

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Future MOD

Dokumentácia k inžinierskemu dielu

Verzia 2.0

Tím č.17 : Future MOD

Vedúci projektu: Ing. Peter Pištek PhD.

Predmet: Tímový projekt II

Študijný program: Softvérové inžinierstvo, ročník: 1

Akademický rok: 2016/2017, letný semester

Tabuľka 1 História zmien

Verzia	Dátum	Autor	Opis
1. kontrolný bod	6.10.2016	Matúš Pohančenič	Vytvorenie dokumentu
	17.10.2016	Tomáš Baránek	Úvod
	18.11.2016	Matúš Slovík	Globálne ciele pre zimný semester
	19.11.2016	Matúš Slovík	Synchronizácia obsahu
	19.11.2016	Tomáš Sokolík	Transcoding
	19.11.2016	Tomáš Baránek	Identifikácia
	19.11.2016	Radoslav Zápach	Webportál
	19.11.2016	Matúš Pohančenič	Multimediálny obsah
	20.11.2016	Lukáš Mastilák	Mobilný internet
	20.11.2016	Jaroslav Tóth	Analýza os
2. kontrolný bod	14.12.2016	Jaroslav Tóth	Ukladací priestor
	14.12.2016	Jaroslav Tóth	Konfigurácia servera
	14.12.2016	Matúš Slovík	Proces synchronizácie obsahu
	14.12.2016	Radoslav Zápach	Reklama
	14.12.2016	Lukáš Mastilák	Logy
	14.12.2016	Jaroslav Tóth	Use case: Inštalatér (Molpir)
	14.12.2016	Matúš Pohančenič	Use case: Cestujúci (1. časť)
	14.12.2016	Matúš Slovík	Use case: Cestujúci (2. časť)
	14.12.2016	Radoslav Zápach	Use case: Stevardka, 2. časť
	14.12.2016	Lukáš Mastilák	Use case: Stevardka, 1. časť
	14.12.2016	Tomáš Baránek	Use case: Správca flotily
	14.12.2016	Tomáš Sokolík	Use case: Cestujúci (3. časť)
Finálne odovzdanie	26.2.2017	Lukáš Mastilák	Fyzický dátový model pre logy z web systému
	7.3.2017	Lukáš Mastilák	Fyzický dátový model pre systémové udalosti
	9.5.2017	Matúš Pohančenič	Fron-end kapitoly Navigácia, Výber jazyka, Prehliadanie filmov, Prehliadanie obrázkov
	9.5.2017	Matúš Slovík	Globálne ciele pre letný semester
	9.5.2017	Jaroslav Tóth	Modul na vzdialenú správu
	9.5.2017	Jaroslav Tóth	Spúšťanie skriptov pri štarte systému

Verzia	Dátum	Autor	Opis
	10.5.2017	Jaroslav Tóth	Jednotkové testy
	10.5.2017	Tomáš Baránek	Zmena úvodnej kapitoly
	11.5.2017	Radoslav Zápach	Webový portál
	11.5.2017	Radoslav Zápach	Fron-end kapitoly Prehliadanie hudby a Stránkovanie
	11.5.2017	Jaroslav Tóth	Finálne údajové modely
	13.5.2017	Lukáš Mastíľak	Rotácia záznamov
	14.5.2017	Jaroslav Tóth	Elektronické médium (príloha A)

Obsah

1	Úvod	1
2	Globálne ciele	3
2.1	Zimný semester	3
2.1.1	Ciele pre jednotlivé šprinty zimného semestra	3
2.2	Letný semester	4
2.2.1	Ciele pre jednotlivé šprinty letného semestra	4
3	Analýza.....	7
3.1	Mobilný internet.....	7
3.1.1	Pokrytie a rýchlosti v mobilnej sieti	7
3.1.2	Router	9
3.2	Analýza OS	10
3.2.1	Skúsenosti v tíme.....	10
3.2.2	Potreba modifikácie základnej architektúry.....	11
3.2.3	Veľkosť systému po inštalácií	12
3.2.4	Minimálne požiadavky na systémovú pamäť	13
3.2.5	Životný cyklus – interval medzi novými verziami	14
3.2.6	Životný cyklus – dĺžka podpory stabilnej verzie	14
3.2.7	Správa balíkov	15
3.2.8	Podpora zariadení (ovládače)	17
3.2.9	Počet dostupných balíkov	18
3.2.10	Podpora Auto-Recovery.....	19
3.2.11	Serverové použitie	22
3.2.12	Podpora programovacích jazykov.....	24
3.2.13	Virtualizácia.....	24
3.2.14	Výsledky porovnania distribúcií	26
3.2.15	Finálny výber operačného systému.....	29
3.3	Multimediálny obsah.....	29
3.3.1	Porovnanie formátov podporovaných HTML5	29
3.3.2	Návrh riešenia.....	31
3.4	Zaznamenávanie udalosti v systéme (Logy)	31
3.4.1	Typy logov.....	31
3.4.2	Výber formátu pre posielanie logov na Cloud	32
3.4.3	Fyzický dátový model pre logy z web portálu.....	32
3.4.4	Fyzický dátový model pre systémové udalosti	34
3.5	Konzoly.....	34
3.5.1	Analýza súčasného stavu	34
3.5.2	Admin konzola	36
3.5.3	Požiadavky zákazníka.....	39
3.5.4	Definícia konzol	39
3.5.5	Spustenie konzol.....	40
3.6	Synchronizácia obsahu.....	41
3.6.1	Analýza transportných protokolov.....	42
3.6.2	Vyhodnotenie transportných protokolov	43
3.6.3	Proces synchronizácie obsahu	44

3.6.4	Porovnanie súborov s obsahom	44
3.6.5	Aktualizácia systému	44
3.6.6	Configuration Management system	44
3.7	Identifikácia	45
3.7.1	Analýza súčasného stavu	45
3.7.2	Zistenia o identifikácii z dokumentu „Setup of Funtoro HD system“	46
3.7.3	Aktuálne parametre identifikácie	48
3.7.4	Požiadavky zákazníka	48
3.7.5	Logický model Identifikácie	49
3.8	Transcoding	49
3.9	Webportál	51
3.9.1	Úvod	51
3.9.2	Analýza jazykov a frameworkov	51
3.9.3	Zhrnutie	54
3.10	Reklamy	54
3.11	Ukladací priestor	56
3.11.1	Požiadavky	56
3.11.2	Analýza súčasného riešenia na serveri	56
3.11.3	Analýza súborových systémov	59
3.11.4	Analýza a návrh mechanizmu na zálohovanie	65
3.12	Požiadavky na systém	75
3.12.1	Diagram prípadov použitia pre cestujúceho	75
3.12.2	Diagram prípadov použitia pre stewardku	82
3.12.3	Diagram prípadov použitia pre inštalátora(Molpir)	86
3.12.4	Diagram prípadov použitia pre správa flotily	90
4	Architektúra a návrh systému	93
4.1.1	Model komponentov	93
5	Implementácia	95
5.1	Backendové služby	95
5.1.1	Webový portál	95
5.1.2	Spúšťanie skriptov pri štarte systému	96
5.1.3	Modul na vzdialenú správu	97
5.1.4	Vytvorenie databázovej štruktúry (médiá)	115
5.1.5	Napĺňanie databáz	117
5.1.6	Rotácia záznamov	121
5.1.7	Údajové modely	122
5.2	Front-end	124
5.2.1	Navigácia	124
5.2.2	Prehliadanie filmov	125
5.2.3	Prehliadanie hudby	125
5.2.4	Stránkovanie	127
5.2.5	Prehliadanie obrázkov	127
5.2.6	Výber jazyka	128
6	Testovanie	129
6.1	Jednotkové testy	129

6.2	Testovanie nasadených aplikácií	131
6.2.1	Server bez zaťaženia	131
6.2.2	Server s pripojenými 5 zariadeniami	132
6.2.3	Server s pripojenými 10 zariadeniami	132
6.2.4	Server s pripojenými 20 zariadeniami	133
6.2.5	Server s prehrávaním 30 filmov súčasne	133
6.2.6	Zhodnotenie testov	134
	Bibliografické odkazy	135
7	Prílohy	i
7.1	Elektronické médium	i

1 Úvod

Táto dokumentácia slúži ako hlavný dokument k dielu, ktoré vzniká v rámci predmetu „Tímový projekt“. Autormi toho dokumentu sú členovia tímu číslo 17.

Projekt s názvom Future MOD sa realizuje v spolupráci so spoločnosťou Molpir s.r.o., ktorá v tomto projekte vystupuje ako zákazník. Spoločnosť Molpir patrí k popredným výrobcam doplnkov pre automobilový priemysel. Medzi zákazníkov tejto spoločnosti patria veľké prepravné spoločnosti ako: RegioJet (Student Agency), Turancar ... Spoločnosť MOLPIR pre svojich zákazníkov ponúka rôzne doplnky, mimo iné i individuálny multimedialny systém (MOD). Ide práve o systém, ktorý v poslednej dobe môžete vidieť vo vlakoch spoločnosti RegioJet a autobusoch Student Agency ako obrazovky, na ktorých máte možnosť pozerat' rôzny multimedialny obsah.

Úlohou projektu je vytvorit' systém, ktorý odstráni nedostatky proprietárneho systému MOD, ktorý prináša mnohé problémy pri rozšírení súčasného systému.

V kapitole 2 tohto dokumentu sú obsiahnuté ciele projektu pre zimný semester, ktoré sú rozpísané po jednotlivých šprintoch.

Podstatnú časť dokumentu zahŕňajú hĺbkové analýzy, ktoré prebiehali počas podstatnej časti prvého semestra. Táto časť je zdokumentovaná v kapitole 3.

Ďalšou z kapitol je Architektúra a návrh systému. V tejto časti sa venujeme architektúre a návrhu celého systému ako celku.

V kapitole číslo 5 je opísaná implementácia jednotlivých častí systému, ktoré sme sa rozhodli implementovať. Táto kapitola je logicky rozdelená na dve časti a to backend a front-end.

Poslednou časťou tejto dokumentácie je časť v ktorej sú opísané testy, ktoré sme vykonali k tomuto projektu.

2 Globálne ciele

2.1 Zimný semester

Na začiatku existencie každého tímu sa musí vyriešiť niekoľko organizačných záležitostí aby mohol tím napredovať správnym smerom. Patria sem napríklad výber spôsobu komunikácie v tíme, výber nástroja na manažment alebo rozdelenie úloh členov tímu. Náš tím bude riešiť tieto záležitosti hlavne počas úvodných stretnutí a prvého šprintu.

Keďže sa jedná o rozsiahly projekt, pri ktorom sa rieši od základu nový multimediálny systém, je potrebné podrobne špecifikovať a analyzovať všetky požiadavky na systém. Tento krok je časovo náročný ale zároveň nevyhnutný na vytvorenie správnej architektúry systému. Náš tím sa bude ním zaoberať prevažnú časť zimného semestra, výsledkom čoho by mala byť architektúra systému.

Z dôvodov rozsiahlej analýzy bude pravdepodobne obmedzený čas na implementáciu počas zimného semestra a teda predpokladáme iba implementovanie menšieho prototypu na zahodenie, ktorý bude predstavovať vybranú časť modulu systému.

Okrem prototypu bude výsledkom práce počas zimného semestra aj nastavenie všetkých vývojových prostredí a prepojenie so systémom na verziovanie, takže všetko bude pripravené na začiatok vývoja modulov systému v letnom semestri.

2.1.1 Ciele pre jednotlivé šprinty zimného semestra

Šprint 1:

Cieľom tohto úvodného šprintu je vyriešiť organizačné záležitosti v rámci tímu, vybrať nástroje na manažment a komunikáciu. Ďalej je potrebné vytvoriť webové sídlo tímu a začať so zbieraním požiadaviek na systém.

Šprint 2:

Tento šprint by mal riešiť podrobnú analýzu základných požiadaviek na systém. Na základe analýzy vybrať operačného systému pre multimediálny systém a zvolenie programovacích jazykov pre služby a web portál.

Šprint 3:

Pokračovanie v analýze ďalších požiadaviek na systém. V tomto šprinte analýzu zameriame na zachovanie jednotlivých vlastností, ktoré multimediálny systém v súčasnosti ponúka a na nové, o ktoré bude treba systém rozšíriť.

Šprint 4:

V čase tohto šprintu by mali byť analyzované všetky základné požiadavky, z ktorých by mal vzniknúť prvý návrh architektúry systému. Nadalej bude prebiehať analýza zvyšných požiadaviek. Tento šprint by mal byť zároveň posledným v zimnom semestri a predpokladá sa aj vytvorenie prototypu na zohodenie.

2.2 Letný semester

Výsledkom zimného semestra nášho tímového projektu bola ukončená analýza multimediálneho systému na základe požiadaviek zákazníka a vypracovanie architektúry systému. Spolu s týmito analytickými časťami sme počas zimného semestra pripravili všetky potrebné vývojové a testovacie prostredia tak, aby bolo možné v letnom semestri naplno začať s implementáciou navrhnutého riešenia.

Keďže multimediálny systém podľa požiadaviek zákazníka má obsahovať ďaleko viac funkcionality ako by sme boli schopní zvládnuť implementovať za letný semester, rozhodli sme sa vybrať a zamerať sa na najdôležitejšie časti tohto systému. Medzi tieto časti patria:

- modul prehrávania filmov s prerušením prehrávania reklamou,
- modul prehrávania hudby,
- modul zobrazovania obrázkov,
- modul logovania udalostí,
- modul štatistík,
- modul synchronizácie štatistík a logov s cloudom,
- modul vzdialeného prístupu.

Hlavným cieľom letného semestra bude teda implementovať vyššie uvedené moduly, prípadne ďalšie časti systému ak ostane čas.

2.2.1 Ciele pre jednotlivé šprinty letného semestra

Šprint 5:

Cieľom tohto prvého šprintu v letnom semestri bude začať s implementáciou jednotlivých modulov. V tomto šprinte pripravíme jednotlivé šablóny stránok web portálu a začneme s implementáciou modulu prehrávania videa, ktorému sme dali najvyššiu prioritu. Popri týchto úlohách sa bude pracovať aj na serverových moduloch, kde začneme so zbieraním logov.

Šprint 6:

V tomto šprinte budeme prioritne riešiť ďalší modul a to prehrávanie hudby. Popritom sa bude riešiť serverový modul vzdialeného prístupu. Počas tohto šprintu tiež zozbierame pripomienky od zákazníka, na časti implementované v predchádzajúcom šprinte.

Šprint 7:

Tento šprint by sa ako aj predchádzajúce mal zaoberať ďalším modulom, tentokrát modulom zobrazovania obrázkov. Popritom bude riešený modul štatistík a zapracovanie pripomienok k Šprintu 5. Rovnako zozbierame pripomienky k predchádzajúcemu šprintu.

Šprint 8:

Cieľom tohto šprintu bude zapracovať všetky doteraz zozbierané pripomienky na predošlé šprinty a pridanie reklám do filmov. Popritom sa bude riešiť modul synchronizácie štatistík a logov s cloudom.

Šprint 9:

V čase nášho posledného šprintu bude prebiehať študentská vedecká konferencia IITSRC a preto sa aj počas šprintu zameriame na opravenie chýb modulov, ktoré budeme prezentovať. Počas tohto šprintu pravdepodobne už nebudeme dopĺňať žiadnu novú funkcionality a budeme sa predovšetkým snažiť doladiť všetky nedostatky implementovaných modulov.

3 Analýza

V prvej časti tejto kapitoly je obsiahnutý výsledok rozsiahlej analýzy spracovanej na požiadavku zákazníka.

3.1 Mobilný internet

Budúci systém by mal byť prioritne nasadzovaný v dopravných prostriedkoch. Z čoho vyplýva, že systém bude stále v pohybe. Tu vzniká niekoľko problémov so stabilným pripojením do mobilnej siete.

Prvým problémom je použitie rôznych štandardov v mobilných sieťach, ktoré sa odlišujú hlavne prenosovou rýchlosťou. Druhým problémom je rôzny rozsah pokrytia územia mobilnou sieťou s použitím rôznych štandardov. Na niektorých územiach dokonca neexistuje pokrytie žiadnou mobilnou sieťou. Tretím problémom je, že aj keď veľká časť územia je pokrytá nejakou mobilnou sieťou, tak dosahované rýchlosti sú veľmi malé.

Z vyššie uvedených problémov vyplýva, že je potrebné regulovať prenos dát smerujúcich von z vozidla napr. pomocou QoS. Tok dát možno rozdeliť na dve skupiny. Prvá skupina je komunikácia medzi vozidlom a centrálnym serverom. Druhú skupinu tvoria pasažieri vo vozidle prístupujúci na internet.

3.1.1 Pokrytie a rýchlosti v mobilnej sieti

V tabuľke 3 je prehľad o používaných štandardoch v mobilných sieťach a teoretický poskytovaných rýchlostiach.

Tabuľka 2 Teoretické rýchlosti štandardov v mobilných sieťach

Symbol	Štandard	Max. rýchlosť sťahovania	Max. rýchlosť zdieľania
2G	GSM	14.4 Kbit/s	14.4 Kbit/s
G	GPRS	53.6 Kbit/s	26.8 Kbit/s
E	EDGE	217.6 Kbit/s	108.8 Kbit/s
3G	UMTS	384 Kbit/s	128 Kbit/s
H	HSPA	7.2 Mbit/s	3.6 Mbit/s
H+	HSPA+	14.4 – 168.8 Mbit/s	5.76 – 23.0 Mbit/s
4G	LTE	100 Mbit/s	50 Mbit/s

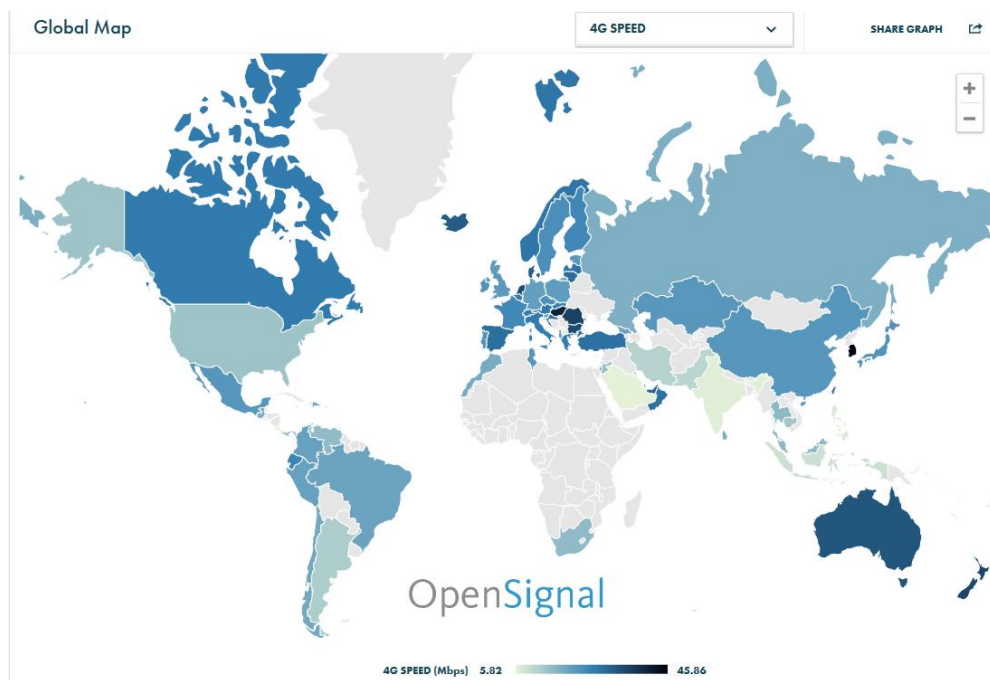
4G	LTE-A	1 Gbit/s	500 Mbit/s
----	-------	----------	------------

Štandard HSPA+ má jedenásť vydaní, ktoré sa líšia okrem iného aj rýchlosťou. Zadaný rozsah je ohraničený rýchlosťou prvého a posledného vydania HSPA+. Nasledujúcej tabuľke 4 sú uvedené rýchlosti od mobilného operátora v SR.

Tabuľka 3 Rýchlosti poskytované mobilným operátorom

Štandard	Rýchlosť sťahovania	Rýchlosť zdieľania	Čas odozvy
LTE	150.0 Mbit/s	35.0 Mbit/s	27 – 50 ms
F-OFDM	5.3 Mbit/s	1.8 Mbit/s	20 – 50 ms
HSPA+	21.8 Mbit/s	1.46 Mbit/s	150 – 350 ms
HSPDA	7.2 Mbit/s	1.46 Mbit/s	150 – 350 ms
UMTS	384.0 Kbit/s	64 Kbit/s	400 – 350 ms
EDGE	238.8 Kbit/s	118.2 Kbit/s	400 – 700 ms
GPRS	57.6 Kbit/s	28.8 Kbit/s	400 – 700 ms

4G sieť dnes poskytuje najlepšie prenosové rýchlosti. Tiež je to sieť do ktorej budovania najviac aktuálne investujú poskytovatelia mobilného internetu. Na obrázku 5 sú zobrazené rýchlosti 4G siete v rôznych štátoch.



Obrázok 1 Rýchlosti 4G siete podľa krajín

3.1.2 Router

Aktuálne používaný typ routera je 3G012 Router mobilný, s integrovanou WiFi, 2x LAN/1xWAN. Pre náš projekt má router význam z pohľadu možnosti nastavenia QoS, demilitarizovanej zóny a zabezpečenej komunikácie. V tabuľke 5 sa nachádza vybraná špecifikácia routera.

Tabuľka 4 Hardvérové a softvérové vlastnosti routera

Parameter	Hodnota
Internetové pripojenie	3G dial-up DHCP/ staické IP/ PPPoE
Počet pripojených klientov	Cez kábel: 253; bezdrôtovo: 30
Správa bezpečnosti	Nastavenie Firewall: blokovanie útoku do LAN; Kontrola P-to-P: Blokovanie určitých web stránok; Mac filtrovanie: blokovanie určitých Mac adries; Kontrola prístupu: limitovaný LAN na prístup na internet; Blokovanie portov: ochrana proti Dos
QoS	Uplink širokého pásma správy nastavení; Downlink širokého pásma správy nastavení
Systémová služba	Virtuálny server: interný virtuálny server; DMZ: nastavenie DMD ako host počas virtuálneho servera, nie je nastavené, aby bol otvorený; Spúšťač portu: automatický prístup na LAN; Sériový port data passthrough a AT príkaz
Režim WLAN bezpečnosti	Otvorený systém; WPA-PSK; WPA2-PSK; WPAPSK; WPA2PSK (mixované s WPA-PSK s WPA2-PSK); WPA1WPA2 (mixované WPA s WPA2
Bezdrôtové rozhranie	IEEE802,11b/g/n
WiFi prenosová rýchlosť	150 Mbit/s
Dátová prenosová rýchlosť	802.11n: až do 150Mb/s 802.11b: 1; 2; 5,5; 11Mb/s 802.11g: 6; 9; 12; 18; 24; 36; 48; 54Mb/s
Výkon prenosu	11n HT40 MCS7: + 13,5dBm; 11b CCK: +18dBm; 11g OFDM: +13,5dBm
Citlivosť prijímača	-66dBm pri 15054Mp/s -73dBm pri 54Mp/s -86dBm pri 11Mp/s

Nevýhodou tohto routra je možnosť konfigurácie iba cez obmedzené grafické prostredie. Ďalším mínusom je slabšia možnosť nastavenia QoS, čo môže byť pre prenos dôležitých údajov kľúčové, vzhľadom nato v akom prostredí sa systém bude používať. Tiež nevýhodou je slabá podpora privátnych sietí a žiadna podpora IPSec protokolu na zabezpečenie komunikácie s centrálnym serverom. Negatívom je aj podpora iba 3G siete. Alternatívu k tomuto routru je router WLINK R200 4G. Poskytuje všetky funkcie, ktoré boli uvedené ako nevýhody pri routri 3G012. Navyše už podporuje aj 4G sieť. Tiež je vhodný na použitie v priemysle a vo vozidlách.

Alternatívne riešenie QoS je možné priamo na servery pomocou nástroja tc. Tc sa používa na riadenie trafiky v kerneli Linuxu. Podporuje nasledujúce vlastnosti:

- Shaping – kontrola rýchlosti prenosu na výstupe
- Scheduling – plánovanie paketov na výstupe
- Policing – kontrola rýchlosti prenosu na vstupe
- Dropping – ak sa prekročí nastavená šírka pásma, tak sa pakety zahodia

Proces kontroly trafiky je vykonávaný pomocou radov, tried a filtrov.

3.2 Analýza OS

Porovnávaných je 10 distribúcií, z toho 9 na báze Linux jadra a 1 na báze Unix jadra. Vždy sú analyzované (ak to analýza umožňuje) najnovšie verzie distribúcií (k dátumu 26/10/2016), ktoré sú dostupné na oficiálnych stránkach. Výber týchto operačných systémov je založený na popularite a odporúčaníach na rôznych webových fórach a diskusných článkoch [1] [2].

- Ubuntu, FreeBSD, Debian, Arch Linux, Fedora, Puppy, CentOS, Tiny Core, Trisquel Mini, Linux Lite

3.2.1 Skúsenosti v tíme

Poznámky:

- Je vysoký predpoklad, že v systéme, ktorý poznáme, dosiahneme vyššiu efektivitu práce ako v systémoch, s ktorými by sme pracovali prvýkrát.
- Jeden člen tímu mohol hlasovať aj viackrát za viaceré distribúcie, s ktorými má nejaké skúsenosti.

Mierka pre vyhodnocovaciu maticu:

- Hodnota 10 = 7 členov v tíme, hodnota 0 = žiadny člen v tíme nemá skúsenosti s distribúciou.
- $X = (10/7) * \text{„počet členov v tíme so skúsenosťami“}$.

Porovnanie operačných systémov z hľadiska skúseností s konkrétnou distribúciou v tíme zobrazuje Tabuľka 5.

Tabuľka 5. Distribúcia skúseností s distribúciami v tíme

Distribúcia	Členovia tímu
Lubuntu	Radoslav Zapach, Tomáš Baránek, Matúš Slovák, Lukáš Mastiľak, Jaroslav Tóth
FreeBSD	Jaroslav Tóth
Debian	Jaroslav Tóth
Arch Linux	-
Fedora	-
Puppy	-
CentOS	Matúš Slovák
Tiny Core	-
Trisquel Mini	-
Linux Lite	-

3.2.2 Potreba modifikácie základnej architektúry

Poznámky:

- Zakladá sa na ciele dosiahnuť veľkosť operačného systému pod hodnotu 1GB a odstrániť GUI/vypnúť GUI, ktorý takisto zbytočne zaberá na disku priestor a vytvára processing.
- Redukovať veľkosť operačného systému je možné odstránením balíkov, ktoré vieme, že nebudeme používať. Ďalšími trikmi sú:
 - Odstránenie nepotrebných lokalizácií (~90 MB).
 - Odstránenie doc-ov a manuálových stránok (~100MB).
- Pravdaže, ďalšie redukovanie veľkosti systému je možné aj pri systémoch, ktorých veľkosť neprekračuje 1 GB a je to aj odporúčané.
- Pri distribúciách Puppy, Trisquel Mini a Linux Lite sme zatiaľ neprišli na to, ako odstrániť GUI (vieme ho iba vypnúť).

Mierka pre vyhodnocovaciu maticu:

- Nie je potrebná žiadna modifikácia systému – 10.
- Potrebné iba odstránenie/vypnutie GUI (aj keď veľkosť systému je pod hranicou 1GB) – 8.
- Potrebná modifikácia pred/po inštalácií zmazaním/pridaním balíčkov – 5.
- Systém nepodporuje modifikáciu zoznamu balíčkov po inštalácií – 1 (nepoužitá možnosť).

Porovnanie operačných systémov (najnovšie **stabilné** distribúcie operačných systémov pre i386) z hľadiska jednoduchosti rekonštrukcie architektúry na „lightweight“ zobrazuje Tabuľka 6.

Tabuľka 6. Porovnanie "lightweight" možností medzi distribúciami

Distribúcia	Opis prechodu na lightweight OS
Lubuntu	Napriek tomu, že výsledná inštalácia základného systému zaberá na disku priestor pod 1GB, inštalácia vyžaduje mať k dispozícii aspoň 5 GB disk – je potrebné modifikovať inštaláčne súbory/balíčky.
FreeBSD	Potrebná modifikácia pred/po inštalácií zmažaním/pridaním balíčkov.
Debian	Nie je potrebná žiadna modifikácia systému pri základnej inštalácií.
Arch Linux	Potrebná modifikácia pred inštaláciou nastavením balíčkov a manuálna inštalácia systému (8 bodov je najobjektívnejší kompromis za manuálnu inštaláciu).
Fedora	Potrebná modifikácia pred/po inštalácií zmažaním ďalších balíčkov.
Puppy	Potrebná modifikácia pred/po inštalácií – odstránenie/vypnutie GUI.
CentOS	Potrebná modifikácia pred/po inštalácií zmažaním/pridaním balíčkov.
Tiny Core	Nie je potrebná žiadna modifikácia systému.
Trisquel Mini	Potrebná modifikácia pred/po inštalácií – odstránenie/vypnutie GUI.
Linux Lite	Potrebná modifikácia pred/po inštalácií – odstránenie/vypnutie GUI.

3.2.3 Veľkosť systému po inštalácií

Poznámky:

- Veľkosť systému iba v CLI móde bez grafických rozhraní (odstránené alebo nenainštalované).
- Ďalšie balíky, ktoré nesúvisia s GUI, nie sú odstránené.

Mierka pre vyhodnocovaciu maticu:

- Rozsah 1 – 10, kde 10 → 0MB a 0 → 2000 MB.
- $X = (-1/200) * SIZE + 10$.

Porovnanie operačných systémov (najnovšie **stabilné** distribúcie operačných systémov pre i386) z hľadiska finálnej veľkosti po inštalácií zobrazuje Tabuľka 7.

Tabuľka 7. Porovnanie veľkostí vybraných distribúcií

Distribúcia	Veľkosť po inštalácií
Lubuntu	650 – 700 MB (priemer 675 MB)
FreeBSD	~ 1500 MB
Debian	650 – 700 MB (priemer 675 MB)
Arch Linux	~ 700 MB

Distribúcia	Veľkosť po inštalácii
Fedora	~ 1100 MB
Puppy	150 – 200 MB (non-live verzia) (priemer 175 MB)
CentOS	700 – 1500 MB (priemer 1100 MB)
Tiny Core	~ 100 MB
Trisquel Mini	500 – 600 MB (priemer 550 MB)
Linux Lite	~ 940 MB

3.2.4 Minimálne požiadavky na systémovú pamäť

Poznámky:

- Hľadali sa informácie o lightweight verziách bez GUI.
- Ako zdroj informácií sa použili hodnoty udané na oficiálnych stránkach distribúcií (minimálne požiadavky). Tieto hodnoty sú ale väčšinou iba približné a skutočný stav obsadenej pamäte je väčšinou aspoň dvojnásobne menší ako udávaná hodnota (spustený iba operačný systém bez ďalších aplikácií).

Mierka pre vyhodnocovaciu maticu:

- Hodnota 0 = 512 MB RAM, hodnota 10 = 32 MB RAM.
- $X = (-1/48) * RAM + 32/3$.

Porovnanie operačných systémov (najnovšie **stabilné** distribúcie operačných systémov pre i386) z hľadiska minimálnych požiadaviek na systémovú pamäť zobrazuje Tabuľka 8.

Tabuľka 8. Porovnanie distribúcií vzhľadom na minimálne požiadavky na systémovú pamäť

Distribúcia	Minimálna veľkosť systémovej pamäte
Lubuntu	128 MB
FreeBSD	96 MB
Debian	256 MB
Arch Linux	256 MB
Fedora	380 MB
Puppy	64 MB
CentOS	392 MB
Tiny Core	46 MB
Trisquel Mini	128 MB
Linux Lite	256 MB

3.2.5 Životný cyklus – interval medzi novými verziami

Poznámky:

- Je ideálne, keď doba medzi jednotlivými vydania distribúcie sú čo najviac konštantné – môžeme sa pripraviť na ďalšie vydanie – aktualizácie, plánovanie aktualizácií.
- Smerodajná odchýlka sa vypočítala z najmenej posledných šiestich vydaní (vyšší počet nebol možný pri niektorých distribúciách, pretože taký počet vydaní ešte nevznikol).
- Počíta sa smerodajná odchýlka s dĺžok v dňoch medzi vydania (menšia je lepšia).

Mierka pre vyhodnocovacu maticu:

- Hodnota 10 = 0 smerodajná odchýlka, hodnota 0 = 600 smerodajná odchýlka.
- $X = 10 - (1/60) * \text{„smerodajná odchýlka“}$.

Porovnanie operačných systémov vzhľadom na smerodajnú odchýlku intervalu medzi vydania distribúcií zobrazuje Tabuľka 9.

Tabuľka 9. Smerodajná odchýlka intervalu medzi vydania distribúcií

Distribúcia	Smerodajná odchýlka intervalu medzi vydania
Lubuntu	10.053
FreeBSD	80.695
Debian	135.181
Arch Linux	1.095
Fedora	50.214
Puppy	492.324
CentOS	506.198
Tiny Core	40.404
Trisquel Mini	166.815
Linux Lite	23.475

3.2.6 Životný cyklus – dĺžka podpory stabilnej verzie

Poznámky:

- Dĺžka podpory stabilnej verzie operačného systému je dôležitá, pretože po tejto dobe sa prestanú do systému dodávať nové aktualizácie, ktoré slúžia na opravu chýb.

Mierka pre vyhodnocovacu maticu:

- Nešpecifikovaná dĺžka podpory → hodnota 0.

- „Rolling“ distribúcia → hodnota 5 (kontinuálne aktualizácie cez jednu vývojovú vetvu zaistia neobmedzenú podporu, na druhej strane je ťažšie plánovať aktualizácie).
- Hodnota 10 = 4000 dní, hodnota 0 = 0 dní.
- $X = (1/400) * \text{„dĺžka podpory“}$.

Porovnanie operačných systémov vzhľadom na priemernú dĺžku podpory stabilnej verzie zobrazuje Tabuľka 10.

Tabuľka 10. Priemernú dĺžku podpory stabilnej verzie vybraných distribúcií

Distribúcia	Priemerná dĺžka podpory stabilnej verzie (počet dní) (LTS)
Lubuntu	705.16
FreeBSD	633.333
Debian	1399
Arch Linux	“rolling” distribúcia
Fedora	413.091
Puppy	nešpecifikované
CentOS	3672.00
Tiny Core	nešpecifikované
Trisquel Mini	1082.111
Linux Lite	1155

3.2.7 Správa balíkov

Poznámky:

- Porovnanie viacerých správčov balíkov pomocou nasledovných zdrojov (najmä doplnenie prvého zdroja na základe podobných funkcionalít rôznych správčov balíčkov) - [3], [4], [5].
- Výsledky porovnania zobrazuje Tabuľka 11.

Tabuľka 11. Porovnanie najpoužívanejších správčov balíkov

Package Manager	Binary and Source Install	Customability	Install from URL	Easy of Develop	Shortened Command
APT	No	2.5	No	2.5	No
DNF	No	5	Yes	5.5	No

Package Manager	Binary and Source Install	Customability	Install from URL	Easy of Develop	Shortened Command
DEB	Yes	8	No	5	Yes
pacman	Yes	8	No	9	No
pkgng (ports)	Yes	10	No	7.5	Yes
TCE	No	1	Yes	1	No
PET	No	1	No	0	No
dpkg	Yes	2	Yes	1	No
RPM	Yes	5	Yes	4	No

Mierka pre vyhodnocovacia maticu:

- Spočítanie hodnôt v riadku, pričom “Yes” = 1, “No” = 0.
- Naškálovanie hodnôt – Hodnota 0 – súčet 0, hodnota 10 – najvyšší súčet zaokrúhlený na desiatku nahor (v tomto prípade to vychádza na 20).

Porovnanie operačných systémov z hľadiska použitých správčov balíkov zobrazuje Tabuľka 12.

Tabuľka 12. Použitie správčov balíkov vo vybraných distribúciách

Distribúcia	Manažér balíkov
Lubuntu	APT (Advanced Packaging Tool)
FreeBSD	pkgng (Next Generation Package Management)
Debian	APT (Advanced Packaging Tool)
Arch Linux	Pacman
Fedora	DNF (Dandified Yum)
Puppy	PET (Pet)
CentOS	RPM (Red Hat Package Manager)
Tiny Core	TCE (Tiny Core Extension)
Trisquel Mini	dpkg (Debian Package Management System)
Linux Lite	dpkg (Debian Package Management System)

3.2.8 Podpora zariadení (ovládače)

Poznámky:

- Analýza sa zameriava na hardware, ktorý bol identifikovaný pri analýze súčasného stavu.
- Pre úložné zariadenia v Linux/Unix nie sú potrebné ovládače, iba podpora súborového systému.
- Pri operačnom systéme FreeBSD sa automaticky predpokladá úplná podpora HW (v prípade, že v novších verziách OS nebudú niektoré zariadenia pracovať správne, je možné z aktuálnej verzie stiahnuť ovládače).
- **Intel 945GSE Chipset (Memory Controller Hub, Integrated Graphics)**
 - Zaručená podpora pre: Linux Fedora 7, Wind River Linux, Red Hat Linux, Ubuntu 8.04 alebo vyššie verzie [6].
 - Základné ovládače sa nachádzajú v Linux Kernel (od 3.4.112). S grafickým adaptérom môžu byť problémy pri iných OS.
- **Intel NM10/ICH7 (Family High Definition Audio, Family PCI Express Port, USB UHCI Controller, Family SMBus Controller)**
 - Základné ovládače sa nachádzajú v Linux Kernel (od 3.4.112).
 - So zvukovým ovládačom môžu nastať problémy - riešenie som našiel pre Debian, Ubuntu, ArchLinux, TinyCore, CentOS a Linux Lite.
 - Riešenie, ktoré s vysokou pravdepodobnosťou bude funkčné aj pri iných distribúciách - [7].
- **Intel 82801(GBM/GHM) (Mobile PCI Bridge, LPC Interface Bridge, IDE Controller, SATA Controller, SCSI Controller)**
 - Priama podpora v Linux Kernel (rôzne verzie) [8].
- **Intel 82574L (Gigabit Network Adapter)**
 - Priama podpora v Linux Kernel aj možnosť stiahnutia ovládačov manuálne, ak by sa vyskytol nejaký problém [9].

Mierka pre vyhodnocovaciu maticu:

- Za každý potenciálny výskyt problému sa odpočítavajú z hodnoty 10 dva body.

Porovnanie operačných systémov z hľadiska podpory potrebných ovládačov hardvérových komponentov zobrazuje Tabuľka 13.

Tabuľka 13. Porovnanie podpory hardvérových komponentov vo vybraných distribúciách

Distribúcia	Opis podpory (problémy)
Lubuntu	-
FreeBSD	-
Debian	Potenciálne problémy s grafickým adaptérom.
Arch Linux	Potenciálne problémy s grafickým adaptérom.
Fedora	Potenciálne problémy so zvukovým ovládačom.

Distribúcia	Opis podpory (problémy)
Puppy	Potenciálne problémy s grafickým adaptérom. Potenciálne problémy so zvukovým ovládačom.
CentOS	-
Tiny Core	Potenciálne problémy s grafickým adaptérom.
Trisquel Mini	Potenciálne problémy s grafickým adaptérom. Potenciálne problémy so zvukovým ovládačom.
Linux Lite	Potenciálne problémy s grafickým adaptérom.

3.2.9 Počet dostupných balíkov

Poznámky:

- Informácie o počte oficiálne podporovaných balíkov, ktorými je možné OS rozšíriť - [10], [11].
- Informácie môžu byť mierne odlišné od skutočného stavu (nepredpokladá sa radikálna zmena hodnôt, ktorá by ovplyvnila výsledky) – oficiálne stránky distribúcií väčšinou neposkytujú štatistické informácie o balíkoch.

Mierka pre vyhodnocovaciú maticu:

- Hodnota 10 = 60000, hodnota 0 = 0 balíkov.
- $X = (1/6000) * \text{“počet balíkov“}$.

Porovnanie operačných systémov z hľadiska počtu dostupných balíkov, ktoré umožňujú rozširovať systém zobrazuje Tabuľka 14.

