Implementing real-time text in VoIP Infrastructure

07.05.2026, Berlin

Henning Westerholt

https://gilawa.com

About our RTC services

Real-time Communication

Management, consulting, training and development for real-time communication solutions for global clients

Kamailio Services

Development of new features and product enhancements for Kamailio and VoIP services, since 2007

Open Source

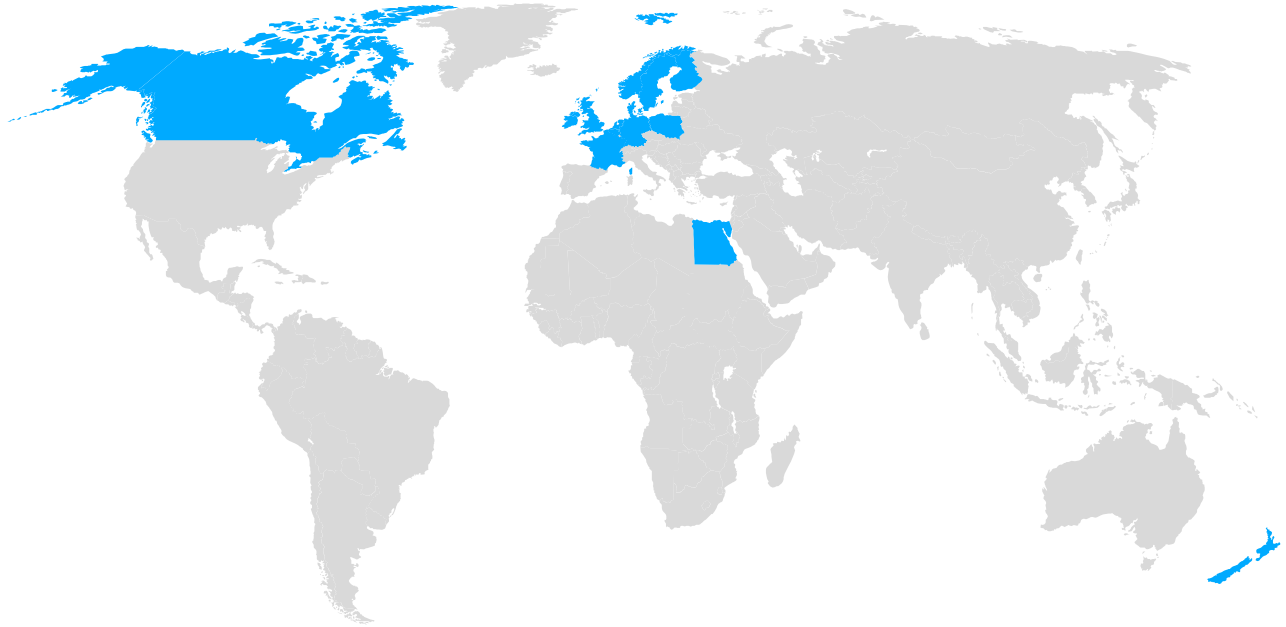
Integration of in-house developments into open-source projects

