Practical WebRTC with OpenSIPS

OpenSIPS Summit, Austin 2015

Who are you people?

- Eric Tamme
  - Principal Engineer
- OnSIP
  - Hosted PBX
  - Hosted SIP Platform
  - Developers of SIP.js

See: sipjs.com, or https://github.com/onsip/sip.js

Federated SIP + KwikyKonf

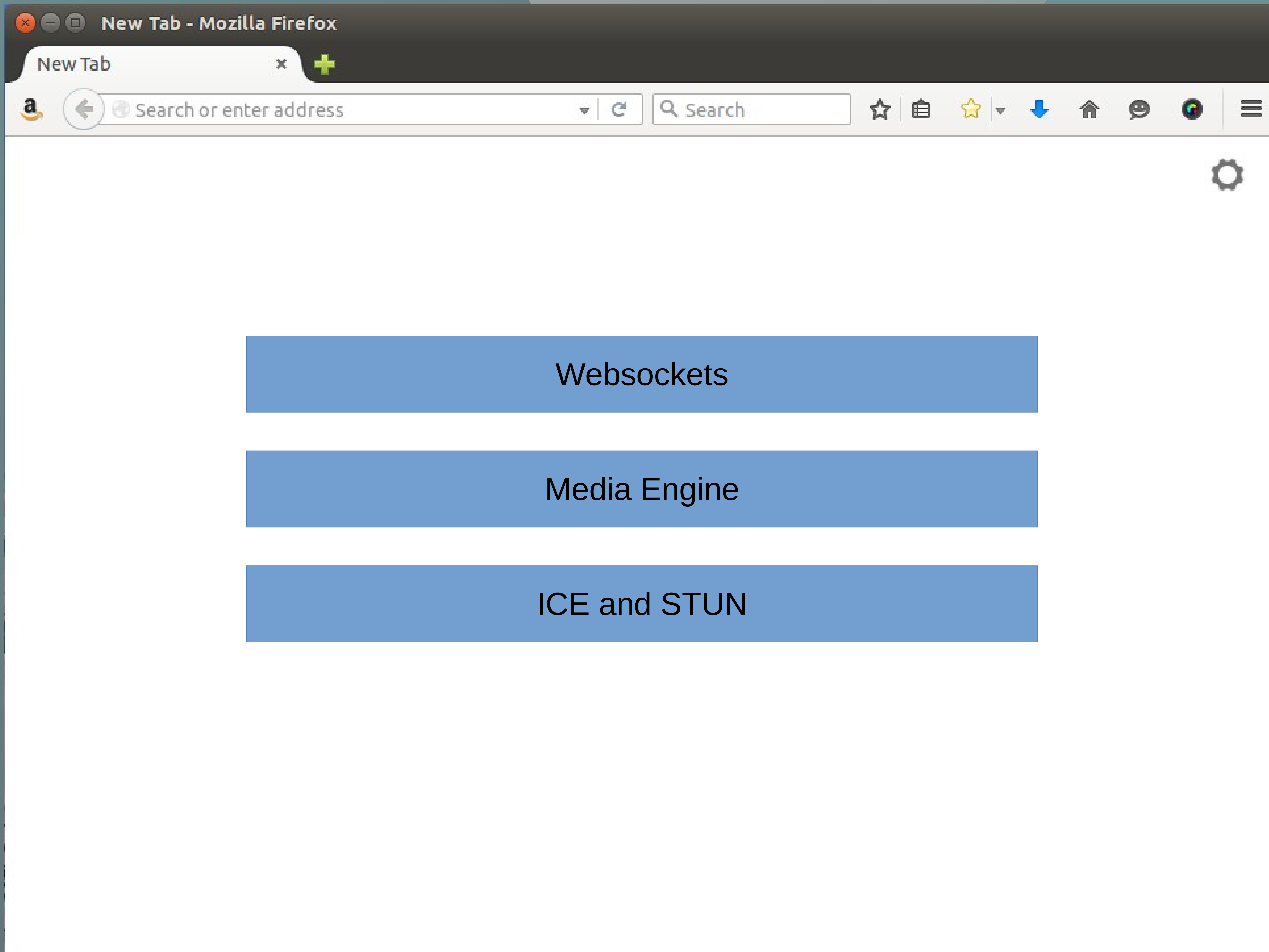
- What is WebRTC, and how does OpenSIPS handle it?
- Build a SIP registrar and proxy server that can handle WebRTC signaling.
- Integrate RTPEngine to provide WebRTC interoperation and media relaying.
- Use SIP.js to build a multi-party WebRTC video chat.

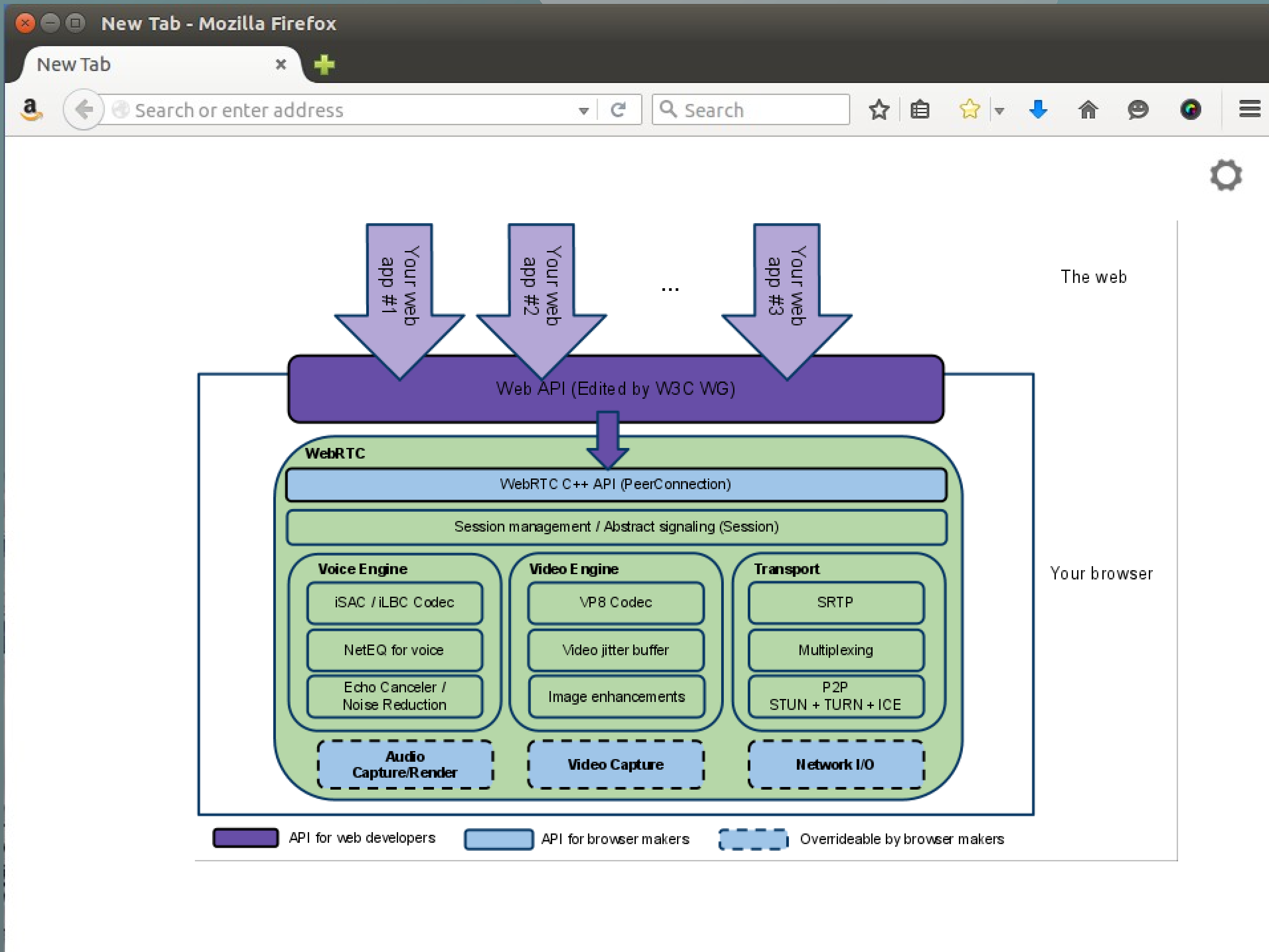
What is WebRTC?

