

**PROJEKAT SISTEMA ZA POMOĆ PRI ONLINE KUPOVINI  
PROPOSAL OF THE SYSTEM FOR HELP IN ONLINE SHOPS**

Igor Dujlović  
Elektrotehnički fakultet u Banja Luci, RS, BiH

**REZIME:** U radu je analizirana mogućnost pružanja dodatnih informacija o proizvodima tokom kupovine na internetu. Dat je kratak opis postojećih sličnih rješenja i analizirane su aktuelne web tehnologije koje se mogu koristiti za razvoj sistema. Osim toga, u radu je prikazan prijedlog sistema koji se može jednostavno integrisati u postojeće sisteme prodaje i koristiti na različitim vrstama uređaja bez instaliranja dodatnog softvera na klijentskoj strani. Implementirani sistem kupcima pruža mogućnost da pozovu trgovca i da na taj način dobiju dodatne informacije o proizvodima ili uslugama koje su, ponuđene na sajtu. Za implementaciju poziva korišten je WebRTC koji omogućava audio, video i tekstualnu komunikaciju i slanje fajlova. Osim toga, sistem omogućava korisnicima da koriste zajednički prostor za grafičko predstavljanje detalja proizvoda.

**KLJUČNE REČI:** WebRTC, komunikacija u realnom vremenu, online kupovina, HTML5, Canvas

**ABSTRACT:** The paper analyzes the possibility of providing additional product information while shopping on the Internet. The paper also gives a brief description of the existing similar solutions and analyze the current web technologies that can be used to develop the system. In addition, the paper presents a proposal for a system that can be easily integrated into existing sale systems and used on different types of devices without installing additional software on the client side. The implemented system provides customers the ability to call the dealer and get more information about products or services offered on the site. WebRTC is used for the implementation of the calls. WebRTC enables audio, video and text communication and sending files. In addition, the system allows users to use the common area for the graphical presentation of the details of the product.

**KEY WORDS:** WebRTC, real-time communication, online shopping, HTML5, Canvas

**UVOD**

Razvojem web aplikacija za prodaju pojavljuje se potreba da se određeni proizvodi približe kupcima, tačnije, da kupci jednostavnije pronađu željene proizvode i da se uvjere da određeni proizvod zadovoljava njihove želje i potrebe. Najčešći oblik pružanja tih informacija je u tekstualnom formatu, uz dodatak određenog broja slika proizvoda uz mogućnost komentiranja i ocjenjivanja. Većina internet prodavnica, kao i web sajtova koji posjeduju mogućnost prodaje proizvoda imaju opcije za kontakt. Kontakt se najčešće obavlja putem elektronske pošte, chat-a ili poziva prema operateru call centra ili trgovcu. Opcije za kontakt mogu biti od presudnog značaja prilikom izbora proizvoda ukoliko kupac ima nedoumice oko izbora proizvoda ili ukoliko informacije ponuđene na stranici nisu dovoljne za donošenje odluke oko kupovine proizvoda. Osim toga, pomoć pri kupovini je naročito značajna ukoliko se radi o prodaji određenih usluga, gdje se kupac i trgovac mogu direktno dogovoriti oko sadržaja i detalja usluge. Takođe, ukoliko web sajt nema opcije za prodaju, a ima pregled proizvoda, tada bi kupac mogao da izvrši narudžbu direktno preko web sajta tokom razgovora sa trgovcem čime bi smanjio troškove poziva preko tradicionalnih telekomunikacionih sistema, što može biti značajno ukoliko se radi o međunarodnom pozivu.

Kontakt u tekstualnoj formi, pomoću email-a ili chat-a, često je spor i nedovoljno efikasan, pa bi za određene situacije direktan poziv bilo mnogo efikasnije rješenje. Osim toga, ukoliko se kupcu ponude i neki drugi načini za prezentovanje proizvoda ili dogovora o pojedinostima, tada kupac može brže donijeti odluku o kupovini i imati isti osjećaj kao da proizvod kupuje u prodavnici, a ne preko interneta. Poželjno je da kupac za dobijanje pomoći ne ulaže dodatna sredstva, čak i u slučaju poziva operateru ili trgovcu, a da pomoć može dobiti sa bilo kojeg uređaja koji koristi pri kupovini (računar, tablet, smar-

tphone...). Kupac bi se na taj način mogao dodatno uvjeriti o detaljima proizvoda, načinu kupovine, uslovima plaćanja i dostave, kao i svim ostalim pojedinostima bez obzira na lokaciju na kojoj se trgovac nalazi. Posebno bi bila zanimljiva opcija u kojoj bi kupac i trgovac mogli interaktivno učestvovati u prezentaciji proizvoda, pri čemu bi se mogli posvetiti svakom detalju proizvoda uz grafičke opise ili oznake, kao i mogućnost dostavljanja dodatne specifikacije ili materijala vezanih za proizvode ili usluge.

**PREGLED POSTOJEĆIH RJEŠENJA I  
MEHANIZAMA ZA POMOĆ PRI KUPOVINI**

Na tržištu je dostupan značajan broj sistema koji omogućavaju određene načine pomoći pri kupovini. Tako se na primjer na velikom broju web sajtova nalaze chat forme ili forme za kontakt pomoću email-a. Postoji i određen broj sistema koji omogućavaju pomoć pri kupovini na različitim uređajima, a bazirani su na WebRTC-u (eng. Web Real-Time Communication) [1]. Tako na primjer, SightCall [2] omogućava upotrebu različitih rješenja za interakciju baziranih na WebRTC-u, ne samo u oblasti trgovine, već i za primjene poput zdravstva, servisa, pomoći itd. Još jedno rješenje koje pruža slične funkcionalnosti je CreaLog [3]. Magento Shopping Assistant [4] takođe omogućava pomoć pri kupovini. Ova rješenja omogućavaju i opciju za pomoć pri kretanju po stranici, traženju proizvoda ili pokazivanje pozicija određenih opcija na stranici pomoću odgovarajućih simbola za označavanje, koje korisnik sajta može lako uočiti. Veliki broj komercijalnih rješenja korisnicima omogućava upotrebu API-ja, pa svoje usluge dostavlja kao servis. Postoji i određen broj istraživanja na temu primjene WebRTC-a u sistemima prodaje ili za pomoć kupcima [5][6].

WebRTC se pokazao veoma efikasnim u implementaciji sistema za saradnju u realnom vremenu [7]. Postoji nekoliko rješenja i istraživanja koja pokazuju da se razmjenom sadržaja preko P2P veze u realnom vremenu mogu kontrolisati akcije kod drugog korisnika, pa se na taj način prenose i izvršavaju akcije jednog korisnika kod drugog. Dobar primjer je upotreba alata za zajednički rad na tekstu ili grafičkom prikazu pomoću HTML5 Canvas elementa ili nekog drugog alata. Međutim, na osnovu istraživanja nije pronađeno rješenje koje koristi crtanje u realnom vremenu na zajedničkom Canvas elementu za potrebe promocije ili opisa proizvoda u trgovini.

