

# Developing rich VoIP SIP applications with SIPSIMPLE SDK

```
with nogil:
    pj_mutex_unlock(lock)
return 0

def send_invite(self, FromHeader from_header not None, ToHeader to_header not None,
                SDPSession sdp not None, Credentials credentials=None, list extra_headers=None):
    cdef int status
    cdef pj_mutex_t *lock = self._lock
    cdef pjmedia_sdp_session *local_sdp
    cdef pjsip_cred_info *cred_info
    cdef pjsip_dialog **dialog_address
    cdef pjsip_inv_session **invite_session
    cdef pjsip_route_hdr *route_set
    cdef pjsip_tx_data *tdata
    cdef PJSIPUA ua
    cdef PJSTR contact_header_str
    cdef PJSTR from_header_str
    cdef PJSTR to_header_str
    cdef PJSTR request_uri_str

    ua = _get_ua()

    with nogil:
        status = pj_mutex_lock(lock)
    if status != 0:
        raise PJSIPError("failed to acquire lock", status)
    try:
        dialog_address = &self._dialog
        invite_session = &self._invite_session
        route_set = <pjsip_route_hdr *> &self._route_set

        if self.state != None:
            raise SIPCoreInvalidStateError('Can only transition to the "outgoing" state from the "None" state,
            if timeout is not None and timeout <= 0:
                raise ValueError("Timeout value must be positive")

        route_header.uri.transport = new(credentials) if credentials is not None else None
        # always send lr parameter in Route header
```

## What is SIPSIMPLE SDK?

- Framework to develop rich SIP applications
- Rich SIP applications?
  - HD audio, MSRP chat, file transfer, desktop sharing, presence, etc.
- Written in Python
  - Core written in C + Python (Cython)
- Flexible
  - Suitable for server and client applications



## What is SIPSIMPLE SDK? (2)

- Extensible
  - High level APIs for all expected functionality
- Standard
  - RFC compliant
- Cross platform
  - Windows, GNU/Linux, Mac OSX (10.5 or higher)
- **Open Source: LGPL**



# Feature set



## Features (VoIP)

- HD audio (G722, speex)
- Secure RTP (SRTP)
- NAT traversal support with ICE
- Multi-party audio conferencing (builtin)



## Features (Instant Messaging)

- Chat sessions over MSRP
- Simple messaging with SIP MESSAGE
- TLS support for MSRP
- NAT traversal using MSRP relay extension or ACM
- Support for subscribing to 'conference' event



## Features (File Transfer & Desktop Sharing)

- File transfer over MSRP stream
- Desktop sharing using VNC over MSRP
  - Remote control!

