Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Inovatívne multimediálne služby

Tím č. 6

Bc. Ivan Barlog

Bc. Jozef Filipek

Bc. Peter Krajči

Bc. Vladimír Kružliak

Bc. Tomáš Škriečka

Ročník: 1.

Predmet: Tímový projekt

Vedúci projektu: Ing. Tomáš Kováčik, PhD.

Študijný program: Počítačové a komunikačné systémy a siete

Ak. rok: 2012/2013

Obsah

Zoznam použitých skratiek

AOSP – Android Open Source Project

API – Application Programming Interface

CEA – Central Electricity Authority

DOM – Document Object Model

DVB – Digital Video Broadcasting

EMU – Emulácie

FTP – File Transfer Protocol

HBB TV – Hybrid Broadcast Broadband Television

HDMI – High-Definition Multimedia Interface

HDTV – High-Definition Television

HTML – HyperText Markup Language

HTTP – Hypertext Transfer Protocol

HTTPS – Hypertext Transfer Protocol Secure

IDE – Integrated Development Enviroment

IdM – Identity Management

LCD – Liquid Crystal Display

LED – Light- Emitting Diode

OIPF – Open IPTV Forum

PVR – Personal Video Recorder

REST – Representational State Transfer

RSS – Really Simple Syndication

RTP – Real-time Transport Protocol

RTSP – Real Time Streaming Protocol

SATA – Serial ATA

SDK – Software Development Kit

TV – Television

USB – Universal Serial Bus

VoD – Video on Demand

W3C – World Wide Web Consortium

WP3 – Work Package

Zoznam použitých obrázkov

1. Úvod

Tento dokument bol vytvorený v rámci predmetu Tímový projekt na tému Inovatívne multimediálne služby. Cieľom je vybrať vhodnú mobilnú platformu pre ktorú navrhneme, implementujeme a otestujeme multimediálnu aplikáciu. Dôraz sa kladie hlavne na inovatívnosť nápadu, vhodnosť platformy, používateľské rozhranie rešpektujúce súčasné trendy a použiteľnosť, praktickosť riešenia, funkčnosť programu overenú priamo na zariadení a možnosti prepojenia s inými platformami (t.j. prepojenie s rovnakou aplikáciou na inej mobilnej platforme ako aj spolupráca aplikácie s hybridnými televíznymi prijímačmi a IPTV. Výstupom bude funkčná aplikácia prijatá do APP store daného výrobcu platformy.

2. Analýza

V časti **Analýza** sa venujeme všetkým technológiám, ktoré je potrebné ovládať na zvládnutie tohto tímového projektu. Základ budú tvoriť technológie Hbb TV a HBB Next. Taktiež analyzujeme set top box AZ box a emulátor Samsung TV.

2.1 HBB TV

HBB Tv je nový priemyselný štandard poskytujúci otvorenú, pre biznis neutrálnu platformu, ktorá kombinuje televízne služby prenášané pomocou broadcastu so službami, ktoré sú prenášané broadbandovo a takisto internetové služby zákazníkom, ktorí používajú spojené televízie a set top boxy. Zakladajúci členovia HbbTv konzorcia spolu s veľkou skupinou podporovateľov spoločne vyvinuli HbbTv špecifikáciu za účelom vytvorenia globálneho štandardu pre hybridné zábavné služby. [2]

Obr. 2.1: Logo Hbb TV [2]

HbbTv špecifikácia je založená na existujúcich štandardoch a webových technológií vrátané OIPF, CEA, DVB a W3C. Štandard poskytuje možnosti a funkcionalitu požadovanú na poskytovanie bohatých broadcastových a internetových služieb. Použitím štandardných internetových technológií

umožňuje rapídny aplikačný vývoj. Definuje minimálne požiadavky, čím zjednodušuje implementáciu v zariadeniach a necháva priestor pre rôznotvárnosť.

Súčasné HbbTv služby:

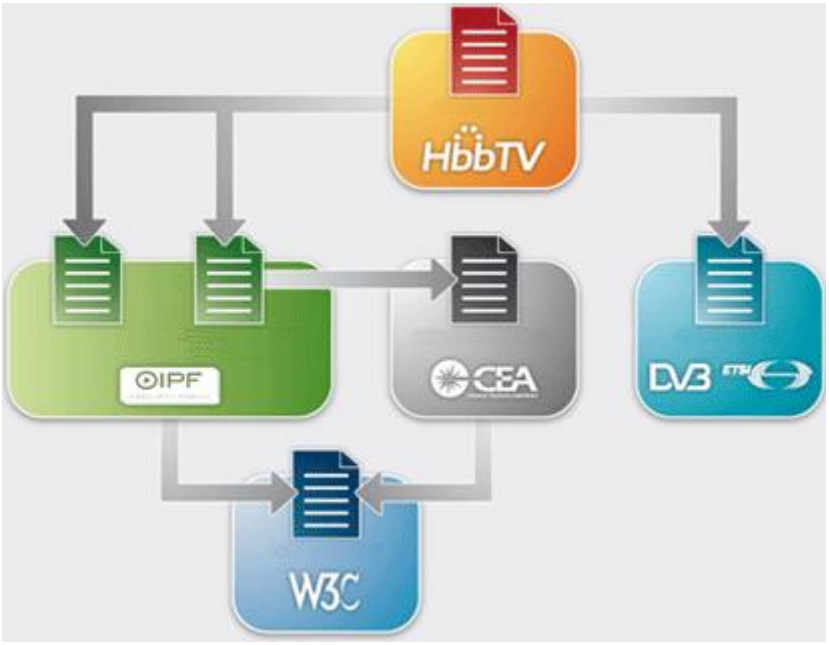
- video textové aplikácie so vstavanou grafikou a videom
- média aplikácie pre VoD
- prototypy informačných služieb (Francúzska televízia)
- pokročilá teletextová služba (Pro7)
- vizuálne rádio

Špecifikácia:

Posledná verzia HbbTv, ktorá bola schválená, je HbbTv 1.5. Verzia 1.5 bola schválená 1. Augusta 2012. Verzia 1.5 podporuje http adaptívne vysielanie (založené na MPEG-DASH), zlepšujúc tak vnímanú kvalitu video prezentácie na pomalom alebo vyťaženom internetovom pripojení. Umožňuje poskytovateľom obsahu chrániť obsah DASH s viacerými DRM technológiami založenými na MPEG CENC špecifikácii. Verzia 1.5 značne zlepšuje prístup k broadcastovému TV programu, umožňujúc operátorom vytvoriť plný, 7-denný televízny program ako HbbTv aplikácie, ktoré môžu byť rozosielané na všetky HbbTv prijímače.

Vznik:

HbbTv špecifikácia bola vyvinutá priemyselnými lídrami, aby sa efektívne zvládlo rýchlo sa zvyšujúce množstvo dostupného obsahu cieleného na dnešného koncového používateľa. Je založená na elementoch existujúcich štandardov a webových technológií vrátane OIPF, CEA-2014 (CE-HTML), W3C (HTML a pod.) a DVB Aplikačnú Signalizačnú Špecifikáciu. Obrázok Obr. 2.2 zobrazuje vzťah medzi HbbTv a existujúcimi štandardmi.



Obr. 2.2: Vzťah medzi HbbTv a podporovanými štandardmi [2]

HbbTv špecifikácia nie je závislá na konkrétnom broadcastovom spojení alebo IP spojení. Je kompatibilné s akýmkoľvek z týchto dvoch spojení, ale získa najviac v prostredí s pripojenými broadband a broadcast spojeniami. Fungovanie HbbTv s týmito spojeniami je zobrazené na obrázku Obr. 2.3: Prepojenie HbbTv s vysielacími službami.

Obr. 2.3: Prepojenie HbbTv s vysielacími službami [2]

2.2 HBB Next

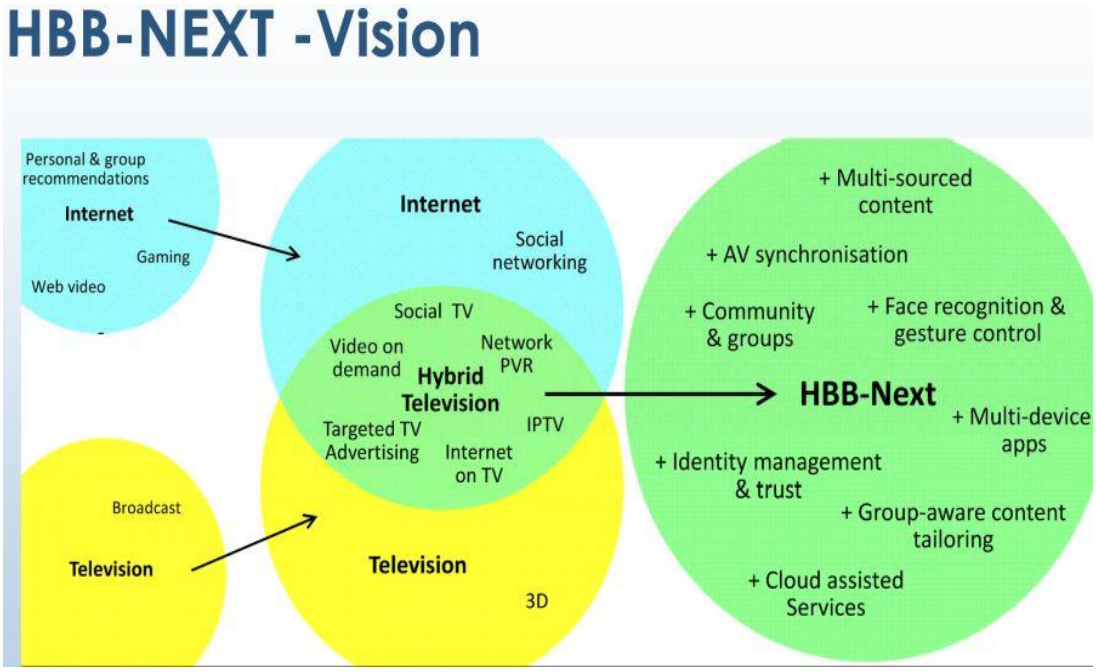
Ako už názov napovedá, HBB-Next má byť ďalšou generáciou vo vývoji štandardu HBB. Technológia spojenia prenosu služieb broadcastom a zároveň spojenie zariadení s internetom je prevratným krokom vo vývoji v tomto odbore. Štandard HBB a teda technológia HBB Tv ešte vo svete nie sú veľmi rozvinuté, a okrem toho ponúkajú ešte množstvo možností ako ich zlepšiť.

HBB-Next je teda nadstavba na už dosiahnuté pokroky v tejto oblasti, ktorá obsahuje veľa aspektov, v ktorých sa má tento štandard zlepšiť a tým sa posunúť do ďalšej úrovne.

Projekt sa usiluje o uľahčenie konvergencie broadcastového vysielania a sveta internetu vynachádzaním technológii, ktoré majú používateľovi obohatiť televízny zážitok sociálnymi sieťami, ovládaním prostredníctvom rôznych zariadení, odporúčania obsahu šitého na mieru, rovnako ako bezproblémové miešanie rozhlasového a televízneho vysielania, doplnkového obsahu na internete a obsahu vytváraného používateľmi.

Projekt má priniesť niekoľko faktorov, aby aplikácie nezávislé od zariadení dokázali organizovať obsah z viacerých zdrojov, podporovať zloženie obsahu v reálnom čase a distribuovať ho používateľom bez ohľadu na ich geografickú polohu. Vyvinutá bude sofistikovaná technológia pre jednoduché a zrozumiteľné použitie obsahu. Neodmysliteľnou súčasťou bude bezpečnosť dát, a ochrana súkromia používateľa, ale takým spôsobom, aby bol pre neho transparentný. [4]

Víziu HBB-Next, ako aj spojenie internetu a televízie a obohatenia štandardu HBB-Next prehľadne ukazuje obrázok Obr. 2.4: Vízia HBB-Next [6].



Obr. 2.4: Vízia HBB-Next [6]

Jadro kompletného štandardu HBB-Next sa skladá z niekoľkých modulov, ktoré zabezpečujú vyššie uvedené funkcionality:

- **Manažér profilov (Profile Manager)** – tento komponent je zodpovedný za riadenie profilov používateľov a služieb. Profily obsahujú služby a k nim príslušné nastavenia.

- **Manažment Identít (Identity Management)** – riadi informácie o používateľoch, zariadeniach a vzťahoch medzi nimi. Zahŕňa priraďovanie používateľov k profilom, autorizáciu, autentifikáciu dát a identitu zariadení.

- **Manažér bezpečnosti (Security Manager) –** umožňuje autorizáciu k aplikáciám a službám. Dovoľuje autentifikáciu používateľov, poskytuje informácie o identite používateľa, pričom spolupracuje s manažmentom identít.

- **Viacmodelová identifikácia (Multi-model Identification) –** umožňuje identifikáciu jedného alebo viacerých používateľov pomocou kamier, mikrofónov a pod.

- **Dôvera a reputácia (Trust & Reputation) –** Zbiera ohodnotenia aplikácii od používateľov, na základe tejto odozvy, vypočíta hodnotenie aplikácie. Toto je potom zobrazované napr. na portáloch s aplikáciami.

- **A/V Synchronizácia (A/V Content Synchronization) –** Synchronizačný modul, ktorý poskytuje funkcionalitu synchronizovať dva prúdy prichádzajúce cez DVB a IP. Tieto prúdy môžu pochádzať od jedného poskytovateľa, ktorý rôzne časti vysiela cez DVB a cez IP alebo niekto iný vysiela dodatočné informácie cez IP k DVB vysielaniu.

- **Systém odporúčaní (Recommendation Engine) –** tento modul umožňuje odporúčanie obsahu používateľom alebo aj ich skupinám. Funguje pre broadcast aj pre broadband.

- **Služba preferencí progilu (Preference Profile Engine) –** súčasť systému odporúčaní, umožňuje ukladanie a zobrazovanie informácii o preferenciách jednotlivých používateľov alebo ich skupinách.

- **Systém personalizácie (Personalization Engine) –** tento komponent je zodpovedný za vytváranie personalizovaných skupinových profilov na základe profilov jednotlivých používateľov. Podľa parametrov z týchto profilov vytvorí skupinové parametre aby vytvoril spoločné prostredie.

- **Notifikačná služba (Notification service) –** modul zameraný na posielanie notifikácii o udalostiach tým používateľom, ktorý si to vyžadujú. Udalosťami sa rozumejú napr. zmena kanálu, osoba vstúpi do miestnosti (opustí ju), a pod.