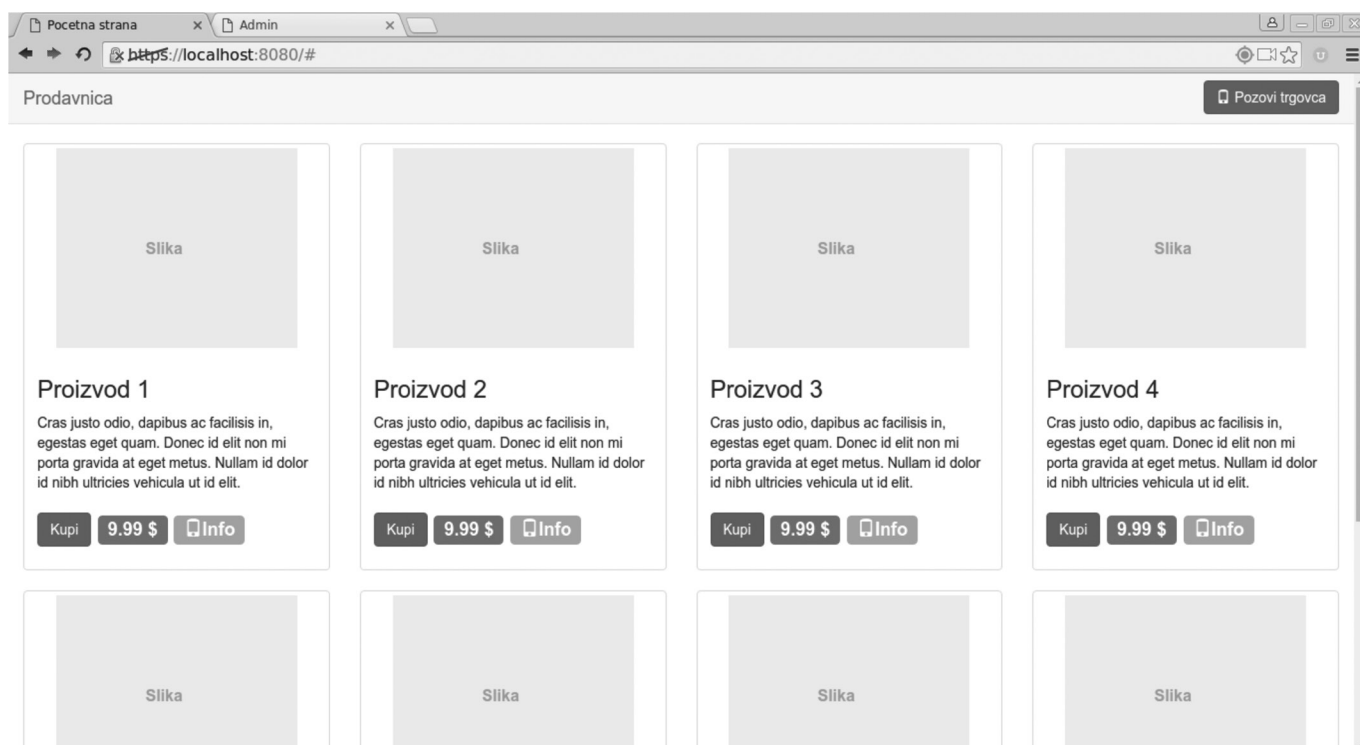
### PRIJEDLOG NOVOG RJEŠENJA

Na osnovu opisanih prednosti i nedostataka postojećih rješenja, može se zaključiti da bi novi sistem za pružanje pomoći pri kupovini, ali i samu kupovinu proizvoda ili usluga direktnom interakcijom kupca i trgovca trebao da posjeduje sljedeće osobine i funkcionalnosti:

- Mogućnost da kupac uspostavi vezu sa trgovcem (najmanje tekstualna komunikacija, ili audio/video poziv)
- Mogućnost da trgovac kupcu pošalje dodatne materijale vezane za proizvod ili uslugu
- Mogućnost interaktivnog opisa proizvoda
- Mogućnost rada više operatera na strani trgovca sa kupcima, tako da se obezbijedi da više kupaca istovremeno može razgovarati sa trgovcima
- Sistem mora da zadovolji osnovne sigurnosne zahtjeve
- Sistem treba da bude jednostavan za integraciju u veće poslovne sisteme prodavnica ili preduzeća
- Kupci i trgovci treba da budu u mogućnosti da sistem koriste sa različitih uređaja

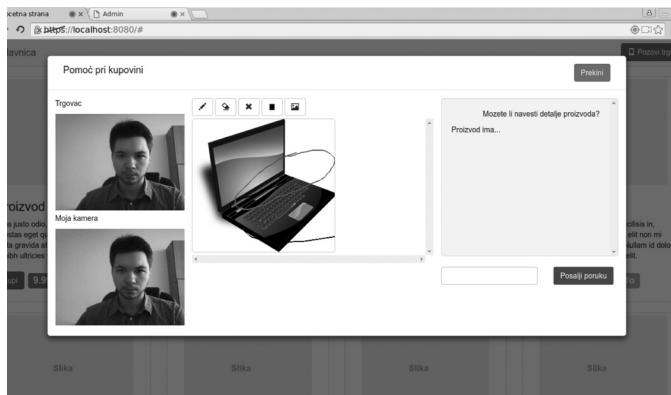
Kako bi se obezbijedila mogućnost interaktivne pomoći pri kupovini u internet prodavnicama, ili samoj prodaji, implementirano je rješenje koje ispunjava prethodno identifikovane potrebe i ciljeve. Osnovna ideja pri realizaciji rješenja bila je da se sistem može jednostavno integrisati u postojeće web aplikacije, pa su zbog toga pri implementaciji korištene isključivo web tehnologije. Pošto se sistem može koristiti u različitim vrstama aplikacija (prodavnice, web sajtovi firmi, udruženja i slično), tačnije u svim poslovnim sistemima gdje je potrebno povezati kupce i trgovce u web okruženju, implementacija funkcionalnosti vezanih za samu prodaju je izostavljena. Umjesto toga, napravljen je demo izgled stranice na kojoj se nalaze opcije za poziv prema trgovcu ili dežurnom operateru koji nije vezan za konkretan proizvod, ili poziv prema trgovcu uz slanje informacije da kupca zanima konkretan proizvod ili usluga, kao što je prikazano na slici 1. Korisnički dio aplikacije sastoji se iz dva dijela, jedan koji je namijenjen kupcu i drugi koji koriste trgovci.

