



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**PROPOJENÍ SOCIÁLNÍ SÍTĚ TWITTER S TELEVIZNÍM
VYSÍLÁNÍM**

TWITTER CONNECTION WITH A TV BROADCAST

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MAREK FIALA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. RADEK BURGET, Ph.D.

BRNO 2018

Abstrakt

Tato diplomová práce se zabývá možností propojení digitálního televizního vysílání DVB se sociální sítí Twitter. Cílovou platformou je Hybridní televizní platforma HbbTV umožňující kombinaci televizního vysílání s daty získanými pomocí širokopásmového internetu. Vytvořený systém je složen z HbbTV aplikace a serveru, který slouží ke spojení se sociální sítí Twitter a získání dodatečných informací o právě běžícím pořadu. Nasazením výsledného systému v reálném vysílání by mělo potenciálně dojít ke zvýšení počtu tweetů (příspěvků) souvisejících s televizním vysíláním, zvýšení povědomí o technologii HbbTV a oslovení mladší generace diváků. V konečném důsledku vše může vést i k mírnému nárůstu sledovanosti.

Abstract

This master thesis focuses on the possible connection between the digital television broadcasting DVB and the Twitter social network. The target platform is the Hybrid Broadcast Broadband TV platform, which combines television broadcasting with a data received from broadband. The created system is composed of a HbbTV application and a server, that connects the application with the Twitter and searches for additional data about the current video content. Usage of the resulting solution in real television broadcasting could potentially increase the amount of tweets related to the television broadcasting, increase the knowledge about HbbTV technology and attract young generation of viewers. All that can result into slight increased number of viewers.

Klíčová slova

HbbTV, Twitter, DVB, televizní vysílání, sociální sítě, TF-IDF.

Keywords

HbbTV, Twitter, DVB, Video Broadcasting, social network, TF-IDF.

Citace

FIALA, Marek. *Propojení sociální sítě Twitter s televizním vysíláním*. Brno, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Radek Burget, Ph.D.

Propojení sociální sítě Twitter s televizním vysíláním

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana doktora Radka Burgeta. S uvedením do kontextu a spuštěním vzorové aplikace mi pomohli kolegové z firmy Mautilus s.r.o. inženýr Martin Heneš a Petr Mazanec. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Marek Fiala
20. května 2018

Poděkování

Chtěl bych poděkovat mému vedoucímu doktoru Radku Burgetovi za vedení mé práce a dále také mým kolegům z firmy Mautilus s.r.o. inženýru Martinu Henešovi a Petru Mazancovi, kteří jsou autoři myšlenky této práce a kteří mi svými radami pomohli při řešení programové části.

Obsah

1	Úvod	3
2	Digitální televizní vysílání a HbbTV	4
2.1	DVB	5
2.2	Systém HbbTV	6
2.2.1	Cíl platformy HbbTV	6
2.2.2	Architektura HbbTV	6
2.2.3	Terminál HbbTV 2.0	8
2.2.4	Standard HbbTV	10
2.3	HbbTV Aplikace	11
2.3.1	Rozdělení	11
2.3.2	Ovládání	11
2.3.3	Poskytnuté API	11
2.3.4	Rozlišení aplikace	12
3	Servery třetích stran a jejich aplikační rozhraní	13
3.1	Sociální síť Twitter	13
3.1.1	Autentizace	14
3.1.2	Vyhledávání	15
3.1.3	Vytvoření nového tweetu, retweet a přidání do oblíbených	16
3.2	Servery s dodatečnými metadaty o multimediálním obsahu	17
3.2.1	The Movie Database (TMDb)	17
3.2.2	The Open Movie Database (OMDb)	18
3.2.3	Porovnání dat získaných ze serverů třetích stran	20
4	Systém pro získávání vysílání relevantního obsahu a hashtagů	21
4.1	Návrh systému	21
4.2	Architektura systému	23
5	Získávání relevantního obsahu a hashtagů	25
5.1	Získání dodatečných informací o pořadu	25
5.2	Vyhledávání vhodných příspěvků na sociální síti Twitter	29
5.3	Vyhodnocení relevantnosti příspěvků	31
5.4	Vyhodnocení vhodných hashtagů	31
5.5	Metody pro hodnocení relevantnosti tweetů	32
5.5.1	Porovnání různých variant metody TF	33
6	Implementace	35

6.1	Server	35
6.1.1	Blok pro zpracování požadavků	35
6.1.2	Blok pro získávání dodatečných informací o médiu	37
6.1.3	Blok pro interakci se sociální sítí Twitter	37
6.2	HbbTV aplikace	39
6.2.1	Scéna pro přihlášení	39
6.2.2	Scéna s příspěvky	40
6.2.3	Scéna pro vytvoření nového příspěvku	41
6.2.4	Scéna nastavení a dialogy	42
6.2.5	Implementační detaily HbbTV aplikace	44
7	Testování	47
7.1	FireHbbTV	47
7.2	Testování na reálných zařízeních	49
7.2.1	Referenční zařízení	52
8	Závěr	54
	Literatura	55
	A Obsah přiloženého paměťového média	57
	B Spuštění HbbTV aplikace a serveru	58
	C Autorizace pomocí PINu na sociální síti Twitter	59

Kapitola 1

Úvod

Cílem této práce je navrhnout a realizovat řešení umožňující propojení digitálního televizního vysílání DVB se sociální sítí Twitter. Platformou zvolenou k tomuto účelu je hybridní televizní platforma HbbTV, která umožňuje kombinaci televizního vysílání a dat získaných pomocí širokopásmového internetu. Motivací ke vzniku této práce je snaha o zvýšení počtu příspěvků (tzv. *tweetů*) souvisejících s televizním vysíláním.

Potenciálním nasazením výsledného systému do reálného vysílání by mělo dojít ke zvýšení povědomí televizních diváků o technologii HbbTV a také nárůstu využívání HbbTV aplikací. Rovněž by možnost propojení se sociální sítí mohla oslovit mladší generace diváků a zvýšit jejich loajalitu. V konečném důsledku by se vše mohlo projevit mírným nárůstem sledovanosti.

Nejdůležitějším požadavkem pro výslednou aplikaci je získání vhodných předmětů zájmu (tzv. *hashtagů*) a odpovídajících příspěvků ze sociální sítě Twitter, které budou souviset s právě probíhajícím pořadem. Aplikace by rovněž měla poskytnout uživateli možnost vložením vlastního příspěvku se schopností přidat vygenerované hashtagy.

Kapitola 2

Digitální televizní vysílání a HbbTV

Televize je pro většinu z nás v současnosti pouze zařízení umožňující zobrazovat televizní vysílání ve formě audiovizuálního obsahu pro televizní programy, případně pouze přehrávat zvukový obsah pro rádiové stanice. Tento typ obsahu je nejčastěji přenášen pomocí standardu digitálního televizního vysílání (DVB), který je přiblížen v podkapitole 2.1. V minulosti se používalo analogové televizní vysílání, které je nyní zastaralé a pro tento účel se již nepoužívá.

S příchodem Internetu do světa televizí se však tento pohled začal pomalu měnit. První využití tohoto média lze vidět u takzvaných Smart televizí, které umožňovaly uživateli spouštět různé aplikace (widgety) například pro sledování archivů televizních společností, hraní her nebo prohlížení Internetu. Widgety bylo možné stáhnout na televizní přijímač ze serverů výrobce konkrétního zařízení, tím však nebyla zajištěna kompatibilita se zařízeními jiných značek.

Právě tato nekompatibilita vedla ke vzniku zcela nové nezávislé televizní platformy HbbTV (viz podkapitola 2.2), která navíc umožnila kombinaci digitálního televizního vysílání s širokopásmovým internetem. Aplikace pro tuto televizní platformu se souhrnně označují jako HbbTV Aplikace (popsány v podkapitole 2.3).

2.1 DVB

DVB (Digital Video Broadcasting) je systém digitálního televizního vysílání, který zahrnuje následující varianty:

- DVB-C (Cable)- kabelové digitální televizní vysílání
- DVB-H (Handheld)- digitální televizní vysílání pro mobilní zařízení
- DVB-T (Terrestrial)- pozemní digitální televizní vysílání
- DVB-S (Satellite)- družicové digitální televizní vysílání
- DVB-SH (Satellite handheld)- družicové digitální televizní vysílání pro mobilní zařízení

U některých z těchto variant je možné se setkat rovněž s druhou generací (například druhá generace DVB-T se označuje jako DVB-T2).

V systému DVB je televizní obraz a jeho zvukový doprovod zdigitalizován a spolu s přidruženými datovými službami (například Teletext) přenášen společným datovým kanálem v podobě tzv. digitálního multiplexu. Zmíněný multiplex může obsahovat několik televizních programů (v DVB-T obvykle 4-6 programů). Dále může obsahovat také sadu rozhlasových programů a nebo další doplňkové služby (například EPG). [15]

2.2 Systém HbbTV

Platforma Hybridní televize HbbTV (Hybrid broadcast broadband TV) je systém kombinující televizní vysílání s širokopásmovým internetem.

Tato platforma byla poprvé představena veřejnosti v roce 2009 během francouzských veletrhů IBC a IFA. Následující rok došlo ke schválení specifikace pro platformu HbbTV¹. Na schválení normy se podílela řada evropských organizací jako např. Evropský ústav pro telekomunikační normy (ETSI), Evropská vysílací unie (EBU) nebo Evropský výbor pro elektrotechnickou normalizaci (CENELEC). Následně vzniklo také stejnojmenné konsorcium sdružující některé renomované výrobce televizních přijímačů a televizních společností². [8]

2.2.1 Cíl platformy HbbTV

Cílem platformy HbbTV je sjednotit výrobce přijímačů a vysílačů, a tímto zajistit univerzálnost použití HbbTV aplikací.

Tuto univerzálnost má zajistit logo HbbTV (obrázek 2.1). Pokud výrobek nese toto logo, může si být uživatel jistý, že všechny HbbTV aplikace budou na tomto zařízení fungovat. [8]



Obrázek 2.1: Logo HbbTV
Zdroj: [7]

2.2.2 Architektura HbbTV

Architekturu platformy HbbTV s hybridním terminálem, který umožňuje příjem televizního vysílání (Broadcast) a současné připojení se širokopásmovým internetem (Broadband), lze vidět na obrázku 2.2.

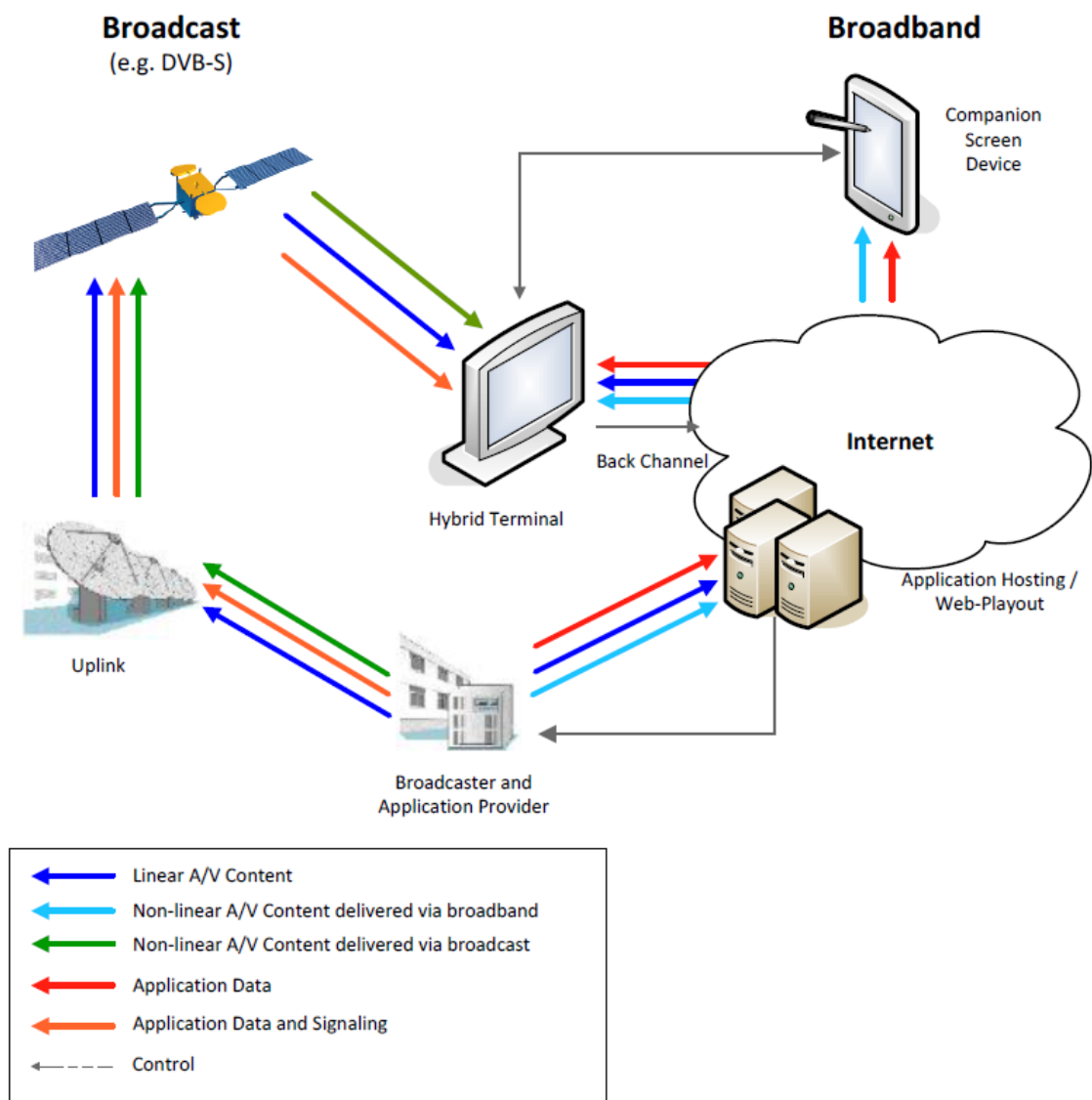
Terminál je schopen být paralelně připojen k jednomu nebo více platformám digitálního vysílání DVB (DVB-T, DVB-S nebo DVB-C), nebo může komunikovat obousměrným kanálem s širokopásmovou sítí internet.

Pomocí připojení k DVB může zařízení přijímat standardní audiovizuální obsah (lineární audiovizuální obsah) přenášený v reálném čase spolu s aplikačními daty a signalizačními daty aplikací HbbTV. I v případě, že terminál není připojen k širokopásmovému internetu, může přijímat aplikace závislé na televizním vysílání.

Připojení k síti internet umožňuje přijímat aplikační data a také nelineární audiovizuální obsah. Pro HbbTV 2.0. je dále možné komunikovat přes toto připojení s dalšími zařízeními (například tablety nebo mobilními telefony) - těmto zařízením se v terminologii HbbTV říká *Přidružené obrazovky* (Companion Screens), či s jinými terminály HbbTV ve stejné lokální síti. Komunikací s dalším zařízením (nejedná se o další terminály HbbTV) je uživateli umožněna další interakce s HbbTV aplikací (například je zde prostor pro informace o aktuálním filmu, kvízové hry apod.). [7]

¹Pod označením ETSI TS 102 796 V. 1.1.1

²Astra, Canal+, France Télévisions, Open TV, TF1, Philips, Humax, Opera Software a další



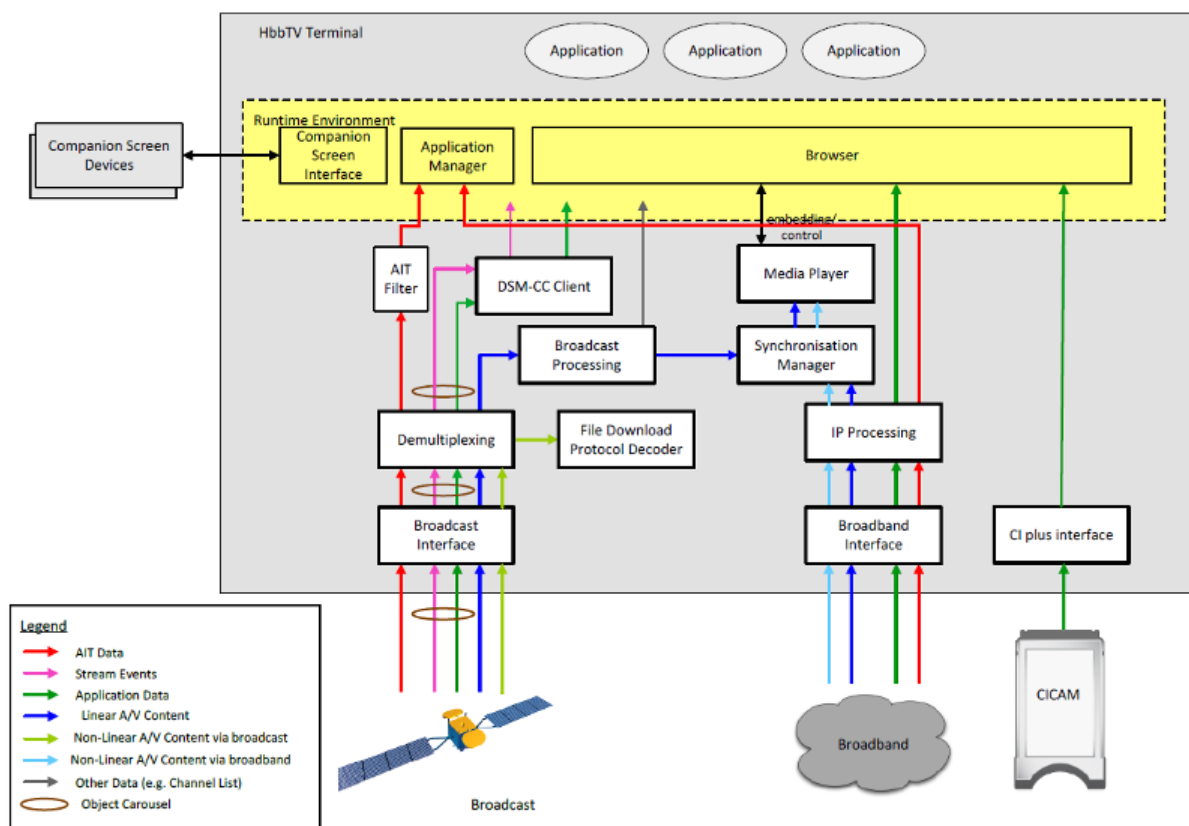
Obrázek 2.2: Architektura systému HbbTV 2.0.³

Zdroj: [7]

³Pro systém HbbTV 1.0 je schéma velmi podobné, avšak se zde nenachází propojení s přidruženými obrazovkami.

