

Farsight 2

Videoconferencing made easy

Olivier Crête



Origins of Farsight

- aMSN
- Free IM had no VoIP
- Each proprietary IM protocol had its own thing
- Philippe's end of studies project
- Hacked into aMSN, gaim

Original goals of Farsight

- Enable Free Software IM to do audio/video like other platforms
- Abstract the streaming from different IM protocols
- Hide the complexities of media streaming (of GStreamer)

State of Farsight

- Current used in Telepathy Stream Engine
- On Nokia Internet Tablets
- Very stable
- Thin core, complex plugins
- Complex RTP plugin
- Unmaintained plugins for MSN, Yahoo

RTP plugin

- 1 to 1 audio & video calls
- Codec detection, negotiation
- Transmitters plugins
 - Unicast UDP
 - ICE (libjingle)
- DTMF
- Confort Noise (on Nokia Tablets)
- RTCP

Limitations

- Only one to one calls
- No lip-sync
- Video support broke abstraction
- Hard to use with non-trivial GStreamer pipelines
- No sRTP
- Hacks for Nokia Tablets (DSP, CN)

Farsight 2: Goals

- High level objects
- Interface, helper libraries
- RTP is reference, most standard, most capable
- Also, MSN, Yahoo, etc
- One GStreamer element per protocol
- Elegance
- Automated test coverage
- Good documentation

New RTP plugin

- Keep good things from older versions
 - Codec detection
 - Codec negotiation
 - gst elements: DTMF, CN, RTP payloaders, etc
- Use GStreamer rtpmanager
 - Multi-party
 - Lip-sync
 - Complete RTP feature set
 - Including full RTCP, SSRC collision detection, etc

Transmitters

- Multi Unicast UDP (with STUN)
- Multicast UDP
- Interactive Connection Establishment (ICE)
- Pidgeons, etc

High level objects

- Codec
- Candidate
- Participant
 - One person with synchronized streams
- Session
- Stream
- Conference

Session

- One type of media (audio, video, etc)
- One local media source
 - One microphone
 - One camera
 - File
 - etc
- Multiple stream from other participants
- RTP session

