



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Милан Новаковић

Имплементација слике-у-слици на Андроид ТВ платформи

ДИПЛОМСКИ РАД
- Основне академске студије -

Нови Сад, 2017



КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:			
Идентификациони број, ИБР:			
Тип документације, ТД:	Монографска документација		
Тип записа, ТЗ:	Текстуални штампани материјал		
Врста рада, ВР:	Завршни (Bachelor) рад		
Аутор, АУ:	Милан Новаковић		
Ментор, МН:	проф. др Илија Башичевић		
Наслов рада, НР:	Имплементација слике-у-слици на Андроид ТВ платформи		
Језик публикације, ЈП:	Српски / латиница		
Језик извода, ЈИ:	Српски		
Земља публиковања, ЗП:	Република Србија		
Уже географско подручје, УГП:	Војводина		
Година, ГО:	2017		
Издавач, ИЗ:	Ауторски репринт		
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, ФО: (поглавља/страна/ цитата/табела/слика/графика/прилога)	7/44/0/1/17/0/0		
Научна област, НО:	Електротехника и рачунарство		
Научна дисциплина, НД:	Рачунарска техника		
Предметна одредница/Кључне речи, ПО:	Дигитална телевизија, Андроид, ДТВ сервиси		
УДК			
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН:			
Извод, ИЗ:	У овом раду приказано је једно решење за проширење слика-у-слици режима рада у оквиру апликације и улазног ТВ сервиса на пријемнику дигиталног телевизијског сигнала заснованог на Андроид платформи. Циљ овог рада је реализација програмске подршке која омогућује истовремено приказивање два ТВ канала, као и могућност коришћења слике-у-слици који је прописан од сране Googlea. У раду су детаљно описаны сви модули који су изменjeni или направљени зарад слика-у-слици функционалности, почев од највиших слојева програмске подршке закључно са најнижим слојевима.		
Датум прихватања теме, ДП:			
Датум одбране, ДО:			
Чланови комисије, КО:	Председник:	проф. др Мирослав Поповић	
	Члан:	доц. др Иван Каштелан	Потпис ментора
	Члан, ментор:	проф. др Илија Башичевић	



KEY WORDS DOCUMENTATION

Accession number, ANO:			
Identification number, INO:			
Document type, DT:	Monographic publication		
Type of record, TR:	Textual printed material		
Contents code, CC:	Bachelor Thesis		
Author, AU:	Milan Novakovic		
Mentor, MN:	PhD Ilija Basicevic		
Title, TI:	A realization of picture-in-picture feature for Android TV platform		
Language of text, LT:	Serbian		
Language of abstract, LA:	Serbian		
Country of publication, CP:	Republic of Serbia		
Locality of publication, LP:	Vojvodina		
Publication year, PY:	2017		
Publisher, PB:	Author's reprint		
Publication place, PP:	Novi Sad, Dositeja Obradovica sq. 6		
Physical description, PD: (chapters/pages/ref./tables/pictures/graphs/appendices)	7/44/0/1/17/0/0		
Scientific field, SF:	Electrical Engineering		
Scientific discipline, SD:	Computer Engineering, Engineering of Computer Based Systems		
Subject/Key words, S/KW:	Digital television, Android, DTV service		
UC			
Holding data, HD:	The Library of Faculty of Technical Sciences, Novi Sad, Serbia		
Note, N:			
Abstract, AB:	This paper presents an extension of Android TV application to support picture-in-picture feature. The goal is to create software that allows displaying two different TV channels in the same time, as well as possibility of using picture-in-picture, according to requirements from Google. This paper contains detailed description of all the modules that are changed or made for picture-in-picture functionality, from the top layer to the bottom layer of software.		
Accepted by the Scientific Board on, ASB:			
Defended on, DE:			
Defended Board, DB:	President:	PhD Miroslav Popović	
	Member:	PhD Ivan Kaštelan	Menthor's sign
Member, Mentor:	PhD Ilija Bašičević		

Zahvalnost

Zahvalujem se mentoru prof. dr Iliji Bašičeviću na stručnoj pomoći tokom izrade ovog rada.

Posebno se zahvalujem Sretenu Tanackoviću i Darku Dejanoviću na stručnoj pomoći, savetima, utrošenom vremenu i strpljenju tokom izrade rada.

Zahvalujem se svojim prijateljima i kolegama na podršci i pomoći tokom školovanja.

Na kraju, želim da se zahvalim svojim roditeljima i bratu, devojci, i ostalim članovima porodice na brizi, motivaciji i ogromnoj podršci.

SADRŽAJ

1.	Uvod.....	1
2.	Teorijske osnove.....	3
2.1	Digitalna televizija.....	3
2.2	Android.....	5
2.2.1	Android Platforma.....	5
2.2.2	Android Open Source Project.....	5
2.2.3	Google Mobile Service.....	6
2.2.4	Android programska podrška za TV uređaje – TIF.....	7
2.2.4.1	Komponente Android programske podrške za TV uređaje.....	7
2.2.4.2	TV snabdevač.....	8
2.2.4.3	TV ulazni rukovaoc.....	9
2.2.4.4	TV ulaz.....	10
2.2.4.5	TV aplikacija.....	11
2.3	Slika-u-slici.....	12
2.3.1	Upotreba i istorijat klasičnog slika-u-slici proširenja.....	12
2.3.2	Android TV slika-u-slici.....	12
3.	Koncept rešenja.....	14
3.1	Opis ciljne platforme.....	14
3.2	Programska podrška STB uređaja.....	15
3.3	Proširenje programske podrške za PIP funkcionalnost.....	18
3.3.1	Android Nougat PIP.....	18
3.3.2	Klasičan PiP.....	18
4.	Programsko rešenje.....	20

4.1	Realizacija Android PiP podrške u okviru TV korisničke aplikacije.....	21
4.2	Realizacija PiP podrške u okviru ulaznog TV servisa.....	22
4.2.1	Puštanje dva TV programa uporedo.....	22
4.2.2	Mogućnost zamene glavnog i PiP prozora.....	24
4.2.3	Mogućnost uklanjanja i pojavljivanja PiP-a.....	26
5.	Ispitivanje i verifikacija.....	28
6.	Zaključak.....	34
7.	Literatura.....	35

SPISAK SLIKA

Slika 2.1 Standardi digitalne televizije u svetu.....	4
Slika 2.2 TIF Arhitektura.....	8
Slika 2.4 TV snabdevač.....	9
Slika 2.5 TV ulaz treće strane.....	10
Slika 3.1 Broadcom 7252S set-top box.....	15
Slika 3.2 Struktura programske podrške TV prijemnika.....	16
Slika 4.1 Prikaz menija u kome se nalazi opcija za PiP režim rada.....	21
Slika 4.2 Promena kanala i zamena glavnog i PiP prozora.....	26
Slika 5.1 TV aplikacija sa upaljenim svim GUI komponentama i klasičnim PiPom.....	29
Slika 5.2 PiP režim rada nakon uklanjanja grafičkih komponenti i klasičnog PiPa.....	29
Slika 5.3 Povratak u normalan režim rada.....	30
Slika 5.4 Izgled ekrana nakon startovanja aplikacije.....	31
Slika 5.5 Izgled ekrana nakon urađene zamene.....	31
Slika 5.6 Promena kanala nakon prethodne zamene ekrana.....	32
Slika 5.7 Uklanjanje prozora za klasičan PiP.....	32
Slika 5.8 Prelazak na zeljeni kanal i dalje bez klasičnog PiPa.....	33
Slika 5.9 Ekran nakon što je klasičan PiP ponovo aktiviran.....	33

SPISAK TABELA

Tabela 5.1 Ispitivanje funkcionalnosti i rezultati.....28

SKRAĆENICE

TV	- Television, televizija
DTV	- <i>Digital Television</i> , Digitalna televizija
DVB	- Digital Video Broadcasting, Standard za digitalnu video difuziju
PIP	- Picture in picture, Slika-u-slici u okviru televizora
STB	- Set-top Box
TIF	- The Android TV Input Framework, razvojno okruženje za Android TV

1. Uvod

U ovom radu biće prikazano rešenje projektovanja programske podrške za puštanje dva televizijska programa paralelno u prozorima različitih veličina, kao i mogucnost korišćenja aplikacija dok se trenutno pušteni program emituje u smanjenom prozoru za Android bazirane televizijske prijemnike. Ova funkcionalnost se naziva Slika-u-slici (eng. Picture-in-picture,u daljem tekstu PIP). Rešenje je realizovano na Broadcom set-top box uređaju.

Razvoj infrastrukture za prenos digitalnog televizijskog signala, omogućio je i razvoj tehnologija za prikaz informacija koje stižu krajnjem korisniku. Kako su set-top boks uređaji napredovali od prvobitnog pojavljivanja, širio im se skup funkcionalnosti. Jedna od njih je i gore pomenuti PIP koja pretvara televizor, statični uređaj za reprodukciju video i audio sadržaja, multimedijalni uređaj na kom se izuzev mogućnosti za reprodukciju jednog ili paralelno dva televizijska programa, omogućuje gledanje TV sadržaja paralelno sa izvršavanjem različitih aktivnosti, poput dobavljanja informacija o programu koji se trenutno emituje, pretraživanje interneta i slično.

Poslednja verzija Androida omogućava brz izlazak novog TV rešenja na tržište sa svim funkcionalnostima koje se zahtevaju od novih STB uređaja i TV prijemnika. Jedna od njegovih prednosti je što se aplikacije pišu u programskom jeziku Java[1][2], pa je rad sa memorijom olakšan po pitanju zauzimanja i oslobođanja iste. Ovakva rasprostranjenost proizilazi iz činjenice da je otvorenog koda (engl. open-source)[3]. To omogućava proizvođačima uređaja da ga menjaju i prilagođavaju svojim potrebama. Uticaj Android platforme se širio, pa danas postoji značajan broj aplikacija pravljenih za ovu platformu, a ona se nalazi na raznim uređajima: mobilnim telefonima, tabletima, kućnim aparatima, televizijskim uređajima i slično.

Zbog pogodnosti koje nudi Android platforma, cilj ovog rada je implementacija za Sliku-slici režim rada koji je od Nugat (eng. Nougat) verzije Androida uveden kao proširenje na Android TV platformama i implementacija proširenja za gledanje dva programa uporedo u TV aplikaciji.

Osnovne prepreke prilikom izrade ovog rada su bile u omogućavanju istovremenog puštanja 2 TV programa, mogućnosti njihove zamene, kao i promena dimenzije videa glavnog programa prilikom poziva PiP-a kao proširenja na Android TV platformi. Takođe, jedan od glavnih izazova je spojiti tradicionalni pristup gledanja televizije sa svim novostima koje Android platforma uvodi.

Ovaj rad je sačinjen od 7 poglavlja.

Prvo poglavlje sadrži osnovne informacije o radu i kratak opis rada.

U drugom poglavlju date su teorijske osnove koje su neophodne za razumevanje načina funkcionisanja digitalne televizije, Android operativnog sistema koji je proširen podrškom za reprodukovanje televizijskog sadržaja, istorijat PiP-a, kao i Android programska podrška za TV uređaje (TIF).

Treće poglavlje sadrži arhitekturu celog sistema i konceptualno su objašnjena rešenja aplikacija i ostalih komponenti neophodnih za izradu ovog rada.

U četvrtom poglavlju se govori o samoj implementaciji i povezivanju svih modula potrebnih za ispravan rad PiP-a.

U petom poglavlju su opisani načini ispitivanja i verifikacije implementiranih Android DTV aplikacija.

