nnstreamer

Neural Networks as Filters of Stream Pipelines
실시간 데이터 흐름에 신경망을 쉽고 효율적으로 붙이고 구성하기

Samsung Research | AI Center | 함명주

SOSCON 2018
SAMSUNG OPEN SOURCE CONFERENCE 2018

nnstreamer는?

nnstreamer는 신경망들을 Stream Filter로 간주하여 사용할 수 있도록 해주는 **gstreamer** Plugin의 모음

gstreamer는 Multimedia Stream Pipeline을 다루기 위한 “전통” Framework

- Fully Open Sourced
  - https://github.com/nnsuite/nnstreamer 에서 코드 개발 진행
  - 누구나 제안/참여 가능
  - LGPLv2로서 누구나 사용 가능 (상업 사용 포함)

nnstreamer는?

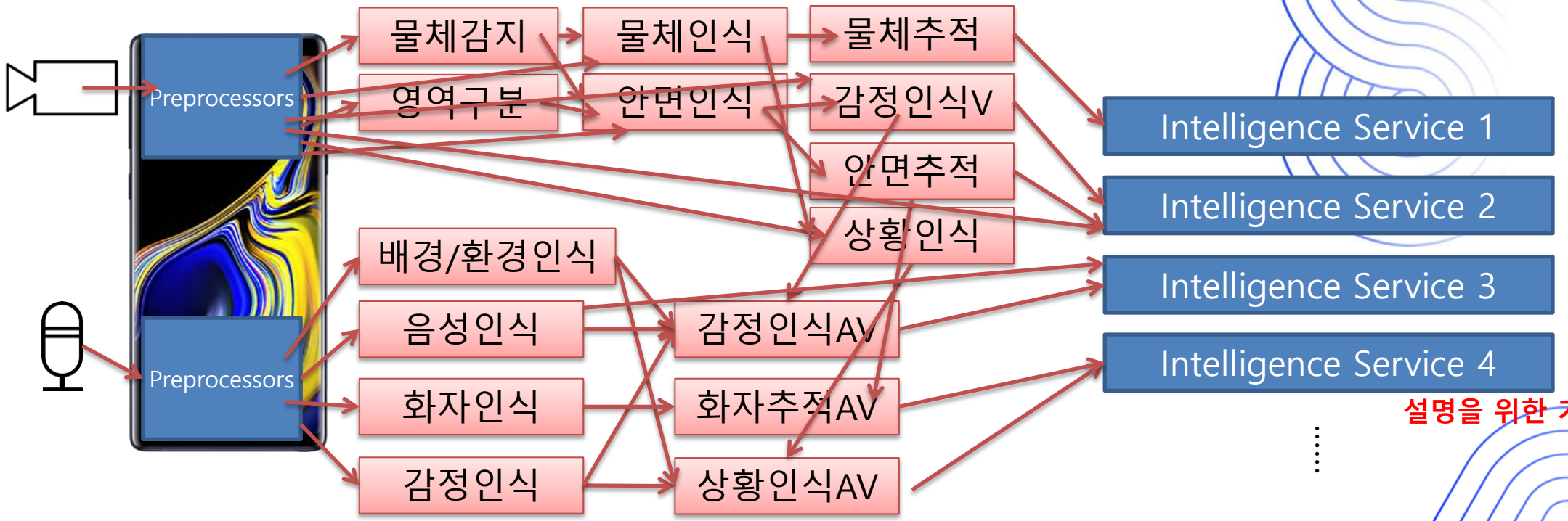
For Everyone!

- OS: Linux (**Tizen**/**Ubuntu**/…), BSD, iOS, macOS, Windows, Android, … … …
- Language: **C**, C++, C#(.NET), Java, Haskell, Qt, **Python**, Perl, Ruby, … … …
- ISA: **x86_64**, x86_32, **armv7**, **aarch64**, …

- PPA for Ubuntu-LTS: ppa:nnstreamer/ppa

Red-Bold: Official per-PR & daily build & test.

nnstreamer로 뭘 만들 수 있나요?



설명을 위한 가상 예제입니다.

- 다양한 신경망 Framework/Model 으로 구성된 Pipeline을 쉽게 구현
- 고성능 Data Stream 처리 기능 제공.
- 기존의 Multimedia Plugin과 Stream Framework의 재사용

Topics

Background	01
Motivation	02
Concepts	03
Details (**`nnstreamer`** Plugins)	04
Usage Example / Demo	05
How-To-Contribute	06
Conclusion	07

Background

Cast Study

Case 1: 자율주행

LIDAR

Camera

RADAR

* 실제 시스템보다 단순화 되었음.

Case 1: 자율주행 센서 / 신경망 모델

실제보다 단순화 된 가상 모델

LIDAR

Camera

RADAR

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

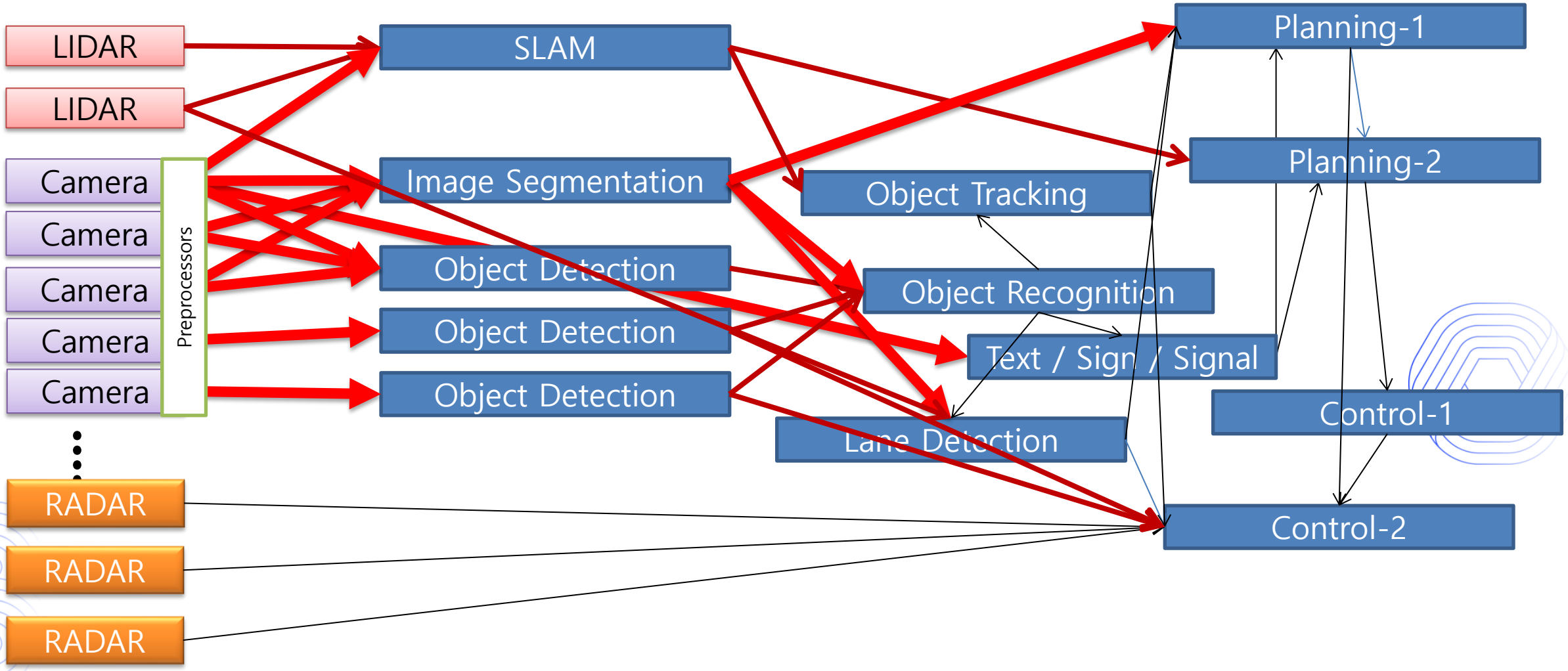
SLAM: Simultaneous localization and mapping

* 실제 시스템보다 단순화 되었음.

Case 1: 자율주행 Data Stream Pipelines

SOSCON 2018

실제보다 단순화 된 가상 모델

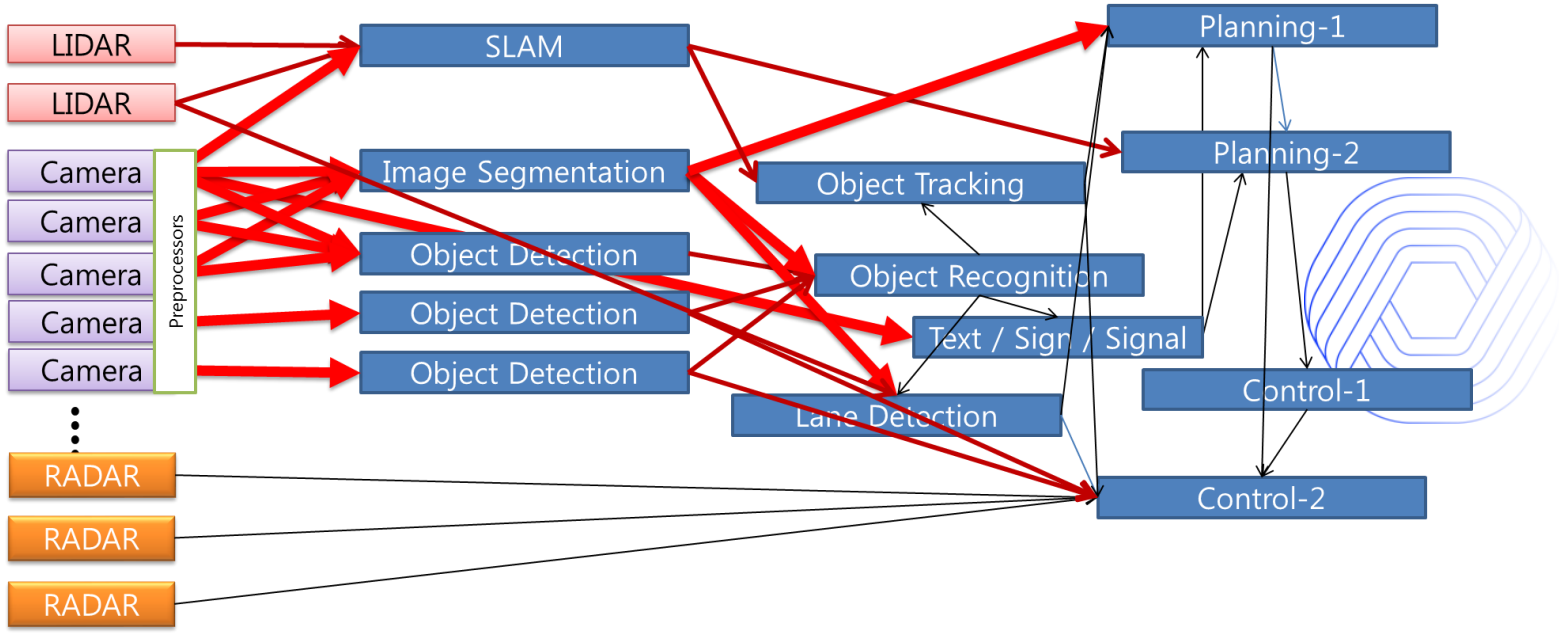


* 실제 시스템보다 단순화 되었음.

Case 1: 자율주행 Data Stream Pipelines

주요 발생 이슈

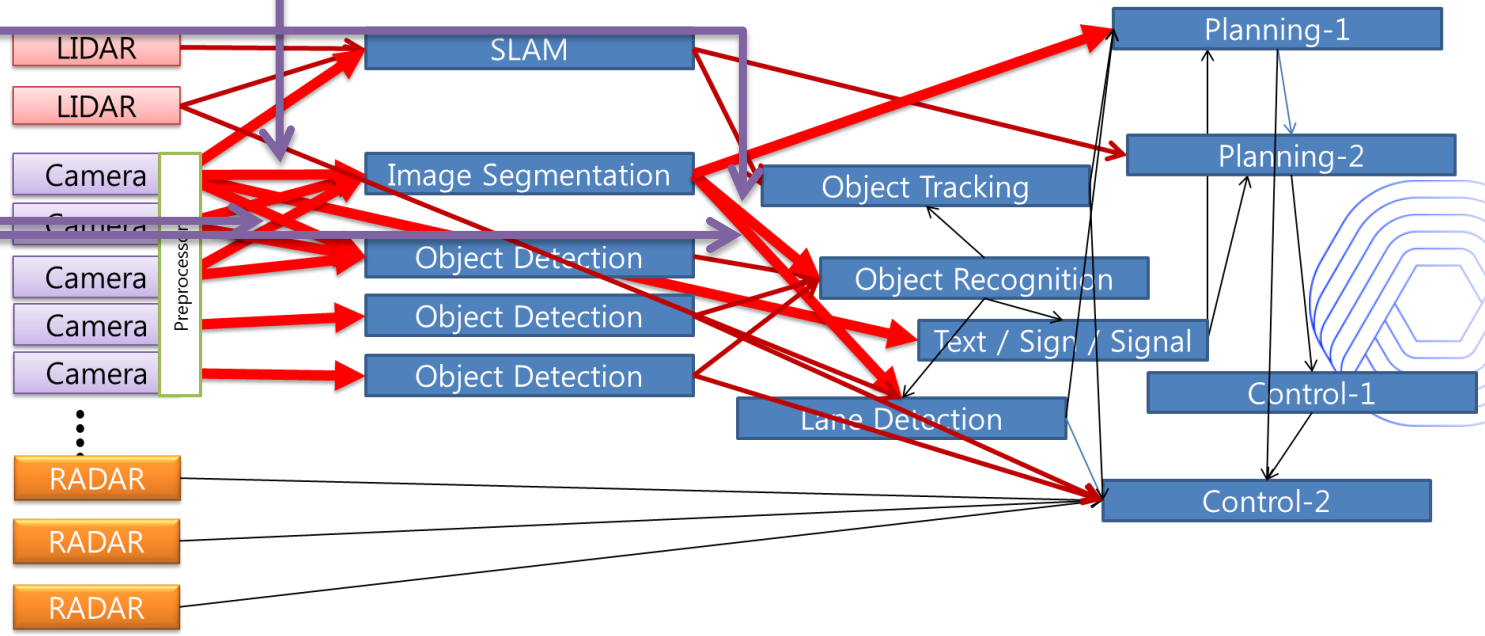
- High Bandwidth
- Complex Pipeline Topology
- Synchronization & Parallelism
- Latency & Jitter Control



Case 1: 자율주행 Data Stream Pipelines w/ ROS

ROS 사용 시 주요 발생 이슈 (Socket!)

