



TAMPEREEN TEKNILLINEN YLIOPISTO

JOONA JÄRVINEN
TILAUSVIDEOPALVELUN TOTEUTUS HBBTV-TEKNIIKOILLA
Diplomityö

Tarkastaja: Professori Tommi Mikkonen
Tarkastaja ja aihe hyväksytty Tieto- ja
sähkötekniikan tiedekuntaneuvoston
kokouksessa 4. syyskuuta 2014

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Tietotekniikan koulutusohjelma

JÄRVINEN, JOONA: Tilausvideopalvelun toteutus HbbTV- tekniikoilla

Diplomityö, 45 sivua, 5 liitesivua

Kesäkuu 2015

Pääaine: Ohjelmistotuotanto

Tarkastaja: professori Tommi Mikkonen

Avainsanat: HbbTV, VoD, Catchup TV, SmartTV, HTML5, CSS3, DVB

Televisioilla voidaan katsoa lineaarista televisiolähetystä. Internet on tuonut sen rinnalle myös PC:llä käytettäviä televisio- ja videopalveluita. Älytelevisioiden mukana on tullut mahdollisuus yhdistää televisio- ja internetpalvelut. Tämän diplomityön aiheena on HbbTV-tilausvideopalvelun toteutus.

Tilausvideopalvelu on sellainen palvelu, jonka avulla sen käyttäjä voi valita katseltavaksi haluamaansa videosisältöä milloin vain. Palvelu on siis sovellus, jolla voi katsoa televisio-ohjelmia tai muuta videosisältöä. Työssä esitellään ja vertaillaan erilaisia TV-sovellustyyppisiä ja -sovellusympäristöjä, sekä esitellään tekniikoita ja standardeja toteutuksen pohjaksi.

Tilausvideopalvelun tulee vastata sille asetettuihin toiminnallisiin-, sekä käyttöliittymävaatimuksiin. Toteutuksen tulee noudattaa HbbTV-standardia, sekä hyödyntää muita HbbTV:een liittyviä standardeja ja tekniikoita, kuten DVB ja MPEG DASH.

Tässä diplomityössä ratkaistaan HbbTV-tilausvideopalvelun toteutuksen vaatimat asiat ja vaatimukset. Työn kirjoittamisen aikana on toteutettu neljä eri HbbTV-tilausvideopalvelusovellusta kahteen eri asiakasprojektiin. Toteutetut käyttöliittymät noudattavat yhdenmukaista valikkorakenteista tietomallia joka mahdollistaa toteuttaa erilaisia näkymiä ja määrittää navigointipolut parametrisoinnin avulla. Sovellukset toteutetaan hyödyntämällä yhdenmukaista rajapintaa joka antaa parametreista riippuen eri palvelun videosisältöä metatietoineen.

HbbTV sopii hyvin tilausvideopalvelun toteutukseen TV-operaattorille. HbbTV on alustariippumaton joten toteutuksia ei tarvitse tehdä erikseen eri laitteille. Jakelutie on yksinkertainen ja sovellukset ovat helposti saatavilla televisiokanavan signaalista. HbbTV-tilausvideopalvelu soveltuu parhaiten TV-operaattorin oman maksullisen ja maksuttoman videosisällön jakamiseen.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Information Technology

JÄRVINEN, JOONA: Implementation of Video on Demand Service Using HbbTV Techniques

Master of Science Thesis, 45 pages, 5 Appendix pages

June 2015

Major: Software engineering

Examiner: Professor Tommi Mikkonen

Keywords: HbbTV, VoD, Catchup TV, SmartTV, HTML5, CSS3, DVB-T/C/S

Linear TV broadcast is available with a television. Internet has brought also TV and video services on PC. With Smart TVs it's possible to combine internet services and TV together. The topic of this thesis is implementing a Video on Demand (VoD) service on Hybrid Broadcast Broadband TV (HbbTV).

A VoD service is a service, which with user can select any available video content to watch any time. The service is an application, that let you choose what you want to see on TV. There is different types of TV-programs and TV-programming interfaces and standards introduced as the base of the thesis.

The implemented VoD service must face the declared functional and user interface requirements. The implementation must follow HbbTV standard, and utilize other standards and techniques linked to HbbTV, such as DVB and MPEG DASH.

In this thesis, the requirements of implementing a HbbTV VoD service have been met. During the writing of the thesis, there is implemented in total four different HbbTV VoD services in two different projects. The implemented applications utilizes an uniform menu data model that allows to implement different views and define navigation paths depending on attributes set. All the applications are implemented same way to utilize an uniform API that reveals video content and metadata of a specific service depending on parameters.

HbbTV is a suitable way to implement a VoD for TV operators. Because HbbTV is platform independent, there is no need to develop different applications for different devices. The distribution channel is simple and easy to access from broadcast signal. HbbTV VoD fits best for distribution of TV operators' own paid or free video content.

ALKUSANAT

Tämä opinnäytetyö on tehty Sofia Digital OY:lle diplomityönä ajalla heinäkuu 2014 – toukokuu 2015. Sofia Digital on kansainvälisesti arvostettu interaktiivisten video- ja televisioratkaisujen tuottaja, joka tuo erilaisiin televisiopäätelaitteisiin uusia lisäarvoratkaisuja ja -palveluita hyödyntämällä uusia digitaalitekologioita. Sofia Digital keskittyy avoimen standardin interaktiivisiin alustoihin, kuten IPTV- ja HbbTV-palveluihin sekä digitaalisten näyttöjen hallintajärjestelmiin. Sofia Digital on perustettu vuonna 2000, ja se palvelee operaattoreita, televisiokanavia ja laitevalmistajia yli 25 maassa. Sofia Digital tarjoaa asiakkailleen ratkaisuja digitaalisen TV-sisällön ja interaktiivisten palvelujen tuottamiseen ja toimittamiseen digibokseihin ja matkapuhelimiin. Asiakkaat ovat TV-operaattoreita ja TV-vastaanottimien valmistajia.

Tämä diplomityö käsittelee HbbTV-tilausvideopalvelun suunnittelua ja toteutusta TV-ympäristössä. Aloitin suunnittelu- ja ohjelmointityön heinäkuussa 2014 ja diplomityön kirjoittamisen heinäkuun loppupuolella. Työskentelin diplomityön tekijänä ja ohjelmistosuunnittelijana useammassa projektissa, joissa toteutetuista ratkaisuista kerron tässä työssä.

Kiitän professori Tommi Mikkosta diplomityöni ohjauksesta ja hyvistä kommentteista työn parantamiseksi. Kiitän työnantajaani Sofia Digitalia diplomityöaiheen antamisesta, joka sopi erittäin hyvin pääasiallisiin työtehtäviini. Kiitän myös kollegoitani Mika Kanervaa, Jukka Eklundia, Jussi Teirikangasta, Aki Niemistä, sekä Ville Holopaista, jotka ovat olleet vahvasti osallisena toteutettujen sovellusten suunnittelussa ja kehityksessä. Haluaisin kiittää myös vaimoani Saara Järvistä joka on osaltaan hyvin ymmärtäväisesti antanut minulle aikaa työn kirjoitusprosessiin.

11. toukokuuta 2015

Joona Järvinen

joona.jarvinen@sofiadigital.com

SISÄLLYS

1	Johdanto.....	1
2	TV-sovellukset ja tekniikat.....	2
2.1	Yleiskuvaus.....	2
2.2	TV sovellusympäristönä.....	3
2.3	Videokuvaan sisällytetty näkymä.....	4
2.4	Natiivi ja hybridi TV-ohjelmisto.....	5
2.5	IPTV.....	6
2.6	HbbTV.....	6
2.6.1	Saatavuus.....	6
2.6.2	Päätelaitteet.....	7
2.6.3	Ohjelmointitekniikat.....	8
2.7	Muiden laitteiden TV/Video -palvelut.....	8
3	Standardit ja organisaatiot.....	9
3.1	DVB.....	9
3.2	HbbTV.....	12
3.2.1	CE-HTML.....	14
3.2.2	Tiedonsiirto ja AIT.....	14
3.2.3	Video ja Broadcast.....	15
3.2.4	HbbTV-sovelluksen loogiset tasot.....	17
3.2.5	Stream eventit.....	18
3.2.6	OIPF DAE.....	19
3.2.7	Suunnitteluohjenuorat.....	20
3.2.8	Testaus ja yhteensopivuus.....	21
3.3	MPEG DASH.....	21
3.4	DRM-Tekniikat.....	23
3.5	Käyttäjien tunnistaminen.....	24
4	Tilausvideopalvelu.....	25
4.1	Yleinen kuvaus.....	25
4.2	Vaatimukset.....	26
4.2.1	Toiminnalliset vaatimukset.....	27
4.2.2	DRM-Vaatimukset.....	27
4.2.3	Autentikointi.....	28
4.2.4	Ohjelmistosisältöjen alueelliset oikeudet.....	28
4.2.5	Käyttöliittymä.....	29
5	HbbTV-Toteutus ja ratkaisut.....	30
5.1	Projektit.....	30
5.2	Sidosryhmät.....	30
5.3	Toteutus.....	31

5.3.1	Sovelluksen käynnistys.....	31
5.3.2	Tilausvideopalvelun käynnistyminen ja suoritus.....	33
5.3.3	Sovelluksen arkkitehtuuri.....	33
5.3.4	Asiakassovellus ja käyttöliittymä.....	35
5.3.5	Käytettävyys ja käyttäjäkokemus.....	37
5.3.6	Palvelinpään toteutus.....	37
5.3.7	Yhteensopivuusongelmat ja ratkaisut.....	39
5.3.8	Muut toteutetut ja tulevat ratkaisut, sekä vaihtoehdot.....	41
5.3.9	HbbTV:n vaihtoehdot.....	41
6	Arviointi.....	42
6.1	Tuotantoon siirtyneet palvelut.....	42
6.2	HbbTV:n soveltuvuus tilausvideopalvelun toteutukseen.....	43
7	Yhteenveto.....	45
	Lähteet.....	46
	Liite 1: Esimerkki: MPEG DASH DRM.....	50
	Liite 2: PlayReady -lisenssin haku.....	53

TERMIT JA NIIDEN MÄÄRITELMÄT

AIT	Application information Table, DVB-siirtovirrassa signaloitu sovellustietotaulu joka pitää sisällään tietoja yhden DVB-palvelun HbbTV-sovelluksista.
API	Application Programming Interface, Sovellusohjelmointirajapinta asiakassovellusten ja palvelinsovellusten väliseen kommunikointiin.
CEA	Consumer Electronics Association, Kulutuselektronikkalaitteyhdistys. Osallistuu toimialan standardien kehitykseen.
CE-HTML	Consumer Electronics Hypertext Markup Language, Kulutuselektronikkalaitteille suunnattu HTML-variantti.
CENC	Common Encryption. Yleinen kryptausmenetelmä, joka mahdollistaa useamman DRM-menetelmän käytön.
Connected TV	Internetiin yhdistetty TV, Usein Smart TV tai älytelevisio.
DAE	Declarative Application Environment, Määrittelevä sovellusympäristökehys.
DASH	Dynamic Adaptive Streaming over HTTP, HTTP-protokollan mukainen kaistanleveyden ja videosoitinkomponentin vaatimusten mukaan muuntautuva videon suoratoistostandardi.
DRM	Digital Rights Management, Digitaalisen immateriaalin käyttöoikeuksien hallinta.
DSM-CC	Digital storage media command and control. Tiedostopohjaisen tiedon synkroninen lähetyksenmekanismi.
DVB	Digital Video Broadcasting. Digi-TV -standardi, joka määrittelee televisiolähetyksen siirtovirran rakenteen.
ETSI	European Telecom Standards Institute. Eurooppalainen tieoliikennetoimialan standardisointiorganisaatio.
HbbTV	Hybrid Broadcast Broadband Television. Standardi, joka määrittelee OIPF-DAE:n pohjalta miten TV-lähetyksen mukana tulevat sovellukset saadaan käyttöön päätelaitteissa.
MPEG	Moving Picture Experts Group. Standardointiryhmä, jonka tehtävänä on määritellä audiovisuaalisen tiedonsiirron standardeja.
MPEG TS	MPEG Transport Stream. MPEG-standardin mukainen siirtovirta, jota käytetään DVB-lähetyksissä.
OIPF	Open IPTV Forum. Lukuisista televiestinnän ja teknologian alan yrityskistä koostuva konsortio, jonka tehtävänä on kehittää standarnin mukaisia teknisiä määrittelyjä.
OIPF-DAE	Open IPTV Forum - Declarative Application Environment. OIPF:n määrittelemä kulutuselektronikkalaitteiden sovellusympäristö.

OITF	Open IPTV Terminal Function. Natiivin toteutuksen rajapintafunktio joka on määritelty OIPF-DAE:ssa.
PMT	Program Map Table. DVB-standardin mukainen ohjelmatietotaulu siirtovirrassa.
SAAS	Software As A Service, ohjelmistopalvelu, jossa ohjelmiston avulla on toteutettu jokin palvelu loppukäyttäjälle.
Smart TV	Älytelevisio. Televisio, jossa voidaan suorittaa erilaisia asennettavia tai valmiiksiasennettuja viihteellisiä sovelluksia, tai käyttää HbbTV-soveluksia.

1 JOHDANTO

Tilausvideopalvelu (Video On Demand) on nimensä mukaisesti sellainen palvelu, josta käyttäjä voi valita haluamansa videosisällön katsottavaksi milloin vain. Sisältö voi olla maksullista tai maksutonta. Tilausvideopalvelun tilaaja ja sisällöntuottaja on tässä työssä käsitellyissä projekteissa aina joko julkinen tai kaupallinen televisio-operaattori. Tilausvideopalvelu voidaan toteuttaa TV-laitteelle joko laitekohtaisena natiivisovelluksena tai HbbTV-tekniikoilla. HbbTV-tekniikoilla tarkoitetaan sovellusohjelmointitekniikoita, joita tarvitaan HbbTV-ohjelmoinnissa: CE-HTML, CSS, JavaScript ja AJAX.

Tässä diplomityössä tutkitaan erilaisia TV-sovellustyyppejä, TV-sovelluksien ohjelmointitapoja, sekä toteutetaan HbbTV -standardin mukaisilla tekniikoilla tilausvideopalveluita, jotka kuuluvat asiakasprojekteihin. Työssä vertaillaan erilaisia TV-sovellustyyppejä ja niiden vahvuuksia, heikkouksia, rajoitteita ja mahdollisuuksia yleisemmällä tasolla, sekä määritellään, miten HbbTV-standardia tukevalle Smart TV:lle pystytään toteuttamaan tilausvideopalvelu siihen liittyvien muiden sovellusten ja ratkaisuiden kanssa kokonaisuudessaan, sekä miksi HbbTV -ratkaisu on erilaisista TV-sovellustyypeistä valittu parhaaksi kyseisenlaisen palvelun toteutuksessa.

HbbTV mahdollista TV-kanavakohtaisen tilausvideopalveluiden toteutuksen osaksi TV-lähetystä lisäpalveluna. Koska sovellukset ovat televisiokanavakohtaisia, HbbTV soveltuu hyvin sovellusalustaksi asiakasyrityksille, jotka hallinnoivat ja lähettävät televisio-ohjelmistoja. Työn kirjoittamisen ohessa on valmistunut kahteen eri asiakasprojektiin kuuluvat HbbTV-tilausvideopalvelut, jotka ovat julkisesti käytössä.

Luvussa 2 määritellään tämän työn konteksti, eli mitä ovat TV-sovellukset ja minkälaista TV-sovellusohjelmointi on. Luvussa 3 määritellään yleisesti HbbTV-ohjelmointiin liittyvät standardit ja niiden taustalla olevat organisaatiot, sekä tarkemmin ne asiat jotka liittyvät tilausvideopalvelun toteutukseen, tiedonsiirtoon, sekä videontoistoon. Luvussa 4 määritellään tilausvideopalvelu ja sen vaatimukset. Luvussa 5 kuvataan projektit, joihin toteutus liittyy, sekä HbbTV-toteutuksen ratkaisut. Luvussa 6 arvioidaan HbbTV-toteutusta sekä ratkaisuja ja jatkokehitystä. Luvussa 7 on yhteenveto koko työstä.

2 TV-SOVELLUKSET JA TEKNIIKAT

TV-sovellus on televisiolle toteutettu television käyttäjää palveleva ohjelmisto, joka suoritetaan television käyttöjärjestelmässä. Sovellus voi olla natiivi käyttöjärjestelmän osana tuleva sovellus, kuten ohjelmaopas; erikseen sovelluskaupasta asennettava natiivisovellus; tai TV-lähetyksen yhteydessä jaeltava ohjelmistoon liittyvä sovellus. [1, s. 7]

Seuraavissa kohdissa esitellään TV-sovellusten ohjelmointiin liittyvät asiat. Kohdassa 2.1 esitellään yleiskuvaus TV-sovelluksista. Kohdassa 2.2 määritellään yleisesti television sovellusympäristö. Kohdassa 2.3 esitellään tapa tuottaa ohjelmoituja sisältöjä televisiokuvaan, jotka eivät ole televisiosovelluksia, koska niitä ei suoriteta television käyttöjärjestelmässä. Kohdassa 2.4 esitellään television natiivi- ja hybridisovellusympäristö. Kohdassa 2.5 esitellään IPTV-jakeluympäristö. Kohdassa 2.6 esitellään HbbTV-ympäristö. Kohdassa 2.7 esitellään muita televisioympäristöjä.

2.1 Yleiskuvaus

TV-sovellukset voidaan jaotella erilaisiin kategorioihin tai sovellustyyppeihin. Tähän vaikuttavat sovelluksien tekniset ominaisuudet ja se, miten TV-sovellus määritellään. Sovellus voi olla mikä tahansa ohjelmallisesti toteutettu sisältö TV:ssä. Sovellustyyppjä voidaan jaotella karkeasti staattisiin sovelluksiin sekä interaktiivisiin sovelluksiin. Interaktiivinen sovellus voi olla joko yksi- tai kaksisuuntainen.

Staattisella sovelluksella tarkoitetaan esimerkiksi jotain TV:ssä näkyvää ohjelmoitua näkymää, kuten uudelle kanavalle siirryttäessä näkyvää ilmoitusta siitä, mikä on kanavan nimi ja numero ja kuinka paljon kello on. Staattiseksi sovellukseksi voidaan myös ajatella TV-kuvaan poltettu ohjelmallisesti tuotettu näkymä, joka tulee lähetyksessä kuvan mukana, ja jota ohjataan lähettäjän päässä, mutta käyttäjä ei voi ohjata sitä. Esimerkiksi uutisohjelmista tutut näkymät, joissa voi pyöriä uutisotsikoita, pörssikurseja, säätietoja, jne.

Yksisuuntainen tarkoittaa, että sovellus saa dataa TV-lähetyksen signaalista ja käyttäjä pystyy ohjaamaan sovellusta laitteen kaukosäätimellä, mutta ei pysty hakemaan muuta dataa kuin sen mitä lähetyksen mukana tulee. Tällainen sovellus on esimerkiksi teksti-TV, joka saa kaiken tekstisisällön lähetyksen mukana ja käyttäjä pystyy vain valitsemaan sivun, jonka sisältö tulee näkyville sitten, kun se lähetetään. Sisältöä ei siis pystytä hakemaan, mutta voidaan päättää, mitä halutaan vastaanotetusta signaalista näyttää.

Kaksisuuntaisia interaktiivisia sovelluksia voivat olla vain sellaiset sovellukset, jotka pystyvät hyödyntämään Internetiä ja näin ollen sekä vastaanottamaan, että hakemaan uutta tietoa. Internetiin kytkettäviä televisioita (Connected TV) kutsutaan älytelevisioiksi (Smart TV). Älytelevisio voidaan määritellä siten, että se on Internetiin

kytkettävissä oleva televisio tai digiboksi, johon voidaan asentaa erilaisia sovelluksia esimerkiksi laitevalmistajan sovelluskaupasta. Älytelevision ja television ero on verrattavissa tavallisen matkapuhelimen ja älypuhelimen eroihin.

2.2 TV sovellusympäristönä

Vaikka TV-sovellukset voivat näyttää melkein samanlaisilta kuin monet PC-sovellukset, asettavat TV:n erityisominaisuudet sekä rajoitteita että mahdollisuuksia. TV-sovelluksilla, PC-sovelluksilla ja myös esimerkiksi mobiilisovelluksilla on kaikilla paljon eroja, vaikka monesti ohjelmointitekniikat voivat olla täysin samoja kussakin tapauksessa.

TV:n erityispiirteitä ovat suuri näyttö, mutta toisaalta sen katseluetäisyys on kauempana kuin PC:llä. TV:n erottelukyky on myös useimmiten pienempi. Tästä syystä TV-ruudulle ei ole mahdollista saada yhtä paljon informaatiota kerralla kuin PC-ruudulle. TV-sovellusten ulkonäön tulisi olla hyvin huoliteltu ja graafinen, koska TV:stä katsellaan kuvia ja videoita, ne ovat useimmiten viihdekäytössä eikä työikäytössä kuten PC:t, näyttäisi dokumentinomaiset näkymät ilman kuvamaista tunnetta huonoilta. Siksi grafiikka ja värisuunnittelu ovat isossa osassa sovellusten kannalta.

TV-sovelluksien suunnittelussa kannattaakin lähteä usein samoilla periaatteilla kuin mobiilisovellusten kautta. Ruudulla on kerrallaan yksi näkymä, ja näkymästä toiseen navigointi tapahtuu näkymä kerrallaan yhdellä komennolla. TV:ssä esimerkiksi nuoli oikealle vaihtaisi seuraavan näkymän samoin kuin mobiilisovellusta kosketusnäytöllä ohjattaessa pyyhkäisy oikealta vasemmalle vaihtaisi seuraavan näkymän. PC:llä useampi asia saattaa sopia yhteen näkymään kerralla, eikä näkymien vaihtoa tarvita niin paljon. Silloin kun PC-sovelluksessa jaetaan informaatiota tai sovelluksen osia toisiin näkymiin, on usein totuttu toteuttamaan niin sanottu välilehtirakenne, jossa hiirellä haluttua tabia painaen voidaan vaihtaa haluttu näkymä. Välilehtiä voidaan tietenkin käyttää kaikissa sovelluksissa, mutta erona PC:hen edellä mainituissa esimerkeissä on se, ettei PC:llä yleensä toteuteta sovelluksiin näkymien välille vain seuraavan ja edellisen näkymän vaihtamiseen mahdollistavaa navigointia, vaan hiirikäyttöiselle ohjaukselle totuttua ohjaamista, joka mahdollistaa suoran siirtymän muihin näkymiin, eikä näkymiä tarvitse selata. [1, s. 10-18]

Toisin kuin PC:llä TV-sovelluksissa ei yleensä ole käytössä kohdistinta, pois lukien sellaiset TV:t, jotka tukevat osoittimenä toimivaa kaukosäädintä. Kuitenkin suurinta osaa televisioista ohjataan yleensä perinteisellä kaukosäätimellä. Tällöin kohdistin on suunniteltava sovellukseen siten, että käyttäjä näkee, mikä elementti, valikko, painike tai muu valittava asia ruudulla on milloinkin valittuna. Toisin sanoen kohdistimen fokus on ilmaistava selkeästi ruudulla, mutta kohdistinta ei yleensä piirretä PC:stä tutun hiiri-osoittimen näköiseksi, vaan se toteutetaan esimerkiksi ympäröimällä valittuna oleva käyttöliittymäkomponentti jollain värillä, joka ilmaisee sen olevan aktiivinen kyseisellä

hetkellä. Kohdistimen tai aktiivisen elementin asema kulloinkin vaikuttaa siis sovelluksen tilaan, toisin kuin PC:llä hiiren sijainti ruudulla, joka ei yleensä vaikuta mitenkään sovelluksen tilaan.

