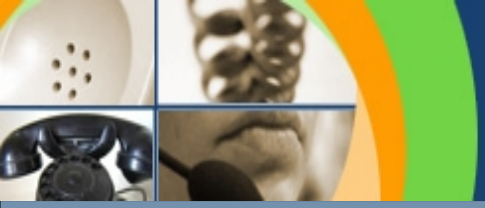


http://www.gnutelephony.org/data/scale8x.odp

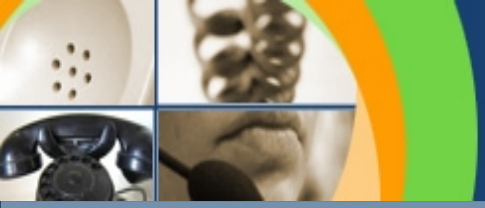
Privacy is ultimately about liberty

Surveillance is always about control

Communication Privacy For Free Societies

David Sugar

Scale/8x 2010



GNU Telephony
Mission Statement

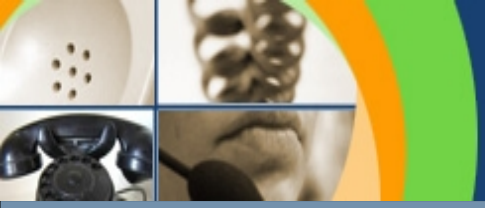
SECURE CALLING PROJECT GOALS:

To empower people, individually and collectively, to communicate and collaborate privately and securely in real-time worldwide

To establish secure communications as the default communication infrastructure

To enable secure anonymous communication worldwide and protect users who exercise their basic human freedom of privacy

To provide secure communication services universally on all computing platforms



GNU Telephony

Why free software

Anyone can review what they receive; no hidden backdoors

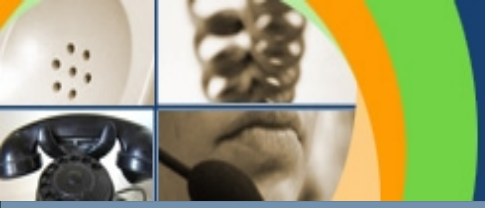
Anyone can modify the software for their specific needs or for specific platforms

Anyone can redistribute the software and help make it widely available

Everyone has universal and unrestricted access to the software worldwide

Everyone can participate on an equal basis in it's development

No-one can remove the software from availability once distributed



GNU Telephony

Challenges we face

Software Patents and Intellectual Monopolies

Anti-privacy laws effecting communication services

Service Blocking and Net Neutrality

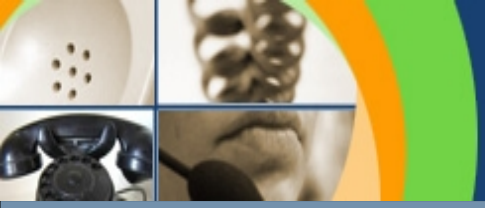
Private commercial data mining

The need for Zero-Knowledge Systems to protect users, zero forward knowledge to protect past conversations, etc

Peer review-able code and verifiable algorithms

Verifiable end-user client software

Trustworthy hardware and client operating systems



GNU Telephony

How we started

1949 George Orwell publishes “1984”

1994 Calea Act introduced into law with promise never to be used for mass domestic surveillance

2001 (spring) Mass domestic communication intercept begins using Calea mandated backdoors

