

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

PORTÁLOVÉ RIEŠENIE IPTV

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

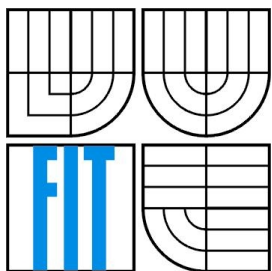
AUTOR PRÁCE
AUTHOR

BC. MICHAL PÁPAI

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

PORTÁLOVÉ ŘEŠENÍ IPTV

IPTV PORTAL

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. MICHAL PÁPAI

VEDOUCÍ PRÁCE
SUPERVISOR

MGR. ROMAN TRCHALÍK

BRNO 2011

Abstrakt

Tato diplomová práce je zaměřena na multicastovou technologii IPTV. Popisuje architekturu IPTV a její jednotlivé komponenty. Práce stručně popisuje architekturu IPTV a jejím hlavním jádrem je detailní analýza požadavků a návrh celkového systému pro sledování digitální televize přes Internet. Tato práce navrhuje a implementuje správu a autentizaci koncových zařízení - settopboxů, efektivní distribuci firmware pomocí multicasu a portálu, který zabezpečuje interakci s navrhovaným systémem. Celý systém je navržen jako komponenta a řeší i problematiku rozhraní pro jednoduché napojení kontrolních, dohledových a účtovacích IS. Navržená komponenta je pak integrována do reálného systému komerčního poskytovatele internetového připojení.

Abstract

This master's thesis is focused on IPTV based on multicast technology. The main aim of this work is to provide detailed analysis of system for watching TV over IP network. It also describes components of IPTV architecture. This master's thesis also discussing about authentication and administration of settopboxes, deals with efficient distribution firmware for STB over multicast and tries to propose and implement portal which provides customer services. It describes the interface development as web service for interconnecting with another management, monitoring and billing information systems. The solution is implemented as a modul into an existing IS of commercial internet service provider.

Klíčová slova

IPTV Middleware, IP STB, IPTV portal, RTSP proxy

Keywords

IPTV Middleware, IP STB, IPTV portal, RTSP proxy

Citace

Michal Pápai: Portálové riešenie IPTV, diplomová práce, Brno, FIT VUT v Brně, 2011

Portálové riešenie IPTV

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Mgr. Romana Trchalíka

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Michal Pápai
25.05.2011

Poděkování

Rád bych tímto poděkoval Mgr. Romanu Trchalíkovi za poskytnuté rady a odborné vedení celé práce.

© Michal Pápai, 2011

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	3
1.1 Stručný obsah kapitol.....	3
2 Architektúra IPTV.....	4
2.1 Live vysielanie.....	5
2.2 Video OnDemand (VoD).....	6
2.3 Middleware.....	7
2.4 Koncové zariadenia (IP STB).....	7
3 Analýza požiadaviek.....	8
3.1 Funkcionalita platformy.....	8
3.2 STB.....	11
3.3 Rozhranie	14
3.4 Bezpečnosť a komunikácia.....	14
3.5 Zdroje.....	15
3.6 Dokumentácia.....	15
4 Návrh systému.....	16
4.1 Idea riešenia.....	16
4.1.1 Elektronický programový sprievodca.....	19
4.1.2 VOD.....	20
4.2 Use case model.....	21
4.2.1 Správa kanálov.....	21
4.2.2 Správa kategórií.....	21
4.2.3 Rodičovský kód.....	22
4.2.4 Samoobsluha.....	22
4.2.5 Videopožičovňa.....	23
4.2.6 Zobrazenie EPG.....	23
5 Realizácia kľúčových častí.....	24
5.1 Portál.....	24
5.1.1 Uživatelské rozhranie.....	24
5.1.2 Prehrávanie obsahu.....	25
5.1.3 Ovládanie	27
5.2 Multicast (mcast) server.....	29
5.2.1 Štruktúra mcast správy.....	30
5.2.2 Šírka pásma multicastového streamu.....	31
5.2.3 Šifrovanie.....	31
5.3 Autentizačný server.....	32
5.3.1 Komunikácia.....	32
5.3.2 Šifrovacie kľúče.....	33
5.4 Aplikačný server.....	34
5.4.1 Štruktúra správy.....	34
5.4.2 Správy.....	36
5.5 RTSP proxy server.....	40

5.5.1 RTSP metódy.....	40
5.5.2 Komunikácia.....	42
5.6 Rozhranie systému.....	44
5.6.1 Webová služba.....	44
5.6.2 Middleware – SOAP server.....	45
5.6.3 CRM systém – SOAP klient.....	47
5.7 Použité technológie pri implementácii.....	48
5.7.1 Aplikačný a RTSP proxy server.....	48
5.7.2 Komponenty middlewaru.....	48
5.7.3 Databázový systém.....	48
6 Záver.....	49
Literatura.....	51
Zoznam príloh.....	52

1 Úvod

Postupná digitalizácia a prechod z analógového vysielania naštartoval rozvoj nových technológií. Digitálna televízia sa vyvíjala niekoľko rokov a poznáme ju najmä ako DVB-S (satelitné vysielanie), DVB-C (káblové vysielanie) respektíve ako mobilné vysielanie DVB-H. Posledné pokroky umožnili poskytovať digitálne televízne služby i prostredníctvom širokopásmových sietí a poznáme ich pod pojmom IPTV (Internet Protocol Television). IPTV predstavuje systém v ktorom sa digitálna televízia šíri prostredníctvom IP protokolu v počítačovej sieti. Nasadenie IPTV pre rôzne typy širokopásmových prístupových sietí ako je napríklad DSL, GPON a v neposlednom rade ethernet umožnilo ISP (poskytovateľom internetového pripojenia) ponúknuť zákazníkovi tzv. triple-play služby (Internet, VoIP, IPTV). Televízia šírená prostredníctvom IP protokolu sa významne odlišuje od tradičných spôsobov vysielania. Jej veľkým benefitom, vďaka IP protokolu, je možnosť obojsmerného komunikačného spojenia medzi poskytovateľom IPTV služieb a koncovým zákazníkom. Takto vie poskytovateľ jednoznačne identifikovať odberateľa digitálnej televízie a na základe toho poskytovať okrem štandardného vysielania aj interaktívne a flexibilné služby ako je napríklad video na želanie (VoD), služby založené na charakteru PPV (Pay Per View), ktoré umožňujú sledovať určité programy v pevne vysielacej dobe na základe predplatného alebo služby typu nPVR a time-shifting. V prípade nPVR sa jedná o personálny sieťový videorekordér, kde jednotlivé programy sa nenahrávajú na strane zákazníka ale priamo u poskytovateľa. Obdobne je to aj v prípade time-shifting (posun v čase), kde zákazník si vie pozrieť jednotlivé programy napr. o 2 dni späť. Aby tieto služby mohli byť zabezpečené a poskytnúť tak zákazníkovi plnohodnotné využívanie, ktoré prináša IPTV, je nevyhnutné nasadenie platformy zvanej middleware.

Každá služba z balíka triple-play má svoje vlastné mechanizmy pre zaistenie managementu infraštruktúry a užívateľov. S rastúcim počtom zákazníkov rastú aj požiadavky pre spoľahlivé a robustné riešenia. Cieľom tejto práce je detailne analyzovať požiadavky pre návrh systému, ktorý by umožnil sledovanie TV prostredníctvom IP protokolu a zároveň implementovať správu tohto systému ako nezávislú komponentu s možnosťou integrácie do IS [7] komerčného poskytovateľa internetového pripojenia.

1.1 Stručný obsah kapitol

- Druhá kapitola tejto práce predstavuje čitateľovi základnú architektúru IPTV. Popisuje možnosti doručenia audia/videoa cez počítačovú sieť a prehľad používaných protokolov. Vysvetľuje význam softwarovej komponenty, middleware, na ktorej stojí celá funkcionálna časť pre sledovanie TV založenej na IP protokole.
- V tretej kapitole je obsiahnutá kompletná analýza požiadaviek na vyvíjaný middleware a portál. V tejto kapitole sú jasne definované funkčné požiadavky, ktoré budú východiskom pre návrh celého systému.
- Štvrtá kapitola predstavuje ideu riešenia s krátkym úvodom do problematiky spracovania niektorých podkomponent a pokračuje návrhom v podobe use-case modelu.
- Piata kapitola popisuje realizáciu kľúčových častí pre zabezpečenie sledovania IPTV
- Posledná kapitola je záver, ktorá obsahuje zhodnotenie vyvíjaného systému s možnosťami jeho ďalšieho rozšírenia.

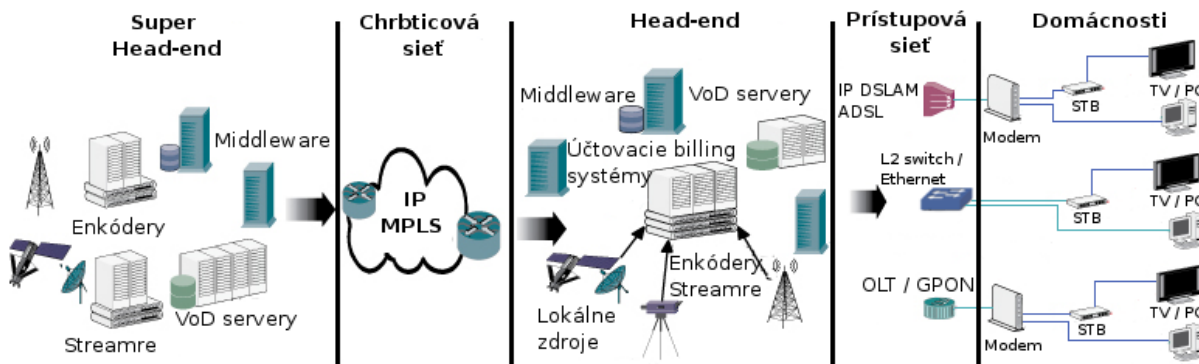
2 Architektúra IPTV

IPTV ako už bolo naznačené v úvode, je digitálna televízia ktorá je šírená prostredníctvom IP protokolu. IPTV by sa dala charakterizovať týmito nasledujúcimi komponentami, ktoré tvoria head-end celého systému:

- Enkóдеры a streamre
- IPTV middleware
- VoD systém
- Systém pre ochranu obsahu

Podľa [1] head-end je logická časť štruktúry, ktorá pozostáva najmä zo zariadení, ktoré spracúvajú digitálny a analógový (dnes už na ústupe) televízny signál. Zdrojom týchto signálov môže byť predovšetkým satelitné, pozemné alebo káblové vysielanie. Väčšinou sa jedná o jedno zariadenie, ktoré kombinuje digitálny enkóder a streamer, z ktorého už vystupujú televízne kanály ako audiovizuálne streamy zabalené do IP paketov vhodné pre ďalšie šírenie prostredníctvom počítačovej siete. V prípade veľkých operátorov tieto streamy nie sú priamo distribuované koncovým zákazníkom ale vstupujú do chrbticovej siete, kde sa agregujú a prípadne distribujú v peeringových centrách ďalším operátorom. V tomto prípade hovoríme o super head-ende.

Lokálni poskytovatelia IPTV služieb majú lokálne head-ends i pre spracovanie regionálnych vysielaní a aby bola zaistená redundancia, odoberajú z peeringového centra takto upravené streamy a inserujú ich do svojej programovej ponuky. Poskytovateľ buduje prístupovú sieť do ktorej sú následne pripojené klientské zariadenia settopboxy (STB), ktoré umožňujú prijať jednotlivé audiovizuálne streamy a zobraziť ich na TV prijímači. Tento model ilustruje obr.č.1.



Obrázok 1: Architektúra IPTV[1]

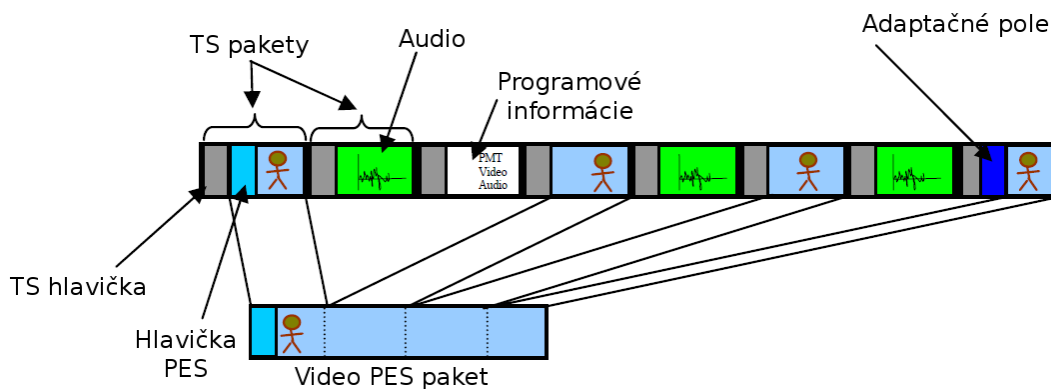
Služby poskytované prostredníctvom IPTV si kladú vysoké požiadavky taktiež i na sieťovú infraštruktúru a jej uzly. Pre doručenie multimediálneho obsahu či už pomocou multicastu (live tv) alebo unicastu (VoD) musí byť zabezpečená garantovaná šírka prenosového pásma a s ňou spojené ďalšie kontrolné IP mechanizmy. Cieľom je teda zabezpečiť poskytovanie IPTV služieb na rovnakej úrovni spoľahlivosti ako bežný operátor káblovej TV.

Nasledujúce dve podkapitoly stručne popisujú metódy a protokoly doručenia multimediálneho obsahu prostredníctvom počítačovej siete.

2.1 Live vysielanie

Prepínače na sieti musia podporovať IGMP protokol (Internet Group Multicast Protocol), ktorý je používaný koncovými zariadeniami (napr. STB) pre prihlásenie a odhlásenie do/z multicastových skupín do ktorých sú streamované jednotlivé TV kanále. Pre efektívny multicasting je nevyhnutné, aby bolo IGMP implementované po celej sieti od zdroja až po najbližší prepínač do ktorého je zariadenie priamo pripojené. Navyše aby sa predišlo zbytočnému zahlcovaniu prostriedkov zasielaním IGMP správ po sieti a predovšetkým posielaním multicastového provozu na zariadenia ktoré si ho explicitne nevyžiadali, zavádza sa technika IGMP snooping. *Prepínač ktorý podporuje IGMP snooping umožňuje analyzovať L2 rámce a hlavičky L3 paketov a zaznamenávať si prípadný IGMP provoz a pak určovať príslušnosti svojich portov buď k linke na ktorej je smerovač alebo koncové zariadenia [6].*

Pre transport audiovizualneho obsahu a programových informácií nad IP protokolom sa používa kontajner MPEG-2 Transport Stream (TS). TS tvoria 188 bajtové pakety, každý so 4 bajtovou hlavičkou. Payload paketu obsahuje paketizované elementárne streamy (PES) typicky video, audio a taktiež môže obsahovať aj programové informácie. PES pakety sú rozdelené do 184 bajtových blokov tak aby vyplnili payload TS paketu. Majú premenlivú veľkosť hlavičky, ktorá je na začiatku payloadu TS paketu. Môže sa stať že PES paket nie je násobkom 184 bajtov, preto je posledný blok doplnený o výplň vid' obrázok č. 2. Pretože hlavičky PES paketov a taktiež adaptačné pole obsahujú časové informácie, nie sú potrebné ďalšie signály pre zaistenie synchronizácie jednotlivých streamov.



Obrázok 2: MPEG-2 Transport Stream multiplexujúci video, audio a programové informácie

2.2 Video OnDemand (VoD)

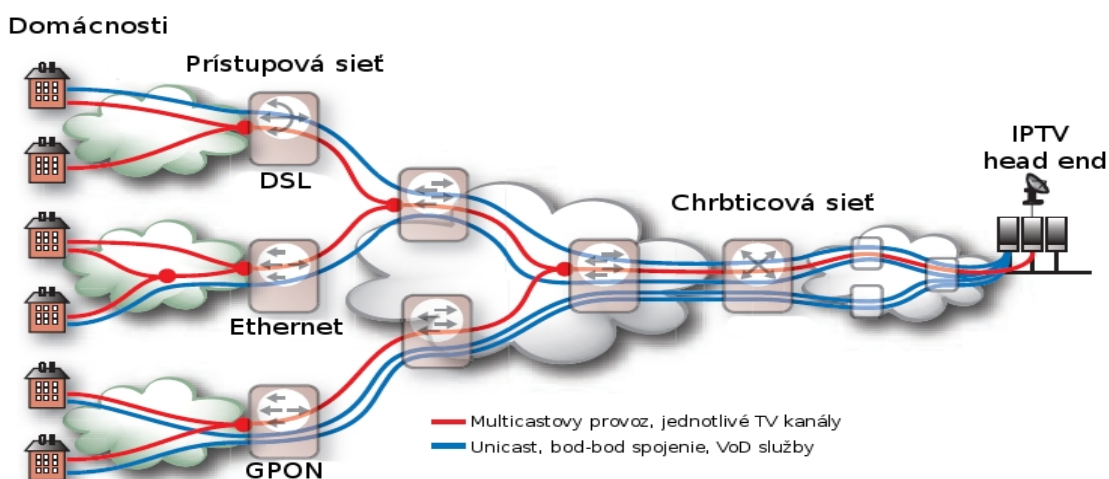
Pre poskytovanie služieb typu VoD pod ktoré spadá i sieťový video rekordér (nPVR) sa využíva unicastový model, ktorý zabezpečuje doručenie unikátneho obsahu pre jedného užívateľa. Unicastové služby vytvárajú bod-bod spojenie medzi užívateľským STB a centrálnym zdrojom v sieti poskytujúci audiovizuálny obsah. Tieto typy služieb vyžadujú vysoké nároky na šírku pásma na chrbticovej sieti, zaručenú kvalitu služieb a odozvu. Očakáva sa, že vďaka IPTV službám, ktoré umožňujú interaktívne sledovanie TV sa bude podiel týchto služieb zvyšovať. V dôsledku toho, je nevyhnutné tento obsah rozložiť a distribuovať koncovým užívateľom čo najbližšie s cieľom minimalizovať unicastového toku a znížiť tak zaťaženie na chrbticovej sieti.

Vo veľa prípadoch užívateľ môže požadovať rôzne streamy pre ten istý obsah, napríklad výber zvukovej stopy pre video, titulky v rôznych jazykoch, alternatívne pohľady kamery. V prípade použitia transportného protokolu MPEG-2 TS ako v prípade živého vysielania sú dve možnosti:

- posielanie všetkých pridružených streamov obsahu a nechať užívateľa si vybrať požadovaný video/audio stream
- alebo na základe požiadavky by server celý tok demultiplexoval, vybral správne PES pakety, a znovu ich zlúčil do výsledného streamu pre užívateľa

V prvej variante ide o zbytočné plytvanie šírky pásma, napríklad ak je dostupných veľa zvukových stôp a užívateľ ich nepotrebuje. Druhá možnosť zvyšuje náklady na výpočetné prostriedky. Na základe týchto faktorov sa pre tento typ služieb využíva protokol RTP (Real Time Protocol). Ďalším faktorom, kde sa javí ako MPEG-2 TS nevýhodné je v používaní komplexnejších vlastností ako je pozastavenie obrazu, zmena smeru resp. rýchlosť prehrávania.

