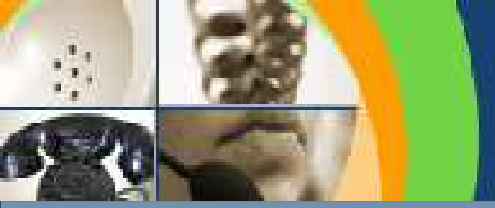


GNU Telephony

Telephony for a free world

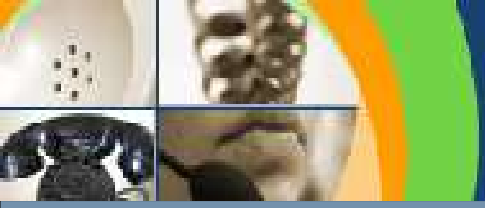
Contents

- Introduction to GNU Telephony
- ZRTP & Secure Media Paths
- VOIP architecture & latency
- Peer media vs Enrollment
- SIP Security
- The GNU Telephony Stack
- Roadmap
- Contact Information & Help

Secure Calling

David Sugar

SecTor 2008



GNU Telephony

Who we are

Founded to develop Free Software for Telecommunications

We have volunteers world-wide developing free telephony solutions

Maintains telephony related packages within the GNU Project

Work with Hipatia on social goals

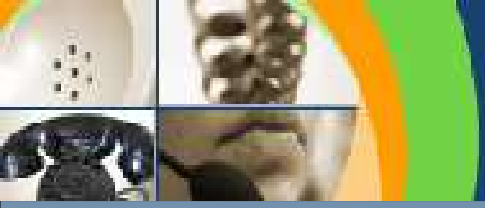
Developing Secure Calling for intercept-free communications

Developing model secure voip telephone network application stack

Developing reference platform for GNU Bayonne for Telecenters

Key people

* Werner Dittmann; ZRTP stack development
* Federico Pouzols; GNU RTP stack
* Aymeric Moizard; GNU oSIP and exosip stack
* Michael De Boer; Twinkle softphone
* David Sugar; GNU SIP Witch, GNU Bayonne, coordinator



GNU Telephony

Challenges we face

Primary Issues in Call Quality Today

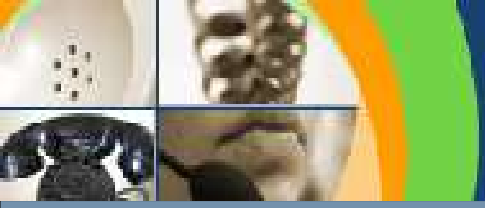
* Network & Codec Latency
* Bandwidth & QoS
* Broad vs Narrow band audio
* Audio conferencing

Concepts we have explored

* Audio as a spacial environment
* Peer vs Spoke-Wheel Architectures
* Smarter clients & endpoint mixing

Political & Social Challenges

* CALEA vs Privacy
* Net Neutrality
* Software Patenting; patent encumbered “mandated” standards
* Service blocking & incumbent carriers
* Data mining of signalling
* Media channel privacy
* Securing future networks