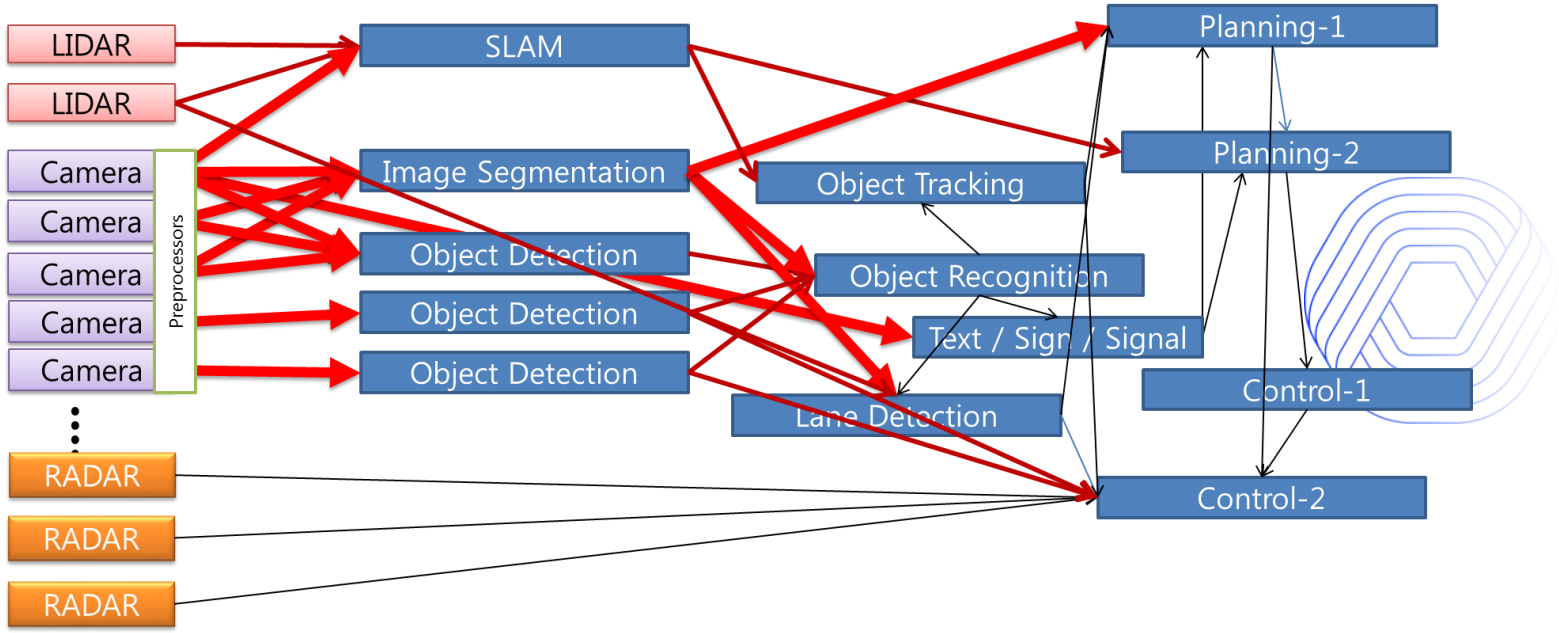
- High Bandwidth
  - 고해상도 이미지 실시간 전송
    - ROS2 / DDS (RTI Connext DDS) may help?
  - 모델간 데이터 전송
  - 중복 데이터 전송
- 고성능 Workstation에서도 **감당하기 힘들 수 있음!!** (Memory Transaction!)



Case 1: 자율주행 Data Stream Pipelines

ROS 사용 시 주요 발생 이슈 (Socket!)

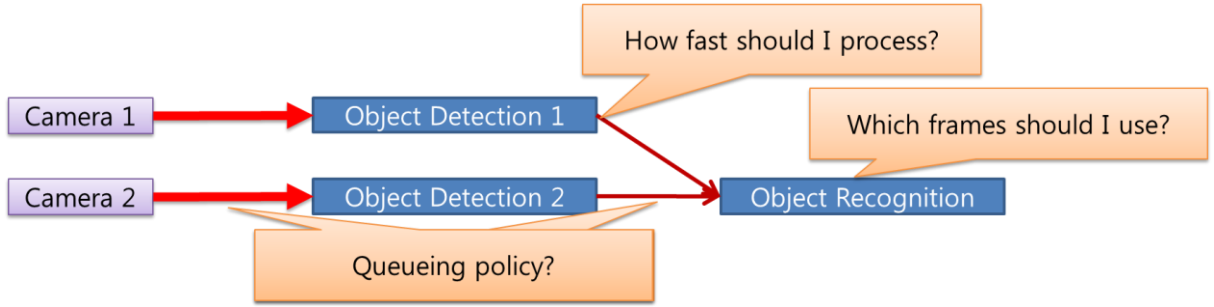
- Complex Pipeline Topology
  - Pub/Sub 구조로 복잡한 Topology도 감당 가능.
  - Pub/Sub 구조는 Overkill???



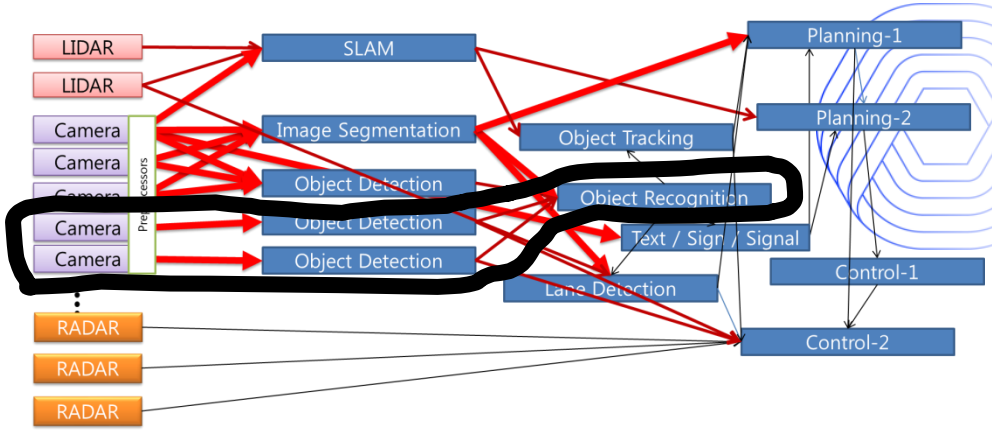
Case 1: 자율주행 Data Stream Pipelines

ROS 사용 시 주요 발생 이슈 (Socket!)

- Synchronization
  - 독립 Process로 동작하여 **골치 아플 수 있음!**



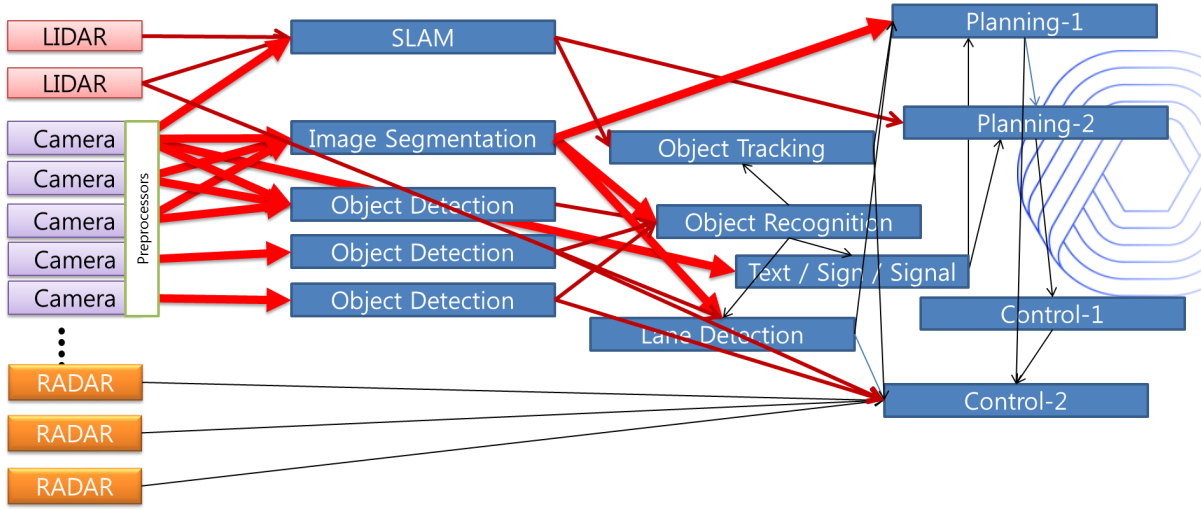
- Parallelism
  - 각 Node는 독립 Process. 문제없음



Case 1: 자율주행 Data Stream Pipelines w/ ROS

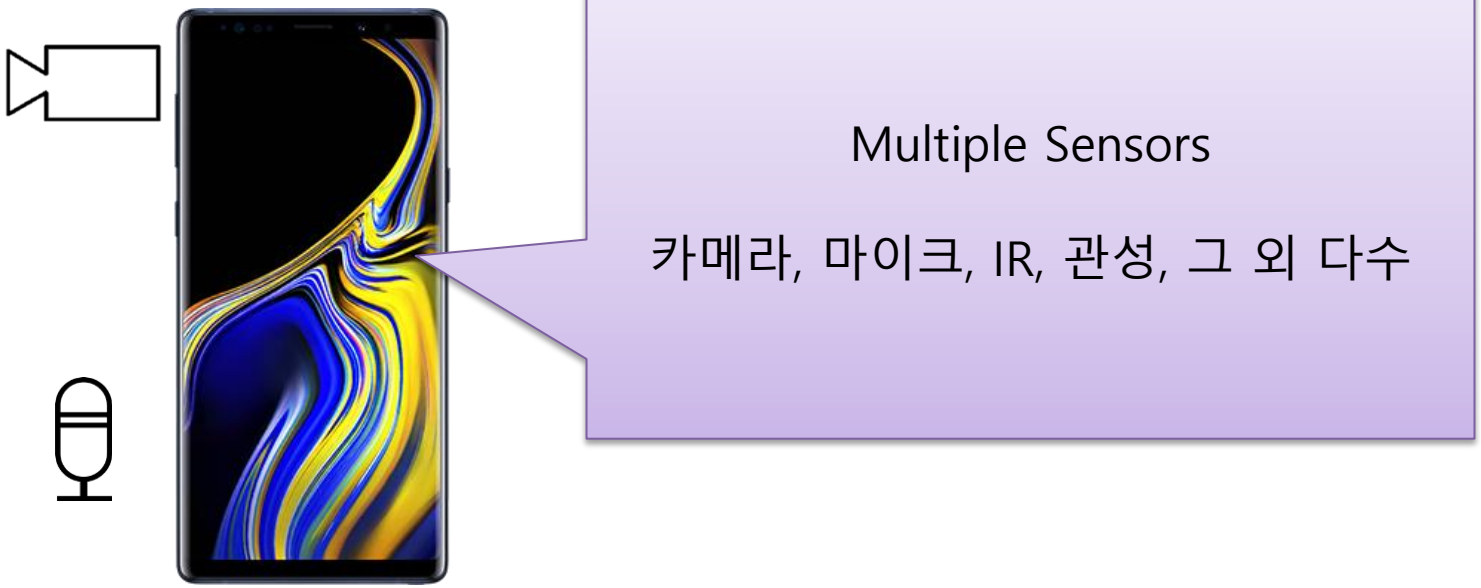
ROS 사용 시 주요 발생 이슈 (Socket!)

- Latency
  - 과도한 Memcpy로 인해 Communication Latency 증가
- Jitter Control
  - 각 Node들에 대한 Resource/Rate 제어 어려움
  - Node간 생성물에 대하 동기화 어려움



Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices



Multiple Sensors

카메라, 마이크, IR, 관성, 그 외 다수

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Multiple AI-based Services

감정인식, 화자인식, STT, 앱연결, 그 외
상상하시는 아무거나...

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

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

마이크 입력을 사용할 기본 요소?

- 배경/환경인식
- 음성인식
- 화자인식
- 감정인식

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

설명을 위한 가상 예제입니다.

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

비디오 입력을 사용할 기본 요소?

- 물체감지
- 물체인식
- 물체추적
- 영역구분
- 안면인식
- 감정인식V
- 안면추적
- 상황인식
- 배경/환경인식
- 음성인식
- 화자인식
- 감정인식

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

설명을 위한 가상 예제입니다.

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

오디오 비디오 입력을 함께 사용할 기본 요소?

물체감지
물체인식
물체추적
영역구분
안면인식
감정인식V
안면추적
상황인식
배경/환경인식
음성인식
화자인식
감정인식
감정인식AV
화자추적AV
상황인식AV

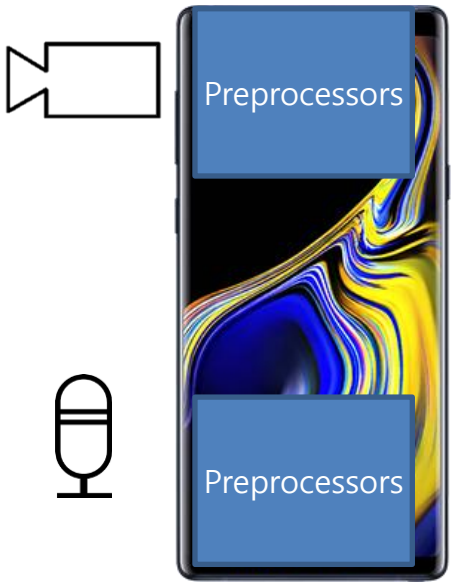
Intelligence Service 1
Intelligence Service 2
Intelligence Service 3
Intelligence Service 4
⋮

설명을 위한 가상 예제입니다.

