SOFTWARE DESIGN

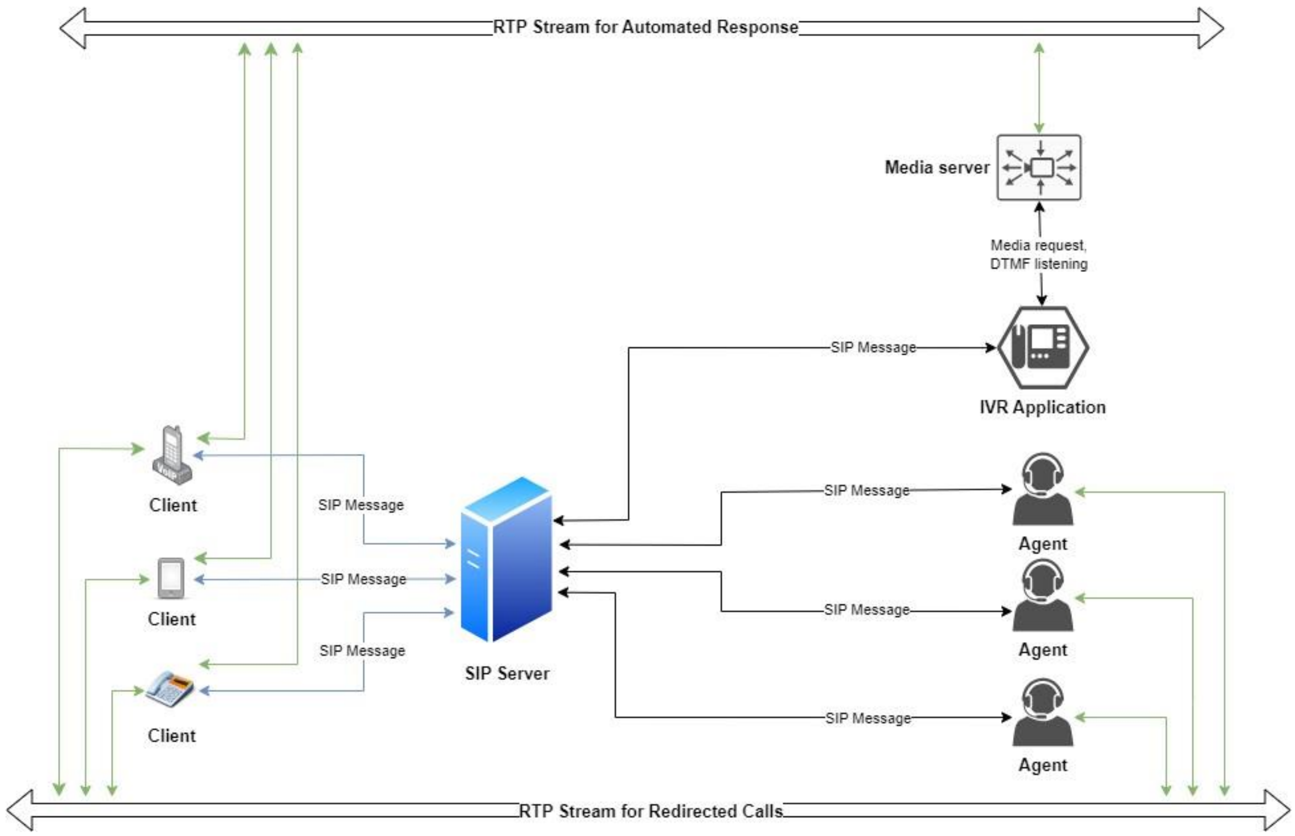
IVR System

Version 1.1

MARUSYS vina

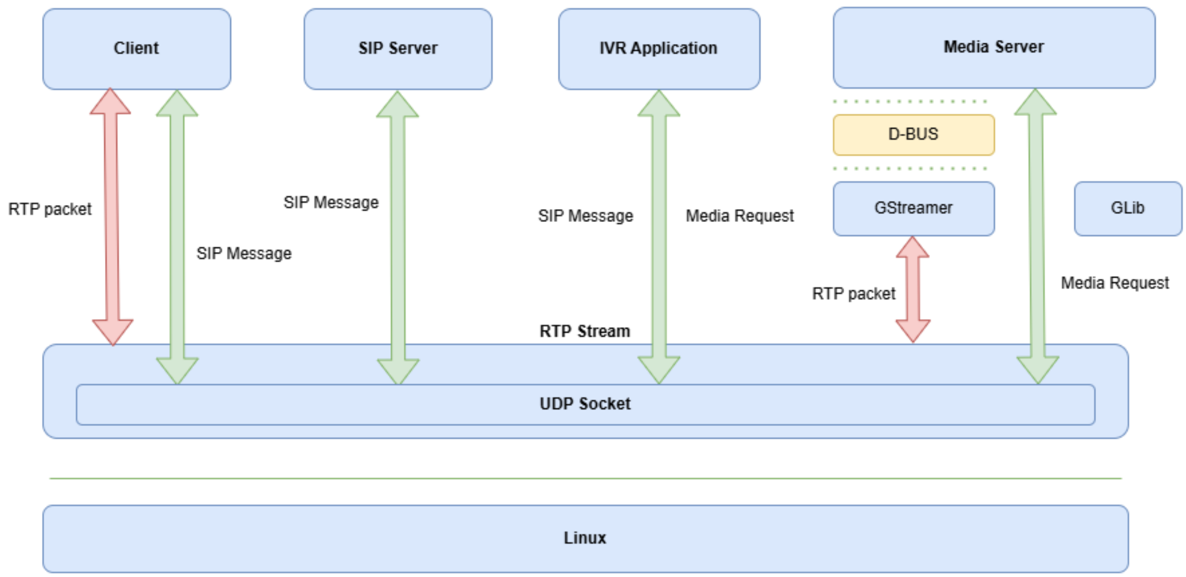
Table of Contents

1. System concepts



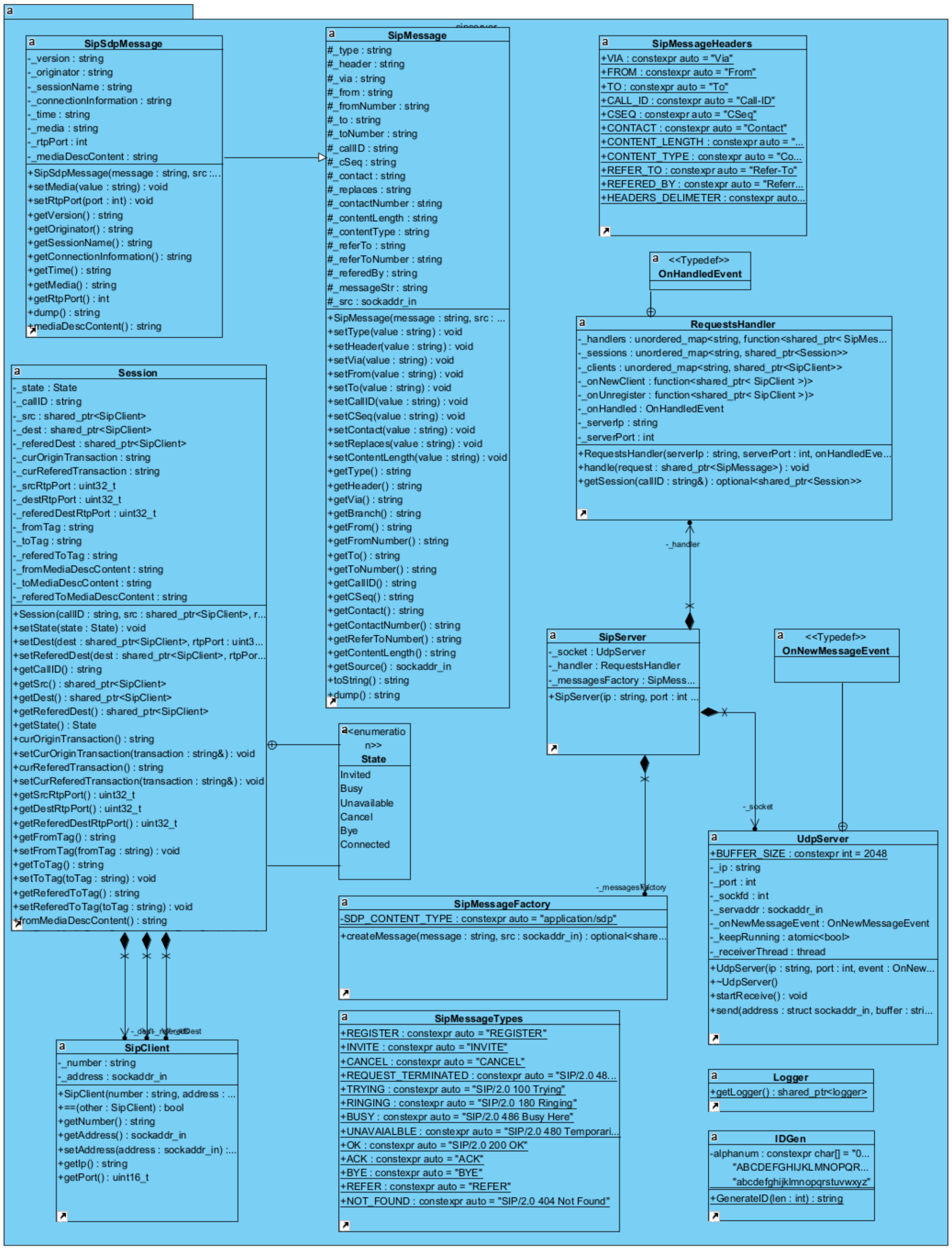
2. Architect overview

2.1. System context diagram



2.2. Class diagram