2.2.3 Terminál HbbTV 2.0

Obrázek 2.3 znázorňuje schéma funkčních komponent terminálu HbbTV 2.0.



Obrázek 2.3: Blokové schéma hybridního terminálu.

Zdroj: [7]

Na tomto obrázku lze vidět, že terminál přijímá prostřednictvím kanálu pro televizní vysílání data z tabulky AIT (Application Information Table), lineární audiovizuální obsah, aplikační data a také data událostí. Poslední dva datové toky jsou přenášeny v objektu karuselu DSM-CC (Digital Storage Media - Command and Control). Nelineární obsah je přijat *dekoderem pro stahování souborů* (File Download Protocol Decoder). V bloku *DSM-CC klient* (DSM-CC Client) se obnovují data z objektu karusel a dále jsou poskytnuta do *běžového prostředí* (Runtime Environment). Běžové prostředí lze vnímat jako abstraktní komponentu, ve které je aplikace vykonávána a prezentována. Toto prostředí je tvořeno *prohlížečem* (Browser) a *aplikačním manažerem* (Application Manager). Aplikační manažer vyhodnocuje data z AIT a řídí životní cyklus interaktivní aplikace. Prohlížeč slouží pro zobrazení a vykonání interaktivní aplikace. [7] [8]

Lineární audiovizuální obsah je zpracováván stejně jako u standardních DVB přijímačů. Zpracovávání audiovizuálního obsahu je prováděno v *bloku pro zpracování televizního vysílání* (Broadcast Processing). Některé informace a funkce z bloku pro zpracování televizního vysílání mohou být přístupné z běžového prostředí. Těmito daty (v obrázku označena jako jiná data) mohou být například: seznam televizních kanálů, informace o běžících a následujících událostech nebo funkce ladění. Lineární obsah lze dále škálovat a následně vkládat

do interaktivní aplikace. Funkce popsané v tomto odstavci zajišťuje blok *přehrávač médií* (Media Player), který slouží k veškerému zpracování audiovizuálního obsahu. [7] [8]

Připojení k širokopásmovému internetu je druhou možností, jak lze získat data ze serverů poskytovatele interaktivní aplikace. Toto rozhraní lze rovněž použít pro získání audiovizuálního obsahu (například obsah vyžadovaný aplikací). *Blok pro zpracování internetového protokolu* (Internet Protocol Processing) zpracovává veškerá data přicházející z internetu. Prostřednictvím tohoto bloku jsou aplikační data poskytnuta běhovému prostředí. Audiovizuální obsah je předán přehrávači médií, který je ovládán běhovým prostředím a může vložit audiovizuální obsah do interaktivní aplikace. Společně s přehrávačem medií může *synchronizační manažer* (Synchronisation Manager) synchronizovat obsah získaný z připojení k širokopásmovému internetu s obsahem získaným z připojení k televiznímu vysílání. [7] [8]

Rozhraní přidružených obrazovek (Companion Screen Interface) umožňuje hybridnímu terminálu nalézt přidružené zařízení a další hybridní terminály. Dále také ostatním hybridním terminálům umožňuje nalezení tohoto zařízení. Prohlížeč s využitím rozhraní přidružených obrazovek může komunikovat a spouštět aplikace na přidruženém zařízení a rovněž z přidruženého zařízení je možno prohlížeči zasílat žádosti o spuštění interaktivní aplikace. Spojení mezi zařízeními je realizováno pomocí Websocketového serveru. Rozhraní přidružených obrazovek v kombinaci s přehrávačem medií zpřístupňuje možnost synchronizace obsahu mezi hybridním terminálem a přidruženým zařízením, nebo mezi dvěma hybridními terminály. [7]

Skrz *rozhraní CI plus* může hybridní terminál požadovat aplikační data z pomocného souborového systému (Auxiliary File System) poskytovaná CICAM (Common Interface Conditional Access Module). [7]

Toto schéma je podobné i pro terminály HbbTV 1.0 a HbbTV 1.5, avšak pro obě varianty chybí rozhraní přidružených obrazovek a pro HbbTV 1.0 dále také rozhraní CI plus.

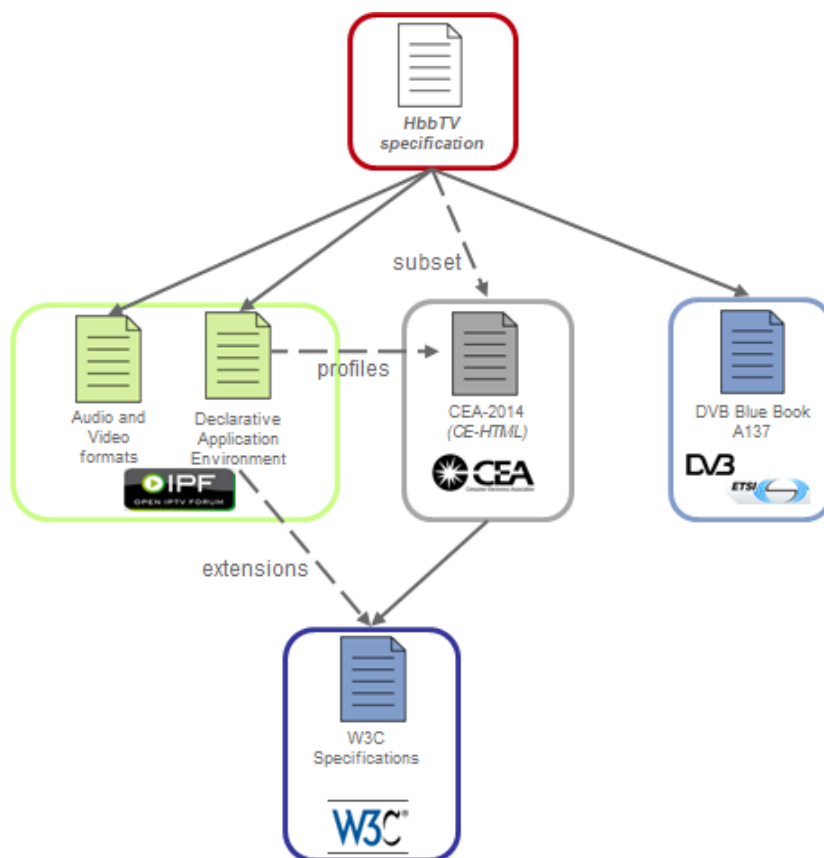
2.2.4 Standard HbbTV

Obrázek 2.4 zobrazuje hlavní technické specifikace a jejich závislost, které jsou použity a odkazovány v HbbTV. HbbTV přebírá velkou část specifikace z Open IPTV Forum (OIPF). OIPF vytvořilo podmnožinu CE-HTML (standard založený na XHTML, který slouží k vytváření webových stránek pro vzdálené uživatelské rozhraní v Universal Plug and Play sítích a Internetu), přidalo další API (například API pro změny kanálu a ladění) a definovalo vkládání lineárního audiovizuálního obsahu do aplikace.

Z CEA HbbTV využívá API pro vkládání nelineárního audiovizuálního obsahu do aplikace, zpracování událostí (například stisky kláves) a specifikace statických obrazových formátů.

Základní prvky pro vývoj aplikací jako XHTML, CSS, DOM (Document Object Model) a ECMAScript jsou ze standardu pro společný web W3C.

Jako standard pro audiovizuální vysílání byla zvolena podmnožina standardů definovaných DVB pro DVB terminály. Z DVB je využita signalizace aplikace, transportu aplikace přes DVB (DSM-CC) a událostí toku. [1] [7]



Obrázek 2.4: Závislosti standardů HbbTV.

Zdroj: [6]

2.3 HbbTV Aplikace

HbbTV Aplikace je kolekce dokumentů HTML, JavaScript, CSS, XML a multimediálních dat. V sekci 2.3.1 se nachází rozdělení aplikací podle přístupu k datům, sekce 2.3.2 se zabývá možnostmi ovládání podle standardu, v sekci 2.3.3 lze nalézt informace o terminálech poskytnutých API a v poslední části 2.3.4 lze nalézt doporučené rozlišení všech HbbTV aplikací.

2.3.1 Rozdělení

Podle standardu HbbTV [6] lze aplikace rozdělit na dva typy:

- *Aplikace nezávislé na televizním vysílání* (Broadcast-independent applications) - tyto aplikace jsou staženy přes širokopásmový internet a mají přístup ke všem datům z tohoto kanálu. Aplikace tohoto typu nejsou spojeny s žádnou službou televizního vysílání.
- *Aplikace závislé na televizním vysílání* (Broadcast-related applications) - tyto aplikace mohou být spuštěny automaticky (tzv. autostart), nebo na žádost uživatele. Tento typ aplikací může být získán buď přes širokopásmový internet, nebo DVB a může získávat data z obou kanálů. Aplikace tohoto typu jsou spojeny s jednou nebo více službami televizního vysílání.

2.3.2 Ovládání

Každou aplikaci je možno ovládat pomocí dálkového ovladače. Konkrétně lze odchytit události pro následující klávesy [7]:

- barevná tlačítka (červené, zelené, žluté a modré)
- směrová tlačítka (UP, DOWN, LEFT, RIGHT)
- klávesa pro potvrzení (ENTER nebo OK)
- klávesa pro návrat (BACK)
- číselné klávesy (0-9)
- tlačítka pro spuštění, pozastavení a zastavení mediálního obsahu (PLAY, PAUSE a STOP)
- klávesy pro převíjení multimediálního obsahu vpřed a vzad (FAST FORWARD a FAST BACKWARD)

V HbbTV 2.0 je dále možno ovládat aplikaci pomocí klávesnice a myši, herního kontrolleru nebo pomocí Přidružené obrazovky.

2.3.3 Poskytnuté API

Standard OIPF [12] poskytuje celou řadu užitečných API, které jsou rovněž součástí standardu HbbTV. Přistupovat ke každému z dostupných API je možno pomocí speciálních objektů. Každé API má svůj vlastní API objekt, který lze získat pomocí značky `object`

s nastavením hodnoty atributu `type` na název konkrétního objektu. Element s daným objektem se musí nacházet v každém HTML kódu stránky, která dané API používá.

Nejdůležitějším API objektem pro každou aplikaci je objekt `application/oipfApplicationManager`, který umožňuje přístup k tzv. *API pro správu aplikace*. Tento objekt řídí například viditelnost aplikace nebo specifikuje to, jaké klávesy se mají do aplikace zasílat.

Díky objektu `application/oipfConfiguration` z *Konfiguračního API* lze přistupovat ke konfiguracím a nastavením uživatele a pro správný chod musí být rovněž součástí aplikace.

Pro získání dat o konkrétním programu lze použít objekt `application/oipfSearchManager`, který komunikuje s tzv. *Metadata API* a který umožňuje komplexní vyhledávání v metadatech. Pomocí tohoto objektu lze získat veškeré dostupné informace o programu, které jsou v hybridním terminálu dostupné. Konkrétně se může jednat například o popis pořadu, čas začátku či konce nebo také informace pro rodiče, zda je daný pořad vhodný pro děti či nikoliv.

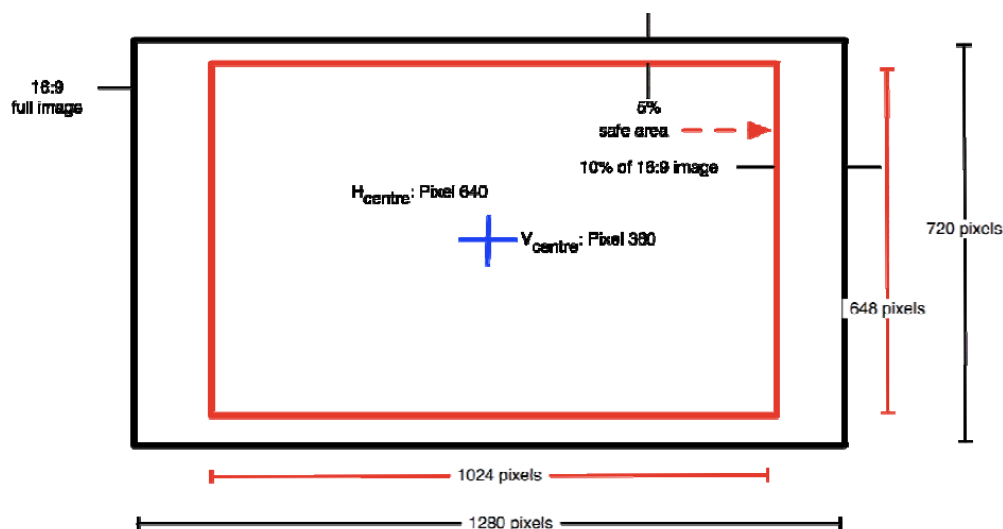
Posledním zde představeným API objektem je objekt `video/broadcast` pro komunikaci s tzv. *API pro plánovaný obsah a hybridní tuner*. Využitím tohoto objektu v aplikaci lze získat například informace o aktuálně zobrazeném kanálu, zobrazit data z jiného kanálu nebo také získat informace o pořadech pro daný kanál.

Seznam všech dostupných API lze nalézt například v [12].

2.3.4 Rozlišení aplikace

Dle standardu OIPF [12] by každá aplikace měla být v rozlišení **1280x720** pixelů, tedy 1280 pixelů na šířku a 720 pixelů na výšku, s poměrem stran 16:9. Minimální rozlišení terminálů musí být tedy 1280x720, pokud je rozlišení zařízení větší, aplikace se automaticky zobrazí s rozlišením 1280x720.

V standardu OIPF [12] je rovněž specifikována tzv. *Bezpečná zóna* (Graphic safe area), která určuje doporučenou zónu pro veškerý zobrazený obsah. Přesné rozměry této bezpečné zóny lze vidět na obrázku 2.5.



Obrázek 2.5: Bezpečná zóna. Zdroj: [12]

Kapitola 3

Servery třetích stran a jejich aplikační rozhraní

Tato kapitola se zabývá servery třetích stran a jejich aplikačními rozhraními. Konkrétně se jedná o API Twitteru (viz podkapitola 3.1), které je určené pro oboustrannou komunikaci mezi aplikací a sociální sítí Twitter. Dále také API serverů, ze kterých je možné získat dodatečná metadata o multimediálním obsahu. V současnosti se jedná o servery The Movie Database [2] a The Open Movie Database [3] (viz podkapitola 3.2).

3.1 Sociální síť Twitter

Twitter je sociální síť, která umožňuje uživateli číst příspěvky jiných uživatelů a dále také vytvářet nové. Těmto příspěvkům se říká *tweety*. Maximální délka tweetu byla původně omezena na 140 znaků, ale v současnosti je již možné vytvářet příspěvky o maximální délce 280 znaků.

Velice důležitou součástí sociální sítě Twitter jsou předměty zájmu, kterým se říká *hashtagy*. Hashtag je slovo nebo fráze, začínající znakem #¹, které vyjadřuje téma daného příspěvku a díky kterému je možné dohledat další příspěvky související s daným tématem. *Retweet* je pojem označující znovusdílení tweetu jiným uživatelem. Proces, který je opakem retweetu, se nazývá *unretweet* a jeho výsledkem je odstranění sdílení tweetu.