Browser based support for

- WebSockets (WS)
- WebSocket Secure (WSS)

- Media engine capable of generating SDP
- DTLS based key exchange for SRTP
- Modern ICE and STUN client


New Tab - Mozilla Firefox
New Tab
Search or enter address
Search
Websockets
Media Engine
ICE and STUN


New Tab - Mozilla Firefox
New Tab
Search or enter address
Search
Your web app #1
Your web app #2
...
Your web app #3
The web
Web API (Edited by W3C WG)
WebRTC
WebRTC C++ API (PeerConnection)
Session management / Abstract signaling (Session)
Voice Engine
iSAC / iLBC Codec
NetEQ for voice
Echo Canceler / Noise Reduction
Video Engine
VP8 Codec
Video jitter buffer
Image enhancements
Transport
SRTP
Multiplexing
P2P STUN + TURN + ICE
Audio Capture/Render
Video Capture
Network I/O
Your browser
API for web developers
API for browser makers
Overrideable by browser makers

WebRTC is ***not*** SIP

No signaling protocol is specified, but SIP is a great tool for the job.

Steps to creating a WebRTC session

- Request a media description from the browser that can be used for an offer.
- Transmit this offer to another browser.
- Accept the answer from the other browser.
- Begin the process of of ICE/STUN negotiation.
- Do DTLS key exchange

Where does OpenSIPS fit in with WebRTC?

- Facilitates signaling generally over WS.
- Provides user location for signaling between users.
- Integrates with RTPEngine for turn server with ICE/STUN support.

Github project links

https://github.com/etamme/federated-sip

https://github.com/etamme/kwikykonf

Federated-SIP Install

```
Create a clean Centos 7 or Debian 8 VM
on public IP (digital ocean)

[yum|apt-get install] -y git

cd /usr/local/src

git clone https://github.com/etamme/federated-sip.git

cd federated-sip

scripts/install.sh

Just hit enter for domain and user
```

Step 1. Build a registrar

- Detect and track user agent capabilities with branch flags

- Allow people to register without authentication so we can generate AOR's on the fly

- Lines 329 and 370 of federated core config.

Step 2. Integrate RTPEngine

- Use known attributes of clients to facilitate interop

- Understand the offer answer models and track required attributes transactionally to handle various scenarios.

- branch_route[rtpengine] and onreply_route

DTLS-SRTP and RTP interop graph

OFFER	WS	NOT WS
WS	FORCE RELAY	RTP/AVP
NOT WS	RTP/SAVPF	RTP/AVP

ANSWER	WS	NOT WS
WS	FORCE RELAY	RTP/AVP
NOT WS	RTP/SAVPF	RTP/AVP

Building RTPEngine offer/answer flags

```
# set rtpengine flags based on whether uac or uas are websockets

if (isflagset(uac_ws) && isbflagset(uas_ws)) {

  $var(rtpengine_flags) = "ICE=force-relay DTLS=passive";
  xlog("L_INFO","$var(prefix) uac and uas are both websockets\n");

} else if (isflagset(uac_ws) && !isbflagset(uas_ws)) {

  $var(rtpengine_flags) = "RTP/AVP replace-session-connection replace-origin ICE=remove";
  xlog("L_INFO","$var(prefix) uac is ws uas is not\n");

} else if (!isflagset(uac_ws) && isbflagset(uas_ws)) {

  $var(rtpengine_flags) = "UDP/TLS/RTP/SAVPF ICE=force";
  xlog("L_INFO","$var(prefix) uas is ws uac is not\n");

} else if (!isflagset(uac_ws) && !isbflagset(uas_ws)) {

  $var(rtpengine_flags) = "RTP/AVP replace-session-connection replace-origin ICE=remove";
  xlog("L_INFO","$var(prefix) neither uac or uas are websocket\n");

}
```

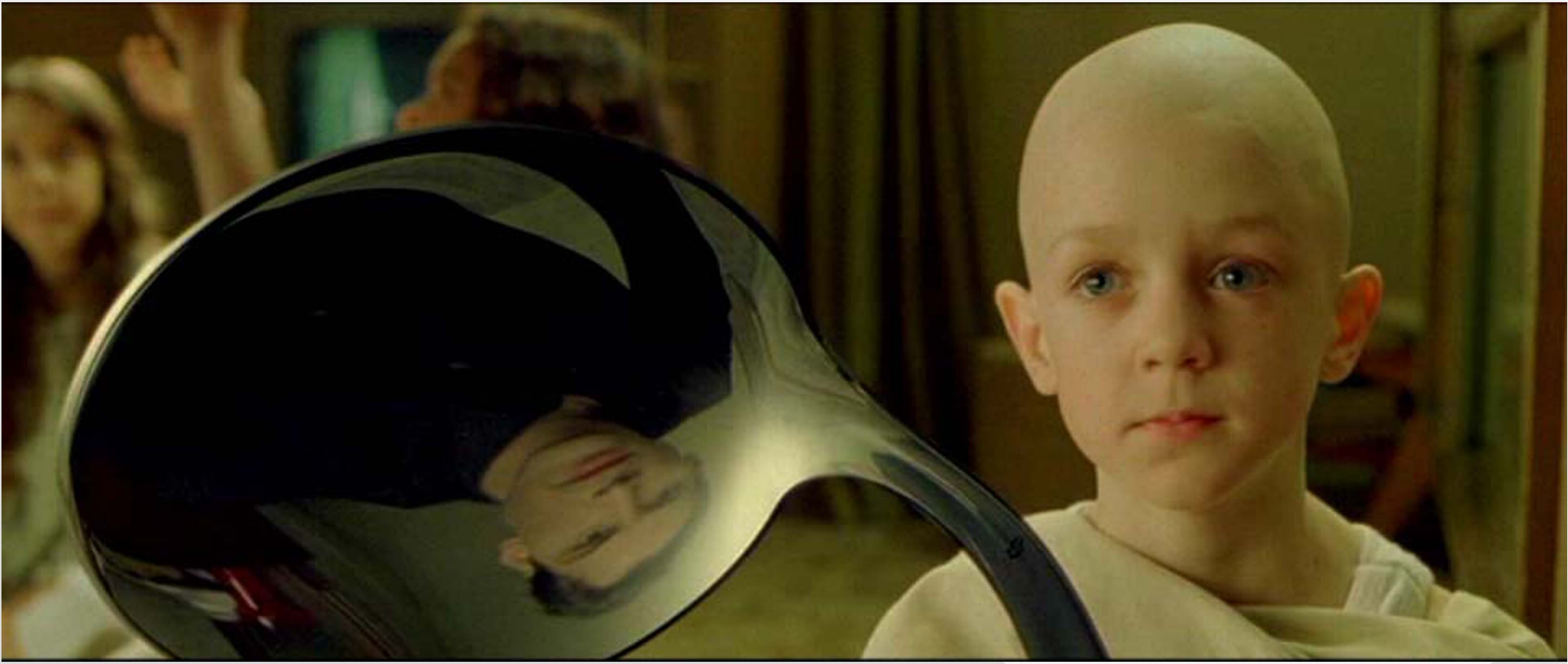
Review so far

- ✓ What is WebRTC, and how does OpenSIPS handle it?
- ✓ Build a SIP registrar and proxy server that can handle WebRTC signaling.
- ✓ Integrate RTPEngine to provide WebRTC interoperation and media relaying.
- Use SIP.js to build a multi-party WebRTC video chat.