GNU Telephony

SDES Media Insecurity

Netherlands

United States

User 1

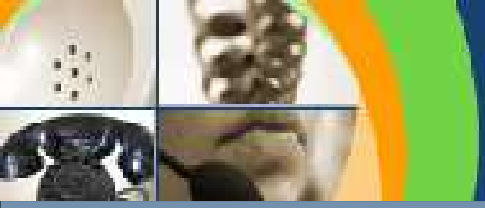
UDP Realtime Audio

Per Session Symetric Encryption

User 2

Realtime mitm uses intercepted keys, undetectable

TCP Snooping SIP Exchange of Private Keys



GNU Telephony

S-RTP & PKI Media Insecurity

United Kingdom

Netherlands

User 1

UDP Realtime Audio

User 2

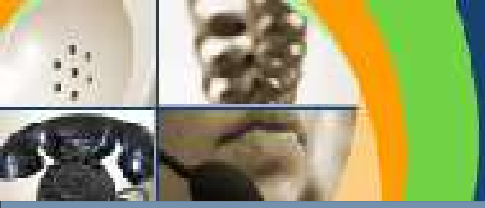
PKI Encryption Static Certificates

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 Media Security

United Kingdom

United States

User 1

UDP Realtime Audio

User 2

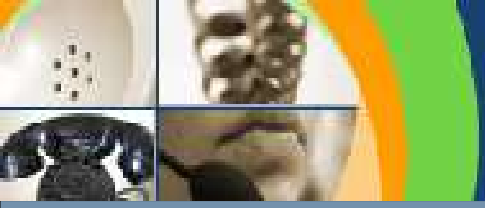
PKI Encryption & Key Exchange

Per session keys
not static, no user
keys for RIPA

Realtime mitm for key exchange
vs SAS validatation

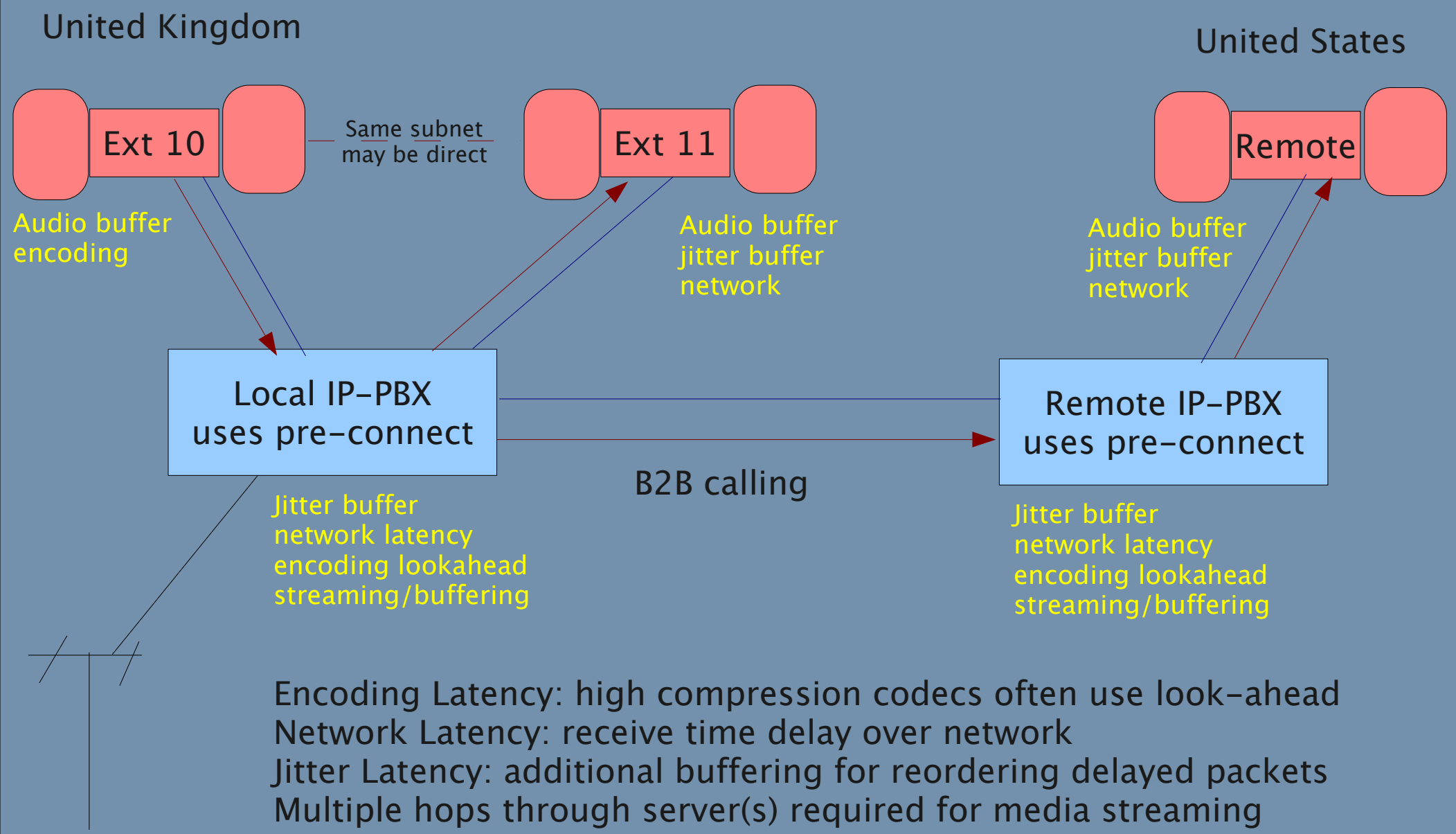
Certificate
Authority

Locally generated keys
no authority to compromise

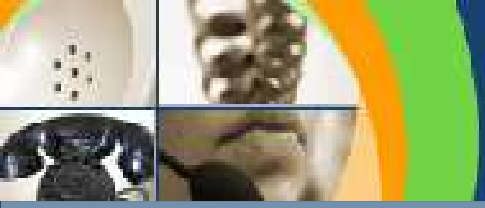


GNU Telephony

IP-PBX Media Latency

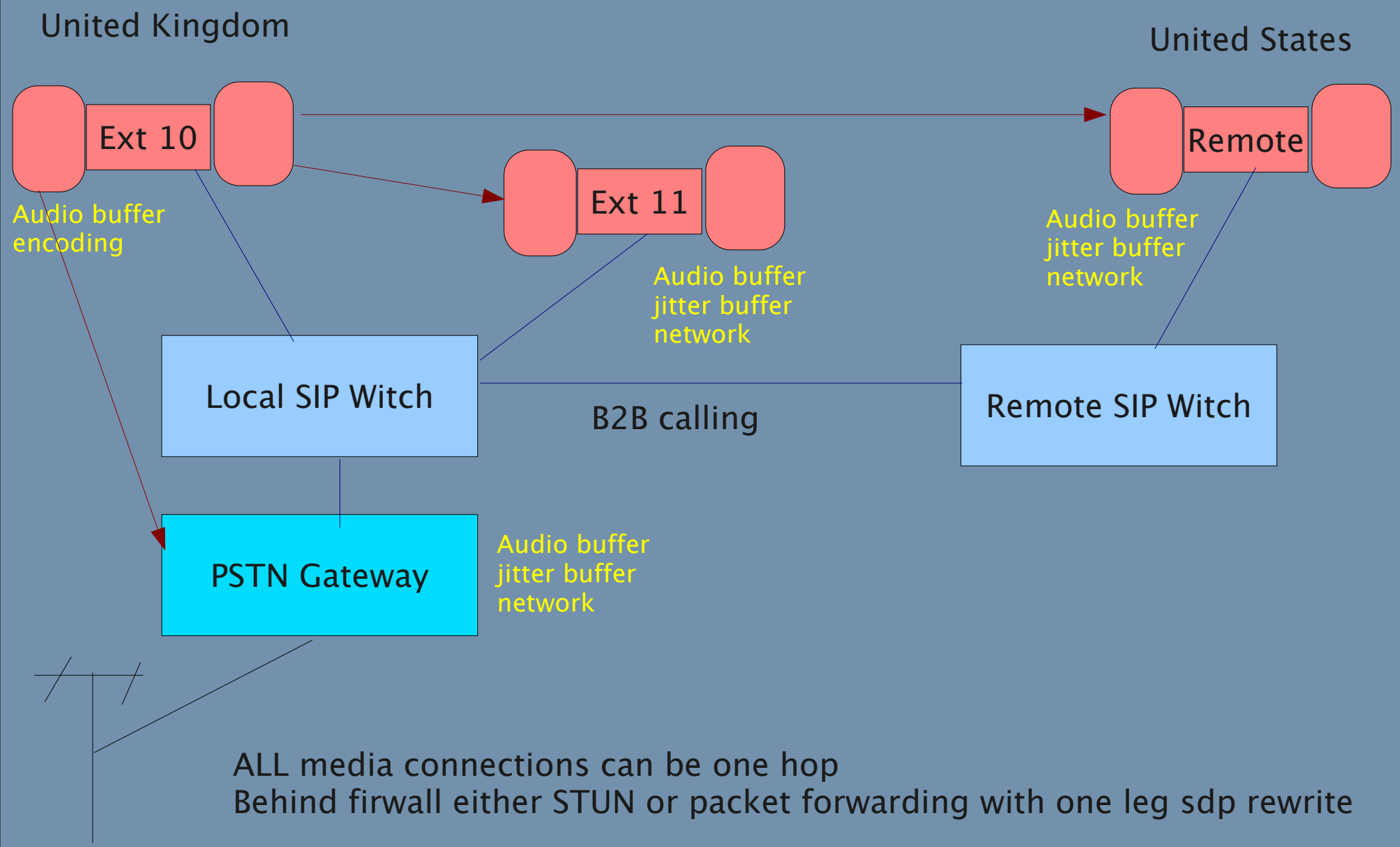


Encoding Latency: high compression codecs often use look-ahead
Network Latency: receive time delay over network
Jitter Latency: additional buffering for reordering delayed packets