Tabuľka 14. Porovnanie počtu dostupných balíkov v distribúciách

Distribúcia	Približný počet balíkov rozširujúcich systém
Lubuntu	58 318
FreeBSD	25 468
Debian	56 864
Arch Linux	8 892
Fedora	22 274
Puppy	611
CentOS	17 565
Tiny Core	3543
Trisquel Mini	52 748

Distribúcia	Približný počet balíkov rozširujúcich systém
Linux Lite	1 767

3.2.10 Podpora Auto-Recovery

Úvod do analýzy

- Operačný systém môže ovládať procesy – automatizované spúšťanie, reštartovanie, ukončovanie procesov a pridelovanie prostriedkov pre procesy. Auto-Recovery funkcionálna znamená, že operačný systém alebo iný modul môže detegovať nesprávne ukončenie procesu a pokúsiť sa ho opätovne naštartovať (napr. aplikácia nevrátila pri ukončení hodnotu 0).
- Operačný systém nezodpovedá za integritu dát, s ktorými proces pracoval pri tom, ako zlyhal – za tento aspekt je stále zodpovedná aplikačná vrstva – jadro systému alebo iný auto-recovery modul, sa ale môže postarať o to, aby sa tento proces pri nasledovnom spustení načítal v núdzovom režime (ak je taký režim k dispozícii, napr. pri databázových systémoch je vždy k dispozícii).

Realizácia Auto-recovery v Linux/Unix

1. **Init systémy** – Všetky operačné systémy s jadrom Linux obsahujú zavádzajúci init súbor, ktorý sa stará o automatické spúšťanie a nastavenie procesov pri štarte systému. Existujú 3 základné typy init systémov – **SysVinit, Upstart a SystemD**. Každý z týchto init systémov podporuje „respawn“ funkcionálnu, pri ktorej init systém kontinuálne monitoruje stav procesu a pri chybnom ukončení sa ho pokúsi okamžite reštartovať (systémový / softvérový watch-dog). Je možné zadať trvanie, počas ktorého sa init bude pokúšať nabudieť proces, po tomto čase init môže vykonať reset jadra. Samozrejmosťou je automatické spúšťanie procesov pri štarte OS. Init beží ako oddelený proces, vždy s PID 1; logovanie je riešené vlastnými logovacími procesmi (nie syslog) – napr. journal v SystemD. Príklad (SystemD implementácia; automatické obnovenie MySQL systému po poruche) [12]:

```

sudo nano /etc/systemd/system/multi-
user.target.wants/mysql.service
[Unit]
...
[Install]
...
[Service]
...
Restart=always
...
sudo systemctl status mysql.service
mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service;
   enabled)
   Active: active (running) since Fri 2015-07-31 21:58:03
   EDT; 1h 7min ago
   Main PID: 661 (mysqld safe)

```

```

...
sudo kill -9 661
sudo systemctl status mysqld.service
mysqld.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service;
   enabled)
   Active: active (running) since Fri 2015-07-31 23:06:38
   EDT; 1min 8s ago
     Process: 11218 ExecStartPost=/usr/bin/mysql-systemd-start
   post (code=exited, status=0/SUCCESS)
     Process: 11207 ExecStartPre=/usr/bin/mysql-systemd-start
   pre (code=exited, status=0/SUCCESS)
   Main PID: 11217 (mysqld safe)

```

2. **Využitie cron systému** – Pomocou nástroja cron je možné naplánovať periodické spúšťanie vytvorenej úlohy (môže byť implementovaná vo forme skriptu alebo procedúry v programovacom jazyku). Takto vytvorená úloha by kontrolovala stav procesu a v prípade zistenia, že proces nebeží alebo sa ukončil nesprávne, môže tento proces opätovne spustiť alebo reštartovať. Výhodou je, že v skripte môžeme zadefinovať aj ďalšie kontrolné podmienky, pomocou ktorých sa vyhodnocuje stav procesu (init systém je obmedzený jeho implementáciou) a zadefinovať aj ďalšie cesty, akým spôsobom sa vysporiadať s danou situáciou (napr. rekonfigurácia softvéru). Ďalšou výhodou je, že činnosť cron systém je možné takisto automaticky logovať do súborov pomocou syslog / rsyslog / ngsyslog alebo inej syslog implementácie. Nevýhodou je zložitejšia implementácia a non-real-time spracovanie.

Príklad – bash skript (založený na testovaní, či sa vybraný process nachádza v procesov):

```

#!/bin/bash
if [ ! "$(pidof application)" ]
then
  /opt/path/application &
fi

```

3. Špecializované softvérové balíky

- a. **Monit** – Lightweight Open Source nástroj na správu a monitorovanie Unix/Linux operačných systémov – automatická údržba systémových modulov. Funkcie: proaktívna reakcia na udalosti v systéme (reštartovanie procesov, logovanie udalostí), monitorovanie procesov, monitorovanie súborového systému, monitorovanie a testovanie siete, testovanie programov a skriptov, monitorovanie systémových prostriedkov. Podpora: aix, freebsd, openbsd, linux (x86, x64, arm, macosx, solaris).
- b. **Daemontools** – Kolekcia nástrojov na správu procesov v UNIX systémoch. Funkcie: spúšťanie a reštartovanie procesov (centralizované), logovanie chybových udalostí, rotácia záznamov.
- c. **Supervisor** – Klient/server systém, ktorý umožňuje kontrolovať z jedného bodu viacero procesov – spúšťanie, reštartovanie, monitorovanie. Hlavnou výhodou je možnosť hromadného ovládania procesov, spúšťanie procesov

ako child procesov (jednoduchšie riadenie) a možnosť riadenia prístupu.
Podpora: Linux, Mac OS X, Solaris, a FreeBSD.

- d. **Daemonize** – Nástroj zameraný na správu prerušení, I/O signálov a background procesov za účelom ladenia procesov. Podpora: FreeBSD, Fedora, Solaris, Red Hat / CentOS, Ubuntu, Mac OS.

4. **Kontinuálna kontrola oddelenými procesmi** – Pre každý proces, pri ktorom chceme zabezpečiť kontinuálne vykonávanie, je vytvorený ďalší oddelený proces, ktorý v cykle kontroluje stav procesu a na základe analýzy sa rozhodne, či je potrebné reštartovať proces alebo napr. celý systém. Výhodou je možnosť zdefinovania komplexných pravidiel a real-time implementácia. Nevýhodou je decentralizované riešenie bez podporného logovania udalostí (ako je to pri cron a init). V Unix systémoch je to možné riešenie namiesto chýbajúceho Init.

Porovnanie init systémov (používaných v analyzovaných operačných systémoch) zobrazuje Tabuľka 15.

Tabuľka 15. Porovnanie init systémov

Funkcia	Upstart	SysVinit	SystemD	Busybox	OpenRC
Distribovaná konfigurácia procesov	ÁNO	ÁNO	ÁNO	NIE	NIE
Init je procesom	NIE	ÁNO	ÁNO	ÁNO	ÁNO
Tvorba zložitých udalostí a prepojení	ÁNO	NIE	ÁNO	NIE	NIE
Paralelné spúšťanie procesov / služieb	ÁNO	NIE	ÁNO	NIE	ÁNO
Automatická obnova procesov	ÁNO	ÁNO	ÁNO	ÁNO	ÁNO

Mierka pre vyhodnocovaciu maticu:

- Berie sa do úvahy primárny (prvý) init systém.
- Hodnota 5 = všetky init funkcie, hodnota 0 = žiadna init funkcia.
- Hodnota 4 = podpora SW priamo z repozitárov, hodnota 2 = potrebná kompilácia SW zo zdrojového kódu (znížená podpora), hodnota 0 = nie je dostupný žiadny podporný SW.
- Hodnota 1 = podpora viacerých init, hodnota 0 = podpora jedného/žiadneho init.
- X = „počet init funkcií“ + „hodnotenie podpory SW“.

Porovnanie operačných systémov vzhľadom na Auto-Recovery podporu zobrazuje Tabuľka 16.

Tabuľka 16. Porovnanie distribúcií z hľadiska podpory Auto-Recovery

Distribúcia	Podporovaný Auto-Recovery systém
Lubuntu	Init: SystemD, Upstart Software: podpora priamo z repozitárov
FreeBSD	Init: FreeBSD nepoužíva init, ale RC skripty Software: podpora priamo z portov
Debian	Init: SystemD, SysVinit, Upstart, OpenRC Software: podpora priamo z repozitárov
Arch Linux	Init: SystemD, Upstart, initng, Epoch, finit Software: podpora priamo z repozitárov
Fedora	Init: SystemD Software: podpora priamo z repozitárov
Puppy	Init: Busybox init Software: potrebná manuálna kompilácia programov
CentOS	Init: SystemD Software: podpora priamo z repozitárov
Tiny Core	Init: Busybox init Software: potrebná manuálna kompilácia programov
Trisquel Mini	Init: SysVinit, Upstart Software: potrebná manuálna kompilácia programov
Linux Lite	Init: SystemD, SysVinit Software: potrebná manuálna kompilácia programov

3.2.11 Serverové použitie

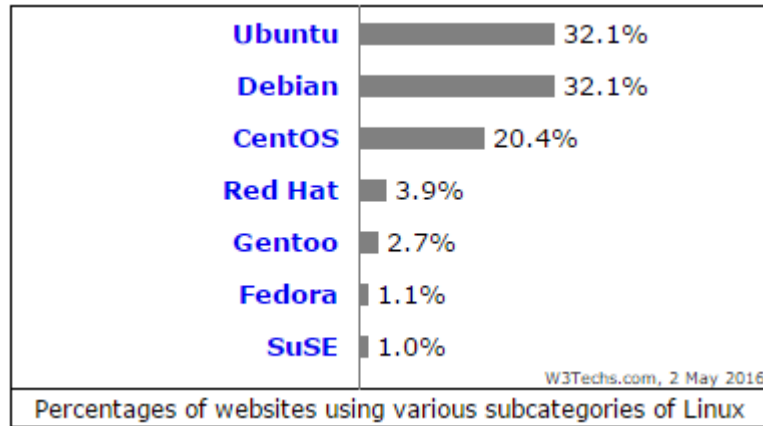
Úvod k analýze

- Na porovnanie použitia distribúcií v dopravnom alebo inom príbuznom priemysle neboli zatiaľ pozbierané štatistické údaje, ktoré by mohli byť použité.
- Používané sú väčšinou proprietárne systémy na báze Linux/Unix, ich bližšia špecifikácia však nie je opísaná (distribúcia, verzia systému ...).
- Existujú rôzne internetové zdroje a články, ktoré sa venujú tejto problematike. Väčšinou sú orientované na vývoj novej embedded Linux distribúcie – Automotive Grade Linux [13]. Tento operačný systém je však ešte vo fáze Demo verzie. Ďalším operačným systémom je Tizen. Tieto distribúcie sú ale viac orientované na používateľské rozhranie ako na serverové rozhranie.

- Z uvedených dôvodov sa táto časť analýzy zaoberá porovnaním použitia operačných systémov na serverové účely (na takéto porovnanie už existuje viac zdrojov informácií).

Získané štatistické informácie s uvedenými zdrojmi (vybrané sú iba tie distribúcie so zoznamov, ktoré boli analyzované v dokumente)

- Obrázok 2, Tabuľka 17, Tabuľka 18.



Obrázok 2. Porovnanie použitia zvolených distribúcií vo webových serveroch [14]

Tabuľka 17. Porovnanie celkového podielu používania distribúcií v serverových aplikáciách – administrátori [15]

Distribution	Responses	Percentage
Ubuntu	633	34%
Debian	539	29%
CentOS	214	12%
Arch Linux	183	10%
Fedora	34	2%

Tabuľka 18. Porovnanie celkového podielu používania distribúcií v serverových aplikáciách – bežní používatelia [16]

Distribution	Non-desktop Users	Percentage
Debian	71	29.78%
Ubuntu	81	28.32%
CentOS	50	13.58%
Arch Linux	13	7.63%
Fedora	2	2.17%

Mierka pre vyhodnocovaciú maticu:

- Hodnota 10 = 100% zastúpenie, hodnota 0 = 0% zastúpenie.
- $X = \text{avg}(\text{“záznamy vybranej distribúcie”})/10$.

3.2.12 Podpora programovacích jazykov

Na operačných systémoch Unix a Linux sú plne podporované rôzne programovacie / skriptovacie jazyky, ktoré môžu byť použité pri vývoji webového portálu - Tabuľka 19.

Tabuľka 19. Porovnanie požiadaviek na knižnice pre programovacie jazyky

Jazyk	Požiadavky (kompilátor; virtuálny stroj; odhadnuté z Arch Linux i686)
Python	python 3.5.2-1 - ~107.5 MB
C/C++	gcc 6.2.1-1 - ~116.0 MB tcc 0.9.26-3 - ~834.6 kB
Perl	perl 5.24.0-2- ~ 46.6 MB
Java	jdk8-openjdk 8.u112-1 - ~ 116.0 MB jre8-openjdk 8.u112-1 - ~ 401.4 kB
Ruby	ruby 2.3.1-1- ~ 22.8 MB
Shell	ZSH/CSH/Bash/... - ~ integrácia v systéme
Go	go 2:1.7.3-1 - ~ 135.3 MB

Z analýzy vyplýva, že je dôležité sa rozhodnúť či sa budú aplikácie kompilovať priamo na serveri alebo na oddelenom virtuálnom stroji, na ktorom bude prebiehať aj vývoj (git zaberá ďalších ~ 32.8 MB). Všetky Unix a Linux operačné systémy majú podporu pre opísané programovacie framework-y. Obmedzená podpora sa môže vyskytnúť pri jazyku Go.

3.2.13 Virtualizácia

Všetky spomenuté operačné systémy alebo distribúcie je možné virtualizovať pomocou VirtualBox alebo Vmware Workstation.

Zvažované možnosti virtualizácie pri vývoji a testovaní:

- Centralizované riešenie – Cloud Linux/Unix distribúcia umiestnená u niektorého poskytovateľa. Problém kvôli plateným licenciám (zadarmo väčšinou iba na obmedzenú I/O kapacitu alebo priamo čas), nemožné jednoducho virtualizovať Android obrazovky.
- Decentralizované riešenie – Každý bude mať svoj virtuálny stroj na VirtualBox alebo Vmware. Problém s riadením synchronizácie, ďalší processing navyše. Výhoda – rýchlejšia odozva ako pri Cloud, možnosť virtualizovať aj Android systémy.

Možnosti pri decentralizovanom riešení:

- Vývojové prostredie môže byť umiestnené priamo vo virtuálnom stroji alebo mimo neho (napr. Visual C++ for Linux Development alebo napr. Remote System Explorer).
- Virtualizácia Android operačného systému v prípade potreby je možná. Virtualizačné nástroje umožňujú takisto vytvoriť oddelené virtuálne siete medzi zariadeniami a aj reálnou sieťou.
- Synchronizácia súborov pomocou git-u.
- Spätná synchronizácia pomocou git-u na MOD serveri, priama inštalácia cez vytvorené balíky alebo konfigurácia.
- Výhoda: môžeme všetci naraz testovať aplikácie pri vývoji, menšia záťaž na server, minimalizovanie rizika poškodenia systému.

Nevýhoda: Všetky hardwarové podmienky je nemožné nasimulovať (vrátane real-time aplikácií).

3.2.14 Výsledky porovnania distribúcií

V tejto sekcii sa nachádzajú dve tabuľky – Súhrnné porovnanie distribúcií bez prepočtu váh (Tabuľka 20) a Súhrnné porovnanie distribúcií s prepočítanými váhami na finálne hodnoty (Tabuľka 21). V druhej tabuľke sa nachádza aj súčet bodov pre všetky distribúcie (posledný riadok).

Tabuľka 20. Súhrnné porovnanie distribúcií bez prepočtu váh

Kritérium	Váha	Lubuntu	FreeBSD	Debian	ArchLinux	Fedora	Puppy	CentOS	TinyCore	TrisquelMini	Linux Lite
Skúsenosti s distribúciami v tíme	6	7.143	1.429	1.429	0	0	0	1.429	0	0	0
Podpora rôznych CPU architektúr	7	3	5	10	2	8	3	8	5	2	2
Potreba modifikácie základnej architektúry systému na lightweight	10	5	5	10	8	5	8	5	10	8	8
Veľkosť systému so základnými modulmi bez GUI (ak ho je možné odstrániť)	10	6.625	2.5	6.625	6.5	4.5	9.125	4.5	9.5	7.25	5.3
Vyladenie systému pre serverové operácie	4	7.5	10	7.5	5	7.5	5	10	7.5	5	5
Minimálne požiadavky na CPU	5	7.778	9.633	7.778	7.778	6.667	8.522	6.111	9.633	6.111	3.333
Minimálne požiadavky na systémovú pamäť	10	8	8.667	5.333	5.333	2.75	9.333	2.5	9.708	8	5.333
Životný cyklus - interval medzi novými verziami	8	9.832	8.655	7.747	9.982	9.163	1.795	1.563	9.327	7.22	9.609

Kritérium	Váha	Lubuntu	FreeBSD	Debian	ArchLinux	Fedora	Puppy	CentOS	TinyCore	TrisquelMini	Linux Lite
Životný cyklus - dĺžka podpory stabilnej verzie	10	1.763	1.583	3.498	5	1.033	0	9.18	0	2.705	2.888
Správa balíkov	8	2.5	9.75	2.5	9	5.75	0.5	5.5	1.5	2.5	2.5
Používateľská podpora	4	8.352	0.157	0.249	0.7	3.711	0.237	0.258	0.055	0.06	0.01
Podpora zariadení (ovládače)	10	10	10	8	8	8	6	10	8	6	8
Popularita za posledný rok	4	8	2.355	9.285	3.98	5.46	2.72	4.185	1.28	0.735	2.62
Počet dostupných balíkov	9	9.72	4.245	9.477	1.482	3.712	0.102	2.928	4.193	8.795	0.295
Auto-Recovery podpora	5	10	4	10	10	9	4	9	4	6	8
Použitie v serverových aplikáciách	7	3.147	0	3.029	0.589	0.156	0	1.533	0	0	0

Tabuľka 21. Súhrnné porovnanie distribúcií s prepočtom váh

Kritérium	Lubuntu	FreeBSD	Debian	ArchLinux	Fedora	Puppy	CentOS	TinyCore	TrisquelMini	Linux Lite
Skúsenosti s distribúciami v tíme	42.858	8.574	8.574	0	0	0	8.574	0	0	0
Podpora rôznych CPU architektúr	21	35	70	14	56	21	56	35	14	14
Potreba modifikácie základnej architektúry systému na lightweight	50	50	100	80	50	80	50	100	80	80
Veľkosť systému so základnými modulmi bez GUI (ak ho je možné odstrániť)	66.25	25	66.25	65	45	91.25	45	95	72.5	53

Kritérium	Lubuntu	FreeBSD	Debian	ArchLinux	Fedora	Puppy	CentOS	TinyCore	TrisquelMini	Linux Lite
Vyladenie systému pre serverové operácie	30	40	30	20	30	20	40	30	20	20
Minimálne požiadavky na CPU	38.89	48.165	38.89	38.89	33.335	42.61	30.555	48.165	30.555	16.665
Minimálne požiadavky na systémovú pamäť	80	86.67	53.33	53.33	27.5	93.33	25	97.08	80	53.33
Životný cyklus - interval medzi novými verziami	78.656	69.24	61.976	79.856	73.304	14.36	12.504	74.616	57.76	76.872
Životný cyklus - dĺžka podpory stabilnej verzie	17.63	15.83	34.98	50	10.33	0	91.8	0	27.05	28.88
Správa aktualizácií	20	78	20	72	46	4	44	12	20	20
Používateľská podpora	33.408	0.628	0.996	2.8	14.844	0.948	1.032	0.22	0.24	0.04
Podpora zariadení (ovládače)	100	100	80	80	80	60	100	80	60	80
Popularita za posledný rok	32	9.42	37.14	15.92	21.84	10.88	16.74	5.12	2.94	10.48
Počet dostupných balíkov	87.48	38.205	85.293	13.338	33.408	0.918	26.352	37.737	79.155	2.655
Auto-Recovery podpora	50	20	50	50	45	20	45	20	30	40
Použitie v serverových aplikáciách	22.029	0	21.203	4.123	1.092	0	10.731	0	0	0
Vyhodnotenie	770.2	624.732	758.632	639.257	567.653	459.296	603.288	634.938	574.2	495.922

3.2.15 Finálny výber operačného systému

Tímové rozhodnutie: Debian

Kľúčové dôvody:

- **Pri virtualizácii sa nám nepodarilo správne nainštalovať Lubuntu na limitovanom úložisku (2 GB) aj napriek tomu, že finálna inštalácia na väčších úložiskách nezaberá viac ako 1 GB.**
- V tejto časti návrhu a analýzy ešte nepoznáme všetky potrebné balíky, doplnky, codec-y ..., ktoré budeme pri vývoji potrebovať (existuje aj pravdepodobnosť, že takéto vývojové rozhodnutia sa menia aj počas vývoja). Debian má oveľa širšiu podporu balíkov ako napr. Tiny Core.
- Napriek väčšej základnej veľkosti systému, Debian podporuje najviac nástrojov na tvorbu custom inštalácií s presnou voľbou, ktoré balíky a časti systému sa majú inštalovať – jednoduchá redukcia systémovej inštalácie – odhad 300 - 400 MB (odskúšané odstránenie doc-ov, man. stránok a naindexovaných repozitárov).
- Vysoká používateľská podpora.

3.3 Multimedialný obsah

Na prehrávanie videí cez internetový prehliadač sa v minulosti používali najmä rôzne prídavné komponenty ako napr. flash. Postupný vývojom technológií sa však podarilo vytvoriť novú verziu HTML5, ktorá umožňuje realizáciu prehrávania videa bez potreby inštalovania ďalších komponentov. Prináša pritom ďalšie výhody v podobe rýchlejšieho spúšťania videa, či lepšej kvality. Problémom však je, že jednotlivé formáty videí, ktoré HTML5 podporuje, nie sú zároveň podporované na všetkých platformách. Ak chceme preto umožniť prehrávanie videa na všetkých platformách, je potrebné použiť kombináciu aspoň dvoch formátov.

- Je možné používať aj viaceré zdrojové súbory
- Možné aj prepínanie medzi zvukovými a textovými stopami

3.3.1 Porovnanie formátov podporovaných HTML5

MP4 / H.264 Codec

Jeden z najpoužívanejších video formátov, ktoré ako kontajner môžu obsahovať zároveň video, zvuk, titulky, obrázky aj kapitoly v jednom súbore. Podporovaný je spoločnosťami Apple a Microsoft. Je to licencovaný kódex, ktorý poskytuje dobrú kvalitu videa a zároveň malú veľkosť súboru na disku.

- Jednoduchosť: 4/5
- Kvalita: 4/5
- Kompresia: 4/5
- Cena: Zadarmo(Zatiaľ)

Ogg/Theora Codec

Je to voľný otvorený štandard pre kódovanie videí. Generuje vysokú kvalitu videa so znateľne väčšou veľkosťou súboru na disku. Problémom je tiež nájst' vhodný nástroj na konvertovanie videí do ogg/Theora.

Je to voľný softvér, takže je veľmi využívaný.

- Jednoduchosť: 1/5
- Kvalita: 3/5
- Kompresia: 2/5
- Cena: Zadarmo

WebM Codec(Vorbis Opus)

Je to video formát od spoločnosti Google, vytvorený pre využitie na Webe(formát pre HTML5). Používa VP8 alebo VP9 kódeky, ktorý poskytuje spoločnosť Google. Štruktúra súboru je založená na Matroska media kontajner. Poskytuje vysokú kvalitu videa, no so znateľne spomaleným procesom dekódovania v porovnaní z H.264. Aj keď spoločnosť Google prezentuje tento formát ako najlepší možný, väčšina nezávislých zdrojov hovorí o podobnej kvalite, prípadne minimálnych rozdieloch v porovnaní s formátom MP4.

Výhodou je, že tento kódek je open source.

- Jednoduchosť: 3/5
- Kvalita: 4/5
- Kompresia: 4/5
- Cena: Zadarmo

Tabuľka 22 Podpora kódekov na jednotlivých platformách

Codec	Android	Chrome		Firefox		IE	iOS	Opera		Safari	
		Win	Mac	Win	Mac	Win		Win	Mac	Win	Mac
MP4 or H.264	3.0	9	7	21	35	9	3	25	25	5	5
Ogg/Theora	2.3	9	7	3.6	3.6	X	X	10.63	10.63	X	X
WebM	2.3	9	7	3.6	3.6	9	X	10.63	10.63	X	X

Tabuľka 23 Hodnotenie parametrov jednotlivých formátov videí

	MP4	Oqq	WebM
Kvalita	1	2	3
Veľkosť	2	3	1
Kompatibilita a využitie	1	3	2

3.3.2 Návrh riešenia

Systém by sme mohli navrhnúť tak, že všetky videá, ktoré budú na serveri uložené, budú vo formáte MPEG-4. Tento formát totiž umožňuje aj obsiahnutie viacerých zvukových a textových stôp, medzi ktorými môžeme prepínať. Jeho prehrávanie na webovom portáli by bolo realizované pomocou HTML5, kde bude potrebné implementovať možnosť výberu konkrétnej zvukovej, či textovej stopy počas prehrávania na klientovi.

Problémom je fakt, že zákazníci nakupujú filmy vo formáte AVI. Je teda potrebné zabezpečiť konverziu medzi formátmi AVI do MP4. Ako najvhodnejšie riešenie sa zdá poskytnutie takéhoto softvéru zákazníkovi s prednastavenými hodnotami, potrebnými podľa špecifikácie systému. Po tom ako si zákazník konvertuje film, môže ho nahrať do systému.

Možné je však aj konvertovanie každého filmu v čase prehrávania pomocou FFMPEG. Takýto proces by však mohol znamenať veľké zaťaženie pre server, najmä v prípade, že by veľké množstvo klientov chcelo prehrávať takéto filmy.

3.4 Zaznamenávanie udalosti v systéme (Logy)

Na základe požiadavky zákazníka, je nutné zaznamenávať udalosti v systéme. Medzi tieto udalosti patrí zaznamenávanie činnosti multimediálneho servera a zaznamenávanie stavu jednotlivých zariadení v sieti.

3.4.1 Typy logov

Na multimediálnom servere sa budú vytvárať nasledujúce druhy logov:

- Auditné
- Serverové (všetky logy čo sa týkajú OS: /var/log + pripojenie/odpojenie disku):
 - lastlog
 - messages
 - daemon.log

Servisné (funkčnosť/stav hubov, monitorov)

- Identifikačné (štruktúra zapojenia zariadení)
- Web portál

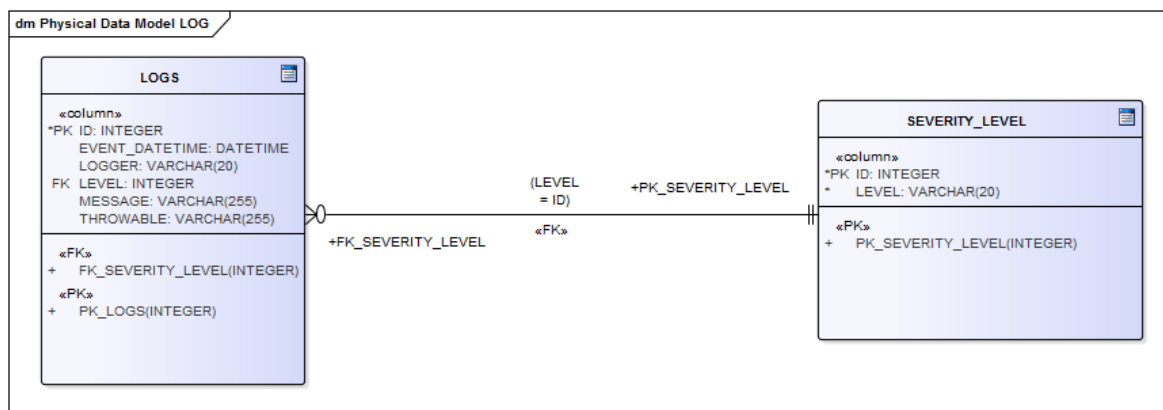
3.4.2 Výber formátu pre posielanie logov na Cloud

Tabuľka 24 Výber formátu pre odosielanie logov

	JSON	XML
Počet znakov pri uložení tej istej informácie	140	167
Dátový formát	áno	nie
Jazyk	nie	áno
Čitateľnosť (človek a stroja)	lepšia	horšia
Namespace	nie	áno
Tagovanie	nie	áno
Náročnosť na CPU	menšia	horšia
Náročnosť na pamäť	rovnaká	rovnaká
Mohutnosť zápisu	menšia	väčšia
Veľkosť (aj v prípade kompresie do zip)	menšia	väčšia

Pre náš projekt bude vhodnejšie použiť na prenos logov na cloud formát JSON.

3.4.3 Fyzický dátový model pre logy z web portálu



Obrázok 3 Fyzický dátový model pre zaznamenávanie logov zo systému MOD

Popis stĺpcov v tabuľke LOGS:

ID – jedinečný identifikátor, každej položky

EVENT_DATETIME – čas výskytu udalosti vo formáte (YYYY-MM-DD HH:MM)

LOGGER – ktorý systém zaznamenal udalosti

LEVEL – úroveň dôležitosti vyskytnutej udalosti od 0 do 7 a 0 má najvyššiu prioritu

MESSAGE – opis zaznamenanej udalosti

THROWABLE – názov vzniknutej výnimky

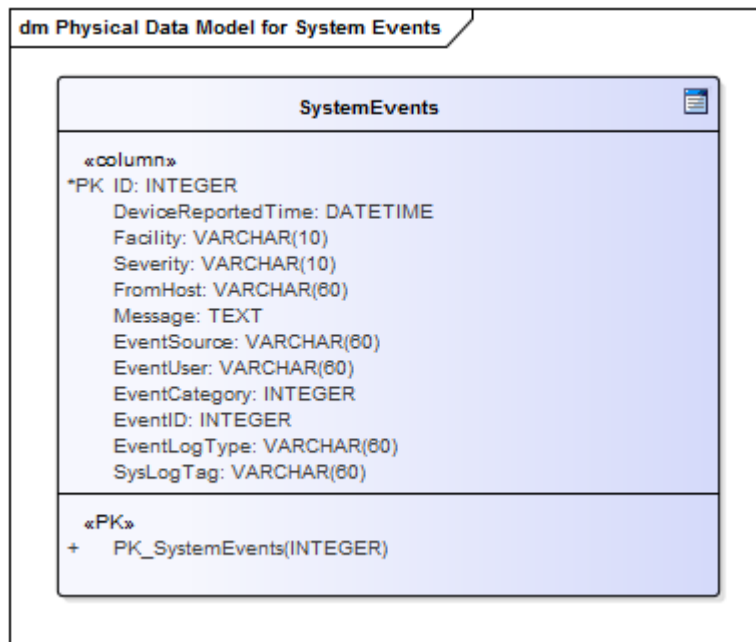
Príklad.: 1, '2017-03-07 13:33 ', 'Java', 3, 'blaaa', 'NullPointerException'

Obsah tabuľky SEVERITY_LEVEL:

Tabuľka 25 Sevrity level

ID	LEVEL
0	emergency
1	alert
2	critical
3	error
4	warning
5	notice
6	informational
7	debug

3.4.4 Fyzický dátový model pre systémové udalosti



Obrázok 4 Fyzický dátový model pre systémové udalosti

Popis stĺpcov v tabuľke:

DeviceReportedTime – čas vygenerovania udalosti

Facility – meno zariadenia, ktorého sa týka vzniknutá udalosť

Severity – dôležitosť správy

FromHost – meno systému od ktorého bola prijatá správa

Message – opis vygenerovanej udalosti

SyslogTag – TAG správy

3.5 Konzoly

3.5.1 Analýza súčasného stavu

Súčasný systém ma v ponuke dva typy konzol a to:

- Driver konzolu
- Admin konzolu

Driver konzola

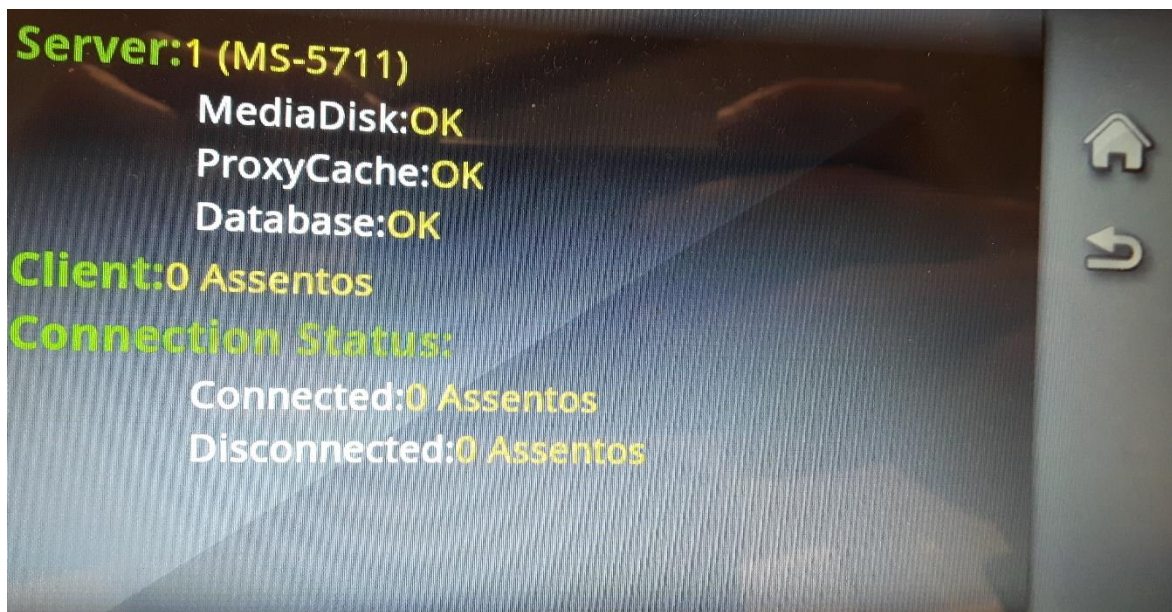
Driver konzola je rozhranie učené pre stewardku v autobuse. V rámci autobusu je väčšinou jedna z obrazoviek nastavená ako táto špeciálna Driver konzola. Po zadaní správneho hesla sa používateľ tejto obrazovky, teda vo väčšine prípadov stewardka dostane do rozhrania ktoré ponúka rôzne špeciálne funkcie systému.

Každý monitor sa môže nastaviť ako Driver konzola, táto možnosť je z Admin konzoly, ktorá je opísaná v ďalšej kapitole. Takto môže nastať, že v systéme je i viacero obrazoviek na ktorých je zapnutá Driver konzola.

Driver konzola sa zapína priamo po naboťovaní a spustení Androidu na obrazovkách automaticky. Teda tie obrazovky, ktoré boli pred bootovaním (pred vypnutím systému, pri poslednom nastavení) nastavená v móde „Driver mode“ sa zapnú v tomto špeciálnom režime.

System

Prvá položka menu v Driver konzole „Systém“ sú systémové informácie zobrazené na obrázku Obrázok 5 Položka System v Driver konzole.



Obrázok 5 Položka System v Driver konzole

Reboot

Táto položka menu slúži na reboťnutie systému a obrazoviek. Možnosti reboťtu sú viaceré. Dá sa napríklad reboťnúť systém celkovo (reboot MOD servera), ďalej je tu možnosť reboťnutia všetkých obrazoviek, treťou možnosťou je reboot samotného monitoru na ktorom sa nachádzame.

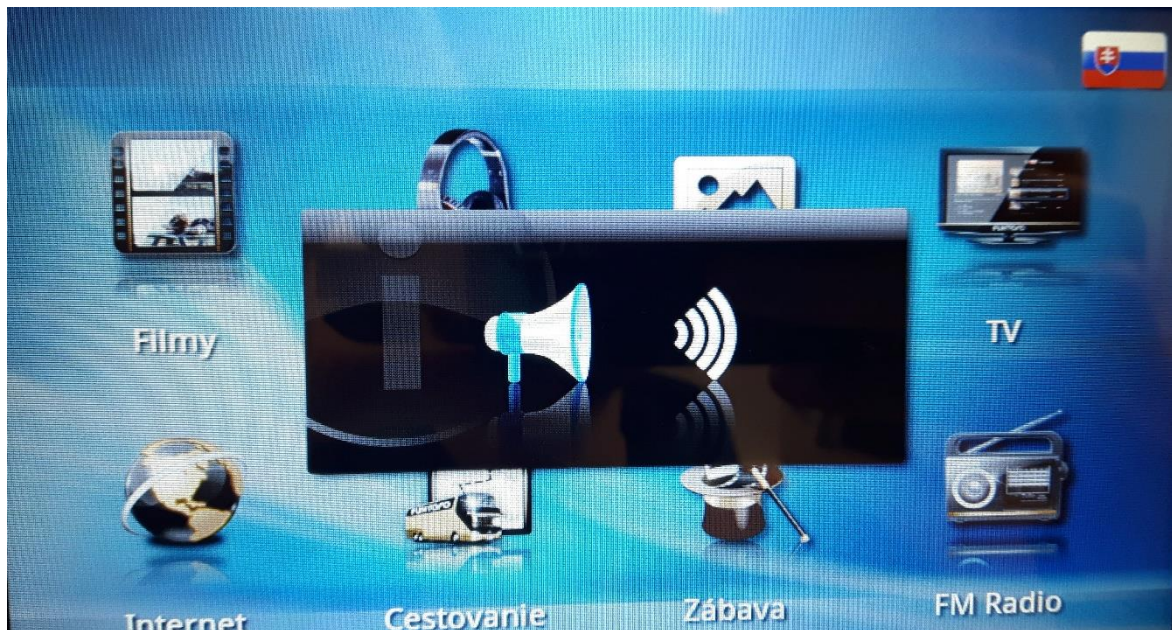
Delete Lastgood

Táto funkcia by mala zabezpečiť to, že monitor si vymaže pamäť spustených vecí, napríklad ak bol na monitore pustená film, tak voľbou tejto funkcie sa všetky monitory dostanú do pôvodného stavu. (Túto funkciu sa mi nepodarilo pustiť na serveri dostupnom v laboratórium FIIT Molpir a nie dosť dobre fungovala i priamo vo firme Molpir).

Announcement(hlásenie)

Touto funkciou stewardka môže povedať cestujúcim dôležitý oznam, ktorý si vypočujú na svojich monitoroch pomocou slúchadiel pripojených k monitoru. Počas hlásenia sa im

zobrazí ikona (pozri obrázok) ktorú nie je možné vypnúť a v tomto momente je systém cestujúcim nepoužiteľný.



Obrázok 6 Hlásenie Driver konzola

Station

Táto funkcia v laboratórium FIIT Molpir nefunguje.

Pay Movie

Stewarde pomocou tejto voľby vypína a zapína určitý druh platených fimov.

Route

Táto funkcia v laboratórium FIIT Molpir nefunguje nakoľko nie je pripojená GPS anténa.

Stewardess Call

Stewardka si zobrazí kto využil na monitore funkciu zavolania stewardky. Ukáže sa presné číslo sedadla, ktoré toto volanie vykonalo.

Show Seat No

Po stlačení tejto voľby sa zobrazia na všetkých monitoroch ich pridelené čísla (väčšinou sa jedná o čísla sedadiel nastevných v Admin konzole).

Movie Manager

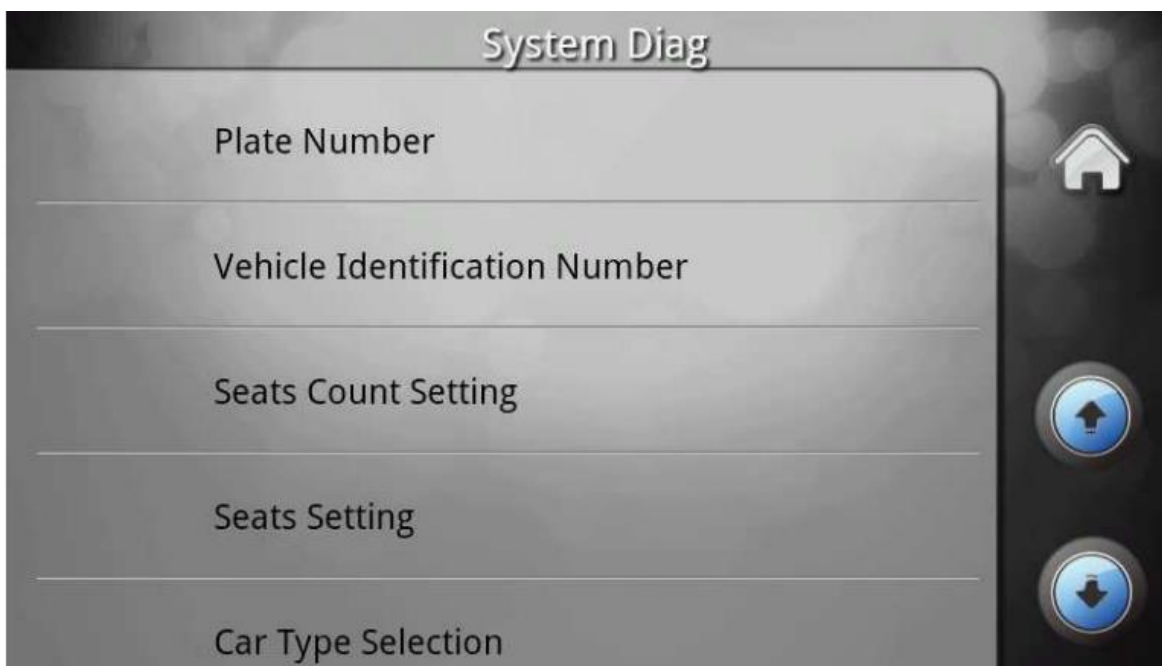
Funkcia ktorou stewardka dokáže povoliť respektíve zakázať presný film ktorý sa má alebo nemá v systéme prehrávať.

