RidgeRun - Texas Instruments Edge AI GStreamer Plugins

Marco Herrera-Valverde

Agenda

- Collaboration background
- Project background
- Project relevance
- GStreamer vs OpenVX
- Project outcome
- Implementation and challenges:
  - Memory management
  - Hierarchical functionality factorization
  - Multi channel support
- Conclusions
- Questions

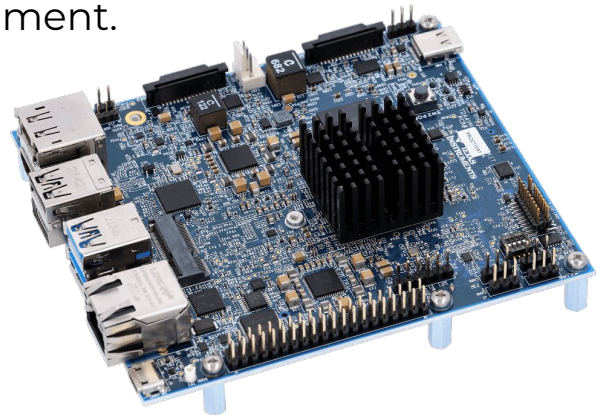
Collaboration Background

RidgeRun

- RidgeRun is a software development and service integration company that specializes in embedded systems across various industries.
- Our areas of expertise include:
  - Embedded Linux.
  - Artificial Intelligence.
  - Computer Vision.
  - FPGA video.
  - **GStreamer.**

Texas Instruments

- Texas Instruments is a well known design and manufacturing company in the analog and embedded semiconductor industry.
- Presence in multiple markets:
  - Personal electronics.
  - Automotive industry.
  - Communications equipment.
  - Enterprise systems.



TDA4VM Edge AI starter kit

Project Background

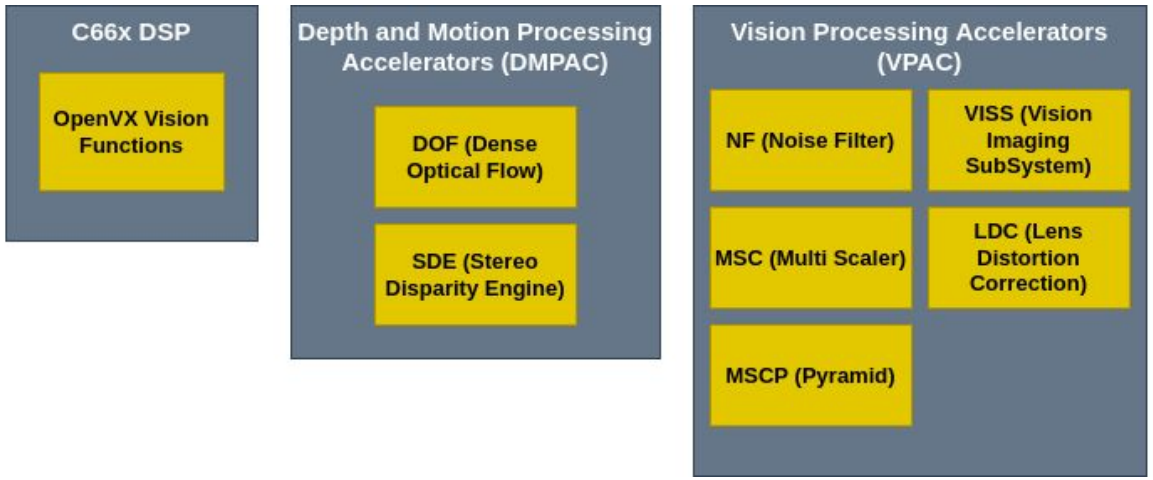
- Jacinto 7 hardware accelerators for vision applications through the TIOVX API.
- GStreamer as a multimedia interface to access computer vision accelerators.
- Functionalities supported by the project:
  - Color space conversion.
  - Multi output downscaling.
  - Dense optical flow.
  - Image signal processing.
  - Mosaicing.
  - Deep learning pre processing.
  - Lens distortion correction.
  - Pyramid conversion.

gstreamer

+

OpenVX

+



GStreamer vs TIOVX API

- Provides flexibility and ease of use.
- Easy prototyping of complex applications.
- Large ecosystem of elements for multimedia handling.
- Community support, makes application development easier.

