tänapäeval toimub tehnika areng sedavõrd kiiresti, et tehnoloogiad ja kontspetsioonid, mis tundusid veel kümnekond aastat tagasi revolutsioonilised, hakkavad juba tasapisi ajale jalgu jääma ning vajavad täiendusi. Sama tendents kehtib ka BitTorrenti ja voogedastuse puhul. Pakutakse lahendusi, mis pole tehnoloogiselt midagi uut ning pigem esitatakse vanu süsteeme lihtsalt nii-öelda uues moodsamas vormis või suurendatakse võimekust massiivsete serveriparkidega.

Hetkeolukord (AS-IS) telesarjade voogedastuses on tehnoloogilise poole pealt suuresti kinni kahte tüüpi lahendustes. Esimesed neist kujutavad endast veebilehti või rakendusi, mille kaudu saab striimida otse serverisse üles laetud faile. Taolised teenused on näiteks Netflix ja Hulu, mis omavad kuumakset ning mis pole ka kõigis riikides kättesaadavad. Lisaks eksisteerib ka suur hulk kahtlase väärtusega veebilehti, mis on küll nii-öelda täiesti tasuta, kuid mille sisu on tõenäoliselt illegaalne ning mille iga nupuvajatus avab kõikvõimalike erinevaid reklaame. Puhtalt serveripõhise voogedastuse lahenduse põhiprobleem on suur ressursikulu. Iga kasutaja lisandumine suurendab koormust serverile ning teenusepakkuja võrguühendusele. Teist tüüpi lahendused on oma olemuselt allalaetavad programmid, mis kasutavad striimimiseks küll *torrente*, kuid pole see eest platvormist sõltumatud. Windows, Mac ja Linux platvormidel võimaldavad *torrent* voogedastustust näiteks programmid Powder Player ning Popcorn Time. Nutiseadmete jaoks on taolise lahenduse leidmine märksa keerukam ning üldjuhul tuleb paigaldada eraldi *torrent* kui ka voogedastuse rakendus. Android platvormil töötab näiteks Flud ja VLC kombinatsioon. Nagu näha, siis on allalaetavate programmide peamine puudus see, et need pole platvormist sõltumatud – iga platvormi jaoks tuleb luua eraldi rakendus.

Antud töö eesmärgiks on välja töötada lahenduse kontspetsioon (TO-BE), mis võimaldaks telesarjade voogedastust partnervõrgu kaudu veebibrauseri vahendusel. Taolise süsteemi loomine vähendaks esiteks oluliselt koormust serveritele ning põhisisu jagamise eest vastutaksid nii-öelda kasutajad ise (*peer-to-peer*). Telesarjade failid ei asuks enam pelgalt ühes kindlas serveris, mis peab suutma teenindada kõiki huvitatuid kliente. Partnervõrgu süsteemi puhul oleks vastav sisu kättesaadav ka kõigilt teistelt kasutajatelt, kes seda omavad või parasjagu striimivad. Tagamaks lisaks ka platvormist sõltumatuse võiks süsteem olla kättesaadav veebilehitseja vahendusel. See võimaldaks ligipääsu erinevatest seadmetest lisatarkvara loomata. Näiteks töötaks süsteem ühtemoodi nii personaalarvutist,

nutitelefonist kui ka tahvelarvutist.

Lahenduse kontspetsiooni väljatöötamisel kasutatav meetodika on oma olemuselt tegevusuuring (ingl k *action reasearch*), mille peamine uurimisküsimus (ingl k *research question*) on järgmine: „Kuidas luua süsteem, mis võimaldab telesarjade voogedastust partnervõrgus veebilehitseja kaudu?“. Vastavalt uurimisküsimusel tuleb leida vastused ka järgmistele alamküsimustele:

- „Millised on tehnoloogilised piirangud?“;
- „Kuidas luua legaalne ja eetiline süsteem?“;
- „Mis on lahenduse peamised eesmärgid ning nõuded?“;
- „Kust pärida telesarjade informatsioon?“;
- „Millist *torrent* liidestuse suhtlusformaati kasutada?“;
- „Millised on sobivaimad tehnoloogiad lahenduse loomiseks?“.

Töö jaguneb viieks peatükiks. Esimeses peatükis antakse ülevaade lahenduse loomisel kasutatavatest tehnoloogiatest ning nende tööpõhimõtetest. Täpsemalt vaadeldakse BitTorrent protokoll, voogedastust ning WebRTC ja WebTorrenti olemust. Teises peatükis käsitletakse BitTorrent protokolliga legaalsuse ja eetika küsimusi ning püstitatakse nõuded loodavale lahendusele lähtudes legaalsusest ja eetilistest aspektidest. Kolmandas alajaotises teostatakse süsteemi strateegiline ning detailanalüüs, kus esitatakse süsteemi eesmärgid, põhiobjektid, funktsionaalsed ja mittefunktsionaalsed nõuded ning üldistatud domeenimudel. Töö neljandas peatükis vaadeldakse erinevaid väliseid andmeallikaid, valitakse välja sobivaim telesarjade rakendusliides ning pakutakse välja suhtlusformaati kasutaja poolt configureeritavaks *torrent* liidestuseks. Viimases peatükis peatutakse süsteemi tehnilisel teostusel. Kirjeldatakse nii süsteemiarhitektuuri, tehnoloogisi valikuid kui ka andmebaasi disaini. Lõpetuseks antakse lühiülevaade loodud lahendusest ning pakutakse välja võimalikud edasiarendused.

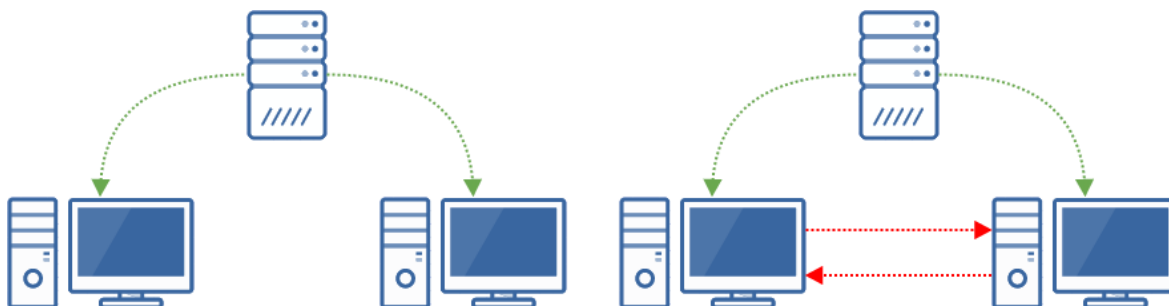
1. Ülevaade kontspetsioonidest ja tehnoloogiatest

Antud peatükis tutvustatakse lühidalt lahenduse loomisel kasutatavaid kontspetsioone ning selle tehnoloogilist tausta.

1.1. BitTorrent protokoll

BitTorrent mõistega seondub enamikule inimestest üldjuhul piraatlus ning ebaseaduslik autoriõigustega kaitstud materjalide jagamine. Seos pole kindlasti alusetu, sest suurem osa piraatlusest toimub tõesti üle BitTorrent protokoll. Samas pakub protokoll ka hulganiisti võimalusi täiesti legitiimseteks eesmärkideks. Näiteks saab BitTorrenti kaudu alla laadida erinevaid Linuxi distributsioone oluliselt kiiremalt kui näiteks tavalist FTP või HTTP protokoll kasutades. Lisaks jagatakse BitTorrent vahendusel ka üha enam muud leaalset sisu, näiteks vabavara tarkvara ja filme. BitTorrent sobib eelkõige populaarsete failide jagamiseks, sest faili erinevad osad on saadaval paljudelt erinevatelt kasutajatelt. Tegu on nii-öelda võimendusefektiga (Cohen 2013). Vanemate ning ebapopulaarsete failide jaoks antud protokoll seevastu väga hästi ei sobi. Loodava rakenduse konteksti klappib BitTorrent oma olemuselt suurepäraselt, sest enamiku telesarjade uute osade vastu on kasutajate huvi suur.

BitTorrent näol on tegu protokolliga, mis muudab failide jagamise lihtsaks ja esmaselt jagajalt vähem ressursi nõudvaks. Tulemus saavutatakse sellega, et lisaks esialgsele jagajale, jagavad faili ka kõik allalaadijad (Cohen 2013). Erinevalt tavalisest klient-server failijagamise süsteemist ei lisa allalaadijate suur kasv erilist koormust jagaja võrguühendusele.



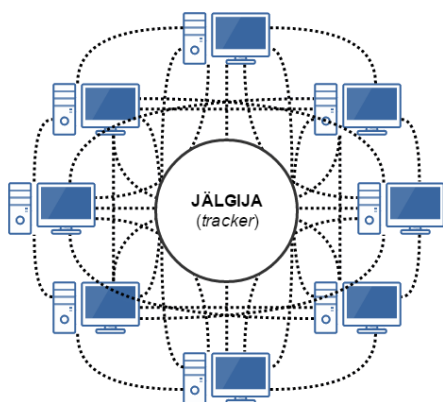
Joonis 1. BitTorrent protokoll põhivoog

Joonisel 1 on kujutatud BitTorrent protokoll põhivoogu. Vasakpoolne joonis näitab klient-server lähenemist faili jagamisele – kõik allalaadijad saavad faili ühest serverist. Kui antud juhul eeldada, et serveri üleslaadimise kiirus on sama, mis iga faili soovija allalaadimise kiirus, siis kahekordistub allalaadimiseks kuluv aeg iga soovija lisandumisel. Parempoolne joonis näitab BitTorrenti lähenemist samale probleemile – lisaks serverile jagab faili ka iga allalaadija, millega kaasneb nii serveri koormuse kui ka allalaadimiseks kuluva aja vähenemine.

Arhitektuur

BitTorrent arhitektuur koosnev üldistatult järgmistest osadest: (Sarioiu jt 2001)

- staatiline metaandmete fail (ingl k *metainfo*);
- jälgija (ingl k *tracker*);
- esmane üleslaadija (ingl k *seed*);
- allalaadiv kasutaja (ingl *leecher*).



Joonis 2. BitTorrent partnervõrk koos jälgijaga

Esimene samm faili avaldamiseks BitTorrent kaudu on jagatavast failist metaandmete faili loomine. Faililaiendiks on *torrent*, kust pärineb ka antud protokoll nime laiendalaialt levinud nimi. Metaandmete fail sisaldab jagatava faili nime, suurust, räsi informatsiooni (ingl k *hashing information*) ning jälgija URL-i. Antud metainfot vajavad kõik kasutajad, kes soovivad alla laadida faili, millest *torrent* loodi. *Torrent* luuakse mõnda vabavaralist programmi kasutades ning üldjuhul toetavad seda ka enamik BitTorrent kliente. BitTorrent klient on vajalik ka failide jagamiseks ning allalaadimiseks. Tuntumatest klientidest võib välja tuua näiteks uTorrenti ning Vuze'i.

Jälgija ülesandeks on hetkel faili allalaadivate kasutajate kohta info hoidmine ning üksteise leidmise hõlbustamine. Otseselt jälgija ise allalaa-

dimise protsessis ei osale ega oma ka koopiad failidest. Kasutajad ja jälgija vahetavad infot üle HTTP protokolliga. Esiteks jagab kasutaja jälgijale infot selle kohta, mis faili ta alla laadida soovib, millist porti kuulab jmt. Jälgija saadab vastuseks nimekirja kasutajatest, kes samuti antud faili allalaevavad ning kuidas nendega ühendust saada. Rühma partneritest (ingl k *peer*), kes kõik jagavad sama *torrenti* tuntakse mõiste all *parv* (ingl k *swarm*).

Selleks, et fail muutuks kättesaadavaks peab esmane üleslaadija (ingl k *seed*) alustama jagamist. Tegu on kasutajaga, kes omab kogu faili sisu. Allalaadivat kasutajat, kellel pole ühtegi osa failist tuntakse kui kaani (ingl k *leecher*). Esmane üleslaadija peab jagama vähemalt ühe täieliku koopia failist. Niipea kui terve fail on jaotatud teiste allalaadijate vahel, võib esmane üleslaadija jagamise peatada. Allalaadimine ei seisku niikaua, kuni eksisteerib piisavalt kasutajaid ning kõik faili osad on olemas.

Jagatavast failist *torrenti* loomisel tükeldatakse see väiksemateks, üldiselt 256KB või 512KB suurusteks, osadeks. SHA-1 räsid loodud tükkidest pannakse metaandmetega kaasa. Andmeid kontrollitakse võrreldes allalaetud osade ning metaandmetes olevate räside väärtusi (Cohen 2013). Antud mehhanism tagab andmete terviklikkuse ning selle et kasutajad laevad alla soovitud osi.

Algoritmid

Erinevalt BitTorrentist toimub enamikus teistes failijagamise protokollides jagamine nii-öelda üks-ühele, st kasutaja valib ise kust sisu tõmmata. BitTorrent puhul see mudel aga ei toimi, sest korraga on võimalik faili alla laadida paljudelt erinevatelt kasutajatelt. See nõuab omakorda algoritme, et teada, millistelt kasutajatelt, milliseid failiosasid küsida. Kogu see protsess peab toimuma lõppkasutaja sekkumiseta ning olema iseseadistuv. Probleemi võib suuresti jagada kaheks. Esiteks tuleb teada saada parim järjekord failiosade allalaadimiseks ning seejärel käsitleda olukordi, kus osad kasutajad võivad endalt allalaadimist blokeerida. Parima järjekorra eest vastutab osade valiku algoritm (ingl k *piece selection algorithm*) ning blokeerivate kasutajate eest lämbumise algoritm (ingl k *choking algorithm*) (Xuemin jt 2010, 670).

See, kuidas BitTorrent valib millises järjekorras osasid alla laadida, omab suurt mõju protokolliga jõudlusele. On tähtis, et ei tekiks olukordi, kus kõigil kasutajatel on ühed samad enimlevinud osad ning ülejäänuid pole kellelgi. Osade valiku algoritmi ülesandeks ongi

võimalikult kiiresti dubleerida erinevaid osi erinvatele kasutajatele. Tulemuseks on alla laadimise kiiruse kasv ning see et kõik osad on võrgus olemas isegi siis kui mõni kasutaja lahkub.

Suuresti tagatakse see järgneva nelja eeskirja abil: (Xuemin jt 2010, 675)

- range eeskiri (ingl k *strict policy*) – seni kuni terve osa on kokku pandud, tuleb alla laadida vaid antud osa alamosasid;
- haruldased enne (ingl k *rarest first*) – leida kasutajate hulgas kõige haruldasemad osad ning kõigepealt alla laadida just need;
- juhuslik esimene osa (ingl k *random first piece*) – valida ja alla laadida suvaline osa;
- lõppmängu režiim (ingl k *endgame mode*) – alamosasid, mida kasutajal pole, tuleb küsida kõigilt teistelt kasutajatelt.

Lisaks osade valiku algoritmile on protokollis jõudluse tagamiseks väga oluline lämbumise algoritm, mis välistab selle, et „muidusööjad“ hävitavad partnervõrgu dünaamika. Kui kasutaja hakkab teisi blokeerima, siis ta nii-öelda lämmatatakse, mis kajastub otseselt allalaadimise kiiruses. Lämbumise algoritmi põhiline eesmärk on ühtlustada üles- ja allalaadimise kiiruseid – mida rohkem jagad seda kiiremini saad ka ise alla laadida.

Kogu see funktsionaalsus on protokollis implementeeritud selliselt, et lõppkasutaja ei pea ise sekkuma. Otsused võtab vastu iga kasutaja BitTorrent klient põhinedes eelpool toodud algoritmidele. Reeglistik, kuidas protokoll ühendused loob ja seadistab, muudab selle väga dünaamiliseks ja töökindlaks.

1.2. Voogedastus

1990ndate alguses muutusid personaalarvutid piisavalt võimsaks, et taasesitada videoid ja helisalvestisi. Neid esimesi arvutimeedia vorme ei jagatud üldiselt voogedastuskanalite kaudu vaid laaditi veebiserverist alla arvuti kõvakettale või mängiti tagasi CD-ROM-ilt (Fechey-Lippens 2010). Toona oli peamiseks väljakutseks meedia suurus ning ribalaiuse

piirangud. Nimelt polnud Interneti kiirused veel sellised, et oleks suutnud mõistliku ajaga jagada suuremat sorti faile.

Voogedastus sai võimalikuks alles 2000ndate aastate alguses koos ribalaiuste ning arvutisüsteemide võimsuse kasvuga. Voogedastuse ehk striimingu (ingl k *streaming*) all mõeldakse üldiselt meedia edastamist arvutivõrgu kaudu, kus andmeid töödeldakse ühtlase voona, mis võimaldab neid taasesitada samal ajal, kui järgnevaid andmeid alla laetakse (Kurose jt 2012, 593). Selle tulemusena ei pea kasutaja ootama, kuni allalaadimine on lõppenud, mis võib aeglase ühenduse ja suurte failide puhul võtta tunde, ning saab neid koheselt vaatama või kuulama hakata.

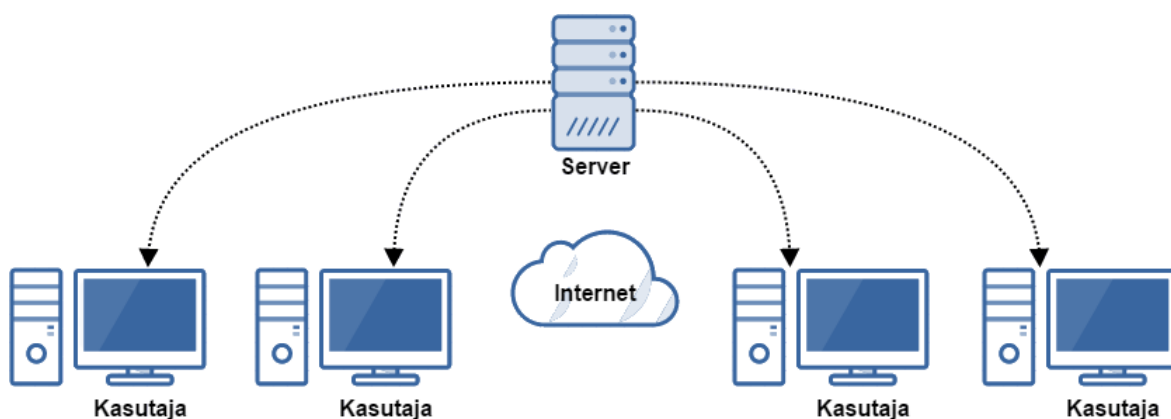
Voogedastuse peamised eeliseid teiste traditsiooniliste meedia kohaletoimetamise viiside ees on järgmised:

- pole vaja oodata kuni allalaadimine on lõppenud;
- voogedastatud faile ei kirjutata kõvakettale ning pärast taasesituse lõppu hävitatakse need automaatselt jätmata andmetest koopiat;
- voogedastuse kaudu on võimalik edastada meediat reaajas;
- toetab interaktiivsust, võimaldades sisu edastada mittelineaarsel viisil, näiteks saab kasutaja liikuda videos suvalisele positsioonile;
- hea võimalus vähendada veebilehtede staatilisust.

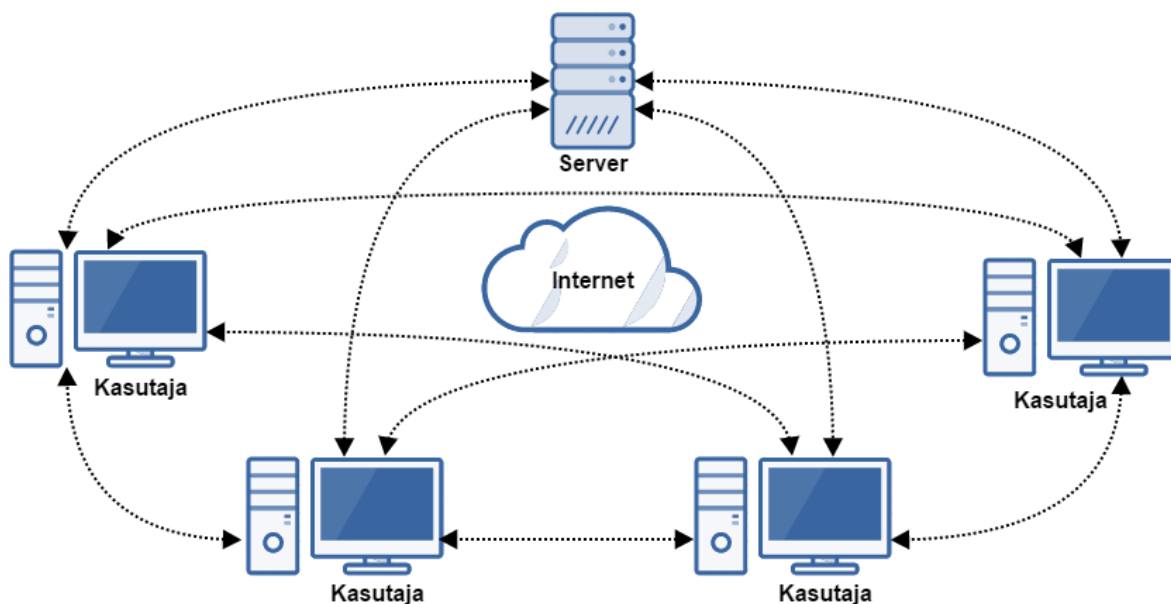
Voogedastust ei kasutata üldiselt n-ö tavaliste andmete edastuseks. Kuna audio ja video on ajast sõltuvad meediumid, siis on nende sujuvalt mängimiseks oluline et informatsioon jõuaks kohale õigeaegselt ja heas seisukorras. Kuna avalik Internet on aga üles ehitatud kui asünkroonne süsteem, kus pole oluline, mis järjekorras paketid kohale jõuavad, siis tekitab see voogedastuse jaoks mõningaid probleeme (Adobe Dynamic Media Group 2001). Seega ongi TCP poolt pakutav veaparanduse mehhanism üldiselt voogedastuse juures ohverdatud ning kui andmed lähevad edastuse käigus kaotsi, siis on nad kadunud jäädavalt. Tulemuseks on mittetäiuslik ning mõnikord isegi loetamatu esitus. Õnneks on aga video ja audio n-ö „väga andestavad“ meediumid ning kui paar paketti kaduma läheb, on fail siiski vaadatav või kuulatav. Failide, mille jaoks on oluline täielik informatsiooni konsistents, voogedastus, oma regulaarsel kujul, ei sobi.