3.5.2 Admin konzola

Admin konzola je bližšie rozpísaná v dokumente Identifikácia, preto je v tejto časti opísaná len časť systému, ktorá nie je obsahom dokumentu identifikácia.

Admin konzola slúži na diagnostiku a prvotné nastavenie systému. Taktiež je cez ňu možné meniť nastavenia už existujúceho systému (zmena sedadla, zmena počtu hubov...).

Táto konzola je určená prioritne pre technika, ktorý inštaluje systém do vozidla, prípadne ho spravuje.



Obrázok 7 HomeScreen Admin konzola 1



Obrázok 8 HomeScreen Admin konzola 2

Možnosti menu v Admin konzole (funkcie opísané v časti Identifikácia)

- **Plate Number** – nastavenie evidenčného čísla vozidla
- **Vehicle Identification Number** – nastavenie VIN vozidla

- **Seats Count Setting** – nastavenie celkového počtu (sedadiel) monitorov vo vozidle
- **Seats Setting** – priradenie čísiel sedačiek k jednotlivým monitorom
- **Car Type Selection** – nastavenie typu vozidla
- **Hubs Count Setting** – nastavenie počtu HUB-ov a priradenie jednotlivých monitorov k HUBom
- **TV Channels Setting** – nastavenie počtu TV kanálov
- Network Test – test siete
- TouchScreen Testing – testovanie dotykovej obrazovky
- Date Time Settings – nastavenie dátumu a času
- **Information** – rozširujúce informácie
- Reboot Selection – možnosti reštartovania systému
- Device Mode – zmena režimu aktuálneho monitora (Guide = Driver konzola / štandardný režim monitora)
- Reading Lamp – nastavenie lampy na čítanie
- Memory Check – test pamäti systému
- Software Update – update SW verzie monitorov

Network test – test siete

Táto možnosť slúži na otestovanie rýchlosti sieťovej komunikácie a to buď jednotlivo na požadovanom monitore alebo postupným otestovaním všetkých monitorov naraz. Po zvolení tejto možnosti sa nám zobrazí obrazovka s dvomi možnosťami- Voľbou „single“ otestujeme „driver“ konzolu (resp. na požadovanom klientskom monitore kliknutím na OK), voľbou „serial“ sa test spustí postupne na všetkých monitoroch, pričom samotný individuálny test trvá približne 3 sekundy, otestovanie približne 70 pripojených monitorov okolo 3 minút.

TouchScreen Testing -testovanie dotykovej obrazovky

Táto voľba slúži na otestovanie správnej funkčnosti dotykovej plochy monitora.

Date Time Settings – nastavenie dátumu a času

Táto možnosť slúži na nastavenie dátumu a času v systéme. Po zvolení tejto možnosti sa nám zobrazí ponuka s tromi nastaveniami (dátum, čas, časové pásmo) a vo vrchnej časti obrazovky sa nachádzajú aktuálne hodnoty týchto troch veličín.

Reboot Selection – možnosti reštartovania systému

Táto možnosť slúži na reštartovanie systému a to konkrétne pomocou troch uvedených možností:

- reštart tohto monitora
- reštart všetkých monitorov
- reštart celého systému

Device Mode – zmena režimu aktuálneho monitora (Guide = Driver konzola / štandardný režim monitora)

Táto možnosť slúži na zmenu režimu monitora z režimu DRIVER (Driver konzola = špeciálne funkcie pre obsluhu systému) na CLIENT (cestujúci) alebo naopak. Aktuálny

režim je zobrazený za dvojbodkou (DRIVER / CLIENT). Zmeníme ho kliknutím na túto možnosť.

Reading Lamp – nastavenie lampy na čítanie

Táto možnosť slúži na povolenie/zakázanie LED osvetlenia na čítanie v 10“ monitoroch, ktoré môžu byť osadené LED osvetlením (v závislosti od modelu/verzie 10“ monitora a jeho zástavbovej súpravy). Aktuálny stav je zobrazený za dvojbodkou (ENABLE/DISABLE), zmeníme ho kliknutím na túto možnosť.

Memory Check – test pamäti systému

Po kliknutí na túto možnosť a spustení tlačidlom „Start“ sa zobrazia informácie o aktuálnom vyťažení systémovej pamäte monitora

Software Update – update SW verzie monitorov

Táto voľba slúži na zmenu firmware a SW aplikácie monitoru a nie je určená pre bežného používateľa !

3.5.3 Požiadavky zákazníka

Driver konzola

Tajný klik na obrazovke – umožňuje nastavenie monitoru do východzieho stavu. Resetovanie obrazovky, aby tam neostalo niečo bežať. Príklad s vystúpením a nastúpením ďalšieho cestujúceho.

Problém so systémom rezervácie lístkov, nie je možné to zovšeobecniť aj keď dá sa robiť pre jednotlivých zákazníkov na ich objednávku.

Pre danú oblasť nie je možné prehrávať nejaký obsah (alkohol, cigarety...). Riadenie obsahu aj reklamy. Toto robí väčšinou stewardka na ľubovoľnom monitore v systéme.

Admin/ instal konzola

Diagnostika beží na pozadí a odosiela sa do centrály, ale chceme aj diagnostiku priamo z monitoru.

3.5.4 Definícia konzol

Súčasný model systému disponuje dvoma typmi konzol Driver / Admin. Funkcie týchto dvoch konzol sú z veľkej časti špecifické vzhľadom na použitie. No na druhej strane existuje viacero funkcií, ktoré sa opakujú v oboch konzolách a bolo by dobre definovať, kto túto funkciu má vykonávať. Taktiež sa tu vyskytujú funkcie, ktoré by daný používateľ nemal vôbec meniť.

Steward konzola

Táto konzola bude slúžiť pre personál vozidla teda zvyčajne pre stewardku v autobuse, alebo vlaku.

- **Zobrazenie informácií o systéme**
 - Počet sedadiel

- Označenie servera
- Počet zapnutých / vypnutých monitorov
- ŠPZ
- **Reboot** – zobrazí sa zoznam sedadiel (graficky) na ktorom bude možnosť označenia sedadla na ktorom sa má rebootnúť displej.
- **Reload** – v starom systéme „Delete Lastgood“
- **Hlásenie**
- **Hlásenie zastávky**
- **Kto ma volal** – v strom systéme „Stewards call“
- **Správca obsahu**
- **Spustenie vynúteného obsahu**

Admin(Správca flotily) konzola

Špeciálna konzola pre správcu systému (servisný technik prepravcu), v spoločnostiach kde takýto technik nie je túto funkciu môže obsluhovať s priloženým manuálnom stewardka, alebo iný zamestnanec firmy. Ide o zmeny v nastaveniach, ktoré sa dejú počas prevádzky systému, čiže nie priamo pri inštalácii systému. Táto konzola umožňuje základu diagnostiku systému.

- **Test siete**
- **Touchscreen test**
- **Nastavenie dátumu**
- **Zmena počtu sedadiel**
- **Prečíslovanie monitorov**

Instal(Inštalatér Molpir) konzola

Dôvodom vytvorenia tejto konzoly je oddelenie konzoly pre prvotnú inštaláciu systému a konzoly pre úpravu už nastaveného systému.

- **Informácie o vozidle**
 - ŠPZ
 - VIN
 - Počet sedadiel
 - Počet HUBov
- **Update Androidu na monitoroch**
- **Nastavenie čísiel sedadiel** – na každom monitore sa objaví obrazovka do ktorej treba zadať číslo sedadla.
- **Ďalšie potrebné systémové nastavenia**

3.5.5 Spustenie konzol

Nakoľko nové riešenie bude implementované pomocou web portálu bude najlepšie spustenie konzoly pomocou tajného tlačidla GUI, tým pádom odpadne Android servis, ktorý by musel pozorovať kedy bolo stlačené tlačidlo monitoru. Tlačidlo by malo byť transparentnej farby a ideálne spustenie by malo byť nejakou kombináciou stlačenia dvoch častí displeja.

Spustenie konzol (tajný klik)

Na obrázku je znázornené spustenie rozhrania pre výber konzoly.



Postup spustenia:

1. 2 krát stlačte neviditeľné tlačidlo umiestnené v zobrazení času.
2. 3 krát stlačte neviditeľné tlačidlo umiestnené v logu firmy.

3.6 Synchronizácia obsahu

Autobus bude obsahovať lokálnu databázu obsahu, podľa ktorej sa bude rozhodovať o aktualizácii obsahu. Taktiež bude mať databázu so štatistikami prehratia / zobrazenia daného obsahu. Takýto prístup je dôležitý jednak kvôli sledovaniu obľúbenosti filmov a podkladom pre fakturáciu prehratia reklám.

Všeobecné požiadavky:

- musí prebiehať automaticky – bez inicializácie používateľom
- používatelia si nemôžu nič všimnúť – synchronizácia musí prebiehať v pozadí
- po pridaní novej položky obsahu musí prebehnúť prídanie do databázy obsahu (lokálne aj *cloud*) a inicializovanie štatistík prehratí/zobrazení
- pred odobratím existujúcej položky musí prebehnúť záloha štatistík na *cloud*, vymazanie z databázy obsahu (lokálne aj *cloud*) a samotné odstránenie súborov
- musí podporovať pokračovanie v prerušenom sťahovaní

Podporované spôsoby synchronizácie:

- prostredníctvom Wi-Fi
- prostredníctvom mobilnej siete
- výmenou disku s obsahom

System musí umožňovať zvoliť prioritu prenosu. Jedná sa o výber medzi Wi-Fi a mobilnou sieťou, kedy je v určitých prípadoch vhodnejšie na synchronizáciu využiť mobilný prenos a inokedy zase neprípustné.

Výmena disku je špeciálnym prípadom, ktorý musí autobus vedieť detegovať a následne spustiť zálohu pôvodných štatistík a aktualizáciu databázy obsahu.

3.6.1 Analýza transportných protokolov

FTP – File Transfer Protocol (a FTPS)

Výhody:

- umožňuje pokračovať v prerušenom sťahovaní
- vhodný v prípadoch, kedy je potrebné cez internet preniesť väčší objem dát
- ponúka veľké množstvo klientskych aplikácií
- možnosť vytvoriť frontu súborov na stiahnutie
- bez obmedzení veľkosti súboru

Nevýhody:

- nízka bezpečnosť (všetky nastavenia, *login* a heslo posielané ako *plain text*)
- problémy s nastavením *firewall*-u (porty)
- vyžaduje dva kanály (dáta, riadenie)

Problémy z bezpečnosťou rieši FTPS (FTP over SSL).

Implementácia FTP protokolu pre *nix systémy:

Tabuľka 26 Porovnanie klientskych aplikácií využívajúcich FTP protokol

Klient	Open Source (10%)	Nemá GUI (20%)	Veľkosť (30%)	Pokračovanie v sťahovaní (20%)	Podpora OS (10%)	Má knižnicu (10%)	Skóre (100%)
Wget	10	10	10	10	10	0	9
Curl	10	10	2	10	10	10	8
FileZila	10	0	2	10	10	0	5
Gftp	10	0	6	0	10	0	4

SFTP (SSH File Transfer Protocol)

Výhody:

- využíva jeden kanál pre dáta aj riadenie
- bezpečnosť
- umožňuje pokračovanie v sťahovaní

Nevýhody:

- pomalšie než FTP
- menej open-source klientských aplikácií

FSP – File Service Protocol

Výhody:

- založený na UDP – menej *traffic overhead*-u
- pokračovanie v sťahovaní
- nevyžaduje proces na každého klienta

Nevýhody

- absencia autentifikácie
- pomalšie než FTP

UFTP – Encrypted UDP based FTP with multicast

Výhody:

- rýchlosť pri bezdrôtovom prenose
- možnosť *multicast*-u pri viacerých klientoch sťahujúcich ten istý súbor
- možnosť nastavenia veľkosti blokov
- klient si vie vypýtať chýbajúce bloky (pokračovanie v sťahovaní)
- bezpečnosť – podporuje šifrovanie

Nevýhody

3.6.2 Vyhodnotenie transportných protokolov

Tabuľka 27 Vyhodnotenie transportných protokolov

Protokol	Pokračovanie v sťahovaní (20%)	Bezpečnosť (30%)	# klientských aplikácií (10%)	Rýchlosť – bezdrôtová sieť (30%)	Riadenie zahltenia (10%)	Skóre (100%)
FTP	6	0	10	8	10	6
FTPS	6	8	8	6	10	7
SFTP	6	7	6	5	10	6
FSP	10	0	2	2	0	3
UFTP	10	10	2	10	0	8

Z vyššie uvedenej analýzy vyplýva, že najvhodnejšími protokolmi na prenos sú FTP a UFTP. Rozdiel medzi týmito dvomi protokolmi je v transportnej vrstve, kde jeden využíva TCP prenos a druhý UDP prenos. Keďže prenos bude prebiehať vždy cez bezdrôtovú sieť, volíme ako vhodnejší protokol založený na UDP prenose, čiže UFTP.

3.6.3 Proces synchronizácie obsahu

Proces sa bude spúšťať pravidelne po uplynutí zadefinovaného intervalu. Jednotlivé kroky sú znázornené na obrázku s diagramom synchronizácie obsahu.

Po úspešnom ukončení procesu aktualizácie multimediálneho obsahu sa potvrdí jeho dokončenie odoslaním správy na cloud, ktorý zaeviduje aktuálnosť obsahu na danom servere (v autobuse).

V prípade potreby uvoľnenia miesta sa v súbore zadefinujú na vymazanie a tie sa pri najbližšom reštarte vymažú.

3.6.4 Porovnanie súborov s obsahom

Po stiahnutí súboru s aktualizovaným obsahom content databázy sa vytvoria dva súbory. Počas porovnávania aktuálneho a aktualizovaného zoznamu sa budú zapisovať definície súborov na vymazanie/stiahnutie do príslušných súborov.

3.6.5 Aktualizácia systému

Po stiahnutí súboru s aktualizovaným obsahom databázy obsahu sa vytvoria dva súbory. Počas porovnávania aktuálneho a aktualizovaného zoznamu sa budú zapisovať definície súborov na vymazanie/stiahnutie do príslušných súborov.

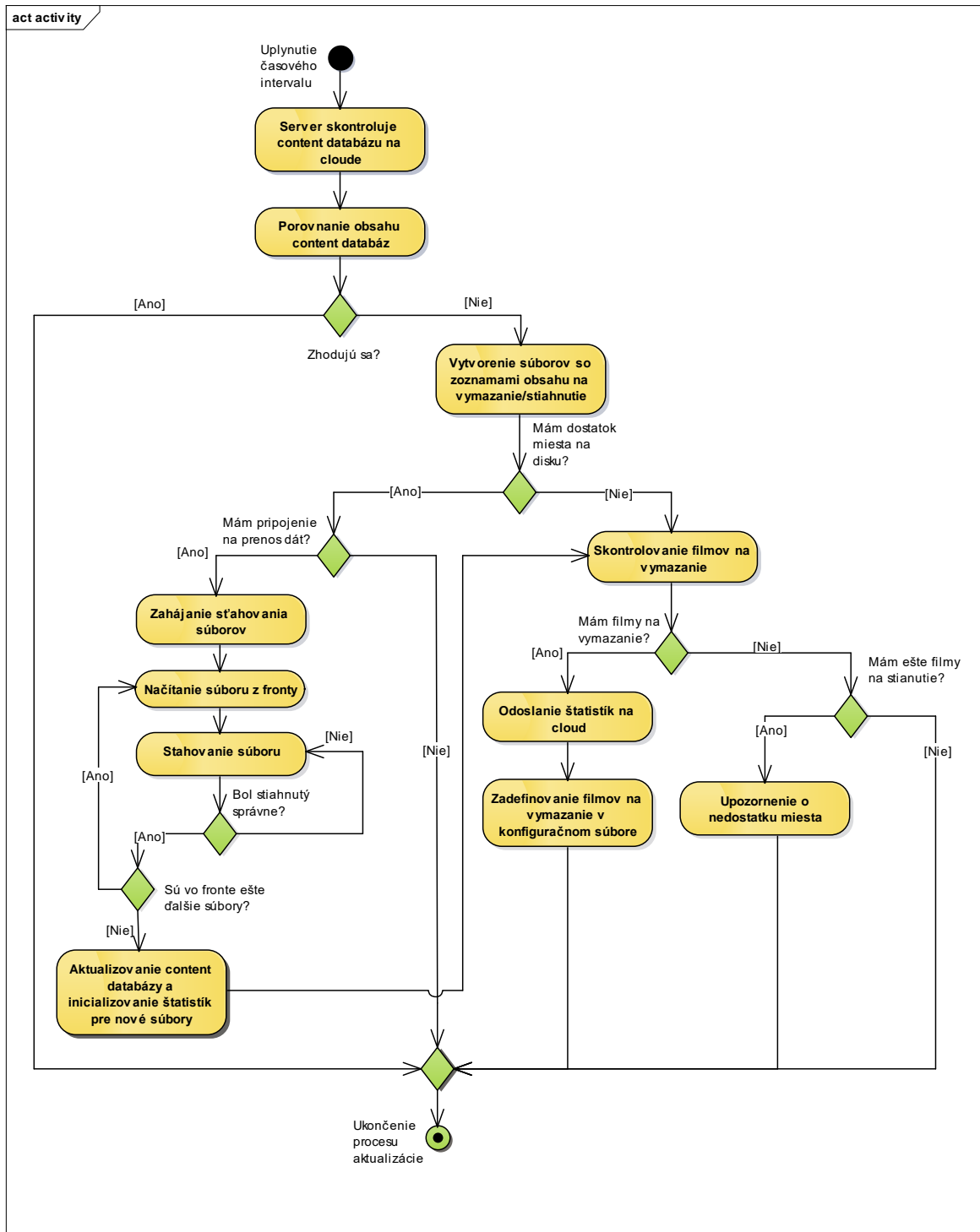
3.6.6 Configuration Management system

Systém na správu operačného systému na diaľku. Využíva sa pri aktualizáciách systémov, zvyšovaní funkcionality, výkonu, udržiavania a pod.

Puppet

Je dizajnovaný na správu konfigurácie Unix a Windows operačných systémov. Používateľ definuje systémové zdroje a ich stavy pomocou Puppet deklaratívneho jazyka alebo jazyka Ruby DSL. Tieto informácie sú potom uložené v súbore Puppet manifest. Puppet potom získava systémové informácie prostredníctvom pomocného programu, ktorý sa nazýva Facter a kompiluje Puppet manifest. Výstup kompilácie je špecifický pre daný operačný systém a obsahuje zdroje a ich závislosti.

Softvér je navrhnutý s architektúrou Client-Server. Client je nazývaný agent a server je nazývaný master. Puppet master je inštalovaný na jednom alebo viacerých serveroch a agent je inštalovaný na operačnom systéme, ktorý je spravovaný. Agent komunikuje so serverom a sťahuje si aktualizácie, ktoré potom aplikuje na daný systém.



Obrázok 9 Diagram procesu synchronizácie obsahu

3.7 Identifikácia

3.7.1 Analýza súčasného stavu

Identifikácia je v systémoch MOD (Media On Demand) dôležitým prvkom diagnostiky systému. Používa sa pri diagnostike systému.

V teraz dostupnej verzii systému pre server MOD je dostupná identifikácie na báze MAC adries a funguje na základe konfigurácie.

3.7.2 Zistenia o identifikácii z dokumentu „Setup of Funtoro HD system“

Odkaz na dokument: http://shop.molpir.sk/Prilohy/25415000856_0.pdf

Do rozhrania v ktorom je možné nastavovať informácie o vozidle, systéme, obrazovkách sa dostaneme cez Systémovú diagnostiku (System / Instal consola).

Možné nastavenia v tomto režime (tie, ktoré nás zaujímajú z hľadiska identifikácie):

1. Plate Number – setting of vehicle registration plate number
2. Vehicle Identification Number – setting of VIN of the vehicle
3. Seats Count Setting – setting of the total amount of (seats) monitors in the vehicle
4. Seats Setting – assigning the seat numbers to on particular monitors
5. Car Type Selection – setting of car type
6. Hubs Count Setting – setting of the HUBs amount and assignment of the particular monitors to HUB
7. TV Channels Setting – setting of amount of TV channels
8. Information – additional information

PLATE NUMBER

Plate number slúži ako identifikátor vozidla v diagnostickom centre, CLOUD, obsah, štatistiky. Je veľmi dôležité ho v súčasnom systéme nastavovať.

Väčšinou sa tu nastavuje štátna poznávacia značka(ŠPZ).

VEHICLE NUMBER

Klasické VIN číslo.

K súčasnej autentifikácii neexistuje žiadna dokumentácia a je teda plne na nás ako bude identifikáciu riešiť.

SEATS COUNT SETTING

Uvádza sa celkový počet monitorov vo vozidle, vrátane Driver console. Celkový počet nesmie byť číslo sedadiel.

SEATS SETTING

Po zvolení možnosti Seats Setting sa zobrazí obrazovka s aktuálne nastavenými parametrami daného monitora. Táto obrazovka sa zobrazí na všetkých pripojených monitoroch. Červené číslo predstavuje číslo monitora, pod ním sa nachádza číslo HUB-u, ku ktorému je daný monitor pripojený. V ľavej časti obrazovky sa nachádzajú celkové počty monitorov a HUB-ov a v pravej časti obrazovky sa nachádzajú tri tlačidlá na zmenu týchto parametrov (číslo monitora a HUB-u, ku ktorému je pripojený).

SET ALL SEATS

Po zvolení tejto možnosti sa na všetkých monitoroch zobrazí biela jednotka a je možné postupne nastaviť všetky monitory „kliknutím“. Kliknutím na požadovaný monitor sa monitor nastaví na číslo jedna, zobrazené číslo monitora sa zmení na červené a na všetkých ostatných monitoroch sa zobrazené biele číslo zvýši o jedna. Rovnakým spôsobom postupne nastavíme všetky monitory (priradíme monitory jednotlivým sedačkám).

SET THIS SEAT

Táto možnosť umožňuje individuálne nastavenie daného monitora. Po zvolení tejto možnosti sa zobrazí textové pole, do ktorého je možné vložiť číselnú hodnotu.

SET HUB NO

Táto možnosť umožňuje nastavenie čísla HUB-u, ku ktorému je pripojený aktuálny monitor. To je podobné ako v predchádzajúcom prípade.

CAR TYPE SELECTION

Táto možnosť umožňuje výber preddefinovaných profilov priradenia monitorov k HUB-om (schém zapojenia). Profil presne definuje, ktoré sedadlo je pripojené ku ktorému HUBu.

Táto možnosť je vhodná pre opakované nastavenia rovnakého typu monitorov a HUBov vo viacerých autobusoch s rovnakou architektúrou MOD systému (forma „vzorového“ zapojenia). Ukážka profilu:

To znamená, že k HUB-u č. 1 sú pripojené monitory 3, 6, 9, 12, 15 a 18 atď.

Samotné Car Type profily sú definované XML súborom v Profile MOD systému (zákaznícky Profil aplikácie), ktorý využíva MediaManager pri tvorbe SSD disku s mediálnym obsahom. V prípade zavedenia nového typu inštalácie je potrebné upraviť/doplniť príslušný .xml súbor v adresári „Car_Type“ na SSD disku resp. v príslušnom „Profile“.

HUBS COUNT SETTING

Vložte celkový počet monitorov HUB-ov v danom vozidle. Po zadaní počtu HUB-ov potvrdíte zadaný údaj tlačidlom Submit.

TV CHANNELS SETTING

Vložte celkový počet TV kanálov (AV vstupov) do Capture Boxov, ktoré sú k dispozícii. Po zadaní počtu TV kanálov (AV vstupov) potvrdíte zadaný údaj tlačidlom Submit.

Maximálny možný počet AV vstupov je 16 (pri použití 4ks Capture Boxov).

INFORMATION

Táto možnosť slúži na zobrazenie systémových (hardvérových a softvérových) informácií. Hlavná obrazovka obsahuje 5 výberových možností.

BUS INFO (HW) – HARDVÉROVÉ INFORMÁCIE O AUTOBUSE

Medzi hardvérovými informáciami o autobuse (Bus Info HW) nájdeme informácie, ako aktuálne nastavené evidenčné číslo vozidla, VIN, celkový počet monitorov (sedačiek) a aktuálne GPS súradnice.

Po pripojení funkčnej GPS antény sa na tejto obrazovke zobrazujú aktuálne GPS koordináty. Keď anténa nie je pripojená alebo neposiela platné NMEA vety, tak sa namiesto GPS koordinát zobrazí znak: N/A

BUS INFO (SW) – SOFTVÉROVÉ INFORMÁCIE O AUTOBUSE

Medzi softvérovými informáciami o autobuse (Bus Info SW) sú zobrazené verzie aplikácií bežiacich na serveri, verziu MediaManager-a a profil, ktorý bol použitý pri generovaní disku.

CLIENT INFO (HW) – HARDVÉROVÉ INFORMÁCIE O KLIENTOVI

Medzi hardvérovými informáciami o klientovi je zobrazená verzia bootloadera, priradené číslo monitora a jeho IP adresu. (viď. obr. 20). IP adresa monitora sa na krátku chvíľu zobrazí aj po vypnutí monitora v ľavom dolnom rohu obrazovky.

CLIENT INFO (SW) – SOFTVÉROVÉ INFORMÁCIE O KLIENTOVI

Obrazovka so softvérovými informáciami o klientovi.

CLIENT INFO (PROFILE) – INFORMÁCIE O PROFILE KLIENTA

Obrazovka obsahujúca informácie o profile klienta (verzia, popis, posledná zmena)

3.7.3 Aktuálne parametre identifikácie

Klient

Verzia profilu, popis profilu, posledná zmena.

Vozidlo

Plate number, VIN, počet sedadiel, typ vozidla, počet HUBov, počet TV kanálov, GPS.

MOD

Verzia aplikácií, verzia MediaManagera.

Monitor

Počet sedadiel, počet HUBov, IP adresa, verzia Bootloadera.

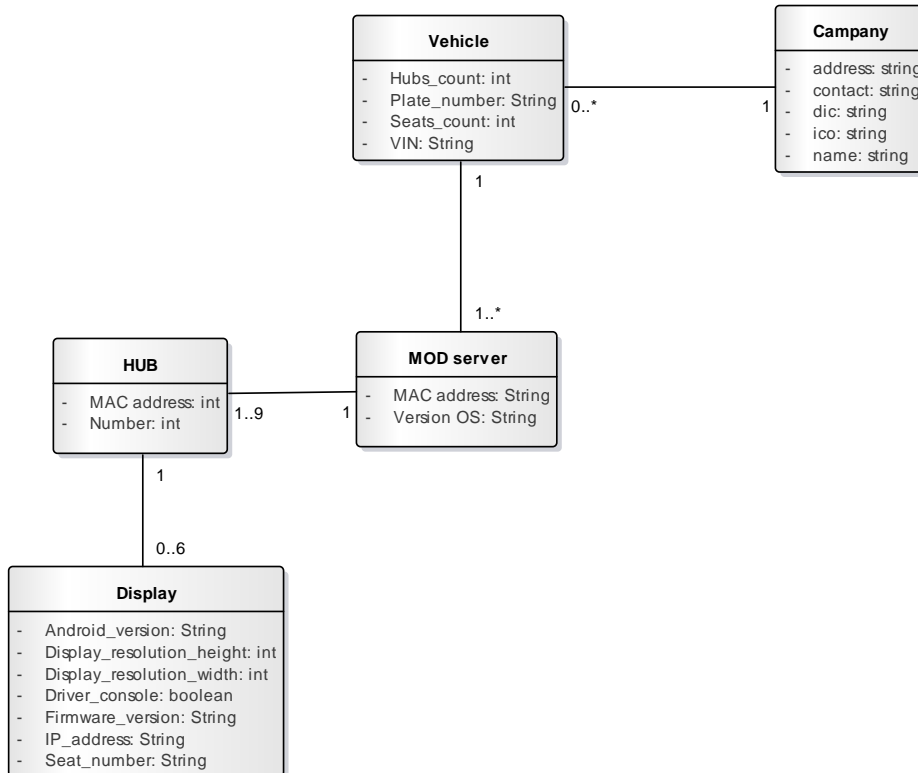
3.7.4 Požiadavky zákazníka

Zákazník požaduje od novo navrhovaného systému identifikáciu jednotlivých komponentov systému, teda: HUB, monitorov, autobus, vlak. Systém by mal tieto údaje

povoliť zadať inštalatérovi, taktiež by tu mala byť možnosť zmeny týchto údajov pri výmene komponentu. Zákazník vyžaduje evidenciu MAC adries, taktiež treba rátať s možnosťou rozšírenia identifikácie pomocou sériových čísel. Sériové čísla nesú v sebe mimo iného informáciu o záruke daného komponentu.

3.7.5 Logický model Identifikácie

V tejto časti je navrhnutý logický model Identifikácie. Je založený na rozdelení zákazníkov podľa spoločností.



Obrázok 10 Logický model identifikácie

3.8 Transcoding

Tabuľka 28 Programy / plagíny pre kódovanie/dekódovanie videa/audia

Program/Plugin	Zameranie	Formáty	Podpora	Velkosť	Info
faad-2 v. 2.7.3	audio dekodér	MPEG-2, MPEG-4	nizka		zastaralý (2010)
ffmpeg v. 3.1.2	audio, video, subtitle, image	takmer všetky dostupné	velmi vysoka	50 - 60 MB	aktualný (2016)

twolame v. 0.3.13	audio koder	MP2	nizka		zastaraly (2010)
vlc v. 2.2.4	audio, video, subtitle, image	takmer všetky dostupné	velmi vysoka	50 - 60 MB	aktualny (2016), využíva libavcodec z ffmpeg
liba52 v. 0.7.4	stream decoder	ATSC A/52	nizka		zastaraly (2002)
libmpeg2 v. 0.5.1	stream decoder	MPEG-2	nizka		zastaraly (2008)

Fungovanie SAT a CAM streamu v momentálnom systéme

Momentálne prebieha transcoding videa v MOD systéme pomocou implementovaných pluginov aj pomocou prídavného hardvéru. Prídavný HW sa používa pri konverzií dát zo satelitu - z analógových na IP stream. Prídavným HW je 4 vstupový Capture Box. Tento spôsob dovoľuje aby naraz bežalo 16-32 kanálov súčasne. Podľa informácií od zákazníka, je potrebné tento systém zmeniť.

SAT Receivery - kompresia videa – MPEG-2 / H.264, kompatibilné s MPEG-1

SAT Receivery - kompresia audia – MPEG-1, MPEG-2 Layer I a II, MP3

Capture Boxu - kompresia videa (FMS5713) – H.264 (D1 rozlíšenie)

Capture Boxu - kompresia audia (FMS5713) - 16 bit. PCM

Pri CAM streame prebieha prenos prostredníctvom RTSP – real-time video streaming protocol.

IP Cam – kompresia videa – H.264 / MPEG-4 / MJPEG

IP Cam – kompresia audia – PCM / AAC

Max PPI ľudskeho oka

Po štúdií problematiky z viacerých zdrojov vyšiel záver, že hodnoty do 300 PPI sú úplne postačujúce pre ľudské oko v prípade, že nebude možné príliš veľké zoomovanie videa. Nasledujúca štatistika popisuje špecifikácie priemerného ľudskeho oka:

Human Eye Specifications (typical):

- Sensor (Retina) : 22mm diameter x 0.5mm thick (section); 10 layers
- Resolution : 576MP equiv.

- Visual Acuity : ~ 74 MP (Megapixels) (printed) to show detail at the limits of human visual acuity
- ISO : 1 - 800 equivalent
- Data Rate : 500,000 bits per second without colour or around 600,000 bits per second including colour.
- Lens : 2 lenses - 16mm & 24mm diameter
- Dynamic Range - Static : contrast ratio of around 100:1 (about 6 1/2 f-stops) (4 seconds)
- Dynamic Range - Dynamic : contrast ratio of about 1,000,000:1 (about 20 f-stops) (30 minutes)
- Focal Length : ~ 3.2mm - (~ 22mm 35mm equiv)
- Aperture : f2.1 - f8.3 (f3.5 dark-adapted is claimed by the astronomical community)
- FOV Field of View : 95° Out, 75° Down, 60° In, 60° Up
- Color Space - 3D (non-linear) RGB
- Color Sensitivity : 10,000,000 (ten million)
- Color Range : 380 to 740 nm
- White Balance : Automatic (constant perceived color under different lighting)
- Refresh Rate : foveal vision (high-quality telescopic) - 3-4fps; peripheral vision (very inaccurate) - up to 90fps

3.9 Webportál

3.9.1 Úvod

Webový server má v zásade dve možnosti, ako získať informácie, ktoré vracia klientom:

- sú to buď dopredu pripravené dátové súbory (HTML stránky), ktoré webový server bez zmeny poskytne klientovi (tzv. statický obsah)
- až na základe požiadavky klienta sú dáta zhromaždené (prečítané zo súboru, databázy, alebo nejakého koncového zariadenia), sformátované a pripravené k prezentácii vo formáte HTML a poskytnutá webovému prehliadaču (tzv. dynamický obsah)

K dynamickému vytváraniu obsahu sa používa množstvo rôznych technológií (C++, PHP, ASP, ASP.NET, JSP a pod.). Statický obsah je schopný server poskytnúť významne rýchlejšie než dynamický. Na druhej strane pomocou dynamického obsahu je možné poskytnúť omnoho väčší obsah informácií a je možné reagovať i na rôzne „ad hoc“ požiadavky klientov. Preto sa v praxi v mnohých prípadoch oba spôsoby poskytovania obsahu kombinujú – napríklad pomocou cachovania.

3.9.2 Analýza jazykov a frameworkov

C++

C++ má ďaleko od populárneho jazyka pre vývoj webových aplikácií z niekoľkých dôvodov: nedostatok vhodných nástrojov, zručností programátorov, kompilačných závislostí a veľa iných.

Napriek tomu existujú situácie, kedy web programovanie v C++ príde vhod. Najmä web sídla a aplikácie so stovkami a tisíckami zásahov za sekundu, kde vysoký výkon, efektívnosť a škálovateľnosť sú nevyhnutné.

Využitie CppCMS:

- Linicom – reklamný engine, 10 mil. requestov za deň dynamicky generovaného obsahu
- FZB-SG – bitcoin zmenáreň, bežaca na serveri s 1GB ram
- Filebase.ws – torrent tracker, 3 mil. návštev, 5000 online používateľov

Java

Výhody:

- Robustný jazyk
- Škálovateľný pre webové aplikácie
- Bezpečný
- Multithreading
- Obrovské množstvo stabilných knižníc a rôznych web frameworkov

Nevýhody:

- Väčšie nároky na zdroje (hlavne ram)
- Viacej „obaľovacieho“ kódu pri programovaní, oproti iným jazykom (Python, ruby, javascript, ...)

Node.js

Open source JavaScript runtime postavené na Google Chrome V8 JavaScript engine. Tento engine kompiluje JavaScript do natívneho strojového kódu, čo má za výsledok rýchlejší beh a exekúciu. Node.js je taktiež asynchrónny a udalosťami riadený, vďaka čomu nenastávajú blokové situácie napr. pri čítaní súborov, práci s databázou a podobne.

Výhody:

- Rovnaký jazyk ako na frontende – javascript
- Ľahko škálovateľný
- Lepší výkon oproti podobným jazykom
- Podporuje cache-ovanie
- Jednoduchá podpora rôznych vývojových nástrojov (node package manager)

Nevýhody:

- Menej robustných knižníc
- Meniace sa vývojové API (existuje ale LTS verzia)

- Asynchrónny model môže byť zložitejší na pochopenie (ale má svoje výhody)
- Nepodporuje multithreading (dá sa vyriešiť viacerými inštanciami a load balancerom)

PHP

Výhody:

- Jednoduché na naučenie
- Veľká komunita
- Veľa doplnkov a zdrojových kódov
- Podporuje ho väčšina web serverov

Nevýhody:

- Ošetrovanie chýb je slabé
- Považované za menej bezpečné, oproti iným jazykom
- Veľa možností programovania – kód nemusí mať jasnú štruktúru (procedurálne - OOP)

Python

Všeobecný programovací jazyk. Navrhnutý, aby umožnil programátorom vyjadriť sa s čo najmenším množstvom kódu.

Výhody:

- Jednoduchý na naučenie
- Čitateľná a organizovaná syntax
- Veľká podpora komunity
- Jednoduchý na otestovanie, prototypovanie
- Znovupoužiteľný cez balíky a moduly
- OOP

Nevýhody:

- Problémy s viacerým procesmi, jadrami
- Pomalší ako Java
- Limitácie pri prístupe k DB

Ruby

Výhody:

- Veľmi vysoko úrovňový jazyk
- Čisto OOP
- Pokročilé techniky na manipuláciu stringov a textu
- Ľahko pripojiteľný k MySQL, Oracle, ...
- Veľké programy sú ľahko udržiavateľné

- Čistá syntax
- Podporuje viac procesov, threadov
- Bezpečnosť

Nevýhody:

- Ťažší na naučenie
- Menej informačných zdrojov
- Slabý výkon
- Vývoj a aktualizácie sú pomalšie

3.9.3 Zhrnutie

Všetky spomenuté programovacie jazyky sú široko rozšírené a použité na tvorbu a beh veľkého množstva webov. K všetkým jazykom tiež existujú rôzne frameworky. Pri výbere konkrétneho jazyka navrhujeme zvážiť tieto hlavné podmienky:

- Výkon/pamäť – server je umiestnený v každom autobuse zvlášť a je hardvérovo obmedzený (1 jadro, 1GB RAM, 1,5 GB pamäť)
- Jednoduchý jazyk na naučenie, vývoj
- Nasadzovanie web aplikácie (keďže server nebude stále pripojený na internet)

V kategórii výkonu prehrávajú jazyky Python a Ruby, keďže sú interpretované. Java, Node.js a PHP sú na druhom mieste a víťazom je samozrejme C++. Java má ale reputáciu „žrčča“ pamäte a to by mohlo byť v našom prípade problém.

Jednoduchosti jazyka na naučenie v tom sú všetky jazyky viac-menej podobne, s výnimkou C++ (a možno Ruby), kde musí byť programátor obozretnejší na viac vecí. Dodal by som, že Node.js je vlastne JavaScript, ktorý budeme pravdepodobne aj tak potrebovať na front-ende našej aplikácie.

Ako som už spomínal, všetky tieto jazyky sú veľmi rozšírené a to znamená ja to, že majú vyriešené nasadzovanie vyvinutých aplikácií na server. V PHP a Jave to funguje iba skopírovaním a prepísaním potrebných súborov. Node.js má zase veľmi chválený npm – node package manager, do ktorého vieme pridať našu aplikáciu a potom ju na servery pomocou neho stiahnuť a nasadiť.

3.10 Reklamy

Analýza formátov reklám

- Video – rovnaké ako filmy – mp4
- Obrázky – nezáleží, preferované jpg
- Text – podľa použitia, dĺžky, formátovania – záznam v DB, alebo textový súbor na disku

- Vlastný dizajn webportálu (napr. student agency) – pomocou css a prípadnou modifikáciou web stránok – vyvíjať portál pomocou MVC, aby čo najmenej kontrolného kódu bolo vo web stránkach.