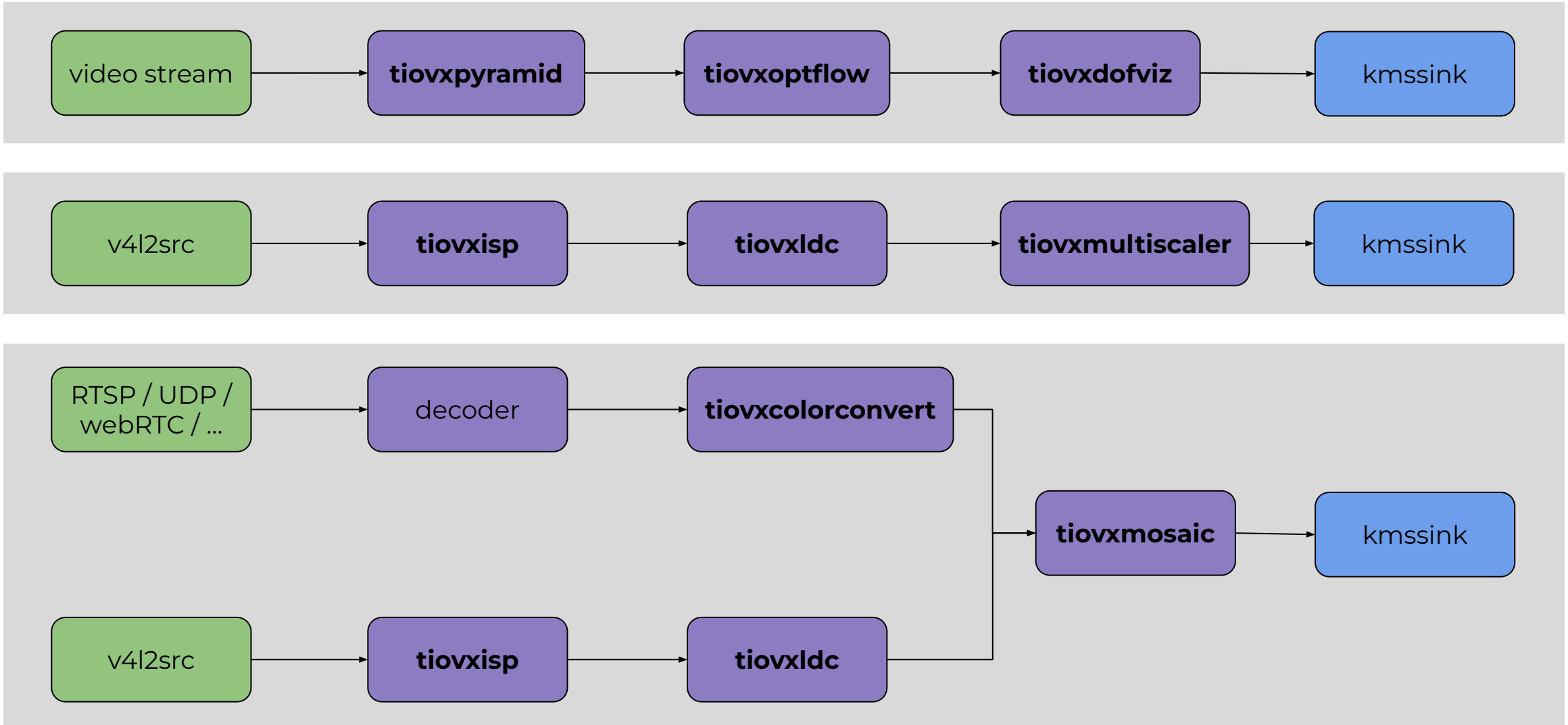
- Detailed control of graph processing.
- Detailed control of memory usage.
- More complex implementations.
- Higher efficiency, ideal for low-power, real-time applications.