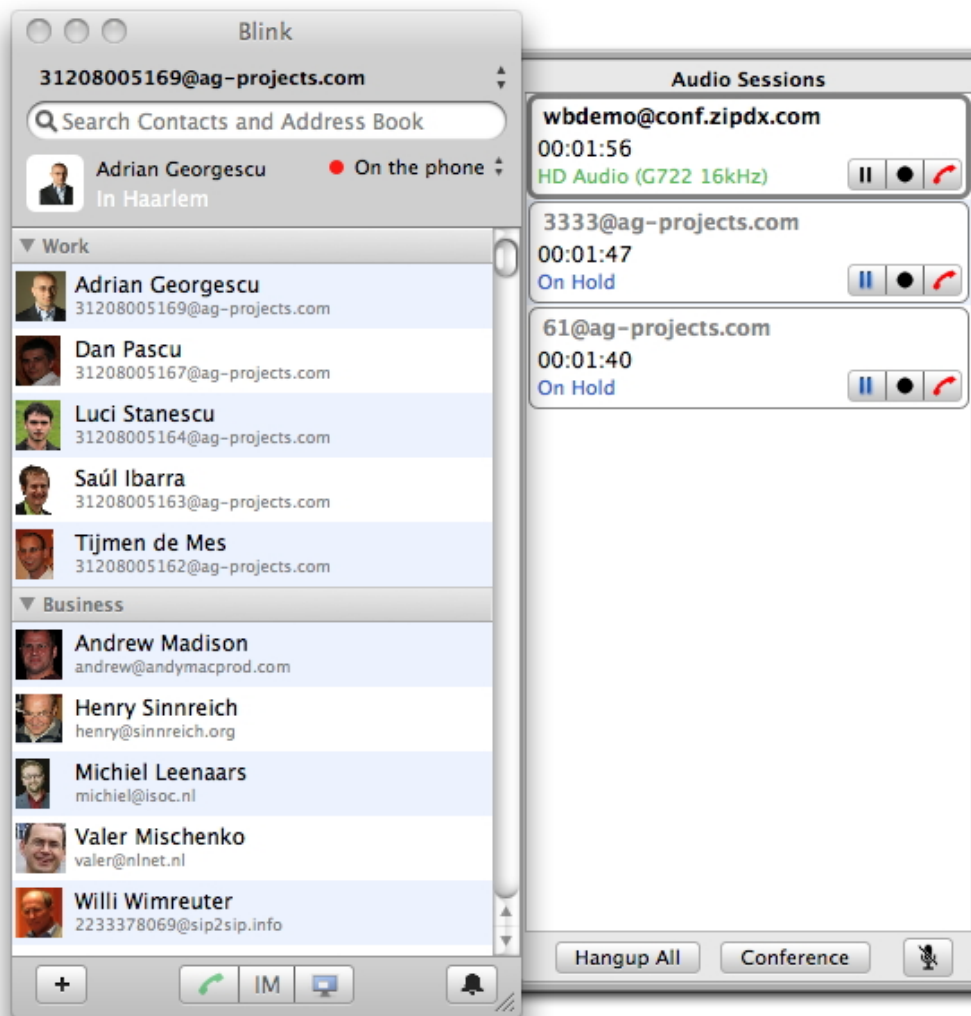


# Implementation examples

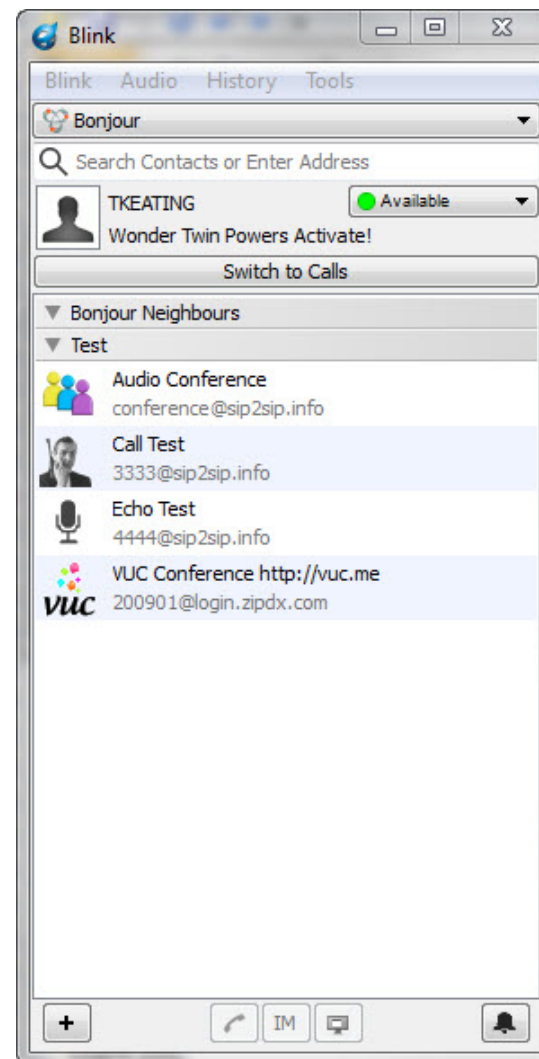
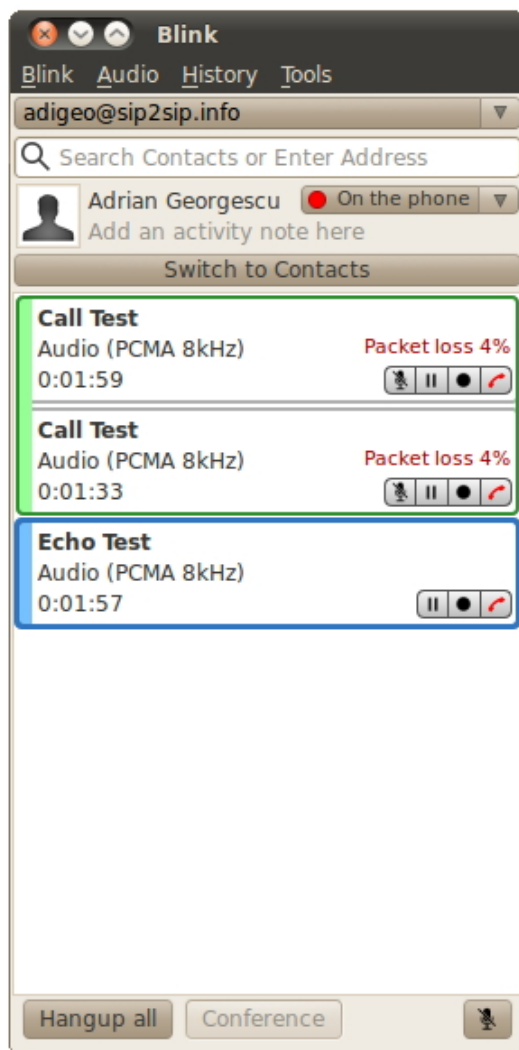




