nnstreamer

Neural Networks as GStreamer Filters

Adding arbitrary neural networks into the pipeline!

Samsung Research | AI Center, Seoul

MyungJoo Ham, Ph.D.

GStreamer Conference 2018

Who am I?

MyungJoo Ham

- Samsung (Korea)

- Dev. Infra & SW Platform for "on-device" AI Projects
  - nnstreamer is a part of "infra" for "on-device" AI projects.
    - Newbie in GStreamer. (all of nnstreamer members as well)

- Previously
  - Linux Kernel
    - maintainer: devfreq, extron
  - Tizen
    - Ex-architect: "System F/W" and "Base" domains (1.0 ~ 2.4)
    - Build & Release process "restructure" (for 4.0)
    - .NET Core Runtime (Linux/ARM) (for 4.0)

What is nnstreamer?

Let gstreamer handle “tensor”!

- Tensor: multi-dimensional array
- Why tensor?: It’s input/output of neural networks.

What is nnstreamer?

Set of gstreamer plugins

- Invoking arbitrary neural network frameworks & models as “stream filters”
- Manipulating stream data, “tensor”, for such filters

Neural networks are yet another media filter.
Data for/from neural networks are yet another media data.

What is nnstreamer?

- Fully Open Sourced (LGPL 2.1)
  - https://github.com/nnsuite/nnstreamer
  - We want to upstream to “gst-plugins-good” later.
- Open for participation
  - Current developers are GStreamer newbies!
- Transparent to HW & SW platforms.

nnstreamer “Official Releases”

- Release: Tizen (for-next), Ubuntu (16.04, 18.04)
  - Ubuntu PPA: ppa:nnstreamer/ppa
  - WIP: build.opensuse.org
- CI Testing: Tizen (x64, arm), Ubuntu, Yocto
  - WIP: Android, aarch64

Topics

GStreamer Conference 2018

Background

Cast Study

GStreamer Conference 2018

Case 1: Autonomous Vehicle

LIDAR

Camera

RADAR

* **This is a simplified example.**

Case 1: Autonomous Vehicle, Neural Network Models

Much simplified from the real case

LIDAR

Camera

RADAR

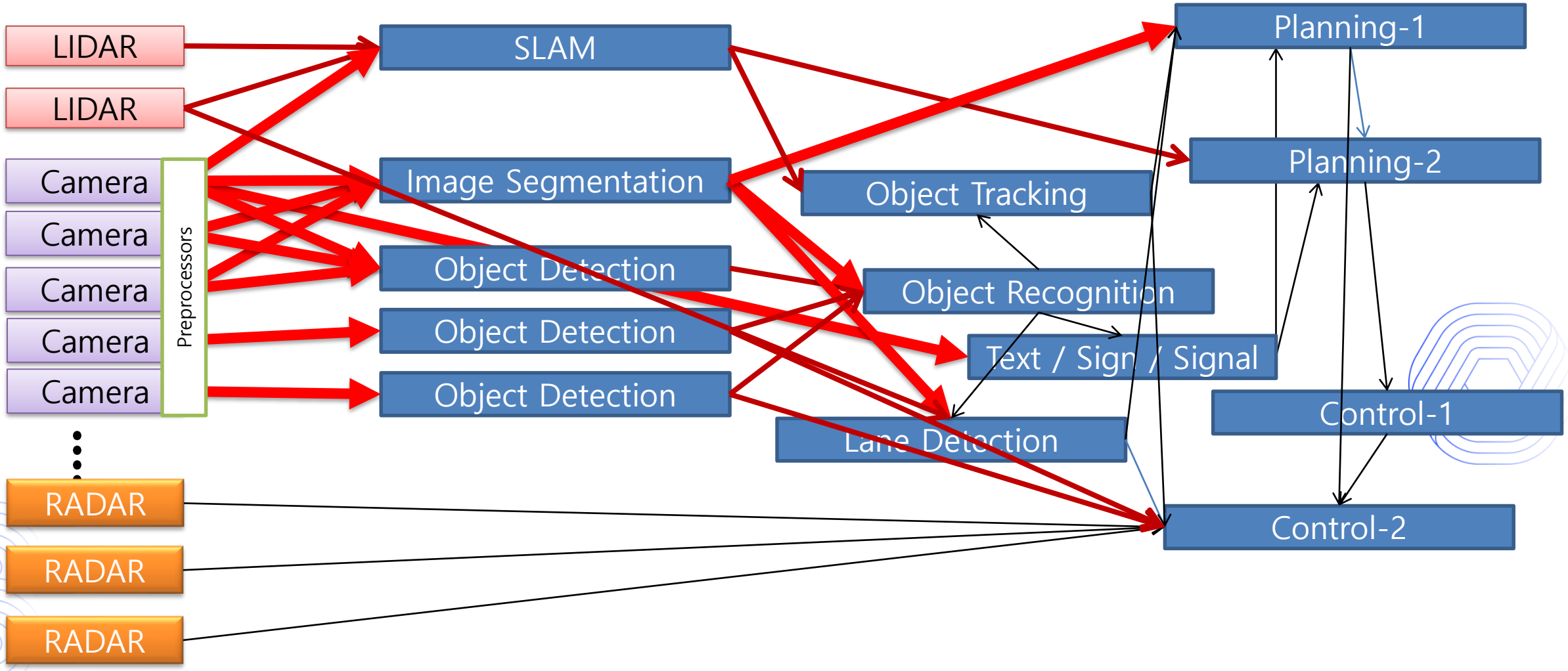
- Object Detection
- Object Recognition
- Object Tracking
- Image Segmentation
- Lane Detection
- SLAM
- Planning
- Control
- •
- •
- •
- •
- •
- •

SLAM: Simultaneous localization and mapping

* **This is a simplified example.**

Case 1: Autonomous Vehicle, Data Stream Pipelines

Much simplified from the real case

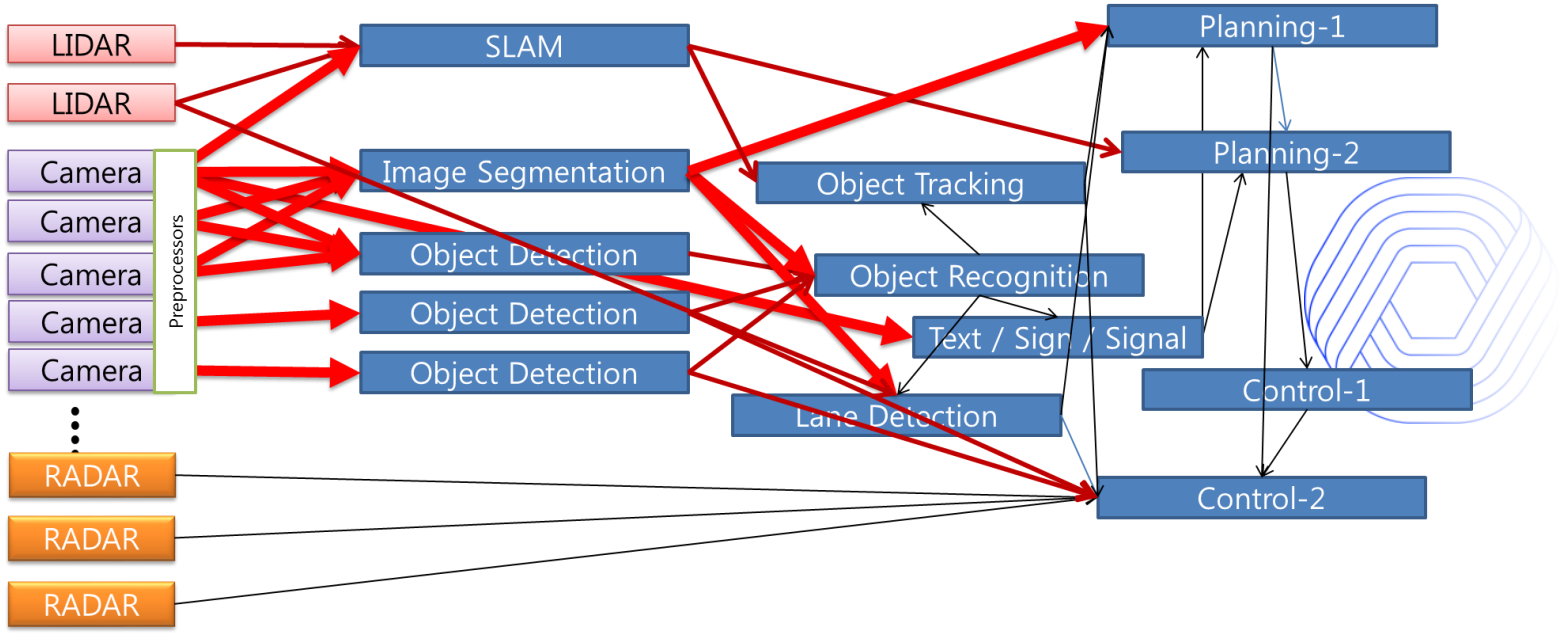


* **This is a simplified example.**

Case 1: Autonomous Vehicle, Data Stream Pipelines

Common Issues

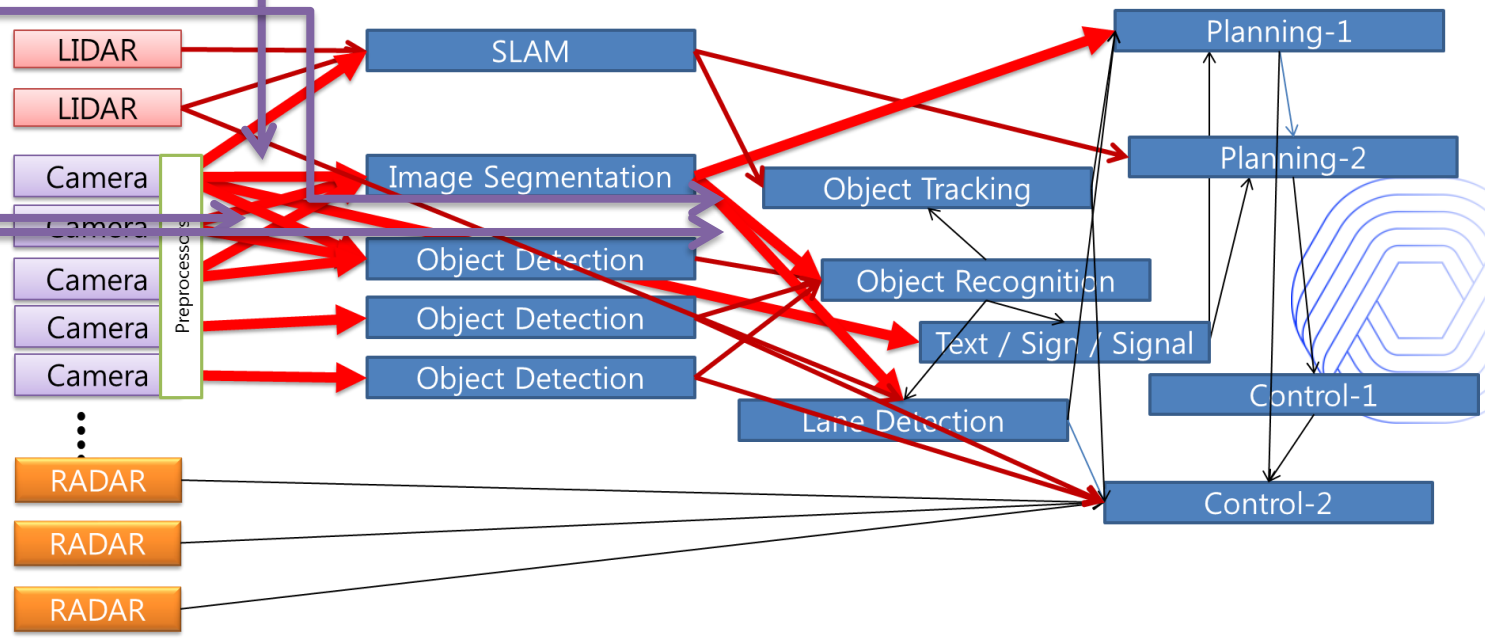
- High Bandwidth / Real-Time
- Complex Pipeline Topology
- Synchronization & Parallelism
- Latency & Jitter Control



Case 1: Autonomous Vehicle, Data Stream Pipelines w/ **ROS**

Common Issues with ROS (Socket!)

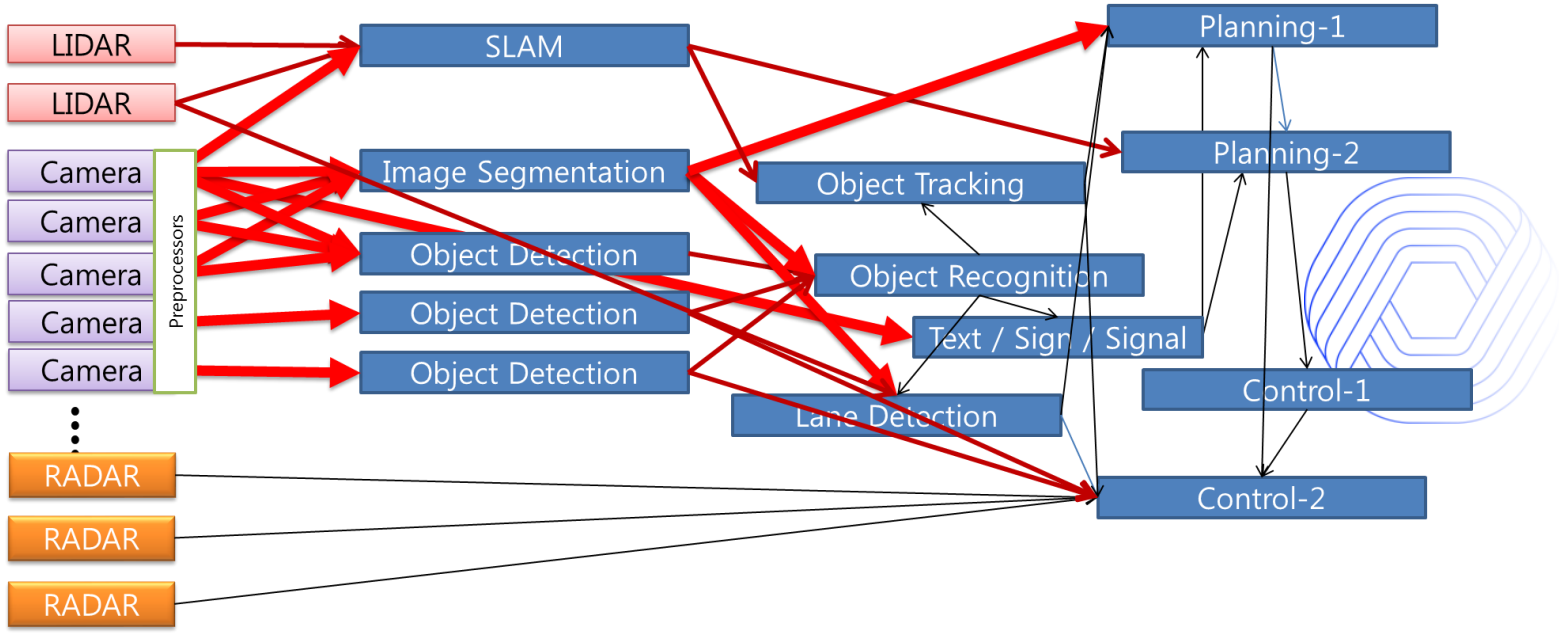
- High Bandwidth / Real-Time
  - High-Res Video in Real-Time
    - ROS2 / DDS (RTI Connext DDS) may help?
  - Transmission between models
  - Duplicated transmission
- Even with high-end workstation



Case 1: Autonomous Vehicle, Data Stream Pipelines

Common Issues with ROS

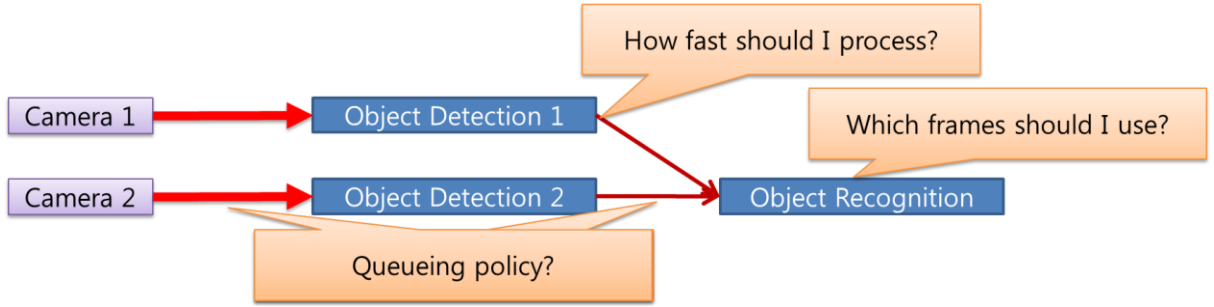
- Complex Pipeline Topology
  - Pub/Sub supports complex topology
  - May be it's overkill… with side-effects