Project Outcome

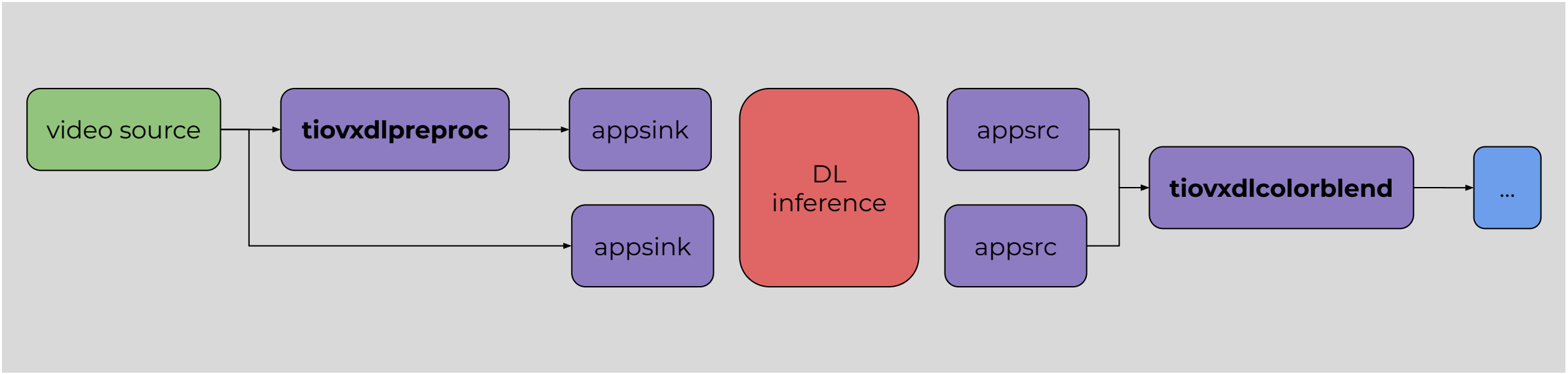
- Open source hardware accelerated elements.
- Keeping GStreamer pipeline functionalities.
- Custom caps and data types.
  - **`application/x-tensor-tiovx`**
  - **`application/x-pyramid-tiovx`**
  - **`application/x-dof-tiovx`**
- Custom and efficient memory management.
- Robust and scalable structure for future development.
- Useful pipelines for the Jacinto 7 platform.

1	tiovxcolorconvert
2	tiovxmultiscaler
3	tiovxdlpreproc
4	tiovxisp
5	tiovxldc
6	tiovxdemux
7	tiovxmosaic
8	tiovxmux
9	tiovxpyramid
10	tiovxdelay
11	tiovxdlpreproc
12	tiovxdof
13	tiovxdofviz