Technical Consultancy

Consulting and implementing performance optimization, availability and security

GILAWA

Our Clients



Who They Are

Internet Service Providers and Telephone Providers

Where They Are

Germany, Europe, North America, Middle East and Asia

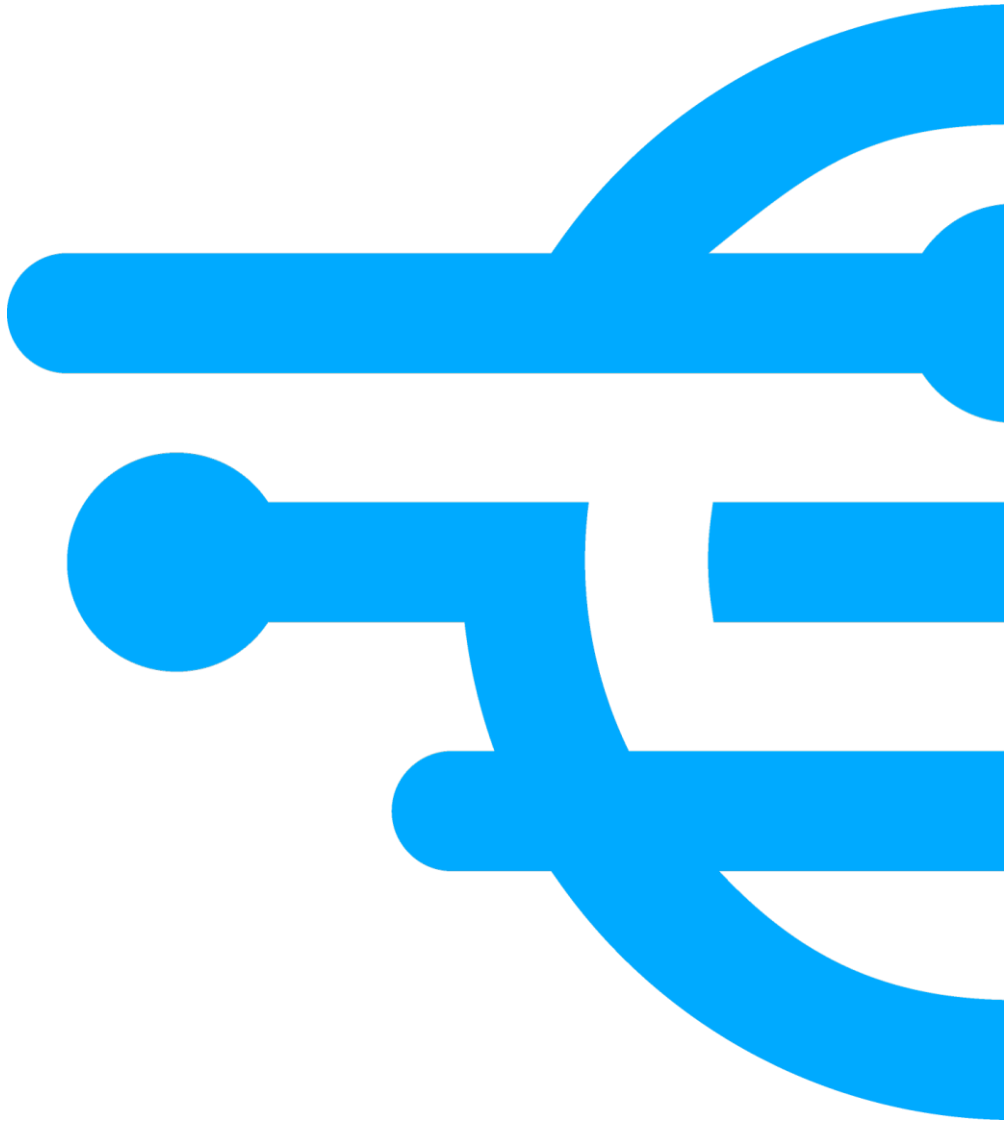
Why They Choose Us

- Independent and neutral service provider
- No proprietary products for end users
- Flexible, no preferred suppliers

GILAWA

GILAWA

Implementing RTT in VoIP infrastructure



Agenda

- What is real-time text?
- Motivation and legal requirements
- Technical implementation of RTT
- Implementation with Kamailio and Asterisk
- Current status and next steps

GILAWA

General remark

- This work started end of 2024, the first PR version was created in the beginning of 2025 – more about it later
- Now one year later we are hopefully close to getting it merged
- This was a group effort from our side, thanks to the whole team!
- Thanks also to our customer for sponsoring this work and for their patience!

What is real-time text ?

Real-time text (RTT) is text transmitted instantly as it is typed. Recipients can immediately read the text while it is being written, without waiting.

This works differently e.g. as using common instant messaging tools or office collaboration tools, where messages are sent, when the writer is ready.

There are different technical means to transmit RTT, based on various technical standards.

GILAWA

Motivation

- Main audience are people which cannot hear well or are deaf
- It eliminates the need for special devices such as TTY machines or OTT apps
- For anyone who prefers to use text-based communication but in real-time
- It provides full support for international character set and emojis
- For emergency situations where it’s impossible to speak
- It can be combined with standard audio and video calls, “Total Conversation”

GILAWA

Legal requirements (1/2)

- European regulation 2019/882 of 17th April 2019 (Accessibility requirements for products and services)
- European regulation 2023/444 of 16th December 2022 supplementing Directive 2018/1972

- “To ensure effective access to emergency services via emergency calls to the single European emergency number 112”
- In Germany mandated by TR. Notruf 2.1 from the BNetzA
- Other countries have similar requirements

GILAWA

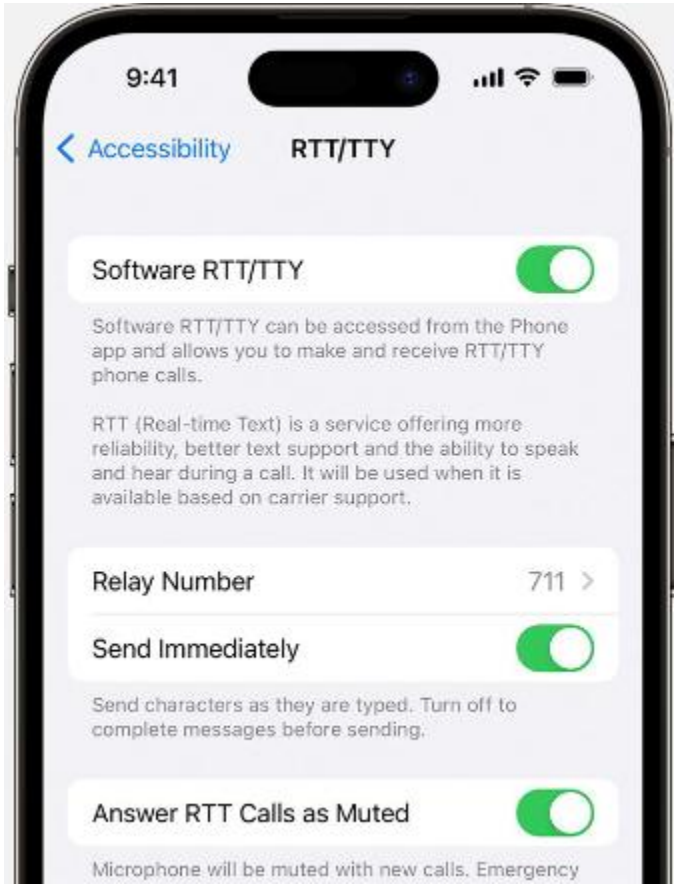
Legal requirements (2/2)