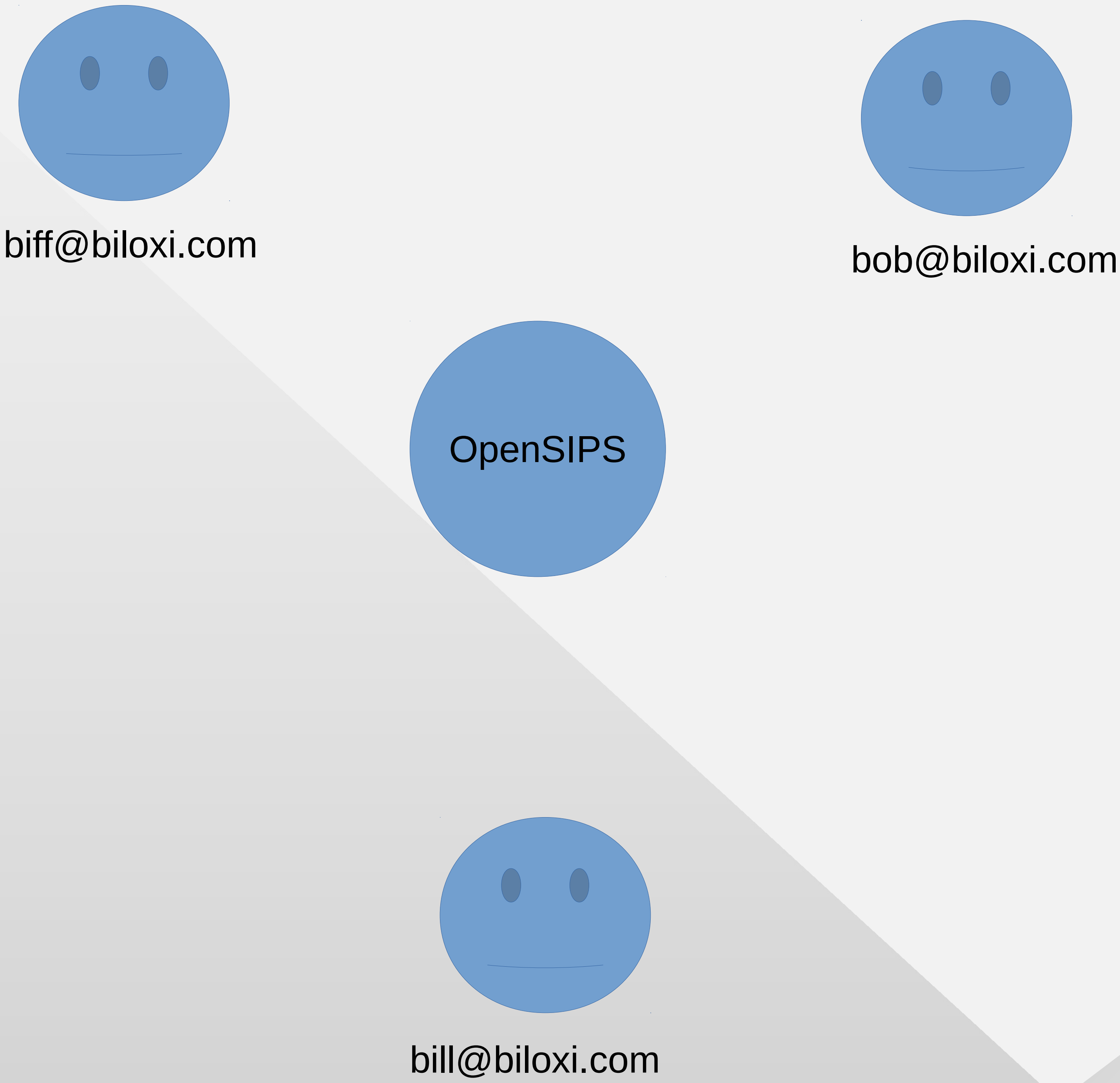
Questions

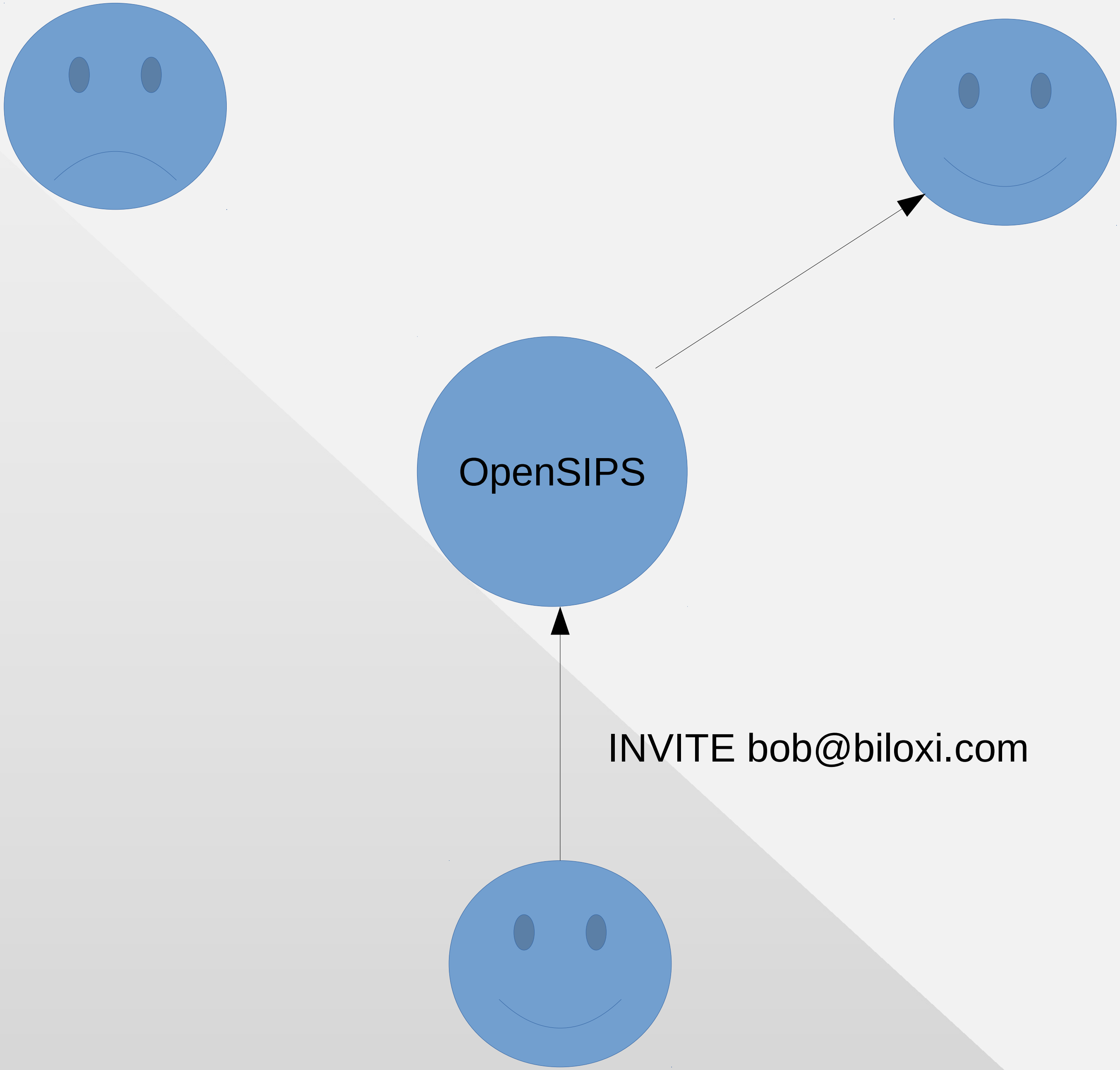
Next up...