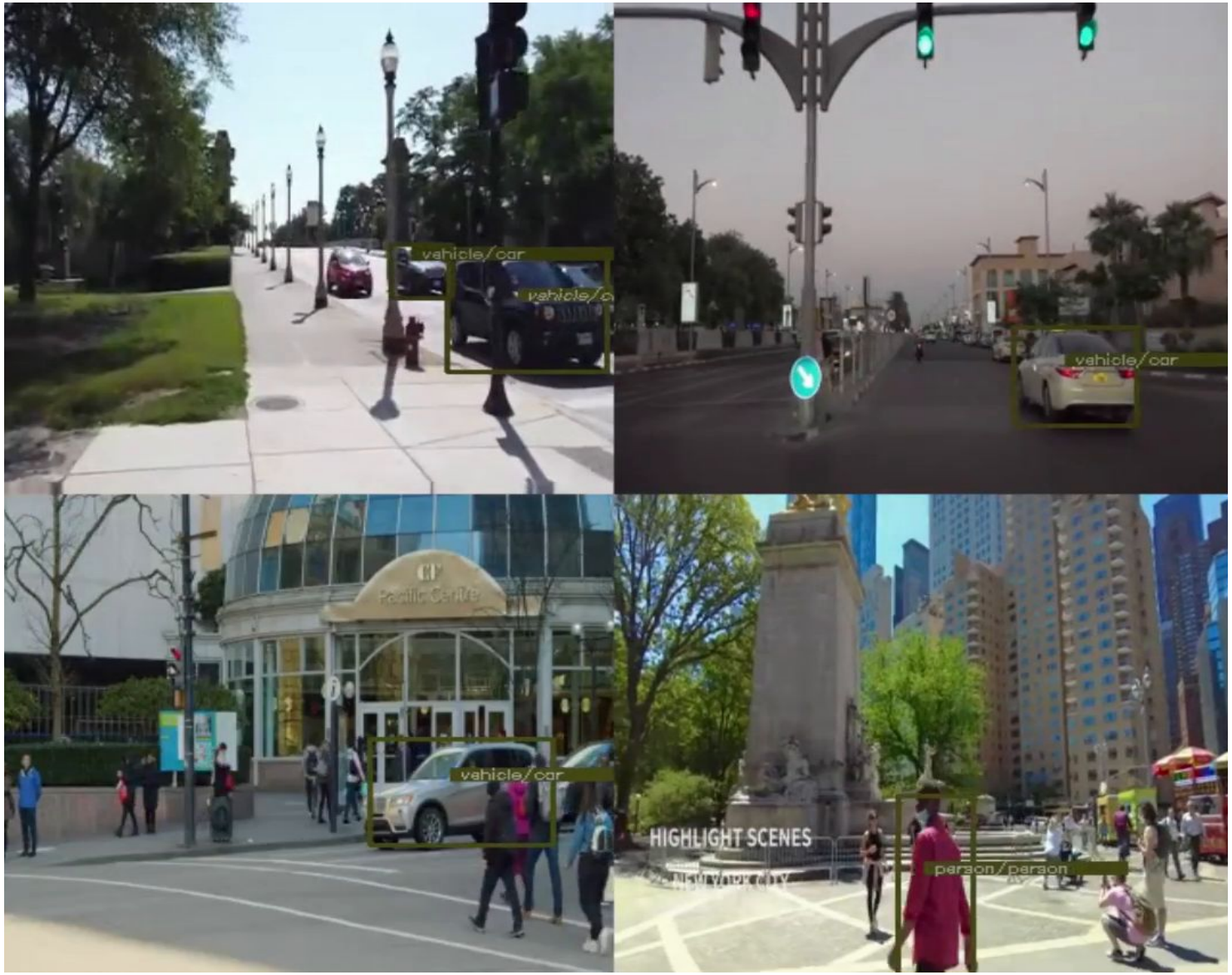
Example Pipelines



Example Pipelines



Example Output



Project Relevance

- Showcase an important use for GStreamer in the embedded world.
- Highlights GStreamer's potential and its current relevance as a multimedia framework.
- Project architecture and useful patterns.
- Applicability to many other APIs.
- Useful set of open source elements.