Obrázek 3.1: Logo Twitteru.

Zdroj: [16]

¹Díky tomuto znaku se předmětům zájmu říká hashtagy - anglické slovo pro dvojkřížek je hashtag.

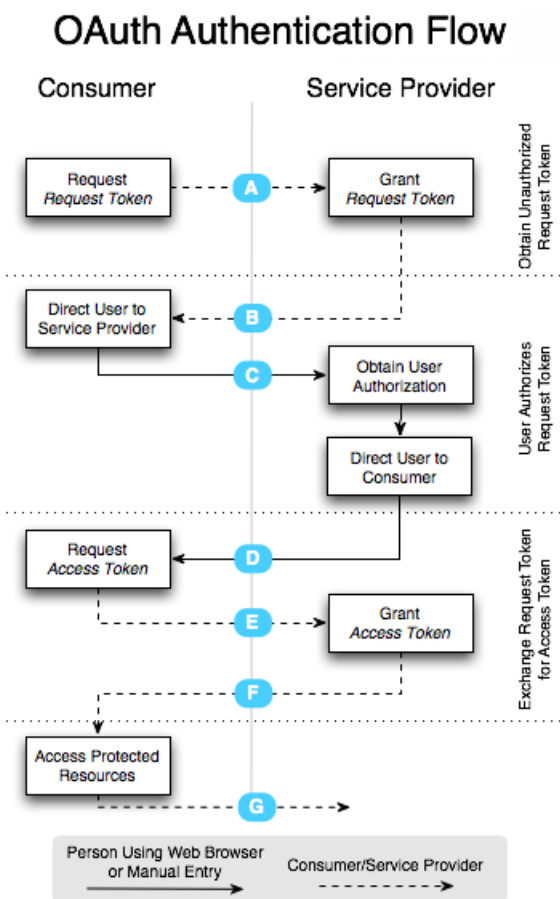
V sekci 3.1.1 je vysvětlen způsob přihlášení k této sociální síti, sekce 3.1.2 a 3.1.3 obsahují informace o použití Twitter API pro vyhledávání, vytváření nového tweetu, retweet a přidání do oblíbených.

3.1.1 Autentizace

K autentizaci uživatele k sociální síti Twitter je využita *metoda založená na PINu* (PIN-based authorization). Tato metoda je určena pro externí aplikace, které nemají přímý přístup k webovému prohlížeči. Samotná metoda využívá protokol *OAuth*. [16]

Protokol OAuth

Protokol OAuth je navržen pro bezpečnou API autentizaci jednoduchým a jednotným způsobem pro webové, mobilní i desktopové aplikace. Na obrázku 3.2 lze vidět schéma autentizace pomocí tohoto protokolu.



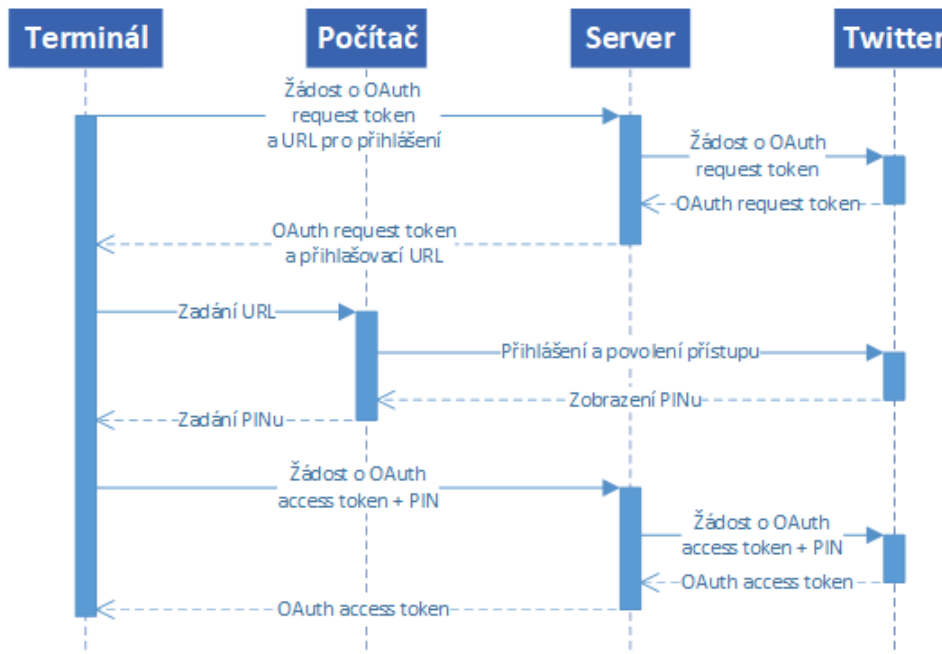
Obrázek 3.2: Schéma autentizace pomocí protokolu OAuth.

Zdroj: [9]

V případě použití protokolu OAuth uživatel nezadá své přístupové údaje k nějaké službě (například Twitter) přímo do externí aplikace, ale je odkázán na URL na serveru dané služby. Zde se klasickým způsobem přihlásí a dále potvrdí, že daná externí aplikace smí

přistupovat k jeho účtu. Následně je externí aplikaci předán klíč, který jí umožňuje přístup k potřebným údajům. Výhodou tohoto způsobu je, že citlivé přihlašovací údaje putují jen mezi uživatelem a cílovou službou. [9]

Schéma autentizace k sociální síti Twitter je zobrazeno na obrázku 3.3.



Obrázek 3.3: Schéma autentizace k sociální síti Twitter.

3.1.2 Vyhledávání

API Twitteru podporuje následující vyhledávací operátory, které jsou potenciálně užitečné pro účely této práce²: [16]

Operátory	Nalezené Tweety obsahující...
sledovat teď	„sledovat“ a „teď“. Výchozí operátor.
"sledovat teď"	slovní spojení „sledovat teď“
dobro OR zlo	buď „dobro“ nebo „zlo“, nebo obě.
dobro -zlo	„dobro“ ale ne „zlo“.
#haiku	hashtag „haiku“.
film :)	„film“ s pozitivním přístupem.
film :(„film“ s negativním přístupem.
film -filter:retweets	„film“ a dále jsou odfiltrovány všechny retweety.

Tabulka 3.1: Vyhledávací operátory podporované Twitter API.

Všechny dotazy musí být kódovány pomocí URL kódování a nastaveny jako hodnota parametru q.

²Seznam všech dostupných operátorů je možné nalézt na stránkách Twitter Developer [16].

Twitter API rovněž podporuje tyto rozšiřující parametry³: [16]

Parametr	Popis
count	Počet vrácených tweetů na jedné stránce (maximální počet je 100).
lang	Omezuje jazyk vyhledávaných tweetů, specifikováno dle kódu ISO 639-1 code.
tweet_mode	Parametr určující, zda se mají vrácené tweety ořezávat na 140 znaků (výchozí chování) nebo zda se mají ponechat s původním textem - parametr je třeba nastavit na hodnotu <code>extended</code> .

Tabulka 3.2: Rozšiřující parametry podporované Twitter API.

URL serveru pro získání odpovídajících tweetu k danému dotazu je <https://api.twitter.com/1.1/search/tweets.json>. Pro vyhledávání je potřeba použít HTTP metodu GET.

Vyhledávání je ze strany Twitteru limitováno na 180 volání za 15 minut pro daného aktivního uživatele.

3.1.3 Vytvoření nového tweetu, retweet a přidání do oblíbených

Pro vytvoření nového tweetu je třeba pouze požadovaný text zakódovat pomocí URL kódování, nastavit jako parametr `status` a poslat na URL serveru <https://api.twitter.com/1.1/statuses/update.json>.

Pro retweet a unretweet stačí zaslat požadavek na URL <https://api.twitter.com/1.1/statuses/retweet/:id.json>, resp. <https://api.twitter.com/1.1/statuses/unretweet/:id.json>, kde `:id` musí být nahrazeno id odpovídajícího tweetu.

Přidání tweetu do oblíbených je realizováno pomocí zaslání požadavku na URL <https://api.twitter.com/1.1/favorites/create.json> s parametrem `id` nastaveného na identifikační číslo požadovaného tweetu. Pro odebrání tweetu z oblíbených je požadavek podobný jako pro přidání, jen URL je <https://api.twitter.com/1.1/favorites/destroy.json>

Všechny volání z této sekce musí použít HTTP metodu POST.

³Jedná se o seznam použitých parametrů v této práci. Seznam všech dostupných operátorů lze nalézt na stránkách Twitter Developer [16].

3.2 Servery s dodatečnými metadaty o multimediálním obsahu

Z EIT tabulky lze získat velké množství užitečných informací o médiu, avšak většinou se jedná spíše o technické informace jako například název, popis či začátek a konec programu. Tyto údaje nejsou pro uživatele ani pro vyhledávání na síti Twitter vhodné a tedy je třeba získat rozšiřující informace ze serverů třetích stran s dodatečnými metadaty o multimediálním obsahu. V současnosti se jedná o servery The Movie Database (viz sekce 3.2.1) a The Open Movie Database (další informace v sekci 3.2.2).

3.2.1 The Movie Database (TMDb)

The Movie Database (TMDb) je televizní a filmová databáze založená a spravovaná komunitou. API serveru je použitelné jak pro nekomerční tak i pro komerční použití. Používání API je zcela zdarma, ale provozovatel si vyhraduje právo zpoplatnit API pro komerční užití v budoucnu. Limitem tohoto API je maximální množství žádostí v počtu 40 každých 10 sekund a nutnost vygenerování osobního API klíče pro přístup. Velkou výhodou tohoto serveru je možnost vyhledávání názvu v mnoha světových jazycích. [2]



Obrázek 3.4: Logo TMDb. Zdroj [2].

Nejběžnější způsob použití tohoto API je nejdříve vyhledat film, TV show nebo osobu. Například pro film Jack Reacher takto:

```
https://api.themoviedb.org/3/search/movie?api_key={api_key}&query=Jack+Reacher
```

kde {api_key} musí být přepsán na vygenerovaný osobní API klíč.

Tento příkaz vrátí několik položek, z nichž je pro vyhledávání nejužitečnější položka `results`, která obsahuje seznam objektů s informacemi o filmech. Příklad objektu s informacemi o filmu může být:

```
{
  "poster_path": "/IfB9hy4JH1eH6HEfIgIGORXi5h.jpg",
  "adult": false,
  "overview": "Jack Reacher must uncover the truth behind a major government
    conspiracy in order to clear his name. On the run as a fugitive from the law,
    Reacher uncovers a potential secret from his past that could change
    his life forever.",
  "release_date": "2016-10-19",
  "genre_ids": [53, 28, 80, 18, 9648],
  "id": 343611,
  "original_title": "Jack Reacher: Never Go Back",
  "original_language": "en",
  "title": "Jack Reacher: Never Go Back",
  "backdrop_path": "/4ynQYtSEuU5hyipcGkfD6ncwtwz.jpg",
  "popularity": 26.818468, "vote_count": 201, "video": false, "vote_average": 4.19
}
```

Z výsledku vyhledávání je získán identifikátor id, například pro film výše je to 343611, a následně sestaven dotaz na získání detailu:

```
https://api.themoviedb.org/3/movie/343611?api_key={api_key}
```

Odpovědí na tento dotaz je seznam obsahující podrobné informace o daném filmu:

```
{
  "adult": false,
  "backdrop_path": "\nDS8rddEK74HfAwCC5CoT6Cwzlt.jpg",
  "belongs_to_collection": {
    "id": 403374, "poster_path": "\/7baSuTFki8PQ9SL06ECYBfAW2K8.jpg",
    "name": "Jack Reacher Collection", "backdrop_path": "\/vViRXFnSyGJ2fzMbcc5sqTKswcd.jpg"
  },
  "budget": 60000000,
  "genres": [{ "id": 28, "name": "Action" }],
  "homepage": "http://www.jackreachermovie.com/",
  "id": 343611,
  "imdb_id": "tt3393786",
  "original_language": "en",
  "original_title": "Jack Reacher: Never Go Back",
  "overview": "Jack Reacher must uncover the truth behind a major government conspiracy in order to clear his name. On the run as a fugitive from the law, Reacher uncovers a potential secret from his past that could change his life forever.",
  "popularity": 13.584471,
  "poster_path": "\/IfB9hy4JH1eH6HEfIgIGORXi5h.jpg",
  "production_companies":
    [{"id": 21777, "logo_path": null, "name": "TC Productions", "origin_country": ""}],
  "production_countries": [{"iso_3166_1": "US", "name": "United States of America"}],
  "release_date": "2016-10-19",
  "revenue": 162146076,
  "runtime": 118,
  "spoken_languages": [{"iso_639_1": "en", "name": "English"}],
  "status": "Released",
  "tagline": "Justice is Coming.",
  "title": "Jack Reacher: Never Go Back",
  "video": false,
  "vote_average": 5.4,
  "vote_count": 2226
}
```

Server dále podporuje speciální parametr `append_to_response`, který umožňuje rozšířit dotaz na detail o další pod-dotazy. Například dotazy

```
https://api.themoviedb.org/3/movie/157336?api_key={api_key}
https://api.themoviedb.org/3/movie/157336/videos?api_key={api_key}
```

lze pomocí parametr `append_to_response` spojit do jednoho požadavku:

```
https://api.themoviedb.org/3/movie/157336?api_key={api_key}
&append_to_response=videos
```

3.2.2 The Open Movie Database (OMDb)

The OMDb API je RESTful webová služba, která slouží k získávání informací o filmech. Veškerý obsah na tomto severu je a spravován samotnými uživateli. Před použitím API je potřeba vygenerovat osobní API klíč pro přístup. [3]

API poskytuje pouze dva typy požadavků, a to vyhledávání podle názvu a získání informací podle jména nebo IMDb identifikátoru.

Požadavek na vyhledávání může vypadat například takto:

```
http://www.omdbapi.com/?apikey={api_key}&s=jack%20reacher
```

kde {api_key} musí být přepsán na vygenerovaný osobní API klíč.

Odpovědí na tento dotaz je:

```
{
  "Search": [
    {
      "Title": "Jack Reacher",
      "Year": "2012",
      "imdbID": "tt0790724",
      "Type": "movie",
      "Poster": "https://images-na.ssl-images-amazon.com/images/M/MV5BMTM1NjUxMDI3OV5BM15BanBnXkFtZTcwNjg1ODM3OA@@._V1_SX300.jpg"
    },
    {
      "Title": "Jack Reacher: Never Go Back",
      "Year": "2016",
      "imdbID": "tt3393786",
      "Type": "movie",
      "Poster": "https://images-na.ssl-images-amazon.com/images/M/MV5BODQ3ODQ3NDI4NV5BM15BanBnXkFtZTgwMDY1Mzk5OTE@._V1_SX300.jpg"
    }
  ],
  "totalResults": "2",
  "Response": "True"
}
```

Požadavek na získání informací podle IMDB id může vypadat například takto:

```
http://www.omdbapi.com/?apikey={api_key}&i=tt0790724
```

Odpovědí je pak:

```
{
  "Title": "Jack Reacher", "Year": "2012",
  "Rated": "PG-13", "Released": "21 Dec 2012",
  "Runtime": "130 min", "Genre": "Action, Crime, Mystery",
  "Director": "Christopher McQuarrie",
  "Writer": "Lee Child (based on the book \"One Shot\" by),
    Christopher McQuarrie (written for the screen by)",
  "Actors": "Tom Cruise, Rosamund Pike, Richard Jenkins, David Oyelowo",
  "Plot": "A homicide investigator digs deeper into a case involving a trained military sniper
    who shot five random victims.",
  "Language": "English", "Country": "USA",
  "Awards": "1 win & 5 nominations.",
  "Poster": "https://images-na.ssl-images-amazon.com/images/M/MV5BMTM1NjUxMDI3OV5BM15BanBnXkFtZTcwNjg1ODM3OA@@._V1_SX300.jpg",
  "Ratings": [
    {
      "Source": "Internet Movie Database",
      "Value": "7.0/10"
    }
  ]
}
```

```

],
"Metascore": "50", "imdbRating": "7.0",
"imdbVotes": "263,876", "imdbID": "tt0790724",
"Type": "movie", "DVD": "07 May 2013",
"BoxOffice": "\$80,033,643", "Production": "Paramount Pictures",
"Website": "http://www.jackreachermovie.com/",
"Response": "True"
}

```

3.2.3 Porovnání dat získaných ze serverů třetích stran

Při bližším pohledu na příklady odpovědí na získání detailu výše je patrné, že se v obou nacházejí podobné položky - například název pořadu v položce `title` nebo `Title`, položka pro shrnutí děje `overview` nebo `Plot`. Dále si lze povšimnout, že odpověď ze serveru OMDb navíc již obsahuje položky pro členy produkce a hlavní herce. Tyto položky jsou velmi důležité pro další použití získaných dat pro tuto práci, a tak se může zpočátku zdát, že odpověď ze serveru TMDb není příliš vhodná pro další užití.