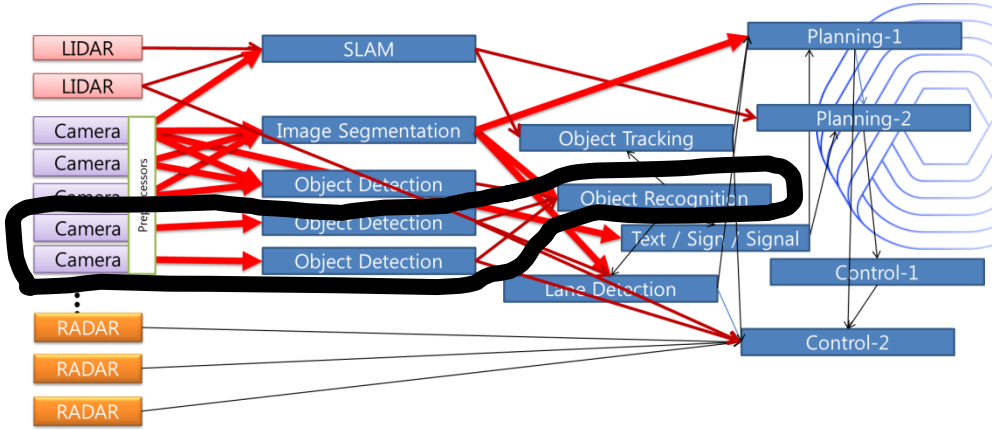


Case 1: Autonomous Vehicle, Data Stream Pipelines

Common Issues with ROS

- Synchronization
  - Independently scheduled processes → **Headaches!**

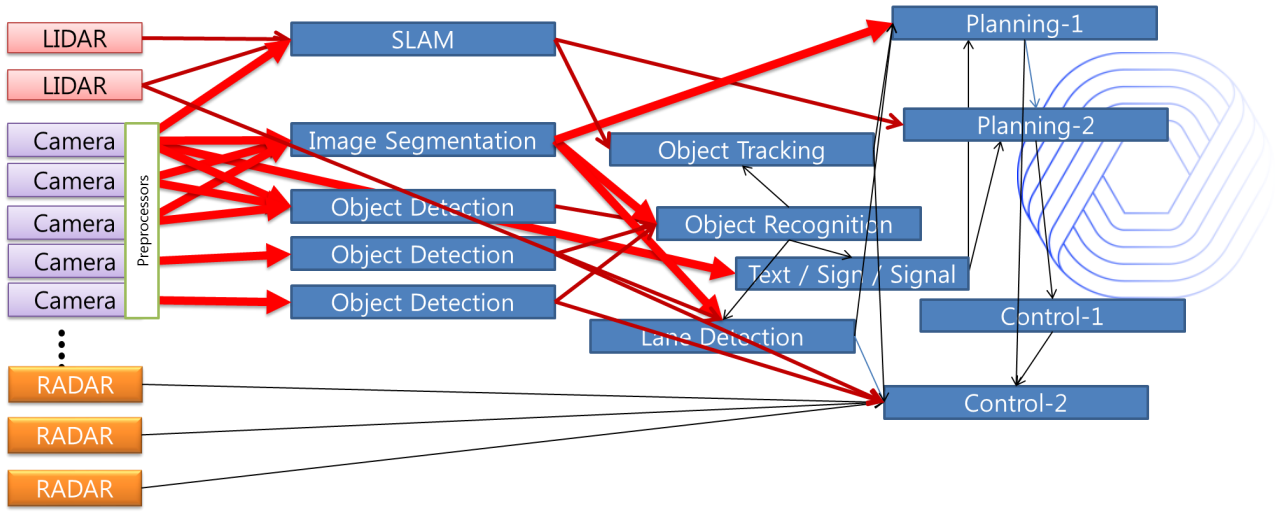




Case 1: Autonomous Vehicle, Data Stream Pipelines w/ **ROS**

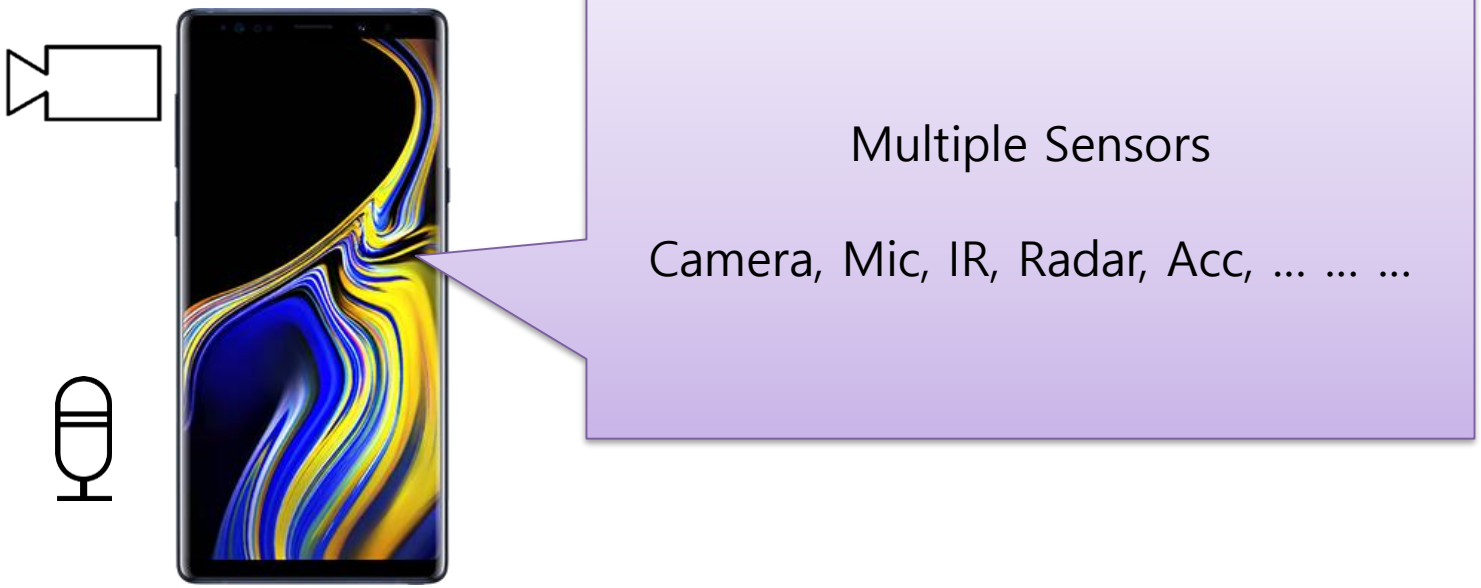
Common Issues with ROS (Socket!)

- Latency
  - Communication Latency due to Memcpy
- Jitter Control
  - Resource/Rate Control
  - Making it more difficult to synchronize



Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices



Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Multiple AI-based Services

Emotion Recognition, Speaker Recognition, Speech-to-Text, Face Detection & Recognition, and anything you imagine…

- Intelligence Service 1
- Intelligence Service 2
- Intelligence Service 3
- Intelligence Service 4
- ⋮

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Common neural networks with Mic inputs

- Env. Recog
- Speech Recog
- Speaker Recog
- Emotion Recog

- Intelligence Service 1
- Intelligence Service 2
- Intelligence Service 3
- Intelligence Service 4
- ⋮

These examples do not reflect actual products

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Common neural networks with camera inputs

Obj. Det.
Obj. Recog
Obj. Trk
Segmentation
Face. Det
Emotion RecogV
Face. Trk
Sit. Recog

Env. Recog
Speech Recog
Speaker Recog
Emotion Recog

Intelligence Service 1
Intelligence Service 2
Intelligence Service 3
Intelligence Service 4

These examples do not reflect actual products

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Combining results from both audio and video

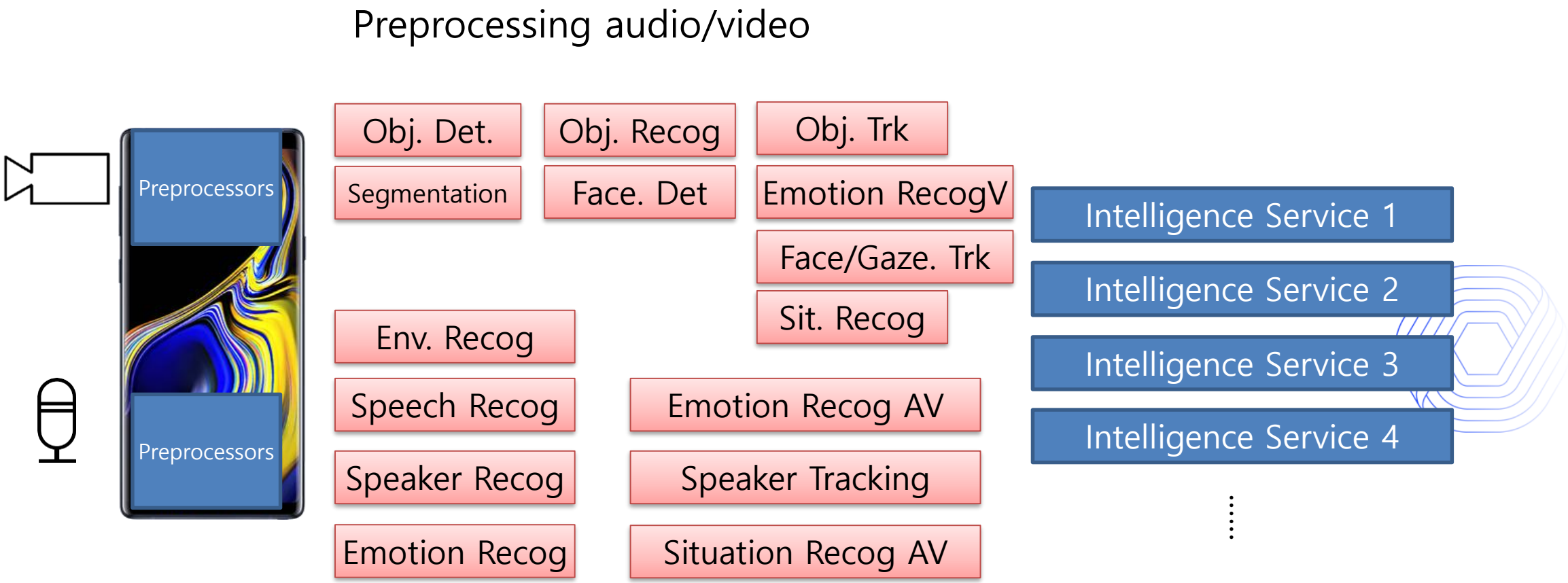
Obj. Det.
Obj. Recog
Obj. Trk
Segmentation
Face. Det
Emotion RecogV
Face. Trk
Sit. Recog
Env. Recog
Speech Recog
Speaker Recog
Emotion Recog
Emotion Recog AV
Speaker Tracking
Situation Recog AV

Intelligence Service 1
Intelligence Service 2
Intelligence Service 3
Intelligence Service 4
⋮

These examples do not reflect actual products

Case 2: Multi-Modal Interactions

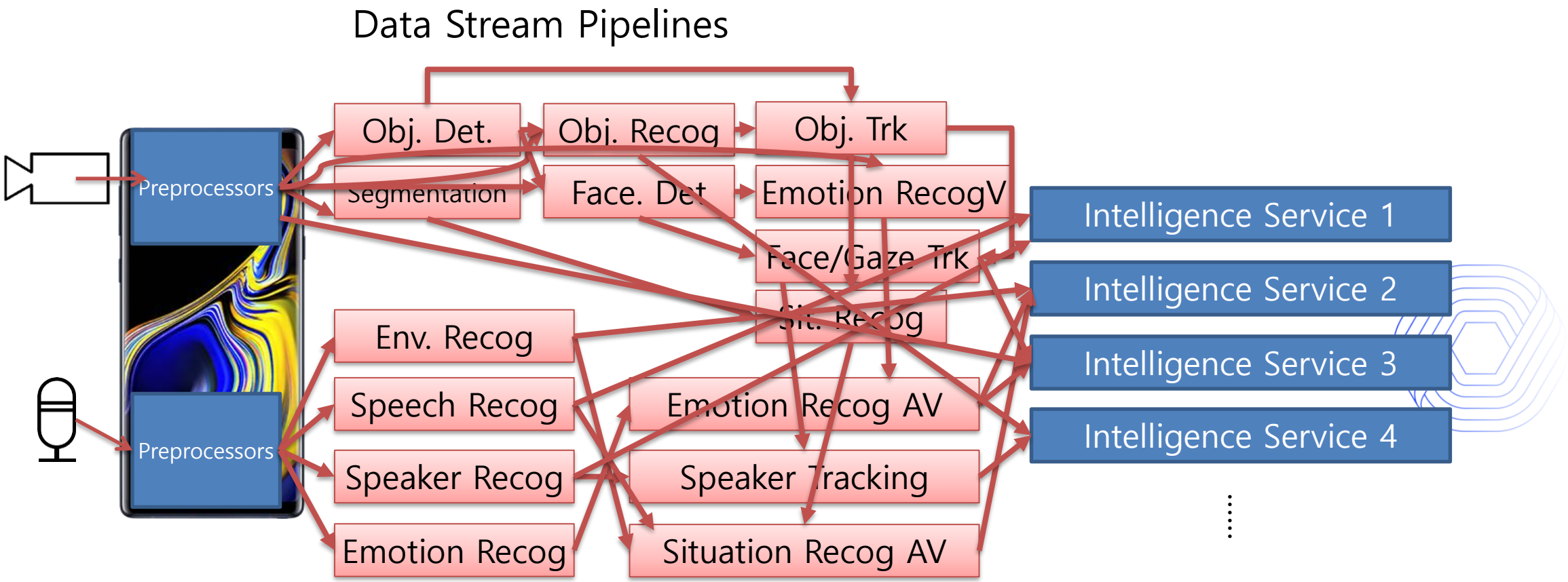
Robots, Consumer Electronics, Mobile Devices



These examples do not reflect actual products

Case 2: Multi-Modal Interactions

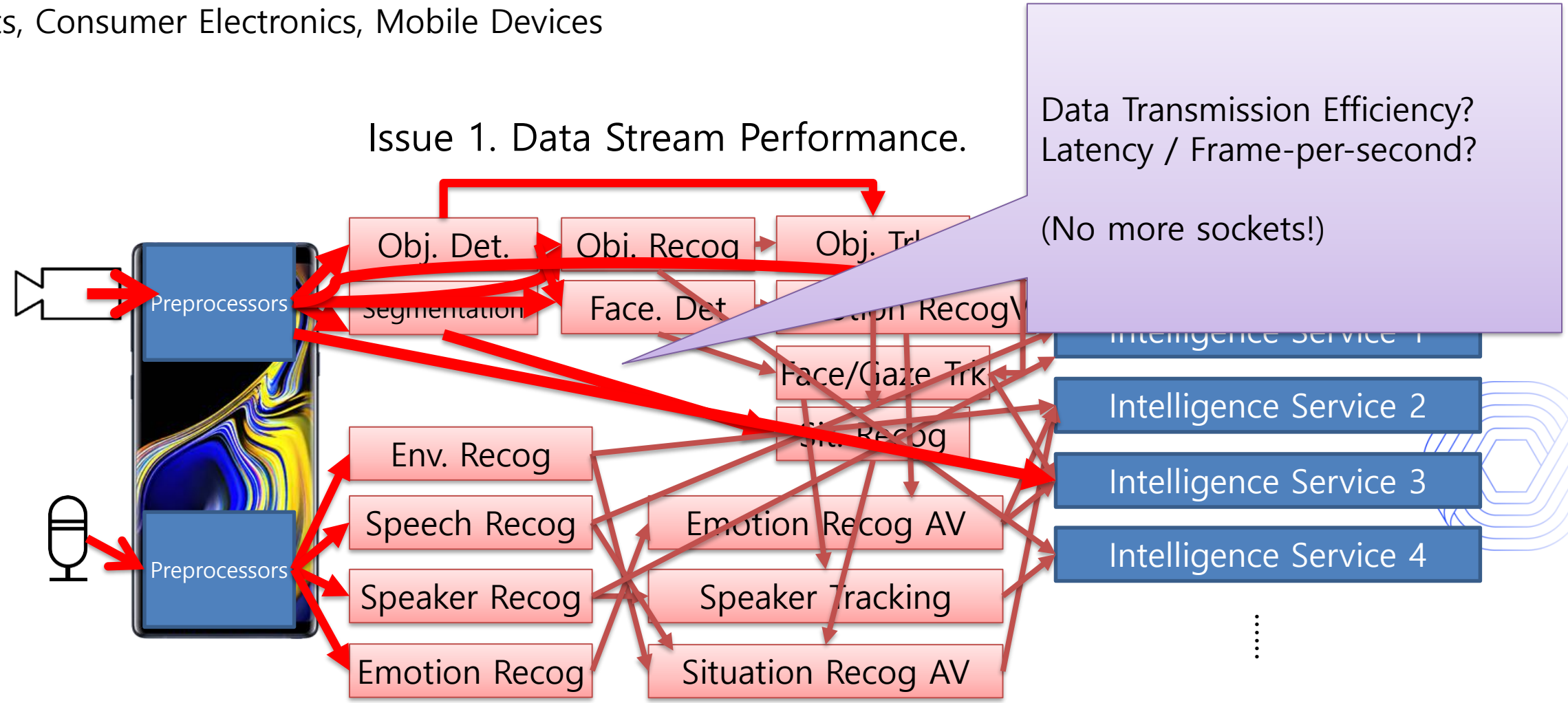
Robots, Consumer Electronics, Mobile Devices



These examples do not reflect actual products

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

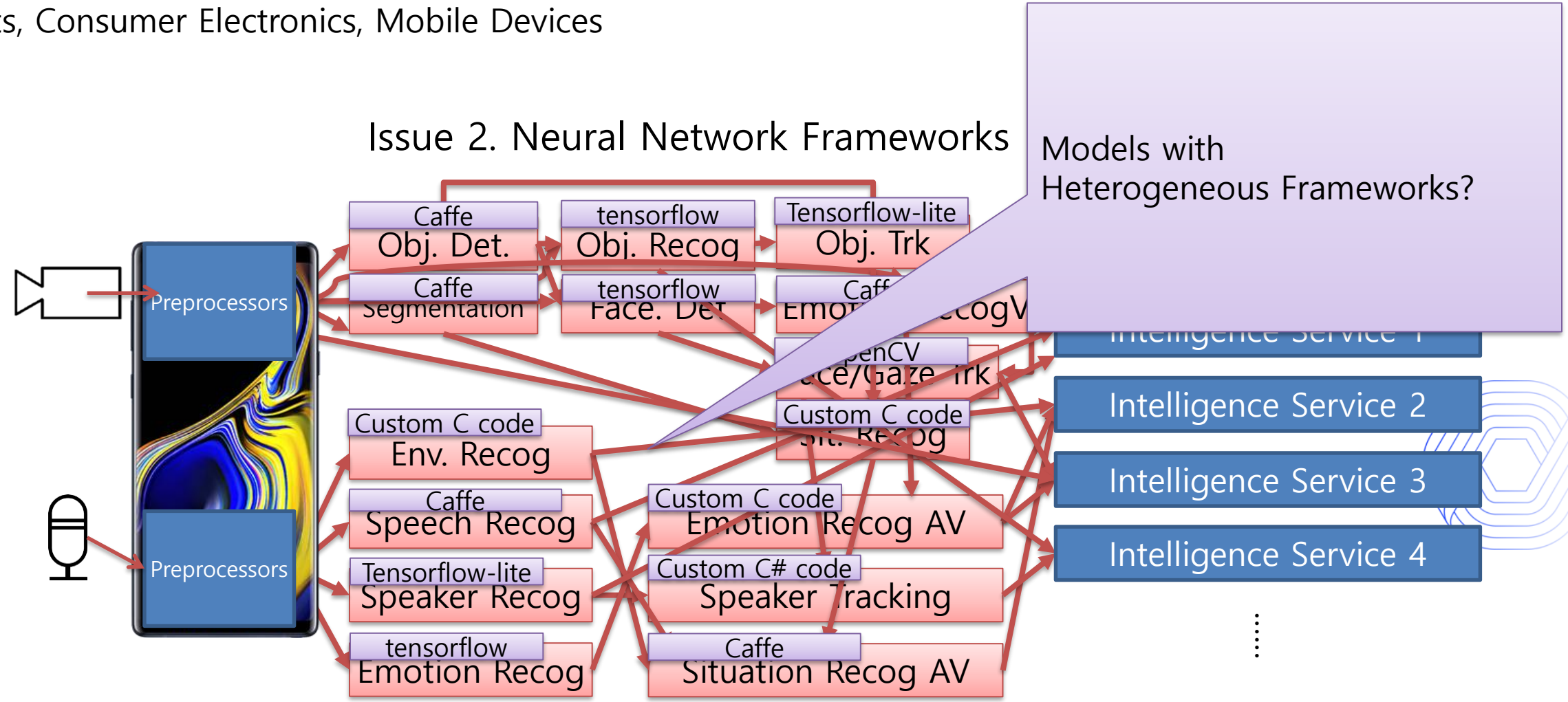


These examples do not reflect actual products

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 2. Neural Network Frameworks

