

Anatomy of GNOME Calls

Evangelos Ribeiro Tzaras

2022-02-06

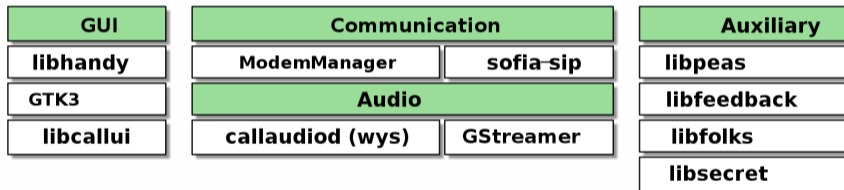
- GNU/Linux user > 10 years
- FOSS enthusiast
- Early PinePhone adopter (Braveheart edition)
- Mobian + Debian Maintainer (DebianOnMobile team)
- Freelance Free Software Developer
- Working with Purism on the Librem 5 phone
- Maintainer of GNOME Calls

What is GNOME Calls anyway?

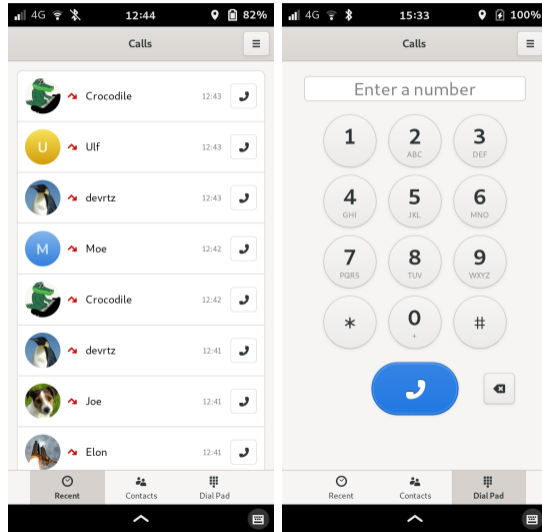
- Dialer application
- Written in C using GTK/GObject
- It does PSTN (regular telephony)
- It does SIP (VoIP)

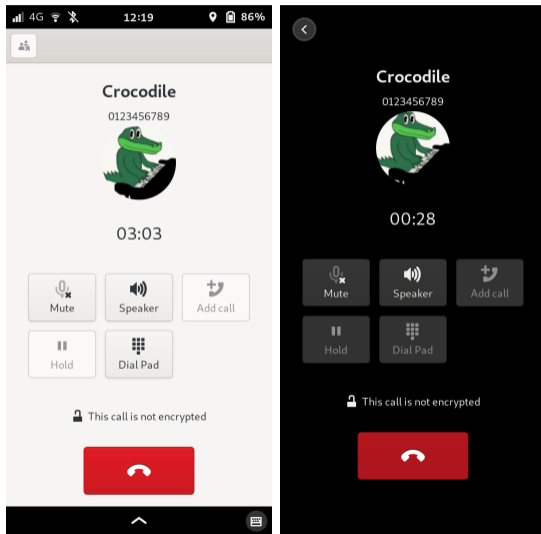
How does it work?

What is needed to place and receive calls?



How does it look?





What does it do?

- MM daemon talks to hardware and exposes objects on DBus
- libmm-glib convenient API to talk to MM daemon, GObject for voice capable modems, calls, etc
- Watch for `/org/freedesktop/ModemManager1/`
- Grab all exported objects (`org.freedesktop.DBus.ObjectManager`)
- Voice capable modems implement `org.freedesktop.ModemManager1.Voice`

API usage

- MModemVoice represents voice capable modem
- MCall represents a call
- `mm_modem_voice_create_call()` -> MCall
- `mm_call_start()`, `mm_call_accept()`, `mm_call_hangup()`, `mm_call_send_dtmf()` etc

callaudiod

- callaudiod talks with PulseAudio and exposes controls on DBus
- Switch audio profiles, enable speaker, mute microphone
- libcallaudio provides convenient API to talk with callaudiod
- `call_audio_select_mode()`, `call_audio_enable_speaker()`, `call_audio_mute_mic()`

wys

- uses loopback devices to set up audio routing between host and modem
- wys watches for new calls on DBus
- not needed for the PP because audio is routed in hardware

What does it do?