2.2.1. Sip Server

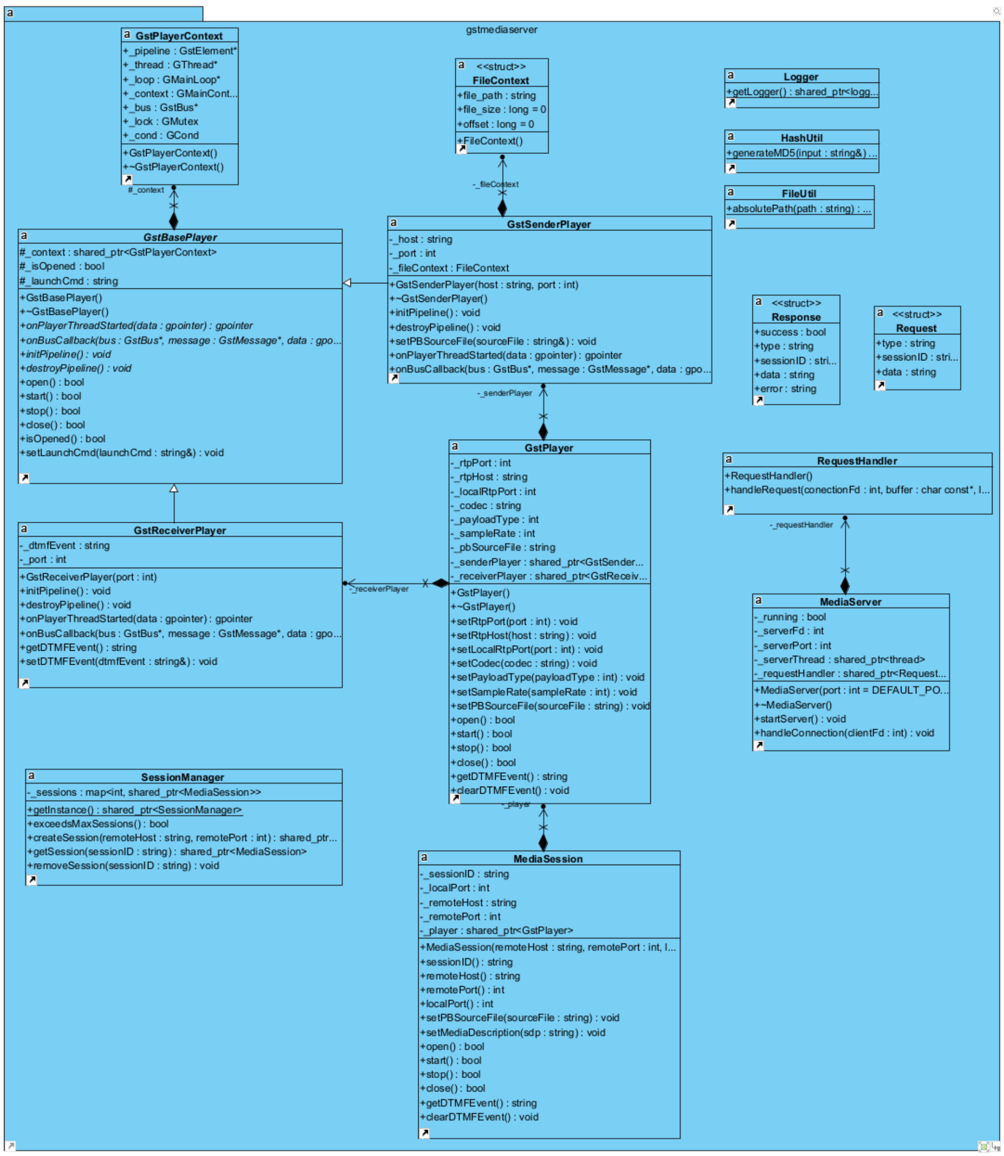


Class roles:

- SipServer: create the socket to send/recv the UDP messages and message handler to process coming message.
- SipMessage: define SIP message structure.
- SipSdpMessage: define sdp message structure.

- SipMessageFactory: create SIP/SDP message base on a string message.
- Sessions: storage information of session's related objects, such as: sockaddr, tag, rtp port …
- SipClient: storage information of the SIP client: account, address.
- RequestsHandler: process all incoming message: REGISTER, INVITE, REFER, BYR, 200 OK …