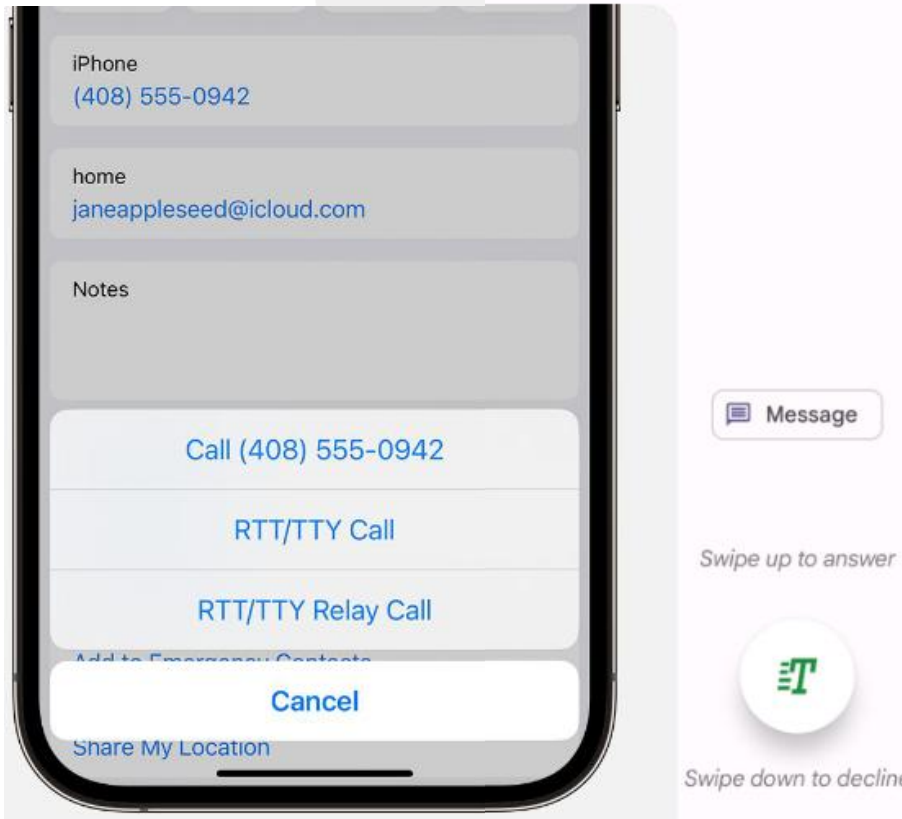
- Different approaches in EU
  - France, Germany, Netherlands, Italy and Sweden are introducing RTT as part of their national legislation (usually in core netw.)
  - Denmark, Finland, Norway, Spain and Switzerland are using OTT apps
- Older report from eena: link

- There are multiple approaches, even countries that aim for core network integration are using in parallel OTT apps
- The legal implementation deadline in the EU is 2027
- Changes in core network, user devices and PSAPs necessary

GILAWA

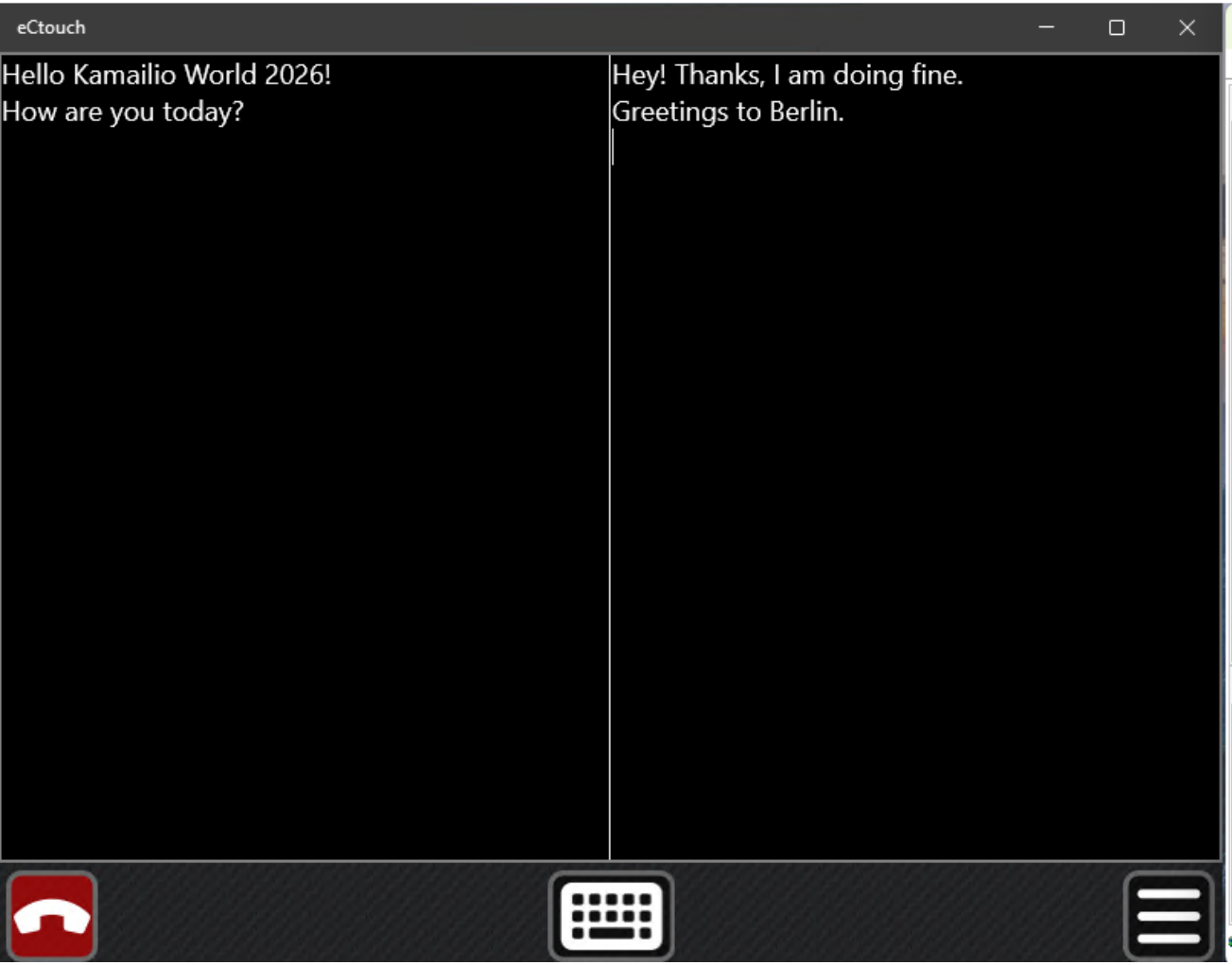
How does it actually look like?

