

LINK Academy

Java

Aplicație de gestionare a fluxurilor video preluate de la camere IP

Profesorul cursului:
Bogdan Posa

Cursantul:
Matișevschii Constantin
Data predării
26.10.2020

Cuprinsul lucrării

Rezumatul lucrării	3
Cuvintele cheie	3
• WebRTC	3
• RTSP.....	3
• RTP	3
• Janus	3
• FFMPEG.....	3
• OpenCV.....	3
• Spring Framework	3
Descrierea aplicației	4
Structura aplicației	4
Descrierea componentelor	5
• Spring boot.....	5
• Janus video server	5
• Canal video.....	5
• Containerul de canale.....	6
Librăriile și aplicațiile utilizate.....	6
• BACK-END	6
• FRONT-END	6
Concluzia	6
Conținutul utilizat de pe Internet	7

Rezumatul lucrării

Aplicația creată de mine are ca scop centralizarea camerelor video IP(de orice model) într-o singură interfață, precum și oferirea unei interfețe unice pentru vizualizarea fluxurilor video furnizate de ele și ,optional, posibilitatea de a activa detectori (bazați pe analiza video) care nu sunt prezenti în camerele video IP simple. Aplicația a fost creată pentru a fi instalată pe un Raspberry pi model 4B (un mini calculator cu preț redus care ar avea toate resursele hardware necesare pentru a rula această aplicație). Astfel, utilizatorul final va obține un mini calculator ieftin, cu o aplicație preinstalată pe el, ce îi va oferi posibilitatea de a vizualiza live imagini video de la camerele IP, montate la el acasă, folosind telefonul mobil sau laptop-ul.

Cuvintele cheie

- **WebRTC** - (eng. Web Real-Time Communication) este un proiect open-source gratuit care oferă browserelor web și aplicațiilor mobile comunicații în timp real (RTC) prin interfețe simple de programare a aplicațiilor (API).
- **RTSP** - (eng. Real Time Streaming Protocol) este un protocol de control al rețelei, destinat pentru utilizarea în divertisment (eng. *entertainment*) și sisteme de comunicații de control al fluxului serverelor media (eng. *streaming media servers*).
- **RTP** - Protocolul de transport în timp real (sau RTP de la engl. *Real-time Transport Protocol*) este un protocol prin intermediul căruia se pot transmite informații de tip media (sunete, imagini) prin rețea de telecomunicații.
- **Janus** - este un server WebRTC dezvoltat de Meetecho conceput pentru a fi unul cu scop general. Ca atare, nu oferă nici o funcționalitate în sine, în afară de implementarea mijloacelor de configurare a unei comunicații media WebRTC cu un browser.
- **FFmpeg** - FFmpeg este un proiect software gratuit și open-source care constă dintr-o suită mare de biblioteci și programe pentru manipularea fișierelor și fluxurilor video, audio și multimedia. În centrul său se află programul FFmpeg în sine, conceput pentru procesarea pe linie de comandă a fișierelor video și audio.
- **OpenCV** - OpenCV (eng. Open Source Computer Vision Library) este o bibliotecă de funcții de programare destinate în principal viziunii computerizate în timp real. Biblioteca este multiplatformă și gratuită pentru utilizare sub licența BSD open source.
- **Spring Framework** - Spring Framework este o colecție de framework-uri mai mici, majoritatea putând să opereze independent de celelalte, însă oferă mai multă funcționalitate atunci când sunt folosite împreună. Abstract classes, factories și beans sunt construite în aşa fel încât programatorul să se poată concentra pe business logic.

Descrierea aplicației

Proiectul dat are ca scop oferirea unei interfețe în care utilizatorul final are posibilitatea să adauge camere video IP de diferite brânduri. El are nevoie doar de un laptop sau telefon mobil pentru a se conecta la aplicație ca să vizualizeze imaginile video live de pe camerele video adăugate în aplicație.

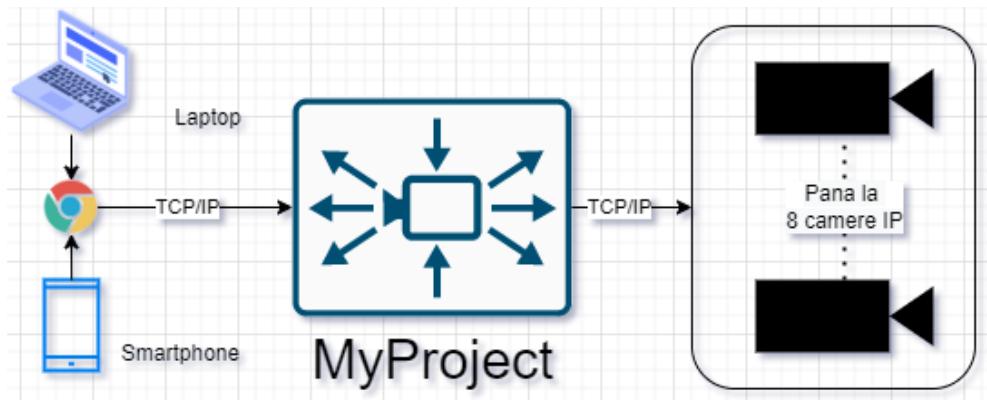


Figura 1. Schema bloc generală

Structura aplicației

Aplicația este formată din următoarele componente:

- **Spring boot**
- **Janus video server**
- **Containerul de canale** (până la 8 canale video)

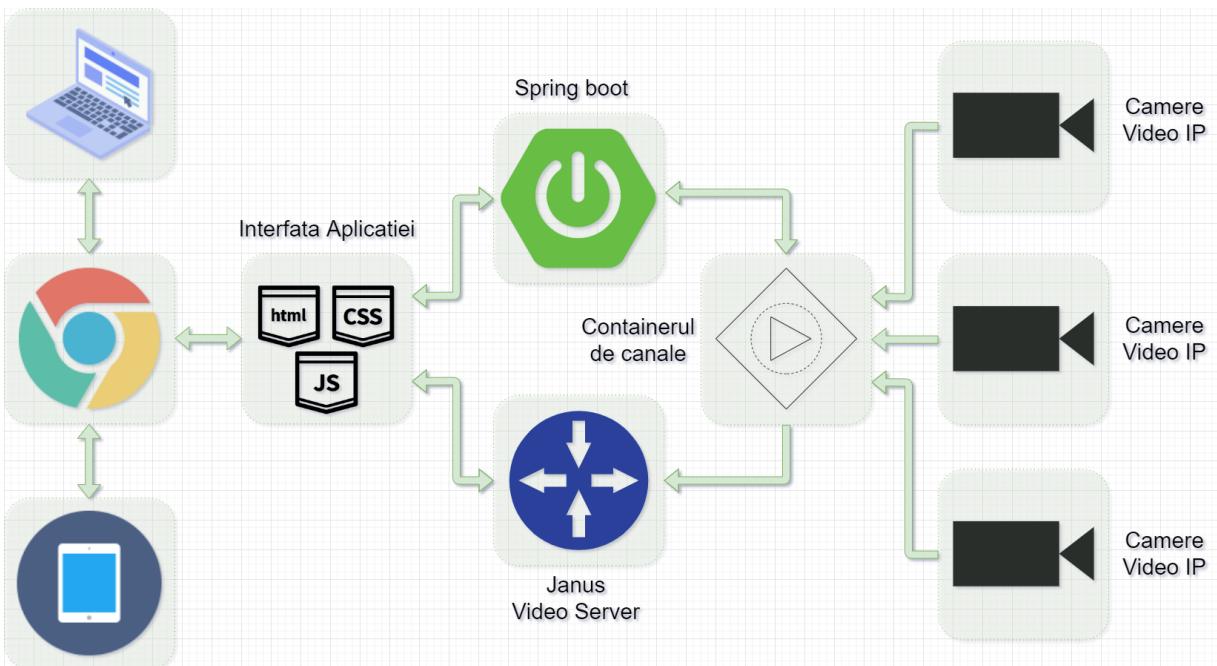


Figura 2. Componentele aplicației

Descrierea componentelor

- **Spring boot** - pentru a avea acces de la distanță la aplicație este utilizat framework-ul Spring boot, care permite utilizatorului să se conecteze la restul componentelor aplicației prin intermediul paginilor html furnizate în browser.
- **Janus video server** – pentru afișarea fluxului video live în browser, este utilizată o aplicație externă Janus, care joacă rol de puncte între fluxul video al camerei IP și browser. Fiecare funcționalitate din Janus poartă numele de “plugin”. Pentru a utiliza componenta de video streaming a serverului Janus, am instalat adițional plugin-ul “Streaming plugin”, descris în figura 3.



Figura 3. Janus (Streaming plugin)

- **Canal video** – un canal video este componenta care se conectează la o camera IP, preia fluxul video principal și secundar și le transmite către serverul Janus. Pe lângă partea de video streaming, în canalul video mai există și două memorii tampon: una din memorii tampon este asociată fluxului video principal, cealaltă fluxului video secundar, ambele fiind destinate stocării de poze generate (la un interval fix de timp) din fluxurile video. Memoriile tampon sunt create pentru a fi folosite de către detectorul de mișcare, care, la rândul său, citește imagine și identifică mișcarea.