2004 CALEA proposed for VoIP, Internet Common Congress Held

2006 GNU ZRTP stack Introduced

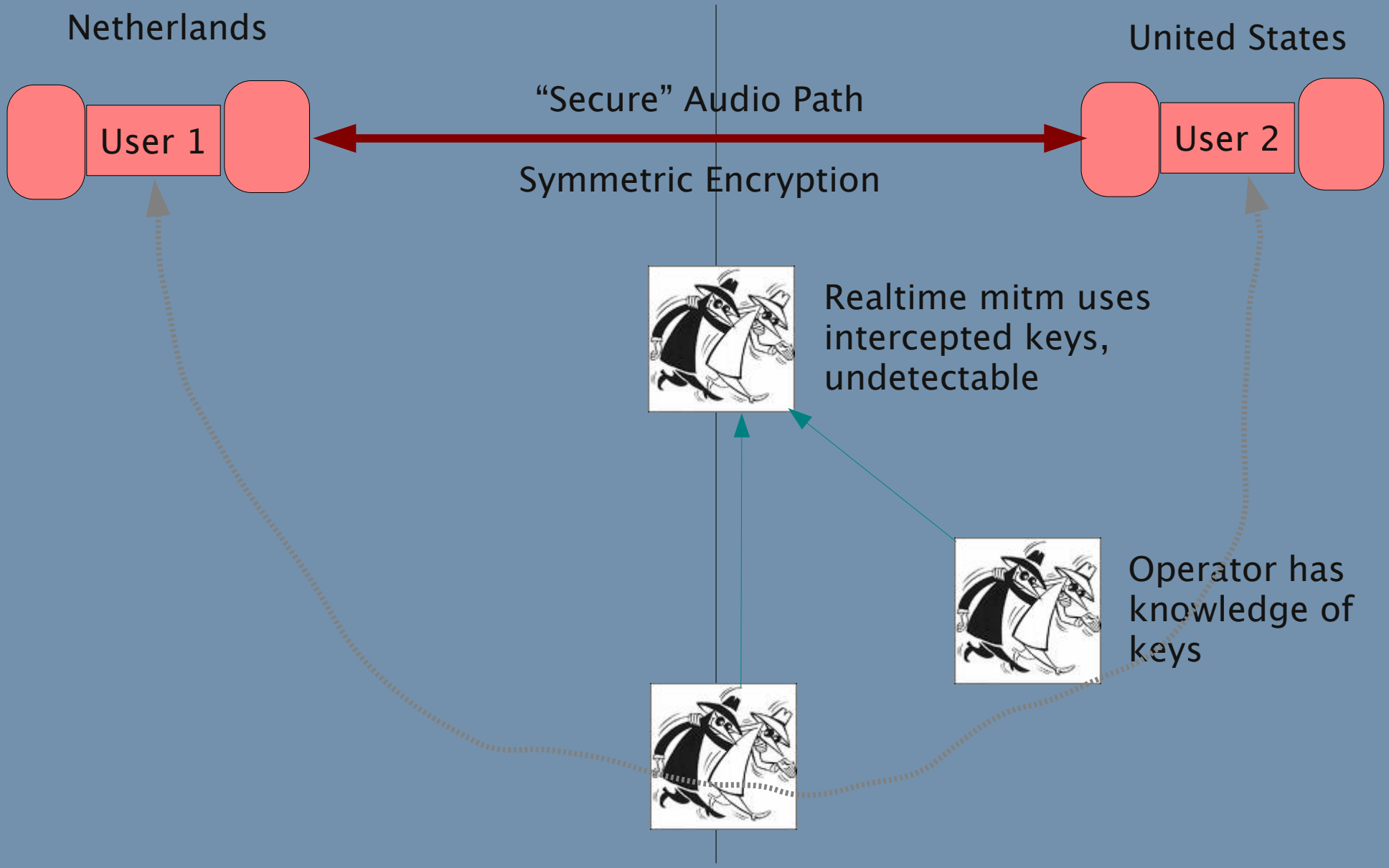
2007 GNU Secure Calling Project started

2008 GNU SIP Witch Introduced as secure phone switch

2010 Secure Calling in Ubuntu 10.04 and Fedora F13 GNU/Linux

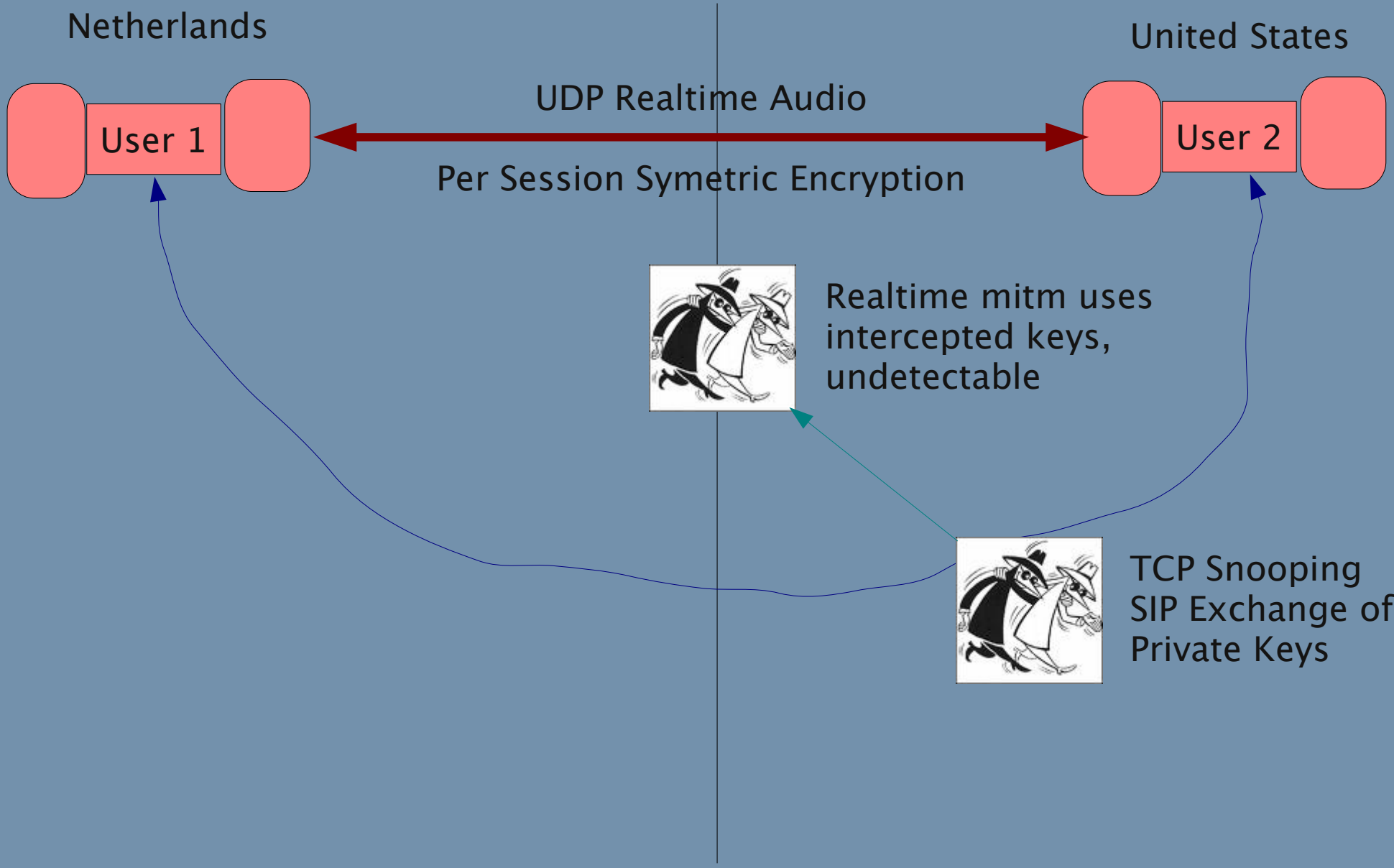
GNU Telephony

Classic Media Insecurity



GNU Telephony

SDES Media Insecurity



GNU Telephony

S-RTP & PKI Media Insecurity

United Kingdom

Netherlands

User 1

UDP Realtime Audio

User 2

PKI Encryption Static Certificates

Certificate Stolen
or “RIPA” all Past
& Present
calls compromised

Realtime mitm
False identity or decrypting
compromised certs

Certificate
Authority

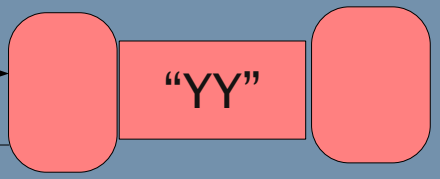
Poisoned/Weak Certificates or
copied to third party outright. All
Past & Present calls compromised

GNU Telephony

ZRTP and SAS

Sending Public Key XX

Sending Public Key YY



Sends Local Public Key XX
Has Local Private Key for XX
Gets Remote Public Key YY
SAS Generated Hash XXYY
SAS Matches, confirmed over voice

Sends Local Public Key YY
Has Local Private Key for YY
Gets Remote Public Key XX
SAS Generated Hash XXYY
SAS Matches on voice

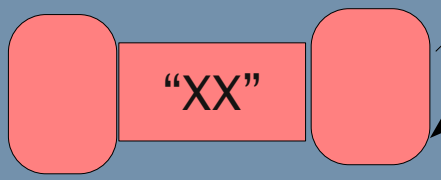
Sending Key XX

Send Key ZZ

Sends Key ZZ

Send Key YY

MITM does not have private keys for XX or YY, so must create a new fake key ZZ



Sends Local Public Key XX
Has Local Private Key for XX
Gets Remote Public Key ZZ
SAS Generated Hash XXZZ
SAS does not match when checked over voice!

