

**FLUENDO**

# **Flumotion: Streaming media with Python and GStreamer**

Thomas Vander Stichele  
thomas (at) fluendo (dot) com

*LCA 2007, Sydney / January 19<sup>th</sup> 2007*

# Calibration

- Adjusting the talk to the audience

# Overview

- Introduction
- Flumotion architecture
- internals
- new in 0.4
- making a product
- happy hacking
- the future

# Introduction

- Who is Fluendo ?
- overview of Flumotion
- overview of GStreamer and Twisted
- multimedia and Free Software
- a sample run

# Fluendo

- A **Spanish** company
- built around **GStreamer**
- focusing on:
  - streaming server and platform
  - plugins and applications for GStreamer
  - consultancy for multimedia and GStreamer
  - Elisa

# Flumotion

- A **streaming** server
- GPL or Advanced license
- written in **Python** using GStreamer, Twisted
- (currently) focused on **live** streaming
- distributed and extensible



# History

- June 29 2004: First public use, 90 Mbit/sec
- Oct 20 2004: 0.1.0
- Mar 03 2006: 0.2.0
- Dec 22 2006: 500 Mbit/sec on one stream
- Jan 19 2007: 0.4.0 (\*)

# Flumotion

- Multiple formats
- Multiple client types
- focused on usability
- low resource usage (5000+ happy viewers)
- end to end:
  - Java applet (Cortado) (\*)
  - DirectShow filters and ActiveX installer



# Flumotion features: components

- **sources:** Firewire, soundcard, videocard, webcam, test, looper
- **codecs:** Vorbis, Theora, mulaw, JPEG, smoke (but also ...)
- **consumption:** HTTP streaming, RTSP streaming, archiving
- **effects:** colorbalance, overlaying, volume

# Flumotion features

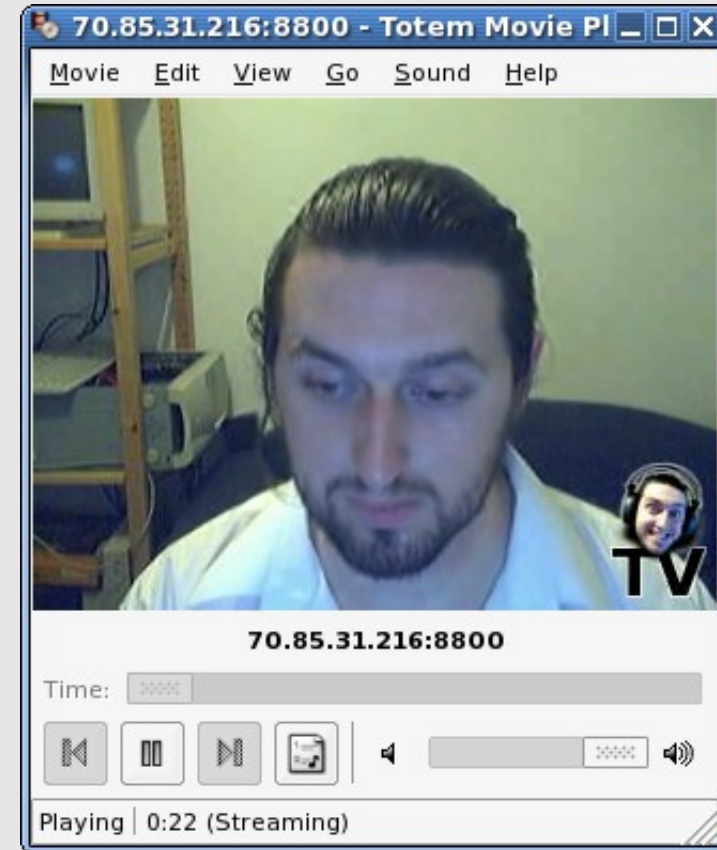
- friendly UI
- localized
- authentication
- code distribution from manager
- runtime checks of software, devices, ...