RTP multimediálne dáta nemultiplexuje, ale miesto toho sú posielané samostatne v jednotlivých RTP streamoch. Z tohto dôvodu nie je potrebné explicitné demultiplexovanie na strane klienta. Každý RTP stream musí niesť informáciu, aby koncové zariadenie vedelo dané streamy v prípade potreby synchronizovať. RTP využíva pridružený protokol, RTP Control Protocol, ktorý poskytuje kvalitu služieb, monitorovanie a informácie potrebné pre synchronizáciu. Pre detailnejší popis je možné nahliadnuť do [4].



Obrázok 3: Multicastový a unicastový provoz[5]

2.3 Middleware

Middleware, ako ďalšiu komponentu IPTV headendu, tvorí rada aplikácií do jednotného softwarového systému ktorý zabezpečuje IPTV služby koncovým užívateľom. IPTV middleware je typicky klient/server architektúra, kde klienti sú STB. Middleware implementuje základné operácie pre chod IPTV systému a funkcie ktoré umožňujú zákazníkovi sledovať IPTV. Predovšetkým ide o tieto funkcie:

- autentizácia STB
- výber dostupných TV kanálov
- elektronický programový sprievodca (EPG)
- VoD služby

Podľa [2] middleware zahrňuje:

- Management a správu užívateľov – Middleware poskytuje dynamické užívateľské rozhranie (UI) pre STB. Mimo iného tiež spravuje a monitoruje STB pre účtovacie a bezpečnostné účely. Ďalej poskytuje možnosti pre vzdialené ovládanie STB na báze WEB technológií a možnosti vzdialených aktualizácií. Rôzne UI môžu byť pridelené rôznym užívateľom alebo viacerým užívateľom v skupine.
- Management obsahu – Software musí poskytovať flexibilnú kontrolu nad obsahom. Napríklad definovať rôzne ceny za rôzne typy obsahu, alebo vytvárať rôzne programové balíky pre rôzne typy obsahu.
- Management infraštruktúry – Middleware dovoľuje administrátorom možnosť konfigurovať role a funkcie a kontrolovať vstupné a výstupné parametre. Middleware musí poskytovať manuálne a automatické nástroje pre vyvažovanie záťaže, monitorovať dostupnosť všetkých serverov a v prípade poruchy alebo výpadku aktivovať záložné procedúry, ktoré zaisťujú hladký chod služieb.

Postupom času výrobcovia zariadení a spoločnosti zaoberajúce sa produktami pre poskytovanie IPTV služieb vytvorili dve metodológie a programovacie paradigmy pre tvorbu IPTV middlewaru a s ním spojené UI. Prvý spôsob je budovanie UI tak ako ho popisuje [2], ktorý implementuje 90% spoločností. Ďalšia možnosť je tvorba užívateľského rozhrania ako samostatná aplikácia inštalovaná do STB. Komunikácia IPTV middlewaru s okolím pre oba prístupy je znázornená v prílohe č. 1.

2.4 Koncové zariadenia (IP STB)

Koncové zariadenia obecné tzv. set-top-boxy slúžia pre príjem, dekódovanie a zobrazenie televízneho signálu na TV prijímači. V našej práci budeme ďalej pracovať len s IP STB, pre ktoré vytvárame portál. Sú to zariadenia väčšinou postavené na báze system-on-chip s OS linux. Na rozdiel od bežných STB ako ich poznáme pre digitálne terestriálne vysielanie resp. satelitné, nemajú v sebe žiadny front-end a teda pre okamžité nasadenie je to nemožné. Výnimku tvoria hotové riešenia od spoločností, ktoré dodávajú dané riešenie na kľúč. Väčšina týchto riešení je nákladná a častokrát je návratnosť investícií pre menších lokálnych poskytovateľov až desiatky rokov. Vývoj systému pre sledovanie IPTV často závisí na zariadeniach, ktoré majú byť podporované zo strany middlewaru. V posledných rokoch sa na trhu objavujú STB s firmwarom, ktorý obsahuje varianty špeciálne upraveného webového prehliadača a proprietárnym prehrávačom pre streamovaný multimediálny obsah. Tento prehrávač je možné ovládať pomocou JavaScriptu interpretovaného v prostredí vstavného web prehliadača. Toto umožňuje rýchly vývoj aplikácií-portálov pre takýto typ zariadení, čo je aj predmetom našej práce.

Pri výbere zariadení sme sa zamerali na STB s voľne prístupnou špecifikáciou JS API prípadne SDK pre programovací jazyk C. Ďalším faktorom bola aj cena týchto zariadení. Vybrané zariadenia a ich špecifikácia sa nachádza v prílohe č. 2.

3 Analýza požiadaviek

Nasledujúca kapitola obsahuje popis analýzy a návrh systému pre sledovanie IPTV. V tejto práci bude naznačený návrh systému a návrh užívateľského rozhrania pre koncové zariadenia.

Analýza pozostáva z bodov, ktoré sú kladené na systém a užívateľské rozhranie podľa [2]. Nasledujúce body požiadaviek sú rozšírené o body, ktoré boli získané komunikáciou so zadávateľom projektu. Ako prvotná požiadavka bola formulovaná na vytvorenie systému pre sledovanie IPTV už vo vybudovanej sieťovej infraštruktúre ISP. V priebehu niekoľkých schôdzok so zadávateľom projektu bol postupne vypracovaný zoznam detailnejších požiadaviek, ktoré boli získané využitím modernej metódy tzv. Brainstorming¹. Jednotlivé požiadavky vyplývajúce z komunikácie sú rozdelené do logických celkov. Každá požiadavka je charakterizovaná niekoľkými bodmi:

- poradové číslo a výstižný názov požiadavky v rámci analýzy
- znenie požiadavky
- priorita pri návrhu a implementácii systému (1- najnižšia... 5- najvyššia)
- predpokladaná náročnosť pri riešení (analýza rizík) (1- najnižšia... 5- najvyššia)

3.1 Funkcionalita platformy

Požiadavka č. 1 – Podpora lokalít

Do budúcnosti sa predpokladá nárast zákazníkov a rozširovanie sieťovej infraštruktúry. Preto je nevyhnutné zaviesť hierarchiu lokalít, na základe ktorých sa budú odvíjať funkcie a poskytované služby koncovým užívateľom.

Priorita: 1

Náročnosť: 3

Požiadavka č. 2 – Správa kanálov

Systém bude umožňovať pridávanie/editáciu kanálov. Každá položka bude mať nasledujúce atribúty:

- názov
- typ (TV, Rádio)
- kategória kanálu (spravodajský, filmový, hudobný, dokumentárny)
- implicitné nastavenie rodičovského zámku
- logo stanice
- multicast IP:port

Priorita: 5

Náročnosť: 2

Požiadavka č. 3 – Podpora viacerých zdrojov

Predpokladá sa, že budú zavedené redundantné streamovacie servery pre zabezpečenie spoľahlivosti poskytovaných služieb. Pri sledovaní TV v určitej lokalite môže dôjsť k výpadku niektorého z nich a je nevyhnutné zabezpečiť automatické prepnutie sledovaného kanálu na alternatívny zdroj.

Priorita: 5

Náročnosť: 4

1 <http://en.wikipedia.org/wiki/Brainstorming>

Požiadavka č. 4 – Programové balíčky

Jednotlivé kanály budú radené do programových balíčkov. Každý programový balík bude obsahovať názov a zoznam kanálov.

Priorita: 4

Náročnosť: 2

Požiadavka č. 5 – Programová ponuka

Pre zvýšenie granularity medzi tarifami bude zaradená ešte vlastná programová ponuka, ktorá bude obsahovať názov, popis, jednotlivé programové balíčky a atribút, ktorý bude povoľovať možnosť zakúpiť si danú ponuku v samoobsluže priamo cez portál STB.

Priorita: 4

Náročnosť: 2

Požiadavka č. 6 – Tarify

Tarify slúžia na bezprostredné pridelenie k zákazníckym STB účtom. Tarifa bude mať nasledujúce atribúty:

- názov
- popis
- štartovacia cena (jednorázová/aktivačná cena za službu)
- základný balík z programovej ponuky a jeho cenu za mesiac prevádzky

Ďalej zoznam možných doplnkových programových ponúk, ktoré si koncový zákazník bude môcť samostatne zakúpiť (ďalej tematické balíčky) a ich ceny za mesiac prevádzky. Pri každej ponuke sa bude musieť uviesť minimálny počet mesiacov predplatného. Počas tejto lehoty si zákazník v samoobsluže nebude vedieť zrušiť zakúpený tematický balíček. Tarifácia bude prebiehať mesačne a síce generovaním potrebných informácií pre operátorov billingový systém vo forme exportov resp. priamym dotazom.

Priorita: 5

Náročnosť: 3

Požiadavka č. 7 – Správa VoD kategórií

Od systému sa predpokladá, že bude podporovať nasadenie VoD, a z tohto hľadiska bude nutné zaviesť i možnosť rozdelenia multimediálneho obsahu do jednotlivých cenových kategórií. Kategórie budú špecifikované cenou a dobou zakúpenia, po ktorú si bude môcť zákazník daný film znova prehrať.

Priorita: 3

Náročnosť: 3

Požiadavka č. 8 – VoD balíky

VoD balíky budú vymedzené ako v požiadavke číslo 4, pričom špecifikácia zdrojov bude upresnená pri nasadení VoD riešenia.

Priorita: 2

Náročnosť: 4

Požiadavka č. 9 – Systém zliav a akcií

Systém bude mať umožniť aplikovať akciové ponuky s určitým časovým úsekom jej platnosti. V dôsledkom tohto budú následne upravené billingové výstupy a akcie patrične zvýraznené v samoobsluže.

Priorita: 3

Náročnosť: 3

Požiadavka č. 10 – Nastavenia systému

Kladené požiadavky na nastavenie systému:

- verzia portálu – Vynútenie pre STB používať určitý typ a verziu portálu
- povoliť/nepovoliť VoD na globálnej úrovni
- zapnúť/vypnúť „peoplemeter“ (viď požiadavka číslo 16.)
- povoliť/nepovoliť samoobsluhu na STB
- nastavenie generovania billing dát

Priorita: 3

Náročnosť: 4

Požiadavka č. 11 – Zasielanie správ na koncové zariadenia

Zasielanie správ zákazníkom priamo na TV obrazovku. Správy budú informačného charakteru typu pop-up. Správy budú odosielané na vybrané alebo všetky STB podľa filtru (minimálne však podľa lokality a typu tarify) a budú mať nasledovné atribúty:

- čas zobrazenia (od/do)
- perióda opakovania (denne, týždenne, atď.)

Priorita: 4

Náročnosť: 4

Požiadavka č. 12 – Riadiace správy

Okrem informačných správ bude STB reagovať na vzdialené príkazy zadané operátorom. Tieto príkazy sa budú v priebehu implementácie a nasadenia platformy dopĺňať.

Priorita: 3

Náročnosť: 4

Požiadavka č. 13 – Správa zariadení STB

Každý STB bude v systéme jednoznačne identifikovaný na základe jeho MAC adresy. Od systému sa očakáva pridávanie/editácia zariadení s minimálnymi atribútmi:

- MAC adresa
- model
- sériové číslo

Priorita: 5

Náročnosť: 2

Požiadavka č. 14 – STB účty

Pre STB účty sú požadované nasledovné atribúty:

- stav (billing, suspend)
- miesto pripojenia
- lokalita

- heslo pre rodičovský zámok
- tarifa
- tematické baličky
- povolenie VoD
- priradenia STB

Priorita: 5

Náročnosť: 2

Požiadavka č. 15 – Evidencia zákazníkov

Systém umožní uchovávať v DB základné informácie o koncových zákazníkoch, a to:

- ID
- osobné údaje
- adresa
- kontakt

Pričom veškeré údaje okrem ID nemusia byť povinné. Predpokladá sa, že poskytovateľ má svoj IS v ktorom vedie evidenciu svojich zákazníkov.

Priorita: 2

Náročnosť: 2

Požiadavka č. 16 – Štatistiky

Od navrhovaného systému sa predpokladá monitorovanie sledovanosti jednotlivých programov. Tieto dáta budú ukladané pre prípadné vyhodnotenia pomocou iných CRM systémov.

Priorita: 3

Náročnosť: 4

Požiadavka č. 17 – Samoobsluha

Zakazníci si budú môcť priamo prostredníctvom STB zakúpiť ľubovoľný tematický balíček na určité obdobie podľa pridelenej tarify (viď požiadavok č. 6). Okrem programových ponúk si zákazník môže vybrať film z video požičovne podľa kategórií definovaných v požiadavke č. 7.

Priorita: 4

Náročnosť: 5

3.2 STB

Požiadavka č. 18 – Základné rozvrhnutie OSD

Pri prepínaní kanálov sa bude zobrazovať číslo aktuálneho kanálu spolu s nasledovnými informáciami:

- názov kanálu
- názov sledovaného programu
- krátky popis programu
- aktuálny čas a dátum, čas do konca programu (vhodné reprezentovať pomocou časovej osy)
- a čas a názov bezprostredne nasledujúceho programu na danom kanály

Tieto informácie sa po krátkom čase skryjú. Ďalej sa vyžaduje, aby tieto informácie bolo možné opäť vyvolať i stlačením vhodného tlačítka na diaľkovom ovládači.

Priorita: 5
Náročnosť: 4

Požiadavka č. 19 – Menu

Zákazník si bude môcť tlačítkom menu vyvolať ponuku, ktorá bude mať nasledujúce možnosti:

- nastavenia STB
- samoobsluha (ak je povolená)
- virtuálna video požičovňa (VoD)
- EPG (elektronický programový sprievodca)

Priorita: 4
Náročnosť: 4

Požiadavka č. 20 – Kontextové menu

Portál bude umožňovať vyvolať kontextové menu pre práve sledovaný kanál (tlačítko sa môže líšiť v závislosti na typu modelu STB) v ktorom sa budú dať prepnúť zvukové stopy, titulky, prípadne pomer obrazu.

Priorita: 5
Náročnosť: 4

Požiadavka č. 21 – Rýchle voľby

Jednotlivé akcie z hore uvedenej požiadavky sa budú dať vyvolať rýchlymi tlačidlami (zvyčajne označené ako modrá, zelená, žltá, červená na DO – v závislosti od modelu)

Priorita: 3
Náročnosť: 3

Požiadavka č. 22 – Virtuálna videopožičovňa

Vo virtuálnej videopožičovni sa zákazníkovi zobrazí zoznam zakúpených filmov a po výbere sa mu zobrazí jeho detailný popis s možným prehratím. Taktiež sa mu bude vhodne zobrazovať zostávajúci čas do vypršania zakúpenej lehoty. Po zakúpení vybraného filmu bude umožnené okamžité prehratie a bude zaradený do zoznamu zakúpených filmov.

Priorita: 3
Náročnosť: 5

Požiadavka č. 23 – Možnosti nastavenia

V submenu nastavení bude zákazníkovi sprístupnené nastavenie obrazu a typ výstupu (HDMI, S-video, v závislosti od modelu STB), taktiež bude vedieť nastaviť predvolenú úroveň hlasitosti. Ďalej možnosť nastavenia predvolenej zvukovej stopy, tituliek a pomer obrazu.

Priorita: 4
Náročnosť: 4

Požiadavka č. 24 – Personalizácia kanálov a rodičovský zámok

Zákazník si bude vedieť spravovať poradie kanálov a umožniť aplikovania rodičovského zámku na požadovaný kanál. Zapnutie/vypnutie rodičovského hesla a taktiež jeho zmena bude umožnená priamo v nastaveniach STB.

Priorita: 4
Náročnosť: 4

Požiadavka č. 25 – Personalizácia kategórií TV kanálov

Zákazník si bude môcť okrem predvolených kategórií TV kanálov nadefinovať vlastnú kategóriu a priradiť do nej kanály, ktoré môžu byť súčasne naraz v niekoľkých kategóriách. Prechádzanie týchto kanálov bude vhodne zasadené do menu prípadne asociované vhodným tlačítkom na DO.

Priorita: 3

Náročnosť: 3

Požiadavka č. 26 – Jazykové mutácie portálu

Požaduje sa aby portál STB bol minimálne dvojjazyčný a síce v anglickom a slovenskom jazyku a jeho zmena bola možná priamo v menu nastaveniach STB.

Priorita: 1

Náročnosť: 2

Požiadavka č. 27 – Elektronický programový sprievodca

Portál bude umožňovať zákazníkovi zobrazit' EPG pre konkrétny kanál, a síce na niekoľko dní dopredu (resp.dozadu – nie je podmienkou) vo vertikálnom prevedení, kde na jednotlivých riadkoch bude uvedený čas a názov programu.

Priorita: 4

Náročnosť: 4

Požiadavka č. 28 – Detaily programu

V náväznosti na predchádzajúcu požiadavku bude možné zobrazit' detailný popis vybraného programu. Túto vlastnosť bude možné aplikovať i na práve sledovaný kanál vhodným tlačítkom na DO resp. zasadením do kontextového menu.

Priorita: 4

Náročnosť: 2

Požiadavka č. 29 – Internetový prehliadač

V závislosti od nastavenia systému bude umožnené na STB spustiť integrovaný internetový prehliadač. Predpokladá sa, že zariadenia nebudú vybavené externou klávesnicou a zadávanie adresy a pohyb po stránkach bude pomocou DO. Preto je návrh virtuálnej klávesnice veľmi dôležitý. V nastaveniach si zákazník bude vedieť editovať záložky a domovskú stránku.

Priorita: 3

Náročnosť: 4

Požiadavka č. 30 – Prehrávanie streamu

Po spustení portálu, STB začne okamžite prehrávať naposledy sledovaný kanál. V prípade že túto skutočnosť nie je možné zistiť, bude sa prehrávať prvý v poradí. Považuje sa za samozrejmé aby pri pohyboch v jednotlivých menu a podmenu, resp. prechádzaním EPG bol stream neprerušovaný a zobrazovaný v pozadí s vhodnou priehľadnosťou.

Priorita: 5

Náročnosť: 3

Požiadavka č. 31 – Portabilita nastavení

Nastavenia ktoré boli modifikované pri personalizácii kanálov budú uložené na aplikačnom servere, aby zostali pre zákazníka dostupné pri prípadnej výmene STB.

Priorita: 3
Náročnosť: 4

3.3 Rozhranie

Požiadavka č. 32 – Univerzálne rozhranie

Od systému sa vyžaduje vytvorenia univerzálneho rozhrania pre správu a kontrolu celej platformy. Toto rozhranie je vstupno/výstupným bodom pre napojenie ostatných CRM systémov.

Priorita: 5
Náročnosť: 5

Požiadavka č. 33 – Integrácia do opeerátorovho IS a billing systému

V rámci projektu sa predpokladá základná implementácia rozhrania z vyššie uvedenej požiadavky. Zadávatel' projektu požaduje integráciu do IS popísaného v [7] ako ďalší samostatný modul.

Priorita: 2
Náročnosť: 4

3.4 Bezpečnosť a komunikácia

Požiadavka č. 34 – Autentizácia STB

Do siete bude pripojené len to zariadenie, ktoré prejde autentizačným mechanizmom. Autentizácia bude prebiehať na základe porovnania MAC adresy STB oproti DB a s možným ďalším overením pomocou zadania PUK kódu pri prvom pripojení STB do siete.