Kada kupac pozove trgovca, otvara se dodatni dijalog na kojem se nalaze prostor za razmjenu tekstualnih poruka, prostor za grafičko predstavljanje detalja proizvoda sa opcijama za crtanje i prikaz slika, prostori za video prikaz kupca i trgovca. Video razgovor se može koristiti u slučaju da kupac želi da vidi trgovca, i obrnuto. Osim toga, upotrebom kamere omogućava se da trgovac detaljnije snimi i prikaže proizvod kupcu u realnom vremenu, ali i kupac može da detaljnije objasni trgovcu šta želi, ili da prikaže određeni postojeći proizvod. Osim toga, upotrebom kamere moguće je izvršiti pomoć pri popravci, sastavljanju ili upotrebi već kupljenog proizvoda, zbog toga što se opcije mogu koristiti u realnom vremenu kao da su kupac i trgovac na istoj lokaciji. Jedna od primjena alata za crtanje može biti da trgovac prikaže sliku proizvoda i zatim izvrši označavanje određenih dijelova proizvoda, što može



Slika 1 – izgled aplikacije na strani kupca

pomoći boljem objašnjenju načina upotrebe ili karakteristika proizvoda. Takođe, i kupac ima iste mogućnosti tako da mogu razgovarati o proizvodu. Izgled ekrana koji kupac ima na raspolaganju prikazan je na slici 2.



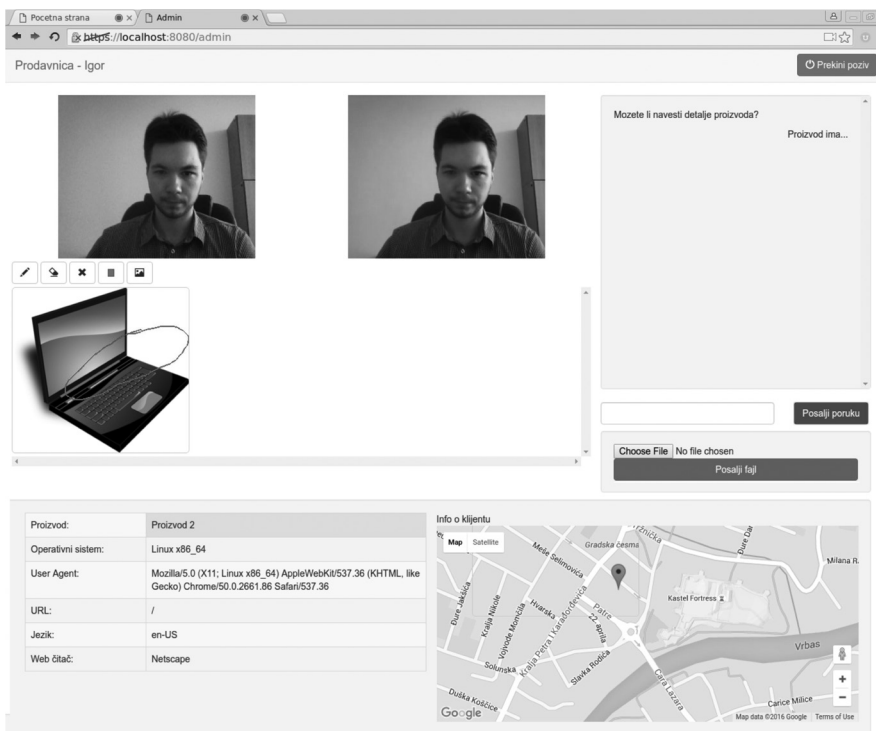
Slika 2 – izgled aplikacije na strani kupca tokom razgovora

Dio aplikacije koji koriste trgovci sastoji se iz elemenata za prikaz video sadržaja kupca i trgovca, prostora za crtanje, prostora za tekstualnu komunikaciju, opcije za slanje fajlova proizvoljnog formata kupcu i prostora za prikaz informacija o kupcu. U ovoj verziji se prikazuju samo osnovne informacije poput proizvoda koji kupac gleda, uređaja koji koristi, jezika, lokacije (ukoliko kupac dozvoli njeno otkrivanje). Na osnovu ovih dodatnih informacija moglo bi se izvršiti prilagođenje sadržaja koji je dostupan kupcu. Kao što je već navedeno, ova verzija aplikacije nije integrisana na neki prodajni sistem, ili sistem koji je specijalizovan za određenu kompaniju ili organizaciju, pa zbog toga nije obrađen način identifikovanja kupca u sistemu. Zbog toga se svi kupci prijavljuju anonimno. Do-

datna opcija koja bi se mogla obezbijediti, a koja može biti korisna trgovcu, je da se prikaže istorija kupovine ili interesovanja kupca (koje proizvode je gledao, koliko detaljno, kada...), što bi trgovcu omogućilo da kupcu ponudi i neke druge slične proizvode. Na osnovu informacija o lokaciji i jeziku operativnog sistema koji kupac koristi, može se znati jezik kojim će se obaviti razgovor sa kupcem. Osim toga, lokacija na kojoj se kupac nalazi može uticati i na načine plaćanja ili dostavu proizvoda. Izgled aplikacije za trgovca prikazan je na slici 3.

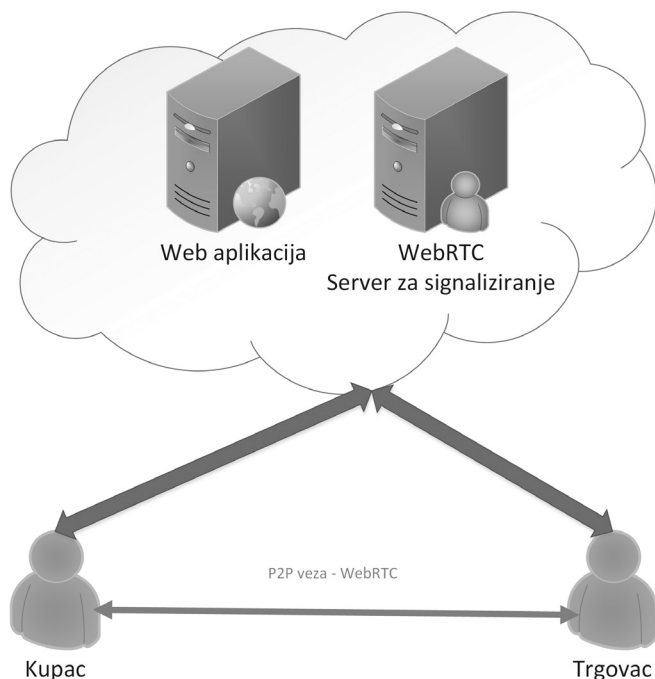
### ARHITEKTURA SISTEMA

Pošto je jedan od zahtjeva sistema da se omogući uspostavljanje video poziva uz korištenje dodatnih funkcionalnosti istovremeno sa većim brojem korisnika, poželjno je da centralni server bude što manje opterećen. Zbog toga je poželjno da se veza između svakog kupca i trgovca ostvari direktno, na način da se audio/video sadržaj prenosi direktno peer-to-peer (P2P) vezom. Na ovaj način centralni server ne bi bio dodatno opterećen, a klijenti se u svakom slučaju mogu nalaziti na proizvoljnoj lokaciji. Osim toga, pošto se sistem treba koristiti u okviru web aplikacije na različitim uređajima i platformama, poželjno je da korisnici sistema ne moraju instalirati dodatne aplikacije ili dodatke (eng. plug-in). Zbog svega navedenog za implementaciju komunikacije između korisnika sistema iskorišten je WebRTC. Za uspostavljanje WebRTC P2P veze potreban je posrednik za otkrivanje puta između učesnika veze, i taj server se naziva server za signaliziranje [1]. Sistem, osim navedenog, posjeduje i web aplikaciju koja obezbijeduje grafički prikaz svih funkcionalnosti i interakciju sa sistemom. Ovaj dio se može jednostavno dodati na neke već postojeće sisteme. Osim navedenih elemenata, sistem za uspostavljanje