Šesto poglavlje sadrži kratak opis o stvarima koje su realizovane u radu.

U sedmom poglavlju je dat spisak literature koja je korišćena prilikom izrade ovog rada.

2. Teorijske osnove

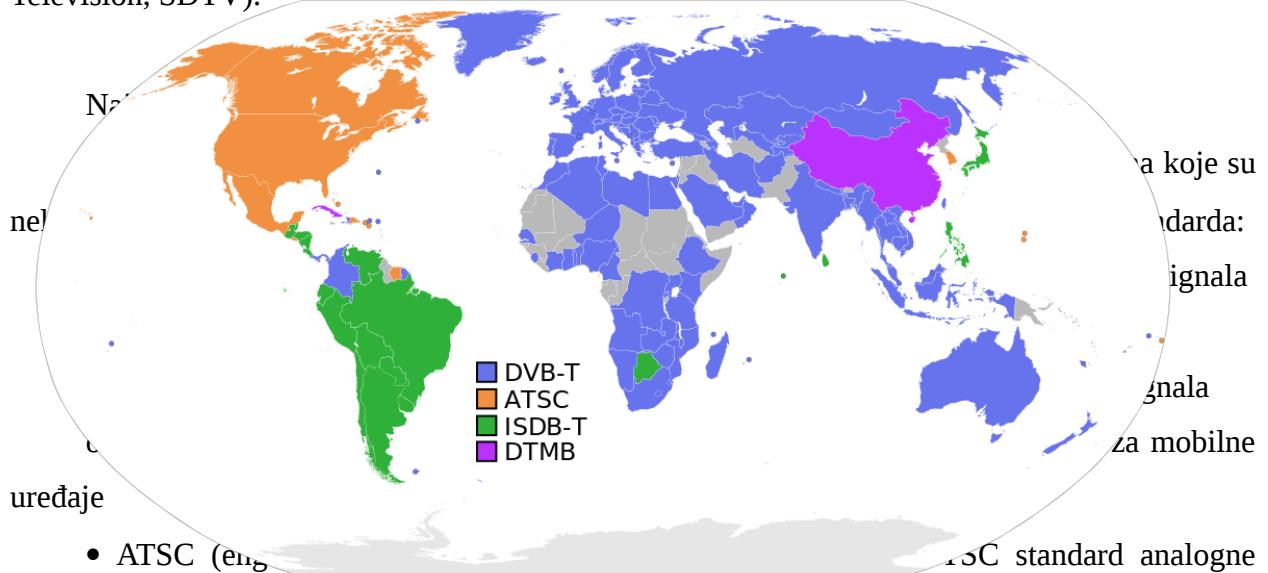
2.1 Digitalna televizija

Velika promena u televiziji, u tehničkom smislu, nastaje uvođenjem digitalne televizije. DTV je zamenila postojeće standarde prenosa televizijskog signala analognim putem i donela je mnogo više televizijskim gledaocima od značajno poboljšanog kvaliteta audio i video signala. Pored osnovne funkcije koja predstavlja prenos video i audio sadržaja, prenose se i dodatne informacije u digitalnom formatu kao što je npr. elektronski programski vodič (eng. Electronic program guide – EPG)[4].

Digitalna televizija dozvoljava prenos mnogo više kanala i koristi novu tehnologiju emitovanja za prevod servisa u binarni format. Svaki servis je kompresovan i konvertovan u digitalni tok podataka koristeći algoritme za kompresiju video sadržaja [5]. Ovaj oblik digitalne kompresije pakuje najmanje pet puta više kanala u dat distributivni mrežni opseg. Prethodno pomenute kompresije omogucuju da se prenose samo delovi slike koji su se promenili u odnosu na prethodni okvir podataka (engl. Frame) umesto da se šalje cela slika čime se smanjuje količina podataka koji se šalju da bi se rekonstruisala originalna slika. Nekoliko digitalnih signala se može preneti istovremeno pošto je prostor koji je potreban za prenos digitalnog servisa mnogo manji nego što je potreban za prenos analognog signala. Ovo predstavlja prvu značajnu evoluciju u televizijskoj tehnologiji još od pojave televizora u boji 1950-ih godina. TV sadržaj se smešta u MPEG transportni tok podataka (eng. Transport Stream – TS) koji služi za prenos TV sadržaja kao i prenos raznih kontrolnih tabela.

Digitalni prenos podataka obezbeđuje isporuku slike i zvuka boljeg kvaliteta koji u prenosu više ne mogu biti ometani interferencijom sa drugim signalima bez obzira na rastojanje na koje se slika i zvuk prenose. Slika i zvuk koju digitalni signal nosi su isti kao i na izvoru emitovanja, sve dok signal ne postane toliko slab da prijem više nije moguć. Postoji nekoliko različitih

digitalnih standarda koji se pojavljuju iz različitih regiona sveta kako prikazuje Slika 2.1. Standardi koriste MPEG-2 tehnologiju za audio i video kodiranje i multipleksiranje kako bi se omogućio adekvatan protok podataka potrebnih za podršku televizije visoke rezolucije (engl. High Definition Television, HDTV) ili televiziju standardne rezolucije (engl. Standard Definition Television, SDTV).



Slika 2.1 Standardi digitalne televizije u svetu

Koristi se za prenos zemaljskog, kablovskog i satelitskog signala.

- ISDB (engl. Integrated Services Digital Broadcasting) – Standard za definisanje digitalnog radio i televizijskog signala u Japanu, Brazilu i drugim zemljama. Ne podržava teletekst i prevod.
- DTMB (engl. Digital Terrestrial Multimedia Broadcast) – Zastupljen je u Kini, Hongkongu i Makau. Definiše standard za prenos zemaljskog signala ka fiksnim i mobilnim mrežama.

2.2 Android

2.2.1 Android Platforma

Android platforma sadrži programsku podršku za mobilne, prenosne uređaje, uređaje digitalne televizije (televizori i set-top-boks uređaji) i automobilsku industriju koji u sebe uključuje operativni sistem, sprežni sloj za povezivanje Java i C (eng. C) programskog jezika i pojedine aplikacije koje su dostupne korisnicima. Operativni sistem baziran je na Linuks (eng. Linux) jezgru.

Android je platforma koja je trenutno najrasprostranjenija za mobilne uređaje, ali su sve češće dostupna određena prilagođenja Android platforme za druge uređaje koji rade u realnom vremenu poput: tablet uređaja, televizijskih prijemnika i drugih.

Programski jezik Java prevodi se u bajt-kod. Ovaj kod se izvršava u Java virtuelnoj mašini. Za razliku od Java, drugi programski jezici se prevode u mašinski kod koji se direktno izvršava u procesoru. Neophodno je prilagoditi deo koda, napisanog u drugom programskom jeziku, tako da se može izvršiti iz Java programa. Programska podrška televizijskog prijemnika je napisana u C kodu, pa se zbog toga koristi JNI kako bi se obezbedila komunikacija sa C kodom i prosleđivanje podataka C kodu iz Java programa i obrnuto.

2.2.2 Android Open Source Project

Android ima aktivnu zajednicu programera[7] koji koriste Android projekat otvorenog koda (eng. Android open source project, skr. AOSP) da bi razvili i proizveli njihove lične verzije ovog operativnog sistema. Izdanja koja napravi ova zajednica obično donose nove dodatke i nadogradnje uređajima brže nego što bi to uradio zvanični proizvođač, sa uporedljivim nivoom kvaliteta. Takođe oni nastavljaju sa pružanjem podrške za zastarele uređaje koji više ne dobijaju zvanične nadogradnje ili spuštaju Android na uređaje koji su prvobitno bili namenjeni za neke druge operativne sisteme. Ova izdanja često pružaju korisniku sva prava i dozvole i sadrže izmene koje nisu obezbeđene od strane prodavca, kao što je mogućnost promena takta i napona procesora uređaja.

Istorijski gledano, proizvođači uređaja i mobilni operatori obično nisu pružali podršku za razvoj programske podrške nižeg sloja (eng. Firmware) od strane trećeg lica (eng. Third party). Proizvođači su iskazali zabrinutost u vezi neprikladnog rada uređaja koji koriste nezvaničnu verziju programske podrške i zbog troškova podrške koji proizilaze is toga. Osim toga, modifikovani firmware-i su nudili oredjene dodatke koje su proizvođači posebno naplaćivali. Kao rezultat toga, tehničke prepreke kao što su zaključan program za pokretanje STB uređaja i ograničeni pristup sistemskim dozvolama su veoma česti. Međutim, kako je softver koji je razvijen od strane zajednice postao više popularan i nakon izjave Kongresne biblioteke u Sjedinjenim Američkim Državama koja dozvoljava otključavanje mobilnih uređaja, proizvođači su popustili u pogledu razvoja softvera treće strane (HTC, LG Nokia i Sony), pružanjem podrške i podsticanjem razvoja softvera. Kao rezultat toga, tokom vremena potreba da se zaobiđu ograničenja fizičke arhitekture za instaliranjem nezvaničnog softvera se smanjila zato što se sve veći broj uređaja isporučuje sa otključanim ili sa mogućnošću otključavanja programa za pokretanje STB uređaja, slično Nexus seriji telefona, iako u većini slučajeva zahteva od korisnika da se odreknu garancije na uređaj. Međutim, uprkos tome što su proizvođači prihvatali

softver razvijen od strane zajednice, neki operateri u SAD i dalje zahtevaju da telefoni budu zaključani, frustrirajuće i programere i kupce.

2.2.3 Google Mobile Service

Google Mobile Services (skr. GMS)[8] su aplikacije i servisi obezbeđeni od strane Googla u nameri da poboljšaju korisničko iskustvo prilikom korišćenja mobilnih uređaja. GMS se razlikuje od Androida jer svi servisi i aplikacije ostaju pod potpunom “odvojenom licencom” od strane Googla. Neki od pripadnika ovih aplikacija i servisa su:

- Gmail – Googlov lični email snabdevač
- Chrome – Googlov internet pretraživač
- Google+ - Googlova društvena mreža
- Google Maps
- YouTube – sajt za reprodukciju video sadržaja
- Google Docs – Aplikacija koja omogućuje korisniku da pravi, menja ili posmatra različite vrste dokumenata
- Google Translate – Googlova stranica za prevodenje reči sa jednog jezika na drugi
- Waze – Pomaže korisnicima prilikom svakodnevnih vožnji. To je programska podrška u realnom vremenu koja daje informacije o saobraćaju

Dok AOSP obezbeđuje uobičajene funkcionalnosti poput elektronske pošte i poziva, GMS nije deo AOSP-a. GMS je jedino dostupan putem Googlove licence i sa sobom donosi pun opus popularnih aplikacija i servisa koji su bazirani na oblaku. Najbolja strana GMS-a je to što prilikom njegove ugradnje na uređaj ne zahteva dodatno nadoknade i plaćanja.

GMS aplikacije su unapred instalirane na sistemsku particiju Android uređaja. Naravno, korisnik može da instalira takođe i ostale aplikacije koje izaberu u Googlovoj prodavnici (eng. Google Play Service)

2.2.4 Android programska podrška za TV uređaje – TIF

Android programska podrška za TV uređaje (eng. TV Input Framework, TIF)[9] definiše aplikativnu programsku spregu i standardizuje način implementacije dopremanja emitovanog sadržaja do TV aplikacije. TIF omogućuje implementaciju TV ulaza preko kog je moguće reprodukovati sadržaj sa mrežnog poslužioca i/ili iz DVB transportnog toka podataka kao I pretragu televizije uživo i preporuke putem meta podataka objavljene od strane TV izvora. Programska podrškane teži ka tome da implementira TV standarde ili regionalne zahteve, ali proizvođačima uređaja omogućava da lakše ispune regionalne digitalne TV standarde bez ponovne implementacije.