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

오디오 비디오 미디어 전처리



물체감지 | 물체인식 | 물체추적

영역구분 | 안면인식 | 감정인식V

안면추적

상황인식

배경/환경인식

음성인식 | 감정인식AV

화자인식 | 화자추적AV

감정인식 | 상황인식AV

Intelligence Service 1

Intelligence Service 2

Intelligence Service 3

Intelligence Service 4

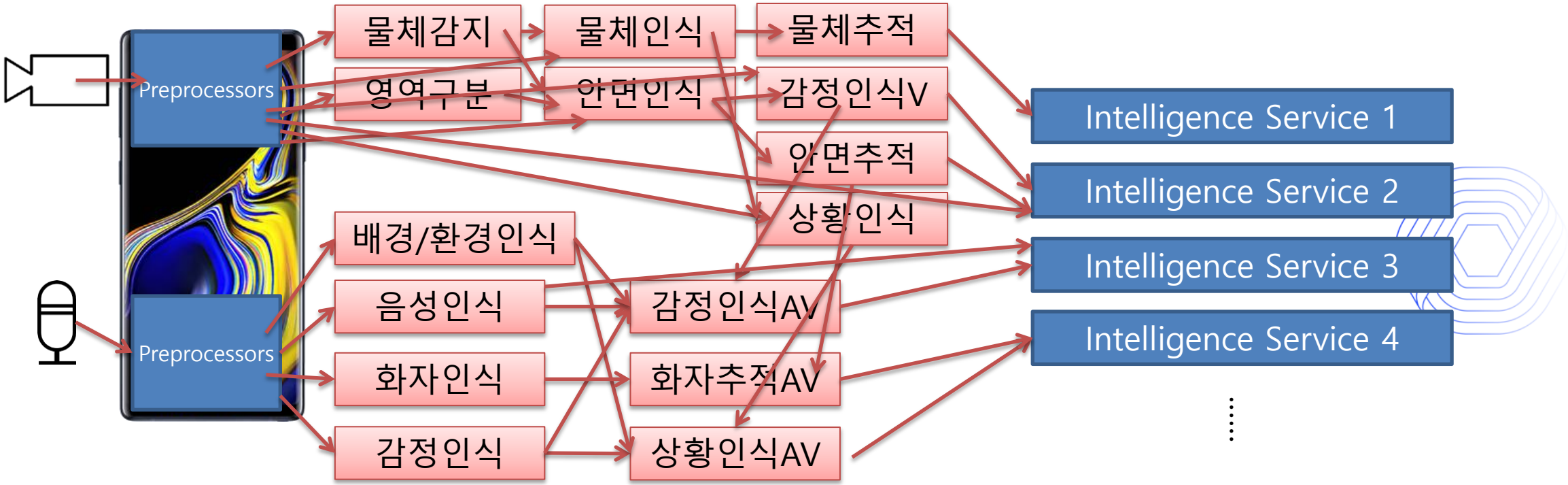
⋮

설명을 위한 가상 예제입니다.

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Data Stream Pipelines



설명을 위한 가상 예제입니다.

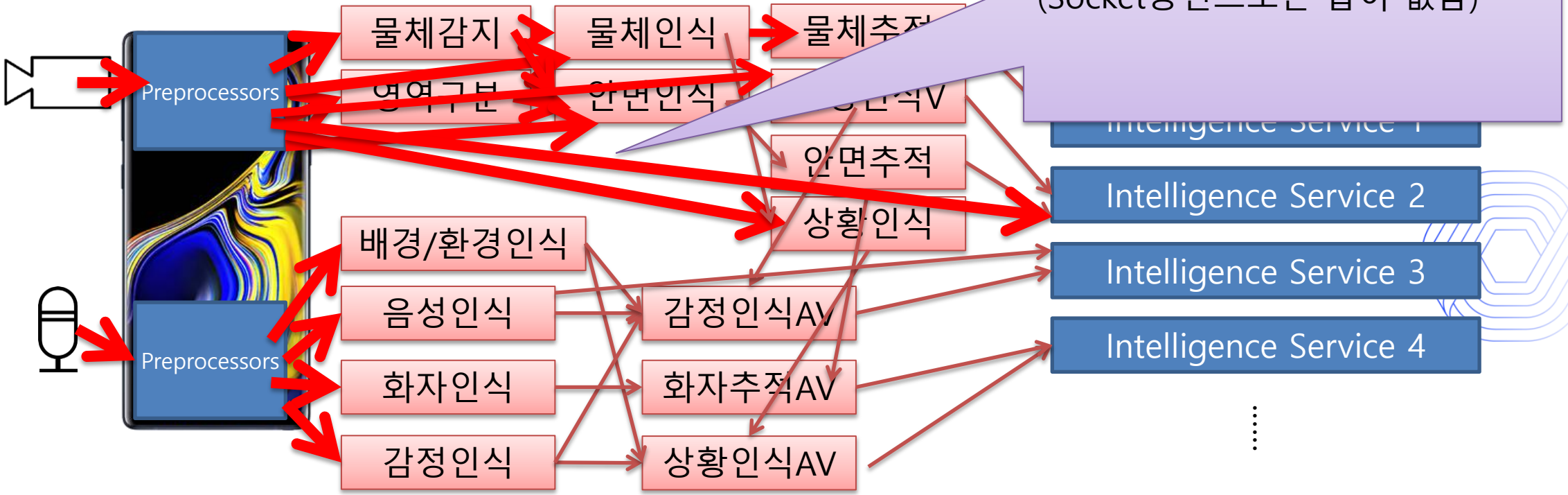
Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 1. Data Stream Performance.

데이터 전송 효율성 / 속도문제!

(Socket통신으로는 답이 없음)

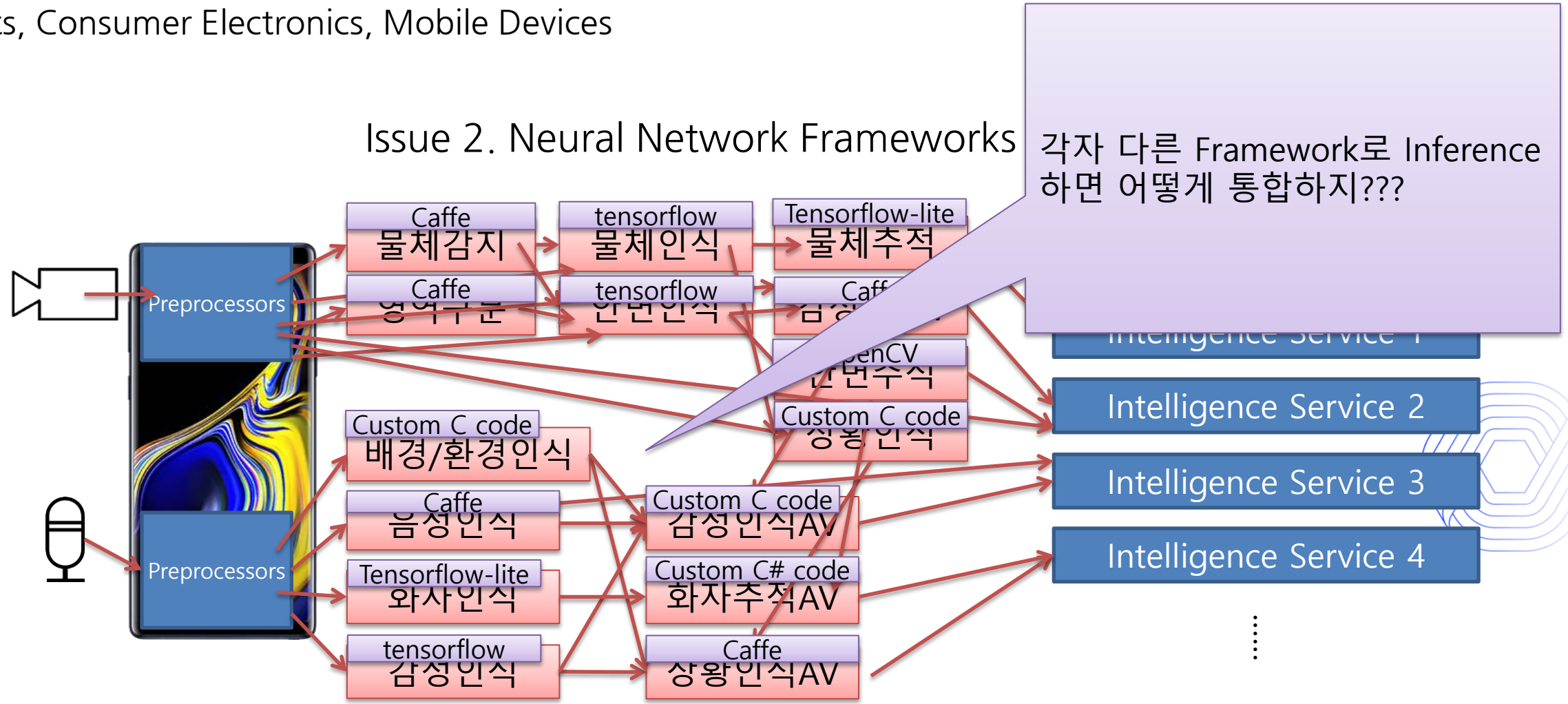


설명을 위한 가상 예제입니다.

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 2. Neural Network Frameworks



설명을 위한 가상 예제입니다.

Case 2: Multi-Modal Interactions

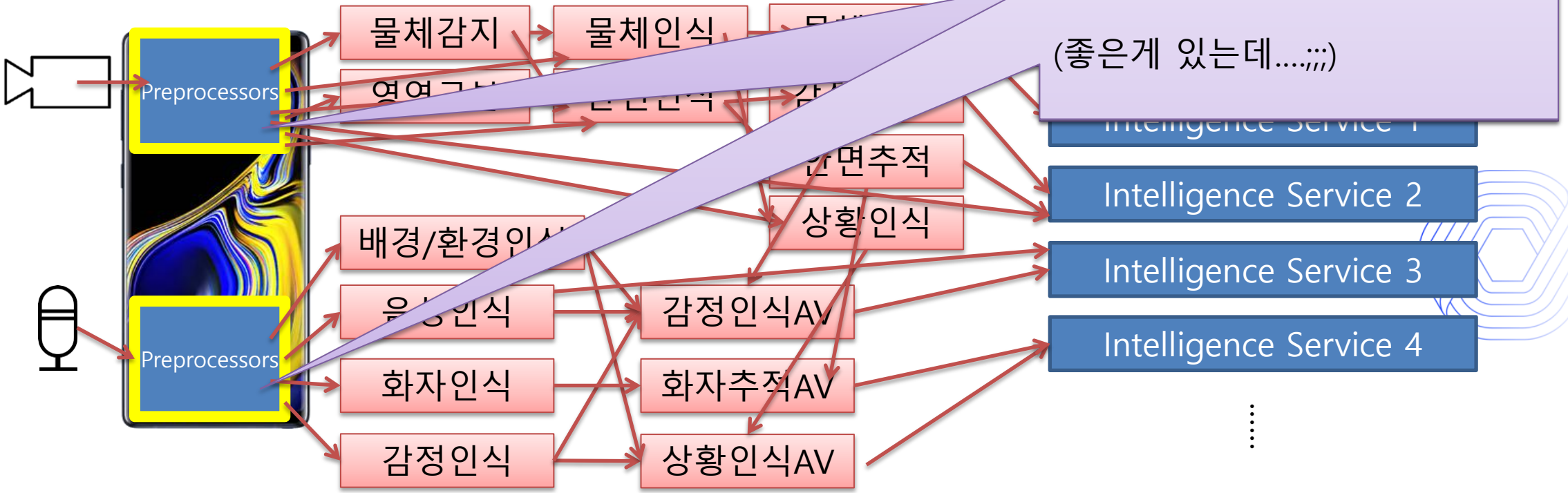
Robots, Consumer Electronics, Mobile Devices

Issue 3. Multimedia Data Processing

Media Preprocessor Filter들은 어떻게 만들고 통합하지?

Data Stream 제어는??

(좋은게 있는데....;;;)

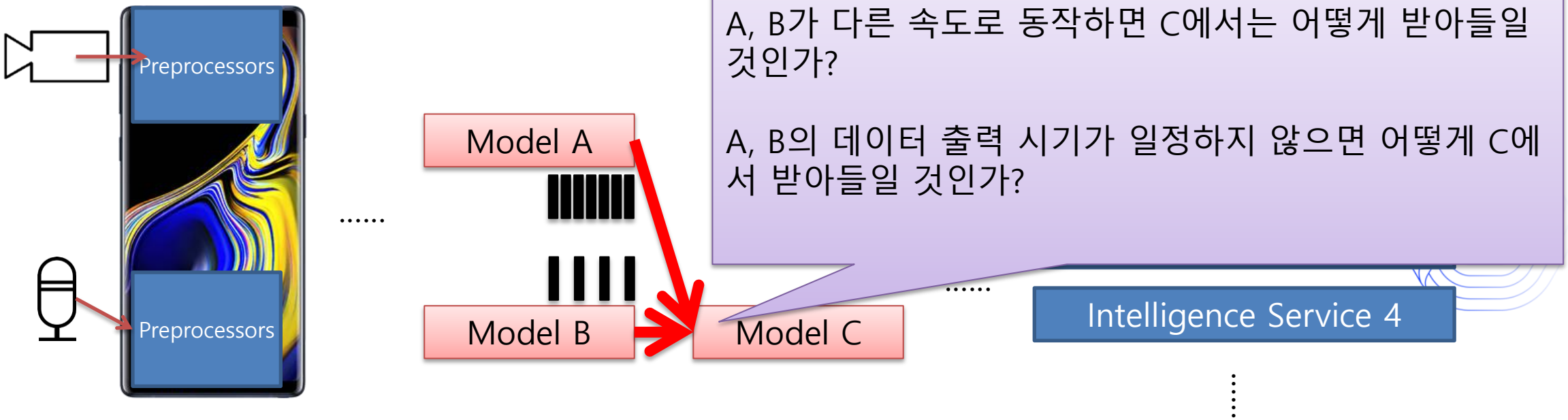


설명을 위한 가상 예제입니다.

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 4. Data Stream Synchronization

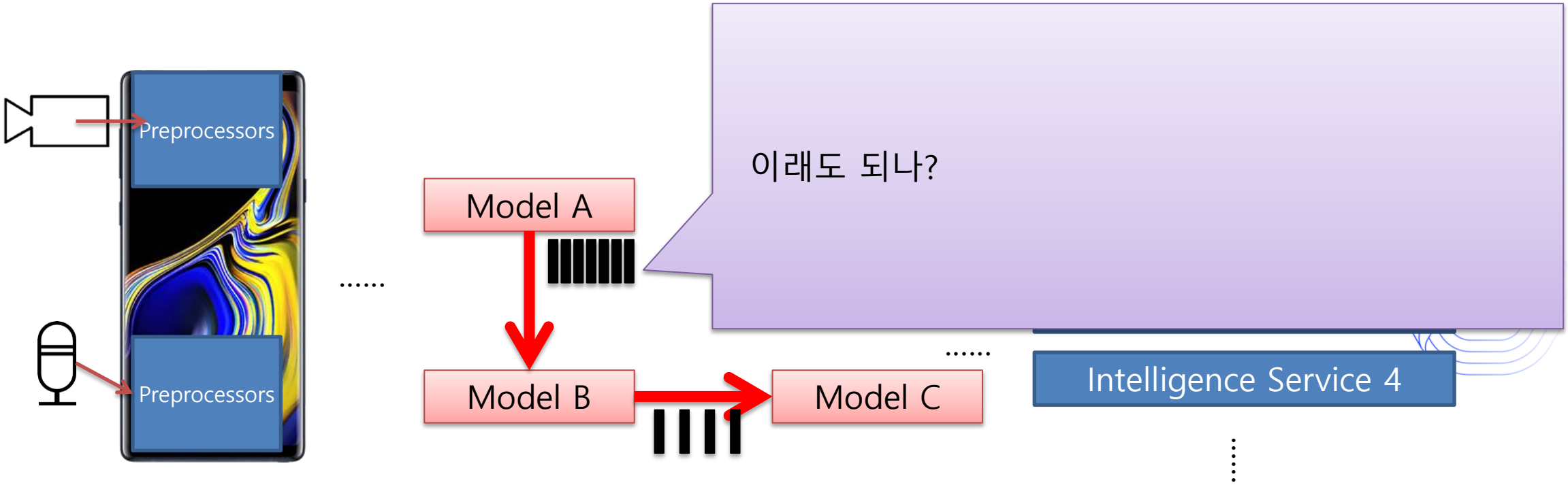


설명을 위한 가상 예제입니다.