# Flumotion target users

- people and companies that want to
  - **extend** the basic use scenario
  - use media sources in **multiple** ways
  - provide media in different **codecs**
  - nice **user experience**
  - Use a **free** platform

# Flumotion users

- Conferences:
  - GUADEC
  - FUDCON
  - FISL
  - LCA (?)
- Radio stations
- A series of mosques
- RTVE
- a bunch of random people



# Twisted

- written in Python
- **event-based** framework
- central concepts:
  - **reactor** (“main loop”)
  - **deferreds** (“promise of result”)
  - **PB** (client/server protocol)
  - cooperative multitasking



# GStreamer

- written in C
- **multimedia** library
- **elements** in pipelines
- abstracting away lots of MM libraries
- **optimized** for speed
- **threaded**
- comes with bindings



# Multimedia FOSS

- patents are a **problem** for distributions
- GPL is **not compatible** with patents
  - murky wording
  - not yet tested in court
- **licensing** (and choice for Flumotion + comparison to others)
- DRM
- **support** royalty-free codecs and Xiph !

# sample run

The screenshot displays a Linux desktop environment. The main window is 'localhost:7531 - Flumotion Administration', which shows a list of components and their status. The 'http-video' component is highlighted. To the right, a terminal window shows the output of a command, including the text '[ OK ]'. In the foreground, a 'localhost:8802 - Totem Movie Player' window is open, displaying a video of a man wearing headphones. The video player interface includes a progress bar, playback controls, and a volume slider. The system tray at the bottom shows various application icons and the system clock indicating 'Fri Feb 25, 17:13'.

Mood	Component	Worker	PID
😊	audio-encoder	default	7444
😊	audio-source	default	7436
😊	disk-audio	default	7437
😊	disk-audio-video	default	7445
😊	disk-video	default	7446
😊	http-audio	default	7440
😊	http-audio-video	default	7439
😊	http-video	default	7442
😊	multiplexer-audio	default	7448
😊	multiplexer-audio-video	default	7443

```
servicer Feb 25 15:44:07
e.py:249)
5)
servicer Feb 25 15:46:32
e.py:249)
5)
default.pid
orker.default.pid'? y
default
[ OK ]
```

localhost:8802  
Movie Edit View Go Sound Help  
Fluendo  
localhost:8802  
Time: [Progress Bar]  
[Play] [Pause] [Next] [Full Screen] [Volume]  
Playing | 1:19 (Streaming)

# Flumotion architecture

- manager, worker, and admin
- components
- planets, atmosphere and flows

# processes

- one **manager** process
- one or more **workers**
  - connect to manager, receive code
  - run **components** as a forked process
- one or more **admin** clients
  - connect to manager, receive code
  - sends commands to manager
  - presents UI for components through manager
- all across the network



# components

- run inside a spawn of the **worker** (\*)
- produce/consume/convert/combine **feeds**
  - GStreamer data protocol
  - Twisted connection setup
- or provide services (authentication, testing)
- combination of
  - set of parameters
  - collection of GStreamer elements
  - wizard pages for configuration
  - administration pages