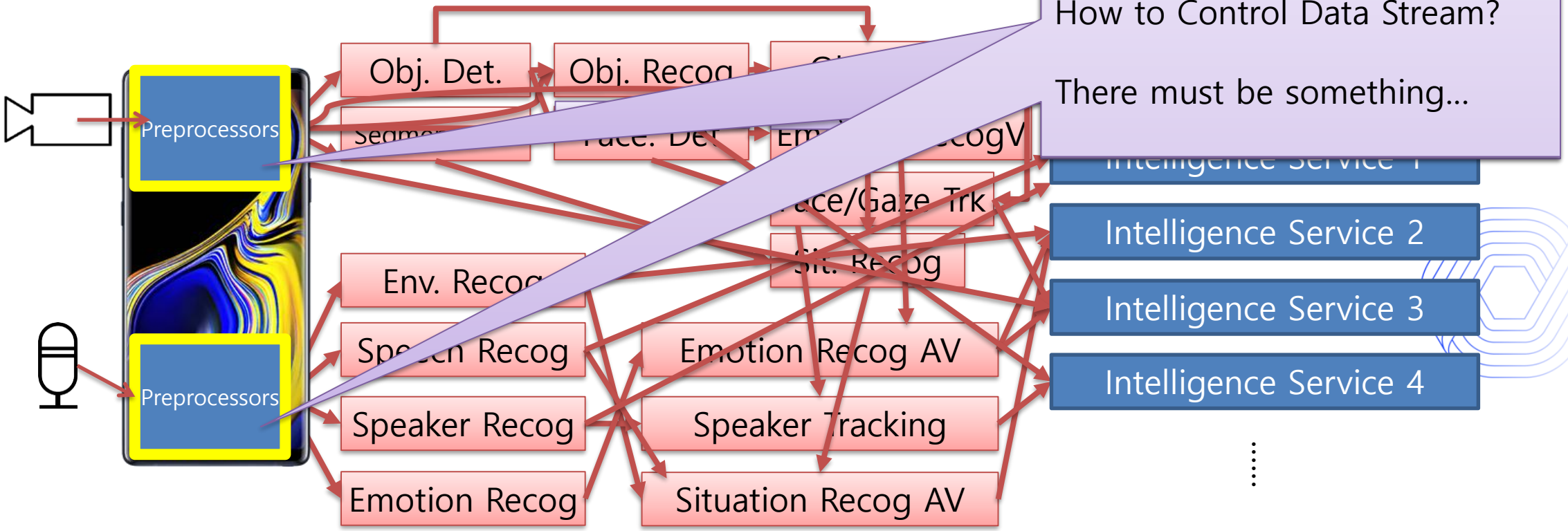


These examples do not reflect actual products

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 3. Multimedia Data Processing

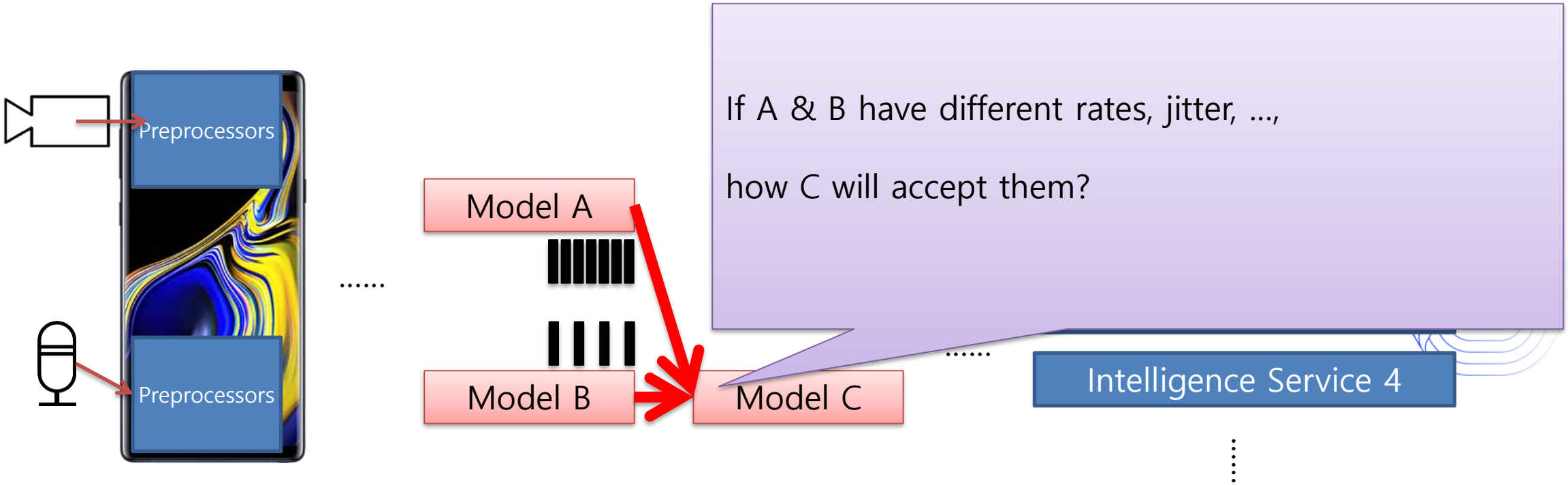


These examples do not reflect actual products

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 4. Data Stream Synchronization

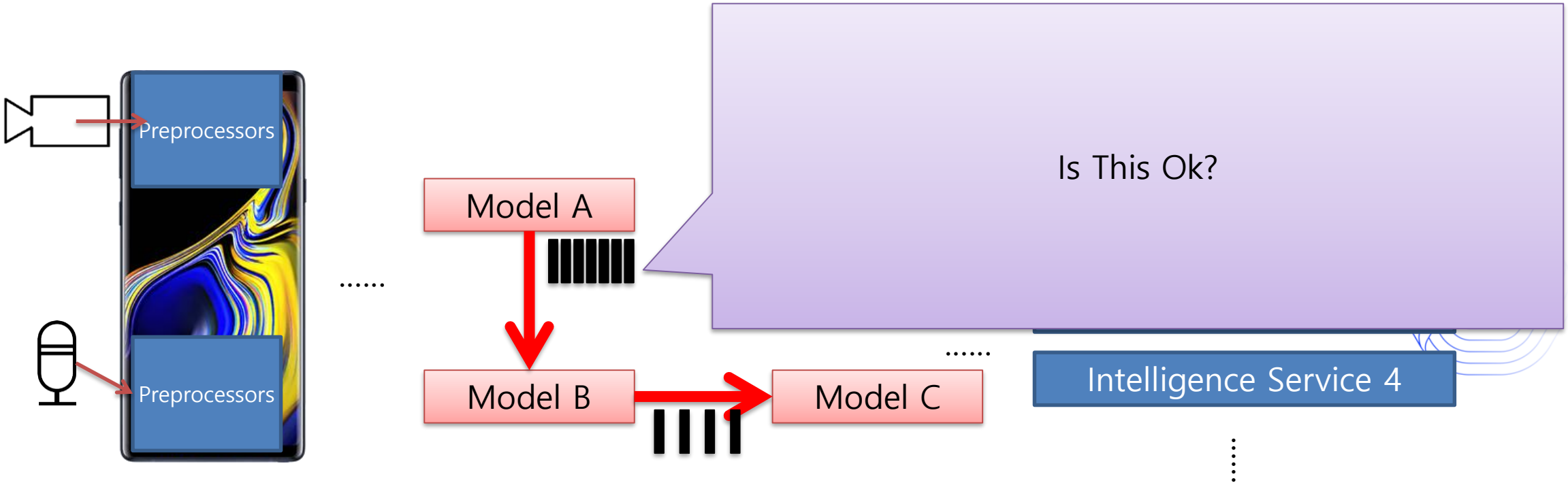


These examples do not reflect actual products

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 5. Data Stream Rate Control

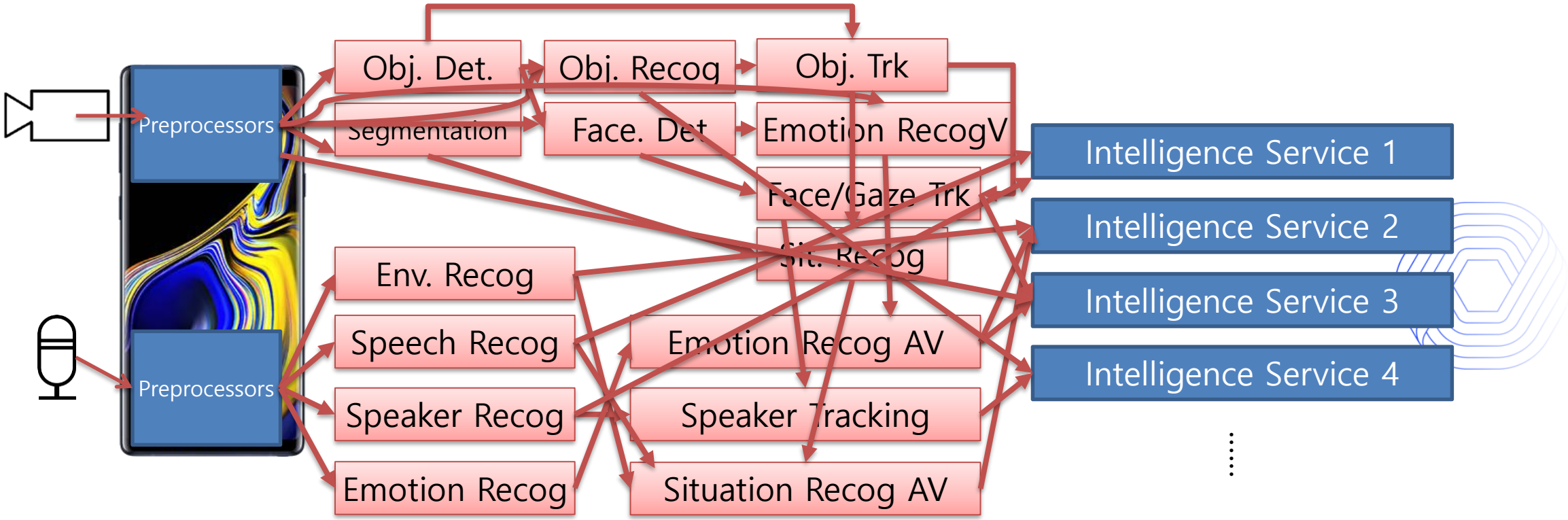


These examples do not reflect actual products

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 6. Dynamic Stream Pipelines

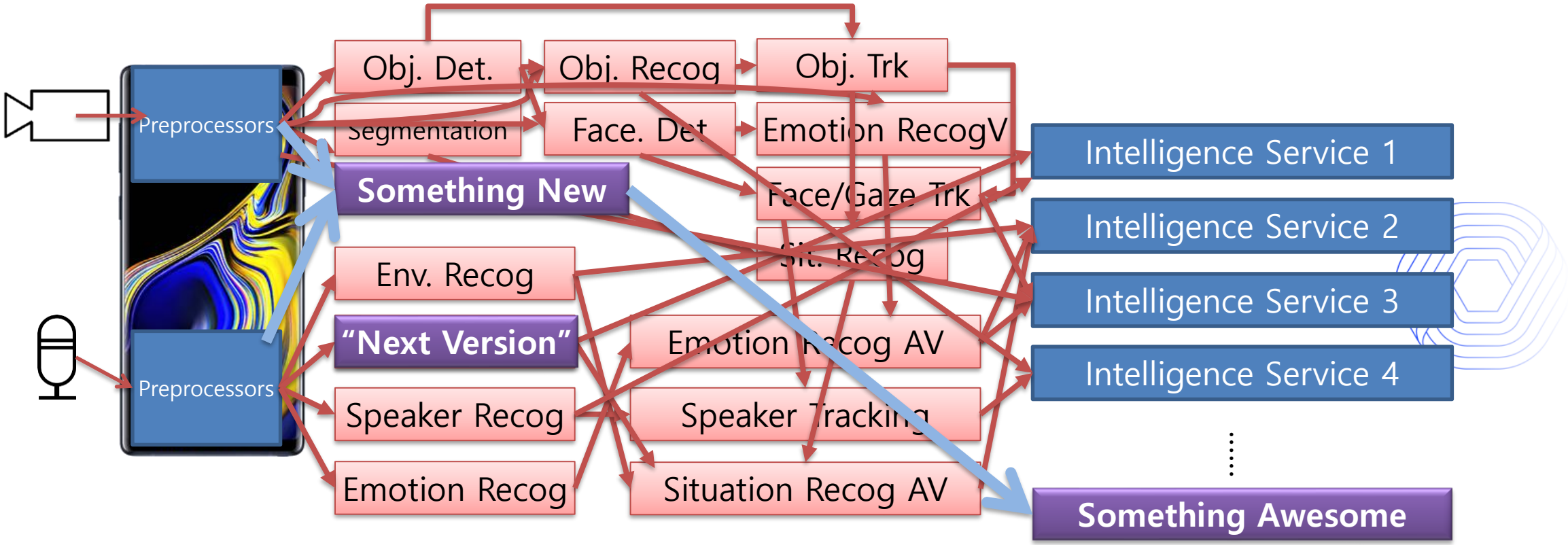


These examples do not reflect actual products

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 6. Dynamic Stream Pipelines

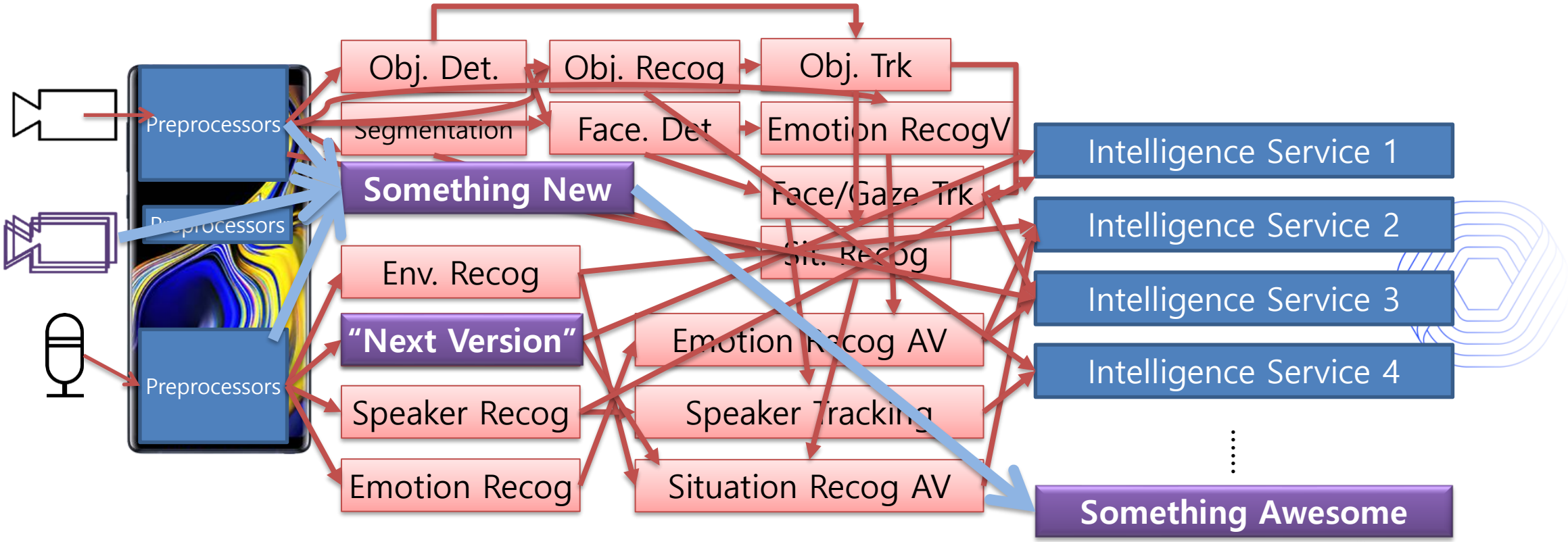


These examples do not reflect actual products

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 6. Dynamic Stream Pipelines



These examples do not reflect actual products

Motivation

Why **nnstreamer chose gstreamer?**

GStreamer Conference 2018

https://pixabay.com/photo-2490210/ MyungJoo Ham

DO NOT REINVENT THE WHEEL

https://pixabay.com/photo-2490210/ MyungJoo Ham

Do **NOT** Reinvent the Wheel

Do NOT Reinvent the Wheel

gstreamer for intelligence!

Neural networks on devices? Why? (a.k.a. edge-computing?)

- Real time processing with large data streams!
  - $ and time
- Privacy and data protection.
  - If you cannot protect it, don't store it.
- Neuromorphic processing units, DSPs, Neural processors, …
  - > O(100) TOPS on your hands soon!

Do NOT Reinvent the Wheel

gstreamer for intelligence!

Why gstreamer for neural networks on devices?

- You, gstreamer developers, have already solved what we are facing!
  - Large data stream processing in real-time, even in embedded devices (TVs, mobile devices, …)
  - Multi data stream synchronization with parallelism
  - Support complex stream pipeline topology with dynamicity
  - Various media preprocessors
  - Support custom plugins for new hardware, features, …
  - Various language bindings
  - Number of developers who are already familiar with
    - In Samsung, we've been using gstreamer for Tizen projects (All of TV, wearable, CE, … & some of mobile)
    - We have someone to consult in house!
  - …

Do NOT Reinvent the Wheel

gstreamer for intelligence!