Tänapäeval toetavad voogedastust mitmed erinevad protokollid: TCP/IP, HTTP, RTSP, P2P jt (McGath 2013). Isegi HTML5 video märgend lubab endale sisuks anda voogedastatava faili. Kuna antud töö käigus loodav süsteem peab olema võimalikult platvormist sõltumatu, siis on üheks eelduseks just brauseri ning HTML5 voogedastuse tugi. Lisaks on erinevate protokollide toe põhjal näha, et voogedastuse lahendus võib olla tsentraliseeritud (kasutajale jagab faili vaid server) või detsentraliseeritud (üksteisele jagavad faile kõik kasutajad). Kuna loodav süsteem peab toimima partnervõrgus, siis tuleb kasutada detsentraliseeritud voogedastust.

Joonistel 3 ja 4 on toodud võrdlusena tsentraliseeritud ja detsentraliseeritud voogedastuse lahenduse mudelid.



Joonis 3. Voogedastuse tsentraliseeritud lahenduse mudel



Joonis 4. Voogedastuse detsentraliseeritud lahenduse mudel

1.3. WebRTC ja WebTorrent

Antud töö käigus loodav voogedastussüsteem peab olema platvormist sõltumatu ning töötama partnervõrgus. Vältimaks erinevate aplikatsioonide loomist, uuritakse täpsemalt kas süsteemi loomine nii-öelda veebilehena kasutades partnervõrku on üldse võimalik ning kui siis kuidas.

Voogedastus partnervõrgus pole oma olemuselt midagi uut ning eksisteerib mitmeid vabavaralisi programme, mis lubavad alla laetud *torrentit* striimida, näiteks uTorrent ning Powder Player. Eelmainitud rakendused kasutavad oma sisemuses tavalist BitTorrenti TCP/UDP transpordiprotokolli. Kuna loodav rakendus peab olema kättesaadav hoopis veebilehena brauseris, siis on HTTP protokolliga piirangute tõttu TCP/UDP kasutamine välistatud ning tuleb leida alternatiiv.

Järgnevalt vaadeldaksegi TCP/UDP transpordiprotokolli ainukese saadaval oleva alternatiivina WebRTC-d ning seda kasutavat WebTorrent klienti.

WebRTC

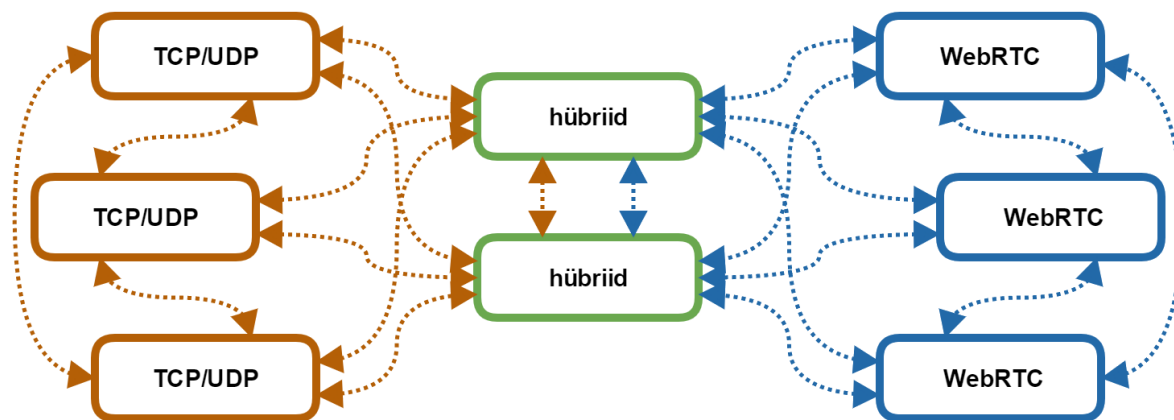
WebRTC (Web Real-Time Communication) näol on tegu rakendusliidesega, mis on mõeldud eelkõige brauserite vaheliseks hääli- ja videokõnedeks ning kasutajate vaheliseks failiedastuseks ilma ühegi pistikprogrammita (ingl k *plugin*) (Johnston jt 2013, 1). Erinevalt näiteks WebSocket ja XMLHttpRequest protokollidest pole WebRTC nii-öelda klient-server mudel, kus andmeid vahendab kasutajatele keskne server. Nimelt lubab WebRTC RTC-DataChannel liides otsest suhtlust brauserite vahel serveri vahenduseta. Kogu suhtlus ja andmevahetus on lisaks ka krüpteeritud (Johnston jt 2013, 109).

1. veebruari 2016 seisuga toetavad WebRTC liidest järgmised brauserid ja platvormid: Microsoft Edge, Google Chrome, Mozilla Firefox, Opera, Android ja iOS (Google Inc 2016). Populaarsetest veebilehitsejatest puudub tootjapoolne WebRTC tugi vaid Safaril ning Microsoft Internet Exploreril. Tõenäoliselt jõuab vähemalt Safarisse tugi küllaltki varsti ning hetkel saab selle soovi korral lisada mõnda vabavaralist pistikprogrammi kasutades.

WebTorrent

WebTorrent on oma olemuselt veebipõhine BitTorrent klient, millel on ka suurepärase voogedastuse tugi. Erinevalt standardsest BitTorrent lahendusest ei kasuta WebTorrent transpordiprotokollina TCP/UDP-d vaid hoopis eelnevalt kirjeldatud WebRTC-d (Abouk-hadijeh 2016).

Pakutakse kahte klienti: puhtalt brauseripõhist, mis suudab ühenduda vaid nii-öelda veebikasutajatega (ingl k *web peer*) ning hübriidi, mis suhtleb nii tavaliste TCP/UDP kui ka WebRTC ühendustega. Kuna loodav rakendus on brauseripõhine, siis sobib vaid esimene variant, mistõttu peab suhtlus toimuma vaid veebikasutajatega.



Joonis 5. WebTorrenti hübriidmudel

WebTorrent kitsaskohaks antud rakenduse loomisel võib esialgu osutada just protokollis uudsus ning see et populaarsed BitTorrent kliendid hetkel veel WebRTC ühendusi ei toeta. Seetõttu võib *torrentite* jagajate arv hetkeseisuga antud protokollis olla veel üpriski puudulik. WebTorrent loojad tegelevad, aga aktiivselt sellega, et tugi jõuaks peagi kõikidesse populaarsetesse BitTorrent klientidesse, millega peaks kaasnema ka jagajate probleemi lahenedmine.

2. Loodava lahenduse õiguspärasus

Järgnevalt antakse ülevaade BitTorrenti legaalsusest ja eetikast ning loodava süsteemi õiguspärasusest.

BitTorrent legaalsus

Failide jagamine partnervõrgus pole oma olemuselt illegaalne. Õiguslikud küsimused teevad alles siis kui tegu on autoriõigusega kaitstud materjaliga. Näiteks BitTorrent protokolliga kasutamine mõne isetehtud filmi jagamiseks on legaalne, kuid mõne uuema telesarja või filmi allalaadimine kasutades mõnda populaarset *torrentit* on juba tõenäoliselt illegaalne.

Olgugi et üldjuhul on filmide ja telesarjade allalaadimine illegaalne, võib tuua järgmised erandid:

- Aegunud autoriõigused – üldjuhul on loometöö autoriõigustega kaitstud 70 aastat ning pärast seda autoriõigused enam ei kehti ja töö muutub üldkasutatavaks. Seega on meedia, mis on vanem kui 70 aastat, allalaadimine BitTorrent kaudu täiesti seaduslik (*Autoriõiguse seadus* § 41).
- Creative Commons litsentseeritud – meediat, millel on Creative Commons litsents, võib autoriõigusi rikkumata vabalt allalaadida (*Creative Commons License*).
- Puuduvad autoriõigused – loometööd, mis ilmusid enne 1. aprilli 1989 ilma autoriõigusteta, võib seaduslikult alla laadida (Templeton 1995).
- Kokkulepe autoriõiguste omajaga – jagamiseks faile avalikkusega peab jagajal olema antud faili omaja nõusolek. Antud mudelit kasutab näiteks populaarne telesarjade ja filmide voogedastuse teenus Netflix (Bell 2016).

Lühidalt kokkuvõttes võib öelda järgmist:

- partnervõrgu tehnoloogiad on seaduslikud;
- BitTorrent on seaduslik;
- *torrent* failide jagamine on seaduslik;
- partnervõrgu, BitTorrenti ja *torrent* failide kasutamine autoriõigusega kaitstud materjali jagamiseks on **ebaseaduslik**.

Partnervõrgus failide jagamisele võib tuua paralleeli relva kasutamisest – relva omamine ja kasutamine näiteks lasketiirus, omades selleks vastavaid õigusi on legaalne, kuid selle tarvitamine näiteks panga röövimiseks enam mitte.

Eestis käsitleb intellektuaalse omandi vastaseid süütegusid karistusseadustiku 14. peatükk, millest tasub eraldi välja tuua paragraafid 223 ja 225:

§ 223. Teose ja autoriõigusega kaasnevate õiguste objekti ebaseaduslik üldsusele suunamine

(1) Teose või autoriõigusega kaasnevate õiguste objekti ebaseadusliku avaliku esitamise, üldsusele näitamise, edastamise, taasedastamise või kättesaadavaks tegemise eest ärilisel eesmärgil – karistatakse rahalise karistuse või kuni üheaastase vangistusega.

(2) Sama teo eest, kui selleks on kasutatud piraatkoopiat, – karistatakse rahalise karistuse või kuni kolmeaastase vangistusega.

(3) Käesoleva paragrahvi lõikes 1 või 2 sätestatud teo eest, kui selle on toime pannud juriidiline isik, – karistatakse rahalise karistusega.

(4) Kohus konfiskeerib käesoleva paragrahvi lõikes 2 sätestatud süüteo vahetuks objektiks olnud eseme. (*Karistusseadustik* § 223)

§ 225. Infoühiskonna teenuse ja meediateenuse ebaseaduslik vastuvõtmine

(1) Tasulisele infoühiskonna teenusele või tasu eest edastatavale meediateenusele või nendele juurdepääsu võimaldavale teenusele ebaseaduslikku juurdepääsu võimaldavate seadmete või tarkvara ärilisel eesmärgil valmistamise, edasiandmise, paigaldamise, hooldamise, omamise või reklaamimise eest – karistatakse rahatrahviga kuni 300 trahviühikut.

(2) Sama teo eest, kui selle on toime pannud juriidiline isik, – karistatakse rahatrahviga kuni 3200 eurot. (*Karistusseadustik* § 225)

BitTorrent eetika

Lisaks õiguslikele küsimustele tuleb BitTorrenti kasutamise puhul kindlasti käsitleda ka eetika punkti. Olgugi et autoriõigustega kaitstud materjali allalaadimine on ebaseaduslik, leitakse tihtipeale, et mõningates olukordades võib see olla justkui andestatav.

Näitena võib tuua järgmised olukorrad:

- Proovimine – telesarjade DVD-d, arvutimängud ning -programmid on üldjuhul küllaltki kallid ning proovimata on üpris raske otsustada, kas need ka reaalselt inimesele meeldivad. Seega tundub näiteks telesarjade puhul mõistlik paar osa *torrenti* kaudu alla laadida ja veenduda, et sari huvi pakub, enne kui nii-öelda legaalne versioon osta. Sama printsiip kehtib ka arvutimängude- ning programmide puhul, kus tootja ei paku endapoolset prooviversiooni.
- Duplikaadid teises failivormingus – tihtipeale omatakse meediast legaalset koopiat mõnes sellises failivormingus, mida mõni seade ei toeta. Näiteks omatakse Kindle või paberkandja versiooni raamatust, mida soovitakse lugeda hoopis nutitelefonis. Selle stsenaariumi puhul tundub eetiline alla laadida sama raamat teises formaadis, olgugi et tehniliselt on see illegaalne.
- Kõrgem kvaliteet, mida legaalset saada pole – sageli pole soovivat meediat soovitavas kvaliteedis seaduslikult kuskilt saada ja ainuke võimalus on illegaalne allalaadimine. Üldjuhul seondub see just muusikaga, kus soovitakse n-ö kadudeta kvaliteediga faile (FLAC), kuid müügil on vaid MP3-ed. Analoogselt võib tuua situatsiooni, kus filmist või telesarjast omatakse DVD-d, kuid hiljem on müügile tulnud ka Blu-ray, mis on saadaval vaid USA-s. Mõlemal juhul ei tundu ebaeetiline illegaalse kõrgema kvaliteediga versiooni soetamine kui eelnevalt omatakse madalama kvaliteediga versiooni.

Partnervõrgu ning BitTorrent eetika küsimused on üpriski suhtelised ning on väga raske kindlalt defineerida, mis on eetiline ja mis mitte. Tegu, mis mõnele tundub eetiline võib teiste jaoks osutuda täiesti vastuvõetamatuks. Vastupidiselt legaalsuse aspektile ei jookse eetika puhul kuskil selget piiri ning seega jääb see antud teenuse loomisel pigem teisejärguliseks.

Nõuded loodavale lahendusele lähtudes legaalsetest ja eetilistest aspektidest

Vältimaks *torrenti* sisu autoriõigustega seonduvaid probleeme on mõistlik süsteem disainida selliselt, et *torrenteid* ega ka nende sisu ei hoitaks kuskil endapoolsetes serverites. Lähtutakse printsiibist, et loodava lahenduse puhul on tegu nii-öelda vahendiga, mis võimaldab *torrenteid* brauseri kaudu voogedastada. Süsteem hoiab vaid telesarjade ja kasutajate informatsiooni ning oskab töödelda kasutaja poolt sisestavat *torrenti* teenuse vastust. Kasutaja võtab vastutuse selle eest, kas tema poolt sisestatud teenusest leitavate *torrentite* sisu on legaalne või mitte. See, millist teenust kasutatakse ei mängi süsteemi tööpõhimõtetes mingit rolli ning ainukeseks eelduseks on formaadi vastavus defineeritud nõuetele. Antud nõuded esitatakse järgnevates peatükkides.

Lühidalt kokkuvõttes lähtutakse süsteemi loomisel järgmistest printsiipidest:

- süsteem ei paku ise *torrenteid* ega teenust nende leidmiseks;
- iga kasutaja sisestab *torrent* teenuse URL-i, mille vastust süsteem töötleb;
- süsteem eeldab, et kasutaja sisestatud *torrent* teenus pakub sisu, mis on legaalne, vastasel juhul on tegu kasutustingimuste rikkumisega.

Seega lasub põhiline vastutus just kasutajal ning tema poolt sisestatud *torrent* teenuse pakkujal. See, kas süsteem, mis ei suuda iseseisvalt tuvastada, kas *torrenti* sisu on legaalne või mitte, on pigem juba eetika küsimus ning jääb antud töö skoobist välja.

3. Lahenduse analüüs

Käesolevas alajaotises teostatakse loodava lahenduse strateegiline ning detailanalüüs. Antud töö keskendub eelkõige just kliendipoolsete nõuete väljaselgitamisele. Tuuakse välja ka mõned peamised süsteemi administraatori pädevusalad, kuid kõigil neil lähemalt ei peatuta ning suurem osa neist jääb töö skoobist välja. Funktsionaalsete nõuete seadmisel lähtutakse põhimõttest, et süsteem peab algselt olema võimalikult lihtsakoeline ning ei tohiks omada üleliigset funktsionaalsust. Sama põhimõte kajastub ka funktsionaalsete nõuete põhjal koostatud andmemudelil. Mittefunktsionaalsete nõuete seadmisel lähtutakse parimatest praktikatest ning kasutatavate tehnoloogiate piirangutest. Kindlasti tuleb ära märkida ka see, et nii telesarjade kui ka *torrent* sisendandmed saadakse välistest süsteemidest ning seega käsitletakse neid käesolevas peatükist vaid pealiskaudselt. Täpsem lahkamine toimub alajaotises „Välise andmeallikate võrdlus“. Analüüsi etapi tulemuseks on telesarjade info ning *torrent* liidestuse üldistatud domeenimudel, mida on võimalik aluseks võtta ka teiste sarnaste süsteemide loomisel.

3.1. Süsteemi eesmärgid

Alljärgnevalt on toodud loodava süsteemi põhilised eesmärgid:

- pakkuda klientidele käepärast võimalust voogedastada telesarjade *torrente* veebibrauseri vahendusel;
- anda kliendile mugav ülevaade hetkel populaarsetest telesarjadest ning võimaldada neid ka soovi korral nime järgi otsida;
- võimaldada kliendil kasutada erinevaid *torrent* liidestuse teenuseid konfigureerides vaid vastavat veebiaadressi välja;
- pakkuda kliendile võimalust lisada telesarju enda lemmikute hulka ning selle põhjal genereerida käepärane nimekiri viimastest linastunud osadest;
- võimaldada soovi korral kliendil ka *torrente* tavapärasel viisil alla laadida;
- silmas pidada kliendi privaatsust ning mitte salvestada süsteemi kriminaliseerivaid andmeid;

- anda administraatorile võimalus vaadata registreeritud kasutajaid ning neid aktiveerida ja deaktiveerida;
- võimaldada süsteemi sisemist nii tavalist kasutaja/parool lahendust kui ka väliste süsteemide (Facebook, Google, Twitter) liidestusi kasutada;
- võimaldada administraatorile ülevaadet sisselogimise katsetest, nende õnnestumise ning ebaõnnestumiste põhjustest.

3.2. Pädevusalad

Süsteem hõlmab esialgselt vaid kahte pädevusala:

- kliendi pädevusala;
- administraatori pädevusala.

Nagu juba eespool mainitud, siis keskendutakse antud peatükis just eelkõige kliendi pädevusalale, sest süsteemi põhifunktsionaalsus on suunatud just kliendile. Administraatori pädevusala pakub nii-öelda toetavat funktsionaalsust, mida on võimalik hiljem soovi korral täiendada. Samuti on vastava vajaduse tekkimisel võimalik pädevusalasid süsteemi lisada.

3.3. Põhiobjektid

Järgnevalt on välja toodud süsteemisisesed põhiobjektid koos ingliskeelsete vastetega:

- kasutaja (ingl k *user*);
- volitus (ingl k *authority*);
- kasutaja volitus (ingl k *user authority*);
- kasutaja telesari (ingl k *user show*);

- sotsiaalse kasutaja seos (ingl k *social user connection*);
- audit sündmus (ingl k *audit event*);
- audit sündmuse andmestik (ingl k *audit event data*);
- püsiv märgend (ingl k *persistent token*).