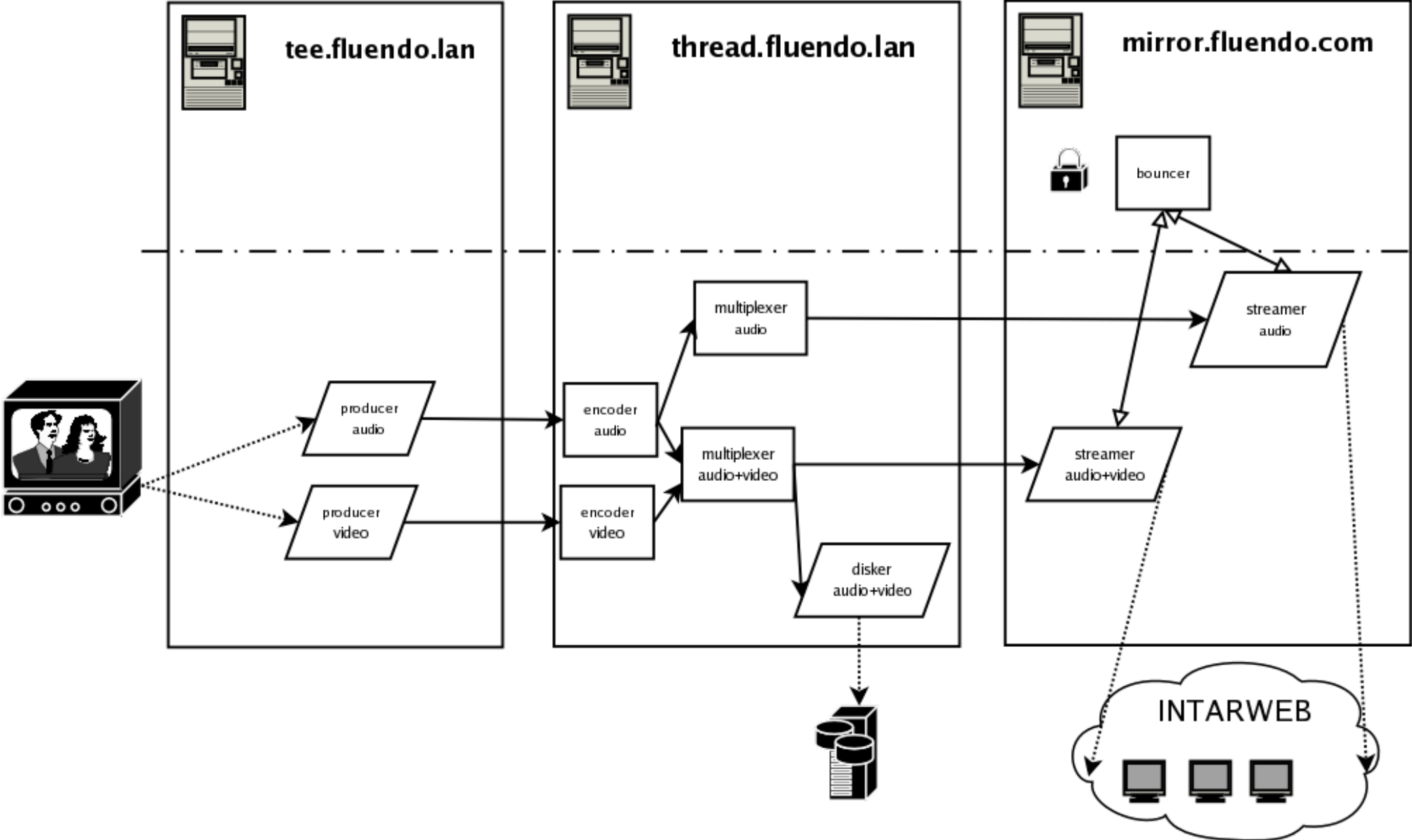


# configuration

- **planet** is managed by a manager
- contains one or more **flows**
  - separated from each other
  - containing feed components
- contains one **atmosphere**
  - containing components used by flow components



# example



# Flumotion internals

- streaming
- bundles
- wizard
- authentication
- state objects
- translatables
- Feel free to discuss these with me !

# Streaming

- connection setup in python
- our code steals the socket from reactor
- we hand the fd to GStreamer
- all actual streaming is done on the C level

# Bundles

- **partitioned module namespace** to import code from
- **transferred** from the manager
- **cached** locally in workers and admin clients
- method “registers” bundle and modifies modules so **normal import statements work**

# Bundles

- BundledFile: .hasChanged()
- Bundle: .setZip(), .getZip()
- Bundler: .add(), .bundle()
- BundlerBasket: registry of bundles
  - .add(), .depend()
  - getBundlerByName()
  - getBundlerNameByFile(), ByImport()

# Bundles

- Unbundler: `.unbundle()`
- `flumotion.common.common.registerPackagePath()`
  - go in `sys.path`
  - find package candidates
  - add to `__path__` of already loaded packages and modules



# State objects

- job processes have state from job to manager
  - pid, mood, ...
- components have state from manager to admin
  - mood, type, worker requested, ip, ...
- planet, flows, atmosphere have state -> tree
- Implements MVC for admin (\*)