Analýza možností ako zobrazovať reklamy

Dve hlavné možnosti:

- Vyvinúť vlastné riešenie v JavaScripte (napr. overlay stránky pri spustení reklamy) – úplná kontrola nad vykonávaním, jednoduchý systém. Zložitosť môže extrémne narásť pri komplexnejšej práci s reklamami a ošetrovaním rôznych hraničných prípadov.
- Využiť API videoprehrávača – zložitejšie na prvotnú implementáciu, limitácia treťou stranou. Stabilný a otestovaný softvér s komunitou, poskytujúci širokú škálu využiteľných metód.

Analýza podporovaných reklám

Video reklama počas prehrávania filmu:

- Preroll – na začiatku prehrávania
- Midroll – počas prehrávania (po určitom čase)
- Postroll - po skončení prehrávania

Súčasný stav – výber náhodnej reklamy z priečinka, bez ďalšej kontroly

Požiadavky, možné vylepšenia – umožniť manažovanie reklám:

- Preferované reklamy pre daný film, kategóriu, jazyk, ...
- Zvýšiť/znížiť pravdepodobnosť prehratia konkrétnej reklamy
- Zadať maximálny počet prehraní
- Limitovať prehrávanie reklamy pomocou dátumov od-do

Logovanie reklám

Tabuľka reklamy:

id, nazov, popis, subor, typ(video/text/...), prehravat_od, prehravat_do, dlzka(?)

Tabuľka logov (previazaná s tabuľkou reklamy):

Možnosti:

- id, id_reklamy, cas - presnejšie logovanie, zaberie viac miesta (každý záznam jeden riadok)
- id, id reklamy, cas_od, pocet - vieme iba celkový počet od poslednej synchronizácie, ktorý sa navyšuje (pri synch. sa vynuluje, nastaví nový čas od)
- id, id_reklamy, den, pocet – kompromis, počet prehraní za daný deň

Prototyp na prehratie videa

Prototyp na prehratie videa je možné spraviť jednoduchými úpravami demo_example súborov alebo stiahnutím zdrojových kódov daného playera a vytvorením .html súboru. Obidve možnosti boli odskúšané.

Matica video prehrávačov

Najpoužívanejšie videoprehrávače na webe: <https://www.datanyze.com/market-share/online-video/>. Väčšina prehrávačov z tohto zoznamu je platených.

jPlayer:

- Rozšírený, ale postupne zaostáva
- Využíva ďalšiu knižnicu jQuery
- Bol vyvinutý primárne pre Flash, HTML5 video bolo dorábané neskôr
- Je potrebné ho dodatočne štylovať
- Open source

VideoJS

- Open source, aktívne vyvíjaný množstvom dobrovoľníkov
- Množstvo voliteľných prídavkov (reklama, ...)
- Podpora titulkov, kapitol, zvukových stôp, ...
- Možnosť (ale nie nevyhnutnosť) customizácie

3.11 Ukladací priestor

3.11.1 Požiadavky

Zoznam známych požiadaviek a poznámky na vlastnosti súborového systému:

- Enkryptovanie alebo uzamykanie súborového systému nie je potrebné riešiť z dôvodu nedostupnosti integrovanej pamäti serverového zariadenia.
- Je potrebné vyriešiť formátovanie štruktúry SSD diskov (lokálne vs pomocou externého nástroja).
- Je potrebné vyriešiť zaznamenávanie udalostí v súborovom systéme – odpájanie, pripájanie zariadení.
- Uchovávanie informácií o aktuálnych súboroch na disku.

3.11.2 Analýza súčasného riešenia na serveri

Súborové systémy

Použité súborové systémy priradené k partíciám (príkaz `df -hT`):

Filesystem	Type	Size	Used	Avail	Capacity	
Mounted on						
/dev/ada0p2	ufs	1.7G	1.5G	48M	97%	/
devfs	devfs	1.0k	1.0k	0B	100%	/dev
/dev/ada0p3	ufs	96M	37M	50M	43%	/share
tmpfs	tmpfs	1.7G	2.4M	1.7G	0%	/tmp
procfs	procfs	4.0k	4.0k	0B	100%	/proc

/dev/md0	ufs	30M	1.9M	26M	7%	/var
devfs	devfs	1.0k	1.0k	0B	100%	
/var/named/dev						
/dev/ada1s1	ext2fs	54G	15G	36G	30%	/data
/dev/ada1s2b	ufs	3.5G	16k	3.2G	0%	/disk2

Kľúčové výsledky analýzy:

- Na SSD disku je použitý súborový systém ext2fs (/dev/ada1s1 - /data).
- Pre operačný systém je použitý súborový systém UFS (/dev/ada0p2 - /).
- Špeciálne particie (/tmp, /proc, /var/named/dev, /dev) majú pridelené vlastné špecializované súborové systémy.
- Nie je použitý LVM (Logical Volume Manager) ako ďalšia úroveň abstrakcie nad particiami.

Politika zálohovania

Kľúčové výsledky analýzy:

- V operačnom systéme FreeBSD bola identifikovaná iba Dual Boot záloha.
- Dual Boot sa aplikuje iba na vytvorený obraz jadra systému. Táto záloha je statická – bola pravdepodobne vytvorená pri návrhu operačného systému.

```
mv /boot/kernel /boot/kernel.old
...
# výskyt poruchy
mv /boot/kernel.old /boot/kernel
boot /boot/kernel/kernel
```

- V prípade zaregistrovanej poruchy systému (systém sa nedokáže naboťovať) sa pri bootovaní použije záložné jadro systému. Proces identifikácie poruchy je automatizovaný (používateľ si pri bootovaní nemá možnosť zvoliť, z ktorého kernel-u chce naboťovať systém).
- V súbore /var/log/messages sa nachádzajú informácie o úspešných / neúspešných bootovaniach systému.
- Ďalšie zálohovacie mechanizmy, ktoré by pokrývali iné moduly systému, neboli identifikované.

Adresárová štruktúra

Systémové adresáre sú rozvrhnuté tak, ako je to prezentované aj v [17]. Súbory a adresáre na SSD disku sú implementované nasledovne (adresár /data/, particia /dev/ada1s1):

```
# obrázkové reklamy - rozlíšenie 800x480 a 1024x600
drwxr-xr-x  3 root  wheel          4096 Dec  4  2013 ad
# manuálne vytvorené zálohy zo systému a iných neidentifikovaných
súborov
drwxr-xr-x  3 root  wheel          4096 Dec  4  2013 backup
# Content Management System (CMS) súbory - pravdepodobne
nastavenia generované pri formátovaní SSD
drwxr-xr-x  5 root  wheel          4096 May 17  22:10 cms
# aktuálne nastavenia servera, obrazoviek (aj "last good" dáta) a
súbory databázového systému
drwxr-xr-x  8 root  wheel          4096 Oct 13  22:59 funtoro
```

```
# nastavenia GPS
drwxr-xr-x  2 root  wheel          4096 Dec  4  2013 gps
# súbory domovskej stránky webového protálu
drwxr-xr-x  6 root  wheel          4096 Dec  4  2013 homepage
# obrázky
drwxr-xr-x  3 root  wheel          4096 Dec  4  2013 images
# úvodné video
drwxr-xr-x  2 root  wheel          4096 Dec  4  2013 introduction
# hudba
drwxr-xr-x  2 root  wheel          4096 Oct 25 18:11 music
# RSS nastavenia
drwxr-xr-x  2 root  wheel          4096 Dec  6  2013 rss
# nastavenia servera
-rw-r--r--  1 root  wheel           208 Dec 12  2013 server.conf
# videá (filmy)
drwxr-xr-x  3 root  wheel          4096 Dec  4  2013 videos
```

Logovanie

Výpis nastavení logovania (/etc/syslog.conf):

```
...
*.notice;authpriv.none;kern.debug;lpr.info;mail.crit;news.err
    /var/log/messages
...
```

Rotácia záznamov (/etc/newsyslog.conf):

```
# logfilename [owner:group] mode count size when
flags [/pid_file] [sig_num]
...
/var/log/messages          5    100    @0101T JC
...
```

Zoznam spustených procesov po štarte systému:

```
/etc/rc.d/cleanvar
/etc/rc.d/ip6addrctl
/etc/rc.d/devd
/etc/rc.d/ipfw
/etc/rc.d/newsyslog
/etc/rc.d/named
/etc/rc.d/rpcbind
/etc/rc.d/dmesg
/etc/rc.d/nfsd
/etc/rc.d/virecover
/usr/local/etc/rc.d/gpsd
/usr/local/etc/rc.d/mysql-server
/usr/local/etc/rc.d/bcastcmdserv
/usr/local/etc/rc.d/traplog
/usr/local/etc/rc.d/ectimer
/usr/local/etc/rc.d/announcement
/usr/local/etc/rc.d/taskld
/usr/local/etc/rc.d/svscan
/usr/local/etc/rc.d/oal
/usr/local/etc/rc.d/igmpproxy
/usr/local/etc/rc.d/gps_receiver
/etc/rc.d/sshd
```

```
/usr/local/etc/rc.d/apache24
/etc/rc.d/cron
/etc/rc.d/mixer
/etc/rc.d/gptboot
```

Kľúčové výsledky analýzy:

- V nastavení procesu syslogd sa nachádza aj záznam, ktorý sa stará o vytváranie systémových logov, ktoré zahŕňajú aj záznamy o súborovom systéme (/var/log/messages).
- Rotácia týchto záznamov je nastavená v súbore (/etc/newsyslog.conf).
- Daemon na zbieranie logov je ale vypnutý, pretože sa nenachádza medzi bežiacimi procesmi po štarte operačného systému (/etc/rc.d/ nastavenia).

Vzdialený prístup

Riešenie vzdialeného prístupu:

- V systéme sa musí nachádzať proces, ktorý inicializuje SSH spojenie smerom ku cloud-u (nedokázali sme ho konkrétne identifikovať) s parametrom reverzného počúvania na zadanom porte.

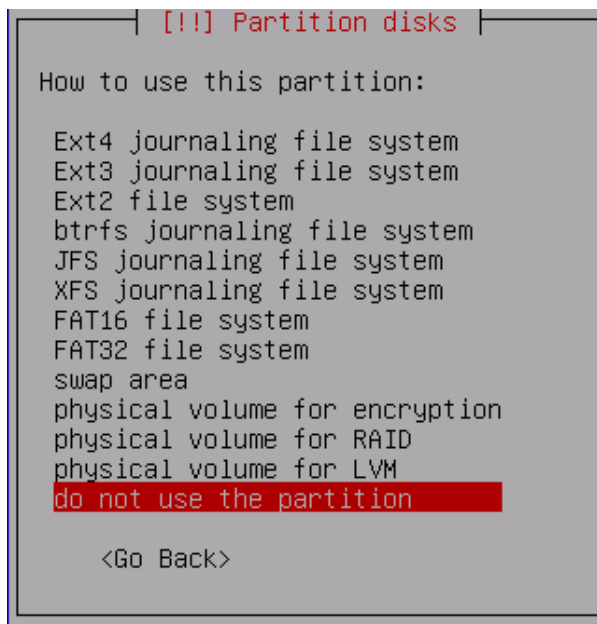
```
# na MOD serveri
ssh -R 10000:localhost:22 username@ip_address_of_cloud
# v cloude
ssh username@localhost -p 10000
```

- Súborový systém na serveri je možné vzdialene modifikovať aj pomocou http správ, cez ktoré sa MOD dopytuje na dostupné aktualizácie (/usr/local/etc/rc.d/apache24).

3.11.3 Analýza súborových systémov

Podpora

Natívne podporované súborové systémy na operačnom systéme Debian 8.6.0 i386 znázorňuje Obrázok 11.



Obrázok 11. Podporované súborové systémy pri inštalácii

Všetky súborové systémy podporované na operačnom systéme Debian, nielen pri inštalácii operačného systému [18]:

exfat, ext2, ext3, ext4dev, ext4, jfs, xfs, reiserfs, 9p, adfs affs, kafs, autofs4, autofs, befs, bfs, btrfs, cifs, coda, configfs, cramfs, dlm, ecryptfs, efs, fat, freevxfs, fuse, gfs2, hfs, hfsplus, hpfs, isofs, jbd2, jbd, jffs2, minix, msdos, ncpfs, nfs, ntfs(depreciated), ocfs2, omfs, qnx4, romfs, sysv, ubifs, udf, ufs, vfat

Ďalšia časť analýzy porovnáva najčastejšie používané súborové systémy **ext4, btrfs, jfs, xfs, fat32, ntfs, exfat** a **reiserfs**.

Porovnanie

Tabuľka 29 znázorňuje porovnanie zvolených súborových systémov z hľadiska funkcionality, ktorá je dôležitá pri riešenom projekte.

Tabuľka 29. Porovnanie súborových systémov z hľadiska funkcionality [19]

Funkcionalita	ext4	btrfs	jfs	xfs	reiserfs	fat32	exfat	ntfs
Maximálna veľkosť súboru	16 TiB	16 EiB	4 PiB	8 EiB	1 EiB	4 GiB	16 EiB	16 EiB
Windows podpora	nepriama	NIE	NIE	NIE	NIE	ÁNO	ÁNO	ÁNO
Ukladanie informácií o majiteľovi	ÁNO	ÁNO	ÁNO	ÁNO	ÁNO	NIE	NIE	ÁNO

Funkcionalita	ext4	btrfs	jfs	xf	reiserfs	fat32	exfat	ntfs
POSIX oprávnenia	ÁNO	ÁNO	ÁNO	ÁNO	ÁNO	NIE	ÁNO	ÁNO (addon)
Časové značky (vytvorenie, modifikovanie)	ÁNO	ÁNO	ÁNO	NIE	ÁNO	ÁNO	ÁNO	ÁNO
Access Control Lists (ACL)	ÁNO	ÁNO	ÁNO	ÁNO	ÁNO	NIE	ÁNO	ÁNO
Rozšírené atribúty	ÁNO	ÁNO	ÁNO	ÁNO	ÁNO	NIE	NIE	ÁNO
Hard/Soft linky	ÁNO	ÁNO	ÁNO	ÁNO	ÁNO	NIE	NIE	častočne
Journaling	ÁNO	ÁNO	iba metadáta	iba metadáta	ÁNO	NIE	NIE	iba metadáta
Logovanie zmien	NIE	NIE	NIE	ÁNO	NIE	NIE	NIE	ÁNO
Rozširovanie systému	online aj offline	iba online	iba offline	iba online	online aj offline	nie je možné	nie je možné	online aj offline
Orezávanie systému	iba offline	iba online	nie je možné	nie je možné	iba offline	nie je možné	nie je možné	online aj offline
Sparse súbory	ÁNO	ÁNO	ÁNO	ÁNO	ÁNO	iba pri kompr.	iba pri kompr.	ÁNO
Bloková subalokácia	NIE	ÁNO	ÁNO	NIE	ÁNO	NIE	NIE	NIE

Windows podpora – Súborový systém je možné namountovať v operačných systémoch Microsoft Windows natívne alebo pomocou stiahnutého ovládača.

POSIX oprávnenia – Štandardné Linux / Unix oprávnenia pre čítanie, zapisovanie a vykonávanie súborov pre vlastníka / skupinu vlastníka / ostatných používateľov (rwx model).

Journaling – Transakcie na úrovni súborového systému (podobne ako databázový systém) – v prípade zlyhania napájania disku sa pri štarte operačného systému vykoná rollback, takže je minimalizované riziko poškodenia súborového systému (pre tento projekt dôležité). Dva typy journaling-u: journaling na úrovni metadát (hlavičky súborov a ukazovatele na dáta) a journaling na úrovni celých blokov (aj dáta súborov).

Sparse súbory – Prázdne súbory alebo súbory, ktoré obsahujú postupnosť rovnakých znakov sa automaticky komprimujú na úrovni súborového systému (výsledok – zaberajú menej pamäťového priestoru).

Metrika pre vyhodnocovacia maticu:

- Medzi najdôležitejšie funkcionality patrí: Windows podpora, POSIX oprávnenia, časové značky a journalling. Týmto funkcionalitám je priradená váha 10. Ostatné funkcionality majú priradenú váhu 1.
- Alokácia hodnoty pre jednu bunku – žiadna funkcionality (NIE) = 0, polovičná alebo nepriama funkcionality = 0.5 a úplná funkcionality = 1.
- Maximálna veľkosť súboru $\leq 4\text{GiB}$ \rightarrow ohodnotenie 0, inak ohodnotenie 1.

Tabuľka 30 zobrazuje porovnanie súborových systémov pomocou praktických testov, ktoré sú opísané v [20], [21], [22] a [23]. Opis testov a nástrojov, ktoré sa použili je zachytený v Tabuľka 31 (žltou farbou sú zvýraznené bunky sa najlepšimi hodnotami) [20], [21], [22] a [23].

Tabuľka 30. Porovnanie súborových systémov na základe testov [20], [21], [22] a [23]

tst	ext4	btrfs	jfs	xfs	reiserfs	fat32	exfat	ntfs
1	186.31	93.65	-	215.61	-	-	-	-
2	276.08	167.08	-	277.39	-	-	-	-
3	429.26	497.77	-	430.49	-	-	-	-
4	203.08	236.83	-	200.27	-	-	-	-
5	353.2	396.4	-	354.83	-	-	-	-
6	211.77	237.8	-	209.2	-	-	-	-
7	290.33	145.01	-	240.84	-	-	-	-
8	4807	4238	-	5068	-	-	-	-
9	35.14	145.27	49	57.75	49.65	-	-	-
10	83.7	88.73	262.9	90.77	66.13	-	-	-
11	167.47	170.1	284.5	175.7	120.8	-	-	-
12	83.87	88	263.47	89.93	64.8	-	-	-
13	337.35	355.39	242.28	333.53	269.84	-	-	-
14	229.42	140.46	126.45	212.58	114.14	-	-	-
15	-	-	-	-	-	54.77	56.63	56.67
16	-	-	-	-	-	12343	22332	30211
17	-	-	-	-	-	7028	7189	7189
18	-	-	-	-	-	7754	22332	22377
19	-	-	-	-	-	21.6	24.1	24.2
20	-	-	-	-	-	58	103.7	101.9

tst	ext4	btrfs	jfs	xfs	reiserfs	fat32	exfat	ntfs
21	-	-	-	-	-	253.2	513.3	512.1
22	-	-	-	-	-	383.9	389.7	391.7
23	65.94	132.58	-	62.3	-	-	-	47.45
24	215520	192625	-	194166	-	-	-	137566
25	320.8	280.67	-	284.07	-	-	-	119.13
26	550031	1721568	-	2188248	-	-	-	51603

Tabuľka 31. Opis použitých testovacích nástrojov a postupov [20], [21], [22] a [23]

tst	Testovací nástroj	Špecifikácia testu	Disk
1	Flexible IO Tester v2.1.13	Type: Random Write - IO Engine: POSIX AIO - Buffered: Yes - Direct: No - Block Size: 4KB - Disk Target: Default Test Directory - Result: MB/s	240GB OCZ VERTEX3
2	Flexible IO Tester v2.1.13	Type: Random Read - IO Engine: POSIX AIO - Buffered: Yes - Direct: No - Block Size: 128KB - Disk Target: Default Test Directory - Result: MB/s	240GB OCZ VERTEX3
3	Flexible IO Tester v2.1.13	Type: Sequential Read - IO Engine: POSIX AIO - Buffered: Yes - Direct: No - Block Size: 4KB - Disk Target: Default Test Directory - Result: MB/s	240GB OCZ VERTEX3
4	FS-Mark v3.3	Test: 1000 Files, 1MB Size, MB/s	240GB OCZ VERTEX3
5	FS-Mark v3.3	Test: 5000 Files, 1MB Size, 4 Threads, MB/s	240GB OCZ VERTEX3
6	FS-Mark v3.3	Test: 4000 Files, 32 Sub Dirs, 1MB Size, MB/s	240GB OCZ VERTEX3
7	Compile Bench v0.6	Test: Initial Create, MB/s	240GB OCZ VERTEX3
8	PostMark v1.51	Disk Transaction Performance, TPS	240GB OCZ VERTEX3
9	Flexible IO Tester v2.1.11	Test: Intel IOMeter File Server Access Pattern - Disk Target: Default Test Directory [sec]	128GB Crucial_CT128MX1
10	FS-Mark v3.3	Test: 1000 Files, 1MB Size, files/sec	128GB Crucial_CT128MX1
11	FS-Mark v3.3	Test: 5000 Files, 1MB Size, 4 Threads, files/sec	128GB Crucial_CT128MX1
12	FS-Mark v3.3	Test: 4000 Files, 32 Sub Dirs, 1MB Size, files/sec	128GB Crucial_CT128MX1
13	Compile Bench v0.6	Test: Compile, MB/s	128GB Crucial_CT128MX1
14	Compile Bench v0.6	Test: Initial Create, MB/s	128GB Crucial_CT128MX1
15	PCMark 7	Application Loading, MB/s	Samsung 830, 256 GB
16	IOMeter 2006.07.27	Database benchmark pattern, average I/O operations/second	Samsung 830, 256 GB
17	IOMeter	Web server workload, average I/O operations/second	Samsung 830, 256 GB

tst	Testovací nástroj	Špecifikácia testu	Disk
	2006.07.27		
18	IOMeter 2006.07.27	Workstation workload patterns, average I/O operations/second	Samsung 830, 256 GB
19	AS SSD 4K Random	Read, MB/s	Samsung 830, 256 GB
20	AS SSD 4K Random	Write, MB/s	Samsung 830, 256 GB
21	AS SSD Sequential Troughput	Read, MB/s	Samsung 830, 256 GB
22	AS SSD Sequential Troughput	Write, MB/s	Samsung 830, 256 GB
23	SQLite v3.8.10.2	Test Target: Default Test Directory, sec	Samsung SSD 950 PRO 256GB
24	Flexible IO Tester v2.1.13	Type: Random Read - IO Engine: Libaio - Buffered: No - Direct: Yes - Block Size: 4KB - Disk Target: Default Test Directory - Result: IOPS	Samsung SSD 950 PRO 256GB
25	FS-Mark v3.3	Test: 5000 Files, 1MB Size, 4 Threads, files/sec	Samsung SSD 950 PRO 256GB
26	BlogBench v1.0	Test: Read, final score	Samsung SSD 950 PRO 256GB

Metrika pre vyhodnocovaci maticu:

- Založená na sekvenčnom porovnávaní blokov podobných testov. Jednotlivé bloky testov sa následne porovnávajú medzi sebou pomocou referenčných súborových systémov, ktoré sa nachádzajú vo viacerých blokoch.
- Výsledkom je poradie súborových systémov od 10 po 3 (väčšia hodnota je lepšia).

Vyhodnotenie na základe navrhnutých metrick znázorňuje Tabuľka 32. Žltou farbou sú zvýraznené bunky s najvyššími hodnotami.

Tabuľka 32. Vyhodnotenie súborových systémov

Funkcionalita	ext4	btrfs	jfs	xfs	reiserfs	fat32	exfat	ntfs
Funkcionalita	38	38	32.5	22.5	29.5	20.5	32.5	34
Výkon	9	8	10	7	6	3	4	5

Návrh

Výber súborových systémov (základná štruktúra Debian inštalácie s android adresárom):

- / – Na základe porovnania (Tabuľka 32) vyplýva, že je potrebné analyzovať tri súborové systémy – ext4, btrfs a jfs. Pri btrfs je častokrát problematická podpora a nestabilita, ktorá v prípade serverového použitia je kritická. Ext4 zbytočne

vytvára vyššiu záťaž na CPU a má väčšie pamäťové nároky kvôli úplnému journaling-u, preto nie je ideálne ho použiť pri obmedzených pamäťových nárokoch. Navyše, MOD server má veľmi dobré vlastnosti napájania – existuje veľmi nízka pravdepodobnosť okamžitého výpadku servera a porušenia súborového systému. Z tohto dôvodu sme sa rozhodli pre súborový systém **jfs**.

- **/android** – Dôležité je rýchle kopírovanie súborového systému na obrazovky – **jfs** dosahuje na flash médiách lepšie výsledky ako ostatné súborové systémy (Tabuľka 32).
- **/tmp** – Nie je potrebné sa starať o konzistenciu súborov na vysokej úrovni a nie je potrebná široká podpora vlastností súborového systému - použitie **jfs**.
- **/home** – Rovnako ako /tmp – **jfs**.
- **/var** – Pri logovacích a ďalších variabilných záznamoch je dôležitým aspektom častý zápis a čítanie zo súborov – ak by sme sa riadili podľa odporúčaní v [24], tak by bol jednoznačne zvolený súborový systém reiserfs. V operačnom systéme Debian však podpora k tomuto súborového systému skončila (je ho možné pripojiť, ale vývoj ovládačov bol ukončený). Jfs je druhým najstabilnejším súborovým systémom, ak je potrebné pracovať naraz s veľkými aj malými súbormi (bez priameho určenia), ktoré sa častokrát menia – použitie **jfs**.
- **swap** – Na swapovací priestor sa použije „SwapSpace“ nástroj, ktorý umožňuje dynamicky alokovať voľný priestor na disku pre swapovacie účely.
- **SSD dáta** – Je potrebná vysoká úroveň zachovania konzistencie pri výpadku servera, vysoký výkon pri čítaní a multiplatformová podpora – rozhodli sme sa pre súborový systém **ext4**. Ext4 má open-source podporu na operačných systémoch Windows, dôležitejšie ale je, že v testoch ďaleko preyšuje súborové systémy fat32, exfat a ntfs.

Návrh rozloženia priestoru na disku:

- Pomer $/: /android : /tmp : /home : /var = 928\text{MiB} \cdot a : 512\text{MiB} : 64\text{MiB} : 32\text{MiB} : 512\text{MiB}$. Z toho vyplýva, že 2GiB disk sa rozdelí postupne na partície s veľkosťami 928MiB, 512MiB, 64MiB, 32MiB a 512MiB.
- Koeficient $a = \sqrt{\frac{DISK\ SIZE}{2048}}$, zaručuje, že pri väčšej kapacite disku sa bude veľkosť root partície rýchlejšie navyšovať ako pri ostatných partiáciách.

Návrh LVM (Logical Volume Manager):

- Vytvorená ďalšia abstrakcia nad základným súborovým systémom.
- Výhody – Väčšia flexibilita pri tvorbe logických partiícií a zmene veľkosti týchto partiícií za behu operačného systému v prípade potreby – rozhodli sme sa pre implementáciu LVM.

3.11.4 Analýza a návrh mechanizmu na zálohovanie

Požiadavky

Rozhodujúce požiadavky a obmedzenia na zálohovanie častí súborového systému:

- Veľkosť integrovanej pamäti je obmedzená na približne 2 GiB – je neuskutočniteľné vytvárať kompletnú zálohu a ukladať ju na integrovaný disk

(veľkosť orezaného systému Debian je približne 300 – 400 MiB, ale je potrebné počítať s Android systémom, aplikáciami a záznamami).

- Zákazníci si neželajú, aby sme používali SSD disk na ukladanie záloh operačného systému (SSD disky sú drahé vzhľadom na zakúpený priestor).
- Zálohy je stále potrebné vykonávať aj napriek kontinuálnom riadení inštalácie aktualizácií – môže nastať kolaps integrovanej pamäti. Je potrebné si uchovávať zálohu celého operačného systému, aby sme mohli obnoviť systém v prípade potreby.
- Požiadavka na Dual-Boot mechanizmus, ktorý zahŕňa zálohovanie súborov jadra operačného systému.

Dual-Boot mechanizmus

Analýza

GRUB (GNU GRand Unified Bootloader) podporuje Dual-Boot mechanizmus pre neobmedzený počet zreťazených bootovacích záznamov, ktoré môžu predstavovať jeden operačný systém alebo iba oddelené jadro operačného systému. V prípade zlyhania naboovania prvého jadra operačného systému, bootloader sa pokúsi naboovať z druhého jadra systému – fallback riešenie.

Pred nastavením GRUB je potrebné vytvoriť záložnú kópiu jadra systému (záloha):

- Nakopírovanie súborov pod novú verziu:
 - /boot/vmlinuz-[version number]
 - /boot/initrd.img-[version number]
 - /lib/modules/[version-number]/

Výhodou tohto prístupu je jednoduchosť, prípadne možnosť vytvorenia priamej kópie súčasného jadra systému. Nevýhodou je nízka konfigurovateľnosť.

- Kompilácia nové jadra systému zo zdrojového kódu pomocou balíkov fakeroot, kernel-package a linux-source-2.6. Výhodou tejto možnosti je voľba modulov a ovládačov, ktoré sa budú nachádzať v novom kernel-y, čo môže viesť k zníženiu veľkosti záložného jadra. Nevýhodou je časová náročnosť na porozumenie procesu kompilácie [25].
- Inštalácia kernel-u pomocou príkazov apt-cache search linux-image a sudo apt-get install linux-image-flavour (v tomto prípade ale nie je možné použiť rovnakú verziu kernel-u). Výhodou tejto možnosti je automatické pridanie GRUB záznamov do /boot/grub/grub.conf súboru a jednoduchosť (dve príkazy). Nevýhodou inštaláčného prístupu je nízka konfigurovateľnosť nové jadra systému.

Príklad (GRUB v1, pri GRUB v2 je syntax podobná) – súbor /etc/default/grub.cfg [26]:

```
default saved
timeout 10
fallback 1 2

title A
root (hd0,0)
kernel /kernel
savedefault fallback
```

```

title B
root (hd1,0)
kernel /kernel
savedefault fallback

title C
root (hd2,0)
kernel /kernel
savedefault

```

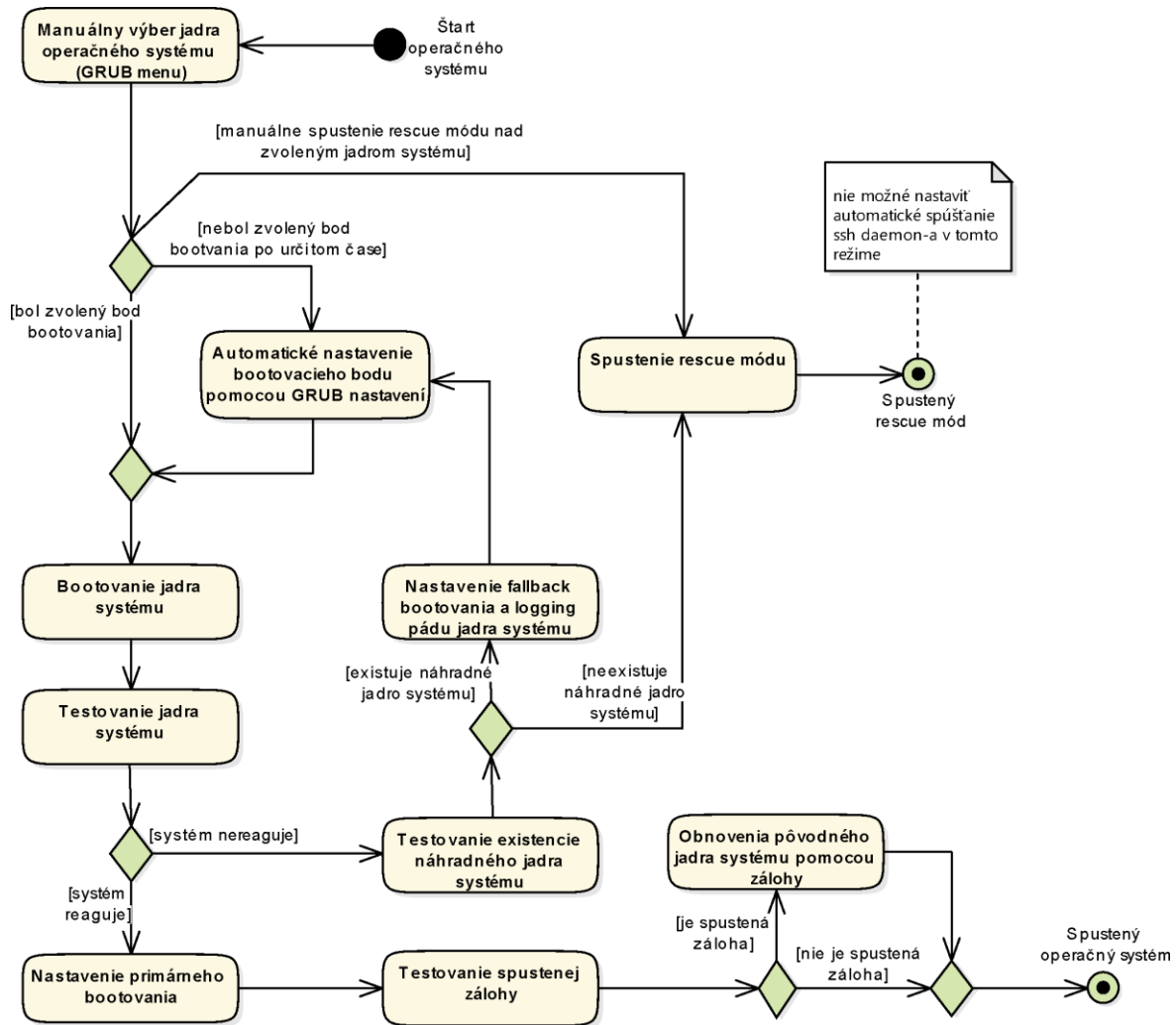
Riadok „default saved“ určuje, že na bootovanie sa má prioritne použiť uložené bootovacie nastavenie, ktoré sa vytvára za behu operačného systému. Riadok „timeout 10“ určuje dobu, počas ktorej sa má pred bootovaním systému zobrazovať ponuka na zmenu systému (v sekundách). Riadok „fallback 1 2“ definuje, že k bootovaciemu záznamu s indexom 0 („A“) sa má použiť fallback s indexom 1 („B“) a k bootovaciemu záznamu s indexom 1 („B“) sa má použiť fallback s indexom 2. Každý bootovací záznam je definovaný názvom („title“), koreňovým zariadením alebo partíciou („root“), cestou ku kernel-u („kernel“) a automatickým nastavením aktuálneho systému ako predvoleného systému na bootovanie („savedefault fallback“) (v opačnom prípade by sa proces bootovania nemohol dostať k tretiemu záznamu).

Problémom je, že ak sa poradie bootovania dostane k systému „B“, ale medzitým sa systém „A“ opraví, poradie bootovania sa automaticky neaktualizuje. Je potrebné v štartovacom skripte zadefinovať, aby sa systém vždy pokúšal bootovať systém zo záznamu s indexom 0 – príkaz „grub-set-default 0“ vo vytvorenom skripte v adresári /etc/init.d/.

Po aktualizácii súboru /etc/default/grub.cfg je potrebné aplikovať vytvorené nastavenia pomocou príkazu „update-grub“. Zabezpečenie GRUB proti neoprávnenému prístupu do single-user módu je možné pomocou príkazu password v GRUB nastaveniach (autentifikácia append módu).

Návrh

Obrázok 12 zobrazuje návrh automatizovaného procesu bootovania operačného systému v prostredí GRUB. Tento proces je založený na detekcii zlyhania jadra operačného systému a automatickom nasadení záložného systému so zaznamenaním zlyhania.



Obrázok 12. Návrh Dual-Boot mechanizmu

Prístup do GRUB nastavení pred bootovaním systému bude autentifikované pomocou nastaveného hesla (ochrana proti neoprávnenému zásahu do systému). Na serveri sa bude nachádzať jedno hlavné jadro a jedno záložné jadro systému. Obe jadrá systému budú podporovať aj rescue mód.

Záloha operačného systému

Obmedzenia, ktoré je potrebné brať do úvahy pri zálohovaní operačného systému

- Zálohy nie je možné ukladať na žiaden s diskov, ktoré sa nachádzajú na serveri.
- Kontinuálne zálohy cez sieť nie je možné realizovať kvôli obmedzenej sieťovej kapacite na strane cloud-u.
- Stále je potrebné uchovávať zálohu operačného systému na oddelenom mieste – server sa môže poškodiť vrátane celého súborového systému.

Dôsledky opísaných obmedzení

- Nie je reálne používať kontinuálne zálohovacie systémy (napr. inkrementálne zálohy), pretože cloud by sa zahltil údajmi.

- Automatická obnova celého súborového systému je nereálna – ak zlyhá celý operačný systém a nepomôže nasadenie záložného jadra systému, je potrebné interagovať so systémom lokálne (lokálne spustiť rescue mode) – stráca sa význam obnovy systému cez sieť.
- Zlyhanie pôvodného jadra systému je možné automaticky vyriešiť skopírovaním záložného jadra systému a vzdialene cez ssh protokol.

Návrh

- V prostredí cloud-u sa budú testovať nasadenia aktualizácií a nového software-u – na cloude sa budú vždy nachádzať najnovšie verzie systémov.
- Cloud bude riadiť aktualizovanie softvérových balíkov a systému na serveroch pomocou ssh protokolu – cloud má prehľad o nainštalovaných softvérových moduloch a teda môže v prípade potreby spätne rekonštruovať celý súborový systém.
- Lokálna inštalácia operačného systému je vždy nutná!

Záloha softvérových balíkov

Analýza

- Linux operačné systémy neposkytujú možnosť na automatický rollback softvérových balíkov na predchádzajúcu verziu bez vytvorenia obrazu celého súborového systému.
- Je ale možné čítať históriu vykonaných aktualizácií softvérových balíkov pomocou logovacieho súboru /var/lib/dpkg/info/. Pomocou tohto súboru je možné identifikovať posledné verzie cieľového balíka a nainštalovať predposlednú verziu (nie je potrebné najprv odinštalovať predposlednú verziu). Z toho vyplýva, že celý rollback proces je možné automatizovať.
- Príklad (v Debian je to rovnaké, pretože používajú rovnakého správcu balíkov) []:

```
find /var/lib/dpkg/info/ -name \*.list -mtime -3 | sed
's#\.list$##;s#.#/##'
sudo apt-cache policy PROGRAM:
*** 3.6.7+build3+nobinonly-0ubuntu0.10.04.1 0
      500 http://de.archive.ubuntu.com/ubuntu/ lucid-
updates/main Packages
      500 http://security.ubuntu.com/ubuntu/ lucid-
security/main Packages
      100 /var/lib/dpkg/status
      3.6.3+nobinonly-0ubuntu4 0
sudo apt-get install PROGRAM=3.6.3
```

- Rollback softvérových balíkov sa musí minimalizovať – na strane cloud-u je potrebné testovať inštalácie nových verzií pred ich reálnym nasadením. Je potrebné brať do úvahy aj rôznorodosť inštalácií serverov (rôzne servery budú mať rôzne požiadavky a obmedzenia na balíky).

Návrh

- Na strane cloud-u – Analýza dostupných aktualizácií a ich zaradovanie do skupín, ktoré predstavujú implementácie serverov (napr. niektoré hardwarové špecifikácie serverov nebudú podporovať novú aktualizáciu).
- Na strane servera – Dopytovanie sa na server a kontrola dostupnosti nových aktualizácií. Naprogramovanie aplikácie, ktorá automaticky vykoná rollback zvoleného softvérového balíka na predposlednú verziu softvérového balíka, ak sa vyskytne nejaký problém.

Návrh adresárovej štruktúry a lokálnych spravovacích mechanizmov

Základná adresárová štruktúra systému vytvorená po inštalácii operačného systému Debian 8.6.0 i386 je zobrazená na Obrázok 13. Opis tejto adresárovej štruktúry je v Tabuľka 33.

```

root@multimedia-server:/# ls -lh
total 76K
drwxr-xr-x  2 root root 256 Nov 25 19:47 android
drwxrwxr-x  2 root root 4.0K Nov 25 19:53 bin
drwxr-xr-x  3 root root 4.0K Nov 25 19:56 boot
drwxr-xr-x  3 root root 4.0K Nov 25 19:47 data
drwxr-xr-x 18 root root 3.2K Nov 25 19:56 dev
drwxr-xr-x 63 root root 8.0K Nov 25 19:56 etc
drwxr-xr-x  3 root root  8 Nov 25 19:55 home
lrwxrwxrwx  1 root root  33 Nov 25 19:49 initrd.img -> /boot/initrd.img-3.16.0-4-686-pae
drwxr-xr-x 14 root root 4.0K Nov 25 19:53 lib
drwxr-xr-x  3 root root 16 Nov 25 19:47 media
drwxr-xr-x  2 root root  1 Nov 25 19:48 mnt
drwxr-xr-x  2 root root  1 Nov 25 19:48 opt
dr-xr-xr-x 187 root root  0 Nov 25 19:56 proc
drwx-----  2 root root 16 Nov 25 19:48 root
drwxr-xr-x 13 root root 480 Nov 25 19:56 run
drwxr-xr-x  2 root root 8.0K Nov 25 19:56/sbin
drwxr-xr-x  2 root root  1 Nov 25 19:48 srv
dr-xr-xr-x 13 root root  0 Nov 25 19:58 sys
drwxrwxrwt  7 root root  40 Nov 25 19:56 tmp
drwxr-xr-x 10 root root  64 Nov 25 19:48 usr
drwxr-xr-x 11 root root 4.0K Nov 25 19:48 var
lrwxrwxrwx  1 root root  29 Nov 25 19:49 vmlinuz -> boot/vmlinuz-3.16.0-4-686-pae

```

Obrázok 13. Základná štruktúra súborového systému po inštalácii systému (/)

Tabuľka 33. Opis základných adresárov

Adresár	Opis
/android	Súbory operačného systému Android, z ktorých sa tento systém bootuje.
/bin	Základné aplikácie operačného systému pre všetkých používateľov (su, ls, mv ...) a špeciálne bootovacie a „rescue“ súbory.
/boot	Kernel obraz, initrd obraz, grub súbory a nastavenia (všetko pre bootovanie).
/data	Namontovaný obsah SSD disku pre multimediálne dáta.
/dev	Súbory predstavujúce pripojené zariadenia.
/etc	Nastavenia aplikácií a služieb operačného systému.
/home	Lokálne dáta používateľa.

