Presence/Collaboration: architecture and state of implementation

Guillaume Desmottes
(guillaume.desmottes@collabora.co.uk)

Collabora Ltd. (www.collabora.co.uk)

18 November 2008

Plan

- Introduction
  - About Me
  - Use cases
  - Requirements
- Telepathy
  - Introduction
  - D-Bus API
  - Implementation
- Collaboration in Sugar
  - Tubes
  - Activity sharing

About Me

Collabora

Use case: children at school

- Use school’s network
- Connected and authenticated to a server
- See buddies in their school
- Collaborate together

Use case: children at school

- Use school’s network
- Connected and authenticated to a server
- See buddies in their school
- Collaborate together
  → Server oriented architecture

Use case: children under a tree

- Simple Mesh network
- No server
- See and collaborate with buddies around

Use case: children under a tree

- Simple Mesh network
- No server
- See and collaborate with buddies around
  → Server less architecture

Requirements

- See available buddies (alias, color, key...)
- See shared activities (type, title, color, participants...)
- Share and join activities
- Collaborate!

Requirements

- See available buddies (alias, color, key...)
- See shared activities (type, title, color, participants...)
- Share and join activities
- Collaborate!
  → Server **and** Server less mode

Requirements

- See available buddies (alias, color, key...)
- See shared activities (type, title, color, participants...)
- Share and join activities
- Collaborate!
  → Server **and** Server less mode
  → Need an abstraction layer

Plan

Introduction
- About Me
- Use cases
- Requirements

Telepathy
- Introduction
- D-Bus API
- Implementation

Collaboration in Sugar
- Tubes
- Activity sharing

Telepathy

`http://telepathy.freedesktop.org`

Telepathy

- Presence
- Instant Messaging
- Audio/Video
- Collaboration

Telepathy: D-Bus API

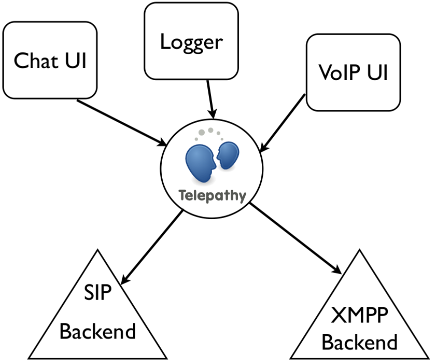
- Object Oriented D-Bus API
- API compatible
- Extensible

Example: Chat with Alice

- *conn = RequestConnection (’jabber’, {’account’: ’bob@jabber.org’, ’password’: ’foo’})*
- *conn.Connect()*
- *chan = conn.CreateChannel({’ChannelType’: ’org.freedesktop.Telepathy.Channel.Type.Text’, ’TargetHandleType’: HANDLE_TYPE_CONTACT, ’TargetID’; ’alice@wonderland.org’})*
- *chan.Send(MSG_TYPE_NORMAL, ”Hi Alice!”)*

Telepathy: Implementation

- Connection Managers: Jabber SIP, XMPP link-local, IRC, MSN, libpurple
- Clients: Empathy, Nokia Internet tablets, Fama, Kopete
- Library: telepathy-glib, telepathy-python


Chat UI
Logger
VoIP UI
Telepathy
SIP Backend
XMPP Backend

Plan

- Introduction
  - About Me
  - Use cases
  - Requirements
- Telepathy
  - Introduction
  - D-Bus API
  - Implementation
- Collaboration in Sugar
  - Tubes
  - Activity sharing

Telepathy in Sugar

- telepathy-gabble: Jabber (XMPP)
- telepathy-salut: XMPP Link-Local (Bonjour)
- Presence, Alias, BuddyProperties, BuddyActivities
- Activity properties
- Collaboration: Tubes!

Telepathy Tubes

Telepathy Tubes

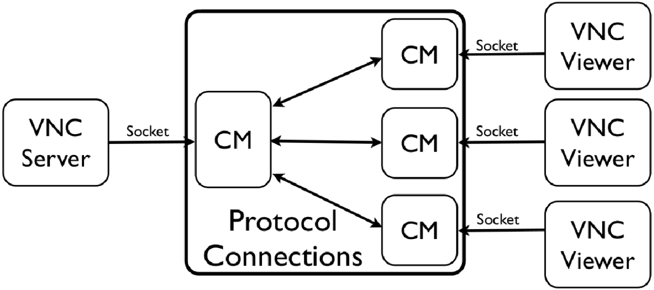
- Alice and Bob can communicate together
- Why not their applications?
- 1-1 tube
- multi users tube

D-Bus Tube

- D-Bus over network
- Participants expose objects on the (networked) bus
- Remote method call
- Catch signals
  → Pure object protocol

Stream tubes

- Initiator exports a socket
- Participants connect to CM socket
- CM relay data
- NAT traversal
- Transparent for the applications



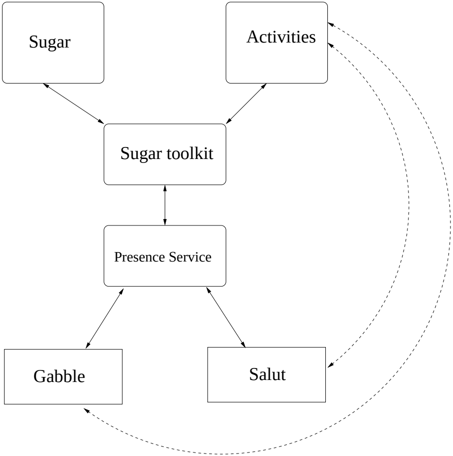
Share an activity on Sugar

- Create a room
- Offer a tube
- Activity connects to the tube
- Announce the activity

Sugar Presence Service

- Sugar specific component
- Manage accounts
- Expose Buddy and Activity objects
- Set user’s info
- Activity sharing helper

Telepathy in Sugar



Teaser: future of Collaboration

- Gadget
- Better Telepathy integration
- New Telepathy features
- Enable other Telepathy backends
- Presence Service MUST die!

Planning about Collaboration
2 PM