- **Viacmodálne používateľské rozhranie (Multi-modal User Interface) –** tento komponent má na starosti rozpoznávanie a autentifikáciu používaľov napr. pomocou hlasu alebo rozpoznaním tváre. Zároveň tento modul zabezpečuje ovládanie set-top-boxu pomocou gest alebo hlasu.

- **Repozitáre s EPG metadátami (EPG metadata repository) –** zobrazuje informácie o programe a jeho obsahu pre broadband aj broadcast služby.

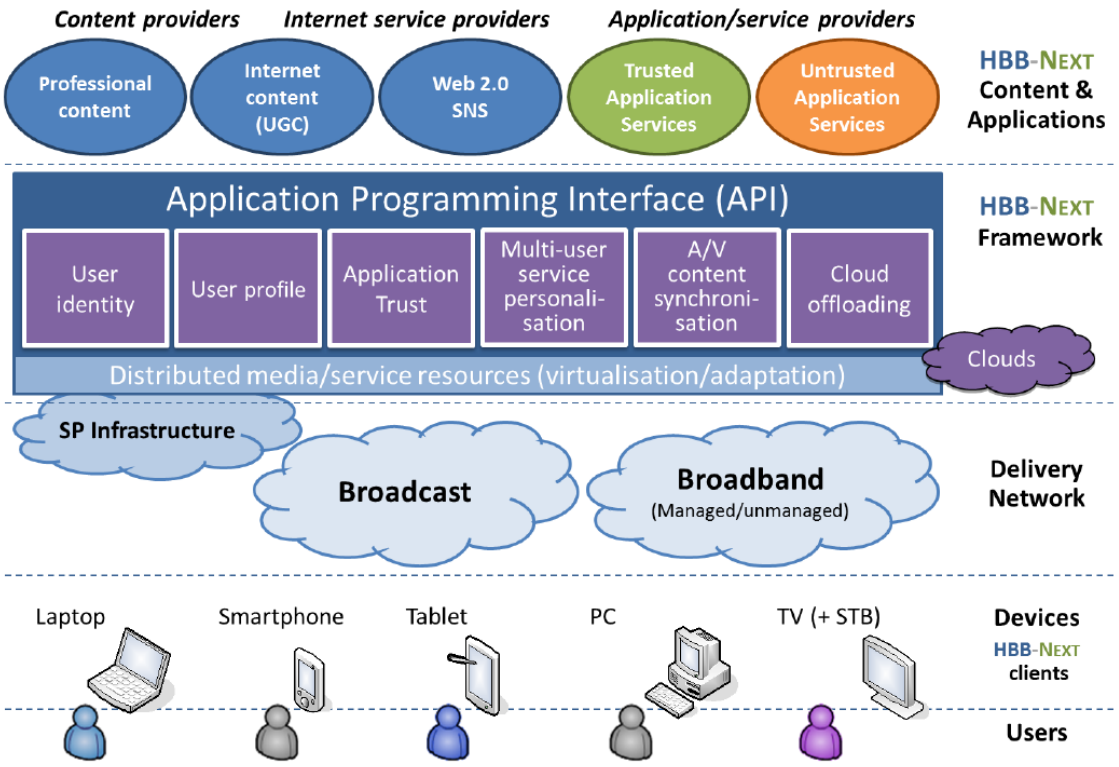
- **Cloud offloading –** poskytuje funkcionality, ktoré nemôžu priamo fungovať na zariadeniach, napr. kvôli slabému výkonu alebo cenovým dôvodom. Tieto funkcionality bežia na sieťovom serveri.

- **Vyhľadávanie služieb (Service Discovery) –** vyhľadávanie dostupných služieb či už na internete alebo z DVB vysielania. [5]

Z celkového uhla pohľadu architektúru HBB-Next tvoria:

- HBB-Next obsah a aplikácie
- HBB-Next rámec – API, ktoré tvoria už spomenuté moduly
- doručovacia sieť – broadcast a broadband
- HBB-Next zariadenia
- používatelia

Túto architektúru zreteľne vidieť na obrázku Obr. 2.5.



Obr. 2.5: Architektúra HBB-Next [5]

2.3 AZ Box Premium HD+

Obr. 2.6: AZBox Premium HD+ [10]

AZ Box Premium HD je kombináciou HDTV satelitného modulového a multimediálneho prijímača na Linuxovej báze a domáceho multimediálneho centra pripojeného k internetu. Disponuje aj SATA rozhraním pre pripojenie interného HDD (2.5“, 3.5“). Externý pevný disk je možné pripojiť cez USB rozhranie, ktoré je v súčasnej dobe štandardným vybavením moderných satelitných prijímačov. Tento model je vybavený DVB-S/S2 tunerom no je možné ho rozšíriť o druhý tuner. Prijímač podporuje obrovské množstvo multimediálnych formátov vrátane HD formátu *mkv*.

Oficiálny firmvér dodávaný v novom prijímači má integrovaný internetový prehliadač, YouTube aplikáciu, FTP, RSS čítačku. Je to však firmvér, do ktorého nemožno zasahovať. Pre naše potreby sme si mohli vybrať medzi rôznymi neoficiálnymi verziami firmvéru (na našom AZBox-e bola nainštalovaná Enigma 2 v 1.7), ktoré poskytujú rozšírené možnosti využitia prijímača a sú poskytované ako „open-source“. Vybrali sme si firmvér Open AZBox PLI mod.

2.3.1 Technická špecifikácia

- operačný systém Linux
- procesor Sigma Designs SMP8634LF MIPS-CPU 300 MHz
- systémová pamäť DDR 128 MB
- video pamäť DDR 128 MB
- tuner DVB-S2, možné rozšírenie o ďalší tuner DVB-S2, DVB-T, DVB-C
- DiSEqC 1.0, 1.2, 1.3 USALS
- Flash 8MB, DOM 256MB
- Disk(y) interný HDD (2,5" a 3,5") a externž HDD cez USB
- Video PAL/ NTSC

- Video Decoder MPEG-1, MPEG-2, MPEG-4.2, MPEG4.10(H.264), WMV9, VC-1, Divx 3/4/5, Xvid
- Audio Decoder Dolby AC3, DTS, MPEG 1 Layer 1/2/3 (MP3), AAC, WMA9, OGG, FLAC
- Rozlišení 576i (480i), 576p (480p), 720p, 1080i, 1080p
- WiFi 802.11b/g miniPCI
- Sieť 10/100 Base-T Ethernet
- 1x čítačka kariet
- 2x CI-Slot pre dekódovacie moduly
- 1 x USB 2.0
- VFD Display
- IR Modul
- Zadný panel 1 x HDMI
- 1 x Component (YCbCr)
- 1 x RCA Video
- 1 x RCA Audio (L/R)
- 1 x RJ45
- 1 x USB 2.0
- 1 x Optický audio výstup S/PDIF
- 1 x SCART
- Sieťový vypínač
- Rozmery 340 x 243 x 66 (mm)
- Napájanie AC 12V 3,4A / 24V 0.8A
- Spotreba max. 60W
- Standby 5W

2.3.2 Funkcie

- nahrávanie PVR
- internetový a YouTube prehliadač
- RSS čítačka
- upgrade firmvéru cez USB a LAN
- sledovanie iného programu počas nahrávania
- pluginy, EMU a rozširenia

2.3.3 Open AZBox PLI MOD HD

Tento firmvér je postavený na bázi OpenPLi 2.1 Enigma 2 opensource a preto disponuje podobnými funkciami ako Enigma 2. Na preinštalovanie firmvéru sme použili softvér AZUp, ktorý umožňuje jednoduchým spôsobom nahrať do prijímača nový firmvér poprípade aj iný kernel.

Medzi pridanú funkcionalitu patrí multimediálny prehrávač AZPlay, vylepšená je taktiež funkčnosť čítačky kariet. Neobsahuje však internetový prehliadač, ktorý v našej práci budeme potrebovať a preto je potrebné ho doinštalovať (plugin, crosscompile).

Tento firmvér neobsahuje taktiež ani žiadne kompilátory (gcc a pod.), a preto sme museli všetky ostatné potrebné programy, ktoré budeme v rámci projektu potrebovať, skompilovať na inom počítači pomocou tzv. cross-kompilácie pre danú architektúru zariadenia. Využili sme Linuxový program crosstool-ng a ako cieľovú architektúru sme použili MIPS. Následne sme potrebné programy skompilovali a cez telnet skopírovali do zariadenia.

2.4 Identity Management Module (IdM) – modul pre správu identít

Modul pre správu identít je jedným z modulov architektúry HBB Next. Tento modul drží informácie o používateľoch, zariadeniach a o ich vzájomných prepojeniach. Tento modul poskytuje API, na základe ktorého je možné komunikovať s ním.

V našom riešení bude komunikovať s týmto modulom serverová časť, ktorá bude vybavovať rôzne požiadavky od rôznych používateľov z rôznych zariadení. Tým pádom v našom riešení musíme byť oboznámení s API tohto modulu.

Pri opise tohto modulu čerpáme z [11]. Nakoľko je samotné IdM momentálne vo vývoji, pri vývoji nášho riešenia budeme dbať ohľad na zmeny v momentálnom API tak, aby sme boli schopní s IdM komunikovať aj v prípade zmien.

2.4.1 Dátový model

Dátový model IdM obsahuje tri hlavné entity – používateľ (user), zariadenie (device) a súvislosť (context). Jednotlivé entity sú podrobne opísané v [D3.2] v časti 4.2 Data model.

Používatelia majú medzi sebou trvalé vzťahy (napr. otec, syn, matka, ...) a taktiež majú trvalé vzťahy so zariadeniami (otec vlastní telefón a podobne). Súvislosti v tomto prípade opisujú

krátkodobé (ale aj dlhodobé) vzťahy medzi používateľmi a zariadeniami. Tieto vzťahy môžu mať tzv. váhu, resp. roly, ktoré približujú typ vzťahu. Používatelia môžu spravovať iných používateľov (napr. vlastník zariadenia môže vytvoriť ďalšieho používateľa, ktorý môže toto zariadenie používať) alebo zariadenia (napr. vlastník zariadenia môže toto zariadenie konfigurovať).

Vzťahy medzi jednotlivými používateľmi a používateľmi a zariadeniami vyjadrujú tzv. súvislosti. Tieto súvislosti hrajú hlavnú úlohu v implementácii IdM. Súvislosti sa používajú na mapovanie zvyčajného a aktuálneho kontextu používateľa.

Zvyčajný kontext (súvislosť) reprezentuje súvislosti, v ktorých zvykne používateľ byť. Aktívny kontext (tiež aktuálny) hovorí o momentálnom kontexte používateľa, ktorý je aktívny v momentálnom čase a používa sa pre interakciu. Kontext (tiež súvislosť) ako taký sa používa pre statické zoskupenie používateľov a zariadenia.

Vzťahy jednotlivých entít

Používateľ môže byť prepojený s ostatnými používateľmi.

Používateľ môže mať mnoho zariadení. Zariadenie musí mať jedného alebo viacerých vlastníkov – vzťahom je váha resp. rola.

Používateľ alebo zariadenie môže byť v reálnom čase aktívny iba v jednom kontexte. V jednom aktívnom kontexte môže byť pripojených viacero používateľov alebo zariadení, pričom váhy ich vzťahov sa odvádzajú od používateľa (riadiaci používateľ z angl. controlling user) alebo zariadenia (riadiace zariadenie z angl. controlling device), ktoré riadia kontext.

Používateľ alebo zariadenie môže byť súčasťou viacerých zvyčajných kontextov. Zvyčajný kontext môže zahŕňať používateľov alebo zariadení.

Používateľ môže mať len jeden profil. Profil sa vždy spája s konkrétnym používateľom. Ku profilom môžu byť priradené rôzne služby.

2.4.2 IdM API

Pre poskytovanie informácií pre ostatné moduly používa IdM modul REST API.

REST rozhranie by malo poskytnúť tieto funkcie:

- získať, pridať, upraviť a vymazať používateľa zo systému
- pridať, upraviť a vymazať používateľa z kontextu
- získať aktívnych používateľov daného kontextu (v reálnom čase)
- získať používateľov súvisiacich s daným kontextom
- získať aktívne zariadenia daného kontextu (v reálnom čase)
- získať zariadenia súvisiace s daným kontextov
- vyhovieť požiadavkám riadenia bezpečnosti
- získať, pridať, upraviť a vymazať zariadenie zo systému
- pridať, upraviť a vymazať zariadenie z kontextu

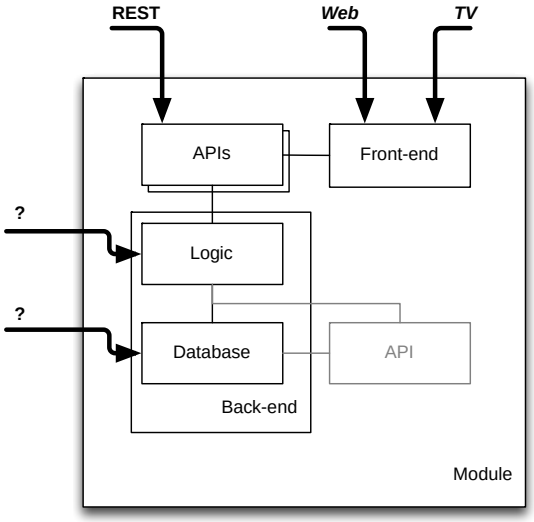
- pripojiť a odpojiť používateľa od zariadenia
- získať zoznam používateľových zariadení
- získať, pridať, upraviť a vymazať roly používateľa
- získať, pridať, upraviť a vymazať relácie medzi používateľmi

IdM pre transport správ používa vynútene protokol HTTPS, pričom pre ladenie sa môže použiť aj HTTP protokol.

API obsahuje 4 základné metódy, ktoré možno použiť nad objektom (v tomto prípade je objekt používateľ, zariadenie alebo kontext v systéme):

- GET – pre získanie objektu
- POST – pre vytvorenie objektu
- PUT – pre upravenie objektu
- DELETE – pre vymazanie objektu

Na Obr. 2.7 je znázornená vnútorná štruktúra architektúry IdM. Nás bude zaujímať hlavne API a Front-end.



Obr. 2.7: Vnútorná štruktúra architektúry modulu IdM

Ukážku IdM možno nájsť na adrese http://hbbnext.ngidm.org, pričom si je možné vyskúšať cez webové rozhranie, ako IdM funguje.