Slika 3 – izgled aplikacije na strani trgovca tokom razgovora

P2P veze koristi i STUN (eng. Session Traversal Utilities for NAT) i TURN (eng. Traversal Using Relays around NAT) servere, čija je uloga detaljnije opisana u nastavku. Šema arhitekture sistema prikazana je na slici 4.



Slika 4 – arhitektura implementiranog rješenja

## WEBRTC

WebRTC je skup protokola, standarda i JavaScript API-ja koji omogućavaju uspostavljanje direktne veze između klijenata preko koje se obavlja audio i video komunikacija, razmjena tekstualnih poruka i datoteka. Klijenti u WebRTC vezi su web čitači, a za obavljanje komunikacije nije potrebno instalirati dodatne plugin-e [8].

WebRTC pruža JavaScript API koji se koristi iz web aplikacije za uspostavljanje veze i prenos podataka, a osim toga sadrži brojne funkcionalnosti koje nisu vidljive krajnjem programeru: dobijanje i obrada audio i video sadržaja, upotreba protokola, uspostavljanje veze između korisnika, otkrivanje korisnika, zaštita komunikacije, slanje podataka o komunikaciji (prekid, promjena parametara...). WebRTC se može koristiti na svim uređajima na kojima je instaliran web čitač koji podržava WebRTC. Postoji mogućnost da se WebRTC integriše sa komunikacionim sistemima poput VOIP, SIP (eng. Session Initiation Protocol) ili postojeće sisteme javne telefonije PSTN (eng. Public Switched Telephone Network) [9]. WebRTC API se sastoji iz [1]:

- MediaStream – API koji omogućava korištenje audio i video tokova (eng. stream)
- RTCPeerConnection – API koji omogućava razmjenu audio i video sadržaja
- RTCDataChannel – API koji omogućava razmjenu tekstualnog sadržaja i poruka

## PREUZIMANJE AUDIO I VIDEO TOKOVA SA UREĐAJA I MEDIASTREAM API

MediaStream API [1] omogućava pristup audio i video uređajima bez korištenja drugih plugin-a ili drajvera. Takođe, audio i video tokovi se automatski obrađuju kako bi se poboljšao kvalitet, omogućila sinhronizacija audio i video sadržaja, uskladio kašnjenje, redukovao eho, spriječile greške pri transportu i slično. Audio i video tokovi enkoduju se odgovarajućim kodecima (Opus, VP8). Kod prijema tokova vrši se dekodovanje primljenih tokova i oni se koriste na uređajima poput ekrana i zvučnika. Svi navedeni procesi su implementirani u web čitaču, automatski se izvršavaju i direktno su podržani kroz API.

Pristup i dobijanje audio i video tokova se obavlja pomoću MediaStream objekta, koji se sastoji iz jedne ili više traka (MediaStreamTrack) koje su sinhronizovane. Ulaz može biti uređaj poput mikrofona ili kamere, fajl sa uređaja ili mrežni čvor. Izlaz MediaStream objekta može se koristiti u audio ili video HTML5 elementu, za obradu pomoću JavaScripta ili za slanje udaljenom čvoru. Audio i video tokovi mogu se dobiti odvojeno. Za pristup audio i video tokovima koristi se getUserMedia() metoda kojoj se prosljeđuju parametri. Dobijanje i korištenje video i audio uređaja se može razlikovati u različitim web čitačima. Razlike su vezane za objekte koji su implementirani u web čitačima, pa je potrebno obezbijediti podršku za sve web čitače. Web čitač je obavezan da od korisnika eksplicitno zatraži dozvolu za korištenje medijskih uređaja i da tokom korištenja prikaže odgovarajuću poruku. Pri implementaciji sistema koje se prezentuje u radu iskorištena je biblioteka adapter.js, koja olakšava prepoznavanje web čitača i pristup medijskim resursima.

## RTCPEERCONNECTION API

RTCPeerConnection API [1] je zadužen za upravljanje ciklusom P2P veze između web čitača. Pomoću ovog API-ja moguće je uspostaviti, održavati, koristiti i završiti P2P vezu. RTCPeerConnection automatski obavlja poslove vezane za prenos multimedijalnog sadržaja u realnom vremenu, poput prikrivanja gubitka paketa, smanjivanja eha, prilagođavanja mrežnog protoka, regulisanje pojačanja i slično. RTCPeerConnection je takođe zadužen za:

- upravljanje ICE procesom uspostavljanja veze preko NAT-a
- slanje STUN keepalive poruka povezanim čvorovima
- čuvanje lokalnih i udaljenih medijskih tokova
- automatsko izvršavanje ponovnog procesa spajanja, ako je to potrebno
- upravljanje pozivima iz aplikacije preko API-ja

## USPOSTAVLJANJE P2P VEZE

Da bi se uspostavila P2P veza potrebno je da postoji put preko mreže kojim je moguće poslati poruku od jednog do drugog čvora. U današnjim mrežama većina čvorova se nalazi iza NAT dijelova mreže, pa zbog toga često nije moguće direktno pronaći čvor kako bi se uspostavila veza. Takođe, potrebno je razmotriti slučajeve u kojima čvor ne mora da prihvati uspostavljanje veze ili može biti nedostupan ili zauzet.