Priorita: 5
Náročnosť: 4

Požiadavka č. 35 – Komunikácia cez šifrovaný kanál

Komunikácia medzi STB a middlwarom bude šifrovaná pomocou SSL a vygenerovaných certifikátov s určitou platnosťou. Tieto certifikáty budú uložené v STB a ich správu si zabezpečí zadávatel' projektu.

Priorita: 5
Náročnosť: 4

Požiadavka č. 36 – PIN nákupu

V súvislosti s požiadavkami č. 17 a č. 22, bude nákup vždy potvrdený PIN kódom, ktorý bude po zákazníkovi vyžadovaný. Po zadaní sa bude považovať nákup za záväzný a bude zákazníkovi riadne účtovaný.

Priorita: 4
Náročnosť: 4

Požiadavka č. 37 – Zabezpečenie portálu proti odcudzeniu

Vytvorený portál nebude jednoduchým postupom možný odcudziť.

Priorita: 5

Náročnosť: 5

3.5 Zdroje

Požiadavka č. 38 – Minimalizácia nákladov na HW

Požaduje sa, aby portál bol navrhnutý, implementovaný a podporovaný na STB v súčasnosti s nízkou cenou na trhu. Taktiež sa predpokladá že serverová časť (middleware) nebude vyžadovať vysoké požiadavky na výpočetné zdroje.

Priorita: 3

Náročnosť: 3

Požiadavka č. 39 – Minimalizácia SW vybavenia

Systém musí byť vytvorený s využitím voľne dostupných technológií a prostriedkov za účelom maximálneho zníženia nákladov na jeho vývoj a údržbu. Požiadavky na využitie konkrétnej technológie nie sú špecifikované.

Priorita: 3

Náročnosť: 3

3.6 Dokumentácia

Požiadavka č. 40 – Špecifikácia rozhrania

Rozhranie IPTV middlewaru v serverovej časti bude zdokumentované pre prípadné budúce nasadenie iných CRM systémov.

Priorita: 3

Náročnosť: 2

Požiadavka č. 41 – Užívateľská príručka pre administrátora systému

Vytvorenie užívateľskej príručky obsahujúcej popis užívateľského grafického rozhrania. Súčasťou dokumentácie budú i príklady použitia systému a vysvetlené princípy práce s novými modulmi.

Priorita: 1

Náročnosť: 1

Požiadavka č. 42 – Užívateľská príručka pre koncového zákazníka

Vytvorenie jednoduchej užívateľskej príručky obsahujúcej popis užívateľského rozhrania STB a prácu s ním.

Priorita: 1

Náročnosť: 1

Požiadavka č. 43 – Programátorská dokumentácia

Za účelom sprehľadnenia a uľahčenia prípadných úprav jadra systému vygenerovať programátorskú dokumentáciu s pomocou dokumentačných nástrojov použitých technológií.

Priorita: 1

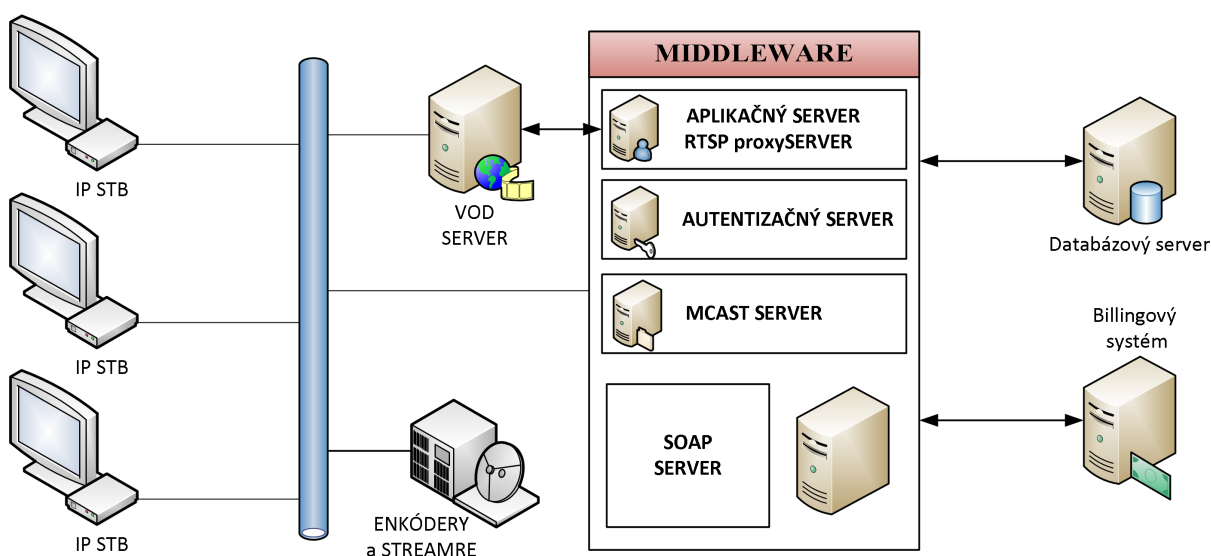
Náročnosť: 1

4 Návrh systému

V tejto kapitole sa oboznámime s návrhom celkového systému, ktorý zahrňuje návrh samotného middlewaru ako klient/server architektúru vid' kap. 2.3. Najskôr bude predstavená idea riešenia spolu so základným vymedzením pojmov, ktoré budeme ďalej v našej práci používať. Na základe analýzy požiadaviek z predchádzajúcej kapitoly je navrhnutý model prípadu použitia zo strany zákazníka.

4.1 Idea riešenia

Na obrázku č. 4 je zobrazený návrh komplexného systému pre zabezpečenie sledovania IPTV prostredníctvom počítačovej siete. Aby si čitateľ utvoril prehľad nad týmto systémom, v nasledujúcich statiach budú stručne charakterizované jednotlivé komponenty.



Obrázok 4: Návrh systému pre sledovanie IPTV

Middleware tvorí serverovú časť systému, a je základom pre celý chod platformy. V našom projekte middleware pozostáva z týchto podkomponent:

- **Autentizačný server** – umožňuje overenie povolených STB
- **Aplikačný server** – spracováva dotazy od STB
- **Multicast server (mcast)** – poskytuje distribúciu nastavení, firmware / portálu pre STB
- **Soap server** – rozhranie pre napojenie CRM / billing systému

Portál je softwarovo vybavenie STB (klientská časť MW), ktoré sa skladá z UI a umožňuje interakciu zákazníka s IPTV platformou. Predovšetkým sa jedná o úkony spojené s prehliadaním a prepínaním TV kanálov, zobrazovaním podrobných informácií o TV programe, prehliadaním a výberom filmu z virtuálnej videopožičovne. Ďalej umožňuje zákazníkovi zmenu osobných nastavení ako je napr. vlastné poradie kanálov a prístup do samoosluhy, prostredníctvom ktorej je možné objednávať a rušiť rozšírenú televíznu ponuku. Portál nie je súčasťou firmwaru STB, ale je to ako balíček, ktorý sa pri každom bootovaní stiahne zo siete. Aby sa splnila požiadavka č. 37, tento balík

je uložený na serveri a distribuovaný v šifrovanej podobe. Počas bootovania STB sa tento portál stiahne, dešifruje a rozbalí pomocou kľúča, ktorý je získaný pri autentizácii.

Autentizačný server tvorí podstatnú podkomponentu systému. Poskytuje služby pre overenie STB zariadení. Každé zariadenie sa môže nachádzať v troch stavoch:

- enabled – zariadenie je aktívne, bez obmedzenia funkcionality
- suspended – zariadenie je aktívne, a síce s obmedzením (napr. pozastavené sledovanie tv kanálov)
- disabled – zariadenie nie je aktívne

Autentizačným mechanizmom prejdú len tie zariadenia, ktoré sú aktívne, t.j. sú v stavoch enabled alebo suspended. Po úspešnej autentizácii STB získa šifrovací kľúč, ktorým dešifruje portál a pripraví prostredie pre jeho spustenie.

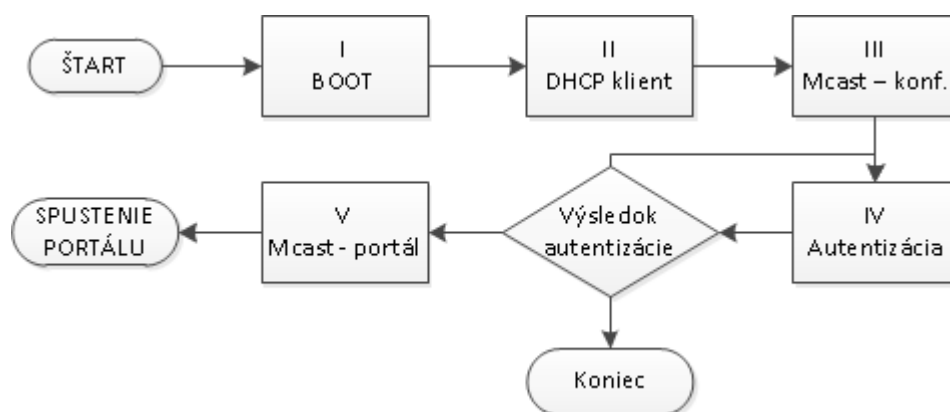
Autentizáciou nemusia prejsť len zariadenia ktoré sú v stave disabled, ale i ostatné a to z niekoľkých dôvodov:

- STB nemá pridelenú IP adresu alebo nie je pripojené k sieti
- výpadok mcast servera resp. nesprávna konfigurácia siete
- nedostupný autentizačný server

Preto tie zariadenia, ktoré neprejdú autentizačným mechanizmom budú automaticky reštartované a cyklus sa znovu zopakuje.

Obrázok č. 4 popisuje sled udalostí, ktoré sú vykonané po zapnutí STB. Ich význam je nasledovný:

- I – boot – zavedenie potrebných ovládačov
- II – pridelenie IP adresy od DHCP servera
- III – pripojenie sa do multicastovej skupiny pre získanie konfiguračného súboru
- IV – autentizácia
- V – príjem portálu a dešifrovanie na základe výsledku autentizácie
- VI – spustenie portálu



Obrázok 5: Proces pri zapnutí STB

Aby sa predišlo zaťaženiu siete a predovšetkým linky ktorá smeruje k serveru, je pre distribúciu globálneho konfiguračného súboru a najmä distribúciu portálu poprípade firmwaru pre STB

zariadenia, kde sa predpokladá objem výsledného balíka rádovo v jednotkách MB, zavedený **multicastový server (mcast server)**. Koncové zariadenia obsahujú mcast klienta s pevne nastavenou multicastovou adresou na ktorej je cyklicky posielaný konfiguračný súbor v otvorenej podobe. Po prijatí tohoto súboru si STB nastaví parametre a pripojí sa do multicastovej skupiny pre príjem portálu, ktorý je taktiež cyklicky streamovaný do siete.

Aplikačný server spracováva správy od STB a riadi ich pomocou buď samostatných správ alebo vložených príkazov v odpovedi na správu. Komunikácia medzi STB a aplikačným serverom bude rozvedená v kapitole 5.4. V nasledujúcej podkapitole 4.4.1. je diskutovaný spôsob pre návrh elektronického programového sprievodcu, ktorý je súčasťou analýzy požiadaviek.

Kontrola nad týmto systémom a jeho ovládanie je zabezpečené prostredníctvom **SOAP serveru**, ktorý umožňuje jeho prepojenie s nezávislými CRM systémami. Toto prepojenie je realizované pomocou webovej služby. Bližší popis a detaily komunikácie je možné nájsť v kapitole 5.6.

4.1.1 Elektronický programový sprievodca

Ďalšou úlohou aplikačného serveru je poskytovanie údajov pre elektronického programového sprievodcu EPG zabudovaného v portáli STB. Tu si môžeme položiť otázku, akým spôsobom budú tieto informácie poskytované a vo výsledku zobrazené na TV. Ponúkajú sa nám tri možnosti zaobchádzania s EPG datmi, ich ukladaním a interpretáciou.

Prvá možnosť, ktorá sa javí ako efektívna je exportovať určitý časový úsek (napr. 3 dni dopredu od aktuálneho dňa) do súboru, ktorý by bol v pamäti STB zariadenia. Tento súbor by sa v pravidelných intervaloch aktualizoval s ohľadom na stanovený časový úsek. Táto varianta predstavuje ale jednak réžiu na strane STB a na strane MW, kde budú globálne uložené tieto EPG dáta. Okrem tejto réžii narastajú aj ďalšie požiadavky na dostupné zdroje STB. Ako hlavná nevýhoda tohto riešenia vyplýva už zo samotného spôsobu implementácie portálu v jazyku JavaScript a jeho interpretácie pomocou web prehliadača. Problém spočíva z bezpečnostných opatrení, ktoré obmedzujú prístup a prácu s lokálnymi súborami z prehliadača. Nie príliš šťastné riešenie by bolo, ak by sa tento súbor s EPG informáciami transformoval na napr. asociatívne pole resp. vhodne navrhnuté pole objektov a pri spustení portálu by sa pracovalo s týmto poľom. Týmto riešením by sme sa taktiež dostali do toho istého stavu, ktorý je vlastne treťou možnosťou (viď ďalší odstavec) ale s väčšími nárokmi na pamäť. Okrem toho by sme si privodli komplikácie s možným nedostatkom voľnej pamäti, nakoľko tieto EPG informácie pre 100 kanálov a spomenutý časový úsek 3 dni zaberajú cca 3 až 4 MB, ktoré by sa museli natiahnuť do pamäti RAM pri tomto type riešenia.

Druhú variantu tvorí návrh jednoduchého http serveru umiestneného v STB zariadení. Táto odľahčená verzia serveru by na základe požiadavky poslanej z portálu vyhľadávala požadované EPG data v súbore a vrátila ich do portálu. Tento spôsob je flexibilnejší čo sa týka samotnej aktualizácie epg dát, ale nároky na pamäť neeliminuje.

Ako posledná tretia možnosť je poskytovať EPG informácie priamo z aplikačného serveru ku klientskému portálu. Pri tomto štýle riešenia sa neexportuje žiadny EPG súbor s časovým intervalom, ale dáta sú vyhľadávané na aplikačnom serveri až v čase, keď príjde požiadavka na ich zobrazenie. Komunikácia prebieha ako v predchádzajúcom prípade, pomocou http protokolu. Negatívna vlastnosť tejto varianty je menšie milisekundové oneskorenie zobrazenia informácií na TV a zvýšenie provozu na sieti pri príliš častých požiadavkách. Tento jav nám ale nemusí vadieť, pretože zavedením lokalít vieme do budúca toto vytťaženie rozložiť na viacero serverov pri náraste klientov.

Takto „ušetrenú“ pamäť môžeme využiť na ďalšie rozšírenia, ktoré su zhrnuté a diskutované na konci tejto práci. Z týchto dôvodov je v tomto projekte EPG implementované tak ako je popísané v tomto odstavci, tj. tretia varianta.

4.1.2 VOD

V tomto diplomovom projekte bude pre demonštráciu virtuálnej video požičovne z požiadavky č. 22 použitý voľne dostupný open source VLC media player. VLC obsahuje komponentu VideoLAN Manager (VLM), ktorá je určená na ovládanie viacerých multimediálnych streamov použitím iba jednej spustenej inštancie VLC. Tento manažér podporuje pre správu streamov http a telnet rozhranie.

VLM podporuje dva typy médií (médiu je zložené zo zoznamu vstupov – audiovizuálne streamy a výstupov – kam a ako bude obsah streamovaný):

- vod – typické Video on Demand – médium bude spustené ak klient oň požiada
- broadcast – podobné live TV kanálom – spustené a ovládanie administrátorom

Čitateľ by predpokladal, že bude využitá práve tá prvá varianta, typ vod, avšak vstavaný RTSP server neumožňuje žiadnu autentizáciu žiadateľov VOD obsahu a ani nemá žiadnym spôsobom získať informácie o tom, či má daný žiadateľ predplatený VOD obsah. Dôsledkom tohto je nevyhnutné pre našu prácu navrhnuť RTSP proxy, ktorý bude komunikovať s VLM a zostavovať session, kde typom média bude broadcast.

Teraz uvedieme zoznam príkazov a popis parametrov pre zostavenie broadcast vysielania, ktoré budú využité v implementácii RTSP proxy servera:

- **new name broadcast**
Vytvorenie nového vysielania typu broadcast, kde *name* je unikátne meno novej session
- **setup name input path_to_file**
Špecifikácia cesty k súboru vysielaného obsahu pre vyššie definovanú session
- **setup name output stream_properties**
Špecifikácia streamovacích parametrov je v tvare `#rtsp {mux=ts,dst=IP_ADDR,port=PORT}` kde *IP_ADDR* je IP adresa STB a *PORT* je číslo portu na ktorom zariadenie očakáva stream.
- **control name play | pause | seek <pos> | stop**
Kontrolné príkazy na začatie prehrávania, pozastavenie, nastavenia na určitú pozíciu v % alebo na úplné zastavenie streamu.
- **del name**
Odstránenie vysielaného média

Po ustanovení a inicializácii prehrávania vyžiadaného obsahu klientom sa vytvorí unicastové spojenie medzi VLC a STB, kde audiovizuálny obsah je transportovaný pomocou RTP protokolu. Klient má možnosť prehrávaný obsah pozastaviť, pretočiť alebo vypnúť. Tieto kontrolné správy spracováva RTSP proxy server, ktorý následne komunikuje s VLM vyššie uvedenými príkazmi.

4.2 Use case model

Pre zobrazenie funkčnej štruktúry portálu z pohľadu zákazníka bol zvolený use case model. Tento model (viď príloha č. 3) definuje správanie portálu a tvorí tak grafické zobrazenie analýzy funkcionálnych požiadaviek kladených na portál. Nasledovné podkapitoly dopĺňujú a upresňujú zobrazené kľúčové prípady použitia. V tomto use case modeli vystupuje iba jeden aktér a to je zákazník.

4.2.1 Správa kanálov

Zákazník si pomocou portálu vie prispôbiť poradie zakúpených kanálov a nastavovať rodičovský zámok na vybraný kanál.

Scenáre prípadov použitia

- **Zmeniť poradie kanála**
 1. zákazník vyvolá menu pre správu kanálov
 2. zobrazí sa zoznam aktuálne zakúpených kanálov s aktuálnym poradím
 3. zákazník si zvolí konkrétny kanál zo zoznamu a iniciuje jeho premiestnenie
 4. zákazník môže menu kedykoľvek opustiť alebo vykonané zmeny uložiť
 5. v prípade uloženia, portál odošle aktuálnu konfiguráciu na server a menu zatvorí

- **Zabezpečiť kanál rodičovským zámkom**
 1. zákazník vyvolá menu pre správu kanálov
 2. zobrazí sa zoznam aktuálne zakúpených kanálov s indikátorom stavu zámku
 3. zákazník si navolí požadovaný kanál a iniciuje zmenu zámku
 4. v prípade, že bol kanál odomknutý sa zamkne
 5. v opačnom prípade portál zobrazí výzvu na zadanie rodičovského kódu
 6. ak bol kód zadaný správne vybraný kanál sa odomkne
 7. zákazník môže menu kedykoľvek opustiť alebo vykonané zmeny uložiť
 8. v prípade uloženia, portál odošle aktuálnu konfiguráciu na server a menu zatvorí

4.2.2 Správa kategórií

Navrhovaný systém obsahuje implicitné kategórie TV kanálov a ich pridelenie k nim, avšak zákazník si vie definovať vlastné kategórie a spravovať ich zoznam.