RidgeRun

+

gstreamer

+

Texas Instruments

Memory Management

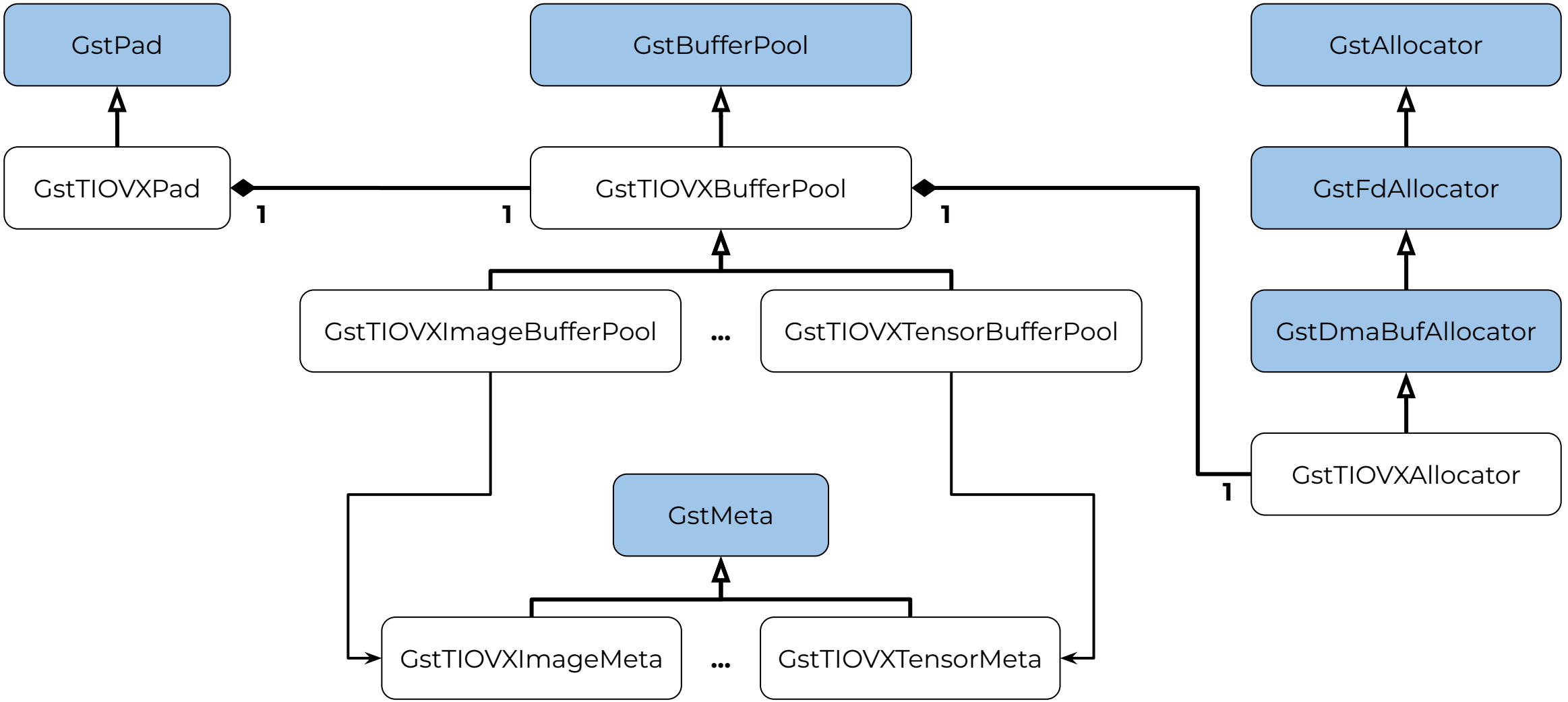
Reduced memory allocation

- Memory allocation is expensive.
- Object pool design pattern.
  - Use of pre allocated objects.
  - Objects are reused at runtime.
  - Reduces allocation overhead.
- GStreamer uses a GstBufferPool.
- Leverage GStreamer’s memory handling mechanisms.
- The project has specific memory requirements.

