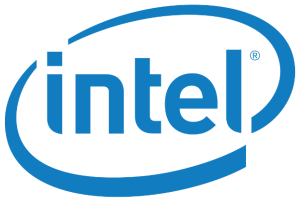


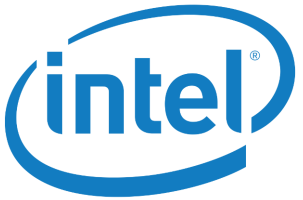
Linux Kernel

Linux features

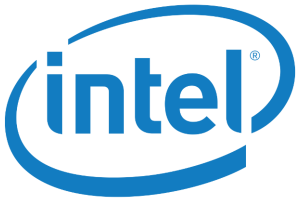
Rafal Kapela

June 26, 2016

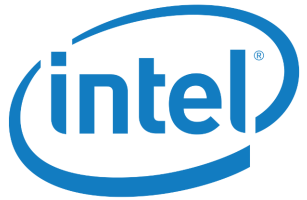
Outline



Outline



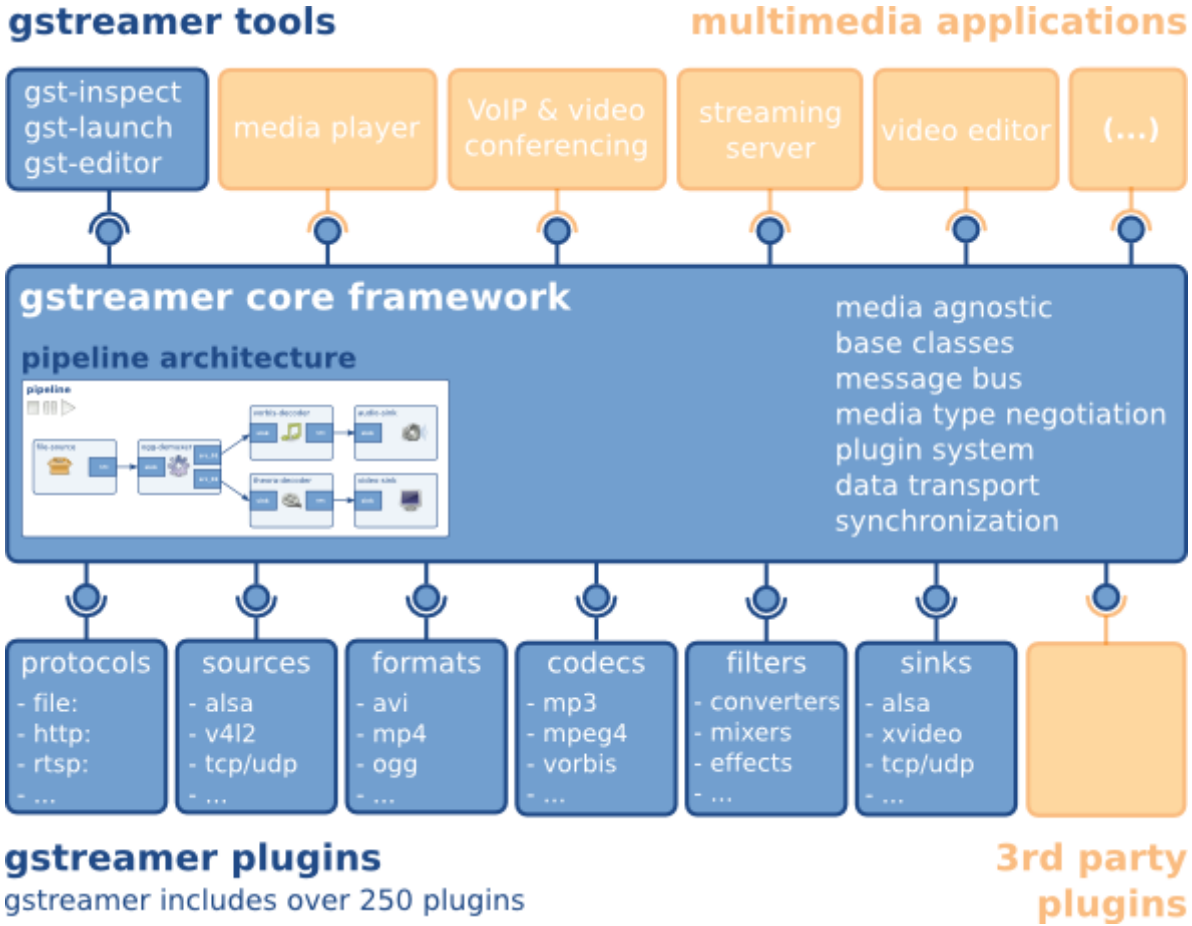
What is GStreamer?