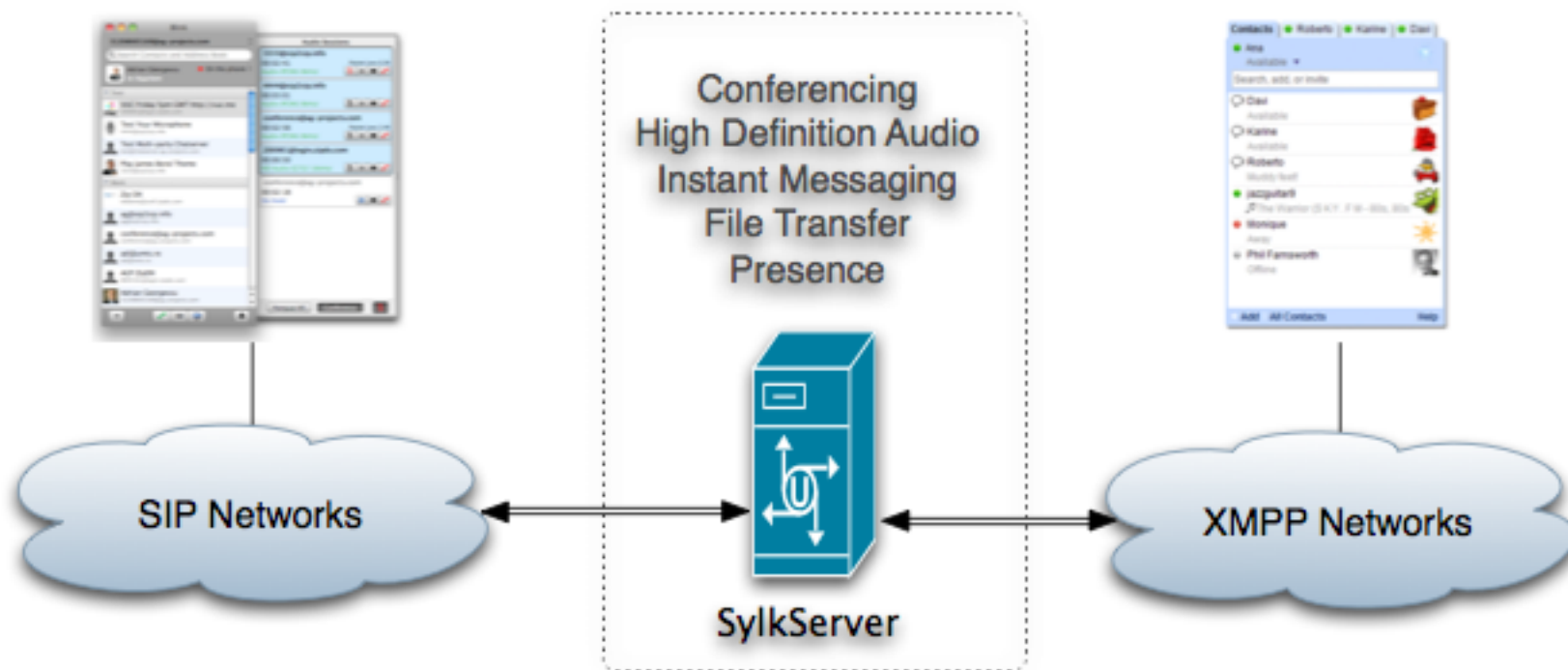
# Implementation example: Blink Cocoa



# Implementation example: Blink Qt



# Implementation example: SylkServer



## SylkServer birth

Lets fix the chatserver and bring it up to date.