Za otkrivanje odredišnog čvora može se iskoristiti posrednik (server) između dva čvora koji će biti dostupan za oba čvora i koji može da razmjenjuje poruke i parametre. U tom slučaju čvorovi se javljaju serveru i daju informacije o svom stanju. Kada jedan čvor želi da se poveže sa drugim, informacije o njemu može da dobije od servera. Nakon toga čvor može da pokuša da uspostavi vezu sa drugim čvorom. Uloga servera u povezivanju i upravljanju vezom između čvorova naziva se signaliziranje. Veoma važno ograničenje predstavlja mrežna struktura, NAT i firewall. Uloga servera u uspostavljanju veze između dva čvora je:

- Otkrivanje korisnika (lokacija na mreži, informacije o korisniku)
- Signaliziranje
- Zaobilazanje NAT-a i firewall-a
- Uspostavljanje veze preko zamjenskog servera ukoliko se ne može uspostaviti P2P veza

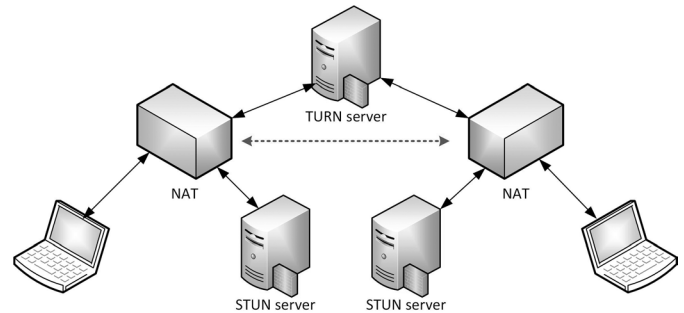
Server za signaliziranje, može biti implementiran na različite i ne postoji konkretna preporuka ili pravilo vezano za implementaciju servera za signaliziranje. Moguće je realizovati proizvoljno rješenje (HTTP, WebSocket...), koristiti SIP, Jingle, ISUP (eng. ISDN User Part) [1].

Ukoliko čvorovi imaju pristup serveru za signaliziranje moguće je pokrenuti proces za uspostavljanje P2P veze. Razmjena informacija o parametrima P2P veze obavlja se pomoću SDP (eng. Session Description Protocol) protokola. Pomoću SDP protokola se ne prenose audio i video sadržaj, već se definiše vrsta sadržaja koji se prenosi (audio, video, aplikacioni podaci), parametri mrežnog prenosa, kodeci, informacije o mrežnom protoku kao i ostali metapodaci. WebRTC aplikacije ne rade direktno sa SDP protokolom. Za definisanje i upravljanje vrijednostima SDP parametara zadužen je JavaScript Session Establishment Protocol (JSEP) koji se poziva pomoću metoda iz RTCPeerConnection API-ja.

Veliki problem pri uspostavljanju P2P veze čak i uz korištenje servera za signaliziranje predstavljaju mrežna ograničenja zbog NAT-a i firewall-a. NAT mehanizam omogućava zamjenu javnih adresa privatnim, na rubnim dijelovima mreže. Drugim riječima, privatne adrese su sakrivene iza javne adrese preko koje ostvaruju pristup internetu. Na taj način više hostova koristi istu javnu adresu čime se ostvaruje ušteda javnih adresa. Uloga NAT-a je prevođenje privatnih u javne adrese i obrnuto. Zbog toga prilikom uspostavljanja P2P veza hostovi često ne znaju javnu adresu preko koje izlaze na internet, već samo znaju svoju privatnu adresu. Ukoliko se pošalje paket na javnu adresu, bez informacije o privatnoj adresi odredišnog host-a paket će biti odbačen u većini slučajeva (osim ukoliko je konfigurisan port-forwarding ili sličan mehanizam na ruteru koji ima javnu adresu). Navedena ograničenja se mogu riješiti pomoću STUN, TURN i ICE (eng. Interactive Connectivity Establishment) protokola [10].

STUN protokol se koristi za otkrivanje javne adrese i porta hosta. Pomoću STUN protokola šalje se zahtjev do STUN servera koji se nalazi na javnoj mreži, a čija IP adresa je poznata. Aplikacija šalje zahtjev prema STUN serveru, a kao odgovor prima IP adresu i port koji STUN server pročita iz primljenog zahtjeva. STUN protokol posjeduje mehanizam koji vrši provjeru putanja do klijenta kako bi provjeravao status po-

vezanosti (eng. keepalive ping). Prilikom uspostavljanja P2P veze, klijenti prvo šalju zahtjev prema STUN serveru a potom mogu iskoristiti dobijene informacije o IP adresi i portu za uspostavljanje veze. STUN protokol nije efikasan u 8% slučajeva [1], zbog mrežnih ograničenja. U tim slučajevima potrebno je primjeniti posrednički server (eng. relay) preko kojeg će se razmjenjivati sadržaj. Za te namjene koristi se TURN, što je prikazano na slici 5.



Slika 5 – upotreba STUN i TURN servera

TURN protokol se primjenjuje kada nije moguće uspostaviti direktnu vezu između čvorova. TURN server se nalazi na javnoj mreži, a oba klijenta se povezuju na isti TURN server i nakon povezivanja šalju sadržaj preko TURN servera, koji je zadužen za dostavljanje sadržaja drugom čvoru. U navedenom slučaju, između čvorova nije uspostavljen veza i kompletan sadržaj ide preko TURN servera. TURN server treba da ima kapacitet za prenos sadržaja za više korisnika.

### INTERACTIVE CONNECTIVITY ESTABLISHMENT (ICE)

U najjednostavnijem slučaju uspostavljanja P2P veze oba čvora se nalaze u lokalnoj mreži ili imaju javne IP adrese. Tada se nakon dobijanja IP adresa, generišu SDP zahtjevi nakon čije razmjene je uspostavljena P2P veza. U praksi se čvorovi često nalaze ili u različitim privatnim mrežama, ili je jedan čvor na javnoj mreži. U tim slučajevima samo pomoću SDP-a ne bi bilo moguće uspostaviti P2P vezu, jer put između čvorova nije poznat. Rješenje koje WebRTC posjeduje u ovim slučajevima je ICE agent.

ICE agent [1][8] je zadužen za dobijanje lokalne IP adrese i portova (kandidati), provjeru veze između čvorova i održavanje veze slanjem keepalive poruka. Nakon što se postavi lokalni ili udaljeni opis sesije (SDP), ICE agent automatski započinje proces otkrivanja kandidata za lokalni čvor. Ovaj proces se obavlja u sljedećim koracima:

1. ICE agent od operativnog sistema dobija lokalne IP adrese,
2. Ako je konfigurisan, ICE agent od STUN servera dobija javnu IP adresu i port čvora,
3. Ako je konfigurisan, ICE agent koristi TURN server kao posrednika u komunikaciji, pod uslovom da nije moguće uspostaviti direktnu vezu između čvorova.

Kada se pronade novi kandidat, ICE agent registruje kandidata u RTCPeerConnection objekat i aktivira se onicecandida-

te funkcija. Proces otkrivanja se izvršava u pozadini, a nakon što završi moguće je generisati i poslati SDP zahtjev. Kada drugi čvor dobije kandidate, postavlja se opis udaljene sesije, odnosno lista kandidata udaljenog čvora. Tada ICE agent može početi sa provjerom mrežnog puta između čvorova. Provjera mrežnog puta počinje slanjem STUN zahtjeva koji treba da završi dobijanjem STUN odgovora. Kada je STUN odgovor primljen P2P veza se može uspostaviti jer je pronađen mrežni put između čvorova. Ukoliko se ne uspije uspostaviti veza nakon provjere svih kandidata tada se koristi TURN server kao posrednik, a sva komunikacija prolazi preko TURN servera.