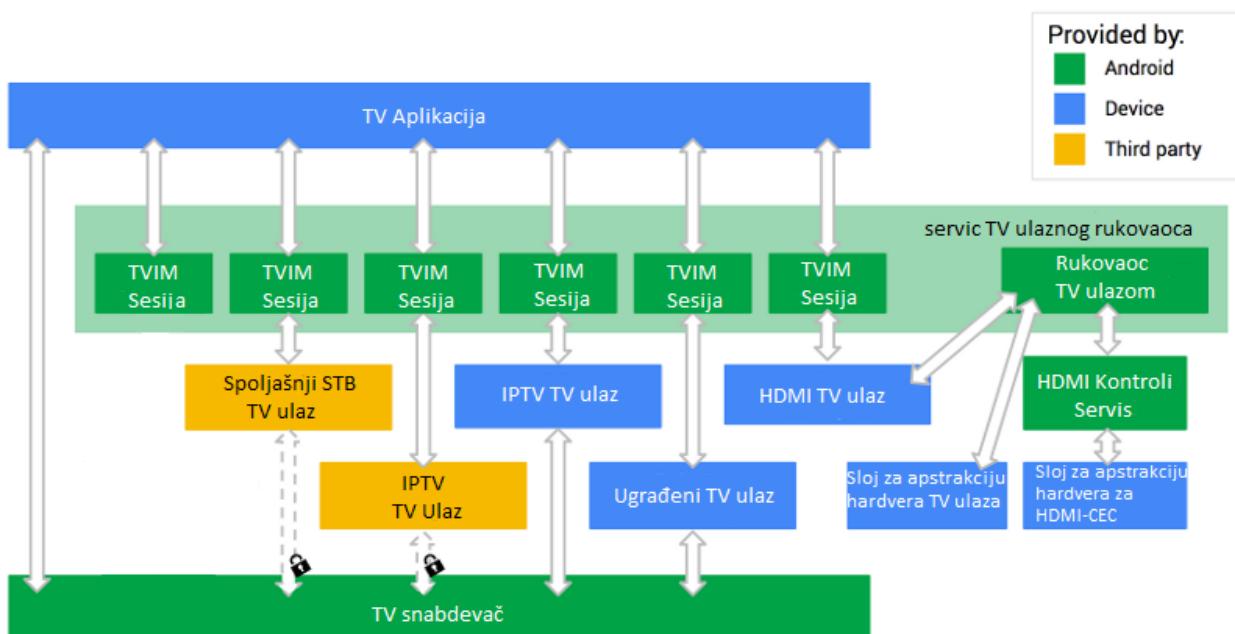
2.2.4.1 Komponente Android programske podrške za TV uređaje

TV aplikacija putem TIF-a pristupa ugrađenim modulima koji su isporučeni od strane proizvođača ili neke druge strane putem TV ulaznog rukovaoca.

TIF se sastoji od:

- TV aplikacija (eng. TV Application): aplikacija pomoću koje korisnik rukuje sa hibridnom programskom podrškom za TV uređaje
- TV ulazni rukovaoc (eng. TV Input Manager): omogućuje TV ulazima da komuniciraju sa TV aplikacijom
- TV snabdevač (eng. TV Provider): baza podataka sa kanalima, programima i pratećim dozvolama
- TV ulaz (eng. TV Input): aplikacija koja predstavlja jedan izvor TV sadržaja
- Ulagna TV fizička arhitektura (eng. TV Input Hardware Abstraction Layer): fizička definicija arhitekture koja omogućuje sistemskim TV ulazima pristup fizičkoj arhitekturi specifičnom za televiziju
- Roditeljska kontrola (eng. Parental Control): tehnologija koja omogućuje blokiranje kanala i programa
- HDMI-CEC: tehnologija koja omogućuje daljinsku kontrolu raznih uređaja putem HDMI-CEC poruka

Ove komponente će detaljnije biti objašnjene u tekstu koji sledi. Dijagram ispod vizuelno oslikava arhitekturu TIF-a.



Slika 2.3 TIF Arhitektura

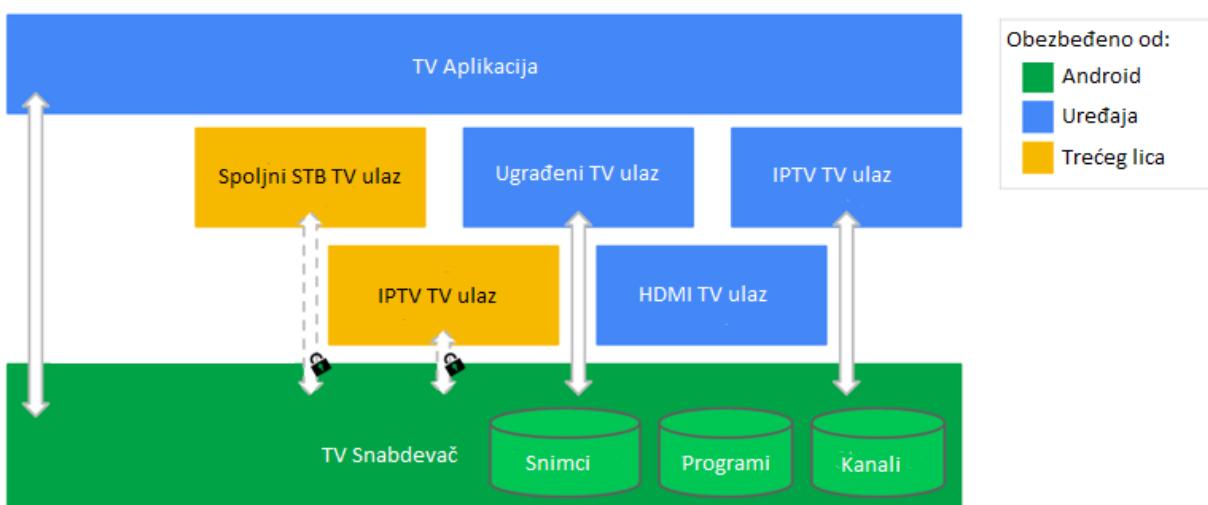
Na osnovu arhitekture sa slike se može zaključiti:

- Da korisnik može da vidi i da bude u direktnom kontaktu sa TV aplikacijom
- TV aplikacija prikazuje sadržaj sa TV ulaza

- TV aplikacija nije u mogućnosti da direktno komunicira sa TV ulazima, već je komunikacija ostvarena pomoću rukovaoca TV ulazom koji identificuje stanje TV ulaza za TV aplikaciju

2.2.4.2 TV snabdevač

TV snabdevač je komponenta koja u sebi sadrži bazu u kojoj se čuvaju lista kanala, programa (EPG) i snimljenog sadržaja. TV snabdevač takođe obezbeđuje i rukuje pratećim dozvolama kako bi TV ulazi bili u mogućnosti da vide samo svoje zapise, i na taj način obezbedile konzistentnost podataka koji su popunjeni od strane drugih TV ulaza.



Slika 2.4 TV snabdevač

Samo aplikacije koje se nalaze u privilegovanoj sistemskoj particiji mogu da čitaju podatke iz baze podataka TV snabdevača, dok samostalne aplikacije mogu da pristupe samo onim poljima u bazi podataka koje su oni popunili. Kao dodatak standardnim poljima za kanale i programe, baza podataka TV snabdevača takođe nudi i polja za binarno velike objekte (eng. BLOB- Binary Large Object,), u svakoj tabeli koju TV ulazi koriste za skladištenje proizvoljnih podataka. Binarni podaci mogu da sadrže prilagođene informacije, kao što je na primer frekvencija povezanog frekventnog odabirača, i mogu biti obezbeđeni u nekoj drugoj formi. Dostupno je i polje, column_searchable, koje može da izuzme neke kanale iz pretrage (kako bi podržale specifične zahteve pojedinih država za zaštitu sadržaja) kao i da omogući glasovnu ili tekstualnu pretragu TV sadržaja koja je jedna od osnovnih funkcionalnosti Android TV platforme.

Sva polja su vidljiva svakome sa pravom pristupa odgovarajućem redu. Nijedno polje nije direktno pristupačno korisniku. Korisnik samo vidi ono što TV aplikacija, sistemske aplikacije ili TV ulazi prikažu.

2.2.4.3 TV ulazni rukovaoc

TV ulazni rukovaoc (eng. TV Input Manager) obezbeđuje centralnu sistemsku aplikativnu spregu (eng. API, Application Programming Interface) za ceo TIF. TV ulazni rukovaoc nadgleda interakciju između aplikacija i TV ulaza i obezbeđuje funkcionalnost roditeljske kontrole. Sesija TV ulaznog rukovaoca mora biti kreirana jedan-na-jedan sa TV ulazom. Ovaj deo TIF-a može da dozvoli pristup aplikacijama TV ulazu tako da aplikacije mogu da:

- Listaju TV ulaze i proveravaju njihove statuse
- Naprave sesiju i rukuju slušaocima

Što se tiče sesija, TV aplikacije mogu da se povežu na TV ulaze samo preko URI-ja koji su dodale u bazu podataka TV snabdevača, osim prolaznih TV ulaza na koje se mogu povezati koristeći `TvContract.buildChannelUriForPassthroughInput()`. TV ulazi koji su obezbeđeni i imaju potpisani sertifikat od strane proizvođača uređaja ili druge aplikacije instalirane u sistemsku particiju imaju potpun pristup bazi podataka TV snabdevača. Ovaj pristup može da se iskoristi da se realizuju aplikacije koje će pretraživati sve dostupne TV kanale i programe.

Aplikacije mogu da naprave i prijave `TvInputCallback` koji će da pozove `android.media.tv.TvInputManager` ukoliko se promeni stanje TV ulaza ili ukoliko se doda ili ukloni neki od TV ulaza. Na primer, TV aplikacija može da reaguje kada se neki TV ulaz isključi tako što će ga prikazati kao isključenog i zabraniće mogućnost selektovanja istog.

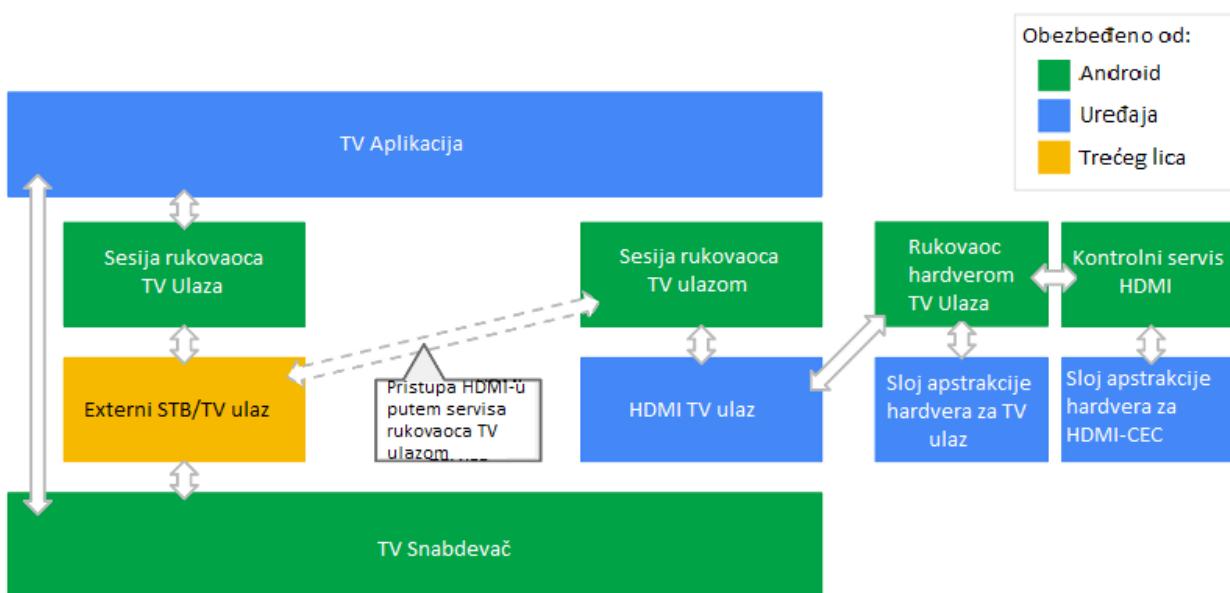
Rukovaoc TV ulazima apstrahuje komunikaciju između TV aplikacije i TV ulaza. Standardna sprega između rukovaoca TV ulazima i TV ulaza dozvoljava da više proizvođača naprave svoje TV aplikacije dok pomažu svim samostalnim TV ulazima da rade sa svim TV aplikacijama.

