Telepathy

Writing a IM/Voip client in 20 lines of python

Raphaël Slinckx
A good client only needs...
A good client only needs...

...a few simple things
A good client only needs...

Link Local

...a few simple things
A good client only needs...

Link Local

...a few simple things

File Transfer
A good client only needs...

Link Local

Video Calls

File Transfer

...a few simple things
A good client only needs...

- Link Local
- Video Calls
- NAT Traversal
- File Transfer

...a few simple things
A good client only needs...

- Link Local
- OSCAR/AIM
- Video Calls
- MSN
- File Transfer
- NAT Traversal

...a few simple things
A good client only needs...

- Link Local
- Presence
- MSN
- Contact List
- Smileys
- NAT Traversal
- OSCAR/AIM
- Video Calls
- File Transfer

...a few simple things
A good client only needs... a few simple things
A good client only needs...

Link Local
OSCAR/AIM
Presence
Video Calls
MSN
Contact List
Smileys
NAT Traversal
SIP
Yahoo!
Voice Calls
Jabber
Nudges!!
File Transfer
...a few simple things
Each project reinvents the wheel.
Each client lives in its own tiny world.
Telepathy

WTF??
I. D-Bus API specification
2. An ecosystem
The big picture

- Chat UI
- Logger
- VoIP UI
- D-Bus
- SIP Backend
- XMPP Backend
The big picture

- Chat UI
- Logger
- VoIP UI
- SIP Backend
- XMPP Backend

Telepathy
Advantages
Keep the butterfly trapped
Connections are shared

Connection user@gmail.com

DBus

Abiword
Chat
Desktop
Logger
Do one thing and do it well
Language independence
Keeps the programmers happy
Language independence
Keeps the programmers happy
License independence
Keeps the lawyers happy
Object-oriented specification
Telepathy

1. The spec
Object hierarchy

Connection Managers

Connections

Channels

XMPP CM

SIP CM

user@gmail.com

contact@jabber.org

+3212345

Text with foo@ gmail.com

Text with bar@ gmail.com

StreamedMedia with sip:baz@sip.be
Connection Manager
Connection

user@gmail.com connected on Google Talk
Presence Interface
Get / Set own presence. Get contacts’ presence.
Capabilities Interface

Set own capabilities. Has video? Has voice?
Channels

user@gmail.com chatting with contact@gmail.com
Contact List Channel

Group Interface

Get / Add members. Members changed signal
Text Channel

Password Interface

Join password protected IRC chatrooms
StreamedMedia Channel

DTMF Interface

Emit key tones on a SIP conversation
2. The ecosystem
Connection Managers

Gamble
Jabber
XMPP

Salut
Link Local
XMPP
(bonjour)

SIP

Idle
IRC
Libraries

**telepathy-python** Implement CM and clients in python

**libtelepathy** Implement clients in C+glib

**telepathy-glib** Implement CM in C+glib

**libempathy** Higher level objects to build clients

**libempathy-gtk** GTK widgets to build graphical clients
Desktop integration
Desktop integration
Mission Control
Mission Control

→
Connection Managers

→
Connections

→
Channels

XMPP CM

SIP CM

user@ gmail.com

contact@ jabber.org

+3212345

Text with foo@ gmail.com

Text with bar@ gmail.com

StreamedMedia with sip:baz@sip.be
Connection aggregation
Change my presence from offline to online
Account management

Store account credentials centrally. Clients can connect without asking for config.
Channel dispatch
Open a chat window talking to user@gmail.com
Desktop integration

Empathy
Gossip

Jabber-only

Monolithic
Gossip Telepathy

Any Telepathy protocol
Monolithic
+VoIP support
Empathy

Any Telepathy protocol
Small components
Same great UI.
A lot less hairy code.
Desktop integration
Summer of Code
Jokosher Integration
Contacts as instruments
Live radio interviews with VoIP
Michael Sheldon
VoIP/Video Widgets

Augment libempathy(-gtk) with VoIP related widgets

Elliot Fairweather
File Transfer

Add link-local file transfer to empathy

Marco Barisione
Telekinesis
Easy file transfers over a local network
Marco Barisione would like to send you a file

Contact ID: demian@cordof

Do you want to accept this file?

Name: logo.png
Size: 14.4 KB
Location: Desktop

[Yes]  [No]
Transfer complete

Contact ID: demian@cordof

The file has been transferred successfully.

Name: Document.txt
Size: 42 bytes

100 %
Collaborative applications

Telepathy Tubes
If I can chat with a contact, why can’t applications?
What’s a tube?