Sends Local Public Key YY
Has Local Private Key for YY
Gets Remote Public Key ZZ
SAS Generated Hash ZZYY
SAS does not match!

GNU Telephony

ZRTP Media Security

United Kingdom

United States

User 1

UDP Realtime Audio

User 2

PKI Encryption & Key Exchange

Per session keys
not static, no user
keys for RIPA

Realtime mitm for key exchange
vs SAS validation

Locally user generated keys

Keys generated per session

User has zero knowledge of keys

Users can validate each others keys

Peer reviewable and verifiable

Certificate Authority

Locally generated keys
no authority to compromise

GNU Telephony

ZRTP & PBX enrollment

United Kingdom

United States

Ext 10

Ext 11

Remote

"Appears secure"

SAS relay valid if switch trusted

Destination insecure! But no SAS to confirm

Local IP-PBX uses pre-connect

Remote IP-PBX uses pre-connect

Audio path decrypted in server

Interconnect maybe insecure. SAS cannot relay cross-node

Destination insecure! But also no SAS to confirm

Cannot call securely between nodes

IP-PBX Server must be "trusted"

GNU Telephony

ZRTP & PBX Passthrough

United Kingdom

United States

Ext 10

Ext 11

Remote

"Appears secure"

SAS relay valid if switch trusted

Destination insecure! But no SAS to confirm

Local IP-PBX uses pre-connect

Remote IP-PBX uses pre-connect

Audio path should remain encrypted in server, but what if config is falsified?

Interconnect maybe insecure. SAS cannot relay cross-node

Destination insecure! But also no SAS to confirm

Cannot call securely between nodes

Enrollment is used, IP-PBX holds keys, can falsify encrypted path in switch

GNU Telephony

SIP Witch & Media Security

No uncertainty about end-to-end security in voip media path

Secure with direct media path & zrtp

Ext 10

Ext 11

Remote

Secure with direct media path & zrtp

Local SIP Witch

Remote SIP Witch

No audio to centrally decrypt

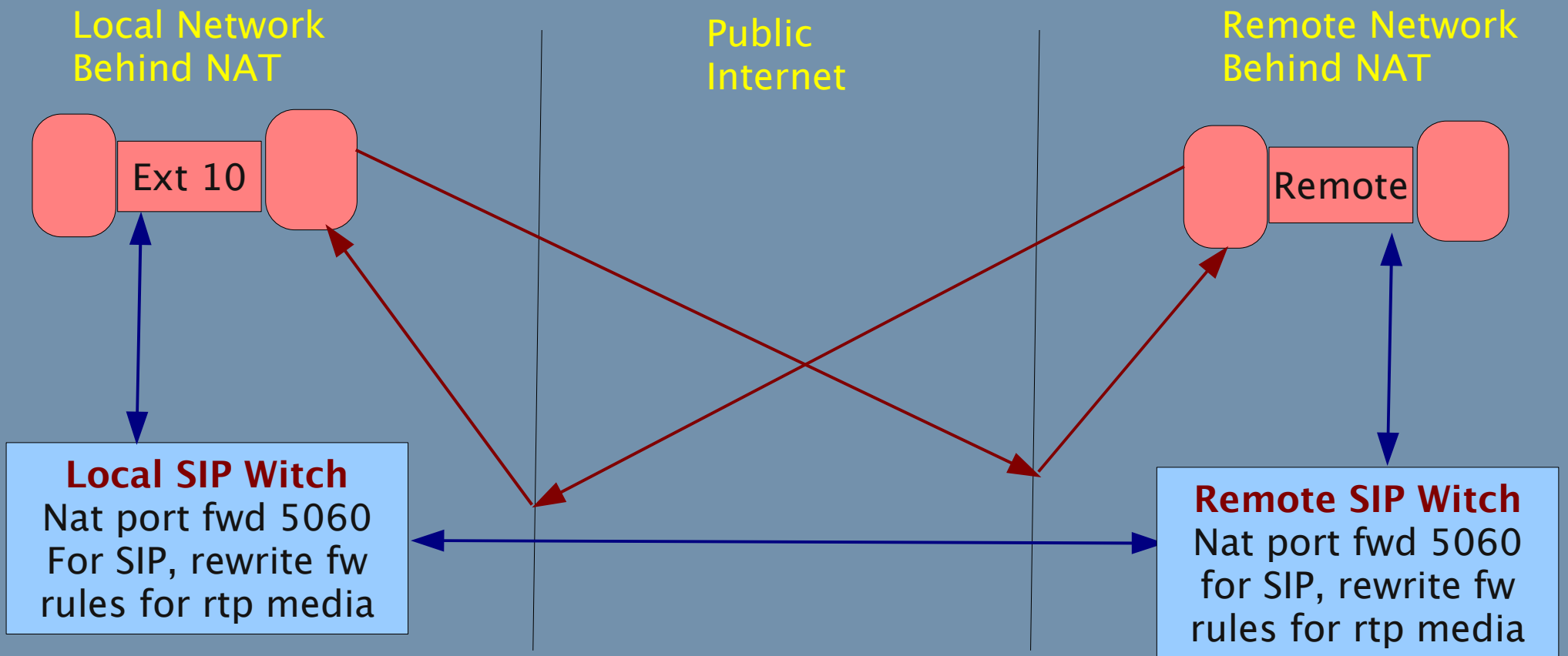
No audio to centrally decrypt, no media interconnect