2.2.4.4 TV ulaz

TV ulazi (eng. TV input) su Android aplikacije u smislu da imaju `AndroidManifest.xml` i instaliraju se. Jedan TV ulaz predstavlja jedan izvor TV sadržaja na set-top boksu. Android TV podržava unapred instalirane sistemske aplikacije, aplikacije sa potpisanim sertifikatom od strane proizvođača uređaja i samostalne (eng. Third party) TV ulaze.

Neki ulazi, kao što je na primer set ili ugrađeni ulaz frekventnog odabираča, mogu biti obezbeđeni samo od strane proizvođača zato što oni komuniciraju direktno sa fizičkom arhitekturom. Drugi, kao na primer IPTV ili spoljni STB, mogu biti u vidu APK datoteka. Jednom kada se preuzmu i instaliraju takve aplikacije, novi ulazi mogu biti izabrani u samoj TV aplikaciji.

U nastavku teksta će biti opisan primer spoljašnjeg samostalnog STB TV ulaza.



S obzirom da TV ulaz ne može direktno da pristupi HDMI video izvoru koji pristiže, mora da pristupi preko TV rukovaoca ulazom i da koristi TV ulaz dobavljen od strane proizvođača uređaja.

Pomoću rukovaoca TV ulaza, spoljni STB TV ulaz može da komunicira sa HDMI TV ulazom i da ga „pita“ da prikaže video putem HDMI1. Na taj način STB TV ulaz može da kontroliše TV dok HDMI TV ulaz, koji je obezbeđen od strane proizvođača, obrađuje video.

2.2.4.5 TV aplikacija

Sistemska TV aplikacija (eng. TV application)[11] prezentuje TV sadržaj korisniku. Referentna TV aplikacija (Live TV) je obezbeđena zajedno sa Android platformom, koja može biti korišćena takva kakva jeste, može se promeniti, proširiti ili zameniti od strane proizvođača uređaja.

Najmanje što sistemska TV aplikacija mora da podrži na verziji Android Nugat je:

1. Podešavanje i konfiguracije:
 - Automatska detekcija TV ulaza
 - Da omogući TV ulazima da inicijalizuju postavke kanala
 - Kontrolu roditeljskih podešavanja
 - Promene (eng. Editing) kanala
2. Gledanje:
 - Pristup i navigacija svim TV kanalima
 - Pristup informacionim podacima TV programa
 - Prikaz elektronskog programskog vodiča
 - Podrška za više zvučnih i prevod traka
 - Prikaz PIN zahteva roditeljske kontrole (eng. PIN Challange)
 - Popunjavanje rezultata pretrage za TV kanale i programe
 - Rukovanje snimanjem digitalnog sadržaja i podrška za TV recording API
 - Podrška za slika-u-slici režim rada

- Podrška za gledanje sadržaja ispočetka (eng. Timeshift)

Ovaj skup funkcija će se povećavati sa novijim Android verzijama kako se i TIF API-ji budu proširivali. Da bi se proverilo da li aplikacija ima sve potrebne komponente postoji aplikacija CTV verifikator[10] koja obezbeđuje ispitivanje usklađenosti. Proizvođači moraju da implementiraju TV aplikaciju koja uključuje rezultate pretrage koji će biti uključeni u globalnu pretragu kako bi osigurali najbolje korisničko iskustvo. Referentna, Live TV, aplikacija pruža implementaciju koja obezbeđuje rezultate samostalnih ulaza kao i rezultate ugrađenih ulaza.

2.3 Slika-u-slici

2.3.1 Upotreba i istorijat klasičnog slika-u-slici proširenja

Slika-u-slici (eng. Picture-in-picture)[11] je karakteristika određenih TV prijemnika i njima sličnih uređaja. Jedan program je prikazan preko celog TV ekrana (eng. Full screen) dok se u isto vreme jedan ili više programa prikazuju u umetnutim, manjim prozorima. Zvuk u ovakvom načinu korišćenja televizora obično reprodukuje sa glavnom (eng. Main) programa koji je prikazan preko celog ekrana. Slika-u-slici se najčešće koristi da bi se gledao jedan program dok se čeka početak nekog željenog programa ili dok na željenom programu traju reklame. Ovaj režim rada zahteva 2 različita frekventna birača kanala ili neki drugi izvor signala kako bi se mogli snabdeti i glavni i mali prozor. Neki televizori imaju u sebi ugrađena dva birača kanala, dok ostali koji imaju jedan zahtevaju spoljni, dodatni izvor signala što može biti spoljni birač kanala, DVD plejer ili kablovski STB.

Dodavanje slike u već postojeću sliku je urađeno mnogo pre nego što je PIP bio dostupan u potrošačkim proizvodima. Princip slike u slici se prvi put pojavio 1976. godine za vreme otvaranja Olimpijskih igara u Montrealu, gde je tokom cele ceremonije bio umetnut video prikaz olimpijskog plamena.

Prvobitna potrošačka implementacija ovog dodatka je bio Multivision set-top box[12] koji nije postigao značajan komercijalni uspeh. Nakon toga PIP je postao dostupan kao karakteristika naprednih televizijskih prijemnika (televizora). Prva široko rasprostranjena potrošačka implementacija slike-u-slici je napravljen od strane Philips kompanije 1983 godine u njihovim najsuklјim televizorima. Bilo je omogućeno puštanje jednog kanala u crno-beloj varijanti, a prozor je bio smešten u jednom od četiri ugla televizijskog ekrana. U to vreme televizija je i dalje bila analogna, što je ponajviše uticalo na cenu izrade ovog dotatka, kao i samu cenu televizora. U današnje vreme, kada je digitalna televizija u ekspanziji, posredstvom set-top box-a

mnogo je jeftinije napraviti podršku za sliku-u-slici, tako da su i cene televizora nakon digitalizacije znatno opale.

2.3.2 Android TV slika-u-slici

U Android 7.0 (Android Nougat) korisnici mogu da gledaju video u prozoru koji je smešten u uglu ekrana dok se kreće kroz ili izmedju aplikacija. Slika-u-slici režim omogućava aplikacijama da puštaju video u dodatom prozoru dok se druga aktivnost (eng. Activity) nesmetano izvršava u pozadini. PIP prozor omogućuje korisnicima da rade nekoliko stvari istovremeno (eng. Multitask) dok gledaju Android TV, što u mnogome pomaže korisnicima da budu više produktivni.

Korisnik putem aplikacije može da odluči kada će preći u PIP režim. Neki od primera korišćenja ovog režima su:

- Kada korisnik želi da pretražuje neke druge sadržaje na svoj televizijskom prijemniku.
- Kada korisnik završava sa gledanjem odredjene epizode željenog sadržaja. U tom slučaju se video sadržaj koji se prikazuje prebacuje u smanjeni prozor dok se na glavnom delu ekrana ispisuju informacije od sledećoj epizodi i slično.
- Kada korisnik želi da pronađe neke dodatne sadržaje o trenutno puštenom videu. Video se u ovom slučaju takođe prebacuje u umanjeni ekran dok se glavni ekran oslobađa za pronalazak dodatnih informacija

Prozor predviđen za PIP je veličine 240x135dp (eng. Density-independent Pixels) i prikazan je u jednom od četiri ugla ekrana, što se podešava sistemski. Korisnik može da otvori padajući PIP meni u kome mogu da izaberu da li će se PIP prebaciti u full screen mode ili će ga skroz ugasiti pritiskom i držanjem taster HOME na svoj daljinskom upravljaču. U slučaju da korisnik pokrene neku aplikaciju za reprodukciju multimedijalnog sadržaja, na primer sa Youtube-a, PIP prozor se automatski uklanja.

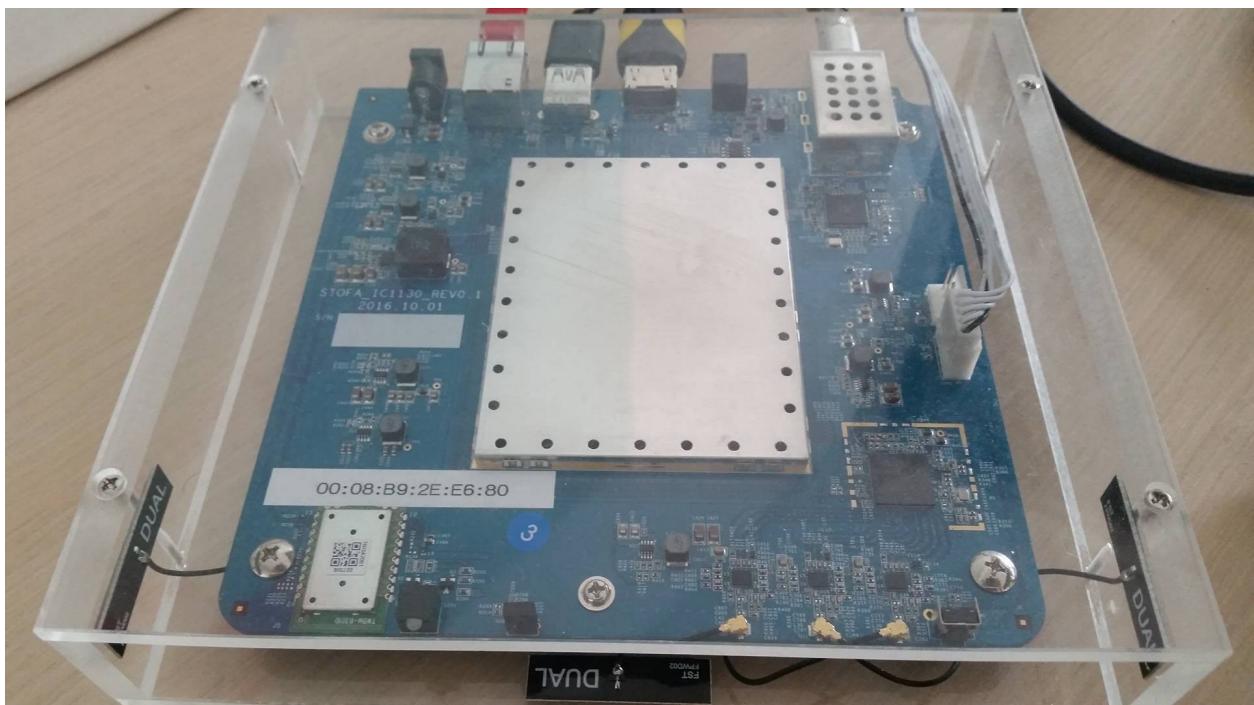
3. Koncept rešenja

Integracija digitalne televizije u okviru Android platforme zahteva postojanje više slojeva programske podrške. Svaki od slojeva predstavlja prilagođen servis za druge slojeve. U nastavku je opisana ciljna platforma, predloženo okruženje korišćeno u rešavanju datog problema, odnosno programska podrška TV prijemnika i potrebni elementi sistema.