Vše se mění při doplnění požadavku o parametr `append_to_response` nastavený na hodnotu `credits`. Díky tomuto parametru je do odpovědi přidána položka `credits`, která obsahuje podrobný seznam členů produkce, herců i filmových postav. Navíc server TMDb podporuje také vyhledávání názvu pořadů i v jiném než anglickém jazyce⁴. Odpověď je rovněž možné pomocí parametru `language` lokalizovat.

Vzhledem k tomuto je server TMDb vybrán jako hlavní server pro získávání dodatečných informací o médiu, a proto budou ukázány níže pouze příklady s tímto serverem. Server OMDb se použije pouze při neúspěšném získání informací ze serveru TMDb a slouží tedy jako záložní server.

⁴Například informace o českém filmu „Bába z ledu“ lze najít pomocí požadavku https://api.themoviedb.org/3/search/movie?api_key={api_key}&query=b%C3%A1ba%20z%20ledu.

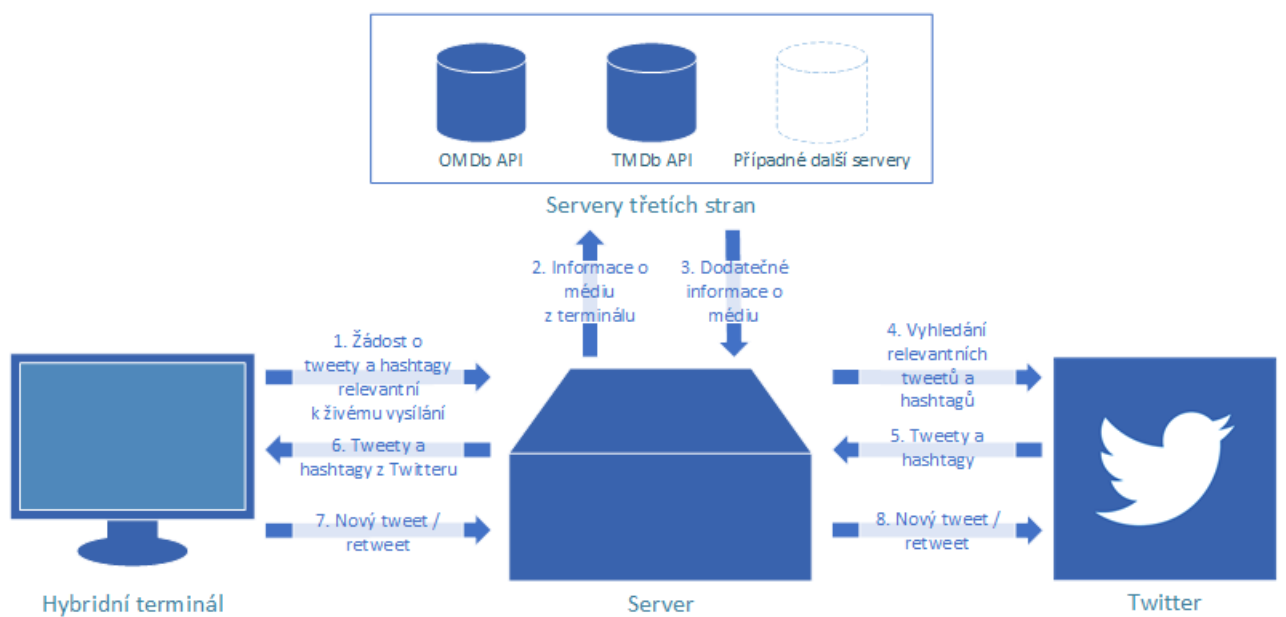
Kapitola 4

System pro získávání vysílání relevantního obsahu a hashtagů

Tato kapitola se zabývá návrhem systému sloužícího k vyhledávání příspěvků a hashtagů ze sociální sítě Twitter, které jsou relevantní k právě běžícímu vysílání. V podkapitole 4.1 je čtenáři přiblížen návrh systému. V následující sekci 4.2 se nachází popis navržené architektury.

4.1 Návrh systému

Předpokládané užití systému pro získávání vysílání relevantních tweetů a hashtagů ze sociální sítě Twitter a dále také k vkládání nových tweetů, případně retweet, je následující: Uživatel si spustí HbbTV aplikaci na svém televizním přijímači (terminálu). Aplikace zašle informace, kterými mohou být například název a popis pořadu či identifikátor kanálu na server. Server se z obdržených informací pokusí nalézt na serverech třetích stran dodatečné informace o médiu. Server ze všech získaných informací nalezne na sociální síti Twitter vhodné tweety a hashtagy a následně vše předá terminálu, který data zobrazí. Uživatel nyní může vidět na svém televizním přijímači aktuálně běžící pořad a rovněž tweety jejichž obsah se zabývá tímto pořadem. Pokud uživatel uzná za vhodné, je mu poskytnuta možnost retweetu zobrazeného příspěvku či přidání do oblíbených, a nebo může vytvořit zcela nový tweet. V případě, že se rozhodne vytvořit nový tweet, jsou mu automaticky nabídnuty získané hashtagy, které může jednoduše přidat do příspěvku. Nový obsah, retweet, nebo přidání do oblíbených je zasláno na server, který vše předá sociální síti Twitter. Výše popsané schéma je zobrazeno na obrázku 4.1.



Obrázek 4.1: Schéma vyhledávání existujících a vkládání nových relevantních příspěvků na sociální síti Twitter.

4.2 Architektura systému

Architektura systému propojujícího sociální síť Twitter a televizní vysílání je tvořena ze čtyř abstraktních prvků (viz obrázek 4.2).

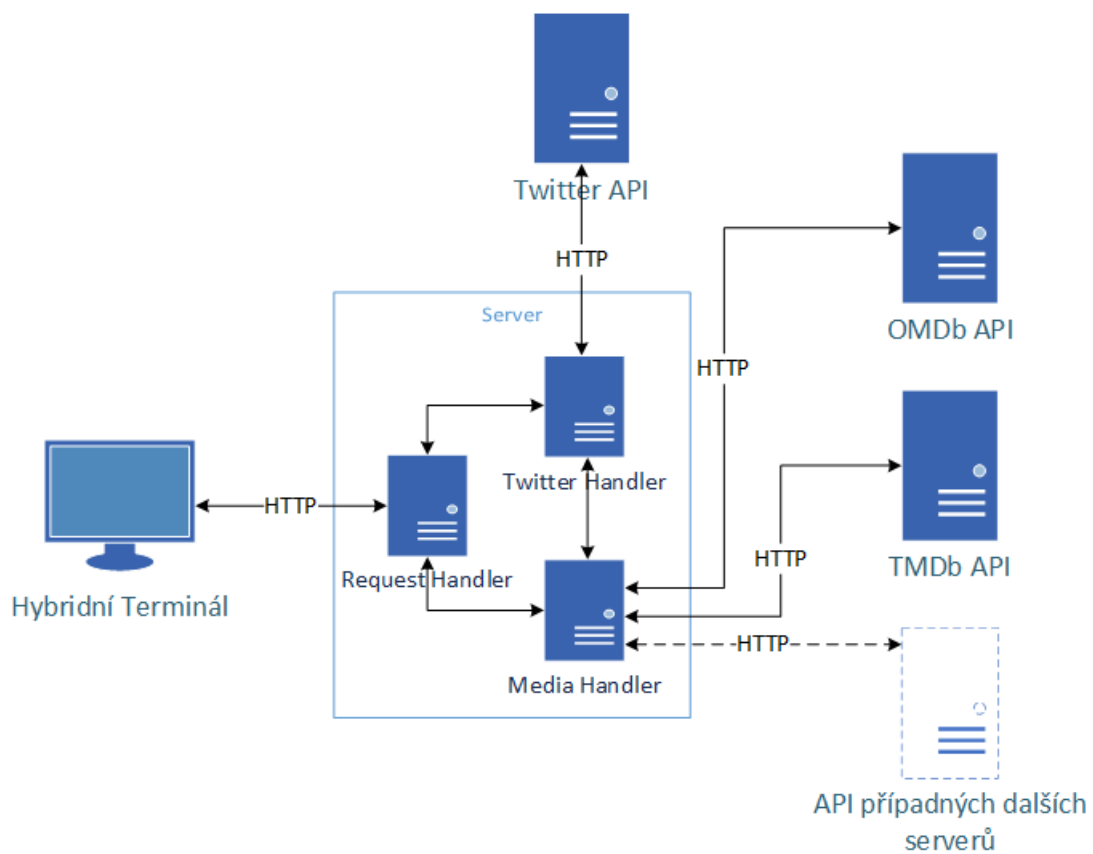
Prvním z nich je *Terminál* (přesněji HbbTV terminál), na kterém běží aplikace sloužící pro zobrazení vysílání relevantního obsahu z Twitteru a dále také pro vkládání nového obsahu na Twitter. Terminál je jediným prvkem celého systému, se kterým může uživatel interagovat.

Druhou částí systému je *Server*, který obsahuje tři abstraktní bloky, z nichž každý slouží pro komunikaci s jinou částí systému. Blok zpracovávající příkazy (*Request Handler*) zpracovává příkazy zaslané uživatelem skrz Terminál. Tyto příkazy jsou složeny z požadované operace a případně také z informací o aktuálně vysílaném pořadu (například název, identifikace kanálu, . . .). V případě, že se jedná o operaci, kterou lze přímo delegovat na Twitter API (vlození příspěvku, retweet, unretweet, přidání do oblíbených, odebrání z oblíbených), jsou informace z příkazu předány bloku pro práci se sítí Twitter (*Twitter Handler*), který sestaví žádost pro Twitter API a následně jí okamžitě odešle. Další možností je operace získání relevantních hashtagů a relevantního obsahu ze sítě Twitter (tweets). Pro tuto operaci je nejdříve potřeba získat dodatečné informace o aktuálně vysílaném pořadu ze serverů třetích stran (popsáno níže). Toto je realizováno v bloku pro média (*Media Handler*). Informace získané pomocí bloku pro média a rovněž také informace zaslané z Terminálu jsou následně předány bloku pro práci se sítí Twitter, který se pomocí Twitter API pokusí nalézt co nejrelevantnější výsledek a následně jej vrátí zpět do terminálu.

Třetí částí systému je samotná síť Twitter a její API.

Poslední čtvrtou částí jsou servery třetích stran s metadaty o multimediálním obsahu. Tyto servery, jak již bylo zmíněno výše, slouží pro získání dodatečných informací o aktuálním pořadu a v současnosti se jedná o servery OMDb API a TMDb API.

Komunikace mezi jednotlivými částmi je realizována pomocí protokolu HTTP. Primárně se jedná o metodu GET, která není použita pouze mezi serverem a Twitter API při vkládání nového obsahu, retweetu, unretweetu, přidání do oblíbených nebo odebrání z oblíbených. V tomto případě je použita metoda POST.



Obrázek 4.2: Navržená architektura řešení pro zobrazení vysílání relevantního obsahu na síti Twitter v rámci HbbTV vysílání a pro vkládání dalšího obsahu na Twitter.

Kapitola 5

Získávání relevantního obsahu a hashtagů

Proces získávání relevantního obsahu a hashtagů ze sociální sítě Twitter začíná obdržáním požadavku o vyhledání. Jakmile se toto stane, server vyhledá dodatečné informace ze serverů třetích stran, a z nich následně získá data vhodná pro vyhledávání a vyhodnocování relevantnosti příspěvků (konkrétně popsáno v sekci 5.1). Následně server prohledá sociální síť Twitter a nalezne odpovídající příspěvky (viz sekce 5.2). Posledním krokem (popsaným v sekci 5.3) je vyhodnocení získaných příspěvků a jejich seřazení podle relevantnosti. V tomto kroku jsou rovněž pro uživatele vybrány vhodné hashtagy (sekce 5.4).

V závěru této kapitoly se nachází sekce 5.5, která se zabývá porovnáním různých způsobů určení relevantnosti daného tweetu.

5.1 Získání dodatečných informací o pořadu

Server po přijetí požadavku na vyhledání relevantního obsahu na Twitteru nejprve z požadavku získá název vysílaného pořadu. Následně na serveru The Movie DB¹ zašle požadavek na vyhledání položky obsahující tento název. Server The Movie DB vrátí seznam nejvíce podobného obsahu. Z tohoto seznamu je vybrána položka, jejíž název přesně odpovídá názvu z požadavku na vyhledávání. V případě neúspěchu je do dalšího zpracování použita první položka (server vrací položky sestupně dle relevantnosti a tudíž by se mělo jednat o požadovaný pořad).

```
https://api.themoviedb.org/3/search/movie?query=Batman&api_key={api_key}
```

Ukázka 5.1: Požadavek na vyhledávání filmu „Batman“ na server The Movie DB².

¹Pro druhý server The Open Movie Database je postup téměř totožný a proto zde nebude uveden.

²{api_key} v požadavku je třeba nahradit vygenerovaným osobním API klíčem.

```

{
  "page": 1,
  "total_results": 107,
  "total_pages": 6,
  "results": [
    {
      "vote_count": 2673,
      "id": 268,
      "video": false,
      "vote_average": 7.1,
      "title": "Batman",
      "popularity": 29.53771,
      "poster_path": "\kbf3g9crrADGmc2AMAM1LBgSm2h.jpg",
      "original_language": "en",
      "original_title": "Batman",
      "genre_ids": [14, 28],
      "backdrop_path": "\2blmxp2pr4BhwQr74AdCfwgfM0b.jpg",
      "adult": false,
      "overview": "The Dark Knight of Gotham City begins his war on crime with his
first major enemy being the clownishly homicidal Joker, who has seized
control of Gotham's underworld.",
      "release_date": "1989-06-23"
    },
    {
      "vote_count": 9016,
      "id": 272,
      "video": false,
      "vote_average": 7.5,
      "title": "Batman Begins",
      "popularity": 41.509599,
      "poster_path": "\dr6x4GyyegBWtinPBzipY02J21V.jpg",
      "original_language": "en",
      "original_title": "Batman Begins",
      "genre_ids": [28, 80, 18],
      "backdrop_path": "\65JWXDCAfwHhJKnDwRnEgVB411X.jpg",
      "adult": false,
      "overview": "Driven by tragedy, billionaire Bruce Wayne dedicates his life to uncovering
and defeating the corruption that plagues his home, Gotham City. Unable to work within
the system, he instead creates a new identity, a symbol of fear for the criminal
underworld - The Batman.",
      "release_date": "2005-06-10"
    }
  ]
}

```

Ukázka 5.2: Odpověď na vyhledávání filmu „Batman“ ze serveru The Movie DB.

Z položky, získané v předchozí části, je vyjmut její identifikátor, který je použit pro zaslání dalšího požadavku na server The Movie DB. Odpověď na tento požadavek obsahuje rozsáhlý text ve formátu *JSON* s podrobnými informacemi o daném pořadu.

```
https://api.themoviedb.org/3/movie/268?api_key={api_key}
&append_to_response=credits
```

Ukázka 5.3: Požadavek na získání podrobností o filmu „Batman“ na serveru The Movie DB⁴.

V požadavku je kromě API klíče použit rovněž parametr `append_to_response` nastavený na hodnotu `credits`. Díky tomuto parametru lze v jednom požadavku spojit dvě operace - získání informací o filmu a získání herců a členů produkce (viz níže).

Pro účely dalšího zpracování jsou využity následující položky:

- **genres** - Seznam žánrů, do kterých je pořad zařazen. Položky tohoto seznamu jsou tvořeny z identifikátoru žánru `id` a jména žánru `name`.
- **original_title** - Původní název.
- **title** - Název.
- **credits** - Objekt obsahující dva seznamy:
 - **cast** - seznam herců, v němž každá položka obsahuje jméno herce `name` a jméno ztvárněné postavy `character`.
Z tohoto seznamu je použito prvních 7 herců.
 - **crew** - seznam členů produkce, kde každá položka obsahuje jméno člena `name` a pozici `job`.
Z tohoto seznamu jsou použity tyto položky⁵:
 - **Director** - režisér
 - **Novel** - autor knižní předlohy
 - **Screenplay** - scénárista
 - **Original Music Composer** - skladatel hudby
 - **Producent** - producent

⁴{api_key} v požadavku je třeba nahradit vygenerovaným osobním API klíčem.