Kohdistinta suunniteltaessa tulee miettiä sen havaittavuutta, jotta käyttäjä tietää sen olevan liikuteltavissa nuolinäppäimillä. Tätä havainnollistamaan voidaan joissain tilanteissa kohdistimeen tai sen vierelle asettaa nuoli-ikoneita. Esimerkkinä tällaisesta voisi olla alas aukeavassa valikossa oleva kohde. Valitun kohteen kohdalla voidaan liikkua ylös, alas, sekä mahdollisesti oikealle, josta aukeaa alivalikko. Käyttäjä ei voi tietää oikealle aukeavasta valikosta tai siitä, että sinne voidaan navigoida, muuta kuin kokeilemalla, ellei se ilmene selkeästi kohdistetusta elementistä tai muutoin käyttöliittymästä. Kohdistimen tulee myös erottua selkeästi taustastaan. Mikäli valintoja olisi ruudulla vain kaksi, ei käyttäjä voi tietää kumpi valinnoista on kohdistettu, jos kohdistuksena käytetään eri väristä taustaa.

Yksi tärkeä osa TV-sovellusohjelmoinnissa on käytettävyys, koska TV-sovellukset ovat yleensä viihdekäyttöön [1, s.10]. Tämä on tärkeää siksi, jotta käyttäjät voivat saada hyvän käyttäjäkokemuksen sovelluksen käytöstä ja olla sovelluksen toimintaan, helppokäyttöisyyteen ja suorituskykyyn tyytyväisiä. Mikäli viihde-elektroniikkalaitteissa ja niiden sovelluksissa on valittamista, voi se vaikuttaa negatiivisesti tuotteen markkinaa ja brändiin.

Sovelluksia toimittavan tahon kannalta käyttäjien tyytyväisyys on hyvin tärkeää, koska eri toimittajat kilpailevat asiakkaista ja yhteistyökumppaneista keskenään. Asiakkaiden on helppo valita toimittaja sovelluksilleen sen perusteella, kenen sovellukset ovat menestyneet ja käyttäjien on helpompi vaihtaa huonompi sovellus parempaan, kuin koko laite, mikäli sovellus ei ole laiteriippuvainen.

2.3 Videokuvaan sisällytetty näkymä

Ensimmäinen tapa sisällyttää TV-lähetykseen muuta informaatiota kuin kameran läpi kuvattu videokuva, joka edelleen on paras omalta osaltaan monissa tilanteissa, on liittää videokuvaan ohjelmallisesti näkymiä. Tällä tekniikalla voidaan lisätä muun muassa tekstiraitoja TV-ohjelmiin, tekstejä ja kuvia uutislähetyksiin, inforuutuja sekä muuta sisältöä. Eräs täysin tällä tekniikalla toteutettu palvelu voisi olla jokin inforuutu-sovellus. Tällaisella sovelluksella on paljon hyödyllisiä käyttökohteita ja se soveltuu hyvin nimenomaan tiedon jakeluun.

Huonona puolena on se, ettei tällaisella sovelluksella voi olla minkäänlaista suoraa käyttäjäinteraktiota, koska sovellus on päätelaitteella vain videokuvana näkyvä tavallinen TV-lähetys. Toisaalta sitä voidaan pitää myös tämän tekniikan vahvuutena, jos halutaan, että kaikille katsojille näkyy sama näkymä ja he eivät voi, eikä heidän tarvitse voida vaikuttaa siihen mitenkään.

Epäsuora interaktio tässäkin sovellustyypissä on mahdollista. Sitä on käytetty laajalti erilaisissa chat- ja peliohjelmissa, joissa käyttäjät voivat osallistua keskusteluun tai peliin kännykällä tekstiviestein tai soittamalla annettuun numeroon. Tällaisia peliohjelmiä, joissa soitetaan lähetykseen, ovat yleensä tietokilpailutyypiset televisio-ohjelmat, joissa televisiossa esitettäviin arvoituksiin tai tehtäviin voi vastata soittamalla annettuun numeroon. Tämä on hyvin kaupallinen tapa hyödyntää televisiota sellaisina aikoina kun katsojia on vähän eikä katseluohjelmia kannata lähettää. Ne katsojat, jotka hiljaisena aikana katsovat peliohjelmaa, kiinnostuvat tarjolla olevasta palkintorahasta, ja yleensä verrattain helppojen tehtävien ratkaisun löytyminen kannustaa soittamaan lähetykseen.

2.4 Natiivi ja hybridi TV-ohjelmisto

Natiiveja TV-sovelluksia voivat olla laitevalmistajien itse laitteeseen integroimat sovellukset tai siihen ladattavat kolmannen osapuolen TV-sovellukset. Eri valmistajilla voi olla omia sovelluskauppoja, joita voi käyttää internetiin kytketyllä Smart TV:llä. Sovelluskauppa toimii kuten nykyiset mobiilisovelluskaupat. Sovelluskehittäjät voivat ladata kauppaan omia sovelluksiaan kuluttajien saataville.

Laitevalmistajien sovelluskaupat voivat tarjota kehittäjille erilaisia mahdollisuuksia ohjelmoida TV-sovelluksia. TV-sovellusalustoja on kuitenkin lukuisia ja joidenkin TV-valmistajien suljettu kehitysympäristö vaikeuttaa sovelluskehittäjien mahdollisuuksia tuottaa sovelluksia ja palveluita eri alustoille [2]. Ratkaisuksi sovellusalustojen pirstaloitumiseen on kehitetty hybridisovelluksia, joissa sovelluksen suurin osa on toteutettu kaikissa laitteissa toimivalla HTML5-kielellä [3]. Muun muassa Operan sovelluskauppaan voi julkaista HTML5-hybridisovelluksia, jotka ovat verrattavissa mobiililaitteiden hybridisovelluksiin. HTML5-hybridisovellukset eroavat HbbTV-sovelluksista pääasiallisesti siinä, että ne ladataan sovelluskaupasta, eivätkä ne tule lähetyksen mukana saumattomasti. Lisäksi ne ovat laitekohtaisia ja tarvitsevat laitekohtaisen rajapinnan päästäkseen käsittelemään esimerkiksi DVB-lähetykseen liittyviä tietoja, kun taas HbbTV-sovellus käyttää vain OIPF -rajapintoja ja sama sovellus toimii kaikissa HbbTV:ta tukevissa laitteissa. [4]

Tilausvideopalvelu voi hyvin olla natiivisovellus. Tunnetuin sellainen lienee Netflix, joka on saatavilla useisiin eri TV-valmistajien uusiin malleihin. Myös televisiokanavien omia sovelluksia on ladattavissa sovelluskaupoista tai löytyy esiasennettuna, kuten Yle Areena, MTV3 Katsomo ja Ruutu. Samsungin älytelevisioissa on vuoden 2012 malleista lähtien ollut esiasennettuna muun muassa Yle Areena. Yle Areena -sovellus on myös kehitteillä lisäksi LG:n ja Sonyn televisioille. Sovellus on tarkoitus julkaista 2015. [5]

2.5 IPTV

IPTV on kokonaan IP-verkossa toimiva kanava tai videopalvelu, jossa näkyy reaaliaikaisesti samat kanavat kuin DVB -lähetyksissä tai valikoidut ja lisätyt kanavat. Palvelu voi maksaa erikseen ja sen hintaan voi vaikuttaa se, miten laaja kanavavalikoima siihen kuuluu. Tunnettuja palveluntarjoajia eli IPTV-operaattoreita Suomessa ovat muun muassa Elisa, Sonera, Anvia ja Maxivision [6].

IPTV:n katseluun tarvitaan Internetyhteyden lisäksi palvelusopimus joltakin palveluntarjoajalta sekä televisioon kytkettävä laite, jolla IPTV-palveluita voidaan käyttää, tai tietokone, jolla jotakin IPTV-sovellusta voidaan käyttää. Eri palveluntarjoajat tarjoavat sekä digibokseja, että IPTV-bokseja. Jotkin operaattorit myyvät niin sanottuja viihdebokseja joiden avulla voi katsoa sekä tavallisia antenni- tai kaapeli-TV -lähetyksiä, että IPTV-kanavia.

2.6 HbbTV

HbbTV (Hybrid Broadcast Broadband TV) tarkoittaa standardia ja tekniikkaa, joka mahdollistaa **DVB**-lähetyksen ja laajakaistayhteyden saumattoman integraation. HbbTV-standardi on ensimmäisenä otettu käyttöön Saksassa, Ranskassa, sekä Englannissa, minkä jälkeen HbbTV-palvelut ovat lisääntyneet vähitellen muissa maissa [7]. Maailmanlaajuisesti HbbTV-sovelluksia on nykyään tai viimeistään ensi vuonna tarjolla suurimmassa osassa Euroopan maita, Turkissa, Lähi-Idässä, Malesiassa, Vietnamissa, Singaporessa, Indonesiassa, sekä Australiassa. [8]. Näistä uusista maista Sofia Digital on ensimmäisenä tuomassa HbbTV-sovelluksia Suomeen, Viroon, Turkkiin, sekä Malesiaan.

HbbTV -sovellukset ovat vielä harvinaisia ja monelle tuntemattomia Suomessa. HbbTV-lähetykset alkoivat Suomessa Digitan antenniverkossa maaliskuussa 2013 [9]. HbbTV-sovelluksia voidaan käynnistää älytelevisioissa (**Smart TV**), mikäli laite tukee HbbTV-standardia, on kytketty Internetiin ja lähetyksessä tulee tarvittava signaali, joka käynnistää itse sovelluksen. Yleisimpiä palveluita ovat ohjelmaopas, uutiset, säätiedot, tilausvideot. Muita palveluita voisi olla tiettyyn ohjelmaan liittyvät sovellukset, kuten äänestys viihdeohjelmassa, tai twitter-feed keskusteluohjelmassa.

2.6.1 Saatavuus

HbbTV on saatavilla noin 95:ssä prosentissa vuoden 2012 jälkeen myydyissä Smart TV -laitteissa joko suoraan tai ohjelmistopäivityksen avulla. Listaa HbbTV-laitteista pitää Digita OY. Testattujen HbbTV-laitteiden listalla on AntennaReady HD -merkinnän saaneet laitteet. Tämä tarkoittaa käytännössä myös sitä että HbbTV toimii. [10]

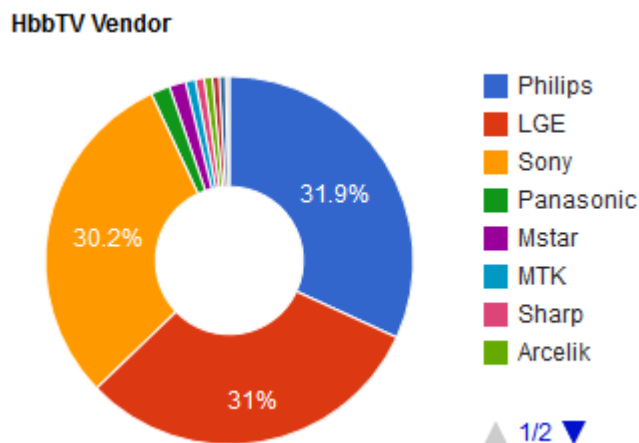
HbbTV on näkynyt Suomessa maaliskuusta 2013 lähtien kaikilla pääkanavilla TV1, TV2, MTV3, Nelonen, Liv ja Jim maanpäällisissä Digitan TV-lähetyksissä [9], sekä joidenkin kaapeli-TV-operaattorien kaapeliverkossa. Tulevaisuudessa HbbTV tulee

todennäköisesti yleistymään ja tulee saataville myös kaapeli- ja satelliitti-TV-lähetysiin. Kaapelioperaattoreitakin, kuten muita teleoperaattoreita, IPTV-palveluntarjoajia tai muita yhteisantennijärjestelmien ylläpitäjiä sitoo siirtovelvoite (must carry), jonka mukaan niiden on välitettävä tietyt valtakunnalliset yleiskanavat verkossaan, niiden televisio- ja radio-ohjelmat, sekä niihin liittyvät oheis- ja lisäpalvelut [11]. Tästä huolimatta HbbTV ei vielä näy monenkaan alan toimijan verkossa, koska AIT-taulua ei välitetä vastaanottimille. Tämä ei vaadi kaapelioperaattoreilta suuria investointeja, koska jos alkuperäinen signaali toimitetaan muuttumattomana, HbbTV näkyy päätelaitteissa, mutta monet kaapelitelevisio-operaattorit ovat erikseen poistaneet sen. [11] [12]

Suomi on HbbTV:n omaksumisessa jäljessä useihin muihin Euroopan maihin verrattuna. Keväällä 2015 suomen kanavilla näkyy antennilähetyksissä vain ohjelmaopas kaikilla peruskanavilla, sekä MTV3:lla Juurinyt -uutissovellus. Sofia Digitalilla on kuitenkin kokemusta erilaisten sovellusten kehityksestä asiakasprojekteissa ulkomaisten televisio- ja radioyhtiöiden kanssa, sekä monia erilaisia sovelluksia lähes valmiina siirrettäväksi tuotantoon myös Suomessa.

2.6.2 Päätelaitteet

HbbTV-laitteita on ollut markkinoilla vuodesta 2011 lähtien. Kuvassa 1 näkyy otos Suomessa olevista internetiin kytketyistä HbbTV-laitteista 4-8.4.2015 HbbTV-kanavilta kerättyjen tietojen perusteella. Tiedot on kerätty Sofia Digitalin tietokannasta vastaanottimien User Agent -merkkijonon perusteella yksilöityjen IP-osoitteiden ja niiden mukaan yksilöityjen User Agent -merkkijonojen mukaan. Laittevalmistajien jakauman mukaan voidaan päätellä prioriteetit sille, mitkä ovat tärkeimpiä ja mitkä marginaalisia merkkejä kun testataan yhteensopivuutta sovelluskehityksessä. Määrällisesti tutkitun ajan sisällä vastaanottimia vieraili kanavilla noin 47000 kappaletta. Luvut vastaavat siis todellisia käyttäjiä jotka saavuttavat tällä hetkellä HbbTV-palvelut suomessa, vaikka HbbTV-televisioita on huomattavasti enemmän.



Kuva 1: HbbTV Laitteet - Valmistajat

Laitteilla on maakohtaisia käyttöprofileita, mikä tarkoittaa sitä, että niissä voi olla erilaisia asetuksia sen mukaan missä maanosassa ne ovat markkinoilla, ja sen mukaan mille profiilille se asetetaan käyttäjän toimesta. Jotkin televisioidot vaativat käyttäjän toimenpiteitä HbbTV:n käyttöön ottoon. Eroina profileissa voi joidenkin valmistajien televisioissa esimerkiksi olla Saksan profiilissa HbbTV aktiivisena, mutta Suomen profiilissa ei. Usein kyse on vain asetusvalinnasta, jossa on kaksi vaihtoehtoa: Päälle ja pois. Jotkin televisioidot vaativat HbbTV:tä varten erillisen sovelluksen, jonka voi ladata sovelluskaupasta. [13]

2.6.3 Ohjelmointitekniikat

HbbTV-sovelluksia voidaan ohjelmoida CE-HTML -merkkaukielellä, joka on suunniteltu kulutustuotteille ja viihdelaitteille, kuten älytelevisioille. CE-HTML sisältää suuren osan HTML5:n sisällöstä, mutta ei kaikkea. Sen sijaan se sisältää OIPF:n määrittelemiä objekteja rajapintoihin, joiden avulla pystytään kommunikoimaan laitteen kanssa ja päästään käsiksi DVB-lähetukseen ja kaikkeen sen sisältämään metadataan. Natiiveissa sovelluksissa voi olla eri tapoja ohjelmoida riippuen laitevalmistajasta ja laitteen käyttäjärjestelmästä, sekä kehitysympäristöstä. [14]

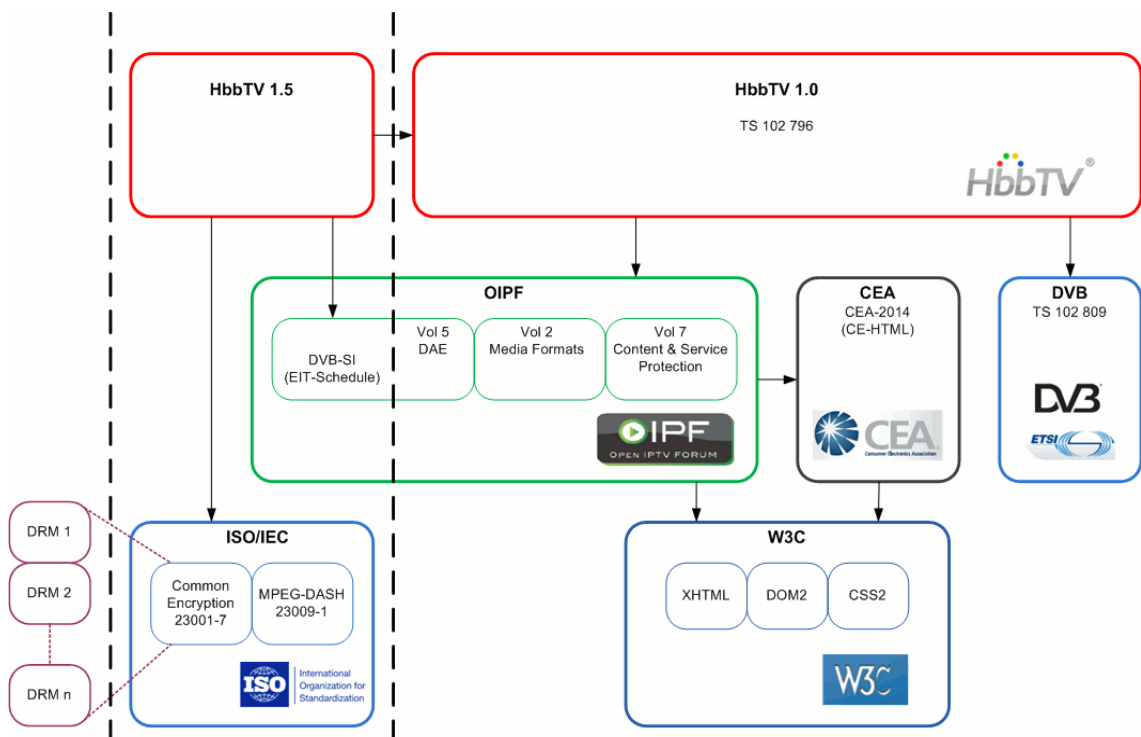
2.7 Muiden laitteiden TV/Video -palvelut

Vastaavia videopalveluita tai sovelluksia kuin televisiolla ja PC:llä on saatavilla myös muille laitteille, kuten älypuhelimille ja tablet-tietokoneilla. Samoista tilausvideosovelluksista voidaan tehdä eri päätelaitteelle sovelluksia siten, että ne käyttävät kuitenkin samaa palvelinrajapintaa. Yhden rajapinnan kautta voidaan jakaa sisältöä etenkin tilausvideopalveluissa. Etuna tästä on se, että kun halutaan uusi sovellus uudelle alustalle, riittää tehdä jo olemassa olevaa rajapintaa tukeva asiakassovellus. HbbTV-tilausvideopalveluita on siksi verrattain helppo tehdä, koska TV-operaattoreilla on jo valmiit palvelinrajapinnat omiin Internet-pohjaisiin ratkaisuihinsa.

3 STANDARDIT JA ORGANISAATIOT

HbbTV ja siihen liittyvät standardit kuuluvat kaikki ETSI TS 102 796 v1.2.1 (European Telecommunication Standardization Institute Technical Specification 102 796 v1.2.1) -määrittelyn piiriin. Kuvassa 2 näkyvät kaikki tässä luvussa käsitellyt standardit, joista käsitellään tärkeimmät HbbTV-ohjelmointiin liittyvät asiat seuraavissa kohdissa [7].

Kohdassa 3.1 käsitellään DVB-standardi, joka on digitaalisen television jakelun standardi. Kohdassa 3.2 käsitellään HbbTV-standardin tärkeimmät HbbTV-palveluiden toteuttamiseen liittyvät asiat yleisellä tasolla. Kohdassa 3.3 käsitellään MPEG DASH, joka on tilausvideopalveluiden videontoiston kannalta tärkeä standardi. Kohdassa 3.4 käsitellään DRM-tekniikat, eli tilausvideopalvelun digitaalisen videosisällön oikeuksien hallintamekanismit. Kohdassa 3.5 käsitellään käyttäjän tunnistamisen mahdollistava CI-standardi, jonka avulla DVB-siirtovirrasta, sekä HbbTV-palveluista voidaan tunnistuksesta riippuen avata salattuja palveluita.



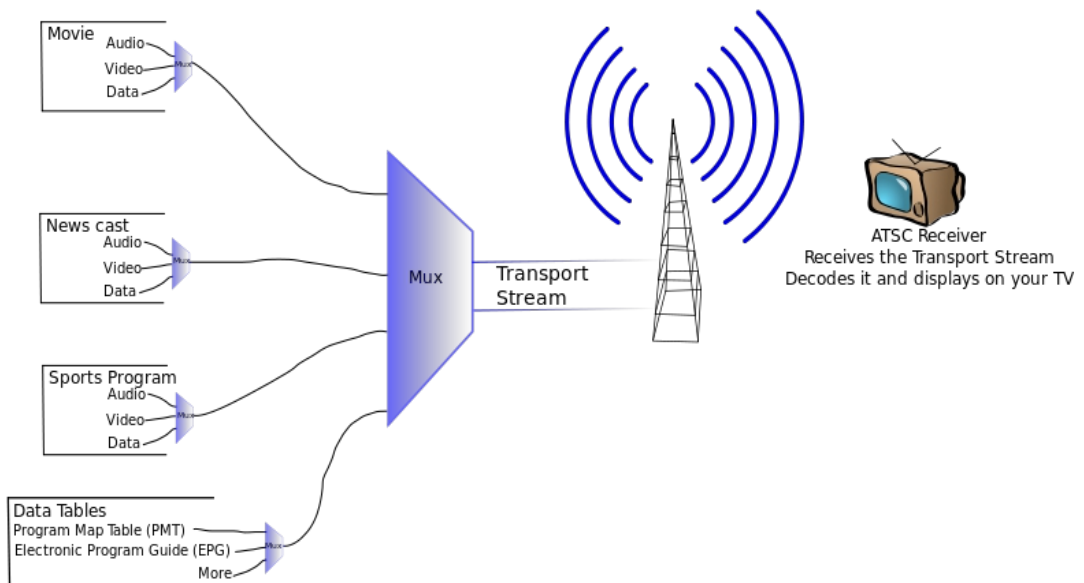
Kuva 2: ETSI TS 102 796 v1.2.1 - Standardit ja niiden riippuvuudet [15, s. 17].

3.1 DVB

DVB (Digital Video Broadcasting) on digi-TV -standardi, jota noudatetaan kaikissa suomen TV -lähetyksissä, sekä maailmanlaajuisesti. DVB -tyyppjä on useita erilaisia, joista tärkeimmät ovat maanpäällinen DVB-T (Terrestrial), DVB-C (Cable), sekä DVB-S

(Satellite). Verkko-operaattorit jakavat ohjelmatoimiluvanhaltijoiden jakamaa lähetystä näiden kanavien kautta. Suomessa noin puolet talouksista vastaanottavat DVB-signaalinsa antenniverkosta ja puolet kaapeliverkosta. Noin 5% talouksista vastaanottaa myös satelliittilähetyksiä, mutta se on yleensä rinnakkainen vaihtoehto kaapelin ja antennin kanssa [1, s.2].

