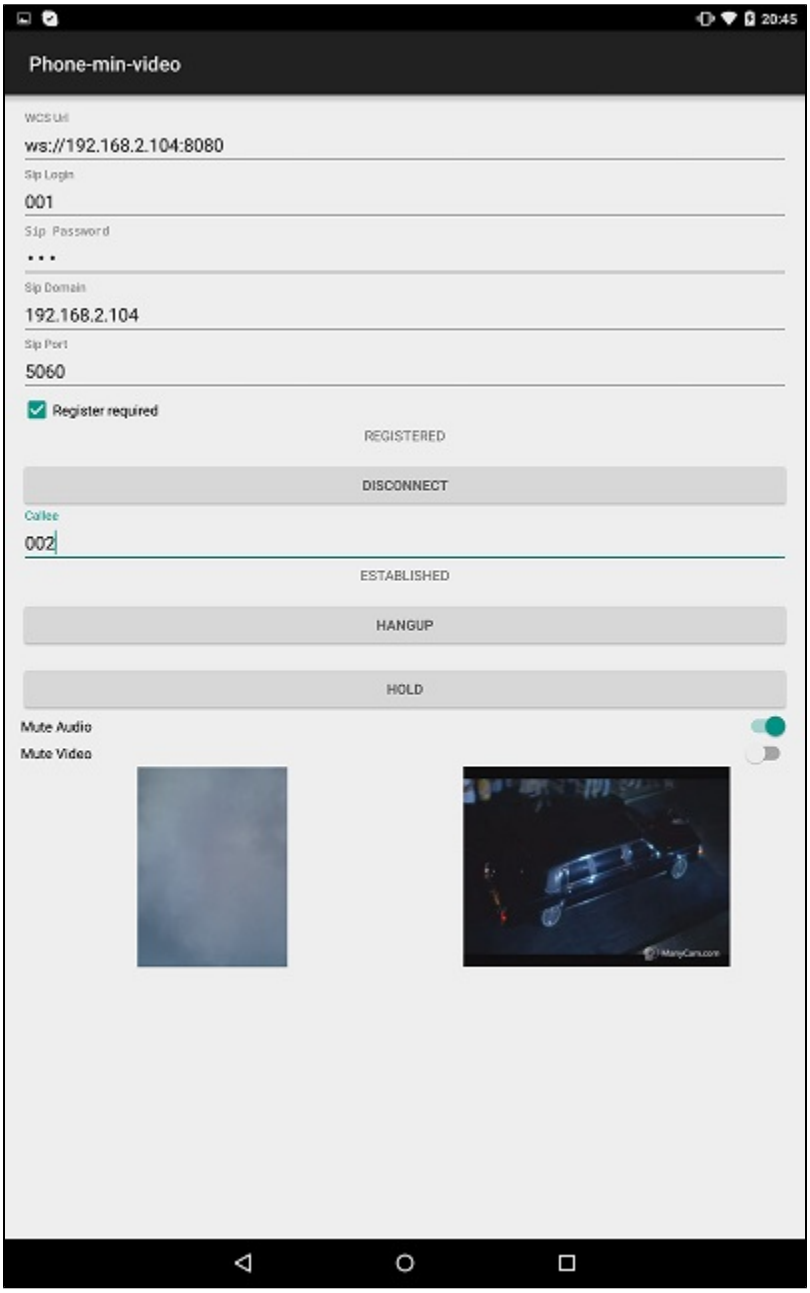
Android Phone Video

Example of Android application for video calls

On the screenshot below the example is displayed when a call is established.

The interface of he application is the same as in the example Phone, except that controls for muting/unmuting audio and video are added and two videos are played

- left - video from the camera of this user
- right - video from the other call party



Work with code of the example

To analyze the code, let's take class PhoneMinVideoActivity.javaof the phone-min-video example, which can be downloaded with corresponding build 1.0.1.38.

Functions of initialization, placing an outgoing call and answering incoming call work the same as described in the example Phone.

Differences from the example Phone:

1.Session creation.

Flashphoner.createSession()code

Object SessionOptionswith the following parameters is passed to method createSession() when session for connection to WCS server is created

- URL of WCS server
- SurfaceViewRenderer, which will be used to display video from the camera
- SurfaceViewRenderer, which will be used to play video from the other party

```
SessionOptions sessionOptions = new SessionOptions(mWcsUrlView.getText().toString());
sessionOptions.setLocalRenderer(localRender);
sessionOptions.setRemoteRenderer(remoteRender);
session = Flashphoner.createSession(sessionOptions);
```

2. Outgoing call.

Session.createCall(), Call.call()code

CallOptions object with the following parameters is passed to createCall() method:

- callee SIP username
- video constraints

```
case CALL_REQUEST_CODE: {
    if (grantResults.length == 0 ||
          grantResults[0] != PackageManager.PERMISSION_GRANTED ||
          grantResults[1] != PackageManager.PERMISSION_GRANTED ) {
        Log.i(TAG, "Permission has been denied by user");
    } else {
        mCallButton.setEnabled(false);
        /**
          * Get call options from the callee text field
          */
        CallOptions callOptions = new CallOptions(mCalleeView.getText().toString());
        callOptions.getConstraints().updateVideo(true);
        call = session.createCall(callOptions);
        call.on(callStatusEvent);
        /**
          * Make a new outgoing call
          */
        call.call();
        Log.i(TAG, "Permission has been granted by user");
    }
    break;
}
```

3. Answering incoming call.

Call.answer()code

```
case INCOMING_CALL_REQUEST_CODE: {
    if (grantResults.length == 0 ||
           grantResults[0] != PackageManager.PERMISSION_GRANTED ||
           grantResults[1] != PackageManager.PERMISSION_GRANTED ) {
        call.hangup();
        incomingCallAlert = null;
        Log.i(TAG, "Permission has been denied by user");
    } else {
        mCallButton.setText(R.string.action_hangup);
        mCallButton.setTag(R.string.action_hangup);
        mCallButton.setEnabled(true);
        mCallStatus.setText(call.getStatus());
        call.getCallOptions().getConstraints().updateVideo(true);
        call.getCallObject().setHasVideo(true);
        call.answer();
        incomingCallAlert = null;
        Log.i(TAG, "Permission has been granted by user");
    }
}
```

4. Mute/unmute audio and video.

Call.unmuteAudio(), Call.muteAudio(), Call.unmuteVideo(), Call.muteVideo()code

```
mMuteAudio = (Switch) findViewById(R.id.mute_audio);
/**
  * Mute or Unmute audio for the SIP call
  * Mute if it is not muted.
  * Unmute if it is muted.
  */
mMuteAudio.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (call != null) {
            if (isChecked) {
                call.muteAudio();
            } else {
                call.unmuteAudio();
            }
        }
    }
});
mMuteVideo = (Switch) findViewById(R.id.mute_video);
/**
  * Mute or Unmute video for the SIP call
  * Mute if it is not muted.
  * Unmute if it is muted.
  */
mMuteVideo.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (call != null) {
            if (isChecked) {
                 call.muteVideo();
            } else {
                 call.unmuteVideo();
            }
        }
    }
});
```