PSTN Gateway

PSTN gateway path may be secure but destination is not but clear boundries between secure & insecure domains

GNU Telephony

NAT and Media Proxy

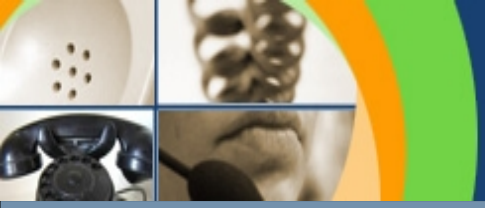


Rewrite of firewall rules to packet forward rtp media on the fly

Integrated rewrite of SIP SDP based on public appearing addresses

Clients have no need for NAT support; all done in one place in sipwitch!

Low cpu overhead, minimal latency, and stateful; server dies but calls remain alive!



GNU Telephony

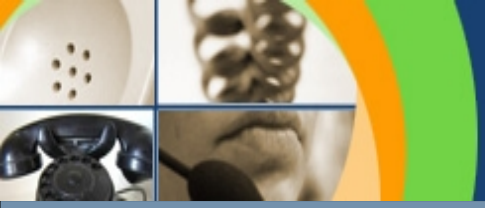
Traditional Roles

SIP Telephone Switch:
* call forward and multi-nodal
* multi-party ring & registration
* multi-node and routing
* class of service/profiles
* reduced traffic on trusted nets
* feature code dialing (todo)
* hunting & acd (todo)
* speed dialing (todo)

SIP Embedded Gateway:
* map subscriber to multi-party
* arm, mips port proven
* compilable for embedded
* rtp media proxy
* very low overhead
* xmlrpc remote management

Internet Hosted Service:
* media peering possible
* virtualizes well
* can run as user w/o root

Secure Call Domain adjunct:
* cross-register with IP-PBX
* fwd insecure to IP-PBX
* clean domain division



GNU Telephony

SIP Witch on the Desktop

Use existing SIP softphone clients

Use your system Login account as a SIP login

Single sign-on for multiple remote accounts

Single place to implement NAT correctly!