# State objects

- StateCacheable/StateRemoteCache
  - .addKey(), .addListKey()
  - .get(), .set(), .append(), .remove()
- IStateListener for listeners to StateRemoteCache changes



# Keycards and bouncers

- “something” wants a service provided
- **keycards** are created: serializable
  - abstracts away
    - type
    - information
    - challenge/response
  - sent to a **bouncer** component
  - back and forth
  - can be expired by bouncer

# Keycards and bouncers

- Bouncer can be anywhere in the network
- Is very easy to write
- Specific bouncers written for customers
  - GET request and token
  - Filtered on IP
  - HTTP authentication

# Translations

- An error happens in a component
- An admin client connects to the manager after the error ...
- ... and the admin client needs to show the translated error
- How do you make gettext do this ?



# Translations

- Create a Translatable class
- That holds the format string, the arguments, and the domain
- Send this object to the admin
- Admin downloads bundles with .mo file for the language and translates locally
- show (\*)





# Wizard

- scenario-based
- pulls in code while you make choices
- checks requirements worker-side(\*)
  - tells manager to ask worker to run a **callable** from a module
  - manager forwards
  - worker gets bundle, runs and returns

# New in 0.4

- porter
- internal connection abstraction
  - one feed server/worker
- reconnection
- on demand http server
- iCal bouncer and diskier
- component and properties cleanup
- feeder information

# use cases

- picture-in-picture streams
- multi-audio streams
- live annotation using annodex
- high-res slides using Ogg + MNG + OpenOffice
- multiple codecs and bandwidths
- flash overlaying

# Flumotion future ideas

- playlist component
- live integration with PiTiVi
- screencasts
- yelp integration
- bug reporting tools
- stream preview
- almost-live translation
- other admins (web, flash, mud, ...)

# Flumotion and RTSP

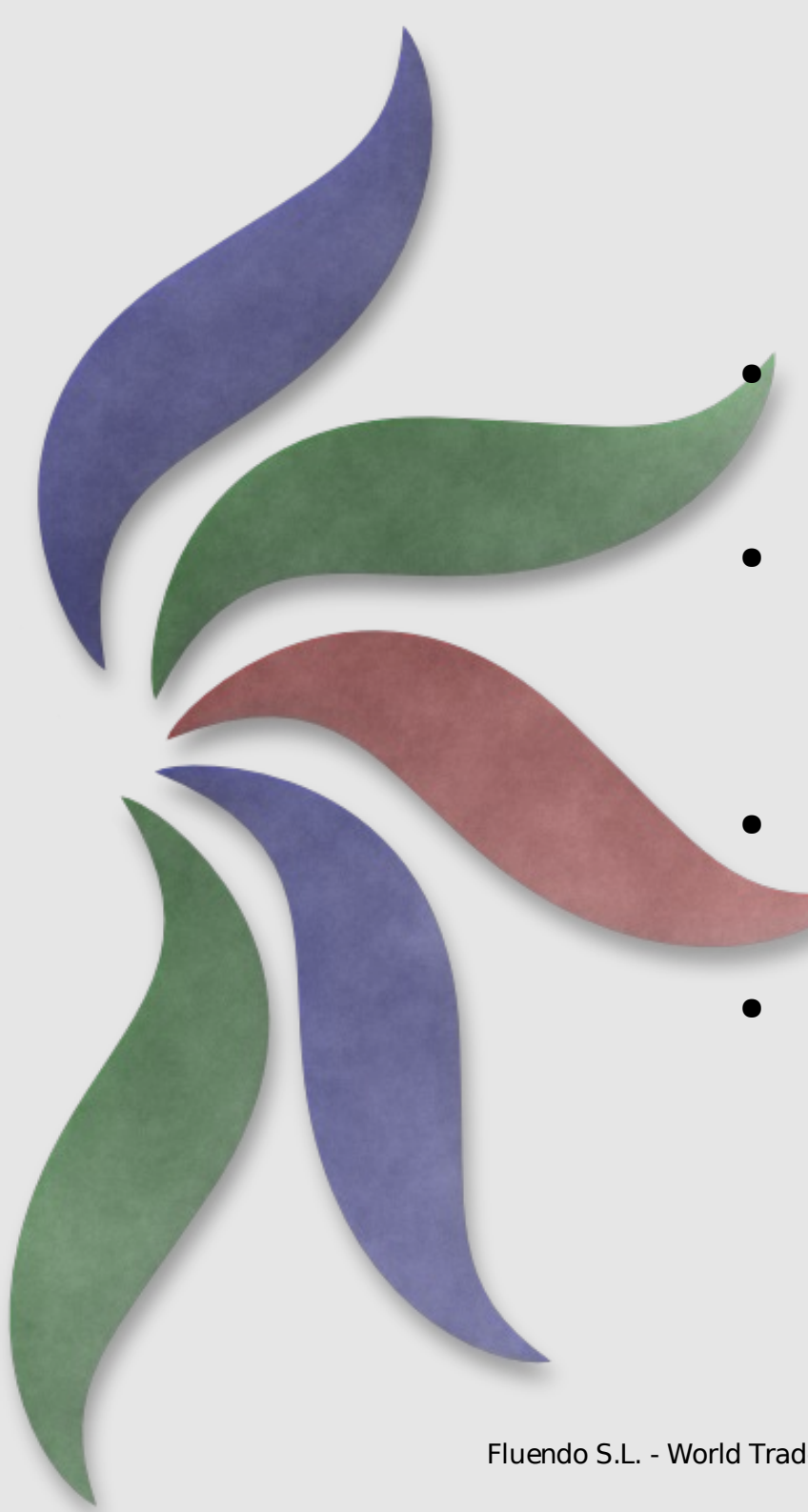
- RTSP and RTP allow for shorter startup delay
- More complex to implement
- Fluendo funded Xiph/RTP work