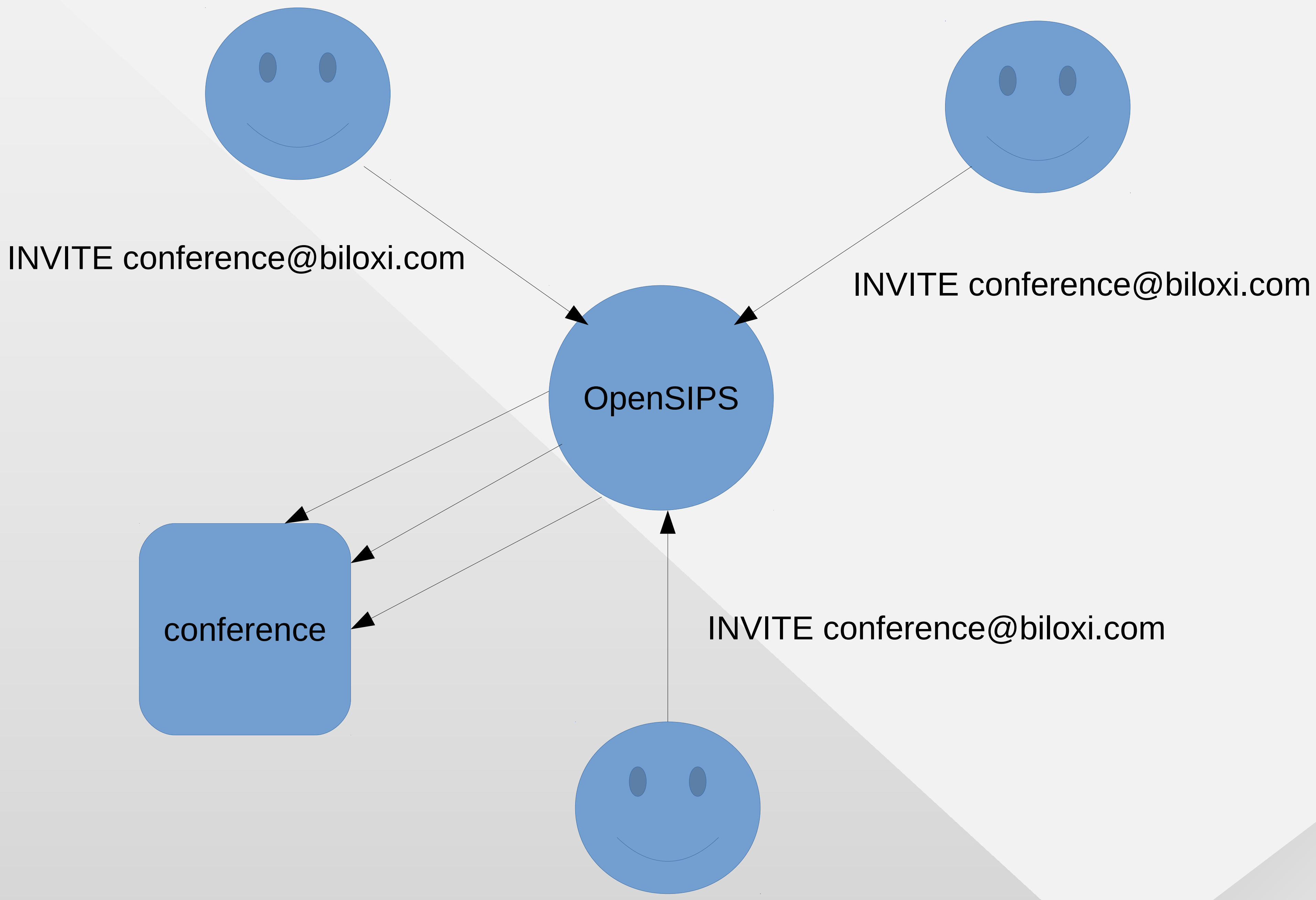
Build a multiparty video chat with SIP.js, OpenSIPS, and RTPEngine