Multiple hops through server(s) required for media streaming

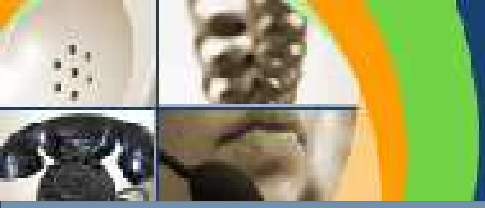


GNU Telephony

SIP Witch & Media Latency



ALL media connections can be one hop
Behind firwall either STUN or packet forwarding with one leg sdp rewrite



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

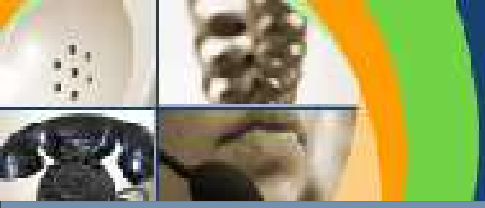
Audio path decrypted in server

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

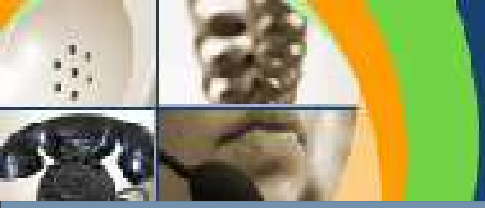
Audio path should remain encrypted in server, but what if config is falsified?