On olemassa myös muita DVB-standardeja. DVB-standardien variaatioissa suurin ero on jakelutien erojen vaatimat muutokset, sekä vastaanottavan laitteen mukaisesti tehdyt optimoinnit. Näitä voivat olla eri tarkoituksiin erilaisten taajuuksien käyttäminen, erilainen kanavanippujen signalointi, sekä erilaiset videokuvan resoluutiot. Esimerkiksi DVB-H (Handhelds), joka on matkapuhelimille optimoitu DVB-standardi. Sen lähetyksissä on pienempi kuvantarkkuus ja sen tulisi toimia 300km/h liikkuvassa junassa ja pienemmällä sähkönkulutuksella. Tälle kilpailevia standardeja ovat MediaFLO, DMB ja MBMS. Standardi hyväksyttiin 2004 Euroopan mobiili-TV-standardiksi ja otettiin suomessa kaupalliseen käyttöön joulukuussa 2006, mutta se ei levinnyt riittävän laajalle, vaikka Nokia tuki sitä vahvasti. DVB-H näkyi viimeisenä suomessa ja Digita sulki DVB-H -verkon kokonaan maaliskuussa 2012. [17]



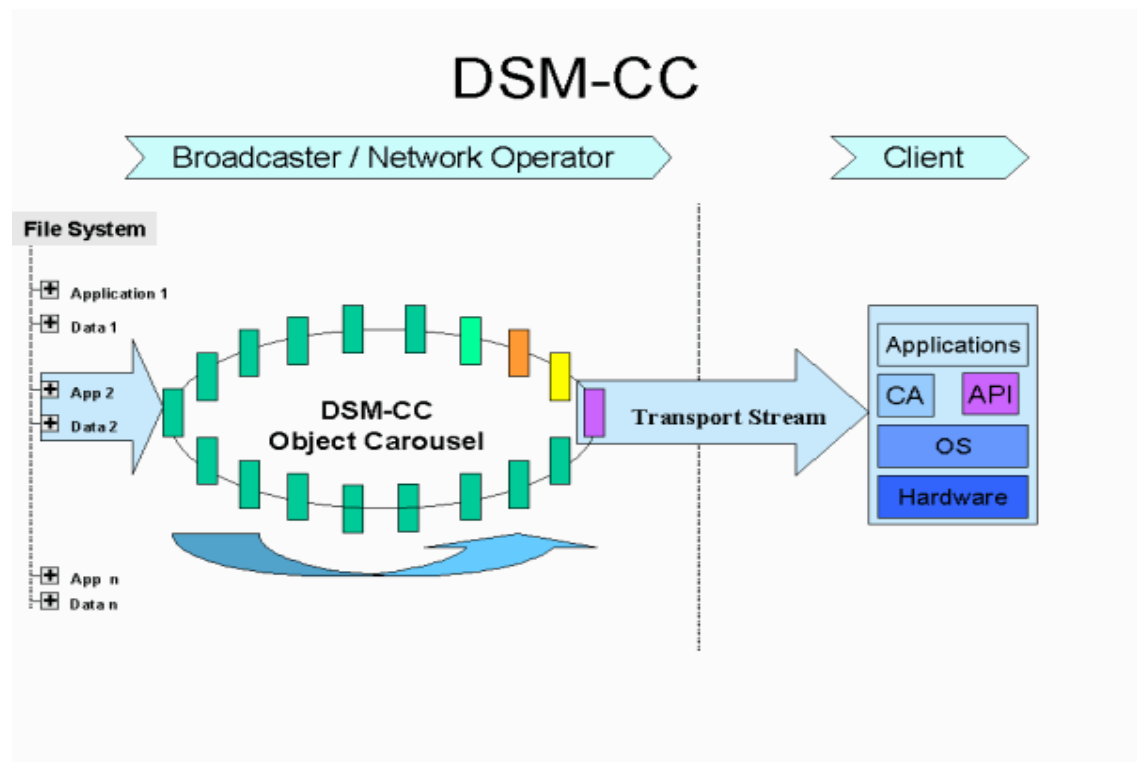
Kuva 3: DVB-siirtovirta (MPEG Transport Stream)

(http://en.wikipedia.org/wiki/File:MPEG_Transport_Stream_HL.svg)

MPEG Transport Stream (TS) on siirtovirta, jonka avulla DVB-lähetykset muodostetaan (Kuva 3). Muodostaminen tapahtuu monivalitsimien avulla eri signaaleista (Multiplexing). Tässä ensimmäisessä vaiheessa video, ääni, sekä metadata, kuten tekstitykset segmentoidaan peräkkäisiksi tietolohkoiksi yhdeksi signaaliksi. Sen jälkeen eri ohjelmat niin ikään segmentoidaan eri taajuuksille, joista muodostuu TV-kanavia. TV-vastaanotin saa lähetksen yleensä joko kaapelin (DVB-C), antennin (DVB-T), tai satelliittilautasen (DVB-S) kautta. HbbTV-terminaali lukee siirtovirrasta

AIT:n sekä muut komponentit (demultiplexing). Näin video, ääni, sekä metatiedot saadaan eriteltyä kanavakohtaisesti ja jatkokäsiteltyä halutulla tavalla. Mikäli AIT:ssa on *autostart*-parametrilla asetettu HbbTV-sovelluksen URL, sovellus ladataan heti internetin yli ja käynnistetään.

DSM-CC Object Carousel on järjestelmä, jonka avulla voidaan lähettää tiedostoja synkronisesti yksisuuntaisessa tietovirrassa (Kuva 4). Koska päätelaite ei voi valita lähetyksestä mitä tietoja se kulloinkin tarvitsee, tiedot lähetetään jatkuvasti DSM-CC:n avulla yksi kerrallaan. (Filecast). Tämäkin tieto niin ikään tulee samassa siirtovirrassa kuin kaikki muukin TV-lähetyskohtainen tieto. Hyvänä esimerkkinä karusellin toiminnasta on teksti-TV, jossa jokainen sivu tulee vuorollaan. Halutun sivun saa näkyville vasta sitten kun se on tullut lähetyksen mukana, eli toisin sanoen ”karuselli pyörähtää ympäri”, kun teksti-TV vastaanottaa saman sivun jälleen. Silloin se tosin voi olla päivittynyt ja sisältää eri tietoa. [18]



Kuva 4: DSM-CC Object Carousel

DSM-CC-objektikarusellin tiedostoihin päästään käsiksi päätelaitteessa DVB-URL:n avulla. DVB-URL alkaa aina merkeillä ”*dvb://*”, jota seuraa hexadesimaaliarvoisista luvuista koostuva DVB-tripletti. DVB-URL voisi olla esimerkiksi ”*dvb://8282.80.1*”. DVB-tripletti koostuu kolmesta komponentista, joiden avulla voidaan tunnistaa siirtovirtaan sisällytetyt komponentit. Esimerkiksi ne ovat pisteellä eroteltuna, järjestyksessään [19]:

1. Original Network Id = 0x8282. Määrittää yksilöllisesti verkon tunnisteen.
2. Transport Stream Id = 0x80. Määrittää yksilöllisesti verkon sisäisen siirtovirran.

3. Service Id = 0x1. Määrittää yksilöllisesti yksittäisen palvelun (Kanava) siirtovirrassa.

Nämä kolme komponenttia yhdessä yksilöivät yhden palvelun siirtovirrassa, joka vastaa televisiokanavan sisältöä. Sisältöön kuuluu palvelukohtaiset komponentit siirtovirrassa, kuten video, ääni, teksti ja muut palvelut, joita voi olla erilaiset taulut, kuten AIT, EIT, jne. [20]

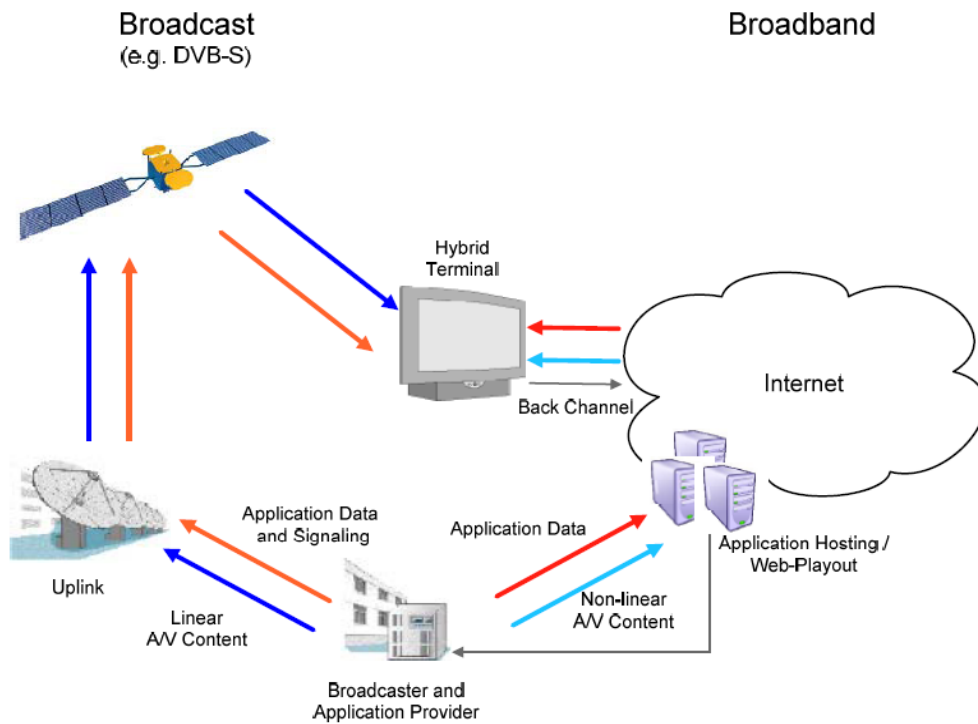
3.2 HbbTV

HbbTV on maailmanlaajuinen avoin standardi, jota on vuodesta 2008 lähtien kehitetty yhdistämään TV-kanavien jakeluun vuorovaikutteisia internetpohjaisia sovelluksia. Standardin tekninen määrittely ja uusin versio on ETSI TS 102 796 v1.2.1 (2012-11), joka on päivitetty syyskuussa 2012. Tämä sisältää 3 dokumenttia, alkuperäisen standardin TS 102 796 v1.1.1, korjauksen 2:n TS 102 796 v1.1.1:n HbbTV:ltä, sekä HbbTV spesifikaation 1.5. ETSI (European Telecommunications Standards Institute) hyväksyi standardin version 1.1.1 heinäkuussa 2010. [21]

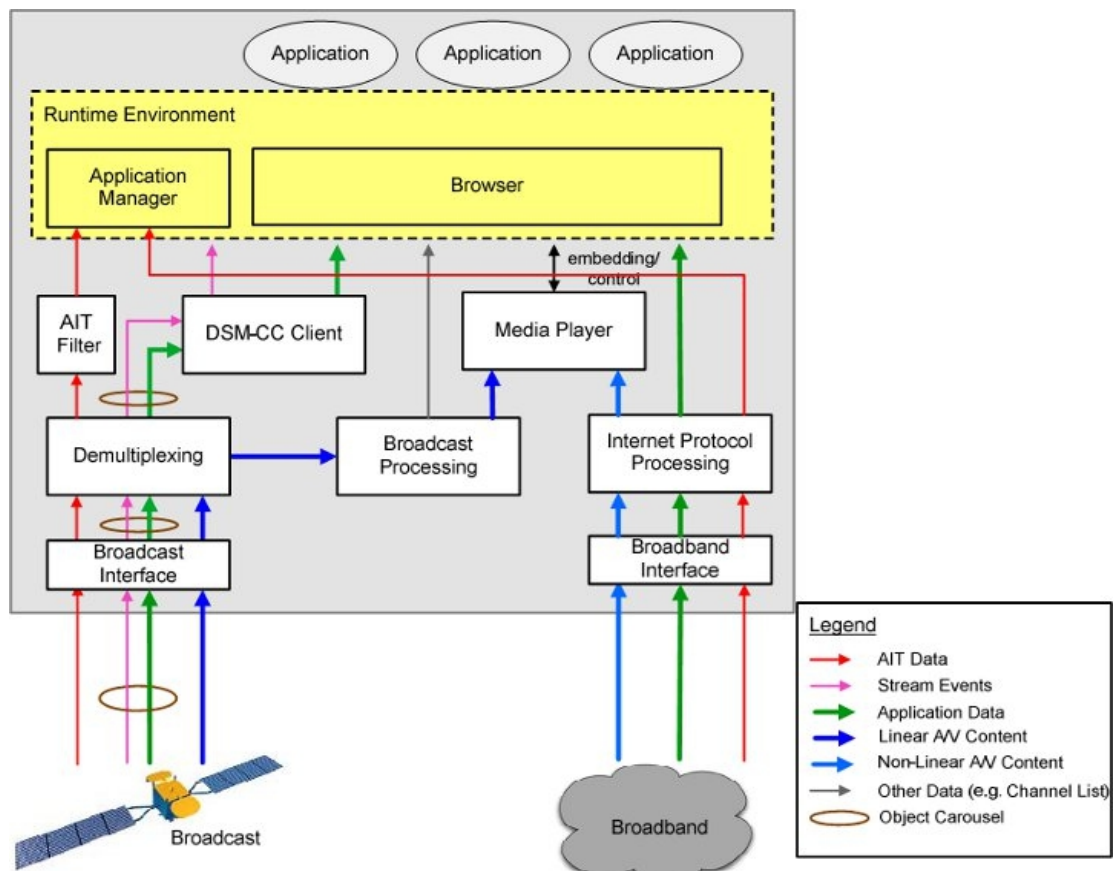
Edellä mainitun teknisen määrittelyn on tuottanut Joint Technical Committee (JTC), Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECTrotechnique (CENELEC) ja European Telecommunications Standards Institute (ETSI) yhdessä [15]

HbbTV:n ja siihen liittyvien standardien ja määritysten kehittäminen jatkuu tulevaisuudessa. TS 102 796 v2.0 on jo työn alla ja HbbTV konsortiolla on yli 60 jäsentä [7]. HbbTV 2.0 -standardin esitys on hyväksytty tammikuussa 2015, ja sen mukaan uusimmat uudistukset ovat täydellinen HTML5-tuki, HEVC -High Efficiency Video Coding (h.265), joka mahdollistaa kuvan koosta ja videon päivitystaajuudesta riippuen keskimääräisesti jopa noin 40% paremman kompression, kuin h.264 AVC. Lisäksi CS (Companion Screen) on laitteiden puolesta tuettu omalla CS-komponentillaan. [22]

Kuvassa 5 on esitetty HbbTV-järjestelmän arkkitehtuuri yleisellä tasolla. HbbTV terminaali, eli HbbTV-yhteensopiva televisio vastaanottaa TV-operaattorin (Broadcaster) DVB-signaalin jotakin siirtotietä käyttämällä, tässä tapauksessa satelliittilähetyksen kautta. Lähetyksen siirtovirtaan signaloidaan TV-operaattorin järjestelmässä lineaarisen TV-lähetyksen (video, ääni) lisäksi sovellusdataa, jonka avulla HbbTV-terminaali voi käynnistää ohjelmistoon liittyviä HbbTV-sovelluksia. Paluukanavana HbbTV-terminaali käyttää Internetiä, jonka kautta sovelluksia voidaan käyttää sovelluspalvelimilta, joissa on TV-operaattorin sovelluksia. Internet-yhteys mahdollistaa ei-lineaarisen TV-lähetyksen katsomisen, eli tilausvideopalvelun toteuttamisen. [23]



Kuva 5: HbbTV: Järjestelmäarkkitehtuuri



Kuva 6: HbbTV-terminaalin komponentit [15, s.14]

Kuvassa 6 on kuvattu HbbTV-terminaalin, eli HbbTV-standardia tukevan vastaanottimen komponentit. Broadcast, eli TV-operaattorin lähettämä signaali vastaanotetaan DVB-T/C/S -standardin mukaisesti pakattuna signaalina ja avataan terminaalin sen hetkellä viritetyn kanavan mukaan jokin palvelu siirtovirrasta (demultiplexing). AIT -signaali kertoo mitä HbbTV-palveluita kanavalla on saatavilla. Application Manager käynnistää *autostart* -parametrilla signaloidun sovelluksen joko URL:n perusteella, tai DSM-CC -karusellista riippuen valitusta siirtotiestä (Application Data). Sovellus suoritetaan HbbTV-selainkomponentissa (Browser). Sovellus voi ottaa kontrollin DVB-videosta ja sen komponenteista videokomponentin (Media Player) avulla, sekä näyttää sen sijaan suoratoistovideota laajakaistayhteyden kautta. [23]

Seuraavissa alakohdissa käsitellään tarkemmin edellä mainitut asiat, sekä HbbTV-standardin noudattamien CE-HTML ja OIPF-standardien tärkeimmät rajapinnat, komponentit, suunnitteluohjenuorat ja yhteensopivuusvaatimukset.

3.2.1 CE-HTML

CE-HTML (CEA-2014) on Consumer Electronics Assosiationin (CEA) määrittelemä kuluttajatuotteiden ohjelmointiin tarkoitettu HTML tekniikoiden nippu. CE-HTML perustuu DOM:iin (Document Object Model), joka on standardoitu malli muodostaa hierarkkisia puurakenteisia dokumentteja. Dokumentissa olevilla elementeillä voi olla omia ominaisuuksia, sekä lapsielementtejä hierarkian alemmalla tasolla, ja sisarelementtejä hierarkian samalla tasolla. [24] CE-HTML:ssä määritellään seuraavat HbbTV:tä koskevat määrittelyt [21][15, s.15]:

- sovellusohjelmointikielet (XHTML, CSS and JavaScript sisältäen AJAX:n)
- miten sisällytetään ei-lineaarista A/V-sisältöä sovellukseen
- DOM-eventtien käsittely, kuten napinpainallustapahtumat (key events).[24]
- kuvaformaattit (still images).

3.2.2 Tiedonsiirto ja AIT

HbbTV mahdollistaa suoran käyttäjäinteraktion TV-lähetyksen yhteydessä. HbbTV -signaali tulee broadcast-lähetyksen siirtovirrassa (MPEG Transport Stream). Se on signaloitu jokaisen kanavanipun jokaiselle kanavalle erikseen AIT-tauluun (Application Information Table). Siirtovirtaa muodostettaessa voidaan määrittää, mikäli halutaan useampia HbbTV-sovelluksia AIT-tauluun ja mikä niistä on *autostart*-sovellus, joka käynnistyy heti kun televisio viritetään kanavalle.

HbbTV-sovellukset voidaan jaella eri siirtoteitse, joko Internetin kautta jota ne käyttävät myös paluukanavana, tai TV-lähetyksessä kokonaan. AIT-taulussa voidaan määrittellä lista sovelluksista, joita on käytettävissä. Jokaiselle sovellukselle määritellään parametreja, kuten sovelluksen automaattinen käynnistys: *autostart/present*, ensisijainen siirtotie: *carousel/web* (Karuselli/Internet).

Koska *autostart*-sovellus käynnistyy aina heti kun se luetaan AIT-aulusta ja ladataan, käyttöliittymää ei yleensä voi avata suoraan TV-lähetysten kuvan päälle, vaan se käynnistyy visuaalisesti näkymättömänä. TV-lähetysten kuvan päälle tulemisen sijaan HbbTV-standardin mukaan hyviin käytäntöihin kuuluu, että sovelluksen olemassaolosta ilmoitetaan pienellä herätteellä kuvan reunassa. Tätä kutsutaan ”Red Button” -herätteeksi. Sovelluksen käyttöliittymä voidaan siis avata painamalla kaukosäätimen punaista nappia, kun ruudulle ilmestyy tämä heräte. Yleensä tämä sovellus on *launcher* -tyyppinen valikkosovellus, jonka tarkoituksena on tarjota lista kaikista kanavan lisäpalveluista, toisin sanoen HbbTV-sovelluksista.

Kaikki ohjelmatiedot, sovellusosoitteet, tekstitykset ja muu metadata voidaan signaloida siirtovirtaan suoraan. Itse HbbTV-sovellukset tarvitsevat internetyhteyden käynnistykseen ja kaksisuuntaisen tiedonsiirron saavuttamiseen.

3.2.3 Video ja Broadcast

HbbTV-sovellus käyttää TV-lähetysten (Broadcast) näyttämiseen HbbTV-standardin mukaista *video/broadcast* -objektia. Objekti määritellään, *<object>*-elementillä *type*-attribuutin avulla: *<object id="video" type="video/broadcast">*, jonka jälkeen sen rajapinta saavutetaan JavaScriptillä valitsemalla objekti DOM:ista, esimerkiksi: *document.getElementById('video').playState*. HbbTV 2.0 standardissa on mahdollista käyttää myös HTML5-spesifistä *<video>*-elementtiä videoiden, sekä TV-lähetysten toistamiseen.

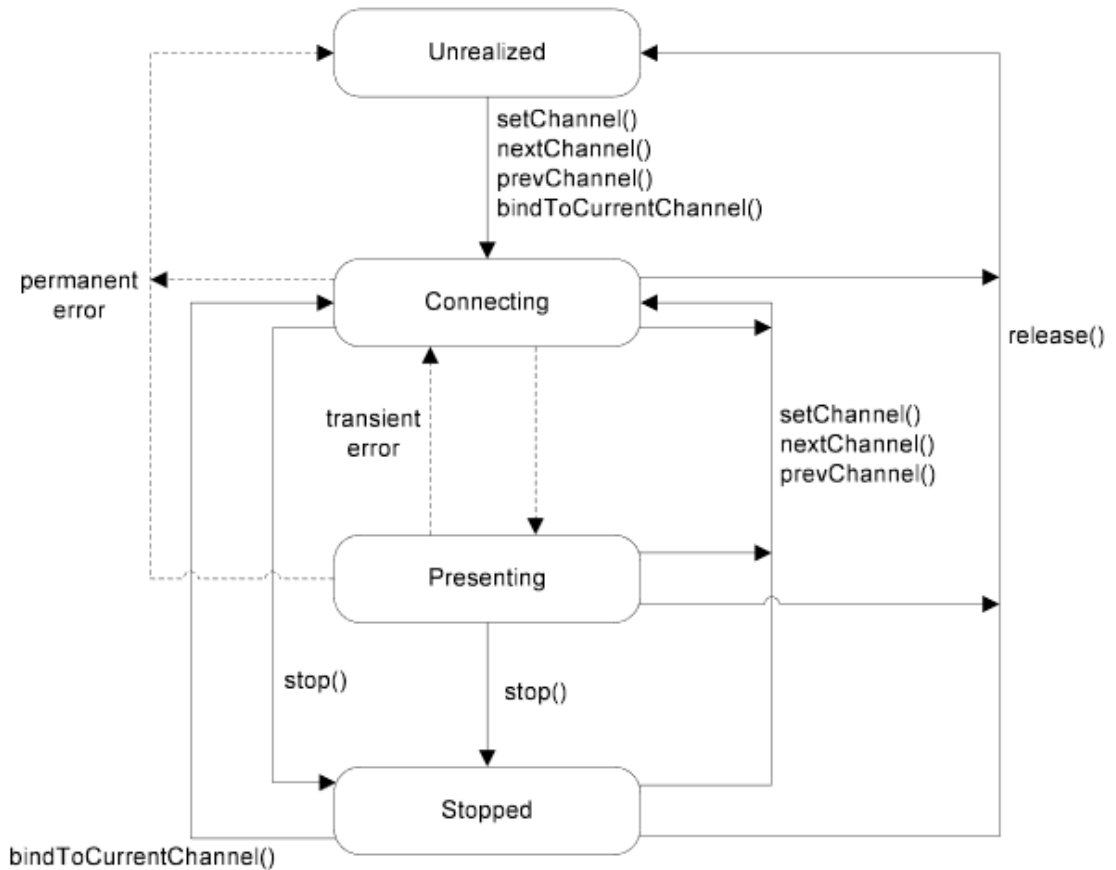
Kuvassa 7 on *video/broadcast* -objektin tilakone. Video alustuu ”Unrealized” tilaan kun elementti kytketään DOM:iin. Funktiolla *bindToCurrentChannel* valitaan videolähteeksi TV-lähetysten siirtovirrasta se kanava joka sillä hetkellä on viritettynä, jonka jälkeen objekti alkaa lukemaan ja piirtämään videokuvaa. Funktioilla *setChannel*, *nextChannel* ja *prevChannel* voidaan vaihtaa kanavaa. [23]

Muuta videolähdettä, kuten Internetin yli ladattavaa sisältöä varten käytetään kullekin videotyypille omaa mimetyyppiä objektin tyyppinä. Mimetyyppejä on useita erilaisia kaikille erilaisille videoformaateille. Mp4 videolle *video/mp4*, MPEG Transport Stream videolle *video/mpeg*, MPEG DASH -videolle *application/dash+xml*. Myös pelkälle audiolle on omat tyyppinsä, esimerkiksi Mp3 audioraita: *audio/mp3*.