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 5. Data Stream Rate Control

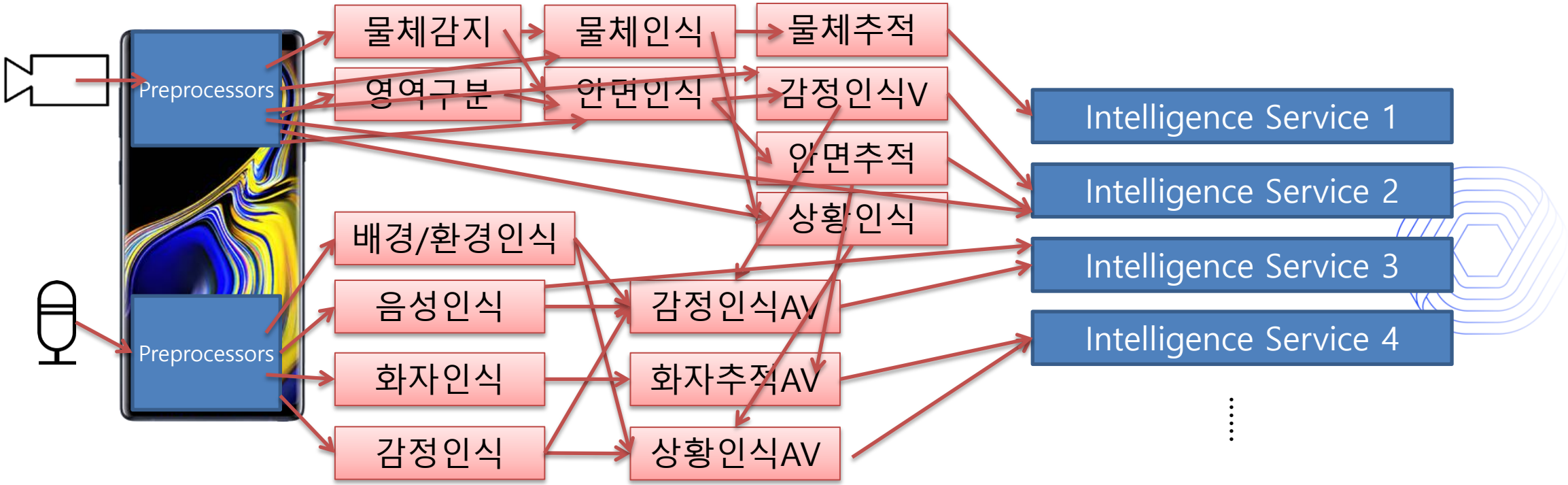


설명을 위한 가상 예제입니다.

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 6. Dynamic Stream Pipelines



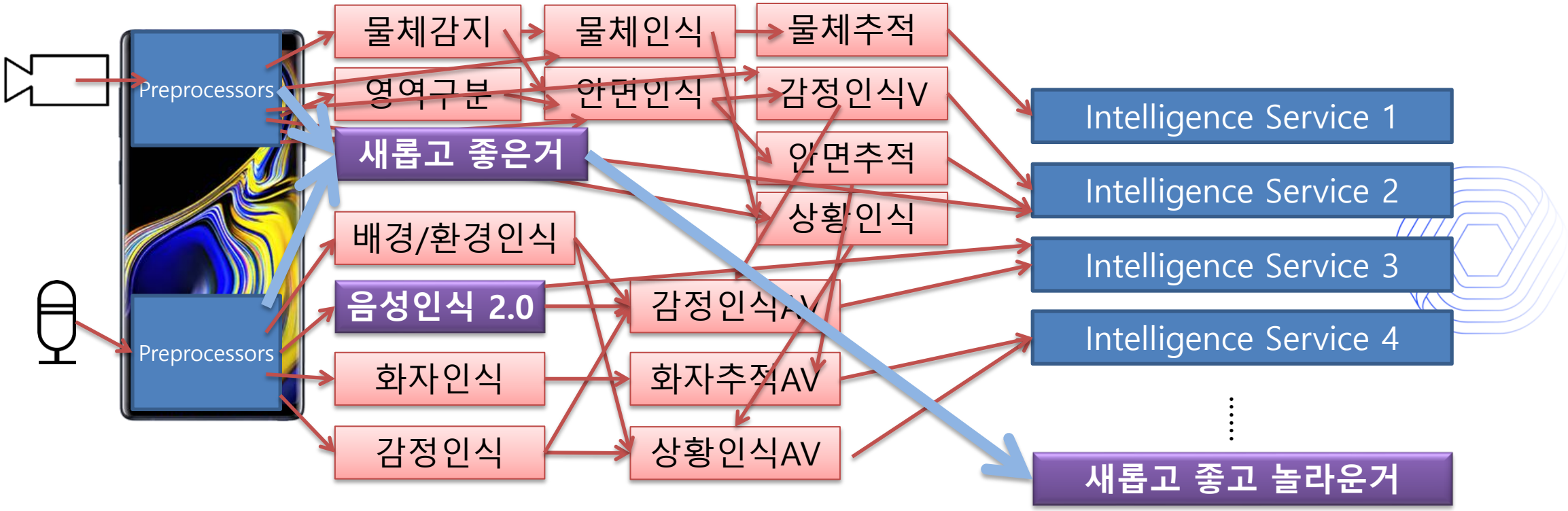
설명을 위한 가상 예제입니다.

Case 2: Multi-Modal Interactions

SOSCON 2018

Robots, Consumer Electronics, Mobile Devices

Issue 6. Dynamic Stream Pipelines



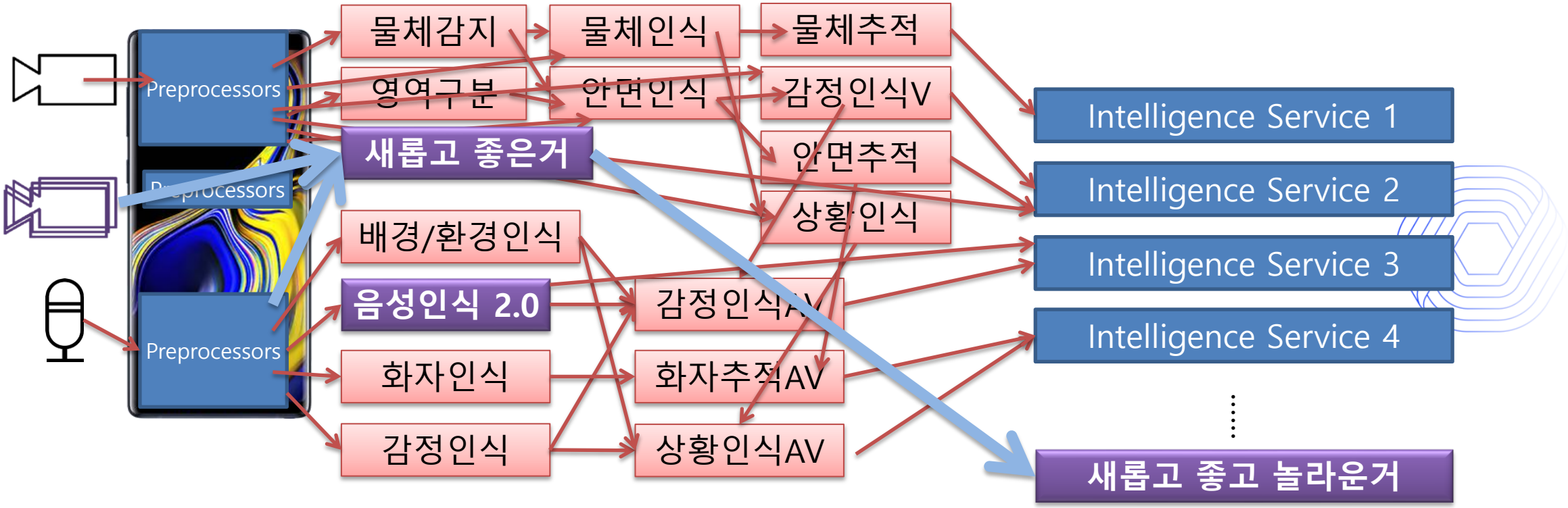
설명을 위한 가상 예제입니다.

Case 2: Multi-Modal Interactions

SOSCON 2018

Robots, Consumer Electronics, Mobile Devices

Issue 6. Dynamic Stream Pipelines



설명을 위한 가상 예제입니다.

Motivation

Why nnstreamer?

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

DO NOT REINVENT THE WHEEL

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

Do NOT Reinvent the Wheel

Do NOT Reinvent the Wheel

gstreamer for intelligence!

What is gstreamer?

- Pipeline 기반 Open Source (LGPL) Multimedia Framework (2001.1. 첫 공개)
- Linux 계열 표준
  - Linux, BSD, macOS, iOS, Windows, … 지원.
  - TV: 삼성, LG, Sony, …
- Audio/Video 재생, 기록, 스트리밍/방송, 변환, 편집 등에 광범위하게 사용됨
  - Stream Filter로 사용 가능한 수백가지 Plugin 제공.
  - Plugin을 제작하기 위한 환경 포함
  - C/C++/C#/Java/Python/Perl/CLI/… 에의 표준 API 제공
  - Pipeline 구성을 표현하는 것 만으로 동작하는 Application이 완성됨!
    - GUI Tool로 Pipeline 그려서 만들기도…

Do NOT Reinvent the Wheel

gstreamer for intelligence!

Why gstreamer?

- Data Stream Pipeline 을 표현, 관리, 실행하는 안정된 표준 Framework
- 언급된 Issue는 이미 수십년 된 Multimedia의 주요 Issue!
  - 이미 해결한 문제 다시 풀지 말자!
- 신경망 == Data Stream Filter!

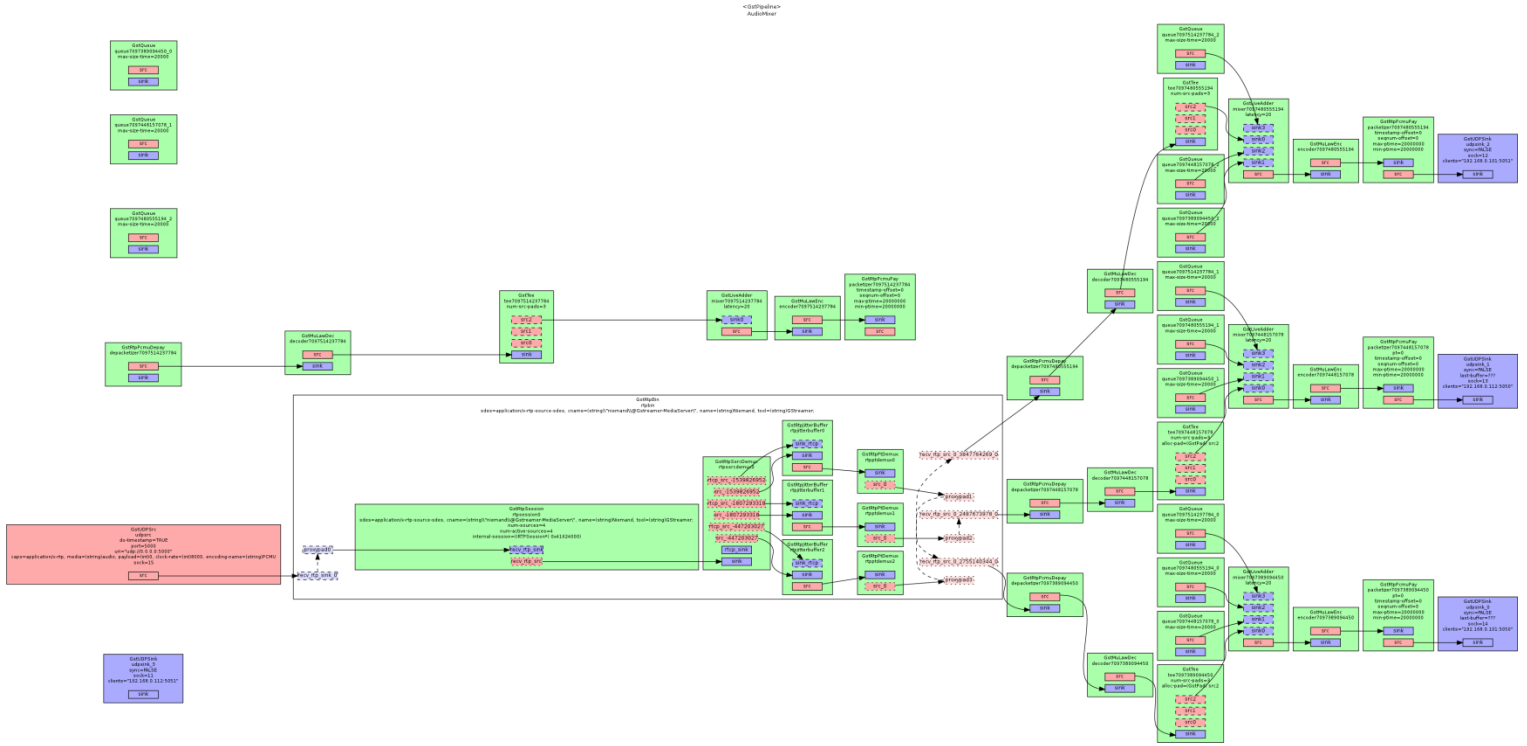
Do NOT Reinvent the Wheel

gstreamer SOSCON 2018

gstreamer for intelligence!

Why gstreamer?

- Configure & run complex stream pipelines, dynamically in run-time!