Scenáre prípadov použitia

- **Vytvoriť novú kategóriu**
 1. zákazník vyvolá menu pre správu kategórií
 2. zobrazí sa zoznam zakúpených kanálov
 3. zákazník iniciuje vytvorenie novej kategórie
 4. portál vyzve zákazníka na vloženie názvu kategórie
 5. po potvrdení je nová kategória vytvorená

- **Pridať / odstrániť kanál z kategórie**

1. zákazník vyvolá menu pre správu kategórií
2. zobrazí sa zoznam zakúpených kanálov
3. zákazník si navolí požadovanú kategóriu ktorú chce modifikovať
4. zákazník si vyberá zo zoznamu dostupných kanálov a iniciuje pridanie/odobratie do/zo zoznamu navolenej kategórie
5. ak sa daný kanál v kategórii ešte nenachádza tak sa pridá, v opačnom prípade sa odoberie
6. zákazník môže menu kedykoľvek opustiť alebo vykonané zmeny uložiť
7. v prípade uloženia, portál odošle aktuálnu konfiguráciu na server a menu zatvorí

4.2.3 Rodičovský kód

Pri odblokovaní kanálu chráneného rodičovským zámkom je vyžadované od zákazníka vloženie kódu. Tento kód je implicitne nastavený na predom definovanú hodnotu, avšak zákazník si v priebehu používania služby tento kód vie kedykoľvek zmeniť

Scenár prípadu použitia

- **Zmeniť rodičovský kód**
 1. zákazník vyvolá menu pre nastavenia rodičovského zámku
 2. zákazník iniciuje zmenu kódu
 3. portál vyzve zákazníka na vloženie súčasného kódu, nového kódu a jeho potvrdenie
 4. zákazník zadá vyžadované parametre
 5. ak vstupné parametre neboli zadané správne, zobrazí sa varovná informácia
 6. v opačnom prípade je kód zmenený

4.2.4 Samoobsluha

Na základe požiadavky číslo 17 bolo vytvorené rozhranie a menu pre nákup televíznych kanálov priamo cez TV obrazovku. Zákazník si vie predplatiť televíznu ponuku podľa aktuálnej tarify, poprípade zrušiť predplatné.

Scenáre prípadov použitia

- **Zakúpiť televíznu ponuku**
 1. zákazník vyvolá menu samoobsluhy
 2. zobrazí sa zoznam dostupných ponúk s cenou, minimálnou dĺžkou predplatného a zoznamom kanálov v ponuke
 3. zákazník si vyberie ponuku zo zoznamu a iniciuje zakúpenie
 4. portál vyžiada od zákazníka PIN kód nákupu
 5. zákazník vloží PIN, ak bol správny novo zakúpené kanály sú k dispozícii
 6. v opačnom prípade sa zobrazí varovná hláška s možnosťou opakovania akcie

- **Odhlásiť televíznu ponuku**
 1. zákazník vyvolá menu samoobsluhy

2. zobrazí sa zoznam aktívne predplatených ponúk s dátumom zakúpenia
3. zákazník si navolí ponuku zo zoznamu a iniciuje odhlásenie ponuky
4. v prípade ak sa odhlasuje ponuka, ktorá od doby zakúpenia ešte nepresiahla minimálnu dĺžku predplateného, zákazník bude mať počas tejto lehoty kanály dostupné
5. v opačnom prípade sa kanály obsiahnuté v odhlasovanej ponuke okamžite odstránia z dostupných kanálov

4.2.5 Videopožičovňa

Umožňuje zákazníkovi si prehliadať dostupné videá na VoD serveri s možnosťou zakúpenia na určitý čas definovaný v systéme.

Scenáre prípadov použitia

- **Zakúpiť video**

1. zákazník vyvolá menu virtuálnej video požičovne
2. zobrazí sa zoznam dostupných videí s krátkym popisom a cenovou kategóriou
3. zákazník si navolí video zo zoznamu a iniciuje zakúpenie
4. portál vyžiada od zákazníka PIN kód nákupu
5. zákazník vloží PIN, ak bol správny, kupované video sa objaví medzi zakúpenými s možnosťou okamžitého prehratia
6. v opačnom prípade sa zobrazí varovná hláška s možnosťou opakovania akcie

- **Prehrať video**

1. zákazník vyvolá menu virtuálnej video požičovne
2. zobrazí sa zoznam zakúpených videí a čas dokedy je možné zakúpené video vzhliadnuť
3. zákazník si vyberie zo zoznamu video a iniciuje jeho prehratie
4. video sa začne okamžite prehrávať s možnosťami pozastavenia, pretáčania a zastavenia prehrávania
5. ak sa skončí prehrávanie, automaticky sa prepne na naposledy sledovaný TV kanál

4.2.6 Zobrazenie EPG

Elektronický programový sprievodca tvorí podstatnú časť užívateľského portálu.

Scenár prípadu použitia

- **Zobraziť detaily programu**

1. zákazník iniciuje zobrazenie epg
2. portál umožní zákazníkovi zvoliť a zobraziť detailné informácie aktuálne sledovaného programu
3. alebo zobraziť informácie o programoch vysielaných aj na ostatných televíznych kanáloch a to i na niekoľko dní dopredu

5 Realizácia kľúčových častí

5.1 Portál

Funkcionalita portálu závisí predovšetkým od možností vybraného HW teda IP STB. Pre našu prácu ako už bolo spomenuté sme zvolili zariadenia, ktoré sú uvedené spolu so špecifikáciou v prílohe č.2.

V ďalšom výklade budú jednotlivé postupy ilustrované pomocou JS API rozhrania modelu MAG200, ktoré nepodlieha NDA².

5.1.1 Užívateľské rozhranie

V našom projekte bolo vytvorené užívateľské rozhranie portálu v súlade s kladenými požiadavkami na vyvíjaný systém. Toto rozhranie tvorí grafickú a funkcionálnu stránku portálu v STB. Ukážku základného OSD je možné nájsť v prílohe č. 5 a č. 6.

Po naboťovaní STB je spustená aplikácia - proprietárny webový prehliadač s prehrávačom. Ako argument tejto aplikácie je HTML dokument. V tomto dokumente je obsiahnutá všetka funkcionalita vrátane grafickej stránky UI portálu. Portál ako celok je zložený ako klasická web stránka, kde jednotlivé prvky sa vytvárajú dynamicky na základe požiadavky na ich zobrazenie (napr. vyvolanie menu). Manipulácia s týmto dokumentom je na bázi objektovo orientovaného prístupu (DOM – Document object model). DOM je API umožňujúce prístup či modifikáciu obsahu, štruktúry alebo štýlu dokumentu, či jeho častiam. DOM reprezentuje HTML dokument ako stromovú dátovú štruktúru. Táto štruktúra je modifikovaná pomocou sady skriptov napísaných v jazyku JavaScript interpretovaných po načítaní dokumentu. Pre uľahčenie práce s DOM modelom sme využili knižnicu jQuery[10]. Taktiež je táto knižnica využívaná pre komunikáciu s aplikačným serverom. Komunikácia prebieha na princípe požiadavka/odpoveď a je podrobne popísaná v kapitole 5.4.

Okrem tohto prehliadača je na pozadí inicializovaný audio/video prehrávač. Ovládanie tohto prehrávača je zabezpečené pomocou API rozhrania v javascripte. Predovšetkým sa jedná o funkcie spojené s:

- prehrávaním audia/videoa (play, stop, pause a pod.)
- úroveň hlasitosti
- nastavenie obrazu (jas, kontrast, a pod.)
- výber audio/video stopy

V kapitole 5.1.3 bude bližšie popísaná metóda na prehrávanie obsahu. Ak má čitateľ záujem o popis ostatných metód môže tak nahliadnuť do [18].

2 http://en.wikipedia.org/wiki/Non-disclosure_agreement

API STB definuje dve okná:

- grafické okno – zobrazuje obsah html dokumentu
- video okno – okno v ktorom je prehrávaný multimedialny obsah

Implicitne je grafické okno na popredí a na výstupe je zobrazovaná html stránka. V prípade požiadavky na prehratie videa, je tak toto video na pozadí a nie je ho vidieť na video výstupe. Túto situáciu je nutné riešiť pomocou priehľadnosti a je rozobraná v nasledujúcej stati.

Priehľadnosť

Zobrazenie video okna a priehľadnosť pre menu s vykreslením OSD v režime, že sledovaný kanál beží na pozadí (viď príloha č. 6) je zabezpečené pomocou troch metód API rozhrania. Predpokladajme, že farba pozadia html stránky je #030201 (RGB zápis pre takmer čiernu farbu) a prvky portálu (jednotlivé menu) odlišnou farbou. Potom nasledovné metódy vytvoria požadované chovanie:

- *SetChromaKey("0x030201", "0xFFFFFFFF");*
Prvý parameter určuje kľúč farby, popredie. Ak je pixel na okne portálu v tejto farbe je po vymaskovaní považovaný za priehľadný. S maskou 0xFFFFFFFF dosiahneme úplnú priehľadnosť tejto farby.
- *SetWinAlphaLevel(0,200); SetWinAlphaLevel(1,255);*
Táto metóda umožňuje nastaviť priehľadnosť na globálnej úrovni pre celé obrazovky. Prvým parametrom je určené pre ktoré okno sa nastavuje priehľadnosť. 0 – grafické okno (portál) 1 – video okno (výstup prehrávača) . Druhý parameter je samotná hodnota priehľadnosti, ktorá môže byť v rozmedzí 0 až 255, pričom 0 značí kompletnú priehľadnosť a 255 absolútnu nepriehľadnosť.
- *SetMode(1);*
Aktivuje nastavenia určené metódou SetChromaKey pre video okno. Týmto krokom je celé pozadie portálu priehľadné a sú vidno len komponenty UI pri ich zobrazení.

5.1.2 Prehrávanie obsahu

Prehrávanie obsahu zabezpečuje metóda *Play(string playStr)*, kde argumentom je reťazec v tvare `<solution> <URL> [atrack:<anum>] [vtrack:<vnum>]`

Význam jednotlivých parametrov je nasledovný:

- **solution**: typ audiovizuálneho obsahu a/alebo metóda vysielania
 - rtsp – prehrá obsah z RTSP servera
 - rtp – prehrá obsah vo formáte MPEG2TS
 - ffmpeg – umožňuje prehrať súbory vo formátoch: avi, mkv, mpg, mp4, mov, wmv, mp3
 - auto – automatická detekcia typu, kontajneru, a kódeku média
- **url**: špecifikuje zdroj, ktorý sa má prehrať a môže byť v týchto formátoch
 - /path – cesta k súboru na lokálnom súborovom systéme
 - rtp://addr:port – multicastový rtp stream špecifikovaný adresou a portom
 - rtsp://addr:port/path – obsah uložený na VOD serveri
 - udp://addr:port – multicastový udp stream špecifikovaný adresou a portom

- **atrack** a **vtrack** sú nepovinné voľby, ktoré umožňujú zvoliť audio/video stopu (hodnota PID v prípade MPEG2TS)

Po zavolaní tejto metódy sa automaticky začne prehrávanie obsahu. Pri tejto akcii je vyvolaná udalosť, ktorú je možné programovo odchytiť. Vďaka podpore udalostí v javascripte a API STB je možné asynchrónne spracovávať správy zo vstavaného prehrávača. Udalosti zachytáva tzv. callback funkcia, ktorá je vždy zavolaná pri výskyte udalosti. Udalosť je identifikovaná hodnotou argumentu, ktorá je predaná tejto funkcii. Pre zaregistrovanie callback funkcie je potrebné vytvoriť objekt *stbEvent* na globálnej úrovni a priradiť mu nami definovanú callback funkciu *EventCallback*. Tento postup zobrazuje nasledovný kód:

```
var stbEvent =
{
  onEvent : function(data){},
  event : 0
}

stbEvent.onEvent = EventCallback;

function EventCallback(event)
{
  if(event==5)
  {
    // obsluha výpadku streamu
  }
}
```

V súčasnej verzii API 1.17 [18] pre MAG 200 je definovaných niekoľko udalostí z ktorých sú využité len tri (viď tabuľka č.1). Udalosť č. 1 je využitá pri ukončení prehrávania VOD. Pri výskyte tejto udalosti portál automaticky prepne na živé televízne vysielanie, ktoré bolo sledované pred začiatkom prehrávania VOD. Udalosť s kódom 2 je interne použitá pre nastavenie globálnej premennej, ktorá nesie informáciu či je prehrávač aktívny. Od tejto hodnoty sa potom odvíjajú aj iné funkcie. Poslednú udalosť a síce pre náš projekt veľmi kľúčovú je udalosť s hodnotou 5. Pri príchode tejto udalosti callback funkcia zabezpečí prepnutie na alternatívny zdroj(e) v prípade že je tento zdroj definovaný. Ak je definovaných viacero zdrojov, cyklicky sa prechádzajú a vyberie sa z nich prvý funkčný. V situácii ak každý jeden zdroj generuje udalosť 5 t.j. STB sa nevie pripojiť na multicastovú skupinu resp. RTSP server, zákazníkovi je zobrazená informácia o nedostupnosti kanála resp. videa.

Číslo udalosti	Popis udalosti
1	Prehrávač došiel nakoniec multimedialneho obsahu, alebo detekoval nesúvislosť toku dát.
4	Prehrávanie obsahu bolo inicializované. Udalosť generovaná vždy pri zavolaní metódy Play()
5	Chyba pri otváraní obsahu. Audio/video nebolo nájdené na serveri alebo spojenie so serverom bolo odmietnuté. Táto udalosť je vo skutočnosti generovaná aj pri výpadku už nadviazaného spojenia.

Tabuľka 1: Zoznam použitých udalostí IP STB MAG 200

5.1.3 Ovládanie

Interakcia s portálom je zabezpečená štandardným spôsobom pomocou diaľkového ovládača. Samotná detekcia stlačených tlačítok na DO je realizovaná na úrovni operačného systému zavedením podporných modulov. Toto je väčšinou vyriešené už s dodávaným firmwarom. Pre informáciu môžeme uviesť, že v prípade modelu MAG 200 je obsluha DO riešená pomocou linuxového balíka lirc. Tento balík obsahuje sadu démonov, ktorý zachytávajú IR signály z DO a dekodujú ich. Okrem démonov balíček obsahuje užívateľské programy, ktoré reagujú na dekodované signály. Jedným z nich je aj emulácia klávesnice, pomocou ktorého je zabezpečená interakcia s portálom.

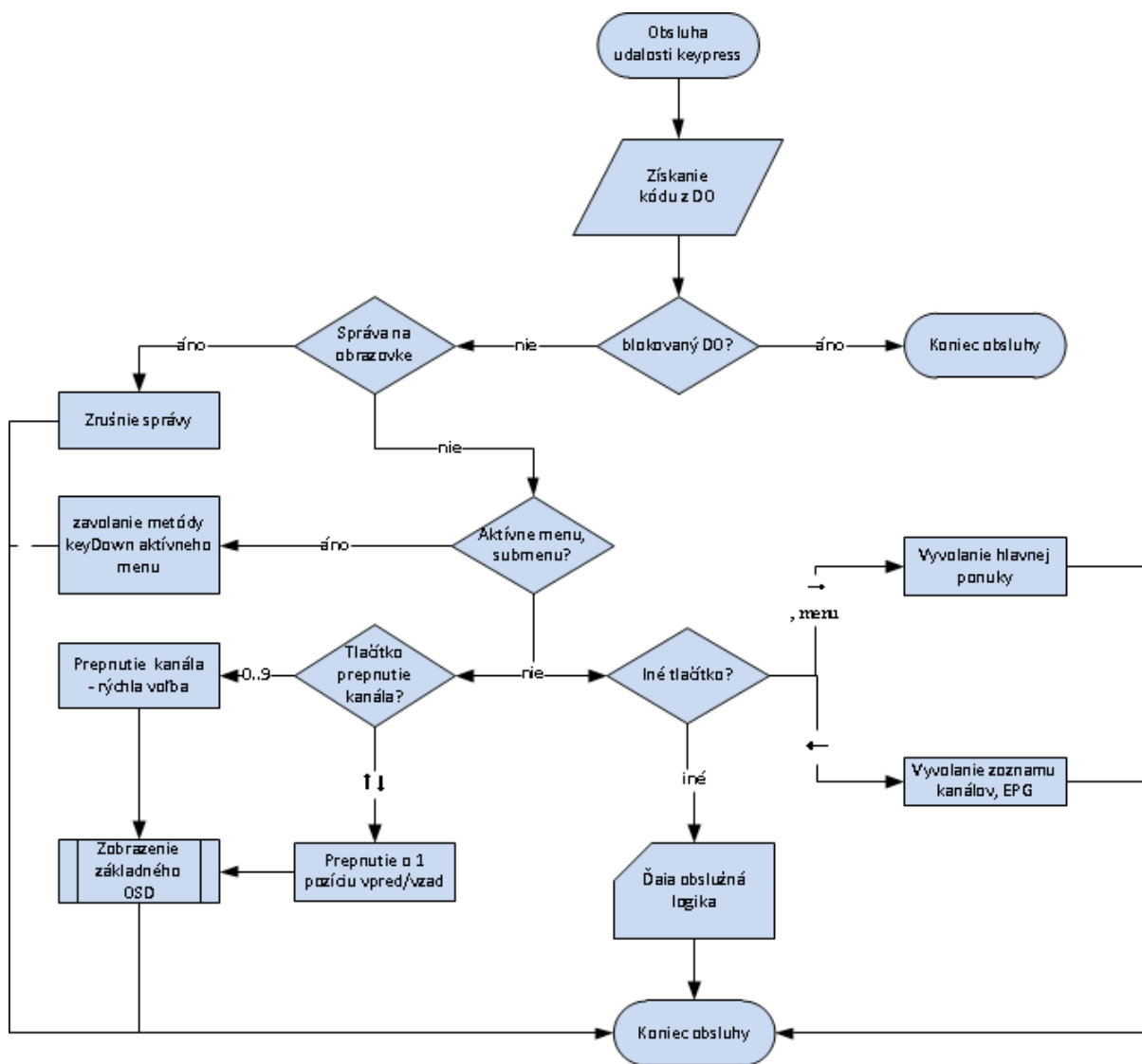
V html stránke je na koreň dokumentu aplikovaná JS metóda `addEventListener`, ktorá zaregistruje callback funkciu `keyDown()`, ktorá obsluhuje udalosť `keypress`.

```
document.addEventListener("keypress", keyDown, false);
```

Vo funkcii `keyDown` je implementovaný stavový automat a tvorí tak celú logiku ovládania portálu. Priebeh tohto automatu je znázornený na obrázku č. 6. Ak sa portál nachádza v stave, keď je aktivovaný nejaký vizuálny objekt napr. menu, je zavolaná metóda `onKeyDown` tohto objektu a riadenie je tak v kompetencii príslušného objektu (napr. pohyb v menu). Funkcie ako je regulácia hlasitosti, zapnutie/vypnutie STB, atď, je na na diagrame zobrazené ako „ďalšia obslužná logika“.