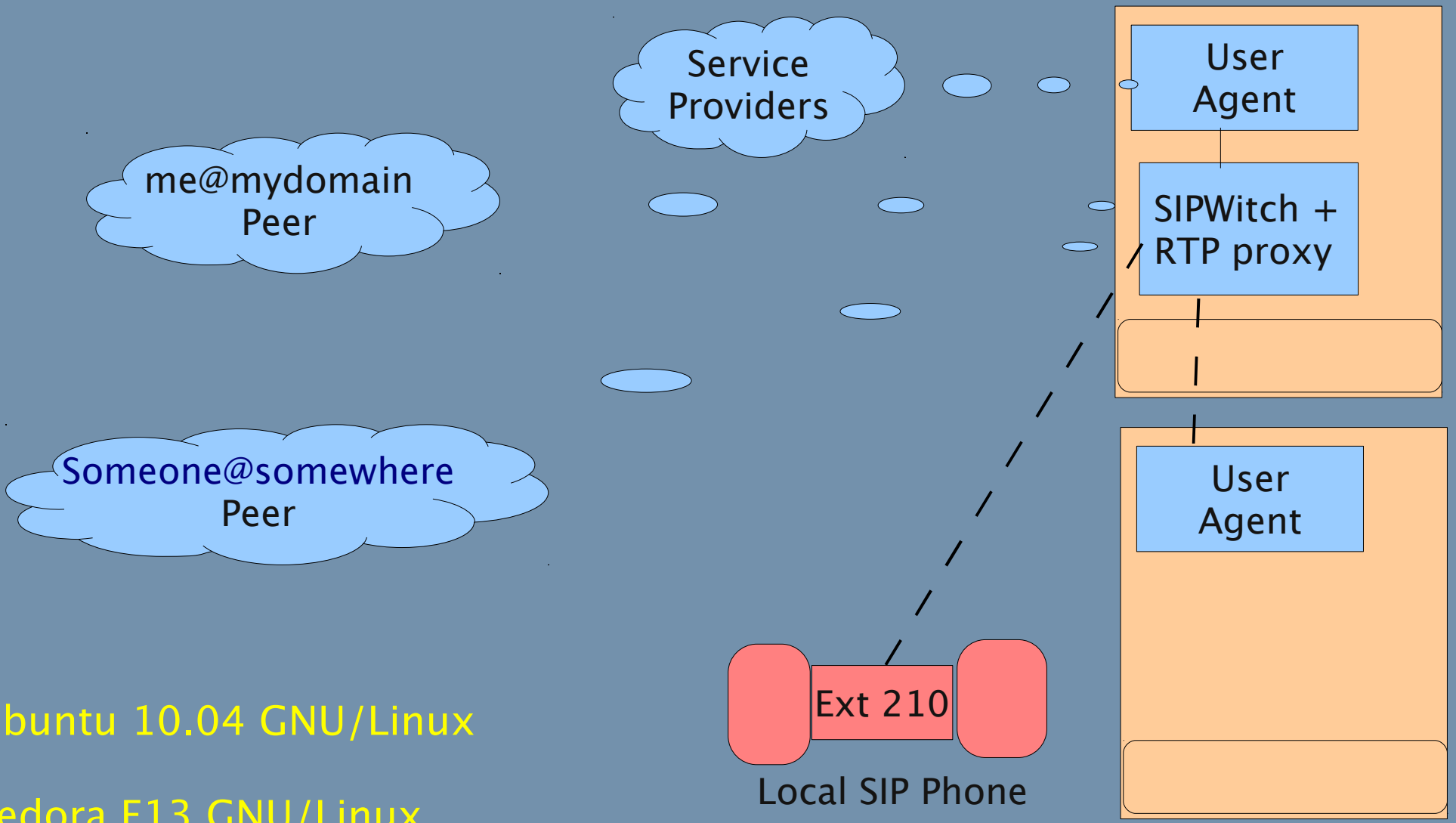
Automatic self configuration!

Simplified service provider provisioning

Creative routing and redirection; a “Gstreamer” for VoIP!