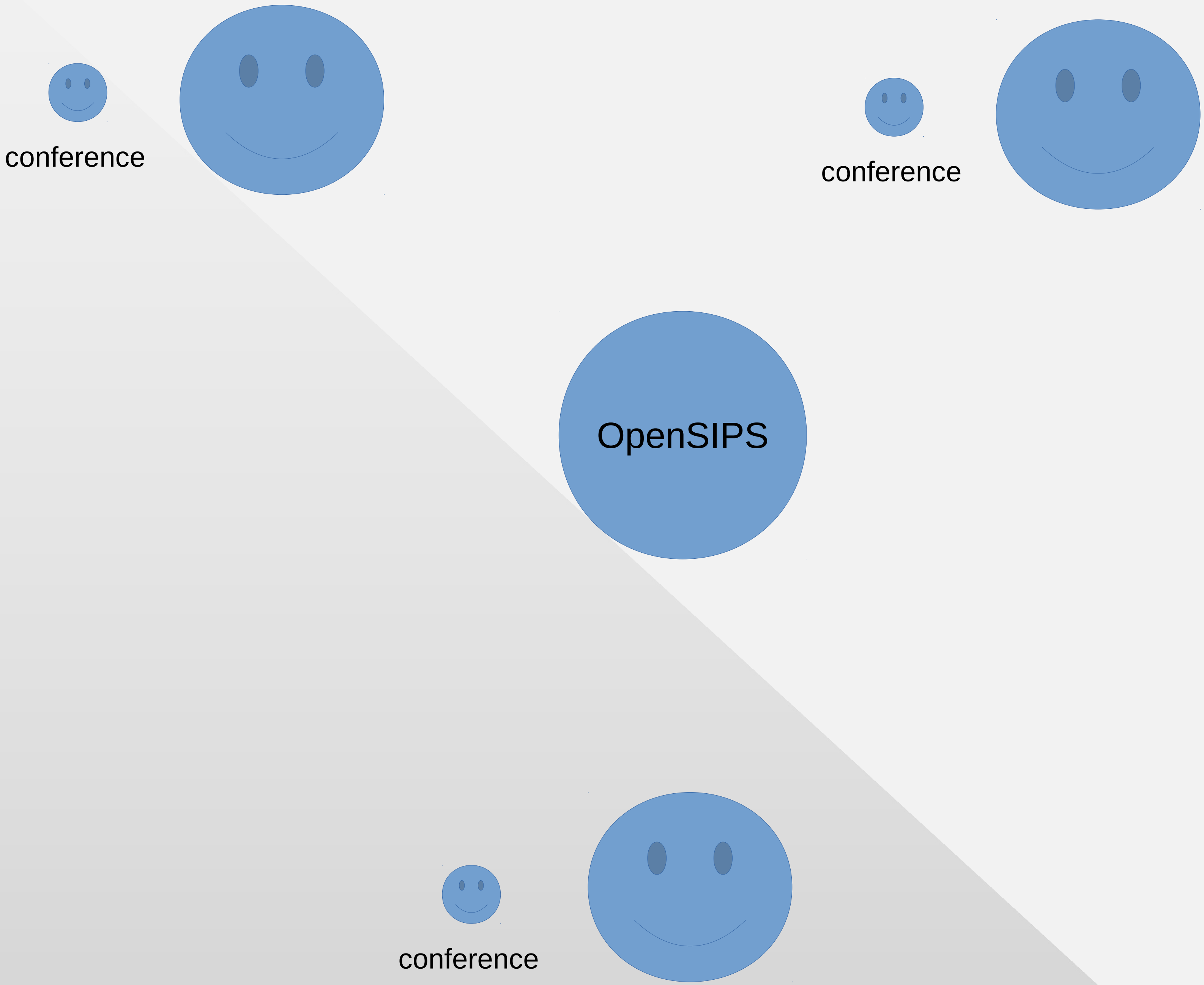


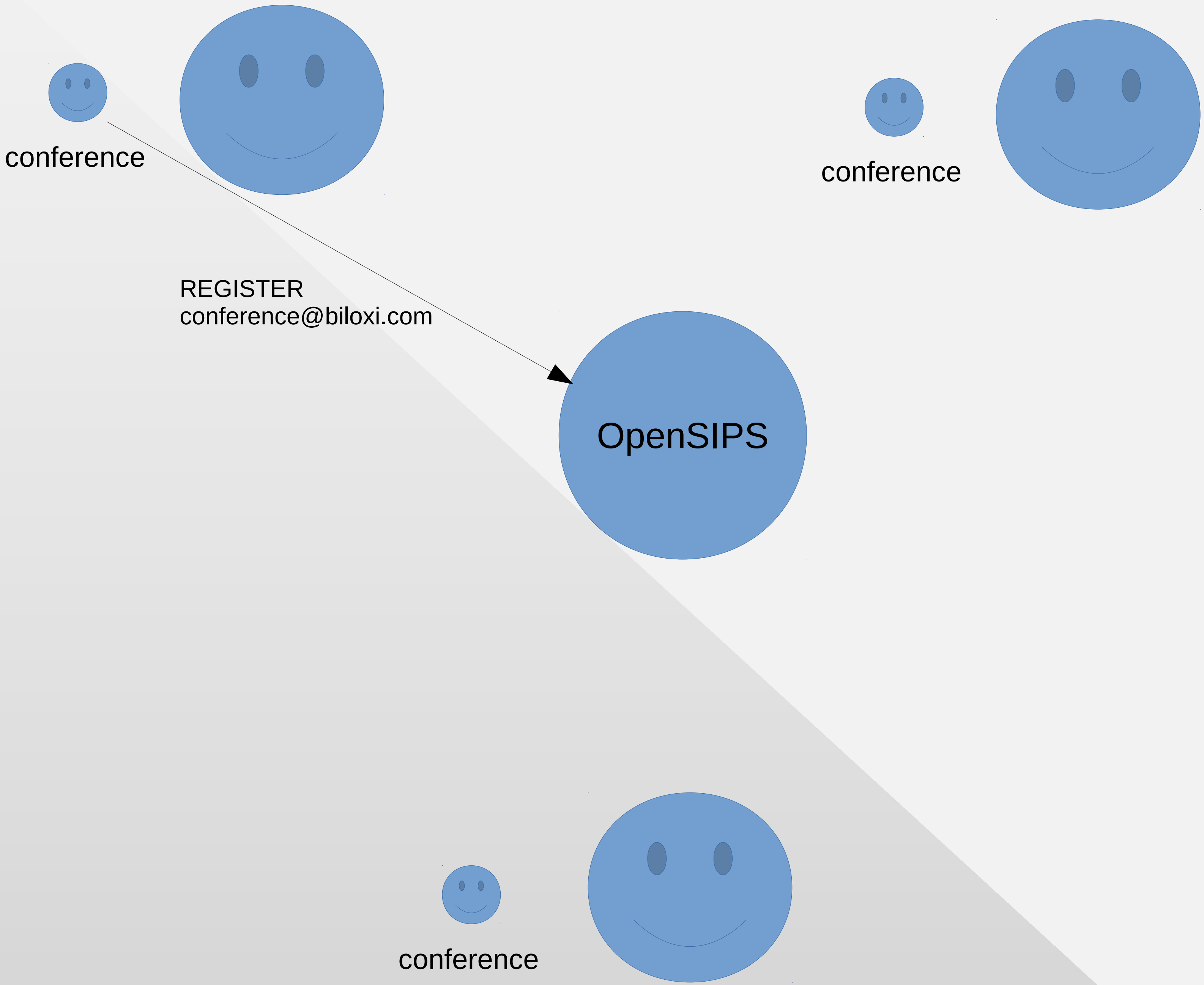
There is no spoon

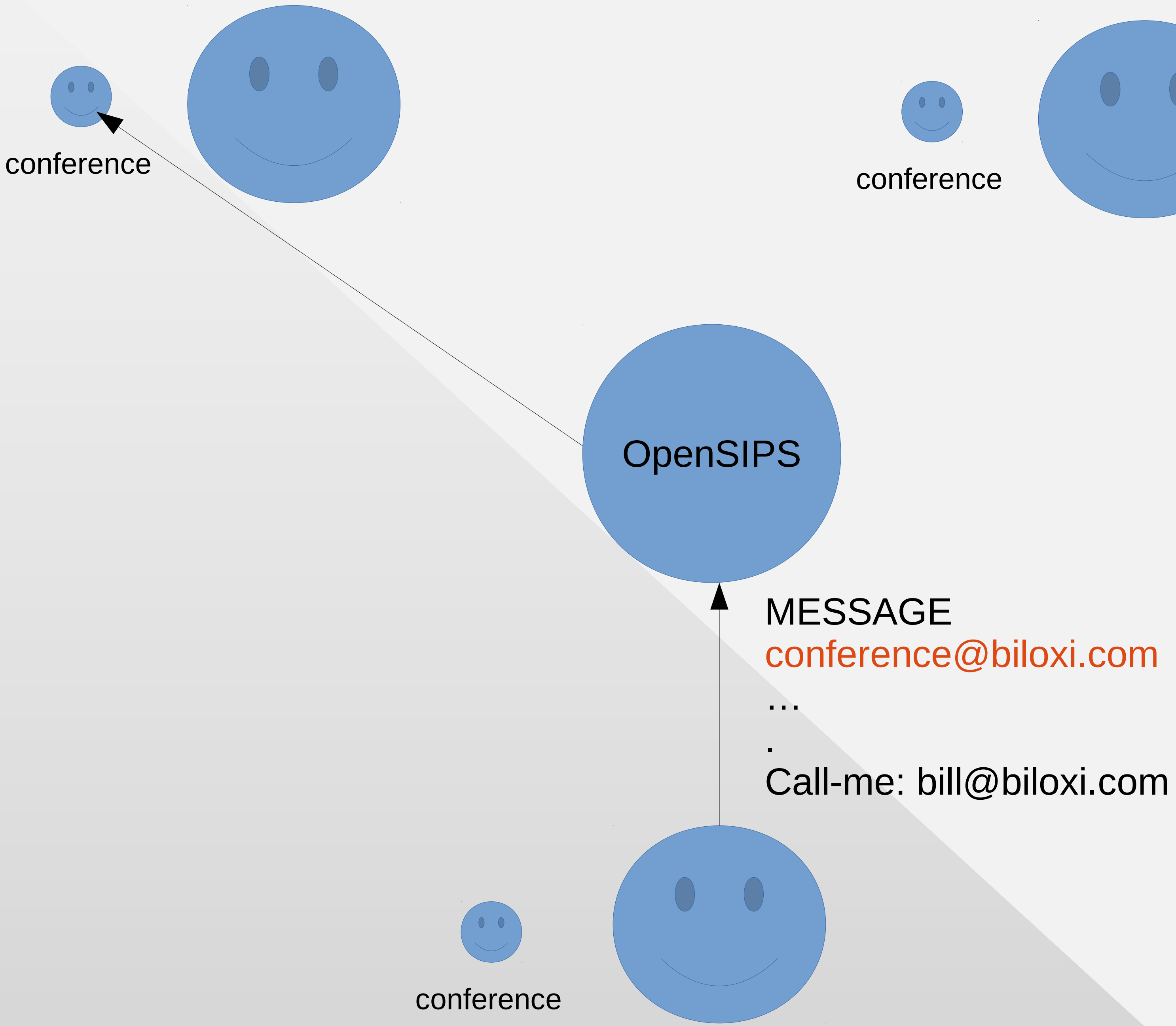

biff@biloxi.com
bob@biloxi.com
OpenSIPS
bill@biloxi.com