# Making a product

- UI standards
- focus on usability
- requirements
- documentation
- unit tests
- automated testing
- QA/release engineering



# the Python challenge

- 
- “scripting” language
  - no typing in declarations
  - dynamic constructs
  - Speed
  - **Design**
  - **document** your API properly
  - **unit tests** (example)
  - implement the most-used paths lower-level (GUADEC example)

# the Python nirvana

- great bindings
  - GTK, GStreamer
- easy prototyping
- quick development during programming (\*)
- great features
  - rebuild modules and objects on the fly
  - Twisted

# UI standards

- programmers try to avoid thinking about UI
- we try to follow the HIG
- the HIG does not talk about implementation
- unittest UI (show flumotion.test.test\_dialogs, flumotion.test.test\_greeter) ?
  - stolen from gazpacho
  - weird but works

# usability

- move to run-time checking
  - devices, elements, library versions, ... (show library version check)
- use a graphic language to describe your program
  - planets, moods, ...
- centralize all actions in one program

# requirements

- Works with GStreamer 0.8 and 0.10
- Migration path
- Continuous tension between developers and deployers
- Run-time checking relaxes build-time requirements



# Stack hacking

- A bug shows up using Flumotion
- Could be in
  - Flumotion
  - Gst-python
  - GStreamer
  - library
  - Kernel driver

# release engineering

- unittests and coverage
  - only tests object functionality to some level
- automated builds
  - make sure you catch problems early on the project level
  - <http://build.fluendo.com:8070> (do false commit in test\_common)
- test scenarios for common functionality
  - additional to unit tests

# Happy Hacking



# Flumotion template

- Sample add-on project
- Has:
  - Build infrastructure
  - Component code
  - UI code
  - Glade file
  - translations

# OLPC

- Simple install of Fedora packages
- Firmware upgrade
- flash upgrade
- some tweaking of video producer



# The future



# The future

- Flumotion
  - more codecs, components, scenarios
  - RTP (GStreamer)
  - high availability
  - hi-res slides in stream using MNG
  - Annodex: browsable media

# The future

- Fluendo
  - advanced servers
  - streaming platform
  - distributable software
  - codecs: Windows Media, Real, MP3
  - DVD player
  - building out GStreamer
  - collaborations with companies



# TODO

- upload release

# Questions ?

- thomas <at> fluendo <dot> com
- <http://www.fluendo.com/>
- <http://www.flumotion.net/>