2.5 Samsung Smart TV 5300

Samsung Smart TV 5300 je multimediálna televízia s podporou Smart TV aplikácií. Táto televízia disponuje veľmi dobrou konektivitou a je plná moderných technológií. Veľmi dobre zobrazuje svetlé scény, avšak naopak tmavé scény jej robia mierne problémy. Nepodporuje technológiu 3D, avšak tú by sme ani nevyužili.

Obrazovka o rozlíšení 1920x1080 pixlov na báze LED, má veľmi dobré zobrazovacie uhly, živé farby a zároveň má nízku spotrebu. Obnovovacia frekvencia na úrovni 100Hz sa postará o to aby bol obraz vždy čistý a ostrý.

Obr. 2.8: Samsung Smart TV 5300 [13]

Z hľadiska Smart TV obsahuje veľké množstvo aplikácií ako sú napríklad: Skype, Facebook, Twitter, Youtube prehrávač a mnohé iné. Tieto sú dostupné spoločne ostatnými službami pomocou Smart Hub.

Technická špecifikácia:

- typ zobrazovacej jednotky LCD /w LED
- veľkosť displeja 40“
- maximálne rozlíšenie 1920 x 1080
- digitálny tuner
- obnovovacia frekvencia 100Hz
- rozmery so stojanom (Š x V x H) 972,6 x 606,5 x 247,8mm
- váha so stojanom 10,9kg
- konektivita: HDMI 3x (v1.3), USB 2x, výstup na slúchadlá, kompozitný vstup, vstup na anténu, optický výstup digitálneho zvuku, Ethernet, scart,

Pre náš projekt bude táto televízia slúžiť len ako zobrazovacie zariadenie, avšak mali sme pripravenú aj alternatívu, keby sa nám nepodarilo nahrať alternatívny firmvér do set-top boxu a vtedy by sme všetku funkcionalitu presunuli na televíziu. Z tohto dôvodu sme preskúmali aj možnosti akým spôsobom sa dá získať kompletný prístup do televízie.

Televízie Samsung majú operačný systém, ktorý je založený na Linuxe. To znamená, že pokiaľ získame plný prístup do operačného systému televízie vieme tam dostať v podstate akýkoľvek náš program. Plný prístup sa dá získať a následne vieme spravovať našu televíziu tromi spôsobmi.

Prvý najznámejší spôsob je pomocou USB. Tento typ získania plných práv sa zakladá na tzv. hotel móde, pri ktorom sa pomocou tohto módu nahrávajú nastavenia do televízie. Takýmto spôsobom vieme spustiť SamyGo [14] aplikáciu, ktorá nám sprístupní plné práva a taktiež vieme povoliť vzdialený prístup pomocou telnetu alebo ssh.

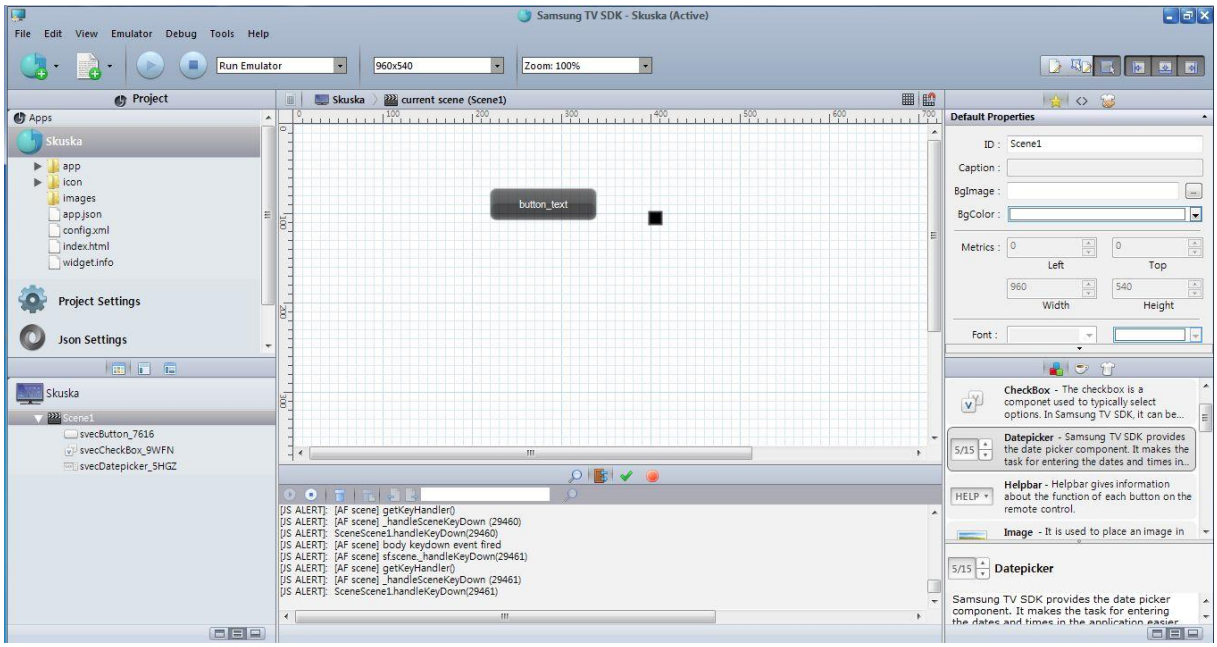
Ďalší spôsob ako získať plné práva je pomocou aktualizácie firmvéru. Nahratím upraveného firmvéru priamo zo stránok SamyGO a tak môžeme začať pracovať na televízii bez obmedzení. Tento spôsob je obmedzený len na niektoré modely.

Pri starších televíziách sa dá získať plný prístup pomocou portu Ex-link. Tento port slúži na servisné zásahy a preto je práca s ním nebezpečná pre neskúseného užívateľa. Tento port vyzerá na strane televízie ako klasický stereofónny kolík používaný na slúchadlách. Na druhej strane je to sériový port so špeciálnym zapojením vývodov. Ex-link sa používa aj na obnovenie chodu televízie pokiaľ sa dostane po neodbornom zásahu do stavu, z ktorého sa nevie spamätať ani reštartovaním.

2.6 Vývoj aplikácií Samsung Smart TV

Keďže nemáme žiadne predošlé skúsenosti s vývojom aplikácií pre platformu Samsumg Smart TV, príde nám vhod veľká podpora zo strany výrobcu tejto značky. Na stránkach [12] môžeme nájsť veľké množstvo návodov pre úplných začiatočníkov, ako aj pre pokročilých vývojárov. Okrem návodov sa tam nachádza aj rozsiahle medzinárodné fórum, kde sa môže každý obrátiť so svojim problémom.

Pokiaľ chceme vyvíjať aplikáciu pre Samsung Smart TV máme na výber dve vývojové prostredia. Prvé je dobre známe Eclipse IDE a druhé je Samsung IDE. Na nasledujúcom obrázku môžeme vidieť vývojové prostredie Samsung IDE.



Obr. 2.9: Vývojové prostredie Samsung IDE

Po spustení Samsung IDE vytvoríme nový projekt. Na výber máme základný projekt s využitím HTML a CSS, projekt v jazyku JavaScript a projekt v jazyku Flash. Vytvorením projektu sa nám zobrazí štandardné rozloženie prvkov známe z iných vývojových prostredí, takže nepotrebujeme veľké množstvo času na prispôsobovanie sa. Následne môžeme začať do projektu pridávať grafické prvky jednoducho pomocou chytenia a presunutia na potrebné miesto, tak isto môžeme tieto prvky upravovať buď v tabuľkovom nastavení na pravej strane obrazovky alebo priamo úpravami v kóde aplikácie.

Na samotný vývoj aplikácie nám nestačí len vývojové prostredie, ale potrebujeme aj prostredie v, ktorom otestujeme nami vytvorenú aplikáciu. Pre tieto účely nám postačí vstavaný emulátor pre obidve vývojové prostredia. V emulátore máme zobrazenú obrazovku televízie ako aj ovládač, takže vieme jednoducho otestovať každú aplikáciu. Tento emulátor môžeme vidieť na nasledujúcom obrázku.

Obr. 2.10: Emulátor Samsung Smart TV

Najnovšia stabilná verzia Samsung SDK pre rok 2012 je verzia 3.5.2. Táto verzia podporuje najnovšie technológie, ktoré sa používajú vo webových prehliadačoch ako sú napríklad: HTML 5, DOM 3, CSS3, JavaScript, Flash 10.1, HTTP, HTTPS, RTP/RTSP a množstvo iných. Okrem štandardnej verzie SDK 3.5.2, je dostupná aj skúšobná verzia 4.0.

2.7 Platforma Android

Android je operačný systém založený na báze Linuxu navrhnutý prevažne pre mobilné zariadenia s dotykovými displejmi ako sú smartfóny a tablety. V súčasnosti je Android vyvíjaný spoločnosťou Google v spojení s Open Headset Alliance. Google zverejnil zdrojové kódy Androidu pod licenciou Apache. AOSP, vedený spoločnosťou Google má na starosti údržbu a vývoj Androidu.

Android pozostáva z jadra založenom na jadre Linuxu vo verzii 2.6 a 3.x (Android 4.0 a vyšší) s middleware softvérom, knižnicami a API napísanými v jazyku C. Aplikačný softvér beží na aplikačnom frameworku, ktorý obsahuje knižnice kompatibilné s jazykom Java. Hlavná hardvérová platforma pre Android je ARM architektúra, ktorá podporuje Android x86.

Aplikácie do zariadení s OS Android sú voľne prístupné cez Google Play. V súčasnosti sa na Google Play nachádza viac ako 600 000 aplikácií, vrátane bezplatných a platených aplikácií. Ako potvrdenie o vytvorení korektnej aplikácii, bude naša finálna android aplikácia umiestnená na Google Play.

3 Špecifikácia požiadaviek

Témou nášho tímového projektu je vytvorenie základnej architektúry pre štandard HBB Next vrátane vytvorenia mobilnej aplikácie, ktorá by zvyšovala funkcionalitu našej architektúry. Mobilná aplikácia bude navrhnutá pre platformu Android.

Základom našej architektúry je server, ktorý bude komunikovať s klientmi (t.j. set top box, mobilná aplikácia) a so systémom Identity Manager, ktorý slúži na overovanie používatľov.

Funkcie servera:

- komunikácia s AZ BOX
- komunikácia s klientmi
- komunikácia s IdM
- spracovanie požiadaviek od klientov

Funkcie mobilnej aplikácie:

- textová komunikácia s ostatnými klientmi
- zobrazovanie zoznamu používateľov

4 Návrh riešenia

V tejto časti sa nachádza náš predbežný návrh architektúry. Obsahuje popis hlavných komponentov a komunikácie medzi nimi.

4.1 Dostupné vybavenie

Na realizáciu nášho návrhu a konečnej implementácie bude využité nasledujúce vybavenie:

Televízia: Samsung Smart TV Class 5300, Series 5
Blu-Ray prehrávač: SAMSUNG BD-E5500
Set Top Box: AZBOX Premium HD+ (firmware enigma2 v1.7)
Vysielacia karta: Dectec DTA 110

4.2 Návrh architektúry

4.2.1 Serverová časť

Serverová časť bude slúžiť na komunikáciu s ostatnými časťami našej architektúry, resp. architektúry HBB Next. Server bude spracúvať a vybavovať požiadavky zariadení (mobilné zariadenia, set top boxy, inteligentné televízory) a posúvať ich ďalej ostatným modulom. Komunikácia medzi koncovými zariadeniami a jednotlivými modulmi bude vždy smerovať práve cez túto serverovú časť a bude samozrejme obojstranná.

Implementácia serverovej časti

Pri riešení použijeme ako server linuxový stroj. Riešenie bude implementované v jazykoch C/C++ za pomoci štandardných knižníc a knižníc Boost – poprípade, ak to bude potrebné – aj iné knižnice.

Serverová časť bude navrhnutá tak, aby korešpondovala s návrhom modulov HBB Next – v praxi to znamená, že bude navrhnutá tak, aby vedela jednoducho komunikovať, s ktorýmkoľvek modulom HBB Next architektúry.

Komunikácia bude prebiehať na vrstve IP v podobe HTTP požiadaviek. Na serverovej časti bude implementovaná RESTful aplikácia, ktorá bude vedieť komunikovať aj s ostatnými modulmi ako je napríklad Identity Management Module (IdM) – modul pre správu identít a podobne.

4.2.2 Klientská časť

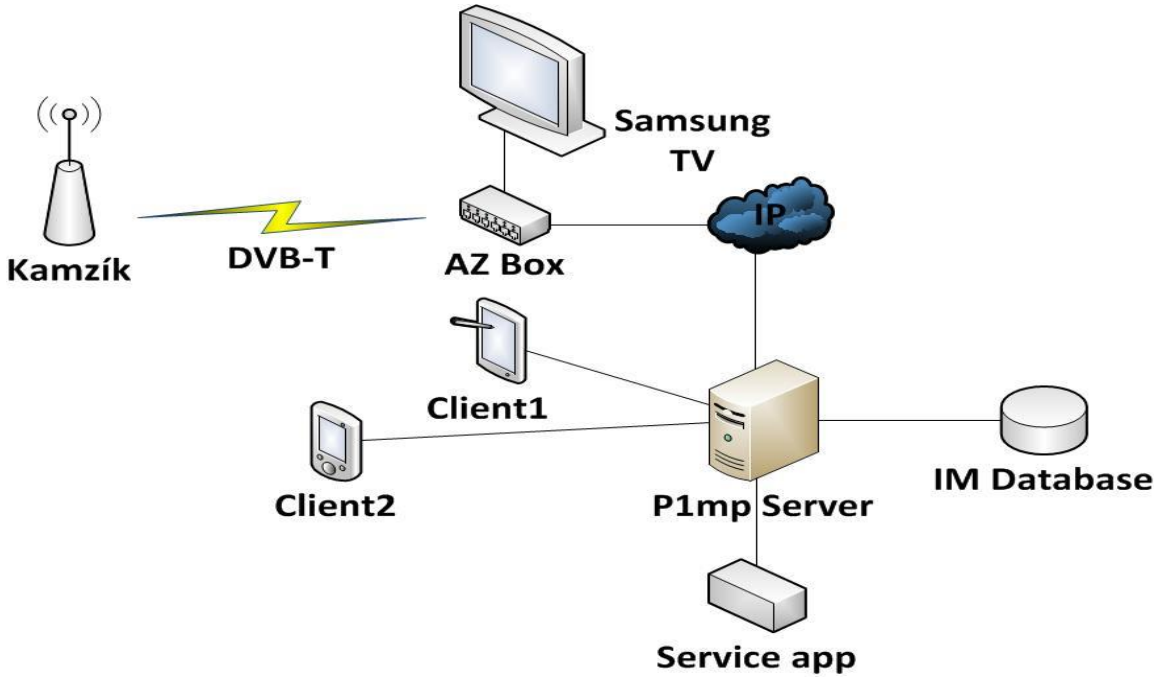
Medzi klientmi budú použití televízia Samsung a aplikácia na platforme Android. Obidvaja klienti budú prepojení so serverom a pomocou komunikačnej služby na serveri budú medzi sebou komunikovať.