Adresár	Opis
/lib	Knižnice potrebné pre beh operačného systému a aplikácií, ktoré sú umiestnené v adresári.
/media	Mountovací bod pre USB úložiská alebo napr. CD/DVD-ROM.
/mnt	Ďalšie dočasne namountované súborové systémy.
/opt	Ďalšie aplikácie (stiahnuté, skompilované).
/proc	Aktuálny stav procesov, pamäte, siete a ďalších real-time prostriedkov.
/root	Domovský adresár pre root používateľa (iba pre root).
/run	Pamäťový priestor pre služby jadra systému načítané pri bootovaní systému.
/sbin	Systémové nástroje pre správu operačného systému.
/srv	Serverovo špecifické dáta (napr. súbory webovej stránky)
/sys	Súborový systém, ktorý je používaný jadrom systému.
/tmp	Dočasné súbory aplikácií.
/usr	Môže obsahovať ďalšie aplikácie, knižnice a jadro systému.
/var	Meniace sa dáta (logy, databázové súbory, linky na balíky).

Návrh adresárovej štruktúry na SSD disku (Tabuľka 34 obsahuje opis adresárov zobrazených v nasledujúcom bloku):

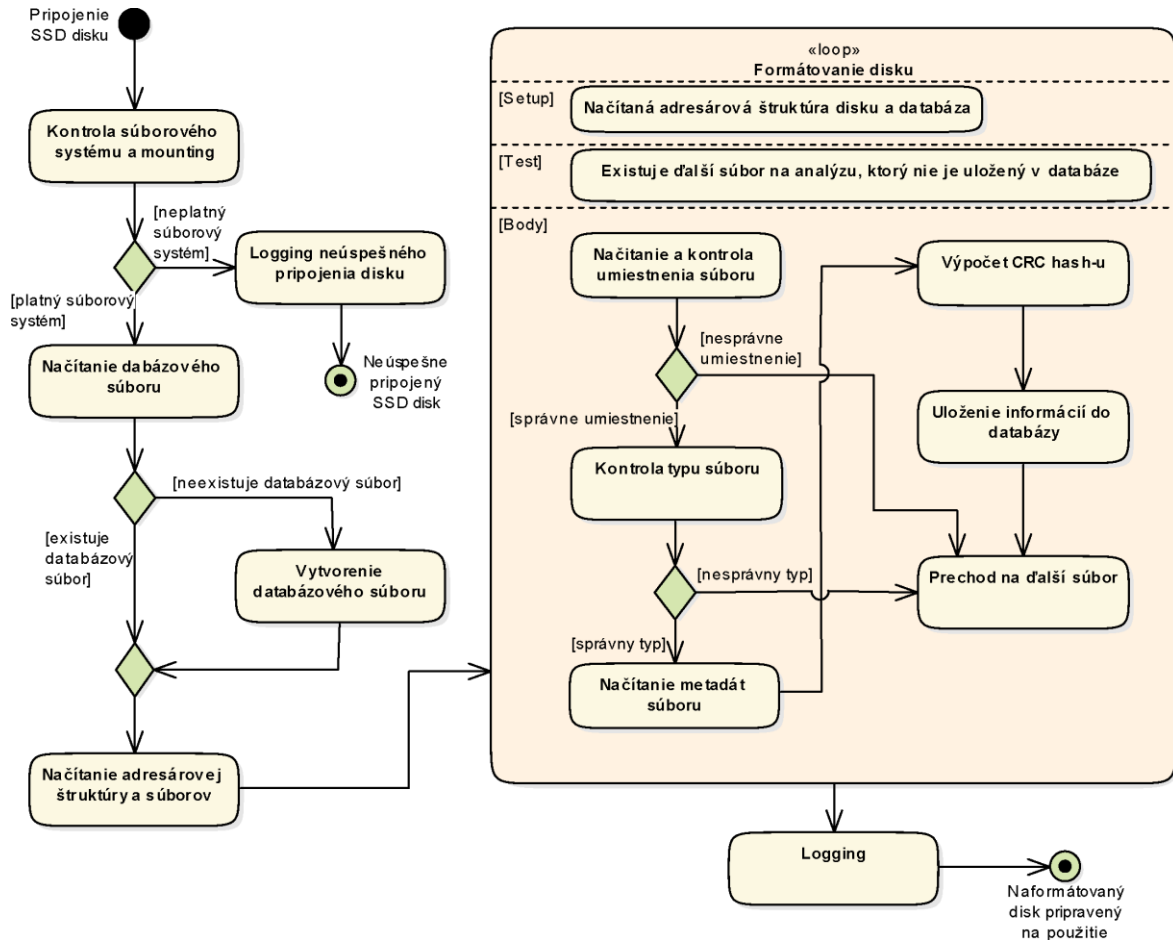
<pre> / /images /album01/... /album02/... ... /videos /videos/movies/... /videos/movies/movie01/... /videos/movies/movie02/... /videos/intro/... /videos/adverts/... /music /album01/... /album02/... /docs/... </pre>
--

Tabuľka 34. Opis adresárov na SSD disku

Adresár/súbor	Opis
/images	Obrázkové materiály, ktoré môžu byť zobrazované na obrazovkách. Môže obsahovať ďalšie podadresáre charakterizujúce albumy. Tieto adresáre môžu

Adresár/súbor	Opis
	obsahovať aj textové súbory opisujúce celý album („album.txt“) a náhľady albumov („previewXX.jpg“, kde XX je poradové číslo náhľadu). Preferovaným formátom je jpg a rozlíšenie pod 2048x1536.
/videos/movies	Adresáre, ktoré obsahujú mp4 súbory a jpg náhľady na filmy („previewXX.jpg“, kde XX je poradové číslo náhľadu). Formát (codec) filmov bol navrhnutý v kapitole „Multimediálny obsah“ (mp4).
/videos/intro	Úvodné video a bezpečnostné pokyny. Rovnaký codec ako pri movies.
/videos/adverts	Rôzne typy reklám. Podporované a opísané formáty reklám sa nachádzajú v kapitole „Reklamy“.
/music	Adresár music môže obsahovať ďalšie podadresáre reprezentujúce rôzne albumy. Preferovaným formátom je mp3 s bitovou šírkou 192 kbps. Ďalej sa tu môžu nachádzať aj cover-y albumu („coverXX.jpg“, kde XX je poradové číslo náhľadu).
/docs	Dokumenty vo formáte pdf.

Obrázok 14 zobrazuje proces načítania pripojeného SSD disku a aktualizácie lokálnej databázy na SSD disku (základné kroky, ktoré sú postavené na analýze typu súboru a jeho metadát vzhľadom na správne umiestnenie súboru). Samozrejmosťou je logovanie pripojenia a odpojenia súborového systému pri úspešnom aj neúspešnom pripojení. Databázový súbor je uložený na SSD disku.



Obrázok 14. Proces pripojenia SSD disku

Zabezpečenie adresárovej štruktúry

Súčasná riešenie:

- Jeden používateľ (root).
- Prepínanie prístupu k adresárovej štruktúre z read-only na read-write alebo opačne v prípade potreby (napr. prístup z cloud-u za účelom modifikácie konfiguračných súborov).
- Identifikované nevýhody takéhoto riešenia:
 - Pri vydatenom útoku Man-in-the-Middle útočník získa prístup k celému súborovému systému prostredníctvom spúšťania ľubovoľných príkazov v rámci platných príkazov deklarovaných v príkaze.
 - Žiadne oddelenie právomocí medzi používateľmi a takisto medzi spúšťanými procesmi (procesy si môžu navzájom modifikovať súbory).
 - Konfigurácia kvót pre procesy a používateľov je takmer nemožná.

- Multifaktorová autentifikácia pri vzdialenom prístupe chýba, pri SSH pripojení sa používateľ musí priamo prihlásovať ako root a aj vykonávať všetky príkazy ako root.

Možné spôsoby zabezpečenia adresárovej štruktúry:

- POSIX oprávnenia – Založená na priradovaní oprávnení pre jednotlivé súbory a adresáre vzhľadom na vlastníka, skupinu a ostatných používateľov. Základné oprávnenia zahŕňajú čítanie, zápis a spúšťanie (rwx).
- Rozšírené atribúty súborov – Umožňujú uzamykanie súborov na základe viacerých atribútov ako pri POSIX oprávneniach (napr. oprávnenie pre mazanie súboru, oprávnenie pre dopĺňanie do súboru).
- Prístupové listy (ACL) – Umožňujú priradovať oprávnenia s väčšou granularitou ako je to pri POSIX oprávneniach (napr. oprávnenie spúšťať súbor pre dvoch rôznych používateľov, ktorí nie sú v rovnakej skupine, pre ostatných používateľov stále musí byť prístup minimálny).
- Kvóty LVM partícií – Maximálna veľkosť súborov, maximálny obsadený priestor z partície konkrétnym používateľom a ďalšie kvóty.

Návrh riešenia:

- Viacero používateľov (nie iba root).
- Rozdelenie používateľov pre niektoré procesy, aby ich adresný priestor na disku bol izolovaný iba pre súbory, ku ktorým musia mať ohraničený prístup.
- Aspoň jeden používateľ pre vzdialený prístup (nie priamo root) – viacfaktorová autentifikácia. Priame prihlásenie do root-a je umožnené iba z konzolového pripojenia.
- Riadenie vzdialeného prístupu pomocou firewall-u a priamo pomocou procesov, ktoré to umožňujú (napr. sshd).

3.12 Požiadavky na systém

3.12.1 Diagram prípadov použitia pre cestujúceho

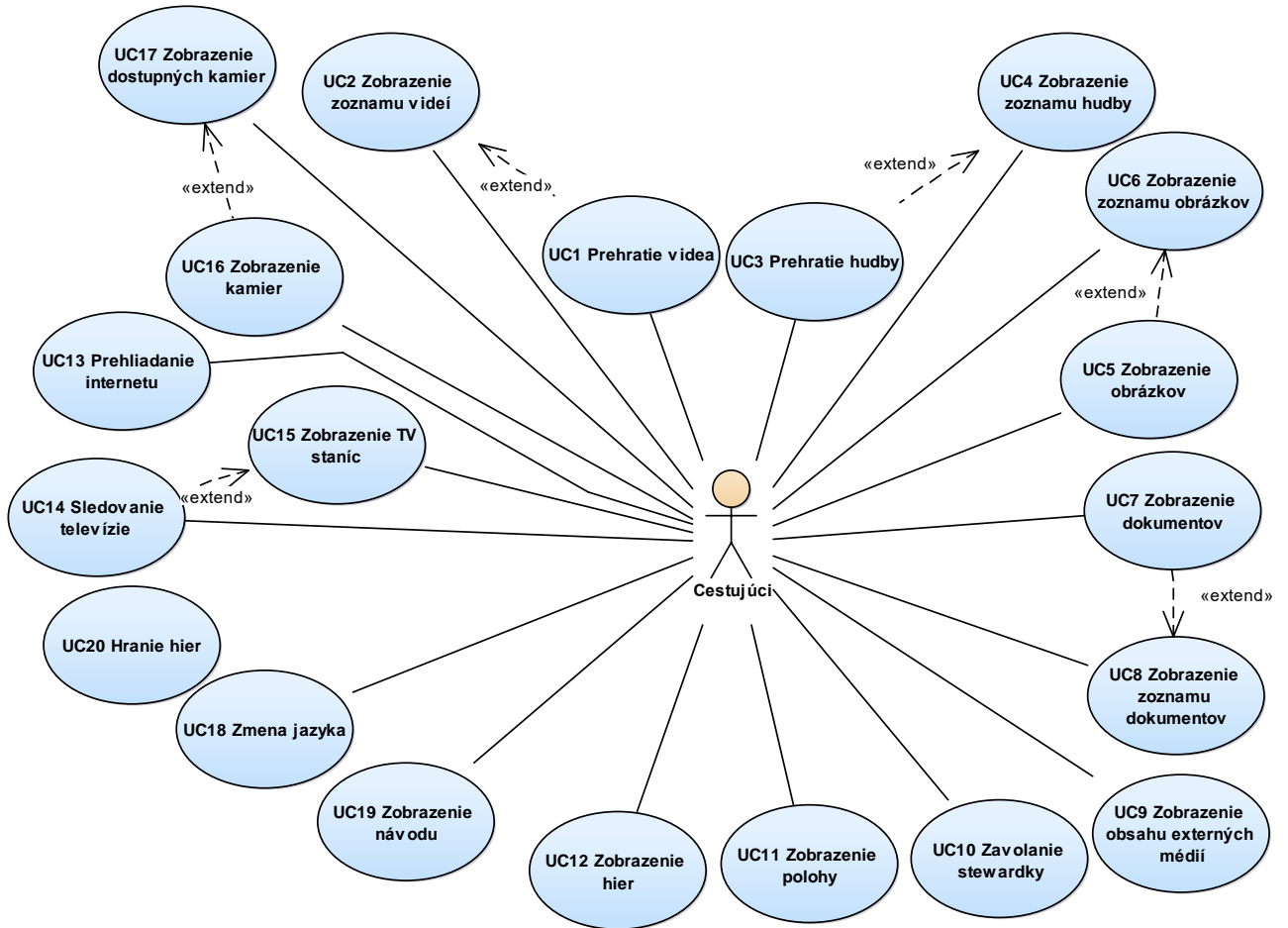



Figure 1: Cestujúci-model




UC1 Prehratie videa

SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. Používateľ klikne na tlačidlo prehrať video. 2. Systém spustí prehrávanie videa. Alternate: 2a. Ukončenie skor 3. Systém ukončí prehrávanie videa.
<p> Alternate. Ukončenie skor</p> <ol style="list-style-type: none"> 1. Používateľ ukončí prehrávanie videa počas jeho prehrávania.



UC10 Zavolanie stewardky

SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. Cestujúci stlačí tlačidlo obsluhy v hlavnom menu. 2. Systém zobrazí informáciu o zavození stevardky. 3. Systém zašle notifikáciu na stevardskú obrazovku s číslom sedadla.

UC11 Zobrazenie polohy


SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. Cestujúci stlačí tlačidlo mapy na domovskej obrazovke. 2. Systém zobrazí aktuálnu polohu autobusu na mape. <ul style="list-style-type: none"> Alternate: 2a. Cestujúci stlačí tlačidlo priblíženia Alternate: 2b. Cestujúci stlačí tlačidlo oddialenia. 3. Cestujúci stlačí tlačidlo návratu. 4. Systém zobrazí domovskú obrazovku.
<p> Alternate. Cestujúci stlačí tlačidlo oddialenia.</p> <ol style="list-style-type: none"> 1. Systém oddiali mapu.
<p> Alternate. Cestujúci stlačí tlačidlo priblíženia</p> <ol style="list-style-type: none"> 1. Systém priblíži mapu.

UC12 Zobrazenie hier





SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. Cestujúci stlačí obrázok hier na domovskej obrazovke. 2. Systém zobrazí zoznam dostupných hier. <ul style="list-style-type: none"> Alternate: 2a. Cestujúci klikne na vybranú hru v zozname. 3. Cestujúci stlačí tlačidlo návratu. 4. Systém zobrazí domovskú obrazovku.
<p> Alternate. Cestujúci klikne na vybranú hru v zozname.</p> <ol style="list-style-type: none"> 1. Systém zobrazí informácie o vybranej hre. 2. Cestujúci stlačí tlačidlo späť.

UC13 Prehliadanie internetu



SCENARIOS

SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. Používateľ stlačí tlačidlo s názvom Internet 2. Systém otvorí nové okno prehliadača na prehliadanie internetu 3. Používateľ stlačí tlačidlo s názvom Späť 4. Systém sa vráti do hlavného menu

UC14 Sledovanie televízie

SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. Používateľ vyberie jednu z dostupných TV staníc na sledovanie 2. Systém zobrazí popis k danej TV stanici 3. Používateľ stlačí tlačidlo s názvom Sledovať <ul style="list-style-type: none"> Alternate: 3a. návrat 1 Alternate: 3b. návrat 2 4. Systém spustí sledovanie danej TV stanice 5. Používateľ stlačí tlačidlo s názvom Späť <ul style="list-style-type: none"> Alternate: 5a. návrat 3 6. Systém sa vráti na zoznam dostupných TV staníc na sledovanie
<p> Alternate. návrat 1</p> <ol style="list-style-type: none"> 1. Používateľ stlačí tlačidlo s názvom späť 2. Systém sa vráti na zoznam dostupných TV staníc na sledovanie
<p> Alternate. návrat 2</p> <ol style="list-style-type: none"> 1. Používateľ stlačí tlačidlo s názvom Domov 2. Systém sa vráti do hlavného menu
<p> Alternate. návrat 3</p> <ol style="list-style-type: none"> 1. Používateľ stlačí tlačidlo s názvom Domov 2. Systém sa vráti do hlavného menu

UC15 Zobrazenie TV staníc

SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. Používateľ stlačí tlačidlo s názvom TV 2. Systém zobrazí dostupné TV stanice na sledovanie <ul style="list-style-type: none"> Alternate: 2a. Žiadny obsah na zobrazenie 3. Používateľ stlačí tlačidlo s názvom Späť 4. Systém sa vráti do hlavného menu
<p> Alternate. Žiadny obsah na zobrazenie</p>

SCENARIOS

1. Systém vypíše hlášu, že nie sú momentálne dostupné žiadne TV stanice
2. Požívateľ stlačí tlačidlo s názvom Späť
3. Systém sa vráti do hlavného menu

UC16 Zobrazenie kamier

SCENARIOS

 Basic Path. Basic Path


1. Používateľ vyberie jednu z dostupných kamier na sledovanie
2. Systém spustí sledovanie vybranej kamery
3. Požívateľ stlačí tlačidlo s názvom Späť
Alternate: 3a. Návrat 1
4. Systém sa vráti na zoznam dostupných kamier na sledovanie

 Alternate. Návrat 1


1. Používateľ stlačí tlačidlo s názvom Domov
2. Systém sa vráti do hlavného menu

UC17 Zobrazenie dostupných kamier

SCENARIOS

 Basic Path. Basic Path

1. Používateľ stlačí tlačidlo s názvom Kamery
2. Systém zobrazí dostupné kamery na sledovanie
Alternate: 2a. Žiadny obsah na zobrazenie
3. Používateľ stlačí tlačidlo s názvom Späť
4. Systém sa vráti do hlavného menu

 Alternate. Žiadny obsah na zobrazenie


1. Systém vypíše hlášu, že nie sú momentálne dostupné žiadne kamery na sledovanie
2. Používateľ stlačí tlačidlo s názvom Späť
3. Systém sa vráti do hlavného menu

UC18 Zmena jazyka

SCENARIOS

 Basic Path. Basic Path


1. Používateľ stlačí tlačidlo s názvom Výber jazyka
2. Systém zobrazí dostupné jazyky na zmenu
3. Používateľ vyberie jeden z dostupných jazykov
Alternate: 3a. Návrat 1
4. Systém vykoná zmenu jazyka
5. Používateľ stlačí tlačidlo s názvom Späť

SCENARIOS
6. Systém sa vráti do hlavného menu
 Alternate. návrat 1 1. Používateľ nezvolí žiadny z dostupných jazykov na zmenu a stlačí tlačidlo s názvom Späť 2. Systém sa vráti do hlavného menu


UC19 Zobrazenie návodu

SCENARIOS
 Basic Path. Basic Path 1. Používateľ stlačí tlačidlo s názvom Návod 2. Systém zobrazí prehľadanie stránok návodu 3. Používateľ stlačí tlačidlo s názvom Späť 4. Systém sa vráti do hlavného menu



UC2 Zobrazenie zoznamu videí

SCENARIOS
 Basic Path. Basic Path 1. Používateľ prejde do časti Filmy, kde zobrazí zoznam ponúkaných filmov. 2. Systém vylistuje celý zoznam filmov, podľa svojej content databázy.

UC20 Hranie hier

SCENARIOS
 Basic Path. Basic Path 1. Cestujúci vyberie požadovanú hru zo zoznamu hier. 2. Systém spustí zvolenú hru.

UC3 Prehratie hudby


SCENARIOS
 Basic Path. Basic Path 1. Používateľ klikne na tlačidlo prehrať pieseň. 2. Systém spustí prehrávanie piesne. Alternate: 2a. Ukončenie skôr 3. Systém ukončí prehrávanie piesne.
 Alternate. Ukončenie skôr

SCENARIOS

1. Používateľ ukončí prehrávanie piesne počas jej prehrávania.

UC4 Zobrazenie zoznamu hudby


SCENARIOS

 Basic Path. Basic Path

1. Používateľ prejde do časti Hudba, kde zobrazí všetky audio súbory.
2. Systém zobrazí zoznam audio súborov podľa content databázy.

UC5 Zobrazenie obrázkov


SCENARIOS

 Basic Path. Basic Path

1. Používateľ klikne na ikonu obrázku.
2. Systém zobrazí obrázok.

UC6 Zobrazenie zoznamu obrázkov


SCENARIOS

 Basic Path. Basic Path

1. Používateľ prejde do časti Obrázky,
2. Systém zobrazí zoznam obrázkov podľa obsahu content databázy.

UC7 Zobrazenie dokumentov


SCENARIOS

 Basic Path. Basic Path

1. Cestujúci si zo zoznamu dokumentov vyberie dokument na zobrazenie.
2. Systém zobrazí vybraný dokument.
3. Cestujúci stlačí tlačidlo návratu.
4. Systém sa vráti do zoznamu dokumentov.

UC8 Zobrazenie zoznamu dokumentov


SCENARIOS

 Basic Path. Basic Path

1. Cestujúci stlačí tlačidlo dokumentov na domovskej obrazovke.
2. Systém zobrazí zoznam dokumentov.
3. Cestujúci stlačí tlačidlo návratu.
4. Systém zobrazí domovskú obrazovku.

UC9 Zobrazenie obsahu externých médií

SCENARIOS

 Basic Path. Basic Path

1. Cestujúci pripojí externé médium do obrazovky.
2. Systém zobrazí informáciu o pripojení externého média
3. Systém pridá ikonu externého média do menu.
4. Cestujúci kliknutím na ikonu externého média zobrazí obsah tohto média.
5. Systém zobrazí obsah externého média.
6. Cestujúci stlačí tlačidlo návratu.
7. Systém zobrazí domovskú obrazovku.

3.12.2 Diagram prípadov použitia pre stewardku

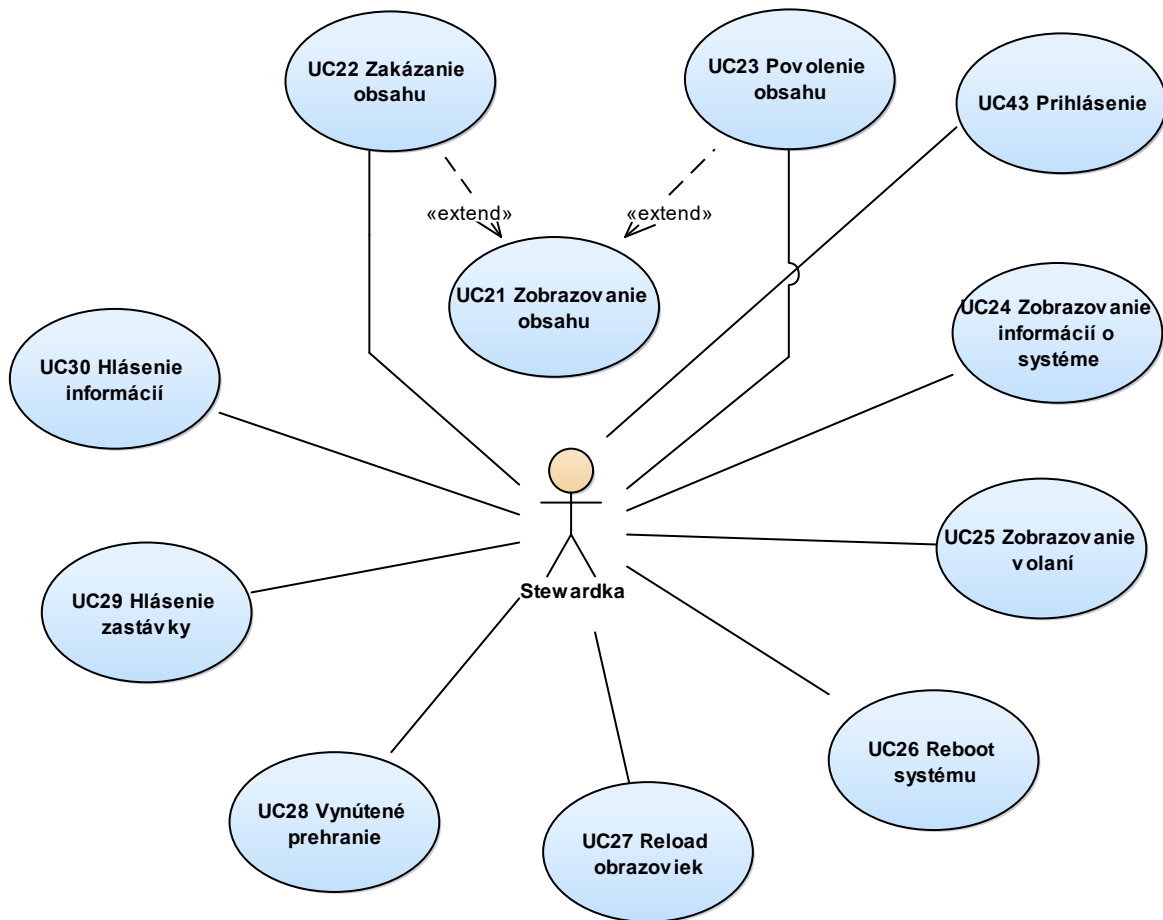





Figure 2: Steward-mode

UC21 Zobrazovanie obsahu


SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. Systém zobrazí zoznam aktuálne povoleného obsahu 2. Systém zobrazí možnosť povolenie obsahu, ktoré je realizované cez Povolenie obsahu Alternate: 2a. Povolenie obsahu 3. Systém zobrazí možnosť zakázanie obsahu, ktoré je realizované cez Zakázanie obsahu Alternate: 3a. Zakázanie obsahu 4. Systém zobrazí aktualizovaný zoznam povoleného obsahu
<p> Alternate. Povolenie obsahu</p> <ol style="list-style-type: none"> 1. Stewardka vyberie možnosť povolenie obsahu. 2. Systém zobrazí zoznam dostupného obsahu. 3. Stewardka zaškrtnie obsah v zozname, ktorý chce povoliť. 4. Systém zaktualizuje zoznam povoleného obsahu.
<p> Alternate. Zakázanie obsahu</p>

SCENARIOS

1. Stewardka vyberie možnosť zakázanie obsahu.
2. Systém zobrazí zoznam dostupného obsahu.
3. Stewardka odškrtnie obsah v zozname, ktorý chce zakázať.
4. Systém zaktualizuje zoznam povoleného obsahu.


UC22 Zakázanie obsahu


CONNECTORS

-  **Extend** «extend» Source -> Destination
 From: UC22 Zakázanie obsahu : UseCase, Public
 To: UC21 Zobrazovanie obsahu : UseCase, Public

UC23 Povolenie obsahu


CONNECTORS

-  **Extend** «extend» Source -> Destination
 From: UC23 Povolenie obsahu : UseCase, Public
 To: UC21 Zobrazovanie obsahu : UseCase, Public

-  **UseCaseLink** Source -> Destination
 From: Stewardka : Actor, Public
 To: UC23 Povolenie obsahu : UseCase, Public

UC24 Zobrazovanie informácií o systéme

SCENARIOS

 Basic Path. Basic Path

1. Stewardka klikne na odkaz zobrazíť informácie o systéme
2. Zobrazí sa list dostupných informácií o systéme: OS obrazoviek/servera, informácie o vozidle (VIN, SPZ, ...), čas posledného reštartu, čas poslednej aktualizácie, počet pripojených obrazoviek, ...

UC25 Zobrazovanie volaní

SCENARIOS

 Basic Path. Basic Path

1. Stewardka klikne na odkaz zobrazíť volania
2. Zobrazí sa list všetkých volaní za posledných X hodín s číslom obrazovky a časom volania

UC26 Reboot systému

SCENARIOS


 Basic Path. Basic Path

SCENARIOS

1. Stewardka klikne na odkaz reboot systému
2. Systém sa rebootuje (so všetkým čo s tým súvisí - aktualizácia, reload obrazoviek, ...)

UC27 Reload obrazoviek


SCENARIOS

 Basic Path. Basic Path

1. Stewardka klikne na odkaz reload obrazoviek
2. Systém vymaže všetky používateľské nastavenia na obrazovkách a reloadne home page

UC28 Vynútené prehranie


SCENARIOS

 Basic Path. Basic Path


1. Stewardka klikne na odkaz vynútené prehratie
2. Zobrazí sa list všetkých možných videí s kategóriou vynútené prehratie
3. Stewardka klikne na požadované video
4. Zobrazia sa detaily daného videa
5. Stewardka klikne na tlačidlo vynútené prehratie
6. Systém pozastaví činnosť a prehrá vybrané video na všetkých zariadeniach, ktoré nie sú v admin konzole

UC29 Hlásenie zastávky


SCENARIOS

 Basic Path. Basic Path

1. Systém zobrazí zoznam trás.
2. Stewardka vyberie trasu.
 - Alternate: 2a. Upraviť trasu
 - Alternate: 2b. Pridať trasu
3. Systém zaktualizuje plánovanú trasu.
4. Systém bude oznamovať jednotlivé zastávky na vybranej trase.

 Alternate. Upraviť trasu

1. Stewardka zvolí možnosť upraviť trasu.
2. Systém zobrazí vybranú trasu.
3. Stewardka vyberie zastávku.
4. Stewardka odstráni alebo nahradí zastávku novou zastávkou.
5. Stewardka uloží vykonané zmeny.
6. Systém zaznamená zmeny v trase.

 Alternate. Pridať trasu

1. Stewardka zvolí možnosť pridať trasu.
2. Systém zobrazí formulár pre novú trasu.

SCENARIOS

3. Stewardka pridá nové zastávky.
4. Stewardka uloží novú trasu.
5. Systém zaznamená novú trasu.

UC30 Hlásenie informácií

SCENARIOS

Basic Path. Basic Path

1. Systém zobrazí možnosť hlasová a písomná správa.
2. Stewardka vyberie možnosť hlasová správa.
Alternate: 2a. Písomná správa
3. Systém pozastaví všetok prehrávaný obsah.
4. Systém zobrazí informáciu o aktivácii mikrofónu.
5. Systém zobrazí možnosť ukončiť hlasovú správu.
6. Stewardka zvolí možnosť ukončiť hlasovú správu.
7. Systém spustí pokračovanie prehrávania pozastaveného obsahu.

Alternate. Písomná správa

1. Stewardka zvolí možnosť písomná správa.
2. Systém zobrazí formulár pre správu.
3. Stewardka napíše správu do formulára.
4. Stewardka potvrdí odoslanie správy.
5. Systém pozastaví všetok prehrávaný obsah.
6. Systém zobrazí správu na všetkých monitoroch na 30 sekúnd.

UC43 Prihlásenie

Prihlásenie stewardky do oddelenej a uzamknutej časti webového portálu.

SCENARIOS

Basic Path. Basic Path

1. Stlačenie vybranej kombinácie tlačidiel na základnej obrazovke web portálu.
2. Systém zobrazí rozhranie pre steward-ku.

Alternate. Nesprávne prihlasovacie údaje

1. Systém zobrazí inštalatérovi chybu s informáciou o nesprávne zadanom prístupovom hesle.

Alternate. Nesprávne prihlasovacie údaje Inštalatér

3.12.3 Diagram prípadov použitia pre inštalatéra(Molpir)

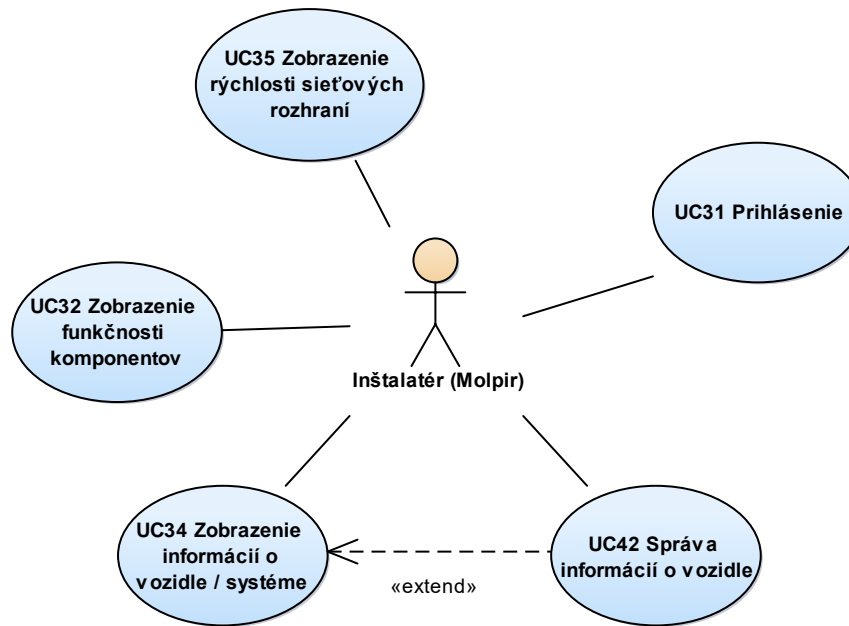


Figure 3: Inštalatér(Molpir) - model

Inštalatér (Molpir)

Inštalatér z Molpir-u vykonávajúci úvodnú inštaláciu celého systému a prepojení medzi obrazovkami, serverom a Internetom.

UC31 Prihlásenie

Prihlásenie správcu flotily do oddelenej a uzamknutej časti webového portálu.

SCENARIOS

Basic Path. Basic Path

1. Stlačenie vybranej kombinácie tlačidiel na základnej obrazovke web portálu.
2. Systém zobrzí rozhranie pre stevard-ku.
3. V prihlasovacom menu pre stevardku stlačenie kombinácie tajných tlačidiel.
4. Systém zobrazí prihlasovacie rozhranie pre správcu flotily.
5. V prihlasovacom rozhraní pre správcu flotily stlačenie kombinácie tajných tlačidiel.
6. Systém zobrazí prihlasovacie rozhranie pre prihlásenie do inštalátorskej konzoly.
7. Inštalatér zadá heslo.
8. Systém zobrazí ovládaci panel inštalatéra.

Alternate: 8a. Nesprávne prihlasovacie údaje Inštalatér


Alternate. Nesprávne prihlasovacie údaje

1. Systém zobrazí inštalatérovi chybu s informáciou o nesprávne zadanom prístupovom hesle.

Alternate. Nesprávne prihlasovacie údaje Inštalatér

CONSTRAINTS

CONSTRAINTS

 Pre-condition. Prihlásený inštalatér.

[Approved, Weight is 0.]

UC32 Zobrazenie funkčnosti komponentov


Otestovanie / zobrazenie funkčnosti HW / SW / NET modulov / komponentov.

SCENARIOS

 Basic Path. Basic Path

1. Inštalatér otvorí okno na testovanie alebo zobrazovanie funkčnosti pripojených komponentov.
2. Zobrazenie okna na zobrazovanie a testovanie funkčnosti komponentov.
3. Kliknutie na tlačidlo obnovy informácií o komponentoch.
4. Načítanie HW, SW a sieťových informácií, interakcia so sieťovými komponentami pomocou sieťových protokolov.
5. Kliknutie na podpriechinok hub-y.
6. Načítanie informácií o pripojených hub-och.
7. Kliknutie na podpriechinok obrazovky.
8. Načítanie informácií o pripojených obrazovkách.
9. Kliknutie na podpriechinok stream-zariadenia.
10. Zobrazenie informácií o stream-ing zariadeniach.
11. Kliknutie na podpriechinok súborový systém.
12. Zobrazenie informácií o pripojených úložiskách a ich dostupnej kapacite.
13. Kliknutie na podpriechinok sieť.
14. Zobrazenie informácií o rozhraniach a počte packet-ov in / out a počte zahodených packet-ov.
15. Kliknutie na podpriechinok systémové služby.
16. Zobrazenie informácií o spustených službách v systéme.
17. Kliknutie na podpriechinok HW komponenty.
18. Zobrazenie informácií o pripojených HW moduloch / ich funkčnosti (aj RAM test a GPS modul).
19. Kliknutie na podpriechinok NTP (časové nastavenia).
20. Zobrazenie informácií o dostupnosti nastavených NTP serverov.
21. Kliknutie na tlačidlo "Touch-screen test".
22. Vyvolanie testovacích okien vo webových portáloch na obrazovkách.
23. Vytvorenie spätnej väzby používateľom na obrazovkách.
24. Zobrazenie súhrnných informácií o úspešnosti touch-screen testov.
25. Návrat do hlavného menu pomocou návratového tlačidla.
26. Zobrazenie hlavného ovládacieho panelu.
27. Kliknutie na odhlásenie inštalatéra.
28. Zobrazenie okna pre stevardku.

CONSTRAINTS

 Pre-condition. Prihlásený inštalatér.

[Approved, Weight is 0.]

UC34 Zobrazenie informácií o vozidle / systéme

Možnosť zobrazenia informácií o:

- Plate Number – nastavenie evidenčného čísla vozidla
- Vehicle Identification Number – nastavenie VIN vozidla
- Seats Count Setting – nastavenie celkového počtu (sedadiel) monitorov vo vozidle
- Seats Setting – priradenie čísel sedačiek k jednotlivým monitorom

- Car Type Selection – nastavenie typu vozidla
- Hubs Count Setting – nastavenie počtu HUB-ov a priradenie jednotlivých monitorov k HUBom
- TV Channels Setting – nastavenie počtu TV kanálov
- Android systému
- Číslovanie monitorov

SCENARIOS


Basic Path. Basic Path

1. Kliknutie na tlačidlo zobrazenie / úprava informácií o vozidle.
2. Otvorenie okna pre zobrazovanie alebo modifikáciu informácií o vozidle.
3. Kliknutie na tlačidlo zobrazenia súhrnných informácií o systéme.
4. Zobrazenie informácií vo formáte adresárov, ktoré je možné otvárať a podrobnejšie analyzovať.
5. Kliknutie na odhlásenie inštalatéra.
Alternate: 5a. Modifikácia položiek
6. Zobrazenie okna pre stevardku.

Alternate. Modifikácia položiek

1. Používateľ klikne na tlačidlo úprava (pri konkrétnej úlohe).
2. Vyvolanie use case UC42 Správa informácií o vozidle.

CONSTRAINTS

 Pre-condition. Prihlásený inštalatér.

[Approved, Weight is 0.]

UC35 Zobrazenie rýchlosti sieťových rozhraní

Zobrazenie informácií o priepustnosti sieťových rozhraní. Sú užitočné pre správne nastavenie rozloženia prepojení.

SCENARIOS

Basic Path. Basic Path

1. Kliknutie na tlačidlo zobrazenia sieťovej rýchlosti.
2. Zobrazenie okna s informáciami o maximálnej priepustnosti sieťových rozhraní spolu s označeniami rozhraní a ich L2 / L3 adresami a prepočet priepustnosti na jeden monitor / externé mobilné zariadenie.
3. Kliknutie na tlačidlo s návratom do hlavného menu.
4. Zobrazenie hlavného menu.
5. Kliknutie na odhlásenie inštalatéra.
6. Zobrazenie okna pre stevardku.

CONSTRAINTS

 Invariant. Prihlásený inštalatér.

[Approved, Weight is 0.]

UC42 Správa informácií o vozidle

Modifikácia informácií o vozidle v prípade potreby.