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

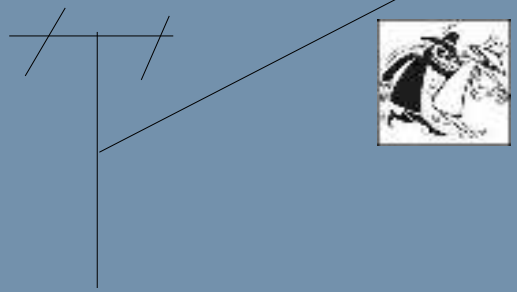
Local SIP Witch

Remote SIP Witch

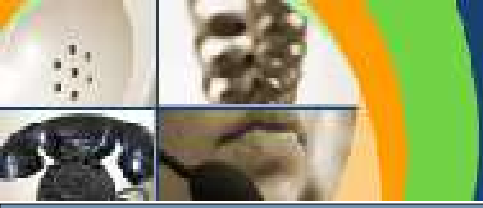
No audio to centrally decrypt

No audio to centrally decrypt, no media interconnect

PSTN Gateway

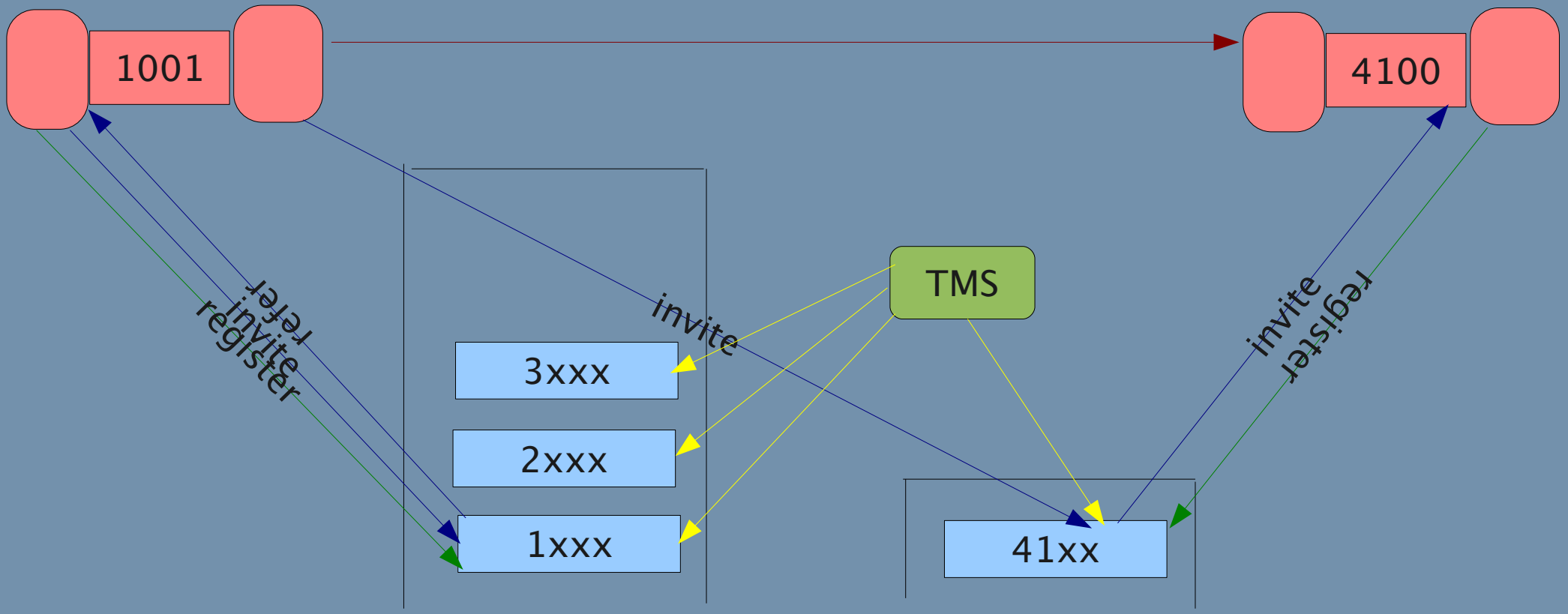


PSTN gateway path may be secure but destination is not but clear boundries between secure & insecure domains