Prvotný návrh architektúry je zobrazený na obrázku Obr. 4.1. Zobrazuje komunikáciu medzi jednotlivými členmi architektúry. Klienti komunikujú so serverom, ktorý je spojený s IdM databázou a kontroluje korektné využívanie služieb klientmi.

AzBox návrh

Úlohou set-top-boxu v našej architektúre je okrem štandardného prijímania videa a zvuku cez DVB vysielanie aj prijímanie HBB-TV aplikácii prostredníctvom tohto vysielania. Vysielanie týchto aplikácii je už implementované a funkčné. Potrebné teda je spojazdniť prijímanie HBB-TV aplikácii, a to takým spôsobom, že pri demultiplexácii prijímaného toku, sa okrem video a audio stopy z neho rozbalí aj spomínaná aplikácia do samostatného adresára.

Pre spustenie prijatej aplikácie je potrebné, aby na set-top-boxe bol nainštalovaný webový prehliadač, ktorý je schopný spúšťať HBB-TV aplikácie.



Obr. 4.1: Návrh architektúry

5 Prototyp

Táto časť opisuje prototyp tímového projektu tímu číslo 6. Náš prototyp sa skladá z niekoľkých separátnych častí. Na každej časti pracujú rôzni členovia tímu, ale vo finálnej verzii budú takmer všetky tieto časti vytvárať jeden celok – architektúru – a umožňovať komunikáciu medzi televízorom Samsung Smart TV a mobilným klientom.

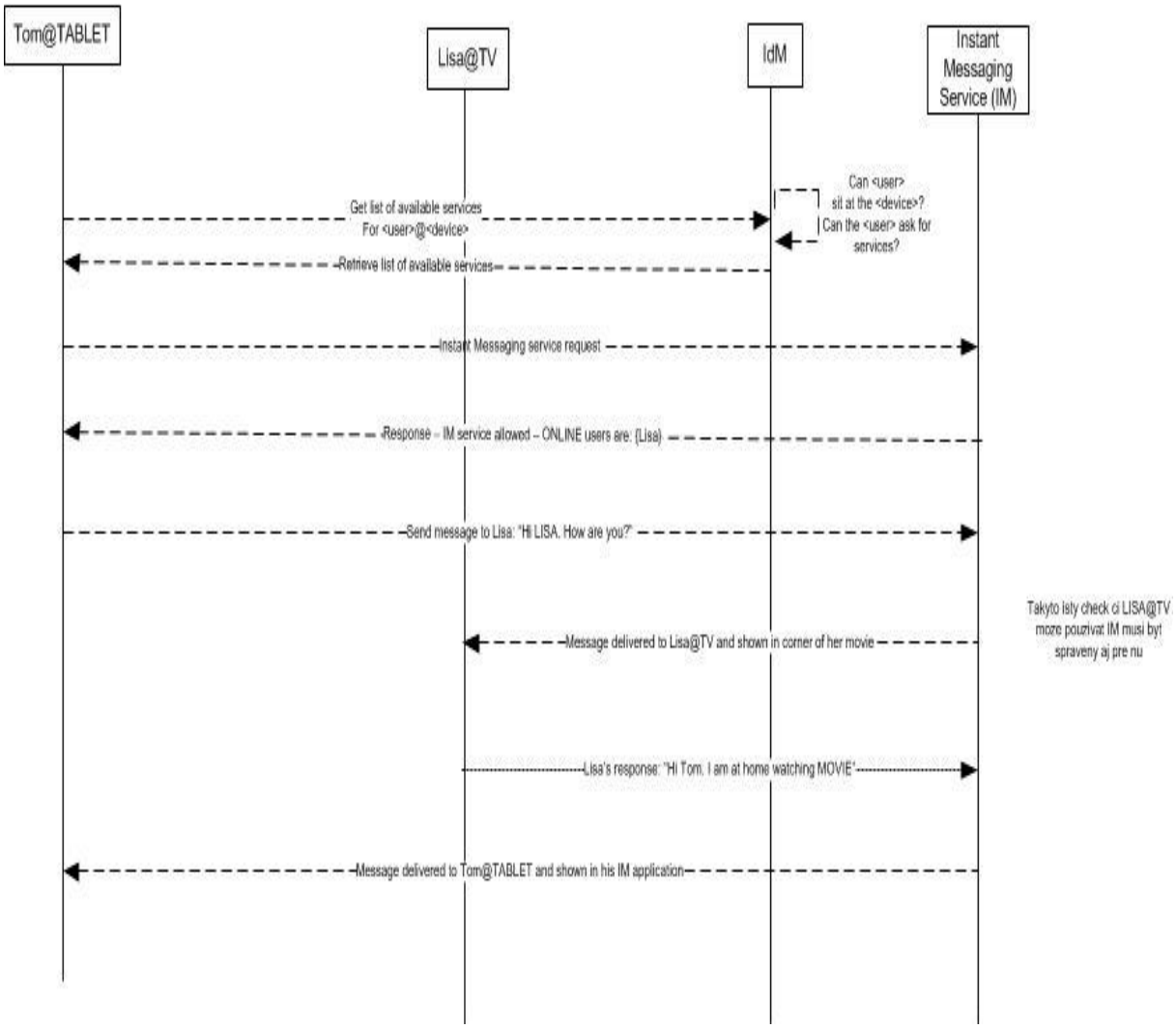
Ako bolo spomenuté vyššie, prototyp sa skladá z niekoľkých častí a v tejto kapitole sú tie časti popísané spolu aj s menami členov tímu, ktorí sú zodpovední za ich vývoj. Naša architektúra sa skladá z nasledujúcich častí:

- Mobilná aplikácia (Android)
- Služba Instant Messaging (IM)
- Televízna aplikácia (Samsung Smart TV)
- Vysielacia/Prijímacia časť (AZ Box)

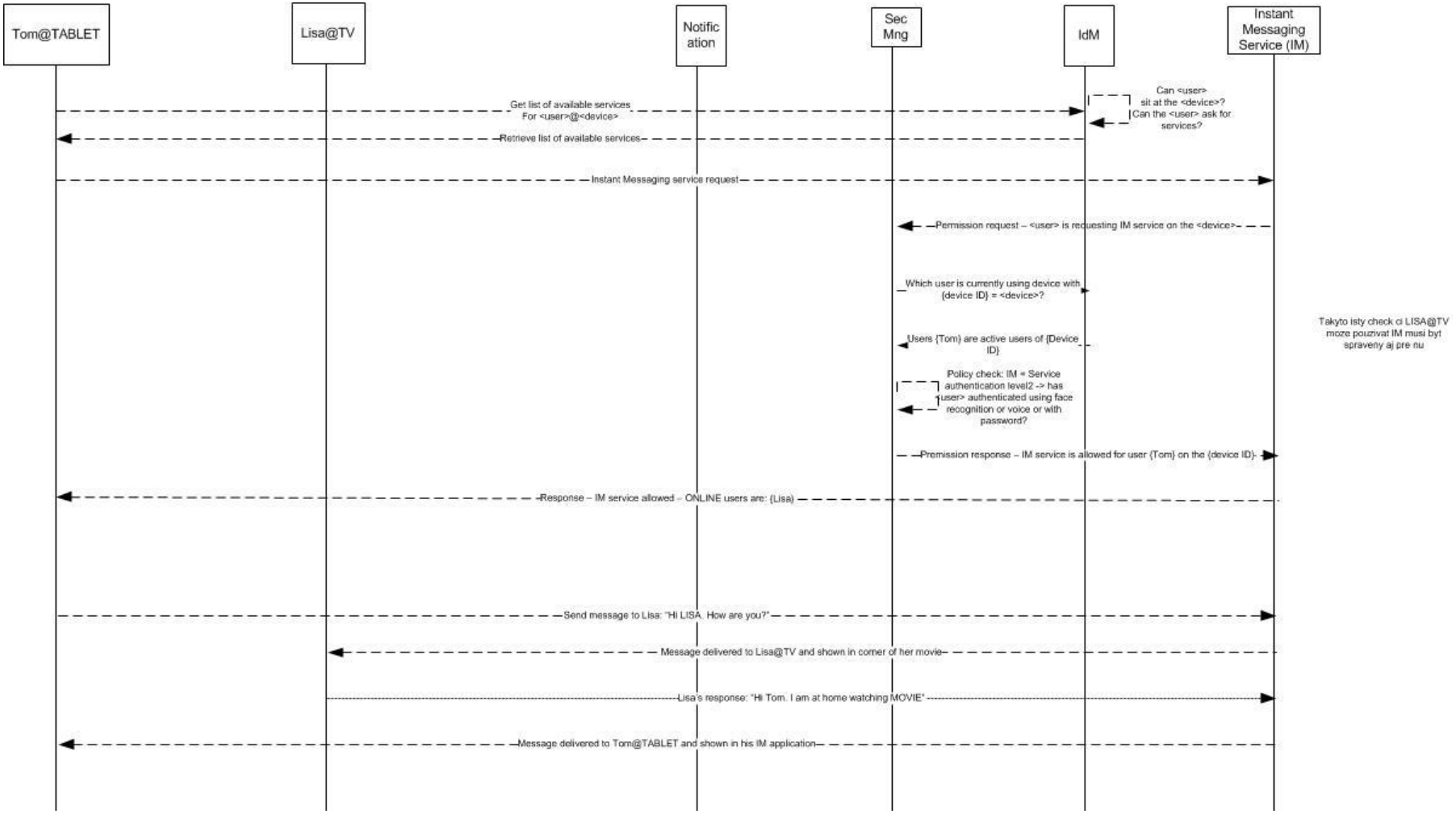
5.1 Architektúra prototypu

Na obrázku Obr. 5.1 je zobrazená architektúra nášho prototypu. Je to zjednodušená verzia finálnej architektúry, ktorú nám poskytol vedúci tímového projektu. Bol z nej odstránený prvok Security Manager, pretože v čase písania prototypu ešte nebol hotový a nemáme k nemu žiaden prístup. Za cieľ sme si dali spojazdniť komunikáciu medzi mobilnou aplikáciou a IdM a medzi serverom a mobilnou aplikáciou. V nasledujúcich častiach budú bližšie popísané čo sa nám podarilo spojazdniť, ako sme to dokázali a čo plánujeme doimplementovať.

Na obrázku Obr. 5.2 je znázornená kompletná architektúra, ktorá nám bola pridelená.



Obr. 5.1: Architektúra prototypu



Obr. 5.2: Kompletná architektúra

5.2 Mobilná aplikácia (Android)

V tejto časti predstavím, čo je doteraz hotové v mobilnej instant messaging aplikácii. Na obrázku Obr. 5.1 je mobilná aplikácia popísaná slovami *Tom@TABLET,* aj keď v skutočnosti sa aplikácia vyvíja pre smartfón.

V rámci architektúry priamo komunikujem s časťami IdM (Identity Manager) a IM (Instant Messaging). Časť IdM má na starosti Sebastian Schumann (Telekom) a časť IM má na starosti Ivan Barlog (člen nášho tímu).

5.2.1 Implementácia mobilnej aplikácie

Ako bolo spomenuté vyššie, implementáciu sme sa rozhodli implementovať na platforme Android a pre testovacie účely sme použili smartfón HTC One V.

Prototyp aplikácie komunikuje s časťami IdM a IM. Samozrejme táto komunikácia ešte nie je úplne kompletná, ale základ je hotový. Pôvodným plánom bolo vytvoriť pomocou aplikácie používateľský účet v IdM aby sa bolo možné pod týmto účtom prihlásiť. Bohužiaľ, poskytnuté IdM nedovoľuje registráciu používateľa mimo webového rozhrania a na modifikáciu dát pomocou mobilnej aplikácie, t.j. získanie informácií o používateľovi alebo zmazanie používateľa (to je všetko čo mi IdM dovolí), je potrebné použiť prístupový token, ktorý je vygenerovaný pre používateľov webového rozhrania IdM, nie pre používateľov o ktorých získavam informácie z IdM.

Jediné čo treba zadať do kódu aplikácie, je táto adresa:
http://hbbnext.ngidm.org/api/v2/users?access_token=95e20b2b06c2de7d5fc4452c85b032af

To mi dovoľuje získať informácie od používateľov a mazať používateľov na základe ich ID. Komunikácia s IdM funguje na základe HTTP požiadaviek (z angl. request). HTTP GET request slúži na získanie informácií o používateľoch a HTTP DELETE slúži na vymazanie informácii o používateľoch. IdM mi poskytuje informácie o používateľoch v JSON správach z ktorých si vyberiem údaje, ktoré ma zaujímajú, resp. potrebujem.

5.2.2 Rozhranie mobilnej aplikácie

Aplikácia v štádiu prototypu vykonávať tieto funkcie:

- Získanie informácií o používateľovi/používateľoch z IdM
- Vymazať používateľa z IdM na základe ID
- Prihlásiť/odhlásiť používateľa z IM
- Počúvať na správy od IM

Komunikácia s IdM je vykonávaná na základe HTTP requestov. Pre získanie informácií používam GET request a pre vymazanie DELETE request. Obidva requesty sa realizujú pomocou ID používateľa v systéme.

GET:

http://hbbnext.ngidm.org/api/v2/users/2?access_token=95e20b2b06c2de7d5fc4452c85b032a

DELETE:

http://hbbnext.ngidm.org/api/v2/users/2?access_token=95e20b2b06c2de7d5fc4452c85b032a

Príkaz GET získa informácie o používateľovi s ID 2 a request DELETE vymaže používateľa s ID2.

Komunikácia s IM je vykonávaná pomocou HTTP requestov a otvorených socketov. HTTP komunikácia slúži na prihlásenie a odhlásenie používateľa.

