GStreamer 1.0

No longer compromise flexibility for performance

Edward Hervey
edward@collabora.com
ELC 2012

GStreamer

- Open Source Multimedia Framework
- Set of libraries and plugins
- Direct Acyclic Graphs of elements
- API for plugins (to export features)
- API for applications

GStreamer 0.10

- 0.10 series (0.10.0 Dec 5 2005)
- Used widely and continuously improved
- More popular and solid than anticipated

0.10 Limitations

- Performance issues
- Some use-case very cumbersome to handle (hw-accel)
- Missing information
- Caps tightly coupled to buffer/memory
- Deprecated API

Enter GStreamer 1.0

- Talked about since 2007
- New challenges
  - Embedded Platforms
  - GPU
  - Dynamic pipelines
  - Re-negotation

Goals

- Improve performance
- Allow more use-cases
- Avoid vendor 'hacks'
- Minimize downstream patches

GStreamer 1.0

- API/ABI cleanups
- Memory Management
- (Re)Negotiation
- Dynamic Pipelines
- Open the road to better performance

- We'll stick to what's relevant to the embedded community

Memory management

- 0.10
  - One buffer => One 'data' field (pointer)
  - Content entirely specified by caps
  - No control over memory access
- Problems
  - Different content layout => new caps
  - More fields => Override data (or subclass)
- => Incompatibility/Maintenance Hell

Collabora

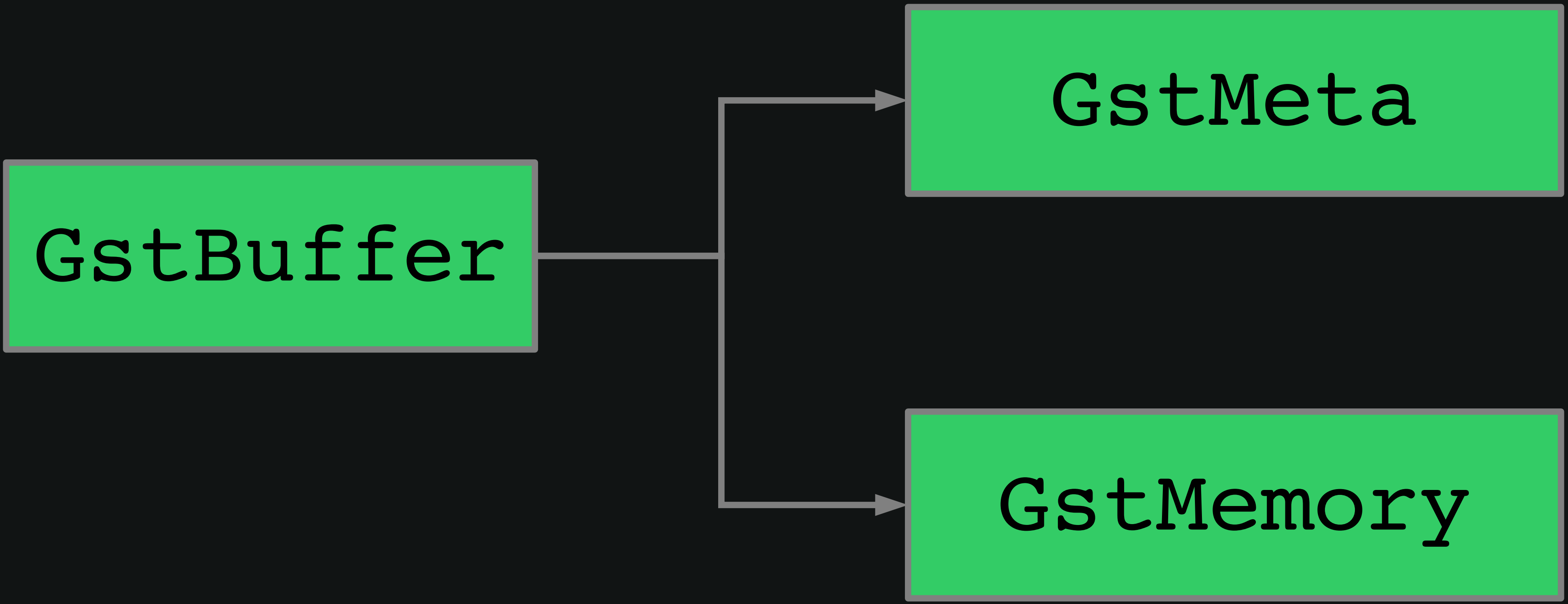
Memory management

- 0.10 Examples
  - Stride
    - video/x-raw-yuv-strided,stride=4096,...
    - Incompatible with all existing video elements :(
  - Non-contiguous planes
    - GstVendorBufferIncompatible
    - Also need specific caps to avoid other elements from prodding into (invalid/unknown) 'data' field
  - *<Insert the hack you had to do>*

Memory management

- 1.0
  - Memory separate from GstBuffer
  - Caps separated from GstBuffer
  - Generic Metadata system for GstBuffer

GstBuffer



$$\begin{tikzcd}
 & \text{GstMeta} \\
\text{GstBuffer} \arrow[ur] \arrow[dr] & \\
 & \text{GstMemory}
\end{tikzcd}$$

(Re)Negotiation

- 0.10
  - Linked with buffer allocation (comes from downstream)
- Problems
  - Slow
  - Doesn't work when upstream need to re-negotiate

(Re)Negotiation

- In 1.0, negotiation is entirely decoupled from buffer allocation
- GST_QUERY_ALLOCATION

Performance

- Re-use buffers
- Explicit concept of GstBufferPool

Impact of change

- Application porting minimal
- 'Naive' plugin porting minimal
- “Throw away the hacks”
  - Re-use existing features