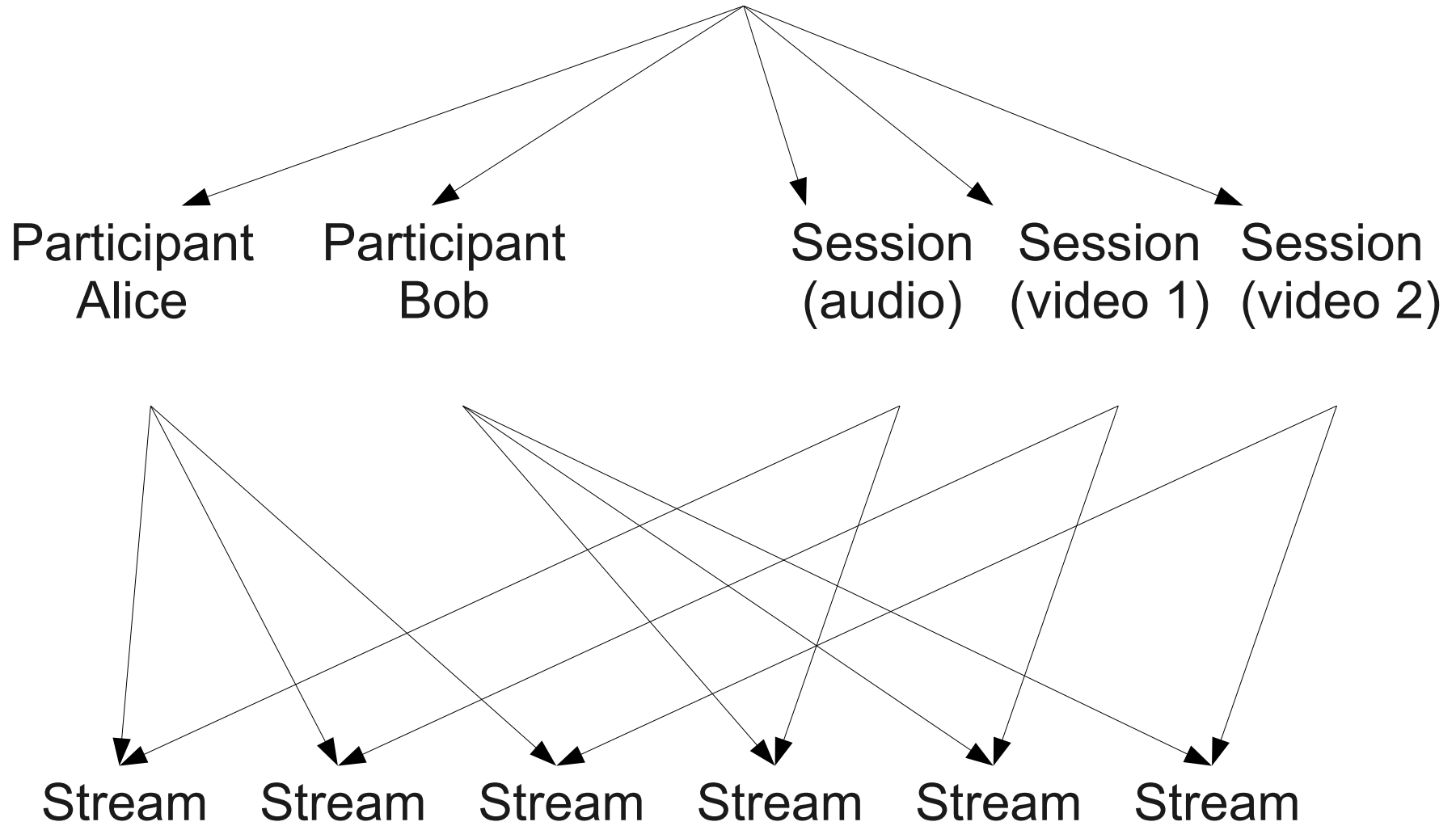
Stream

- One participant in one session
- Use for communication with participant
 - Codecs
 - Candidates
- Remote media comes out of here

Conference

- The GStreamer element
- Multiple synchronized sessions
- Contains everything else