SCENARIOS

Basic Path. Basic Path

1. Otvorenie okna pre modifikáciu Android systému.
2. Zobrazenie okna pre modifikáciu Android systému.
3. Nahranie obrazu pre Android systém.
4. Aplikovanie zmien v systéme.
5. Zatvorenie predchádzajúceho okna a otvorenie okna pre modifikáciu "Hubs Count Setting".
6. Zobrazenie okna pre modifikáciu "Hubs Count Setting".
7. Nastavenie počtu HUB-ov a priradenie jednotlivých monitorov k HUBom.
8. Uloženie nastavení v systéme.
9. Zatvorenie predchádzajúceho okna a otvorenie okna pre modifikáciu "Plate number".
10. Zobrazenie okna pre modifikáciu "Plate number".
11. Modifikácia evidenčného čísla v textovom poli a uloženie nastavení.
12. Uloženie nastavení v systéme.
13. Zatvorenie predchádzajúceho okna a otvorenie okna pre modifikáciu "Vehicle Identification Number".
14. Zobrazenie okna pre modifikáciu "Vehicle Identification Number".
15. Modifikácia VIN vozidla v textovom poli a uloženie zmien.
16. Uloženie nastavení v systéme.
17. Zatvorenie predchádzajúceho okna a otvorenie okna pre modifikáciu "Seats Count Setting".
18. Zobrazenie okna pre modifikáciu "Seats Count Setting".
19. Modifikácia celkového počtu sedadiel vo vozidle.
20. Uloženie nastavení v systéme.
21. Zatvorenie predchádzajúceho okna a otvorenie okna pre modifikáciu "Seats Setting".
22. Zobrazenie okna pre modifikáciu "Seats Setting".
23. Spustenie operácie vyvolania číselných okien na monitoroch.
24. Systém aktivuje priradovacie okná na všetkých monitoroch.
25. Nastavenie číslovania vozidiel priamo na monitoroch a uloženie nastavenia potvrdením.
26. Spracovanie nastavení a uloženie priradených čísel sedačiek k jednotlivým monitorom.
27. Zatvorenie predchádzajúceho okna a otvorenie okna pre modifikáciu "Car Type Selection".
28. Zobrazenie okna pre modifikáciu "Car Type Selection".
29. Výber typu vozidla pomocou combo-box-u a potvrdenie výberu pomocou tlačidla.
30. Uloženie nastavení v systéme.
31. Zatvorenie predchádzajúceho okna a otvorenie okna pre modifikáciu multicast kanálov.
32. Zobrazenie okna pre modifikáciu multicast zdrojov.
33. Zadeinovanie multicast zdrojov pomocou IP adries, protokolu a portov (interktívne nastevnie pomocou textového poľa a combo-boxu). Potvrdenie nastavení.
34. Uloženie nastavení v systéme.
35. Návrat do hlavného menu pre správu informácií o vozidle.
36. Zobrazené hlavného menu pre správu informácií o vozidle.

3.12.4 Diagram prípadov použitia pre správa flotily

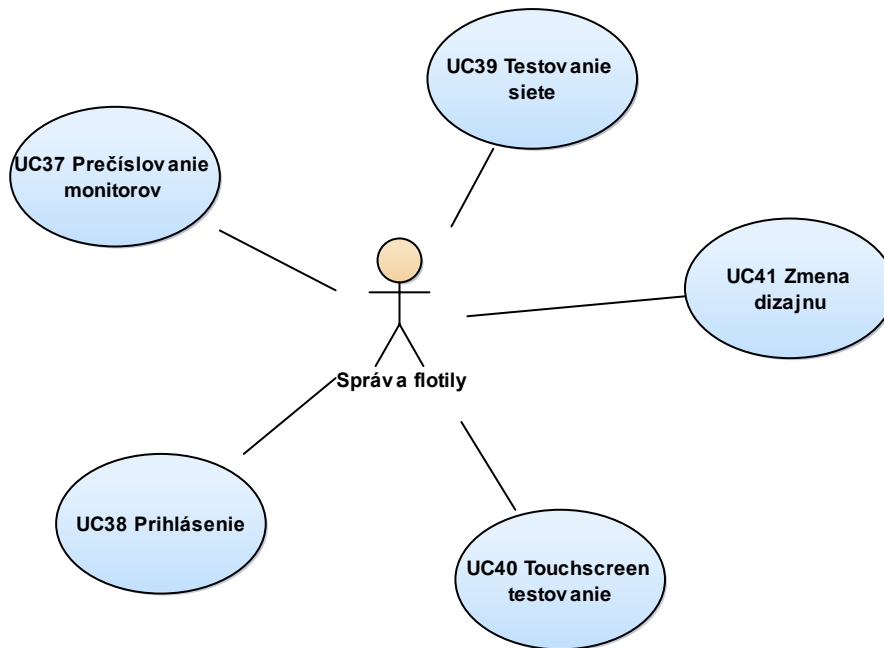


Figure 4: Správca flotily - model

UC37 Prečíslovanie monitorov

Zmena číslovania dispeljov.

SCENARIOS

Basic Path. Basic Path

1. Prihlásenie používateľa v role správca flotily(<include> Prihlásenie).
2. Správca flotily vyberie položku test siete.
3. Systém zobrazí na každom jednom monitore jeho aktuálne číslo.
4. Správca flotily vyberie možnosť zmeny čísla monitoru na monitore ktorému chce zmeniť číslo.
5. Systém zobrazí okno v ktorom je možnosť meniť číslo monitora.
6. Správca flotily zadá číslo do príslušného okna a potvrdí svoju voľbu.
7. Systém zmení číslo monitora na správcom zvolené číslo.

Alternate: 7a. Číslo je obsadené

Alternate. Číslo je obsadené

1. Systém zobrazí chybovú hlášku o obsadenosti tohto čísla a požiada o znovazadanie čísla.



UC38 Prihlásenie

Prihlásenie správca flotily do oddelenej a uzamknutej časti webového portálu.

UC38 Prihlásenie


SCENARIOS

Basic Path. Basic Path

SCENARIOS
<ol style="list-style-type: none"> 1. Stlačenie vybranej kombinácie tlačidiel na základnej obrazovke web portálu. 2. Systém zobrzí rozhranie pre stevard-ku. 3. V prihlasovacom menu pre stevardku stlačenie kombinácie tajných tlačidiel. 4. Systém zobrazí prihlasovacie rozhranie pre správcu flotily. 5. V prihlasovacom rozhraní pre správcu flotily stlačenie kombinácie tajných tlačidiel. 6. Systém zobrazí prihlasovacie rozhranie pre prihlásenie do inštalátorskej konzoly. 7. Inštalatér zadá heslo. 8. Systém zobrazí ovládaci panel inštalatéra. <p>Alternate: 8a. Nesprávne prihlasovacie údaje Inštalatér</p>
<p> Alternate. Nesprávne prihlasovacie údaje</p> <ol style="list-style-type: none"> 1. Systém zobrazí inštalatérovi chybu s informáciou o nesprávne zadanom prístupovom hesle.
<p> Alternate. Nesprávne prihlasovacie údaje Inštalatér</p>


UC39 Testovanie siete

Test vytvárenosti systému.

SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. Prihlásenie používateľa v role správcu flotily (<include> Prihlásenie). 2. Správca flotily vyberie položku test siete. 3. Systém zobrazí pripojené prípadne odpojené komponenty systému (huby, monitory) a zaťaženosť siete.


UC40 Touchscreen testovanie

Test dotykovej obrazovky displeja.

SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. Prihlásenie používateľa v role Správca flotily(<include> Prihlásenie). 2. Správca flotily vyberie položku test touchscreen. 3. Systém zobrazí obrazovku na testovanie obrazovky. 4. Správca flotily kliká na zobrazené miesto na displeji. 5. Systém skontrolu správnosť kliknutia.

UC41 Zmena dizajnu

Zmena dizajnu systému.

SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. Prihlásenie používateľa v role správcu flotily (<include> Prihlásenie). 2. Výber možnosti "Zmena dizajnu". 3. Zobrazenie možností na správu dizajnu.

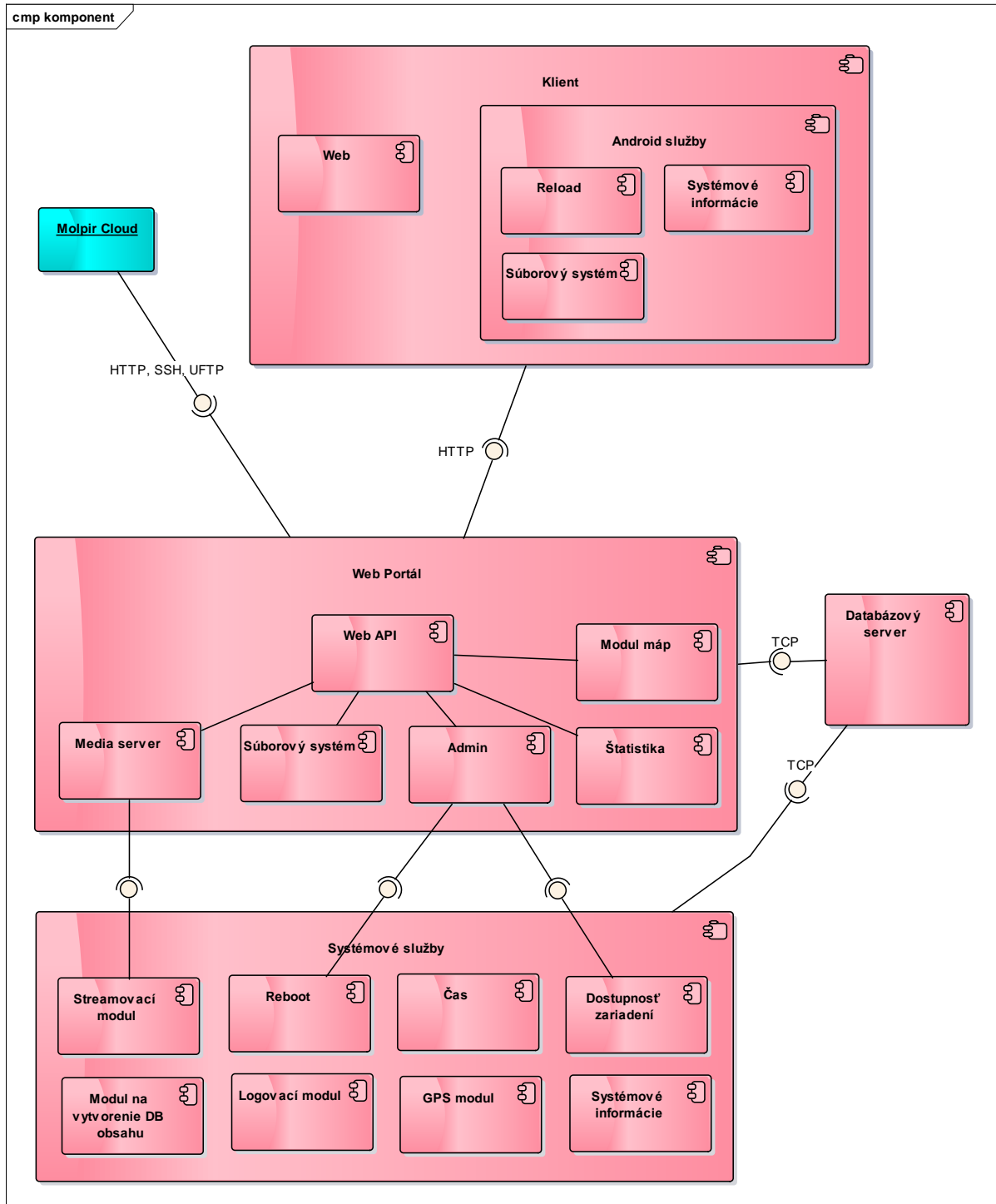
SCENARIOS

4. Nastavenie parametrov pre zmenu dizajnu.
5. Zmena dizajnu systému.

4 Architektúra a návrh systému

Táto kapitola obsahuje architektonický návrh systému v ktorom je zobrazenie navrhnutých modulov.

4.1.1 Model komponentov



Jednotlivé komponenty, medzi ktorými nie je znázornené rozhranie komunikujú prostredníctvom databázového servera.

Klient:

- Reload – resetovanie používateľských nastavení na tabletoch
- Systémové informácie – verzia androidu, vytáženie CPU, Ram, a iné
- Súborový systém – sprístupnenie súborového systému zariadenia – vlastné USB

Web portál

- Modul máp – zobrazovanie polohy autobusu na mape. Aktuálna poloha z databázy, kde to naplní systémový GPS modul.
- Media server – riadenie prehrávania filmov a streamov a ich transcodingu a voľby audio stop pomocou systémového streamovacieho modulu
- Súborový systém – sprístupňovanie požadovaných súborov (audio, dokumenty, obrázky, ...) z kontent databázy
- Admin – rôzne požadované funkcionality pre stewardku, inštalátora a správcu flotily. Ako napríklad zistenie dostupnosti nastavení, rebootu systému (servera), konfigurácia nastavení (databáza), hlásenie informácií a iné.
- Štatistika – štatistika prehrávania videí

Databázové služby

- Streamovací modul – streamovanie a transcoding videa z kamier, satelitov a iných zdrojov
- Reboot – reštart a aktualizácia systému
- Čas – ntp server, poskytuje aktuálny čas pre klientov
- Dostupnosť zariadení – zistenie dostupností obrazoviek a hubov v sieti – ping
- Modul na vytvorenie DB obsahu – spustí sa po pripojení dátového disku do systému. Vygeneruje kontent databázu, ktorej obsah sa bude zobrazovať klientom na webe
- Logovací modul – zaznamenávanie, formátovanie a ukladanie logov do databázy
- GPS modul – zaznamenávanie aktuálnej polohy a jej uloženie do databázy
- Systémové informácie – informácie o operačnom systéme, využití CPU, Ram, diskového priestoru, a iné

5 Implementácia

5.1 Backendové služby

5.1.1 Webový portál

Webový portál pozostáva z dvoch hlavných častí, a to front-end aplikácie využívajúcej webový framework Angular 2 a back-end časti vyvinutej v programovacom jazyku java, využívajúcej Spring framework. Hlavná časť back-end časti webového portálu je preto poskytovanie REST rozhraní pre získavanie dynamických dát z databázy vo formáte JSON.

Vytvorenie takýchto rozhraní je vďaka použitiu už spomenutého Spring frameworku pomerne jednoduché a založené primárne na anotáciách tried, metód a premenných v java kóde.

Základná konfigurácia Spring frameworku pozostáva z nasledujúcich tried:

- `sk.fiit.team17.config.AppConfig` – obsah tejto triedy je prázdny, dôležité su jej anotácie, ktoré určujú nastavenia aplikácie
 - `@Configuration` – určuje, že sa jedná o konfiguračnú triedu aplikácie
 - `@ComponentScan("sk.fiit.team17")` – určuje, ktoré triedy sa budú prehľadávať pre analýzu spring anotácií
 - `@EnableWebMvc` – pridanie WebMvc modulu do aplikácie. Toto je hlavný modul, vďaka ktorému dokáže aplikácia vytvárať REST rozhrania
 - `@EnableScheduling` – pridanie plánovača do aplikácie. Plánovač sa používa na vykonanie periodických, alebo presne časovo určených úloh
- `sk.fiit.team17.config.AppInitializer` – v tejto triede sa inicializuje aplikácia a načíta sa konfigurácia z triedy `AppConfig`. Tiež sa tu nastavuje mapovanie aplikácie – v našom prípade `"/rest/*"` – takže všetky rozhrania budú začínať na takejto adrese.
- `sk.fiit.team17.config.DatabaseConfig` – konfiguračná abstraktná trieda pre konfiguráciu nového databázového spojenia. Konkretná konfigurácia – umiestnenie databázy, meno, heslo a podobne - sa implementuje až v triede, ktorá dedí od tejto triedy, napríklad v `sk.fiit.team17.config.MediaDBConfig`.

Controller pre poskytovanie REST rozhrania je potom trieda, ktorá obsahuje metódy, ktoré vracajú určitý objekt a očakávajú určité vstupné parametre. Každá takáto metóda predstavuje jedno REST rozhranie. Táto trieda a jej metódy sú doplnené o spring anotácie:

- `@RestController` – anotácia celej triedy. Hovorí, že trieda bude obsahovať REST rozhrania.
- `@RequestMapping` – anotácia metódy. Definuje REST rozhranie a obsahuje hodnoty:
 - `Value` – URL REST rozhrania, napríklad `"/movie/movie_path/{movieId}"`

- Produces – format vyprodukovaného výstupu, napríklad json, xml, alebo text. V našom prípade je stále nastavený ako "application/json"
- Method – použitá HTTP metóda, napríklad GET, POST, PUT, HEAD a podobne.
- @PathVariable – anotácia vstupnej premennej, ktorá sa načíta priamo z URL rozhrania. Musí byť definovaná pomocou zátvoriek { a } v hodnote value v anotácii @RequestMapping
- @RequestParam – anotácia vstupnej premennej, ktorá sa načíta z parametra zadanej URL

5.1.2 Spúšťanie skriptov pri štarte systému

Všetky programy, ktoré sa majú spúšťať pri štarte operačného systému sú umiestnené v adresári /opt. Aktuálny zoznam aplikácií:

```
-rwxr-xr-x 1 fiit fiit 45344 Apr 24 10:37 AdvertDBFiller
-rwxr-xr-x 1 root root 26712 Apr 6 21:31 ControlLoggingDb
-rwxr-xr-x 1 fiit fiit 37544 Apr 30 00:42 InitDb
-rwxr-xr-x 1 fiit fiit 74876 Apr 30 00:21 MediaDbFiller
-rwxr-xr-x 1 fiit fiit 67196 Apr 30 00:38 MusicDbFiller
-rwxr-xr-x 1 root root 54 Apr 20 23:17 permissions.sh
-rwxr-xr-x 1 fiit fiit 65596 Apr 30 00:57 PictureDbFiller
-rwxr-xr-x 1 root root 44 Apr 13 10:13 proxy.sh
-rwxr-xr-x 1 fiit fiit 23938802 Apr 27 08:56 RemoteCloud.jar
-rwxr-xr-x 1 root root 1911 May 9 10:26 startup.sh
-rwxr-xr-x 1 root root 109 Apr 12 18:13 timeset.sh
```

Stručný opis skriptov sa nachádza v Tabuľka 35.

Tabuľka 35. Opis skriptov v adresári /opt

Skript	Význam
AdvertDBFiller	Naplnenie media databázy s reklamami uloženými vo vyhradenom adresári na SSD.
ControlLoggingDb	Rotácia logov (aplikačné a systémové logy) a štatistík.
InitDb	Inicializácia / vytvorenie štruktúry media databázy.
MediaDbFiller	Naplnenie media databázy s filmami.
MusicDbFiller	Naplnenie media databázy s hudobnými albumami.
PictureDbFiller	Naplnenie media databázy s albumami obrázkov.
RemoteCloud.jar	Vlákna vzdialenej správy (komunikácia s cloudom).
timeset.sh	Automatické nastavenie času pri štarte OS z NTP serverov.
permissions.sh	Nastavenie prístupových práv pre namountovanú data partíciu.
startup.sh	Spustenie všetkých ostatných už opísaných skriptov s príslušnými vstupnými argumentami a v správnom poradí. Tento skript obsahuje umiestnenia konfiguračných súborov, certifikačných súborov,

Skript	Význam
	datábazových súborov a prevolených adresárov, z ktorých sú čítané multimedialne dáta.
proxy.sh	Odkaz na startup.sh s možnosťou spustenia v ľubovľnom adresári (tento skript sa volá v systemd).

Opísané skripty sa volajú vždy iba pri štarte operačného systému pred spustením Tomcat služby (webový portál pracuje už s naplnenými databázami). Ak sa nepodarí spustiť proxy.sh, potom sa nespustí ani Tomcat služba. Na druhej strane, skripty sa volajú až po inicializácii sieťového spojenia (aby sa mohol aktualizovať čas a vzdialená správa).

Skript zodpovedný za spustenie Tomcat služby pri štarte OS:

/etc/systemd/system/tomcat.service

Skript zodpovedný za spustenie podporných systémových služieb webového portálu:

/etc/systemd/system/modsystem.service

Bližší opis systémových skriptov / programov sa nachádza v nasledujúcich kapitolách 5.1.3 až 0.

5.1.3 Modul na vzdialenú správu

Implementované časti vzdialenej správy:

- Synchronizácia štatistík s cloudom – Pravidelné odosielanie agregovaných štatistík za zvolenú dobu na cloud (štatistiky z prehrávania filmov, hudby, reklám a zobrazovania obrázkov).
- Synchronizácia systémových logov s cloudom – Pravidelné odosielanie systémových logov na cloud (logy generované a následne uložené službou Rsyslog do databázy).
- Synchronizácia aplikačných logov s cloudom – Pravidelné odosielanie aplikačných logov na cloud (logy generované webovým portálom a ostatnými aplikáciami – výnimky alebo rôzne aplikačné informácie).
- Sťahovanie príkazov z cloud – Príkazy reprezentujú akciu, ktorá sa má vykonať na cloude (napr. spustenie určitého príkazu). Zatiaľ je implementovaný iba príkaz na vytvorenie SSH tunela medzi MOD serverom a vybraným vzdialeným zariadením na vzdialenej strane. Je potrebné vytvárať forwardovací reverzný tunel z dôvodu priamej nedostupnosti MOD servera z Internetu (komunikáciu vždy inicializuje MOD server, nie cloud).

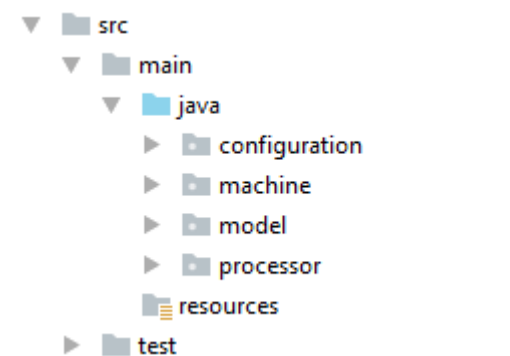
Jednotlivé kľúčové časovače, konfigurácie, nastavenia a formáty správ sú opísané v nasledujúcich kapitolách. Všetka sieťová komunikácia je zabezpečená s nasledujúcimi technikami:

- TLS remote-access tunel: Medzi MOD serverom a klientom – MOD má k dispozícii zoznam dôveryhodných certifikátov, ktoré patria cloud serveru.

- Autentifikačná HTTP hlavička – Každý MOD server sa musí autentifikovať voči cloudu pomocou identifikátora MOD servera a prístupového hesla (tieto informácie sú uložené v globálnom konfiguračnom súbore na serveri).

Štruktúra Java projektu (stručný opis)

Štruktúra projektu je znázorňuje Obrázok 15. Prvý základným adresárom je main – obsahuje všetky zdrojové kódy funkčných softvérových komponentov usporiadané v balíkoch. Adresár test obsahuje jednotkové testy – označenia tried jednotkových testov sú rovnaké ako označenia tried, ktorých metóda sa v danom teste testuje + prefix / postfix „Test“. Nasledujúce odseky sa bližšie venujú významu jednotlivých balíkov ako aj ich obsahu.



Obrázok 15. Štruktúra projektu

Balík „configuration“ – Nastavenia databázových spojení – vytváranie inštancií triedy EntityManagerFactory, cez ktorú je možné spravovať (vytvárať, ukončovať) databázové spojenia. Všetky triedy slúžiace na vytvorenie konfiguračnej triedy musia byť odvodené od abstraktnej triedy DbConfig, ktorá obsahuje JPA nastavenia. Ako objektovo-relačný mapovač sú použité knižnice EclipseLink. Vytvárajú sa celkovo 3 skupiny mapovaní (opis spojení s databázou):

1. Media databáza (DbConfigAdvert) – Práva s reklamami.
2. Logovacia databáza (DbConfigiLogging) – Logovanie udalostí a posielanie logov na cloud.
3. Databáza so štatistikami – Posielanie agregovaných štatistík na cloud.

Balík „machine“ – Tento balík obsahuje triedy, ktoré sú zodpovedné za tvorbu hlavných vlákien aplikácie. Medzi triedy, ktoré tvoria a spravujú vlákna aplikácie patria triedy odvodené od rozhraní ThreadMachine (logy) a ThreadMachine2 (štatistiky). Tieto rozhrania definujú správanie sa vlákien – spustenie vlákna, načítanie dát s databázy alebo označenie poslaných dát v príslušnej databáze. Výnimkou sú triedy pre spravovanie vlákien na načítanie konfigurácie reklám (AdvertThread) a spracovanie prijatých príkazov (CommandQueuriesThread) – je potrebný iný životný cyklus, ktorý je opísaný v poslednej kapitole. Ďalej, nachádzajú sa tu aj abstraktné triedy implementujúce ThreadMachine resp. ThreadMachine2 rozhrania – implementácia niektorých spoločných metód pre všetky

prípady synchronizácie logov a štatistík (napr. aplikačné logy a systémové logy). Konečný zoznam tried na správu vlákien:

- AdvertThread – Vlákno, ktoré má za úlohu periodicky dopytovať cloud o nový konfiguračný súbor pre reklamy.
- ApplicationLogsThread – Periodická synchronizácia aplikačných logov aj vo viacerých balíkoch s viacerými pokusmi.
- SystemLogsThread - Periodická synchronizácia systémových logov aj vo viacerých balíkoch s viacerými pokusmi.
- StatisticThread – Periodická agregácia a synchronizácia štatistík za posledné (konfigurovateľné) obdobie s cloudom.
- CommandQueriesThried – Periodické dopytovanie cloudu po nové príkazy, ktoré má vykonať MOD server (napr. vytvorenie nového SSH reverzného tunela).

Ďalej, v tomto balíku sa nachádza aj trieda zodpovedná za posielanie HTTP správ na cloud – obsahuje metódu na serializáciu a posielanie JSON správ v HTTP POST a metódy na posielanie GET správ za účelom načítania nového príkazu z cloudu alebo nového konfiguračného súboru pre reklamy (trieda HttpsSenderReceiver). Táto trieda je používaná všetkými vláknami. Všetky HTTP spojenia sú šifrované (TLS) a autentifikované obojstranne (cloud certifikátom a klient identifikátorom + heslom).

Poslednou triedou je StartApp – obsahuje hlavnú spúšťačiu metódu main, z ktorej sa spúšťajú všetky potrebné vlákna, ktoré sú zapnuté. Nastavenia vlákien, časovačov, stránok, IP adries alebo certifikačných ciest sú načítavané z XML súborov.

Balík “model“ – Obsahuje ďalšie balíky, ktoré slúžia na mapovanie XML konfiguračných súborov a databázových modelov do Java tried. Každý z podbalíkov obsahuje jednu triedu s označením Manager, ktorá slúži na interakciu s konfiguračným súborom alebo databázovou entitou (čítanie alebo zapisovanie objektov do relácií / konfiguračného súboru -). Opis podbalíkov:

- **adverts** – Balík obsahujúce mapovanie na reláciu s reklamami v media databáze trieda Advert. Manažérom vykonávajúcim databázové transakcie je trieda AdvertDB a jej metódy.
- **commands** – Balík obsahuje triedy reprezentujúce očakávanú štruktúru prijímaných JSON správ obsahujúcich príkazy, ktoré sa majú na MOD serveri vykonať. Každý príkaz je reprezentovaný unikátnym ID (opis ďalej v kapitole) – ID sa nachádza v triede Command. Potom, na základe ID sa podmiennečne mapuje payload na ďalšiu triedu – v tomto prípade je implementovaný iba príkaz na vytvorenie reverzného SSH tunela (trieda SshCommand).
- **logging** – Obsahuje triedy mapujúce systémové a aplikačné logy v logovacej databáze – PortalLog, SystemEvents a podporná agregujúca trieda SeverityLevel. Manažérom je trieda LoggingDb – vyberanie logov z databázy a označovanie poslaných logov značkou.
- **statistic** – Obsahuje triedu mapujúcu reláciu na ukladanie štatistických záznamov (Statistic), triedu reprezentujúcu agregované štatistiky (AggregatedStatistic) a triedu reprezentujúcu formát štatistických údajov odosielaných na cloud

(StatisticsForCloud) (AggregatedStatistic agreguje StatisticsForCloud). Manažérom databázových dopytov je trieda StatisticDB.

- **xmlconfig** – Načítavanie / zápis XML konfiguračných súborov – reklamy a globálny konfiguračný súbor obsahujúci všeobecné nastavenia prahových hodnôt, časovačov, adresy stránok na cloude, IP adresy alebo umiestnenia ďalších súborov. Manager pre globálny konfiguračný súbor je trieda (a jej metódy) XmlManager, manager pre reklamy (oddelený konfiguračný súbor) je trieda XmlManagerAdvert. Ak XML súbor má chybnú štruktúru, potom sa celý XML súbor nenačíta a vytvorí sa príslušná výnimka. Ak XML má dobrú štruktúru, ale neznáme polia alebo polia s nesprávnymi hodnotami, hodnoty v chýbajúcich / chybných poliach sa nahradia predvolenými hodnotami. Takisto, XML súbor sa vytvorí s predvolenými hodnotami, ak neexistuje zatiaľ žiaden súbor na predvolenom umiestnení.

Balík „processor“ – Balík obsahuje všeobecný procesor na spracovanie hlavného tela príkazu – mapovanie payloadu do konkrétnej inštancie payloadu a vykonanie kódu v payloade. Jednotlivé procesory, ktoré slúžia na vykonanie payloadu príkazu podľa konkrétneho ID, sú triedy implementujúce rozhranie Processor. Momentálne, je implementovaný iba jeden špecifický procesor – SshCommandImplementation.

Nastavenia pred kompiláciou

Certifikáty pre webovú komunikáciu

- Všetky certifikačné súbory sa musia nachádzať v adresári (v OS Windows je to D:\var\certs alebo C:\var\certs podľa toho, na ktorej partícii je uložený kompilovaný projekt):

/var/certs

- Aby unit testy prebehli úspešne, je potrebné vygenerovať cloud certifikát lokálne na MOD servery (self-signed), alebo na stroji, na ktorom sa budú spúšťať unit testy – privátny kľúč musí byť známy, kvôli WireMock serveru.
- Pri testovaní, heslo musí byť všade nastavené na reťazec **password** (dôležité!).
- Vygenerovanie certifikátu do **keystore.jks** repozitára (pri reálnom nasadení nasledovný krok nie je potrebný, pretože cloud dodá serveru verejný certifikát):

keytool -genkey -keyalg RSA -alias cloud -keystore keystore.jks -storepass password -validity 360 -keysize 2048

What is your first and last name?

[Unknown]: Jaroslav Toth

What is the name of your organizational unit?

[Unknown]: stu

What is the name of your organization?

[Unknown]: fiit

What is the name of your City or Locality?

[Unknown]: Bratislava

What is the name of your State or Province?

[Unknown]: Slovakia

What is the two-letter country code for this unit?

[Unknown]: SR

Is CN=Jaroslav Toth, OU=stu, O=fiit, L=Bratislava, ST=Slovakia, C=SR correct?

[no]: yes

Enter key password for <cloud>

(RETURN if same as keystore password):

- Nasledujúci krok je opäť potrebný iba pri lokálnom testovaní aplikácie – exportovanie verejného certifikátu z repozitára do súboru **cloud.cer**.
- V reálnom nasadení cloud dodá tento certifikačný súbor na MOD server (napr. pri inštalácií operačného systému).

keytool -export -keystore keystore.jks -alias cloud -file cloud.cer

Enter keystore password:

Certificate stored in file <cloud.cer>

- Importovanie certifikačného súboru **cloud.cer** do lokálneho úložiska dôveryhodných certifikátov - **truststore.jks**.
- Tento krok sa musí vykonať vždy.

keytool -import -file cloud.cer -alias cloudCA -keystore truststore.jks

Enter keystore password: password

Re-enter new password: password

Owner: CN=Jaroslav Toth, OU=stu, O=fiit, L=Bratislava, ST=Slovakia, C=SR

Issuer: CN=Jaroslav Toth, OU=stu, O=fiit, L=Bratislava, ST=Slovakia, C=SR

Serial number: 398b77dd

Valid from: Sat Mar 18 13:42:42 CET 2017 until: Tue Mar 13 13:42:42 CET 2018

Certificate fingerprints:

MD5: 8F:B1:97:69:F2:70:B6:88:23:7A:49:21:4F:E6:02:AC

SHA1: 1E:77:11:27:7C:68:8E:9D:28:2B:67:3F:54:B7:5A:87:89:0F:97:BE

SHA256: 39:34:EC:BE:FD:9F:56:F1:E4:E6:66:4C:7E:A6:03:86:6C:00:CF:33:46:

33:11:91:9F:AA:49:97:36:1A:37:09

Signature algorithm name: SHA256withRSA

Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false

SubjectKeyIdentifier [

KeyIdentifier [

0000: DA 5C AD D3 27 C5 AB 23 45 78 91 7D DF DB EF 99 \.!.#Ex.....

0010: 6F 03 19 25 o..%]]

Trust this certificate? [no]: yes

Certificate was added to keystore

- Odstránenie zbytočného súboru:

rm -f cloud.cer

Nastavenie kľúčov pre SSH

- Všetky certifikačné súbory sa musia nachádzať v adresári (v OS Windows je to D:\var\certs alebo C:\var\certs podľa toho, na ktorej partícii je uložený kompilovaný projekt):

/var/certs

- Aby unit testy prebehli úspešne, je potrebné vygenerovať cloud SSH pár kľúčov lokálne na MOD servery (self-signed), alebo na stroji, na ktorom sa budú spúšťať unit testy.
- Súbor privátneho kľúča musí mať označenie **ssh_rsa** (v prípade unit testov).
- Heslo na sprístupnenie privátneho kľúča nemôže byť nastavené (v prípade unit testov).
- Linux nástroj ssh-keygen je možné nahradiť nástrojom PuTTYGen v OS Windows.
- Vygenerovanie páru kľúčov, príklad pre MOD:

ssh-keygen -t rsa

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id_rsa): /var/certs/ssh_rsa

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /var/certs/ssh_rsa.

Your public key has been saved in /var/certs/ssh_rsa.pub.

The key fingerprint is:

SHA256:G4DaRVMTXHIZP+OsUMSExGGTJ5b25MBqHafXiiaXbok Jaroslav@Jaro-
NB

The key's randomart image is:

```
+---[RSA 2048]-----+
```

```
| oo+o=*B=. |
```

```
| o ...o%== . |
```

```
| .o =.@..o.|
```

```
| o . . o o.+o.o|
```

```
| . . S .+ .o |
```

```
| + =... |
```

```
| . * .. |
```

```
| E + |
```

```
| . |
```

```
+----[SHA256]-----+
```

- Následne je potrebné vyexportovať verejný kľúč z páru kľúčov a uložiť verejný kľúč na MOD serveri do súboru „known_hosts“, ktorý sa tiež nachádza v adresári /var/certs.
- Návod, ak sa vygenerovaný pár kľúčov nachádza na rovnakom stroji (MOD) a v rovnakom adresári /var/certs (krok platný pre úspešné spustenie unit testov):

cat ssh_rsa.pub >> known_hosts

- Obsah súboru known_hosts musí byť prispôbený na formát, v ktorom prvý reťazec charakterizuje vzdialený stroj, ktorý je potrebné autentifikovať, ďalším reťazcom je algoritmus vygenerovaného páru kľúčov a nakoniec nasleduje verejný kľúč. Za verejným kľúčom sa nemôžu nachádzať ďalšie identifikátory (tie musia byť vymazané).
- Príklad platný pre úspešné spustenie unit testov:

vim known_hosts

```
127.0.0.1 ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQBAQCf/KfkD6cjsaC9duDuHVKYjBJtbLx1W
xyuKf49S54ansZ+WiNTweWQJJUQklmt64yUcuVggj6w/4rPptUDV85IPtKvK9ow097un
ScCMofSgF2nqPcbtm4pV2cZ5EIC9fZbI7QOLMP/i70edXniHH/LNZWNjKkr9AdrxuRb
Sbp0c/QtJq8O8hNtxWiZjM2dW1VBAXoNubTusRMmnM8ZVJL2KhWykp6Ds1GyB3Y
CXm0/Ev2upBkwND4NkKNqHi39OO30vUBOvnKjgt0XvGsexKdqlmzNH9yD7/x+wId
Uzv2ty3JuAKOx/EY13xQ9kV9fvVlkuM8P8ytfwpjYgnj6EjtX
```

Databázové súbory

- Všetky databázové súbory sa musia nachádzať v adresári (logging.db, statistics.db a media.db):

/var/db

Kompilácia a zabalenie aplikácie

a) Pomocou vývojového prostredia IntelliJ

1. Importovanie projektu do IntelliJ – typ projektu Maven.
2. Nastaviť automatické importovanie chýbajúcich knižníc (Maven).
3. Spustiť Maven Goal (test vynechať, ak sa nejedná o testovacie prostredie s keystore repozitárom, v opačnom prípade test padne):

clean validate compile test assembly:single

4. Spustiteľný súbor sa vytvorí v adresári /target s názvom:

mod-remote-admin-0.0.1-jar-with-dependencies.jar.

5. Aplikáciu je možné spustiť príkazom:

**java -jar mod-remote-admin-0.0.1-jar-with-dependencies.jar
[LOGGING_DB_PATH] [MOVIE_DB_PATH] [XML_PATH]
[TRUSTSTORE_PATH] [KNOWN_HOSTS_FILE]**

Význam argumentov opisuje Tabuľka 36 (všetky argumenty sú povinné).