Lisaks tuleb üldistatud objektina välja tuua kasutatavate väliste süsteemide vastused:

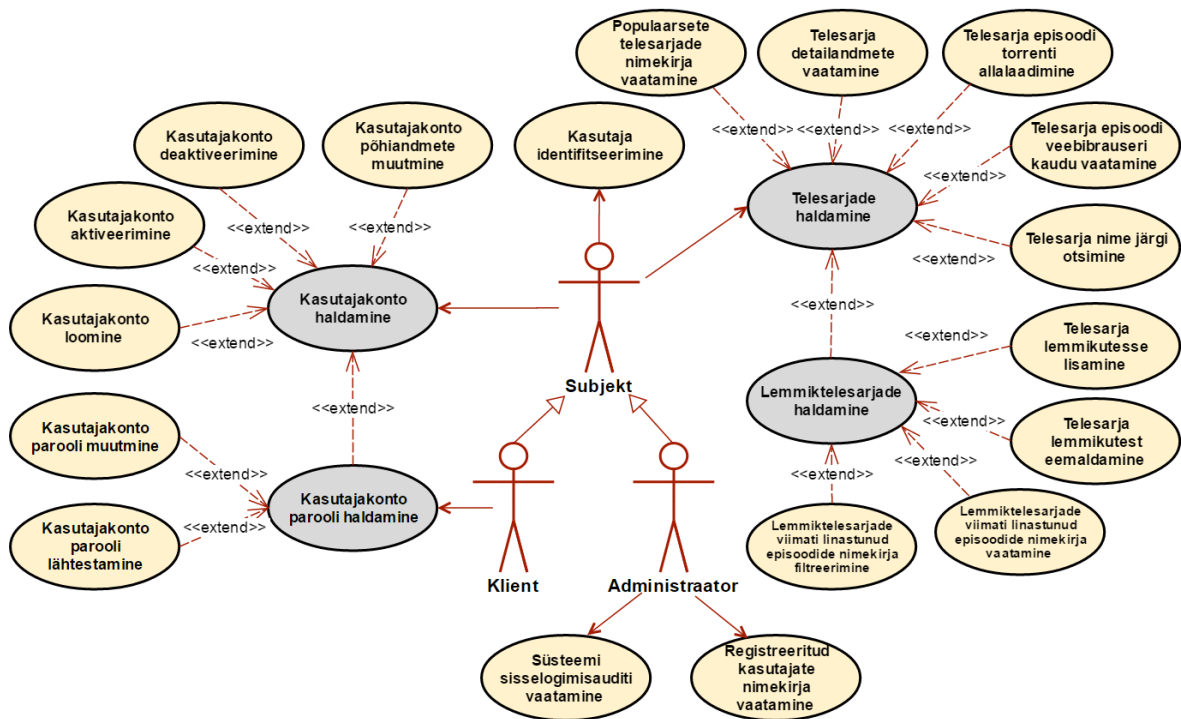
- telesarjade päringu vastus (ingl k *show query response*);
- torrent päringu vastus (ingl k *torrent query response*);
- sotsiaalse kasutaja päringu vastus (ingl k *social user query response*).

Nagu näha, siis ei eralda süsteem kliendi ja administraatori põhiobjekti vaid kasutab ühtset kasutajat. Õiguste eest vastutab kasutaja volituse objekt. Nagu eelnevalt mainitud, siis peab süsteem olema võimalikult lihtsakoeline ning seetõttu on ka kasutaja andmeid sedavõrd vähe, et puudub vajadus klienti ja administraatorit objektipõhiselt eraldada. Edasistes etappides ning joonistel kasutatakse põhiobjektide ingliskeelseid vasteid.

3.4. Funktsionaalsed nõuded

Antud alampeatükis esitatakse loodava süsteemi funktsionaalsed nõuded. Nende kaardistamiseks luuakse primitiivne kasutusjuhtude mudel, mis koosneb UML notatsioonidel põhinevast kasutusjuhtude diagrammist ning kõrgtaseme formaadis kasutusjuhtude kirjeldustest.

Kliendi ja administraatori pädevusala kasutusjuhtude diagramm on toodud joonisel 6.



Joonis 6. Kliendi ja administraatori pädevusala kasutusjuhtude diagramm

Kasutusjuht: Kasutajakonto loomine (FR-001)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Klient soovib ennast registreerida süsteemi kasutajaks. Süsteem avab uue kasutaja registreerimise vormi ning palub sisestada kasutajanime, parooli ning e-maili aadressi. Seejärel saadetakse kliendi poolt sisestatud e-mailile kinnituskiri konto aktiveerimise viitega. Analoogselt saab uusi kasutajaid süsteemi sisestada ka administraator. Alternatiivselt võimaldab süsteem konto loomiseks kasutada ka Google, Facebook ja Twitter kasutajaid, mille puhul aktiveeritakse loodud konto automaatselt.

Kasutusjuht: Kasutajakonto aktiveerimine (FR-002)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Kliendile on eelnevalt loodud kasutajakonto, mis on deaktiveeritud seisundis. Klient soovib kontot aktiveerida ning avab selleks registreerimisel sisestatud e-mailile saadetud kinnituskirjas oleva viite. Viite avamisel aktiveerib süsteem kliendi kasutajakonto ning saadab vastavasisulise kinnituskirja. Alternatiivselt võimaldab süsteem kontosid aktiveerida ka administraatoril.

Kasutusjuht: Kasutajakonto deaktiveerimine (FR-003)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Kliendil eksisteerib süsteemis aktiivses seisundis kasutajakonto, mida soovitakse deaktiveerida. Selleks saadetakse kliendile kinnituskiri deaktiveerimise viitega. Viite avamisel deaktiveerib süsteem kliendi kasutajakonto. Alternatiivselt võimaldab süsteem kontosid deaktiveerida ka administraatoril.

Kasutusjuht: Kasutaja identifitseerimine (FR-004)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Subjekt soovib süsteemi siseneda ning sisestab selleks oma kasutajanime ja parooli või kasutab alternatiivset Google, Facebook või Twitter autentimist. Süsteem kontrollib sisestatud andmeid ning võrdleb neid andmebaasis olevate väärtustega lähtudes turvaalgoritmidest. Kui vastavate andmetega kasutajakonto eksisteerib, siis logitakse kasutaja süsteemi sisse.

Kasutusjuht: Kasutajakonto põhiandmete muutmine (FR-005)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Subjekt uuendab enda kasutajakonto põhiandmeid. Näiteks võidakse muuta oma e-maili aadressi või kasutatavat *torrent* teenuse URL-i.

Kasutusjuht: Kasutajakonto parooli muutmine (FR-006)

Tegutsejad: Klient

Lühikirjeldus: Klient vahetab oma kasutajakonto parooli. Näiteks soovitakse parool muuta keerukamaks ja raskemini äraarvatavaks. Uus parool rakendub järgmisel süsteemi sisenemisel.

Kasutusjuht: Kasutajakonto parooli lähtestamine (FR-007)

Tegutsejad: Klient

Lühikirjeldus: Klient on unustanud oma kasutajakonto parooli ning ei saa enam süsteemi sisse. Probleemi lahendamiseks sisestab ta parooli lähtestamise vormile oma e-maili ning süsteem saadab sellele parooli lähtestamise viite. Viitele vajutades suunatakse kasutaja parooli muutmise vormile.

Kasutusjuht: Populaarsete telesarjade nimekirja vaatamine (FR-008)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Subjekt soovib saada ülevaadet hetkel populaarsetest telesarjadest. Süs-

teem kasutab selleks välist telesarjade rakendusliidest (täpsemalt kirjeldatakse väliseid liideseid alajaotises „Välise andmeallikate võrdlus“). Süsteem töötleb vastava päringu vastust ning esitab selle kasutajale arusaadaval kujul populaarsuse järgi kahanevalt. Kuvatakse nii telesarja pealkirjad kui *posterid*.

Kasutusjuht: Telesarja detailandmete vaatamine (FR-009)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Subjekt soovib vaadata telesarja täpsemaid detaile. Süsteem kasutab selleks jällegi telesarjade rakendusliidest (täpsemalt kirjeldatakse väliseid liideseid alajaotises „Välise andmeallikate võrdlus“). Süsteem töötleb vastava päringu vastust ning esitab kasutajale telesarja detailandmed. Lisaks üldinformatsioonile kuvatakse ka näiteks näitlejad ning episoodid hooegade kaupa. Kui kasutaja on eelnevalt sisestanud *torrent* teenuse URL-i ja süsteem tuvastab episoodile vastava torrenti, siis kuvatakse iga episoodi juures ka vastav viide.

Kasutusjuht: Telesarja lemmikutesse lisamine (FR-010)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Subjekt lisab telesarja enda lemmikute hulka. Süsteem salvestab andmebaasi telesarja- ning kasutajaidentifikaatori seose.

Kasutusjuht: Telesarja lemmikutest eemaldamine (FR-011)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Subjekt eemaldab telesarja enda lemmikute hulgast. Süsteem kustutab andmebaasist telesarja- ning kasutajaidentifikaatori seose.

Kasutusjuht: Lemmiktelesarjade viimati linastunud episoodide nimekirja vaatamine (FR-012)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Subjekt soovib näha nimekirja enda lemmikisarjade viimastest episoodidest koos *torrent* viidetega. Süsteem kasutab nimekirja loomiseks kasutaja poolt sisestatud *torrent* teenust ning andmebaasi salvestatud kasutaja lemmikisarju. Kasutajale kuvatakse episoodid kuupäeva kahanemise järjekorras.

Kasutusjuht: Lemmiktelesarjade viimati linastunud episoodide nimekirja filtreerimine (FR-013)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Subjekt soovib lemmiksarjade nimekirja telesarja pealkirja järgi filtreerida. Süsteem kuvab kasutajale episoodide nimekirjast vaid need, mille pealkiri sisaldab kasutaja poolt sisestatud sõne.

Kasutusjuht: Telesarja nime järgi otsimine (FR-014)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Subjekt soovib otsida telesarja pealkirja järgi. Süsteem kasutab selleks telesarjade rakendusliidest (täpsemalt kirjeldatakse väliseid liideseid alajaotises „Välise andmeallikate võrdlus“). Süsteem töötleb vastava päringu vastust ning esitab kasutajale nimekirja telesarjadest, mis sisaldasid tema poolt sisestatud sõne.

Kasutusjuht: Telesarja episoodi veebibrauseri kaudu vaatamine (FR-015)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Subjekt soovib vaadata telesarja episoodi otse veebibrauseris. Süsteem leiab vastava *torrenti* kasutaja poolt sisestatud *torrent* teenusest ning avab selle brauseri modaalses aknas. Seejärel mängitakse episood jagajate olemasolul kasutajatele tagasi kasutades WebTorrent tehnoloogiat.

Kasutusjuht: Telesarja episoodi *torrenti* allalaadimine (FR-016)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Subjekt soovib telesarja episoodi alla laadida. Süsteem leiab vastava *torrenti* kasutaja poolt sisestatud *torrent* teenusest ning tekitab sellele allalaadimise viite.

Kasutusjuht: Registreeritud kasutajate nimekirja vaatamine (FR-017)

Tegutsejad: Administraator

Lühikirjeldus: Administraator soovib saada ülevaadet kõikidest infosüsteemi registreerunud kasutajatest. Süsteem kuvab nimekirja andmebaasis olevate kasutajate põhiandmetest.

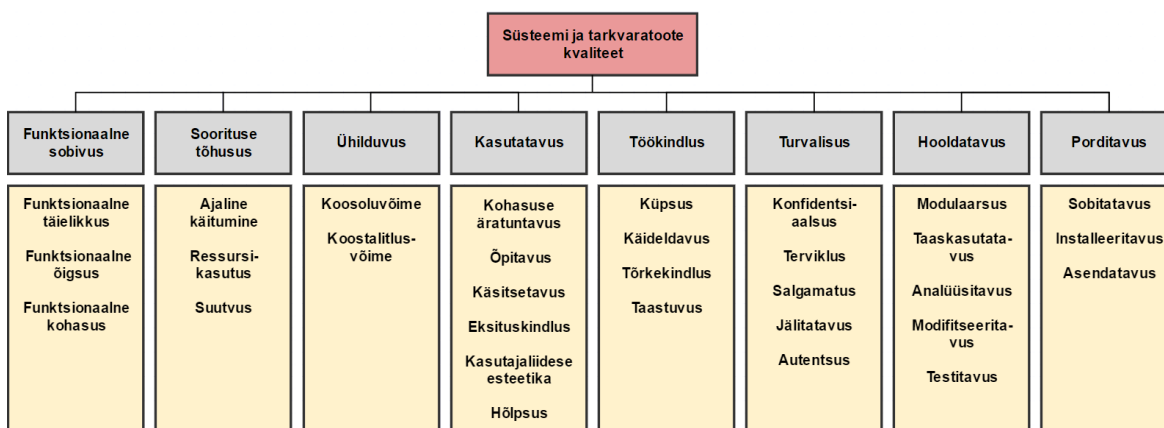
Kasutusjuht: Süsteemi sisselogimisauditi vaatamine (FR-018)

Tegutsejad: Administraator

Lühikirjeldus: Administraator soovib näha süsteemi sisenemise katseid. Süsteem kuvab nimekirja kasutajanimedest, kes on üritanud süsteemi siseneda. Lisaks kasutajanimedele kuvatakse ka kuupäev, kellaaeg, kasutaja IP aadress ning sisselogimise staatused.

3.5. Mittefunktsionaalsed nõuded

Loodava lahenduse mittefunktsionaalsete nõuete kirjeldamisel lähtutakse rahvusvahelisest standardist ISO/IEC 25010:2011. Täpsemalt on aluseks võetud AS Cybernetica poolt tõlgitud eestikeelne versioon EVS-ISO/IEC 25010:2011. Antud standardi on avaldanud Eesti Standardikeskus. Selles esitatud tootekvaliteedimudelit (vt joonis 7) saab kasutada relevantsete kvaliteedikarakteristikute piiritlemiseks, mida saab omakorda kasutada nõuete, nende täidetuse kriteeriumite ja vastavate näitajate kehtestamiseks (Süsteemide ja tarkvara kvaliteedinõuded ja kvaliteedi hindamine 2012, V). Mudel on ülesehitatud kahetasemelise kvaliteedikriteeriumite süsteemina, mille iga põhikriteerium koosneb mitmest erinevast alamkriteeriumist. Taoline jaotus võimaldab mittefunktsionaalseid nõudeid ühtlasemalt määratleda ning tagab selle, et erinevad aspektid oleks kaetud.



Joonis 7. EVS-ISO/IEC 25010:2011 tarkvaratoote kvaliteedimudel

Faktor/Kriteerium: Soorituse tõhusus/ajaline käitumine (NFR-001)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Subjekt teeb süsteemi veebiliideses nupuvajutuse, millele süsteem reageerib vähemalt 5 sekundi jooksul. Süsteemiväliste teenustega suhtlemisel peab süsteem andma veateate kui päringu vastus pole 5 sekundi vältel saabunud. Tingimus kehtib süsteemi madala ning mõõduka koormuse korral (kuni 100 üheaegset kasutajat).

Faktor/Kriteerium: Soorituse tõhusus/ressursikasutus (NFR-002)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Süsteem peab kasutama HTTP vahemälu päiseid staatiliste ressursside hoidmiseks, mis võimaldab veebibrauseritel ja proksidel neid vastavalt ka vahemälust lei-

da. Lisaks peab süsteem võimalusel serveri vahemälus hoidma andmebaasi transaktsioone ja päringuid.

Faktor/Kriteerium: Ühilduvus/koostalitusvõime (NFR-003)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Loodava rakenduse HTML ning CSS kood peab olema valiidne vastavalt World Wide Web Consortiumi poolt koostatud spetsifikatsioonidele (versioonid HTML5 ning CSS3).

Faktor/Kriteerium: Kasutatavus/eksituskindlus (NFR-004)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Ebakorrekse sisestuste korral annab süsteem selgitava veateate, mitte ei esita koodi läbijooksmisel heidetud erindit. Tehnilised veateate detailid peavad olema tavakasutaja eest peidetud.

Faktor/Kriteerium: Kasutatavus/käsitsetavus (NFR-005)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Loodava süsteemi kasutajaliides peab olema võimalikult mugav ning arusaadav. Kõigile rakenduse kuvadele peab olema võimalik navigeerida kolme hiirevajutusega.

Faktor/Kriteerium: Töökindlus/tõrketaluvus (NFR-006)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Väliste teenuste (telesarjade ning *torrent* liideste) probleemid ei tohi häirida terviksüsteemi tööd. Antud olukordadele tuleb reageerida vastava veateatega või teenuse funktsionaalsuse ajutise peatamisega. Näiteks *torrent* liidestuse probleemide korral peab endiselt tööle jääma telesarjade informatsiooni funktsionaalsus.

Faktor/Kriteerium: Töökindlus/taastuvus (NFR-007)

Tegutsejad: ei kohaldu

Lühikirjeldus: Süsteemis olevaid andmeid peab iganädalaselt varundama, et võimalda tõrgete korral eelnevat seisuga taastada. Rakenduse lähtekood peab asuma versioonihalduse süsteemis ning andmebaasi koopiad eraldiseisvas pilveteenuses või kõvakettal.

Faktor/Kriteerium: Turvalisus/salgamatus (NFR-008)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Kõik süsteemi sisselogimise katsed ning nende kohta käiv informatsioon (kasutajanimi, IP aadress, staatused, kuupäevad jms) tuleb jäädvustada.

Faktor/Kriteerium: Turvalisus/konfidentsiaalsus (NFR-009)

Tegutsejad: Klient, administraator (e üldistatult subjekt)

Lühikirjeldus: Süsteemi kasutajate parooli hoitakse andmebaasis räsitud kujul. Nõrgimaks aktsepteeritavaks räsifunktsiooniks on MD5 + lisatud juhuslikkus (ingl. k. *salt*), soovitatavalt võiks aga kasutada SHA256-te koos lisatud juhuslikkusega.

Faktor/Kriteerium: Hooldatavus/modifitseeritavus (NFR-010)

Tegutsejad: ei kohaldu

Lühikirjeldus: Süsteemi välimus peab olema hõlpsasti muudetav. Tähtsamate piltide, ikoonide ning taustavärvide vahetamine/asendamine ei tohiks võtta märkimisväärselt palju aega.

Faktor/Kriteerium: Porditavus/sobitatavus (NFR-011)

Tegutsejad: ei kohaldu

Lühikirjeldus: Andmete talletamiseks kasutatava andmebaasisüsteemi ning rakendusserveri väljavahetamine ei tohiks olla väga töömahukas. Süsteem peab võimaldama üleminekut PostgreSQLilt mistahes teisele relatsioonilisele andmebaasisüsteemile kui ka üleminekut Apache Tomcatilt teistele Java rakendusserveri tarkvaradele.

3.6. Ärireeglid

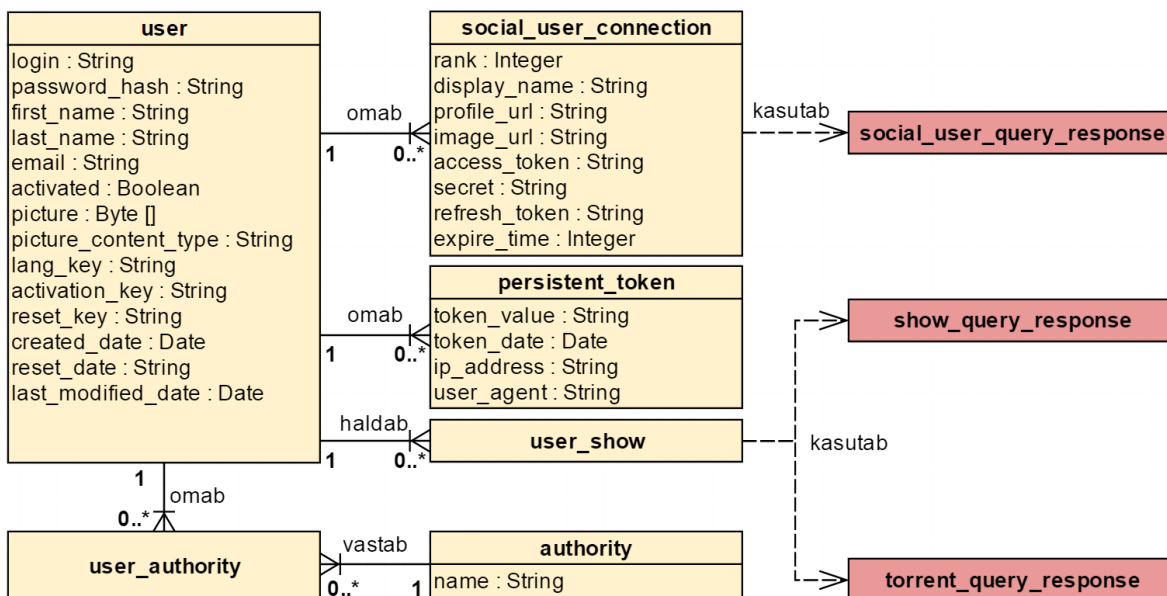
Järgnevalt esitatakse loodava süsteemi ärireeglid, mis pakuvad selgemat ülevaadet põhiobjektide vahelistest seostest. Nende põhjal koostatakse hiljem ka süsteemi kontseptuaalne andmemudel.

- Kasutaja võib omada ühte või enamat volitust.
- Volitus võib olla seotud ühe või enama kasutajaga.
- Kasutaja võib omada ühte või enamat sotsiaalse kasutaja seost.
- Sotsiaalse kasutaja seos peab kuulma ühele ja ainult ühele kasutajale.