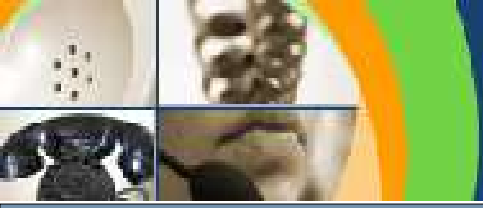


GNU Telephony

Scaling with refer routing

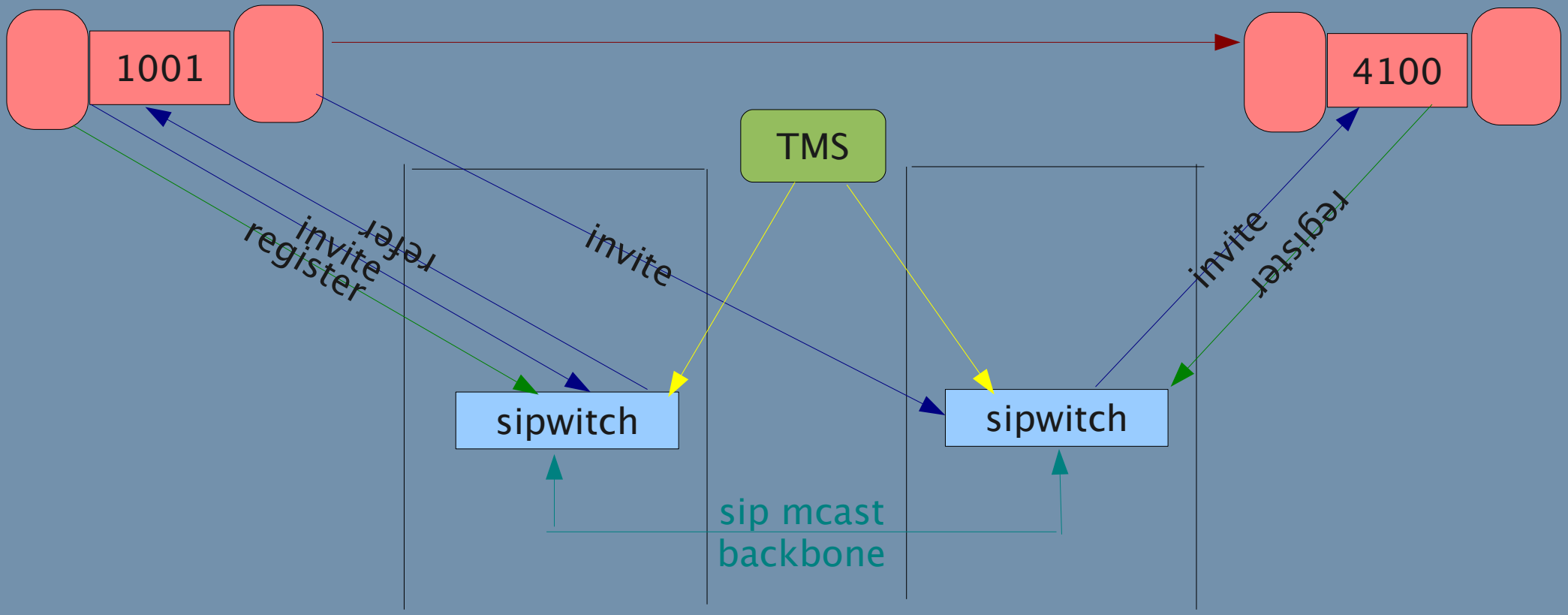


Basic satellite office and multi-switch scenario with sip witch.
Each location has unique routing maps to find other nodes.
Common TMS for all users loaded to all sipwitch call nodes.

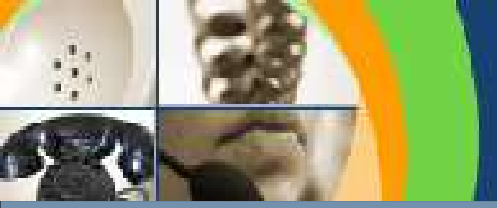


GNU Telephony

Multicast refer routing



From large campus to public Internet scalability
Registration info shared between sipwitch nodes over multicast.
Shared authentication database mirrored to each call node.
Invites referred to switch where destination is registered..



GNU Telephony

SIP Security

SIP and traffic analysis:

SIP content may be data mined. Smart use of headers and non-revelatory configurations reduce knowledge to X connected with Y. observation of ZRTP traffic also reveals IP address X connected with IP address Y. Staged realtime rtp reflectors and anonymous proxys can further reduce value of SIP & core IP data mining.

SIP is not realtime:

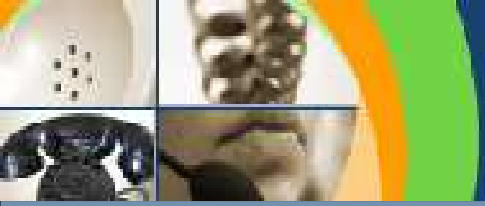
SIP can be hidden in tor/onion routers or dark freenets. Latency much less important for signaling path.

SIP can also be encrypted:

Only offers transport security. Endpoint or server still decrypts SIP messages. Goal is to be non-revealing of useful information.