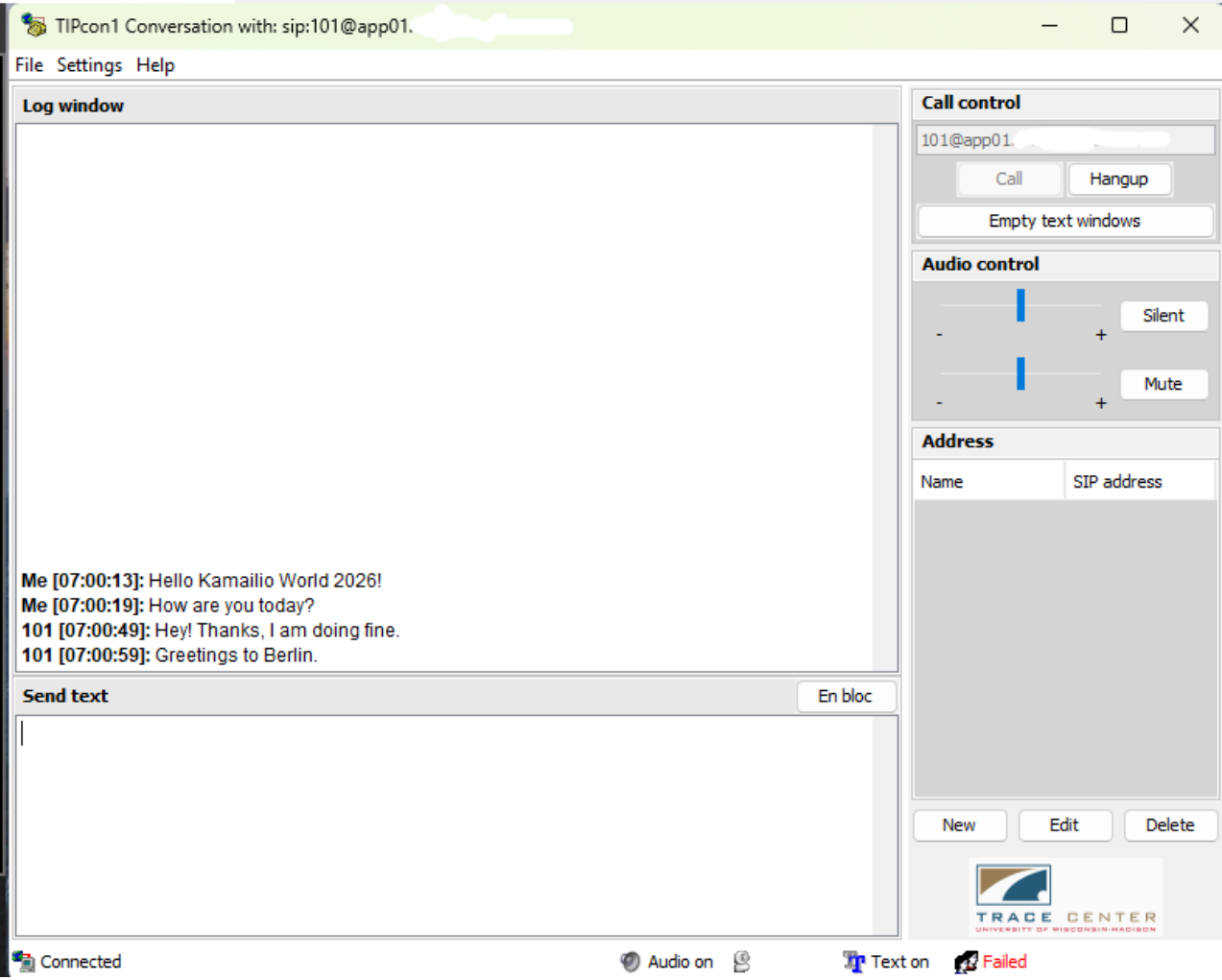




GILAWA

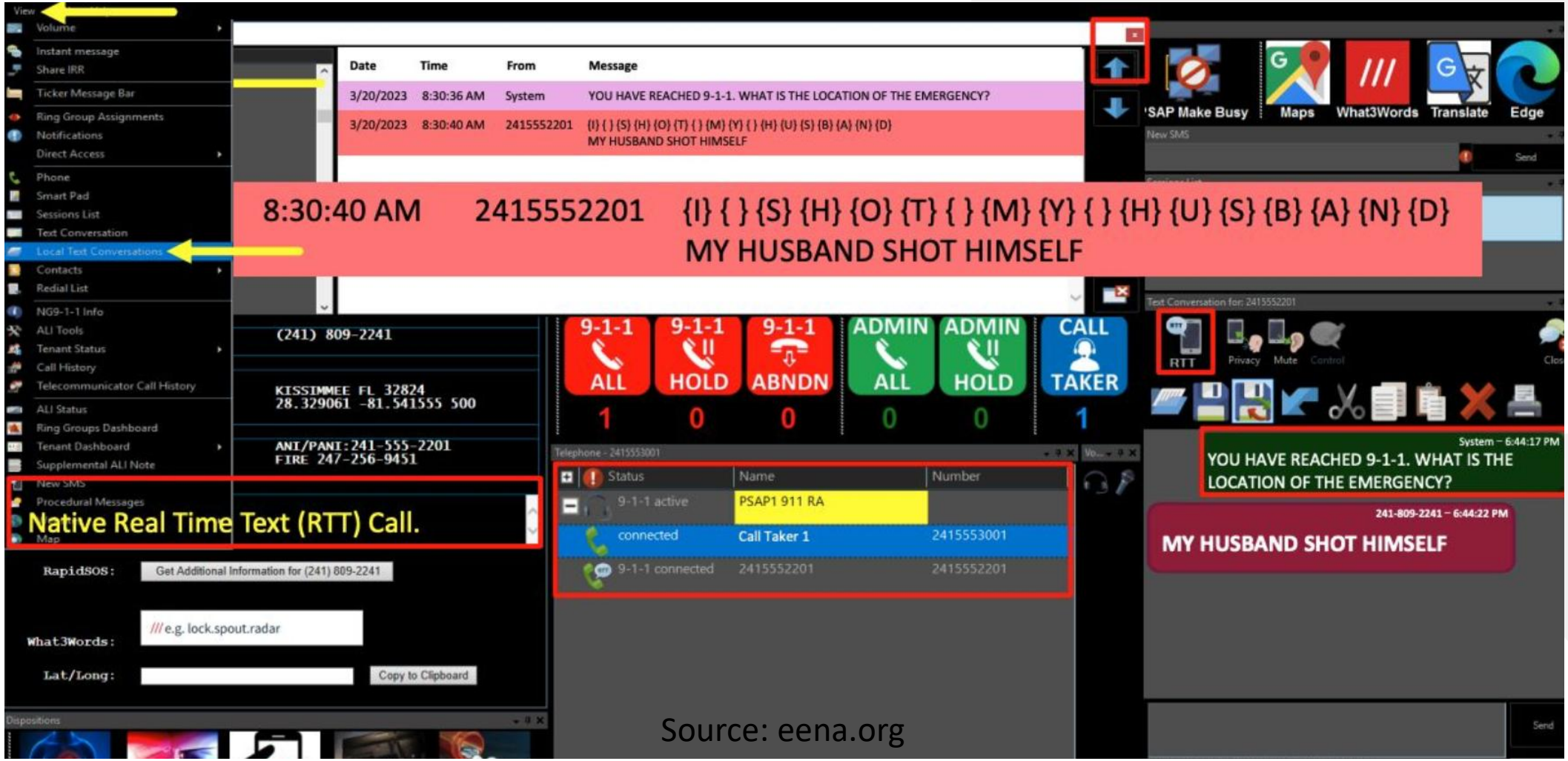
How does it actually look like?





GILAWA

How does it actually look like?