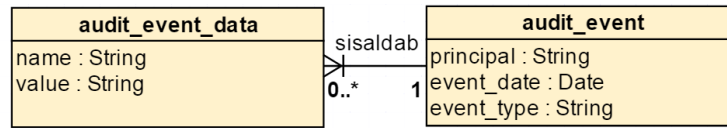
- Kasutaja võib omada ühte või enamat püsivat märgendit (sessiooni).
- Püsiv märgend peab kuulma ühele ja ainult ühele kasutajale.
- Kasutaja võib omada ühte või enamat lemmikut telesarja.
- Telesari võib olla ühe või enama kasutaja lemmik.
- Audit sündmus võib sisaldada ühte või enamat andmestikku.
- Audit sündmuse andmestik peab kuulma ühele ja ainult ühele audit sündmusele.
- Sotsiaalse kasutaja seose leidmiseks kasutakse sotsiaalse kasutaja päringu vastust.
- Telesarjade informatsiooni leidmiseks kasutatakse telesarjade päringu vastust.
- Telesarjade *torrent* informatsiooni leidmiseks kasutatakse *torrent* päringu vastust.

3.7. Kontseptuaalne andmemudel

Vastavalt süsteemianalüüsis määratletud põhiobjektidele ja esitatud funktsionaalsetele ning mittefunktsionaalsetele nõuetele koostatakse loodava süsteemi esialgne andmemudel koos olemi-suhte diagrammiga (vt joonised 8 ja 9). Välistest süsteemidest tulevate andmete seosed on kujutatud olemi-suhte diagrammil punaselt.



Joonis 8. Süsteemi esialgne olemi-suhte diagramm (põhisüsteem)



Joonis 9. Süsteemi esialgne olemitüüpe diagramm (audit)

Joonistel 8 ja 9 kirjeldatud olemitüüpide ning nende atribuutide definitsioonid on täpsemalt lahti seletatud tabelites 1 ja 2.

Tabel 1. Olemitüüpide definitsioonid

Olemitüübi nimi	Definitsioon
user	Infosüsteem kasutaja, kes on huvitatud rakenduse poolt pakutavast funktsionaalsusest.
authority	Volituste komplekt, mis annab ligipääsu teatud tüüpi rakenduse funktsionaalsusele.
user_authority	Kasutajale antud volitused.
user_show	Kasutaja lemmiksarjade kogum.
social_user_connection	Kasutajaga seotud sotsiaalvõrgustike kontod, mis võimaldavad süsteemi sisemist vastavate liideste kaudu.
persistent_token	Kestvad ühendused ehk seansid süsteemis.
audit_event	Kasutaja sisselogimise katset kajastav sündmus.
audit_event_data	Kasutaja sisselogimise katset kajastava sündmuse andmestik.

Tabel 2. Olemitüüpide atribuutide definitsioonid

Olemitüüp	Atribuut	Definitsioon	Näiteväärus
user	login	Liikme unikaalne kasutajanimi, mida kasutatakse süsteemis autentimiseks.	announ
user	password_hash	Kasutaja salasõna krüpteeritud kujul, mida kasutatakse süsteemis autentimiseks.	\$2a\$10\$VEjxo0jq2YG9Rb-k2HmX9S.k1-uZBGYUHdU-cid3g/vfiEl7l-wWgOH/K

Jätkub...

Tabel 2 – Jät kub...

Olemitüüp	Atribuut	Definitsioon	Näiteväärtus
user	first_name	Lapsele tema sünni registree- rimisel antav nimi (Erelt jt 2007, 685).	Ann
user	last_name	Vanemait lapsele kanduv ni- mi (Erelt jt 2007, 685).	Õun
user	email	Kasutaja unikaalne e-posti aadress. Kasutatakse näiteks uue kasutaja aktiveerimise kinnituskirja saatmisel.	ann@oun.ee
user	activated	Tõeväärtus näitamaks, kas kasutajakonto on aktiivne või mitte. Mitteaktiivse kasutaja- kontoga ei saa süsteemi sise- neda.	TRUE
user	picture	Baidimassiiv kasutaja graafi- lisest esindajast süsteemis.	3773303740- 0020JFIF0...
user	picture_content_type	Kasutaja graafilise esindaja sisu tüüp.	image/jpeg
user	lang_key	Kasutaja poolt eelistatud kee- le kood ISO 639-1 formaadis.	en
user	activation_key	Süsteemi poolt genereeritud unikaalne kood kasutajakon- to aktiveerimiseks. Vastav ak- tiveerimise viide saadetak- se konto esmasel registreeri- misel kasutaja poolt sisesta- tud e-posti aadressile.	1029322857- 1582430144
user	reset_key	Süsteemi poolt genereeritud unikaalne kood kasutajakon- to lähtestamiseks. Vastav läh- testamise viide saadetakse kasutaja poolt sisestatud e- posti aadressile.	2660550549- 4396496475

Jät kub...

Tabel 2 – Jätkub...

Olemitüüp	Atribuut	Definitsioon	Näiteväärtus
user	created_date	Kuupäev ja kellaaeg, millal kasutajakonto loodi.	2016-02-08 16:55:50.969
user	reset_date	Kuupäev ja kellaaeg, millal kasutajakonto parool lähetati.	2016-04-07 17:32:19.126
user	last_modified_date	Kuupäev ja kellaaeg, millal kasutajakonto andmeid viimati muudeti.	2016-02-08 16:57:14.973
authority	name	Volituse unikaalne koodnimi süsteemis.	ROLE_USER
social_user_connection	rank	Järjekorra number sotsiaalvõrgu kasutaja unikaalsuse tagamiseks.	1
social_user_connection	display_name	Nimi, kellele sotsiaalvõrgu kasutaja on registreeritud.	Tiit Sei
social_user_connection	profile_url	Sotsiaalvõrgu kasutaja profiili viide.	http://plus.google.com/-1012835423-45737790604
social_user_connection	image_url	Sotsiaalvõrgu kasutaja profiilipildi viide.	http://graph.facebook.com/v2.5/127-9749831084-46/picture
social_user_connection	access_token	Välise liidestuse poolt genereeritud unikaalne juurdepääsuluba kasutaja identifitseerimiseks. Võimaldab piiratud ligipääsu.	BSSVMiqkuh-5cBADkQVVg-hjAWHFp5K-mJOv3JcJib-Mnv42iQPpr-TSZAkglMM-AOvxZAYX2-GYe4P5BJtOp

Jätkub...

Tabel 2 – Jätkub...

Olemitüüp	Atribuut	Definitsioon	Näiteväärtus
social_user_connection	secret	Välise liidestuse poolt genereeritud salajane juurdepääsuluba sotsiaalvõrgu kasutajale. Võimaldab täielikku ligipääsu.	y7Ig9ZCeUR-GPMmwMxD-DgNv5x5Y0u-ZCZAbxstsxfI-0mmw1fzEh-qZCxCGadsf
social_user_connection	refresh_token	Välise liidestuse poolt genereeritud unikaalne värskendus märgend, mida kasutatakse uue juurdepääsuloa omandamiseks.	5cBADkQVVg-hjAWHFP5K-mJOv3JcJib-Mnv42iQPpr-TSZAkglMM-AOvxZAYX2-GYe4P5BJt0
social_user_connection	expire_time	Ajatempel, mis näitab millal vastavale kasutajale antud juurdepääsuluba aegub.	1465235102-101
persistent_token	token_value	Kasutaja püsivale seansile süsteemi poolt genereeritud unikaalne tunnus.	YvT1a+ghC1-ZqHGHJjywq-3w==
persistent_token	token_date	Kuupäev, millal püsiv seanss süsteemi registreeriti.	2016-03-31
persistent_token	ip_address	Internetiaadress, kus seanss kehtib.	88.191.4.74

Jätkub...

Tabel 2 – Jätkub...

Olemitüüp	Atribuut	Definitsioon	Näiteväärtus
persistent_token	user_agent	Kasutajaagent, millel seanss kehtib.	Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.87 Safari/537.36
audit_event	principal	Kasutajakonto nimetus, mida üritati süsteemi sisenemiseks kasutada.	admin
audit_event	event_date	Kuupäev ja kellaaeg, millal sisenemiskatse toimus.	2015-12-30 21:27:32.692
audit_event	event_type	Sisselogimiskatse tulemust seletav kood.	AUTHENTICATION_FAILURE
audit_event_data	name	Sisselogimisauditi andmevälja tüüpnimetus.	message
audit_event_data	value	Sisselogimisauditi andmevälja väärtus.	Bad credentials

4. Välise andmeallikate võrdlus

Nagu juba eespool mainitud, on loodava rakenduse juures eriti tähtsal kohal just välised andmeallikad, sest sisemises andmebaasis talletakse vaid kasutajapõhist informatsiooni. Seega on väga oluline, et välised allikad oleks kvaliteetsed, kasutajasõbralikud ning teeksid vajaliku informatsioon kättesaamise võimalikult mugavaks ja efektiivseks. Antud peatükk keskendubki taolistele andmeallikatele. Esiteks vaadeldakse telesarjade informatsiooni pärimist ning pakutakse sobivaim lahendus. Seejärel vaadeldakse põgusalt ka erinevaid *torrenti* otsingu rakendusliideseid, et saada aimu, milline võiks olla optimaalne sisendformaad, mille kasutaja peab edastama, et süstida antud süsteemi endapoolne *torrentite* informatsioon.

4.1. Telesarjade informatsiooni allikad

Antud peatükis antakse ülevaade erinevatest rakendusliidestest telesarja andmete pärimiseks ning vaadeldakse, milline neist on antud rakenduse loomiseks kõige mõistlikum.

4.1.1. The Movie Database

TMDB on üks suurimaid telesarjade ja filmide info andmebaase, mis pakub ka rakendusliidest vastava info pärimiseks. Enda rakendusliidest reklaamitakse kui kiiret, järjepidevat ja usaldusväärset võimalust kolmandate osapoolte andmete otsinguks. API kasutamine on tasuta, kuid vajab kasutajakonto loomist ning nõuab loodavas rakenduses TMDB-le, kui allikale, viitamist. Lisaks on ära märgitud ka see, et tulevikus võib rakendusliidese kaubanduslikel eesmärkidel kasutamise muutuda tasuliseks. Kindlasti tuleb TMDB puhul ära märkida ka päringute piirang ühelt IP aadressilt, milleks on 30 päringut 10 sekundi jooksul.

Tabel 3. *The Movie Database* rakendusliidese põhiparameetrid

Näitaja	Väärtus
Rakendusliidese aadress	http://api.themoviedb.org/3/

Jätkub...

Tabel 3 – Jät kub...

Näitaja	Väärtus
Autentimine	API key
Protokoll/formaat	JSON, REST
SSL tugi	jah
Erinevate keelte tugi	jah
Limiidid	30 päringut 10 sekundi jooksul ühelt IP aadressilt

Põhifunktsionaalsusena tuuakse välja järgnevad päringuvõimalused:

- kõrgeimalt hinnatud filmid;
- tulevased filmid;
- hetkel jooksvad filmid;
- menuskad filmid;
- menuskad telesarjad;
- kõrgeimalt hinnatud telesarjad;
- hetkel jooksvad telesarjad;
- täna kavas olevad telesarjad.

Telesarjade päringuvõimalustena tuuakse välja järgnev:

- tekstipõhine otsing pealkirja järgi;
- telesarjade otsing parameetrite järgi, näiteks keskmine hinne ja žanr;
- välise identifikaatori põhine otsing, näiteks IMDB identifikaator;
- detailid – põhiinformatsioon, näitlejad, pildid, tõlked, välised identifikaatorid;
- hooajad – põhiinformatsioon, näitlejad, pildid, tõlked, välised identifikaatorid;
- episoodid – põhiinformatsioon, näitlejad, pildid, tõlked, välised identifikaatorid.

Kuna antud rakenduse loomisel on huviorbiidis eelkõige just telesarjadega seonduv, siis vaadeldakse lähemalt järgnevaid päringu tüüpe: populaarsed telesarjad, telesarjade otsing, telesarja detailid. Iga päringu tüübi juures esitatakse näitesisend koos parameetritega, vastuse formaadi klassidiagramm ning vastuse reaktsiooniajad. Reaktsiooniaegade mõõtmiseks kasutatakse lihtsat *bash* funktsiooni, mida saab täpsemalt vaadata lisast 2. Kiiruse mõõtmine on küll üpris subjektiivne ning sõltub mitmetest teguritest, näiteks läbiviija interneti kiirus, vastava serveri töökoormus jne. Siiski võiks see anda esialgse aimduse sellest, milliste reaktsiooniaegadega tasub antud rakendusliidese puhul arvestada.

Populaarsete telesarjade näitesisend:

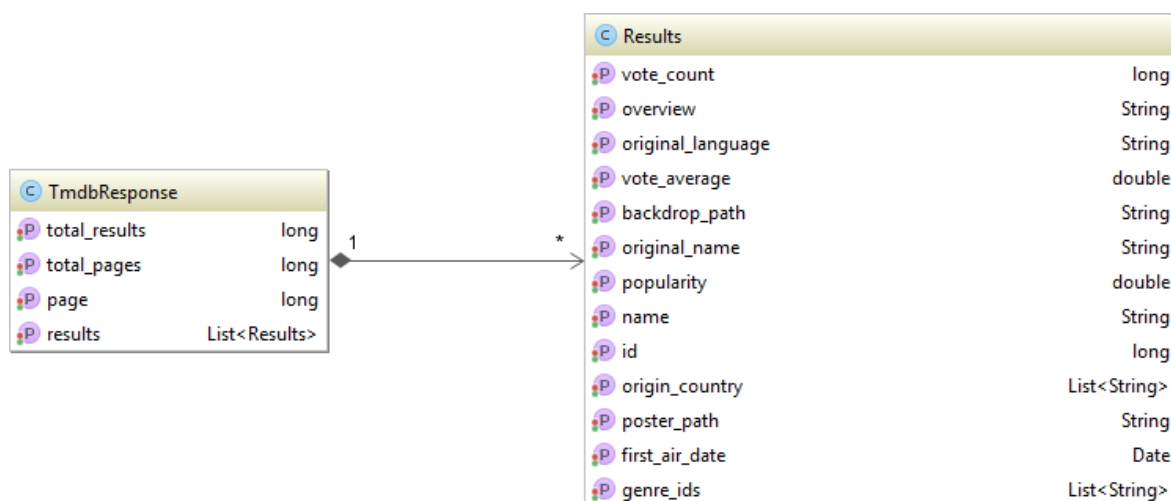
`http://api.themoviedb.org/3/tv/popular?api_key=xyz&page=1`

Populaarsete telesarjade päringu keskmine reaktsiooniaeg: 0,2309 sekundit

Populaarsete telesarjade päringu sisendparameetrid ja vastuse formaadi klassidiagramm on esitatud järgnevalt tabelis 4 ja joonisel 10.

Tabel 4. *The Movie Database* populaarsete telesarjade sisendparameetrid

Parameeter	Formaat	Kohustuslik
api_key	numbrite ja tähtede kombinatsioon	jah
page	1 kuni 1000	ei
language	ISO 639-1 kood	ei



Joonis 10. *The Movie Database* populaarsete telesarjade vastuse klassidiagramm

Telesarjade otsingu näitesisend:

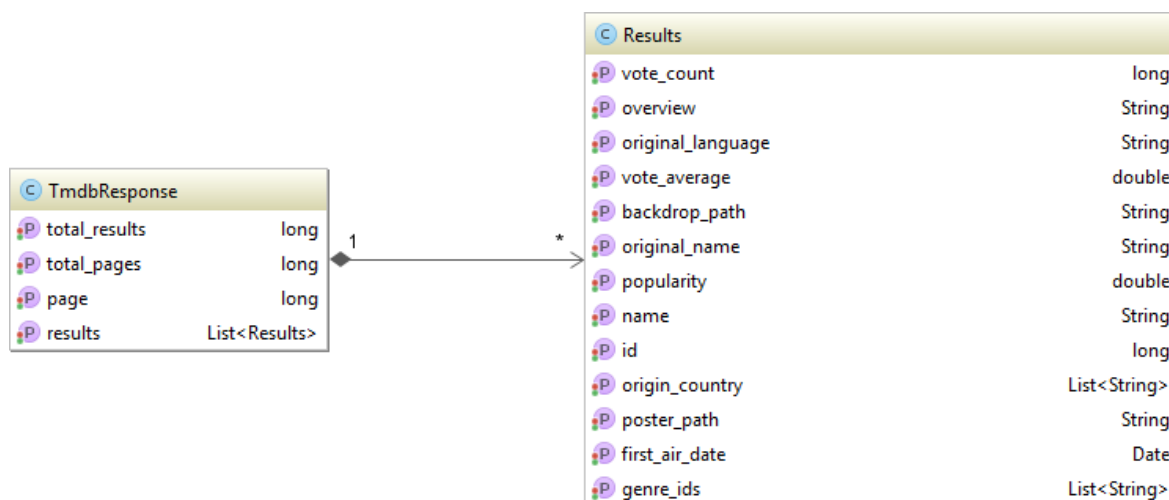
http://api.themoviedb.org/3/search/tv?api_key=x&query=cop

Telesarjade otsingu keskmine reaktsioonaeg: 0,2295 sekundit

Telesarjade otsingu sisendparameetrid ja vastuse formaadi klassidiagramm on esitatud järgnevalt tabelis 5 ja joonisel 11.

Tabel 5. *The Movie Database* telesarjade otsingu sisendparameetrid

Parameeter	Formaat	Kohustuslik
api_key	numbrite ja tähtede kombinatsioon	jah
page	1 kuni 1000	ei
language	ISO 639-1 kood	ei
first_air_date_year	neljakohaline aasta number	ei
search_type	sõne <i>phrase</i> või <i>ngram</i>	ei



Joonis 11. *The Movie Database* telesarjade otsingu vastuse klassidiagramm

Telesarja detailide näitesisend:

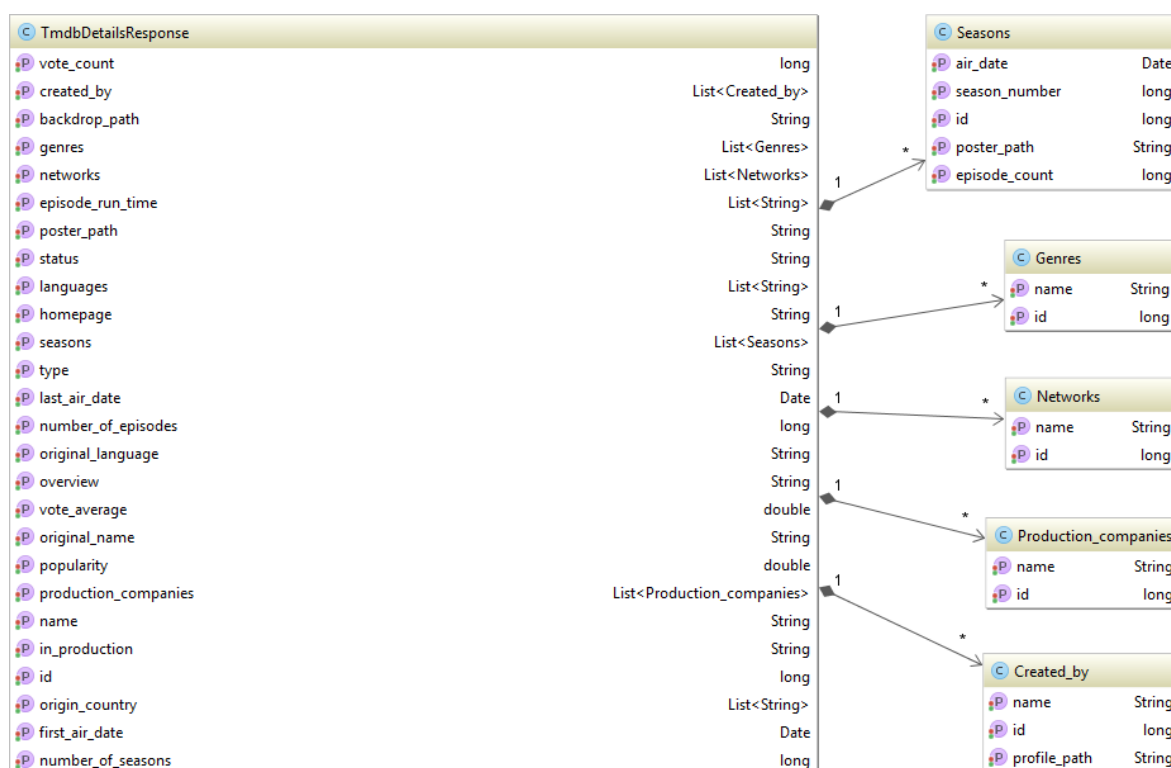
http://api.themoviedb.org/3/tv/1396?api_key=abc123

Telesarja detailide päringu keskmine reaktsioonaeg: 0,2375 sekundit

Telesarja detailide sisendparameetrid ja vastuse formaadi klassidiagramm on esitatud järgnevalt tabelis 6 ja joonisel 12.