Tabuľka 36. Význam vstupných argumentov

Argument	Význam
[LOGGING_DB_PATH]	Umiestnenie databázy na logovanie (odporúčaná cesta: /var/db/logging.db).
[MOVIE_DB_PATH]	Umiestnenie media databázy (odporúčaná cesta: /var/db/media.db)
[XML_PATH]	Umiestnenie globálneho konfiguračného súboru s nastaveniami časovačov a iných konštánt (odporúčaná cesta: /var/config/varData.xml).
[XML_ADVERT_PATH]	Umiestnenie konfiguračného súboru k reklamám.
[TRUSTSTORE_PATH]	Umiestnenie repozitára s dôveryhodnými certifikátmi cloudu (TLS pripojenie).
[KNOWN_HOSTS_FILE]	Umiestnenie súboru, ktorý obsahuje dôveryhodné verejné kľúče cloudu (SSH autentifikácia).

b) Pomocou príkazového riadku

1. Nainštalovať Maven a nastaviť cesty; návod pre Windows OS:

<https://maven.apache.org/guides/getting-started/windows-prerequisites.html>

2. V adresári, kde sa nachádza pom.xml spustiť príkaz (test vynechať, ak sa nejedná o testovacie prostredie s keystore repozitárom, v opačnom prípade test padne):

mvn clean validate compile test assembly:single

3. Aplikáciu je možné rovnako spustiť pomocou java -jar príkazu.

Globálny konfiguračný súbor (XML)

- Konfiguračný súbor sa musí nachádzať staticky na ceste:
\var\config\varData.xml
- V prípade, že tento súbor neexistuje, pri štarte aplikácie sa vytvorí predvolený súbor s predvolenými hodnotami.
- V prípade, že súbor obsahuje zle naformátované hodnoty, ale v správnych poliach, aplikujú na predvolené hodnoty.
- V prípade, že súbor má narušenú štruktúru polí, vygeneruje sa výnimka, ktorá sa uloží do logovacej databázy (aplikačné logy).
- XML súbor sa načítava iba pri štarte aplikácie.
- Príklad predvoleného XML súboru:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<globalConfiguration>
  <maxRetries>3</maxRetries>
  <retriesOffset>1000</retriesOffset>
  <coldStartOffset>20</coldStartOffset>
  <localMachineId>
    <serverId>ANONYMOUS</serverId>
    <secret></secret>
    <truststorePassword>password</truststorePassword>
  </localMachineId>
  <remoteMachineId>
    <ipv4Address>127.0.0.1</ipv4Address>
    <port>8443</port>
    <applicationLogs>/remote/application_logs</applicationLogs>
    <systemLogs>/remote/system_logs</systemLogs>
    <statistics>/remote/statistics</statistics>
    <commands>/remote/commands</commands>
    <adverts>/remote/adverts</adverts>
  </remoteMachineId>
  <applicationLogs>
    <enabled>>false</enabled>
    <interval>28800</interval>
    <maxEntries>100</maxEntries>
  </applicationLogs>
</globalConfiguration>
```

```

        <offset>50</offset>
</applicationLogs>
<systemLogs>
    <enabled>>false</enabled>
    <interval>28800</interval>
    <maxEntries>100</maxEntries>
    <offset>50</offset>
</systemLogs>
<statistics>
    <enabled>>false</enabled>
    <interval>28800</interval>
    <lastHours>48</lastHours>
</statistics>
<queries>
    <enabled>>true</enabled>
    <interval>28800</interval>
</queries>
<rotationLog>
    <interval>43200</interval>
    <sizeDB>1024</sizeDB>
</rotationLog>
<advertGlobalConfiguration>
    <enabled>>true</enabled>
    <interval>28800</interval>
</advertGlobalConfiguration>
</globalConfiguration>

```

- Opis polí XML súboru sa nachádza v Tabuľka 37.

Tabuľka 37. Opis polí XML súboru

Názov	Obm.	Opis
<globalConfiguration>	-	Adresár obsahujúci všetky ostatné nastavenia.
<maxRetries>	> 0	Maximálny počet pokusov pri nadviazaní spojenia s cloud IP adresou na zvolenom porte. Po neúspešných pokusoch sa vygenerujú logy, pričom aplikácia pokračuje v činnosti ďalej (v nasledujúcom cykle budú prebiehať opätovné pokusy).
<retriesOffset>	> 0	Doba v milisekundách medzi ukončením jedného pokusu a začiatkom nasledujúceho pokusu – ochrana pred zahltením pripojenia.
<coldStartOffset>	> 0	Doba po spustení serveru, ktorá musí ubehnúť pred tým, ako sa odošlú prvé správy / záznamy na cloud (v sekundách) – ochrana pred chybami aplikácie, ktoré by mohli cyklicky po spustení aplikácie zahlcovať cloud.
<localMachineId>	-	Adresár obsahujúci údaje, ktoré identifikujú lokálny MOD stroj.
<serverId>	dĺžka	Lokálny textový identifikátor servera – mal by byť unikáty.

Názov	Obm.	Opis
	> 0	Slúži aj ako autentifikačný údaj v http správach – meno.
<secret>	-	Autentifikačné heslo použité v HTTP správach. Cloud môže overovať identitu klienta pomocou kombinácie meno, heslo (zabezpečenie). Obojstranná autentifikácia pomocou certifikátov je pomalá a nepraktická vzhľadom na potrebnú ďalšiu správu certifikátov aj na cloude.
<truststorePassword>	-	Heslo odomykajúce prístup do repozitára, ktorý obsahuje dôveryhodné certifikáty cloudu.
<remoteMachineId>	-	Adresár obsahujúci údaje, ktoré identifikujú vzdialený cloud. Takýchto adresárov, môže byť aj viac (load balancing), aplikácia ale momentálne podporuje iba jeden vzdialený server.
<ipv4Address>	formát IPv4	IP adresa, na ktorom cloud počúva prichádzajúce HTTPS spojenia.
<port>	> 0	Port (zabezpečený), na ktorom cloud počúva prichádzajúce HTTPS spojenia.
<systemLogs>	dĺžka > 0	Webová stránka (bez identifikátora koncového zariadenia), na ktorú sa budú posielat' systémové logy.
<applicationLogs>	dĺžka > 0	Webová stránka (bez identifikátora koncového zariadenia), na ktorú sa budú posielat' aplikačné logy.
<statistics>	dĺžka > 0	Webová stránka (bez identifikátora koncového zariadenia), na ktorú sa budú posielat' štatistiky.
<commands>	dĺžka > 0	Webová stránka (bez identifikátora koncového zariadenia), z ktorej sa bude MOD server dopytovať na príkazy od cloudu (napr. otvorenie ssh spojenia na správu MOD servera – reverzné SSH).
<adverts>	dĺžka > 0	Webová stránka (bez identifikátora koncového zariadenia), z ktorej sa MOD dopytuje na aktuálne nastavenia reklám (sťahovanie nastavení do XML súbory).
<applicationLogs>	-	Adresár obsahujúci nastavenia synchronizácie aplikačných logov. Podobné adresáre s rovnakými nastaveniami (alebo iba niektorými nastaveniami) – <systemLogs>, <statistics>, <queries> a <advertGlobalConfiguration>. V <queries> adresári sa nenachádza maximálny počet záznamov posielaných na server, keďže žiadne záznamy sa neposielajú (http get správa). V <statistics> adresáry nie je špecifikovaný offset a maximálny počet záznamov medzi posielanými balíkmi agregovaných štatistík (namiesto toho je tu parameter

Názov	Obm.	Opis
		<lastHours>)
<enabled>	true / false	Povolenie / zakázanie posielania aplikačných logov, štatistík alebo napr. dopytovania po nových príkazoch z cloudu.
<interval>	> 0	Ako často sa majú záznamy synchronizovať so serverom – doba medzi ukončením poslednej synchronizácie a začiatkom nasledujúcej synchronizácie v sekundách. Pri príkazoch ide o interval dopytovania po nových príkazoch na serveri určených pre aktuálny MOD server. Pri rotácii logov ide o interval vykonávania rotácie v systéme.
<maxEntries>	> 0	Počet záznamov, ktoré sa môžu poslať na cloud v jednom čase. V prípade, že sa na servery nachádza viacero záznamov, ďalšie záznamy sú poslané s oneskorením (offset).
<offset>	> 0	Doba medzi odoslaním jedného balíka záznamov a začiatkom synchronizácie nasledujúceho balíka.
<lastHours>	> 0	Rozpätie času v hodinách, ktoré definuje rozsah štatistík, ktoré môžu byť agregované a poslané naraz v jednom cykle.
<rotationLog>	-	Adresár, ktorý obsahuje nastavenia rotácie logov v databáze.
<sizeDB>	> 0	Maximálna veľkosť logov v databáze (MB) – následne sa začnú mazať staršie záznamy.

Konfiguračný súbor pre reklamy

- Príklad konfiguračného súboru pre reklamy

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<advertsConfiguration>
  <Buses>40</Buses>
  <Adverts>
    <Advert>
      <Path>6f4735e21ae676482dd29af6d0ddb6e6.360.mp4</Path>
      <Advert_ID>5</Advert_ID>
      <PlayFrom>25.04.2017</PlayFrom>
      <PlayTo>28.07.2018</PlayTo>
      <Count>11</Count>
    </Advert>
    <Advert>
      <Path>17cc207beaa13343e8d28083e894606a.720.mp4</Path>
      <Advert_ID>20</Advert_ID>
      <PlayFrom>19.04.2017</PlayFrom>
      <PlayTo>25.05.2017</PlayTo>
      <Count>7777</Count>
    </Advert>
    <Advert>
      <Path>512d43a3df1aa8a9386367ab6eac2fa8.360.mp4</Path>
      <Advert_ID>10</Advert_ID>
      <PlayFrom>19.04.2017</PlayFrom>
```

```

        <PlayTo>25.04.2017</PlayTo>
        <Count>5000</Count>
    </Advert>
</Adverts>
</advertsConfiguration>

```

- Opis polí XML súboru sa nachádza v tabuľke nižšie

Názov	Obm.	Opis
<advertsConfiguration>	-	Adresár obsahujúci nastavenia reklám
<Buses>	> 0	Počet autobusov vo flotile
<Adverts>	-	Adresár obsahujúci zoznam reklám
<Advert>	-	Adresár obsahujúci informácie o konkrétnej reklame
<Path>	-	Cesta k súboru s reklamou
<Advert_ID>	> 0	Identifikátor reklamy v Cloude
<PlayFrom>	DD.MM.RRRR	Dátum od kedy sa bude reklama prehrávať
<PlayTo>	DD.MM.RRRR	Dátum do kedy sa bude reklama prehrávať
<Count>	> 0	Počet prehratí reklám na celú flotilu

Opis HTTP správ

Posielané dátové entity

Opis JSON polí pre entitu „ApplicationLog“ – aplikačné logy generované webovým portálom alebo inou aplikáciou na MOD. Prvý riadok opisuje názov koreňového adresára JSON-u, druhý riadok opisuje presné poradie atribútov, 3-riadok vynecháva neinicializované parametre vo výstupnom JSON-e, ďalšie bloky kódu mapujú JAVA atribúty na JSON názvy atribútov. Z triedy SeverityLevel je použité na identifikáciu iba pole „id“. Táto JSON entita sa vysiela na cloud v http post správe. Opis polí sa nachádza v dokumentácií pre zbieranie logov.

```

@JsonRootName(value = "ApplicationLog")
@JsonPropertyOrder({"logger", "eventDateTime", "severity", "message", "thrown"})
@JsonInclude(JsonInclude.Include.NON_NULL)
public class PortalLog implements DataToPost
...

@JsonProperty("severity")
public SeverityLevel getFkLevel() {
    return fkLevel;
}

@JsonProperty("eventDateTime")
public Long getEventDatetime() {
    return eventDatetime;
}

```

```

@JsonProperty("logger")
public String getLogger() {
    return logger;
}
@JsonProperty("message")
public String getMessage() {
    return message;
}
@JsonProperty("thrown")
public String getThrowable() {
    return throwable;
}

public class SeverityLevel
@JsonProperty
public Integer getId() {
    return id;
}
...

```

Opis JSON polí pre entitu „SystemEvents“ – systémové logy. Prvý riadok opisuje názov koreňového adresára JSON-u, druhý riadok opisuje presné poradie atribútov, 3-riadok vynecháva neinicializované parametre vo výstupnom JSON-e, ďalšie bloky kódu mapujú JAVA atribúty na JSON názvy atribútov. Táto JSON entita sa vysiela na cloud v http post správe. Opis polí sa nachádza v dokumentácii pre zbieranie logov.

```

@JsonProperty(value = "SystemLog")
@JsonPropertyOrder({"facility", "host", "eventDateTime", "severity", "eventSource",
"eventUser", "eventCategory", "eventId", "eventLogType", "syslogTag", "message"})
@JsonPropertyInclude(JsonInclude.Include.NON_NULL)
public class SystemEvents implements DataToPost
...
@JsonProperty("eventDateTime")
public Long getDeviceReportedTime() {
    return deviceReportedTime;
}
@JsonProperty("facility")
public String getFacility() {
    return facility;
}
@JsonProperty("severity")
public Integer getSeverity() {
    return severity;
}
@JsonProperty("host")
public String getFromHost() {
    return fromHost;
}
@JsonProperty("message")
public String getMessage() {
    return message;
}
@JsonProperty("eventSource")
public String getEventSource() {
    return eventSource;
}
@JsonProperty("eventUser")
public String getEventUser() {
    return eventUser;
}
@JsonProperty("eventCategory")
public Integer getEventCategory() {
    return eventCategory;
}
}

```

```

@JsonProperty("eventId")
public Integer getEventId() {
    return eventId;
}
@JsonProperty("eventLogType")
public Integer getEventLogType() {
    return eventLogType;
}
@JsonProperty("sysLogTag")
public String getSysLogTag() {
    return sysLogTag;
}
...

```

Nasledujúca entita slúži na deserializáciu prichádzajúcich príkazov v JSON formáte do entity v JAVA. Atribút „id“ je nemenný – identifikuje o aký príkaz ide. Hodnota „id“ 0 predstavuje prázdny príkaz – žiadne nové aktualizácie / príkazy pre MOD z cloudu (takáto správa by mala byť posielaná ako odpoveď na http get správu, v opačnom prípade sa vygeneruje výnimka kvôli nedostupnosti serveru). Štruktúra poľa „payload“ sa mení v závislosti na hodnote „id“ (v ďalších častiach programu je vetvene mapovaná na ďalšie entity).

```

@JsonRootName(value = "Command")
@JsonPropertyOrder({"id", "payload"})
public class Command
...
@JsonProperty("id")
public void setId(Integer id) {
    this.id = id;
}
@JsonProperty("payload")
public void setPayload(JsonNode payload) {
    this.payload = payload;
}
...

```

Nasledujúca entita (deserializácia JSON-u do POJO entity) slúži na opis príkazu na otvorenie reverzného SSH tunelu medzi MOD serverom (zdroj) a cloudom na zadefinovanej IP adrese (cieľ). Následne je možné tento tunel využiť na administráciu MOD zariadenia z cloudu alebo iného zariadenia. Táto entita sa mapuje do poľa payload v predchádzajúcej entite pri id = 1.

```

@JsonRootName(value = "SshCommand")
@JsonPropertyOrder({"remoteIp", "remotePort", "remoteUser", "password"})
public class SshCommand
...
@JsonProperty("remoteIp")
public void setRemoteIp(String remoteIp) {
    this.remoteIp = remoteIp;
}
@JsonProperty("remotePort")
public void setRemotePort(Integer remotePort) {
    this.remotePort = remotePort;
}
@JsonProperty("remoteUser")
public void setRemoteUser(String remoteUser) {
    this.remoteUser = remoteUser;
}
}

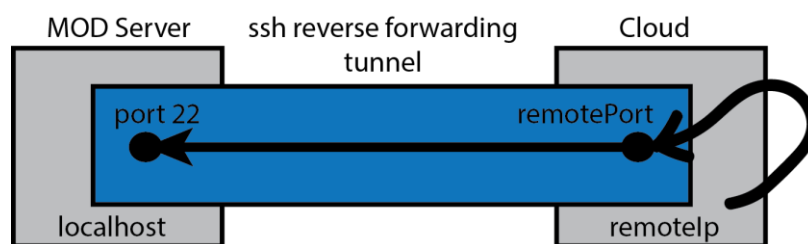
```

```
@JsonProperty("password")
public void setPassword(String password) {
    this.password = password;
}
...

```

Vysvetlenie jednotlivých polí serializovaného JSON-u (Obrázok 16 je na vysvetlenie polí takisto možné použiť – opis vytvárania SSH spojení aj s reprezentatívnyimi shell príkazmi):

- “remoteIp” – IP adresa, na ktorú sa má vytvoriť reverzný forwardovací tunel (manažment konzola).
- “remotePort” – Cieľový port, na ktorý sa bude pripájať používateľ na localhost-e u vzdialenej konzoly (tento port je automaticky presmerovaný cez forwardovací tunel na MOD).
- “remoteUser” – Prihlasovacie meno na manažment konzolu (SSH login).
- “password” – Prihlasovacie heslo na manažment konzolu (SSH password).



```
mod# ssh -fNR [remotePort]:localhost:22 [remoteUser]@[remotelp]
password: [password]
cloud# ssh -p [remotePort] management@127.0.0.1
password: xxxxxxxx

```

Obrázok 16. SSH koncepcia

Nasledovná trieda udáva formát (JSON), v akom sa budú posielat’ agregované štatistiky na cloud (HTT POST). Trieda obsahuje 3 atribúty:

- Štartovací čas (startTime) – Čas, od kedy sú štatistiky agregované.
- Konečný čas (endTime) – Koncový hraničný čas, do ktorého sú štatistiky agregované.
- Zoznam zagregovaných štatistík (List<AggregatedStatistics>) – Zoznam zagregovaných štatistík z prehratia pre každý multimedialny záznam, ktorý sa prehral / zobrazil aspoň jedenkrát.

```
public class StatisticsForCloud {
    private Long startTime;
    private Long endTime;
    private List<AggregatedStatistics> aggregatedStatistics;
    ...
}

```


Nasledujúca trieda znázorňuje formát agregovaných štatistík, v tomto prípade s explicitne špecifikovanými JSON poľami (JsonProperty anotácia), názvom koreňovej entity (JsonRootName) a poradím atribútov, ako sa budú serializovať do výstupného JSONu. Zaznamenáva sa typ média, ktorý je prehrávaný, početnosť prehrania alebo zobrazenia a identifikátor multimediálneho súboru.

```

@JsonRootName(value = "AggregatedStatistics")
@JsonPropertyOrder({"mediaId", "mediaType", "count"})
@JsonInclude(JsonInclude.Include.NON_NULL)
public class Statistic {
    ...
    @JsonProperty("mediaId")
    public Integer getMediaId() {
        return mediaId;
    }
    @JsonProperty("mediaType")
    public Integer getMediaType() {
        return mediaType;
    }
    @JsonProperty("count")
    public Integer getCount() {
        return count;
    }
    ...
}

```

Priradenia identifikátorov média:

```

public static enum TYPE {
    MOVIE(1),
    MUSIC(2),
    PICTURE(3),
    ADVERT(4);
    ...
}

```

Opis štruktúry správ pre posielanie dát na cloud

Správa posielaná v smere MOD server – cloud:

- Cieľová IP adresa a port sú definované v XML konfiguračnom súbore.
- Typ správy – http post request.
- Content-Type – application/json.
- Cieľová stránka – podľa nastavenia v <RemoteMachineId>.
- Autentifikácia – preemptívna autentifikácia - meno („serverId“) a heslo („secret“).
- Telo správy – serializovaná entita typu List<DataToPost>.
- Nastavené TLSv2 overovanie cloud-u.
- Nastavenie cache-ovania autentifikačných údajov.

Správa posielaná v smere cloud – MOD server:

- Cieľová IP adresa – IP adresa PAT brány + preložený port,
- Typ správy – http response.
- Status code – 200.
- Content-type – text/plain.

- Telo správy – reťazec „acknowledged“.

Opis štruktúry správ pre dopytovanie sa na príkazy od cloud-u

Správa posielaná v smere MOD server – cloud:

- Cieľová IP adresa a port sú definované v XML konfiguračnom súbore.
- Typ správy – http get request.
- accept – application/json.
- Cieľová stránka – podľa nastavenia v <RemoteMachineId> – pole <commands>.
- Autentifikácia – preemptívna autentifikácia - meno („serverId“) a heslo („secret“).
- Nastavené TLSv2 overovanie cloud-u.
- Nastavenie cache-ovania autentifikačných údajov.

Správa posielaná v smere cloud – MOD server:

- Cieľová IP adresa – IP adresa PAT brány + preložený port,
- Typ správy – http response.
- Status code – 200.
- Content-type – application/json.
- Telo správy – serializovaný JSON s formátom entity „Command“ (opis entity ako aj možných payload štruktúr sa nachádzajú v predchádzajúcich kapitolách).

Zoznam ID príkazov – sumarizácia

Tabuľka 38 obsahuje zosumarizovaný zoznam všetkých podporovaných príkazov na MOD, ktoré môže server vykonať na príkaz z cloudu.

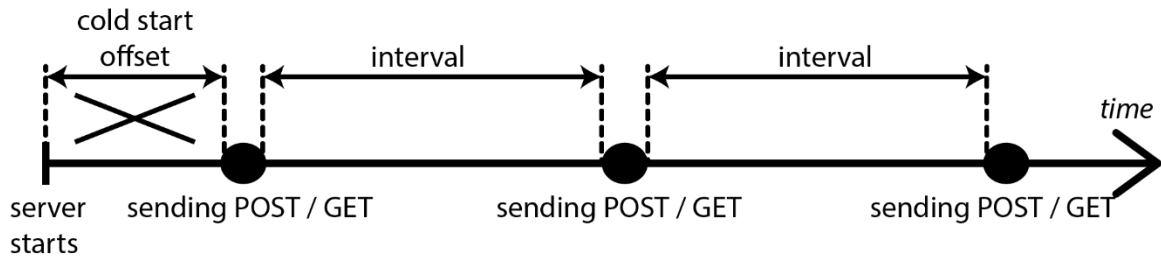
Tabuľka 38. Zoznam príkazov

ID	Názov entity	Opis
0	-	Prázdny príkaz – žiadna nový povel z cloudu, payload neexistuje. Tento príkaz sa používa iba na potvrdenie existencie cloudu.
1	SshCommand	Príkaz na vytvorenie SSH reverzného forwardovacieho tunelu z MOD na zvolený manažovací server.

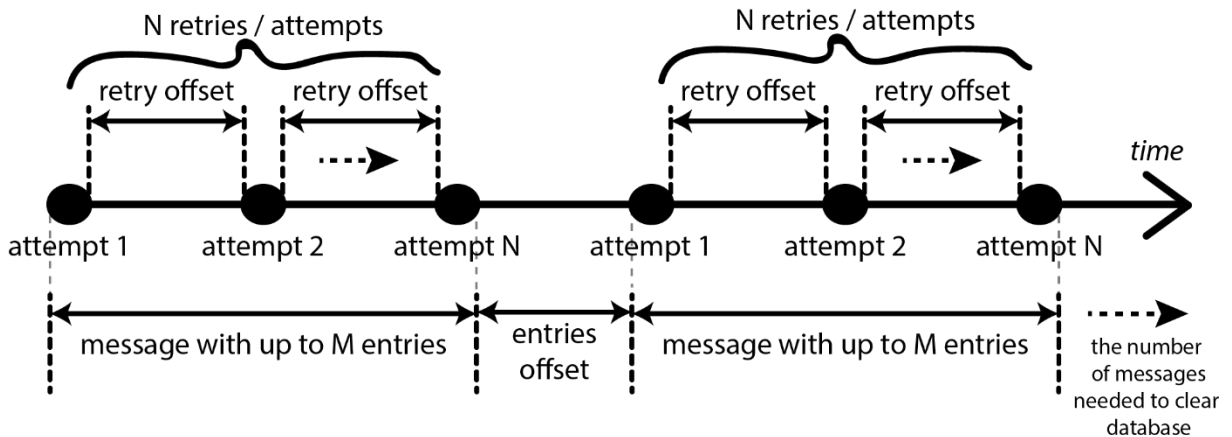
Diagram udalostí v rámci vlákien

Opis časovania http správ je znázorňuje Obrázok 17. Táto postupnosť udalostí platí pre posielanie POST správ, ktoré obsahujú databázové položky vo formáte JSON. Pri GET správach nemá význam fragmentácia správy do viacerých častí (t. j. „number of messages needed to clear database“ = 1);

Global view on threads



HTTPS POST / GET Procedure



Obrázok 17. Časovanie HTTP správ

5.1.4 Vytvorenie databázovej štruktúry (médiá)

Program na vytvorenie databázovej štruktúry sa nazýva InitDb. Skladá sa zo zdrojových súborov init-db.cpp a error.cpp.

Po spustení vytvorí novú databázu s preddefinovanou štruktúrou a naplní tabuľky s hudobnými žánrami a filmovými kategóriami z poskytnutých XML súborov. V prípade nastatia chyby, je táto chyba uložená v logovacej databáze. Ak neexistuje logovacia databáza, program vypisuje chybové hlášky na štandardný výstup.

Knižnice:

- sqlite3 (napr. inštalácia balíka sqlite-autoconf-3170000.tar.gz)
- pugixml (napr. apt-get install libpugixml-dev)

Kompilácia:

```
g++ -o InitDb -std=c++11 init-db.cpp error.cpp -lsqlite3 -lpugixml
```

Spustenie:

```
./InitDb <name_of_db><path_to_logging.db> <genres_xml_path> <categories_xml_path>
```

genres_config.xml súbor musí obsahovať všetky hudobné žánre, s ktorými systém bude pracovať. Hudobné žánre nie sú jazykovo rozlíšené. Súbor musí vždy obsahovať defaultnú kategóriu Other.

- Príklad konfiguračného súboru pre hudobné žánre

```
<Genres>
  <Genre>60's</Genre>
  <Genre>Disco</Genre>
  ...
  <Genre>Other</Genre>
  ...
  <Genre>Top 40</Genre>
</Genres>
```

- Opis polí XML súboru sa nachádza v tabuľke nižšie

Názov	Obm.	Opis
<Genres>	-	Adresár obsahujúci zoznam hudobných žánrov
<Genre>	-	Názov konkrétneho hudobného žánru

category_config.xml súbor musí obsahovať všetky kategórie filmov (minimálne v predvolenom jazyku systému), s ktorými systém bude pracovať. Každá kategória môže mať viacero prekladov (v príklade Akčné, Action).

- Príklad konfiguračného súboru pre filmové kategórie

```
<Categories>
  <Category>
    <Meta
      name="Action"
      lang="en"/>
    <Meta
      name="Akčné"
      lang="sk"/>
  </Category>
  <Category>
    <Meta
      name="Animation"
      lang="en"/>
    <Meta
      name="Animované"
      lang="sk"/>
  </Category>
  ...
</Categories>
```

- Opis polí XML súboru sa nachádza v tabuľke nižšie

Názov	Obm.	Opis
<Categories>	-	Adresár obsahujúci zoznam filmových kategórií
<Category>	-	Adresár obsahujúci zoznam metadát konkrétnej kategórie

<Meta>	-	Metadáta konkrétnej kategórie v danom jazyku Toto pole ma dva atribúty: <ul style="list-style-type: none"> • name – názov kategórie v danom jazyku • lang – kód jazyku
--------	---	---

5.1.5 Naplňanie databáz

Naplnenie databázy s reklamami

Program na naplnenie databázy s reklamami sa nazýva AdvertDBFiller. Skladá sa zo zdrojových súborov advert-db-filler.cpp, advert.cpp a error.cpp.

Po spustení program prečíta konfiguračný súbor, ktorý je zadaný na vstupe. Z konfiguračného súboru vyberie postupne všetky reklamy a vloží ich do databázy. Počet prehraní z konfiguračného súboru sa vydolí počtom autobusov vo flotile a táto hodnota sa uloží ako počet prehraní reklamy do databázy. Pred vložením do databázy sa otestuje, či zadaná cesta k reklame je platný súbor na disku. Ak nie, daná reklama sa nevloží do databázy.

V prípade nastatia chyby, je táto chyba uložená v logovacej databáze. Ak neexistuje logovacia databáza, program vypisuje chybové hlášky na štandardný výstup.

Knižnice:

- sqlite3 (napr. inštalácia balíka sqlite-autoconf-3170000.tar.gz)
- pugixml (napr. apt-get install libpugixml-dev)

Kompilácia:

```
g++ -o AdvertDbFiller -std=c++11 advert-db-filler.cpp advert.cpp error.cpp -lsqlite3 -lpugixml
```

Spustenie:

```
./AdvertDbFiller <path_to_media.db> <path_to_logging.db> <path_to_adverts_directory> <path_to_adverts_config>
```

Priečink s reklamami (path_to_adverts_directory) obsahuje súbory s jednotlivými reklamami.

Príklad a opis konfiguračného súboru pre reklamy je uvedený v kapitole 5.1.3 .

Naplnenie databázy s hudbou

Program na naplnenie databázy s hudbou sa nazýva MusicDBFiller. Skladá sa zo zdrojových súborov music-db-filler.cpp, music.cpp a error.cpp.

Po spustení program vyhľadá všetky priečinky v adresáry s hudbou zadanom na vstupe. Následne prechádza každým z nájdených priečinkov a vyhľadá v ňom XML súbor s metadátami k albumu. Ak priečink takýto XML súbor neobsahuje, priečink sa

preskočí. Z nájdeného XML súboru program vyberie metadáta o danom albume, zoznam všetkých pesničiek a obrázkov s obalom albumu. Následne sa tieto údaje vložia do databázy, predtým sa však skontroluje či sú nájdené cesty k súborom platné. Ak nie, daný súbor sa ignoruje.

V prípade nastatia chyby, je táto chyba uložená v logovacej databáze. Ak neexistuje logovacia databáza, program vypisuje chybové hlášky na štandardný výstup.

Knižnice:

- sqlite3 (napr. inštalácia balíka sqlite-autoconf-3170000.tar.gz)
- pugixml (napr. apt-get install libpugixml-dev)

Kompilácia:

```
g++ -o MusicDbFiller -std=c++11 music-db-filler.cpp music.cpp error.cpp -lsqlite3 -lpugixml
```

Spustenie:

```
./MusicDbFiller <path_to_media.db> <path_to_logging.db> <path_to_albums_directory>
```

- Príklad konfiguračného súboru pre hudobný album

```
<Album>
  <Name>Best of CD1</Name>
  <Artist>Elan</Artist>
  <Year>2000</Year>
  <Genre>Pop</Genre>
  <CoverImage>best_of_cd1.jpg</CoverImage>
  <Songs>
    <Song filename="Elan - Netrpezlivi.mp3" name="Netrpezlivi" />
    <Song filename="Elan - Tulaci podchodoch.mp3" name="Tulaci
podchodoch" />
    ...
  </Songs>
</Album>
```

- Opis polí XML súboru sa nachádza v tabuľke nižšie

Názov	Obm.	Opis
<Album>	-	Adresár obsahujúci metadáta o albume
<Name>	-	Názov albumu
<Artist>	-	Názov interpreta
<Year>	> 0	Rok vydania albumu
<Genre>	-	Žáner albumu
<CoverImage>	-	Cesta k obrázku obalu albumu
<Songs>	-	Adresár obsahujúci zoznam pesničiek

<Song>	-	Informácie o danej pesničke Toto pole ma dva atribúty: filename – názov súboru pesničky name – názov pesničky, ktorý bude zobrazený pri prehrávaní
--------	---	---

Naplnenie databázy s filmami

Program na naplnenie databázy s filmami sa nazýva MediaDBFiller. Skladá sa zo zdrojových súborov media-db-filler.cpp, movie.cpp a error.cpp.

Po spustení program vyhledá všetky priečinky v adresári s filmami zadanom na vstupe. Následne prechádza každým z nájdených priečinkov a vyhledá v ňom XML súbor s metadátami o filme. Ak priečinok takýto XML súbor neobsahuje, priečinok sa preskočí. Z nájdeného XML súboru program vyberú metadáta o danom filme, zoznam všetkých zvukových stôp a tituliek a obrázkov s obalom filmu. Následne sa tieto údaje vložia do databázy, predtým sa však skontroluje či sú nájdené cesty k súborom platné. Ak nie, daný súbor sa ignoruje. Ďalšou podmienkou aby bol film vložený do databázy je, že musí obsahovať metadáta a zvukovú stopu minimálne v predvolenom jazyku servera.

V prípade nastatia chyby, je táto chyba uložená v logovacej databáze. Ak neexistuje logovacia databáza, program vypisuje chybové hlášky na štandardný výstup.

Knižnice:

- sqlite3 (napr. inštalácia balíka sqlite-autoconf-3170000.tar.gz)
- pugixml (napr. apt-get install libpugixml-dev)

Kompilácia:

```
g++ -o MediaDbFiller -std=c++11 media-db-filler.cpp movie.cpp error.cpp -lsqlite3 -lpugixml
```

Spustenie:

```
./MediaDbFiller <path_to_media.db> <path_to_logging.db> <path_to_movies_directory> <path_to_portal_xml>
```

- Príklad konfiguračného súboru pre film

```
<Movie>
  <Picture>avatar.jpg</Picture>
  <Metadatas>
    <Metadata>
      <Name>Avatar</Name>
      <Language>sk</Language>
      <Director>James Cameron</Director>
      <Plot>strucny obsah filmu... </Plot>
      <Stars>Sam Worthington,Zoe Saldana,Stephen Lang</Stars>
      <Country>USA</Country>
      <Year>2009</Year>
      <Time>166</Time>
```

```

    <Category>Akčné,Dobrodužné</Category>
    <MovieFile>avatar_cz.mp4</MovieFile>
    <Subtitles>avatar_cz.vtt</Subtitles>
  </Metadata>
  <Metadata>
    <Name>Avatar</Name>
    <Language>en</Language>
    <Director>James Cameron</Director>
    <Plot>obsah filmu v inom jazyku...</Plot>
    <Stars>Sam Worthington,Zoe Saldana,Stephen Lang</Stars>
    <Country>USA</Country>
    <Year>2009</Year>
    <Time>166</Time>
    <Category>Action,Adventure</Category>
    <MovieFile>avatar_en.mp4</MovieFile>
    <Subtitles>avatar_en.vtt</Subtitles>
  </Metadata>
</Metadatas>
</Movie>

```

- Opis polí XML súboru sa nachádza v tabuľke nižšie

Názov	Obm.	Opis
<Movie>	-	Adresár obsahujúci informácie o filme
<Picture>	-	Cesta k obrázku filmu
<Metadatas>	-	Adresár obsahujúci zoznam metadát v rôznych jazykoch
<Metadata>	-	Metadáta k filmu v konkrétnom jazyku
<Name>	-	Názov filmu
<Language>	-	Kód jazyku, v ktorom sú dané metadáta
<Director>	-	Názov režiséra filmu
<Plot>	-	Stučný obsah filmu
<Stars>	-	Mená hercov hrajúcich vo filme (oddelené čiarkou)
<Country>	-	Názov krajiny pôvodu filmu
<Year>	> 0	Rok natočenia filmu
<Time>	> 0	Dĺžka trvania filmu
<Category>	-	Zoznam kategórií, do ktorých film spadá (oddelené čiarkou)
<MovieFile>	-	Cesta k zvukovej stope filmu (v danom jazyku)
<Subtitles>	-	Cesta k titulčiam filmu (v danom jazyku)

Naplnenie databázy s obrázkami

Program na naplnenie databázy s obrázkami sa nazýva PictureDBFiller. Skladá sa zo zdrojových súbor picture-db-filler-main.cpp, picture-db-filler.cpp, picture.cpp.

Po spustení program vyhľadá všetky cesty súborov v zadanom adresári na vstupe a v jeho podadresároch. Následne prechádza jednotlivé cesty a zisťuje, či niektorá obsahuje XML súbor k albumu. Ak sa nájde XML súbor, tak sa prečítajú informácie o albume ako meno, opis a jazyk. Potom sa vykoná kontrola, či už je daný album v databáze. Ak nie je, tak sa pridá do databázy. Takto sa prejdú všetky cesty. Potom sa z nájdených ciest z predhadzajúceho kroku zisťuje, či daná cesta obsahuje súbor z podporovaných formátov pre obrázky. Podporované sú tieto formáty: jpg, gif, png, tiff. Kontrola formátu obrázku prebieha na základe špecifických znakov na začiatku súboru pre daný formát obrázku. Potom sa informácie o obrázku uložia do databázy.

Knižnice:

- sqlite3 (napr. inštalácia balíka sqlite-autoconf-3170000.tar.gz)
- pugixml (napr. apt-get install libpugixml-dev)

Kompilácia:

```
g++ -o PictureDbFiller -std=c++11 picture-db-filler-main.cpp picture-db-filler.cpp
picture.cpp error.cpp -lsqlite3 -lpugixml
```

Spustenie:

```
./PictureDbFiller <path_to_media.db> <path_to_logging.db> <path_to_picture_directory>
```

- Príklad konfiguračného súboru pre album

```
<?xml version="1.0"?>
<METADATA>
  <NAME>Autá</NAME>
  <DESCRIPTION>Kolekcia obrázkov o autách.</DESCRIPTION>
  <LANGUAGE>sk</LANGUAGE>
</METADATA>
```

- Opis polí XML súboru sa nachádza v tabuľke nižšie

Názov	Obm.	Opis
<METADATA>	-	Metadáta k albumu
<NAME>	-	Názov albumu
<DESCRIPTION>	-	Opis albumu
<LANGUAGE>	-	Kód jazyku, v ktorom sú dané metadáta

5.1.6 Rotácia záznamov

Program na rotáciu záznamov kontroluje periodický záznamy v databáze pre logy (logging.db) a v databáze pre štatistiky (statistics.db). Záznamy, ktoré sú označené ako synchronizované s cloudom (FLAG_SEND = 1), tak sú odstránené z databáz.

Ďalej sa vykonáva aj kontrola veľkostí oboch databáz, aby nezabrali viac pamäťového miesta ako je pre nich vyhradené. V prípade zistenia, že veľkosť databázy presiahla množstvo vyhradeného pamäťového priestoru, tak sa všetky záznamy z databáz odstránia.

Na to, aby sa veľkosť databáz po odstránení záznamov optimalizovala, je potrebné v databázach nastaviť VACUUM:

```
$sqlite3 database_name "VACUUM;"
```

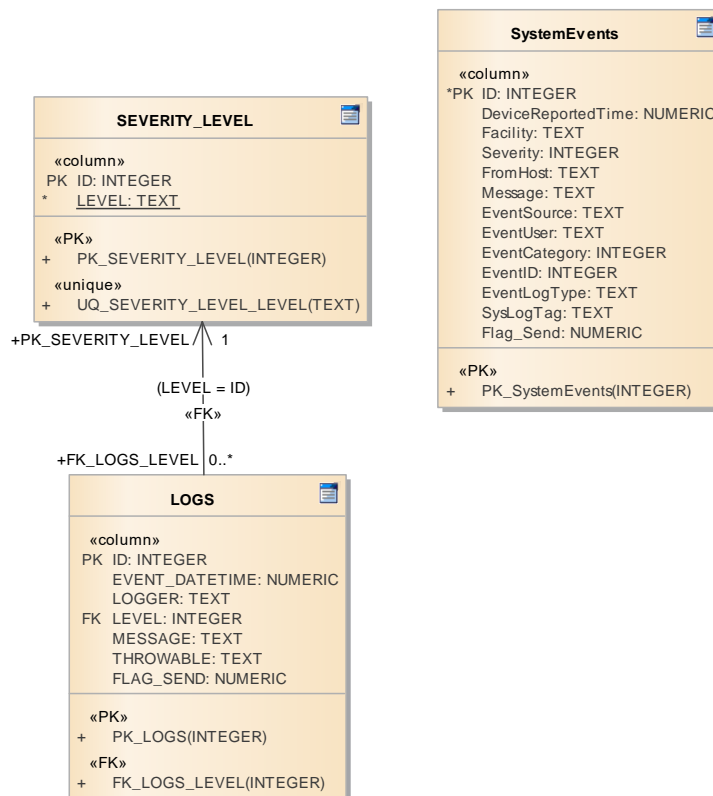
Z globálneho [konfiguračného súboru](#) (element rotationLog) sa čítajú tieto parametre:

- interval – perióda kontroly záznamov v databáze
- sizeDB – maximálna povolená veľkosť databázy

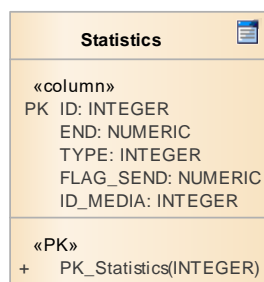
5.1.7 Údajové modely

Finálne údajové modely:

- Media databáza (filmy, obrázky, reklamy a hudba) - Obrázok 18.
- Logovacia databáza (systémové a aplikacné logy) - Obrázok 19.
- Štatistická databáza (štatistiky z prehrávania multimédií a reklamy) - Obrázok 20.



Obrázok 19. Údajový model - logging databáza



Obrázok 20. Údajový model - statistics databáza

5.2 Front-end

Webportál je používateľským prostredím, ktoré umožňuje interakciu cestujúcich s multimediálnym serverom. Implementovaný je ako Angular 2 aplikácia, ktorú budú môcť cestujúci prehliadať na inštalovaných obrazovkách, ale aj na akýchkoľvek vlastných zariadeniach. Vzhľadom na to, že implementácia celej funkcionality, ktorá sa očakáva od toho portálu by bola časovo náročná a nebolo ju možné realizovať v rámci tohto projektu, vybrali sme si časť, ktorú sa nám podarilo úspešne implementovať.

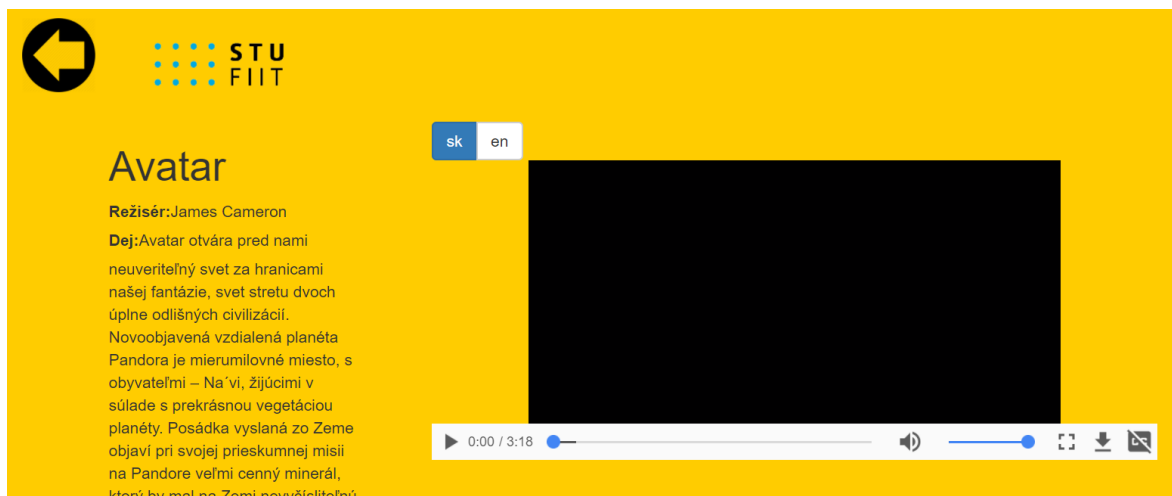
5.2.1 Navigácia

Navigácia v rámci web-portálu je realizovaná klikaním na jednotlivé ikony zobrazené na portáli. V prípade návratu môže používateľ zvoliť návrat o jeden krok späť na šípku, alebo kliknúť na logo spoločnosti, ktoré ho vráti na domovskú stránku.