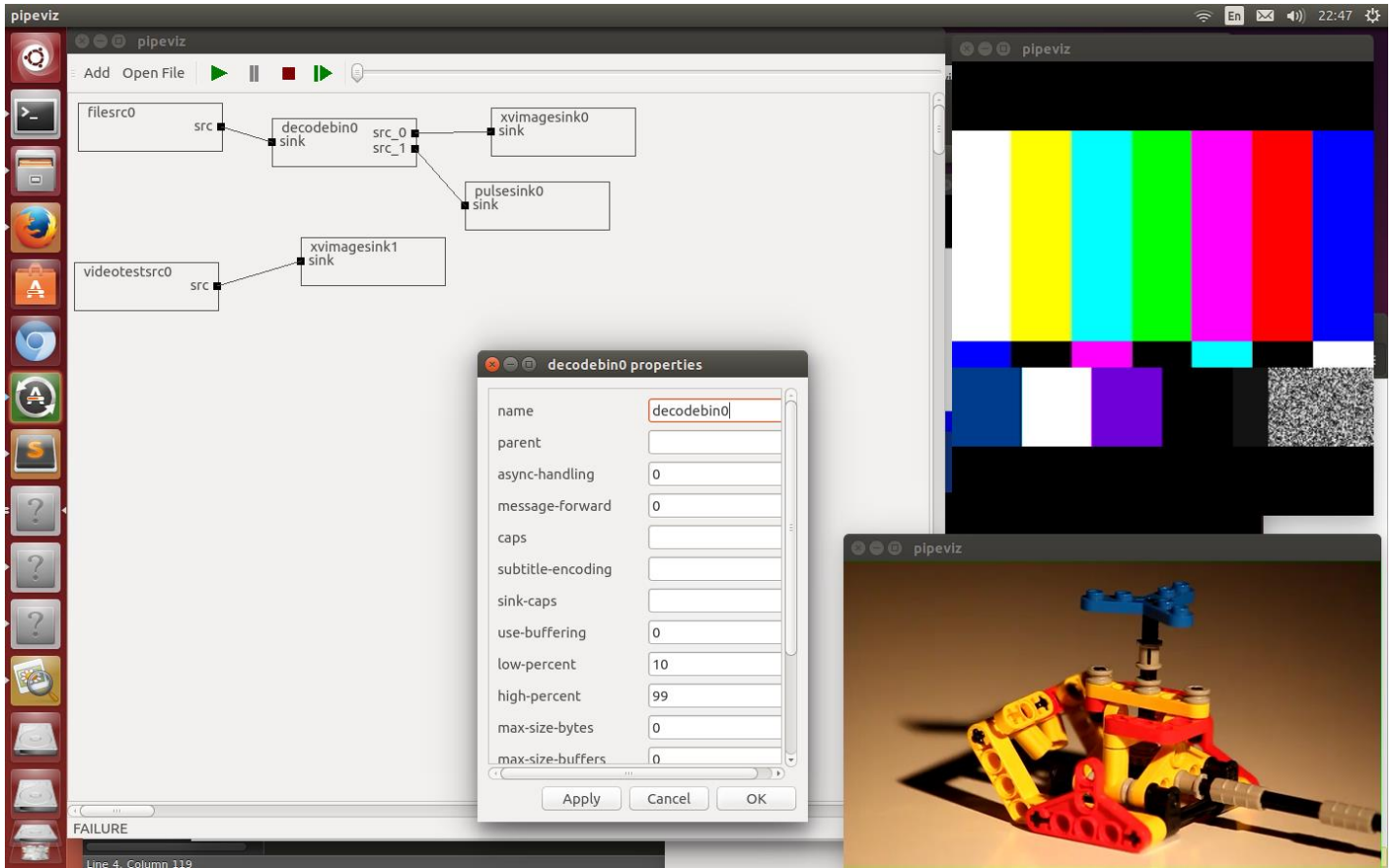
Do NOT Reinvent the Wheel

gstreamer SOSCON 2018

gstreamer for intelligence!

Why gstreamer?

- Configure & run complex stream pipelines… even with GUI tools



https://github.com/virinext/pipeviz

ROS vs GStreamer

비교

	ROS	GStreamer
Community	HIGH (11 years old) Many robot-related plugins	HIGH (17 years old) Many multimedia plugins
Maturity (commercial products)	MEDIUM (Questionable for mass pr oduction)	**HIGH** (Mass Production: TV, Wearable, …)
OS	**Linux**, Windows 10	**Linux**, BSD, Solaris, Android, macOS, iOS, Windows, OS/400
Communication	Socket (ROS2: DDS supported)	**Shared memory chunks (faster!)**
Distributed System	**Fully supported**	Experimental (may use ROS as gstreamer plugins)
Sensors	**More varying.** Many unstable (even sound)	**Video/Audio: good** / Other sensors: unstable
Neural Network	Custom plugins	Custom plugins → **`nnstreamer`**

Don’t Reinvent the Wheel,
PERFECT THE WHEEL!

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

Don’t Reinvent the Wheel, Perfect it!

gstreamer for intelligence ➔ **nnstreamer**

Gstreamer가 신경망을 이해하여 AI Device의 기반이 되도록 하기 위해…

- 다양한 신경망 Framework / Model 들에게 공통으로 주어질 Data Stream 표준
- 다양한 신경망 Framework / Model 들을 Invoke할 방법
- Audio/Video/Text 외 다른 Data Stream을 GStreamer에서 다루도록 할 방법

Don’t Reinvent the Wheel, Perfect it!

gstreamer for intelligence ➔ **nnstreamer**

Gstreamer가 신경망을 이해하여 AI Device의 기반이 되도록 하기 위해…

- 다양한 신경망 Framework / Model 들에게 공통으로 주어질 Data Stream 표준
- 다양한 신경망 Framework / Model 들을 Invoke할 방법
- Audio/Video/Text 외 다른 Data Stream을 GStreamer에서 다루도록 할 방법

➔ nnstreamer

gstreamer for intelligence ➔ **nnstreamer**

➔ nnstreamer 

License: LGPL 2.1

기존 GStreamer LGPL 코드 재사용

Public Open: 2018.7. (v0.0.1)

https://github.com/nnsuite/nnstreamer

Don’t Reinvent the Wheel, Perfect it!

gstreamer for intelligence ➔ **nnstreamer**

Gstreamer가 신경망을 이해하여 AI Device의 기반이 되면…

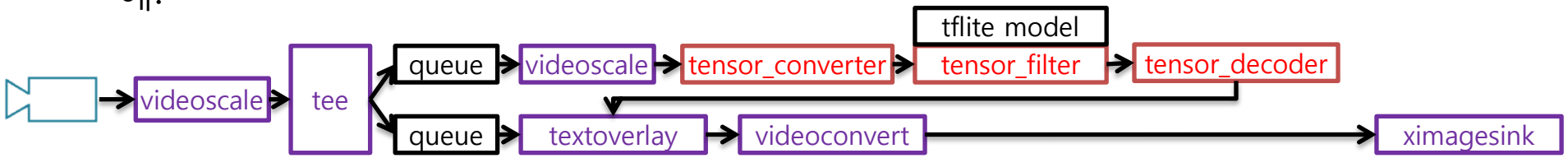
- 신경망 알고리즘 개발자
- 신경망 사용 디바이스/애플리케이션 개발자
- 멀티미디어 개발자

Don’t Reinvent the Wheel, Perfect it!

gstreamer for intelligence ➔ **nnstreamer**

Gstreamer가 신경망을 이해하여 AI Device의 기반이 되면…

- 신경망 알고리즘 개발자
  - 각종 센서 / 미디어 / 데이터 스트림에 쉽고 즉각적인 통합 ➔ 빠른 테스트 / 프로토타이핑
  - 예:




```
$ gst-launch-1.0    v4l2src name=cam_src ! videoscale ! \   # “cam_src”에서 영상 받아서 scale
      video/x-raw,width=640,height=640,format=RGB ! \       # 3x640x480의 RGB Stream
      tee name_t_raw \                                      # Data Stream 분기
      t_raw. ! queue ! videoscale ! video/x-raw,width=224,height=224 ! \   # 224x224 RGB 로 전환
            tensor_converter ! \                                           # “tensor” 포맷으로 전환
            tensor_filter framework=tensorflow-lite model=mobilenetssd.tflite ! \ # tflite 모델 부름
            tensor_decoder mode=labeling option1=labels.txt ! txt. \       # 결과값 해석 방법 지정 ➔ txt
      t_raw. ! queue ! textoverlay name=txt ! \                            # txt로 들어온 문자열을 합성
            videoconvert ! ximagesink                                      # 합성결과를 화면에 출력
```

Don’t Reinvent the Wheel, Perfect it!

gstreamer for intelligence ➔ **nnstreamer**

Gstreamer가 신경망을 이해하여 AI Device의 기반이 되면…

- 신경망 사용 디바이스/애플리케이션 개발자
  - 복잡한 Data Stream Pipeline 의
    - 표현이 쉽다
    - 변경, 관리도 쉽다
    - 실행이 효율적이고 빠르다
    - 구성하는 알고리즘/모델의 교체도 쉽다.

    } 저비용! 고효율!
  - 다양한 언어/OS에 표준 API 지원
    - C, C++, C#(.NET), Java, Python, Perl, Rust, D, Qt, Haskell, Ruby, CLI (shell)
    - Linux, BSD, Solaris, Android, macOS, iOS, Windows, OS/400
  - 다양한 Input 과 Media Filter의 재사용
  - 기존 SW Platform의 재사용 / 통합.

Don’t Reinvent the Wheel, Perfect it!

gstreamer for intelligence ➔ **nnstreamer**

Gstreamer가 신경망을 이해하여 AI Device의 기반이 되면…

- 멀티미디어 개발자
  - 신경망 알고리즘을 Media Filter로 사용하자!

화질 개선, Context 인식, Style 변형, ……

Concepts

How nnstreamer  works?

How GStreamer handle neural networks?

신경망도 또 하나의 Media Filter 일뿐.

그럼 신경망 사이의 데이터는 Video Stream? Audio Stream? Text Stream? …

- ➔ “Tensor Stream”!

- Tensor : Multi-Dimensional Array

How GStreamer handle neural networks?

신경망도 또 하나의 Media Filter. Tensor도 또 하나의 Media Type.

Data Stream으로서의 “Tensor”에 대한 정의

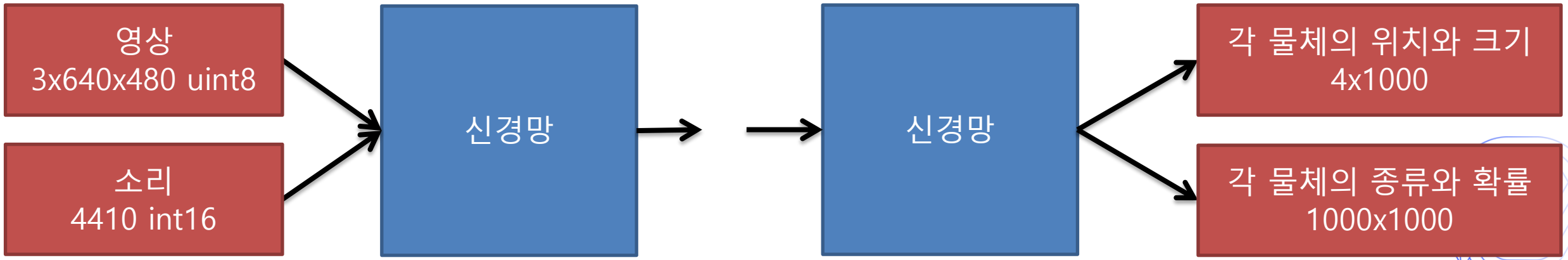
```
other/tensor
        framerate: (fraction) [ 0/1, 2147483647/1 ]
        dim1: (int) [1, 65535]
        dim2: (int) [1, 65535]
        dim3: (int) [1, 65535]
        dim4: (int) [1, 65535]    # 아직 5차원 이상은 지원하지 않습니다.
        type: (string) { uint8, int8, uint16, int16, uint32, int32, uint64, int64, float32
, float64 }
```

- Refer: https://github.com/nnsuite/nnstreamer/wiki/AD%3A-Data-Type%2C-Caps%2C-and-Flow-Design “**GStreamer data types (pad capabilities)**”

Tensor로 다 될까요?

NO!

Tensor 만으로는 충분하지 않은 경우들…

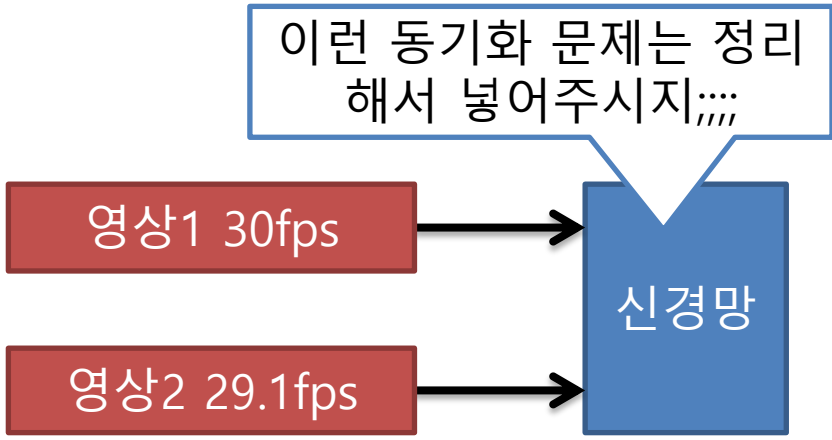


Tensor가 여러 개 뭉쳐야 하네요…

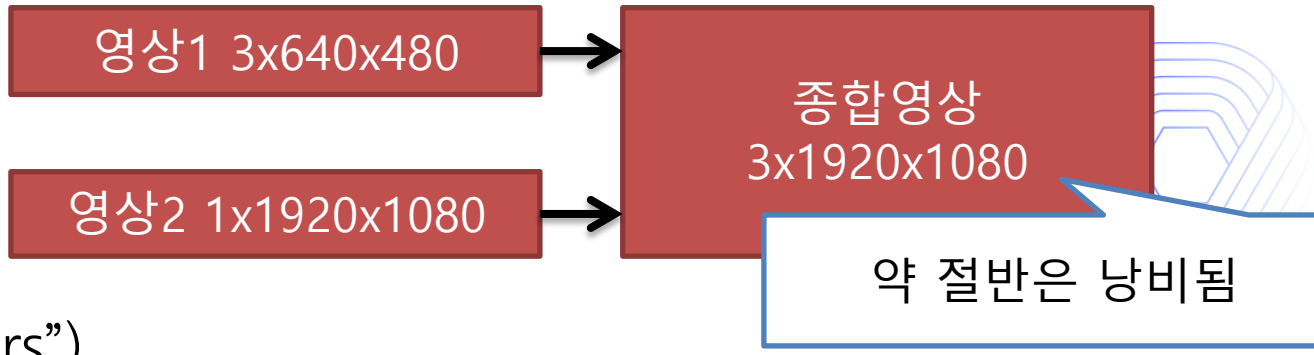
Tensor의 묶음을 표현하자

어떻게?

- Tensor는 그대로 두고 Data Stream 여러 개 사용?
  - Synchronization 문제 추가 발생. (GST 기능 적용 곤란)



- Tensor Dimension만 키워서 사용?
  - 메모리 낭비! 성능 저하!