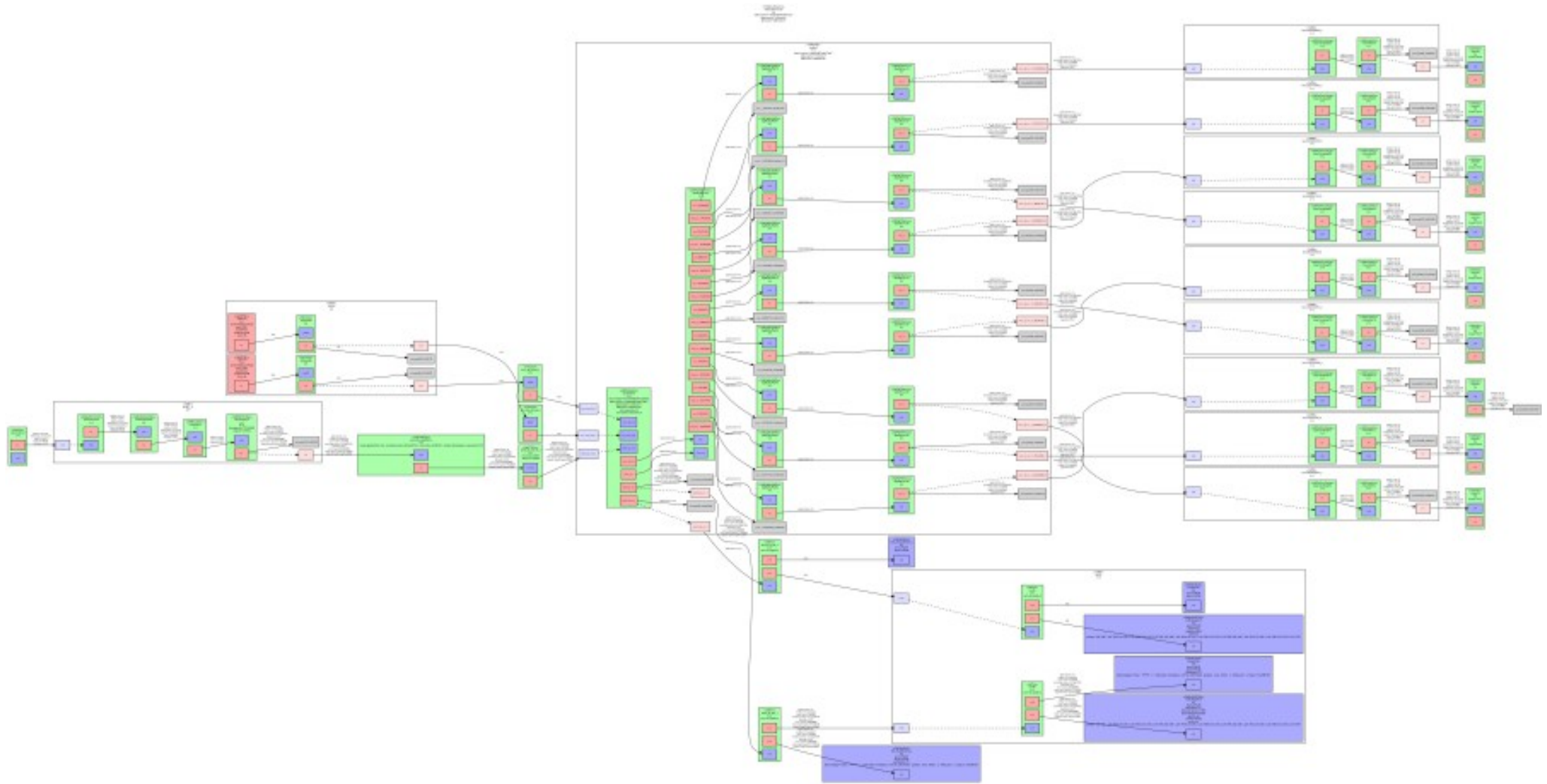
Conference



Current status

- Base RTP implementation
 - Most of Farsight 1 features, except DTMF & CN
 - Multi-party
 - Lip-sync
 - Python bindings
 - Some automated tests
 - Unicast, Multicast transmitters

10 way conference



Demo

- 3 way conference ...
- Oops ???

Example

```
import farsight, gst, gobject
```

```
loop = gobject.MainLoop()
```

```
pipeline = gst.Pipeline()
```

```
conference = gst.element_factory_make ("fsrtpconference")
```

```
conference.set_property ("sdes-cname", "tester@2.3.4.5")
```

```
pipeline.add (conference)
```

```
session = conference.new_session (farsight.MEDIA_TYPE_AUDIO)
```

```
participant = conference.new_participant ("bob@1.2.3.4")
```

```
stream = session.new_stream (participant, farsight.DIRECTION_BOTH, "multicast")
```

```
stream.set_remote_codecs(session.get_property("local-codecs"))
```

```
candidate = farsight.Candidate()
```

```
candidate.ip = "224.0.0.110"
```

```
candidate.port = 3442
```

```
candidate.component_id = farsight.COMPONENT_RTP
```

```
candidate.proto = farsight.NETWORK_PROTOCOL_UDP
```

```
candidate.type = farsight.CANDIDATE_TYPE_MULTICAST
```

```
stream.add_remote_candidate (candidate)
```

```
candidate.port = 3443
```

```
candidate.component_id = farsight.COMPONENT_RTCP
```

```
stream.add_remote_candidate (candidate)
```

```
audiosource = gst.factory_element_make ("audiotestsrc")
pipeline.add (audiosource)
audiosource.get_pad ("src").link(session.get_property ("sink-pad"))

def _src_pad_added (stream, pad, codec, pipeline):
    audiosink = gst.element_factory_make ("alsasink")
    pipeline.add (audiosink)
    audiosink.set_state (gst.STATE_PLAYING)
    pad.link (audiosink.get_pad ("sink"))

stream.connect ("src-pad-added", _src_pad_added, pipeline)

gobject.idle_add (lambda p: p.set_state(gst.STATE_PLAYING), pipeline)

loop.run()
```

The Future

- Complete RTP implementation
 - DTMF
 - Confort noise
 - sRTP
 - Stabilize
- Port Telepathy to use it
- Use it in all Free clients so they can gain AV capabilities

Thank you

- Farsight is brought to you by Collabora
 - Phillippe Kalaf
 - Youness Alaoui
 - Olivier Crête
- Questions?

<http://farsight.freedesktop.org/>

<http://git.collabora.co.uk/>

<http://www.collabora.co.uk/>