Medzi jednotlivými url adresami, ktoré používateľ prechádza sú implementované tzv. resolvers, ktorých úlohou je prostredníctvom triedy *content.setvice.ts* zabezpečiť údaje http volaním na vystavenú REST API. Aplikácia tak čaká na odpoveď volania GET a až následne začne vykresľovať stránku, pre ktoré sú tieto údaje určené.

5.2.2 Prehliadanie filmov

Používateľ má možnosť prehliadať filmy, ktoré sú uložené v multimedialnej databáze. Formát prehrávaných filmov je mp4. Filmy sú delené do kategórií, podľa ktorých používateľ vyberá typ filmu o aký má záujem. Pri výbere konkrétneho filmu sa zobrazí používateľovi nasledujúce zobrazenie:



Obrázok 21 Detail filmu.

Používateľ vidí metadáta k filmu a na pravej strane je mediálny prehrávač. Podľa dostupnosti viacerých zvukových stôp je možné medzi týmito stopami vyberať jednoduchými tlačidlami nad prehrávačom. Prehrávač zahŕňa ovládaciu lištu, pomocou ktorej môže používateľ napríklad zväčšiť prehrávanie na celú obrazovku alebo v prípade dostupnosti zapnúť titulky. Prehrávanie filmu zahŕňa automatické prehratie reklamy v prípade, že bola prehratá prednastavená dĺžka z filmu. Po skončení reklamy prehrávač vráti prehrávanie filmu do pôvodného času prehrávania pred začiatkom prehratia reklamy.

5.2.3 Prehliadanie hudby

Prehliadanie hudby je ďalším modulom aplikácie. Hlavným objektom pri hudbe sú skladby, ktoré patria do albumov. Albumy sú priradené konkrétnym interpretom a napokon interpreti sú zaradené do hudobných žánrov.

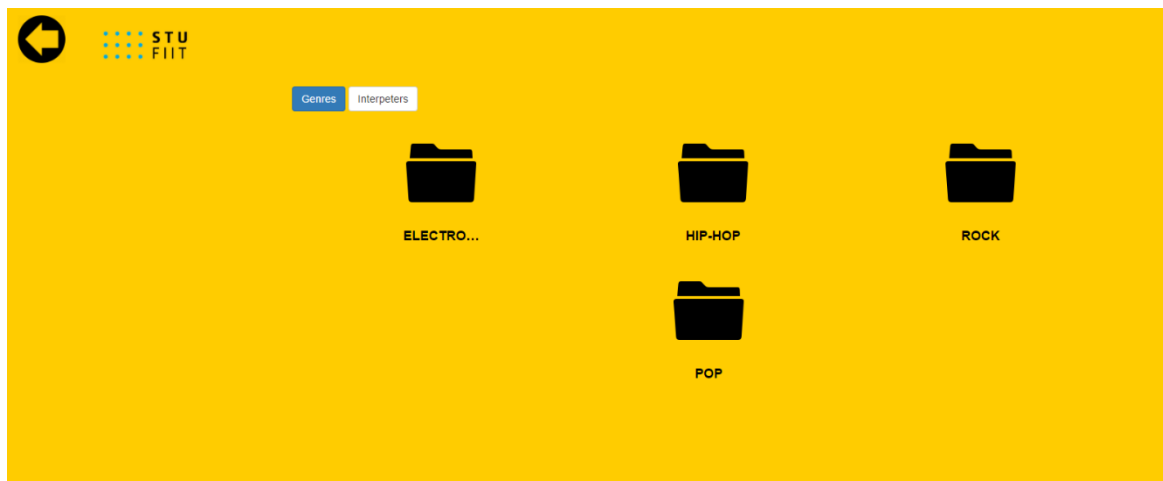
Pri prehrávaní skladieb prechádza používateľ touto hierarchiou zhora nadol v niekoľkých krokoch:

1. Výber hudobného žánru
2. Výber interpreta
3. Výber albumu

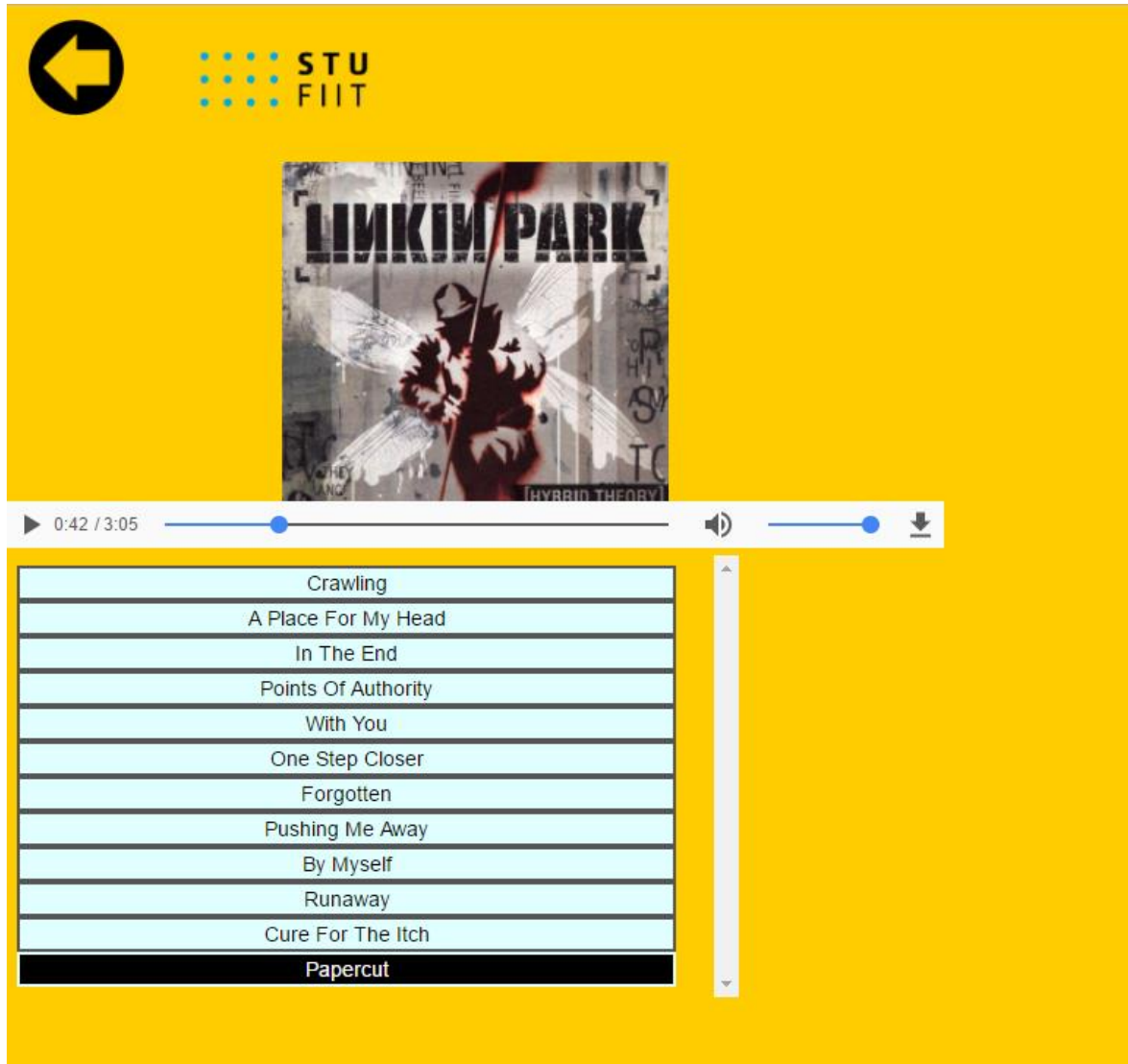
4. Výber pesničky
5. Prehrávanie vybranej skladby

Používateľ má možnosť preskočiť krok jeden, vtedy sa mu zobrazia hneď všetci interpreti.

Pri prehrávaní skladieb sa používateľovi zobrazí daná skladba a spolu s ňou aj playlist iných skladieb zložený na základe ostatných skladieb z albumu. Po skončení prehrávania sa automaticky spustí prehrávanie ďalšej skladby.



Obrázok 22 Výber hudobných žanrov



Obrázok 23 Hudobný prehrávač a playlist

5.2.4 Stránkovanie

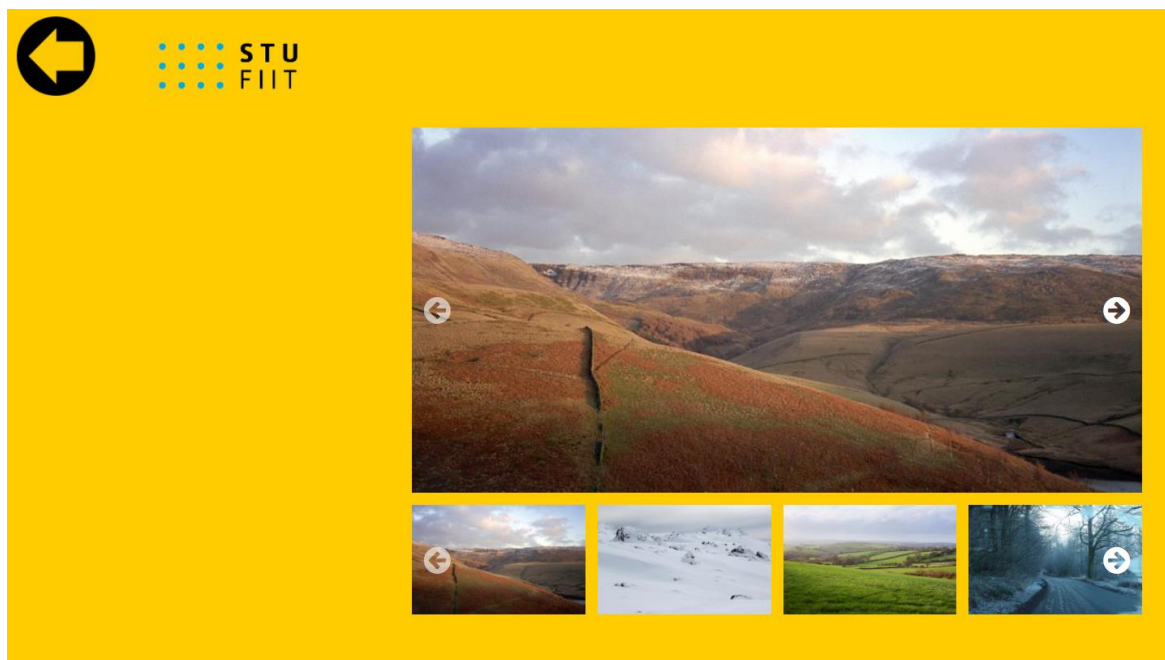
Z dôvodu zamerania aplikácie na mobilné zariadenia, kedy je potrebné brať do úvahy obmedzenú veľkosť obrazoviek, ako aj nepresný ovládací prvok používateľa – prst – sme implementovali vo všetkých obrazovkách, kde sa zobrazuje zoznam položiek stránkovanie.

Jedná sa o zoznam filmov, hudobných žánrov, interpretov, albumov a pesničiek.

Stránkovanie sa vykonáva na strane servera, čo znižuje prenášané dáta po sieti ako aj potrebné pamäťové zdroje v prehliadačoch.

5.2.5 Prehliadanie obrázkov

Obrázky sú uložené do albumov, ktoré webportál používateľovi zobrazuje. Po výbere konkrétneho albumu si používateľ prezerá fotografie tak ako je to zobrazené na nasledujúcom obrázku:



Obrázok 24 Prehliadanie fotografií

Pre zobrazenie obrázkov sa využíva importovaný modul *NgGaleryModule* do ktorej je vložené pole fotografií podľa definície tohto modulu a ten tieto fotografie zobrazí.

5.2.6 Výber jazyka

Používateľom umožňujeme výber jazyka, ktorý bude použitý v rámci portálu. Výber je umožnený na domovskej stránke. Na základe tohto výberu sa potom realizujú dopyty na API webportálu, ktorá podľa vyžiadaného jazyka vracia údaje v danom jazyku. Výber jazyka sa následne ukladá do *translate.tervice.ts* z ktorého môžeme z akéhokoľvek miesta zistiť aktuálny jazyk portálu. Defaultný jazyk portálu ja pritom načítava pri spustení aplikácie so súboru *enviroment.ts*.

6 Testovanie

6.1 Jednotkové testy

Jednotkové testy slúžia na automatizované testovanie časti funkcionality pri kompilácii alebo nasadení software-u. V tomto projekte boli jednotkové testy aplikované iba na aplikáciu webového portálu a na aplikáciu vytvárajúcu vlákna pre účely vzdialenej správy.

Všeobecná štruktúra jednotkového testu aplikovaná v Java aplikácií:

1. V prípade Spring frameworku, je potrebné mať inicializovanú Spring architektúru popri tom, ako budú bežať jednotkové testy (napr. aby bolo možné používať manažerov databázových spojení). Prepojenie jednotkových testov so Spring framework-om je realizované pomocou nasledovných anotácií pri deklarácii triedy (AppConfig – hlavná konfiguračná trieda v projekte webového portálu):

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(classes = {AppConfig.class})
@WebAppConfiguration
public class StatisticsControllerTest {
    ...
}
```

2. Inicializácia konštantných hodnôt a deklarácia niektorých objektov, ktoré sa prenášajú medzi testami, testujú ja, alebo sa inicializujú pri nastavení a následne sa používajú pri viacerých testoch. Najčastejšie sa tu nachádzajú:

- Umiestnenie certifikačných alebo iných súborov, hesla a iné konštatné hodnoty.
- Neinicializovaná premenná testovanej triedy, korej metódy sa budú v triede testovať (napr. vlákna alebo triedy manažerov).
- Neinicializovaná premenná testovanej triedy EntityManagerFactory, ktorá slúži na spravovanie databázových transakcií.
- Objekt WireMockRule slúžiaci na simulovanie HTTP servera (označenie pomocou anotácie @Rule). Príklad konfigurácie so zadaným HTTPS portom, umiestnením certifikačného úložiska a so zadaným prístupovým heslom do certifikačného úložiska:

```
private static final String PATH_KEYSTORE =
    "/var/certs/keystore.jks";
private static final String KEYSTORE_PASS = "password";
@Rule
public WireMockRule wireMockRule = new
WireMockRule(wireMockConfig()
    .httpsPort(RemoteMachineId.DEFAULT_PORT)
    .keystorePath(PATH_KEYSTORE)
    .keystorePassword(KEYSTORE_PASS));
```

- Časť s automatickým naviazaním perzistentnej jednotky na Spring, príklad:

```
private static EntityManagerFactory emf;
@PersistenceUnit(unitName = "statistics")
public void setEntityManagerFactory(EntityManagerFactory emf) {
    StatisticsControllerTest.emf = emf;
}
```

- Ďalšie objekty, ktoré je vhodné inicializovať iba raz a budú sa používať vo viacerých jednotkových testoch.
3. Nastavenia testov – Tu sa nachádzajú úvodné inicializácie objektov, ktoré sa budú testovať a ďalších podporných objektov, ktoré sú pri testoch potrebné (napr. `HttpSenderReceiver` na prijímanie / odosielania HTTP správ alebo vlákno na periodické odosielanie aplikačných logov na cloud). Metóda s nastaveniami sa vždy spúšťa pred samotnými testami. Má označenie `setup()` s anotáciou `@Before`.
 4. Hlavné telá testov – Ide o metódy, ktoré majú v svojom názve reťazec “test” a sú označené anotáciou `@Test`. Tieto metódy testujú funkčnosť jednej zvolenej metódy v testovanej triede. Testovanie sa realizuje pomocou viacerých knižníc alebo metód:
 - a. Testovanie správneho vykonania metódy – Testujú sa očakávané alebo neočakávané výnimky, ktoré sa vyhodia počas vykonávania testovanej metódy. V prípade očakávania vyhodenej výnimky je potrebné upraviť telo anotácie `@Test` nasledovne:
`@Test(expected = JAXBException.class)`
 - b. Testovanie porovnaním očakávaných hodnôt s vypočítanými alebo vrátenými hodnotami (v tomto poradí), príklad:

```
Assert.assertEquals(LocalMachineId.DEFAULT_SERVER_ID,
config.getLocalMachineId().getServerId());
```
 - c. Testovanie pomocou `WireMock` knižnice – testovanie prijatých HTTP správ – format hlavičky a telo správy, príklad:

```
verify(postRequestedFor(urlEqualTo(site))
.withHeader("Content-Type", equalTo("application/json")));
BasicCredentials credentials = new BasicCredentials
(LocalMachineId.DEFAULT_SERVER_ID,
LocalMachineId.DEFAULT_SECRET);
verify(postRequestedFor(urlEqualTo(site))
.withBasicAuth(credentials));
```

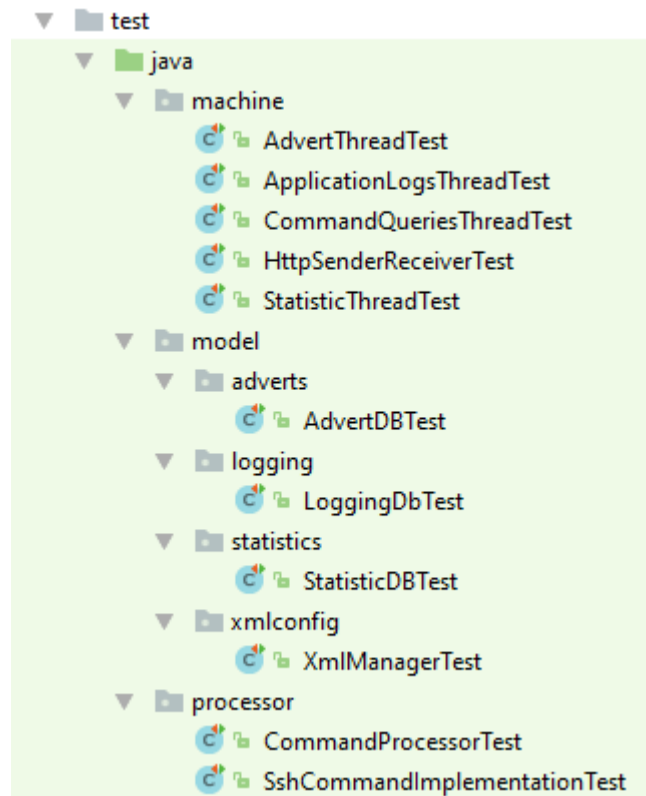
Testy väčšinou sú zložené z viacerých procedúr, napr. vloženie záznamov do databázy, spustenie vlákna, vyberanie hodnôt z databázy a testovanie očakávaných hodnôt. Pri testovaní API je dôležitou súčasťou tela testu aj nastavenie webového servera (`WireMock`), ktorý automaticky odpovedá na HTTP Post / Get správy (simulácia cloudu), príklad nastavenia:

```
String site = RemoteMachineId.DEFAULT_APPLICATION_LOGS;
stubFor(post(urlEqualTo(site))
.withReturn(aResponse()
.withHeader("Content-Type", "text/plain")
.withBody(HttpSenderReceiver.ACKNOWLEDGE_STRING)
.withStatus(200)));
```

5. Ukončenie testov – Po ukončení testovacích procedúr je potrebné zatvoriť simulačný webový server ako aj otvorené HTTP spojenie, prípadne vyčistiť umelo vytvorené záznamy v databázach alebo vymazať dočasné súbory, ktoré sa používali pri testoch (konfiguračné súbory). Táto metóda má vždy označenie `clean()` a je označená anotáciou `@After`. Príklad:

```
@After
public void clean() throws IOException {
removeAllApplicationLogs();
wireMockRule.stop();
httpsSenderReceiver.closeHttpClient();
}
```

Všetky jednotkové testy sú usporiadané v adresári test. Názvy balíkov sú rovnaké ako názvy balíkov tried, ktoré sa v testovacej treiede nachádzajú. Názvy testovacích tried musia obsahovať názov nestovanej metódy s prefixom alebo postfixom „Test“. Príklad usporiadania jednotkových testov pri aplikácii na vzdialenú správu znázorňuje Obrázok 25.



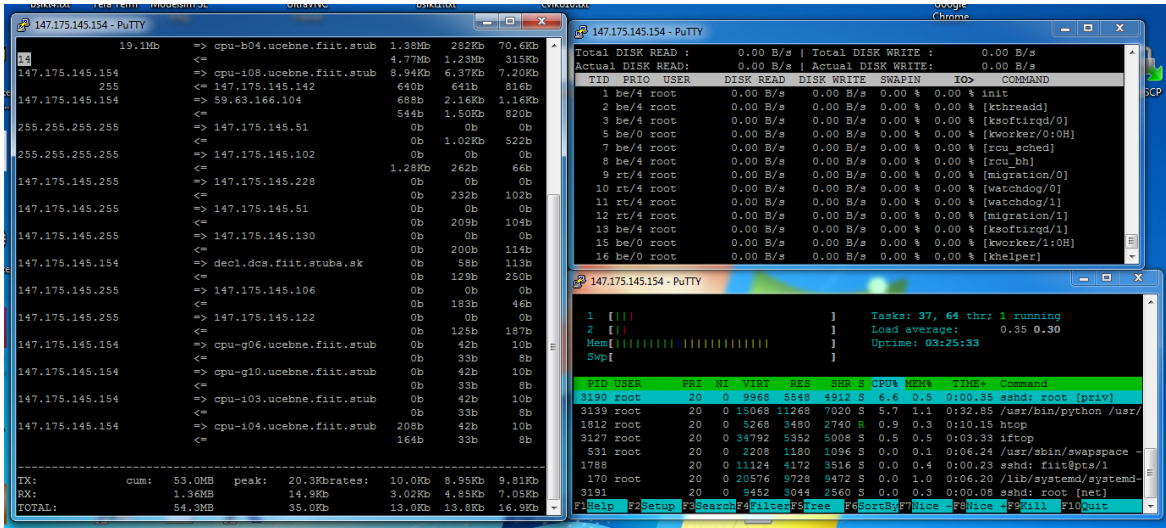
Obrázok 25. Štruktúra testov (vzdialená správa)

6.2 Testovanie nasadených aplikácií

Cieľom bolo otestovať riešenie na stene dostupnej v Molpir laboratóriu, avšak inštalované zariadenia majú operačný systém Android verzie 2, ktorý nepodporuje Angular framework. Z tohto dôvodu sme testovania vykonali na viacerých fyzických počítačoch. Výsledky týchto meraní je možné vidieť na nasledujúcich obrázkoch, kde je zobrazené vyťaženie centrálného servera.

6.2.1 Server bez zaťaženia

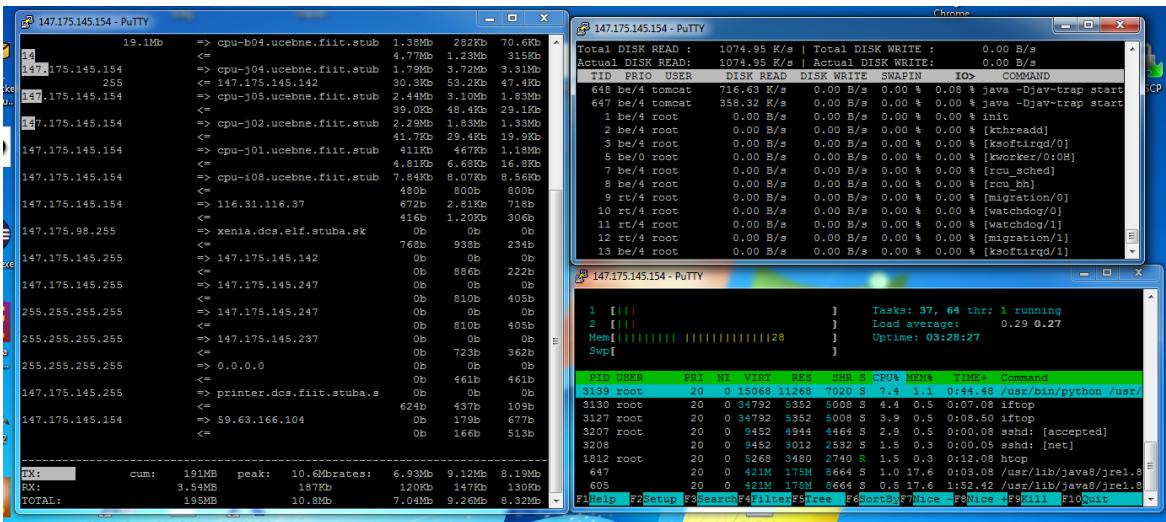
Na obrázku nižšie je možné vidieť server bez zaťaženia, tzn. že naň neboli pripojené žiadne klientské zariadenia.



Obrázok 26 – Server bez zataženia

6.2.2 Server s pripojenými 5 zariadeniami

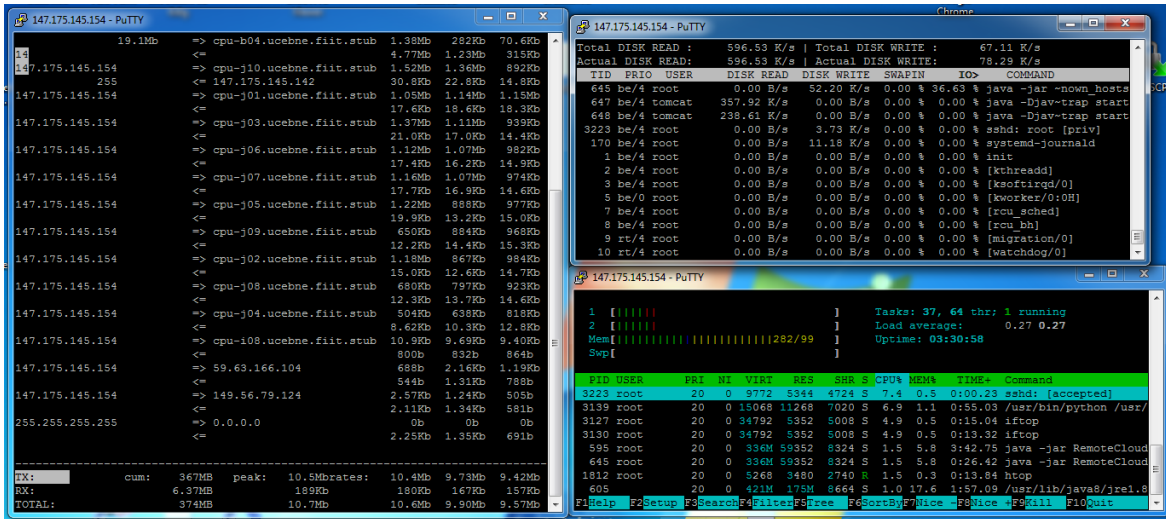
V tomto teste bolo pripojených 5 zariadení. Na každom zo zariadení bol spustený iný film. Na obrázku nižšie je možné vidieť, že hlavným vytáženým zdrojom je sieť a ostatné výpočtové zdroje ostávajú vytážené iba minimálne.



Obrázok 27 – Server s pripojeným 5 zariadeniami

6.2.3 Server s pripojenými 10 zariadeniami

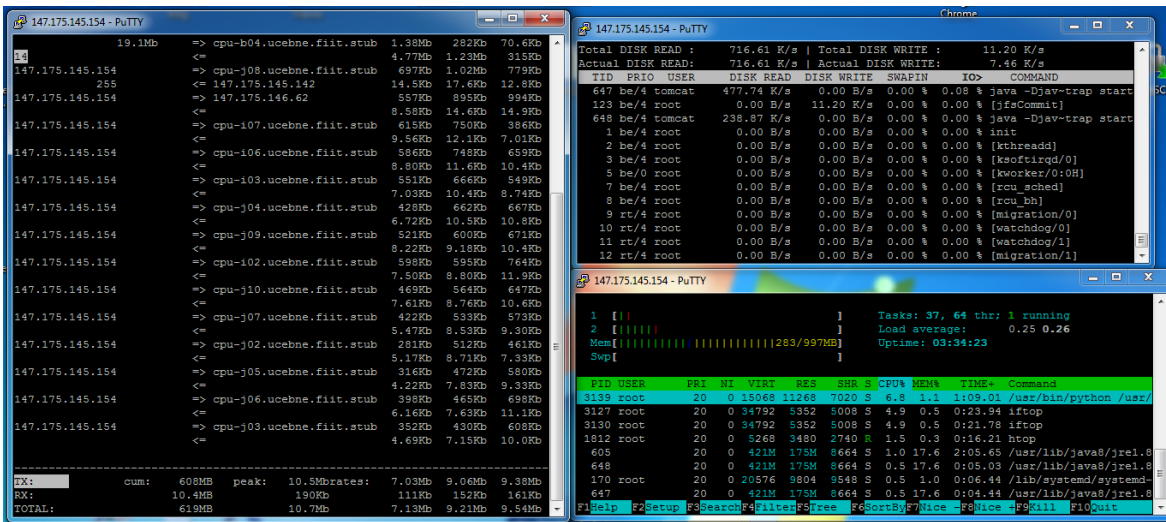
V tomto teste bolo pripojených 10 zariadení. Na každom zo zariadení bol spustený film. Na obrázku nižšie je možné vidieť, že hlavným vytáženým zdrojom je opäť sieť a ostatné výpočtové zdroje ostávajú vytážené iba minimálne. Pri tomto teste sme zistili, že sieť je pravdepodobne obmedzená na 100Mbit/s, pretože filmy sa začali načítavať dlhšie, ale vytáženie servera nestúpalo.



Obrázok 28 – Server s pripojenými 10 zariadeniami

6.2.4 Server s pripojenými 20 zariadeniami

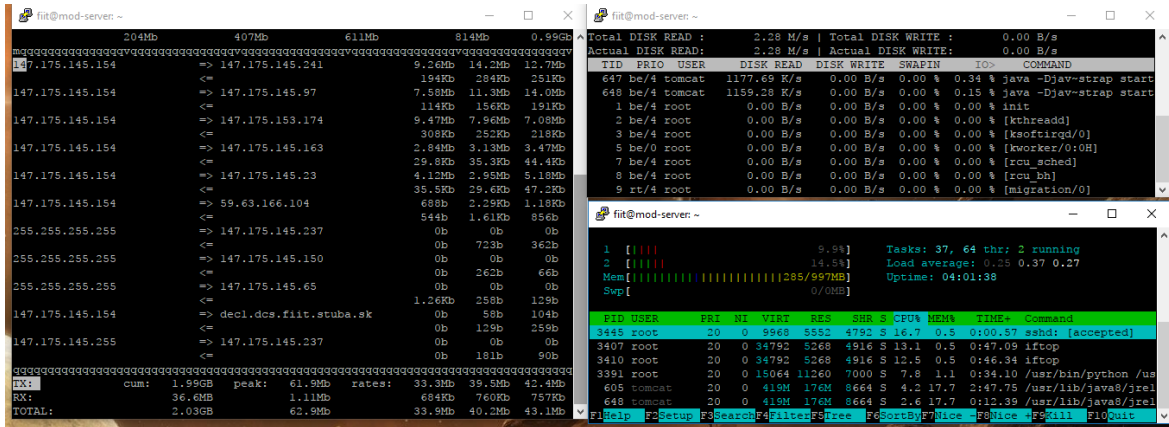
Keďže v predchádzajúcom teste sme zistili obmedzenie siete, v tomto teste sme sítě pripojili 20 zariadení, ale z týchto iba 10 prehrávalo film a ďalších 10 hudbu. Zaťaženie siete ostalo rovnaké (kvôli fyzickému obmedzeniu) a vyťaženie servera bolo stále minimálne.



Obrázok 29 - Server s pripojenými 20 zariadeniami

6.2.5 Server s prehrávaním 30 filmov súčasne

Posledný test sme vykonávali v laboratóriu s priamym pripojením na server cez prepínač s rýchlosťou linky 1 Gbit/s. Sítě bolo pripojených iba 5 fyzických zariadení, no na každom z nich bolo spustených 6 záložiek, kde sa prehrával film. Výsledný počet prehrávaných filmov bol teda 30. Na obrázku vidno, že v tomto teste výrazne stúplo zaťaženie siete, čo potvrdilo aj naše predpoklady o obmedzenej linke v predchádzajúcich testoch.



Obrázok 30 – Server s prehrávaním 30 filmov súčasne

6.2.6 Zhodnotenie testov

Hlavným obmedzením počas testovania bola rýchlosť linky. Z testovani však vyplynulo, že server aj pri spustení viacerých prehrávaní súčasne nejaví zvýšené vyťaženia procesora. Z toho vplyva, že pri použití dostatočne rýchlej linky a optimálnej kvality videa, bude server zvládať aj viacero naraz pripojených zariadení.

Bibliografické odkazy

- [1] F. Stroud, "Top 10 Linux Server Distributions of 2016," ServerWatch, 8th May 2016. [Online]. Available: <http://www.serverwatch.com/columns/slideshows/top-10-linux-server-distributions-of-2015.html>. [Accessed 18th November 2016].
- [2] S. Bhartiya, „Best Lightweight Linux Distros,“ LINUX.COM, 17th August 2015. [Online]. Available: <https://www.linux.com/news/best-lightweight-linux-distros>. [Cit. 18th November 2016].
- [3] B. Horne, „A Comparison of Popular Linux Package Managers,“ The Hornery, 19th February 2016. [Online]. Available: <http://fusion809.github.io/comparison-of-package-managers/>. [Cit. 18th November 2016].
- [4] „Comparison of major Linux package management systems,“ LinuxConfig, [Online]. Available: <https://linuxconfig.org/comparison-of-major-linux-package-management-systems>. [Cit. 18th November 2016].
- [5] „Package Management Cheatsheet,“ DistroWatch.com, 2016. [Online]. Available: <https://distrowatch.com/dwres.php?resource=package-management>. [Cit. 18th November 2016].
- [6] „Intel® Embedded Graphics Drivers,“ Intel, [Online]. Available: <http://www.intel.com/content/www/us/en/embedded/software/iegd/intel-embedded-graphics-drivers-overview.html>. [Cit. 18th November 2016].
- [7] „ArchLinux Forum. [SOLVED] ALSA can't detect Intel ICH7 soundcard,“ ArchLinux, 2nd January 2014. [Online]. Available: <https://bbs.archlinux.org/viewtopic.php?id=175110>. [Cit. 18th November 2016].
- [8] „CONFIG_I2C_I801: Intel 82801 (ICH),“ Linux Kernel Driver DataBase, 2016. [Online]. Available: http://cateee.net/lkddb/web-lkddb/I2C_I801.html. [Cit. 18th November 2016].
- [9] „Downloads for Intel® 82574 Gigabit Ethernet Controller,“ Intel, [Online]. Available: <https://downloadcenter.intel.com/product/32210/Intel-82574-Gigabit-Ethernet-Controller>. [Cit. 18th November 2016].
- [10] Wikipedia contributors, „Comparison of Linux distributions,“ Wikipedia, The Free Encyclopedia, 12th November 2016. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Comparison_of_Linux_distributions&oldid=749149982. [Cit. 18th November 2016].
- [11] „Compare Packages Between Distributions,“ DistroWatch.com, [Online]. Available: <https://distrowatch.com/dwres.php?resource=compare-packages>. [Cit. 18th November 2016].
- [12] S. Hussain, „How To Configure a Linux Service to Start Automatically After a Crash or Reboot,“ DigitalOcean, 19th August 2015. [Online]. Available: <https://www.digitalocean.com/community/tutorials/how-to-configure-a-linux-service-to-start-automatically-after-a-crash-or-reboot-part-1-practical-examples>. [Cit. 18th November 2016].
- [13] „Homepage,“ Automotive Grade Linux, 2016. [Online]. Available: <https://www.automotivelinux.org/>.

[Cit. 18th November 2016].

- [14] M. Gelbmann, „Ubuntu became the most popular Linux distribution for web servers,“ W3Techs, 2nd May 2016. [Online]. Available: https://w3techs.com/blog/entry/ubuntu_became_the_most_popular_linux_distribution_for_web_servers. [Cit. 18th November 2016].
- [15] „Results of the 2015 /r/Linux Distribution Survey,“ Fighting the System, 24th August 2015. [Online]. Available: <https://brashear.me/blog/2015/08/24/results-of-the-2015-slash-r-slash-linux-distribution-survey/>. [Cit. November 18th 2016].
- [16] „2013 /r/Linux Distro Survey Results,“ 3rd March 2013. [Online]. Available: <https://constantmayhem.com/ty-stuff/linuxsurvey/2013.html#primaryServerResults>. [Cit. November 18th 2016].
- [17] “Chapter 3. FreeBSD Basics. 3.5. Directory Structure,“ FreeBSD, [Online]. Available: <https://www.freebsd.org/doc/handbook/dirstructure.html>. [Accessed 11th November 2016].
- [18] „FileSystem,“ Debian, [Online]. Available: <https://wiki.debian.org/FileSystem>. [Cit. 11th November 2016].
- [19] „Comparison of file systems,“ Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Comparison_of_file_systems. [Cit. 11th November 2016].
- [20] M. Larabel, “Linux 4.0 SSD EXT4 / Btrfs / XFS / F2FS Benchmarks,“ Phoronix, 12th April 2015. [Online]. Available: <http://www.phoronix.com/scan.php?page=article&item=linux-40-ssd&num=3>. [Accessed 19th November 2016].
- [21] M. Larabel, “9-Way File-System Comparison With A SSD On The Linux 3.17 Kernel,“ Phoronix, 7th September 2014. [Online]. Available: http://www.phoronix.com/scan.php?page=article&item=9way_linux317_fs&num=1. [Accessed 19th November 2016].
- [22] P. Schmid, “Does Your SSD's File System Affect Performance?,“ tom's HARDWARE, 13th April 2012. [Online]. Available: <http://www.tomshardware.com/reviews/ssd-file-system-ntfs,3166-8.html>. [Accessed 19th November 2016].
- [23] M. Larabel, “Linux 4.7 - Btrfs vs. EXT4 vs. F2FS vs. XFS vs. NTFS Benchmarks,“ Phoronix, 6th August 2016. [Online]. Available: https://www.phoronix.com/scan.php?page=news_item&px=Linux-4.7-FS-5-Way. [Accessed 19th November 2016].
- [24] „Help me choose the best filesystem for my PC [closed],“ Ubuntu, 27th October 2015. [Online]. Available: <http://askubuntu.com/questions/690360/help-me-choose-the-best-filesystem-for-my-pc>. [Cit. 19th November 2016].
- [25] „8.6. Compiling a New Kernel,“ Debian, [Online]. Available: <https://www.debian.org/releases/jessie/i386/ch08s06.html.en>. [Cit. 22nd November 2016].
- [26] „Booting fallback systems,“ GNU, 2006. [Online]. Available: <https://www.gnu.org/software/grub/manual/legacy/Booting-fallback-systems.html>. [Cit. 22nd November 2016].

7 Prílohy

7.1 Elektronické médium

Obsah priloženého elektronického média znázorňuje Tabuľka 39.

Tabuľka 39. Obsah elektronického média

Súbor / adresár	Opis
/deployed mod clonezilla	Adresár obsahujúci zálohu nasadeného systému na MOD (vytvorené voľne dostupných nástrojom Clonezilla).
/javadoc remote management	JavaDoc dokumentácia vygenerovaná z projektu, ktorý rieši vzdialenú správu MOD servera z cloudu.
/javadoc webportal	Adresár obsahuje JavaDoc dokumentáciu webového portálu (administratívna časť web portálu obsahujúca REST API a rozhrania pre komunikáciu s databázovým systémom a konfiguráciami).
/typescriptdoc frontend	Dokumentácia webového portálu – prezentačná vrstva (Angular 2).
/virtual machine mod	Obraz virtuálneho servera slúžiace na testovanie a kompiláciu systémových služieb a nasadenie web portálu.
/static team web	Statická verzia web stránky tímového projektu.
/dokumentácia api.docx /dokumentácia api.pdf	Dokumenácia REST API webového portálu – štruktúra HTTP GET / POST volaní.
/dokumentacia prirucka.docx /dokumentacia prirucka.pdf	Používateľská (inštalačná) príručka – konfigurácia servera, nastavenie vývojového prostredia, nastavenie štartovacích hodnôt portálu a nastavenia hodnôt pre štatistiku.
/dokumentacia projektu.docx /dokumentacia projektu.pdf	Hlavná dokumentácia projektu – inžinierske dielo (finálna verzia). Obsahuje analýzu, návrh a implementáciu projektu.
/dokumentacia riadenia projektu.docx /dokumentacia riadenia projektu.pdf	Dokumentácia riadiacich aspektov projektu – metodiky, výstupy šprintov.