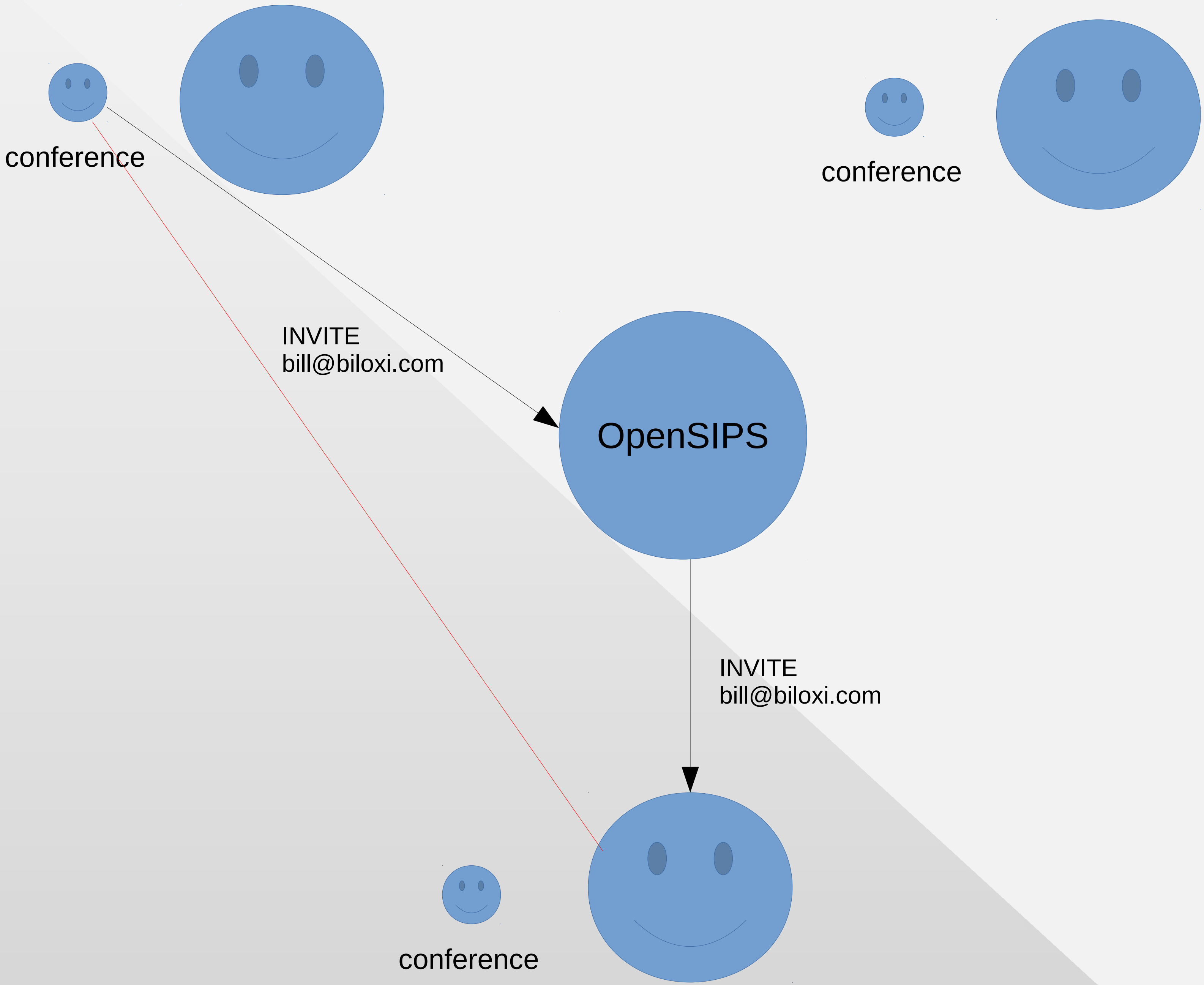

OpenSIPS
INVITE bob@biloxi.com

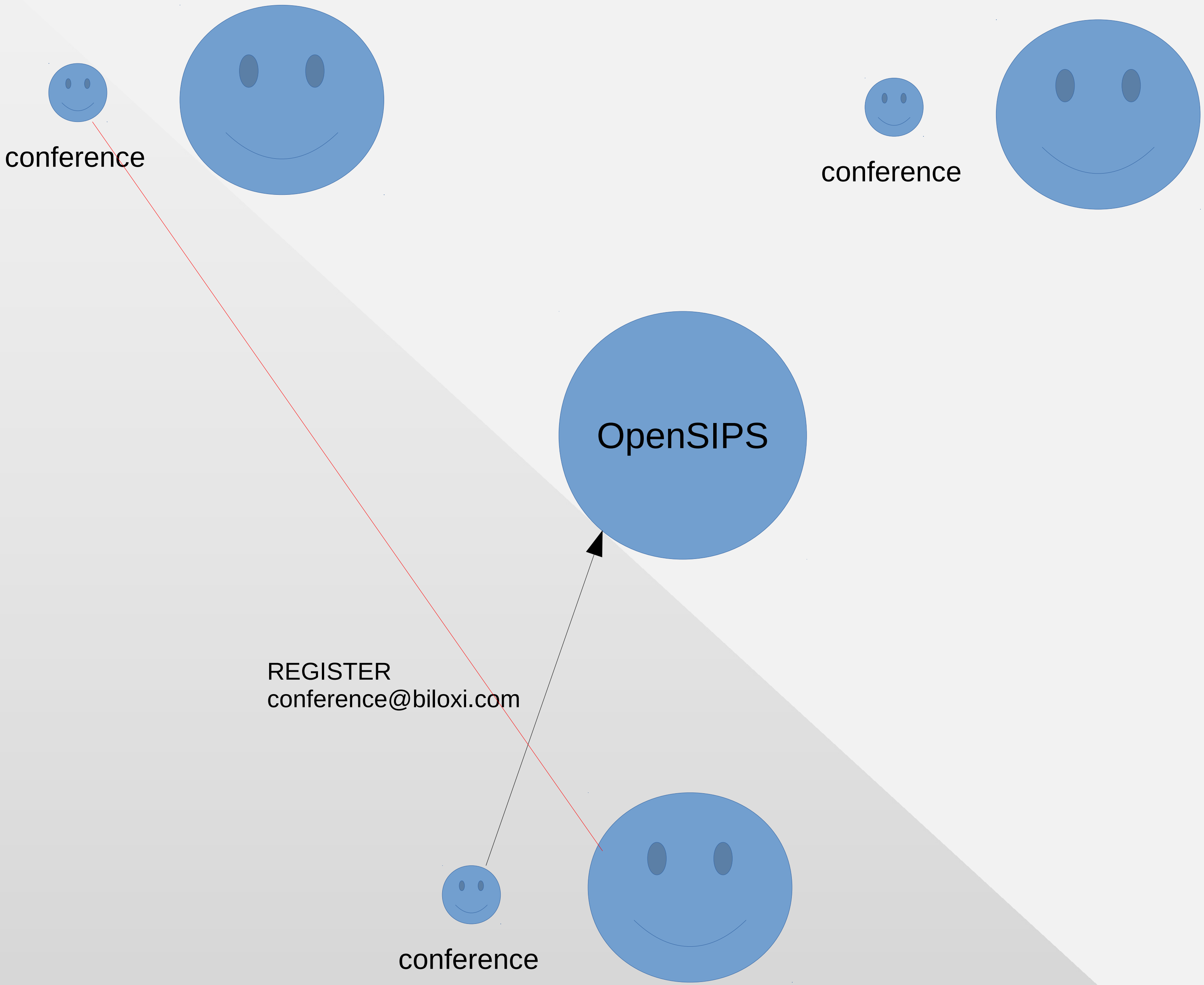

INVITE conference@biloxi.com
INVITE conference@biloxi.com
OpenSIPS
conference
INVITE conference@biloxi.com