Trickle ICE predstavlja mehanizam pomoću kojeg se ubrzava vrijeme traženja kandidata. Kako bi se smanjilo vrijeme potrebno za dobijanje informacija od STUN servera i provjere da li postoji mrežni put između čvorova za sve kandidate, Trickle ICE omogućava inkrementalno prikupljanje i provjeru puteva između čvorova na sljedeći način:

1. Čvorovi razmjenjuju SDP zahtjeve bez ICE kandidata,
2. ICE kandidati se šalju preko servera za signaliziranje čim se otkriju,
3. Provjera mogućnosti povezivanja se izvršava čim se dobije opis kandidata.

Ovaj način povezivanja proizvodi više saobraćaja preko servera za signaliziranje, ali ubrzava proces povezivanja.

Nakon što je uspostavljena P2P veza, razmjena audio, video i tekstualnih podataka se obavlja upotrebom nekoliko protokola:

- Datagram Transport Layer Security (DTLS) – koristi se za razmjenu tajnih ključeva za enkripciju multimedijiskog sadržaja i podataka
- Secure Real-Time Transport Protocol (SRTP) – koristi se za prenos audio i video sadržaja
- Stream Control Transport Protocol (SCTP) – koristi se za prenos podataka

#### DATACHANNEL API

DataChannel API [1] omogućava dvosmjernu P2P komunikaciju slanjem tekstualnih poruka preko jednog ili više kanala. Kanali se otvaraju nakon što je uspostavljena RTCPeerConnection veza. Za svaki kanal definišu se funkcije koje obrađuju različite događaje, pri čemu su posebno značajni događaji vezani za otvaranje i zatvaranje kanala i prijem poruke. Poruke koje se razmjenjuju mogu biti tekstualne ili binarni podaci (blob tip). Slanje poruke obavlja se funkcijom send, iz DataChannel objekta. Slanje fajlova se može implementirati na način da se fajl pročita sa diska korištenjem FileAPI-ja iz HTML5, zatim se podijeli na manje dijelove (chunk) koji se pretvaraju u tekstualne podatke i tako šalju preko DataChannel-a. Na prijemnoj strani potrebno je primati dijelove, čuvati ih na privremenoj lokaciji (buffer) i potom spojiti u cijeli fajl i tako spojen fajl smjestiti na disk uređaja. Drugi način slanja fajlova je pomoću blob tipa, pri čemu nije potrebno dijeliti i spajati fajl već se slanje obavlja direktno, odnosno jednim pozivom send funkcije. Izbor načina slanja zavisi od toga da li je slanje blob fajla podržano u web čitaču ili nije. Kroz DataChannel moguće je slati i JavaScript objekte (JSON), ako se prethodno pretvore u string.

DataChannel nalazi na vrhu steka tri protokola: UDP, koji obezbijedi P2P komunikaciju, DTLS koji obezbijedi enkripciju podataka i SCTP koji obezbijedi kontrolu prenosa

poruka. DataChannel se, kao što je već navedeno, može prilagoditi da se obezbijedi pouzdano dostavljanje poruka u redosljednu slanja za svaki kanal koji je otvoren u aplikaciji na različite načine. Kanali se mogu konfigurirati tako da se obezbijedi slanje poruka u ili bez redosljeda, te da se obezbijedi potpuna ili djelimična pouzdanost kanala. U slučaju da slanje poruke nije uspješno, moguće je automatski ponoviti slanje.

#### WEB APLIKACIJA

Kao što je već navedeno, web aplikacija se zbog univerzalnosti sastoji samo iz dva dijela, za rad trgovca i za kupca, a razvijena je pomoću HTML5, JavaScript-a, CSS3, jQuery i Bootstrap biblioteka. Aplikacija se nalazi na NodeJS serveru. Funkcionalnosti web aplikacije su ranije opisane, a u nastavku će biti objašnjen način implementacije svih značajnih funkcionalnosti.

Veza između kupca i trgovca realizovana je pomoću WebRTC-a. Oba dijela web aplikacije uspostavljaju vezu sa serverom za signaliziranje, čime se omogućava otkrivanje učesnika i razmjena parametara potrebnih za uspostavljanje P2P veze. Komunikacija između web aplikacije i servera za signaliziranje obavlja se pomoću WebSocket-a. Povezivanje sa serverom za signaliziranje obavlja se pri učitavanju stranice. Kada kupac odluči da pozove trgovca, tada se šalje odgovarajuća naredba na server za signaliziranje, pronalazi se prvi slobodan trgovac i započinje se sa uspostavljanjem WebRTC P2P veze. Ukoliko nema slobodnih trgovaca za razgovor, kupac se obavještava o tome. Prilikom uspostavljanja veze, podrazumijevana konfiguracija je da se koristi audio, video i DataChannel, ali u slučaju da određeni korisnik ne posjeduje neki od uređaja, ili ne želi da ga koristi, tada se uspostavlja samo komunikacija preko DataChannel-a. Kada se uspostavi P2P veza, tada se za prikaz video sadržaja koristi video HTML5 element. Prikazane su kamere kupca i trgovca, ukoliko oba učesnika dozvole upotrebu svojih kamera.

Nakon što se uspostavi P2P veza, otvara se DataChannel pomoću kojeg su implementirane opcije za chat, razmjenu fajlova i slanje dodatnih poruka koje su značajne za upotrebu određenih funkcionalnosti sistema. Svaka poruka je u JSON formatu, a prije slanja se pretvara u string kako bi se mogla koristiti u različitim web čitačima na isti način. Poruka ima polje tip, koji određuje namjenu sadržaja poruke, a tipovi mogu biti chat poruka, fajl poruka, poruka za crtanje i poruka sa informacijama o kupcu. Svaki tip poruke ima različita polja, a na osnovu toga se izvršavaju određene funkcionalnosti. Slanje fajla i poruka je implementirano na način koji je prethodno opisan u radu, a bitno je naglasiti da se po prijemu fajla automatski pokreće njegovo preuzimanje.

