GStreamer Audio Applications
What is GStreamer

- Graph based multimedia framework
- Open source
- Cross platform
  - Linux, Windows, MacOS, Symbian, ...
- Multiple languages
  - C, C++, C#, Java, Perl, Python, Vala, ...

http://www.gstreamer.net
What is GStreamer used for

- Media players
- Video editors
- Voip and Video call
- Music editors/composers
- Camera capture
- Transcoding
- ...

1. May 2010
Stefan Kost <ensonic@sonicpulse.de>
What is Buzztard
Concepts

gstreamer tools

gst-inspect
gst-launch
gst-editor

media player
VoIP & video conferencing
streaming server
video editor
(...)

gstreamer core framework
pipeline architecture

protocols
- file:
- http:
- rtsp:
- ...

sources
- alsa
- v4l2
- tcp/udp
- ...

formats
- avi
- mp4
- ogg
- ...

codecs
- mp3
- mpeg4
- vorbis
- ...

filters
- converters
- mixers
- effects
- ...

sinks
- alsa
- xvideo
- tcp/udp
- ...

media agnostic
base classes
message bus
media type negotiation
plugin system
utility libraries
language bindings

3rd party plugins

gstreamer includes over 150 plugins

1. May 2010
Stefan Kost <ensonic@sonicpulse.de>
GStreamer Core

- Pipeline layout
  - Elements, Bins and Pads
- Communication
  - Buffers, Events, Queries and Messages
- Scheduling
  - Tasks, Queues and Clocks
- Registry
GStreamer Core Libs

- Base Classes
  - BaseSrc, BaseTransform, BaseSink, ...
- Check
  - Testing support
- Controller
  - Parameter automation
- ...

1. May 2010
Stefan Kost <ensonic@sonicpulse.de>
GStreamer Plugins Base Libs

• Core is media agnostic
• gst-plugin-base introduces
  • Audio/Video baseclasses and utilities
  • Lots of helper libraries
Elements

- Elements with static pads

![Source Element](source_element.png)
![Filter](filter.png)
![Sink Element](sink_element.png)
Elements

- Elements with "sometimes" pads

![Diagram of elements with "sometimes" pads]

- Elements with "request" pads

![Diagram of elements with "request" pads]
Pads

- Pads are connectors
- Elements have PadTemplates
- Pads are created from PadTemplates
Caps

• PadTemplates have Caps

  audio/x-raw-float
    rate: [ 1, 2147483647 ]
    channels: [ 1, 2147483647 ]
    endianness: 1234
    width: { 32, 64 }

  audio/x-raw-int
    ...

• Pads have caps too
RAW Audio Caps

- good support for multi-channel audio
- audio is interleaved!
  - LRLRLRLR instead of LLLLLRRRRR
- Non interleaved support would be nice to have
Links

- Elements are linked by linking their pads
- It is always src -> sink
- Caps of PadTemplates need to be compatible
Bins

- Elements can be grouped
- Bins are elements themself
Bins & GhostPads

- GhostPads are proxies for internal pads
- Bin + GhostPads = hierarchical graphs
Pipeline

- Toplevel bin

Gstreamer pipeline for a basic ogg player

Pipeline

• Toplevel bin
1. May 2010
Stefan Kost <ensonic@sonicpulse.de>
Communication

- Buffers, Events, Queries & Messages
Threads

- Pads run in streaming threads
Clocks

- Elements can offer a clock
- The pipeline selects **one**
- All sinks synchronize to it
Element States

- NULL: inactive
- READY: check resources
- PAUSED: preroll
- PLAYING: active dataflow, clocks are ticking
Plugins

- Provide system or user wide features
  - /usr/lib/gstreamer-0.10/
  - /home/user/.gstreamer-0.10/plUGINS/
- Features are element factories, type finders, etc.
Registry

• Cache of plugins and the plugin features
  • Introspected from available plugins
• Automatically checked at startup
• Updated if needed
Audio Elements

• Audio I/O
  • alsa, jack, pulse, esound, oss, ...

• Tools
  • audioconvert, audioresample, adder, (de)interleave

• Effects
  • a few native (~15), ladspa, lv2, buzzmachines

• Muxers, Demuxer and Codecs
Sequencer

- GstController subsystem
- Time based parameter changes
A simple C Example

- 145 lines of C for a stupid sound sequence
- Simple pipeline
  `simsyn ! audiodelay ! autoaudiosink`
- Sequencer controls "simsyn"
Resources

  - docs, mailing-lists, ...
- irc://irc.freenode.org/#gstreamer
- http://cgit.freedesktop.org/gstreamer/
- https://bugzilla.gnome.org/browse.cgi?product=GStreamer
- http://www.buzztard.org
- irc://irc.freenode.org/#buzztard
Thank you!
Some statistics

- 10 years of development
- ~500 contributor (30 active)
GObject

• Written in C, but still OO
• We're using GObject
  • Inheritance
  • Interfaces
  • Introspectable properties
  • Signals
Logging

- Powerful logging subsystem
- Level, categories, ...
- Run apps with:

  `GST_DEBUG="*:2,GST_CAPS:4" ./myapp`
Pipeline dumps

- Dump pipelines as graphviz graphs
- Use dot to render them
- Run apps with:

  `GST_DEBUG_DUMP_DOT_DIR=$PWD ./myapp`
Tools

• gst-inspect
  • Show known plugins
• gst-launch
  • Run pipelines on the fly
Tools

- **gst-inspector**
  http://cgit.freedesktop.org/∼cymacs/gst-inspector/

- **gst-debug-viewer**
  http://cgit.freedesktop.org/∼cymacs/gst-debug-viewer/

- **gst-gengui**
  http://code.google.com/p/gst-gengui/

- **gst-tracelib**
  http://cgit.freedesktop.org/∼ensonic/gst-tracelib/
Writing own plugins

• fdo git repo has a template module
  • App template
  • Plugin template
• Or find a similar plugin and copy'n'paste
  • All plugin packages have gst/ext/sys dirs
    – Gst – no dependencies
    – Ext – external library dependencies
    – Sys – platform dependencies (kernel)