⁵pokud se v seznamu nacházejí

```

{
  "adult": false, "backdrop_path": "\/2blmxp2pr4BhwQr74AdCfwgfM0b.jpg",
  "belongs_to_collection": {"id": 120794, "poster_path": "\/adQxUvQW8HXNsvMG5aj71pHh9T.jpg",
    "name": "Batman Collection", "backdrop_path": "\/9SpZvXLZDGGtFqVDBYgqSGI64Kw.jpg"},
  "budget": 35000000,
  "genres": [{"id": 14, "name": "Fantasy"}, {"id": 28, "name": "Action"}],
  "homepage": null,
  "id": 268, "imdb_id": "tt0096895", "original_language": "en",
  "original_title": "Batman",
  "overview": "The Dark Knight of Gotham City begins his war on crime with his first major enemy being the clownishly homicidal Joker, who has seized control of Gotham's underworld.",
  "popularity": 28.53771,
  "poster_path": "\/kBf3g9crrADGMc2AMAM1LBgSm2h.jpg",
  "production_companies": [{"id": 1382, "logo_path": null,
    "name": "PolyGram Filmed Entertainment", "origin_country": ""}],
  "production_countries": [{"iso_3166_1": "GB", "name": "United Kingdom"}],
  "release_date": "1989-06-23", "revenue": 411348924, "runtime": 126,
  "spoken_languages": [{"iso_639_1": "en", "name": "English"}], "status": "Released",
  "tagline": "Have you ever danced with the devil in the pale moonlight?",
  "title": "Batman",
  "video": false, "vote_average": 7.1, "vote_count": 2676,
  "credits": {
    "cast": [
      {
        "cast_id": 5, "order": 0,
        "character": "Jack NapierThe Joker",
        "credit_id": "52fe422fc3a36847f800aa05", "gender": 2, "id": 514,
        "name": "Jack Nicholson",
        "profile_path": "\/z5kVLyn3sxj0wNR1FgVgT6deeRO.jpg"},
      {
        "cast_id": 4, "order": 1,
        "character": "Bruce WayneBatman",
        "credit_id": "52fe422fc3a36847f800aa01", "gender": 2, "id": 2232,
        "name": "Michael Keaton",
        "profile_path": "\/myVdrYNGTgqunLfUSaM8DuVD7DL.jpg"}
    ],
    "crew": [
      {
        "credit_id": "52fe422fc3a36847f800a9f1",
        "department": "Directing",
        "gender": 2, "id": 510,
        "job": "Director",
        "name": "Tim Burton",
        "profile_path": "\/yHEHAHQpN9PfSEQx1UxZPczhcAi.jpg"}
    ]
  }
}

```

Ukázka 5.4: Odpověď požadavku na získání podrobností o filmu „Batman“ ze serveru The Movie DB.

5.2 Vyhledávání vhodných příspěvků na sociální síti Twitter

Z klíčových slov získaných v první fázi jsou vytvořeny následující tři pole:

- A **Pole termů** - obsahuje termíny pro další analýzu, které slouží pro určení relevantnosti daného tweetu
- B **Pole dotazů** - obsahuje slova, která budou použita pro vyhledávání na Twitteru
- C **Pole hashtagů** - obsahuje přímo vygenerované hashtagy, které jsou získány z informací o pořadu

A Pole termů

Toto pole obsahuje původní název, hashtag vytvořený z původního názvu. Pokud je název rozdílný od původního názvu pak se v poli nachází také název a hashtag získaný z názvu. Dále jména herců, jména jimi ztvárněných postav a rovněž hashtagy vytvořené z těchto jmen. Před samotným vložením do pole jsou jména předzpracována (viz níže). Poslední částí tohoto pole jsou jména důležitých členů produkce (viz předchozí sekce).

```
[  
  'Batman', '#Batman',  
  'Jack Napier', '#JackNapier', 'The Joker', '#TheJoker',  
  'Jack Nicholson', '#JackNicholson', 'Bruce Wayne', '#BruceWayne', 'Batman', '#Batman',  
  'Michael Keaton', '#MichaelKeaton',  
  'Vicki Vale', '#VickiVale',  
  'Kim Basinger', '#KimBasinger',  
  'Alfred', 'Michael Gough',  
  'Harvey Dent', 'Billy Dee Williams',  
  'Alexander Knox', 'Robert Wuhl',  
  'Commissioner James Gordon', 'Pat Hingle',  
  'Tim Burton', // "Director"  
  'Warren Skaaren', // "Screenplay"  
  'Danny Elfman', // "Original Music Composer"  
  'Sam Hamm', // "Screenplay"  
]
```

Ukázka 5.5: Pole termů pro film „Batman“

Zpracování jmen herců

Při zpracovávání jmen herců a filmových postav je třeba brát v potaz, že některá obsahují také přezdívku v uvozovkách (například Adrianna "Adrian" Pennino). Tuto přezdívku je třeba před vložením do tohoto pole vyjmout. U některých jmen se lze setkat se znakem lomítka /, které slouží pro specifikace dvou odlišných jmen ať už skutečné, nebo filmové postavy. Je třeba tyto jména rozdělit a zpracovávat je odděleně. Příkladem takového jména může být Jack Napier neboli Joker (v odpovědi serveru TMDb specifikováno jako Jack Napier / The Joker). Problémem u předzpracování jmen jsou rovněž dodatečná slova nesusouvisející nijak s danou postavou. V současnosti bylo nalezeno jediné takové slovo (*voice*), kterým je specifikováno, že daný herec pouze propůjčil svůj hlas dané filmové postavě. S tímto se lze setkat hlavně u animovaných filmů (například film „Zootropolis“).

B Pole dotazů

Pole dotazů je velmi podobné poli termů. Jediným rozdílem je, že některé položky jsou pro optimalizaci počtu volání spojeny mezi sebou pomocí operátoru OR (více o operátorech podporovaných pro vyhledávání v tabulce 3.1).

```
[  
  'Batman" OR #Batman',  
  '(Jack Napier) OR #JackNapier OR (The Joker) OR #TheJoker  
    OR (Jack Nicholson) OR #JackNicholson',  
  '(Bruce Wayne) OR #BruceWayne OR (Michael Keaton) OR #MichaelKeaton',  
  '(Vicki Vale) OR #VickiVale OR (Kim Basinger) OR #KimBasinger',  
  '(Michael Gough)',  
  '(Harvey Dent) OR (Billy Dee Williams)',  
  '(Alexander Knox) OR (Robert Wuhl)',  
  '(Commissioner James Gordon) OR (Pat Hingle)',  
  '(Tim Burton) OR (Warren Skaaren) OR (Danny Elfman)',  
  '(Sam Hamm)'  
]
```

Ukázka 5.6: Pole dotazů pro film „Batman“

C Pole hashtagů

Položky tohoto pole jsou hashtagy vytvořené z pole žánrů, z původního názvu a z názvu.

```
[ '#Fantasy', '#Action', '#Batman' ]
```

Ukázka 5.7: Pole dotazů pro film „Batman“

Jakmile jsou tato tři pole sestavena, dojde k vytvoření požadavků na vyhledávání⁶ z pole dotazů a jejich odeslání na Twitter.

```
https://api.twitter.com/1.1/search/tweets.json?lang=en&tweet_mode=extended&count=100  
&q="Batman" OR #Batman -filter:retweets  
https://api.twitter.com/1.1/search/tweets.json?lang=en&tweet_mode=extended&count=100  
&q=(Jack Napier) OR #JackNapier OR (The Joker) OR #TheJoker OR  
(Jack Nicholson) OR #JackNicholson -filter:retweets  
https://api.twitter.com/1.1/search/tweets.json?lang=en&tweet_mode=extended&count=100  
&q=(Bruce Wayne) OR #BruceWayne OR (Michael Keaton) OR #MichaelKeaton -filter:retweets
```

Ukázka 5.8: Vytvořené vyhledávací dotazy na film „Batman“⁷.

Odpovědi na každý z těchto požadavků je pole tweetů, které odpovídají danému dotazu. Všechny získané tweety jsou vloženy do jednoho pole a spolu s polem termů předány do další vyhodnocovací fáze 5.3. Pole hashtagů je použito až ve fázi 5.4.

⁶Popis vytváření požadavků na vyhledávání se nachází v sekci 3.1.2.

⁷Před použitím se tyto dotazy musí zakódovat pomocí URL kódování.

5.3 Vyhodnocení relevantnosti příspěvků

Prvním krokem při vyhodnocování relevantnosti příspěvků je jejich rozdělení na tokeny pomocí následujícího regulárního výrazu (získaného z článku [4]):

```
1 const STRING_REGEX = new RegExp(  
2   "(?:[;:]| [o0\\-]?[D\\] \\|\\(\\|\\/\\|o0pP])|" + // emotikony  
3   "(?:<[~>]+>)" + // HTML tagy  
4   "(?:@[\\w_]+)" + // @zminovani  
5   "(?:\\#+[\\w_]+[\\w\\' _\\-]*[\\w_]+)" + // hashtagy  
6   "(?:http[s]?://\\/(?:[a-z]| [0-9]| [\\$-_.&+]| [!*\\(\\),]|(?:%[0-9a-f] [0-9a-f]))+)"  
7   + // URL  
8   "(?:(?:\\d+,?)+(?:\\.?.\\d+)?)" + // cisla  
9   "(?:[a-z][a-z\\' _-]+[a-z])" + // Slova s~ a '  
10  "(?:[\\w_]+)" + // ostatni slova  
11  "(?:\\S)" // zbytek  
  , 'g'); // vyhledavat globalne
```

Následně dochází k odstranění tweetů, jež byly označeny jako nevhodné pro tuto práci. Konkrétně se jedná o tweety s pornografickou tematikou, kterých se na této sociální síti nachází nečekaně velké množství. Tweet je označen jako nevhodný v případě, že se v jeho obsahu nachází aspoň jedno ze zakázaných slov. Seznam zakázaných slov lze nalézt ve zdrojových kódech výstupní aplikace.

Po převedení všech obsahů tweetů na tokeny je rovněž pole termů, získané v předchozí části, převedeno na tokeny. Následně dochází pomocí hodnotící metody (viz sekce 5.5) k určení relevantnosti daného tweetu k poli termu. Tímto je získáno bodové ohodnocení (skóre) pro každý tweet.

Následně jsou tweety seřazeny sestupně podle dosaženého bodového ohodnocení a tweety s bodovým ohodnocením 0 (žádná shoda) jsou zahozeny.

Výsledné pole je zasláno společně s hashtagy z fáze 5.4 zpět uživateli.

5.4 Vyhodnocení vhodných hashtagů

Obdobně jako u vyhodnocování relevantnosti příspěvků je také při vyhodnocení hashtagů nejdříve použit regulární výraz pro získání jednotlivých hashtagů. Použitý regulární výraz má následující podobu:

```
1 const HASHTAG_REGEX = new RegExp("(\\#+[\\w_]+[\\w\\' _\\-]*[\\w_]+)", 'g');
```

Poté jsou sečteny výskyty hashtagů ve všech dokumentech a 10 nejčastějších je spojeno s polem hashtagů z sekce 5.2 a následně vráceno uživateli.

Pro výpočet vhodných hashtagů bylo rovněž vyzkoušeno vzít v potaz výsledné ohodnocení dokumentu obsahujícího daný hashtag z hodnotící metody, avšak během testování se ukázalo, že tento způsob nepřináší žádné výhody a výsledek je prakticky totožný jako při pouhém součtu výskytů.

5.5 Metody pro hodnocení relevantnosti tweetů

Pro určení relevantnosti tweetů byla zvolena metoda *Term frequency-inverse document frequency* (TF-IDF). TF-IDF je metoda, která slouží pro získání skóre podobnosti dokumentu s kolekcí dokumentů. Její obrovskou výhodou je, že nastavuje nižší váhu obecně častým slovům jako jsou spojky, předložky apod. Naopak zvyšuje váhu slovům, která se v daném dokumentu objevují často, ale zároveň se nevyskytují často v ostatních dokumentech. Díky tomuto není potřeba před samotným výpočtem odstranit tzv. *stopwords*. Pod tímto termínem se rozumí slova, která jsou v dokumentech častá, ale většinou nenesou žádnou informační hodnotu (například v anglickém jazyce členy „the“ a „a“). [14]

Term frequency (TF) vyjadřuje, jak často se daný term vyskytuje v daném dokumentu. Aby se předešlo zvýhodnění dlouhých dokumentů s více vyhledávanými výrazy, je výsledek normalizován délkou dokumentu.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (5.1)$$

kde $n_{i,j}$ je počet výskytů slova t_i v dokumentu d_j a jmenovatel je počet výskytů všech slov v dokumentu (délka dokumentu).

Inverse Document Frequency (IDF) je definováno jako logaritmus podílu počtu všech dokumentů ku počtu dokumentů, které obsahují daný term.

$$idf_i = \log_{10} \frac{|D|}{|\{d : d \ni t_i\}|} \quad (5.2)$$

kde $\{d : d \ni t_i\}$ je počet dokumentů obsahující term i a $|D|$ celkový počet dokumentů.

Matematicky je TF-IDF definováno jako součin *Term frequency* a *Inverse document frequency*.

$$tfidf = tf \cdot idf \quad (5.3)$$

Tento přístup se ukázal jako funkční, avšak docházelo k nežádoucímu zvýhodnění kratších tweetů, jejichž obsah byl složený například pouze z jména jednoho herce. Hodnotící metoda dle implementace správně přiřadila danému příspěvku velmi vysoké skóre, avšak pro potenciálního uživatele tento tweet nepřidával žádnou přidanou informaci.

Vzhledem k tomuto byly v rámci této práce experimentálně vyzkoušeny i různé varianty metody TF nalezených v prezentaci [11].

5.5.1 Porovnání různých variant metody TF

Varianty metody TF vyzkoušené v této práci jsou konkrétně logaritmická (logarithm), rozšířená (augmented) a nenormalizovaná. Vzorce každé z metod jsou následující⁸:

- Term frequency bez normalizace:

$$tf_{i,j}^{nn} = n_{i,j} \quad (5.4)$$

- Logarithm Term frequency:

$$tf_{i,j}^{log} = 1 + \log(tf_{i,j}^{nn}) \quad (5.5)$$

- Augmented Term frequency:

$$tf_{i,j}^{aug} = a + (1 - a) \frac{tf_{i,j}}{\max_t(tf_j)} \quad (5.6)$$

kde a je vyhlazovací konstanta v intervalu od 0 do 1 a nejčastěji je podle [5] nastavena na hodnotu 0.4 (v minulosti se používala hodnota 0.5) a $\max_t(tf_j)$ je maximální tf daného dokumentu j .

Všechny tyto metody byly otestovány na stejné sadě tweetů a to rovněž s odstraněním stopwords. Seznam stopwords byl pro český a anglický jazyk získán ze stránky [13].

Testování variant probíhalo na pěti kolekcích tweetů získaných z Twitteru pomocí vyhledávání pro následujících pět filmů:

- Deník Bridget Jones (Bridget Jones's Diary)
- Rocky
- Forrest Gump
- Titanic
- Hobit

Pro každou z hodnotících variant byly všechny tweety z dané kolekce ohodnoceny a následně seřazeny sestupně od nejvyššího skóre po nejnižší. Poté došlo k vybrání deseti tweetů s nejvyšším skóre, které byly následně ohodnoceny body dle tabulky 5.1.

Bodů	Text tweetu obsahuje...
5	název pořadu, citaci z pořadu, nebo je daný tweet objektivně relevantní k pořadu.
4	jméno více než jedné z hlavních postav případně i jména herců.
3	jméno jedné z hlavních postav nebo obsahuje výčet více názvů filmů (například seřazení do žebříčku oblíbenosti).
2	jméno herce nebo významného člena produkce.
1	nesouvisející informace k danému pořadu.

Tabulka 5.1: Bodové ohodnocení tweetů v závislosti na obsahu.

Na závěr došlo ke zprůměrování všech výsledných hodnot pro danou variantu metody. Výsledky lze nalézt v tabulce 5.2.

⁸U každé z výše zmíněných variant se liší pouze část TF, část IDF zůstává totožná.

Varianta	Deník Bridget Jones	Forrest Gump	Hobit	Rocky	Titanic	Průměrně bodů
tf	4.8	5.0	3.1	3.0	2.0	3.6
tf^{nn}	4.4	3.6	3.0	4.5	2.7	3.6
tf^{log}	4.5	4.6	2.4	5.0	3.6	4.0
tf^{aug}	4.4	4.8	3.3	4.4	3.3	4.0
tf^*	4.7	5.0	2.4	2.5	1.9	3.3
tf^{nn*}	4.3	5.0	3.4	5.0	2.7	4.1
tf^{log*}	4.4	4.7	3.0	5.0	3.8	4.2
tf^{aug*}	4.1	4.6	2.7	4.7	4.4	4.1

Tabulka 5.2: Průměrné bodové ohodnocení 10 nejlepších tweetů všechny zkoumané metody a pěti vybraných filmů.

* značí odstranění stopwords.

Z výsledků je patrné, že nevhodnějších výsledků dosahuje **logaritmická varianta s odstraněním stopwords**. Použití této varianty je rovněž doporučeno v prezentaci [11]. Přesto je nutno poznamenat, že výše popsané testování se zakládá na osobním hodnocení od dvou rozdílných lidí, tudíž by bylo třeba pro zvýšení věrohodnosti provést další výzkum na větším vzorku subjektů.