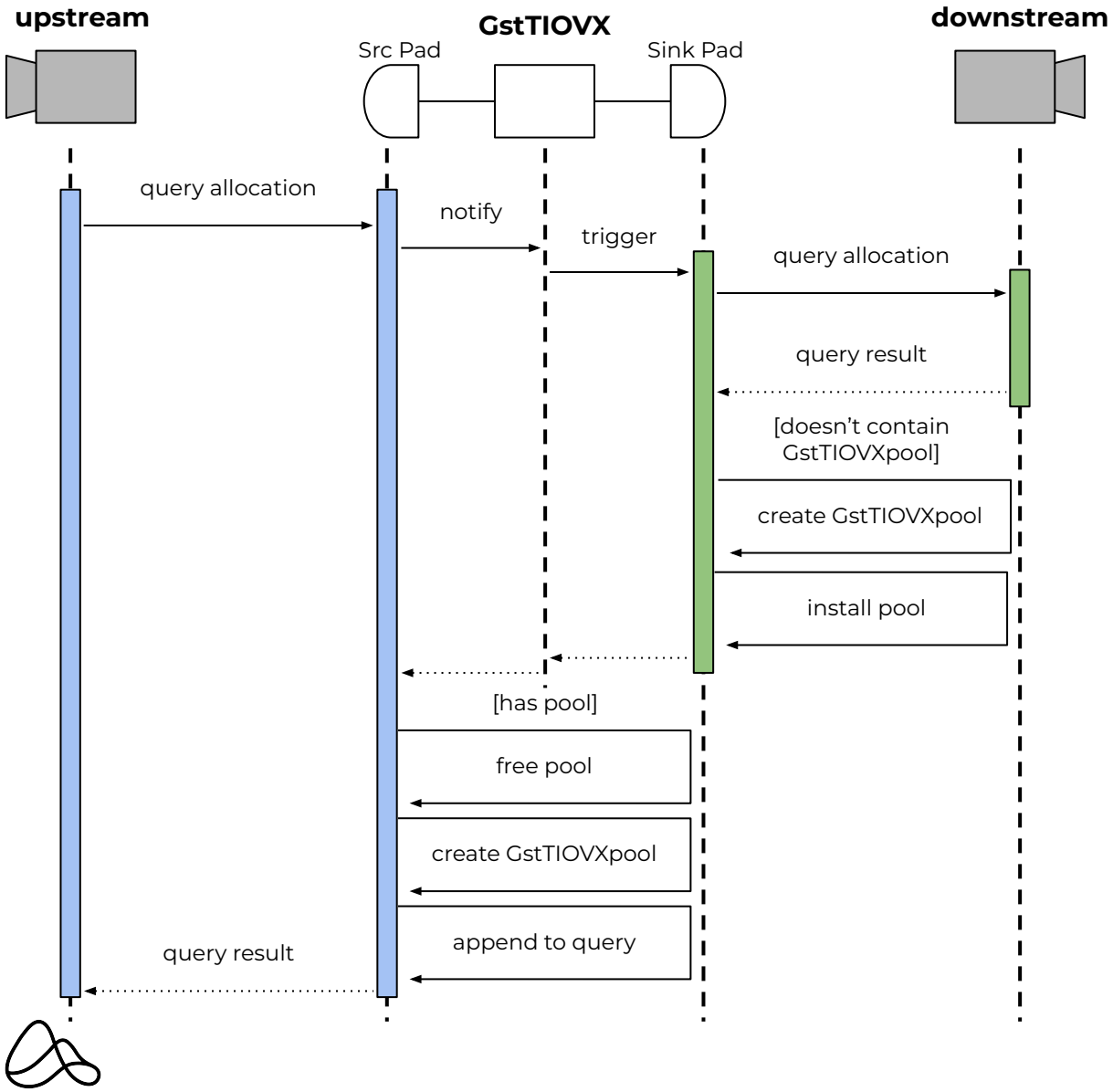
Zero copy configuration

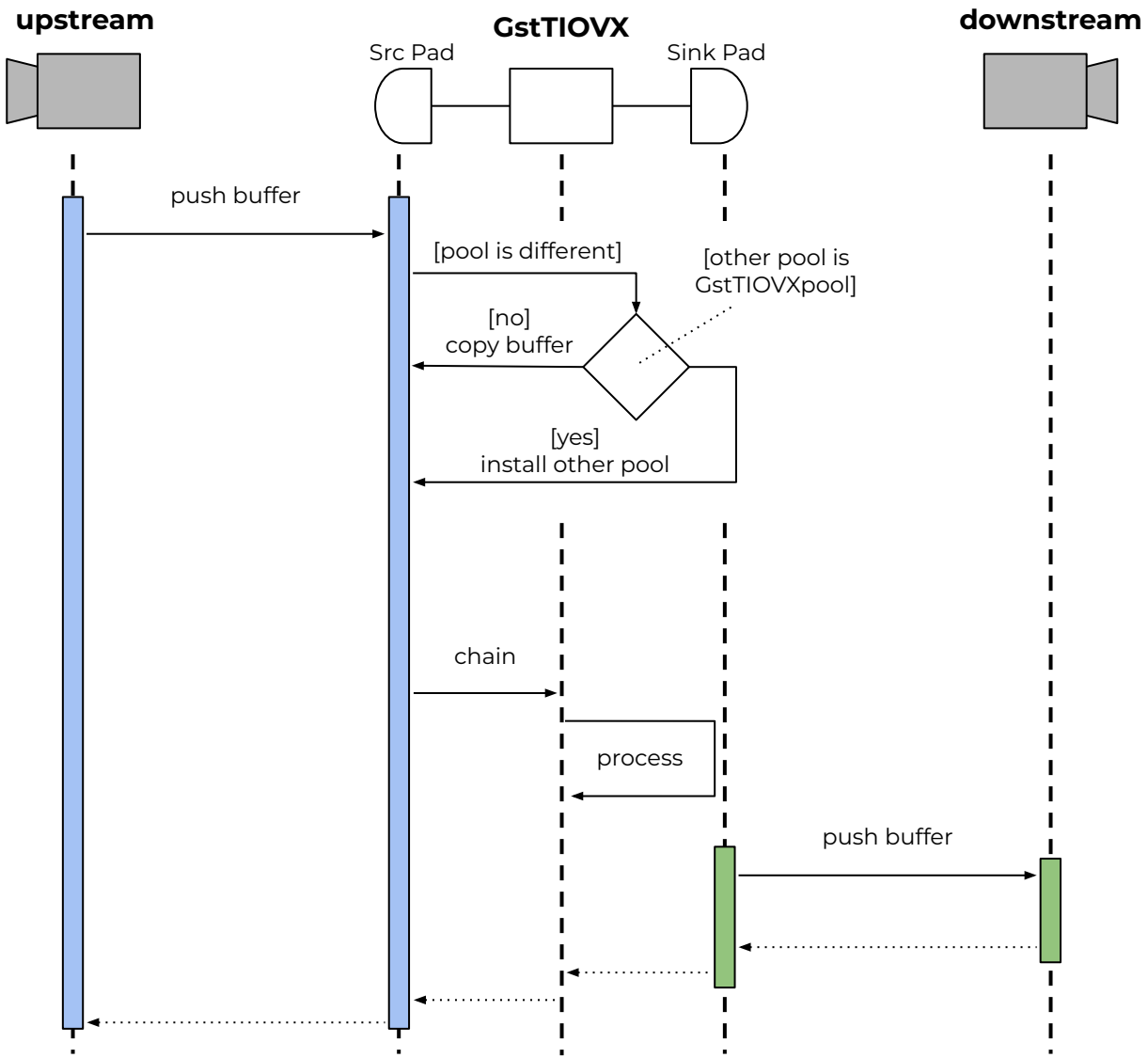
- Avoid copies from user space memory.
- Managing different TIOVX data types with different memory configurations.
- Solved by having custom pool implementations.
- Four main custom components:
  - Buffer pool.
  - Allocator.
  - Pad.
  - Meta.