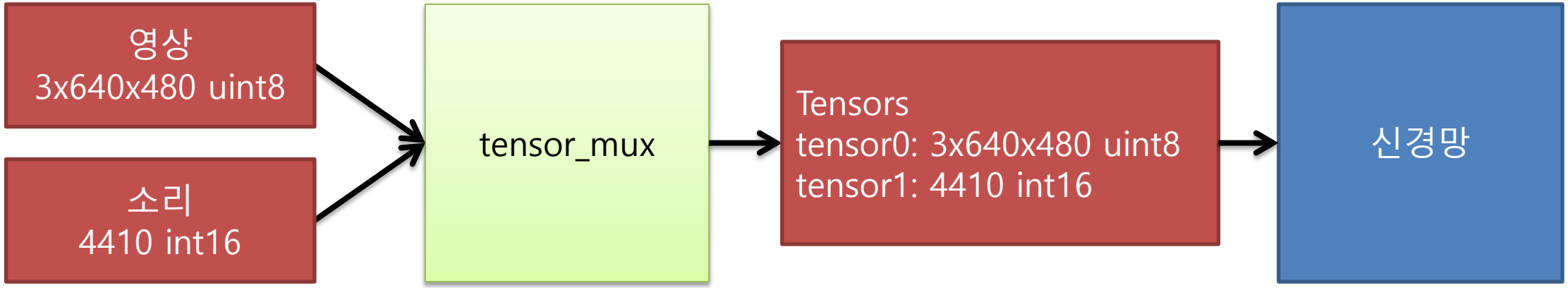


- Tensor의 묶음을 나타낼 표준 제정? (“Tensors”)
  - **nnstreamer** v0.0.2 !

Tensors 로 표현되면…

other/tensors

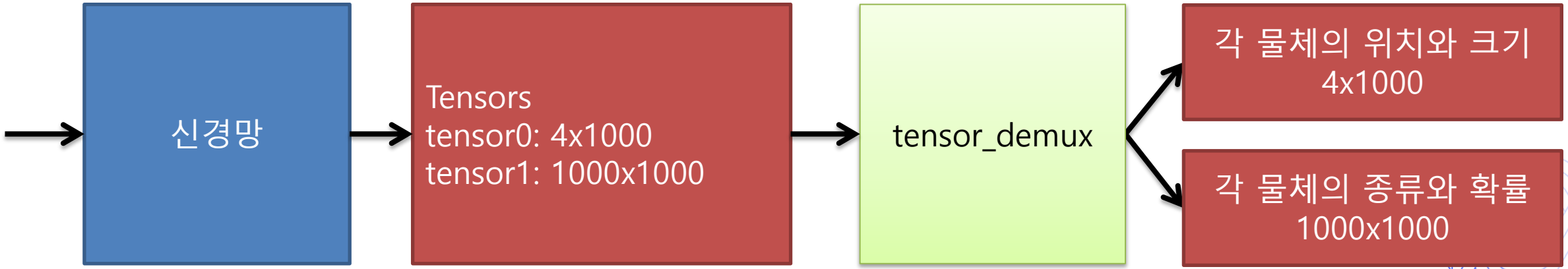
Tensor 만으로는 충분하지 않은 경우였던…



Tensors 로 표현되면…

other/tensors

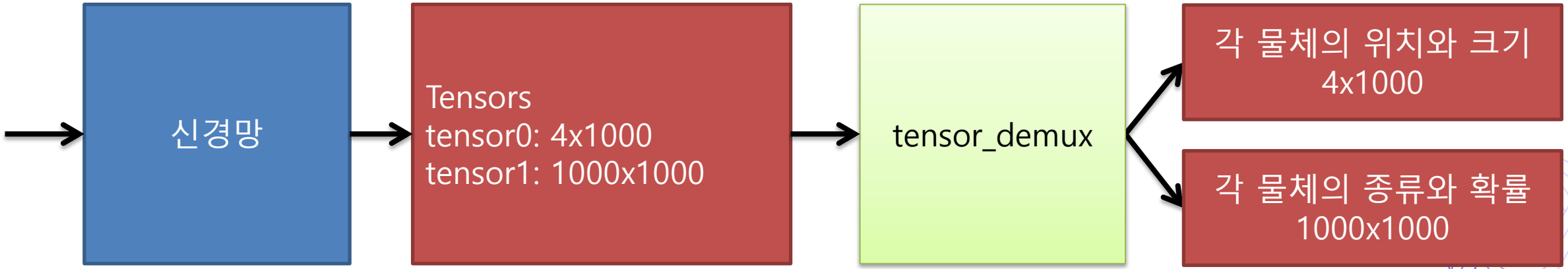
Tensor 만으로는 충분하지 않은 경우였던…



Tensors 로 표현되면…

other/tensors

Mux / Demux 성능



Tensors 성능 문제?

MUX/DEMUX 때 성능 낭비 없나요?

Memcpy-less mux & demux!

- GstBuffer Multi-memory-chunk 기능 사용.
- 대신. 최대 16개의 Tensor만 뭉쳐집니다.
- 16개 넘기려면 Gstreamer 재빌드 필요. (Hardcoded Parameter)

Tensors. 그 생김새는?

신경망도 또 하나의 Media Filter. Tensor도 Tensors도 또 하나의 Media Type.