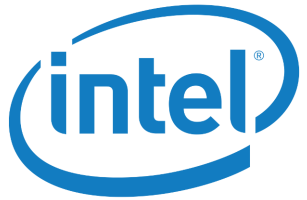
GStreamer is a framework for creating streaming media applications. The fundamental design comes from the video pipeline at Oregon Graduate Institute, as well as some ideas from DirectShow.
Specifically, GStreamer provides:

- an API for multimedia applications
- a plugin architecture
- a pipeline architecture
- a mechanism for media type handling/negotiation
- a mechanism for synchronization
- over 250 plug-ins providing more than 1000 elements
- a set of tools

GStreamer overview

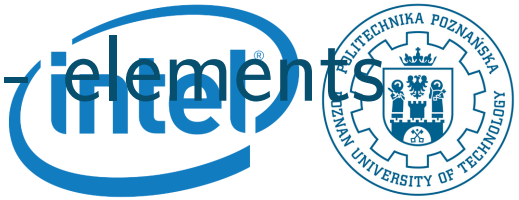


GStreamer design principles



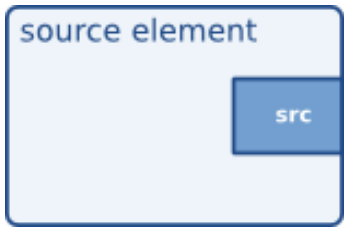
- Clean and powerful
  - Gstreamer provides clean interface to application programmer and plugin programmer.
- Object oriented
  - GStreamer adheres to GObject, the GLib 2.0 object model
- Extensible
  - All GStreamer Objects can be extended using the GObject inheritance methods.
- Allow binary-only plugins
- High performance
- Clean core/plugins separation
- Provide a framework for codec experimentation

GStreamer programming model - elements



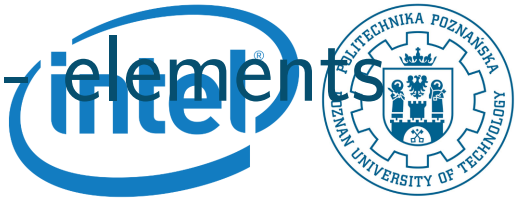
For the application programmer, elements are best visualized as black boxes. On the one end, you might put something in, the element does something with it and something else comes out at the other side.

- Source element



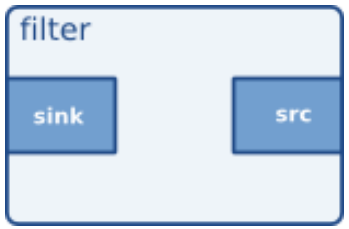
Source elements generate data for use by a pipeline, for example reading from disk or from a sound card.

GStreamer programming model - elements



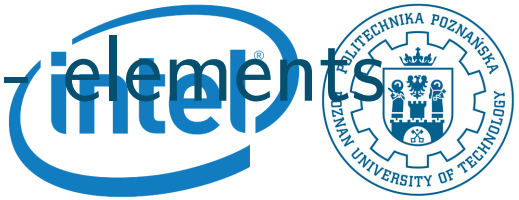
For the application programmer, elements are best visualized as black boxes. On the one end, you might put something in, the element does something with it and something else comes out at the other side.

- Filter element



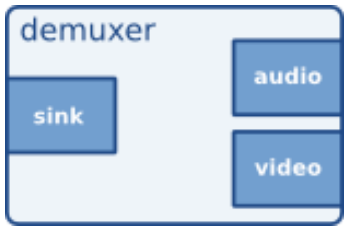
Filters and filter-like elements have both input and outputs pads. They operate on data that they receive on their input (sink) pads, and will provide data on their output (source) pads. Examples of such elements are a volume element (filter), a video scaler (convertor), an Ogg demuxer or a Vorbis

GStreamer programming model - elements



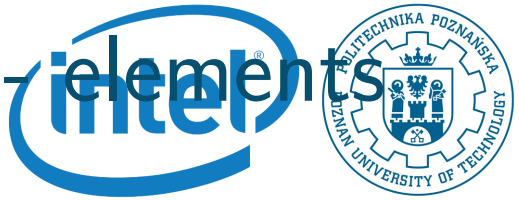
For the application programmer, elements are best visualized as black boxes. On the one end, you might put something in, the element does something with it and something else comes out at the other side.

- Multi-pad filter element



An example of one such element could, for example, be an Ogg demuxer for an Ogg stream containing both audio and video. One source pad will contain the elementary video stream, another will contain the elementary audio stream.

GStreamer programming model - elements

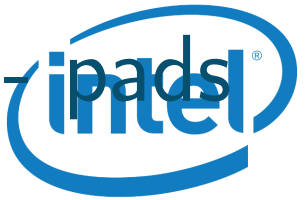


For the application programmer, elements are best visualized as black boxes. On the one end, you might put something in, the element does something with it and something else comes out at the other side.

- Sink element

Sink elements are end points in a media pipeline. They accept data but do not produce anything. Disk writing, soundcard playback, and video output would all be implemented by sink elements.