Objekty ktoré majú vlastnú logiku riadenia sú:

- hlavná ponuka
 - nastavenia
 - virtuálna videopožičovňa
 - samoobsluha
- zoznamu kanálov a pohyb v EPG
- pohyb v kontextovom menu pre výber audio stopy
- virtuálna klávesnica



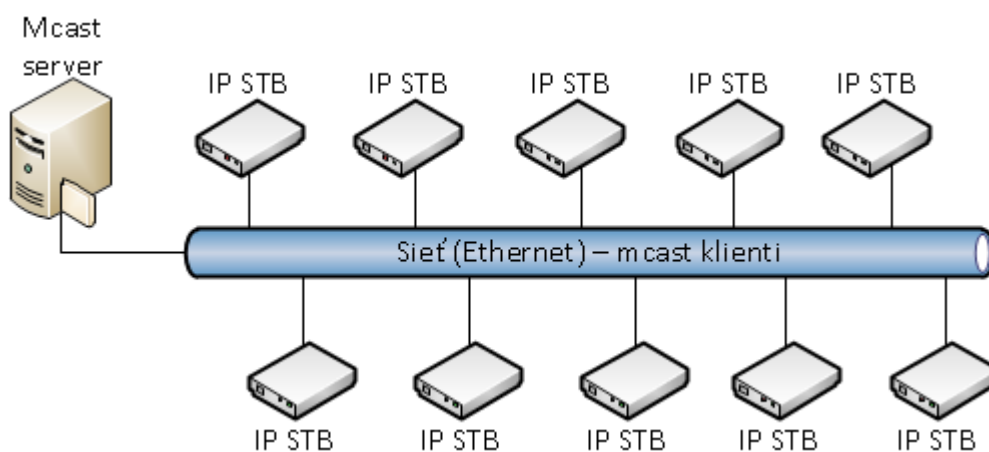
Obrázok 6: Tok obsluhy udalosti keypress

5.2 Multicast (mcast) server

V kapitole 4.1. sme naznačili spôsob distribúcie konfiguračných súborov a samotného portálu resp. firmwaru pre STB. Pre zníženie záťaže linky, ktorá smeruje k middlewaru, je implementácia a nasadenie distribúcie súborov pomocou multicastu nevyhnutná. Táto požiadavka je priamo úmerná s rastúcim počtom zákazníkov.

Súbor resp. súbory zabalené do archívu, server cyklicky posiela do siete pomocou multicastu s využitím transportného protokolu UDP. Sieťová topológia a aktívne prvky musia podporovať IGMP snooping. Zariadenie, ktoré chce prijať distribuovaný súbor, odošle požiadavku IGMP join na predom špecifikovanú multicastovú skupinu. Aby klienti nemuseli čakať až na začiatok nového cyklu, sa pri každom odosielanom pakete vkladá informácia o počte už odoslaných bajtov (offset) distribuovaného súboru. Táto logika umožňuje klientovi prijímať súbor okamžite hneď ako sa pripojí do skupiny. Výsledný súbor poskladá na základe informačnej správy, ktorá obsahuje základné údaje o prenášanom súbore vrátane jeho kontrolného súčtu. Tieto informačné správy sa posielajú periodicky každých päť sekúnd v spoločnej multicastovej skupine ako je distribuovaný súbor.

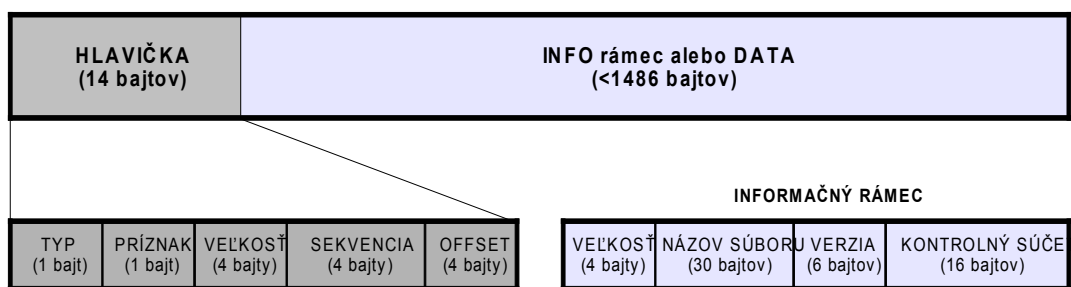
Po prijatí súboru sa vypočíta kontrolný súčet a porovná sa so súčtom získaným z informačnej správy. Ak sa oba súčty nezhodujú, klient sa pokúsi opakovane získať súbor zo siete. V opačnom prípade je archív rozbalený. Archív môže obsahovať skript, ktorý sa po rozbalení spustí a vykoná update súborov resp. iné príkazy obsiahnuté v skripte.



Obrázok 7: Distribúcia súborov cez multicast

5.2.1 Štruktúra mcast správy

Na obrázku č.8 je zobrazené zloženie dátovej časti multicastového UDP datagramu. Skladá sa z hlavičky po ktorej nasleduje informačný rámec alebo obsah distribuovaných dát. Aby nedochádzalo k fragmentácii na nižších vrstvách TCP/IP protokolu je maximálna veľkosť tejto správy limitovaná na 1500 bajtov. Táto veľkosť je odvodená z ethernet IP datagramu [2]. Dôvodom takého to návrhu je aby sa jednoduchým spôsobom dalo softwareovo zabezpečiť riadenie šírky pásma multicastového toku na úrovni samotného mcast servera, vid' nasledujúcu podkapitolu.



Obrázok 8: Dátová časť UDP datagramu

Hlavičku, ktorá je vkladaná do každého paketu, tvoria nasledovné polia:

- **typ** – nesie informácie o type mcast správy
hodnoty: 1 – informačná správa; 2 - data
- **príznak** – šifrovanie obsahu správy
hodnoty: 0 – správa je v otvorenej podobe; 1 – správa je šifrovaná symetrickým kľúčom
- **veľkosť** – aktuálne prenášaná veľkosť správy spolu s hlavičkou
- **sekvencia** – za každým kompletne odoslaným súborom sa toto číslo inkrementuje o jedna
- **offset** – pozícia prenášaných dát od začiatku súboru

Informačná správa je posielaná v pravidelných intervaloch a obsahuje tieto položky:

- **veľkosť** – veľkosť distribuovaného súboru v bajtoch
- **názov súboru** – maximálna dĺžka názvu je 30 znakov
- **verzia** – verzia portálu / firmwaru
- **kontrolný súčet** – MD5 obsahu súboru

5.2.2 Šírka pásma multicastového streamu

Ak by pásmo nebolo riadené, distribuovaný súbor by sa cyklicky odosielať v závislosti od priepustnosti TCP/IP stacku operačného systému. To by vyústilo takmer vo všetkých prípadoch obsadení a zahltení celej kapacity prenosového média čo je samozrejme neprípustné. Preto je nevyhnutné zaviesť časové medzery medzi posielaťmi mcst správami. Pre riadenie požadovanej šírky pásma sa používa funkcia *sleep* na dočasné pozastavenie vykonávania kódu na dobu špecifikovanú argumentom funkcie. Čas je udávaný v mikrosekundách a je vypočítaný podľa nasledujúceho vzorca:

$$\text{sleep} = 1/(\text{speed}/(\text{MTU}-\text{headerSize})) * 1000000$$

kde *speed* je požadovaná šírka pásma v B/s

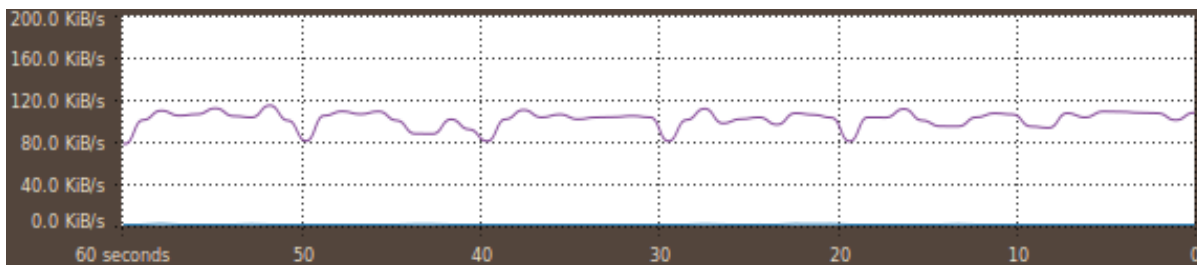
MTU je maximálna veľkosť prenášanej správy tj 1500 bajtov

a *headerSize* je z pravidla 14 bajtov

Príklad: Pre konštantnú šírku 100 kB/s potrebujeme pozastaviť posielať na 14511 mikrosekúnd.

Meranie a overenie výsledku

Jednoduchým meraním pomocou linuxovej utility System Monitor sme overili odchádzajúci provoz zo sieťového rozhrania na ktorý sa cyklicky posielať portál pomocou mcst serveru. Na obr. č. 9 je výstupný graf merania pre nastavený tok 100 kB/s. Priemerná hodnota merania sa pohybovala okolo 105 kB/s.



Obrázok 9: Šírka pásma multicastového prúdu (100 kB/s)

5.2.3 Šifrovanie

Implementáciu požiadavky č. 37 (zaistenie bezpečnosti portálu/firmwaru proti odcuzdeniu) pri distribúcii po sieti je možné realizovať niekoľkými spôsobmi. Distribuovaný súbor bude už v čase rozosielať šifrovaný alebo posielať súbor sa bude šifrovať on-the-fly pri posielať jednotlivých častí. V oboch prípadoch ak je distribuovaný súbor šifrovaný, je nastavený príznak v hlavičke mcst správy. Na základe tohto príznaku sa mcst klient bude vedieť zariadiť a prijatý súbor sa snažiť dešifrovať. Pre šifrovanie sa využíva symetrický šifrovací algoritmus AES so 128 bitovým kľúčom s použitím openssl knižnice [11]. Tento symetrický kľúč získa STB pri autentizácii, ktorá sa vykoná pred samotným príjmom distribuovaného súbora.

5.3 Autentizačný server

Úlohou autentizačného serveru je overenie identity STB zariadení a ustanovenia bezpečného spojenia medzi koncovým zariadením a serverom, spôsobom aký bol uvedený v kap. 4.1. Autentizačný server ako časť middlewaru ustanovuje šifrované spojenie na základe obojstrannej autentizácie cez SSL.

SSL (Secure Socket Layer) je vrstva, ktorá je vložená medzi aplikačnú a transportnú vrstvu. SSL spojenie je založené na princípe asymetrickej šifry, ktorá sa skladá z dvojice navzájom previazaných kľúčov – verejný a súkromný. Každá z komunikujúcich strán šifruje správy verejným kľúčom príjemcu. Takto je zaručené, že správy bude vedieť dešifrovať len majiteľ privátneho kľúča.

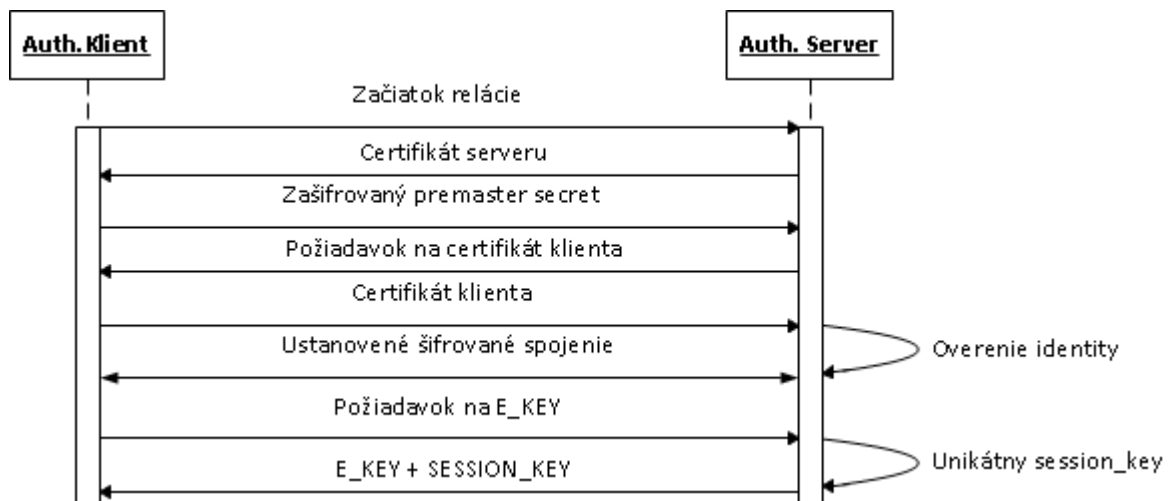
Pre ustanovenie bezpečného kanála cez nezabezpečenú sieť by postačila aj jednosmerná SSL autentizácia. Obojstrannú autentizáciu využijeme pre jednoznačné overenie identity STB, ktorým nahradíme klasické autentizačné metódy ako je napríklad meno a heslo.

5.3.1 Komunikácia

Komunikácia medzi STB (auth.klient) a middlewarom (auth.server) prebieha podľa obr. č. 10 a zobrazuje nasledovné:

1. Klient iniciuje SSL spojenie poslaním požiadavky na ustanovenie šifrovaného kanálu.
2. Server pošle klientovi odpoveď na jeho požiadavku, ktorá obsahuje certifikát serveru. Klient má k dispozícii certifikát CA (certifikačnej autority) podľa ktorého overí prijatý certifikát serveru a jeho samotnú identitu. Prijatý certifikát taktiež obsahuje verejný kľúč servera.
3. Na základe doposiaľ obdržaných informácií vygeneruje klient základ šifrovacieho kľúča (premaster secret), ktorým sa bude šifrovať následná komunikácia. Ten zašifruje verejným kľúčom serveru a pošle mu ho. Server použije svoj súkromný kľúč k rozšifrovaniu základu šifrovacieho kľúča. Z tohto základu vygenerujú ako aj server, tak aj klient hlavný šifrovací kľúč.
4. Pre overenie identity STB, si server od klienta vyžiada klientský certifikát podpísaný tou istou CA. Každé zariadenie vlastní unikátny certifikát, kde v zložke CN (Common name) je obsiahnutá jeho MAC adresa.
5. Klient zašle certifikát. Na základe tohoto certifikátu server overí správnosť klientského certifikátu tj. jeho platnosť, vydavateľa certifikátu atď. V tomto kroku server identifikuje zariadenie pre ktoré bol tento certifikát vydaný podľa CN. Ak sa dané zariadenie nachádza v databáze aktívnych zariadení, server dokončí autentizačnú procedúru. V opačnom prípade je ustanovené spojenie automaticky zrušené zo strany servera.
6. Po úspešnom ustanovení šifrovaného spojenia, klient zašle požiadavku na získanie šifrovacieho kľúča (E_KEY).
7. Server odošle E_KEY a SESSION_KEY klientovi

Význam týchto kľúčov z kroku 6. a 7. je rozvedený v nasledujúcej podkapitole.



Obrázok 10: Autentizácia klienta

5.3.2 Šifrovacie kľúče

E_KEY

E_KEY je symetrický 128 bitový šifrovací kľúč, ktorým je zašifrovaný portál resp. súbory, ktoré sú distribuované cez mcst server.

SESSION_KEY

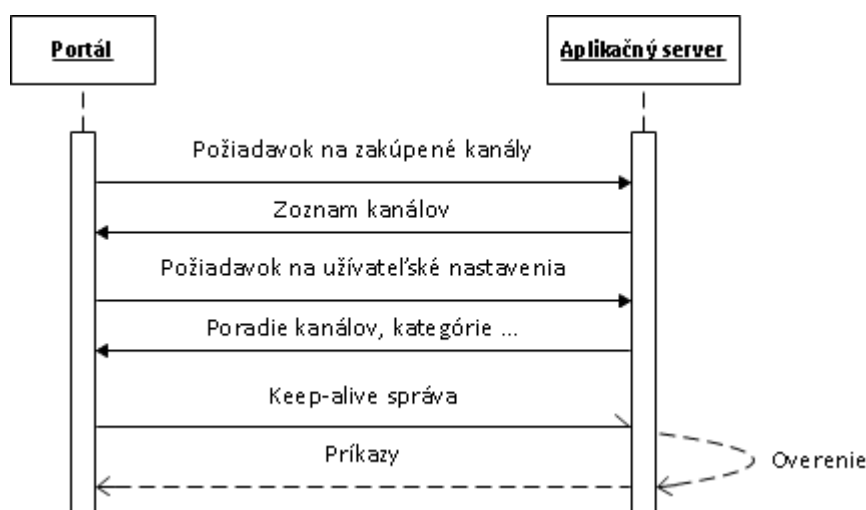
Pretože ďalšia komunikácia STB s MW môže prebiehať i cez nezabezpečený kanál, v závislosti od použitého koncového zariadenia (viď kap. 5.4.), autentizačný server vygeneruje unikátny relačný kľúč pre túto komunikáciu. SESSION_KEY je 128 bitový identifikátor pridelený STB a je platný do doby nasledujúcej autentizácie. MW spáruje autentizované zariadenie s týmto vygenerovaným kľúčom spolu s IP adresou. V skutočnosti sa pre identifikáciu používa iba prvých 32 bitov a ďalej v texte je identifikátor označovaný ako SESSION_ID. Generovanie SESSION_KEY zabezpečuje funkcia *UUID (Universally unique identifier)*³, ktorá vráti 32 hexadecimálnych čísiel. Systém generovania tohto čísla sa môže v rôznych implementáciach líšiť, a síce väčšinou je odvodené z časového razítka, pseudonáhodného čísla a použitého hardwaru. V našom projekte je využitá vstavaná funkcia DB servera, ktorá tieto unikátne kľúče generuje a ukladá priamo do tabuľky pre autentizované zariadenia.

3 http://en.wikipedia.org/wiki/Universally_unique_identifier

5.4 Aplikačný server

Aplikačný server, ako ďalšia podkomponenta MW, spracováva požiadavky od STB. Komunikácia medzi STB (portálom) a MW (aplikačným serverom) prebieha zasielaním správ cez HTTP/HTTPS protokol. Pretože vstavaný prehliadač a poskytnuté API v STB zariadení nevedia overiť certifikát aplikačného servera, je i napriek bezpečnej komunikácii cez HTTPS protokol nevyhnutné šifrovať správy na aplikačnej úrovni vzhľadom k tomu, že sa pri tejto komunikácii prenášajú citlivé údaje a riadiace správy, ktoré povolujú jednotlivé služby pre zákazníka. Tieto správy sú šifrované pomocou symetrického algoritmu AES s použitím vygenerovaného SESSION_KEY.

Obrázok č. 11 ilustruje priebeh komunikácie STB s MW po inicializácii portálu. Detaily týchto správ ako ich štruktúra je rozobraná v nasledujúcich podkapitolách.



