Adding Audio/Video Call Functionality to SimpleX Apps

To extend the functionality of the SimpleX mobile apps in pursuit of supporting all kinds of communication, we seek to add the ability for already connected users to call each other with audio and optionally video.

Desired Functionality

- [ ] The content (audio/video data) of the calls is encrypted
- [ ] The setting up of the call session is secure and encrypted (via the SimpleX protocol)
- [ ] Contacts can audio call each other
- [ ] Contacts can video call each other
- [ ] When on a call users can mute/unmute their mic
- [ ] When on a video call users can show/hide their video feed
- [ ] Users will be notified of other calls when already engaged in a call
- [ ] Incoming calls trigger a notification which offers the chance to accept or reject the call. Accepting the call opens the app to a call page.
- [ ] (TBC) Calls will be entered into chat history as immutable messages with styling differing from typical messages

Proposed Implementation

The calls themselves should be handled by WebRTC. This requires some initial messaging to set up the details of the session (routing, codecs, message priorities) and then the data of the call is passed peer-to-peer through the WebRTC channel resulting from the session instantiation. In order to secure the communications, the initial communication to set up the session will be handled through the existing SimpleX communication channel between users. The content sent through the WebRTC session will also be encrypted using keys (exchanged through SimpleX). Full details of the workflow for setting up WebRTC calls can be found here.

To take advantage of existing development and infrastructure, we propose to build this functionality using webviews in our apps which will have web pages using JavaScript APIs to handle WebRTC elements.

Setting Up the Session

In essence, we can use SimpleX to handle the signalling with ICE agents performing negotiation at either end in the SimpleX mobile app. This requires the sharing of Session Description Protocol information which can be serialised as JSON. These can be passed as a new message type in the SimpleX API.

There are a few key features required to set up a call session.

1. Generation of Offers
2. Negotiation of Offers
3. Instantiating the call over the network
4. Showing the users the video streams

These elements can be included in a new 'call' message type which includes the information alongside the nature of the call (i.e. audio only or video).

User state etc can be updated as calls are connected/disconnected.

Initial Prototype

To get off the ground, we will develop an initial prototype (not for app store release). This prototype will simply set up a video call with no changes to the SimpleX API (as we will simply pass messages as JSON through other channels).

This prototype will demonstrate how to set up calls using WebRTC and demonstrate how to pass information to and from webviews in app.

The workflow will be as follows

1. 2.
   3.

Queries

Do we need to set up and destroy virtual IP addresses for additional security?

For initial implementation it is sufficient to warn users that they may be exposing their IP to the recipient.

Is it beneficial to have an additional layer of encryption for the media content (under the principle of zero trust or otherwise)?

Yes. We can implement a 'frame encryption' method in the SimpleX API which given a key and some content returns the encrypted content. Similarly, we will have a decryption call. Keys can be call specific and formed using typical DH key exchange.

Who runs the STUN/TURN servers? For the initial prototype we can use publicly available servers. For a full release implementation, SimpleX will need to run its own routing servers to support ICE and possibly STUN/ TURN. Open source implementations for these elements exist.