Why gstreamer for neural networks on devices? / Continued

Neural networks are yet another stream filters! (Gst elements)

Don't Reinvent the Wheel,
PERFECT THE WHEEL!

https://pixabay.com/photo-2490210/ MyungJoo Ham

Don’t Reinvent the Wheel, **Perfect** it!

gstreamer for intelligence ➔ **nnstreamer**

What Gstreamer needs to understand “neural networks”

- Data Stream Standards for various neural network frameworks & models
- Methods to invoke various neural network framework & models
- Methods to convert between traditional GST data and neural network data
- Methods to handle neural network data streams

Don’t Reinvent the Wheel, **Perfect** it!

gstreamer for intelligence ➔ **nnstreamer**

What Gstreamer needs to understand “neural networks”

- Data Stream Standards for various neural network frameworks & models
- Methods to invoke various neural network framework & models
- Methods to convert between traditional GST data and neural network data
- Methods to handle neural network data streams

➔ nnstreamer

Don’t Reinvent the Wheel, **Perfect** it!

gstreamer for intelligence ➔ **nnstreamer**

➔ nnstreamer 

License: LGPL 2.1

Public Open: https://github.com/nnsuite/nnstreamer

2018.07: 0.0.1

2018.10: 0.0.2

Current: 0.0.3-rc1

Don’t Reinvent the Wheel, **Perfect** it!

gstreamer for intelligence ➔ **nnstreamer**

➔ nnstreamer 

License: LGPL 2.1

Public Open: https://github.com/nnsuite/nnstreamer

We will try to upstream to gst-plugins-good ☺

Don't Reinvent the Wheel, **Perfect** it!

gstreamer for intelligence ➔ **`nnstreamer`**

Benefits of nnstreamer @ gstreamer

- For **neural network** developers
- For **device/application** developers using neural networks
- For **gstreamer** developers

Don’t Reinvent the Wheel, **Perfect** it!

gstreamer for intelligence ➔ **`nnstreamer`**

What if Gstreamer understands neural networks…

- Neural Network Model / Device / Application Developers
  - Fast prototyping, testing, and releases of models with real-time sensor inputs

    Low Cost, High Efficiency, Short Time-to-market, High Reusability, … … …
  - “Create applications by describing the topology only?!!!”
    - Just bring your neural network models!

Don’t Reinvent the Wheel, **Perfect** it!

gstreamer for intelligence ➔ **nnstreamer**

What if Gstreamer understands neural networks…

- Multimedia Framework / Plugin Developers
  - Neural Network Models as Media Filters!

Upscaling, Picture Quality Improvement, Context Awareness, Style Transfer, ……

Concepts

How **nnstreamer**  works?

GStreamer Conference 2018

How GStreamer handle neural networks?

NN (neural network) is yet another media filter.

Data between NNs: Video Stream? Audio Stream? Text Stream? …

➔ “Tensor Stream”!

- Tensor : Multi-Dimensional Array

How GStreamer handle neural networks?

NN is yet another media filter. Tensor is yet another media type.

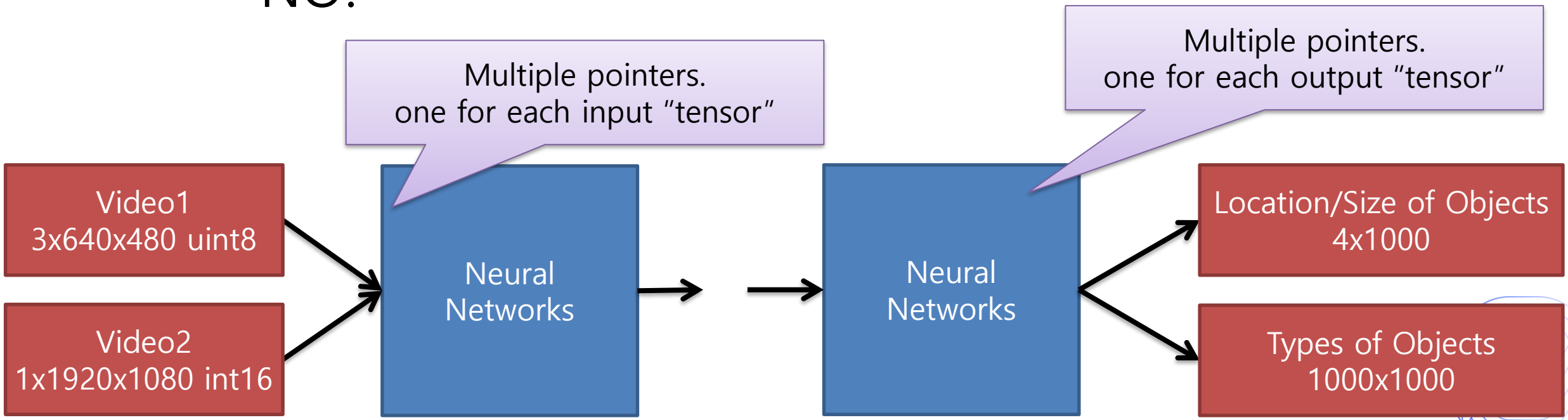
"Tensor" as "Data Stream"

```
other/tensor
        framerate: (fraction) [ 0/1, 2147483647/1 ]
        dim1: (int) [1, 65535]
        dim2: (int) [1, 65535]
        dim3: (int) [1, 65535]
        dim4: (int) [1, 65535]    # Up to 4-dimensions (v0.0.3)
        type: (string) { uint8, int8, uint16, int16, uint32, int32, uint64, int64, float32
, float64 }
```

Caps-and-Flow-Design "**GStreamer data types (pad capabilities)**"

Is "other/tensor" enough?

NO!

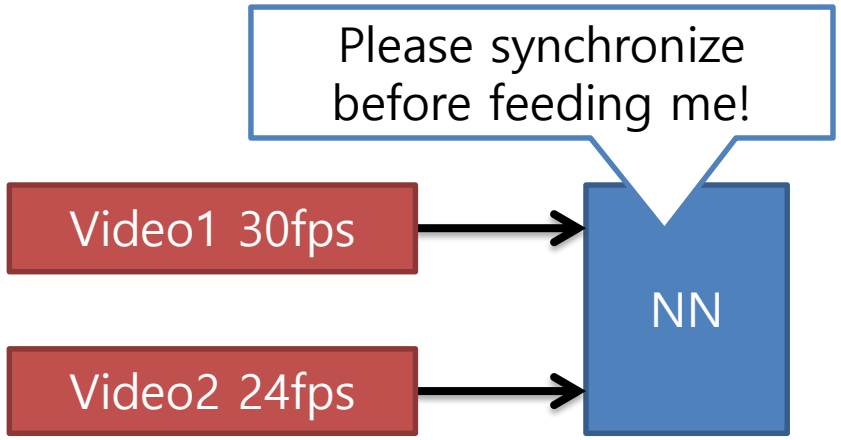


Need Container of Tensors!

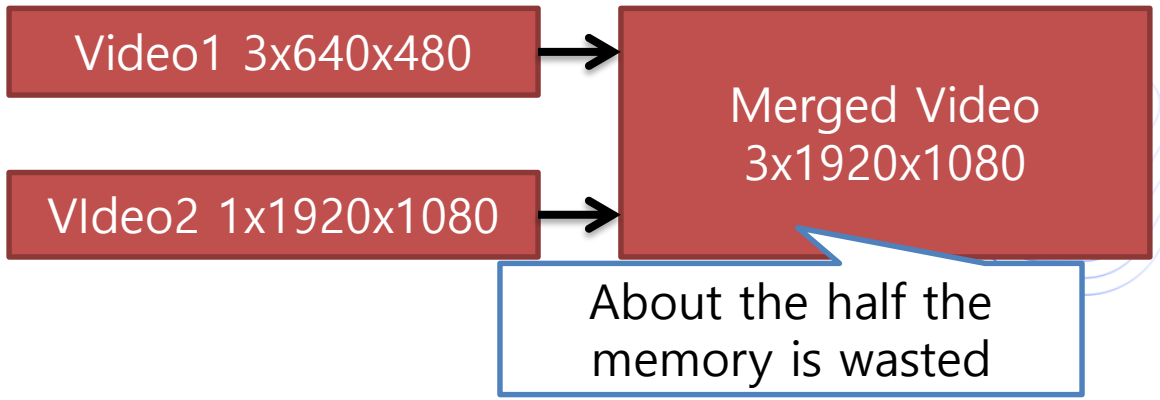
A Set of Tensors, How?

Suggested mechanisms…

- Multiple Streams, Each with a Tensor
  - Additional Synchronization Problems!!!



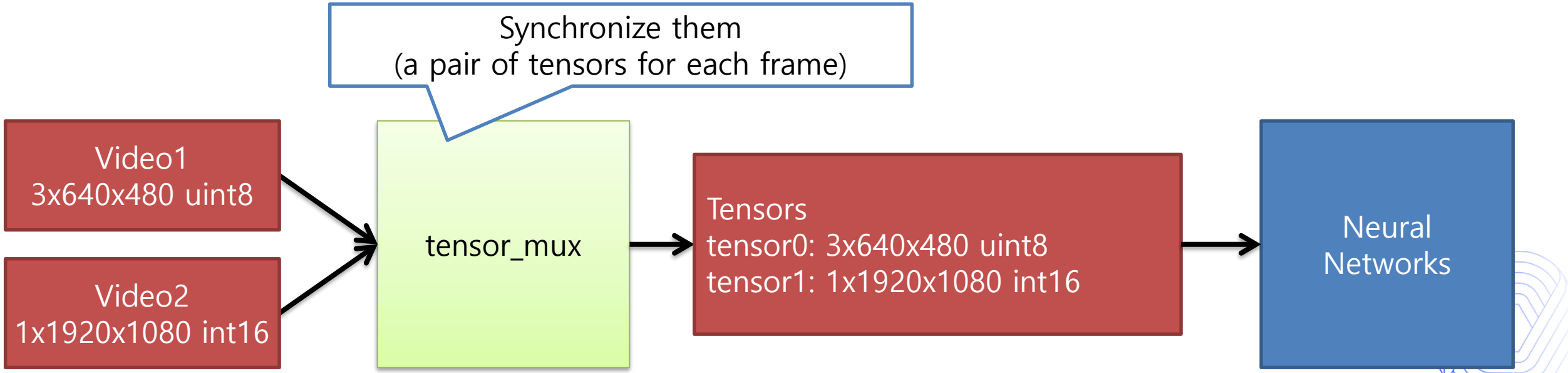
- Merge Tensors into a Tensor
  - 20x4 + 4x10 ➔ 20x10x2 or 24x10 or …
  - Inefficiency!



- Container of Tensors (“other/tensor**s**”)
  - **`nnstreamer`** v0.0.2 !

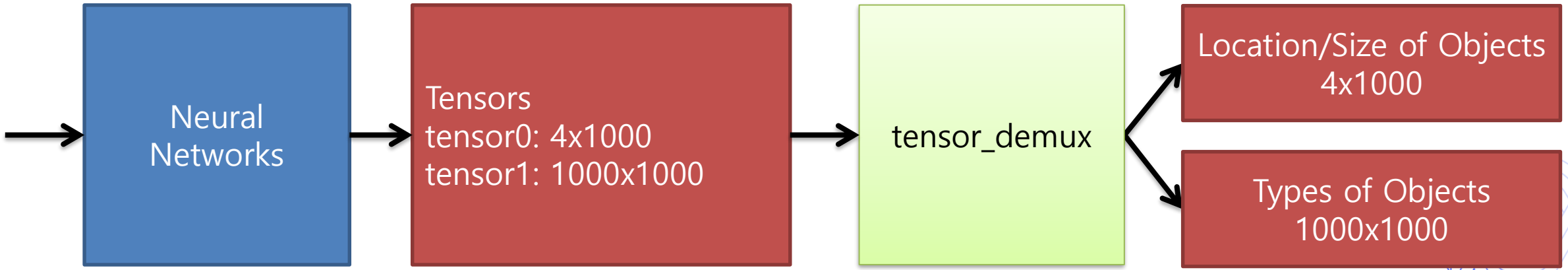
Tensors

other/tensors



Tensors

other/tensors



Tensors Performance? (Mux/Demux)

Any Performance Deterioration?

Memcpy-less mux & demux!

- GstBuffer Multi-memory-chunks!
- Thus, up to 16 Tensors in other/tensors

Tensors.

NN is yet another media filter. Tensor and Tensros are yet another media types.

"Tensor**s**" as Data Stream

```
other/tensors
        num_tensors = (int) [1, 16] # GST_MAX_MEMCHUNK_PER_BUFFER
        framerate: (fraction) [ 0/1, 2147483647/1 ]
        types = (string) Typestrings
        dimensions = (string) Dimensions

Typestrings = Typestring | Typestring, Typestrings
Dimensions = Dimension | Dimension, Dimensions

Typestring = (string) {float32, float64, . . . , int16, uint16, int8, uint8}
Dimension = (string) [1-65535]:[1-65535]:[1-65535]:[1-65535]
```

- Refer: nnstreamer wiki, "**GStreamer data types (pad capabilities)**"

Tensor File Format

other/tensorsave, ".tnsr" files

A file type definition for other/tensor & other/tensors

- NYI @ v 0.0.2
- Drafted at nnstreamer wiki.

nnstreamer plugins

What nnstreamer  actually does

GStreamer Conference 2018

nnstreamer plugins

Categories of Plugins

https://github.com/nnsuite/nnstreamer/blob/master/Documentation/component-description.md

- Converter
  - Conventional Media → Tensor
  - Tensor → Conventional Media

nnstreamer plugins

Categories of Plugins

https://github.com/nnsuite/nnstreamer/blob/master/Documentation/component-description.md

- Filter

  Invoke neural network frameworks and their models

  - Invoke Tensorflow, Caffe, TF-Lite, …
  - Invoke custom shared objects.

nnstreamer plugins

Categories of Plugins

https://github.com/nnsuite/nnstreamer/blob/master/Documentation/component-description.md

- Utility
  - Stream Pipeline Control
    - Mux, Demux, Merge, Split, Aggregate, ...
  - Tensor Transform
    - Add, Mul, Normalize, Sum, Transpose, Dimension-Change, Type-Cast, ...

nnstreamer plugins

Categories of Plugins

https://github.com/nnsuite/nnstreamer/blob/master/Documentation/component-description.md

- Connector:

  Connecting with Out-of-Gstreamer World

  - Non Audio/Video Sensors (LIDAR, RADAR, UWB, GPS, …)
  - Tensor-Sink to Applications
  - ROS-src / ROS-sink
  - …

nnstreamer plugins: tensor_converter

v0.0.1 (video) / v0.0.2 (all)

nnstreamer/gst/tensor_converter/

- Video (video/x-raw)
  - RGB/BGRx. Other Colorspaces TBD.
  - 3:width:height uint8 (RGB) (== uint8[1][height][width][3]) (rank 3)
- Audio (audio/x-raw)
  - Channel:Frames-per-buffer <type> (rank 2)
- Text (text/x-raw)
  - Frames-per-buffer uint8 (rank 1)
- Binary Blob (application/octet)
  - Pipeline writers should specify the dimensions.

nnstreamer plugins: **tensor_converter**

v0.0.1 (video) / v0.0.2 (all)

`nnstreamer/gst/tensor_converter/`

- Example:

```
(video stream) ... ! tensor_converter ! ... (tensor stream)
```

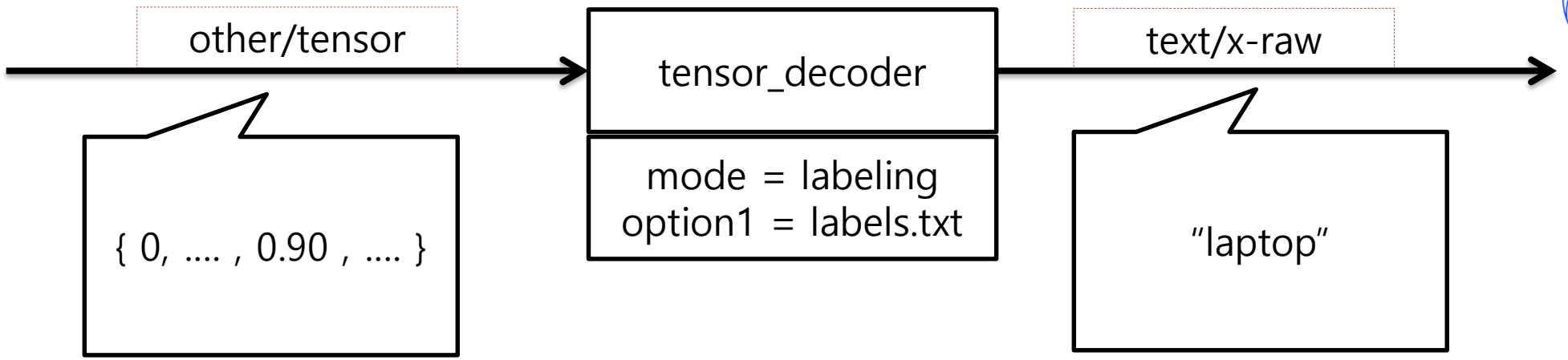
nnstreamer plugins: tensor_decoder

v0.0.2 (partial)

`nnstreamer/gst/tensor_decoder/  (Heavily WIP)`

- For 0.0.3: bounding box, object labeling, … (video output)
- Example:

`(tensor stream) ... ! tensor_converter ! ... (text/video/audio stream)`



nnstreamer plugins: **tensor_filter**

v0.0.1 (tf-lite/custom) / v0.0.2 (multi-tensor) / v0.0.3+ (more frameworks)

`nnstreamer/gst/tensor_filter/`

- A NN Framework supported as a "Sub-plugin".
  - 0.0.2: Tensorflow-Lite, Custom
  - 0.0.3+: Caffe/Caffe2, Tensorflow
  - Each sub-plugin can be added as a ".so" (e.g., tensor_filter_caffe.so)

- Custom subplugin
  - Neural network binaries with **nnstreamer** API
  - Refer: "`tensor_filter_custom.h`"

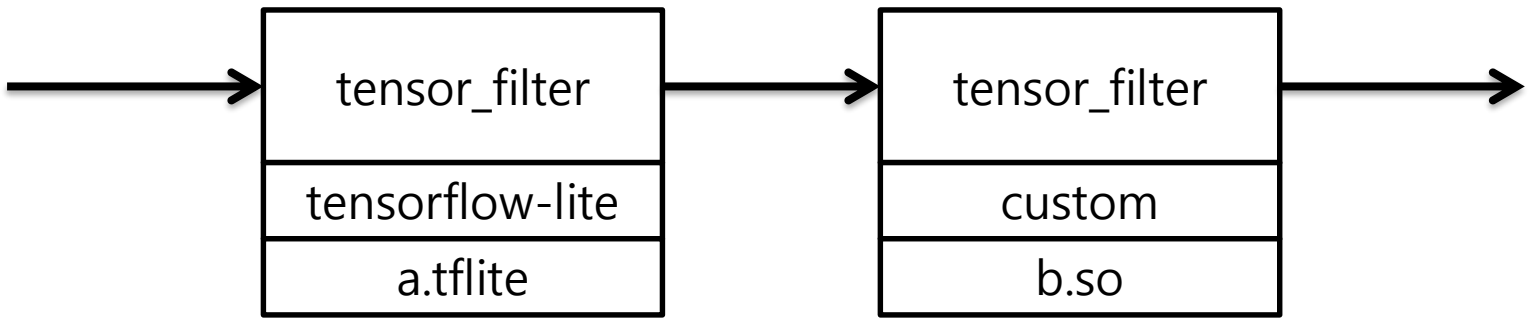
nnstreamer plugins: tensor_filter

v0.0.1 (tf-lite/custom) / v0.0.2 (multi-tensor) / v0.0.3+ (more frameworks)

