Flexible data flow framework
http://www.upipe.org/

Christophe Massiot
cmassiot@upipe.org
What is Upipe?

- Handles flows of data in a « pipeline »
- Processes them using filters called « pipes »
- Defines APIs:
  - To configure and feed data into pipes
  - To get out-of-band events from pipes
  - To store data in an efficient manner (refcounts, zerocopy) with arbitrary attributes
  - To interact with an event loop/threads
- Provides a set of basic pipes
## Potential applications

- Transcoders
- Multiplexers
- Play-out systems
- Mosaics
- Embedded, lightweight, media players
- Embedded demonstration platforms
Why yet-another-multimedia-framework?

- Existing frameworks are 15 years old; new trends emerged since:
  - Super-scalar architectures
  - Event-driven loops (à la libevent)
  - Frameworks (designed for multimedia players) are more and more used for professional applications, for which a single high-level API is not convenient

- Maintenance made more difficult by:
  - Lack of modularity, complexity
  - Confusion between processing vs. decision
Developing Upipe

• Started 18 months ago with new principles:
  • Specified bottom-up, from the simplest to the most complicated, different API levels are possible
  • All modules of code are autonomous and are unit-tested separately (including memory leaks)
  • SIMPLICITY
  • Emphasis on documentation
  • Designed for professional applications — not “media players that should play every single poorly encoded file”
  • Core under MIT, modules under GPLv2+ or LGPLv2+
Who we are

- Upipe development sponsored by OpenHeadend
  - Use Upipe in products (encoders, video processing, mosaics)
  - Provide development services on Upipe
  - Written by Christophe Massiot and Benjamin Cohen
- C. Massiot co-founded “VLC media player” 15 years ago, ran the television head-end of a large telco for 10 years
In a nutshell
Available modules for video pipelines

- Input/output: file, udp (multicast), http
- Containers: native TS demux/mux + libavformat support
- Codecs: libavcodec & x264 support
- Filters: swscale, native deinterlacing
- Display: GLX, Mac OS X audioqueue
- Other utility modules:
  - Lock-less queue between threads
  - “dup” pipe
  - “null” pipe
  - Trick play, dejitter
Example: glxplay pipeline
Example: urecordhash pipeline
Keep in touch!

http://upipe.org/