That will require almost a complete rewrite!

Then lets improve it: add audio support, subscriptions to the conference event, other applications...



Talk is cheap.  
Show me the code.



# SylkServer application skeleton

```
1  from application.python.util import Singleton
2  from sylk.applications import ISylkApplication, sylk_application
3  from zope.interface import implements
4
5  @sylk_application
6  class SkelApplication(object):
7      __metaclass__ = Singleton
8      implements(ISylkApplication)
9
10     __appname__ = 'skelapp'
11
12     def __init__(self):
13         pass
14
15     def incoming_session(self, session):
16         # Handle incoming INVITE session
17         pass
18
19     def incoming_subscription(self, subscribe_request, data):
20         # Handle incoming SUBSCRIBE
21         pass
22
23     def incoming_sip_message(self, message_request, data):
24         # Handle incoming MESSAGE
25         pass
26
27
```

## SylkServer application: James Bond

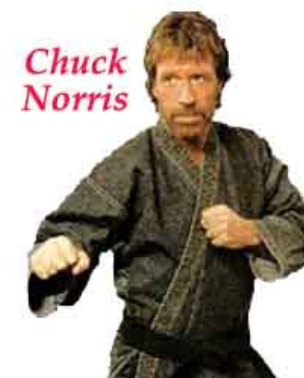
- Reject call if no audio is proposed
- Answer the call immediately
- Play the James Bond theme song

**007**



# SylkServer application: Chuck Norris Messenger

- Get a bunch of nice Chuck Norris Facts
- Reply with a random one for each incoming SIP MESSAGE





# SylkServer application: Chuck Norris Messenger 2

- Get a bunch of nice Chuck Norris Facts
- Handle incoming sessions with MSRP chat
- Reply with a random Chuck Norris fact to any message received



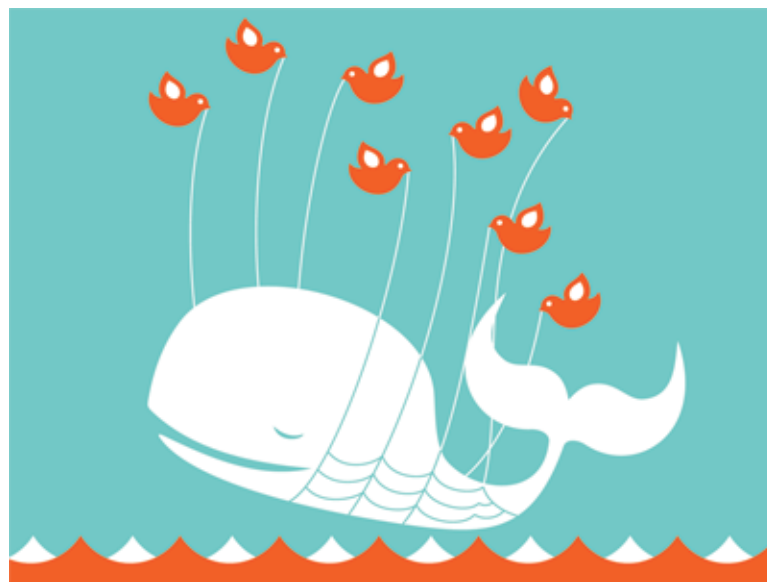
## SylkServer application: AudioConf

- Single room audio conference
- Bridge everyone into the same room when they join



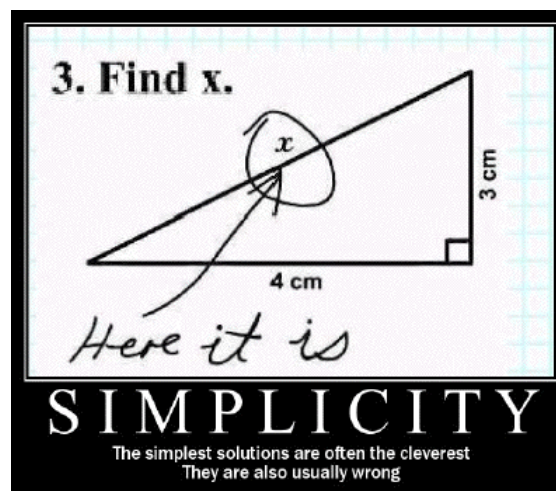
## SylkServer application: Twitter Search