Obrázok 11: Komunikácia medzi STB (portálom) a MW (aplikačným serverom) po spustení portálu

5.4.1 Štruktúra správy

Pre správy bol zvolený textový formát JSON (JavaScript Object Notation). Tento formát sa vyznačuje svojou univerzálnosťou a nezávislosťou na programovacích jazykoch, preto je vhodný na výmenu dát medzi aplikáciami. JSON je založený na podmnožine programovacieho jazyka JavaScript a dvoch univerzálnych štruktúrach, kolekcia párov názov-hodnota (objekt) a usporiadaný zoznam.

Každá správa ktorá je odosielaná na MW je zložená z pevnej hlavičky a variabilného obsahu správy podľa typu požiadavky. Hlavička obsahuje kód správy (c) a SESSION_ID (s) podľa ktorého je identifikované zariadenie, ktoré správu odoslalo. Hlavička spolu s obsahom správy môže vyzeráť nasledovne :

```
{ h:{c:1, s:"df9d7c42"}, m:{currPIN:0000, newPIN:1234} }
```

Táto správa predstavuje JSON objekt, kde jeho prvok *h* označuje hlavičku a *m* obsah správy. Konkrétne táto správa je odosielaná pri požiadavke na zmenu PIN kódu nákupu v samoobsluže.

Odpoveď na správu

Odpovedná správa je úplne variabilná a jej obsah môže byť i celkom prázdny (prázdny objekt {}). Teraz si definujeme 3 typy odpovedných správ využívané v komunikácii s aplikačným serverom, ktoré majú pseudo štruktúru:

- typ 1 – informačná správa informatívneho charakteru
- typ 2 – informačná správa výstražného charakteru
- typ 3 – riadiaca správa, ktorá obsahuje príkazy pre portál

Prvé dva typy spôsobia zobrazenie oznamu na TV obrazovke zákazníka. Tretí typ je rezervovaný na riadiace príkazy. V súčasnosti je podporovaný jeden príkaz a to vynútenie prepnutie na multicastovú skupinu určenú parametrom správy.

Štruktúra odpovednej informačnej správy je nasledovná:

```
{ type: type, voidable: voidable, title: title, text: text }
```

kde *type* je typ informačnej správy 1 alebo 2, *voidable* je atribút podľa ktorého je zákazníkovi umožnené oznam zrušiť.

V prípade, že je atribút nastavený na hodnotu 1, zákazník vie oznam zrušiť pomocou tlačítka na DO. V opačnom prípade tj. ak je atribút nastavený na 0 je zákazníkovi zablokovaná interakcia s portálom a oznam nejde zrušiť. Táto funkcia nájde uplatnenie v situácii, ak chce operátor obmedziť na určitý čas resp. pozastaviť služby z dôvodu neuhradenia poplatkov. Zrušenie tohto obmedzenia je možné zaslaním ďalšej správy s hodnotou atribútu *voidable* 1. Posledné dva parametre správy sú sebvysvetľujúce a ich hodnotou sú textové reťazce.

Štruktúra riadiacej správy:

```
{ type: 3, voidable: voidable, cmd: {type: type,url: url} }
```

kde atribút *voidable* má analogický význam ako predchádzajúce typy správ. Parameter *cmd* definuje objekt príkazu ktorý sa má vykonať. Ako sme uviedli, momentálne je podporovaný jeden typ a to je *type=1* pre vynútenie prepnutie multicastovej skupiny na adresu špecifikovanú atribútom *url*.

Zvyšné odpovedné správy majú variabilný počet a názvy atribútov v závislosti od typu požiadavky a sú predstavené v nasledujúcej podkapitole.

5.4.2 Správy

V tabuľke č. 2 sú uvedené implementované správy. Každá z týchto správ bude detailne popísaná s príkladom komunikácie, z ktorej bude zrejmé ich zloženie.

Kód správy (c)	Názov správy	Kód správy (c)	Názov správy
1	KeepAlive	200	ChangePIN
100	GetUserChannels	201	GetAvaibleOffers
101	SetUserChannels	202	BuyOffer
110	GetUserCategories	203	GetBoughtOffers
111	SetUserCategories	204	CancelOffer
301	GetAvaibleVOD		
302	BuyVOD		
303	GetBoughtVOD		

Tabuľka 2: Prehľad správ posielaných portálom

KeepAlive (1)

Keep-alive správy sa používajú na testovanie dostupnosti aplikačného serveru. V prípade jeho výpadku by portál dočasne obmedzil funkcie ako je napríklad samoobsluha či virtuálna video požičovňa. Avšak hlavnou úlohou týchto správ je testovanie stavu predplatených služieb zákazníka, posielanie správ-oznamov na TV obrazovku a popríklad riadenie STB pomocou kontrolných správ. Mimo toho, sa tieto správy využívajú aj na zber štatistických údajov (peoplemeter) sledovanosti televíznych kanálov, kde každá odoslaná správa obsahuje jeden parameter, ID práve sledovaného TV kanála. KeepAlive správy sú zasielané asynchrónne portálom každých päť minút.

Aplikačný server pri spracovávaní keep-alive správ prechádza frontou čakajúcich oznamov pre zákazníka. Ak fronta nie je prázdna, vyzdvihne sa najstarší oznam a odošle sa odpovedná informačná správa. Ďalšou frontou ktorou server prechádza sú riadiace správy pre STB. Tieto správy obsahujú preddefinované typy príkazov. Tohto času je podporovaný jediný príkaz a to, vynútené prepnutie na požadovanú multicastovú skupinu. Ak sú obe fronty prázdne, server posiela prázdnu správu.

Príklad:

Prichodzia správa

STB → MW: { c:1,s:SESSION_ID,m:{ currChId:10 } }

Odpovedné správy

Odoslanie informačnej správy v odpovedi na keep-alive správu

STB ← MW: { type:1,voidable:1,title:"Oznam",text:"Zajtra sa uskutočni planovana odstavka" }

Zablokovanie portálu pomocou informačnej správy

STB ← MW: { type:2,voidable:0,title:"Upomienka",text:"Poskytovanie služieb je pozastavene" }

Vynútenie prepnutie na multicastovú skupinu 239.1.1.1 a port 1234

STB ← MW: { type:2,voidable:1,title:"Zmena PIN",text:"Zadany PIN sa nezhoduje s aktualnym" }

GetUserChannels (100)

Tento typ správy je bez vstupných parametrov a je zasielaný pri inicializácii portálu. Server pri spracovaní tohto požiadavku vyexportuje zoznam TV kanálov, ktoré má zákazník predplatené. Okrem tohto zoznamu je v odpovednej správe zasielaná aj konfigurácia užívateľských nastavení (poradie kanálov a rodičovský zámok).

Príklad:

STB → MW: { c:100,s:SESSION_ID,m:{} }

STB ← MW: { [{chId:1,ord:2,lock:0 },{chId:2,ord:3,lock:1 },{chId:3,ord:1,lock:0 }] }

SetUserChannels (101)

Správa s požiadavkou na uloženie užívateľskej konfigurácii na strane MW. Vstupným parametrom je zoznam kanálov a ich poradie definované užívateľom a stavy rodičovského zámku. Táto správa sa posieľa v okamihu keď zákazník iniciuje uloženie konfigurácie. Server na túto správu odpovedá informačnou správou typu 1 v prípade že uloženie bolo úspešné, v opačnom prípade je odpovedná správa typu 2 a informuje zákazníka o neúspechu.

Príklad:

STB → MW: { c:101,s:SESSION_ID,m:{ [{chId:1,ord:2,lock:0 },{chId:2,ord:3,lock:0 }] } }

STB ← MW: { type:1,voidable:1,title:"Poradie kanalov",text:"Zmeny uspesne ulozene " }

GetUserCategories (110)

Správa je podobne ako č. 100 bez vstupných parametrov a je zasielaná po spustení portálu. Server v odpovednej správe pošle zoznam zákazníkom definovaných kategórií a zoznam kanálov priradených k nim.

Príklad:

STB → MW: { c:110,s:SESSION_ID,m:{} }

STB ← MW: { [{name:"Sport",chs:[3] }, {name:"Doku",chs:[2,5] }, {name:"Film",chs:[1,6] }] }

SetUserCategories (111)

Analogicky k správe č. 101, nastavuje vlastné kategórie a k nim priradené kanály definované zákazníkom.

Príklad:

STB → MW: { c:111,s:SESSION_ID,m:{ [{name:"Music",chs:[8,11,20] }] } }

STB ← MW: { type:1,voidable:1,title:"Kategorie",text:"Zmeny uspesne ulozene" }

ChangePIN (200)

Táto správa sa posiela pri požiadavke o zmenu PIN kódu nákupu. PIN kód sa overuje až na strane middlewaru, preto ako vstupné hodnoty sa so správou posielajú zadaný aktuálny PIN a nový PIN. V prípade, že zadaný aktuálny PIN sa zhoduje so skutočným, je požiadavka na zmenu úspešná. O tomto stave je zákazník informovaný oznamom, ktorý je zobrazený po obdržaní informačnej správy.

Príklad:

STB → MW: { c:200,s:SESSION_ID,m:{ currPIN:0000, newPIN:1234 } }

STB ← MW: { type:2,voidable:1,title:"Zmena PIN",text:"Zadany PIN sa nezhoduje s aktualnym" }

GetAvailableOffers (201)

Pri každom vstupe do samoobsluhy sa aktualizuje zoznam aktuálnej ponuky programových TV balíčkov. Tento zoznam je získaný pomocou požiadavky zaslaním tejto správy bez vstupných parametrov. Zoznam je generovaný na základe tarify, ktorú má zákazník zakúpenú a obsahuje informácie podľa požiadaviek 5. a 6.

Príklad:

STB → MW: { c:201,s:SESSION_ID,m:{ } }

STB ← MW: { [{id:1,name:"Film+",price:3.99,minDuration:1,chs:[6,17,23]}] }

BuyOffer (202)

Správa odoslaná pri požiadavke na zakúpenie tematického balíčka. Ako vstupný parameter je ID tohto balíka a PIN kód nákupu. V prípade, že bol PIN zadaný správne, zákazník má od tohto momentu rozšírenú programovú ponuku o kanály obsiahnuté v tomto balíku. Výstup tejto správy – odpoveď je informačná správa o výsledku spracovania.

Príklad:

STB → MW: { c:202,s:SESSION_ID,m:{ id:1, pin:1234 } }

STB ← MW: { type:1,voidable:1,title:"Samoobsluha",text:"Tematicky balicek uspene zakupeny" }

GetBoughtOffers (203)

Vráti zoznam aktuálne zakúpených tematických balíčkov. V tomto zozname sú uvedené základné informácie o ponuke vrátane dňa jej zakúpenia a doby jej platnosti. Táto správa je bez vstupných parametrov.

Príklad:

STB → MW: { c:203,s:SESSION_KEY,m:{ } }

*STB ← MW: { [{id:1,name:"Film+",bPrice:3.99,bDate:"2011-04-04",vDate:"2011-05-04"},
{id:2,name:"Doku2",bPrice:1.99,bDate:"2011-05-01",vDate:"0000-00-00"}] }*

CancelOffer (204)

Správa odoslaná pri požiadavke na odhlásenie tematického balíčka. Ako jediný vstupný parameter je ID balíka. Pri odhlasovaní sa od zákazníka nevyžaduje PIN. Odpoveďou je informačná správa o výsledku spracovania.

Príklad:

STB → MW: { c:204,s:SESSION_ID,m:{ id:2 } }

STB ← MW: { type:1,voidable:1,title:"Samoobsluha",text:"Vas balicek bude platny do 2011-06-01" }

GetAvaibleVOD (301)

Ak sa zákazník prepne do menu virtuálnej video požičovne portál generuje tento typ správy. Správa je bez vstupných parametrov a odpoveďou je zoznam dostupných videí, ktoré si zákazník môže zakúpiť. V tomto zozname sú uvedené základné charakteristiky o videi ako je názov, rok, jazyk a krátky popis. Okrem toho je tu uvedená jeho cena a doba po ktorú si môže zákazník video pozrieť po zakúpení.

Príklad:

STB → MW: { c:301,s:SESSION_ID,m:{ } }

STB ← MW: { [{id:1, name:"Vod1", price:0.99, dur:24, dscr:{year:2010, len:120, lang:SK-CZ, plot:"Popis videa" } }] }

BuyVOD (302)

Analogicky ako je to u správy č. 202. Vstupným parametrom je ID VOD ponuky.

Príklad:

STB → MW: { c:302,s:SESSION_ID,m:{ id:1, pin:1235 } }

STB ← MW: { type:2,voidable:1,title:"VideoPozicovna",text:"Nespravny PIN nakupu!" }

GetBoughtVOD (303)

Vráti zoznam aktuálne zakúpených videí z virtuálnej video požičovne. V tomto zozname sú uvedené základné informácie o zakúpení a najmä zostávajúci čas, dokedy si zákazník vie dané video pozrieť. Táto správa je bez vstupných parametrov.

Príklad:

STB → MW: { c:303,s:SESSION_ID,m:{ } }

*STB ← MW: { [{id:2,name:"Source Code", bPrice:1.99, bDate:"2010-05-05", leftHours: 48},
{id:1,name:"Scream 4", bPrice:0.99, bDate:"2010-05-04", leftHours: 11}] }*

5.5 RTSP proxy server

Obecne, RTSP proxy je server, ktorý poskytuje služby ako napríklad smerovanie požiadaviek klientov na jeden alebo viacero serverov s multimediálnym obsahom t.j. vie zabezpečiť vyvažovanie záťaže. Proxy server taktiež môže podporovať cachovanie obsahu, a pri dotaze na video sa prehrávanie spustí z lokálanej cache. V našom prípade bude RTSP proxy server zabezpečovať autentizáciu a autorizáciu klientov.

Komunikácia medzi STB a RTSP proxy je založená na protokole s rovnakým názvom RTSP (Real-Time Streaming Protocol). Je to textový protokol pre riadenie multimediálnych tokov s podobnou syntaxou a metódami ako HTTP/1.1. Narozdiel od http protokolu, ktorý je bezstavový, rtsp má stav a definované metódy pre obojsmernú komunikáciu. Informácie ohľadne šírenia multimediálneho obsahu prostredníctvom počítačovej siete vie čitateľ dohľadať v tejto literatúre [13].

5.5.1 RTSP metódy

RTSP definuje niekoľko metód a síce pre splnenie našich požiadaviek a zabezpečenie celkovej funkčnosti ovládania je potrebné naimplementovať nasledovné metódy (pri každej metóde je uvedený krátky popis a príklad komunikácie):

OPTIONS

Vráti zoznam metód, ktoré server akceptuje

Príklad:

```
K→S: OPTIONS * RTSP/1.0
      CSeq: 1
S→K: RTSP/1.0 200 OK
      CSeq: 1
      Public: DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE
```

DESCRIBE

Získa popis multimediálneho obsahu na požadovanej URL adrese špecifikovanej klientom. Tento popis je vo formáte SDP⁴ a obsahuje informácie ako je počet dostupných audio stôp, dĺžka média atď. V tomto príklade je odpoveď zo servera ilustratívna a neobsahuje všetky parametre.

Príklad:

```
K→S: DESCRIBE rtsp://192.168.1.1/1.sdp RTSP/1.0
      CSeq: 2
S→K: RTSP/1.0 200 OK
      CSeq: 2
      Content-Base: rtsp://192.168.1.1/1.sdp
      Content-Type: application/sdp
      Content-Length: 76

      m=video 0 RTP/AVP 3
      a=control:track1
      a=range:npt=0-2760.4368
      ...
```

4 http://en.wikipedia.org/wiki/Session_Description_Protocol

SETUP

Táto metóda slúži k nastaveniu parametrov samotného prenosu multimedialného toku. Klient v požiadavke uvedie transportný protokol a porty na ktorých bude očakávať multimedialný obsah. Server pri obdržaní tejto požiadavky vytvorí novú reláciu pre toto spojenie a potvrdí parametre.

Príklad:

```
K→S: SETUP rtsp://192.168.1.1/1.sdp/track1 RTSP/1.0
      CSeq: 3
      Transport: RTP/AVP;unicast;client_port=4005-4006
S→K: RTSP/1.0 200 OK
      CSeq: 3
      Transport: RTP/AVP;unicast;client_port=4005-4006
      Session: 12345678
```

PLAY

Ak server obdrží požiadavku s touto metódou spustí prehrávanie multimedialného obsahu ustanoveného metódou SETUP. Ako parameter tejto správy môže byť uvedený časový rozsah, ktorý má byť prehraný.

Príklad:

```
K→S: PLAY rtsp://192.168.1.1/1.sdp RTSP/1.0
      CSeq: 4
      Range: npt=0-2760.4368
      Session: 12345678
S→K: RTSP/1.0 200 OK
      CSeq: 4
      Range: npt=0-2760.4368
      Session: 12345678
```

PAUSE

Dočasne preruší prehrávanie multimedialného toku. Pre pokračovanie sa očakáva požiadavka s metódou PLAY.

Príklad:

```
K→S: PAUSE rtsp://192.168.1.1/1.sdp RTSP/1.0
      CSeq: 5
      Session: 12345678
S→K: RTSP/1.0 200 OK
      CSeq: 5
      Session: 12345678
```

GET_PARAMETER

Obecne sa táto metóda používa na získanie parametrov špecifikovaných v požiadavke zo servera respektíve v tomto prípade je možné túto správu zaslať aj na klienta.

Príklad:

```
S→K: GET_PARAMETER 192.168.1.1/1.sdp RTSP/1.0
      CSeq: 6
      Content-Type: text/parameters
      Session: 12345678
      Content-Length: 16
      packets_received
```

```

K→S: RTSP/1.0 200 OK
      CSeq: 6
      Content-Length: 20
      Content-Type: text/parameters

```

```

packets_received: 10

```

TEARDOWN

S touto metódou server ukončí prehrávanie a uvoľní všetky alokované prostriedky

Příklad:

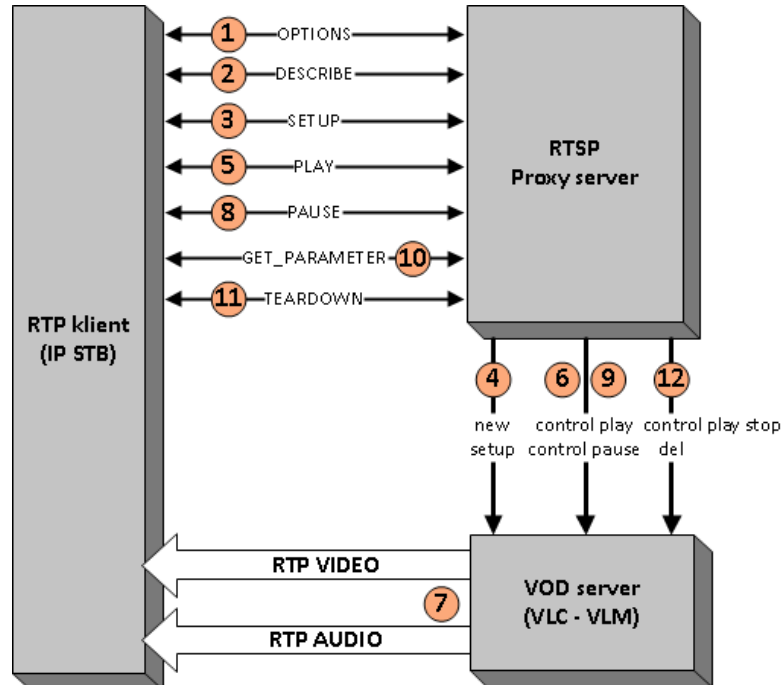
```

K→S: TEARDOWN rtsp://192.168.1.1/1.sdp RTSP/1.0
      CSeq: 7
      Session: 12345678
S→K: RTSP/1.0 200 OK
      CSeq: 7

```

5.5.2 Komunikácia

Na obrázku č. 12 je znázornená komunikácia RTSP proxy servera ako prostredníka medzi STB a VOD serverom. Pre komunikáciu RTSP proxy s VOD sa využíva telnet rozhranie a príkazy uvedené v kapitole 4.1.2. RTP klient (STB) komunikuje s RTSP proxy podľa RFC 2326 t.j.metódami zmienými v predchádzajúcej podkapitole. Čísla u jednotlivých požiadavkách určujú poradie ich vykonávania.