GStreamer programming model - pads

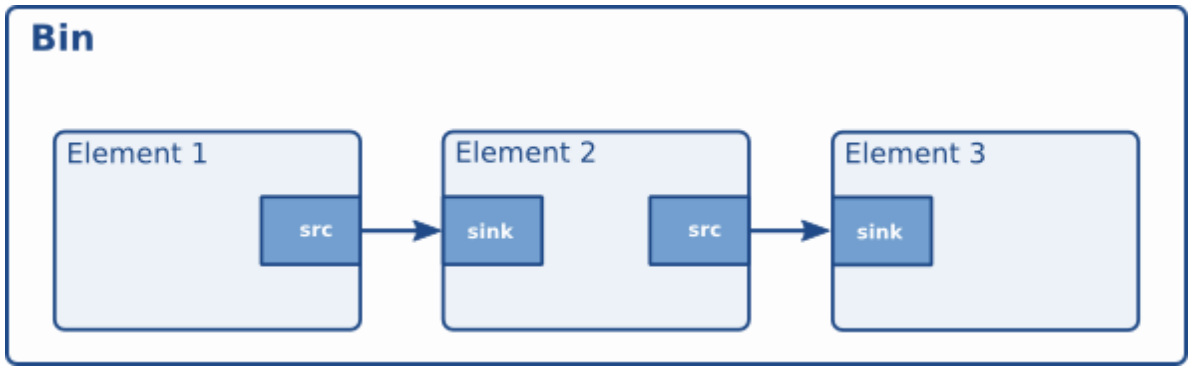


Pads are element's input and output, where you can connect other elements. They are used to negotiate links and data flow between elements in GStreamer.
Pads capabilities are:

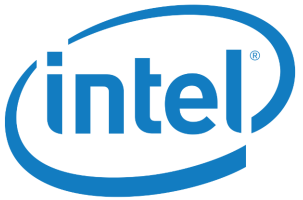
- Links are only allowed between two pads when the allowed data types of the two pads are compatible.
- Data types are negotiated between pads using a process called caps negotiation. Data types are described as a GstCaps.
- Data flows out of one element through one or more source pads, and elements accept incoming data through one or more sink pads.
- Source and sink elements have only source and sink pads, respectively.

GStreamer programming model - bins and pipelines

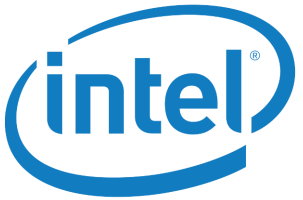
Bins allow to combine a group of linked elements into one logical element. This is extremely powerful when you are going to construct complex pipelines since it allows you to break up the pipeline in smaller chunks.



Outline



GStreamer - examples



toolkit installation

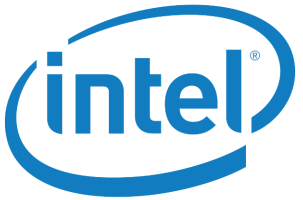
```
sudo apt-get install gstreamer-tools
```

exemplary pipeline

```
gst-launch videotestsrc ! autovideosink
```

GStreamer - examples

GStreamer - examples



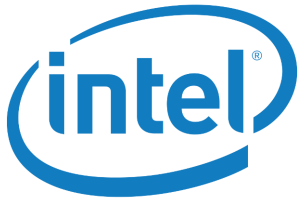
plugin inspection

gst-inspect videotestsrc

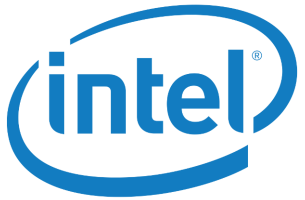
```
Plugin Details:
  Name:                 videotestsrc
  Description:          Creates a test video stream
  License:              LGPL
  Source module:        gst-plugins-base

Pad Templates:
  SRC template: src
    Availability: Always
    Capabilities:
      video/x-raw-yuv
                 format: YUY2
           color-matrix: { sdtv, hdtv }
            chroma-site: { mpeg2, jpeg }
                  width: [ 1, 2147483647 ]
                 height: [ 1, 2147483647 ]
              framerate: [ 0/1, 2147483647/1 ]
```

Outline



GStreamer - examples

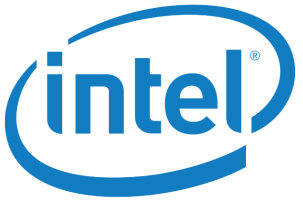


P

resentation of live GStreamer pipelines.

Resources

If you want to gain some knowledge by your own...



Wikipedia – Embedded system

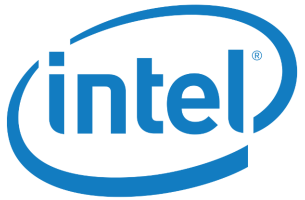
`http://en.wikipedia.org/wiki/Embedded_system`

Embedded System Market – Global Industry Analysis

`http://www.prnewswire.com/`

Free Electrons - embedded Linux experts

`http://free-electrons.com/`



Questions ?

Rafal Kapela
rafal.kapela@put.poznan.pl