Video-objekteilla on myös muita metodeja käytettävissä, kuten soittosijainnin asetus. Tämän avulla voidaan toteuttaa eteen ja taakse -kelaustoiminnot. Objektit osaavat myös puskuroida ja niillä on useampia tiloja (*video.playState*) jotka on eritelty listauksessa 1. Tiloja voidaan havainnoida tilan vaihtuessa suorittamalla tapahtuman käsittelijäfunktio. Listauksessa 1 asetetaan *<object id="video" type="video/mp4">* -objektille lähde-URL sekä täydennettävä tilanmuutoksen käsittelijä joka tunnistaa erilaiset objektin tilat, mutta ei sisällä varsinaista toiminnallisuutta. [14, s. 263]

HbbTV-standardin versiot ennen versiota 2.0 eivät tue HTML5-spesifisiä video ja audio -komponentteja. HbbTV 2.0 -standardiin on esitetty täydellistä HTML5-tukea,

joka mahdollistaa paremman videon- ja äänenkäsittelyn, kuten ääniraidan tarkemman analysoinnin esimerkiksi taajuusalueiden ja eri äänikanavien lukemisen, sekä manipuloinnin.



Kuva 7: Tilakone video/broadcast -objektille

```

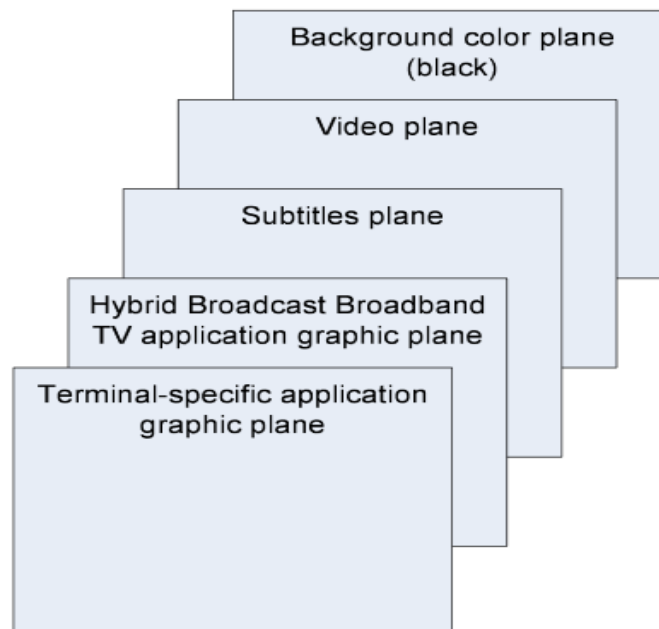
1 var video = document.getElementById('video');
2 video.data = "/testvideos/mp4/test7.mp4";
3 video.play(1);
4 video.onPlayStateChange = function() {
5     switch (video.playState)
6     {
7         case 0: // stopped
8         case 1: // playing
9         case 2: // paused
10        case 3: // connecting
11        case 4: // buffering
12        case 5: // finished
13        case 6: // error
14            switch (vidobj.error) {
15                case 0: // A/V format not supported
16                case 1: //cannot connect to server or lost connection
17                case 2: // unidentified error
18                case 3: // insufficient resources
19                case 4: // content corrupt or invalid
20                case 5: // content not available
21                case 6: // content not available at given position
22                case 7: // content blocked due to parental control
23            }
24        }
25 }
  
```

Listaus 1: Video-objektin alustaminen ja tapahtumien käsittely.

3.2.4 HbbTV-sovelluksen loogiset tasot

HbbTV-sovelluksen grafiikka ja videokuva jakautuvat eri tasoille. Kuvassa 8 on esitetty HbbTV-terminaalin loogiset kuvatasot [23, s. 46]. Alin taso on määritelty olevan musta tausta joka näkyy alta, mikäli muilla tasoilla ei ole sitä peittävää kuvaa tai grafiikkaan. Seuraavalla tasolla on videokuva, jossa näkyy TV-lähetys. Tämän tason resoluutio on television suurimman resoluution mukainen. Seuraava taso on tekstityksille, jotka luetaan videon kanssa toisena komponenttina siirtovirrasta. Neljännellä tasolla on itse HbbTV-sovellus. HbbTV-sovellus pystyy siirtämään videotason myös ylemmäs, ei kuitenkaan omaa tasoaan ylemmäksi. Ylimmällä tasolla on aina laitteen oma graafinen piirtotaso. Television natiivit valikot ja grafiikat piirtyvät aina päällimmäiseksi, eikä HbbTV-sovellus voi piirtää sen päälle.

Kuvassa 9 on esimerkkikuva jossa näkyy kaikki piirtotasot. Taaimmaisena on musta tausta (Background color plane). Sitä ylemmällä tasolla on videotaso (Video plane) joka on skaalattu HbbTV-sovelluksensa ohjelmallisesti pienemmäksi kuin normaali koko, joka peittää koko taustatason. Videotason päällä on tekstitystaso (Subtitles plane). Tekstitys on usein poltettu videokuvan yhteyteen, mutta tässä se on omana komponenttinaan videokuvan päällä. Tekstitason päällä on HbbTV-sovellustaso (HbbTV application graphic plane), jossa on nähtävillä videosoitin-elementti alempien tasojen päällä. Päällimmäisenä on avattuna television natiivi ohjelmalista, joka piiryy aina kaikkein päällimmäiseksi ja myös videosoitin, kuten muut HbbTV-sovelluksen elementit jäävät sen taustalle.



Kuva 8: HbbTV-terminaalin loogiset piirtotasot



Kuva 9: Esimerkki: HbbTV-terminaalien loogiset piirtotasot

3.2.5 Stream eventit

Stream eventit ovat keino välittää tietoa broadcast-lähetyksen kautta. Niiden hyöty ilmenee parhaiten nopeassa tiedonsiirrossa palvelimelta tai TV-lähetyksestä TV-katselijoille TV-ohjelmaan liittyvien tapahtumien näyttämiseen. Ilman Stream Eventejä vastaavan toiminnallisuuden toteuttaminen HbbTV:llä vaatisi asiakassovelluksilta palvelimen pollaamista Internetin yli saadakseen tiedon tapahtumasta. Kun katsojia voi olla samanaikaisesti useita tuhansia, se vaatisi hyvin suuren palvelinkapasiteetin, sekä todennäköisesti hidastaisi tiedonkulkua. Stream eventit signaloidaan objektikarusellissa.

Stream eventejä voidaan käyttää ohjelmaan kuuluvien tapahtumien lähettämiseen TV-päätelaitteille. Tällaisia voivat olla esimerkiksi urheiluohjelman aikana tapahtuvat asiat joista voidaan jakaa lisätietoa reaaliaikaisesti sen lisäksi mitä TV-ruudulla näkyy. Toinen esimerkki voisi olla uutislähetys, jossa kuhunkin lähetyksessä käsiteltyyn asiaan liittyviä uutisia voi lukea stream eventtinä lähetetyn ruudun reunaan ilmestyvän ”Lue lisää tästä aiheesta” -tyyppisen linkin kautta jollakin napilla, jolloin lisätiedot voidaan näyttää suoraan ruudulla.

Stream eventit toimivat myös ilman Internet-yhteyttä, joten tiedot voidaan lähettää kokonaan broadcastin mukana ja näin voidaan toteuttaa reaaliaikaista informaatiota sisältäviä interaktiivisia HbbTV-sovelluksia myös ilman Internetiä. Tietynlainen HbbTV-sovellus, joka ei vaadi paluukanavaa, voidaan siis toteuttaa hyvin ilman internetiä. Tällainen voisi olla tietoruutu-tyyppinen sovellus jonka tehtävänä on näyttää tietoja jotka liittyvät televisiolähetykseen.

Paluukanavaton Stream eventtejä hyödyntävä HbbTV-sovellus voi olla esimerkiksi urheilukilpailun seurantaan liittyvä lisäpalvelu. Tv-lähetykseen voidaan signaloida kilpailun ajaksi oma HbbTV-sovelluksensa, joka saa Stream eventeinä tietoja kilpailun tilanteesta. Interaktiivisuutta voidaan toteuttaa käyttäjän ja television välille siten, että hän voi valita, minkälaisia tietoja hän haluaa nähdä ruudulla. Hän voi myös sulkea HbbTV-sovelluksen halutessaan ja katsoa pelkkää lähetystä.

3.2.6 OIPF DAE

OIPF DAE (Declarative Application Environment), määrittelee HbbTV-sovelluksissa käytettävissä olevia olioita ja niiden rajapinnat. HbbTV:tä tukevien selainten ja selainkomponenttien tulee noudattaa tässä dokumentissa määritettyjä pakollisia ominaisuuksia. Jotkin määrittelyn rajapinnoista eivät ole pakollisia vielä HbbTV 1.5 -standardissa. [14]

OIPF-oliot määritellään CE-HTML -dokumenteissa object-elementin avulla määrittelemällä tyypiksi jokin OIPF-luokka. HbbTV-ohjelmoinnissa tärkeimmät OIPF-rajapintaluokat ovat seuraavat:

- *oipfApplicationManager*. Luo, alustaa, aktivoi, hallinnoi ja poistaa sovelluksen.
- *oipfConfiguration*. Tarjoaa tietoa päätelaitteesta, sen asetuksista, sekä tietoa sen tukemista komponenteista, kuten onko laitteessa PlayReady DRM-tuki tai CI-moduuli [14, s82].
- *OipfSearchManager*. Tarjoaa TV-ohjelmien metatietojen hakurajapinnan. Rajapinnasta voidaan hakea samoja tietoja mitä näkyy perinteisessä ohjelmaoppaassa, esimerkiksi kaikkien kanavien ohjelmat kahden viikon ajalta [14, s171].
- *OipfDRMAgent*. Mahdollistaa DRM-suojattujen digitaalisten tallenteiden katsomisen.

Seuraavassa listauksessa on koodiesimerkki OIPF-objektien käyttöönotosta CE-HTML -dokumentin body-osassa. Listauksessa 2 määritetään kolme objektielementtiä, joiden type-parametri määrittää mikä upotettu OIPF-objekti on kyseessä.

```
1 <object id="appmgr" type="application/oipfApplicationManager"></object>
2 <object id="oipfcfg" type="application/oipfConfiguration"></object>
3 <object id="srchmgr" type="application/oipfSearchManager"></object>
```

Listaus 2: OIPF-objektit HTML DOM:ssa.

Attribuutilla *Id* ei ole olioon merkitystä, mutta sen avulla JavaScript:llä getElementById -funktiolla voidaan valita HTML-elementti, jonka jälkeen sitä on helppo käyttää ohjelmallisesti. HbbTV-sovellus luodaan *oipfApplicationManager* -luokan instanssin avulla, jonka jälkeen sen rajapinta on kutsuttavissa. Tyypillinen ohjelman alustaminen (Listaus 3) voidaan tehdä esimerkiksi näin:

```

1 try{
2   var appmgr = document.getElementById("appmgr");
3   app = appmgr.getOwnerApplication(document);
4   registerKeys(0);
5   registerKeyListener();
6   app.show();
7   app.activate();
8 } catch{ /* halt */ }

```

Listaus 3: HbbTV-sovelluksen alustaminen.

Esimerkissä riveillä 4 ja 5 kutsuttavien itsemääriteltävien funktioiden tarkoituksena on määrittää, mitä ohjainkomentoja HbbTV-sovellus kuuntelee ja miten se reagoi niihin. *registerKeys* -funktion tarkoitus on määrittää *oipfApplicationManager*:lle tilanteesta riippuen, mitä näppäinkomentoja se kuuntelee ja mitä ei. Tyypillisesti HbbTV-sovellus ei varaa mitään muita TV:n käyttöön tarvittavia kaukosäätimen painikkeita kuin sen käynnistämiseen tarvittavat napit, silloin kun sovellus on taustalla. Esimerkiksi funktio voisi olla tällainen, jossa parametrin avulla määritellään bittikuvio jonka avulla *oipfApplicationManager* varaa halutut painikkeet käyttöön:

```

1 function registerKeys(mode) {
2   var mask;
3   // ui hidden, only color buttons registered
4   if (mode == 0) {
5     mask = 0x1 + 0x2 + 0x4 + 0x8; // color buttons
6   }
7   else {
8     mask = 0x1 + 0x2 + 0x4 + 0x8 + 0x10 + 0x20 + 0x40
9           + 0x80 + 0x100;
10  }
11  try {
12    app.privateData.keySet.setValue(mask);
13  } catch (e) {}
14 }

```

Listaus 4: Kaukosäätimen näppäinten varaaminen HbbTV-sovelluksen käyttöön.

3.2.7 Suunnitteluohjenuorat

Erityisesti HbbTV-sovelluksia varten, mutta myös pitkälti muitakin TV-sovelluksia varten on olemassa ohjesääntöjä ja suosituksia siitä, miten sovellukset tulisi suunnitella. HbbTV-sovelluksen tulisi käyttäytyä standardin mukaisesti, kuten teknisessä määrittelyssä on määritelty. Seuraavat asiat tulee ottaa huomioon sovelluksen suunnittelussa ja toteutuksessa [23, s. 19]:

- ”Red button” -heräte. Sovellus käynnistyessään näyttää visuaalisen ilmoituksen saatavilla olevista HbbTV-palveluista. Ilmoituksessa kerrotaan että kaukosäätimen punaista nappia painamalla saadaan lisäpalveluita näkyville.
- Navigointi tapahtuu siten, että ruudulla olevat elementit näyttävät visuaalisesti selvästi kulloinkin aktiivisena olevan elementin ja nuolinäppäimillä voidaan vaihtaa loogisesti valittua elementtiä näkymässä siihen suuntaan mitä nuolinäppäintä painetaan.

- Nappien looginen käyttö. Navigointi nuolinapeilla kuten edellä mainittu, sekä värinapeilla tulee olla vakiotoiminnallisuus, mikäli mahdollista.
- Ison ruudun ja katseluetäisyyden huomioiminen.
- Kaukosäätimen rajoitteet.

3.2.8 Testaus ja yhteensopivuus

Testaus eri päätelaitteilla on tärkeää, koska HbbTV -sovellukset toimivat eri alustoilla, erilaisten laitteiden ja ohjelmistoversioiden päällä. Sovelluksia voidaan kehittää PC:llä kuten muitakin selaimessa toimivia HTML/JS -sovelluksia. HbbTV-sovelluksien kehitykseen ja testaukseen kehitetty Firefox-laajennos FireHbbTV [25] emuloi HbbTV-laitetta parhaansa mukaan, mutta se ei riitä testaukseen, vaan sovellukset on testattava eri laitteilla erikseen, jotta voidaan varmistua siitä, että eri laitteiden selainkomponentit toimivat oletetulla tavalla, sekä laitteiden suorituskyky on riittävä. Sovelluslaajennus on kuitenkin tärkeä apuväline kehittäjälle.

Kun laitteita on usein testattava erikseen, joudutaan laite kerrallaan testaamaan uudet ja muuttuneet ominaisuudet jokaisella laitteella. Kriittisintä on testata yleisimmillä laitteilla ja eri selainversioilla. Laitevalmistajat pystyvät näkemään television natiivien sovellusten tuottamat debug-viestit ja pystyvät laitteeseen kytketyltä tietokoneelta seuraamaan ja ohjaamaan laitteen toimintaa. Tällaista mahdollisuutta ei ole HbbTV-kehityksessä. Yksi yksinkertainen tapa nähdä laitteen selaimen tiedot ja ohjata sitä etänä tietokoneelta on alun perin mobiililaitteiden kehitykseen suunniteltu, mutta myös muiden ”Black Box” -laitteiden kehityksen apuvälineeksi sopiva Jsconsole [26], jonka avulla voidaan nähdä JavaScript-konsolin tulosteet, sekä sovelluksen DOM. Koska Jsconsole on kaksisuuntainen, voidaan laitetta myös ohjata tietokoneelta.

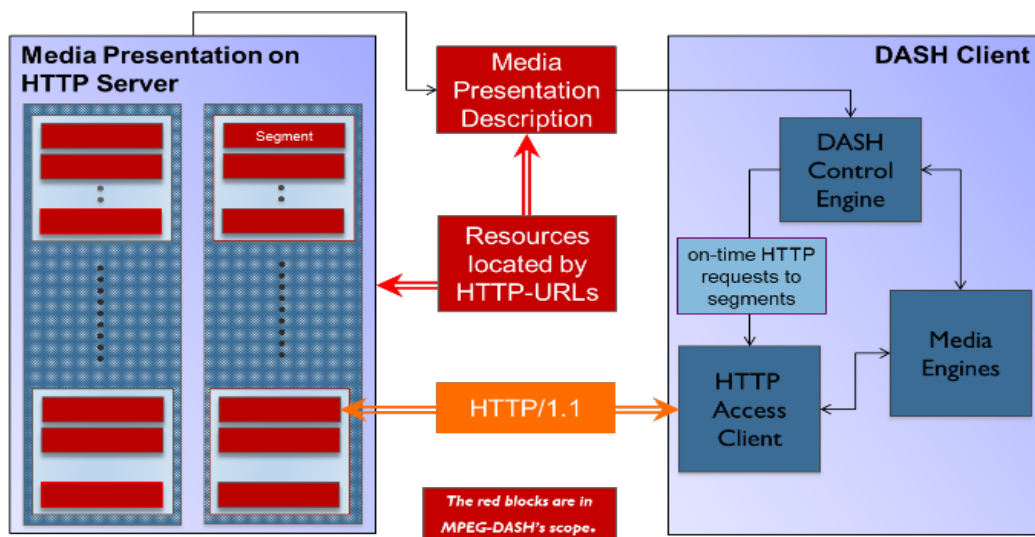
3.3 MPEG DASH

MPEG DASH (Dynamic Adaptive Streaming over HTTP), on standardi, joka määrittelee miten videovirtaa voidaan ohjata ja sen parametreja muuttaa päätelaitteen ominaisuuksien ja interaktion kautta [16]. MPEG DASH on HbbTV-standardissa määritelty tuettavaksi. [23, s.44-45]. MPEG DASH -videot muodostuvat useammasta mediatiedostosta jotka on segmentoitu, sekä määrittelytiedostosta, joka määrittelee saatavilla olevat komponentit ja niiden attribuutit. MPEG DASHin toiminta perustuu siis kahteen HTTP-yhteyteen, jotka ovat seuraavat [27, s. 7-9]:

1. XML-muotoisesta manifestitiedoston haku (.mpd) jonka avulla MPEG DASHia tukeva videosoitinkomponentti voi avata A/V -suoratoiston (**Media Presentation Description**)
2. Mediasuoratoisto, joka avataan videosoitinkomponentilta.

DASH on riippumaton videon enkoodauksesta. MPEG DASH siis tukee videonkoodauksessa muun muassa H.264 AVC (Advanced Video Coding) pakkausstandardia, joka on julkaistu vuonna 2003 ja on nykyään laajalti käytössä videokuvan pakkauksessa. H.265 HEVC (High Efficiency Video Coding) on H.264:n seuraaja, joka pystyy noin kaksinkertaistamaan pakkausration edeltäjäänsä nähden. HEVC on tulevaisuudessa yleistävä standardi, mutta sitä tukevat laitteet eivät ole vielä kovin yleisiä. MPEG DASH tukee molempia ISO-perustaisia tiedostomuotoja MP4 (.mp4) ja MPEG 2 Transport Stream (.ts) [28]

MPEG DASH toimii XML-manifestien avulla. Kuvassa 10 on havainnollistettu MPEG DASHiin liittyvät komponentit ja niiden välinen toiminta. MPEG DASH asiakassovellus ensin lataa kuvaustiedoston mediasisällöstä (XML-manifesti), jonka jälkeen lähettää HTTP pyynnön tiedoston perusteella saatavilla olevan suoratoistovirran URL:iin, josta avataan siirtovirta asiakassovellukselle HTTP-vastauksena. Videosoitinena (Video Client) toimii CE-HTML object, jonka mimetyyppinä on määriteltä *application/dash+xml*.



Kuva 10: MPEG DASH - Standardin komponentit [16]

MPEG DASH-videoita voidaan generoida erilaisilla työkaluilla muun muassa yleisestä MP4 -videomuodosta MP4Box -komentoriviohjelman avulla. Komennolla `MP4Box -dash 1500 -out Manifest.mpd "input.mp4"#audio:id=audio "input.mp4"#video:id=video` luetaan videotiedoston `input.mp4` audio- ja videokomponentit, ja muodostetaan segmentoidut MPEG DASH-videotiedostot, sekä XML-manifesti. [29] Tavallisesti manifesti, sekä media sijoitetaan samaan hakemistoon, jonka nimeksi annetaan sama kuin manifestitiedoston nimi ilman tiedostopäätettä. Näin yhteen MPEG DASH-videoon liittyvät mediatiedostot löytyvät samasta hakemistosta.