Tabel 6. *The Movie Database* telesarja detailide sisendparameetrid

Parameeter	Formaat	Kohustuslik
api_key	numbrite ja tähtede kombinatsioon	jah
language	ISO 639-1 kood	ei
append_to_response	komaga eraldatud sõned (telesarjade rakendusliidese meetodid)	ei



Joonis 12. *The Movie Database* telesarja detailide vastuse klassidiagramm

4.1.2. Trakt.tv

Trakt.tv erineb TMDB-st suuresti selle poolest, et lisaks telesarjade ja filmide informatsioonile pakutakse ka teavet selle kohta, mida kasutajad hetkel vaatavad. Näiteks on võimalik pärida paljud Trakt.tv kasutajad vaatavad hetkel kõige uuemat Star Warsi või mõnda populaarset telesarja. Enda põhiliste infoallikatena tuuakse välja TMDB, TVDB ja Fanart.tv andmebaasid. Nagu ka TMDB on Trakt.tv rakendusliidese kasutamine tasuta. Endapoolt loodav rakendus tuleb küll registreerida, kuid erinevalt TMDB-st ei tooda välja mingeid

olulisi piiranguid. Lisaks ei tooda kuskil välja ka seda, et oma rakenduses peaks Trakt.tv-le, kui allikale, viitama. Erinevalt TMDB-st saadetakse parameetrid nagu näiteks API key ja rakendusliidese versioon HTTP päises, mitte URL parameetrina. Niisamuti on ka lehekülgede numeratsiooni informatsioon vastuse HTTP päises.

Tabel 7. *Trakt.tv* rakendusliidese põhiparameetrid

Näitaja	Väärtus
Rakendusliidese aadress	<code>https://api-v2launch.trakt.tv</code>
Autentimine	API key, OAuth 2, Token
Protokoll/formaat	JSON, REST
SSL tugi	jah
Erinevate keelte tugi	jah (pealkirjad ja tutvustused)
Limiidid	puuduvad või pole välja toodud

Erinevalt TMDB-st ei tooda välja mingeid kindlaid põhifunktsionaalsusi, kuid üldjoontes tundub rakendusliidese funktsionaalsuse poolest olevat üpriski sarnane. Välja võiks tuua järgmised päringuvõimalused:

- tulevased filmid;
- menuskad filmid;
- populaarsust koguvad filmid;
- menuskad telesarjad;
- tulevased telesarjad;
- populaarsust koguvad telesarjad.

Telesarjade päringuvõimalustena tuleks välja tuua järgnev:

- tekstipõhine otsing pealkirja järgi;
- välise identifikaatori põhine otsing, näiteks IMDB identifikaator;
- detailid – põhiinformatsioon, näitlejad, meeskond, pildid, välised identifikaatorid;

- hooajad – põhiinformatsioon, pildid, välised identifikaatorid;
- episoodid – põhiinformatsioon, pildid, tõlked, välised identifikaatorid.

Nagu ka TMDb puhul vaadeldakse lähemalt järgnevaid päringu tüüpe: populaarsed telesarjad, telesarjade otsing, telesarja detailid. Iga päringu tüübi juures esitatakse näitesisend koos parameetritega, vastuse formaadi klassidiagramm ning vastuse reaktsiooniajad.

Populaarsete telesarjade näitesisend:

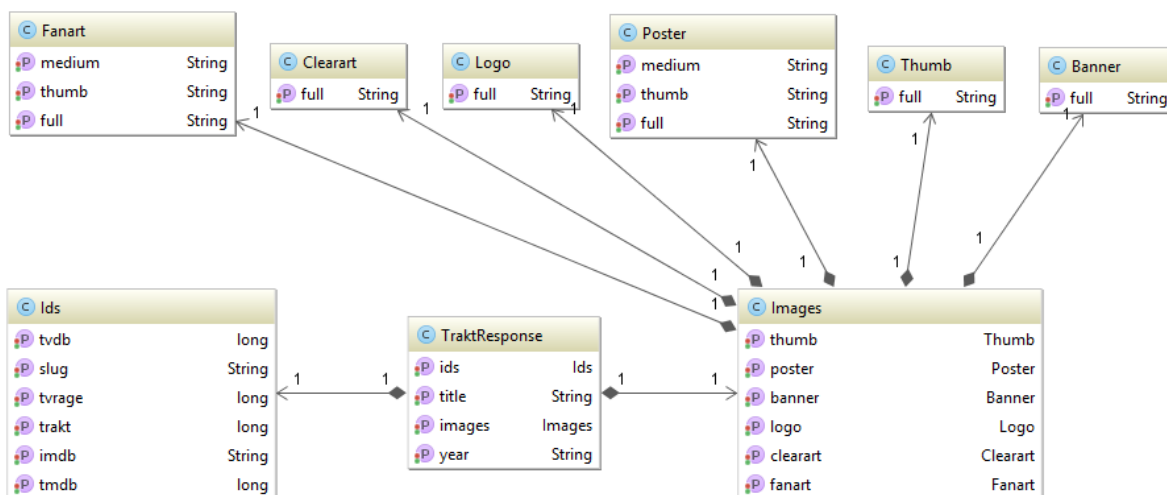
<https://api-v2launch.trakt.tv/shows/popular?page=1>

Populaarsete telesarjade päringu keskmine reaktsioonaeg: 0,1623 sekundit

Populaarsete telesarjade päringu sisendparameetrid ja vastuse formaadi klassidiagramm on esitatud järgnevalt tabelis 8 ja joonisel 13.

Tabel 8. *Trakt.tv populaarsete telesarjade sisendparameetrid*

Parameeter	Tüüp	Formaat või väärtus	Kohustuslik
Content-Type	HTTP päis	application/json	jah
trakt-api-version	HTTP päis	2	jah
trakt-api-key	HTTP päis	numbrite ja tähtede kombinatsioon	jah
page	URL parameeter	positiivne täisarv	ei
limit	URL parameeter	positiivne täisarv	ei
extended	URL parameeter	sõned: min, images, full, metadata	ei



Joonis 13. Trakt.tv populaarsete telesarjade vastuse klassidiagramm

Telesarjade otsingu näitesisend:

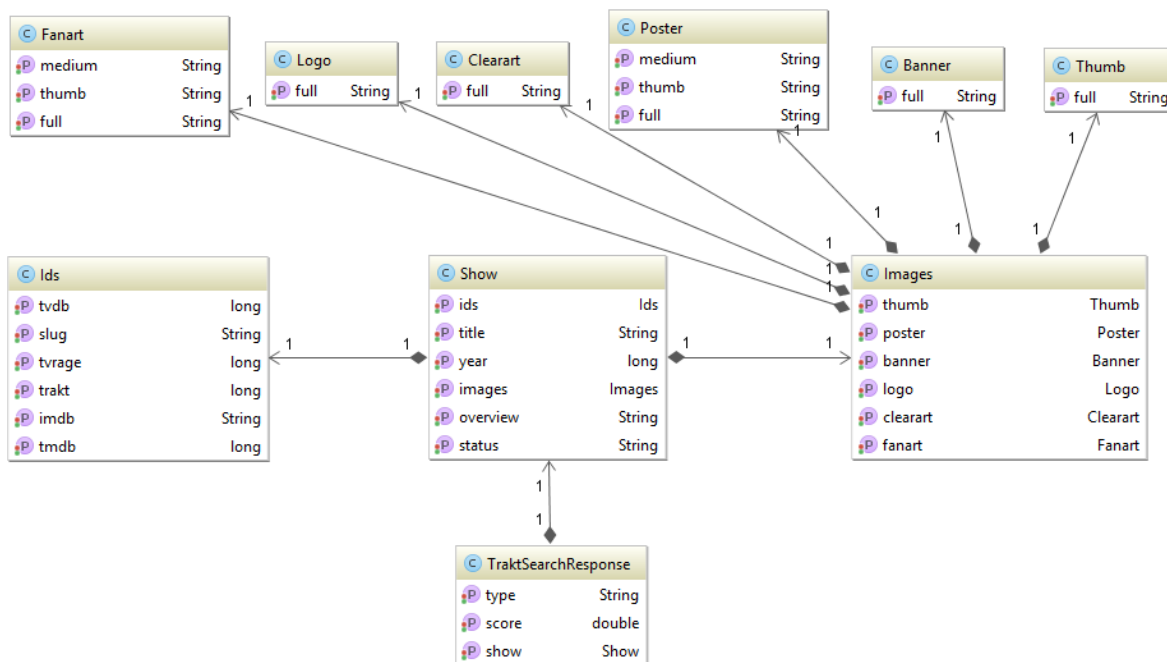
<https://api-v2launch.trakt.tv/search?query=cop&type=show>

Telesarjade otsingu keskmine reaktsioonaeg: 0,1655 sekundit

Telesarjade otsingu sisendparameetrid ja vastuse formaadi klassidiagramm on esitatud järgnevalt tabelis 9 ja joonisel 14.

Tabel 9. Trakt.tv telesarjade otsingu sisendparameetrid

Parameeter	Tüüp	Formaat või väärtus	Kohustuslik
Content-Type	HTTP päis	application/json	jah
trakt-api-version	HTTP päis	2	jah
trakt-api-key	HTTP päis	numbrite ja tähtede kombinatsioon	jah
query	URL parameeter	CGI vormindatud sõne	jah
type	URL parameeter	Telesarjade jaoks väärtus: show. Teised väärtused: movie, episode, person, list.	ei
year	URL parameeter	neljakohaline aasta number	ei
page	URL parameeter	positiivne täisarv	ei
limit	URL parameeter	positiivne täisarv	ei



Joonis 14. Trakt.tv telesarjade otsingu vastuse klassidiagramm

Telesarja detailide näitesisend:

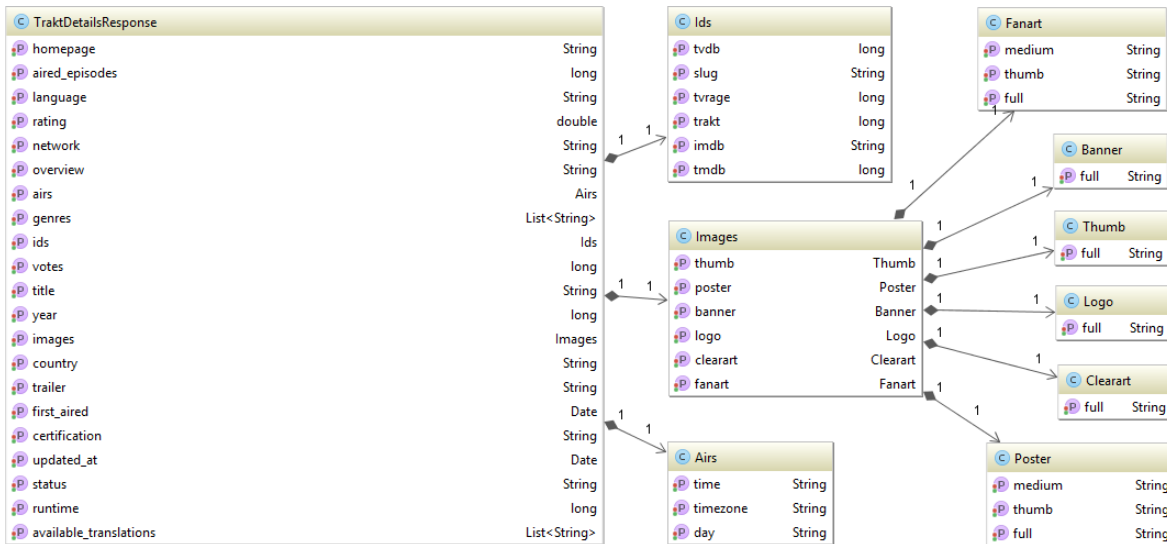
<https://api-v2launch.trakt.tv/shows/tt0944947>

Telesarja detailide päringu keskmine reaktsioonaeg: 0,1622 sekundit

Telesarja detailide sisendparameetrid ja vastuse formaadi klassidiagramm on esitatud järgnevalt tabelis 10 ja joonisel 15.

Tabel 10. Trakt.tv telesarja detailide sisendparameetrid

Parameeter	Tüüp	Formaat või väärtus	Kohustuslik
Content-Type	HTTP päis	application/json	jah
trakt-api-version	HTTP päis	2	jah
trakt-api-key	HTTP päis	numbrite ja tähtede kombinatsioon	jah
extended	URL parameeter	sõned: min, images, full, metadata	ei



Joonis 15. Trakt.tv telesarja detailide vastuse klassidiagramm

4.1.3. Muud liidesed

Olugugi et antud rakenduse loomiseks järgnevad rakendusliidesed ei sobi, tuleks eraldi välja tuua ka Rotten Tomatoes ning The Open Movie Database.

Rotten Tomatoes näol on samuti tegu paindliku liidesega filmide informatsiooni pärimiseks. Rõhk on siinjuures just sõnal film, sest antud liidesel puudub telesarjade pärimise tugi ning seetõttu jäi see ka täpsemast vaatlusest kõrvale.

The Open Movie Database rakendusliidesel seevastu on küll nii filmide kui telesarjade tugi, kuid pakutakse vaid n-ö otsingut pealkirja ja väliste identifikaatorite järgi. Pole aga võimalust pärida korraga nimekirja näiteks hetkel populaarsetest telesarjadest. Antud põhjusel jäi ka tolle rakendusliidese täpsem uurimine sinnapaika.

4.1.4. Valiku kriteeriumid ning sobivaima liidese valik

Sobivaima liidese valiku kriteeriumite püstitamisel lähtutakse süsteemianalüüsi faasis püstitatud funktsionaalsetest ja mittefunktsionaalsetest nõuetest. Täpsemalt on oluline, et telesarjade rakendusliides võimaldaks järgnevat funktsionaalsust:

- populaarsete telesarjade nimekirja vaatamine (FR-008);
- telesarja detailandmete vaatamine (FR-009);
- telesarja nime järgi otsimine (FR-014).

Lisaks ei tohi rakendusliidese päringud võtta liiga kaua aega (Soorituse tõhusus/ajaline käitumine [NFR-001]).

Eespool võrreldud liidestustest pakuvad soovitud funktsionaalsust mõlemad, nii The Open Movie Database kui ka Trakt.tv. Lisaks täidavad mõlemad liideseid ka ajalise käitumise mittefunktsionaalse nõude. Keskmiselt on päringud Trakt.tv-sse natukene kiiremad (0.16s vs 0.23s), kuid vahe on üpris minimaalne. Seega sobivad rakenduse loomiseks mõlemad liidestused ning parima valikuks peab kaaluma muid lisafaktoreid.

TMDB eeliseks on see, et detailandmete päringu vastus tagastab rohkem andmeid. Samas on ka Trakt.tv vastavate andmete leidmise päring üpriski detailne ning loodava lahenduse jaoks täiesti piisav. Lähtudes sellest, et rakendus disainitakse kolmekihilise arhitektuurina, kus telesarjade päringuid teostab rakendusserver, siis võib TMDB kasutamisel komistuskiviks osutuda 30 päringu limiit ühelt IP aadressilt. TMDB liidest kasutades tuleks rakendus disainida selliselt, et telesarjade päringute eest vastutaks kliendi veebibrauser, Trakt.tv puhul on võimalikud mõlemad variandid. Trakt.tv kasuks räägib lisaks veel see, et kõik päringud tagastavad identifikaatorite objekti ning võimaldab seega telesarjade vasteid leida mitmest erinevast süsteemist. Antud funktsionaalsus võib kasuks tulla näiteks telesarjade ja *torrent*ite sidumisel. Trakt.tv liides pakub peale selle ka hulganisti muid võimalusi ning on seega perspektiivikam. Näiteks võib loodavat rakendust hiljem täiendada selliselt, et kasutajad saaksid sellega siduda oma Trakt.tv konto. Kõike eelnevat arvesse võttes tuleb tõdeda, et Trakt.tv liides on antud süsteemi loomiseks parem valik.

4.2. Torrenti otsingu rakendusliideseid

Leidmaks mõistliku formaati süsteemi liidestamiseks välise *torrent* allikatega vaadeldakse antud peatükis populaarseimate teenusepakkujate otsingu vastuseformaate. Täpsemal teenuste analüüsil ei peatuta. Tuleb tõdeda, et käsitletavate teenuste poolt tagastatud sisu legaalsus on kaheldav, kuid selle eest annavad liideseid üpriski hea aimduse sellest,

milline võiks loodava rakenduse poolt kasutatav suhtlusformaad välja näha. Nagu juba eespool mainitud, siis sisestab vastava teenuse URL-i kasutaja ning see pole süsteemis eelnevalt defineeritud. Näiteks võib rakenduse kasutaja luua oma *torrent* informatsiooni teenuse, millega loodav lahendus peab suutma suhelda. Antud alajaotise tulemusena defineeritakse suhtlusformaad, millele väline *torrent* teenus peab vastama.

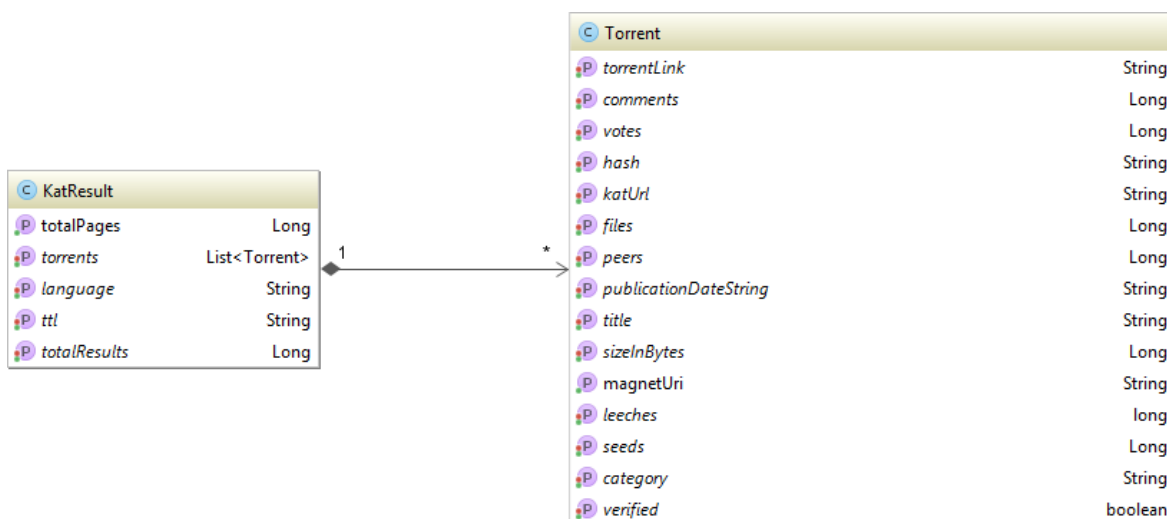
4.2.1. KickassTorrents

KickassTorrents (lühendatult KAT) näol on tegemist ühe populaarseima *torrent* otsingumootoriga maailmas, mida külastab igapäevaselt üle miljoni kasutaja. Teenuse pakkuja enda väitel ollakse kooskõlas autoriõiguse seadustega ning kõik kasutajate poolt raporteeritud ebaseaduslikud failid eemaldatakse. Lühidalt võib veebilehe olemuse kokku võtta lausega: „Juriidiliselt on kõik korrektne“. Lähtudes eetikast võib aga üpriski kindlalt väita, et tegeletakse piraatlusele kaasaaitamisega.

KAT võimaldab IMDB identifikaatori järgi otsida ainult filme, telesarjade *torrente*id on võimalik otsida vaid sõne põhiselt, s.t. leitakse kõik need *torrentid*, mille nimetuses sisaldus kasutaja poolt sisestatud väärtus. Järgnevalt on toodud otsingu näitesisend koos vastuse formaadi ning klassidiagrammiga (vt joonis 16).