3.1 Opis ciljne platforme

DTV prijenik koji je korišćen pri izradi ovog rada je Android platforma bazirana na Broadcom 7252S čipsetu, sa 2 video dekodera koji mogu da dekoduju i 4K video sadržaj. Komponente koja ova platforma poseduje su:

- 2 Video dekodera 4K UHD HEVC 2048p60/MP4/MPEG2/VP9/HDR
- 6 Audio dekodera MPEG-1 layers 1,2, AAC LC, HE-AACv1/2, MP3
- Dvojezgarni procesor: ARMv7 Processor rev 3 (v7l) @1200MHz
- RAM memorija - SDRAM 2GB DDR4 (64-bit memory bus)
- Wi-Fi i BlueTooth 802.11ac 3x3 2.4GHz/5GHz
- 4 frekvencijska odabirača - RF NorDig compliant DVB-C quad tuner (BCM31584)
- USB 3.0 priključak
- HDMI 2.0a Tx (HDCP 2.2 & CEC supported) izlaz
- Ethernet utičnicu
- 2D grafički procesor brzine 2x 800Mpix/s
- 3D grafički procesor brzine 2.7Gpix/s OpenGL ES 2.0/3.0



Slika 3.6 Broadcom 7252S set-top box

3.2 Programska podrška STB uređaja

Funkcionalnost Android TV platforme je orijentisana više ka načinu kako prezentovati korisnicima TV sadržaj i ne sadrži direktnu podršku za upravljanje DVB transportnim tokom podataka kao ni upravljanje fizičkim komponentama DTV prijemnika. Da bi se omogućilo upravljanje fizičkim komponentama DTV prijemnika i upravljanjem DVB transportnim tokom podataka na postojeće slojeve Android platforme dodat je deo DTV programske podrške [13]. Na slici 3.2 prikazana je struktura programske podrške TV prijemnika zasnovanom na Android platformi.



Slika 3.7 Struktura programske podrške TV prijemnika

TV Aplikacija je komponenta koja je definisana od strane TV ulaznog radnog okvira i ona predstavlja grafičko rešenje Android TV aplikacije. Zadužena je za prikaz svih grafičkih elemenata, odnosno informacija dobijenih u komunikaciji sa programskom podrškom za TV uređaje i opsluživanje upravljačkih komandi pristiglih od strane krajnjeg korisnika. TV aplikacija se oslanja na programsku podršku za TV uređaje. **Programska podrška za TV uređaje**, kao i TV aplikacija su opisane u poglavlju 2.2.4 Android programska podrška za TV uređaje – TIF.

Servis programske podrške za TV uređaje predstavlja standardni Android servis koji objedinjuje DTV funkcionalnost i obezbeđuje skup DTV funkcija izložen gornjim slojevima (eng. Application programming interface, u daljem tekstu API). Pošto Android servis može da pribavlja i isporučuje podatke u pozadini, dok je u isto vreme prikazana grafička sprega kojoj servis obezbeđuje te podatke, TIF propisuje ovaj mehanizam koji predstavlja sponu između sprege za rukovanje DTV srednjim slojem i TV ulaznim okvirom. Servis programske podrške za TV uređaje je opisan u okviru 2.2.4.4 Tv ulaz poglavljia.

DTV servis srednjeg sloja je proširenje DTV srednjeg sloja i on pruža direktnu spregu sa Linux servisom srednjeg sloja pružajući dodatne funkcionalnosti visokog nivoa realizovane u programskom jeziku Java.

Sloj za spregu sa DTV srednjim slojem (eng. Middleware Abstraction Layer) je sprega koja omogućava programskoj podršci srednjeg sloja da komunicira sa višim slojevima. Ona omogućava višim slojevima da koriste funkcionalnosti srednjeg sloja bez poznavanja unutrašnjih detalja realizacije

Programska podrška srednjeg sloja(eng. Middleware) obezbeđuje DTV funkcionalnost. Zadužena je za rukovanje i obradu digitalnog toka podataka, kao i za podešavanje DTV prijemnika, upravljanje listom programa, audio i video prikazom i pristup informacijama o emisijama. Takođe, obezbeđuje podršku za prikaz prevoda i teleteksta, zakazivanje podsetnika, funkcionalnost snimanja video sadržaja, izvršavanje MHEG aplikacije, prikaz HbbTV sadržaja itd. Komunikaciju sa fizičkom arhitekturom ostvaruje preko najnižeg sloja programske podrške, HAL sloja.

Sloj za apstrakciju fizičke arhitekture (eng. Hardware Abstraction Layer, skr. HAL) je najniži nivo programske podrške i predstavlja apstrakciju fizičke arhitekture TV prijemnika. Zadužen je za rukovanje složenim podacima koje dobija od fizičke arhitekture i obezbeđuje brojne usluge višim slojevima, kao što su podešavanje frekvencije, jačine zvuka, upravljanje formatima slike itd. Takođe, raspolaze skupom funkcija koje apstrahuju funkcionalnosti vezane za korišćeni operativni sistem (sinhronizacija, kritične sekcije, semafori itd) i obaveštava više slojeve o nastalim događajima. Uvođenjem ovog sloja, proizvođači DTV programske podrške ne moraju poznavati specifičnosti fizičke arhitekture, odnosno omogućeno je izvršavanje programske podrške DTV srednjeg sloja na Android platformi.

3.3 Proširenje programske podrške za PIP funkcionalnost

3.3.1 Android Nougat PIP

Od Nougat verzije Androida, sve TV Aplikacije koje podležu Google sertifikaciji moraju da imaju implementiranu PiP podršku. Ova podrška se implementira u samoj korisničkoj aplikaciji, kojoj treba omogućiti da menja svoju veličinu (eng. Resizable). Nakon ulaska u PiP režim rada potrebno je promeniti dimenzije video sadržaja, kako bi se sprečilo odsecanje videa i omogućilo da se ceo video sadržaj prikaže u PiP prozoru. Ova promena veličine video prozora se odvija u najnižim nivoima DTV srednjeg sloja. Na ovaj nacin se omogućava korisnicima da mogu da gledaju TV sadržaj uporedno sa nekom od druge funkcionalnosti koje nudi Android platforma.

3.3.2 Klasičan PiP

Klasican pip predstavlja reprodukciju dva TV programa uzivo gde se drugi TV program prikazuje u umanjenom prozoru. Da bi se omogućilo puštanje dva različita TV programa potrebno je upravljati sa više različitih fizičkih komponenti istovremeno. Te komponente su:

- Frekvencijski odabirač
- Demultiplexer
- Audio/Video dekoderi
- HDMI Izlaz

Da bi se pustila dva različita kanala potrebno je obezbediti za različite dekodere i frekvencijeske odabirače.

S obzirom da se PiP u ovom radu zasniva na puštanju odabranog kanala u glavnom prozoru i njegovog sledećeg u umanjenom prozoru potrebno je dobaviti sve potrebne informacije kako bi se omogućila reprodukcija željenog kanala.

Takodje, potrebno je omogućiti zamenu glavnog i PIP prozora. Ovaj deo zahteva reakciju na događaj, odnosno da se prozori zamene nakon pritiska tastera na tastaturi ili daljinskom upravljaču. Pomenute akcije se iniciraju kroz TV aplikaciju, a upravljačka logika je smestena u okviru DTV srednjeg sloja. Da bi se događaj registrovao u DTV srednjem sloju, zarad promene prozora, on se mora proslediti iz TV aplikacije. Da bi ceo sistem ostao u konzistentnom stanju treba pravilno zahtevati i zapamtiti resurse koji su korišćeni prilikom korišćenja ove funkcionalnosti.

U koliko korisnik želi da gleda isključivo sadržaj sa glavnog ekranra, potrebno je prekinuti reprodukovanje sadržaja u PIP prozoru. Ova funkcionalnost, kao i prethodno opisana zamena prozora, se poziva nakon prosleđenog događaja, odnosno pritisnutog tastera. Nakon toga

TV program pušten u PiP prozoru se zaustavlja. Ako korisnik želi da vrati PiP prikaz, pokretanje TV sadržaja u PiP prozoru se vrši pritiskom na to isto dugme.

Kako bi TV aplikacija imala dostupne funkcionalnosti koje nisu deo Android programske sprege za TV uređaje, koristi se Android4TV programska podrška[14] (personalni video snimač, itd). U Android4TV programskoj podršci ne postoji potrebna podrška za realizaciju prethodno opisane funkcionalnosti za PiP režim rada. Bilo je potrebno omogućiti poziv funkcija za zamenu prozora koje su implementirane u nižim nivoima programske podrške.

4. Programsko rešenje

U ovom poglavlju biće detaljno opisano programsko rešenje za obe verzije slika-u-slici funkcionalnosti koje su detaljno opisane u prethodnim poglavljima. Android aplikacija koja koristi funkcionalnosti programske podrške televizijskog prijemnika se razvija korišćenjem skupa alata i programskih elemenata za razvoj programske podrške u Android okruženju (engl. Android Software Development Kit – Android SDK). Kao što je prikazano u prethodnom poglavlju aplikativni deo se oslanja na niže delove, pa će korišćene funkcije kao i funkcije na koje se one oslanjaju u najnižim slojevima biti opisane. Ovo programsko rešenje se može podeliti na dve celine :

- Realizacija Android PiP funkcionalnosti u okviru korisničke TV aplikacije
- Realizacija Klasičnog PiP-a u okviru TV ulaznog servisa

Android aplikacija je sačinjena od zasebnih programskih modula, čija je uloga prikaz grafičkih komponenti, prikupljanje komandi od strane daljinskog upravljača i modula koji predstavlja upravljačku logiku aplikacije. Upravljačka logika predstavlja programsku realizaciju načina na koji će određeni događaji biti prosleđeni programskoj podršci televizijskog prijemnika, kao i načine kako će informacije iz programske podrške televizijskog prijemnika biti prikazani krajnjem korisniku putem različitih grafičkih komponenti aplikacije. U delu zaduženom za upravljačku logiku se prozivaju određene funkcije programske podrške televizijskog prijemnika, gde se pomoću Android4TV programske sprege, objekti koji su bili parametri Java funkcija prepisuju u C strukture koje se dalje prosleđuju MAL sloju programske podrške televizijskog prijemnika. Iz sloja za spregu sa srednjim slojem dolazi do pozivanja određenih metoda iz programske podrške DTV srednjeg sloja, to jest njenih određenih modula, u zavisnosti od definisane akcije, koji završavaju na HAL sloju.

4.1 Realizacija Android PiP podrške u okviru TV korisničke aplikacije

Android aplikacija sama po sebi ne može da radi u PiP modu. U koliko želimo da naša aplikacija podržava PiP potrebno je u okviru manifest aplikacije, u deo koji opisuje aktivnost, dodati atribute kako bi naša aplikacija imala mogućnost promene veličine, `android:resizeableActivity = "true"`, kao i samu podršku za PiP, `android:supportsPictureInPicture = "true"`.