`nnstreamer/gst/tensor_filter/`

- Example:

```
... ! tensor_filter framework=tensorflow-lite model=a.tflite !
      tensor_filter framework=custom model=b.so ! ...
```



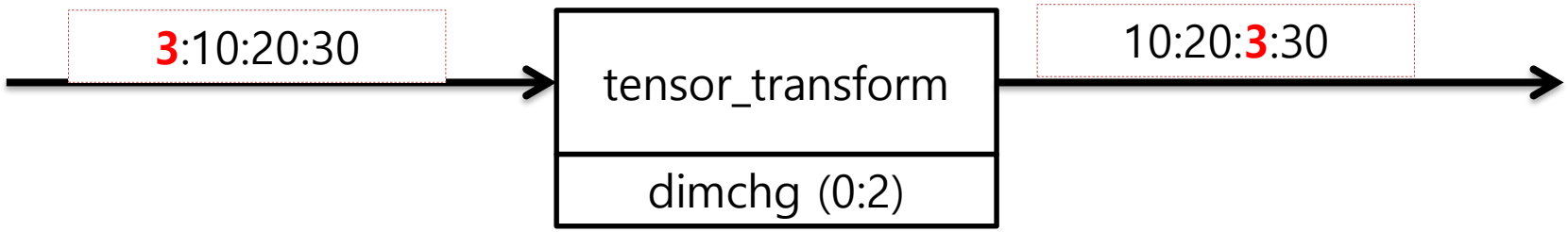
nnstreamer plugins: utility / **tensor_transform**

v0.0.1 +

`nnstreamer/gst/tensor_transform/ (Heavily WIP)`

- Transpose, Type-Cast, Normalization, Mul, Add, …
- WIP: SIMD-supprot / Multi-operator (e.g., “normalize/avg, mul/3, add/5”)
- Example:

```
... ! tensor_transform mode=dimchg option=0:2 ! ...
```



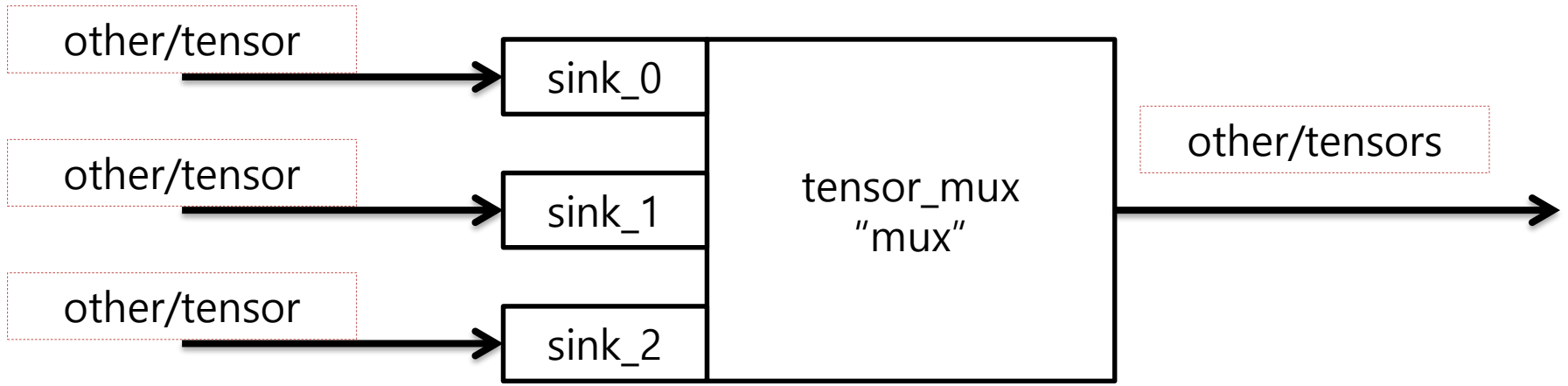
nnstreamer plugins: utility / **tensor_mux**

v0.0.2

`nnstreamer/gst/tensor_mux/`

- Create “other/tensors” from “other/tensor”s.
- Responsible: synchronization between tensors streams.
- Example:

```
... mux.sink_0 ... mux.sink_1 ... mux.sink_2 ... tensor_mux name=mux ! ...
```



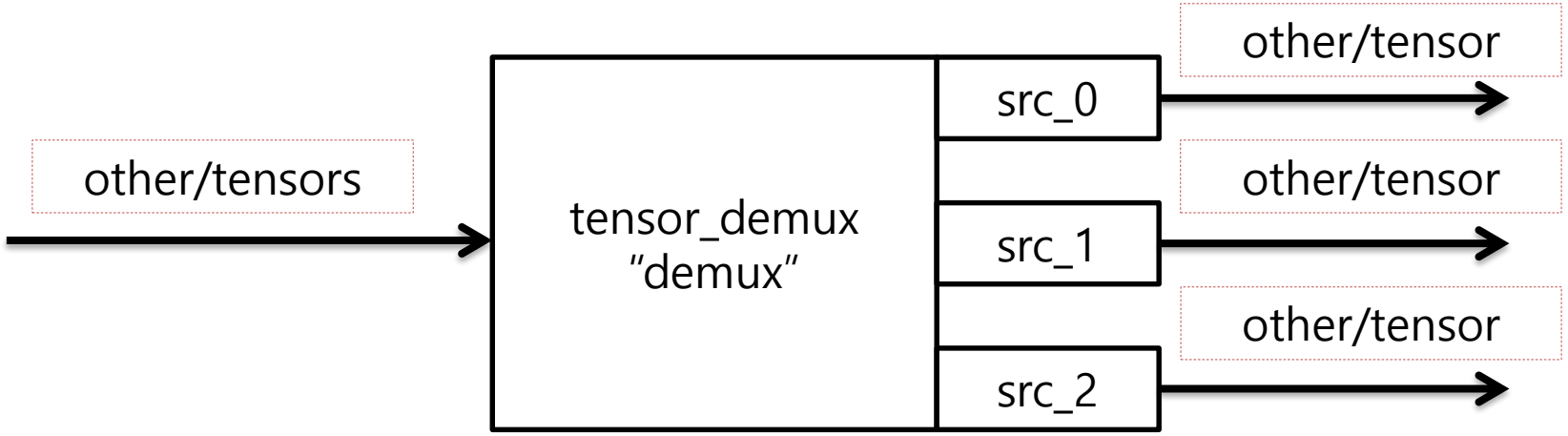
nnstreamer plugins: utility / **tensor_demux**

v0.0.2

`nnstreamer/gst/tensor_demux/`

- "other/tensor"s from "other/tensors"
- May choose specific "tensor" w/ tensorpick property.
- Example:

```
... ! tensor_demux name=demux ... demux.src_0 ... demux.src_1 ... demux.src_2
```



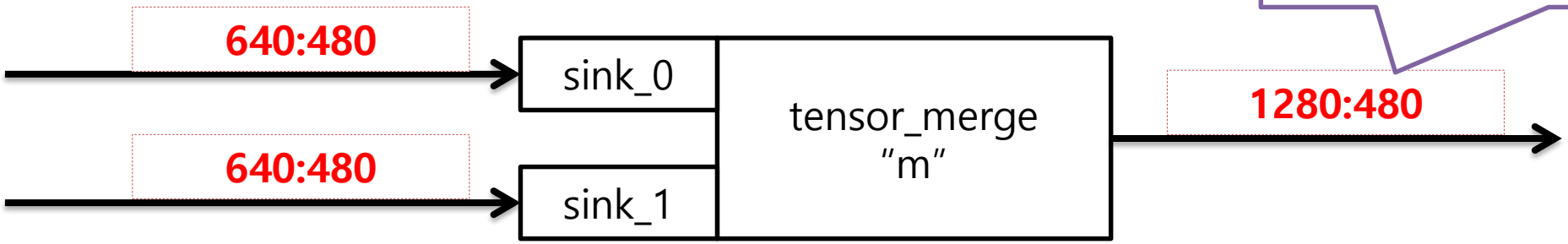
nnstreamer plugins: utility / **tensor_merge**

v0.0.2 (partial)

`nnstreamer/gst/tensor_merge/  (features implemented partially)`

- "other/tensor"s → Larger "other/tensor"
- A lot of memcpy. Proper for "stitching"
- Example:

```
... m.sink_0 ... m.sink_1 ... tensor_merge name=m ! ...
```



There will be other methods to "stitch" tensors. (NYI)

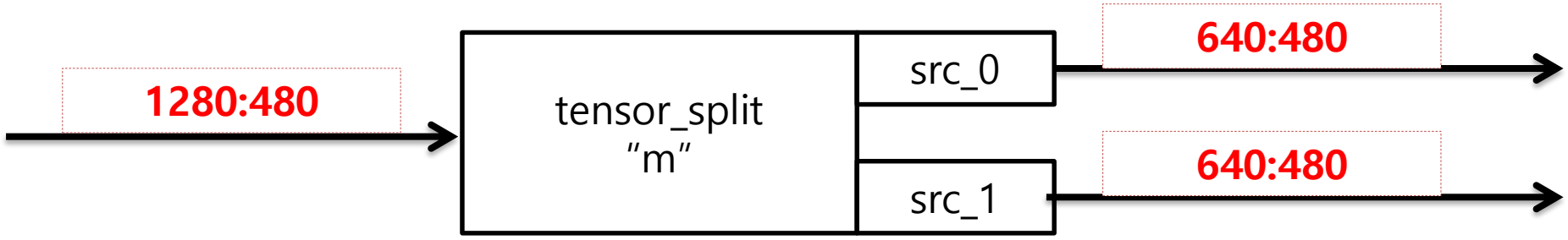
nnstreamer plugins: utility / **tensor_split**

future item

`nnstreamer/gst/tensor_split/  (NYI)`

- Cut “other/tensor” into smaller “other/tensor”s.
- A lot of memcpy. Proper for “cutting”
- Example:

```
... ! tensor_split name=m ! ... m.src_0 ... m.src_1 ...
```



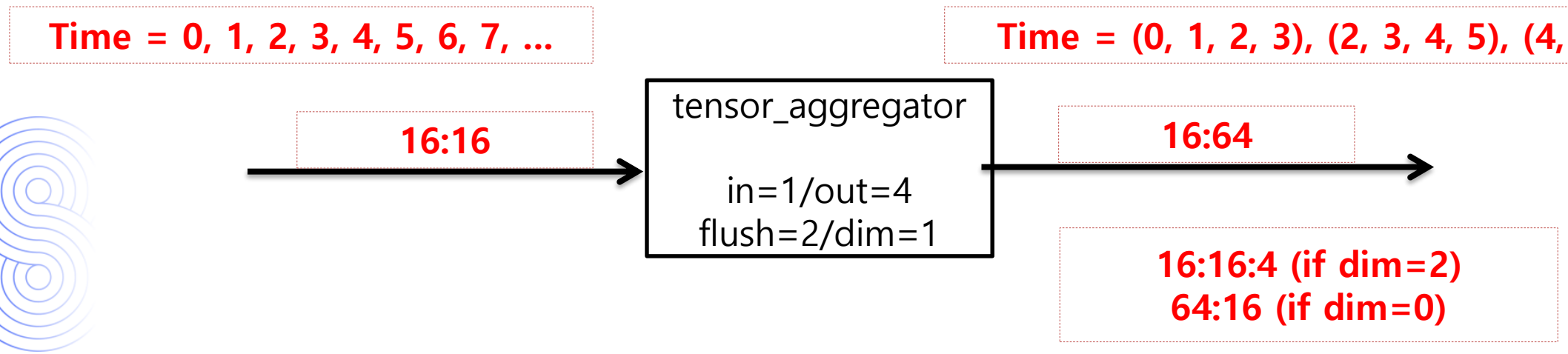
nnstreamer plugins: utility / **tensor_aggregator**

v0.0.2

`nnstreamer/gst/tensor_aggregator/` (features implemented partially)

- Aggregate frames in a sequence to a frame
  (e.g., aggregate recent 10 frames into 1 frame)
- Example:

```
... ! tensor_aggregator frames-out=4, frames-flush=2, frames-dim=1 !  ...
```



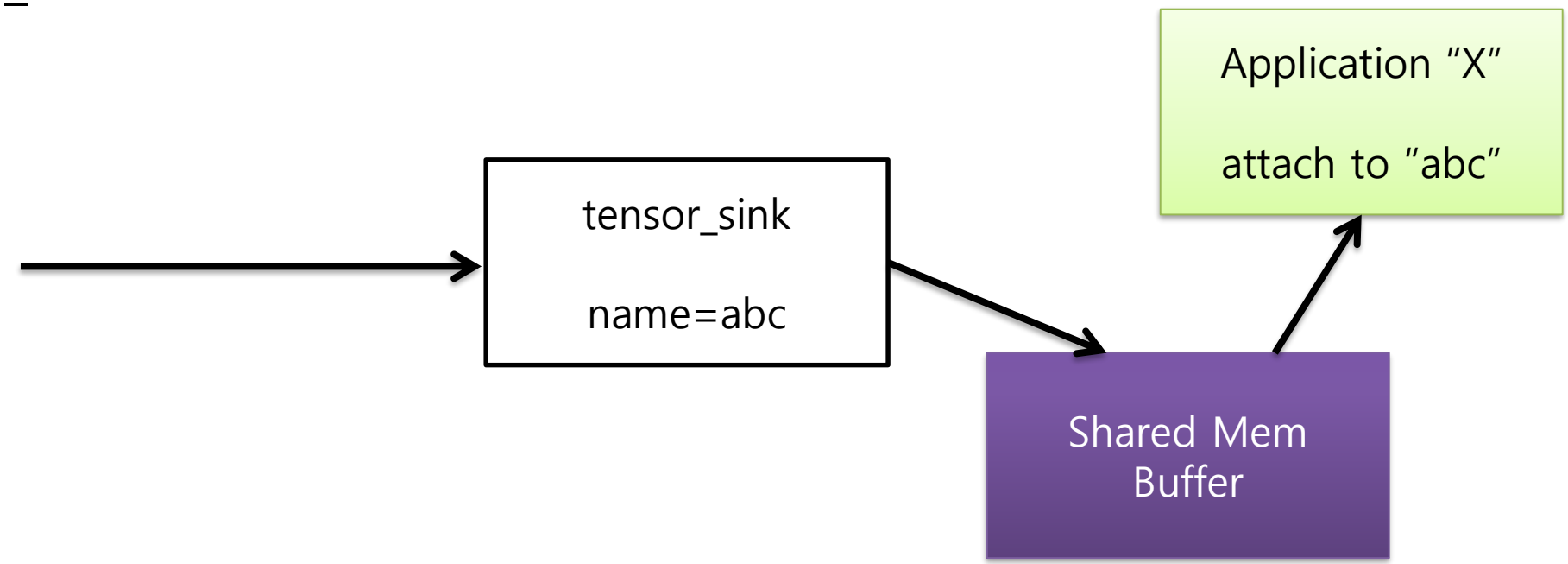
nnstreamer plugins: connector / **tensor_sink**

v0.0.1

`nnstreamer/gst/tensor_sink/`

- Connect with external application as “sink” via shared memory
- Example:

```
... ! tensor_sink name=abc
```



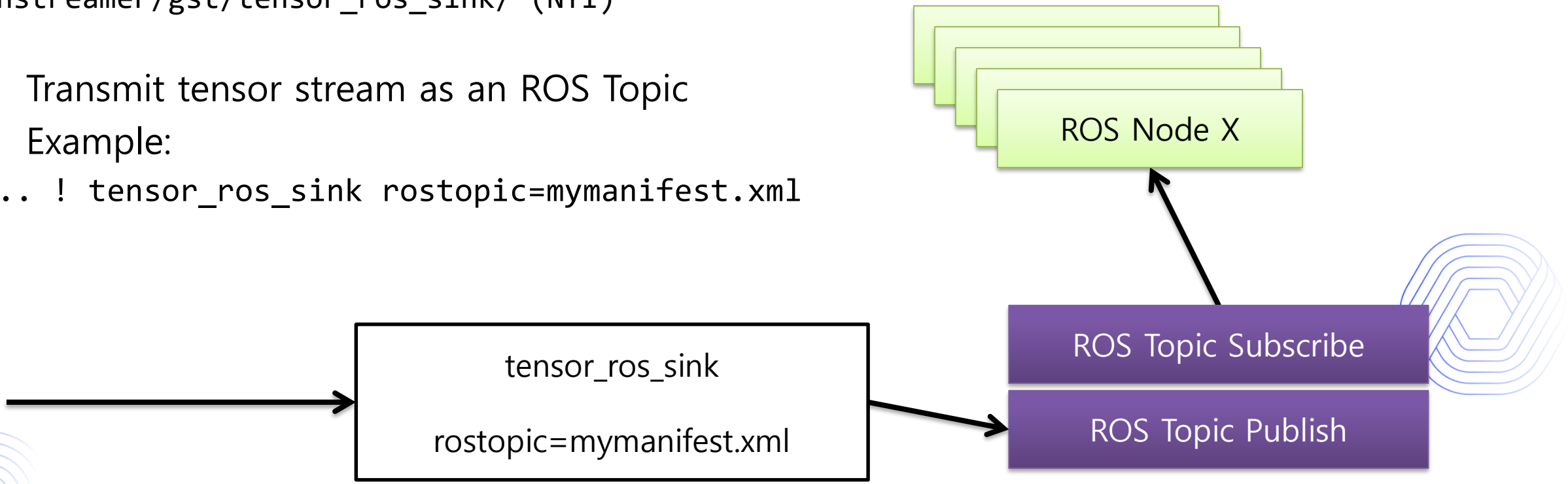
nnstreamer plugins: connector / **tensor_ros_sink**

Future Work

`nnstreamer/gst/tensor_ros_sink/ (NYI)`

- Transmit tensor stream as an ROS Topic
- Example:

```
... ! tensor_ros_sink rostopic=mymanifest.xml
```