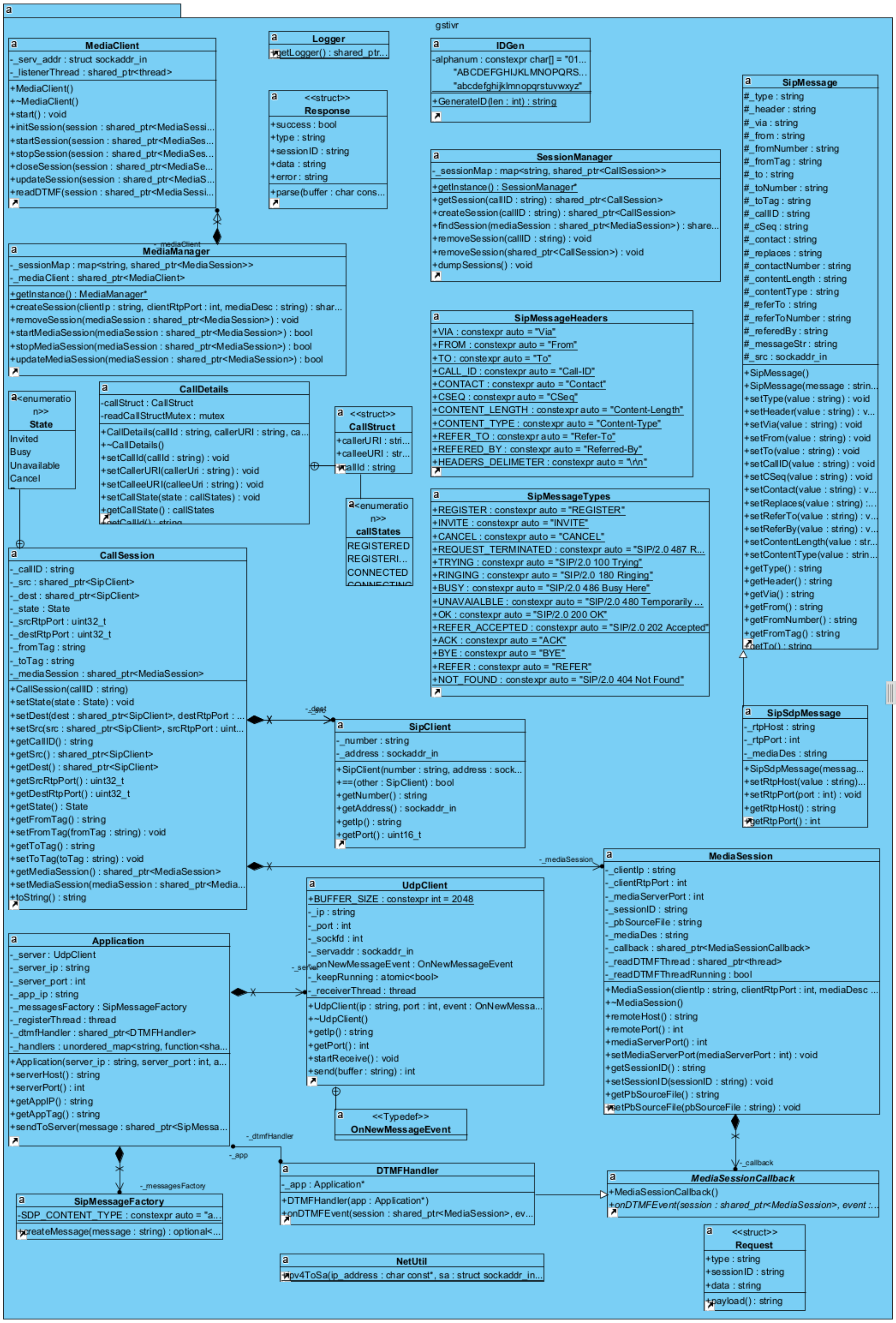
2.2.2. Media Server



Class roles:

- MediaServer: create a thread to receive request from IVR application via socket.
- SessionManager: manage the sessions: initialize, start, stop, close.
- MediaSession: storage information of session: client RTP info, Sending Player for playback audio file via RTP, Receiving Player for detecting DTMF event.
- RequestHandler: Processing the request from client: initialize, start, update, stop, close the session.
- GstBasePlayer: a base player which will create a worker thread for pipeline work: looper, callback receiver.
- GstReceiverPlayer: build a gstreamer pipeline from rtpbin, udpsrc and some other elements, for receiving RTP packages from client, parsing DTMF event.
- GstSenderPlayer: build a gstreamer pipline from filesrc/appsrc and some other elements, for playback via RTP a requested audio file .
- GstPlayer: a player which contains both of Sender Player and Receiver Player, it looks a wrapper for both.
- GstPlayerContext: It's a model class which contains player's properties: thread, context, mainloop, bus, lock, pipeline …