- SIP User agent
- SIP signalling

What does it do?

- SIP User agent
- SIP signalling

How do we use it?

- Applications provide callbacks for event types `nua_i_invite`, `nua_r_invite`, etc
- `sofia-sip-ua-glib` provides GSource for glib main loop integration

Registering with a server

```
nua_register(register_handle, NUTAG_M_USERNAME ("alice"),  
             NUTAG_REGISTRAR_URL("example.org"), TAG_END ());
```

Registering with a server

```
nua_register(register_handle, NUTAG_M_USERNAME ("alice"),  
             NUTAG_REGISTRAR_URL("example.org"), TAG_END ());
```

REGISTER request

```
REGISTER sip:example.org SIP/2.0  
From: <sip:alice@example.org>;tag=NrcZe62gcB8Fm  
To: <sip:alice@example.org>  
CSeq: 962983170 REGISTER
```

Placing a call

```
nua_invite(invite_handle, SIPTAG_TO_STR ("bob@example.org"),  
           SOATAG_USER_SDP_STR (sdp_string), TAG_END ());
```

sofia-sip: How to place a call?

Placing a call

```
nua_invite(invite_handle, SIPTAG_TO_STR ("bob@example.org"),  
           SOATAG_USER_SDP_STR (sdp_string), TAG_END ());
```

INVITE request

```
INVITE sip:bob@example.org SIP/2.0  
From: <sip:alice@example.org>;tag=58FHS0F6acF2p  
To: <sip:bob@example.org>  
Content-Type: application/sdp  
  
c=IN IP4 185.162.223.25  
m=audio 25128 RTP/AVP 8 0  
a=rtpmap:8 PCMA/8000  
a=rtpmap:0 PCMU/8000
```

What and how?

- Send and receive RTP media over the network
- Send pipeline: Payload encode audio data and send it over UDP socket

```
$ gst-launch pulsesrc ! alawenc ! rtpmap ! udpsrc host=.. port=..
```

- Receive pipeline: Extract received audio from RTP packets, decode and play them

```
$ gst-launch udpsink port=.. ! rtpmap ! alawdec ! pulsesink
```


Auxiliary libraries

libfolks

Contact integration

libfeedback

Provides audio (ringing), haptic (vibration motor) and visual feedback (LED)

libsecret

Used to store and retrieve VoIP credentials

libpeas

Plugin system (mm, sip, dummy, ofono)

Cellular

- Supplementary call services (call holding, waiting, call transfer, etc)
- Conference calls
- Voice mail

VoIP

- SRTP
- Video calls

Want to learn more?

Links

- Repository: <https://gitlab.gnome.org/GNOME/calls>
- Documentation: <https://gnome.pages.gitlab.gnome.org/calls>

Get in contact

- devrtz@fortysixandtwo.eu
- [@devrtz:fortysixandtwo.eu](https://matrix.to/#/!devrtz:fortysixandtwo.eu) (Matrix)
- [@evangelos.tzaras:talk.puri.sm](https://matrix.to/#/!evangelos.tzaras:talk.puri.sm) (Matrix)
- [@devrtz@mastodon.online](https://mstdn.social/@devrtz) (Fediverse)

Thanks!

- Arnaud Ferraris for creating Mobian
- Henry-Nicolas Tourneur for showing me the ropes of Debian packaging
- Guido Günther for all the code reviews and explanations
- Purism for building the Phosh/GNOME ecosystem
- GNOME for hosting and generally being a helpful community
- All of you for listening!