Obrázok 12: Bloková schéma RTSP proxy

Komunikácia z obrázku č. 12 by sa dala charakterizovať takto:

1. Pri požiadavke na prehranie VOD sa STB zariadenie pripojí na RTSP proxy a s metódou OPTIONS získa „funkčnosť“ servera.
2. Klient požaduje informácie o multimedialnom obsahu. V tejto fáze server autentizuje zariadenie na základe jeho IP adresy, ktorá sa spárovala so STB pri prvotnej autentizácii. Následne prebieha autorizácia - proxy získa informácie o zákazníkovi a overuje stav predplatených služieb, tj. či má zákazník právo prehrávať vyžiadaný obsah. Za normálnych okolností neúspešná autorizácia nemôže nastať, pretože portál na strane zákazníka neposkytne služby, ktoré nie sú predplatené resp. ich platnosť skončila. Neautorizovaný požiadavok skončí s chybovým kódom 403 Forbidden
3. V prípade úspešnej autentizácie RTP klient obdrží odpoveď s kódom 200 OK, a pokračuje zaslaním požiadavky pre ustanovenie relácie s RTSP proxy.
4. Ak je v sieti použitých niekoľko VOD serverov, ktoré obsahujú rovnaký obsah ako je vyžiadaný, môže proxy server rozhodnúť na ktorý bude ďalej smerovať požiadavku klienta a to na základe jeho lokality. So zvoleným VOD serverom vytvorí telnet sedenie a ustanoví novú reláciu.
5. Po úspešnom ustanovení spojenia, RTP klient posielá požiadavku na prehratie obsahu.
6. Proxy server zkonštruje správu pre VLM a pošle príkaz na štart streamovania vyžiadaného obsahu.
7. V tomto okamihu VLM vytvorí unicastové spojenie medzi VLC a STB a začne streamovať data cez RTP protokol.
8. Požiadavok klienta na dočasné zastavenie prehrávania.
9. Príkaz na pozastavenie streamovaného obsahu.
10. Pre testovanie živosti RTP klienta sa používa metóda GET_PARAMETER s prázdny telom. Ak by došlo k prerušeniu spojenia, streamovaný obsah by sa automaticky neukončil a pokračoval by až na koniec obsahu. Kvôli takýmto stavom je zariadenie kontrolované každých 30 sekúnd a v prípade neobdržania odpovedi sa nadviazané spojenie ukončí a prehrávanie zastaví (12)
11. Ukončenie relácie zo strany klienta
12. Príkaz na ukončenie streamovaného obsahu a uvoľnenie prostriedkov

5.6 Rozhranie systému

V doterajších podkapitolách sme popisovali vzťahy a komunikáciu portálu resp. STB s middlewarom a návrh tohto systému. Táto podkapitola sa bude zaoberať realizáciou rozhrania pre správu a kontrolu celej platformy. Ovládanie systému je zabezpečené pomocou webovej služby. Tento prístup je univerzálny a umožňuje ľahkú integráciu do účtovacích, kontrolných a dohľadových systémov. Tieto systémy môžu byť implementované v rôznych programovacích jazykoch. V našom projekte je vytvorená jednoduchá implementácia tohto rozhrania ako modul už do súčasného billing systému predstaveného v [7].

5.6.1 Webová služba

Webová služba umožňuje aplikáciám komunikovať a vymieňať informácie prostredníctvom siete. Tieto aplikácie fungujú na báze architektúry klient-server, kde komunikácia prebieha prostredníctvom SOAP správ. Typicky sú tieto správy doručované protokolom HTTP v XML tvare. SOAP správu tvorí koreňová XML značka *Envelope* ktorá obsahuje dve vnorené značky, nepovinnú hlavičku *Header*, a telo, označené značkou *Body*. Hlavička sa väčšinou používa na prenos pomocných informácií ako je napríklad autentizácia klienta. V tele správy sa prenášajú informácie identifikujúce volanie vzdalenej procedúry a obsahuje hlavnú značku s rovnakým názvom ako je volaná procedúra. Zanorené značky potom predstavujú parametre volania. Pre bližšie pochopenie tejto problematiky ako i úvod do webových služieb odporúčame nahliadnuť do [16].

Komunikácia prostredníctvom SOAP správ môže vyzerat' nasledovne:

Požiadavka na získanie informácií o kanály:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ns="urn:mw">
  <SOAP-ENV:Body SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <ns:getChannel>
      <chId>21</chId>
    </ns:getChannel>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Tieto informácie sú na strane servera získané prostredníctvom funkcie *getChannel*, kde vstupným parametrom je ID kanála (*chId*). Funkcia vracia odpoveď v štruktúre *Channel*. Táto štruktúra obsahuje id, názov, typ, kategóriu a zdroje kanála. Zdroj je taktiež zložený typ a obsahuje id, multicastovú IP a port, poznámku a lokalitu pre ktorý je tento zdroj primárny. Jeden kanál môže mať definovaných viacero zdrojov, preto je položka zdroje v odpovedi ako pole zdrojov dátového typu *ChSource*.

Odpoveď na požiadavku: (na príklade je skrátenejší výpis koreňovej značky)

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
...
  xmlns:ns="urn:mw">
  <SOAP-ENV:Body SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <ns:Channel>
      <chId>21</chId>
      <name>Discovery</name>
      <type>TV</type>
      <category>DOCUMENTARY</category>
      <sources SOAP-ENC:arrayType="ns:ChSource[1]">
        <item>
          <sId>1</sId>
          <localityId>0</localityId>
          <mip>239.1.1.57</mip>
          <port>1234</port>
          <note></note>
        </item>
      </sources>
    </ns:Channel>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

5.6.2 Middleware – SOAP server

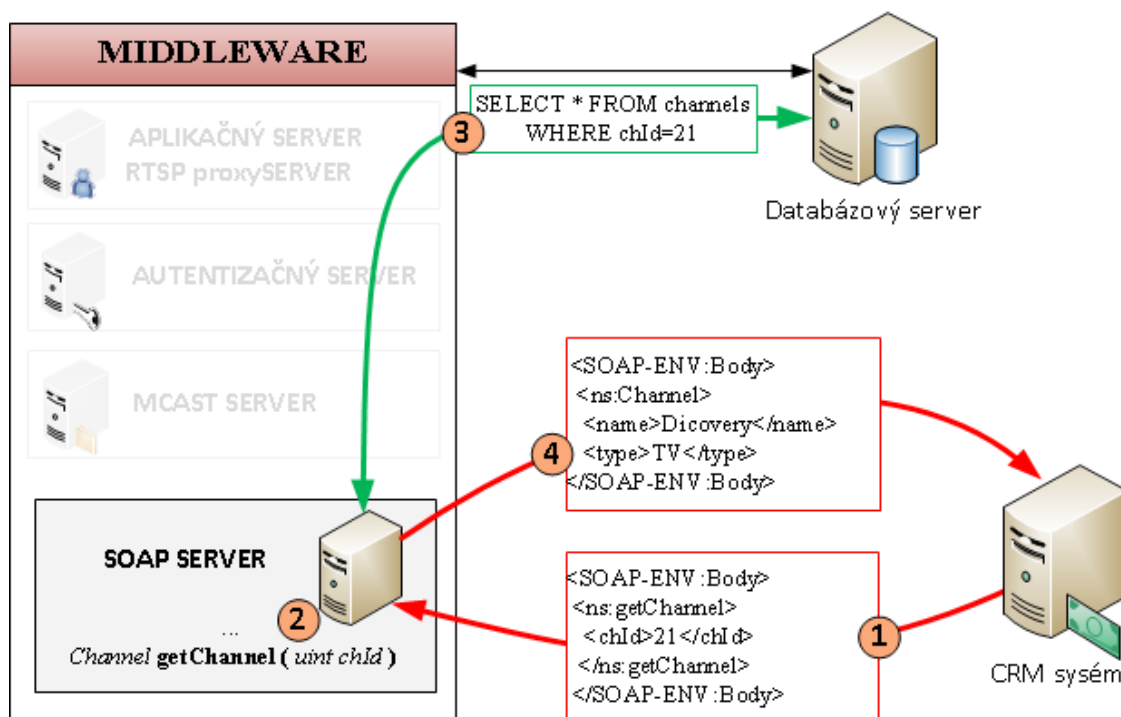
Webovú službu v našej navrhutej architektúre poskytuje podkomponenta middlewaru označená ako SOAP server. Tento server zabezpečuje vstupno/výstupné rozhranie pre správu celého systému pomocou ostatných CRM systémov. SOAP server predovšetkým implementuje a poskytuje možnosti správy v týchto oblastiach:

- správa lokalít
- správa kanálov a zdrojov
- správa programových balíkov
- správa programových ponúk
- správa VOD
- správa tarifov
- správa koncových zariadení (STB)
- správa zákazníkov a ich účtov
- nastavenia systému

Na obrázku č. 13 je naznačená komunikácia medzi MW a CRM systémom. Komunikácia vychádza z ukázkových SOAP správ uvedených v kapitole 5.5.1 (na obrázku pre prehľadnosť nie sú uvedené kompletné).

Táto komunikácia by sa dala charakterizovať týmito krokmi:

1. Operátor iniciuje zobrazenie detailu nejakého kanála (v príklade 21). CRM systém so SOAP klientom pošle požiadavku na zobrazenie informácií o kanály.
2. SOAP server čaká na prichádzajúce spojenie. Pri novej požiadavke rozbalí správu a podľa nej zavolá funkciu *getChannel* s hodnotou parametru obsiahnutej v správe.
3. Klient v tomto prípade požaduje informácie, ktoré sa nachádzajú v databáze. Zavolaná funkcia sa spojí s DB serverom a položí mu správne sformulovaný SQL dotaz.
4. Výsledok SQL dotazu je spracovaný a zabalený naspäť do SOAP správy a odoslaný klientovi.



Obrázok 13: SOAP komunikácia

Aby sme mohli vzdialene volať funkcie prostredníctvom SOAP protokolu musíme mať k dispozícii rozhranie webovej služby. Pre popis rozhrania sa používa jazyk WSDL (Web Service Definition Language). WSDL je jazyk založený na XML, ktorý popisuje dostupné operácie služby, dátové typy, štruktúru SOAP správ, miesto uloženia služby a i. Preto WSDL dokument by sa dal prirovnať k hlavičkovým súborom .h v jazyku C, ako zoznam názvov funkcií s ich typmi parametrov a návratových hodnôt.

WSDL popis je obdobou IDL popisu pri vzdialenom volaní procedúr, a preto existujú automatizované nástroje, ktoré z WSDL vedia vygenerovať zástupný kód (stub) pre volanie danej služby vo zvolenom programovacom jazyku [16]. Avšak existuje i opačný prístup, kde sa WSDL generuje z kódu príslušného programovacieho jazyka. Tento prístup sme si zvolili i my s použitím SOAP toolkitu spomenutého v kapitole 5.6.

5.6.3 CRM systém – SOAP klient

Nezávislosť na technickom prevedení na strane CRM systému (voľba programovacieho jazyka a spracovanie) zaisťuje webová služba ako komunikačný protokol, v našom prípade SOAP, a špecifikácia rozhrania tejto služby v jazyku WSDL na strane middlewaru. Ako sme v úvode naznačili, celá správa navrhnutého systému pre sledovanie IPTV je implementovaná ako modul už do stávajúceho IS komerčného poskytovateľa internetových služieb. Tento IS je kompletne napísaný v jazyku PHP ako modulárny systém. Časť modulu, ktorá zobrazuje informácie o kanály ilustruje nasledovný kód (pre jednoduchosť je vynechaná logika spojená s interpretáciou výsledku ako html):

```
$soap = new SoapClient('/cesta/k/popisu_wsdl/middleware.wsdl');
$chInfo = $soap->getChannel(21);
print_r($chInfo);
```

Výstup:

```
Array
(
    [chId] => 21
    [name] => Discovery
    [type] => TV
    [category] => DOCUMENTARY
    [sources] => Array
        (
            Array
            (
                [sId] => 1
                [localityId] => 0
                [mip] => 239.1.1.57
                [port] => 1234
                [note] => Zo satelitu
            )
        )
)
```

Vzhľadom na to, že celá logika je presunutá na SOAP server, klient pozostáva iba z volania vzdialenej procedúry/funkcie a interpretácie výsledku tohto volania. Vyššie uvedený zdrojový kód zobrazuje volanie metódy *getChannel()*. Komunikácia a detail posielaných správ pri volaní tejto metódy bol vysvetlený v predchádzajúcich kapitolách.

Tento diplomový projekt sa nezaobrá kompletnou implementáciou SOAP klienta pre nami navrhnuté rozhranie, ale sú implementované len časti nevyhnutné pre zabezpečenie prevádzky IPTV služieb. Tento fakt plynie napríklad i z požiadavky č. 15, kde evidencia zákazníkov na strane middlewaru nevyžaduje ani jeden povinný atribút okrem identifikátora zákazníka. Bolo by neefektívne udržiavanie týchto informácií na dvoch miestach, pretože ISP vo svojom systéme má zahrnutú kompletnú agendu pre správu kontaktov svojich zákazníkov a ich predplatených službách.

5.7 Použité technológie pri implementácii

5.7.1 Aplikačný a RTSP proxy server

Ako sme uviedli v kapitole 5.3 portál komunikuje s aplikačným serverom pomocou http protokolu. Z tohto dôvodu sme zvolili nasadenie voľne dostupného web servera apache⁵. Tento server podporuje obsluhu niekoľko stoviek požiadaviek za sekundu a v kombinácii so ssl modulom je zabezpečená aj šifrovaná komunikácia. Ako programovací jazyk bol zvolený PHP, ktorý je platformovo nezávislý s jednoduchou integráciou do web servera. PHP je dynamický skriptovací jazyk a v súčasnosti veľmi rozšírená technológia umožňujúca programovanie na strane serveru s množstvom rozširujúcich modulov.

RTSP proxy server je taktiež napísaný v jazyku PHP a síce ako samostatná aplikácia. Táto aplikácia beží ako konkurentný server, t.j. pri pripojení klienta vytvorí nové vlákno, v ktorom začne jeho obsluhu a trvá až pokiaľ klient neukončí spojenie respektíve nenastane timeout.

5.7.2 Komponenty middlewaru

Zostávajúce komponenty navrhovaného systému (mcast server, autentizačný server, soap server) sú implementované v jazyku C++ ako viac vláknová aplikácia. Pri spustení aplikácie sú všetky potrebné konfiguračné nastavenia získané z databázy. To umožňuje dynamicky meniť konfiguráciu bez nutnosti prihlasovať sa na systém kde služba beží. Potrebnú konfiguráciu je možné meniť výlučne cez navrhnuté rozhranie webovej služby. Webová služba (soap server) je postavená na báze knižnice gSOAP, ktorá je vysoko optimalizovaná pre výkon. Tento toolkit podporuje konštrukcie ako v čistom C tak i C++ s podporou STL. Bezpečnostné mechanizmy u mcast servera a autentizačného servera sú riešené pomocou openssl knižnice.

5.7.3 Databázový systém

Pre účely ukladania a správy dát bol zvolený databázový systém MySQL. Je to multiplatformová databáza s podporou viacerých typov úložísk. V našom projekte je zvolený engine typu InnoDB. Toto úložisko je vhodné v prípadoch, keď sa tabuľky často menia, navyše umožňuje používanie cudzích kľúčov a transakčné spracovanie.

Štruktúra databázy je zobrazená v dátovom diagrame v prílohe č. 4. Z tohto ERD diagramu boli vyňaté niektoré tabuľky, ktoré sa priamo nedotýkajú samotného návrhu systému pre sledovanie IPTV (napr. užívatelia a ich práva pre management systému).

5 <http://www.apache.org/>

6 Záver

Cieľom tejto práce bolo navrhnuť a implementovať systém pre sledovanie IPTV v rámci lokálnej siete poskytovateľa internetového pripojenia (ISP). Navrhovaný systém mal zautomatizovať správu koncových zariadení – settopboxov (STB), televíznych kanálov, videopožičovni a v neposlednej rade správu užívateľských účtov a napojenie na stávajúci systém, ktorý som vyvinul v rámci svojej BP[7].

Práca popisuje architektúru a základné prvky systému, ktoré sú vhodné k tomuto účelu. Ďalej sa potom zameriavam na detailnú analýzu požiadavkov a na jejich základe navrhujem vlastné riešenie, ktoré som následne implementoval a uviedol do reálnej prevádzky. V rámci práce som vykonal implementáciu webového portálu na strane STB, pomocou ktorého je zabezpečená interakcia s IPTV službou. Navrhnutý a implementovaný systém umožňuje autentizáciu a vzdialenú konfiguráciu STB, služby videopožičovni, zobrazovania informácií na TV (zoznam programov, výpis jednotlivých staníc, a pod.) a ďalšie. Celý systém bol od počiatku navrhovaný a implementovaný ako komponenta do stávajúceho IS a vďaka použitému SOAP protokolu poskytuje nezávislosť na programovacom jazyku, tzn. túto komponentu je možné zahrnúť i do iných IS s minimálnym množstvom práce a nákladov.

Portál je navrhnutý tak, aby sa dal ľahko modifikovať a poskytuje rovnaké funkčné a zobrazovacie vlastnosti ako webová stránka. Umožňuje tak zobrazovanie informácií a poskytovanie služieb, ktoré nemusia nutne súvisieť s televízou. Napr: informácie o počasiu vo forme widgetov, informácie o stave siete, obraz z vybraných kamier poskytovateľa a pod. Možným rozšírením stávajúceho systému je možné doimplementovať rôzne komerčné i nekomerčné služby ako je podpora emailového klienta vrátane kontroly a notifikačných správ, objednávkový systém služieb (pizza, taxi, atď.). Ďalej je možné integrovať podporu sociálnych sietí, tak ako sa s tým už môžeme stretnúť u dnešných „smart“ televízorov.

Uvedené rozšírenia, ktoré robia TV interaktívnou implementuje takmer každá komerčná IPTV platforma, no však čo chýba a pre poskytovateľa veľmi užitočná vlastnosť je distribuovaný dohľad nad multicastmi monitorovaný priamo na IP STB zákazníka. Integrácia dohľadového systému priamo do STB a zapracovaním tejto funkcionality do IPTV middlewaru tak tvorí kompletný systém pre správu a management IPTV. Táto predstava je smer, ktorým by sa chcel prioritne zaoberať pri ďalšom vývoji mňou navrhnutého systému.