Source: eena.org

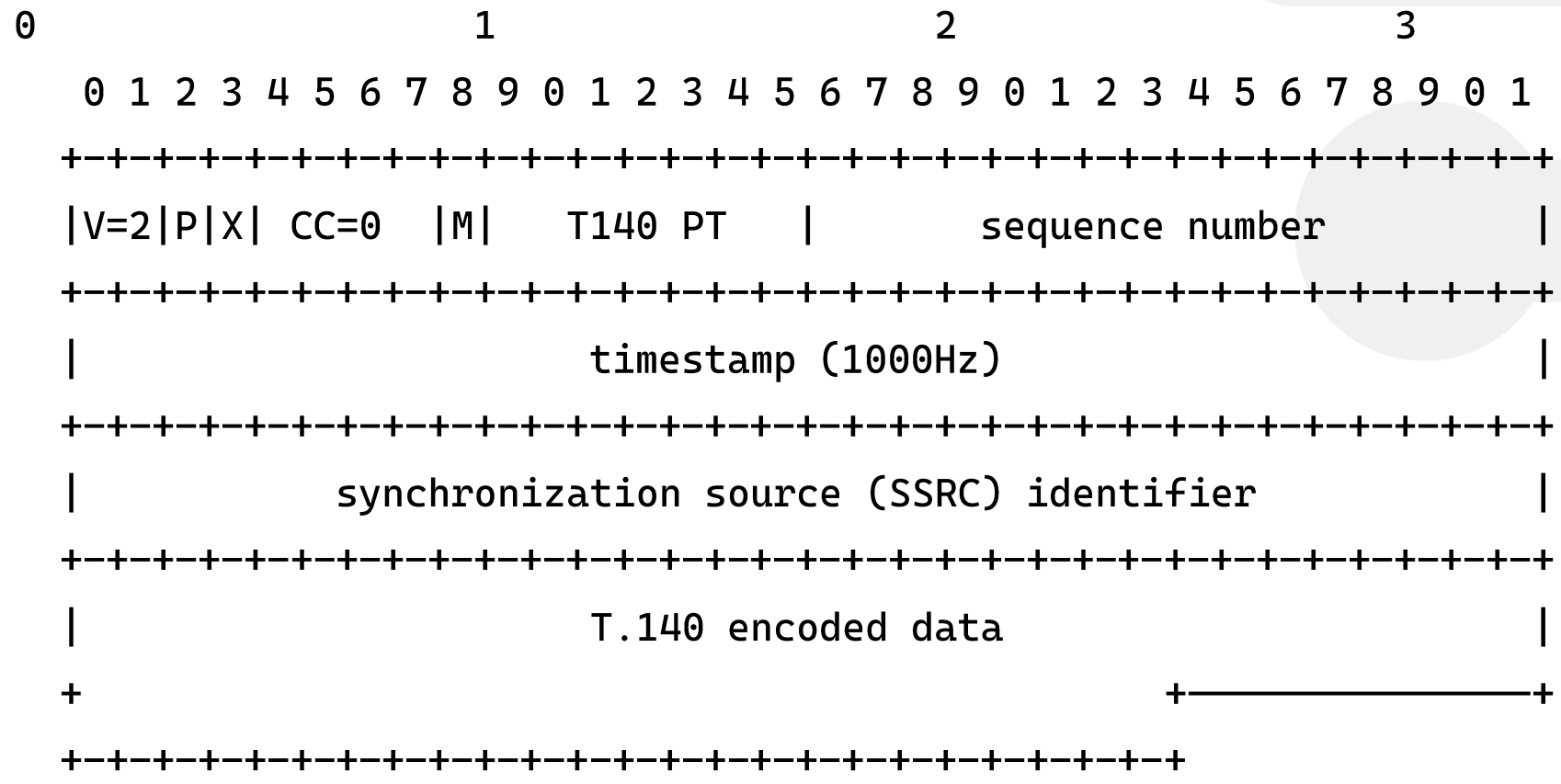
Technical implementation of RTT

- RTT calls are standard SIP calls
- Text format defined in standard ITU-T T.140
- RTP payload for RTT defined in RFC 4103
- Newer developments like RFC 9071 for multi-party RTT calls published more recently