Prostor na kojem trgovac i klijent mogu postavljati slike, crtati po postavljenim slikama ili praznoj površini implementiran je pomoću HTML5 Canvas elementa. Opcije za crtanje, brisanje, izbor boje i postavljanje slike su implementirane u posebnom meniju iznad Canvas prostora. Oba korisnika mogu istovremeno izvršavati različite akcije na prostoru za crtanje. Nakon što neki korisnik izabere određenu opciju iz menija, registruje se i obrađuje svaki pokret i događaj na komponenti za crtanje. Podržani su događaji na klik i dodir, kako bi se omogućila podrška za različite vrste uređaja. Svi registrovani događaji se obrađuju i ogovarajuće akcije se evidentiraju u komponenti (cr-

tanje, brisanje, postavljanje slike...). Svaka akcija se ima početak (touchstart ili mousedown), izvršavanje (touchmove, mousemove) i kraj (touchend, mouseup). Prilikom registrovanja i obrade svakog događaja lokalno, u privremenu strukturu podataka se evidentira svaki događaj, i nakon završetka akcije ta struktura se šalje drugom korisniku. Prikupljeni podaci se sastoje iz oznake iskorištene opcije i dodatnih polja, poput koordinata tačaka za crtanje krivih linija, boje linije, koordinata prostora koji se briše i slično. Kada neki korisnik dobije poruku za crtanje, tada se na osnovu oznake čitaju događaji koje je drugi korisnik napravio i oni se izvršavaju lokalno. Izvršavanje akcija koje je drugi korisnik napravio ne utiče na izvršavanje lokalnih događaja, već se oni mogu izvršavati istovremeno bez blokiranja. Kada se koristi opcija za postavljanje slike kao pozadine u prostoru za crtanje, tada korisnik prvo bira i učitava određenu sliku sa uređaja, a ona se zatim lokalno dodaje u prostor za crtanje i šalje drugom korisniku preko DataChannel-a, u obliku odgovarajuće poruke. Po prijemu se slika prikazuje u prostoru i na taj način oba korisnika mogu da vide i rade na istom sadržaju.

Izgled web aplikacije je realizovana upotrebom Bootstrap biblioteke, pa se pomoću nje na jednostavan način izgled prilagođava dimenzijama uređaja na kojem se koristi. Implementirane funkcionalnosti su dostupne preko web čitača uređaja, pa ukoliko web čitač na proizvoljnom uređaju (tablet, računar, pametni telefon...) podržava WebRTC, tada se i sistem može koristiti. Kao što je već navedeno, da bi se opcija za crtanje mogla koristiti na različitim vrstama uređaja podržana je detekcija događaja vezanih i za klik miša i za dodir ekrana.

Dodatne informacije o kupcu preuzimaju se dijelom iz JavaScript objekata (location, navigator), a dijelom sa stranice (informacija o proizvodu koji korisnik gleda). Dobijanje lokacije kupca obavlja se pomoću HTML5 Geolocation API-ja, za šta kupac mora dati eksplicitnu saglasnost. Na administratorskoj stranici se podaci o geolokaciji prikazuju na mapi pomoću Google Maps API-ja.

### SERVER ZA SIGNALIZIRANJE

Server za WebRTC signaliziranje obezbjeđuje otkrivanje korisnika i uspostavljanje P2P WebRTC veze između njih. Osim toga, ovaj server je zadužen za praćenje trajanja svih poziva, što se može evidentirati u bazu podataka, kao i kontrolu dostupnosti i povezivanja.

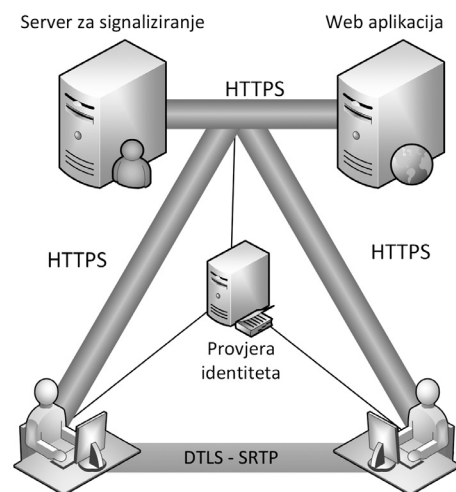
Server je implementiran pomoću NodeJS-a. Komunikacija između servera i web aplikacije obavlja se preko HTTPS protokola. Server koristi socket.io biblioteku za uspostavljanje veze sa web aplikacijom. Biblioteka socket.io omogućava dvosmjernu komunikaciju u realnom vremenu, koja je bazirana na događajima. Svaki događaj ima svoju oznaku, i u kodu se definišu funkcije koje obrađuju svaki događaj. Za uspješnu komunikaciju između servera za signaliziranje i web aplikacije potrebno je definisati funkcije za obradu događaja na obje strane, dok se iniciranje događaja može uraditi slanjem odgovarajućih poruka prema jednom kraju veze ili broadcast poruka svim učesnicima.

Na serveru se čuva lista svih trgovaca koji su prijavljeni na server. Za svakog trgovca čuva se korisničko ime i status dostupnosti (slobodan, zauzet). Svaki trgovac ima svoju sobu za razgovore, čiji je identifikator korisničko ime. Ova soba

predstavlja prostor gdje se povezuju trgovac i kupac, kako bi međusobno razmjenjivali poruke za uspostavljanje P2P veze. Trgovcu se dodjeljuje soba za razgovor čim završi proces prijave na sistem, slanjem odgovarajuće poruke. Tada se trgovac dodaje u listu, na posljednje mjesto. Kada određeni kupac aktivira opciju poziva, tada se na serveru pronalazi prvi trgovac koji je slobodan, i kupac se registruje u njegovu sobu za razgovor. Nakon toga, trgovac postaje zauzet i počinje se sa uspostavljanjem poziva. Sve dok je trgovac zauzet, drugi kupci ga ne mogu kontaktirati. Kada se završi poziv, ili se uopšte ne uspostavi iz proizvoljnog razloga, tada trgovac postaje slobodan i drugi kupci ga mogu kontaktirati. Ukoliko trgovac ili kupac odluče za završe razgovor ili zatvore odgovarajuću web stranicu, na server pristiže odgovarajuća poruka koja se prosljeđuje drugom korisniku.

### SIGURNOST SISTEMA

Sigurnost implementiranog sistema zasnovan je na principima WebRTC-a [11] [12]. WebRTC uvijek koristi zaštićenu komunikaciju između krajnjih korisnika (P2P veza) pomoću DTLS-SRTP protokola. Za zaštitu komunikacije između krajnjih korisnika i servera za signaliziranje upotrebljen je TLS protokol. TLS obezbjeđuje tajnost, integritet i autentifikaciju. Tajnost podataka koji se razmjenjuju tokom komunikacije se obezbjeđuje upotrebom simetričnih algoritama. Integritet razmijenjenih poruka se postiže upotrebom hash funkcija, a autentifikacija je moguća na serverskoj i klijentskoj strani. Autentifikacija se postiže upotrebom digitalnih sertifikata. Serverska autentifikacija je obavezna, dok klijentska nije. DTLS obezbjeđuje integritet i ekripciju, ali ne obezbjeđuje autentifikaciju. U WebRTC vezama, DTLS automatski generiše samopotpisani digitalni sertifikat za sve učesnike. Ovako generisani sertifikati ne mogu biti potvrđeni od strane drugog sertifikacionog tijela, što onemogućava pouzdanu autentifikaciju. Navedeno ograničenje se može prevazići upotrebom drugog načina autentifikacije na aplikativnom sloju (npr. prijava na aplikaciju prije uspostavljanja WebRTC veze). SRTP protokol obezbjeđuje enkripciju, integritet, autentifikaciju poruka i sprečava ponovno slanje podataka. Principijelna šema WebRTC sigurnosnog modela prikazana je na slici 6 [13].