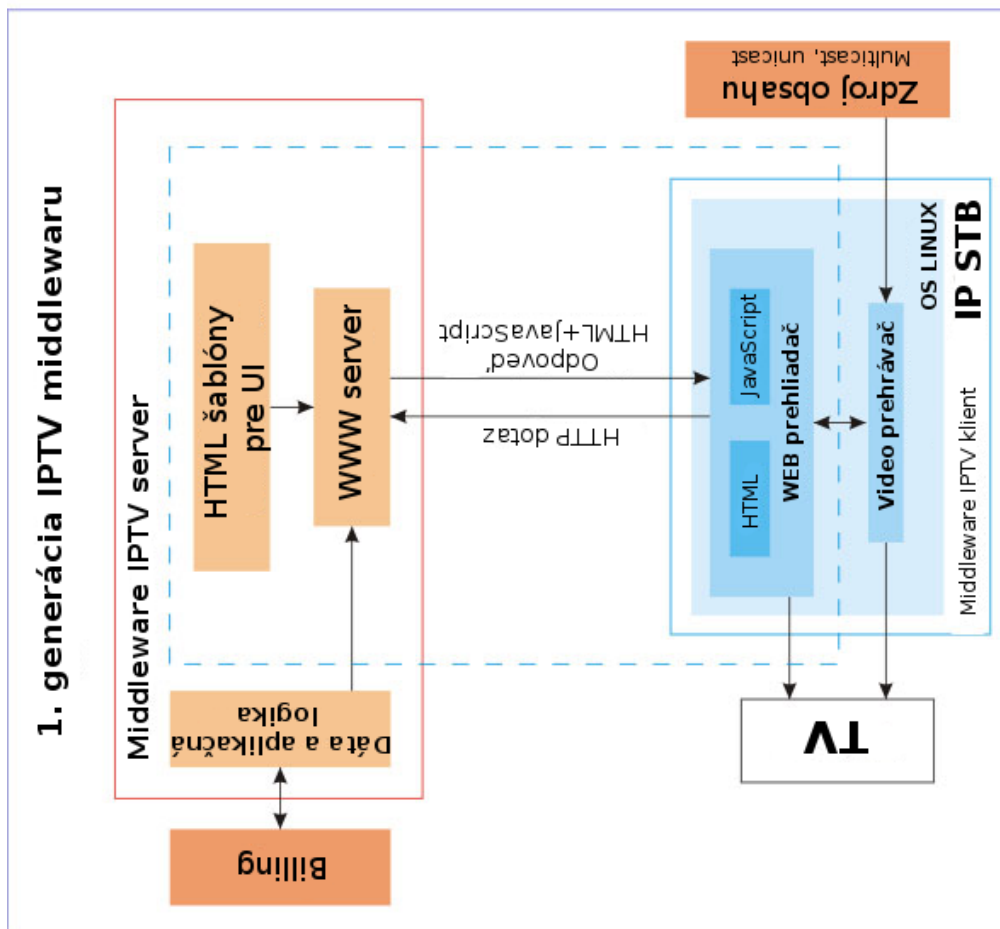
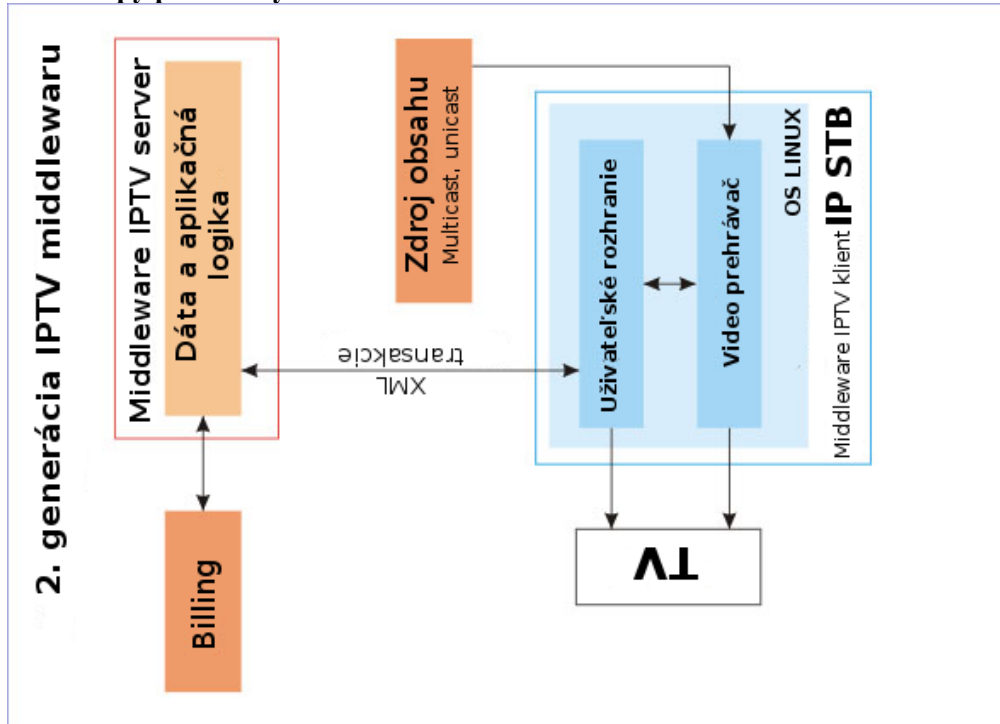
Literatura

- [1] Hens, F. J., Cabellero, J. M. Triple Play: Building the converged network for IP, VoIP and IPTV, Wiley, 2008. ISBN 978-0470753675.
- [2] Minoli, D. IP multicast with applications to IPTV and mobile DVB-H, Wiley, 2008. ISBN 978-0-470-25815-6.
- [3] ISO/IEC 13818-1:2000 (ITU-T Recommendation H.222.0). Generic coding of moving pictures and associated audio information: Systems, 2000.
- [4] Schulzrinne, H., Casner, S., Frederick, R. RTP: A Transport Protocol for Real-Time Applications, STD 64, RFC 3550, 2003.
- [5] Arberg, P., Cagenius, T., Tidblad, O., Ullerstig, M., Winterbottom, P. Network infrastructure for IPTV, Ericsson. Review No. 3, 2007, [cit. 2010-12-20]. Dostupné z URL: http://www.ericsson.com/ericsson/corpinfo/publications/review/2007_03/
- [6] Veselý, V. Architektúry systémů na internetu se skupinovým adresováním, diplomová práce, Brno, FIT VUT v Brně, 2009
- [7] Pápai, M. Systém pro administraci komerční sítě, bakalářská práce, Brno, FIT VUT v Brně, 2009
- [8] Powell, T. Ajax: The Complete Reference, McGraw-Hill, 2008, ISBN 978-0-07-149216-4
- [9] Zakas, N. Professional JavaScript for Web Developers, Wiley, 2009, ISBN 978-0-470-22780-0
- [10] Lindley, C. jQuery Cookbook, O'Reilly Media, 2009, ISBN 978-0-596-15977-1
- [11] Viega J., Messier M., Chandra P. Network Security with OpenSSL, O'Reilly Media, 2009, ISBN 978-0-596-10345-3
- [12] Schulzrinne H., Rao A., Lanphier R. Real Time Streaming Protocol (RTSP), RFC 2326, 1998
- [13] Simpson, W. Video Over IP, Elsevier, 2008, ISBN 978-0-240-81084-3
- [14] Yaghmour, K. Building embedded LINUX systems, O'Reilly Media, 2003, 978-0-596-00222-0
- [15] Gallivan, K.A., Engelen, R.A. The gSOAP toolkit for Web services and peer-to-peer computing networks, Cluster Computing and the Grid 2nd IEEE/ACM International Symposium CCGRID2002 (2002), 117-124.
- [16] Kuba, M.: Web services. [online][cit. máj 2011]. Dostupné z URL: http://www.ics.muni.cz/~makub/soap/MartinKuba_WebServices_Datakon2006_clanek.pdf
- [17] Welling, L., Thomson, L. PHP a MySQL rozvoj webových aplikací. Softpress, Praha, 2002. ISBN 8086497836
- [18] JavaScript API for controlling IPTV devices. [online][cit. máj 2011]. Dostupné z URL: [http://soft.infomir.com.ua/mag200/Doc/JS%20API%20MAG200%20\(R%201.17\).pdf](http://soft.infomir.com.ua/mag200/Doc/JS%20API%20MAG200%20(R%201.17).pdf)

Zoznam príloh

- Príloha 1. Prístupy pre tvorbu IPTV middlewaru
- Príloha 2. Špecifikácia zvolených STB
- Príloha 3. Use case model – portál
- Príloha 4. Dátový diagram návrhu databázy
- Príloha 5. Portál - základné OSD s krátkym EPG
- Príloha 6. Portál - základné OSD s krátkym EPG – reálna ukážka (záber z TV)
- Príloha 7. Užívateľské rozhranie pre administráciu platformy
- Príloha 8. CD so zdrojovými kódmi pre middleware a portál

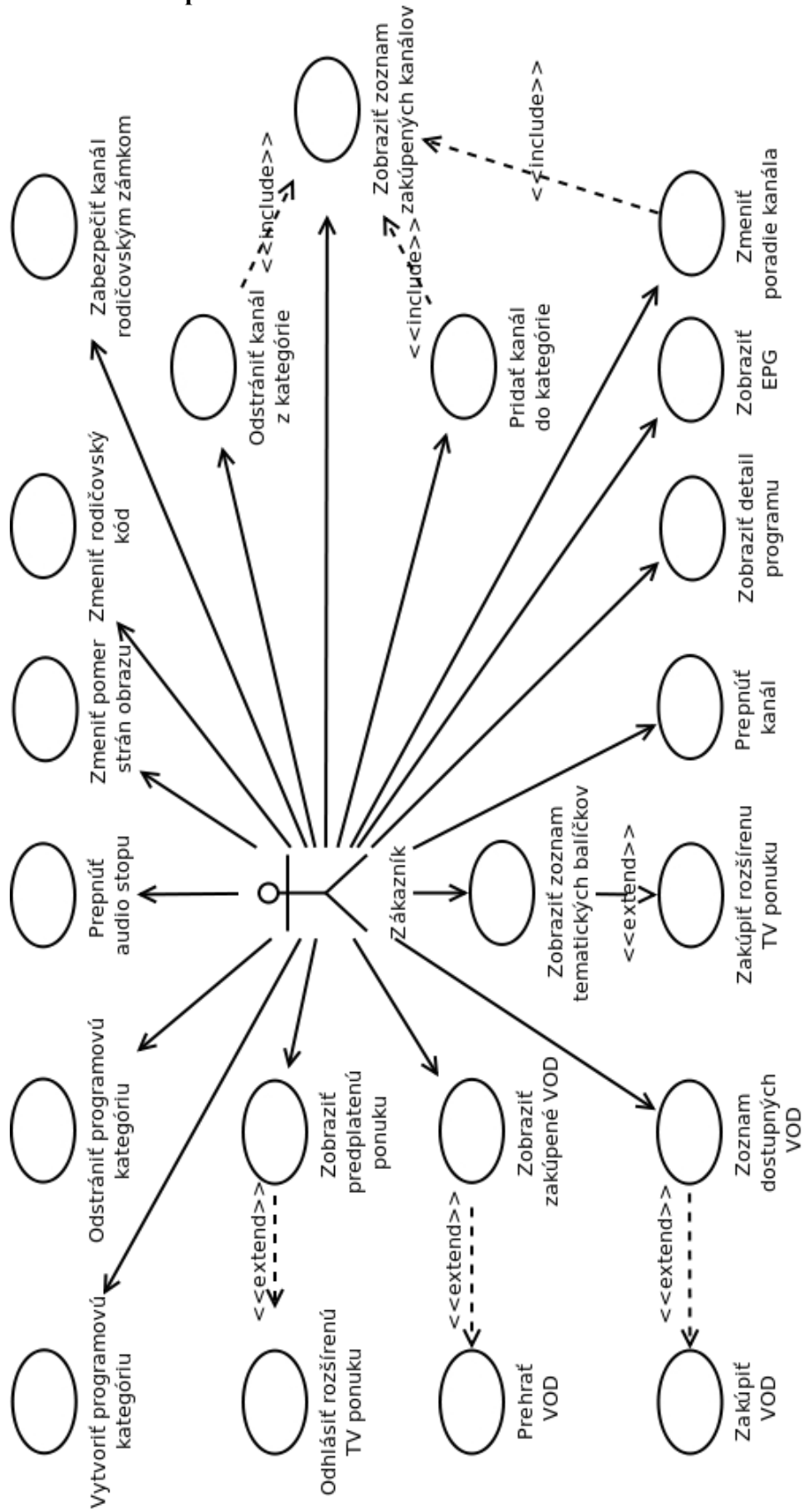
Príloha 1. Prístupy pre tvorby IPTV middlewaru



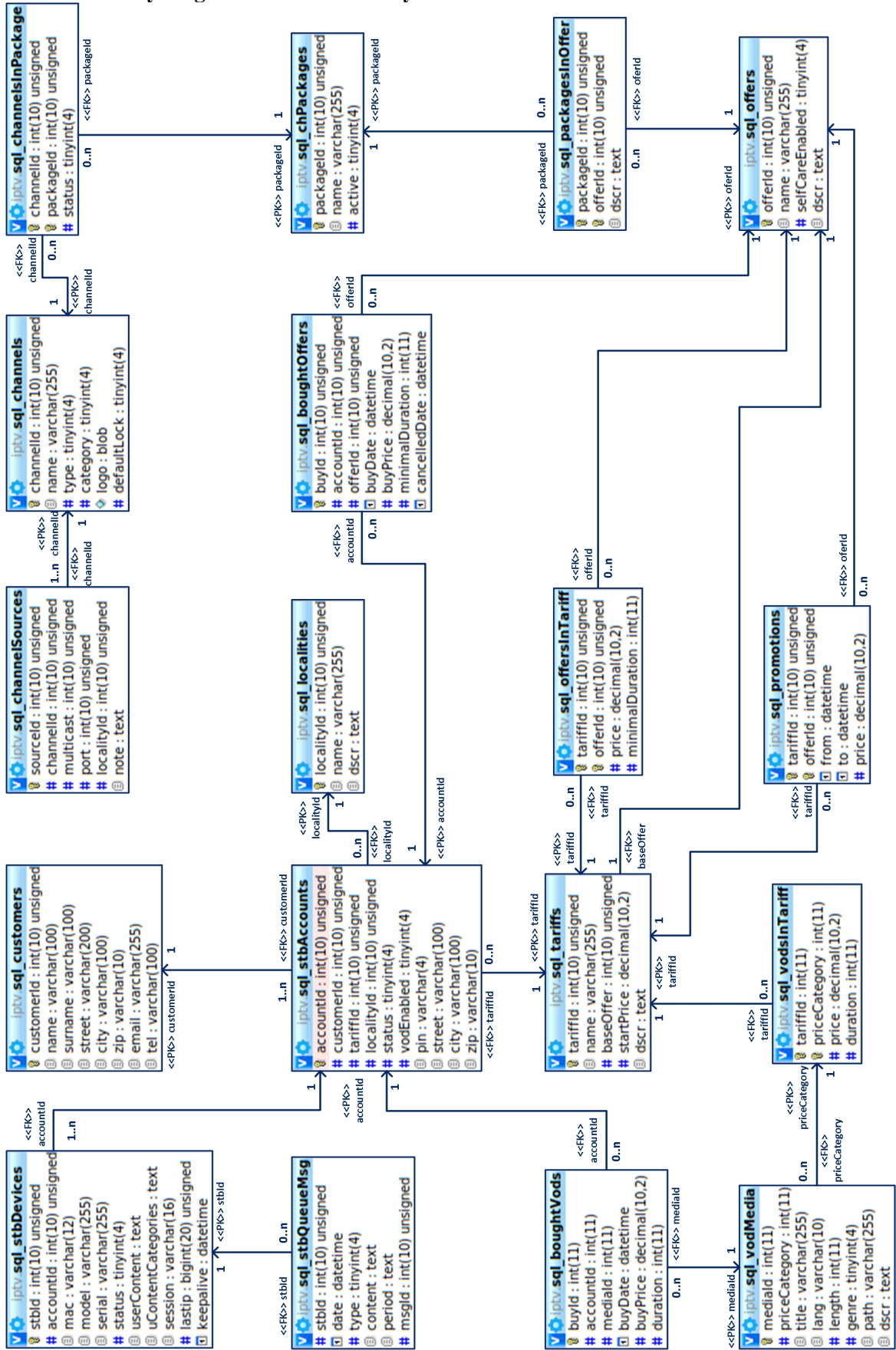
Príloha 2. Špecifikácia zvolených STB

Model	DIB-120	MAG-200
Výrobca	D-Link	Infomir
Prehliadač	Opera HTML 4.0 XML	Webkit HTML 4.01 XHTML 1.0/1.1 XML 1.0, XSLT 1.0, XPath 1.0 SOAP 1.1
Podpora protokolov	RTSP, RTP, UDP	RTSP, RTP, UDP
Video dekodery	MPEG2 MP @ HL / H.264 MPEG-4 part10 MP @ L4	MPEG1/2 MP@HL H.264 HP@level 4.1 MPEG4 part 2 (ASP) VC1 video, WMV-9
Audio dekodery	MPEG-1 layer I/II MPEG-2 layer II/III MPEG-4 AAC	MPEG-1 layer I/II MPEG-2 layer II/III MPEG-4 AAC
Video výstup	HDMI, S-Video, Component RGB	RCA, S-Video, SCART, HDMI, Component RGB or (Y Pr Pb)
USB porty	2	1
RAM DDR	256 MB	128 MB
NAND Flash	64 MB	128 MB
Architektúra	Mipsel	SuperH

Príloha 3. Use case model – portál



Príloha 4. Dátový diagram návrhu databázy



Príloha 5. Portál - základné OSD s krátkym EPG



(Inšpirácia z nangu.tv)

Popis:

- 1 – aktuálne sledovaný kanál
- 2 – názov sledovaného programu
- 3 – krátky popis programu
- 4 – začiatok, koniec, časová osa
- 5 – čas a názov nasledujúcich programov
- 6 – zoznam predplatených kanálov
- 7 – menu

Príloha 6. Portál - základné OSD s krátkym EPG – živá ukážka (záber z TV)



Príloha 7. Užívateľské rozhranie pre administráciu platformy

Nový kanál

Mcast kanál


Názov: Poznámky:

Kategória: ▼

Typ: ▼

Zámok:

Zdroje

Multicast IP:port: 

Pridávanie nového kanálu a definícia multicastových skupín

IPTV tarifa

Názov:

Štartovacia cena: €

Základná ponuka: ▼

Mesačná sadzba: €

Poznámky:

Ostatné ponuky

Ponuka	Cena za mesiac [€]	Minimálna doba predplatného [mesiace]	Stav
Dokument	<input type="text" value="0.99"/>	<input type="text" value="1"/>	<input checked="" type="checkbox"/>
Music	<input type="text" value="0.99"/>	<input type="text" value="1"/>	<input checked="" type="checkbox"/>

Pridávanie novej tarify