- Chat `something@server`
- Get a big chat message with the last 15 tweets with 'something' on them



## SIP client: Hello World 1

- Simple SIP client which calls to [3333@sip2sip.info](mailto:3333@sip2sip.info)
- Uses the default Bonjour account (always present)
- Simplest possible example



## SIP client: Hello World 2

- Extension of helloworld1
- Call any SIP URI
- Hangup
- CLI style interface
- Also very simple

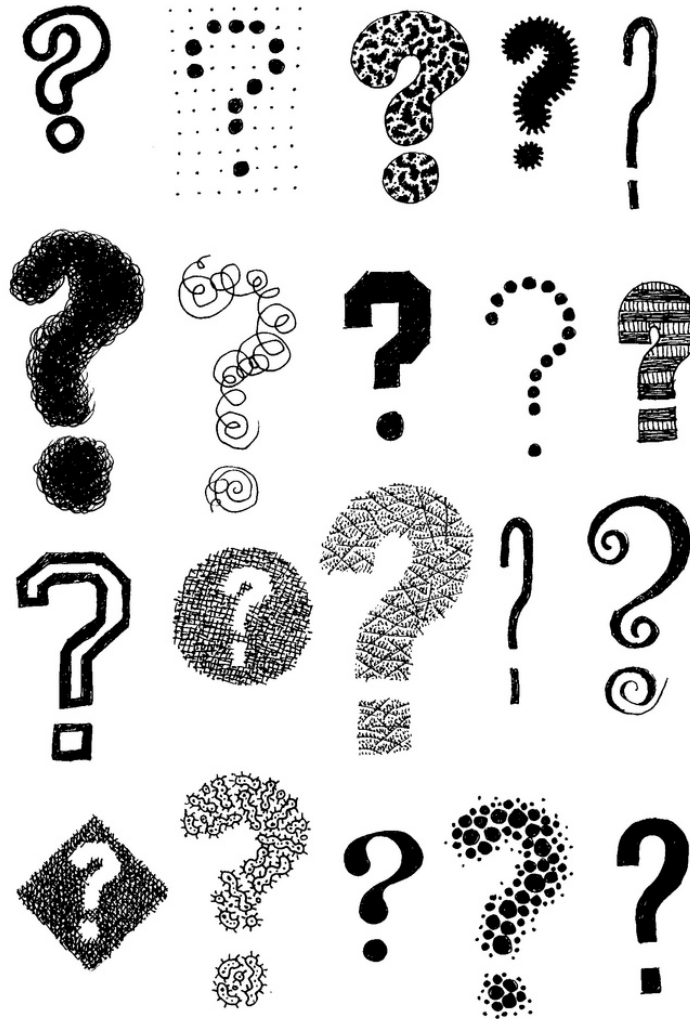


## Resources

- SIPSIMPLE SDK (LGPL)
  - <http://sipsimpleclient.com>
- Blink SIP client (GPLv3)
  - <http://icanblink.com>
- SylkServer SIP application server (GPLv3)
  - <http://silkserver.com>
- Code used in this presentation (GPLv3)
  - <http://github.com/saghul/sipsimple-examples>



# Questions?



# BYE

## BYE sip:audience@fosdem2011 SIP/2.0

Via: SIP/2.0/UDP 192.168.99.23:49919;rport;branch=z9hG4bKPjDb30Dx0sH-ozn9QB.cCCboyU.atR97aM  
Max-Forwards: 70  
From: "saghul" <sip:saul@ag-projects.com>;tag=UCpGKVZbQQx7BUKYtiuPEX668oa9jaU7  
To: <sip:audience@fosdem2011>;tag=as59aef35c  
Call-ID: DEWDfu63OACwYeQk7MrhmRhRq.1cqqis  
CSeq: 10633 BYE  
Route: <sip:81.23.228.129;lr;ftag=UCpGKVZbQQx7BUKYtiuPEX668oa9jaU7;did=641.a8a9c553>  
User-Agent: blink-0.20.2  
Content-Length: 0

twitter



@saghul



saul@ag-projects.com



sip:saul@ag-projects.com