3.4 DRM-Tekniikat

DRM on tärkeä huomioida palvelussa, jossa on sisältöä, jota koskee tekijänoikeudet ja/tai sisältö on maksullista. MPEG DASH ei määrittele DRM:ää mutta tukee kaikkia ISO/IEC 23001-7 Common Encryption -määritettyjä vaihtoehtoja Microsoft PlayReady, Widevine, sekä Marlin. [28]

Sisällönsuojaus DASH MPD-manifestissa voidaan määrittää ContentProtection -elementin avulla. ContentProtection -elementti sisältää schemeIdUri -attribuutin, joka kertoo, mitä sisällönsuojausta milloinkin käytetään. Esimerkiksi *schemeIdUri="urn:uuid:9A04F079-9840-4286-AB92-E65BE0885F95"* määrittelee käytössä olevan Microsoft PlayReady DRM [30]. Sisällön suojaukseen voidaan käyttää useampia DRM-tyyppejä, jotka yksilöidään schemeIdUri -attribuutin avulla. Mikäli useita DRM-vaihtoehtoja on määritetty, mikä vain niistä riittää sisällön avaamiseen [27, s.67-69]. Seuraavassa esimerkissä (Listaus 5) on havainnollistettu MPEG-DASH-MPD-manifestin sisällönsuojausmäärittelyksiä:

```

1      <!-- Common Encryption -->
2      <ContentProtection
3          schemeIdUri="urn:mpeg:dash:mp4protection:2011"
4          value="cenc"
5          cenc:default_KID="5CF725ED-7C2A-4ACB-99D9-028564608B37">
6      </ContentProtection>
7      <!-- PlayReady -->
8      <ContentProtection
9          schemeIdUri="urn:uuid:9A04F079-9840-4286-AB92-E65BE0885F95"
10         value="MSPR 2.0">
11     </ContentProtection>
12     <!-- widevine -->
13     <ContentProtection
14         schemeIdUri="urn:uuid:EDEF8BA9-79D6-4ACE-A3C8-27DCD51D21ED">
15     </ContentProtection>
16     <!-- Marlin -->
17     <ContentProtection
18         schemeIdUri="urn:uuid:5E629AF5-38DA-4063-8977-97FFBD9902D4">
19         <mas:MarlinContentIds>
20
21         <mas:MarlinContentId>urn:marlin:kid:5cf725ed7c2a4acb99d9028564608b37</
22         mas:MarlinContentId>
23         </mas:MarlinContentIds>
24     </ContentProtection>

```

Listaus 5: Esimerkki: MPEG DASH -manifestin DRM-määrittelyt.

Eri DRM-vaihtoehdot voivat toimia eri tavoin. Hyvä ja paljon käytössäoleva PlayReady DRM, tarvitsee 64-kantaiseksi enkoodatun lisenssiavaimen, joka on HbbTV-soveluksessa erikseen noudettava ja liitettävä manifestiin. Lisenssin hakua ei ole vielä toteutettu HbbTV-sovelluksissa, mutta se onnistuu lähettämällä xml-muotoinen HTTP-pyyntö lisenssin URL-osoitteeseen. Vastauksena PlayReady-palvelin lähettää toimivan lisenssiavaimen <License>-tagissa (Liite 2) joka voidaan lisätä MPD-tiedoston <mspr:pro>-tagiin. Seuraavassa esimerkissä (Listaus 6) <mspr:pro> -elementin sisään tuleva 64-kantaiseksi koodattu lisenssiavain on lyhennetty tilan säästämiseksi:

```
1 <ContentProtection schemeIdUri="urn:uuid:9a04f079-9840-4286-ab92-  
   e65be0885f95" value="2.0">  
2   <mspr:pro>ngMA..AgA=</mspr:pro>  
3 </ContentProtection>
```

Listaus 6: Playready-lisenssiavain MPEG DASH-manifestissa

3.5 Käyttäjien tunnistaminen

Käyttäjä on tunnistettava, kun halutaan tarjota maksullisia palveluita. Tähän on käytössä useampia tapoja, mutta HbbTV-sovelluksissa käytännöllisimpänä voidaan pitää kahta eri tunnistusmekanismia: TV:n CI-moduulipaikkaan asetettavalla TV-maksukortilla tunnistautuminen, sekä tekstiviestitse TV-laitteen parittaminen puhelinnumeroon, jolloin laskutus voidaan tehdä puhelinliittymän laskuun.

4 TILAUSVIDEOPALVELU

Tilausvideopalvelu on palvelu, josta käyttäjä voi valita haluamansa videon toistettavaksi päätelaitteessa. Tavallisesti tilausvideopalvelut ovat olleet laitevalmistajien toteuttamia sovelluksia digitelevisioissa ja digibokseissa. Myös pc:lle suunnattuja samantyyppisiä palveluita on ollut jo pitkään internetissä. Yhteistä kaikille tilausvideopalveluille on se, että videoiden toistamiseen tarvitaan internet-yhteys vaikka itse sovellus muuten olisikin natiivi.

Tässä luvussa kuvataan tilausvideopalvelu yleisesti, siihen liittyvät yleiset vaatimukset, sekä HbbTV:n kannalta tärkeät asiat joita tulee ottaa huomioon. Kohdassa 4.1 kuvataan yleisesti, mikä on tilausvideopalvelu ja minkälaisia yleisiä variaatioita palvelusta on olemassa. Kohdassa 4.2 kuvataan tilausvideopalvelun yleiset toiminnalliset vaatimukset, DRM- ja autentikointivaatimukset, alueellisten oikeuksien vaatimukset, sekä käyttöliittymävaatimukset.

4.1 Yleinen kuvaus

Tilausvideopalvelu on ohjelmistopalvelu (SAAS), jonka loppukäyttäjiä ovat kaikki TV-käyttäjät, jotka saavuttavat palvelun sisällön. Kuvassa 11 on esitetty tilausvideopalvelun yleinen käyttötapauskaavio. Kuvan käyttäjän toiminnot on esitetty sinisissä käyttötapausvaihtoehdoissa josta seuraa vihreällä ympyröity toiminnallisuus.

Palvelutyyppejä, alustoja ja laskutustyyppettä on erilaisia ja niitä on kuvattu tässä luvussa. Näistä voi myös olla yhdistelmiä, mutta jaottelu on esimerkinomainen otos. Tilausvideopalvelualustoja on eri tyyppisiä, joista TV-palveluita voivat olla seuraavat:

1. pakettina myytävään viihdepalveluun kuuluva digiboksi, jossa on sisäänrakennettu tilausvideopalvelu. (Elisa Viihde, Sonera Viihde). Pakettiin kuuluu yleensä Internetliittymä sekä digiboksi. (Natiivi, järjestelmätaso)
2. Smart TV-sovelluksena toteutettu laitevalmistajakohtainen palvelu (Natiivi)
3. HbbTV-sovelluksena toteutettu televisiokanavakohtainen palvelu (HbbTV).

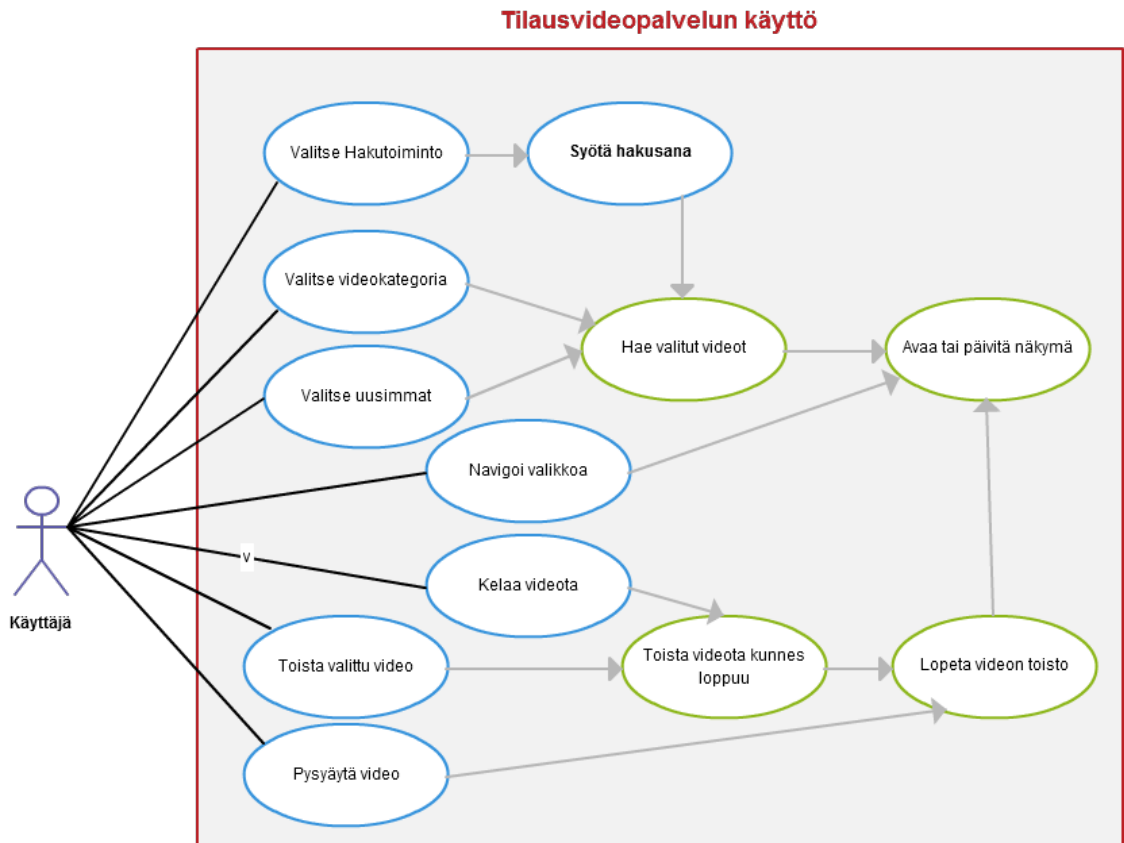
Tilausvideopalvelutyyppejä IP-jakeluteitse on erilaisia:

1. tilausvideopalvelu, jossa voi olla mitä vain videoita saatavilla (VoD)
2. Catchup TV, televisiokanavalla näytettyjen jaksojen katsominen myöhemmin
3. reaaliaikainen suora videolähetys.

Palvelun laskutustyyppettä on erilaisia:

1. Palvelu on täysin ilmainen (Suomessa esimerkiksi Internetpalvelut: Yle Areena, MTV3 Katsomo, Ruutu, Liv Play) (Palveluissa voi olla lisäksi myös maksullista sisältöä).

2. Yksittäisestä videosta maksetaan (Suomessa esimerkiksi: Elisa Viihde, Sonera Viihde). Tilausvideot yleensä maksavat noin 5 euroa, mutta ne voivat olla ilmaisiakin. palvelun käytöstä veloitetaan samassa laskussa kuin internetistäkin kuukausittain.
3. Palvelun käytöstä maksetaan kuukausimaksua (Netflix, Viasat).



Kuva 11: Tilausvideopalvelun käyttö

Tilausvideopalvelun ansaintalogiikka vaihtelee. Erilaiset palveluntarjoajat ja palvelualustat määrittelevät miten palvelusta voi hyötyä rahallisesti. Ilmaisisissa palveluissa (Alusta 4) yleensä raha tulee mainoksista, mutta ei aina. YLE:n tapauksessa mainoksia ei käytetä, vaan rahoitus tulee verorahoista, nykyään YLE-verosta. Muissa ilmaisisissa palveluissa, jotka ovat yleensä Catchup-tyyppisiä, rahoitus tulee mainosten näyttämällä, kuten televisiossa.

4.2 Vaatimukset

Vaatimukset HbbTV-tilausvideopalvelulle on määritelty useampien asiakkaiden toimesta, eikä kaikissa palveluissa ole samanlaisia vaatimuksia. Seuraavissa alakohdissa on määritelty yleiset vaatimukset mitkä koskevat tai voivat koskea HbbTV-tilausvideopalvelua. Käytännössä vaatimukset voivat olla määritelty tarkemmin

sovelluskohtaisesti ja vaatimuksiin voivat vaikuttaa asiakkaan omat järjestelmät, kuten DRM-ratkaisuiden käyttö ja niiden valinta.

4.2.1 Toiminnalliset vaatimukset

Sovelluksen avulla voidaan katsoa käyttäjän valitsemaa videosisältöä. Koska käyttäjä valitsee toistettavan videon, on sovelluksen oltava interaktiivinen ja sen on toteutettava kaksisuuntainen kommunikointi videolähteeseen. Tästä syystä sovellus tarvitsee paluukanavaksi internet-yhteyden, vaikka itse sovellus voitaisiinkin signaloida DSM-CC -tiedostokarusellissa. Teoriassa, kuten muitakin tiedostoja, videotakin voisi välittää sitä kautta, mutta käytännössä ei, koska silloin kaikki videot olisi lähetettävä laitteelle, ja ne vievät liikaa tilaa. Päätelaitteelta vaaditaan siis riittävän nopea laajakaistayhteys, jotta videoiden toisto on mahdollista Internetin yli. Koska sovellus toimii Internetin kautta, se myös ladataan sovelluksen ulkoisen URL:n mukaan, joka joko avataan linkillä toisesta sovelluksesta tai käynnistetään suoraan AIT-taulusta. Käytännössä sovellus käynnistetään aina linkillä, eikä automaattisesti kanavalle tultaessa.

Sovelluksen tulee toimia määrätyillä laitteilla, jotka ovat yleensä kaikki HbbTV 1.2.1 -laitteet, sekä testatut ja valitut HbbTV 1.1.1 -laitteet. Mikäli sovellus ei toimi jollain laitteella, vaikka se olisi kelvollista HbbTV-sovelluskoodia, voidaan ongelmalliset tapaukset joko ratkaista laitekohtaisesti tai jättää tukematta kyseistä laitetta palveluntarjoajan päätöksen mukaan.

Sovelluksissa on erilaisia vaatimuksia käyttöliittymän toiminnasta ja muista seikoista. Tärkeimmät toiminnallisuudet ja ominaisuudet, jotka tilausvideopalvelussa tulee olla toteutettuna, on listattu alla:

- Sovellus avataan linkillä haluttaessa erillisestä *Launcher*-sovelluksesta, joka käynnistyy kanavalla *autostart*-parametrilla automaattisesti.
- Kaukosäätimellä navigoitava valikkorakenne.
- Etusivuvalikot: Nostovideot (promo), Uusimmat videot (viimeisimmät).
- Joko ohjelmakategoriavalikot, tai kanavavalikot.
- Hakutoiminto.
- Videosoitin, joka pystyy toistamaan mp4-, ts- ja mpd-videoita.
- Videoita on voitava kelata eteen ja taakse, sekä pysäyttää ja keskeyttää katselu.
- Ennen valitun videon toistoa voi olla mainoksia, joita ei voi ohittaa eikä kelata.

4.2.2 DRM-Vaatimukset

Sekä maksullisissa että ilmaisissa palveluissa, joissa käyttäjä saa käyttöönsä tekijänoikeudella suojattua materiaalia, tulee huolehtia siitä, ettei materiaalia voi laittomasti kopioida ja levittää. Tätä kutsutaan digitaaliseksi käyttöoikeuksien hallinnaksi (DRM), Engl: Digital Rights Management. DRM:n sovellusalueita ovat muun muassa elokuvat, TV-ohjelmat, musiikki, sekä tietokoneohjelmat, jotka ovat

digitaalisessa muodossa. Kuitenkin vaatimuksena tätä ei vielä ole ollut nykyisten projektien sisällä, eikä sitä ole vielä toteutettu ja käyttöön otettu toteutettavissa ja toteutetuissa HbbTV-tilausvideopalveluissa. Nykyisten projektien HbbTV-sovelluksissa ei ole vielä vaadittu, eikä toteutettu DRM:ää, mutta kehitteillä oleviin uusiin, tai päivitettyihin sovelluksiin, on tulossa PlayReady-DRM, kun palveluiden sisällöntarjoajat, eli asiakkaat, saavat valmiiksi oman osuutensa DRM-palvelimesta.

4.2.3 Autentikointi

Käyttäjät on voitava autentikoida, mikäli palvelusta on tilattavissa maksullista sisältöä. Autentikointiin, eli käyttäjän tunnistamiseen tarvitaan jokin käyttäjätili ja tapa todentaa käyttäjän olevan tilin omistaja. Tilin avulla voidaan suorittaa veloitus useammalla menetelmällä, kuten luottokortilta tai kirjaamalla veloitus matkapuhelinlaskuun. Maksutapoja voivat myös olla yleisesti internetissä käytettävissä olevat maksupalvelut, mutta ne eivät ole TV-maailmassa kovinkaan käyttäjäystävällisiä.

CI -moduuli on yksi paljon käytetty TV:ssä oleva tai siihen hankittava komponentti, johon voidaan hankkia maksukortti. Maksukortin avulla voidaan tunnistaa käyttäjä ja veloittaa maksuja palveluista. Maksukortteja käytetään nykyään maksullisten TV-kanavien salauksen purkamiseen vain palvelusta maksaneille asiakkaille, joilla on maksukortti. HbbTV-rajapinta mahdollistaa myös CI-moduulin käytön, jonka avulla maksuja voidaan rekisteröidä. CI-moduulia on testattu ja siitä on tehty demosovellus HbbTV:lle, mutta sitä ei ole otettu missään tuotantosovelluksessa käyttöön.

Tämän hetkisten vaatimusten mukaan tullaan tulevaisuudessa näillä näkymin toteuttamaan ensimmäisenä olemassa olevan palvelun sisäänkirjautuminen käyttäjätunnuksella ja salasanalla. Näin Internetpohjaisen tilausvideopalvelun käyttäjätunnusta käytetään myös televisiolla autentikointiin ja saadaan myös maksullista sisältöä HbbTV-sovellukseen. Maksullisesta sisällöstä ei siis makseta HbbTV-sovelluksen kautta, vaan siihen on olemassa omat mekanisminsa palveluntarjoajan kautta. HbbTV-sovellus kirjautuu käyttäjätunnuksella palveluun, jolloin maksetut palvelut voivat näkyä myös HbbTV-sovelluksessa. Ilman käyttäjätunnusta saatavilla on ilmaiset tarjonnat.

4.2.4 Ohjelmistosisältöjen alueelliset oikeudet

Usein TV-ohjelmille, sarjoille ja elokuville lähetysoikeudet on lisensoitu alueellisesti. Tämän takia tilausvideoita katsovan vastaanottimen maantieteellinen sijainti voi vaikuttaa siihen, saako sisältöä näyttää. Maantieteellinen sijainti voidaan selvittää valtion tarkkuudella n. 95-99% varmuudella IP-osoitteen perusteella [31].

Täysin varmaa tapaa selvittää laitteen sijaintia ei ole. Alueelliset rajoitukset on mahdollista kiertää käyttämällä välityspalvelinta, mutta HbbTV-sovelluksessa, kuten muissakin vastaavia vaatimuksia soveltavissa ohjelmistoissa tarkin mahdollinen paikannus on IP-pohjainen.

4.2.5 Käyttöliittymä

Käyttöliittymäsuunnittelussa yleisesti tärkeitä tavoitteita, jotka ovat erityisen tärkeitä myös HbbTV-sovellusten käyttöliittymissä, ovat **saavutettavuus**, **käytettävyys** ja **suorituskyky**. Lisäksi käyttöliittymäsuunnittelussa pitää ottaa huomioon loppukäyttäjän käyttämän laitteen erityisominaisuudet. Suurin osa älytelevisioista ovat kuvasuhteeltaan ja resoluutioltaan HD-laatua ja niitä ohjataan kaukosäätimellä, joissa on peruspainikkeet käytössä.

Käyttöliittymä on skaalattu laitteen ruudulle riippumatta laitteen todellisesta resoluutiosta ja kuvasuhde on vakio. Siksi HbbTV-sovelluksen käyttöliittymän tulee aina olla 1280*720px. Staattinen kuvasuhde helpottaa käyttöliittymän tekemistä ja varmistaa sen, että se näkyy samanlaisena joka laitteessa.

Suorituskyky on tärkeä osa-alue, joka tulee ottaa huomioon niin HbbTV-sovelluksissa, kuin muissakin sovelluksissa. Viihdekäyttöön suunnatussa palvelussa suorituskyky liittyy myös hyvin paljon käyttäjäkokemukseen. Sovellus ei saa jumiutua liian kuormittavan käyttöliittymän takia. Jotta sovellus toimisi myös heikommilla laitteilla, käyttöliittymäkoodi on pidettävä mahdollisimman kevyenä.

5 HBBTV-TOTEUTUS JA RATKAISUT

Tässä luvussa käsitellään tilausvideopalvelun HbbTV-sovelluksen ratkaisut. Sovelluksia on toteutettu eri asiakkaiden projekteihin. Kohdassa 5.1 esitellään työssä tarkasteltavat projektit. Kohdassa 5.2 määritellään projektien sidosryhmät. Kohdassa 5.3 ja sen alakohdissa määritellään HbbTV-sovelluksen toteutukseen liittyvät yksityiskohdat.

5.1 Projektit

Työssä tarkasteltavat tilausvideopalveluratkaisut kuuluvat useampaan projektiin. Projektit ovat eri TV-operaattoreille toteutettuja palvelukokonaisuuksia, jotka sisältävät vähintään saman tapaisen Launcher- ja tilausvideopalvelusovelluksen. Tilausvideopalvelu on kaikille TV-operaattoreille hyödyllinen palvelu, koska heillä on saatavilla palveluun kaikki se sisältö, mitä TV-lähetyksessä lähetetään. Tästä syystä samalla suunnittelumallilla voidaan toteuttaa useampia projekteja, joissa on samantyyppisiä vaatimuksia. Seuraavissa kohdissa käsitellään yleisemmällä tasolla useamman samantyyppisen projektin tuotteena syntyviä palveluita, sekä niiden vaatimusten mukaisten ongelmien ratkaisuja.

Sofia Digitalilla on syksyn 2014 aikana ja sen jälkeen samanaikaisesti toteutettu neljää eri projektia eri asiakkaille eri maista, joissa on mukana tilausvideopalvelu ja lisäksi useita muita sovelluksia. Monille asiakkaille yksi tärkeä tuote on tilausvideopalvelu, joka tässä vaiheessa on ns. Catchup-TV, eli palvelu, josta voi katsoa myöhemmin televisiolähetyksessä tulleita ohjelmia IP-verkon yli. Koska sovellukset voivat olla kanavakohtaisia, jokainen kanava voisi periaatteessa tuottaa omasta materiaalistaan tilausvideoita ja näin ollen kysyntää samantyyppisille sovelluksille mutta hieman vaihtelevin ominaisuuksin on paljon. Kaupallisten kanavien palvelut rahoitetaan mainoksilla, kuten broadcast-lähetyksissäkin. Jatkossa erilaisten maksullisten tilausvideoiden katsominen HbbTV:llä tulee mahdolliseksi. Olemassa olevia DRM-ratkaisuita integroidaan HbbTV-sovelluksiin ja testataan.

Toteutuksessa käydään läpi toteutettuja sovelluksia, jotka kaikki ovat rakennettu samalle generiselle pohjalle. Kaikissa eri versioissa käytetään samaa tietomallia ja datarajapintaa, hieman erilaisilla asiakassovellusrajapinnoilla ja räätälöidyillä käyttöliittymillä, joissa kuitenkin on samanlaiset perusratkaisut. Erot käyttöliittymien osalta ovat pitkälti CSS-variaatioita ja samantyyppisiä näkymiä eri parametreilla.

5.2 Sidoryhmät

HbbTV-tilausvideopalvelun tärkeimpiä sidoryhmiä ovat palvelun toimittaja, ylläpitäjä, tilaaja sekä tiedonlähteenä käytettävä taho joka tarjoaa tarvittavat rajapinnat tiedonsaantiin. Näissä projekteissa Sofia Digital on sovellusten ja myös joidenkin

laitteiden sekä ylläpidon ja kehityksen toimittaja. Tilaajina ovat maailmanlaajuisesti eri asiakkaat, joilla on tarve saada HbbTV-palveluita kanavilleen. Tietolähteitä on jokaisella asiakkaalla omasta materiaalistaan saatava metadata ja itse videot tai muu data. Lisäksi sovelluksiin mahdollisesti liittyvien komponenttien toteuttamiseen voidaan tarvita kolmansien osapuolien tarjoamia tietolähteitä ja rajapintoja.

Erilaiset sisällönhallintajärjestelmät ja pilvipalvelut, jotka toteuttavat videometadatan ja datan jakelun sekä rajapinnat, ovat tiedonsaantiin liittyviä kolmansia osapuolia, jotka voivat olla Sofia Digitalin yhteistyökumppaneita tai Sofia Digitalin asiakkaiden palveluntarjoajia. Tällaisia voivat olla muun muassa Azure [32], Brightcove VideoCloud [33], Akamai [34], sekä muut vastaavat palvelut.