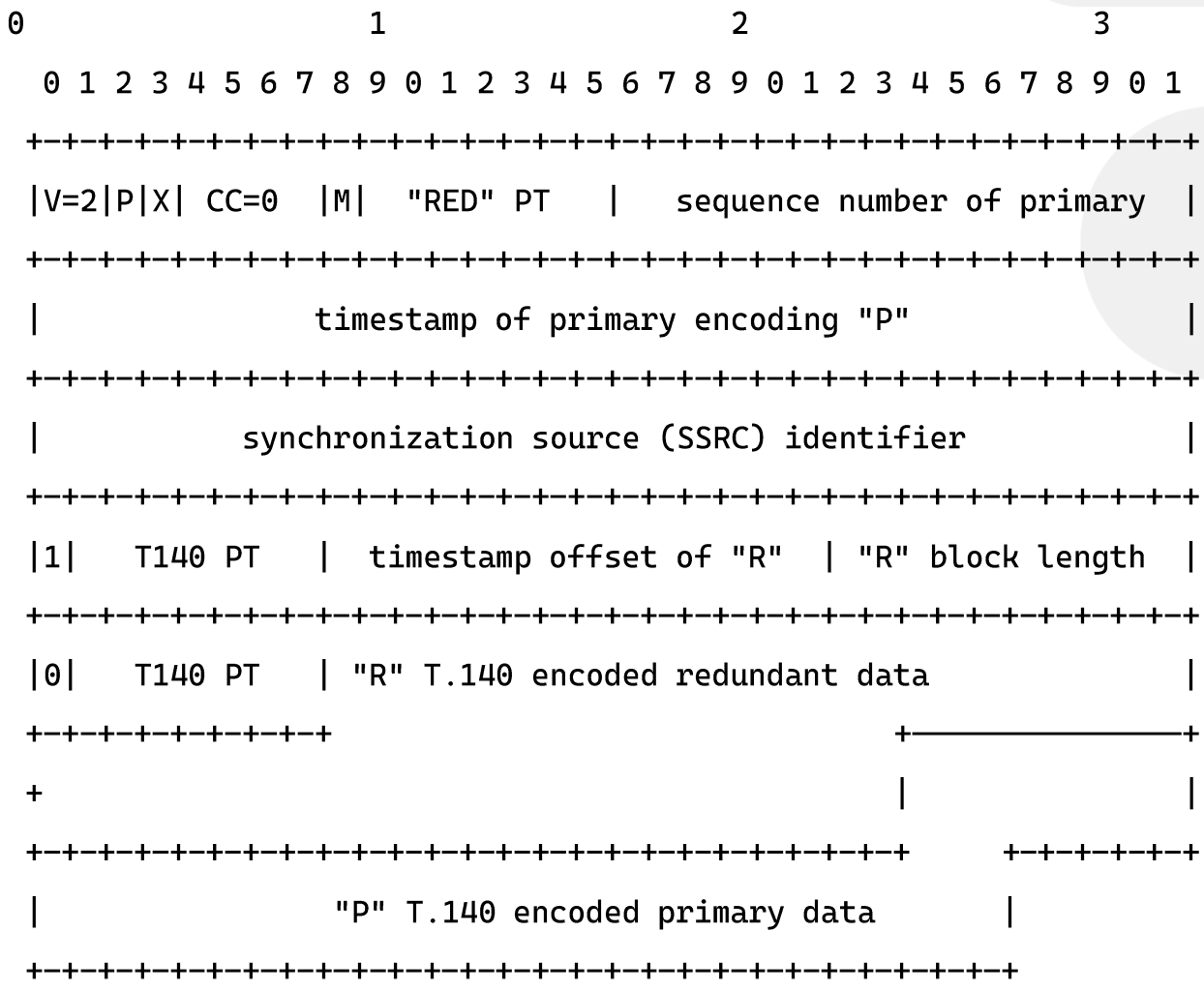
- From a media/RTP point of view the RTT calls are just using a dedicated text media stream
- Usual and well-known SDP offer-answer negotiation
- Stand-alone text stream or combined with audio/video