U grafičkoj korisničkoj spregi (eng. Graphic User Interface, skr. GUI) urađeno je dodavanje elementa za ulazak u PiP režim rada. U ovoj aplikaciji, element za ulazak u PiP režim rada je dodat kao stavka menija (eng. Menu) koja se zove “Picture in mode”.

U koliko korisnik odabere opciju za ulazak u PiP režim rada, poziva se slušalac



Slika 4.8 Prikaz menija u kome se nalazi opcija za PiP režim rada `OnMenuItemClickListener` i putem njega se poziva metoda `enterPictureInPictureMode()`.

Kad je aplikacija u PiP režimu rada, aplikacija treba isključivo da prikazuje video sadržaj. Potrebno je ukloniti grafičke korisničke elemente pre nego što aplikacija pređe u PiP režim rada, a potrebno je elemente vratiti kada se aplikacija vrati u režim rada preko celog ekrana (eng. Full screen mode). Da bi se elementi sklonili, odnosno ponovo pojavili, potrebno je preklopiti `Activity.onPictureInPictureModeChanged(boolean isInPictureInPictureMode)` metodu, koja je dostupna u Android SDK-u, u kojoj je potrebno ukloniti ili ponovo prikazati

elemente. U koliko je aplikacija u PiP režimu rada, polje koje predstavlja zabranu korišćenja dugmadi postavljamo na tačno. Takođe, menjamo vidljivost menija i liste kanala ,koja se nalazi sa leve strane ekrana i u kojoj korisnik bira koji program želi da pusti, pomoću `channelsFrame.setVisibility(View.INVISIBLE)`. Ako se korisnik vraća iz PiP režima rada u normalan režim zabrana korišćenja dugmadi postavlja se na netačno, čime je ponovo omogućeno korisniku da menja kanale i bira opcije iz glavnog menija.

Pošto su aplikacija koja prikazuje video sadržaj i servis u kome se zapravo vrši reprodukcija tog sadržaja dva odvojena modula potrebno je voditi računa o promeni površine ekrana u kojoj treba da se prikaže video. Da bi PiP režim rada radio ispravno, potrebno je da ceo video sadržaj bude prikazan u PiP prozoru. Potrebno je promeniti dimenzije videa koji se trenutno reproducuje. Kako bi se detektovala promena površine u kojoj treba da se prikaže video sadržaj (eng. Video surface) prijavljena je funkcija povratnog poziva (eng. Callback) `static void sb_geometry_update(void *s, unsigned int x, unsigned int y, unsigned int width, unsigned int height)` u okviru rukovaoca (eng. Driver) za prikaz grafičkih ravnih Androida. Funkcija povratnog poziva je dodata u najniži sloj srednjeg sloja, HAL, u okviru modula za apstrahovanje TV ekrana (eng. TV display abstraction layer, skr. TDAL). U njenom telu se poziva funkcija `void doScaling (unsigned int x, unsigned int y, unsigned int width, unsigned int height)` koja je zadužena za promenu dimenzija video površine, gde su x i y koordinate gornjeg levog ugla prozora, a width i height visina i širina prozora.

Takođe, ako je prilikom ulaska u PiP režim rada omogućena funkcionalnost klasičnog PiP-a, potrebno je samo glavni ekran prikazati u umanjenom prozoru.

4.2 Realizacija PiP podrške u okviru ulaznog TV servisa

4.2.1 Puštanje dva TV programa uporedo

Implementacija PiP podrške je uradjena tako da može bilo koja kombinacija kanala da se prikaze u PiP i glavnom prozoru. Prilikom izrade ovog rada cilj je bio puštanje 2 televizijskog programa istovremeno. Tokom implementacije odabrana su 2 različita načina izbora kanala za prikaz u PiP prozoru:

1. Izabrani program iz liste se prikazuje u glavnom prozoru, a sledeći program u listi se automatski prikazuje u umanjenom prozoru
2. Mogućnost proizvoljnog izbora kanala iz liste koji će biti prikazan u umanjenom prozoru

U nastavku dokumenta biće opisana oba režima.

Kada korisnik želi da promeni program (eng. Zapp) TV servisu je potrebno da se zakači na frekvenciju (eng. Tune) kanala putem kojeg se emituje odabrani program. U TV ulaznom servisu

(eng. TV Input Service), u okviru klase `TvSession.java` `public boolean onTune(Uri channelUri)` je metoda koja se poziva prilikom promene kanala i putem koje se dobavljaju informacije o kanalu koji će biti reprodukovani. Te informacije su smeštene u polju `mCurrentChannel`, koje predstavlja objekat klase `ChannelDescriptor`. Sledeće, obezbeđene su informacije o narednom kanalu iz liste, zarad njegove reprodukcije u umanjenom prozoru, koje će biti smeštene u polju `mNextChannel`, objektu istoimene `ChannelDescriptor` klase. Ove informacije se dobavljaju putem metode `public ChannelDescriptor getNextChannel(ChannelDescriptor currentChannel)` klase `ChannelManager.java` u kojoj se na osnovu indeksa odabranog kanala, iz liste svih pronađenih kanala, pronalazi sledeći kanal. U koliko je odabran poslednji kanal u listi, kanal koji će biti reprodukovani u malom prozoru je kanal koji je prvi u listi kanala. Dalje se poziva se metoda za reprodukciju video i audio sadržaja `startPlayback()`. U okviru ove metode se pozivaju metode, u kojima se proverava kog je tipa kanal koji treba da bude pušten, a koje se nalaze u klasi `DtvManager.java`, `start(ChannelDescriptor CurrentChannel)` i `startPip(ChannelDescriptor NextChannel)`. U zavisnosti od toga da li je tip prenosa kanala zemaljski (eng. Terrestrial, skr. TER), kablovski (eng. Cable, skr. CAB), satelitski (eng. Satelite, skr. SAT) ili internet (skr. IP, koji može biti unicast, multicast ili adaptivni (HLS ili DASH)) pristupa se različitim metodama za reprodukciju. Ova provera se radi i za program u glavnom i za program u malom prozoru. U ovom radu u fokusu ce biti kablovski tip kanala.

Sledeće u niz metoda kroz koje se pozivaju da bi na ekranu bio prikazan video su `private boolean startDvb(ChannelDescriptor currentChannel)`, koja je zadužena za rukovanje i puštanje sadržaja u okvirima glavnog prozora, i `private boolean startDvbPip(ChannelDescriptor nextChannel)`, za rukovanje i puštanje video sadržaja u umanjenom prozoru. U okviru ovih metoda se startuje TV servis `startService()`. Ova metoda se nalazi u okviru `tv_service-a` koja je u sastavu Android4TV programske podrške. Android4TV programska podrška komunicira posredstvom sloja za spregu sa DTV srednjim slojem sa najnižim delovima srednjeg sloja, HAL-om, u kojem dolazi do komunikacije sa rukovaocima (eng. Driver) i onda i do samog prikaza video sadržaja na ekranu. Isti sled događaja i poziva je i za PiP kanal, s jedinom razlikom koja se ogleda u promeni dimenzija prozora u kom će se videti video sadržaj, pa je zarad toga opisano samo programsko rešenje za reprodukciju glavnog kanala.

Korisnik može na proizvoljan način da odabere kanal koji želi da pusti u sporednom prozoru. Ovaj proces započinje otvaranjem liste kanala i pronalaženjem kanala koji korisnik želi da reprodukuje u sporednom prozoru. Umesto tastera za reprodukciju odabranog kanala u glavnom prozoru, korisnik pritiska namenjen za puštanje kana u sporednom prozoru (na tastaturi

taster "P"). Prilikom pritiska tastera, pravi se novi intent konstruktorom kom se kao parametar prosleđuje string "`com.iwedia.tiftestapp.CHANNEL`". Novonapravljenom intentu se potom zakači vrednost "p_pressed" i redni broj kanala u listi. Nakon što je intent poslat svim aktivnim aplikacijama i servisima, TV ulazni servis prihvata intent koji sadrži "`com.iwedia.tiftestapp.CHANNEL`" pomoću primaoca (eng. Receivera) koji reaguje samo na vrednosti koje filter (eng. IntentFilter) odredi kao validne. Unutar samog BroadcastReceivera preklopljena je (eng. Override) metoda `public void onReceive(Context context, Intent intent)`. U ovoj metodi se parsira sadržaj intenta i bira koja akcija će biti sledeća izvršena. Ovo parsiranje se vrši pomoću metoda `intent.getExtras().getString("value")`. U koliko je vrednost koja se dobije nakon parsiranje "p_pressed" poziva se metoda `setNextChannel(int channelIndex)` gde se na osnovu rednog broja kanala iz liste dobavljuju podaci o samom kanalu i koji se potom smeštaju u polje `mNextChannel`. Po pribavljanju podataka dolazi do ponovnog startovanja servisa.

4.2.2 Mogućnost zamene glavnog i PiP prozora

Mogućnost zamene glavnog i PiP prozora je funkcionalnost koja omogućuje korisniku brže prebacivanje na program koji ga interesuje ili na program koji čeka da počne.

Početak implementacija se nalazi iznad TV ulaznog servisa, odnosno u TV korisničkoj aplikaciji. Pošto do zamene prozora dolazi tek nakon pritiska na taster potrebno je nižim delovima, odnosno TV ulaznom servisu, proslediti zahtev za zamenu kanala. Registrovan je onKeyListener kako bi došlo do okidanja događaja prilikom pritiska na taster na tastaturi, odnosno daljinskom. Kada se taster pilikom pritiskanja otpusti poziva se metoda `public boolean onKeyUp(int keyCode, KeyEvent event)` koja je implementirana u MainActivity.java klasi u okviru korisničke aplikacije. U ovoj metodi dolazi do ispitivanja koji je taster pritisnut. Ukoliko je pritisnut taster „S“ `keyCode` koji je prosleđen funkciji će imati vrednost `KeyEvent.KEYCODE_S`. U tom slučaju se pravi novi intent pomoću konstruktora kom se kao parametar prosleđuje string "`com.iwedia.tiftestapp.SWAP`". Tom intentu se potom nakači dodatak sa vrednošću `"s_pressed"` i zatim se intent šalje svim aktivnim aplikacijama i servisima.

TV ulazni servis očekuje intent koji sadrži gore navedeni string i na osnovu tog stringa servis prepoznaje da je u pitanju intent koji je poslala korisnička aplikacija na zahtev korisnika. U okviru `TvService.java` klase, da bi ovaj način komunikacije između aplikacije i servisa funkcionisao registrovan je `BroadcastReceiver`. Prvo je napravljen IntentFilter kome je dodata vrednost striga koji je korišćen prilikom kreiranja intenta na predajnoj strani, a nakon toga se registruje primalac (eng. Receiver) koji će reagovati samo na vrednosti koje filter odredi kao

ispravne. Unutar samog BroadcastReceiver-a prekopljena je (eng. Override) metoda `public void onReceive(Context context, Intent intent)`. U ovoj metodi se parsira sadržaj intenta i bira koja akcija će biti sledeća izvršena. Ovo parsiranje se vrši pomoću metoda `intent.getExtras().getString("value")`. U koliko je vrednost ovog stringa „`s_pressed`“ poziva se metoda `callSwap()` koja pripada instanci objekta `mCurrentSession` klase `TvSession.java`.

Metoda `callSwap()` je zadužena za poziv metode `swapScreens()` klase `DtvManager`, nad već instanciranim `mDtvManager` objektom.