Memory Management



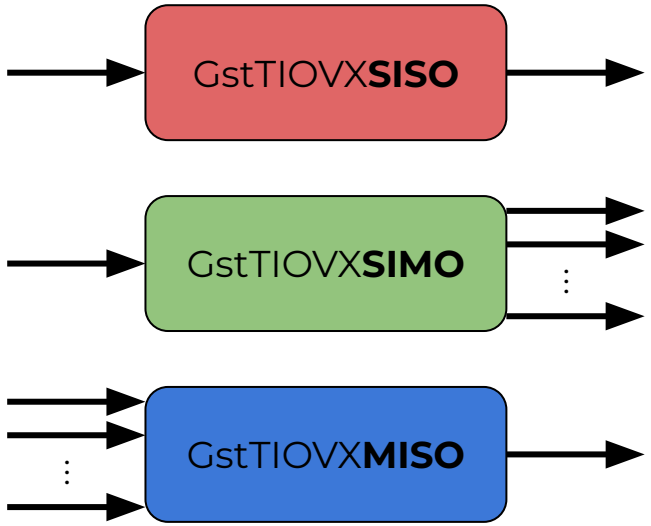
Memory Management





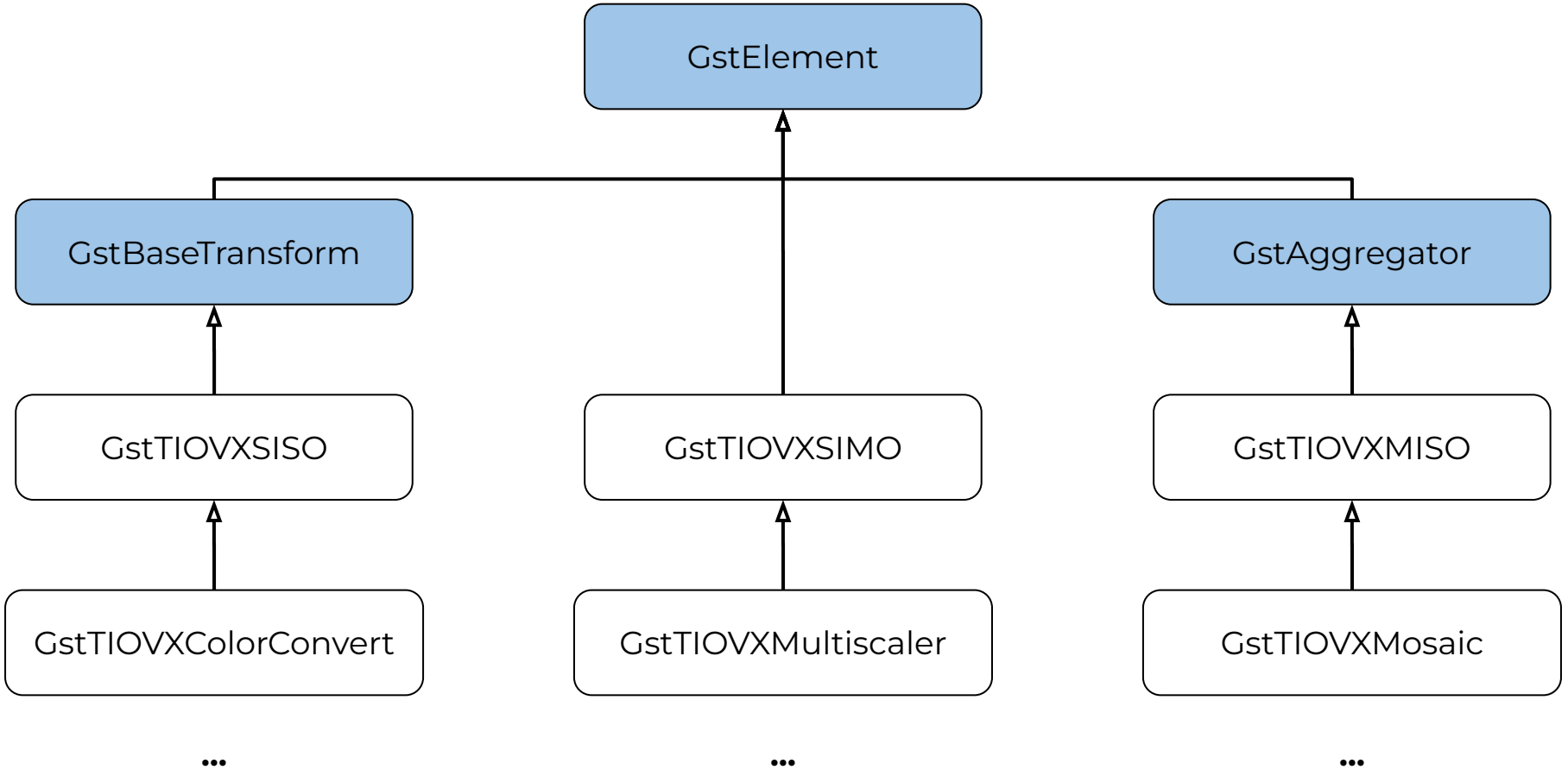
Hierarchical Functionality Factorization