Kapitola 6

Implementace

Jak již bylo zmíněno výše, celá aplikace je navržena a implementována v architektuře klient-server. Architektura tohoto typu byla zvolena proto, aby se docílilo nezávislosti na televizním kanálu a také aby se snížila výpočetní náročnost aplikace na hybridním terminálu.

Pro implementaci je použit programovací jazyk JavaScript s využitím konstrukcí a funkcí z jazyka ECMAScript verze 6 (ES6). Důvodem k použití ES6 je větší přehlednost kódu díky použití modulů. Aby se docílilo zpětné kompatibility je kód přetransformován do ECMAScriptu verze 5 (ES5) pomocí nástroje *Babel*¹.

6.1 Server

Serverová část aplikace je určena pro ekosystém *Node.js*². Pro větší přehlednost asynchronního běhu je použit modul *Promise*. Dalšími moduly, které jsou použity, jsou *QRCode*, modul pro generování QR kódu z vloženého textu, a modul *OAuth*, který zajišťuje autentizovanou komunikaci s Twitterem. Celý server pak běží s využitím modulu *http*, který je přímo součástí platformy *Node.js*.

Samotný server lze rozdělit do tří abstraktních bloků (jak již bylo popsáno v sekci 4.2). Těmito bloky jsou *blok pro zpracování požadavků od uživatele* (popsán v sekci 6.1.1), *blok pro získávání dodatečných informací o médiu ze serverů třetích stran* (viz sekce 6.1.2) a *blok pro interakci se sociální sítí Twitter* (popis v sekci 6.1.3).

6.1.1 Blok pro zpracování požadavků

Hlavní třídou tohoto bloku je třída `RequestHandler`, jejíž metoda `onRequest` je nastavena jako vykonavatel příchozích požadavků při vytváření serveru pomocí metody `createServer` modulu `http`.

Metoda `onRequest` je navržena tak, aby bylo možno přidat do serveru nové moduly bez nutnosti dalších uprav. Vzhledem k tomu, že server slouží v současnosti pouze k interakci se sociální sítí Twitter, obsahuje tato metoda pouze ověření, zda URL požadavku obsahuje podřetězec `twitter` či nikoliv. Jestliže ano, pak je zavolána metoda `handleTwitterRequest`. V opačném případě je okamžitě zaslána odpověď se stavovým kódem 404, což značí nesprávný požadavek.

Metodou `handleTwitterRequest` je určeno komunikační rozhraní s Twitterem, které se v současnosti skládá z položek níže.

¹Oficiální stránka nástroje Babel je <https://babeljs.io/>.

²Oficiální stránka systému Node.js je <https://nodejs.org/>.

Komunikační rozhraní poskytnuté serverem má tyto položky, které jsou pro přehlednost rozděleny do dvou skupin:

■ Pro přihlášení do sociální sítě Twitter:

- getRequestToken** - získání OAuth Request tokenu a OAuth Request token secret, URL a QR kód pro přihlášení. **Bez parametrů.**
- getAccessToken** - získání OAuth Access tokenu a OAuth Access token secret.

parametry:

- `pin` - PIN
- `token` - OAuth Request tokenu
- `tokenSecret` - OAuth token secret

■ Pro další interakci se sociální sítí Twitter:

- getTimeline** - získání aktuálních příspěvků z účtu uživatele³.
- search** - získání relevantních příspěvků k pořadu s názvem specifikovaným v parametru `query`.

parametry:

- `query` - název pořadu pro vyhledávání
- `lang` - jazyk vyhledávaných příspěvků dle kódu ISO 639-1. Nepovinný parametr s výchozí hodnotou `en` pro anglický jazyk.

- postTweet** - publikace nového příspěvku na sociální síť Twitter.

parametry:

- `status` - obsah pro nově vytvořený příspěvek

- retweet** - retweetnutí příspěvku specifikovaného parametrem `id`.

parametry:

- `id` - id příspěvku, pro který se má provést retweet

- unretweet** - odstranění retweetnutí příspěvku specifikovaného parametrem `id`.

parametry:

- `id` - id příspěvku, pro který se má provést unretweet

- favoritesCreate** - přidání příspěvku specifikovaného parametrem `id` do oblíbených.

parametry:

- `id` - id příspěvku, který se má přidat do oblíbených

- favoritesDestroy** - odebrání příspěvku specifikovaného parametrem `id` z oblíbených.

parametry:

- `id` - id příspěvku, který se má odebrat z oblíbených

³„Timeline“ je v terminologii Twitteru označen seznam aktuálních příspěvků od účtů, které daný uživatel sleduje.

Každá výše popsaná položka pro interakci se sociální sítí Twitter musí kromě výše popsaných parametrů obsahovat dále také tyto parametry pro autentizaci daného uživatele:

parametry:

- `accessToken` - OAuth Access token
- `accessTokenSecret` - OAuth Access token secret

6.1.2 Blok pro získávání dodatečných informací o médiu

Pro tento blok je hlavní třídou třída `MediaHandler`, která obsahuje pouze konstruktor, v němž jsou vytvořeny instance tříd pro komunikaci se servery třetích stran, a metodu `searchQuery`, která pro název specifikovaný ve vstupním parametru `query` hledá odpovídající položku na těchto serverech.

V současnosti jsou pro vyhledávání použity servery TMDb (pro více informací viz 3.2.1) a OMDb (viz sekce 3.2.2). Ke každému z těchto serverů je přiřazena třída, která zajišťuje komunikaci a také vyextrahování informací. Pro server TMDb se jedná o třídu `TMDBHandler` a pro OMDb je to třída `OMDBHandler`. Aby se docílilo co možná největší přehlednosti, a také aby se zvýšila jednoduchost přidání potenciálních dalších serverů, je pojmenování, účel i pořadí vykonávání metod v obou modulech totožný. Liší se pouze konkrétní implementace, například získávání informací z odpovědi, vytváření požadavků a podobně.

Vstupní metodou pro vyhledávání je metoda `processQuery`, která nejdříve převede vyhledávaný řetězec na malá písmena a následně předá řízení metodě `searchForTitle`. V této metodě je zaslán požadavek o vyhledávání řetězce na daný server s metadaty o multimediálním obsahu (URL požadavku je získáno z metody `getQueryURL`). Po přijetí odpovědi se v metodě `parseItemList` nalezne položka s úplnou shodou názvu s vyhledávaným řetězcem. V případě neúspěšného nalezení úplné shody je jako nejvíce relevantní vrácena první položka. Následně je z této položky získán identifikátor, který je zaslán do metody `getItemInfo`. Dalším krokem je vytvoření nového požadavku o získání podrobných informací o dané položce. Toto je implementováno v metodě `getItemInfo` a URL požadavku je získáno z metody `getDetailUrl`. Poslední fází získávání dodatečných informací o médiu je zpracování odpovědi z předchozího požadavku, které je řešeno v metodě `findUsefullInformations`. Podrobně je zpracování popsáno v sekci 5.1 a také v části sekce 5.2.

6.1.3 Blok pro interakci se sociální sítí Twitter

Do tohoto bloku jsou zařazeny dvě třídy - třída `TwitterHandler` pro interakci se sociální sítí Twitter a třída `TweetProcessor`, která zpracovává získané příspěvky a vyhodnocuje jejich relevantnost.

TwitterHandler

Třída `TwitterHandler` vykonává příkazy ze sekce 6.1.1. Při samotné inicializaci třídy se provede vytvoření nové instance `oauth.OAuth` z externího modulu `OAuth`, přes kterou je vedena veškerá komunikace se sociální sítí Twitter.


```

1  this.oauth = new oauth.OAuth(
2      TWITTER_CONFIG.requestTokenUrl,
3      TWITTER_CONFIG.accessTokenUrl,
4      TWITTER_CONFIG.consumerKey,
5      TWITTER_CONFIG.consumerSecret,
6      '1.0',
7      'oob',
8      'HMAC-SHA1'
9  );

```

Ukázka 6.1: Vytváření nové instance `oauth.OAuth`⁴.

Všechny příkazy ze sekce 6.1.1 lze jednoduše převést a zaslat přímo na API Twitteru. Jedinou výjimkou je příkaz `search`, který je nejrozsáhlejší operací serveru. Prvním krokem je metoda `getSearch`. Ta nejdříve získá dodatečné informace o daném pořadu (viz předchozí sekce) a následně se získanými daty zavolá metodu `sendMultipleTermQueries`. V této metodě se vytvoří a odešlou požadavky o vyhledání na API Twitteru z termů získaným v předchozím kroku. Ze všech odpovědí je následně vytvořeno jedno pole příspěvku, které je předáno do zpracování.

TweetProcessor

Ve třídě `TweetProcessor` je vyhodnocena relevantnost všech získaných příspěvků vzhledem k poli vyhledávaných položek. Vyhodnocování je zahájeno v metodě `findMostAccuredTweetsTo`. Zde se nejprve odstraní nevhodné tweety (dle seznamu v souboru `forbiddenWords.js`) a poté se pro zbývající tweety (v metodě `createDocumentFromTweet`) vytvoří odpovídající dokument, složený ze seznamu tokenů, získaného aplikací regulárního výrazu 5.3 na text (konkrétně položka `full_text`) tweetu, a mapy jejich výskytů. Při samotné tokenizaci jsou z dokumentu vyjmuta tzv. stopwords pro daný jazyk (dle seznamu v souboru `stopwords.js`). Mapa výskytů určuje kolikrát se daný token v daném tweetu objevil a je použita pro rychlé získání počtu výskytů při výpočtu hodnotící metody.

Následně je pro každý dokument pomocí hodnotící metody vypočteno skóre, které určuje jak moc tento dokument odpovídá vyhledávaným výrazům. Dokumenty jsou dále seřazeny od nejvyššího po nejnižší skóre a ty s ohodnocením 0 jsou zahozeny. Pro snížení datové náročnosti jsou ze zbývajících tweetů vytvořeny jejich zmenšené kopie, které obsahují pouze položky užitečné pro HbbTV aplikaci. Mezi položky zmenšené kopie tweetu patří čas vytvoření, úplný text, informace o autorovi tweetu, identifikátor, entity tweetu, počet retweetů, počet vložení do oblíbených a zda byl daný tweet retweetnut či zda byl vložen do oblíbených.

Posledním krokem je výpočet výskytů všech hashtagů (v metodě `countHashtags`) v kolekci dokumentů, jejich seřazení sestupně a následné zkombinování s ostatními hashtagy získanými z informací o daném médiu.

Na závěr je pole hashtagů a pole zmenšených tweetů předáno zpět do třídy `RequestHandler`, která provede zaslání odpovědi obsahující tyto informace.

⁴`TWITTER_CONFIG` je konfigurační objekt, `oob` je zkratkou pro *out-of-band* a značí autentizaci skrz jiný než hlavní komunikační kanál - jedná se o Twitterem požadovaný parametr pro autentizační metodu založenou na PINu (více informací lze nalézt v sekci 3.1.1) a `HMAC-SHA1` je typ autentizačního kódu.

6.2 HbbTV aplikace

Celá HbbTV aplikace je tvořena ze tří hlavních scén, jedné pomocné scény pro nastavení a dialogu. První scénou, se kterou se uživatel setká, je scéna pro přihlášení. Tato scéna je popsána v sekci 6.2.1. Po úspěšném přihlášení je uživatel přesměrován na scénu se zobrazenými příspěvky (viz sekce 6.2.2). Poslední hlavní scénou je scéna pro vytváření a publikaci nového obsahu na sociální síť Twitter (popsána v sekci 6.2.3). Tuto podkapitolu uzavírá sekce 6.2.4, která obsahuje popis scény nastavení a dialogů.

Aplikaci je třeba spustit pomocí stisku červeného barevného tlačítka. O možnosti spustit tuto aplikaci uživatele informuje dialog, který se zobrazí na obrazovce po dobu 5 sekund. Uzavření aplikace je v celé aplikaci možné rovněž pomocí červeného tlačítka. Funkčnost ostatních barevných tlačítek, pokud existuje, lze nalézt v odpovídající sekci s popisem scény.

6.2.1 Scéna pro přihlášení

Scéna pro přihlášení je zobrazena v případě, že se žádný uživatel na tomto zařízení doposud nepřihlásil, nebo pokud se předchozí uživatel odhlásil. To, zda je nějaký uživatel přihlášen, je indikováno tím, zda se v lokálním úložišti (více informací v níže v sekci 6.2.5) nachází neprázdné položky `accessToken` a `accessTokenSecret`.

Nejvýraznějším prvkem této scény je QR kód, který společně se zobrazenou URL adresou, slouží pro přihlášení uživatele na jeho účet na síti Twitter (podrobnosti o přihlášení do sítě Twitter viz 3.1.1). Pokud se chce uživatel přihlásit, je potřeba, aby si načel URL z QR kódu, případně přepsal přímo vygenerované URL, a přešel na tuto adresu ve svém internetovém prohlížeči. QR kód i URL pro přihlášení jsou získány z odpovědi na požadavek `getRequestToken`. Součástí odpovědi je rovněž Request token a Request token secret, které jsou uloženy do `oauthToken` resp. `oauthTokenSecret` pro další fázi autentizace. Po zadání URL do internetového prohlížeče se uživateli zobrazí na stránce Twitteru autentizační formulář. Poté co uživatel potvrdí formulář se zobrazí stránka s PINem, který je třeba zadat do aplikace. Viz obrázky v příloze C.

Pro tento účel je na přihlašovací scéně aplikace vstupní pole, do kterého lze PIN zadat, buď pomocí číselných kláves na ovladači, nebo pomocí numerické virtuální klávesnice na obrazovce. Nutno dodat, že virtuální klávesnice je rovněž součástí aplikace, jelikož většina terminálů neposkytuje žádnou výchozí klávesnici. V případě úplné absence klávesnice by pro některé zredukované typy dálkových ovladačů bylo nemožné PIN zadat. Samotné vstupní pole je rovněž vytvořeno bez použití standardní značky `<input>`, jelikož je o této značce známo, že na některých zařízeních nefunguje korektně.

Posledním funkčním prvkem na přihlašovací scéně je tlačítko pro odeslání PINu. Po jeho stisknutí se obsah vstupního pole pro PIN společně s Request token a Request token secret odešle v požadavku `getAccessToken` na server. Server následně přepošle požadavek na Twitter API, kde dojde k vyhodnocení, zda je PIN správný či nikoliv. V případě korektního PINu je do uživatelské aplikace vrácen Access token a Access token secret. Tyto dvě položky jsou uloženy do lokálního úložiště a slouží pro autentizaci všech následujících požadavků na server. Následně je scéna pro přihlášení skryta a uživatel je přesměrován na scénu s příspěvky (viz 6.2.2). Pokud zasláný PIN není správný, je uživateli zobrazen dialog s odpovídajícím textem, obsah vstupního pole pro PIN je vymazán a dojde k vygenerování nového přihlašovacího URL opět pomocí požadavku `getRequestToken`. Uživatel poté musí opakovat přihlašovací cyklus.



Obrázek 6.1: Vzhled scény pro přihlášení zobrazené v pluginu FireHbbTV.

6.2.2 Scéna s příspěvkky

Scéna s příspěvkky je hlavní scénou celé aplikace, která je aktivní buď ihned po zobrazení, anebo po úspěšném přihlášení uživatele. Hlavním prvkem této scény je „timeline“, což je vertikálně posuvná oblast, která umožňuje zobrazovat tweety získané z vyhledávání viz níže. Dalším prvkem je tlačítko pro vytvoření nového tweetu, po jehož stisknutí přechází aplikace na scénu umožňující publikaci nového příspěvku (scéna popsána v sekci 6.2.3). Další funkčnost je uživateli umožněna skrz stisk standardních barevných tlačítek:

Červené tlačítko - ukončení aplikace.

Zelené tlačítko - přidání aktivního tweetu do oblíbených. Pomocí API metody `favoritesCreate`, kde je jako parametr `id` použit identifikátor tohoto tweetu. V případě, že aktivní tweet se již v oblíbených nachází, je na místo `favoritesCreate` použita metoda `favoritesDestroy` pro jeho odebrání. Parametr `id` zůstává u obou metod totožný.

Žluté tlačítko - přechod aplikace do scény nastavení (viz 6.2.4).

Modré tlačítko - retweet aktivního tweetu pomocí API metody `retweet` či `unretweet` pomocí API metody `unretweet` v případě, že aktivní tweet byl již retweetnut. U obou API metod je jako parametr `id` použit identifikátor tohoto tweetu.

Nyní k samotnému vyhledávání příspěvků. Při zobrazení této scény dojde k odeslání události `getContent`, na její odeslání naslouchá posluchač uvnitř hlavního modulu `main.js`. Tento posluchač po obdržení události vyhledá v objektu `video/broadcast`⁵ název aktuálního televizního kanálu a rovněž název běžícího pořadu.