2.2.3. IVR Application

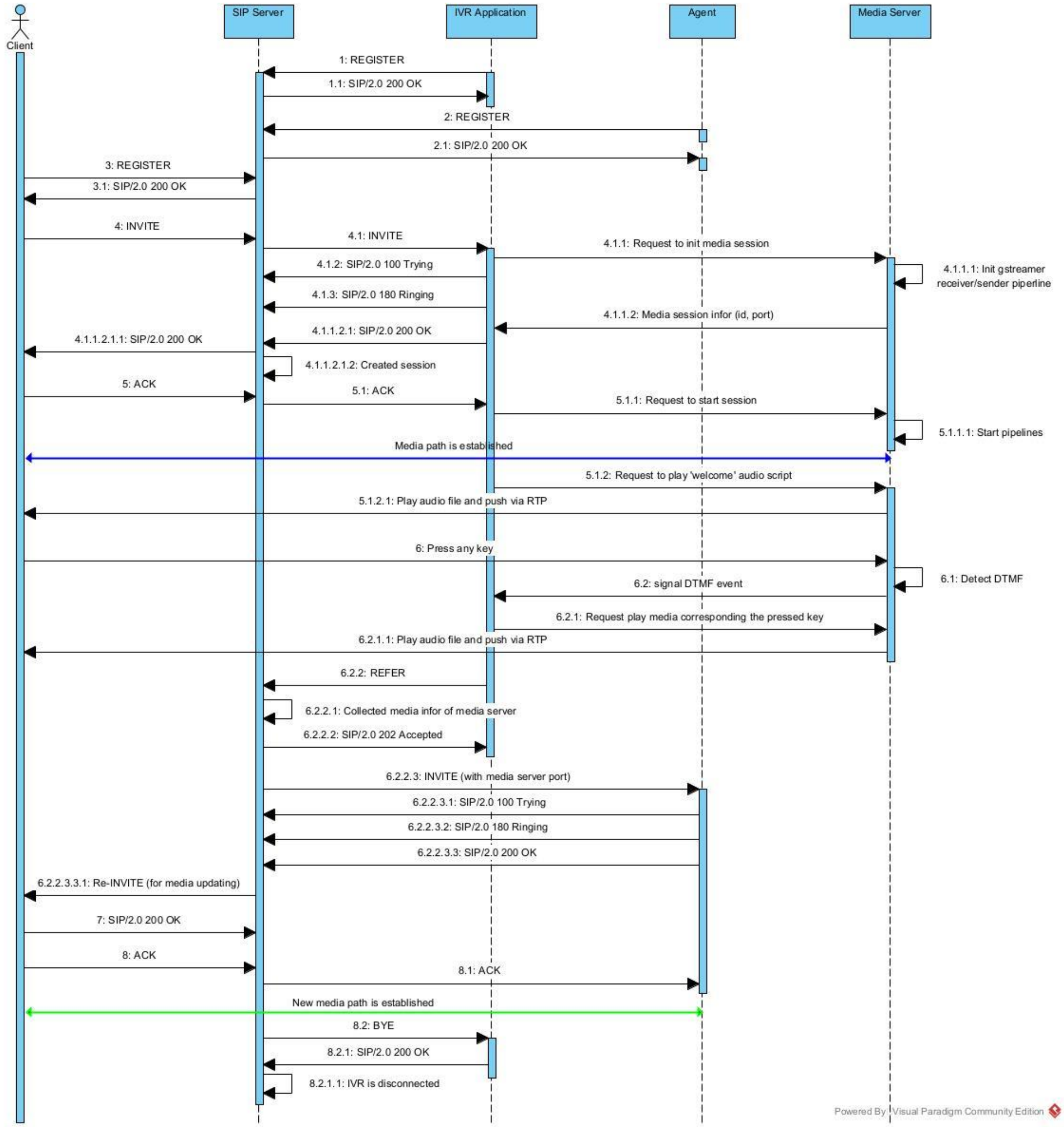


Class roles:

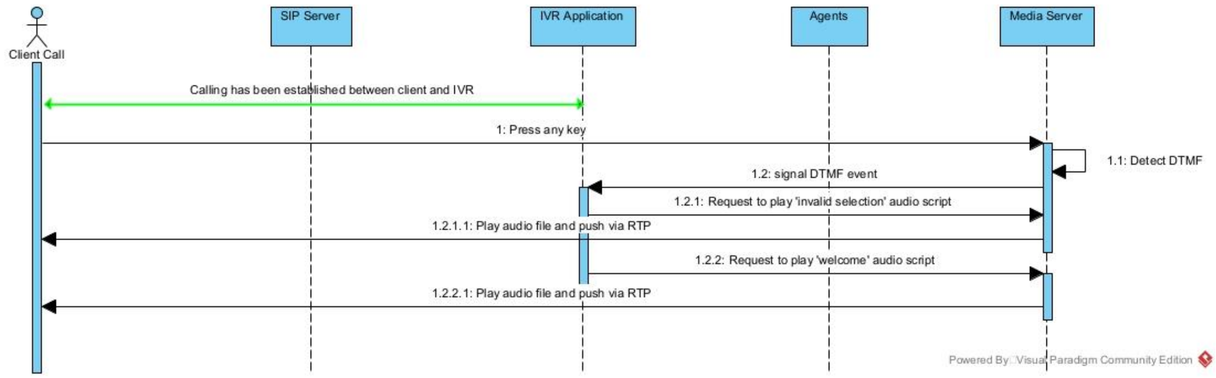
- Application: mainly controlling all requests from SIP server and define corresponding behavior.
- SipMessage: define SIP message structure.
- SipSdpMessage: define sdp message structure.
- SipMessageFactory: create SIP/SDP message base on a string message.
- UDPClient: Send/Receive the SIP message to/from SIP server
- SipClient: storage information of the SIP client: account, address.
- DTMFHandler: process DTMF event
- SessionManager: manager all calling sessions
- CallSession: storage information of callee
- MediaSessionManager: each calling session keeps one media session, and all media session with be managed by this class.
- MediaSession: keep media session information and call back to DTMF handler when the event is fired.
- MediaClient: it's responsible to communicate with media server