GNU Telephony

Domain Calling

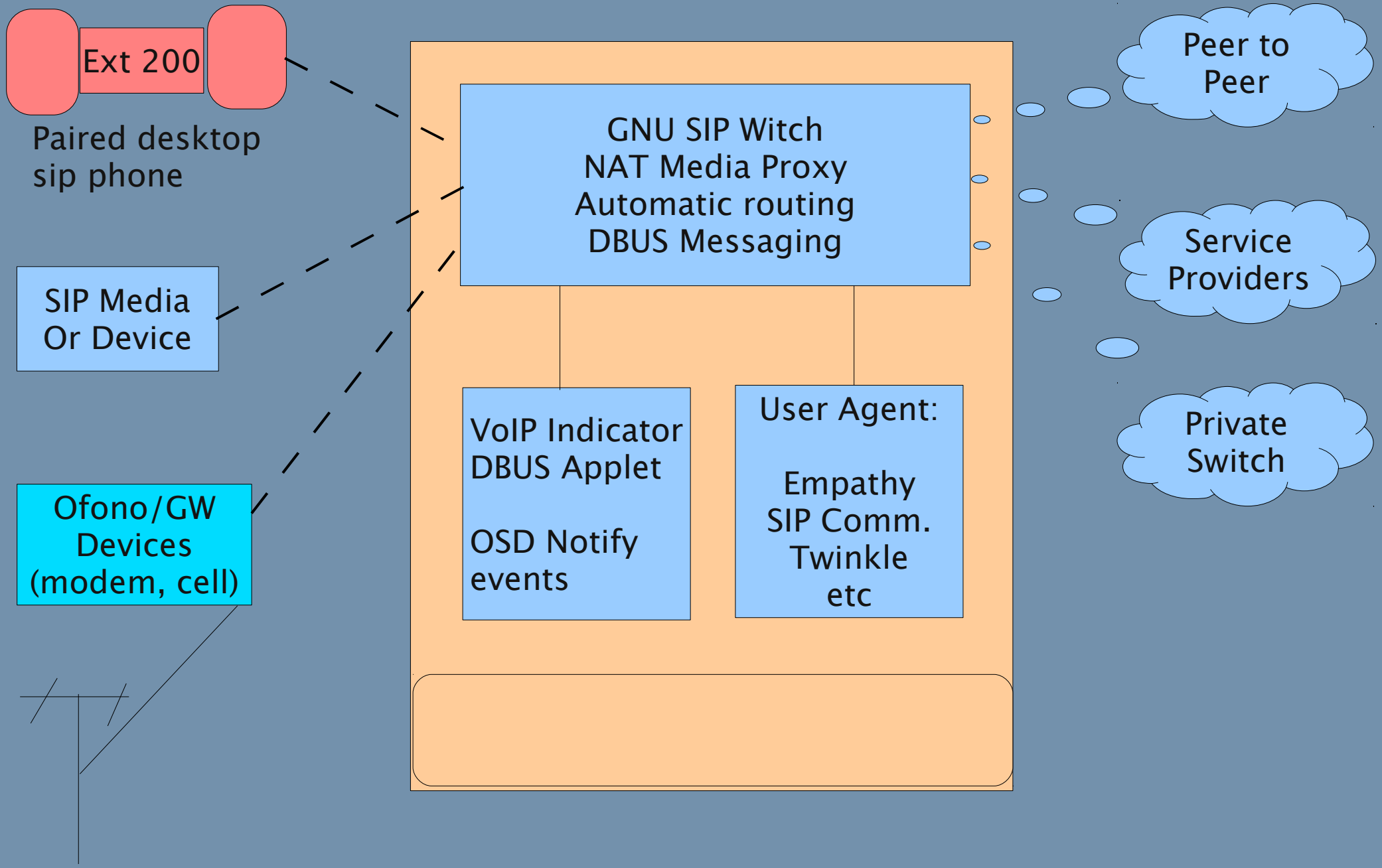


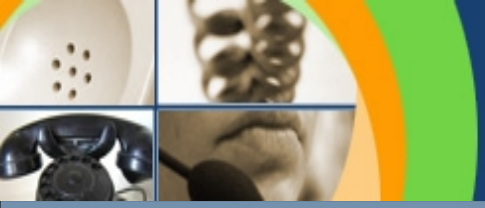
Ubuntu 10.04 GNU/Linux

Fedora F13 GNU/Linux

GNU Telephony

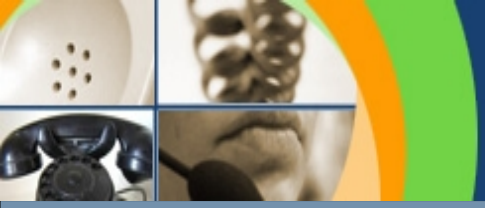
The VoIP Desktop





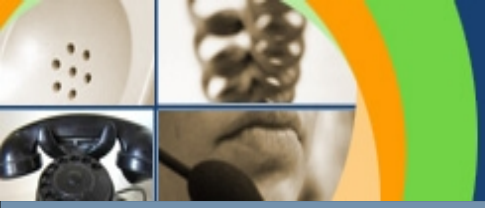
How you can help

Create domain calling networks bottom-up
Test and use various deployment models
Report bugs to sipwitch-devel@gnu.org
Document using different GNU/Linux distros
Help us document basic sipwitch use cases
Test SIP clients and devices
Contribute code to the community
Communicate freely using free software



GNU Telephony
http://www.gnutelephony.org
mailto:dyfet@gnutelephony.org
mailto:sipwitch-devel@gnu.org

Free World Dialup: 688841
sip:dyfet@sip.gnutelephony.org
irc:#bayonne irc.freenode.net
jabber:gnudyfet@gmail.com



HAPPY

Hacking