- Classified by number of inputs and outputs.
- Abstraction of element complexity in base classes.
- Share parent class that represent similar behaviour.
- Facilitates future development and improves scalability.
- Clear distinction between elements and topologies.



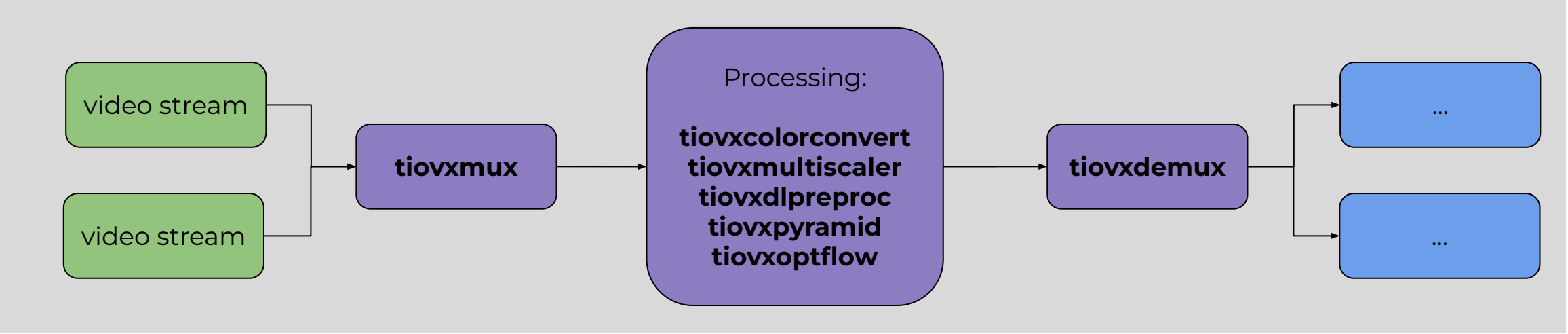

```
gboolean (*init_module)    (GstTIOVXSiso *trans, vx_context context, GstCaps * in_caps, …);
gboolean (*create_graph)   (GstTIOVXSiso *trans, vx_context context, vx_graph graph);
gboolean (*get_node_info)  (GstTIOVXSiso *trans, vx_object_array * input, …);
gboolean (*release_buffer) (GstTIOVXSiso *trans);
gboolean (*deinit_module)  (GstTIOVXSiso *trans, vx_context context);
gboolean (*compare_caps)   (GstTIOVXSiso *trans, GstCaps *caps1, GstCaps *caps2, …);
```

Hierarchical Functionality Factorization



Multi Channel Support

- Custom elements for multi stream batch processing.
- Extended afterwards, no major changes in architecture.
- Fully functional caps negotiation with (memory:batched).



Project Takeaways

GStreamer

- GStreamer works great as a multimedia interface to abstract APIs.
- Efficient and robust custom memory management.
- Similar performance to direct API usage without major overhead.
- GStreamer provides a good compromise between detailed control and ease of use.

Architecture

- Architecture can be adapted to many use cases in the multimedia embedded world.
- Design patterns improve:
  - Project scalability and maintainability.
  - Code readability and organization.
- Project architecture impacts direct users and future developers.

References

- https://developer.ridgerun.com/
- https://github.com/TexasInstruments/edgeai-gst-plugins
- https://github.com/TexasInstruments/edgeai-gst-apps
- https://developer.ridgerun.com/wiki/index.php/Getting_started_with_TI_Jacinto_7_Edge_AI
- https://software-dl.ti.com/jacinto7/esd/processor-sdk-rtos-jacinto7/latest/exports/docs/tiovx/docs/user_guide/index.html

RidgeRun - Texas Instruments Edge AI GStreamer Plugins

Marco Herrera-Valverde