Otsingu näitesisend: `https://kat.cr/json.php?q=daredevil`

Otsingu vastuseformaad: JSON



Joonis 16. KickassTorrents torrent otsingu vastuse klassidiagramm

4.2.2. Strike

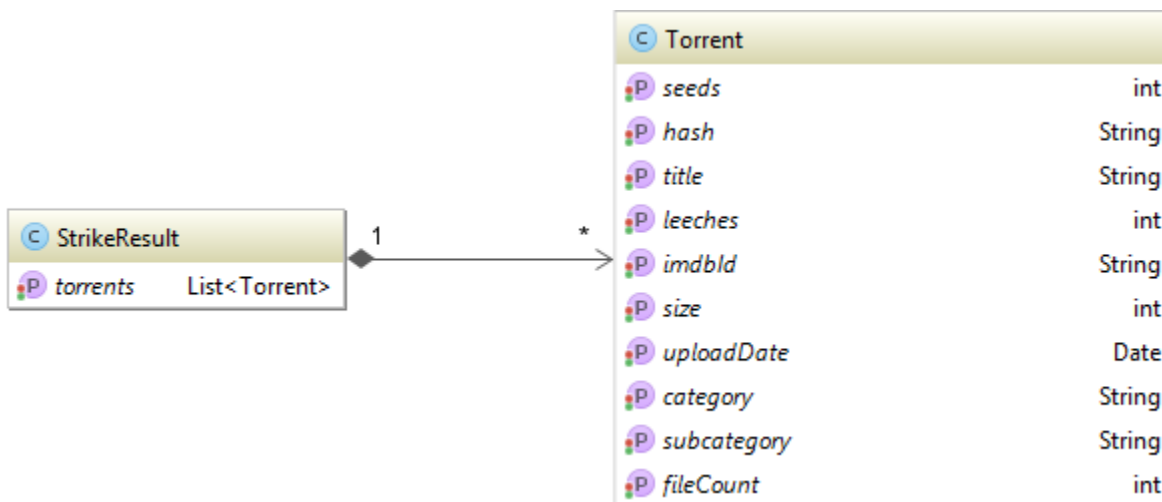
Nagu KickassTorrents oli ka Strike üpriski populaarne *torrent* otsingumootor. Igapäevane kasutajate arv oli samas suurusjärgus ning ka kasutustingimused väga analoogsed.

26. jaanuari 2016 seisuga Strike otsingumootor suleti. Põhjuseks on toodud juriidilised probleemid. Pärast sulgemist muudeti otsingumootori lähtekood avalikuks ning kõigil huvilistel on võimalik seda iseseisvalt jooksutada.

Telesarjade otsing on KAT-le väga sarnane. IMDB identifikaatori järgi otsing puudub täielikult ning kõik toimub sõnepõhiselt. Järgnevalt on toodud otsingu näitesisend koos vastuse formaadi ning klassidiagrammiga (vt joonis 17).

Otsingu näitesisend: `http://getstrike.net/api/torrents/search/ab`

Otsingu vastuseformaad: JSON



Joonis 17. Strike torrent otsingu vastuse klassidiagramm

4.2.3. EZTV

EZTV näol pole otseselt tegu *torrent* otsingumootoriga. Nimetus viitab ühele populaarsemale telesarjade jaotusrühmale, mis tegutses aastatel 2005-2015. Olgugi et rühm on tegevuse ametlikult lõpetanud on EZTV nimetus endiselt mitmetel veebilehtedele kasutusel. Neist populaarseim on tõenäoliselt `http://eztv.ag/`, mille põhjal on loodud

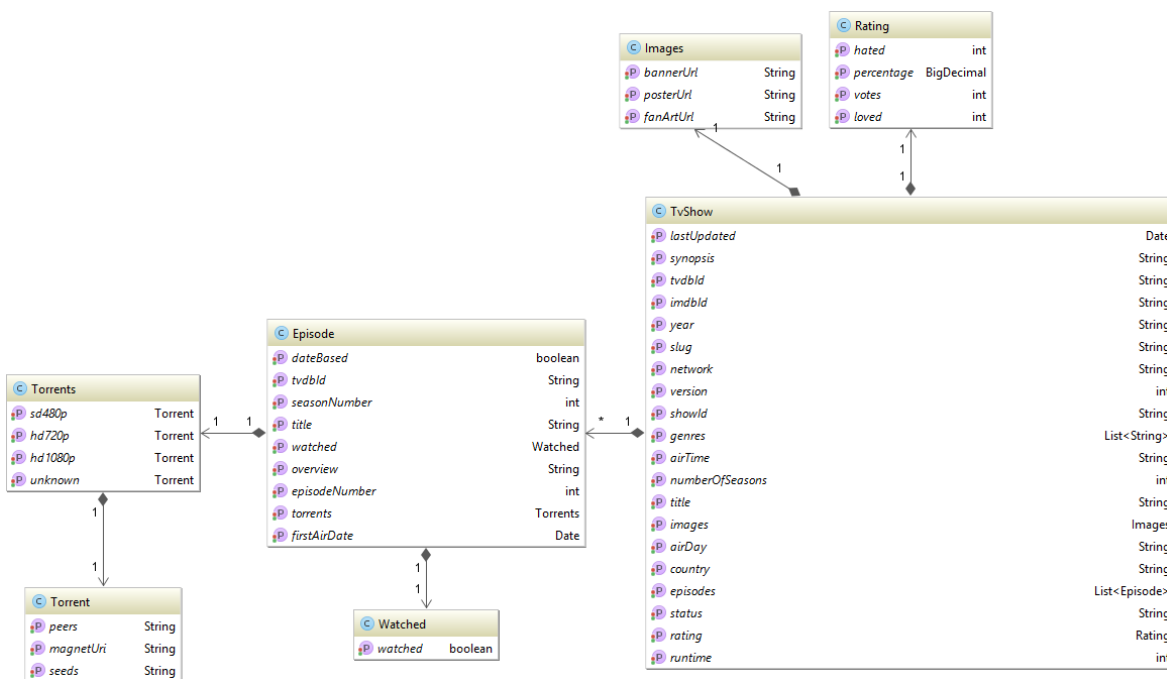
mitmeid teenuseid, mis indektseerivad igapäevaselt antud veebilehe sisu ning võimaldavad selle põhjal mugavat otsingut.

Taolist mähisteeki kasutab näiteks populaarne voogedastus rakendus Popcorn Time. Eri- nevalt KickassTorrentsist ja Strike’st on täielikult loobunud sõnepõhisest otsingust ning kõik toimib IMDB identifikaatori järgi. Järgnevalt on toodud otsingu näitesisend koos vastuse formaadi ning klassidiagrammiga (vt joonis 18).

Otsingu näitesisend:

`http://popcorntime.ws/api/eztv/show/tt1475582`

Otsingu vastuseformaad: JSON



Joonis 18. EZTV torrent otsingu vastuse klassidiagramm

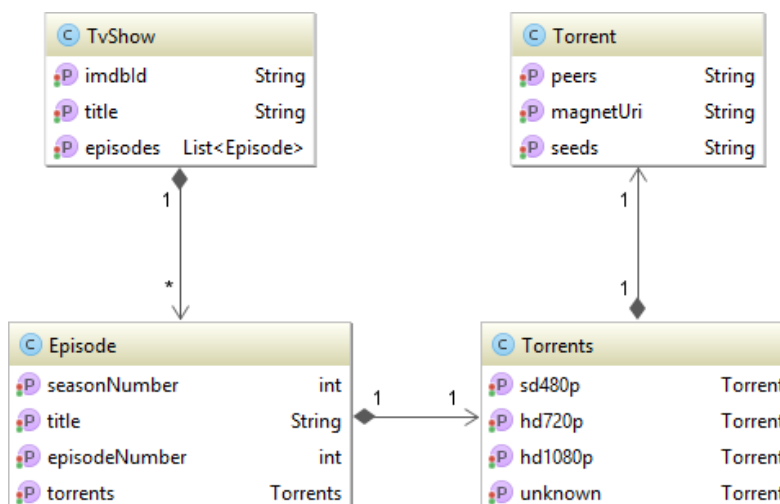
4.2.4. Formaatide võrdlus ning sobivaima defineerimine

Vaadates KAT ning Strike *torrent* otsingu vastuse klassidiagramme on näha, et need on üprisri sarnased. Mõlemad tagastavad n-ö nimekirja *torrent*itest, mis on mähitud eraldi- seisvasse objekti. Kuna otsingud on sõnepõhised, siis pole võimalik kindlalt eristada, kas tegu on just otsitava telesarjaga. Samuti puudub vastuses selgelt eristatav informatsioon sellest, millise episoodiga on tegu. Vastavad andmed on leitavad ainult pealkirjast. Lühi-

dalt kokkuvõttes tundub mõlema rakendusliidese peamiseks probleemiks olevat see, et vastus pole telesarja põhiselt grupeeritud.

Erinevalt kahest eelnevast on EZTV mähisliidese vastuseformaad märksa komplekssem ning põhineb täielikult IMDB identifikaatoril. Seega on tulemus usaldusväärsem ja võimaldab telesarjapõhise grupeerimise. Lisaks on vastus ka selgelt tükeldatud ning informatsioon paremini hoomatav. Tagastatakse alati üks kindel telesari koos kõigi episoodidega, millel *torrent* eksisteerib. Iga episood on eristatav, kas pealkirja või järjekorranumbri alusel. Boonuseks on veel võimalus *torrente* kvaliteedi järgi eristada. Kokkuvõtlikult võib öelda, et antud rakendusliides võimaldab leida ühe kindla telesarja kõikide episoodide *torrentid*, kui need eksisteerivad, ning eristada ka nende videokvaliteeti.

Lähtudes telesarjade liidese valimisel püstitatud kriteeriumist on ka *torrentite* puhul oluline IMDB identifikaatori olemasolu. Nimelt peab olema võimalik telesarju ja *torrente* adekvaatselt siduda. Üheks võimaluseks oleks küll ka pealkirjapõhine kokkuviiimine, kuid tihtipeale eksisteerib mitmeid sama nimega telesarju ning seega ei saa sidumise korrektsuses alati kindel olla. IMDB unikaalne identifikaator välistab taolised olukorrad. Vaadeldud rakendusliidestest vastab antud nõudele vaid EZTV, mille eeskujul defineeritakse ka loodava lahenduse poolt kasutatav *torrent* liidestuse suhtlusformaad. EZTV formaad on küll üpriski ideaalne, kuid kuna loodav süsteem saab telesarjade informatsiooni juba mujalt, siis pole kõiki tagastatavaid välja tarvis. Lähtuvalt eespool mainitust on joonisel 19 kujutatud loodava lahenduse *torrent* liidestuse suhtlusformaad. Eeldusel, et kasutaja on rakendusse sisestanud teenuse URL-i, oodatakse defineeritud formaadis vastust järgmiselt aadressilt: `http://TEENUSE_URL/show/IMDB_ID`.



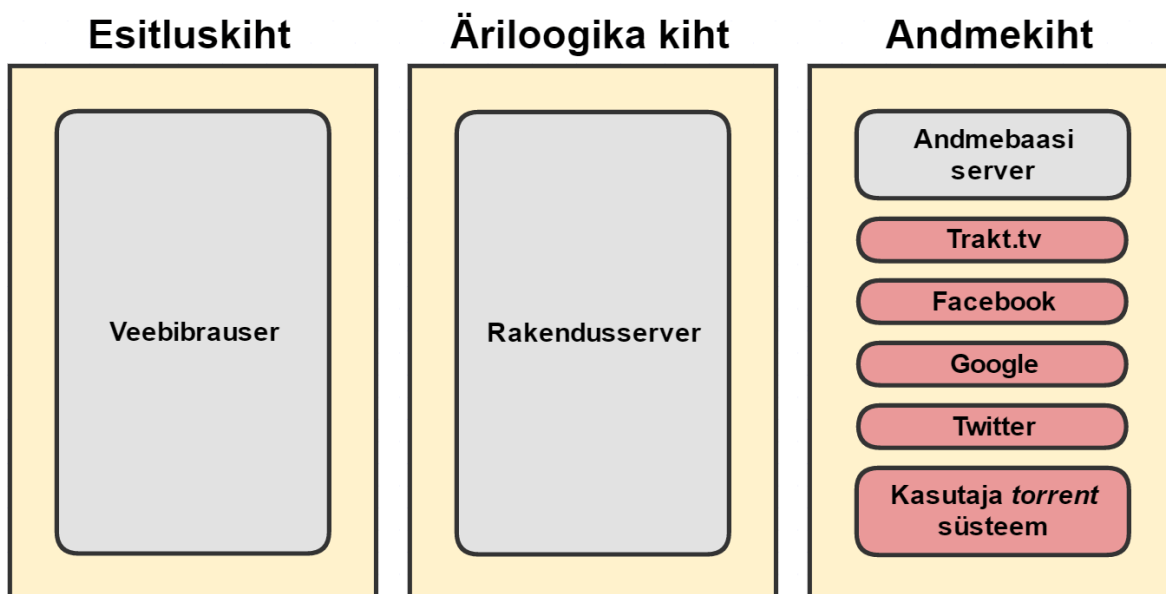
Joonis 19. Loodava lahenduse *torrent* liidestuse suhtlusformaadi klassidiagramm

5. Lahenduse realisatsioon

Antud alajaotises kirjeldatakse loodava süsteemi tehnilisi aspekte. Käsitletakse nii süsteemiarhitektuuri kui ka kasutatavaid tehnoloogiad. Lisaks disainitakse analüüsi faasis kirjeldatud andmebaas. Lõpetuseks esitatakse ka ülevaade loodud süsteemist ning selle võimalikest edasiarendustest.

5.1. Süsteemiarhitektuur

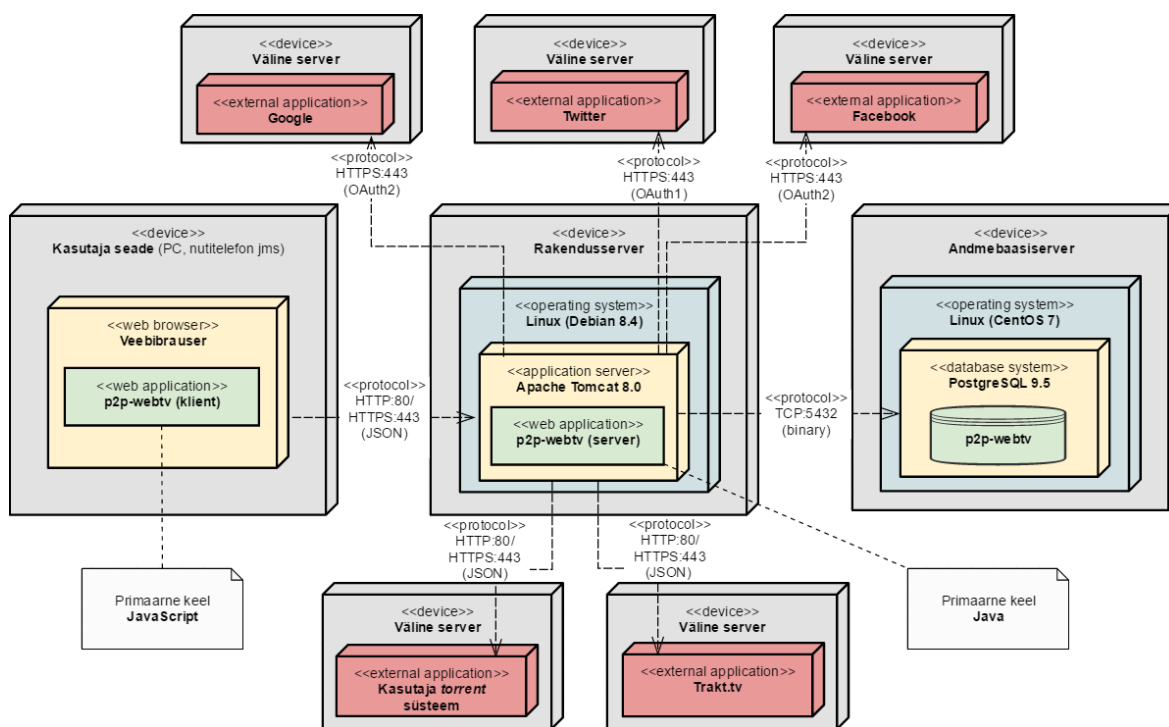
Süsteemi loomisel on aluseks võetud mitmekihilise arhitektuuri (ingl k *n-tier architecture*) mudel, milles andmete esitlemise, töötlemise ja haldamise protsessid on üksteisest eraldatud. Taoline mudel muudab tarkvara loomise paindlikumaks ning täienduste tegemise lihtsamaks. Lähtutakse mudeli laialt levinud kolmekihilise arhitektuuri (ingl k *three-tier architecture*) variatsioonist, kus iga kiht asub arvutivõrgus eraldiseisvas kohas. Kasutajale andmete kättesaadavaks tegemise eest vastutab esitluskihis asuv veebibrauser, mis suhtleb ärioloogikat pakkuva rakendusserveriga. Ärioloogika kiht saab andmed andmekihist, kus süsteemisene informatsioon talletatakse vastavas andmebaasi serveris. Lisaks kasutatakse ka välistest süsteemidest pärinevat informatsiooni (telesarjad, *torrentid*, sotsiaalvõrgu kasutajakontode integratsioon). Ülevaatlik mudel on toodud joonisel 20.



Joonis 20. Kolmekihilise süsteemiarhitektuuri mudel

5.1.1. Tehniline ülevaade

Joonisel 21 on toodud terviklahenduse struktuuri illustreeriv evitusdiagramm. Andmaks paremat ülevaadet iga komponendi eesmärgist on nende sisu ja olemus ka lühidalt lahti kirjutatud.



Joonis 21. Loodava lahenduse süsteemiarhitektuur

Järgnevalt on esitatud joonisel 21 toodud komponentide üldised eesmärgid:

- Andmebaasiserver – süsteemisene komponent, mis talletab selle toimimiseks vajalikku informatsiooni. Näiteks hoitakse andmebaasis teavet kasutajakontodest, kasutaja lemmiktelesarjadest ning süsteemiauditist.
- Rakendusserver – süsteemisene komponent, mis töötleb kasutaja poolt saadud HTTP päringuid. Rakendusserver vastutab süsteemi ärioloogilise toimimise eest suheldes nii sisese andmebaasiserveri kui ka väliste süsteemidega.
- Kasutaja seade – seade läbi mille süsteemi kasutatakse. Kogu süsteemi funktsionaalsus on kättesaadav veebibrauseri vahendusel ning seega sobivad seadmeteks sisuliselt kõik need, millel eksisteerib veebilehtiseja. Erandiks võivad olla vähemtuntud

või iganenud brauserid. Selleks, et kasutada süsteemi telesarja episoodi veebibrauseri kaudu vaatamise funktsionaalsust (FR-015) peab lehitsejal olema WebRTC tugi. Lisaks kasutatakse brauserit ka kasutaja poolt seadistatud *torrent* süsteemi URL-i talletamiseks küpsisena (ingl k *cookie*).

- Väline server (Google, Twitter, Facebook) – süsteemivälised komponendid/rakendusliidesed võimaldamaks süsteemi sisenemist sotsiaalvõrgu konto vahendusel.
- Väline server (Trakt.tv) – süsteemiväline komponent/rakendusliides, mille kaudu päritakse telesarjade informatsiooni. Tagab funktsionaalsed nõuded (FR-008, FR-009, FR-014).
- Väline server (Kasutaja *torrent* süsteem) – süsteemiväline komponent/rakendusliides, mille kaudu päritakse telesarja episoodide *torrent* informatsioon. Tagab funktsionaalsed nõuded (FR-015, FR-016). Antud teenuse URL-i seadistab kasutaja veebilehtiseja kaudu ning selle sisu peab vastama peatükis 4 defineeritud formaadile. Sisestatud väärtus talletatakse kasutaja veebibrauseri küpsisena (ingl k *cookie*).

5.1.2. Valitud platvormid ning nende alternatiivid

Programmeerimiskeeled

Loodava lahenduse süsteemiarhitektuuri diagrammilt (vt joonis 21) on näha, et süsteemi keskseks komponendiks on rakendusserver, mis peab suutma suhelda paljude erinevate teenustega. Seega on mõistlik valida serverikomponendi programmeerimiskeel selliselt, et liidestatus oleks võimalikult mugav. Andmebaasi ja REST arhitektuuri toe poolest sobiksid mitmed erinevad programmeerimiskeeled, näiteks Java, C#, Python, Go jms. Määravaks faktoriks osutub just väliste süsteemide liidestusteekide olemasolu. Nimelt eksistevivad mugavad sotsiaalvõrkude ning Trakt.tv liidestusteegid vaid Javal. Seega on lihtsuse huvides mõistlik kasutada just antud programmeerimiskeelt. Vastasel juhul tuleks ka välise süsteemide liideseid realiseerida nii-öelda käsitsi.

Rakendusserverid

Rakendusserveri tarkvara valimise eelduseks on Java platvormi tugi. Kuna serverikomponent peab sisuliselt olema vaid HTTP server Servlet API toega, siis pole n-ö täismahuliste rakendusserverite nagu JBoss või GlassFish kasutamine vajalik. Piisab üpriski primitiivsest rakendusserverist Apache Tomcat, mis täidab edukalt kõik rakendusele püstitatud nõuded ning mille kasutamine on oma piiratud funktsionaalsuse tõttu ka märksa mugavam. Ühelt rakendusserverilt üleminek ei tohiks osutuda ka ülemäära keerukaks (NFR-011).