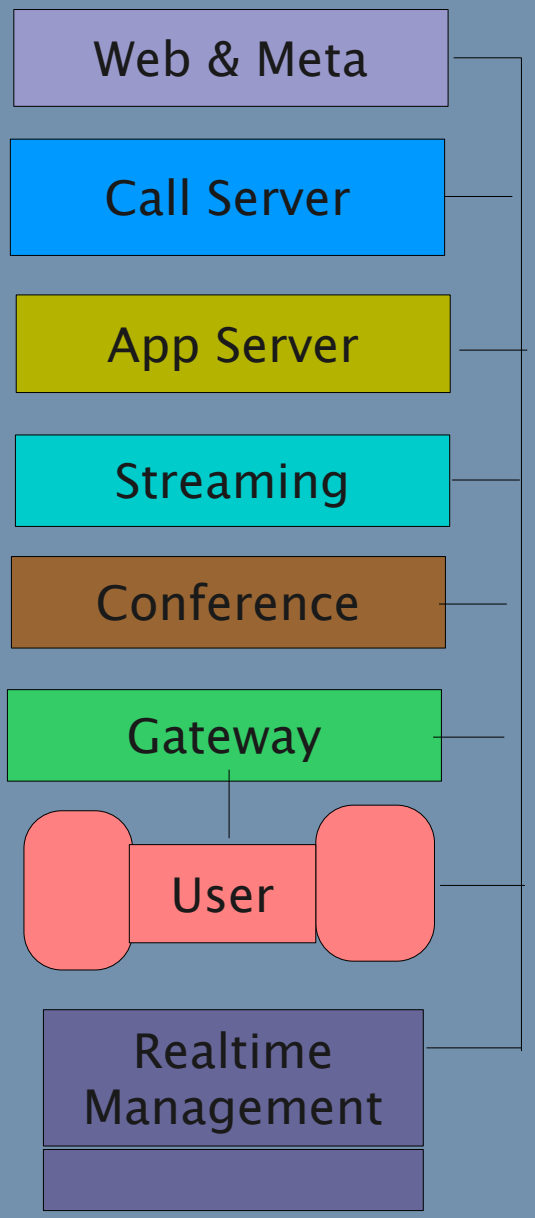
Summery of why SIP Witch

* Lower latency / less call hops
* End-to-end media security
* Network scalable routing
* Failover support possible
* No patent/codec licensing
* Trusted domains reduce SIP transactions (if registered thru calling node, can avoid auth bounce requests)
* Better use of call management possible without preconnect (orbits, acd, multi-path, etc)

