2017

AMC Native WebRTC Client

Kellan Childers  
*Virginia Commonwealth University*

Michael Dowling  
*Virginia Commonwealth University*

Sherdil Khawaja  
*Virginia Commonwealth University*

Follow this and additional works at: [https://scholarscompass.vcu.edu/capstone](https://scholarscompass.vcu.edu/capstone)

Part of the [Computer Engineering Commons](https://scholarscompass.vcu.edu/capstone)

© The Author(s)

Downloaded from  
[https://scholarscompass.vcu.edu/capstone/147](https://scholarscompass.vcu.edu/capstone/147)

This Poster is brought to you for free and open access by the College of Engineering at VCU Scholars Compass. It has been accepted for inclusion in Capstone Design Expo Posters by an authorized administrator of VCU Scholars Compass. For more information, please contact libcompass@vcu.edu.
Introduction

- Traditional call centers and telecommunication hardware is being replaced by thin, browser-based, cloud enabled web services.
- Industry standards for web based communication protocols, such as WebRTC are being established.
- AMC needed to address this new technology, while maintaining a hybrid approach of server-based capabilities, taking advantage of the web-based communication channel, while broadcasting events to the Contact Canvas Server.

Description

- Contact Canvas Agent Palette is the editing platform of the AMC adapter for Salesforce.com, allowing agents to communicate with customers through the AMC adapter/ Softphone.
- Using Agent Palette, the task was to integrate Video Chat using WebRTC into the AMC toolbar.
- Two agents connect with one another over a peer-to-peer connection. Once the connected the two can communicate through video chat and also support Screen Pop.

Components

- We used the AMC adapter for Salesforce.com, including the Agent Palette, and the Salesforce.com Customer Relation Management (CRM).
- The AMC adapter is an HTML Softphone that can be used to voice enable salesforce.com.
- Implemented WebRTC into Contact Canvas Agent to video chat with customers. Socket.io and Node.js were used to communicate with the server side.

Layout

- Agent communicating with another through WebRTC video chat with separate computers.
- Eventually the goal is to use this same technology to establish communication between the agent and the customer.