Lisäksi yksi sidosryhmä, joka ei liity itse sovellukseen, mutta sen saatavuuteen on DVB-signaalin kuljettava osapuoli, joka huolehtii tarvittavasta infrastruktuurista, sekä siitä, että signaali kulkee muuttumattomana vastaanottiin. Infrastruktuuria ovat DVB-T-verkon linkkimastot, sekä niiden kaapeloinnit joista vastaa Suomessa Digita, sekä tyypillisesti muissa maissa jokin siihen valtuutettu valtionyhtiö ja/tai yksi tai useampi muu toimija. DVB-C -kaapeliverkossa on useampia toimijoita jotka potentiaalisesti voivat jaella HbbTV:n vaatiman AIT-tilun tulevaisuudessa DVB-C -signaalissa.

5.3 Toteutus

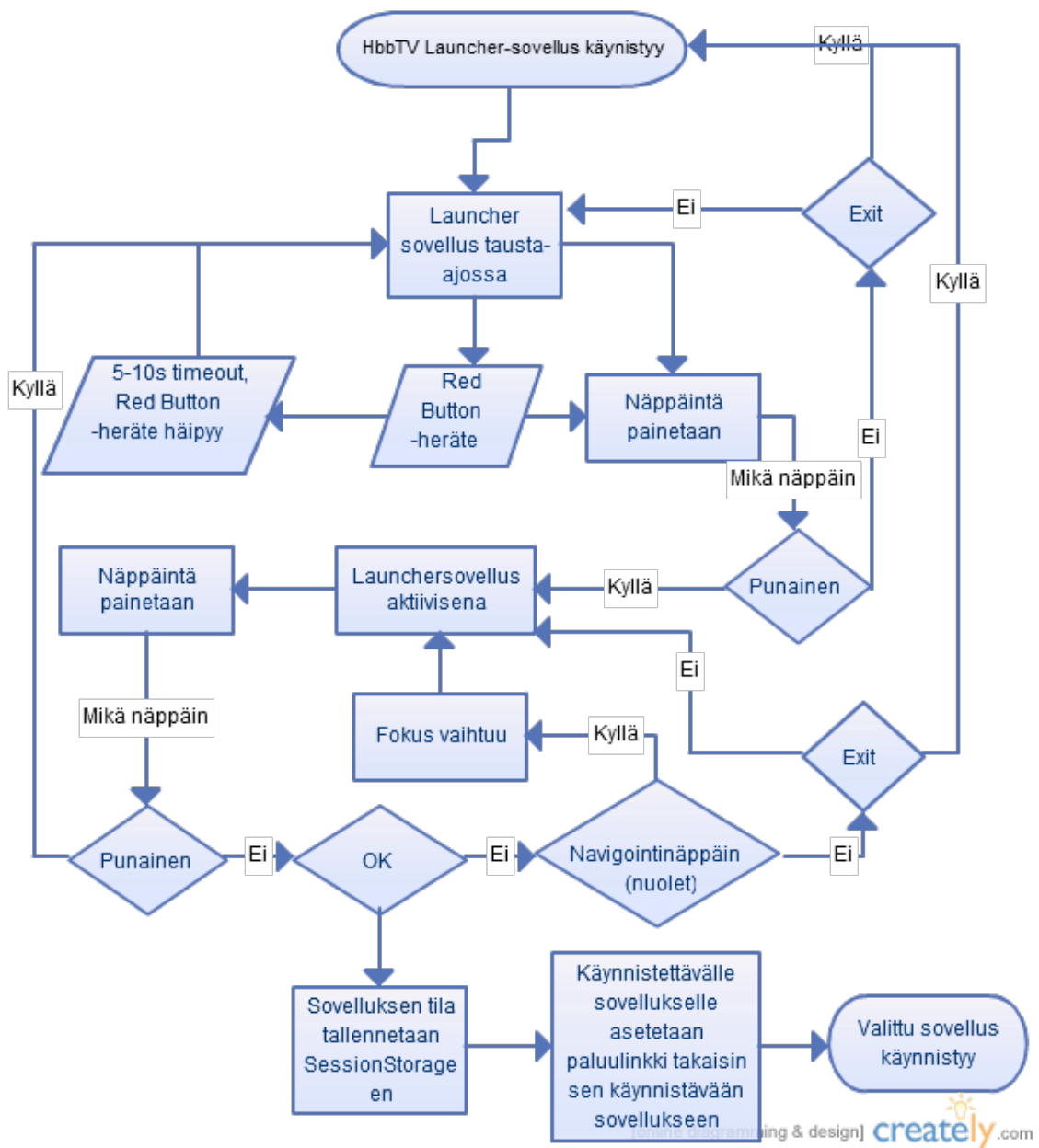
Tilausvideopalveluita on toteutettu useammalle asiakkaalle samalla sovellusarkkitehtuurilla ja pitkälti samoilla tekniikoilla. Teknisiä ongelmia ja eroja aiheuttavat erilaiset lähderajapinnat, sisällönsuojausmenetelmät (DRM), sekä videoformaatit ja mahdolliset lisenssit ja niiden käyttö HbbTV:ssä. Ulkoisen lähteen PlayReady-lisenssin liittäminen valmiiseen MPEG-DASH -manifestiin on uutta HbbTV:ssä eikä ole toteutettu aiemmin julkisesti. Seuraavissa luvuissa esitellään yleisemmällä tasolla järjestelmään liittyvät muut komponentit ja itse tilausvideopalvelu niiltä osin, mitkä ovat sen tärkeimmät ratkaisut HbbTV:n ja yleisesti tarkoituksenmukaisen, geneerisen ja käyttökokemusorientoituneen ohjelmistosuunnittelun kannalta.

5.3.1 Sovelluksen käynnistys

HbbTV-sovellus voidaan käynnistää automaattisesti, kun TV on kytketty Internetiin ja HbbTV-heräte tulee TV-kanavan lähetyksessä AIT-tilussa. Jotta yhdelle kanavalle saadaan monta sovellusta, tehdään niin sanottu Launcher-sovellus, eli päävalikko, josta saa muita sovelluksia käyntiin.

Tämä sovellus voidaan käynnistää AIT-tilun parametrien mukaan Internetistä tai DSM-CC-karusellista. Käytännössä Launcher, kuten muutkin sovellukset ladataan Internetistä. Vaikka itse Launcher ei tarvitsisi Internetiä, sen käynnistämät muut sovellukset todennäköisesti tarvitsevat. Siksi Launcher-sovellus asetetaan AIT-tiluun

ladattavaksi Internetistä korkeammalla prioriteetilla ja karusellista matalammalla prioriteetilla. Mikäli Internet-versiota ei pystytä lataamaan, käynnistetään karuselliversio, joka on signaloitu DSM-CC -karusellissa. Karuselliversio ei sisällä muiden sovellusten käynnistystä, vaan sen tehtävä on vain ilmoittaa käyttäjälle saatavilla olevista HbbTV-palveluista näyttämällä viestin, joka kehottaa kytkemään TV:n Internetiin.



Kuva 12: Tilakone: HbbTV Launcher

Launcher käynnistyy kun TV viritetään kanavalle, jossa on tämä sovellus AIT-aulussa parametrilla *autostart* (Kuva 12). Sovellusvalikko ei voi aueta suoraan TV-kuvan päälle, vaan ruudun reunaan asetetaan muutamaksi sekunniksi näkyville pieni

”Red Button” -heräte, josta ilmenee, että kanavalla on olemassa oheispalveluita ja ne saa päälle punaisesta napista. Punainen nappi on HbbTV-sovelluksissa aina käynnistys tai poistumisnappi. Teknisesti se voisi olla mikä vain, mutta on hyvän käytännön mukaista noudattaa yleistä käytäntöä. [23, HbbTV, s. 16-19]

Launcher-sovelluksen tehtävänä on mahdollistaa useiden muiden sovellusten käynnistäminen. Varsinainen tilausvideopalvelu käynnistyy siis omana instanssinaan erikseen. Kuvassa 12 näkyy tilakaavio Launcher-sovelluksen tilasta. Tila *Launcher*-sovellus on kevyt ja yksinkertainen, eikä siinä ole muita toimintoja kuin sovellusten selaaminen ja niiden käynnistäminen. Launcher kuitenkin avautuessaan voi tarjota muuta informaatiota, kuten kulloinkin lähetyksessä olevan ohjelman tiedot (Nimi, alkuaika, loppuaika, prosessi), seuraavaksi tulevan ohjelman tiedot, sekä TV-lähetyksestä riippumattomia tietoja, kuten säätiedot ja kellonaika. Lisäksi Launcher-sovellus pystyy keräämään jokaisesta television katsojasta tietoja vaikkei hän edes avaisi sovellusvalikkoa ollenkaan.

5.3.2 Tilausvideopalvelun käynnistyminen ja suoritus

Sovellus käynnistetään Launcher-sovelluksesta joko luomalla lähetyks kontekstista ulkopuolinen sovellus (broadcast independend application) HbbTV-sovellusta hallinnoiman *OipfApplicationManager*-komponentin *createApplication(url)* metodilla, tai siirtymällä sovelluksen URL-osoitteeseen, jolloin kanavakonteksti säilyy. Erona on se, että ilman lähetyks kontekstia, ei voida saada tietoja DVB-siirtovirrasta. Se tarkoittaa myös sitä, että standardin mukaan lähetyks ei näy enää, mutta todellisuudessa eri laitteet toimivat tässä tilanteessa erilailla. Yleisin tapa riippumatta sovelluksesta on avata sovellus aina linkkinä, jolloin konteksti säilyy uudessa sovelluksessa. Vaikka tilausvideopalvelu ei välttämättä tarvitse TV-lähetyks taustalle, tämä on silti parempi ratkaisu siinä mielessä, että se toimii laitteesta riippumatta samalla tavalla.

Sovelluksen runko on HTML-dokumentti joka voi olla generoitu PHP:n avulla. Sovelluksen runko pyritään tekemään kevyeksi, jotta se latautuu nopeasti. Kun DOM-rakenne on valmis, aloitetaan lataamaan varsinaista sisältöä käyttäjän valintojen mukaan.

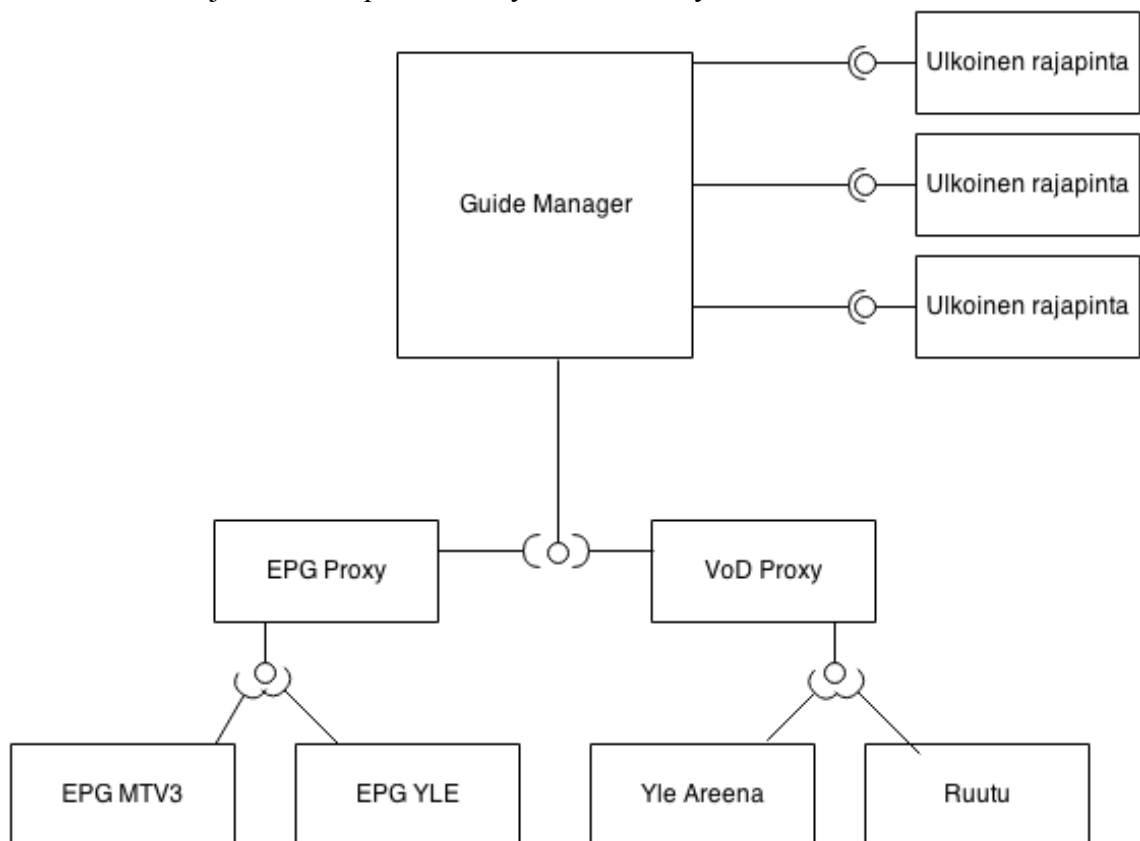
5.3.3 Sovelluksen arkkitehtuuri

Sovellus on toteutettu asiaks-palvelin -arkkitehtuurin mukaisesti, jossa on eriytetty *frontend*-asiakassovellukseen ja *backend*-palvelinsovellukseen (Kuva 13). Asiakassovellukset käyttävät tietojensa hankintaan joko OIPF-rajapintoja, kun halutaan hyödyntää TV-lähetykseseen signaloitua tietoa, tai palvelimella sijaitsevia rajapintoja, joiden avulla saadaan tietokannasta kolmansien osapuolien tietoja. Modulaarisuuden vuoksi, kaikki toteutetut tilausvideopalvelut lukevat saman muotoista JSON- tai XML-syötettä. Palvelimelle on toteutettu niin sanottuja Proxyjä, joilta samantyyppiset asiakassovellukset voivat hakea eri parametrein eri tietoja samassa formaatissa. Näin

asiakassovellukset pystyvät visualisoimaan erilaista sisältöä erilaisiin käyttöliittymiin ja niitä on helpompi monistaa. Kuvassa Proxyjen ja sovellusten nimet ovat havainnollistamistarkoituksessa esitetty näin, eikä tarkoita, että juuri tämän nimisiä palveluita olisi toteutettu.

Proxyt ovat palvelimen tarjoamia suoria yhteyksiä asiakassovelluksille, joiden kautta varsinaisia datarajapintoja käsitellään. Tämä on toteutettu siksi, että datalähterajapinnat eivät ole julkisia ja siellä käytetään sellaisia tunnuksia ja tietoja, joita ei voi asiakassovelluksen puolella näyttää selkokielisenä.

Sovellustyyppikohtaiset Proxyt tarjoavat siis ohjelmointirajapinnan asiakassovelluksille. Tämä on usein hyvin yksinkertainen ja sama proxy voi toimia hyvin useammallekin samantyyppiselle sovellukselle. *Proxyn* tarkoitus on lähinnä abstrahoida ohjelmadatarajapinta asiakassovelluksen näkymättömiin ja tarjota tietoturvallinen ja sovellusoptimoitu väylä tiedonvälitykseen.



Kuva 13: Sovellusten ja taustajärjestelmien rajapinnat

Palvelimella sijaitsevat Proxyt käyttävät jälleen yhdenmuotoista rajapintaa jonka avulla saadaan haettua varsinaisen datarajapinnan toteuttavalta TV-ohjelmien tietoja tarjoavalta Guide Manager -nimisestä järjestelmästä. Ohjelmatietoihin lukeutuu esimerkiksi ohjelmaopastiedot, sekä tilausvideopalvelutiedot (Ohjelman nimi, sarja, alkuaika, loppuaika, kuvake, video-URL, jne.).

Guide Manager puolestaan hakee tietoja omaan tietokantaansa ulkoisista kolmannen osapuolen rajapinnoista, kuten eri TV-operaattoreiden rajapinnoista. Jokaista erilaista rajapintaa varten Guide Manageriin on asennettava niin sanottu Connector joka jäsentää ulkoisen datalähteen tiedot tietokantaan. (Kuva 13). Guide Managerista saadaan sekä tulevat ohjelmatiedot, menneet ohjelmat, sekä muut mahdolliset tilausvideot yhdenmukaisessa JSON-formaatissa. Guide Managerilla taas on mahdollisuus täydentää tietosisältöään erinäisistä lähteistä, joilla on oma rajapintansa (Kuvassa ulkoinen rajapinta). Tällaisia lähteitä ovat eri asiakkaiden rajapinnat, joiden kautta voidaan päivittää tietoja Guide Manageriin. Nyt koska yksittäiset sovellukset eivät hae tietoja ensimmäisestä lähteestä, ne eivät ole suoraan riippuvaisia siitä, eikä jokaisen asiakkaan samantyyppisiin sovelluksiin tarvitse toteuttaa erilaista tiedon parsintaa, vaan tieto tulee Guide Managerista aina samassa formaatissa asiakassovelluksille.

Hyvät rajapinnat ja vastuualueiden jakaminen helpottavat suuremmankin järjestelmän toteutusta, sekä uusien sovellusten luomista samojen hyvien rajapintojen avulla. Etenkin Saman tyyppisen sovelluksen monistaminen voi olla hyvinkin helppoa, koska se ei aina vaadi muuta kuin eri parametrit asiakassovellukselle, sekä uuden Connectorin GuideManageriin. Tämän jälkeen käyttöliittymää voidaan muokata erilaiseksi, mutta perustoiminnallisuudet voidaan toteuttaa aina samalla tavalla.

5.3.4 Asiakassovellus ja käyttöliittymä

Eri TV-operaattoreille toteutetut tilausvideopalvelut ovat asiakassovelluksen rakenteen puolesta hyvinkin samanlaisia. Rakenne on HTML-runko, johon muodostetaan dynaamisesti erilaista sisältöä eri parametrein haettuna. Koko sisältöä on turha ladata käyttöliittymään, joten siihen ladataan käynnistymisen jälkeen asynkronisesti aina tietyn kokoisia näkymiä kerrallaan. Nämä on toteutettu valikkorakenteena, joissa yhden valikon rakenne voidaan ladata kerralla rajapinnasta nopeasti. Sovellusten päävalikkona toimii tyyppillisesti navigoitava kanava- tai kategorialista, josta valitsemalla voidaan avata uusia näkymiä ja ladata niiden sisältöjä ruudulle.

Katseltavat videot voidaan listata erilaisiin näkymiin ja videot ja näkymät voivat muodostaa syvempiä navigoitavia hierarkioita, kuten Kategoriat → Lapset → Haku → mno → Muumit, jonka jälkeen sovelluksen muistiin on ladattu vain kevyitä muutaman valinnan valikoita, sekä yksi listaus haetuista videoista kaikkine sisältöineen.

Kaikki tietorakenteet ovat samankaltaisia valikkorakenteita. Koska HbbTV-sovelluksia ohjataan yleensä jostakin valitusta elementistä tai tilasta toiseen, voidaan valikkorakenteella muodostaa mikä tahansa sovellus. Ruudulle visualisoitavat elementit ovat valikon alkioita. Jokaiseen valikon alkioon voidaan tallentaa attribuutteja tai funktioita jotka aktivoituvat kun alkio saa fokuksen.

Ruudulle piirrettävät navigoitavat ja valittavissa olevat valikot voidaan tallentaa samassa muodossa ja visualisoida eri lailla. Valikolla voi olla erilaisia tyyppisiä joka vaikuttavat sen visualisointiin ja miten siinä navigoidaan. Valikon tyyppistä riippuen

pystytään geneerisesti toteuttamaan erilaisia visualisointeja, sekä navigaatiomalleja. Valikot ovat JavaScript-listassa olevia tietorakenteita, jotka sisältävät kukin yhteen valikkoon liittyviä attribuutteja sekä listan siihen kuuluvista alkioista. Valikot kuvataan tässä JSON-formaatissa ja samoin ne myös siirretään sekä tallennetaan. Peruslistatyyppejä ovat seuraavat:

- **list** – Vaakasuuntainen tai pystysuuntainen lista jossa alkiot voivat sisältää tekstiä tai HTML:ää, eli myös muita HTML-elementtejä. Listaa selataan nuolinäppäimillä eteen ja taakse, tai ylös ja alas sen orientaatiosta riippuen. Vaakalistan alkioista voidaan avata alivalikkoja, jotka aukeavat yleensä sekä alaspäin, että OK-näppäimillä. Pystyorientoitunut lista toimii muutoin samoin, paitsi selaus ja alivalikon avaus tapahtuvat eri nuolinapeilla. Alla olevassa listauksessa 7 on alku valikkolistasta, jossa ensimmäinen valikko on päävalikko ja se sisältää kaksi alkiota, sekä määrittelyt minkä tyyppinen kyseinen päävalikko on (`type=list`), sekä kohdistettuna oleva alkiot (center=0):

```

1 [
2   {"center":0, "type":"list", "items":[
3     {
4       "cid":159,
5       "title":"UUDET",
6       "img":"http://loc.of.icon/gridfs/3538.jpg",
7       "submenu":40,
8       "load":true
9     },
10    {
11      "genre":"lapset,animaatio",
12      "title":"LAPSET"
13    }
14  ],
15  ...
16 ]

```

Listaus 7: Asiakassovelluksen listamallinen tietomalli

- **grid** – Valikko visualisoidaan ruudukkona, johon valikon alkiot ladotaan sarakkeittain täytettynä. Ruudukkoja voi olla erilaisia. Käytössä on ollut kaksirivinen, ja kolmirivinen ruudukko. Näissä navigointi pitää ohjelmoida siten, että nuolinäppäimillä siirryttäessä oikealle siirrytään tietorakenteessa niin monta alkiota eteenpäin, kuin ruudukolla on korkeutta, mikäli ruudukko ladotaan sarakkeittain vasemmasta yläalaidasta alkaen.
- **episodelist** – Valikko visualisoidaan pystylistana, jossa yhden sarjan jaksot ovat ladottu vasempaan reunaan navigoitavaksi listaksi ja ruudun oikeaan reunaan avautuu esikatselunäkymä kulloinkin aktiivisena olevan valikkoalkion mukaan. Listaa navigoidaan nuolinäppäimillä ylös- ja alaspäin, jolloin fokus vaihtuu jaksosta toiseen. Valikossa voidaan ilmaista jakson nimi, lähetysaika, saatavillaoloaika päivinä, sekä pieni kuva. Esikatselunäkymässä näytetään valittuun jaksoon liittyvä iso kuva, jakson nimi, kuvausteksti ja muut mahdolliset tiedot. *OK* tai *Enter* painike käynnistää videon.

Valikosta toiseen siirtyminen on myös johdonmukaista. Mikäli valitulla alkiolla on olemassa määritetty alivalikko, muuttuu se aktiiviseksi ja visualisoidaan ruudulle kun kyseinen alkio valitaan. Esimerkiksi tilausvideopalvelussa voi olla päävalikkona lista ohjelmatyyppejä, kuten Uutiset, Urheilu, Viihde, Elokuvat, Lapset. Jokaisesta kategoriasta voidaan aktivoida uusi valikko, joka onkin tyypiltään ruudukko ja siihen visualisoidaan kategorian ohjelmat kuvina. Kuitenkin tämäkin valikko on samanmuotoinen tietorakenne kuin mikä tahansa navigoitavissa oleva valikko. Yhdenmukaisuus helpottaa geneeristen toimintojen toteutusta ja ennen kaikkea navigointia sovelluksessa jonka tila on tallennettuna tässä samassa tietorakenteessa.