Prihlásenie:

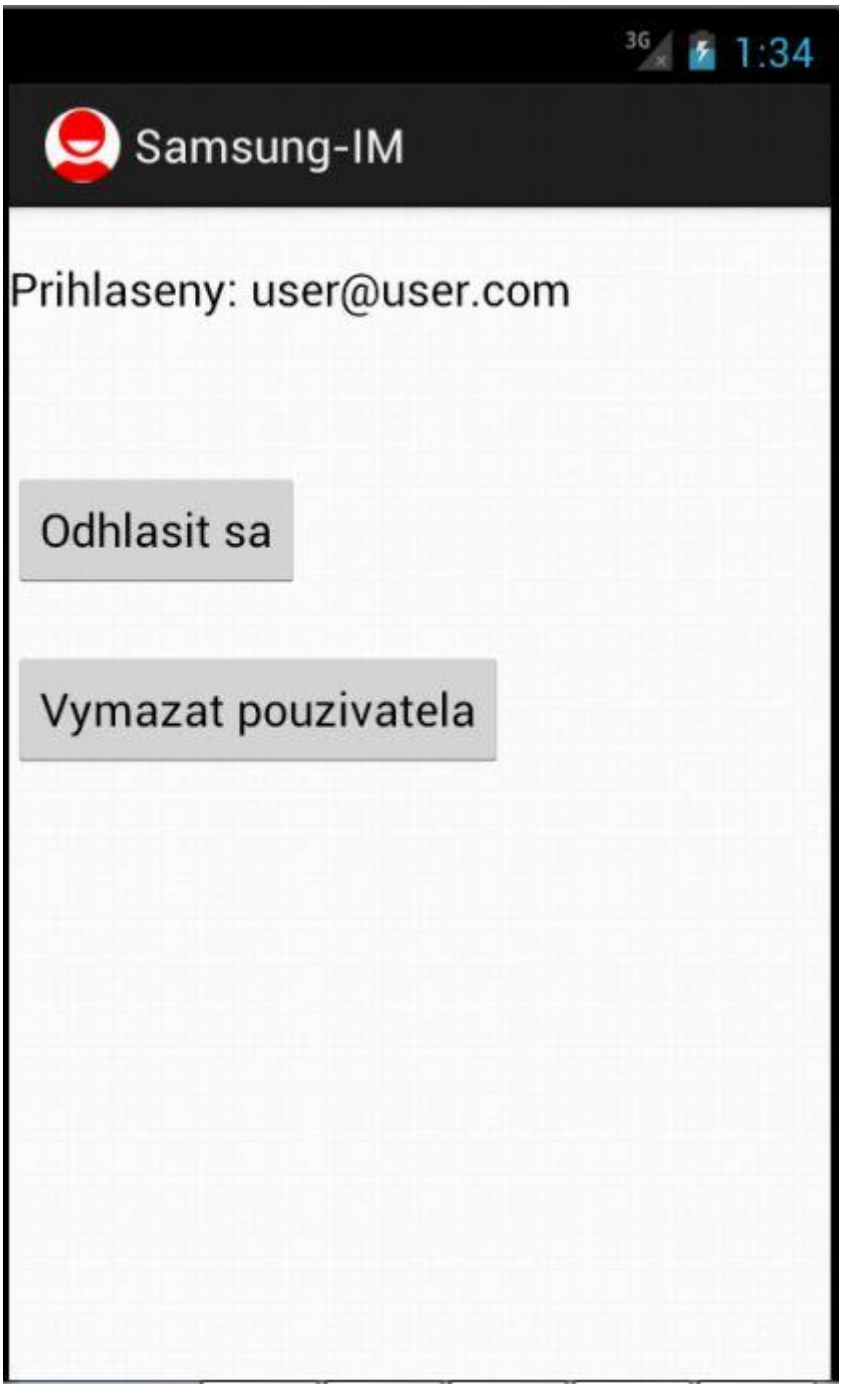
http://147.175.158.173:8080/SrvcM/sign?in=user@user.com

Odhlásenie:

http://147.175.158.173:8080/SrvcM/sign?out=user@user.com

Ako druhý prístup mám otvorený socket, ktorý počúva na konkrétnom porte a IP adrese kvôli prichádzajúcim správam od IMS.

Na obrázku Obr. 5.3 je znázornené používateľské grafické prostredie mobilnej aplikácie. Aplikácia má názov Samsung-IM a na demonštrovanom obrázku je vidieť aký používateľ je prihlásený, odhlásiť používateľa a vymazať používateľa.



Obr. 5.3: GUI mobilnej aplikácie

5.3 Služba Instant Messaging (IM)

Obrázok Obr. 5.1 znázorňuje architektúru, ktorá bola použitá pri implementovaní nášho prototypu. Za úlohu sme dostali implementovať službu Instant Messaging (z angl. rýchle správy, resp. čet) medzi zariadeniami s platformou Android.

V rámci architektúry nám bol poskytnutý prístup k modulu Identity Management (IdM), ktorý vyvíja Sebastian Schumann. Rovnako nám bol prisľúbený prístup k modulu Security Management (SecMng), ale bohužiaľ v čase implementácie prototypu sme k nemu prístup nemali. Tento modul by mal podľa všetkého potvrdzovať ostatným modulom či používateľ má alebo nemá prístup k jednotlivým službám – túto funkčnosť sme v rámci prototypu nijako špeciálne neriešili, ale berieme ju do úvahy a budeme ju vedieť zahrnúť v konečnej implementácii.

Z priloženej architektúry (Obr. 5.1) sme dostali za úlohu vytvoriť serverovú časť, ktorá bude implementovať službu Instant Messaging (implementuje Ivan Barlog – časť 5.3.1) a klientské časti pre platformu Android (implementuje Jozef Filipek – časť 5.2) a platformu Samsung Smart TV (implementuje Peter Krajči – časť 5.4).

5.3.1 Implementácia služby Instant Messaging

Implementáciu tejto služby sme sa rozhodli založiť na aplikačnom serveri Apache Tomcat, ktorý beží na linuxovom stroji. Serverovú časť obsluhujú Java Servlety – klient zadáva na server požiadavky a server na ne odpovedá.

Prototyp sme sa rozhodli okresať (aj kvôli nekompletnej architektúre) na obsluhu troch požiadaviek – prihlásenie sa, odhlásenie sa a odosielanie správ medzi klientami. Server si v pamäti drží UID (user ID) a IP adresu každého prihláseného klienta. Pri odhlásení klienta sa jemu prislúchajúci záznam zo servera vymaže. Pri každom prihlásení alebo odhlásení server rozosiela všetkým prihláseným používateľom pomocou socketovej komunikácie aktualizovanú tabuľku používateľov.

Komunikácia medzi klientami je založená na kombinácii použitia Java Servletu a socketovej komunikácie. Odosielateľ zadá na server požiadavku, v ktorej ako prvý argument priloží UID klienta, pre ktorého je správa určená. Ako druhý argument priloží JSON správu s vlastným UID a samotnou správou pre prijímateľa. Server z tejto požiadavky vyberie UID prijímateľa, vyhľadá ho v tabuľke pripojených používateľov a zvyšnú JSON správu mu odošle pomocou socketovej komunikácie. Výsledné rozhranie pre klienta je bližšie opísané v časti Rozhranie služby Instant Messaging.

V rámci výslednej implementácie máme za cieľ okrem služby Instant Messaging vytvoriť aj manažér služieb a preto na to myslíme už v prototype a snažíme sa ho spraviť čo najabstraktnejšie.

5.3.2 Rozhranie služby Instant Messaging

Služba obsahuje dohromady 3 servlety, Tabuľka 5.1 obsahuje popis týchto troch servletov.

Tabuľka 5.1 Popis rozhrania pre službu Instant Messaging (IM).

URL adresa	argument	popis
/sign	in	argument nadobúda hodnotu UID, používa sa pri prihlásení klienta k službe IM, po zadaní je táto hodnota pripísaná do tabuľky pripojených klientov na serveri vo forme UID=IP
	out	argument nadobúda hodnotu UID, používa sa pri odhlásení klienta zo služby IM, po zadaní je táto hodnota vymazaná z tabuľky pripojených klientov na serveri
/message	to	argument nadobúda hodnotu UID používateľa, ktorému chceme poslať správu
	msg	argument nadobúda hodnotu JSON správy, ktorá je následne preposlaná klientovi na adresu podľa UID uloženého v tabuľke pripojených klientov. Štruktúru JSON správy rieši klient, server túto správu iba prepošle pomocou socketovej komunikácie s klientom.
/contactList		tento servlet nemá žiadne argumenty a slúži len pre potreby ladenia aplikácie, po zadaní sa vypíše obsah tabuľky pripojených kientov na serveri

Ku servletom je možné dostať sa po zadaní nasledujúcej štruktúry ako URL:

```
http://ip-servera:port/SrvcM/servlet?arg1=val1&arg2=val21
```

kde *ip-servera* je verejná IP adresa počítača na ktorom beží Apache Tomcat, *port* je číslo portu, na ktorom tento server počúva, *SrvcM* je názov modulu (Service Manager – správca služieb), *servlet* je jeden zo servletov opísaných v tabuľke vyššie a *arg=val* sú názvy a hodnoty argumentov, ktoré možno reťaziť pomocou znaku &.

Príklad prihlásenia a odhlásenia používateľa s UID „user123“:

```
http://147.175.158.173:8080/SrvcM/sign?in=user123
http://147.175.158.173:8080/SrvcM/sign?out=user123
```

5.4 Televízna aplikácia (Samsung Smart TV)

Na obrázku Obr. 5.1 je zobrazený sekvenčný diagram odoslania správy z mobilnej aplikácie na televíziu aplikáciu. Rovnakým spôsobom bude prebiehať opačná komunikácia, teda z televíznej aplikácie na mobilnú. Avšak predtým ako príde k odoslaniu samotnej správy je potrebná kooperácia viacerých entít. Prvá entita ktorá figuruje pri nadviazaní komunikácie je Identity Manager (IdM). Z IdM dostaneme zoznam dostupných služieb na základe ktorých vyšleme požiadavku na službu Instant Messaging (IM). Služba IM overí poskytnuté prihlasovacie údaje voči ďalšej entite Security Manager-u (SecMng). Funkcionalita SecMng nám však nie je známa detailne, takže ju zatiaľ vynechávame. Rátame však s jej zapojením do konečnej architektúry. Na základe kladnej alebo zápornej odpovede na overenie následne služba IM pošle odpoveď mobilnému zariadeniu o prihlásení. Po úspešnom prihlásení obidvoch zariadení môžu spolu komunikovať cez službu IM.

Na vyvíjanie aplikácie pre televíziu používame najnovšie oficiálne prostredie Samsung SDK 3.5.2 a simulátor pripojený k tomuto prostrediu. Aplikáciu testujeme aj na reálnej televízii a to nasledujúcim spôsobom. Po vytvorení kontrolného bodu a otestovaní na simulátore našu aplikáciu zbalíme vo vývojovom prostredí a nahráme na webový server umiestnený na lokálnej sieti čo je v našom prípade Apache server bežiaci na pozadí. Keď máme takto nachystaný server môžeme nahrať aplikáciu na televíziu. Na to sa potrebujeme v televízii prihlásiť pod špeciálneho užívateľa a to tzv. vývojára. Následne v nastaveniach zmeníme IP adresu odkiaľ sa majú sťahovať nové aplikácie na IP adresu nášho servera a vyberieme voľbu synchronizovať. Nami vytvorená aplikácia sa nainštaluje a je pripravená na použitie.

Pri vytváraní prototypu televíznej aplikácie sme sa inšpirovali oficiálnym návodom [1]. V tomto návode sú uvedené dve aplikácie, televízna a emulácia mobilnej aplikácie vo webovom prehliadači. Tieto komunikujú medzi sebou pomocou http požiadaviek a odpovedí. Čo je do veľkej miery podobné ako v našej aplikácií. Je tu názorne uvedené ako pracovať a komunikovať s externými zariadeniami ako sú napríklad mobilné zariadenia alebo tablety. V návode je nadviazanie spojenia, posielanie správ a dokonca aj súborov medzi emulátorom mobilnej aplikácie a televíznou aplikáciou. Nakoľko máme v tejto oblasti skromné skúsenosti, tak sa nám nepodarilo ešte spraviť vlastný funkčný prototyp. Na jeho vytváraní však svedomito pracujeme a už čoskoro očakávame prvé výsledky.

5.5 Vysielacia/Prijímacia časť (AZ Box)

Úlohou tejto časti tímového projektu je spojazdniť zobrazovanie aplikácií na hybridnej televízii. Aplikácie sú vysielané prostredníctvom pozemného digitálneho vysielanie DVB-T.

Vysielanie bolo implementované ako súčasť diplomového projektu Ing. Romana Broniša teda je implementované a funkčné. Vysielaná aplikácia má byť prijatá na set-top-box pomocou doinštalovaných unixových programov z balíka dvb-apps a súčastí programu OpenCaster. Následne má byť zobrazená vo webovom prehliadači nainštalovanom na set-top-boxe, ktorý je schopný spracovávať a zobrazovať CE-HTML aplikácie.

Na začiatku semestra bola na set-top-box nainštalovaná voľne šíriteľná verzia operačného systému Enigma2 OpenAZbox Pli MOD, ktorá mala predpoklady na to, aby bolo do AZBoxu možné doinštalovať všetky potrebné náležitosti a programy. Bolo vytvorené prostredie na cross-kompiláciu pre túto platformu, keďže neobsahovala žiaden vlastný kompilátor. Cross-kompilácia potrebných programov do AZBoxu bola nakoniec úspešná, aj keď si vyžadovala veľa času kvôli potrebe doinštalovania potrebných knižníc do cross-kompilátora ako aj do AZBoxu.

Napriek úspešnej inštalácii programov pre správne fungovanie prijímania aplikácií, aj po viacerých rôznych pokusoch sa nepodarilo nainštalovať webový prehliadač. Dôvodom je, že webové prehliadače pre operačný systém Enigma2 síce existujú, ale každá distribúcia tohto operačného systému je naviazaná na konkrétny hardvér a pre AZbox Premium HD doteraz nebol implementovaný žiaden webový prehliadač pre Enigmu2.

Z uvedeného dôvodu bolo preto nutné v pokročilom čase semestra zvoliť celkom iný prístup a na set-top-box nainštalovať originálny firmvér, ktorý obsahuje webový prehliadač. Pôvodný firmvér pre AZBox Premium HD síce obsahuje webový prehliadač, ale čo sa týka vývojárskych vlastností je vo veľkej miere obmedzený. Linuxové jadro je príliš staré na to, aby bolo možné štandardnými spôsobmi vytvoriť pre toto prostredie cross-kompilátor. Alternatívnymi spôsobmi bolo nakoniec možné vytvoriť prostredie pre cross-kompiláciu a momentálne je nutné opäť kompilovať potrebné programy pre prijímanie aplikácií cez DVB-T, pri komplikácií ktorých sa vyskytujú nové problémy.