Andmebaasisüsteemid

Loodava süsteemi suurem osa andmeid tuleb välistest süsteemidest ning seega on sisene andmebaas üpriski lihtsakoeline. Sellest tulenevalt on peamisteks andmebaasisüsteemide esitatavatest nõuteks jällegi lihtsus ning vabalt kättesaadavus, mis jätavad populaarsematel relatsioonilistel andmebaasisüsteemidest sõelale PostgreSQLi ning MySQLi. Nõutavat funktsionaalsust pakuvad mõlemad, kuid kuna MySQL ei vasta täielikult SQL standardile, siis võib hilisem üleminek mõnele teisele relatsioonilisele andmebaasisüsteemile osutuda tülikaks (NFR-011). Seega on PostgreSQL antud juhul perspektiivikam valik.

Operatsioonisüsteemid

Loodava süsteemi infraktstruktuur peab olema töökindel ja turvaline. Lisaks peab operatsioonisüsteem olema vabavaraline, mis välistab Microsoft Server tarkvara kasutamise. Seega jäävad valikusse erinevad Linuxi distributsioonid. Serveri operatsioonisüsteemidest on ühed populaarsemad ning töökindlaimad Debian ning CentOS, mistõttu kasutatakse neid ka antud süsteemi loomisel. Mitmekülgsuse tagamiseks on rakendusserveril kasutusel Debian 8.4 ning andmebaasiserveril CentOS 7.

Serveripoolsed tarkvarateegid

Loodava süsteemi serverikomponent on oma olemuselt n-ö täiemahuline Spring rakendus. Springi näol on tegu populaarseima Java rakendusraamistikuga, mis teeb veebipõhiste süsteemide loomise tavapärasest märksa lihtsamaks. Täpsemalt on antud süsteemi puhul kasutusel järgnevad Spring moodulid: Spring Boot (rakenduse tuumkomponent ja konfiguratsioon), Spring Security (autentimine ja ligipääsu kontroll), Spring MVC REST (REST teenuste tugi), Spring Data JPA (andmebaasiga suhtlemise lihtsutamine) ning Spring Social (Software-as-a-Service rakendusliidese pakkujate tugi). Lisaks Springile on kasutusel Hibernate, mis teeb võimalikuks andmebaasi tabelite ning vastavate Java objektide mugava sidumise (Object-relational mapping). Andmetöötluste ning JSON genereerimise lihtsutamiseks kasutatakse teeki nimega Jackson. Projekti kokkuehitamist ning teekide haldust lihtsustab tööriist Gradle.

Kliendipoolsed tarkvarateegid

Rakenduse kliendipoolne osa on ülesehitatud ühelehelise rakendusena (Single-page application), kus muudatused toimuvad veebilehte uuesti laadimata. Antud funktsionaalsust pakub JavaScript raamistik nimega AngularJS. Lisaks on kasutusel AngularJS alammodul Angular Translate, mis pakub mitmekeelsuse tuge. Ühtlase välimuse ning kasutajaliidese korrapära (NFR-005) eest vastutab Bootstrap veebiraamistik. Võimaldamaks torrentite voogedastust veebibrauseris (FR-015) üle WebRTC protokollil on kasutusel JavaScript teek WebTorrent. Kogu kliendipoolse rakenduse ehitab kokku ehitusautomaatika tööriist nimega Gulp, milles tarkvarateekide haldust pakub omakorda paketihaldusvahend Bower.

5.2. Andmebaasi disain

Järgnevas peatükis kirjeldatakse andmebaasi disaini põhimõtteid ning seda kuidas viia lahenduse analüüsi käigus pakutud kontseptuaalne andmemudel vastavusse valitud andmebaasisüsteemile (PostgreSQL 9.5).

5.2.1. Disaini üldised põhimõtted

Nagu juba eespool mainitud, siis hoitakse loodava rakenduse poolt andmebaasis üpris kiirga vähe informatsiooni ning suurem osa andmeid tuleb välistest süsteemidest, mida sisemise andmebaasi ümber ei kirjutata. Enamik rakenduse funktsionaalsust realiseeritakse rakendusserveris ning andmebaasi kasutatakse vaid kasutajaandmete, lemmikisarjade ning auditsündmuste talletamiseks. Näiteks kasutaja lemmikisarja lisamisel salvestatakse andmebaasi Trakt.tv-st päritud IMDB identifikaator, läbi mille on hiljem võimalik leida vastav sari ning selle *torrentid* ka teistest välistest süsteemidest.

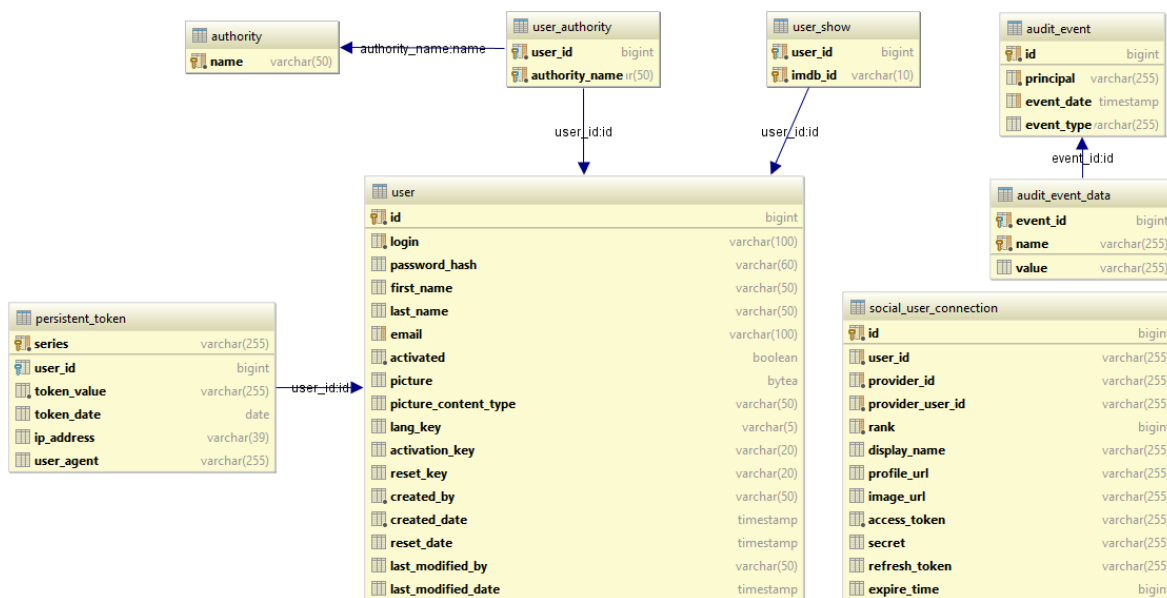
Andmebaasi disainimisel on lähtutud järgmistest põhimõtetest:

- salvestatakse vaid süsteemi toimimiseks hädavajalik informatsioon, mille aluseks on funktsionaalsed nõuded;
- andmete dubleerimise vältimiseks on tabelid viidud kolmandale normaalkujule;
- kõikidele kohustuslikele veergudele on lisatud NOT NULL kitsendus;
- veergudele, mis peavad olema süsteemi mõistes unikaalsed, kuid pole primaarvõtmed, on lisatud UNIQUE kitsendused;
- olemite terviklikkuse tagamiseks on lisatud PRIMARY KEY kitsendused;
- olemite sidumiseks on lisatud FOREIGN KEY kitsendused, millele rakendub ON DELETE CASCADE kompenseeriv tegevus (kõik kasutatavad olemid on sõltuvad);
- sõltuvatest tabelitest andmete leidmise kiirendamiseks on välisvõtme veergudele lisatud B-TREE INDEX;
- kuna enamiku ärireeglite kontrollid teostab rakendus, siis pole neid andmebaasis dubleeritud (puuduvad CHECK kitsendused).

Andmebaasi ja rakenduse vaheliseks suhtluseks on loodud eraldi andmebaasi roll, millel on vaid DML (Data Manipulation Language) õigused (GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES). Sellega tagatakse nii-öelda minimaalne ligipääs süsteemile ning välistatakse DDL (Data Definition Language) operatsioonide käivitamine rakenduse kaudu. Rakenduse enda poolt kasutatavad rollid asuvad vastavas andmebaasi tabelis ning nende õigused ja piirangud on realiseeritud rakenduse tasemel.

5.2.2. Andmebaasi struktuur

Andmebaasi projekteerimise aluseks on analüüsietaapis välja pakutud kontseptuaalne andmemudel ning eelnevalt välja toodud üldised andmebaasi disaini põhimõtted. Joonisel 22 on esitatud loodava lahenduse andmebaasi diagramm.



Joonis 22. Loodava lahenduse andmebaasi diagramm

Diagrammilt on näha, et puuduvad otsesed seosed tabelite *user* ning *social_user_connection* vahel. Nimelt leitakse erinevate sotsiaalvõrkude seosed kasutades erinevaid andmebaasi veerge. Twitter puhul on vastavuses tabeli *user* veerg *login* ning tabeli *social_user_connection* veerg *user_id*. Kuna Google ja Facebook liidesed ei paku kasutajanime välja, siis tuleb seosed leida tabeli *user* veerg *email* kasutades. Lisaks tuleb seoste leidmisel arvestatada ka duplikaatidega. Näiteks kui kasutaja on sidunud nii Facebook kui Google konto, mis on mõlemad registreeritud samale e-posti aadressile, siis eksisteerib *social_user_connection* tabelis 2 kirjet, mille *user_id* veerg on identne. Järelkult tuleb arvesse võtta ka liidese pakkuja identifikaatorit (*provider_id*).

Samuti pole kasutajaga otseselt seotud süsteemiauditid. Nimelt salvestatakse *audit_event* tabelisse ka näiteks ebaõnnestunud sisenemiskatsed koos proovitud kasutajanimega (veerg *principal*). Seega ei saa ka eksisteerida vastavat seost tabelis *user*.

5.3. Ülevaade loodud lahendusest

Antud töö käigus loodud veebipõhise telesarjade voogedastuse lahenduse loomine osutus oma keerukusastmelt oodatust lihtsamaks. Suuresti aitasid sellele kaasa õigesti valitud tarkvarateegid, mis hõlpsustasid oluliselt telesarjade informatsiooni pärimist (Trakt.tv) ning *torrent* voogedastust brauseris (WebTorrent). Antud funktsionaalsuste nii-öelda käitsi realiseerimine oleks töömahtu tõenäoliselt mitmekordselt suurendanud. Oma osa oli kindlasti ka minimalistlikul andmemudelil, mis täitis edukalt püstitud eesmärgid.

Valminud rakenduse lähtekood on avalikult kättesaadav (*p2p-webtv* 13.04.2016) ning litsentseeritud GNU GPLv3 alusel. Vastav litsents lubab lähtekoodi kopeerida, muuta ning levitada kõigil huvilistel, kuid nõuab ka muudetud programmi lähtekoodi avalikustamist. Vastav mudel julgustab kogukondlikult rakenduse täiendustesse panustamist ning kitsakohtade väljatoomist. Läbi selle võiks paraneda kvaliteet ning tekkida ka uusi ideid rakenduse funktsionaalsuse täiendamiseks. Litsents pakub ka kaitset omakasupüüdlike isikute ja ettevõtete eest.

Andmaks paremat ülevaadet loodud lahenduse keerukusest on läbi viidud ka lihtsakoeline mahuanalüüs, mis näitab erinevate faililaienditega failide hulka ning koodiridade arvu (vt tabel 11). Nagu näha, on keerukus kliendi- ja serverikomponendi vahel üpriski ühtlaselt jaotunud. HTML, CSS ning JS koodiread moodustavad ca 61% rakenduse kogumahu, serverikomponendi osa ca 38% ning andmebaasi loomiseks kasutatud SQL skriptid kõigest 1%. Kliendipoolse osa mõningane ülekaal tuleneb ühelehelise rakenduse olemusest, mis nõuab tavalisest märksa suurema mahu JavaScripti kirjutamist. Lisaks jäävad ka enamik püstitatud funktsionaalseid nõudeid kliendikomponendi realiseerida. Serveri pool vastutab eelkõige andmebaasist ning välistest andmeallikatest tulevate andmete töötlemise eest. Mõningast mahtu lisab serverikomponendile ka üldine konfiguratsioon ja turvalisuse tagamine. SQL vähene osakaal tuleneb minimalistlikkust andmemudelist.

Tabel 11. Loodud lahenduse mahuanalüüsi tulemused

Faililaiend	Failide arv	Koodiridade arv
css	1	784
html	27	2056
js	74	3116

Jätukub...

Tabel 11 – Jät kub...

Faili laiend	Failide arv	Koodiridu
java	60	3621
sql	2	121
Kokku	164	9698

Saavutatud tulemusega võib üldjoontes rahule jääda – püstitatud eesmärgid, funktsionaalsed ning mittefunktsionaalsed nõuded said suures osas täidetud. Nagu võis eeldada, siis tekkisid peamised tagasilöögid just WebTorrent tarkvarateegiga, mida kasutatakse veebipõhiseks *torrent* voogedastuseks. Tegu on üpriski uudse tehnoloogiaga ning seetõttu kogu funktsionaalsus veel ka ideaalselt ei tööta. Näiteks tekivad teatud video- ja audioformaatidega probleemid ning neid ei suudeta tagasimängida. Uute meediakonteinerite toe lisamine on väidetavalt töös. Teine probleem on vastavasisuliste *torrent*ite puudumine. Nimelt puudub enamikul populaarsetel *torrent*ite klientidel hetkel tugi jagamiseks sisu ka WebRTC-d kasutavatele veebibrauseri klientidele. Seega pole n-ö tavalistel *torrent*itel hetkel antud protokollis üldjuhul jagajaid. Positiivse küljena teeb see piraatsisu kasutamise loodud teenuses märksa raskemaks. Arvestades seda, et ülesandepüstitus on seatud selliselt, et *torrent*ite teenuse peab iga kasutaja iseseisvalt seadistama, siis on sisu loomine kasutajate kätes. Kõigile lisaks on alati võimalik kasutada *torrent*ite allalaadimise võimalust.

Autorile teadaolevalt samadel põhimõtetel ning tehnoloogiatel põhinevaid voogedastuse teenuseid ei eksiteeri. Telesarjade voogedastus partnervõrgus pole oma olemuselt midagi uut ning selleks eksisteerib mitmeid programme. Ükski neist ei tööta kahjuks ilma lisatarkvara alla laadimata ning kasutatakse TCP/UDP transpordiprotokolli. Loodud lahenduse uudsus seisneb eelkõige WebRTC protokolliga kasutamises, mis võimaldab platvormist sõltumatult *torrent*ite voogedastada veebibrauseri vahendusel ühegi pistikprogrammiga. Lisaks pakutakse igale kasutajale võimalust *torrent*ite teenust iseseisvalt konfigurida. Erinevalt paljudest analoogsetest rakendustest kasutatakse telesarjade ja *torrent* informatsiooni pärimiseks erinevaid väliseid teenuseid, mille vastused suudetakse vastavalt eeldefineeritud suhtlusformaadile siduda.

Lühidalt kokkuvõttes töötab loodud lahendus tehnoloogiliselt üpris hästi. Ainukeste probleemidena võib välja tuua mõningate meediaformaatide toe ning vastavasisuliste *torrent*ite teenuste puudumise.

5.4. Võimalikud edasiarendused

Antud töö käigus realiseeriti telesarjade voogedastusteenuse põhifunktsionaalsus ning nii-öelda teisejärgulised mugavusfunktsionaalsused jäid töö skoobist kõrvale. Rakenduse edasiarendamisel tasuks kindlasti taolise funktsionaalsuse lisamist kaaluda. Näiteks on süsteemi praeguses versioonis võimalik vaadata vaid hetkel populaarsete telesarjade nimekirja. Analoogselt võiks lisada reitingupõhise loendi. Lisaks teeks sobiva telesarja leidmise märka lihtsamaks see, kui nimekirju saaks kindlate kriteeriumite, nagu žanr või aasta, alusel filtreerida. Täiendavalt võiks süsteem pakkuda detailandmete kuval sama laadseid telesarju, mis aitaks kasutajal avastada enda lemmiksarjadele sarnast meediat. Lemmiksarjade põhjal võiks süsteem osata koostada ka nimekirja soovitustest.

Kindlasti võiks ära kasutada välise telesarjade liidese (Trakt.tv) täiendavaid võimalusi. Näiteks saaks Google, Facebook ning Twitter kontodega sisselogimise kõrvale lisada ka Trakt.tv toe. See võimaldaks süsteemi integreerida näiteks kasutaja Trakt.tv kollektsoonid. Täiendavalt saaks antud liidese kaudu episoode markeerida vaadatuks ning neid selle alusel peita. Peitmise funktsionaalsuse jaoks ei pea muidugi kasutama ilmtingimata välist liidestust ning selle saab realiseerida ka n-ö süsteemisiseselt. Antud juhul tuleks, aga tekitada täiendav andmebaasi tabel, mis hoiaks kasutajate ja episoodide seoseid. Trakt.tv liidestusest saaks veel ära kasutada telesarja hetkel vaatavate kasutajate nimistut, mis täiendaks süsteemi sotsiaalseid funktsioone. Näiteks võiks lisada üksteisele sõnumite saatmise võimaluse.

Olgugi et süsteem on algselt mõeldud just telesarjade vaatamiseks, pole tõenäoliselt väga keerukas lisada filmide tuge. Nimelt pakub Trakt.tv lisaks telesarjadele ka filmide informatsiooni. Seega oleks vastavate nimekirjade ja detailandmete kuvade koostamine analoogne ning vastav töömaht üpris väike. Arvatavasti tuleb küll kasutada telesarjadest erinevat *torrent* otsingu liidestust, mis oleks jällegi kasutaja enda poolt konfigureeritav. Kuna filmide puhul pole vaja leida episoode, siis oleks ka suhtlusformaad lihtsakoelisem. Näiteks saab aluseks võtta peatükis 4.2.1 toodud KickassTorrentsi vastuseformaadi ning muuta selle otsing IMDB identifikaatori põhiseks. Sarnaselt telesarjadega võiks eksisteerida populaarsete filmide nimekiri ning detailandmete kuva.

Enne eespool mainitud edasiarenduste realiseerimist tasub mõelda koodibaasi refaktooreerimisele. Kindlasti annab osa eksisteerivast funktsionaalsusest taaskasutatavaks muu-

ta. Näiteks saab olemasolevast populaarsete telesarjade nimekirjast teha nii-öelda meedia nimekirja komponendi, mida on võimalik hiljem kasutada ka filmide või näiteks reitingupõhise loendi koostamisel. Lisaks tuleks komponendiks teha ka voogedastuse modaalne aken, võimaldamaks seda taaskasutada rakenduse erinevates komponentides. Taoline koodibaasi ühtlustamine ja korrastamine kõrgendab kvaliteeti ning muudab täiendava funktsionaalsuse lisamise lihtsamaks.

Kokkuvõte

Antud magistritöö eesmärk oli luua teenus võimaldamaks telesarjade voogedastust partnervõrgus veebilehitseja vahendusel. Selleks tutvuti lähemalt BitTorrent protokolliga ja voogedastuse kontspetsioonidega ning jõuti järeldusele, et HTTP protokolliga piirangute tõttu BitTorrenti tavapärase TCP/UDP mudel ei sobi. Ainukese teadaoleva alternatiivina vaadeldi WebRTC-d ning seda kasutavat WebTorrent klienti. Lühidalt peatuti ka BitTorrenti legaalsusel ja eetilisel ning esitati nõuded loodavale lahendusele lähtudes legaalsusest ja eetilistest aspektidest. Seejärel viidi läbi lihtsustatud süsteemianalüüs ning võrreldi vajaminevaid väliseid andmeallikaid. Võrdluse tulemusena valiti välja sobivaim telesarjade rakendusliides ning pakuti välja suhtlusformaati kasutaja poolt konfigureeritavaks *torrent* liidestuseks. Analüüsi tulemustest lähtuvalt pandi paika rakenduse süsteemiarhitektuur ning töötati välja selle aluseks olev andmebaas. Lõpetuseks anti loodud lahendusest põgus ülevaade ning pakuti välja mõned võimalikud edasiarendused.

Töö olulisim tulemus on uudne brauseripõhine *torrent*-voogedastusteenuse prototüüp, mille lähtekood on avalikult kättesaadav ning litsentseeritud GNU GPLv3 litsentsi alusel. Seega on pakutud kontseptsiooni ja lähtekoodi võimalik aluseks võtta teiste sarnaste süsteemide loomisel. Autorile teadaolevalt teisi samadel põhimõtetel ja tehnoloogiatel põhinevaid voogedastusteenuseid ei eksiteeri. Lahenduse uudsus seisneb eelkõige WebRTC kasutamises ning erinevate väliste süsteemidega liidestuse pakkumises. Väline *torrentite* teenus on pealekauba iga kasutaja enda poolt konfigureeritav. Lisaks rakenduse väljarendamisel võrreldi töö käigus ka erinevaid telesarjade ning *torrentite* andmeallikaid ning toodi välja nende peamised tugevused ja nõrkused. Vastav võrdlus võib olla abiks teiste telesarjade või *torrent* informatsiooni vajavate teenuste arendamisel. Samuti saab töö käigus väljatöötatud domeenimudelit aluseks võtta väliste süsteemide, nagu sotsiaalvõrgud, telesarjad ja *torrentid*, liidestuse disainimisel.