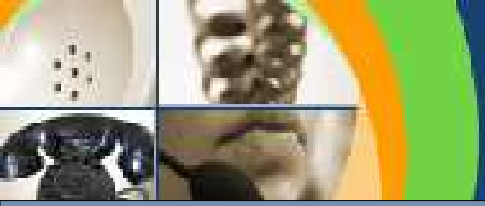


GNU Telephony

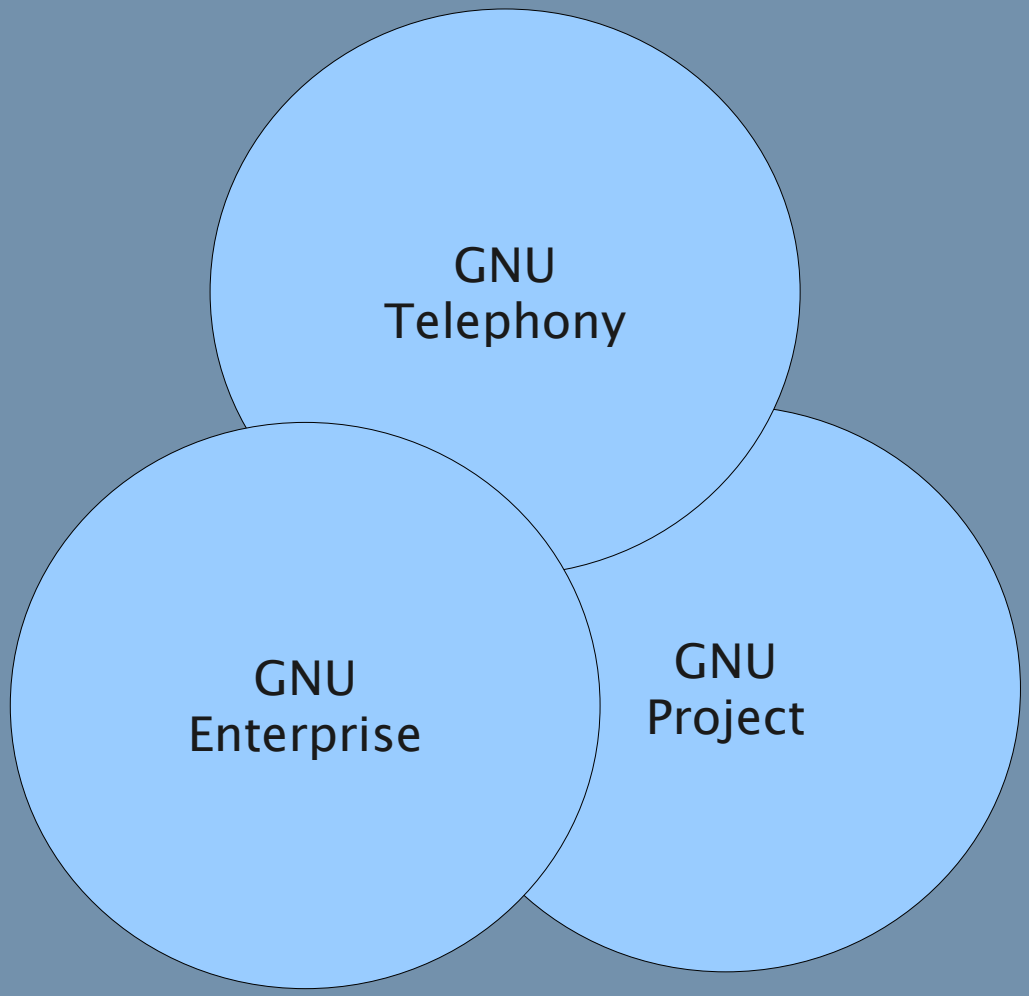
GNU Telephony Stack



* Separate functional roles, replaceable components network centric internet telephony, not IP-PBX
* Externalize meta-data such as subscribers and routes
* Common administration and control of components
* Integrate with API's for telephony hardware
* Minimize audio processing and conversions
* Use traffic engineered scripting engine for appservers
* Apply free software compatible licensing everywhere
* Closely integrate with other projects (gnue, phpgw)
* ALL voip media paths ZRTP capable
* All other ip transport paths (including sip) tls capable



GNU Telephony
Components



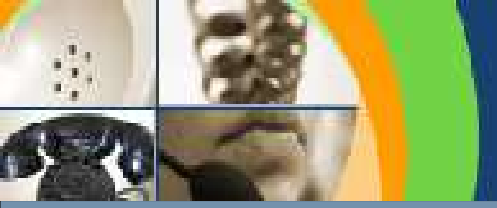
Other Projects: Twinkle Softphone, SFLPhone, fox-toolkit, hoard allocator

Libraries:

* GNU Common C++/uCommon
* GNU ccAudio2 (adding codecs)
* GNU RTP Stack (zrtp support)
* GNU ZRTP4J (new :)

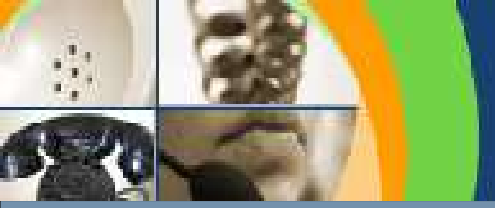
Servers:

* Bayonne Application Server
* Bayonne2 IP PBX
* GNU SIP Witch
* More to come...



GNU Telephony
Roadmap

GNU RTP	-> secure VOIP clients (twinkle) secure softphones & devices secure handheld VOIP (gpe based)
ucommon/sip	-> secure application server (bayonne) scalable gateways (bayonne2/troll) secure media conferencing secure telephone systems (sipwitch)
zrtp4j	-> portable secure clients (sip communicator) secure voip on mobile devices
gnue, boa	-> tms / common management systems
other research	-> constant traffic tunnels (tstunnel) 3d audio positioning in multiparty conferencing community telecenters (bayonne2/sipwitch)



GNU Telephony
http://www.gnutelephony.org
mailto:dyfet@gnutelephony.org

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