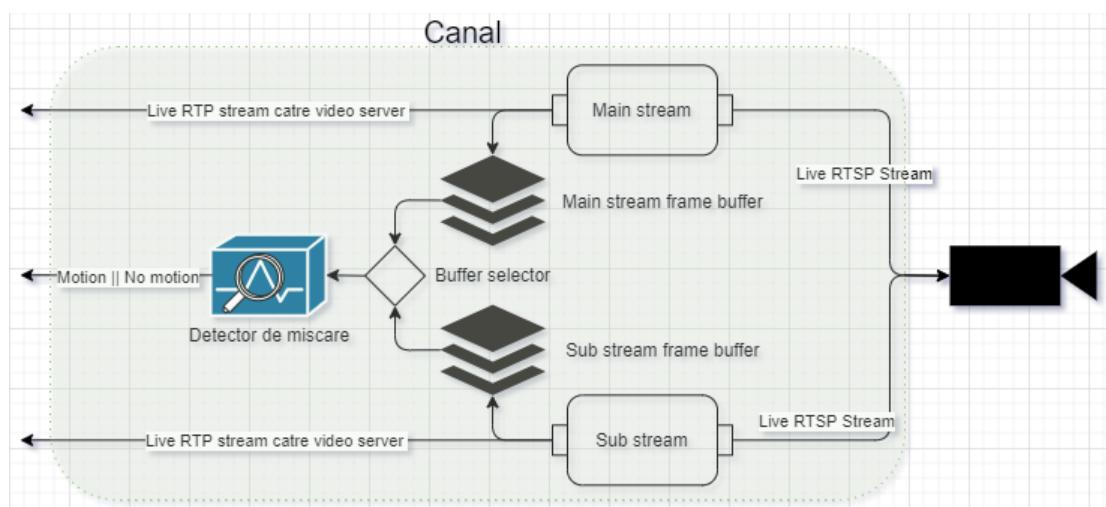


Figura 4. Schema bloc a unui canal

- **Containerul de canale** – reprezintă componenta ce găzduiește toate canalele video ale aplicației. Prin intermediul ei, se fac toate modificările de parametri pentru fiecare canal în parte. De fiecare dată când se modifică un parametru de canal, containerul de canale salvează parametrii acestora într-un fișier extern.

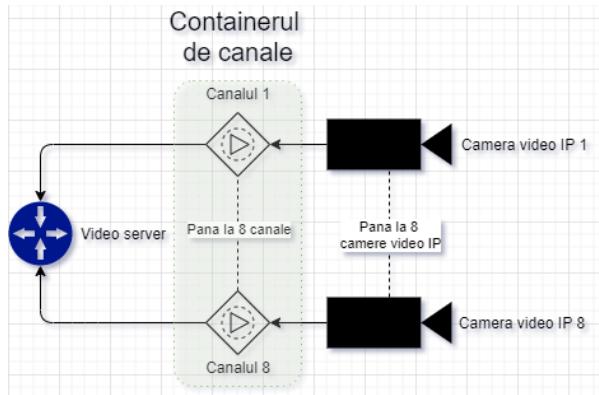


Figura 5. Container de canale

Librăriile și aplicațiile utilizate

- **BACK-END**
 - **Librării**
 - org.springframework
 - org.opencv
 - org.slf4j
 - com.github.kokorin.jaffree
 - java.util.concurrent
 - java.io
 - **Aplicații**
 - FFmpeg
 - Janus WebRTC Server & Janus javascript client
- **FRONT-END**
 - **Librării**
 - Bootstrap
 - Janus
 - Jquery
 - Web_socket
 - Adapter

Concluzia

În procesul elaborării proiectului dat, am întâmpinat mai multe dificultăți, care m-au determinat să studiez mai asiduu. Fiecare framework nou studiat îmi demonstra că cheia succesului constă în învățarea utilizării cât mai eficace a librăriilor deja scrise, fără a fi nevoie să le inventez de la zero.

În această aplicație am folosit mai multe framework-uri, librării, aplicații, care rulează sub controlul lui java sau comunică cu codul scris în limbajul de programare java. Astfel, am înțeles că scriind cod în java, am posibilitatea de a crea orice tip de aplicație. Concluzionând, pot spune, fără îndoială, că m-am îndrăgostit de acest limbaj.

Conținutul utilizat de pe Internet

- <https://ffmpeg.org/download.html>
- <https://ffmpeg.org/documentation.html>
- <https://opencv.org/>
- <https://janus.conf.meetecho.com/>
- <https://janus.conf.meetecho.com/docs/streaming.html>
- <https://github.com/meetecho/janus-gateway>
- <https://getbootstrap.com/docs/4.5/getting-started/introduction/>
- <https://api.jquery.com/>
- <https://start.spring.io/>