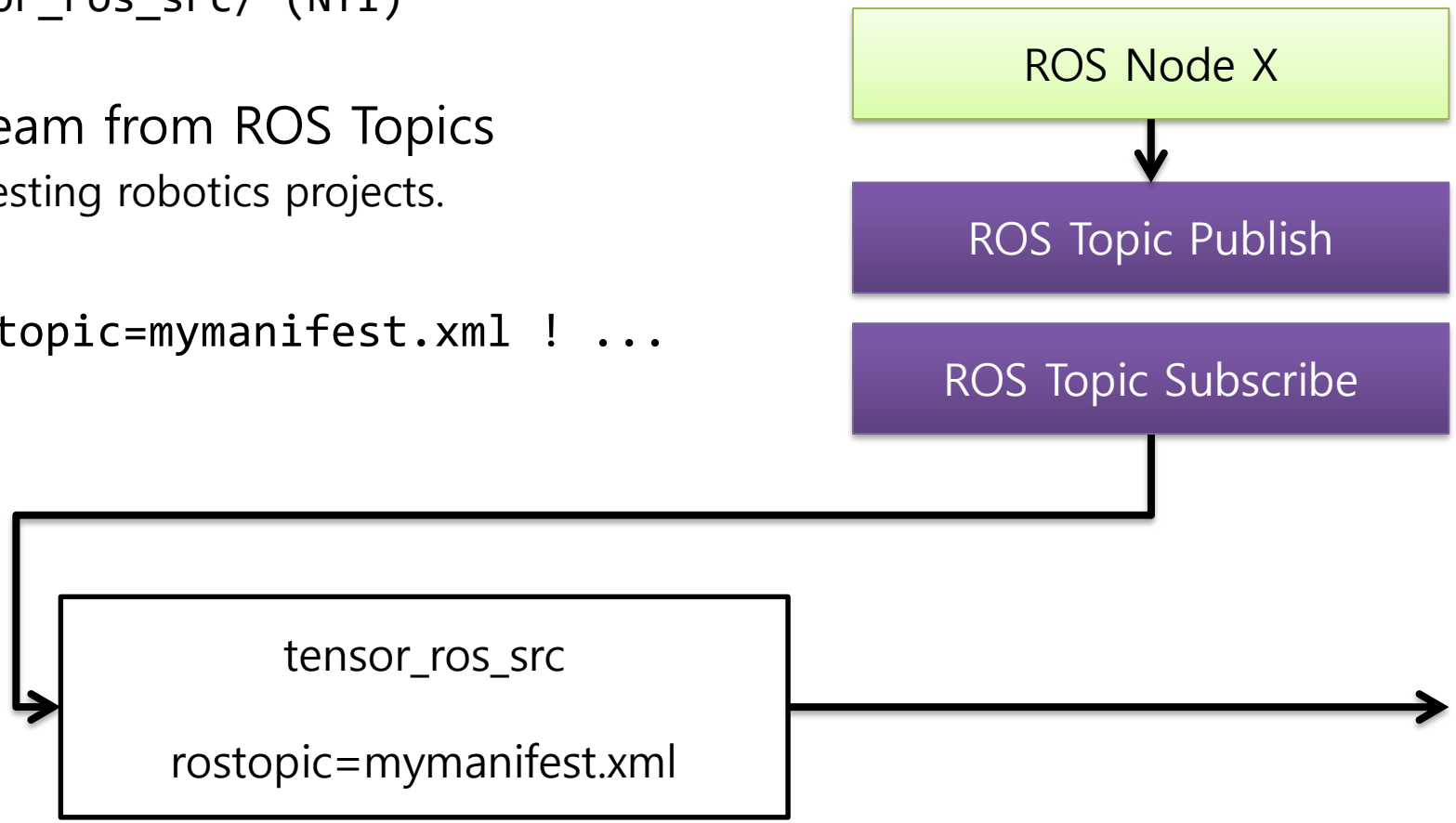
nnstreamer plugins: connector / **tensor_ros_src**

Future Work

`nnstreamer/gst/tensor_ros_src/ (NYI)`

- Create tensor stream from ROS Topics
  - E.g., rosbag for testing robotics projects.
- Example:

```
tensor_ros_src rostopic=mymanifest.xml ! ...
```



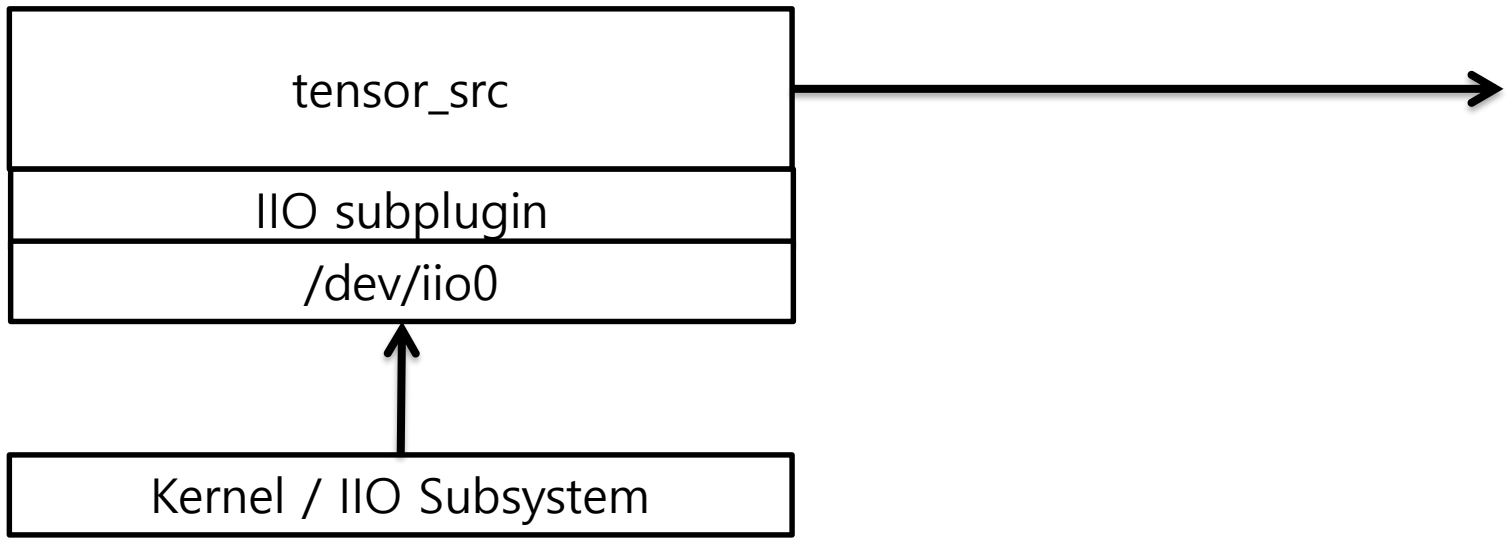
nnstreamer plugins: connector / **tensor_src**

Future Work

`nnstreamer/gst/tensor_src/ (NYI)`

- Tensor streams from other input sources / sensors
- Example:

```
tensor_src mode=IIO node=/dev/iio0! ...
```



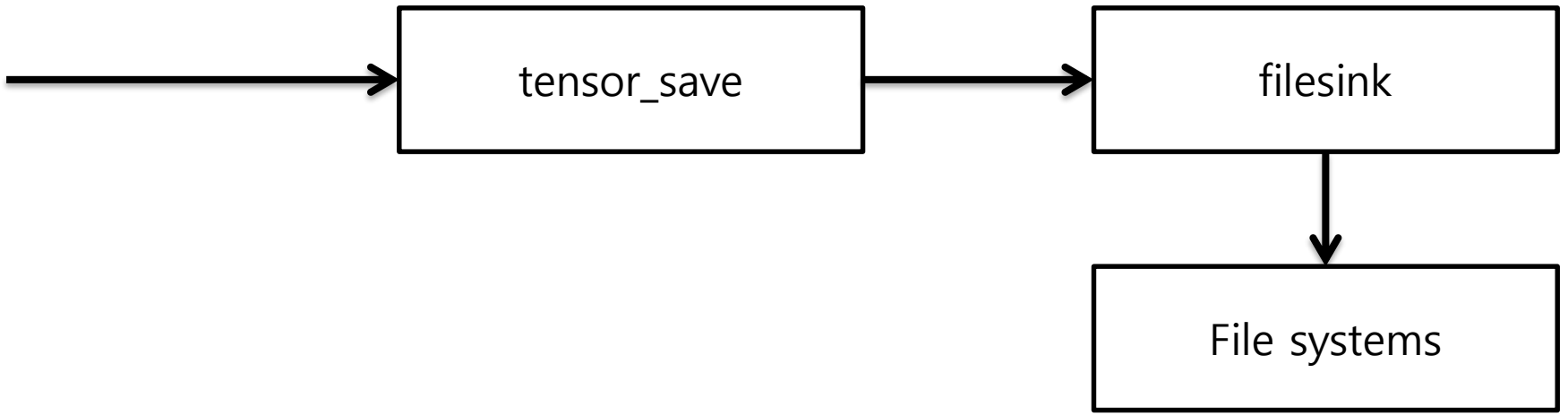
nnstreamer plugins: connector / **tensor_save**

Future Work

`nnstreamer/gst/tensor_saveload/ (NYI)`

- Tensor stream to files (.tnsr)
- Example:

```
... ! tensor_save ! filesink location=abc.tnsr
```



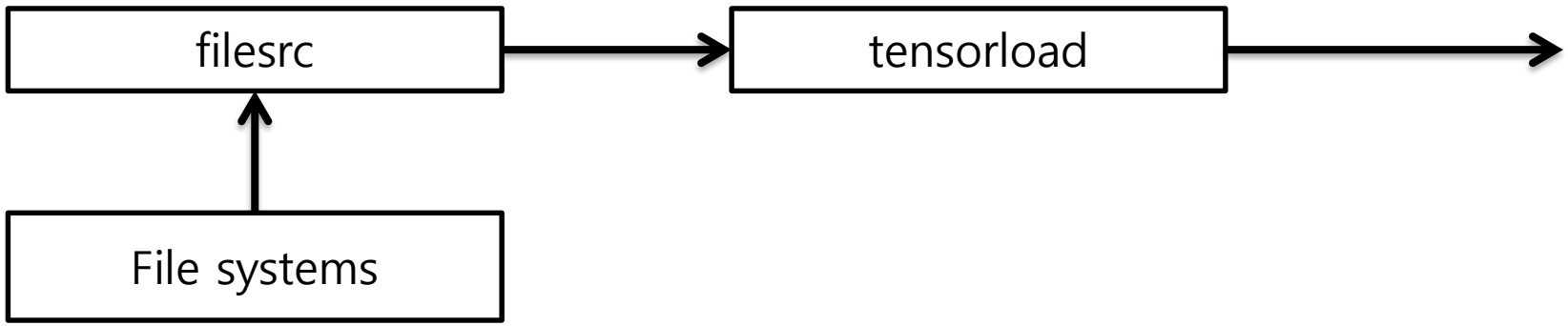
nnstreamer plugins: connector / **tensor_load**

Future Work

`nnstreamer/gst/tensor_saveload/ (NYI)`

- File to Tensor Stream (tensor/tensors)
- Example:

```
1. filesrc location=abc.tnsr ! tensor_loadsave ! ...

2. filesrc location=abc.tnsr ! autodecode ( “tensorsave” Gstreamer 표준 채택시)
```



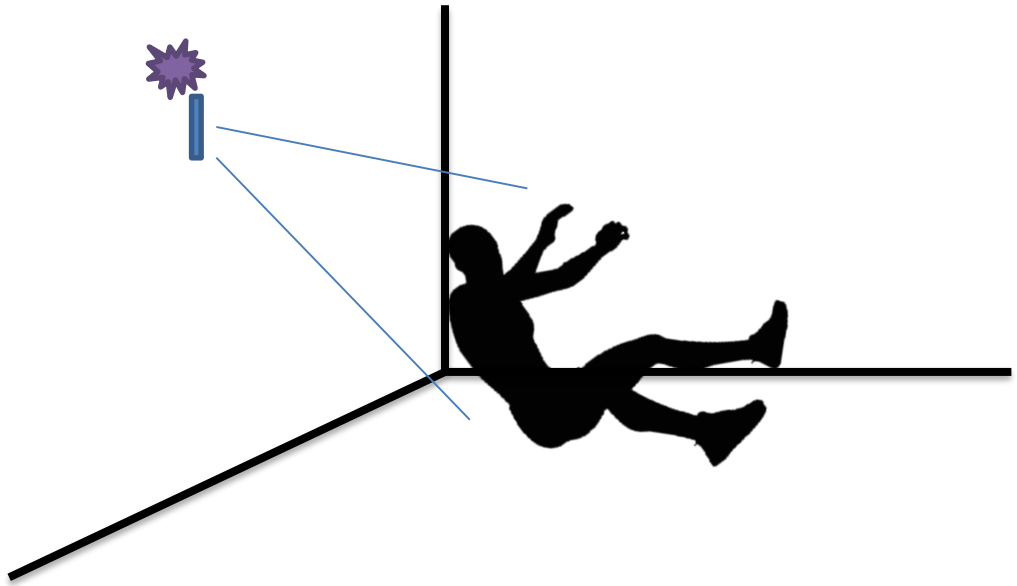
Usage example

nnstreamer  in action

GStreamer Conference 2018

Fall Detection Device, Elderly Care

Usage Example in Products



Images from https://pixabay.com/ko/%EC%8B%A4%EB%A3%A8%EC%97%A3-%EA%B0%80-%EB%96%A8%EC%96%B4%EC%A7%80%EB%8A%94-%EC%B6%95%EC%86%8C-%EC%84%9C%EC%82%AC%EC%8B%9C-%EB%94%94%EC%9E%90%EC%9D%B8-%EC%95%84%EC%9D%B4%EC%BD%98-%EA%B0%9C%EB%85-3172200/

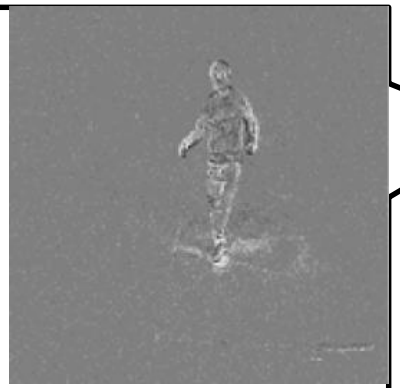
Fall Detection Device, Elderly Care

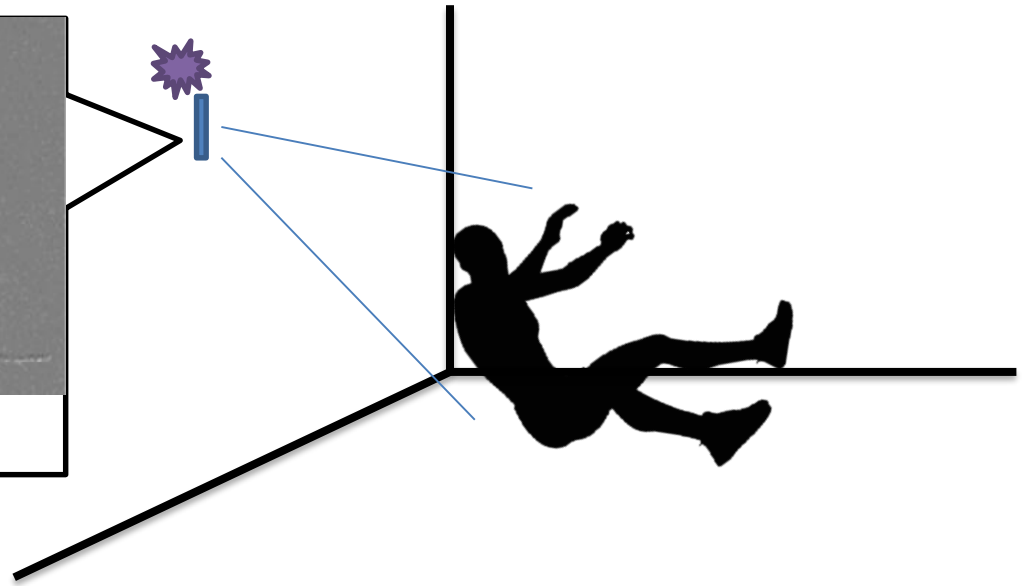
Usage Example in Products

DVS Input

(Dynamic Vision Sensor)

(Samsung Artik 5 IoT)





- H. Lee, et. al., "Embedded Real-Time Fall Detection Using Deep Learning For Elderly Care"

Images from https://pixabay.com/ko/%EC%8B%A4%EB%A3%A8%EC%97%A3-%EA%B0%80-%EB%96%A8%EC%96%B4%EC%A7%80%EB%8A%94-%EC%B6%95%EC%86%8C-%EC%84%9C%EC%82%AC%EC%8B%9C-%EB%94%94%EC%9E%90%EC%9D%B8-%EC%95%84%EC%9D%B4%EC%BD%98-%EA%B0%80%EB%9C%B5-3172200/