⁵Objekt `video/broadcast` je součástí HbbTV API a přistupovat k němu lze pomocí elementu `<object id="video" type="video/broadcast"></object>` v HTML stránce

Následně dochází k odeslání události `newContent`, jež obsahuje obě výše zmíněné položky. Na výskyt události `newContent` naslouchá posluchač ve scéně s příspěvkem. Po jejím obdržení dochází k odeslání názvu pořadu v požadavku `search` na server, název kanálu je převeden na hashtag a uložen pro další fázi.

Jakmile dojde k přijetí odpovědi na požadavek `search` ze serveru, jsou v ní obsažené tweety zobrazeny uživateli a dojde ke sloučení pole hashtagů s hashtagem názvu kanálu. Nově vzniklé pole hashtagů je uloženo do paměti a v případě, že uživatel přejde na scénu pro vytvoření nového příspěvku (viz 6.2.3), jsou mu zde nabídnuty pro potenciální využití.



Obrázek 6.2: Vzhled scény s příspěvkem zobrazené v pluginu FireHbbTV.

6.2.3 Scéna pro vytvoření nového příspěvku

Scéna pro vytvoření nového příspěvku neboli tweetu je zobrazena v případě, že uživatel v sekci 6.2.2 stiskne klávesu pro vytvoření nového příspěvku. Hlavním prvkem scény je pole pro vytvoření obsahu nového příspěvku.

Pro toto pole, podobně jako u vstupu pro PIN ve scéně pro přihlášení, nelze vzhledem k nekompatibilitě některých zařízení použít značku `<input>`. Jako alternativní přístup pro vytvoření vstupního pole byla zvolena dvojice elementů `` mezi nimiž je jeden element `<div>`, jež je pomocí stylování upraven na řádkovou svislou úsečku, která slouží jako kurzor. Při pohybu tímto polem dochází k přesunu z jednoho elementu `` do druhého `` v závislosti na směru pohybu. Při tomto pohybu je třeba dávat pozor na HTML entity, nyní použity entity `<` pro znak `<` či `>` pro znak `>`, a na HTML značky, v současnosti pouze značka pro nový řádek `
`.

Po stisku na hlavní pole se uživateli zobrazí virtuální klávesnice, která obdobně jako virtuální numerická klávesnice použitá ve scéně pro přihlášení, musí být součástí aplikace. Použití systémové klávesnice není u většiny zařízení standardizováno a může docházet k problémům.

Dalším důležitým prvkem nacházejícím se na této scéně je horizontálně posuvná lišta, na které jsou uživatelům zobrazovány nabídnuté předgenerované hashtagy. Tyto hashtagy jsou společně s řízením předány ze scény s příspěvkem 6.2.2. Lišta je vizuálně spojena s klávesnicí.

Posledním prvkem scény pro vytvoření nového příspěvku je tlačítko na vytvoření nového tweetu, po jehož stisknutí dojde ke spojení obsahu elementů `` ve vstupním poli, dekódování případných HTML značek `
` na znaky odpovídající novému řádku `\r\n` a dekódování HTML entit. Následně je celý obsah nastaven parametru `status` a zaslán pomocí API metody `postTweet` na server. V případě, že se podařilo nový tweet úspěšně publikovat, dochází k přesměrování aplikace zpět na scénu s příspěvkem 6.2.2. Pokud vytvoření nového tweetu nebylo úspěšné, je uživatel informován odpovídající chybovou hláškou.

Pomocí žlutého tlačítka může uživatel uzavřít otevřenou klávesnici, nebo v případě, že klávesnice je již skryta, přejít zpět na scénu s příspěvkem 6.2.2. Podobně je na scénu s příspěvkem přesměrován kdykoliv po stisknutí tlačítka zpět. Pokud je klávesnice zobrazena, může uživatel využít modré tlačítko pro smazání posledního znaku a zelené tlačítko pro rychlou změnu mezi písmeny a symboly.



Obrázek 6.3: Vzhled scény pro nový tweet v pluginu FireHbbTV.

6.2.4 Scéna nastavení a dialogy

Scéna nastavení společně s dialogy plní pouze pomocnou funkci, jež uživateli nepřidává žádnou dodatečnou funkci.

Scéna nastavení v současnosti obsahuje pouze dvě položky. První z nich je možnost zvolení jazyka prostředí, na výběr je český a anglický jazyk (výchozí nastavení). Nastavení jazyka se nevztahuje na hlášení získané ze serveru, která jsou vždy v anglickém jazyce. Druhou možností je odhlášení uživatele, které je realizováno tím, že dojde k vymazání položek `accessToken`, `accessTokenSecret`, `oAuthToken` a `oAuthTokenSecret` z lokálního úložiště. Uživatel je následně přesměrován na scénu pro přihlášení 6.2.1. Ze scény lze pomocí stisku žlutého tlačítka nebo tlačítka zpět přejít zpět na scénu s příspěvkem 6.2.2.



Obrázek 6.4: Vzhled scény nastavení v pluginu FireHbbTV.

Dialog je jednoduchý element obsahující místo pro vložení libovolného textu a tlačítko „zavřít“, po jehož stisknutí dojde k uzavření dialogu. Ke zobrazení dialogu může dojít například při zadání špatného PINu nebo při neúspěšném požadavku na server.



Obrázek 6.5: Vzhled dialogu v pluginu FireHbbTV.

Do dialogů lze rovněž zahrnout informační prvek, který uživateli nabízí spuštění aplikace pomocí stisku červeného tlačítka. Tento prvek je zobrazen po dobu 5 sekund po inicializaci aplikace.



Obrázek 6.6: Vzhled informačního dialogu v pluginu FireHbbTV.

6.2.5 Implementační detaily HbbTV aplikace

V této sekci jsou popsány některé důležité implementační detaily, které nelze přímo zařadit k jedné scéně. Konkrétně se jedná o popis navigační logiky, směřování a vizuální prvky, použití lokálního uložště a použité ikony.

Aktivní prvek

HbbTV standard podporuje výchozí způsob navigace pomocí funkce `focus` nebo pomocí `css` navigace. Oba tyto způsoby ovšem bohužel nejsou podporovány na všech televizních zařízeních a je tedy vhodnější vytvořit systém pro navigaci jako součást aplikace.

V celé aplikaci je aktuálně aktivní prvek uchovávan pomocí proměnné `focused` v modulu `focused` (součást aplikace), který rovněž obsahuje metody pro získání nebo změnu aktivního prvku.

Konkrétně lze jako aktivní prvek přiřadit HTML element nebo tzv. *Snippet*, což je JavaScriptový objekt, který zaobaluje uvnitř jednoho prvku více elementů stejného typu. Příkladem snippetu může být například klávesnice, která je-li aktivní, umožňuje navigaci po jednotlivých klávesách a zároveň zajišťuje správný přechod mezi klávesnicí a ostatními prvky.

Vizuální prvky a směřování

Celá aplikace je napsána jako takzvaná Single-page application (SPA), což znamená, že celá aplikace se skládá z jedné HTML stránky. Veškerý zobrazovaný obsah je tedy součástí jedné HTML stránky konkrétně souboru `index.html`. Aby se docílilo větší přehlednosti HTML souboru, nachází se v něm pouze obalující elementy pro každou scénou, snippet či dialog. Během inicializace je následně do těchto obalujících elementů pomocí metody `insertAdjacentHTML` vložen řetězec obsahující HTML kód dané scénou, nebo dialogu. Snippets (například klávesnice nebo lišta s hashtagy) jsou vytvářeny dynamicky vkládáním

jednotlivých potomků (například jednotlivých kláves) do obalovacího elementu pomocí metody `appendChild`.

```
1 <div id="app">
2   <div id="login_wrapper" class="content"></div>
3   <div id="timeline_wrapper" class="content"></div>
4   <div id="new_tweet_input_wrapper" class="content"></div>
5   <div id="settings_wrapper" class="content"></div>
6
7   <div id="keypad"></div>
8   <div id="keyboard"></div>
9   <div id="hashtagline"></div>
10  <div id="dialog"></div>
11  <div id="red_button_opening"></div>
12 </div>
```

Ukázka 6.2: Obalující elementy pro každý vizuální prvek ze souboru `index.html`.

Inicializace všech vizuálních prvků probíhá v modulu `app.js`, který rovněž řídí přepínání mezi scénami a také předání klávesových stisků aktuálně aktivní scéně. Přepínání mezi scénami je implementováno pomocí jednoduchého routeru, který v případě přesměrování schová aktuálně aktivní scénu, nastaví nově zaslanou scénu jako aktivní, následně nové scéně předá případné parametry a zobrazí ji.

Lokální úložiště

Použití lokálního úložiště je vzhledem k velké rozdílnosti zařízení podporujících HbbTV problematické. Existují zařízení, která nepodporují ukládání v `localStorage`, nebo je tato paměť tak malá, že je nemožné toto úložiště použít. Pokud nastane některý z těchto případů, je potřeba data ukládat do HTTP cookie. Uložení do HTTP cookie ovšem nemusí zajistit perzistentní uložení přihlašovacích údajů v případě odpojení zařízení od přívodu energie, nebo při výpadku. Proto je toto úložiště využito pouze v případě, že předchozí možnost selže. O uložení dat se stará modul `Storage`, který obsahuje metodu `get` pro získání položky s daným klíčem a metodu `set` pro uložení hodnoty do položky s daným klíčem. Dále se zde nachází funkce `localStorageTest`, která je zavolána při startu aplikace a určuje, jaké z úložišť se má použít.

```
1 const localStorageTest = () => {
2   const test = 'test';
3   try {
4     localStorage.setItem(test, test);
5     localStorage.removeItem(test);
6     useStorage = true;
7   } catch (e) { // Pokud dojde k-vyjimce je treba vyuzit cookies
8     useStorage = false;
9   }
10 };
```

Ukázka 6.3: Funkce pro rychlé zjištění, které úložiště se má v aplikaci použít.

Ikony

Veškeré ikony (assets) použité v aplikaci jsou získány zdarma ze stránky *FLATICON*⁶. Autor ikony pro retweet je *Daniel Bruce*, ostatní ikony jsou vytvořeny společností *Google* jako součást Material design tématu. Posledním použitým obrázkem je logo TMDb, jehož povinné zobrazení vychází z pravidel používání.

Aby se zamezilo pro uživatele nepříjemnému „prokliknutí“, které je způsobeno postupným načítáním ikon, jsou vloženy pomocí nástroje *Leshy SpriteSheet Tool*⁷ do jednoho společného obrázku tzv. „spritesheetu“.



Obrázek 6.7: Spritesheet obsahující všechny ikony použité v aplikaci.

⁶Adresa stránky FLATICON je <https://www.flaticon.com>.

⁷Adresa nástroje Leshy SpriteSheet Tool je <https://www.leshylabs.com/apps/sstool/>

Kapitola 7

Testování

Testování lze rozdělit na dvě hlavní části, a to na testování v internetovém prohlížeči během vývoje, k čemuž slouží plugin FireHbbTV popsany v sekci 7.1, a testování na reálném HbbTV terminálu, které bylo realizováno pomocí testovacího prostředí firmy Mautilus s.r.o. GlobalMux (viz sekce 7.2).

7.1 FireHbbTV

Během vývoje lze pro testování HbbTV aplikací použít plugin pro internetový prohlížeč Firefox, který se jmenuje *FireHbbTV*. Plugin umožňuje Firefoxu interpretovat jazyk CE-HTML, a tedy i zobrazit HbbTV aplikace. Velkou výhodou je rovněž to, že tento plugin obsahuje automatickou kontrolu syntaxe CE-HTML, která v případě nějaké syntaktické chyby informuje vývojáře.



Obrázek 7.1: FireHbbTV plugin výchozí obrazovka.

Aby ovládání aplikace běžící v FireHbbTV odpovídalo reálnému zařízení, nachází se zde virtuální dálkový ovladač, který pro specifické klávesy na běžné klávesnici simuluje události na reálném HbbTV terminálu. Například stisk směrových tlačítek simuluje stisk navigačních kláves nebo například stisk klávesy „b“ vyvolá událost odpovídající stisku modrého tlačítka na dálkovém ovladači.

```
Left: LEFT
Right: RIGHT
Up: UP
Down: DOWN
OK: RETURN
Back: BACKSPACE

0: 0
1: 1
2: 2
3: 3
4: 4
5: 5
6: 6
7: 7
8: 8
9: 9

Red: R
Green: G
Yellow: Y
Blue: B

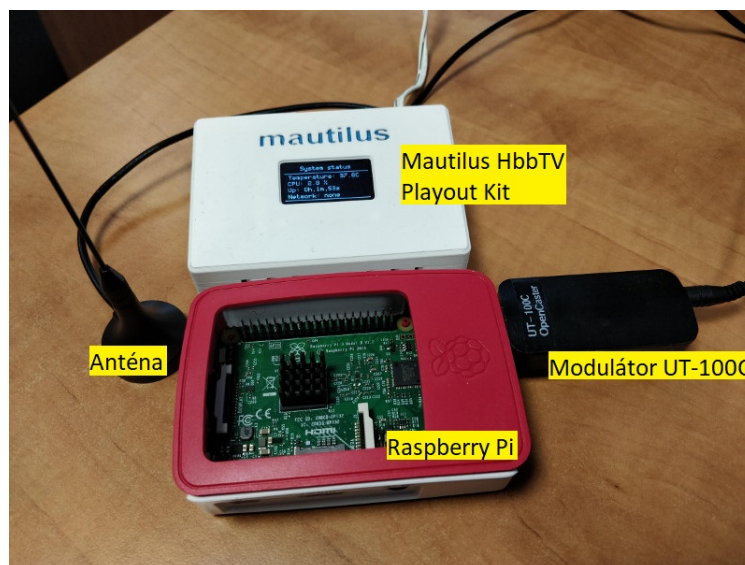
Play: SHIFT+P
Pause: CTRL+P
Play/Pause: P
Stop: S
Fast Forward: >
Fast Rewind: <
```

Obrázek 7.2: FireHbbTV plugin simulace dálkového ovladače.

Tento plugin je zcela zdarma, bohužel však již není od Firefox verze 57 dostupný, tudíž je potřeba mít nainstalovanou starší verzi tohoto prohlížeče.

7.2 Testování na reálných zařízeních

Pro testování na reálných zařízeních bylo zvoleno HbbTV Playout kit od společnosti Mautilus s.r.o. Toto řešení je postavené na volně dostupném nástroji *Opencaster*¹ od italské firmy Avalpa. K vytvoření testovacího prostředí je zapotřebí Raspberry Pi, jako řídicí jednotka, a DVB-T modulátor UT-100C s vysílací anténou. Na řídicí jednotce je použit operační systém Raspbian², na němž běží Opencaster používající skriptovací jazyk Python pro generování transportního streamu. Ilustrační fotografii celého řešení lze vidět na obrázku 7.3. Na této fotografii se nachází rovněž i produkt *Mautilus HbbTV Playout Kit*³, který je komerční variantou výše popsaného zařízení.



Obrázek 7.3: HbbTV Playout kit.

K otestování HbbTV aplikace je nutno nastavit DVB-T modulátor tak, aby parametry odpovídaly české normě DVB-T. Vzhledem k tomuto byla zvolena frekvence 546MHz s šířkou pásma 8MHz. Pro modulaci je použito kódování QPSK (více informací o tomto kódování například viz [10]) s ochranným intervalem $\frac{1}{4}$ a rychlostí kódování $\frac{3}{4}$, což ve výsledku dává celkovou šířku pásma 7.465 Mbps pro daný multiplexer. Toto nastavení zcela postačuje pro otestování jednoho televizního kanálu.

Pro změnu frekvence, signalizace a dalších parametrů slouží webová administrace. Tyto hodnoty jsou uloženy na lokálním úložišti a používají se pro generování multiplexeru a nastavení výstupu modulátoru DVB-T.

¹Stránka repozitáře nástroje Opencaster na serveru Github je <https://github.com/aventuri/opencaster>.

²Webová stránka OS Raspbian je <https://www.raspbian.org/>.

³Více informací o produktu Mautilus HbbTV Playout Kit lze nalézt na stránce <https://www.mautilus.com/products/hbbtv-playout-kit/>.