Metoda `swapScreens()` je ključna za zamenu prozora u samom TV ulaznom servisu. U njoj se pozivaju funkcije nižih slojeva programske podrške. Prvi korak je dobavljanje kontrole servisa nad instancom objekta `DtvManager` i potom pozvanje funkcije `swapScreens()` koja je implementirana u okviru Android4TV programske podrške o čemu će biti reči u nastavku poglavlja. Nakon poziva ovog niza metoda, zamenjuje se vrednost poljima koja predstavljaju resurse za glavni i PiP prozor. Ovo je bitno kako bi se nakon zamene reprodukcija kanala nastavila neometano. Takođe je upamćeno koji je kanal pušten u glavnem, a koji u PiP prozoru zarad funkcionalnosti stopiranja i ponovnog pokretanja PiP-a, da ne bi došlo do reprodukcije dva ista kanala.

Dalje, proširena je Android4TV programsku podršku jer u okviru nje nije postojalo proširenje za zamenu prozora. U ovoj programskoj podršci dodata je funkcija `public Android4TV swapScreens()` i u njoj se putem `DTVManager` klase dobavlja DVB servis metodom `getDvbService()` i na tom servisu se poziva metoda `swapScreens()` koja je dodata u `tvplayer_service` daemon servisu koji je zadužen za predstavljanje srednjeg sloja.

Implementacija funkcije za zamenu prozora je urađena u okviru `tvplayer_service` servisa. Funkcionalnost je dodata u `TvService` i `ITvService` modulima.

U zaglavlju `ITvService`, `ITvService.h` u `Tv_CallMsg_e` enumeraciji dodat je element `SWAP_SCREENS` koji se prosleđuje modulu nakon poziva funkcije za zamenu prozora. Takođe, dodata je deklaracija virtuelne metode `virtual uint32_t swapScreens()` čija definicija je data u `ITvService.cpp` datoteci.

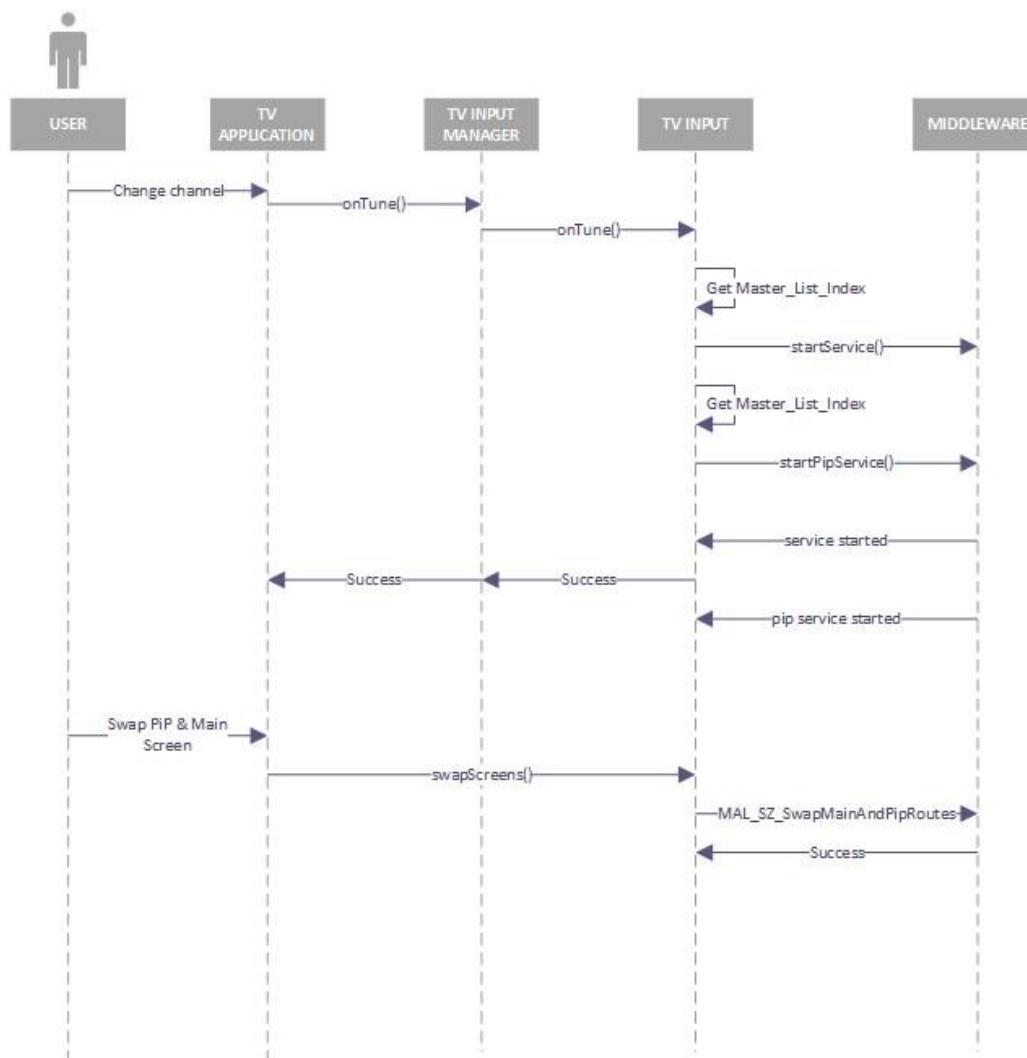
U samoj `ITvService.cpp` datoteci izmenjena je `status_t BnTvService::onTransact(uint32_t code, const Parcel &data, Parcel *reply, uint32_t flags)` funkcija. Izmena se ogleda u dodavanju jednog slučaja u switch-case bloku, odnosno, šta da se dešava kada kod koji bude prosledjen funkciji bude „`SWAP_SCREENS`“. Iz `const Parcel &data` su izvučeni identifikatori resursa glavnog i PiP prozora i pozvana je funkcija `swapScreens` nad ovim identifikatorima.

Što se tiče TvService modula izmene su jednostavnije. U njegovom zaglavlju, `TvService.h` datoteci, deklarisana je funkcija `virtual uint32_t swapScreens()`.

Definicija ove funkcije se nalazi u `TvService.cpp` datoteci. U samom telu funkcije dolazi do poziva funkcije za zamenu prozora koja postoji u sloju za apstrahovanje srednjeg sloja `MAL_SZ_SwapMainAndPipRoutes()`.

Ostale potrebne funkcije su već bile implementirane u slojevima ispod sloja za spregu sa srednjim slojem što implicira da je glavni zadatak bio povezivanje viših slojeva, odnosno TV ulaznog servisa, putem Android4TV programske podrške i `tvplayer_service` daemon-a (servis u pozadini koji rukuje zahtevima ostalih servisa) sa sprežnim slojem za srednji sloj.

Mehanizam promene kanala i zamene glavnog i PiP prozora dat je na slici 4.2.



Slika 4.9 Promena kanala i zamena glavnog i PiP prozora

4.2.3 Mogućnost uklanjanja i pojavljivanja PiP-a

Korisniku je potrebno omogućiti da gleda željeni program bez dodatnih video sadržaja na ekranu, ukoliko on to zahteva. Obrnuto, potrebno je omogućiti da korisnik naknadno uključi PiP prozor.

Uklanjanje i pojavljivanje PiP prozora je zavisno od korisnika, pa samim tim se vezuje za korisničku TV aplikaciju. Isto kao i sa zamenom prozora, uklanjanje odnosno ponovno pojavljivanje PiP-a se odvija na prisak tastera na tastaturi ili daljinskom upravljaču, nakon čega se pokreće niz operacija zarad uklanjanja ili puštanja TV sadržaja u PiP prozoru.

Princip komunikacije je isti kao i u mehanizmu zamene prozora sa stanovišta korisničke TV aplikacije, jedina razlika je u stringu koji se dodeljuje intentu putem konstruktora i čija je vrednost u ovom slučaju "`com.iwedia.tiftestapp.PIP`" i vrednosti dodatka koja je u ovom slučaju "`d_pressed`" jer se uklanjanje/pojavljanje odvija pritiskom tastera "D" na tastaturi. Nakon što je intent napravljen i u njegov objekat smeštene sve potrebne informacije, on se emituje kroz Android.

U TV ulaznom servisu se, po ugledu na mehanizam zamene prozora, registruje primalac emitovanih informacija, BroadcastReceiver, postavljaju se IntentFilteri na odgovarajuću vrednost, i čeka se da intent pristigne u TV ulazni servis. Nakon što prođu sve provere i ako je ustanovljeno da je pritisnut taster "D" poziva se metoda `pipVisibility()` nad objektom `mCurrentSession` klase `TvSession`.

Metoda `pipVisibility()` sluzi samo za komunikaciju sa DtvManager klasom jer TvService ne može direktno da komunicira sa njom. U ovoj metodi dolazi do poziva `changePipVisibility()` nad objektom `mDtvManager` klase `DtvManager`.

Metoda `changePipVisibility()` je metoda u kojoj je smeštena sva logika ove funkcionalnosti. Logika se zasniva na proveri da li je PiP funkcionalnost trenutno aktivna ili ne.

Ako jeste, to automatski znači da korisnik želi da isključi PiP prozor. Polje u kom je smešteno logičko stanje PiP prozora se postavlja na netačno (eng. False) i potom se zaustavlja emitovanje video sadržaja u umanjenom prozoru pomoću DtvManager-a tako što pozovemo metodu `stopService()`. Identifikator resursa koji prosleđujemo treba da odgovara identifikatoru resursa za PiP prozor. Isključivanjem PiP-a je onemogućena i zamena ekrana.

Ako nije, znači da korisnik želi da aktivira PiP prozor. Ovaj slučaj je suprotan od prethodno opisanog. Polje koje predstavlja logičko stanje PiP prozora se stavlja na tačno (eng. True) i potom je potrebno pokrenuti reprodukciju TV sadržaja. To se radi pozivom funkcije `startPip(Channel Descriptor mPipChannel)`, a ostatak poziva je opisan u poglavlju 4.2.1. Nakon prikaza PiP prozora omogućeno je ponovna zamena izmedju glavnog i PiP prozora.