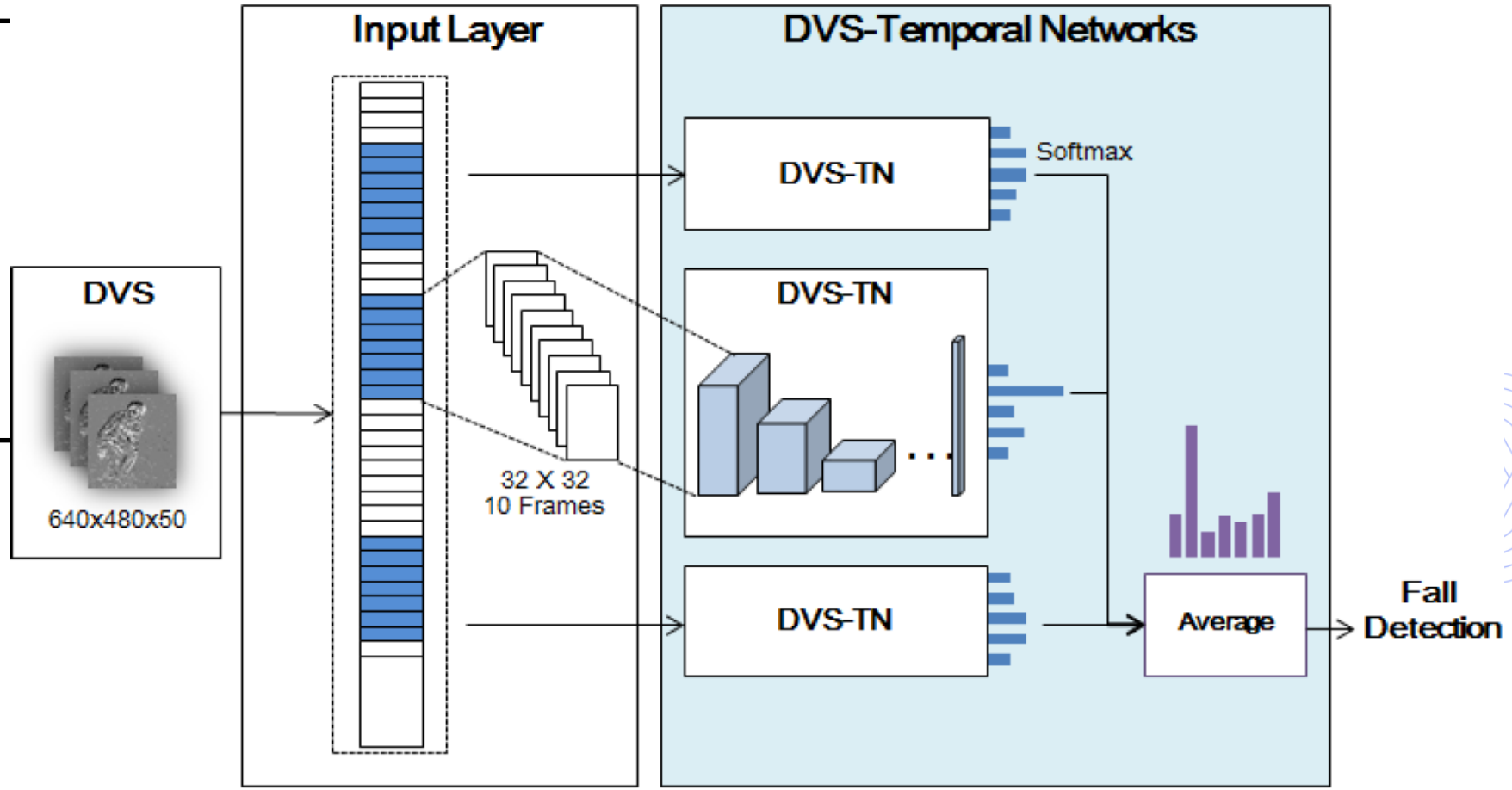
Fall Detection Device, Elderly Care

Usage Example in Products

DVS Input

(Dynamic Vision Sensor)

(Samsung Artik 5 IoT)

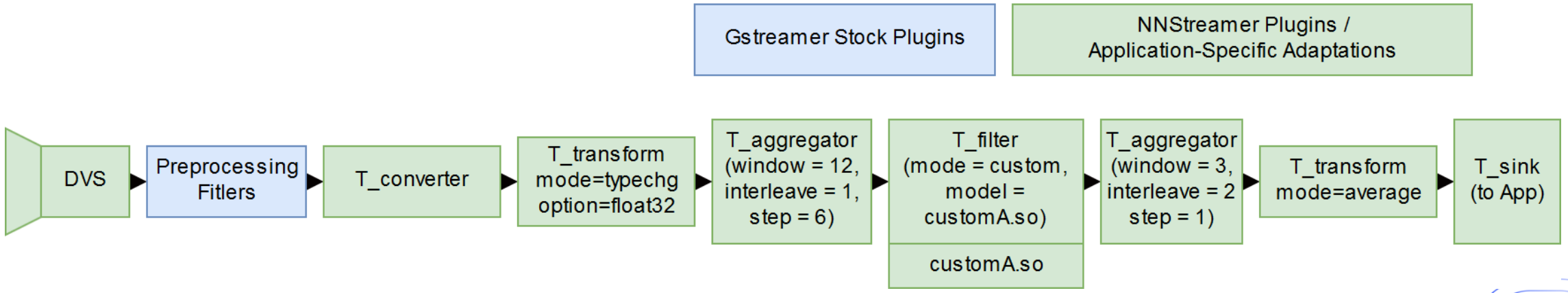


- H. Lee, et. al., "Embedded Real-Time Fall Detection Using Deep Learning For Elderly Care"

Images from https://pixabay.com/ko/%EC%8B%A4%EB%A3%A8%EC%97%A3-%EA%B0%80-%EB%96%A8%EC%96%B4%EC%A7%80%EB%8A%94-%EC%B6%95%EC%86%8C-%EC%84%9C%EC%82%AC%EC%8B%9C-%EB%94%94%EC%9E%90%EC%9D%B8-%EC%95%84%EC%9D%B4%EC%BD%98-%EA%B0%80%EB%9E%8C-3172260/

Fall Detection Device, Elderly Care

실 적용 사례



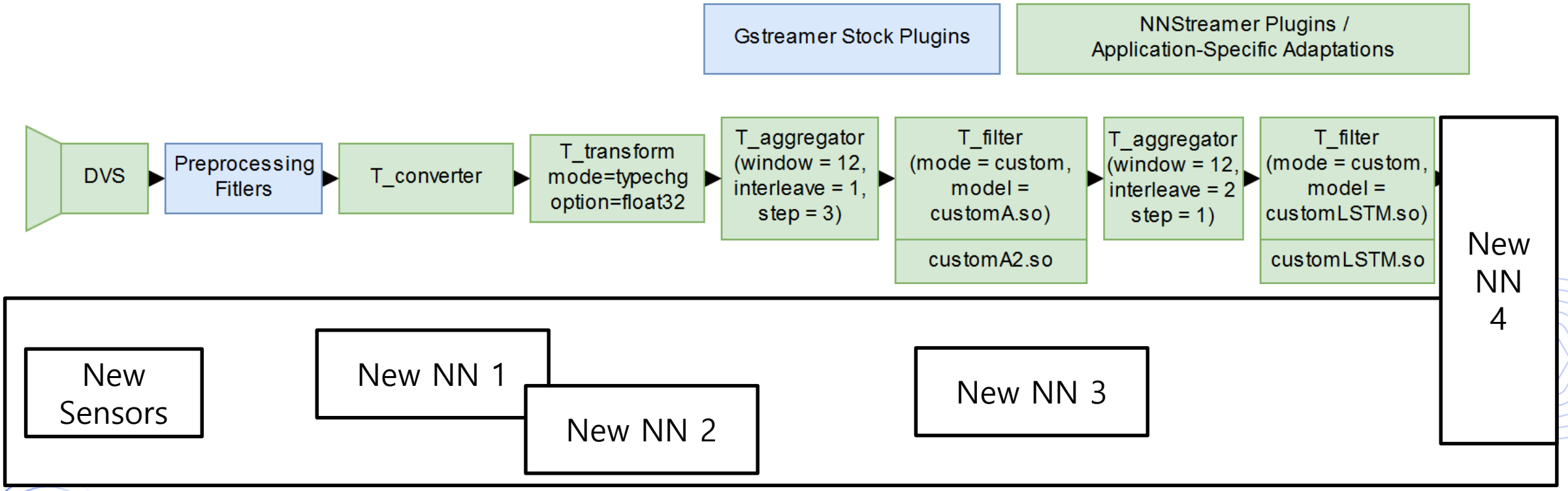
With nnstreamer+gstreamer

- Code Productivity! O(1000) lines → O(10) lines
- Performance (CPU Load)! < 1/4

➔ Enable New Features & More Complex Neural Network Models

Fall Detection Device, Elderly Care, "Next-Gen"

실 적용 사례



- Add LSTM, More CNN, New Sensor, New NNs, …
- Development Time: Several weeks x several developers → Several days x 1 developer
- High Performance. Still available for more features.

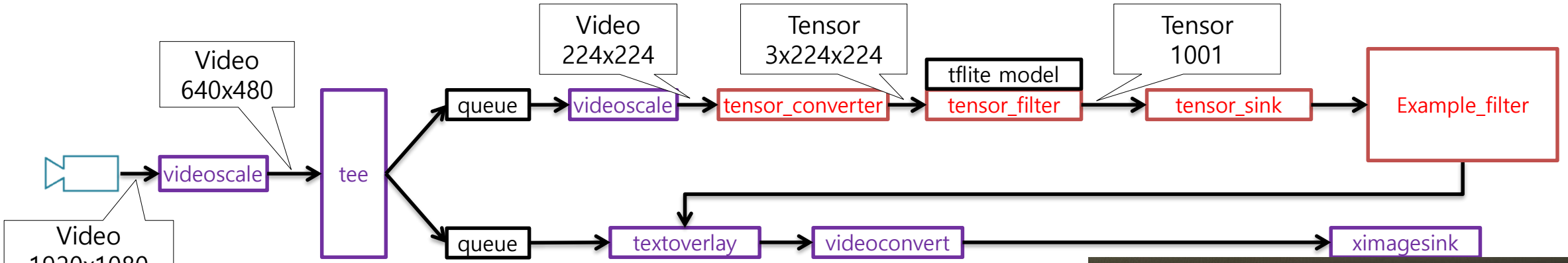
Demonstration

nnstreamer  live action

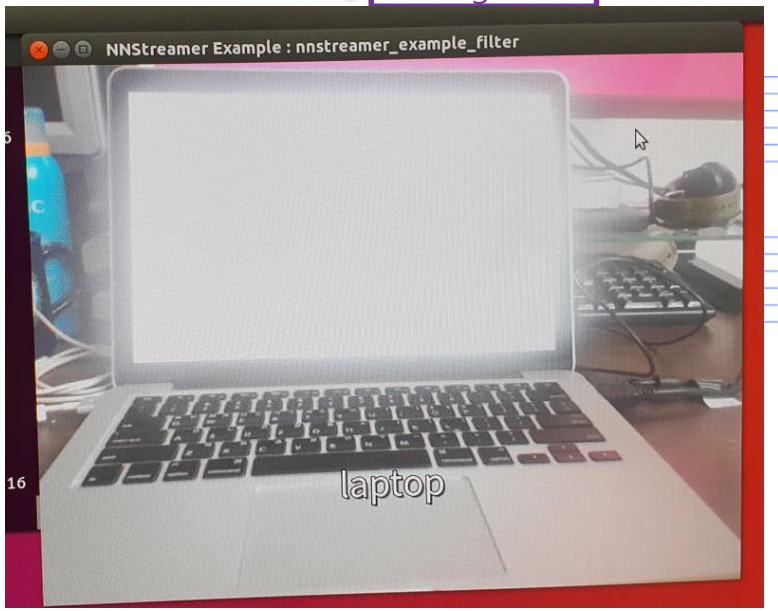
GStreamer Conference 2018

Demonstration

https://github.com/nnsuite/nnstreamer : nnstreamer_example/example_filter (C & Python)

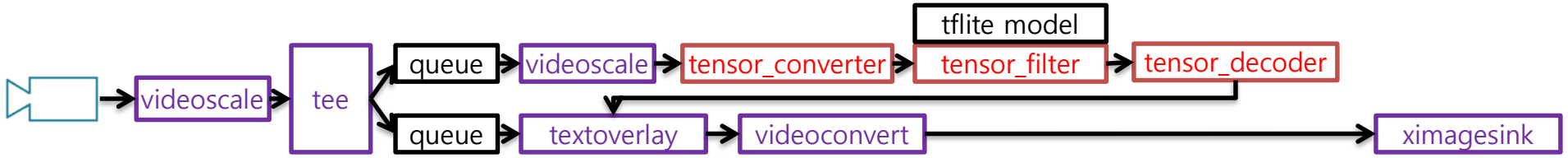


```
$ ./example_filter

or

$ python example_filter.py
```



Demonstration

nnstreamer @ Ubuntu Demo




```
$ gst-launch-1.0    v4l2src name=cam_src ! videoscale ! \   # "cam_src" / scale
      video/x-raw,width=640,height=640,format=RGB ! \       # 3x640x480 RGB
      tee name_t_raw \                                      # Tee the Stream
      t_raw. ! queue ! videoscale ! video/x-raw,width=224,height=224 ! \   # 224x224 RGB for mobilenet
            tensor_converter ! \                                          # "tensor"
            tensor_filter framework=tensorflow-lite model=mobilenetssd.tflite ! \ # invoke tflite
            tensor_decoder mode=labeling option1=labels.txt ! txt. \      # decode the result ➔ txt
      t_raw. ! queue ! textoverlay name=txt ! \                           # txt + video
            videoconvert ! ximagesink                                     # sink
```

CI / CD & How to Contribute

How **nnstreamer**  is developed

How you can contribute

GStreamer Conference 2018

nnstreamer CI: TAOS-CI

Continuous Integration (CI)

Every Github Pull-Request is required to pass CI system

- Anti-Regression
  - Build & Unit Test (Ubuntu-x64, Tizen-x64, Tizen-armv7, Tizen-arm64 (coming soon!))
- Coding Style/Rules
- Integration
  - Tizen/Ubuntu Packaging & UI App Test (with loopback camera & virtual display)

- G. Lim, M. Ham, J. Moon, W. Song, S. Woo, S. Oh, "TAOS-CI: Lightweight & Modular Continuous Integration System for Edge Computing", ICCE, 2019 (Accepted)

nnstreamer CD

Continuous Deployment (CD)

Binary Package Release via CD system

- Ubuntu: launchpad PPA (PPA:nnstreamer/ppa)
- Tizen: OBS
  - nnstreamer packages not opened to public yet
- WIP: Preparing other distros via build.opensuse.org

How to Contribute?

You are welcomed to contribute!

Github (github.com/nnsuite/nnstreamer)

- Communicate via [Issues]
- Code suggestion via [Pull Request]
  - Please check coding style/rule, build-test, unit-test, integration-test before submission.
    - The CI infrastructure has low capacity, yet. Do not use CI as your test machine.
  - New feature requires its own unit test cases.
  - Bug fix requires its own regression test cases.

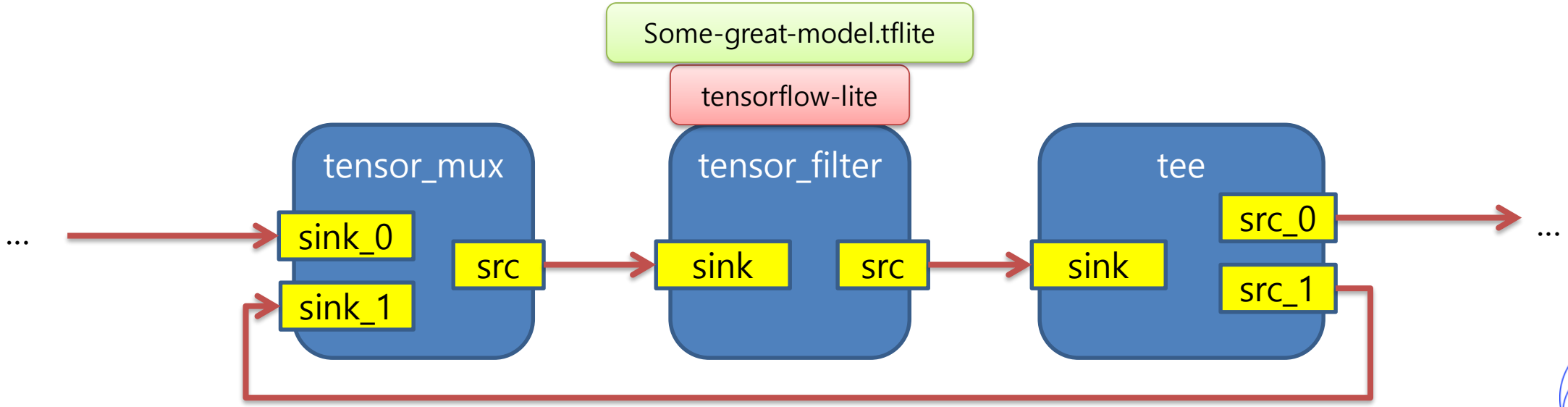
Work In Progress / Plans

Future of **nnstreamer**

GStreamer Conference 2018

Work-In-Progress

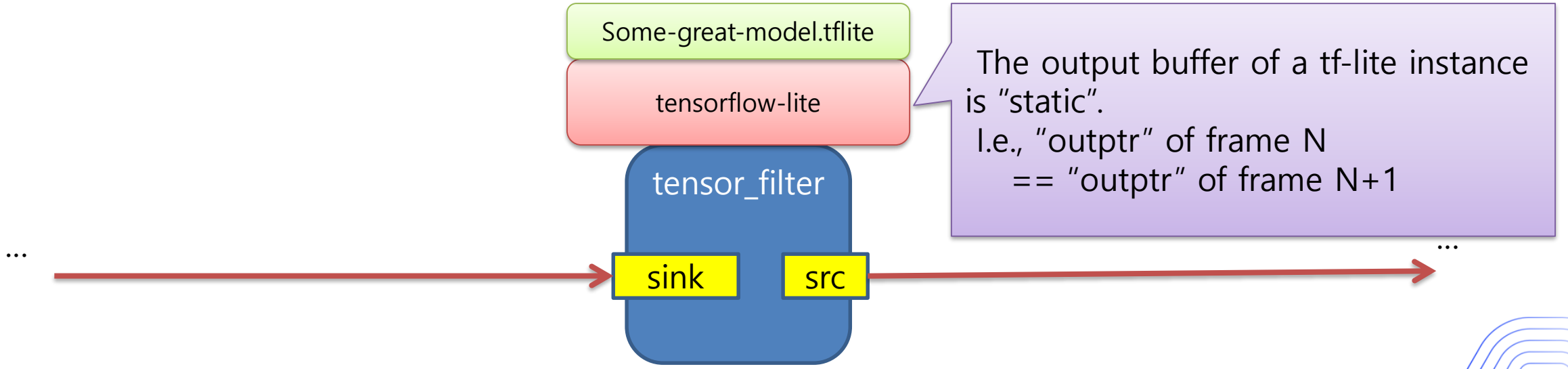
Major Issue: Recurrent Networks



- TIMESTAMP?
- Synchronization? (getting proper frame for sink_1)
-

~~Work-In-Progress~~ (Pending Pull-Request)

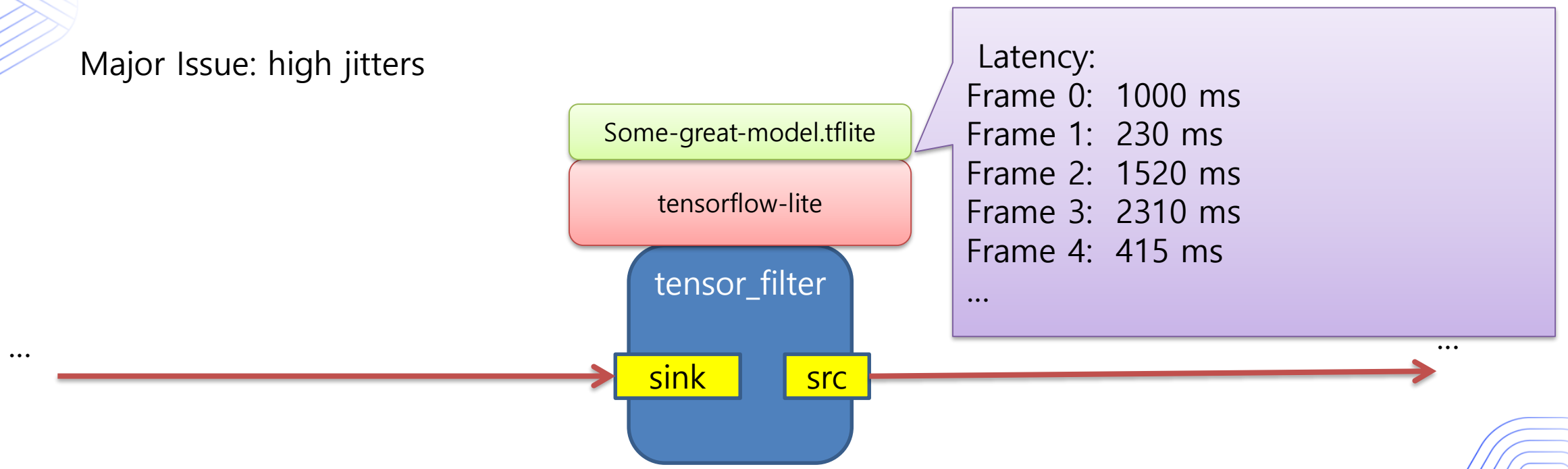
Major Issue: memcpy of the output buffer



- Current workaround: "memcpy" for each frame of tf-lite output buffer.
  - Avoid "tearing" at the next element.
- Suggested solutions:
  - Double/Triple buffering with multiple instances of tensorflow-lite for each instance.
  - Update & upstream tensorflow-lite to allocate output buffers for each frame optionally.