Arbitrary data exchange

TCP/UDP/D-Bus behavior

Perform NAT traversal
D-Bus Tubes

“Virtual” D-Bus

Abiword

Internet

GetText()
Stream Tubes

“Virtual” TCP/UDP Connection

VNC Server

Internet

Socket

Socket

Socket

Socket

VNC Viewer

VNC Viewer

VNC Viewer
Stream Tubes

“Virtual” TCP/UDP Connection

VNC Server

CM

Protocol Connections

CM

CM

CM

Socket

Socket

Socket

Socket

VNC Viewer

VNC Viewer

VNC Viewer

VNC Viewer
Where to use Tubes?

If an application can be used through a D-Bus API
Where to use Tubes?

If an application can be used through a D-Bus API, it can be shared by exposing the API on a D-Tube.
Where to use Tubes?
If an application can be used through a D-Bus API
It can be shared by exposing the API on a D-Tube
Pick a contact, start collaborating!
Hi there world of One Laptop Per Child collaboration!

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Lovely.... :)

Hi there world of One Laptop Per Child collaboration!

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Lovely.... :)

Lovely.... :)
import gtk, empathy

def on_contact_notify(contact):
    print "Alias:" , contact.get_name()
    print "Presence:" , contact.get_status()

def on_contact_added(manager, contact):
    contact.connect('notify', on_contact_notify)

manager = empathy.ContactManager()
manager.setup()
manager.connect("contact-added", on_contact_added)
contacts = manager.get_members()
for contact in contacts:
    on_contact_added(manager, contact)

gtk.main()}
Code Samples

libempathy-gtk

```python
import gtk, empathy, gtkempathy

manager = empathy.ContactManager()
manager.setup()

store = gtkempathy.ContactListStore(manager)
view = gtkempathy.ContactListView(store)
view.show()

w = gtk.Window()
w.add(view)
w.show()

gtk.main()
```
import telepathy, dbus

bus = dbus.Bus()
mc = bus.get_object(
    'org.freedesktop.Telepathy.MissionControl',
    '/org/freedesktop/Telepathy/MissionControl')

mc_presence = dbus.Interface(mc,
    'org.freedesktop.Telepathy.MissionControl')

def on_presence_changed(presence):
    print "Presence changed to: ", presence

mc_presence.connect_to_signal("PresenceStatusActual", on_presence_changed)
mc_presence.SetPresence(PRESENCE_AVAILABLE, "At GUADEC")
import telepathy, dbus
bus = dbus.Bus()
reg = telepathy.client.ManagerRegistry()
reg.LoadManagers()

mgr = reg.GetManager("gabble")
name, path = mgr[CONN_MGR_INTERFACE].RequestConnection("jabber", "user@gmail.com")
connection = telepathy.client.Connection(name, path)
connection[CONN_INTERFACE].Connect()

contact = connection[CONN_INTERFACE].RequestHandles(
    CONNECTION_HANDLE_TYPE_CONTACT, 
    ["user@gmail.com"])][0]
telepathy-python / VoIP and Video call

```python
path = connection[CONN_INTERFACE].RequestChannel(
    CHANNEL_TYPE_STREAMED_MEDIA, CONNECTION_HANDLE_TYPE_NONE, 0, True)
channel = Channel(conn.service_name, path)

se = bus.get_object(
    'org.freedesktop.Telepathy.StreamEngine',
    '/org/freedesktop/Telepathy/StreamEngine')
se_handler = dbus.Interface(se,
    'org.freedesktop.Telepathy.ChannelHandler')

se_handler.HandleChannel(
    connection.service_name, connection.object_path,
    CHANNEL_TYPE_STREAMED_MEDIA, channel.object_path,
    CONNECTION_HANDLE_TYPE_NONE, 0)

channel[CHANNEL_INTERFACE_GROUP].AddMembers([[contact], ""])

channel[CHANNEL_TYPE_STREAMED_MEDIA].RequestStreams(
    contact, [MEDIA_STREAM_TYPE_AUDIO, MEDIA_STREAM_TYPE_VIDEO])
```
# When connected
connection[CONN_INTERFACE_CAPABILITIES].AdvertiseCapabilities(
    [(CHANNEL_TYPE_STREAMED_MEDIA, CREATE|INVITE)], [])

# On incoming channel of type StreamedMedia
pending = channel[CHANNEL_INTERFACE_GROUP].GetLocalPendingMembers()
channel[CHANNEL_INTERFACE_GROUP].AddMembers(pending, "")
room = conn.RequestHandles(
    CONNECTION_HANDLE_TYPE_ROOM,
    ["test@conference.olpc.collabora.co.uk"])[0]

tube = conn.request_channel(
    CHANNEL_TYPE_TUBES, CONNECTION_HANDLE_TYPE_ROOM,
    room, True)

id = tube[CHANNEL_TYPE_TUBES].OfferTube(

addr = tube[CHANNEL_TYPE_TUBES].GetDBusServerAddress(id)
p2pbus = _dbus_bindings.Connection(addr)