Slika 6 – principijelna šema WebRTC sigurnosnog modela

Jedan od osnovnih sigurnosnih principa WebRTC-a je da se prilikom uspostavljanja poziva korisnicima prikažu informacije o učesnicima veze, a potrebno je i obezbijediti mehanizam autorizacije i provjere identiteta svih učesnika. U implementiranom sistemu ne postoji centralizovan mehanizam identifikovanja korisnika, pa se u dijelu aplikacije namijenjenom trgovcima identifikacija obavlja unosom imena koje mora biti jedinstveno. Dio aplikacije koji koriste kupci ne posjeduju mehanizam identifikacije, iz razloga što je pomoć namijenjena svim posjetiocima sajta. Na jednostavan način se predloženi sistem može integrisati u neki postojeći, ili se može dodati mehanizam registracije korisnika, pa se ti podaci mogu koristiti pri identifikovanju učesnika komunikacije.

Prije upotrebe kamere i mikrofona korisnik je obavezan da eksplicitno dozvoli njihovu upotrebu. Tokom korištenja mikrofona i kamere korisniku se mora prikazati oznaka da se navedeni uređaji koriste, a takođe korisnik treba da ima mogućnost prekida korištenja tih uređaja. Sva direktna komunikacija mora biti potvrđena od strane ICE protokola, a iz aplikacije se pomoću JavaScripta ne smije pristupiti identifikatoru transakcije. Dok korisnik ne prihvati uspostavljanje veze ICE mehanizam ne smije početi sa radom, kako bi se zaštitila IP adresa korisnika, odnosno njegova lokacija. Nakon uspostavljanja veze javna adresa korisnika će biti otkrivena. Za razmjenu podataka obavezno je korištenje DTLS protokola, a za razmjenu medijskog sadržaja obavezna je upotreba DTLS-SRTP protokola, čime se obezbijuje zaštita svih prenesenih podataka.

## ZAKLJUČAK

Trenutno postoji značajan broj tehnologija i protokola koji omogućavaju web zasnovanu komunikaciju u realnom vremenu. WebRTC je trenutno jedna od najperspektivnijih specifikacija, koja će se sve češće koristiti zbog svoje jednostavnosti i dobrih osobina. Sva komunikacija i upotreba uređaja poput mikrofona ili web kamere se odvija pod kontrolom web čitača, tako da programeri i krajnji korisnici ne treba da vode računa o kontroli ovih uređaja. Programerima je dostupan API, a osim poznavanja aktuelnih web tehnologija (HTML5 i JavaScript) programeri ne treba da poznaju dodatne tehnologije, čime je implementacija WebRTC-ja u web aplikacijama veoma jednostavna. Svi uređaji koji imaju aktuelne verzije web čitača mogu da koriste WebRTC, čime se omogućava komunikacija između korisnika sa različitim tipovima uređaja, i to bez potrebe razvoja posebne aplikacije za svaki tip uređaja.

Implementiranje web zasnovane komunikacije u realnom vremenu moguće je u svim postojećim web aplikacijama koje su namijenjene trgovini ili bilo kojoj interakciji sa krajnjim korisnicima određenih usluga. Zbog svoje jednostavnosti, WebRTC se jednostavno dodaje na postojeće aplikacije, čime se kupcima pružaju dodatne funkcionalnosti (chat, pozivi, dobijanje dodatnih materijala vezanih za proizvode, uputstva...).

Upotrebom alata za interakciju kupaca i trgovaca, sa ciljem da se za određeni proizvod dobiju dodatne i specifične informacije označavanjem dijelova slike, crtanjem ili bilo kojim proizvoljnim načinom objašnjavanja, postiže se bolji efekat pri trgovini, čime se smanjuju razlike između tradicionalne i online prodaje. Na kraju se može zaključiti da razvoj HTML5 i pratećih tehnologija doprinosi poboljšanju mnogih internet servisa, što se u sistemima za prodaju može iskoristiti za približavanje proizvoda i usluga krajnjem kupcu, što je i opisano u radu.

## LITERATURA:

- [1] Ilya Grigorik, High Performance Browser Networking, O'Reilly Media, Inc, 2013.
- [2] <http://www.sightcall.com>, posjećivano marta 2016.
- [3] <https://www.crealog.com/>, posjećivano marta 2016.
- [4] <https://github.com/meanbee/magento-shopping-assistant>, posjećivano marta 2016.
- [5] N. Hongo, H. Yamamoto and K. Yamazaki, "Web shopping support system for elderly people using WebRTC," Advanced Communication Technology (ICACT), 2014 16th International Conference on, Pyeongchang, 2014, pp. 934-940.
- [6] Salvatore Longo, Ernő Kovacs, Jörn Franke, Miquel Martin, Enriching Shopping Experiences with Pervasive Displays and Smart Things, UbiComp'13, September 8–12, 2013, Zurich, Switzerland
- [7] A. Zeidan, A. Lehmann and U. Trick, "WebRTC enabled multimedia conferencing and collaboration solution," WTC 2014; World Telecommunications Congress 2014; Proceedings of, Berlin, Germany, 2014, pp. 1-6.
- [8] Salvatore Loreto and Simon Pietro Romano, Real-Time Communication with WebRTC, O'Reilly Media, maj 2014.
- [9] Amirante, A., Castaldi, T., Miniero, L., & Romano, S. P. (2013). On the seamless interaction between webRTC browsers and SIP-based conferencing systems. Communications Magazine, IEEE, 51(4), 42-47.
- [10] Alan Johnston, John Yoakum, and Kundan Singh, Taking on WebRTC in an Enterprise, IEEE Communications Magazine, april 2013
- [11] Barnes, R., Thomson, M., Browser-to-Browser Security Assurances for WebRTC, Internet Computing, IEEE (Volume:PP, Issue:99), septembar 2014
- [12] Desmet, Lieven ; Johns, Martin, Real-Time Communications Security on the Web, Internet Computing, IEEE Volume: 18 , Issue: 6, 2014.
- [13] Eric Rescorla, Proposed WebRTC Security Architecture, IETF 82, <http://www.ietf.org/proceedings/82/slides/rwcweb-13.pdf>, posjećeno mart 2016.



**M. Sci. Igor Dujlović, dipl. ing.** Elektrotehnički fakultet Banja Luka, RS, BiH  
**Kontakt:** igor.dujlovic@etfbl.net  
**Oblasti interesovanja:** Internet programiranje, objektno-orientisano programiranje i modelovanje, razvoj mobilnih aplikacija, informacijski sistemi