Work-In-Progress

Major Issue: high jitters

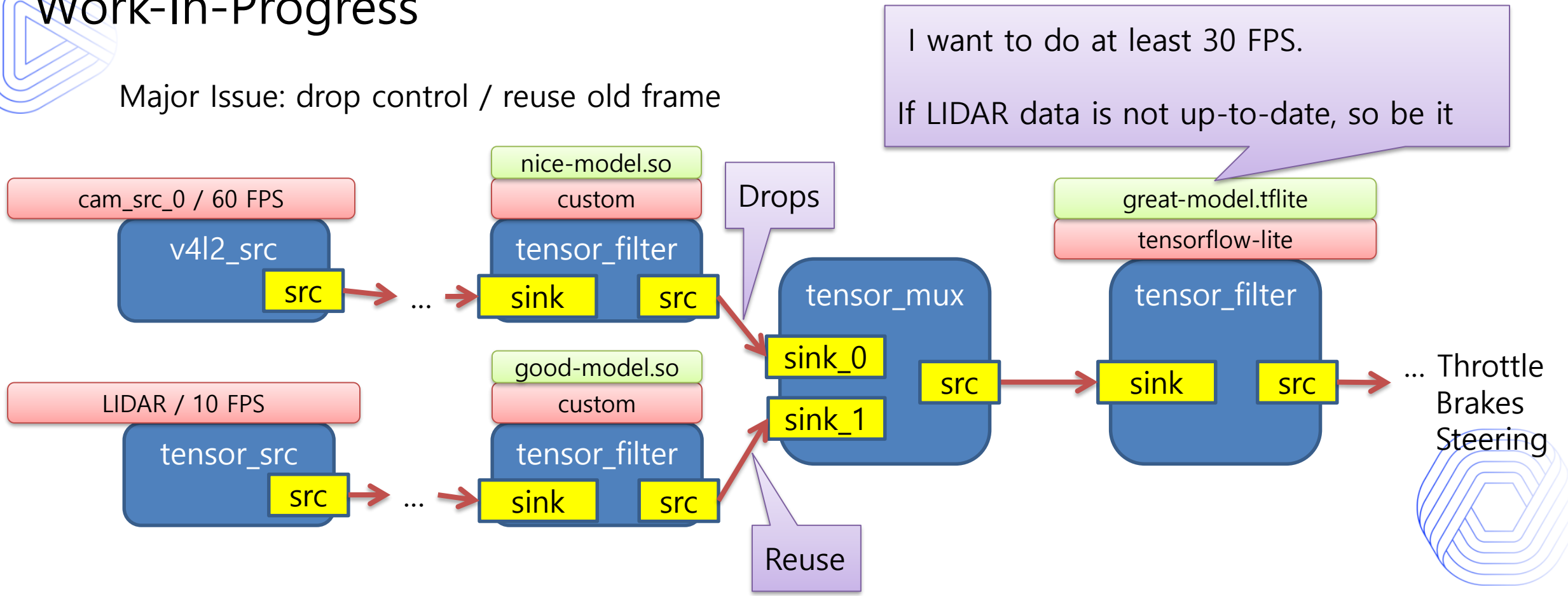


- How are we going to configure framerate? How do we synchronize at “mux” after src?

(we are still GST newbies…)
Previous products w/ nnstreamer had low jitter.
But, prospective products won’t.

Work-In-Progress

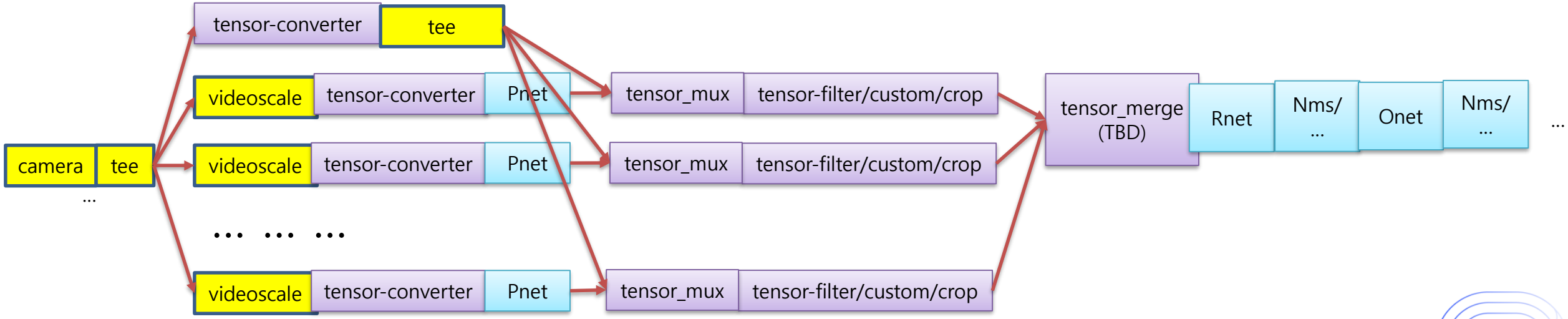
Major Issue: drop control / reuse old frame



- Intentional frame drops / old-frame reuse at “tensor_mux” and “tensor_merge”.
  Especially with high jitters from tensor_filters.

Work-In-Progress / MTCNN as GST Pipeline

Major Issue: we just need to implement some NYI plugins



- Expected to accelerate by easily and fully parallelizing the NN models.
- Keep adding more neural networks w/ tee, proving additional services
  : "Gaze tracking", "Emotion", ...

Work-In-Progress & Plans

OS Integration

- 2018
  - Tizen: the standard target. Tested: arm/x64 / Built (no test): arm64/x86-32
  - Ubuntu: release via PPA. Tested: x64
- 2019
  - Tizen: full SW platform integration & release
  - Android: secondary standard target. Tested: arm or arm64
  - Windows/macOS: built tested only: x64
  - Other Linux Distro: release via openSUSE OBS.

Work-In-Progress & Plans

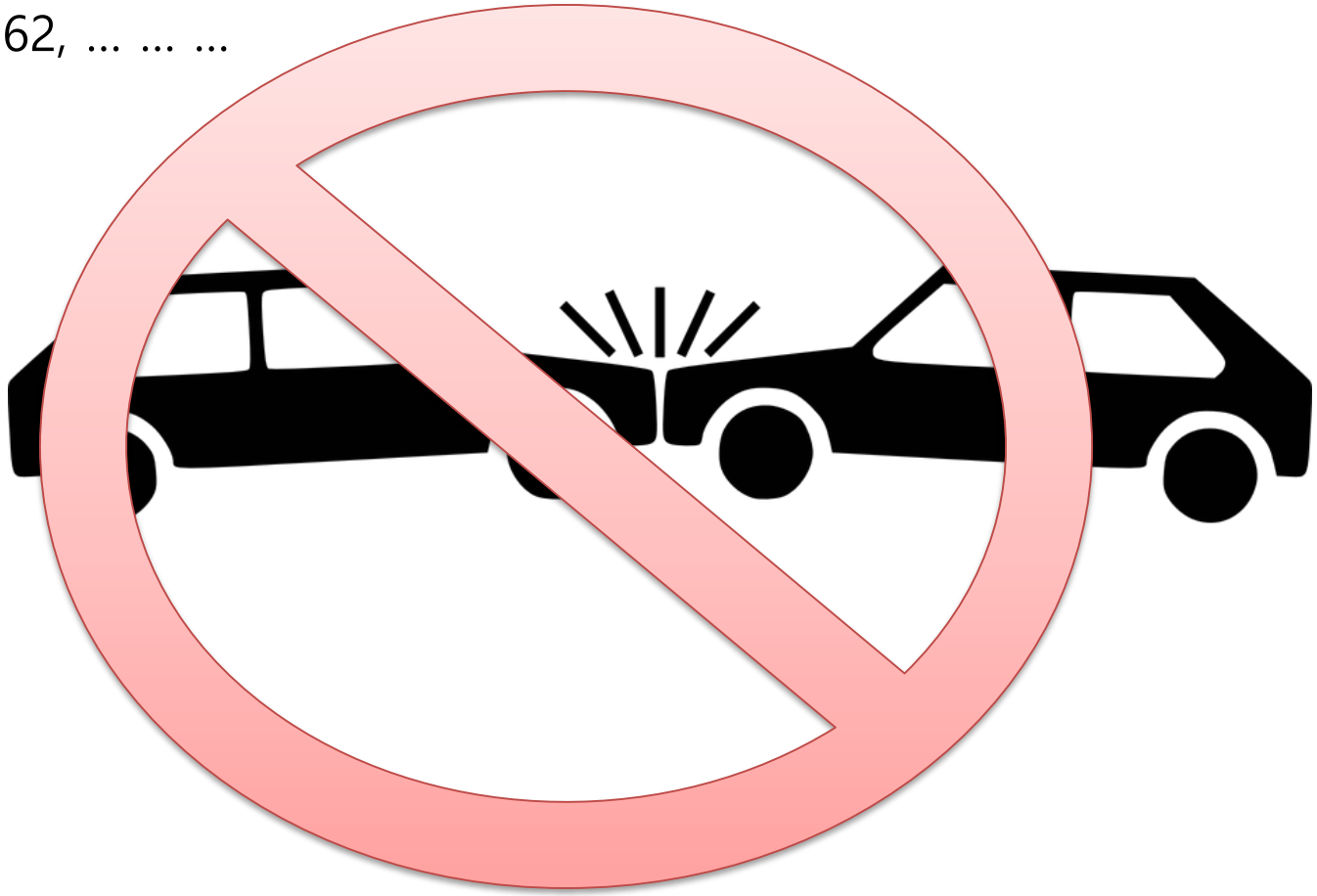
Features

- 2018
  - Tensor_filter: tensorflow
  - Tensor_ros_sink: send ROS topics (pub)
  - Tensor_decoder: bounding boxes, object labeling
  - Tensor_merge: more options required
- 2019
  - Tensor_filter: caffe/caffe2, LSTM (recurrent net) support
  - Tensor_transform: SIMD support. Multi-op support
  - Tensor_source: IIO sensors, LIDAR/RADAR support
  - Tensor_ros_source: receive ROS topics (sub): "FULLY DISTRIBUTED GST PIPELINE!"
  - Tensor_decoder: more incoming!
  - Tensor_save/load
  - Try to become **gst-plugins-good**!
- 2020
  - Support training phases
  - Multi-computer/board support

Plans ?

Automotive Industry

- ASIL-B/D, ISO-26262, … … …



https://pixabay.com/photo-151668/

Conclusion

nnstreamer 

GStreamer Conference 2018

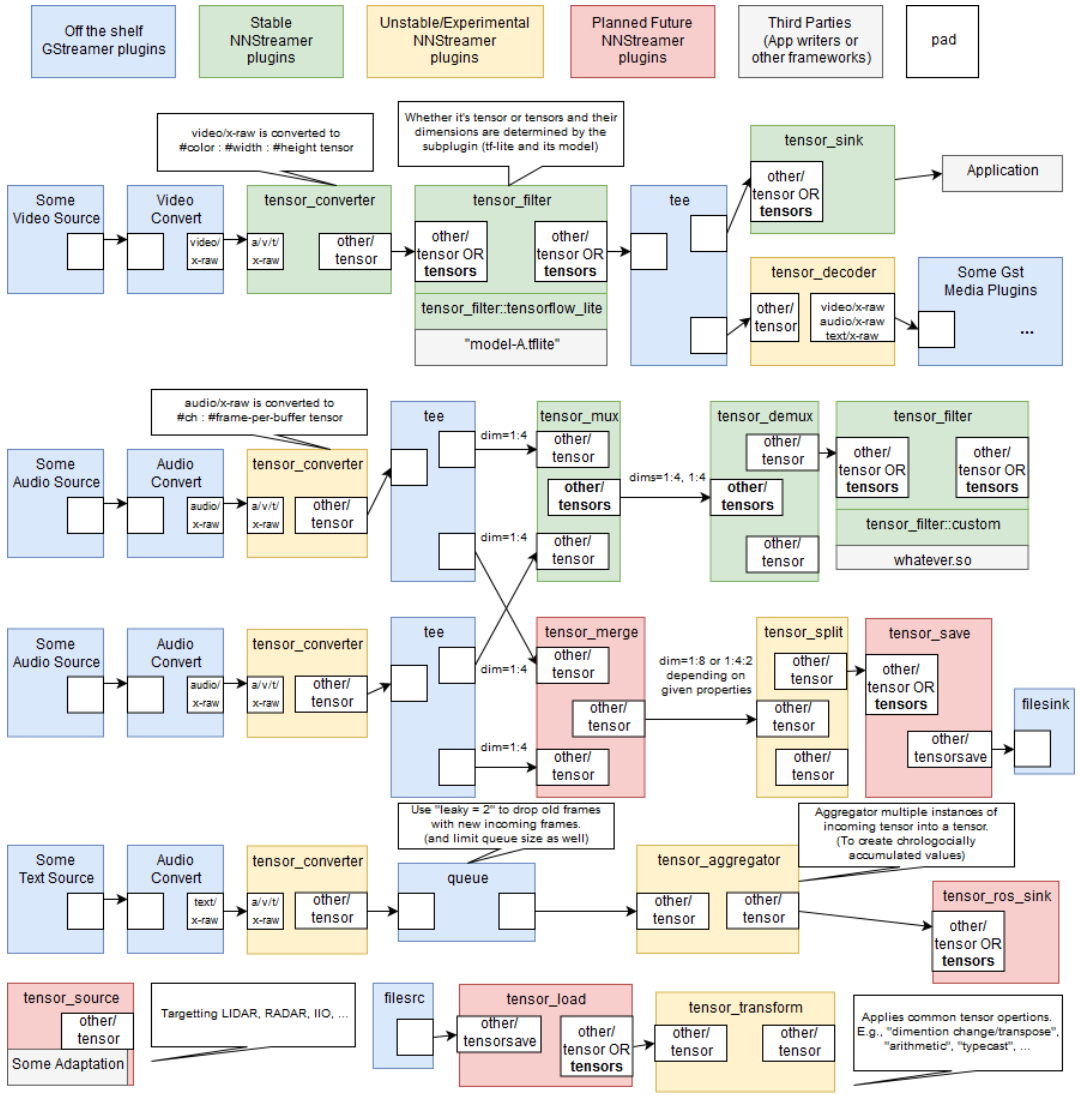
Recommended Applications

Consider **`nnstreamer`** if you are developing...

- Inferencing NN on device, not on cloud/workstations.
- Inferencing NN with data from Audio/Video real-time stream
- Multiple NNs in a device or stream pipeline
  - Esp. complex or dynamic stream topology w/ some sync-policies.
- NNs with high bandwidth stream
- ...

nnstreamer Cheat Sheet

GST Plugins being developed in nnstreamer



nnstreamer Wiki

https://github.com/nnsuite/nnstreamer/wiki

THANK YOU

GStreamer Conference 2018

Appendix

GStreamer Conference 2018

GStreamer Conference 2018

삼성 오픈소스 컨퍼런스 제목 나눔고딕 EB 32pt

소제목 나눔고딕 EB 20pt

소속 | 부서 | 작성자

날짜, 나눔고딕 EB 14pt

GStreamer Conference 2018

제목 나눔고딕 EB 22pt

소제목 나눔고딕 R 16pt 01

소제목 나눔고딕 R 16pt 02

소제목 나눔고딕 R 16pt 03

소제목 나눔고딕 R 16pt 04

소제목 나눔고딕 R 16pt 05

소제목 나눔고딕 R 16pt 06

소제목 나눔고딕 R 16pt 07

GStreamer Conference 2018

제목 나눔고딕 EB 22pt

소제목 나눔고딕 R 18pt

내용 나눔고딕 R 16pt

- 글머리기호 나눔고딕 R 16pt
  - 서브 글머리 기호, 14 pt
- 1슬라이드에 글머리기호 최대 8개 사용

제목 나눔고딕 EB 22pt

소제목 나눔고딕 R 18pt

내용 나눔고딕 R 16pt

- 글머리기호 나눔고딕 R 16pt
  - 서브 글머리 기호, 14 pt
- 1슬라이드에 글머리기호 최대 8개 사용

소스코드제목 나눔고딕 EB 22pt

소제목 나눔고딕 R 18pt

```
<?php if( have_rows('repeater_field_name') ): ?>

        <ul>

        <?php while( have_rows('repeater_field_name') ): the_row(); ?>  <li>sub_field_1 =
        <?php the_sub_field('sub_field_1'); ?>, sub_field_2 =   <?php   the_sub_field('sub_fi
eld_2'); ?>, etc</li>

        <?php

        $sub_field_3 = get_sub_field('sub_field_3');

        // do something with $sub_field_3

        ?>

        <?php endwhile; ?>

        </ul>

        <?php endif; ?>
```

Section Divider

소제목 나눔고딕 EB 22pt

GStreamer Conference 2018

THANK YOU

GStreamer Conference 2018