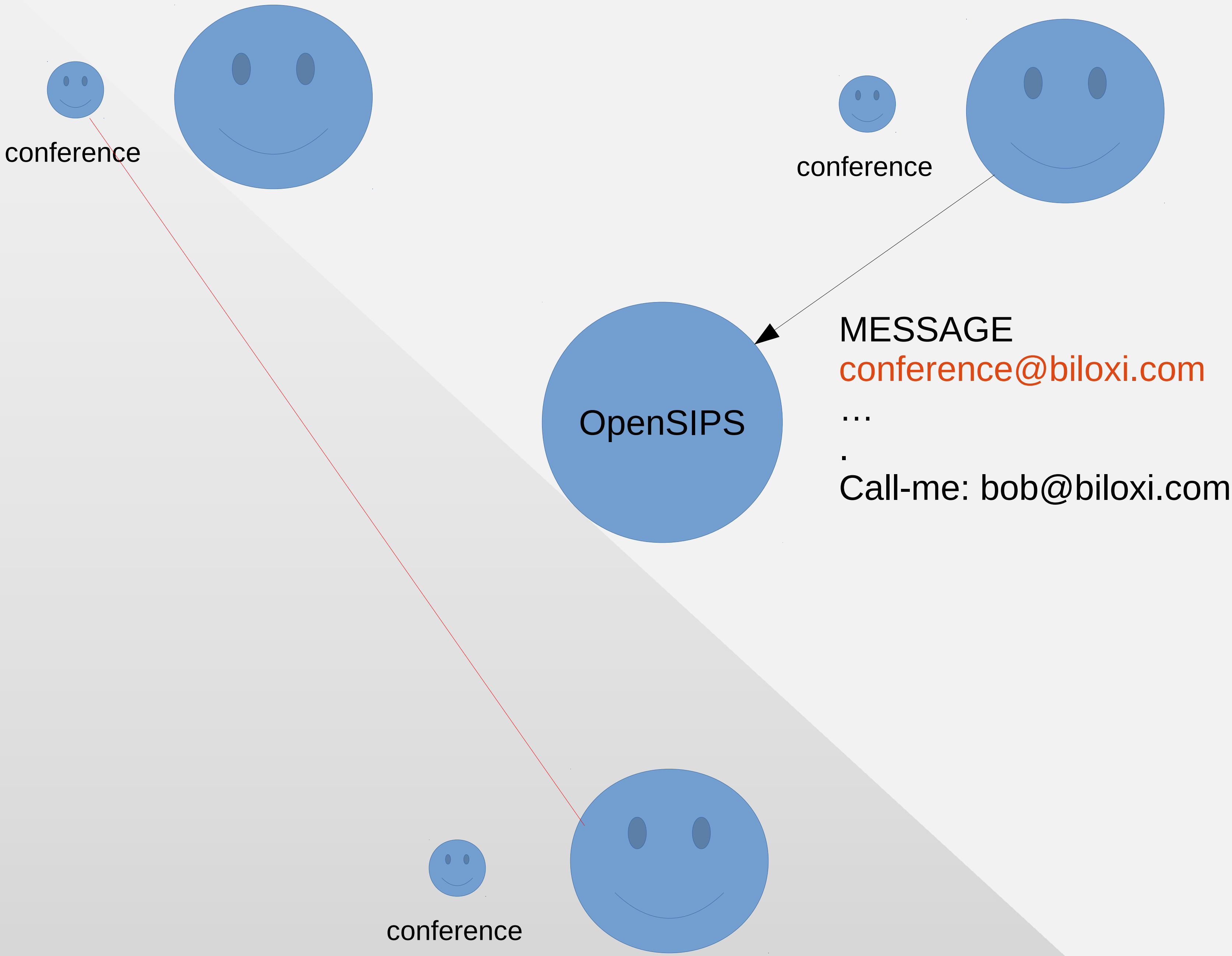

conference
conference
OpenSIPS
conference

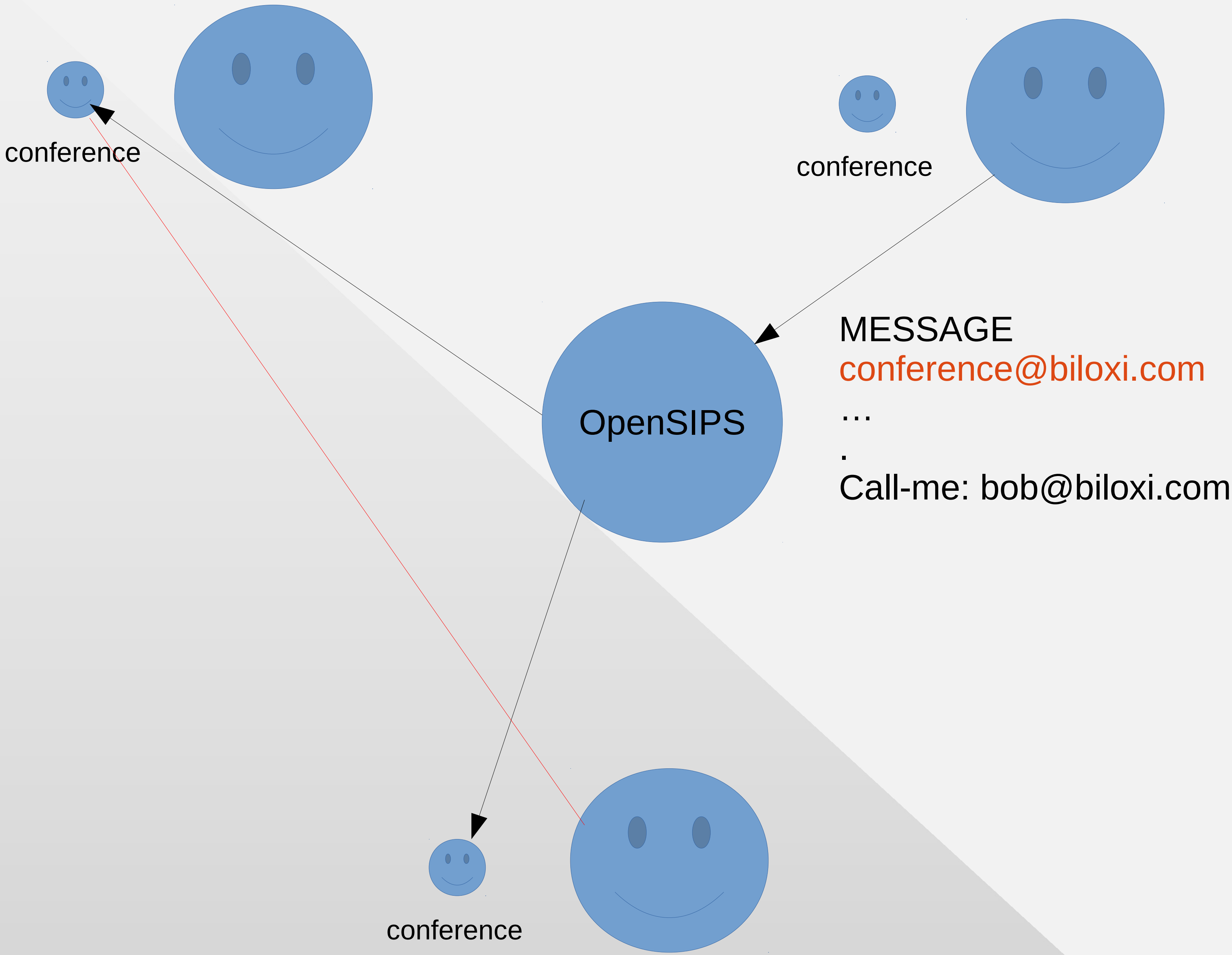

conference
conference
REGISTER
conference@biloxi.com
OpenSIPS
conference


conference
conference
OpenSIPS
MESSAGE
conference@biloxi.com
…
.
Call-me: bill@biloxi.com
conference

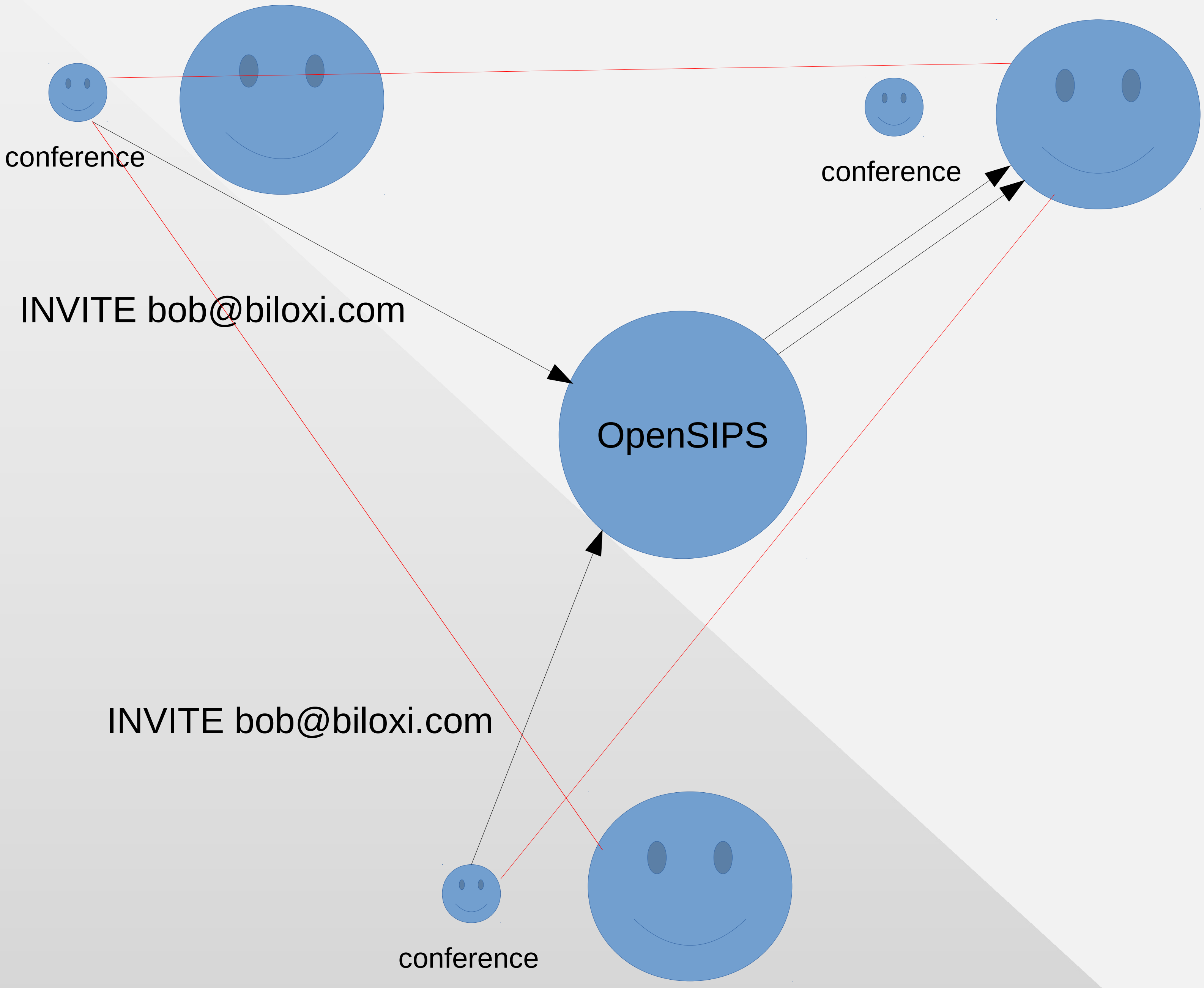

conference
conference
INVITE
bill@biloxi.com
OpenSIPS
INVITE
bill@biloxi.com
conference


conference
conference
OpenSIPS
REGISTER
conference@biloxi.com
conference


conference
conference
MESSAGE
conference@biloxi.com
…
.
Call-me: bob@biloxi.com
OpenSIPS
conference



conference
conference
OpenSIPS
MESSAGE
conference@biloxi.com
…
.
Call-me: bob@biloxi.com
conference


conference
conference
INVITE bob@biloxi.com
OpenSIPS
INVITE bob@biloxi.com
conference

KwikyKonf Code walkthrough

- Create our private UA
  - Send MESSAGE request to the shared AOR
  - Register callbacks to handle adding and removing streams
- Create our shared UA
  - REGISTER the shared AOR
  - Register callbacks to handle adding and removing streams

Trial by fire

Questions?

SIP/Email: eric.tamme@onsip.com

onsip®