Keďže práca na tejto časti tímového projektu bola nepriaznivo ovplyvnená zlou voľbou a neúspešnou prácou s Enigmou2, táto časť nebude figurovať ako súčasť prototypu, ale bude dokončená až počas letného semestra.

6 Implementácia riešenia

V tejto časti budú popísané zmeny oproti riešeniu prototypu a ako sme postupovali vo finálnej implementácii produktu.

6.1 IM service

V tejto časti sa nachádza popísaná IM service, ktorá umožňuje komunikáciu medzi Samsung TV a android zariadením.

6.1.1 Prihlasovanie a odhlasovanie používateľov

V rámci finálnej verzii došlo k prepojeniu IM služby s IdM službou. Z IdM služby sa pri spustení serverovej časti stiahne zoznam používateľov pridaných v IdM, ktorý sa zapíše do štruktúry v triede „Clients“. Táto trieda je postavená na vzore Singleton, teda každý servlet má k nej prístup. Trieda poskytuje jednoduché API na získavanie zoznamu kontaktov a zmenu atribútov pri jednotlivých kontaktoch. Pri prvotnom načítaní sa ku každému kontaktu (ku každému „username“) priradí IP adresa s hodnotou „None“. Táto hodnota indikuje, že používateľ nie je nikde prihlásený. V momente, keď sa používateľ prihlási, zmení sa hodnota IP adresy na IP adresu, z ktorej sa prihlásil. Týmto spôsobom vieme v každom servlete získať IP adresu používateľa na základe jeho prihlasovacieho mena „username“.

6.1.1.1 Nedostatky a možné vylepšenia

V momentálnej verzii chýba overenie PIN kódu voči IdM nakoľko IdM neobsahovalo reálne dáta. V ďalšej verzii by toto overenie určite nemalo chýbať. Takisto sa zoznam používateľov z IdM získava iba v čase štartu serverovej časti a teda nie je zabezpečená žiadna aktualizácia týchto záznamov v rámci serverovej časti – v praxi kontaktuje naša služba službu IdM iba raz. Ďalšou nevýhodou je, že používateľ nemôže byť prihlásený naraz na viacerých strojoch. Tento nedostatok by sa dal riešiť zložitejšou štruktúrou používateľov.

6.1.2 Prijímanie a odosielanie správ

Socketová komunikácia, ktorú sme spomínali v rámci Prototypu sme z finálnej verzie nakoniec museli vypustiť. Dôvodom vypustenia bola nepodpora zo strany Samsung Smart TV aplikácií. Tento nedostatok sme však vyriešili tak, že v rámci servletu, ktorý obsluhuje správy máme príznak, ktorý hovorí o tom, či chceme správu prijať alebo odoslať (viď 6.1.5). Pri vhodne zvolenom intervale overovania nových správ na serveri sa môže komunikácia priblížiť komunikácii v reálnom čase.

Správy na serveri sa ukladajú v štruktúre tvorenej triedou „Messages“, ktorá je takisto ako trieda „Clients“ navrhnutá podľa vzoru Singleton. Rovnako obsahuje jednoduché API na pridávanie a získavanie správ podľa príjemcu – „username“. Po prijatí požiadavky „receive“ sa danému používateľovi odošle JSON správa, ktorá obsahuje všetky správy určené pre príjemcu. Každá správa obsahuje text správy a „username“ odosielateľa správy.

6.1.2.1 Nedostatky a možné vylepšenia

Takisto ako v časti prihlasovania a odhlasovania, ani v tejto časti sme nebrali do úvahy bezpečnostné aspekty. Táto problematika by sa dala určite vylepšiť. Ďalšou nevýhodou je, že neprihlásenému používateľovi nemožno odoslať správu. Aj tento nedostatok by sa dal vyriešiť napríklad implementáciou databázy správ na serveri a teda v prípade, že je používateľ neaktívny, správa sa na serveri zapíše do perzistentného úložiska. V tejto verzii sme prijímanie správ pre nekatívnych používateľov neimplementovali aj preto, lebo správy sa uchovávajú iba v pamäti a teda pri výpadku serveru dochádza k ich strate.

6.1.3 Overovanie kontakt listu

Rovnako ako prijímanie správ, aj zmeny v kontakt liste si musí klient vyžiadať pomocou servletu na to určeného. Ako odpoveď klientovi príde JSON správa s obsahom štruktúry vytvorenej triedou „Clients“ – JSON správa, ktorá ako kľúče obsahuje „username“ jednotlivých používateľov a ako hodnotu obsahuje buď IP adresu (v prípade, že je používateľ prihlásený) alebo reťazec „None“ (v prípade, že je používateľ neaktívny).

6.1.4 Chybové správy a spätná väzba

Implementácia je oproti prototypu rozšírená aj o chybové správy, ktoré indikujú nesprávny formát alebo kombináciu atribútov. Taktiež dokážu odhaliť preklepy v názvoch argumentov. Rovnako ako chybové správy boli implementované správy so spätnou väzbou, ktoré hovoria o stave – napr. pri upozornení, že používateľ, ktorého sa snažíme kontaktovať je neaktívny.

6.1.5 IM Service API

API služby Instant Messaging sa oproti prototypu zmenilo nasledovne:

URL adresa	argument	popis
/sign	in	argument nadobúda hodnotu „username“ zo zoznamu používateľov v službe IdM, používa sa pri prihlásení klienta k službe IM, po zadaní je táto hodnota pripísaná do tabuľky pripojených klientov na serveri vo forme username=IP
	out	argument nadobúda hodnotu „username“ zo zoznamu používateľov v službe IdM, používa sa pri odhlásení klienta zo služby IM, po zadaní je v tabuľke pripojených klientov na serveri vymazaná IP adresa a nahradená textom „None“ – userneme=None
/message	send	argument nenadobúda žiadne hodnoty, slúži iba ako príznak metódy – hovorí službe o tom, že chceme správu poslať; tento argument sa používa spolu s argumentom „to“ a „msg“
	to	argument nadobúda hodnotu „username“ zo zoznamu používateľov v službe IDM, ktorému chceme poslať správu
	msg	argument nadobúda hodnotu JSON správy, ktorá je následne preposlaná klientovi na adresu podľa „username“ uloženého v tabuľke pripojených klientov; štruktúru JSON správy rieši klient, server túto správu iba prepošle keď si ju používateľ vyžiada
	receive	argument nenadobúda žiadne hodnoty, slúži iba ako príznak metódy – hovorí službe o tom, že chceme prijať správy, ktoré sú na serveri; samotné odosielanie server nerieši, prijímanie rieši klient, ktorý túto URL v pravidelných intervaloch kontroluje
/contactList		tento servlet nemá žiadne argumenty a slúži pre zistenie zmien v kontakt liste, po zadaní sa vypíše obsah tabuľky pripojených kientov na serveri, ktorý ďalej spracúvava klient

Použitie API je opísané v časti 5.3.2.

6.2 Mobilná aplikácia (Android)

Predložená kapitola sa venuje implementácii mobilnej aplikácii. V tejto časti sa nachádza doplnenie špecifikácie riešenia, doplnenie návrhu riešenia, používateľské prostredie a konečná implementácia.

6.2.1 Doplnenie špecifikácie riešenia

Počas vývoja aplikácie od prototypu nastalo niekoľko funkcionálnych zmien oproti prototypu mobilnej aplikácie. Upustilo sa od komunikácie mobilnej aplikácie s IdM, pretože v architektúra má mobilná aplikácia pristúpiť k službám IdM a počas riešenia celého projektu neboli v IdM služby. Takisto sa zmenil spôsob prijímania správ. Pôvodne sme to riešili cez sockety, ale museli sme sa tohto

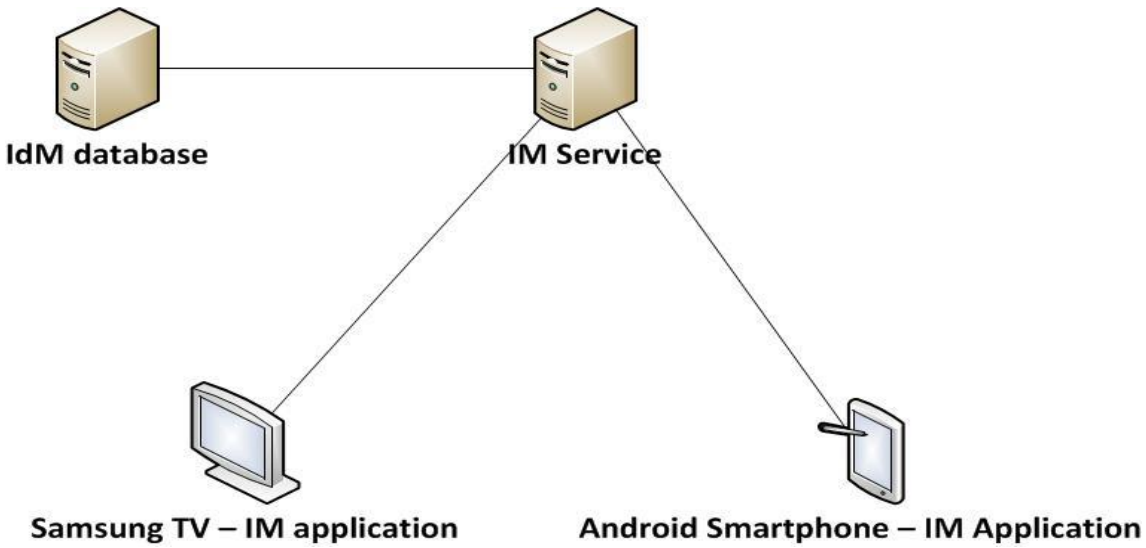
spôsobu vzdať, pretože sa nám to nedarilo implementovať. Namiesto toho sa klient dopytuje po nových správach od servera pomocou HTTP požiadaviek.

Zhrnuté špecifikácie požiadaviek:

- Zobrazovanie zoznamu kontaktov
- Komunikácia s ekvivalentnými IM aplikáciami
- Prihlásenie, odhlásenie používateľa

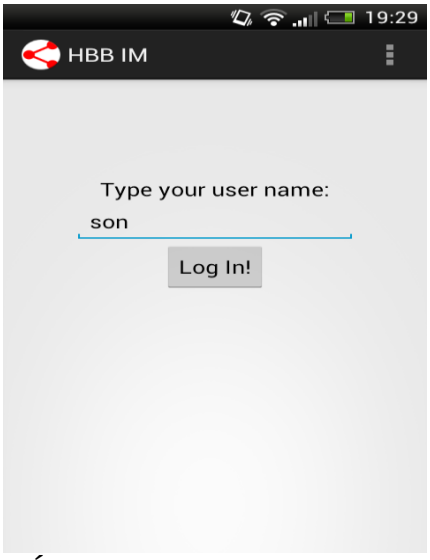
6.2.2 Doplnenie návrhu riešenia

V priebehu implementácie som pridal nové obrazovky do aplikácie a aplikácia bola doplnená o ďalšiu funkcionalitu. Celkovo sa aplikácia skladá z troch obrazoviek. Na úvodnej obrazovke sa používateľ prihlasuje pomocou svojho prihlasovacieho mena uloženého v IdM. Po tomto kroku sa používateľovi zobrazí zoznam kontaktov a aj to či je niekto z kontaktov online alebo offline. Po kliknutí na nejaký z kontaktov s ním môže prihlásený používateľ komunikovať. Architektúra IM aplikácie je zobrazená na obrázku Obr. 6.1. Všetky entity medzi sebou komunikujú pomocou JSON správ a HTTP requestov. Smery komunikácií sú zobrazené na spomenutom obrázku. IM service si udržuje zoznam používateľov z IdM databázy a aj informáciu o tom či je daný kontakt online alebo nie. Pri prihlásení si Android IM application tento zoznam kontaktov stiahne a pravidelne si ho aktualizuje. Správy si medzi sebou posielajú IM aplikácie prostredníctvom IM service. Pre prichádzajúce správy sa IM aplikácie dopytuje IM service pomocou http requestov. IM Service aplikácii odpovie či prišla nejaká správa alebo nie. Ak áno tak na ňu IM aplikácia upozorní pomocou zavibrovania a správu zobrazí na obrazovke.



Obr. 6.1: Architektúra IM služby

Na obrázku Obr. 6.2 je zobrazená úvodná obrazovka mobilnej aplikácie HBB IM. Používanie aplikácie je bližšie popísané v časti Používateľská príručka.



Obr. 6.2: Úvodná obrazovka mobilnej aplikácie

6.3 Azbox

V tejto časti bolo našou úlohou prijímanie digitálneho vysielania DVB-T na set-top-box AZ Box Premium HD+, pričom sa jednalo o vlastné vysielanie, ktoré prenášalo HbbTV aplikácie. Po prijatí dát cez DVB-T ich bolo potrebné rozbaliť do priečinka a spustiť na pripojenom televízore vo webovom prehliadači.

Konkrétny postup zahŕňal inštaláciu programov z balíka dvb-apps a súčastí programu OpenCaster. Pomocou týchto programov bolo možné odchytávanie DVB-T vysielania a jeho následné spracovávanie. Zároveň bolo samozrejme potrebné mať na set-top-boxe nainštalovaný webový prehliadač, v ktorom by sa prijatá HbbTV aplikácia spustila.

Počas tvorby prototypu v zimnom semestri bolo odskúšaných viacero prístupov ako túto úlohu vyriešiť.

Najprv bola ako firmvér set-top-boxu nainštalovaná voľne šíriteľná verzia systému Enigma2 OpenAZBox Pli MOD. Tento firmvér bol postavený na novej verzii kernelu 3.4.4 a bolo možné naň využitím cross-kompilácie nainštalovať programy potrebné na prijímanie. Problémy nastali pri inštalácii webového prehliadača, keďže pre túto kombináciu hardvéru a softvéru ešte neexistoval hotový prehliadač.