GILAWA

RTT basic RTP format



RTT RTP format with redundancy



SDP examples

Normal RTT:

m=**text** 11000 RTP/AVP 98

a=rtpmap:98 **t140**/1000

RTT with redundancy:

m=**text** 11000 RTP/AVP 98 100

a=rtpmap:98 **t140**/1000

a=rtpmap:100 **red**/1000

a=fmtp:100 98/98/98

GILAWA

Implementation with Kamailio

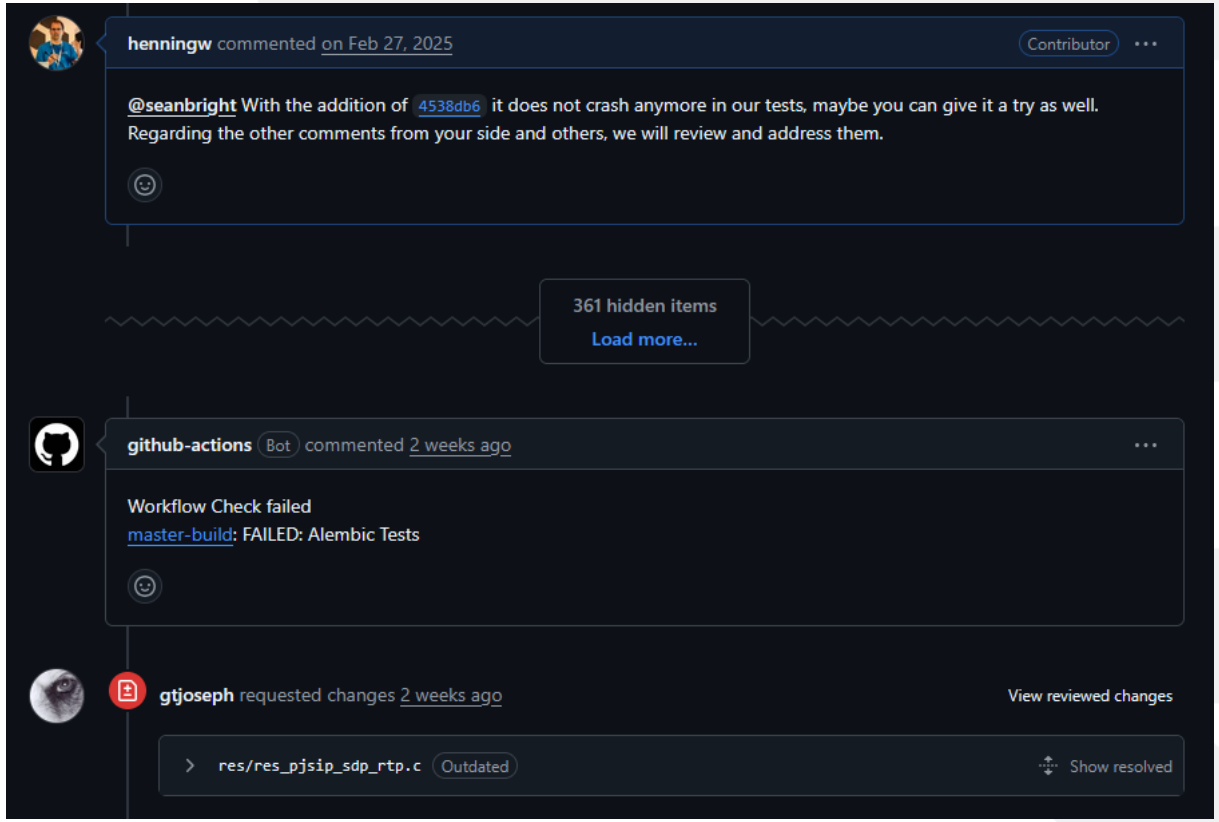
- As Kamailio doesn’t do much about the media, it mostly depends on the user agent
- It seems to work fine with rtpengine as well
- Straightforward to setup now