Kun sovellus sammutetaan, sen tila voidaan palauttaa ennalleen uudelleenkäynnistyksen jälkeen. Näin myös video-alkioiden tiedot säilyvät. Mikäli ohjelmaa on katsottu jonkin verran ja katsomista halutaan jatkaa myöhemmin, säilyy videon soittopositio tallessa. Tallennukseen käytetään HTML 5 -spesifisiä Storage-komponentteja LocalStorage ja SessionStorage. Storage-komponentin tuki on HbbTV-laitteilla suhteellisen laaja, tosin ei täydellinen. Mikäli laitteen HbbTV-selain ei tue Storage-komponenttia, laite ei kykene tallentamaan tilaansa. Tästä ei seuraa suurta ongelmaa, mutta valikot on ladattava kokonaan uudestaan aina sovelluksen käynnistyessä uudelleen. HbbTV 2.0 -versiossa edellytetään täydellistä HTML5-tukea

5.3.5 Käytettävyys ja käyttäjäkokemus

Käyttöliittymäsuunnittelussa on huomioitava erityisesti visuaalisuus, grafiikat, käytettävyys, sekä suorituskyky. Koska JavaScript on tulkittava kieli, ja TV-laitteiden prosessorit huomattavasti heikompia kuin tietokoneilla nykyään, voi suorituskykyongelmia ilmetä, mikäli käyttöliittymässä on paljon elementtejä ja animointia. Tämä tuottaa oman ongelmansa ja joskus on tingittävä muun muassa CSS-animaatioista, jos laitteet eivät pysty niitä suorittamaan sujuvasti. Myös eri laitteissa on havaittavissa huomattavia puutteita liittyen CSS-toiminnallisuuteen, jonka takia laitteita on hyvä eritellä heikompiin (*low-end*) ja parempiin (*high-end*). Heikommille laitteille voidaan toteuttaa räätälöityjä CSS-tiedostoja, jotta ne pystyisivät antamaan paremman käyttäjäkokemuksen käyttäjälleen. [23, s. 16-19]

5.3.6 Palvelinpään toteutus

Yksittäisen tilausvideopalvelun palvelinpuoleksi voidaan laskea kuvan 13 mukainen Proxy. Proxy on PHP:lla toteutettu välikappale, joka hakee tietoja joko Guide Managerista tai kolmannen osapuolen sisällönhallinta- ja jakopalvelun rajapinnasta. Proxyn tehtävä on tarjota kaikille asiakassovelluksille yhtenevä rajapinta ja toteuttaa varsinaisen lähdedatarajapinnan käyttö, sekä salata asiakassovellukselta todellinen lähde ja sen vaatimat tunnistautumismekanismit.

Proxyt ovat usein melko yksinkertaisia ja pieniä. Niiden toimintamalli noudattaa seuraavanlaista sekvenssiä:

1. Proxy lukee asiakassovelluksen lähettämät GET-parametrit ja muodostaa sen jälkeen rajapintakyselyn lähdedatarajapintaan, Guide Manageriin, tai joissain sovelluksissa Brightcove Video Cloud CDN -palveluun, joka on asiakasyrityksen palveluntuottaja. Tarvittavat yhdysavaimet määritellään Proxyn koodissa, eikä ole näkyvillä asiakassovellukselle.
2. Rajapintakyselyn tulos on JSON-muotoista. Yksinkertaisessa tapauksessa se palautetaan sellaisenaan asiakassovellukselle. Monesti proxy saattaa kuitenkin karsia pois, muokata, tai lisätä JSON-tietorakenteeseen kenttiä. Brightcoven rajapintakyselyn tulos on eri formaatissa, joten sieltä haettaessa Proxy muuntaa tuloksen yhdenmukaiseksi ja voi lisätä, muokata, sekä poistaa kenttiä tietorakenteesta.
3. Tulos palautetaan asiakassovellukselle yhdenmukaisessa JSON-muodossa. Http-vastauksen Content-Type -Header on oltava *application/json*.

Varsinainen palvelinrajapinnan tarjoama data tulee Guide Managerista (Kuva 13). Guide Manager on ohjelmoitu Javalla virtuaalikoneelle, jossa toimii Tomcat-palvelin. Guide Manager on toteutettu ohjelmaopastiedon hakuun ja jakeluun jo aiemmin, eikä sen sisäisiä ratkaisuita käsitellä tässä. Guide Managerin osalta käsitellään sen ulkoista rajapintaa Proxy-sovellukselle.

Guide Managerin rajapinta vaatii yhden pakollisen vaihtoehtoisen parametrin, joka on joko *serverid* tai *cid*. *Serverid* on yhden palveluntarjoajan palveluita yhdistävä yksikäsitteinen tunnistenumero. *Cid* puolestaan yksilöi kanavan. Guide Managerista voidaan näin hakea tietoja joko yhden tai useamman kanavan sisällöstä, tai jonkin useita kanavia tarjoavan palveluntarjoajan (TV-lähetisyhtiön) kaikista kanavista. Lisäksi valinnaisia parametreja ovat muun muassa *qf=query field* (hakukenttä), *q=query* (hakuehdot), *days* (montako päivää). Näiden avulla voidaan helposti hakea yhdenmukaisen rajapinnan avulla kaikissa asiakassovelluksen toteutuksissa erilaisia näkymiä. Esimerkiksi jos halutaan hakea uutisia ja säätietoja viikon ajalta neljältä kanavalta, hakuparametrit voisivat näyttää tältä:

```
cid=250,251,252,253&days=-7&qf=genre&q=uutiset,sää
```

Guide Managerin vastaus on aina JSON-muotoinen dokumentti josta käyttöliittymä muodostaa JavaScript-olion. Olio pitää sisällään aina *channels* attribuutin joka sisältää listan hakutuloksen kanavista. Jokainen kanava pitää sisällään alla listattuja tietoja ja EPG-listan. EPG-listassa on hakua vastanneiden videoiden metatiedot, joista tärkeimmät on listattu seuraavassa listauksessa 8.

Listauksessa 8, tietotyyppi string ISO 8601 tarkoittaa YYYY-MM-DDThh:mm:ssZ -muotoista aikaleimaa [35]. Käyttöliittymässä käydään tietorakenne läpi ja muodostetaan kohdassa määritellyn mukainen tietorakenne, jossa on kaikki sovelluksen tarvitsema informaatio. Tietorakenteeseen voidaan lisätä uusia attribuutteja ja muunnella olemassa olevia. Muodostettava tietorakenne on monimutkaisempi kuin rajapinnan palauttama listarakenne.

```

1  {
2    "channels": [
3      {
4        "id": <int>,
5        "serverid": <int>,
6        "title": <string>,
7        "streamurl": <string>,
8        "image": <string>,
9        "epg": [
10       {
11         "id": int,
12         "code": <string md5 hash>,
13         "start": <string ISO 8601>,
14         "end": <string ISO 8601>,
15         "duration": <int>,
16         "availablestart": <string ISO 8601>,
17         "availableend": <string ISO 8601>,
18         "title": <string>,
19         "desc": <string>,
20         "drm": <string>,
21         "genre": <string>,
22         "season": <string>,
23         "serie": <string>,
24         "mediaurl": <string>,
25         "mediaimage": <string>,
26       },
27       ...
28     ]
29   },
30   ...
31 ]
32 }

```

Listaus 8: ”EPG”-listauksen rakenne

5.3.7 Yhteensopivuusongelmat ja ratkaisut

HbbTV-laitteet noudattavat pääsääntöisesti HbbTV-standardia hyvin. Ongelmia ilmenee tilanteissa, joissa eri laitevalmistajien puolesta standardia ei ole tulkittu samoin, standardissa ei ole määritelty riittävän tarkasti jotain tulkinnanvaraista seikkaa, tai laitteiden selainkomponenteissa on HbbTV-standardista riippumattomia erilaisuuksia. Tunteamattomat erot ja yhteensopivuusongelmat aiheuttavat sen, että sovelluksia on testattava mahdollisimman monella laitteella, ennen kuin se voidaan julkaista. Myös jatkokehityksen, muutosten ja virheenkorjausten jälkeen uudet sovellusversiot on testattava huolellisesti mahdollisimman monella laitteella.

Yhteensopivuusongelmia pyritään ratkomaan ja virheitä korjaamaan laitevalmistajien puolesta. Tätä varten Institut für Rundfunktechnik (IRT) järjestää useita yhteensopivuustyöpajoja vuodessa. Tapahtumiin voivat osallistua laitevalmistajien edustajat, sekä HbbTV-sovelluksia kehittävien yritysten edustajat. Tapahtumaan ottavat osaa ilmoittautuneet laitevalmistajat, jotka tuovat paikalle uusimman laitteensa testausta varten. Laitteet ovat yleensä prototyypivaiheessa ja niiden testauksen myötä löydettyjä ongelmia voidaan korjata ennen uuden mallin julkaisua. Seuraavassa listassa on lueteltu IRT HbbTV Workshopissa ja muuten havaittuja ongelmia tuottavia asioita jotka eivät suoranaisesti liity tilausvideopalvelun toteutukseen, koska niitä ei siinä ole käytetty:

1. Kumpi siirtotie on prioriteetiltaan ensimmäinen Carousel, vai Web, ja miten laitteen tulee reagoida, mikäli prioriteetiltaan korkeampi siirtotie ei ole luettavissa tai saatavilla. Jotkin laitteet eivät lataa sovellusta objektikarusellista, mikäli Internet-yhteyttä ei voi muodostaa ja HTTP-URL on priorisoitu ensimmäiseksi siirtotieksi.
2. LG rekisteröi *Stream event listener* -käsittelijäfunktion kun rekisteröintikäskylle annetaan parametriksi Carousel component tag. LG:n televisiot eivät kuitenkaan havaitse Stream eventejä karusellissa.

Seuraavissa kappaleissa on käsitelty laajemmin työpajassa ja testauksessa ilmenneitä yhteensopivuusongelmia jotka ovat vaikuttaneet tilausvideopalvelun käyttöliittymän toteutukseen eniten. Muita vähäisempiä ongelmia esiintyy laitekohtaisten olettamusten myötä, kuten automaattinen kuvasuhteen vaihtaminen toistettavan videon mukaan, joka vaikuttaa myös sovelluksen käyttöliittymään. Tähän ei kuitenkaan voi vaikuttaa HbbTV-sovelluksessa.

Lähetyksen vaimennus. Tilausvideopalvelu on TV-lähetyksestä irrallinen sovellus siinä mielessä, että sieltä katsotaan videoita jotka eivät liity juuri sillä hetkellä televisiosta tulevaan ohjelmaan. Asiakkaan toiveesta jo sovelluksen käynnistyessä tulisi sen hetkinen TV-lähetys saada pois taustalta. TV-kuva voidaan piilottaa mutta äänenvoimakkuutta ei voi säätää HbbTV 1.1.1/1.2.1 -laitteissa.

Lähetyksen vaimentamiseen on olemassa ratkaisuja, joista yhteensopivuusongelmien takia yksikään ei toimi kaikilla laitteilla. Ensimmäinen vaihtoehto, mikä HbbTV-standardin mukaan on oikea tapa avata sovellus ilman lähetykskontekstia, on käynnistää sovellus *OIPFApplicationManager.createApplication()* -metodilla [23, s. 29]. Tämä ei kuitenkaan poista lähetykskontekstia jokaisella laitteella. Toinen, myöskin standardissa selkeästi määritelty tapa kytkeä lähetys pois ja siirtyä lähetyksiin riippumattomaan tilaan on *video/broadcast* -objektin *setChannel(null)* -metodi [36]. Tämä kuitenkin aiheuttaa joidenkin televisioiden kaatumisen. Kolmas edellistä lähellä oleva ratkaisu on yksiselitteinen *video/broadcast* -objektin metodi *stop()*, jonka tulisi siirtää lähetyksen tila tilaan ”*stopped*” jolloin kuva ja ääni poistuvat (Kuva 7)[23, s. 67-68]. Tämä ei myöskään vaikuta muiden lähetyksvirran resurssien saatavuuteen, kuten DSM-CC-karuselliin ja *Stream eventeihin*. Tämä saa kuitenkin jotkin vanhemmat televisiot sellaiseen tilaan, ettei ne enää palauta videokuvaa ja ääntä ollenkaan. Palautuminen on mahdollista vain ottamalla virtajohto kokonaan irti hetkeksi. Koska laitteet toimivat tässä asiassa niin eri lailla ja niissä voi aiheuta hyvin vakavia ongelmia, on tämä toiminnallisuus jätetty toteuttamatta ilman turvallista ratkaisua.

CSS-räätälöinti. CSS3 kuuluu kokonaisuudessaan vain HTML5 selaimiin ja se on osittain tuettu eri HbbTV-laitteiden kanssa. Koska osa laitteista tukee useampia CSS-selektoreita ja -efektejä, kuin toiset, on laitteet testattava erikseen miten hyvin mikäkin noudattaa CSS:ää. CSS-transition ei toimi kaikissa laitteissa ja joissakin se toimii huonosti. Se on kuitenkin visuaalisuudeltaan lisäarvoa tuova ominaisuus, joten sitä

käytetään kaikilla laitteilla jotka sen pystyvät visualisoimaan. CSS-attribuutti *Border-radius* ei toimi kaikilla laitteilla, ja joillakin Samsungin malleilla se aiheuttaa jumiutumista ja toisinaan jopa sovelluksen kaatumisen. *Nth-child* -selektori ei toimi Philipsin ja Toshiba AntGalio -laitteilla. Jos efekti ei toimi, tai se on hyvin hidas, voidaan tehdä määrätyille laitteille oma CSS-tiedosto. Jotta tiedostoja ei tarvitse tehdä liian montaa, ratkaisuna on kaksi tiedostoa, joista oletuksena oleva tiedosto sisältää kaikki CSS-ominaisuudet ja tunnetut mustalla listalla olevat laitteet lataavat ns.”low-end”-CSS-tiedoston, joka on karsittu CSS3-ominaisuuksista.

5.3.8 Muut toteutetut ja tulevat ratkaisut, sekä vaihtoehdot

Mainokset. Muilla kuin julkisilla TV- ja radioyhtiöillä on tällä hetkellä *pre-roll* -mainoksia, eli ennen varsinaista videon toiston aloitusta toistettavia mainoksia jotka on katsottava ennen varsinaisen videon katselua. Jatkossa mahdollista on myös, että halutaan *mid-roll* -mainoksia, joita voidaan ajoittaa katsottavaksi kesken ohjelman.

Geoblock. Videosisältöjen alueelliset rajoitukset on toteutettu osaan tilausvideopalveluita tarpeen mukaan. Jotkin sisällönjakelijärjestelmät huolehtivat siitä itse videosoittimen sijainnin perusteella, mutta toiset eivät. Tapauksissa, joissa geoblock on toteutettu HbbTV-sovellukseen, on hyödynnetty kahta tapaa paremman paikkatiedon selvitykseen. Ensimmäinen on ipinfo.io -palvelu, joka mahdollistaa 1000 rajapintakyselyä päivässä. Tämä ei kuitenkaan riitä, jos käyttäjiä on paljon. Siksi paikkatieto haetaan vain kerran jokaista laitetta kohti ja säilötään evästeeseen, jotta turhia apikutsuja ei tule. [37]

Mikäli ipinfo.io:n paikkatieto ei tunnista aluetta valtion tarkkuudella, se on ipinfo.io:n mukaan muu kuin Viro, tai ilmaisten API-kutsujen määrä on loppunut, tarkistetaan paikkatieto IP-osoitelistalta, jossa on kaikki viron IP-osoitteet aliverkkojen tarkkuudella.

5.3.9 HbbTV:n vaihtoehdot

Tilausvideopalveluita on toteutettu natiiveina sovelluksina eri alustoille, kuten TV-, Web- ja älypuhelinlustralat. Näistä lähinnä HbbTV-toteutusta on PC:lle toteutetut Web-sovellukset, kuten Netflix, Yle Areena ja MTV3 Katsomo. HbbTV-Toteutus eroaakin pääasiassa käyttöliittymäsuunnittelun ja käyttäjäinteraktion osalta niistä. Televisiolle on kuitenkin olemassa vain kolme vartenotettavaa vaihtoehtoa; laitteeseen valmiiksi asennettu sovellus, sovelluskaupasta ladattava sovellus, sekä TV-lähetyksen tarjoama HbbTV-sovellus.

6 ARVIOINTI

HbbTV on kehittyvä standardi ja tulossa yhä laajempaan käyttöön. Se on vielä uutta Suomessa, mutta tulevaisuudessa vielä laajemmassa käytössä. HbbTV -ohjelmointi on nopeaa ja joustavaa tulkittavan JavaScript-ohjelmoinnin myötä, sekä se tarjoaa uudenlaisen tavan toteuttaa interaktiivisia televisio-ohjelmia. Sovellukset ovat myös ketteriä koska ne voidaan syöttää loppukäyttäjille suoraan lähetyksen kautta ja se voidaan milloin vain päivittää, vaihtaa tai myöskin sulkea. Tämä helpottaa ohjelmakohtaisen sovelluksen toteutusta. Se voi olla saatavilla vain ohjelman ajan tai tietyllä hetkellä. Sovelluksia varten tarvitaan siis vähintään AIT-tauluun sovelluksen heräte-URL tai tiedostonimi DCM-CC -karusellissa, sekä palvelin, joka tarjoaa sovelluksen, sekä palvelin joka generoi AIT-taulun siirtovirtaan.

Maksukanavien ja maksullisten sisältöjen avaaminen onnistuu CI+ -Moduulin avulla tai käyttäjätunnus-salasana -kirjautumisella johonkin palveluun. Tilausvideopalvelu on hyvin potentiaalinen sovellus, joka on mahdollista toteuttaa HbbTV-tekniikoilla. Tärkeänä asiana HbbTV-sovelluksissa on käyttöliittymän suorituskyvyn, visualisoinnin ja käytettävyyden huomioiminen, koska kyseessä on kulutuselektronikkalaitteen sovellus, joka on useimmiten suunnattu viihdekäyttöön.

6.1 Tuotantoon siirtyneet palvelut

Työssä käsitellyistä sovelluksista osa on tuotannossa ja osa kehitysvaiheessa. Ensimmäisenä Sofia Digitalin tuottamana HbbTV-tilausvideopalveluna otettiin käyttöön Turkissa TRT:n (Turkin julkinen TV- ja Radioyhtiö) muiden HbbTV-Palveluiden kanssa syksyllä 2014. Kaikki HbbTV-palvelut kokonaisuudessaan ja palvelinten ohjelmistojen kanssa ovat Sofia Digitalin toimittamat. Palvelut ensimmäisessä vaiheessa olivat Launcher -päävalikko, Tilausvideopalvelu, Ohjelmaopas, Uutiset, Sää tiedot, sekä Pörssitiedot. Projekti on edelleen jatkuva ylläpitovaiheessa, sekä uusia sovelluksia kehitetään jatkuvasti. Palvelukokonaisuus on nimeltään TRT ARTI, (Engl. TRT Plus) ja se on tällä hetkellä saatavilla kaikilla TRT:n hallinnoimilla satelliittikanavilla yli maan rajojen. [38]

Alkuvuodesta 2015 Virossa Levira otti käyttöön vastaavanlaisen Sofia Digitalin toimittaman HbbTV-palvelukokonaisuuden. Siihen kuului Launcher, ERR-Tilausvideopalvelu, TV3-Tilausvideopalvelu, Kanal2-Tilausvideopalvelu, Ohjelmaopas, Uutiset, Sää, Internetradiot, sekä lisäksi Tasuta Info TV -kanavalla oleva Portal-sovellus, joka on näyttävämpi koko TV-kuvan peittävä Launcher-sovellus. Palvelukokonaisuus on nimeltään Tasuta TV Hübriid TV, ja se tavoittaa kaikki Leviran antenniverkon kuuluvuusalueella sijaitsevat vastaanottimet virossa. [39] [40]

Tällä hetkellä samalta pohjalta kuin jo julkaistut HbbTV-palvelut ja tilausvideosovellukset kehitteillä on kolme Suomeen tulevaa tilausvideopalvelua.

Ensimmäinen niistä julkaistaan näillä näkymin maanpäällisessä antenniverkossa kesäkuussa 2015 pilottina. Alkuvaiheessa palvelussa ei vielä ole maksullista sisältöä, mutta se tulee saataville myöhemmin. Muiden kehitteillä olevien palveluiden toteutuksella ei ole vielä määräaikaa, vaan ne ovat pilottivaiheessa kunnes asia etenee.

6.2 HbbTV:n soveltuvuus tilausvideopalvelun toteutukseen

HbbTV soveltuu hyvin tilausvideopalvelun toteutukseen, koska sen toteutus on HbbTV-standardin myötä saatavilla kaikille HbbTV-yhteensopiville älytelevisioille. Pääasiassa Catchup TV -videoita, mutta myös muita tilausvideoita on saatavilla useimmilla TV-operaattorilla Internetissä ja näille palveluille on olemassa käyttökelpoiset rajapinnat. Tämä on etuna uusien vastaavien palveluiden toteutukselle samaa rajapintaa hyödyntäen. Näin HbbTV-toteutus on usein hyvin pitkälti vain uusi käyttöliittymä, joka on suunniteltu televisiolle visuaalisesti sopivaksi, sekä kunnioittaen television erilaista käyttäjäinteraktiota verrattuna tietokoneeseen tai mobiililaitteeseen.

Televisiokanavariippumaton tilausvideopalvelu on lähtökohdiltaan erilainen siinä mielessä, että koska se ei ole kanavasidonnainen, se ei myöskään ole televisiolähetykseen liittyvä lisäpalvelu, mitä HbbTV-palvelut luonteeltaan ovat. Siksi televisiokanavayhtiöille toteutetut HbbTV-palvelut eivät kilpaile samoista lähtökohdista natiivien ja erikseen ladattavien sovellusten kanssa.

Eroja HbbTV- ja natiivisovellusratkaisuiden välillä löytyy näennäisesti suorituskyvystä natiivisovelluksen eduksi, koska se voi käyttää vapaammin järjestelmän tarjoamia rutiineita. Kuitenkin hyvä toteutus voi olla käytännössä yhtä nopea tai nopeampi HbbTV-sovelluksena, joten erolla ei ole suurta merkitystä HbbTV-tilausvideopalvelun kannalta. Suurempi ero on todennäköisemmin hinnassa, jakelun helppoudessa, sekä päivitettävyydessä ja monistettavuudessa, koska HbbTV-sovellus periaatteessa toimii missä tahansa HbbTV-yhteensopivassa laitteessa yhdellä toteutuksella. Näin ollen sovelluksista ei tarvitse tuottaa ja ylläpitää eri alustojen sovelluksia. Sovellusta ei myöskään tarvitse päivittää uuteen versioon erikseen, vaan se tapahtuu automaattisesti kun uusi versio julkaistaan sovelluksen lähdeosoitteeseen.

HbbTV sopii siis hyvin tilausvideopalveluiden toteuttamiseen TV-operaattoreille. Tällaisissa palveluissa voidaan jakaa sekä ilmaisia Catchup TV -videoita, sekä maksullisia tilausvideoita. Maksullisten sisältöjen suojaus vaatii DRM-tuen, sekä käyttäjän autentikoinnin, joka vaatii vielä yhteistyötä ja päätöksiä kunkin asiakkaan kesken erikseen. Seuraavassa vaiheessa kehitteillä olevissa palveluissa nämä vaatimukset tullaan täyttämään ja myös maksullista sisältöä tulee saataville eri HbbTV-palveluihin.