Zvolený bol teda alternatívny prístup, ktorý spočíval v opätovnej inštalácii pôvodného firmware AZBoxu. Dôvodom bolo, že tento už mal ako svoju súčasť webový prehliadač. Pri tejto alternatíve však nastal opačný problém. Originál firmvér pre AZBox beží na veľmi starej verzii

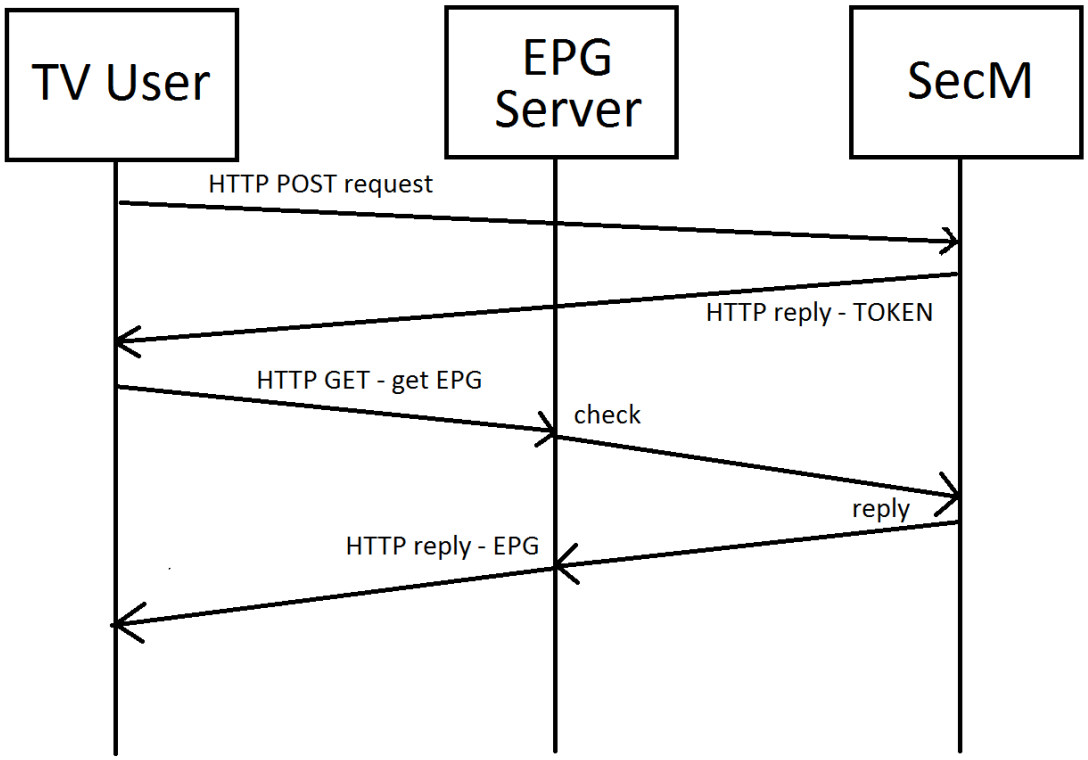
kernelu, kde sa ešte nevyužívali nové Linuxové hlavičky (Linux kernel headers) a teda nebolo možné naň vytvoriť bežnou cestou cross-kompilátor a nájdený cross-kompilátor pre túto architektúru dokázal kompilovať iba staré a jednoduché zdrojové kódy, ktoré nevyužívali práve spomínané Linuxové hlavičky.

Po neúspechoch v zimnom semestri bola táto časť vypustená z plánu a bola nahradená inou úlohou. Nakoľko táto časť nesúvisela s prácou ostatných, jej vypustenie nijak neovplyvnilo prácu ostatných členov tímu.

6.3.1 Získanie EPG (Electronic Program Guide)

Novou úlohou v letnom semestri, na ktorej sa pracovalo namiesto prijímania na AZBox bolo vytvorenie komunikácie ako používateľ TV (TV User) s ďalšími komponentami architektúry HBB-Next: Security Manager (SecM) a EPG server.

Komunikácia bola implementované podľa schémy na obrázku Obr. 6.3.



Obr. 6.3: Získavanie EPG – komunikácia

Prvým krokom je „login“ – vyžiadanie si „tokenu“ od SecM. SecM beží na nejakej IP s portom 8080. Odpovedá na správy typu HTTP POST. V prvej správe sa TV User identifikuje, autentifikuje a tým si vyžiada „token“. Obsah požiadavky HTTP POST je: *id=1&user=2&pin=1234&device=2*. Security manager je potrebné kontaktovať na porte 8080.

Odpoveď od SecM je zabalená do protokolu HTTP, a je vo formáte JSON. Vyzerá nasledovne *{"1":{"user":"2","device":"2","token":"987059828","level":"3"}}*. Z tejto správy je možné vybrať token a uložiť si ho pre ďalšie použitie, pretože ďalšia správa od TV Usera smeruje na EPG server, kde je potrebné týmto tokenom sa autentifikovať.

Formát tretej správy je HTTP GET request a vyzerá nasledovne: *http://IP:8081/?token=987059828&users=2*. IP je IP adresa EPG servera. Odpovedá pre požiadavky na port 8081. A ako dáta posiela prijatý token a číslo používateľa.

Po prijatí tejto správy si EPG server overí token u SecM a následne odpovedá HTTP protokolom TV Userovi a podáva mu EPG informácie. Momentálne EPG informácie tvorí zoznam programov v nešifrovanom textovom formáte (plaintext): *"STV1, STV2, TV JOJ, Markiza, Doma"*.

SecM a EPG server sú už hotové súčasti implementované Bc. Tomášom Kokolevskym. Úlohou nášho tímu bolo len implementovať vysielanie, prijímanie a spracovávanie HTTP správ na strane TV Usera. O túto funkcionalitu bol obohatený program, ktorý pôvodne slúžil na vizualizáciu set-top-boxu s diaľkovým ovládaním v architektúre HBB-Next. Pridaná funkcionalita nemá veľa spoločného s pôvodným programom, no je ovládaná podľa jeho štandardov a teda: Príkaz sa zadáva na štandardný vstup programu „klient“, pričom program „server“ musí byť zapnutý. Formát príkazu: *meno* getepg online. Ako meno by sa malo zadať jedno z mien, ktoré figurovali ako používatelia v pôvodnom programe. Pre našu funkcionalitu môžeme zadať ľubovoľné meno zo zoznamu v programe (napr. Roman alebo Andrej). „getepg“ je príkaz, ktorý spúšťa novú funkcionalitu. „online“ slúžil v pôvodnom programe na udanie stavu, teraz figuruje len ako doplnenie do formátu zadávania príkazov.

Na štandardnom výstupe programu „server“ sa zobrazujú informácie o priebehu komunikácie a nakoniec aj požadovaná odpoveď – EPG informácie.

6.4 Aplikácia pre XMPP based IM

Táto časť v prototype nie je spomenutá. Toto riešenie bolo navrhnuté ako doplnkové. Pôvodne sa však malo jednať o aplikáciu zobrazujúcu zdieľané obrázky na lokálnej sieti. Toto riešenie však už existovalo v podobe voľne šíriteľnej aplikácie a preto sme sa rozhodli ho nahradiť súčasnými trendmi. Ako výsledok tohto rozhodnutia sa ďalej pracovalo na aplikácii, ktorá by umožnila chat s Facebook priateľmi. Toto riešenie bolo zovšeobecnené na IM (instant messaging) na základe XMPP a teda nie len Facebook chat ale aj napr. GTalk.

XMPP (The Extensible Messaging and Presence Protocol) je otvorená technológia pre komunikáciu v reálnom čase, ktorá poskytuje základ pre širokú škálu aplikácií vrátane IM, presence, hlasové a video hovory, spolupráca a všeobecné smerovanie XML dát. Kľúčovú

technológiu skrývajúcu sa za XMPP vynašiel Jeremie Miller v roku 1998, v roku 1999 a 2000 bola doladená komunitou Jabber open-source a v roku 2002 – 2003 ju formalizovala IETF.

Naša aplikácia využíva pripojenie na XMPP server a preto je možné ju využiť pre pripojenie na ľubovoľnú IM službu postavenú na XMPP/Jabber (Facebook chat, GTalk ...) pomocou tzv. JID (Jabber ID) a hesla. V prípade Facebook chatu má JID tvar „FacebookID@chat.facebook.com“, v prípade GTalk-u je to celá emailová adresa Google účtu.

Aplikácie pre Samsung Smart TV sú tvorené ako webové aplikácie a preto naša aplikácia používa HTML, JavaScript a jQuery. Pre zadávanie textu a pokynov pomocou TV ovládača používame štandardné pluginy pre vstup. Pre pripojenie na XMPP server používame jQuery-XMPP plugin. Tento plugin je open-source a rozšírili sme ho o zopár funkcií , ktoré nám umožňujú získať celé meno pripojených priateľov a zobraziť ho tak namiesto ID.

Úvodná HTML strana je jednoduchá a obsahuje polia pre zadanie JID a hesla, chatovacie okno zobrazujúce prijaté a odoslané správy, log panel zobrazujúci systémové informácie a panel pre zobrazenie zoznamu priateľov. Po vyplnení prihlasovacích údajov TV ovládačom a stlačení červeného tlačidla pre pripojenie sa vytvára pripojenie na tzv. BOSH server (Bidirectional-streams Over Synchronous HTTP), slúžiaci na vytvorenie spojenia medzi dvoma entitami (klient - server) pomocou synchronizovaných HTTP požiadavok/odpovedí.

var url ="http://bosh.metajack.im:5280/xmpp-httpbind";

$.xmpp.connect({url:url, jid: jid, password: password,

Ďalšími parametrami funkcie *connect* sú event handlery :

onConnect : po úspešnom pripojení sa vypíše správa „Connected“ a zmení sa stav na online.

onPresence : XMPP server po pripojení posiela informácie o online priateľoch, v tejto funkcii sa následne ukladajú do poľa a posiela sa správa pre zistenie celého mena podľa ID.

onDisconnect : vypíše sa „Disconnected“ a vymaže sa pole obsahujúce zoznam priateľov.

onMessage : po prijatí správy sa do chat okna zobrazí meno odosielateľa a samotná správa.

onError : zobrazí sa chybová hláška.

onIq : spracovanie správy obsahujúcej celé meno používateľa, uloží sa do poľa priateľov paralelne s ID.

Pre správne fungovanie aplikácie je potrebné vypnúť firewall alebo nastaviť výnimku. Po stlačení zeleného tlačidla sa prejde celé pole priateľov a zobrazí sa jeho obsah v bočnom paneli. Následne je možné sa šípkami na TV ovládači pohybovať po zozname a po zvolení konkrétneho priateľa sa kurzor nastaví do poľa v okne pre chat a je možné zadať samotnú správu. Po stlačení „OK“ sa správa zobrazí a zavolá sa funkcia $.xmpp.sendMessage, ktorej parametre sú *to* (príjemca) a *body* (telo správy). Príjemca sa zvolí na základe aktuálnej pozície v zozname priateľov a správa sa odošle. Po stlačení modrého tlačidla sa používateľ odhlási.

6.5 Samsung Smart TV IM aplikácia

Aplikácia Samsung Smart TV IM slúži na posielanie správ buď medzi dvomi a viacerymi Samsung Smart TV alebo medzi android zariadením, či už tabletom alebo smatfónom, a Samsung Smart TV.

Aplikácia sa skladá z troch základných komponentov a to sú:

- Html súbor – slúži ako vstupný bod do programu a sú na ňom zobrazené všetky potrebné komponenty
- CSS súbor – slúži pridávanie štýlov do Html súboru
- Javascript súbor – je v ňom obsiahnutá celá logika programu ako napríklad komunikácia so serverom alebo ovládanie aplikácie pomocou diaľkového ovládania

Po spustení aplikácie nás privíta okno s logickým rozdelením prvkov, na ktoré sme zvyknutý z iných aplikácií slúžiacich na posielanie správ. Vľavo hore máme textové pole na zadanie prihlasovacieho mena. Vedľa neho, v strede hore máme zobrazený zoznam prihlásených užívateľov. Pod týmito dvoma oknami máme pole pre všetky typy správ, či už prijaté, odoslané, chybové alebo stavové. Pod ním máme textové pole pre zadávanie správ. Na pravej strane obrazovky máme pri označených textových poliach zobrazenú klávesnicu na zadávanie textu.

Ovládanie je jednoduché a prirodzené pomocou smerových tlačidiel na ovládači. Ako prvé po spustení aplikácie potrebujeme zadať naše prihlasovacie meno. Následne sa prihlásime stlačením červeného tlačidla. O úspešnosti alebo neúspešnosti každej akcie nás informuje správa v poli pre všetky typy správ. Po úspešnom prihasení potrebujeme vybrať priateľa s ktorým budeme písať. To dosiahneme stlačením tlačidla return, ktoré skryje klávesnicu na zadávanie pomocou ovládača a následne prejdeme pravým tlačidlom na zoznam

priateľov. V zozname priateľov si vyberieme pomocou smerových tlačidiel hore alebo dolu priateľa s ktorým si chceme písať. Následne prejdeme ľavým tlačidlom späť do poľa na písanie správy, kde napíšeme správu a tá sa potvrdením (enter) odošle.

Na komunikáciu so serverom boli použité správy typu XMLHttpRequest (XHR). Tieto správy sa používajú na posielanie požiadaviek priamo na webové servery pričom dokážu spracovať aj odpoveď. Pomocou XHR správ posielame a prijímame správy, získavame informácie o prihlásených užívateľoch a dotazujeme sa na server ohľadom nových správ.

Medzi hlavné funkcie aplikácie patria funkcie na odosielanie správ, prijímanie správ, obnovovanie kontaktov a funkcie spojené s ovládaním aplikácie. Funkcie na prijímanie správ a obnovovanie kontaktov fungujú na pravidelnom dotazovaní sa na server. U týchto funkcií je možné meniť periodicitu dotazovania sa pomocou konštánt *timeoutCl* a *timeoutReceive* v súbore *Main.js*. Štandardné hodnoty sú nastavené na 3000ms pre obnovovanie kontaktov a 2000ms pre dotazovanie sa na server ohľadom nových správ. Ďalšou dôležitou konštantou v programe je reťazec *URL* v súbore *Main.js*. Tento reťazec sa skladá z IP adresy servera a portu na ktorý sa má dotazovať. V prípade zmeny adresy alebo portu servera je ho potrebné modifikovať.