- User agents for testing
  - pjsua version 2.17
  - tipcon1 1.5.1
  - ectouch (not freely available)
  - Last years apple/android mobile phones also support it
- For mobile phone tests you need a test IMS/mobile core

GILAWA

Implementation with Asterisk

- Asterisk has an existing implementation in chan_sip
  - This is of course obsolete
  - It seems to be also broken
- We are working on a chan_pjsip implementation
- This turned out to be difficult
- Longest PR I experienced so far



Problems during implementation

- PR was based on a previous, unfinished PR
- Close integration with many different Asterisk parts necessary
- Implementation choices in Asterisk regarding interfaces and internal integration
- Manual testing during the implementation, difficult test environment, delays from our side
- Unreliable testing clients, pjsua RTT not available during most of the implementation
- Several longer waiting phases for feedback from Asterisk project
- Multiple RTT modes and bridging methods, Asterisk did not supported text-only calls
- Several times changing requirements from Asterisk developer side

GILAWA

Asterisk output

```
-- Executing [101@rtt-test:1] NoOp("PJSIP/102-00000006", "Calling 101") in new stack
-- Executing [101@rtt-test:2] Dial("PJSIP/102-00000006", "PJSIP/101") in new stack
-- Called PJSIP/101
-- PJSIP/101-00000007 is ringing
   > 0x7940a8092d90 -- Strict RTP learning after remote address set to:      ..138.134:8736
   > 0x7940a8097dd0 -- Strict RTP learning after remote address set to:      ..138.134:7958
-- PJSIP/101-00000007 answered PJSIP/102-00000006
   > 0x7940a8051ec0 -- Strict RTP learning after remote address set to: 192.168.188.59:1028
   > 0x7940a80580f0 -- Strict RTP learning after remote address set to: 192.168.188.59:1030
-- Channel PJSIP/101-00000007 joined 'simple_bridge' basic-bridge <1d29056b-2496-47c0-8287-189bc76c8b84>
-- Channel PJSIP/102-00000006 joined 'simple_bridge' basic-bridge <1d29056b-2496-47c0-8287-189bc76c8b84>
   > 0x7940a8092d90 -- Strict RTP qualifying stream type: audio
   > 0x7940a8092d90 -- Strict RTP switching source address to       .178.196:60560
   > 0x7940a8097dd0 -- Strict RTP qualifying stream type: text
   > 0x7940a8051ec0 -- Strict RTP qualifying stream type: audio
   > 0x7940a8051ec0 -- Strict RTP switching source address to       .178.196:1028
   > 0x7940a8097dd0 -- Strict RTP switching source address to       .178.196:60540
= Endpoint phone101 is now Reachable
-- Contact phone101/sip:phone101@      .178.196:57090;x-ast-orig-host=192.168.188.59:0 is now Reachable.  RTT: 74.600 msec
   > 0x7940a8092d90 -- Strict RTP learning complete - Locking on source address       .178.196:60560
   > 0x7940a8051ec0 -- Strict RTP learning complete - Locking on source address       .178.196:1028
   > 0x7940a8097dd0 -- Strict RTP learning complete - Locking on source address       .178.196:60540
   > 0x7940a80580f0 -- Strict RTP qualifying stream type: text
   > 0x7940a80580f0 -- Strict RTP switching source address to       .178.196:1030
   > 0x7940a80580f0 -- Strict RTP learning complete - Locking on source address       .178.196:1030
-- Channel PJSIP/101-00000007 left 'simple_bridge' basic-bridge <1d29056b-2496-47c0-8287-189bc76c8b84>
-- Channel PJSIP/102-00000006 left 'simple_bridge' basic-bridge <1d29056b-2496-47c0-8287-189bc76c8b84>
```

GILAWA

Current status

- Main PR is complete from our side, automated test suite created, this PR will be created soon
- RTT T140 and RED calls between ectouch and tipcon1 and ectouch and pjsua work
- RTT T140 calls between pjsua and pjsua work with audio
- RTT T140 calls between tipcon1 and pjsua work with and without audio
- RED calls betw. tipcon1 and pjsua with and w/o audio, asymmetric codec B side, pjsua issue
- RTT T140 calls between pjsua and pjsua don’t work w/o audio, likely a limitation in Asterisk
- RTT RED calls between pjsua and pjsua don’t yet work due to a pjsua issue
- Native bridging and simple bridging works in Asterisk, with RED „transcoding“ to T140

GILAWA

Future Work

- Finally get the PR merged, waiting for feedback from the asterisk side
- Wait for feedback on the found pjsua issues
- Create an issue for the asterisk audio problem with pjsua
- Follow up on the RTT RED pjsua issues
- Hopefully see it in production soon!

GILAWA

GILAWA

Thank you!

Questions?

https://gilawa.com

GILAWA

Contact Us

+357 24 030678

+49 1579 2475270

mail@gilawa.com

https://gilawa.com