5. Ispitivanje i verifikacija

Prilikom izrade ovog rada obavljena su ispitivanja realizovanih modula i funkcionalnosti. Ručno ispitivanje je vršeno u gore opisanoj TV aplikaciji koja je prethodno implementirana sa osnovnim funkcionalnostima svih TV aplikacija. Drugu vrstu ispitivanja čine nezavisni JUnit ispitni slučajevi, kojima su provereni svi moduli opisani u ovom radu. JUnit ispitni slučajevi ne mogu da daju kompletan dokaz o ispravnosti realizovane funkcionalnosti zbog toga što su to funkcionalni ispitni slučajevi koji ispituju da li je ispravna očekivana funkcionalnost na programskom nivou i ne mogu da proveravaju da li se prikazala slika i slično. Zato su rađeni ručni ispitni slučajevi. Tabela JUnit ispitnih slučajeva data je ispod:

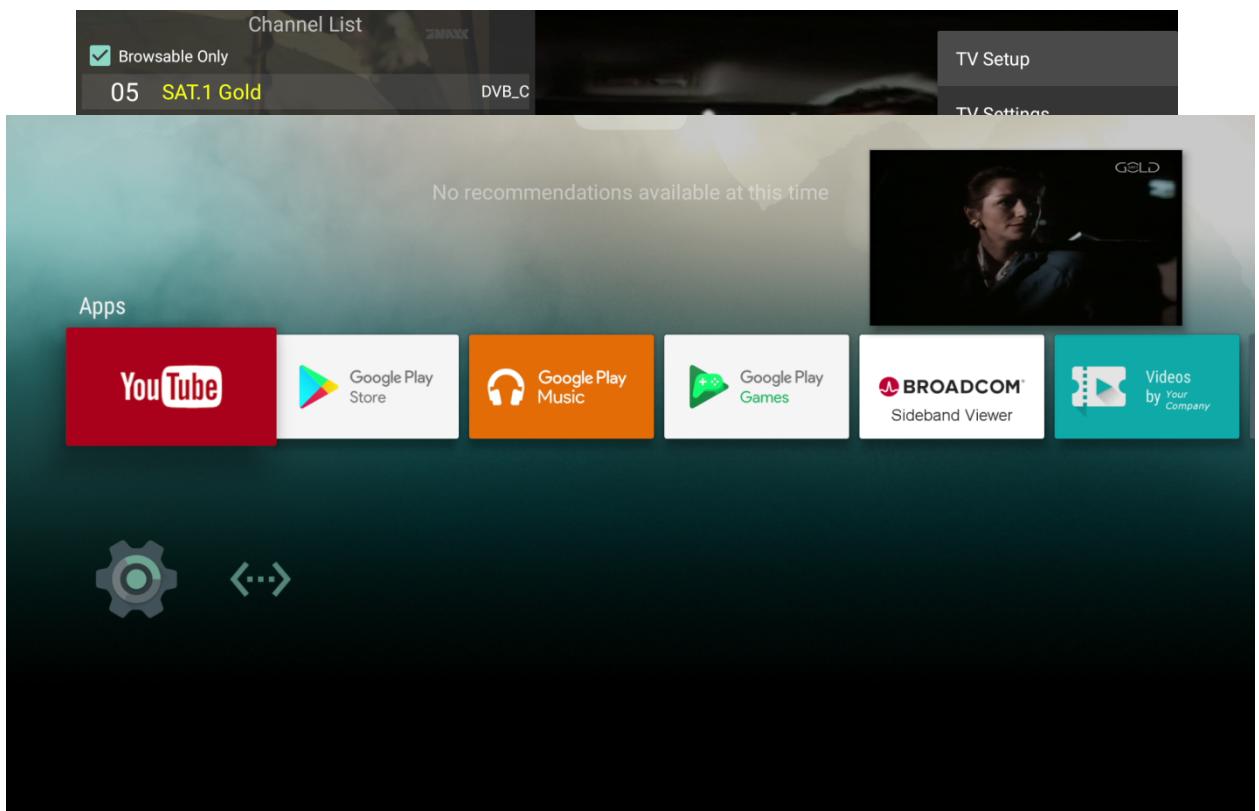
Ispitni slučaj	Rezultati ispitnih slučajeva
Zamena ekrana	Uspešno završen
Uklanjanje PiP prozora	Uspešno završen
Pojavljivanje PiP prozora	Uspešno završen
Promena kanala nakon zamene prozora	Uspešno završen
Promena kanala nakon uklanjanja PiP prozora	Uspešno završen
Paljenje PiP prozora nakon promene kanala	Uspešno završen
Početak rada Android PiP-a	Uspešno završen
Završetak rada Android PiP-a	Uspešno završen

Tabela 5.1 Ispitivanje funkcionalnosti i rezultati

Ispitivanja Googlsovog PiP-a urađeno je za sve moguće ispitne slučajeve zbog kojih je moglo doći do nekonzistentnog stanja aplikacije.

Slučajevi koji su ispitani u procesu ispitivanja su:

1. Ulazak u PiP režim i automatsko uklanjanje svih vidljivih grafičkih komponenti



Slika 5.11 PiP režim rada nakon uklanjanja grafičkih komponenti i klasičnog PiPa

2. Povratak u normalan režim prikazivanja video sadržaja gde se ponovo prikazuje prozor klasičnog PiP-a



Slika 5.12 Povratak u normalan režim rada
Sa prethodnih slika se može potvrditi:

- da korisnička aplikacija ispravno prati događaje na daljinskom upravljaču odnosno tastaturi,
- da se prilikom pokretanja PiP režima rada sve grafičke komponente aplikacije uklanjuju i ostaje jedino video u PiP prozora,
- da program koji se emituje u PiP prozoru odgovara programu koji je bio na glavnom prozoru prilikom poziva PiP funkcionalnosti,
- da je uklonjen prozor klasičnog PiP-a koji se nalazi u gornjem levom uglu ekrana,
- da se prilikom povratka u normalan režim rada u glavnom prozoru prikazuje isti kanal koji je bio odabran pre i za vreme upotrebe Android PiP-a,
- da se nastavlja sa prikazivanjem klasičnog PiP-a, s obzirom da korisnik nije odabrao ukidanje sekundarnog prozora

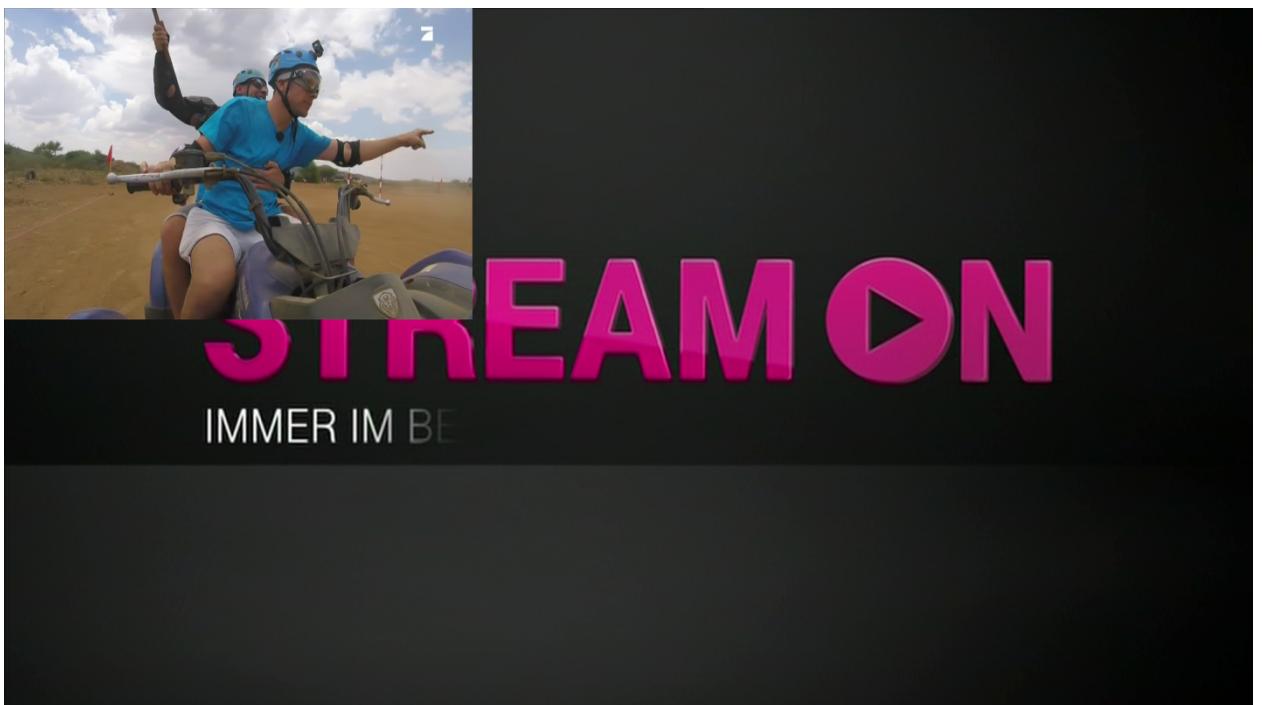
Kao i kod Android PiP-a, i kod klasičnog PiP-a su ispitani svi granični slučajevi na osnovu čega je zaključeno da aplikacija stabilno radi.

Slučajevi koji su ispitani u procesu ispitivanja su:

1. Zamena glavnog i sporednog prozora



Slika 5.13 Izgled ekrana nakon startovanja aplikacije



Slika 5.14 Izgled ekrana nakon urađene zamene

2. Promena kanala nakon odradene zamene



Slika 5.15 Promena kanala nakon prethodne zamene ekrana

3. Ukidanje sporednog prozora



Slika 5.16 Uklanjanje prozora za klasičan PiP

4. Prebacivanje na željeni kanal



Slika 5.17 Prelazak na zeljeni kanal i dalje bez klasičnog PiPa

5. Ponovna aktivacija sporednog prozora



Slika 5.18 Ekran nakon što je klasičan PiP ponovo aktiviran

6. Zaključak

U ovom radu je realizovana programska podrška za Slika-u-slici funkcionalnost, na više različitih načina, na prijemniku digitalnog televizijskog signala zasnovanog na Android Nougat platformi. Prikazana je realizacija sprege između Java programskog jezika i C programskog jezika koja je potrebna zarad realizacije slika-u-slici funkcionalnosti i komunikaciju sa programskom podrškom digitalnog TV prijemnika. Dat je detaljan opis korišćenja slika-u-slici funkcionalnosti. Predloženo rešenje je ostvareno na Broadcom 7252S platformi. Programsко rešenje je ispitano na ciljnoj fizičkoj arhitekturi sa implementiranim programskom podrškom srednjeg sloja televizijskog prijemnika. Detaljnim ispitivanjem je potvrđena funkcionalnost priloženog rešenja.

Neprekidan razvoj Android platforme i posirenje skupa njenih funkcionalnosti pruža mogućnost za dalju nadogradnju i poboljšanje datog rešenja. Realizovan je PiP koji omogućuje krajnjim korisnicima da gledaju TV sadržaj dok koriste ostale funkcionalnosti koje nudi Android. Takođe, realizovana je mogućnost reprodukcije dva TV kanala sa mogućim nadogradnjama poput prikaza sadržaja kanala u umanjenom prozoru dok korisnik prelazi preko liste kanala, prikaza kanala za vreme čitanja elektronskog programskega vodiča, puštanje već snimljenog TV sadržaja pomoću snimača kanala (eng. Personal Video Recording – PVR).

7. Literatura

- [1] J. Bloch: EffectiveJava (2nd Edition), Addison-Wesley, 2008.
- [2] N. Bencomo, F. Losavio, A. Matteo: Java implementations of user-interface frameworks, Technology of Object-Oriented Languages and Systems, 1997
- [3] <https://source.android.com/source/licenses.html>, jun 2017
- [4] U. Reimers: Digital Video Broadcasting, IEEE Communications Magazine, Vol. 36, No. 6 Jun 1998
- [5] M. Robin and M. Poulin: “Digital Television Fundamentals - Design and Installation of Video and Audio Systems,” McGraw-Hill, 1997
- [6] W. Fischer: Digital Video and Audio Broadcasting Technology, 2010.
- [7] <https://source.android.com/>, jun 2017
- [8] <https://www.android.com/gms/>, jun 2017
- [9] <https://source.android.com/devices/tv/>, jun 2017
- [10] <https://source.android.com/compatibility/cts/verifier>, jun 2017
- [11] <https://en.wikipedia.org/wiki/Picture-in-picture>, jun 2017
- [12] [https://en.wikipedia.org/wiki/Multivision_\(television_technology\)](https://en.wikipedia.org/wiki/Multivision_(television_technology)), jun 2017
- [13] M. Vidakovic, N. Teslic, T. Maruna, V. Mihic: Android4TV: a proposition for integration of DTV in Android devices, IEEE 30th International Conference on Consumer Electronics (ICCE), Las Vegas, January 2012
- [14] <http://android4tv.iwedia.com>, jun 2017