Ďalšou často využívanou funkcionalitou v našej aplikácií je Input Method Editor (IME). IME slúži na zadávanie textu pomocou ovládača. IME funguje takým spôsobom, že pri textovom poli kde je možné zadať text, nám zobrazí klávesnicu. IME ponúka 2 typy klávesnice. Prvá je kompletná qwerty klávesnica kde sa môžeme pohybovať pomocou smerových tlačidiel a potvrdením pridáme označené písmeno. Druhá je obmedzená, známa skôr zo starších mobilov, kde volíme písmená pomocou viacnásobného stlačenia čísla. IME podporuje aj zadávanie pomocou hlasu, avšak túto funkcionalitu sme v našej aplikácií nevyužili.

Nemôžeme zabudnúť na ovládanie celej aplikácie, ktoré je realizované funkciou *Main.keyDown()*. Táto funkcia sa zavolá po každom stlačení tlačidla na ovládači a pokiaľ máme zadefinované konkrétne tlačidlo vykoná sa príslušný zoznam príkazov.

Aplikácia splňuje základné požiadavky, ktoré sú kladené na všetky aplikácie tohto charakteru. Čo sa týka užívateľského hľadiska, tak je prehľadná, má logické usporiadanie prvkov a je interaktívna. Čo sa týka logiky funkcií tam je aplikácia zaujímavejšia, pretože spolupracuje s celou nami vytvorenou architektúrou. Uvedomujeme si, že má ešte mnoho nedostatkov, avšak cieľ tejto aplikácie nebol poskytnúť dokonalú aplikáciu na posielanie správ medzi kamarátmi, ale poukázať na možnosti využitia Samsung Smart TV v spolupráci s iným technológiami.

7 Záver

V tejto časti dokumentu sa nachádza záver z jednotlivých semestrov a aj zhrnutie toho čo sme sa naučili a čo nám vypracovanie tohto projektu prinieslo.

7.1 Záver prvého semestra

V rámci prvého semestra tímového projektu sme sa oboznámili s architektúrou HBB Next a s prenosom videa cez DVB vysielanie. Vyšpecifikovali a navrhli sme si niekoľko cieľov nášho tímového projektu. V druhom semestri sa pokúsime implementovať jednoduchú komunikačnú službu typu Instant messaging, ktorá bude fungovať v prostredí HBB Next architektúry a bude demonštrovať možnosť prepojenia komunikácie na zariadeniach s operačným systémom Android so zariadeniami Samsung Smart TV. Ďalším cieľom je pokúsiť sa o rozšírenie Set top boxu o možnosť prijatia a rozbalenia HBB aplikácie, aby bolo možné komunikovať aj touto formou.

7.2 Záver druhého semestra

V rámci druhého semestra sme v rámci tímového projektu implementovali komunikačnú službu, ktorá funguje v prostredí architektúry HBB Next. Služba je založená na aplikačnom serveri, ktorý komunikuje so službou IdM, ktorá je špecifická práve pre architektúru HBB Next. Aplikačný server poskytuje rozhranie HTTP požiadaviek, za pomoci ktorých možno implementovať jednoduchého klienta na rôznych platformách. V rámci tohto projektu sme implementovali chatovacieho klienta postaveného na platformách Android a Samsung Smart TV.

Taktiež sme sa na problematiku komunikácie pozreli z iného hľadiska, pričom sme sa zamerali na trendy súčasnosti. Implementovali sme aplikáciu na komunikáciu, ktorá môže komunikovať s väčšinou moderných chatovacích služieb akými sú Facebook či Google Talk. Aplikácia je implementovaná pre platformu Samsung Smart TV.

V rámci druhého semestra tímového projektu sa nám bohužiaľ nepodarilo nájsť vhodné riešenie, ktoré by zabezpečilo softvérovú modifikáciu Set top boxu tak, aby bolo možné na Set top boxe prijímať a zobrazovať HBB aplikácie. Táto skutočnosť je zdokumentovaná a podrobnosti sú vysvetlené v rámci tohto dokumentu.

Výsledným produktom tohto tímového projektu sú jednoduché demá aplikácií, ktoré môžu pomôcť v ďalšom vývoji aplikácií v rámci architektúry HBB Next. Taktiež tieto aplikácie môžu slúžiť ako dôkaz funkčnosti jednotlivých blokov architektúry HBB Next. Každá zo súčastí výsledného produktu môže byť použitá v ďalších projektoch v rámci našej fakulty.

Za dva semestre tímového projektu sa nám bohužiaľ nepodarilo úplne prepojiť všetky časti a rozhodne sa nám nepodarilo vyskúšať všetky časti novovznikajúcej architektúry HBB Next. Avšak veríme, že aj táto naša práca pomôže vývoju tejto novej architektúry.

Na záver by sme tiež chceli poďakovať nášmu vedúcemu pedagógovi Ing. Tomáš Kováčik, PhD. za odbornú pomoc, vedenie a trpezlivosť.

7.3 Čo sme sa naučili

Dva semestre tímového projektu nás naučili nie len novým technológiám, ale samozrejme naučili nás práci v kolektíve. Okrem práce v kolektíve sme si vyskúšali aké je pracovať pod vedením. Myslíme si, že toto môže byť považované ako príprava na prácu v budúcom zamestnaní a sme radi, že sme si to vyskúšali aj v rámci štúdia.

Síce sme každý pracovali na niečom inom, čiže sme si nevyskúšali aké to je pracovať napr. s verziovacími systémami v kolektíve, ale naučili sme sa komunikovať a spolu rozmýšľať. Taktiež sme sa naučili rozdeliť si medzi sebou úlohy a v prípade, že mal niekto s niečím problém, snažili sme sa všetci mu pomôcť.

Zdroje

[1] HBB Next-NEXT Next Generation Media [online]. [cit 2012-6-11] http://www.hbb-next.eu/index.php/

[2] Introduction: HBB Tv [online]. [cit 2012-2-11] http://www.hbbtv.org/pages/about_hbbtv/introduction.php

[3] Android, the world's most popular mobile platform | Android Developers [online]. [cit 2012-7-11] http://developer.android.com/about/index.html

[4] HBB-Next Project [online] [cit. 2012-11-11] http://www.hbb-next.eu/index.php

[5] E. Mikoczy, M. Probst: Next Generation Hybrid Broadcast Broadband, deliverable 6.1.1: Initial Version of the HBB-NEXT System Architecture, 2012

[6] P.Podhradsky, G. Rozinaj, E. Mikoczy: HBB-Next FP7 Project, 2011

[7] Fórum Satdigitalne.cz [online]. [cit 2012-11-14] http://forum.satdigitalne.cz

[8] Forum Gateway - Sat Universe [online]. [cit 2012-11-14] http://www.sat-universe.com/showthread.php?t=241159

[9] AZBOX PREMIUM HD PLUS - Videotech [online]. [cit 2012-11-14] http://www.videotech.cz/hdtv-prijimace/satelitni-dvb-s/linuxove/azbox-premium-hd-plus/

[10] AZBox HD | AZBox Premium HD+ "PLUS" [online]. [cit 2012-11-14] http://www.azboxhd.cz/azbox-hd-premium-plus.html

[11] HBB NEXT Overview: http://www.hbb-next.eu/documents/HBB-NEXT_D3.2.pdf

[12] Smart TV | SAMSUNG Developers [online]. [cit 2012-11-14] http://developer.samsung.com/smarttv

[13] 40" EH5300 (UA40EH5300J) | OVERVIEW SAMSUNG [online]. [cit 2012-11-14] http://www.samsung.com/hk_en/consumer/tv-av/televisions/led-tv/UA40EH5300JXZK

[14]. SamyGO [online]. [cit 2012-11-14] http://wiki.samygo.tv/index.php5/Main_Page

[15] Creating a Convergence Application [online] [cit 2012-12-13] http://www.samsungdforum.com/Guide/tut00024/index.html

[16] Apache Tomcat [online]. [cit 2013-5-16] http://tomcat.apache.org/

[17] Tutorial: Installing Tomcat 7 and Using it with Eclipse – [online]. [cit 2013-5-16] http://www.coreservlets.com/Apache-Tomcat-Tutorial/tomcat-7-with-eclipse.html

[18]. Apache Tomcat Eclipse Integration [online]. [cit 2013-5-19] http://wiki.samygo.tv/index.php5/Main_Page

[19] JSON.simple - A simple Java toolkit for JSON – [online] [cit 2013-5-13] http://www.samsungdforum.com/Guide/tut00024/index.html

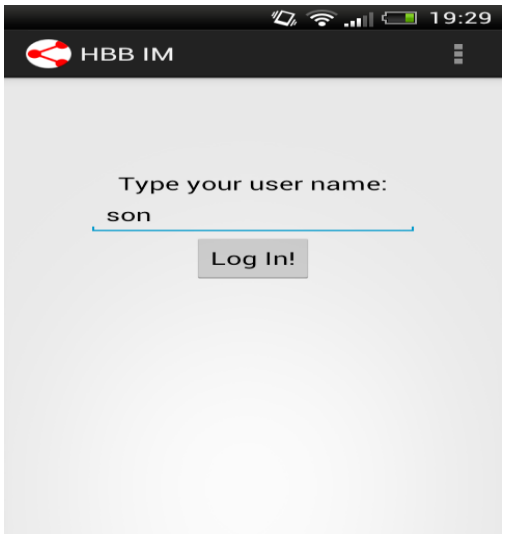
Prílohy

V tejto časti priloženého dokumentu sa nachádzajú jednotlivé prílohy. Medzi prílohy patrí používateľská príručka a systémová príručka.

Príloha A: Používateľská príručka Android IM aplikácie

V tejto časti obsahujúcej používateľskú príručku Android IM aplikácie sa venujeme jednoduchému návodu ako správne používať mobilnú aplikáciu HBB IM. Aplikácia bola navrhnutá a následne implementovaná tak aby nebolo potrebné pre používateľov študovať rozsiahle návody. Aj napriek intuitívnemu ovládaniu prikladáme k predloženej projektovej dokumentácii používateľskú prihlášku.

Po spustení aplikácie sa zobrazí obrazovka, kde si používateľ zadá svoje prihlasovacie meno podľa ktorého sa prihlási Obr. 0.1. Ak náhodou zadá zlé meno alebo sa stane niečo neočakávané, na obrazovke sa viditeľne zobrazí chyba. Po prihlásení sa používateľovi zobrazí obrazovka s kontaktmi a aj s informáciou či je daný kontakt online alebo offline Obr. 0.2. V pravom hornom rohu obrazovky sa používateľ môže stlačením možnosti Log Out odhlásiť a vráti sa do úvodnej obrazovky. Po krátkom stlačení nejakého z kontaktov sa používateľ prepne do chatovacej obrazovky Obr. 0.3, kde sa používateľovi zobrazujú správy od daného kontaktu a kde môže posielať správu danému kontaktu. Pri prijatí správy telefón krátko zavybruje. Rovnako ako pri predchádzajúcej obrazovke sa aj tu môže používateľ odhlásiť pomocou možnosti v pravom hornom rohu.



Obr. 0.1: Úvodná obrazovka pre prihlásenie používateľa

Obr. 0.2: Obrazovka zobrazujúca zoznam kontaktov

Obr. 0.3: Obrazovka zobrazujúca konverzáciu s používateľmi

Príloha B: Používateľská príručka IM Service

Pre správny beh program na server je potrebné mať na server nainštalovaný aplikačný server Apache Tomcat [16] aspoň verzie 6. V našom prípade sme použili linuxový stroj s distribúciou Ubuntu 11.10 (GNU/Linux 3.0.0-20-generic i686) a ako aplikačný server sme použili Apache Tomcat verzie 7.0.34.

V rámci tejto inštalačnej príručky nebudeme rozpisovať, ako je potrebné nainštalovať Apache Tomcat alebo ako ho integrovať s vývojovým prostredím Eclipse, nakoľko tieto postupy možno nájsť na internete napr. [17] a [18]. Namiesto toho si opíšeme, ako napr. nasadiť novú verziu na produkčný server, ktorý bol použitý v rámci nášho tímového projektu.

Adresárová štruktúra a nahrávanie zmien

Adresárová štruktúra pre nahrávanie skompilovaných binárnych súborov (JSP class):

/home/barlog/apps/apache-tomcat-7.0.34/webapps/SrvcM/WEB-INF/
./classes/
./lib/

Adresár „classes“ obsahuje adresárovú štruktúru skompilovaných súborov a adresár „lib“ obsahuje skompilované knižnice, ktoré sa používajú – v našom prípade ide iba o jednu knižnicu na vytváranie a prácu s JSON správami „json-simple-1.1.1.jar“ [19].

Vždy po zmene verzie je potrebné vypnúť Tomcat za pomoci scriptu „shutdown.sh“ a opätovne naštartovať za pomoci skriptu „startup.sh“. Tieto skripty sú súčasťou aplikačného servera a možno ich nájsť v adresári:
/home/barlog/apps/apache-tomcat-7.0.34/bin/

Príloha C:Používateľská príručka pre XMPP based IM

1. Spustenie aplikácie
2. Vyplnenie prihlasovacích údajov (medzi jednotlivými poľami sa prepíname šípkami)
3. Po zadaní prihlasovacích údajov stlačíme RETURN, aby sa skryli panely pre zadávanie textu.
4. Stlačíme červené tlačidlo pre prihlásenie.
5. Stlačíme zelené tlačidlo pre zobrazenie zoznamu prihlásených priateľov.
6. Šípkami si zvolíme priateľa, s ktorým chceme komunikovať a stlačíme OK.
7. Napíšeme správu a stlačíme OK.
8. V prípade, že chceme komunikovať s iným priateľom stlačíme RETURN a zvolíme si ho šípkami.
9. Pre odhlásenie stlačíme modré tlačidlo.

Príloha D:Používateľská príručka pre Samsung Smart TV IM

Na spustenie aplikácie potrebujeme televízor od spoločnosti Samsung s podporou Smart TV. Celý postup ako nahrať aplikáciu na TV je uvedený tu[12]. Po úspešnom nahraní našej aplikácie do TV nasledujú tieto kroky:

1. Spustenie aplikácie
2. Vyplnenie prihlasovacieho mena.

3. Po zadaní prihlasovacieho mena stlačíme RETURN, aby sa skryli panely pre zadávanie textu.

4. Stlačíme červené tlačidlo pre prihlásenie a zobrazenie prihlásených užívateľov.

5. Šípkami si zvolíme priateľa, s ktorým chceme komunikovať a presunieme sa naspäť na písanie správy.

6. Napíšeme správu a stlačíme OK.

7. V prípade, že chceme komunikovať s iným priateľom stlačíme RETURN a zvolíme si ho šípkami.

8. Pre odhlásenie stlačíme zelené.