HbbTV mahdollistaa myös reaaliaikaisen videon toiston, joten palvelun osana voidaan toteuttaa reaaliaikaisia suoratoistopalveluita. Tällaiset voivat olla maksullisia tai maksuttomia lisäpalveluita tavallisen TV-lähetyksen rinnalle, tai sitten ihan uutta

materiaalia mitä televisiosta ei näe muuten, kuten IPTV-kanavia, urheilutapahtumia, sekä konsertteja.

7 YHTEENVETO

Perinteisen lineaarisen televisiolähetyksen videosisällön rinnalla on Internet-pohjaisia televisio- ja videopalveluita. PC:n lisäksi eri alustoille, kuten älytelevisioille on toteutettu tilausvideopalveluita natiivisovelluksina. Tässä työssä esiteltiin yleisesti HbbTV-standardi ja muut HbbTV-ohjelmointiin vahvasti liittyvät standardit. Toteutetuista HbbTV-tilausvideopalveluista esiteltiin yleiset ja toiminnalliset vaatimukset, sekä tekniset ratkaisut, miten niihin on vastattu.

Työssä esiteltiin, minkälainen on TV sovellusympäristönä, millaisia sovelluksia TV:lle voidaan ohjelmoida ja minkälaisia erilaisia tekniikoita niiden toteuttamiseen on olemassa. TV-ohjelmointi ei eroa PC- ja mobiililaitteohjelmasta välttämättä toteutusteknisesti, mikäli käytössä on HTML5 -perustaiset Web-tekniologiat. TV-sovellusympäristö eroaa PC-ympäristöstä ja mobiiliympäristöstä eniten käyttöliittymäsuunnittelussa, koska ruudun resoluutio on TV:ssä vakio, katseluetäisyys on pidempi ja käyttäjäinteraktio tapahtuu kaukosäätimellä, jolla ei ole hiiri- tai kosketusvalintaa.

Ongelmana oli kehittää tilausvideopalvelu HbbTV-tekniikoilla noudattaen HbbTV-standardia ja muita siihen liittyviä standardeja ja hyödyntää HbbTV:n ja videontoiston mahdollistavia tekniikoita. Toteutuksen haasteena oli vastata tilausvideopalvelun toiminnallisiin vaatimuksiin.

HbbTV-tilausvideopalveluita tehtiin useampaan projektiin. Jokaisessa sovelluksessa käytettiin samanlaista geneeristä tietomallia asiakassovelluksen puolella ja samanlaista toiminnallista määrittelyä sisältäen joitakin variaatioita. Tietomalli on valikkorakenne, jossa navigaatio ja visuaalinen mallintaminen on toteutettu valikkotyypin mukaan parametrisoidusti. Kaikki toteutukset hyödynsivät samaa palvelinrajapintaa riippumatta siitä, mikä oli lähdedatana käytettyä lopullinen rajapinta. Geneerisen mallin ansiosta sovellukset ovat helposti ylläpidettäviä ja monistettavia. Asiakassovellukseen voidaan vaihtaa toisen palvelun sisältö vaihtamalla palvelinrajapinnan kutsuihin annettavia parametreja.

Lopputuloksena tämän työn kirjoituksen ajalla saatiin tuotantovalmiiksi kaksi eri asiakkaan projektia, jotka pitivät sisällään yhteensä neljä HbbTV-tilausvideopalvelua. HbbTV sopii hyvin TV-operaattorien maksullisen ja maksuttoman sisällön tarjoamiseen kanavakohtaisen TV-lähetyksen mukana signaloitavan lisäpalvelun, HbbTV-tilausvideopalvelun, avulla. Toteutuksessa ei ole toteutettu käyttäjän autentikointia eikä DRM-suojatun sisällön toistamista, mutta sovellusten kehitys jatkuu ja nämä tulevat myöhemmin osaksi uusia ja olemassa olevia tilausvideopalveluita.

LÄHTEET

- [1] ArviD-Julkaisu 01/2004: ”Digi-TV:n palveluntekijän opas”, Julkaistu 22.4.2004, Viitattu 14.9.2014, ISBN 952-201-000-6
- [2] Yle, artikkeli: Äly-TV sorsii suomalaista sisältöä. Julkaistu 20.6.2012, Viitattu 9.5.2015. Saatavissa: <http://yle.fi/aihe/artikkeli/2012/06/20/aly-tv-sorsii-suomalaista-sisaltoa>
- [3] John Moduling: ”Apps fragmentation is starting to hurt Connected TV”, Julkaistu 18.6.2012, Viitattu 9.5.2015, Saatavissa: <http://www.v-net.tv/apps-fragmentation-is-starting-to-hurt-connected-tv/>
- [4] Opera TV store apps, Viitattu 11.5.2015, Saatavissa: <https://dev.opera.com/tv/>
- [5] Yle, blogi, Kari Haakana : ”Näillä laitteilla käytät Areenaa, syksy 2014 ” 11/2014, Julkaistu 5.11.2014, Viitattu 9.5.2015, Saatavissa: <http://blogit.yle.fi/yle-areena/nailla-laitteilla-kaytat-areenaa-syksy-2014>
- [6] IPTV -operaattorit Suomessa, Viitattu 9.5.2015, Saatavissa: <http://www.kaikkitelevisiosta.fi/verkot/iptv/>
- [7] HbbTV roadmap 2012, Viitattu 10.9.2014 saatavissa: http://www.hbbtv.org/pages/news_events/docs/hbbtv_roadmap_121217.pdf
- [8] HbbTV roadmap 2014, Viitattu 22.4.2015 saatavissa: <http://www.hbbtv-symposiumeurope2014.com/wp-content/uploads/2014/10/2-1-HbbTV-Roadmap.pdf>
- [9] Digita, tiedote 5.3.2013, Viitattu 9.5.2015, Saatavissa: http://www.digita.fi/yhtio/media/tiedotteet/2013/tv-kanavien_runsaat_nettilisallot_nyt_helposti_siirrettavissa_televisioon.2121.news
- [10] AntennaReady HD, Viitattu 9.5.2014, Saatavilla: http://www.testatutlaitteet.fi/fi/fi_1.html

- [11] Viestintävirasto, ”Must carry eli siirtovelvoite” Viitattu 4.5.2015, Saatavissa: <https://www.viestintavirasto.fi/tvradio/jakelujavastaanotto/Kaapeli-tv/mustcarryelisiirtovelvoite.html>
- [12] Laki 7.11.2014/917 Tietoyhteiskuntakaari, Luku 27, §227, viitattu 4.5.2015, saatavissa: <http://www.finlex.fi/fi/laki/ajantasa/2014/20140917>
- [13] Digita, ”Hybridi-TV:n käyttöönotto-ohjeita”, Viitattu 24.5.2015. Saatavilla: http://digita.fi/kuluttajat/tv/hybridi-tv/hybridi-tv_n_kayttoonotto-ohjeita
- [14] OIPF DAE, Viitattu 10.5.2015, Saatavissa: http://www.oipf.tv/docs/oipf-archive/OIPF-T1-R2-Specification-Volume-5-Declarative-Application-Environment-v2_1-2011-06-21.pdf
- [15] ETSI TS 102 796, HbbTV 1.1.1, Viitattu 24.4.2015, Saatavilla: http://www.etsi.org/deliver/etsi_ts/102700_102799/102796/01.01.01_60/ts_102796v010101p.pdf
- [16] DASH – Design Principles and Standards, Thomas Stockhammer: ”Dynamic Adaptive Streaming over HTTP – Design Principles and Standards”, Viitattu 11.5.2015, Saatavissa: http://www.w3.org/2010/11/web-and-tv/papers/webtv2_submission_64.pdf
- [17] Liikenne- ja viestintäministeriön julkaisuja - Kansallinen laajakaistastrategia, s-46, 3/2007, Viitattu 15.9.2014, Saatavissa: http://www.lvm.fi/fileserver/Julkaisuja%203_2007.pdf
- [18], Steven Morris: ”TV Without Borders - How To Become An Expert In DSM-CC”, Viitattu 4.5.2015, Saatavissa : http://www.interactivetvweb.org/tutorials/dtv_intro/dsmcc
- [19] ETSI TS 102 034, Transport of MPEG- 2 TS Based DVB Services over IP Based Networks, Versio 1.4.1, Julkaistu 8/2009, Viitattu 14.4.2015 Saatavilla: http://www.etsi.org/deliver/etsi_ts/102000_102099/102034/01.04.01_60/ts_102034v010401p.pdf
- [20] ETSI TS 102 539, DVB v.1.3.1, Julkaistu 4/2010, Viitattu 11.5.2015, Saatavilla : http://www.etsi.org/deliver/etsi_ts/102500_102599/102539/01.03.01_60/ts_102539v010301p.pdf

- [21] HbbTV.org, About HbbTV -Specification, Viitattu 25.7.2014, Saatavissa: https://www.hbbtv.org/pages/about_hbbtv/specification.php
- [22] Performance comparison HEVC-ACV, Viitattu 2.2.2015. Saatavissa: http://iphome.hhi.de/marpe/download/Performance_HEVC_VP9_X264_PCS_2013_preprint.pdf
- [23] ETSI TS 102 796, HbbTV 1.2.1, julkaistu 11/2012, Viitattu 25.7.2014, Saatavissa: http://www.etsi.org/deliver/etsi_ts/102700_102799/102796/01.02.01_60/ts_102796v010201p.pdf
- [24] WC3 DOM. Le Hégarret, Philippe (2002), World Wide Web Consortium. Viitattu 16.5.2015, Saatavissa: <http://www.w3.org/2002/07/26-dom-article.html>
- [25] FireHbbTV, "Usage / Features Detils" Viitattu 10.9.2014 saatavissa: <http://firehbbtv.iptv.aw.atos.net/#features>
- [26] Jsconsole, Viitattu 21.4.2015, Saatavissa: <http://jsconsole.com/remote-debugging.html>
- [27] ISO/IEC 23009-1:2014, Viitattu 5.5.2015, Saatavissa: http://standards.iso.org/ittf/PubliclyAvailableStandards/c065274_ISO_IEC_23009-1_2014.zip
- [28] Streaming Media: "What is MPEG DASH?", Viitattu 4.5.2015, Saatavissa: <http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-MPEG-DASH-79041.aspx>
- [29] MP4Box, Viitattu 22.4.2015. Saatavissa: <http://gpac.wp.mines-telecom.fr/mp4box/dash/>
- [30] MPEG DASH content protection, Viitattu 24.4.2015, Saatavissa: <http://dashif.org/identifiers/protection/>
- [31] WhatIsMyIPAddress.com "How accurate is IP GeoLocation?" Viitattu 5.5.2015, Saatavissa: <http://whatismyipaddress.com/geolocation-accuracy>
- [32] Azure, Viitattu 16.5.2015, Saatavissa: <http://azure.microsoft.com/fi-fi/>

- [33] Brightcove Videocloud, Viitattu 16.5.2015, Saatavissa: <https://www.brightcove.com/en/online-video-platform>
- [34] Akamai, Viitattu 16.5.2015, Saatavissa: <http://www.akamai.com/>
- [35] W3C "Date and Time formats", Viitattu 4.5.2015, Saatavissa <http://www.w3.org/TR/NOTE-datetime>
- [36] Errata 1 for the HbbTV Version 1.0, s.42. Viitattu 15.5.2015, Saatavissa: https://hbbtv.org/pages/about_hbbtv/HbbTV-errata-1.pdf
- [37] Ipinfo.io Api, Viitattu 24.4.2014. Saatavissa: <https://ipinfo.io/developer>
- [38] Broadband TV News, 8.4.2015, Viitattu 24.4.2015, Saatavilla : <http://www.broadbandtvnews.com/2015/04/08/trt-launches-hybrid-services/>
- [39] Levira, HbbTV-palvelut, Viitattu 24.4.2015, Saatavissa: <http://levira.ee/en/news/levira-launches-hybrid-tv-services-in-estonia-using-sofia-digital-hbbtv-platform/>
- [40] Advanced Television, Levira HbbTV platform Viitattu 24.4.2015, Saatavissa: <http://advanced-television.com/2015/03/13/levira-hybrid-tv-in-estonia-with-sofia/>

LIITE 1: ESIMERKKI: MPEG DASH DRM

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <!-- Created with Unified Streaming Platform(version=1.7.6) -->
3 <MPD
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xmlns="urn:mpeg:dash:schema:mpd:2011"
6   xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011
7     http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-
8     DASH_schema_files/DASH-MPD.xsd"
9   xmlns:cenc="urn:mpeg:cenc:2013"
10  xmlns:mas="urn:marlin:mas:1-0:services:schemas:mpd"
11  type="dynamic"
12  availabilityStartTime="1970-01-01T00:00:00Z"
13  publishTime="2015-05-05T06:03:01.271843Z"
14  minimumUpdatePeriod="PT2S"
15  timeShiftBufferDepth="PT15M"
16  maxSegmentDuration="PT5S"
17  minBufferTime="PT10S"
18  profiles="urn:mpeg:dash:profile:isoff-live:2011">
19  <Period
20    id="1"
21    start="PT0S">
22    <BaseURL>dash/</BaseURL>
23    <AdaptationSet
24      group="1"
25      contentType="audio"
26      minBandwidth="96000"
27      maxBandwidth="96000"
28      segmentAlignment="true"
29      audioSamplingRate="24000"
30      mimeType="audio/mp4"
31      codecs="mp4a.40.5">
32      <AudioChannelConfiguration
33        schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011"
34        value="2">
35      </AudioChannelConfiguration>
36      <!-- Common Encryption -->
37      <ContentProtection
38        schemeIdUri="urn:mpeg:dash:mp4protection:2011"
39        value="cenc"
40        cenc:default_KID="5CF725ED-7C2A-4ACB-99D9-028564608B37">
41      </ContentProtection>
42      <!-- PlayReady -->
43      <ContentProtection
44        schemeIdUri="urn:uuid:9A04F079-9840-4286-AB92-E65BE0885F95"
45        value="MSPR 2.0">
46      </ContentProtection>
47      <!-- widevine -->
48      <ContentProtection
49        schemeIdUri="urn:uuid:EDEF8BA9-79D6-4ACE-A3C8-27DCD51D21ED">
50      </ContentProtection>
51      <!-- Marlin -->
52      <ContentProtection
53        schemeIdUri="urn:uuid:5E629AF5-38DA-4063-8977-97FFBD9902D4">
54      </ContentProtection>
55      <mas:MarlinContentIds>
56      <mas:MarlinContentId>urn:marlin:kid:5cf725ed7c2a4acb99d9028564608b37</
57      mas:MarlinContentId>
58    </mas:MarlinContentIds>
59    </SegmentTemplate
60      timescale="24000"
61      initialization="DR47USP-$RepresentationID$.dash"

```

```

59     media="DR47USP-$RepresentationID$-$Time$.dash"
60     startNumber="1">
61     <SegmentTimeline>
62         <S t="41040766976" d="96256" r="223" />
63     </SegmentTimeline>
64 </SegmentTemplate>
65 <Representation
66     id="audio=96000"
67     bandwidth="96000">
68 </Representation>
69 </AdaptationSet>
70 <AdaptationSet
71     group="2"
72     contentType="video"
73     par="16:9"
74     minBandwidth="500000"
75     maxBandwidth="2500000"
76     minwidth="512"
77     maxwidth="1280"
78     minHeight="288"
79     maxHeight="720"
80     segmentAlignment="true"
81     frameRate="25"
82     mimeType="video/mp4"
83     codecs="avc1.4D401F"
84     startWithSAP="1">
85     <!-- Common Encryption -->
86     <ContentProtection
87         schemeIdUri="urn:mpeg:dash:mp4protection:2011"
88         value="cenc"
89         cenc:default_KID="5CF725ED-7C2A-4ACB-99D9-028564608B37">
90     </ContentProtection>
91     <!-- PlayReady -->
92     <ContentProtection
93         schemeIdUri="urn:uuid:9A04F079-9840-4286-AB92-E65BE0885F95"
94         value="MSPR 2.0">
95     </ContentProtection>
96     <!-- Widevine -->
97     <ContentProtection
98         schemeIdUri="urn:uuid:EDEF8BA9-79D6-4ACE-A3C8-27DCD51D21ED">
99     </ContentProtection>
100    <!-- Marlin -->
101    <ContentProtection
102        schemeIdUri="urn:uuid:5E629AF5-38DA-4063-8977-97FFBD9902D4">
103 <mas:MarlinContentIds>
104
105 <mas:MarlinContentId>urn:marlin:kid:5cf725ed7c2a4acb99d9028564608b37</
mas:MarlinContentId>
106 </mas:MarlinContentIds>
107 </ContentProtection>
108 <SegmentTemplate
109     timescale="25"
110     initialization="DR47USP-$RepresentationID$.dash"
111     media="DR47USP-$RepresentationID$-$Time$.dash"
112     startNumber="1">
113     <SegmentTimeline>
114         <S t="42750700" d="100" r="225" />
115     </SegmentTimeline>
116 </SegmentTemplate>
117 <Representation
118     id="video=500000"
119     bandwidth="500000"
120     width="512"
121     height="288"
122     scanType="progressive">
123 </Representation>
124 <Representation
125     id="video=750000"
126     bandwidth="750000"
127     width="640"
128     height="360"
129     scanType="progressive">
130 </Representation>

```

```
131         id="video=1000000"  
132         bandwidth="1000000"  
133         width="768"  
134         height="432"  
135         scanType="progressive">  
136     </Representation>  
137     <Representation  
138         id="video=1500000"  
139         bandwidth="1500000"  
140         width="864"  
141         height="486"  
142         scanType="progressive">  
143     </Representation>  
144     <Representation  
145         id="video=2500000"  
146         bandwidth="2500000"  
147         width="1280"  
148         height="720"  
149         scanType="progressive">  
150     </Representation>  
151 </AdaptationSet>  
152 </Period>  
153 </MPD>
```

LIITE 2: PLAYREADY -LISENSSIN HAKU

Esimerkistä on poistettu luottamukselliset tiedot ... merkeillä korvaten. Alla HTTP-POST kutsu ja vastaus otsikkoineen sisältöineen.

```

POST http://.../license?timeStamp=2013-09-
02T16:10:37+0300&contract=...&account=source HTTP/1.1
Accept: */*
Accept-Language: fi-FI
Content-Type: text/xml; charset=utf-8
SOAPAction:
"http://schemas.microsoft.com/DRM/2007/03/protocols/AcquireLicense"
msprdrm_server_exception_compat: false
msprdrm_server_redirect_compat: false
UA-CPU: ARM
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; windows Phone 8.0;
Trident/6.0; IEMobile/10.0; ARM; Touch; NOKIA; Lumia 920)
Host: ...
Content-Length: 7360
Connection: Keep-Alive
Pragma: no-cache
Cookie: dtCookie=<cookies>

1 <?xml version="1.0" encoding="UTF-8"?>
2 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3   <soap:Body>
4     <AcquireLicense
5       xmlns="http://schemas.microsoft.com/DRM/2007/03/protocols">
6       <challenge>
7         <Challenge
8           xmlns="http://schemas.microsoft.com/DRM/2007/03/protocols/messages">
9           <LA
            xmlns="http://schemas.microsoft.com/DRM/2007/03/protocols"
            Id="SignedData"
            xml:space="preserve"><Version>1</Version><ContentHeader><WRMHEADER
            xmlns="http://schemas.microsoft.com/DRM/2007/03/PlayReadyHeader"
            version="4.0.0.0"><DATA><PROTECTINFO><KEYLEN>16</KEYLEN><ALGID>AESCTR<
            /ALGID></PROTECTINFO><KID>eCyu+509ykwZZi1tj/
            +v3A==</KID><CHECKSUM>SQujzY5APYc=</CHECKSUM><CUSTOMATTRIBUTES><IIS_DR
            M_VERSION>7.1.1064.0</IIS_DRM_VERSION></CUSTOMATTRIBUTES><LA_URL>...</
            LA_URL></DATA></WRMHEADER></ContentHeader><CLIENTINFO><CLIENTVERSION>2
            .4.0.1708</CLIENTVERSION></CLIENTINFO><RevocationLists><RevListInfo><L
            istID>ioydTlK2p0WxkwlprR5Hw==</ListID><Version>10</Version></RevListI
            nfo><RevListInfo><ListID>...</ListID><Version>11</Version></RevListInf
            o><RevListInfo><ListID>...</ListID><Version>11</Version></RevListInfo>
            <RevListInfo><ListID>...</ListID><Version>13</Version></RevListInfo></
            RevocationLists><LicenseNonce>...</LicenseNonce><EncryptedData
            xmlns="http://www.w3.org/2001/04/xmlenc#"
            Type="http://www.w3.org/2001/04/xmlenc#Element"><EncryptionMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" /><KeyInfo
            xmlns="http://www.w3.org/2000/09/xmldsig#"><EncryptedKey
            xmlns="http://www.w3.org/2001/04/xmlenc#"><EncryptionMethod
            Algorithm="http://schemas.microsoft.com/DRM/2007/03/protocols#ecc256"
            /><KeyInfo
            xmlns="http://www.w3.org/2000/09/xmldsig#"><KeyName>WRRMServer</KeyNam
            e></KeyInfo><CipherData><CipherValue>...</CipherValue></CipherData></E
            ncrryptedData></LA>
            <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
            <SignedInfo>

```

```

10         <CanonicalizationMethod
11     Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
12         <SignatureMethod
13     Algorithm="http://schemas.microsoft.com/DRM/2007/03/protocols#ecdsa-
14     sha256" />
15         <Reference URI="#SignedData">
16             <DigestMethod
17     Algorithm="http://schemas.microsoft.com/DRM/2007/03/protocols#sha256"
18     />
19                 <DigestValue>...</DigestValue>
20             </Reference>
21         </SignedInfo>
22         <SignatureValue>...</SignatureValue>
23         <KeyInfo>
24             <KeyValue>
25                 <ECCKeyValue>
26                     <PublicKey>...</PublicKey>
27                 </ECCKeyValue>
28             </KeyValue>
29         </KeyInfo>
30     </Signature>
31 </Challenge>
32 </challenge>
33 </AcquireLicense>
34 </soap:Body>
35 </soap:Envelope>

```

```

HTTP/1.1 200 OK
Date: Mon, 02 Sep 2013 13:10:38 GMT
Server: Apache
X-Powered-By: Servlet 2.4; JBoss-4.2.3.GA (build: SVNTag=JBoss_4_2_3_GA
date=200807181417)/JBossWeb-2.0
Set-Cookie: sumoSession=deleteMe; Path=/; Max-Age=0; Expires=Sun, 01-Sep-
2013 13:10:38 GMT
Set-Cookie: sumoSession=deleteMe; Domain=...; Path=/; Max-Age=0;
Expires=Sun, 01-Sep-2013 13:10:38 GMT
Content-Encoding: utf-8
Content-Length: 1090
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/xml; charset=UTF-8

```

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><AcquireLicens
5     eResponse
6     xmlns="http://schemas.microsoft.com/DRM/2007/03/protocols"><AcquireLic
7     enseResult><Response
8     xmlns="http://schemas.microsoft.com/DRM/2007/03/protocols/messages"><L
9     icenseResponse
10    xmlns="http://schemas.microsoft.com/DRM/2007/03/protocols"><Version>1<
11    /Version><Licenses><License>WE1SAA...20vCTIh/gOu/Q==</License></Licens
12    es></LicenseResponse></Response></AcquireLicenseResult></AcquireLicens
13    eResponse></soap:Body></soap:Envelope>

```