Antud töös püstitatud uurimisülesanded said lahendatud. Seatud eesmärgid, funktsionaalsed ning mittefunktsionaalsed nõuded said täidetud, mis kinnitab valitud liideste ja domeenimudeli sobivust platvormist sõltumatu telesarjade voogedastusteenuse arendamiseks. Lahenduse peamised probleemid tulenevad pigem WebTorrent kliendi uudsusest. Nimelt puudub enamikul populaarsetel *torrenti* programmidel hetkel tugi jagamiseks ka WebRTC-d kasutavatele veebibrauseri klientidele. Seega on jagajate leidmine veel üpris problemaatiline. Lisaks puudub ka mõningate meediaformaatide tugi.

Summary

The purpose of this thesis was to develop a service for streaming TV series in peer-to-peer network via a web browser. To achieve this goal, BitTorrent protocol and streaming technologies were closely examined and it was concluded that due to the limitations of the HTTP protocol BitTorrent's regular TCP/UDP model is not suitable. As the only known alternatives, WebRTC technology and WebTorrent client, were studied. Briefly, BitTorrent's legal and ethical issues were discussed and the corresponding requirements for the solution presented. Next, simplified systems analysis was conducted and external data sources compared. The results were used to choose the best TV series API for the task and to propose a communication format for the user-configurable torrent interface. The resulting specification was used to establish application's system architecture and its underlying database. Finally, a short overview of the created application was given and several possible improvements proposed.

The main outcome of the thesis is a novel browser-based torrent-streaming service prototype whose source code is publicly available and licensed under GNU GPLv3. Thus, the proposed concept and source code can be used to create other similar systems. To the author's knowledge, other streaming services based on the same principles and technologies, do not exist. Novelty consists particularly in the use of WebRTC and in the offering of multiple interfaces to other systems. Furthermore, the external torrent service is configurable by each user. In addition to application development a variety of TV series as well as torrent data sources were compared and their major strengths along with weaknesses identified. The relevant comparison can be helpful for creating other services that require TV series or torrent information. Also, the developed domain model can be used as a basis for designing interfaces to communicate with external systems that offer social network, TV series and torrent material.

The research tasks outlined in the thesis were completed. The main objectives, functional and nonfunctional requirements were fully met, suggesting that the chosen interfaces and domain model are suitable for creating platform independent TV streaming services. The greatest problems of the proposed solution arise from the novelty of WebTorrent client. Namely, most torrent software does not share the content to web browser clients that use WebRTC. Thus, finding uploaders is currently quite problematic. Additionally, as of now, support for some media formats is missing.

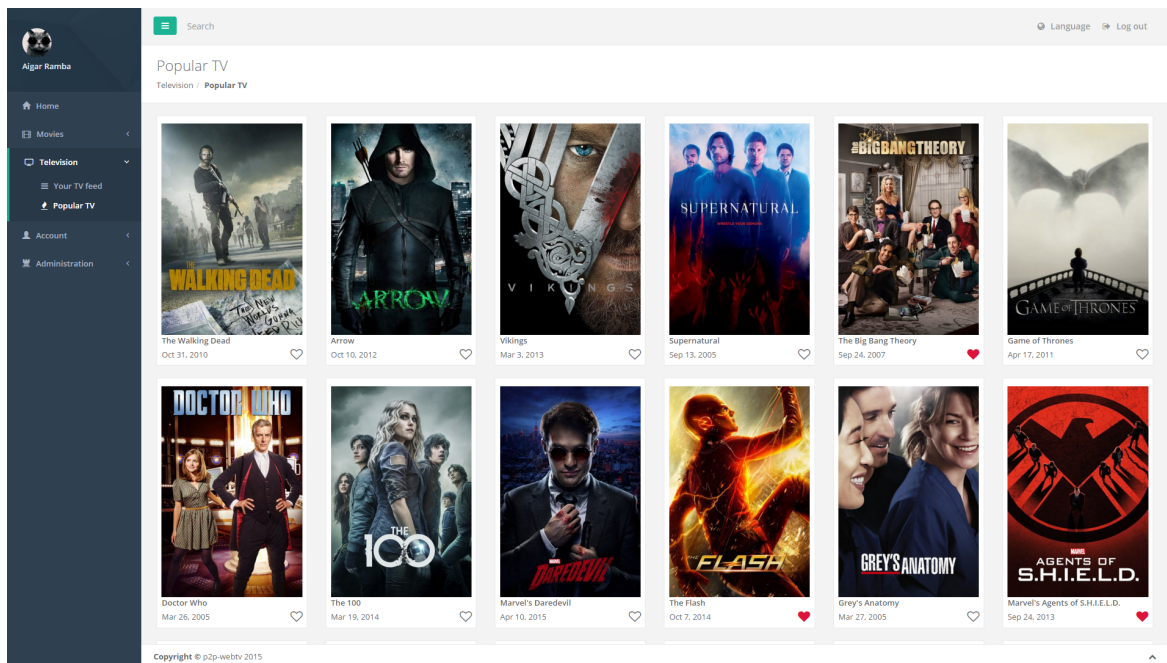
Kasutatud materjalid

- [1] F. Aboukhadijeh. *WebTorrent: Streaming torrent client for node.js and the web*. 2016. [WWW] <https://webtorrent.io/faq> (01.02.2016).
- [2] Adobe Dynamic Media Group. *A Streaming Media Primer*. 2001. [WWW] <http://www.anslab.iastate.edu/Class/Digital%20Images/Adobe%20Streaming%20Media%20Primer.pdf> (01.03.2016).
- [3] J. Bell. *Netflix bound by copyright rules, contracts with studios*. 2016. [WWW] <http://www.cbc.ca/news/business/canadian-netflix-copyright-law-1.3406074> (04.04.2016).
- [4] B. Cohen. *The BitTorrent Protocol Specification*. 2013. [WWW] http://www.bittorrent.org/beps/bep_0003.html (24.02.2016).
- [5] Creative Commons. *Creative Commons License*. 2001. [WWW] <https://creativecommons.org/licenses/> (04.04.2016).
- [6] M. Ereht, T. Ereht, ja K. Ross. *Eesti keele käsiraamat*. Tallinn: Eesti Keele Sihtasutus, 2007.
- [7] A. Fecheyr-Lippens. *A Review of HTTP Live Streaming*. 2010. [WWW] http://files.andrewsblog.org/http_live_streaming.pdf (01.03.2016).
- [8] Google Inc. *WebRTC Supported Browsers and Platforms*. 2016. [WWW] <https://webrtc.org/> (01.02.2016).
- [9] A. Johnston ja D. Burnett. *WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web*. Digital Codex LLC, 2013.
- [10] F. Kurose ja K. Ross. *Computer Networking: A Top-Down Approach*. Pearson, 2012.
- [11] G. McGath. *Basics of streaming protocols*. 2013. [WWW] <http://www.gary-mcath.com/streamingprotocols.html> (01.03.2016).
- [12] *p2p-webtv*. Get your personalized torrent feed and stream it using WebRTC. [WWW] <https://github.com/priitl/p2p-webtv> (04.04.2016).
- [13] Riigi Teataja. *Autoriõiguse seadus*. 2011.
- [14] Riigi Teataja. *Karistusseadustik*. 2014.
- [15] S. Saroiu, P. Gummadi, ja S. Gribble. *A Measurement Study of Peer-to-Peer File Sharing Systems*. Tech. rep. University of Washington, 2001.

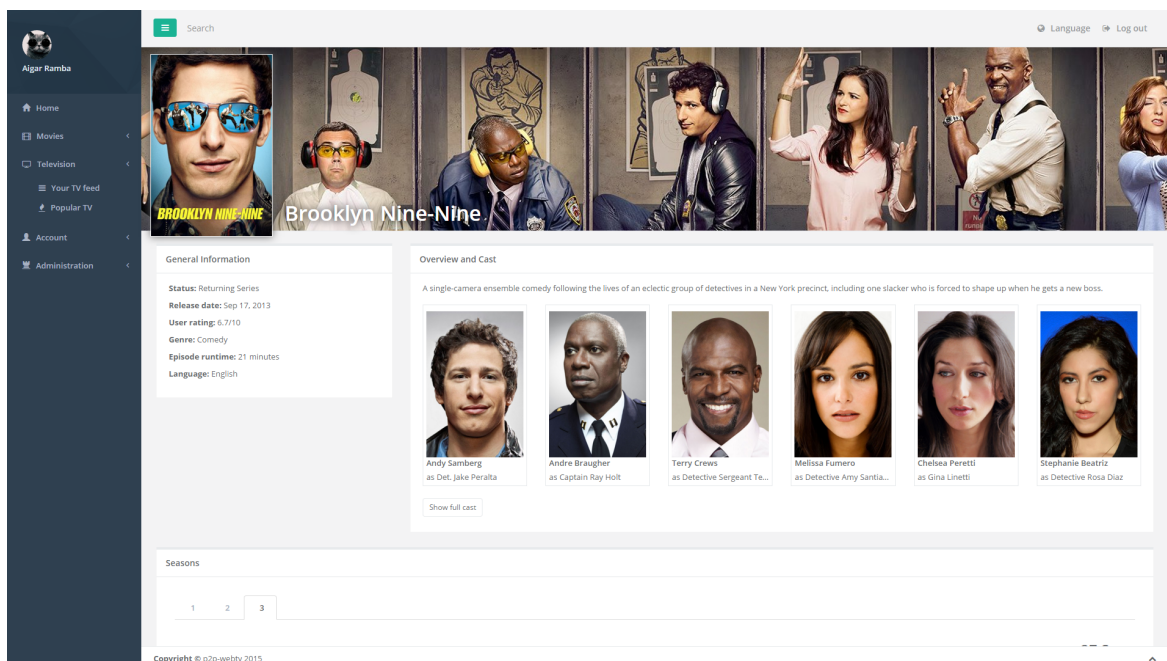
- [16] Süsteemide ja tarkvara kvaliteedinõuded ja kvaliteedi hindamine. *Süsteemide ja tarkvara kvaliteedimudelid: Eesti standard EVS-ISO/IEC 25010:2011*. Tallinn: Eesti Standardikeskus, 2012.
- [17] B. Templeton. *10 Big Myths about copyright explained*. 1995. [WWW] <https://www.w3.org/Legal/copyright-myths.txt> (04.04.2016).
- [18] S. Xuemin jt. *Handbook of Peer-to-Peer Networking*. Springer, 2010.

Lisa 1 – Ekraanitõmmised loodud rakendusest

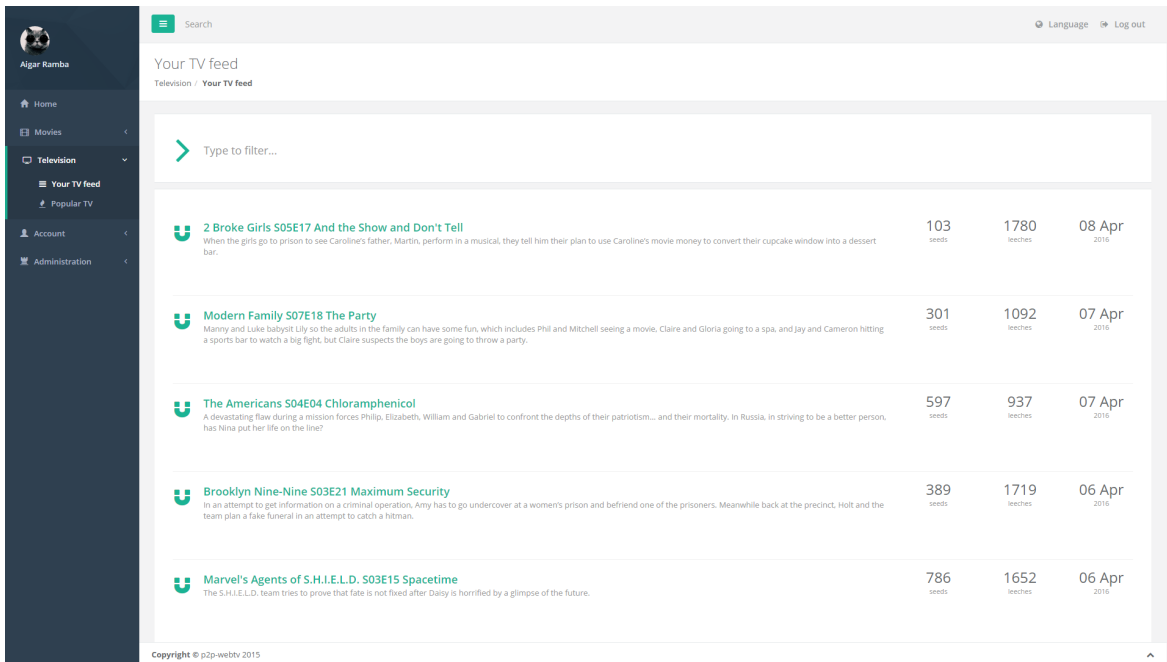
Järgnevalt tuuakse osade funktsionaalsete nõuete ekraanitõmmised loodud rakendusest.



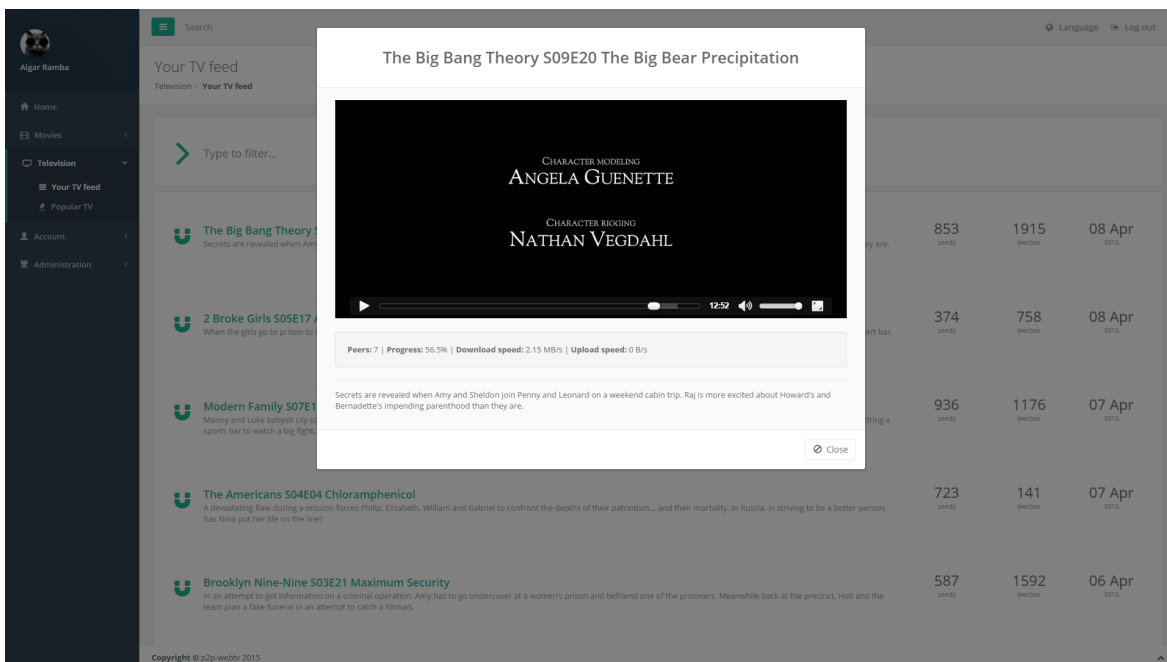
Joonis L1-1. Populaarsete telesarjade nimekirja vaatamine (FR-008)



Joonis L1-2. Telesarja detailandmete vaatamine (FR-009)



Joonis L1-3. Lemmiktelesarjade viimaste episoodide nimekirja vaatamine (FR-012)



Joonis L1-4. Telesarja episoodi veebibrauseri kaudu vaatamine (FR-015)

Users

Administration / User management

User management

Create a new user

ID	Login	Email	Status	Language	Profiles	Created date	Modified by	Modified date	
1087	lilab	gowevafa@yahoo.com	Activated	fr	ROLE_USER	08/04/16 19:45	admin	08/04/16 19:45	
1088	holocomb	zeserekoz@yahoo.com	Activated	en	ROLE_USER ROLE_ADMIN	08/04/16 19:45	admin	08/04/16 19:45	
1090	millerc	dupejcovu@yahoo.com	Activated	en	ROLE_USER	08/04/16 19:46	admin	08/04/16 19:46	
1089	henryb	fyfawag@gmail.com	Deactivated	en	ROLE_USER	08/04/16 19:46	admin	08/04/16 19:46	
1091	onealb	melamey@hotmail.com	Activated	fr	ROLE_USER	08/04/16 19:47	admin	08/04/16 19:47	
1092	shelonn	vicewju@gmail.com	Activated	en	ROLE_USER	08/04/16 19:47	admin	08/04/16 19:47	
1093	hyperadmin	mosefmoq@hotmail.com	Activated	fr	ROLE_USER ROLE_ADMIN	08/04/16 19:48	admin	08/04/16 19:48	
1094	chrisphi	girepia@yahoo.com	Activated	en	ROLE_USER	08/04/16 19:48	admin	08/04/16 19:48	
1095	brancher	wytuz@gmail.com	Deactivated	en	ROLE_USER	08/04/16 19:49	admin	08/04/16 19:49	
1096	cameronm	ketuse@yahoo.com	Activated	en	ROLE_USER	08/04/16 19:50	admin	08/04/16 19:50	
1097	harmoic	dawutefik@gmail.com	Deactivated	fr	ROLE_USER	08/04/16 19:50	admin	08/04/16 19:50	

Copyright © p2p-webtv 2015

Joonis L1-5. Registreeritud kasutajate nimekirja vaatamine (FR-017)

Audits

Administration / Audits

Filtrer par date

De 03/08/2016 à 04/09/2016

Audits

Date	Utilisateur	Status	Autres données
8 avr. 2016 14:47:57	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.132.5
7 avr. 2016 21:44:08	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.132.5
7 avr. 2016 20:35:16	testuser	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.132.5
6 avr. 2016 19:59:10	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.132.5
6 avr. 2016 19:43:43	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.132.5
6 avr. 2016 17:18:32	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.132.5
6 avr. 2016 15:10:57	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.132.5
6 avr. 2016 12:55:00	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.132.5
5 avr. 2016 21:38:57	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.132.5
4 avr. 2016 21:58:49	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.132.5
3 avr. 2016 17:06:19	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.132.5
2 avr. 2016 22:18:05	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.134.251
2 avr. 2016 19:25:01	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.134.251
1 avr. 2016 21:41:31	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.134.251
1 avr. 2016 21:41:28	admin	AUTHENTICATION_FAILURE	Bad credentials
1 avr. 2016 19:19:49	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.134.251
1 avr. 2016 17:44:06	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.134.251
31 mars 2016 22:04:24	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.125.226
31 mars 2016 12:34:43	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 217.71.47.60
30 mars 2016 21:22:18	admin	AUTHENTICATION_SUCCESS	Adresse Distante : 90.191.125.226

Copyright © p2p-webtv 2015

Joonis L1-6. Süsteemi sisselogimisauditi vaatamine (FR-018)

Lisa 2 – Päringu kiiruste mõõtmiseks kasutatav *bash* funktsioon

Allpool on toodud koodilõik, mida saab käivitada käsurealt päringu kiiruste ning nende keskmise reaktsiooniaja mõõtmiseks. Funktsioonile antakse ette kaks paremeetrit: 1) URL, mille vastuse saabumise kiirust soovitakse mõõta, 2) iteratsioonide arv mitu korda päringut käivitatakse. Tulemuseks prinditakse käsureale iga päringu reaktsiooniaeg ning kõigi päringute reaktsiooniaegade keskmine. Antud töös kasutatakse vastavat funktsiooni telesarja rakendusliideste kiiruste väljaselgitamiseks.

```
function response_time() {
    URL="$1" ITERATIONS="$2" TOTAL="0" i="0"
    while [ $i -lt $ITERATIONS ]; do
        CURRENT=`curl $URL -o /dev/null -s -w %{time_total}`
        CURRENT="${CURRENT}/./."
        echo "Current: ${CURRENT}"
        TOTAL=`echo "scale=4; $CURRENT + $TOTAL" | bc`
        i=$((i+1))
    done
    AVERAGE=`echo "scale=4; $TOTAL / $ITERATIONS" | bc`
    echo "-----"
    echo "Average: $AVERAGE"
}
```

Näiteks tagastab "*response_time http://www.ttu.ee/infotehnoloogia-teaduskond/ 5*" vastusena:

```
Current:    0.916
Current:    1.153
Current:    0.855
Current:    0.820
Current:    1.138
Average:    0.976
```