2.3. Sequence diagram

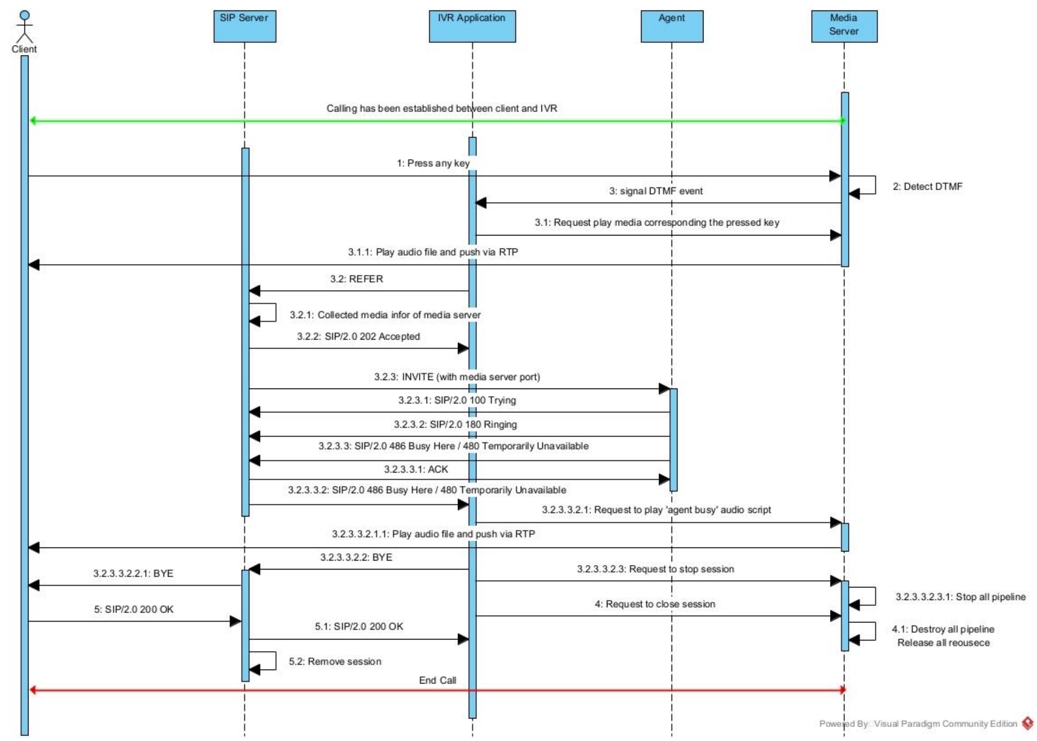
2.3.1. Normal Call



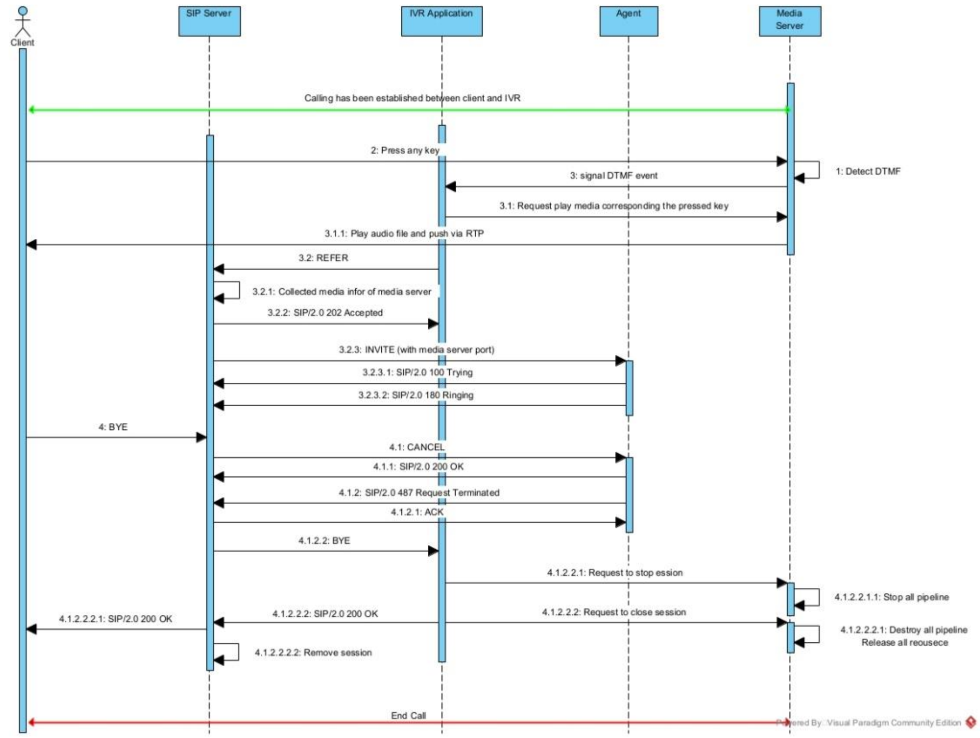
2.3.2. Client pressed invalid key



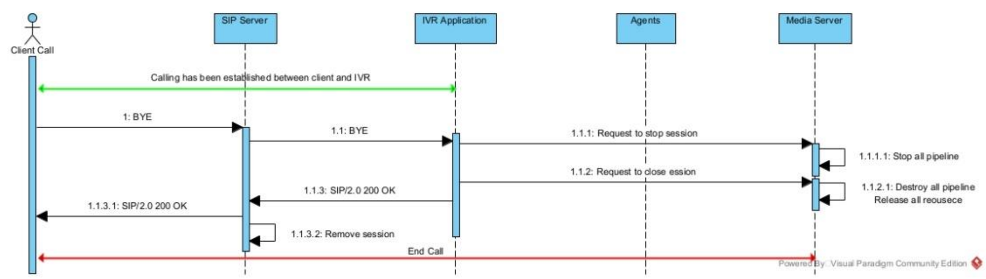
2.3.3. Agent is temporarily unavailable

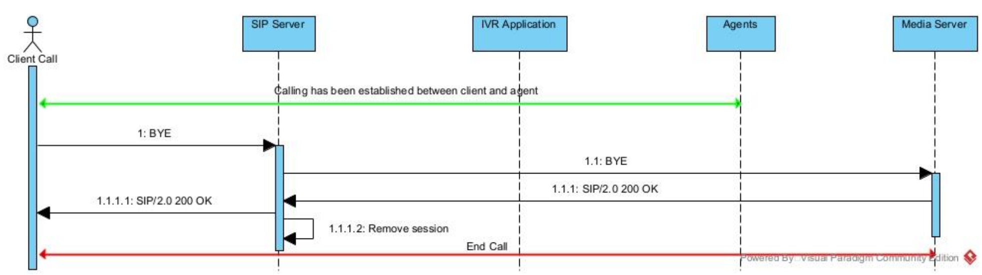


2.3.4. Client cancel call



2.3.5. Client hangs up call





3. Constraints

- At present, this project only support the softphones which have media codec:
  - speex/16000
  - speex/16000

- The project is only supported on Linux.

4. Testing

This project has been tested on Zoiper and Eyebeam already. With other softphones, we don't guarantee that it will work.

5. References

- Wiki: https://en.wikipedia.org/wiki/Session_Initiation_Protocol
- Gstreamer: https://gstreamer.freedesktop.org/documentation/tutorials/index.html?gi-language=c