Status



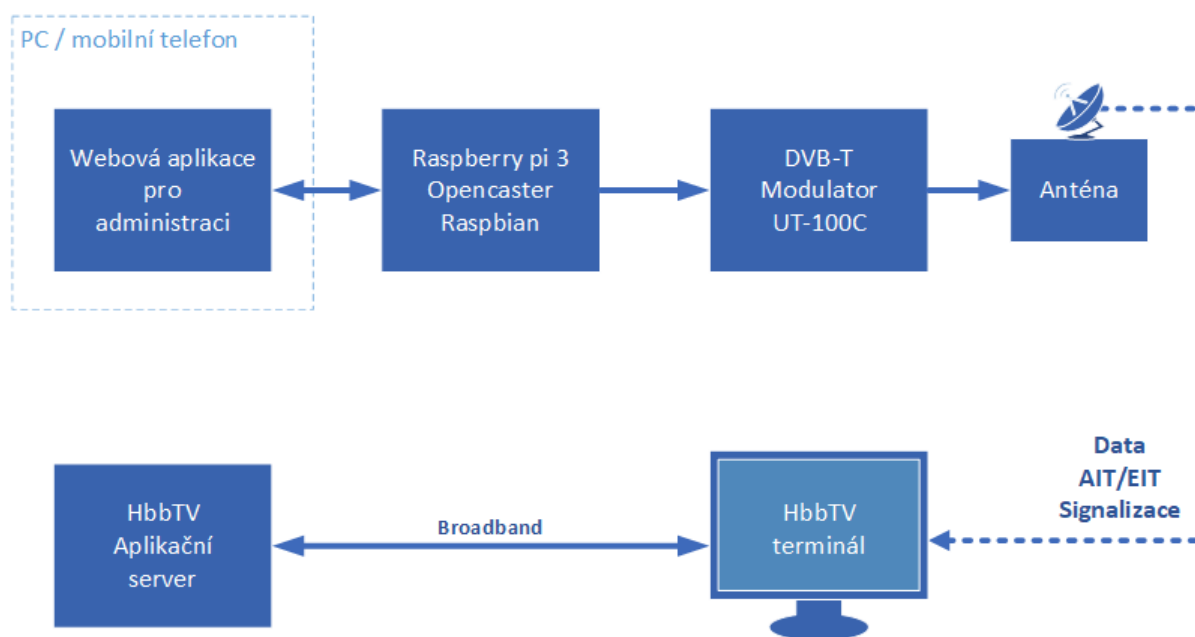
Obrázek 7.4: Webová administrace řešení pro testování na reálných zařízeních - přehled.

Application	Service 1
Name <input type="text" value="Mautilus"/>	Name <input type="text" value="HbbTV Global"/>
Frequency <input type="text" value="546000"/>	Uri <input type="text" value="http://hbttv01.mautilus.com/global_tvcon/"/>
Bandwidth <input type="text" value="8000"/>	Path <input type="text" value="index.html"/>

Obrázek 7.5: Webová administrace řešení pro testování na reálných zařízeních - nastavení.

Aby celý řetězec bylo možné otestovat, je zapotřebí televizor s podporou HbbTV a připojením na internet, ze kterého dojde následně ke stažení aplikace. Jak bylo zmíněno v kapitole 2.2.3, je nutné zajistit, aby koncové zařízení mělo k dispozici AIT (Application information table), která nese informaci o HbbTV aplikaci. Pokud je vše výše popsané splněno, stačí již jen nastavit transportní protokol na HTTP a nastavit automatické spouštění aplikace. Pro otestování aplikace na reálném televizoru je nutné mít naladěný testovací multiplex a přepnout na testovací kanál. Aplikace se poté automaticky spustí.

Schéma celého vysílacího řetězce DVB-T lze vidět na obrázku 7.6.



Obrázek 7.6: Schéma vysílacího řetězce DVB-T.

7.2.1 Referenční zařízení

Firma Mautilus s.r.o., ve které byla výsledná aplikace otestována, disponuje velkým množstvím zařízení (více než 100 zařízení) podporujících HbbTV. Z tohoto množství byly pro referenční testování vybrány následující televizory různých značek a modelů:

Značka	Model	Rok	Úhlopříčka	Rozlišení
Samsung	UE49MU7072	2017	49 palců (125 cm)	Ultra HD (3840 x 2160)
	UE55K5602	2016	32 palců (80 cm)	Full HD (1920 x 1080)
	UE55HU8500TXXH	2014	55 palců (139 cm)	Ultra HD (3840 x 2160)
LG	43UJ701V	2017	43 palců (109 cm)	Ultra HD (3840 x 2160)
	32LF631V	2015	32 palců (80 cm)	Full HD (1920 x 1080)
Sony	KDL-32WD759	2017	32 palců (80 cm)	Full HD (1920 x 1080)
	KD-43XD8099	2016	43 palců (109 cm)	Ultra HD (3840 x 2160)
Grundig	43VLE6625BP	2017	43 palců (108 cm)	Full HD (1920 x 1080)
Skyworth	32E2000	2017	32 palců (81 cm)	HD Ready (1366 x 768)
Panasonic	TX-40DX603E	2016	40 palců (100 cm)	Ultra HD (3840 x 2160)

Tabulka 7.1: Seznam referenčních zařízení, na kterých byla aplikace otestována.

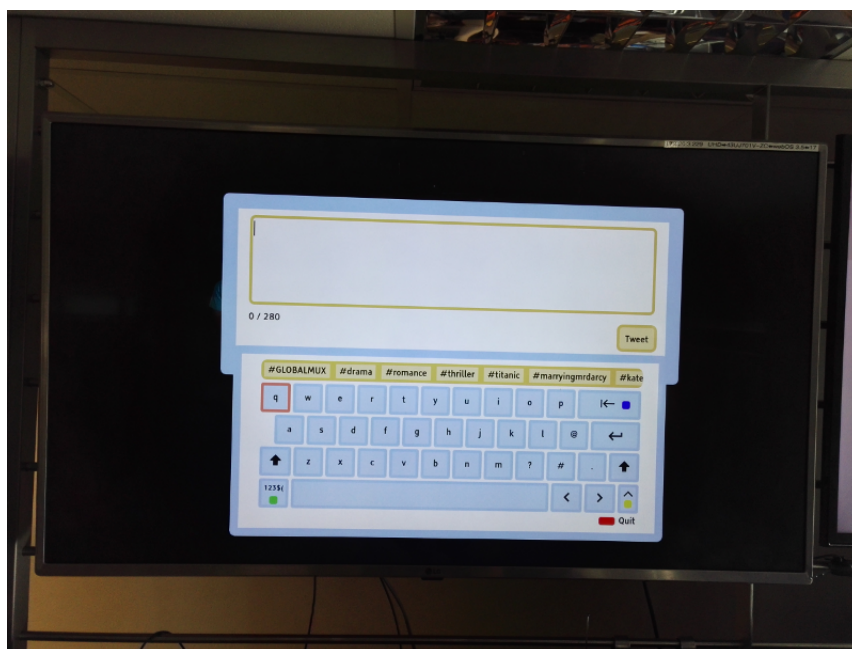
Ukázku tří hlavních scén aplikace na reálných zařízeních lze spatřit na obrázcích níže.



Obrázek 7.7: Scéna pro přihlášení na televizoru Sony KDL-32WD759.



Obrázek 7.8: Scéna s příspěvky na televizoru Sony KDL-32WD759.



Obrázek 7.9: Scéna pro přihlášení na televizoru LG 43UJ701V.

Kapitola 8

Závěr

Cílem této práce bylo navrhnout a realizovat řešení umožňující propojení digitálního televizního vysílání DVB se sociální sítí Twitter pro platformu HbbTV. Výstupem této práce je systém v architektuře klient-server, kde klientem je HbbTV aplikace běžící na hybridním terminálu a server je aplikace pro prostředí Node.js, která slouží pro komunikaci HbbTV aplikace s Twitter API a s dalšími použitými servery. Obě části systému jsou napsány v jazyce JavaScript s využitím konstrukcí z EcmaScriptu verze 6.

Aby se docílilo co nejlepší relevantnosti příspěvků (tweetů) získaných ze sociální sítě Twitter, bylo nutné data získaná z televizoru rozšířit o dodatečná metadata. Tato metadata jsou získána ze serverů The Movie Database a The Open Movie Database.

Pro určení relevantnosti tweetů s daným pořadem byla použita metoda TF-IDF v logaritmické variantě. Tato varianta byla během testování určena jako ta, jež vrací nejvíce relevantní výsledky. Před samotným určením relevantnosti dochází k odstranění nevhodných příspěvků pomocí testu na přítomnost zakázaných slov a také k odstranění slov, která většinou nemají žádnou přidanou informaci v textu, tzv. stopwords.

Výsledný systém byl otestován pomocí testovacího prostředí firmy Mautilus s.r.o. na deseti reálných zařízeních různých značek a modelů.

Dalšími rozšířeními tohoto systému mohou být přidání podpory ovládání pomocí mobilního telefonu dle standardu HbbTV 2.0, rozsáhlejší prohledávání sociální sítě Twitter pomocí premium API nebo nabízení dalších předmětů zájmu přímo z textu popisu daného programu.

Literatura

- [1] *HbbTV Developer Page*. [Online; navštíveno 12.11.2017].
URL <http://www.hbbtv-developer.com/>
- [2] *The Movie Database*. [Online; navštíveno 20.03.2018].
URL <https://www.themoviedb.org/>
- [3] *The Open Movie Database*. [Online; navštíveno 20.03.2018].
URL <http://www.omdbapi.com/>
- [4] Bonzanini, M.: *Mining Twitter Data with Python*. [Online; navštíveno 22.03.2018].
URL <https://marcobonzanini.com/2015/03/09/mining-twitter-data-with-python-part-2/>
- [5] Christopher D. Manning, P. R.; Schütze, H.: *Introduction to Information Retrieval*. Cambridge University Press, 2008, iSBN: 0521865719.
- [6] European Telecommunications Standards Institute: *TS 102 796 - V1.1.1 - Hybrid Broadcast Broadband TV*. [Online; navštíveno 17.10.2017].
URL http://www.etsi.org/deliver/etsi_ts/102700_102799/102796/01.01.01_60/ts_102796v010101p.pdf
- [7] HbbTV Association: *HbbTV 2.0.1 Specification*. [Online; navštíveno 17.10.2017].
URL https://www.hbbtv.org/wp-content/uploads/2015/07/HbbTV-SPEC20-00023-001-HbbTV_2.0.1_specification_for_publication_clean.pdf
- [8] Loveček, V.: *Aplikační server pro HbbTV*. Brno: Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2014.
- [9] Malý, M.: *OAuth – nový protokol pro autentizaci k vašemu API*. [Online; navštíveno 09.12.2017].
URL <https://www.zdrojak.cz/clanky/oauth-novy-protokol-pro-autentizaci-k-vasemu-api/>
- [10] Maxim Integrated Products: *QPSK Modulation Demystified*. [Online; navštíveno 05.05.2018].
URL <https://www.maximintegrated.com/en/app-notes/index.mvp/id/686>
- [11] Nayak, P.; Raghavan, P.: *Introduction to Information Retrieval - Lecture 6: Scoring, Term Weighting and the Vector Space Model*. [Online; navštíveno 19.04.2018].
URL <http://www.uio.no/studier/emner/matnat/ifi/INF3800/v16/slides/stanford-pdf/lecture6-tfidf.pdf>

- [12] Open IPTV Forum: *Open IPTV Forum Release 2 Specification*. [Online; navštíveno 17.10.2017].
URL <http://www.oipf.tv/web-spec/volume5.html>
- [13] Ranks: *Ranks NL*. [Online; navštíveno 21.04.2018].
URL <https://www.ranks.nl/stopwords>
- [14] Russell, M. A.: *Mining the Social Web, 2nd Edition*. O'Reilly Media, Inc., 2013, iSBN: 9781449368180.
- [15] Toman, J.; Procházka, I.: *Digitální vysílání DVB obecně - Technické základy*. [Online; navštíveno 10.12.2017].
URL <http://www.ceskatelevize.cz/vse-o-ct/technika/digitalni-vysilani-dvb-obecne/technicke-zaklady/>
- [16] Twitter, Inc.: *Twitter Developers*. [Online; navštíveno 09.11.2017].
URL <https://developer.twitter.com>

Příloha A

Obsah přiloženého paměťového média

Na přiloženém paměťovém médiu se nachází následující adresáře:

- **src** - zdrojové soubory aplikace
 - adresář **tv-app** - zdrojové kódy HbbTV aplikace
 - adresář **server** - zdrojové kódy serverové části
 - soubor **README.md** - popis instalace a spuštění systému v anglickém jazyce
- **doc** - zdrojové kódy tohoto dokumentu
- **tools** - nástroje umožňující spustit HbbTV aplikaci - instalační soubor pro Internetový prohlížeč Mozilla Firefox verze 56 ve formátu **.exe**
- **results** - výsledky testování různých variant metody TF-IDF s testovacími daty (uvnitř souboru **results.xlsx**)

Příloha B

Spuštění HbbTV aplikace a serveru

Pro správné spuštění aplikace i serveru je nutné mít nainstalovaný ekosystém *Node.js* (získaný například přímo ze stránky tohoto nástroje <https://nodejs.org/en/>) a dále *npm*, což je správce JavaScriptových balíčků.

To, zda je ekosystém správně nainstalovaný lze zjistit zadáním příkazu `npm -v` do terminálu. V případě, že vše proběhlo v pořádku, je nutné zadáním příkazu `npm install` v adresáři HbbTV aplikace (`tv-app`) i v adresáři serveru (`server`) stáhnout požadované balíčky pro *Node.js*.

Před samotným spuštěním je nutné správně nastavit URL serverové části (parametr `serverUrl`) v souboru `config.js`, který se nachází v adresáři `src` ve složce s HbbTV aplikací. URL serveru je ve výchozím nastavení 127.0.0.1 na portu 1337, ale je možné jej změnit v souboru `config.js` v adresáři serveru.

Dalším krokem je sestavení obou nástrojů pomocí příkazu `npm run build` (případně `npm run watch` pro kontinuální sestavování) opět spuštěném v adresářích obou částí systému.

Následně je třeba spustit některý z http serverů, který je nasměrovaný do adresáře s HbbTV aplikací (například server *WAMP*¹) nebo přímo příkazem `npm run start` uvnitř tohoto adresáře.

Pokud server již nebyl spuštěn příkazem `npm run watch`, je třeba jej spustit příkazem `npm run start`.

Po zobrazení hlášky `console.log('Server running at http://ip:port/')` je server úspěšně spuštěn a je možno se na něj připojit pomocí HbbTV aplikace.

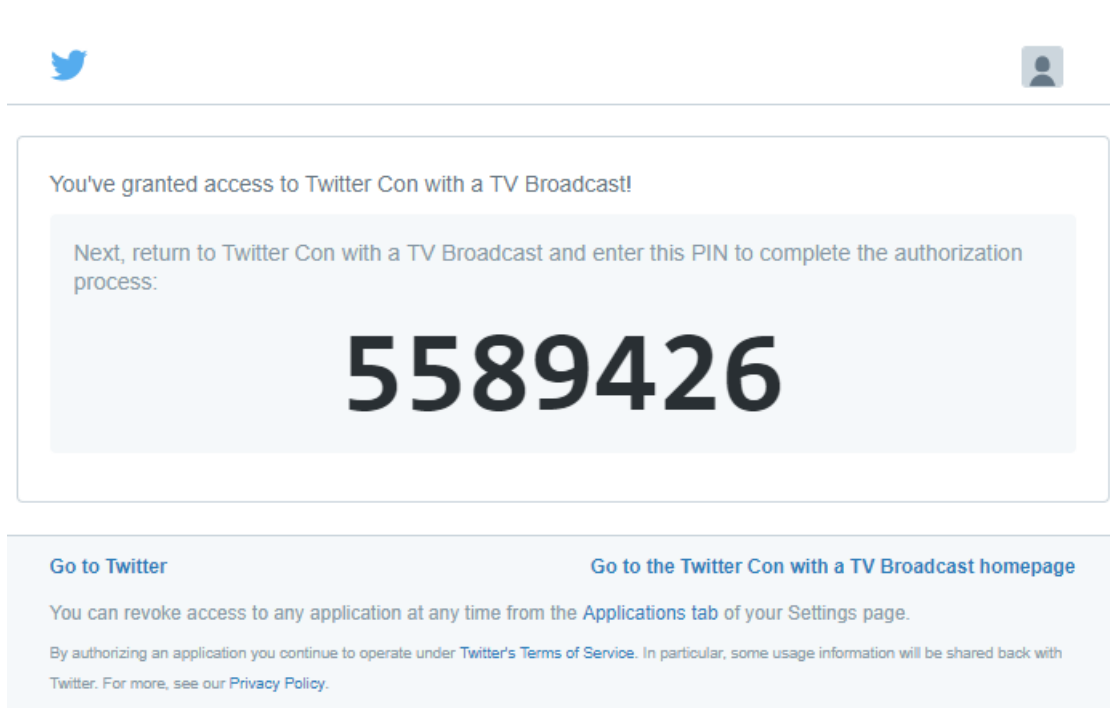
¹Lze získat na adrese <http://www.wampserver.com/en/>

Příloha C

Autorizace pomocí PINu na sociální síti Twitter



Obrázek C.1: Stránka pro autorizaci aplikace pro využívání uživatelského účtu.



Obrázek C.2: Stránka se zobrazeným PINem, který je nutno přepsat do HbbTV aplikace.