Data Stream으로서의 “Tensor**s**”에 대한 정의

```
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: https://github.com/nnsuite/nnstreamer/wiki/AD%3A-Data-Type%2C-Caps%2C-and-Flow-Design “**GStreamer data types (pad capabilities)**”

Tensor File Format

other/tensorsave

파일로 tensor / tensors 를 저장하고 로드하기 위한 표준

- 현재 개발중인 v 0.0.2 에서는 구현하지 않습니다.
- 향후 GStreamer 표준으로 제안될 Format Draft는 Wiki에..

nnstreamer plugins

What nnstreamer  actually does

nnstreamer plugins

어떤 종류의 Plugin 들이…?

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

- Converter
  - 기존 Media Type → Tensor / Tensor → 기존 Media Type
- Filter
  - Caffe/Tensorflow 등 Framework 나 nnstreamer API로 만들어진 신경망 호출
- Utility
  - Stream Pipeline 제어, Tensor 변환/연산 등
- Connector
  - Gstreamer 외부 요소와의 연결 (비 A/V 센서, ROS, Application, Filesystem 등 src/sink)

nnstreamer plugins: tensor_converter

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

`nnstreamer/gst/tensor_converter/`

- Video (video/x-raw)
  - RGB/BGRx 지원. 기타 Colorspace는 향후 지원 할 계획.
  - RGB기준 3:Width:Height uint8 Tensor 출력 (uint8[1][height][width][3])
- Audio (audio/x-raw)
  - Channel:Frames-per-buffer <type> Tensor 출력 (Frames-per-buffer: buffer당 sample 수)
- Text (text/x-raw)
  - Frames-per-buffer uint8 Tensor 출력(Frames-per-buffer: buffer당 text 최대 길이)
- Binary Blob (application/octet)
  - 아무 Binary나 Tensor로 받아 들임. App개발자가 Tensor dimension/type을 지정해야 함.

nnstreamer plugins: tensor_converter

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

`nnstreamer/gst/tensor_converter/`

- 사용 예:

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

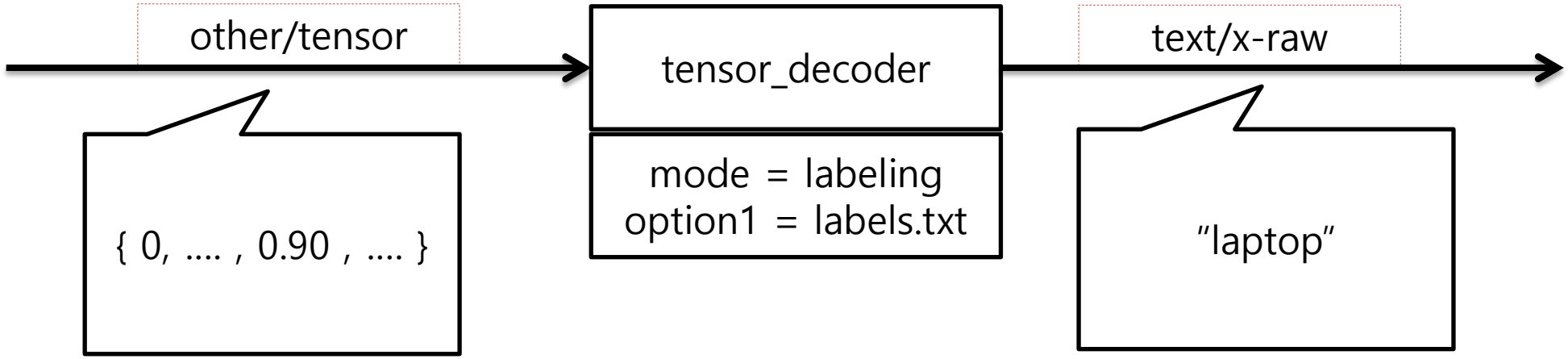
nnstreamer plugins: tensor_decoder

v0.0.2 (partial)

`nnstreamer/gst/tensor_decoder/` (부분 구현)

- 향후 Bounding box, object labeling 등 많이 쓰는 decode 기능 추가. (video output)
- 사용 예:

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



nnstreamer plugins: tensor_filter

v0.0.1/v0.0.2

`nnstreamer/gst/tensor_filter/`

- 각 신경망 Framework 는 하나의 “Subplugin”으로 지원함.
  - 현재 (TF-Lite, Custom) 지원.
  - 향후 Caffe/Caffe2, Tensorflow 추가.
  - Subplugin 형태로서 쉽게 추가 가능함. (별도 .so 파일로 추가)

- Custom subplugin
  - **`nnstreamer`** API 를 이용해 구현한 신경망 Binary를 사용.
  - “`tensor_filter_custom.h`” 참조

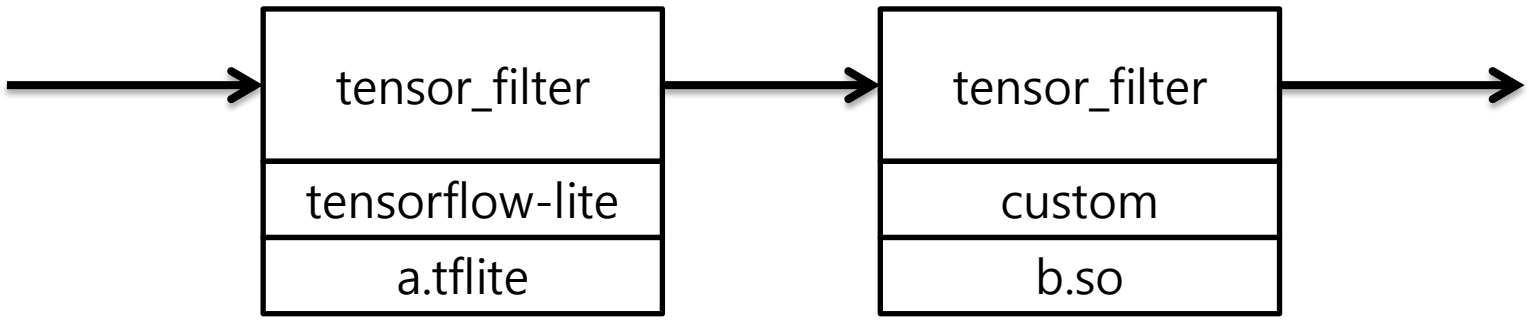
nnstreamer plugins: tensor_filter

v0.0.1/v0.0.2

`nnstreamer/gst/tensor_filter/`

- 사용 예:

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



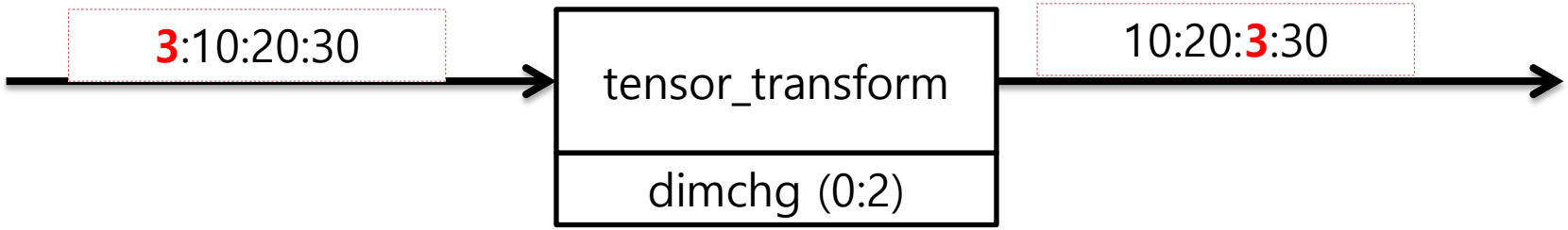
nnstreamer plugins: utility / tensor_transform

v0.0.1/v0.0.2

`nnstreamer/gst/tensor_transform/` (부분구현)

- Transpose, Type 바꾸기, Normalization 하기, 곱하기, 더하기, ……
- TODO: SIMD / Multi-op 지원
- 사용 예:

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



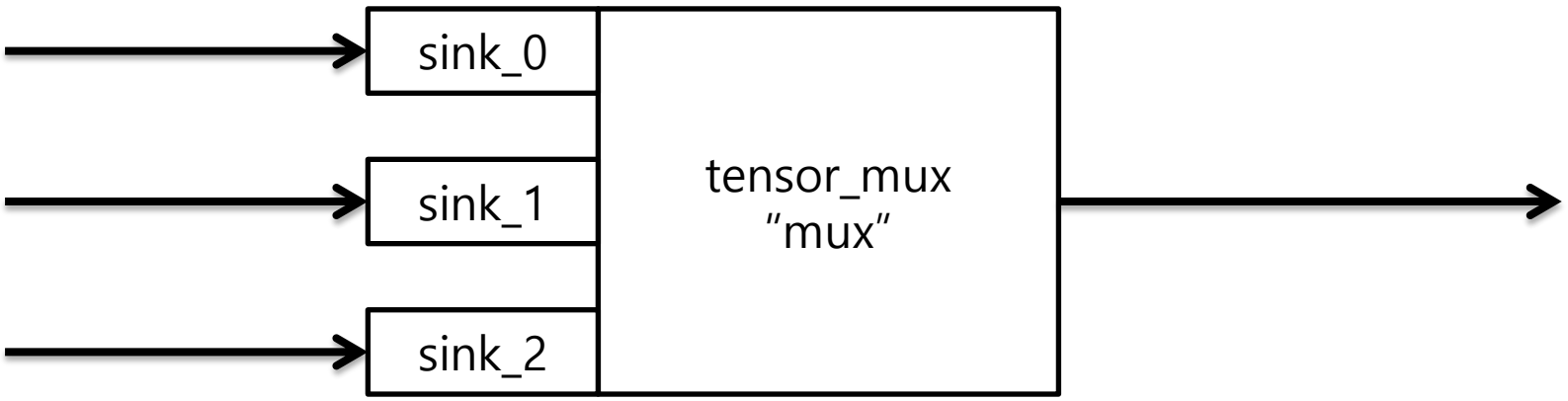
nnstreamer plugins: utility / tensor_mux

v0.0.2

`nnstreamer/gst/tensor_mux/`

- Tensor 들을 묶어서 하나의 Tensors 로 (v 0.0.2 한계: 16)
- 동기화 정책 표현 가능 (v 0.0.3 이후). Memcpy 없음.
- 사용 예:

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



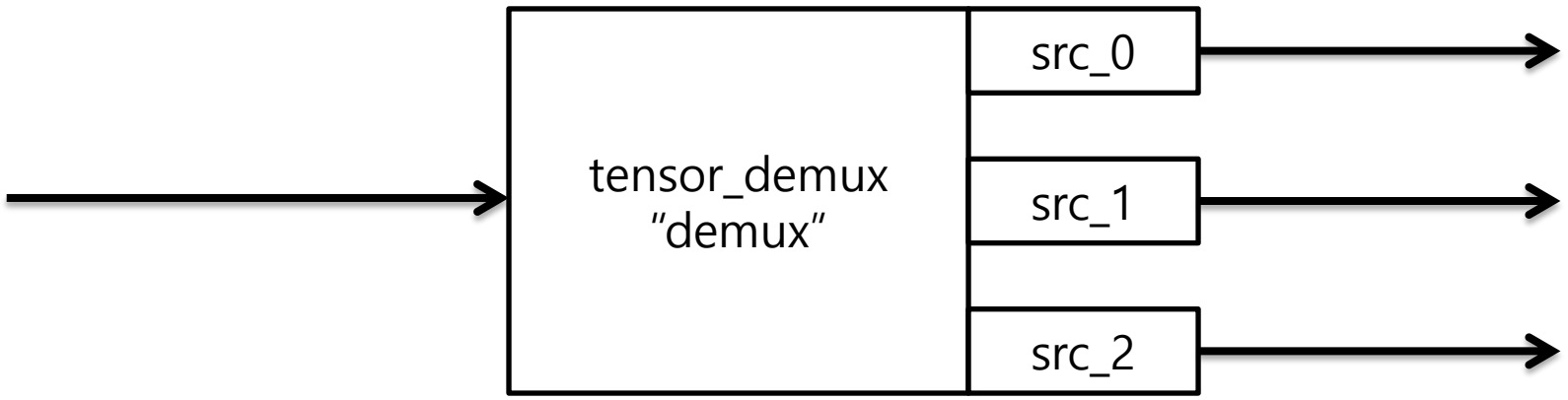
nnstreamer plugins: utility / tensor_demux

v0.0.2

`nnstreamer/gst/tensor_demux/`

- Tensors 들을 풀어서 여러 개의 Tensor 로.
- Memcpy 없음. “tensorpick” 옵션을 통해 개별 Tensor 선택 가능.
- 사용 예:

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



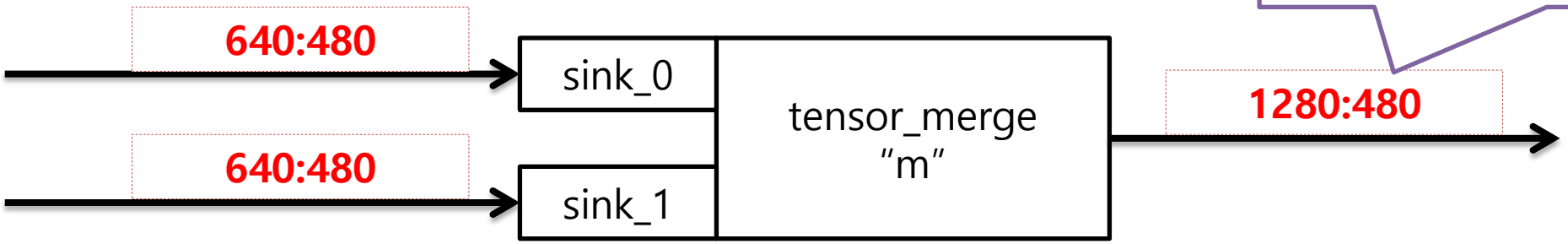
nnstreamer plugins: utility / tensor_merge

v0.0.2 (partial)

`nnstreamer/gst/tensor_merge/` (부분구현)

- Tensor 들을 묶어서 하나의 Tensor 로 (e.g., 좌우 카메라 화면 이어 붙이기)
- 동기화 정책 표현 가능 (v 0.0.3 이후). Memcpy 많음.
- 사용 예:

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



그 외 "640:960", "640:480:2", "2:640:480", "640:480/sum", "640:480/avg", 3개 이상 붙이기, 가로/세로로 4개 붙이기 등 다양한 기능이 향후 추가 가능.

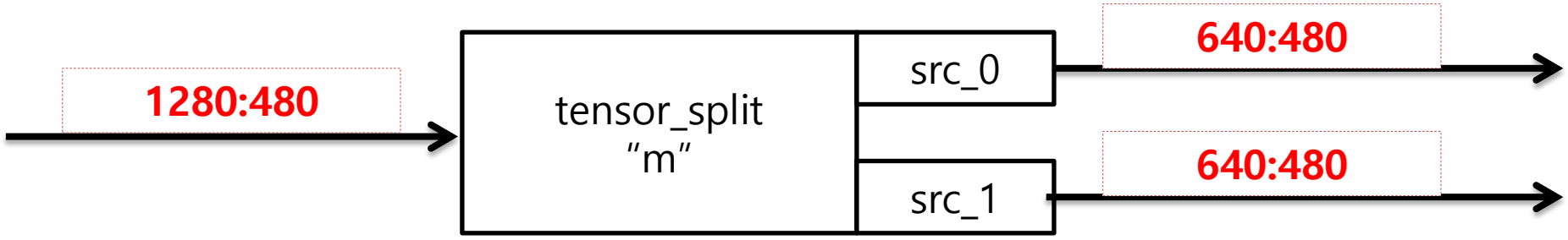
nnstreamer plugins: utility / tensor_split

future item

`nnstreamer/gst/tensor_split/` (미구현)

- Tensor 들을 잘라서 여러 개의 Tensor 로 (e.g., 좌우 화면으로 자르기)
- Memcpy 많음.
- 사용 예:

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



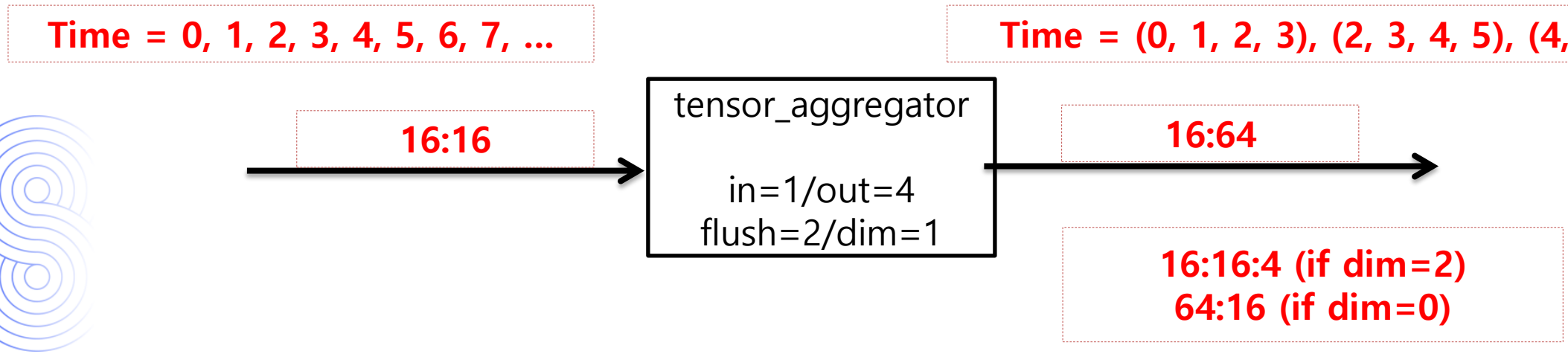
nnstreamer plugins: utility / tensor_aggregator

v0.0.2

`nnstreamer/gst/tensor_aggregator/`

- Tensor Stream의 Frame Sequence를 묶어서 제공함.
  (e.g., 현재부터 과거 10개의 Frame을 한번에 묶어서 신경망에 입력)
- 사용 예:

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



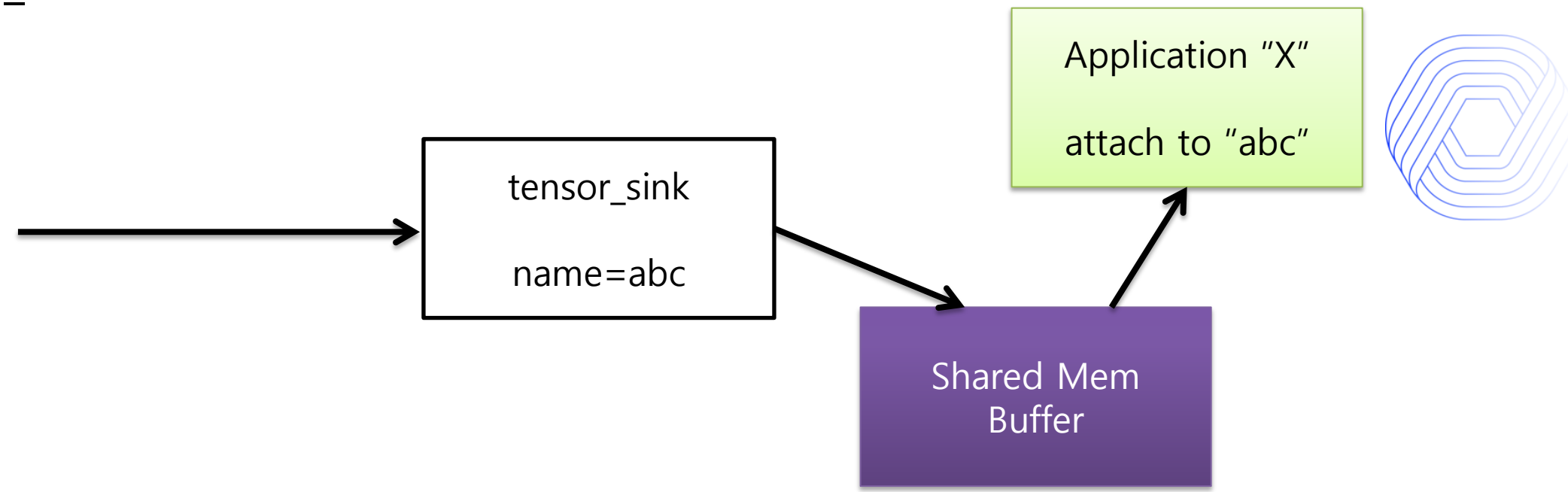
nnstreamer plugins: connector / tensor_sink

v0.0.1

`nnstreamer/gst/tensor_sink/`

- Tensor Stream의 Frame들을 Shared Mem으로 외부 App에게 제공
- 사용 예:

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



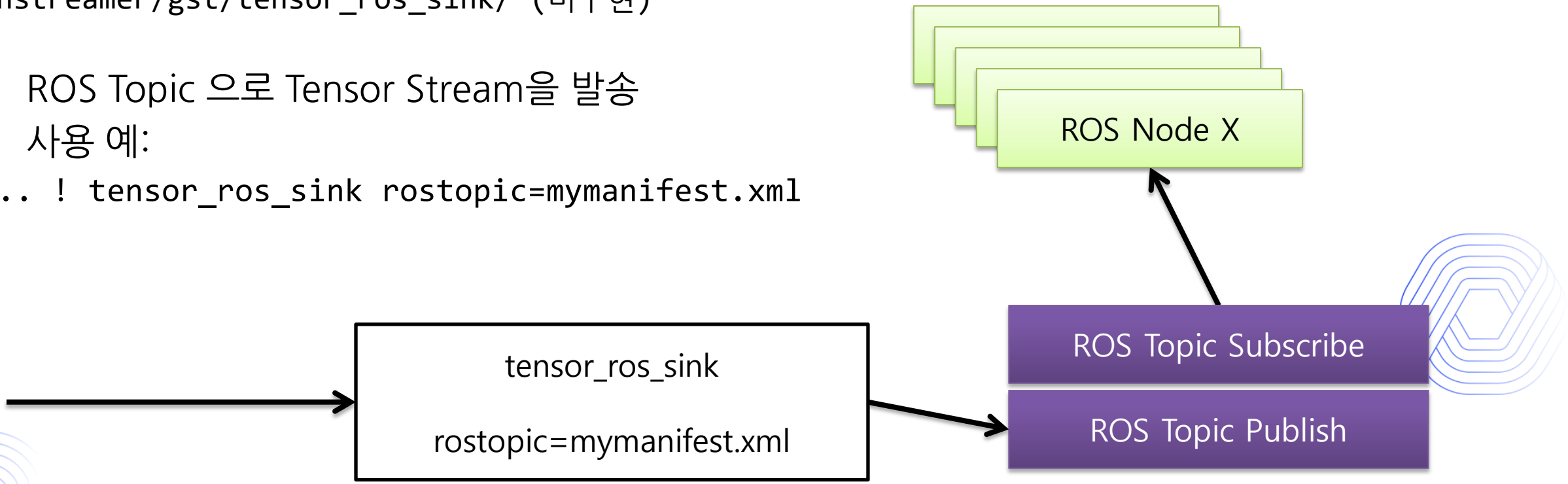
nnstreamer plugins: connector / tensor_ros_sink

Future Work

`nnstreamer/gst/tensor_ros_sink/ (미구현)`

- ROS Topic 으로 Tensor Stream을 발송
- 사용 예:

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



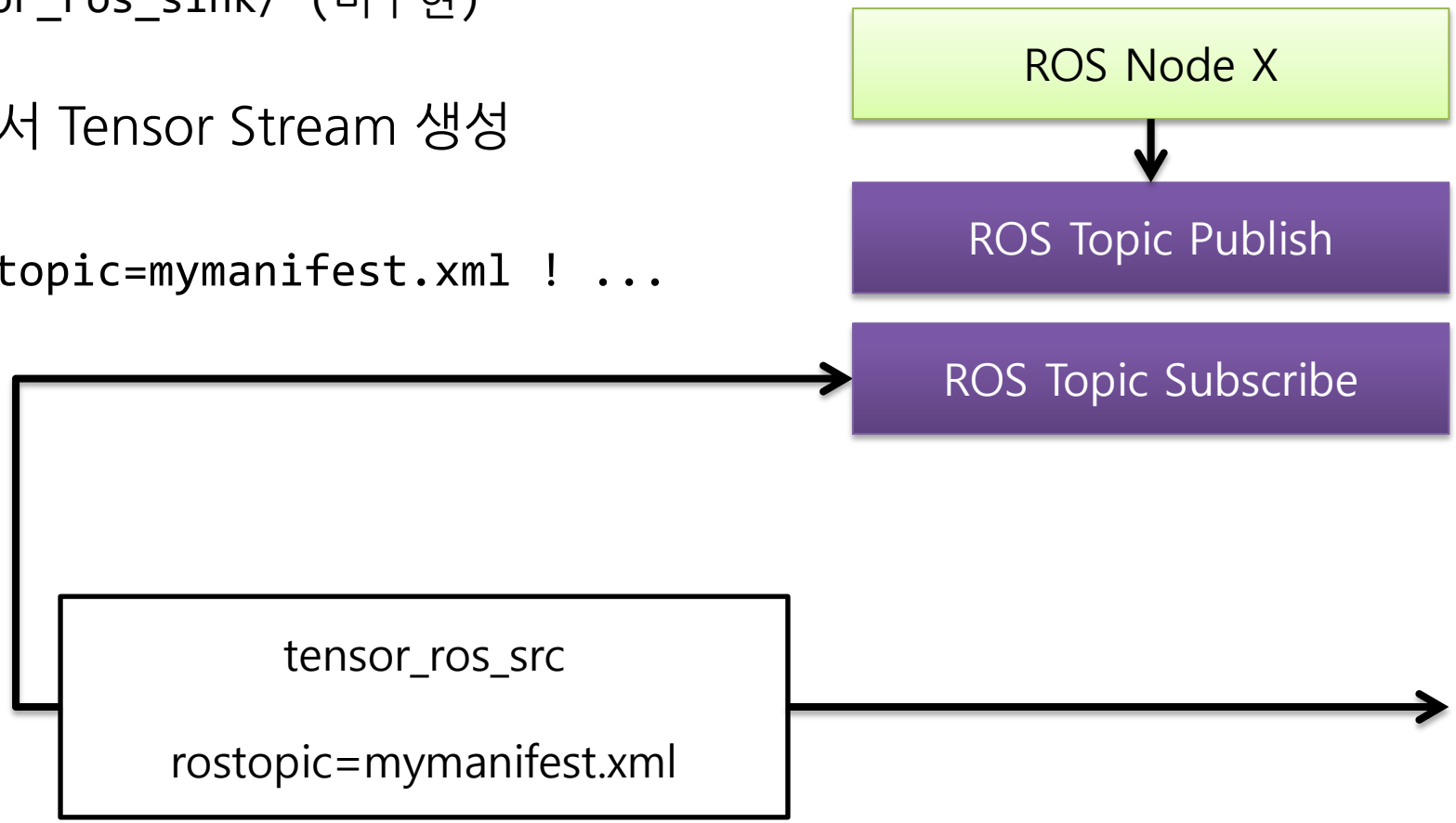
nnstreamer plugins: connector / tensor_ros_src

Future Work

`nnstreamer/gst/tensor_ros_sink/ (미구현)`

- ROS Topic 받아와서 Tensor Stream 생성
- 사용 예:

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



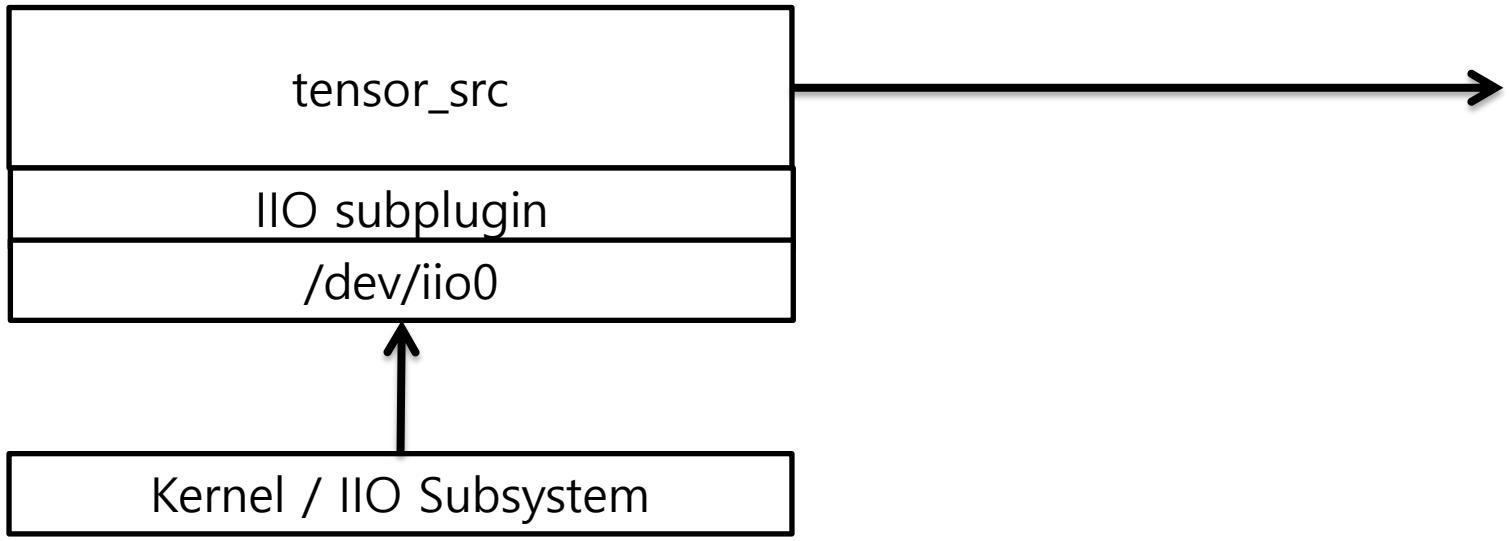
nnstreamer plugins: connector / tensor_src

Future Work

`nnstreamer/gst/tensor_ros_src/ (미구현)`

- 기타 Input Source / Sensor 로부터 Tensor Stream 생성
- 사용 예:

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



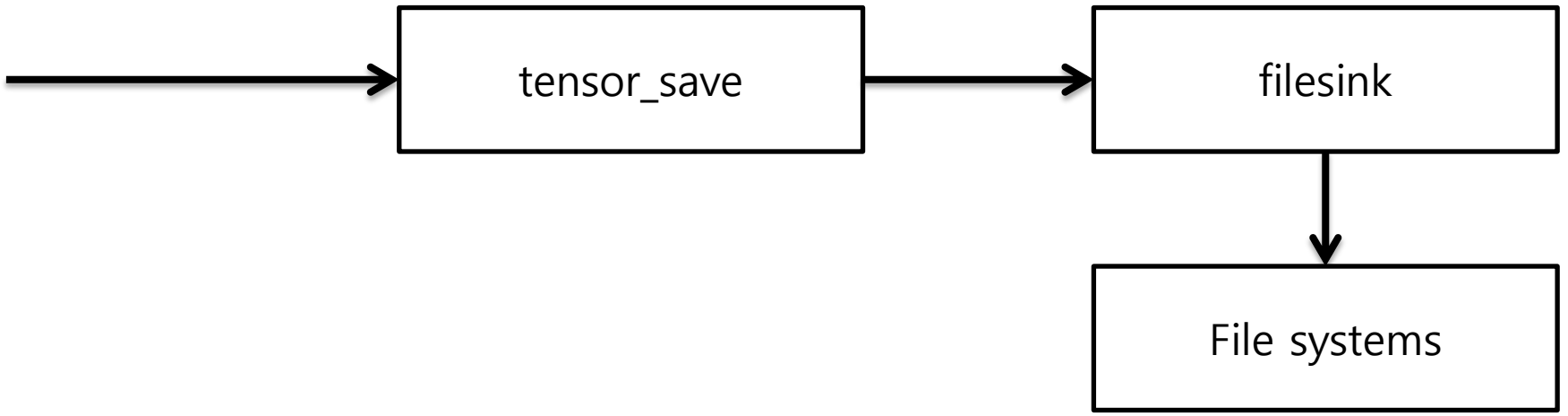
nnstreamer plugins: connector / tensor_save

Future Work

`nnstreamer/gst/tensor_saveload/ (미구현)`

- Tensor Stream을 File 저장을 위한 Headed Data로 전환
- 사용 예:

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



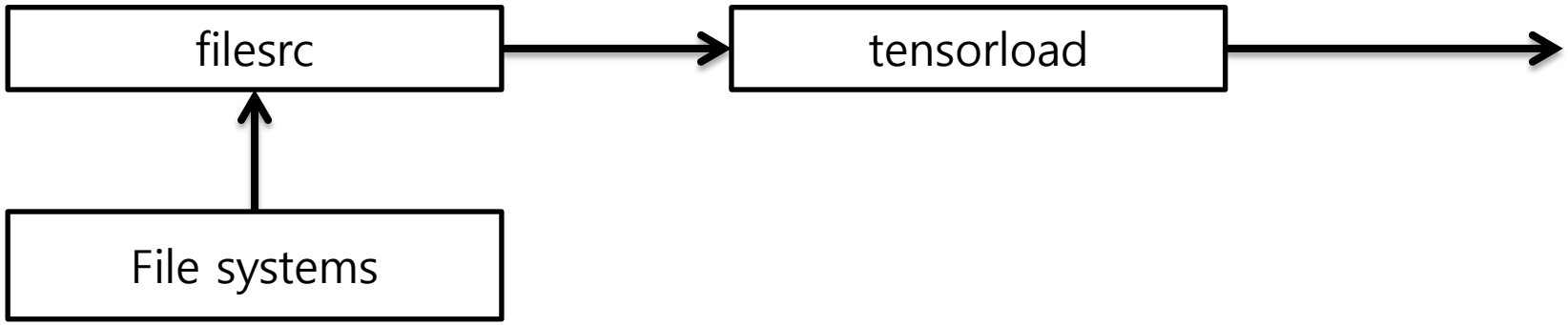
nnstreamer plugins: connector / tensor_load

Future Work

`nnstreamer/gst/tensor_saveload/ (미구현)`

- File 로부터 Tensor Stream (tensor/tensors) 을 생성
- 사용 예:

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

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



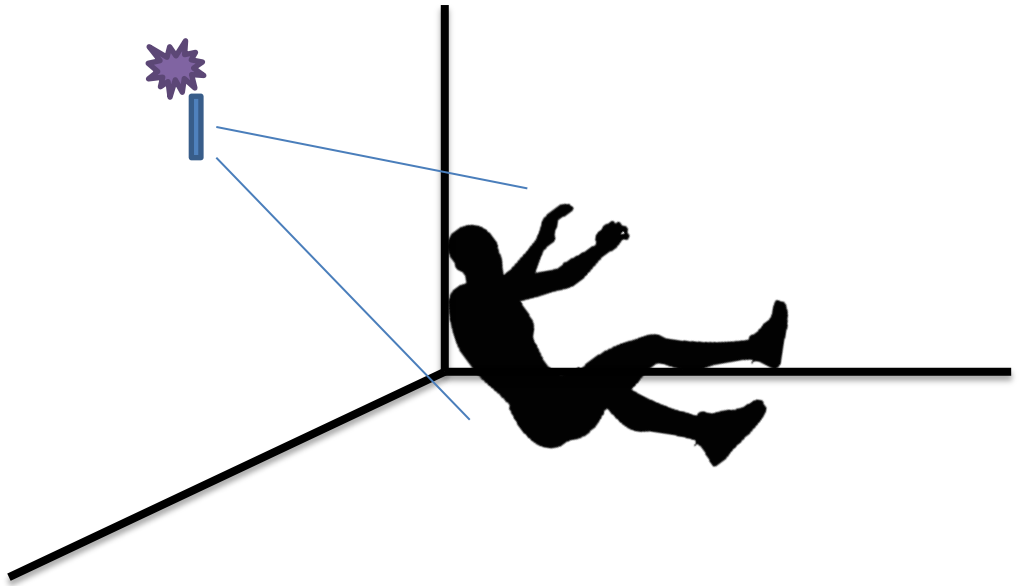
Usage example

nnstreamer  in action

Fall Detection Device, Elderly Care

SOSCON 2018

실 적용 사례

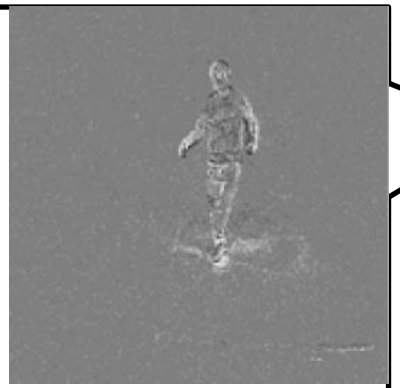


Images from: https://pixabay.com/ko/%EC%8B%A4%EB%A3%A8%EC%97%A3-%EA%B0%80-%EB%96%A8%EC%96%B4%EC%A7%80%EB%8A%94-%EC%B6%95%EC%86%8C-%EC%84%9C%EC%82%AC%EC%8B%9C-%EB%94%94%EC%9E%90%EC%9D%B8-%EC%95%84%EC%9D%B4%EC%BD%98-%EA%B8%88%EC%9C%B5-3172280/

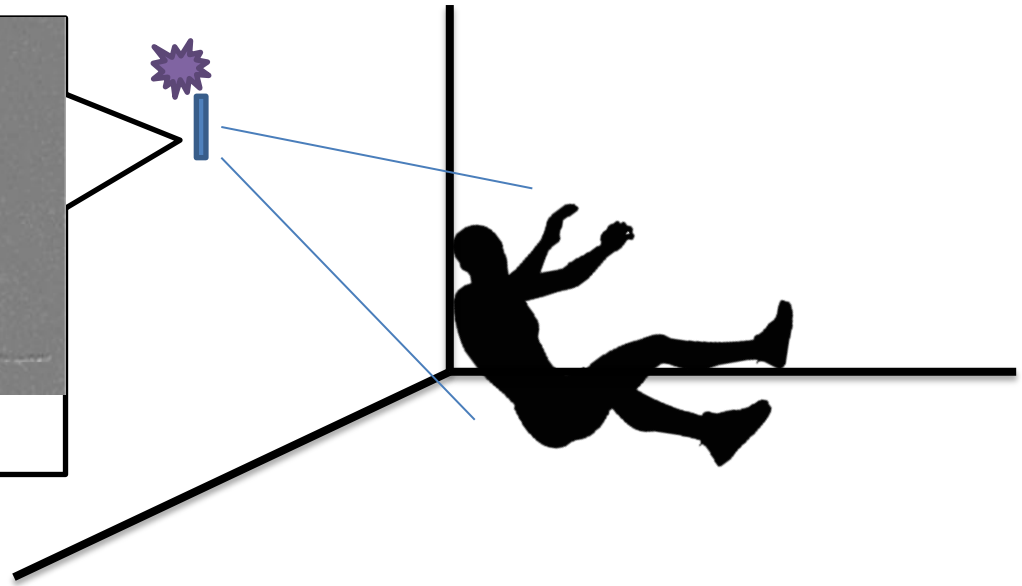
Fall Detection Device, Elderly Care

실 적용 사례

DVS 센서



(Samsung Artik 5 IoT)



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

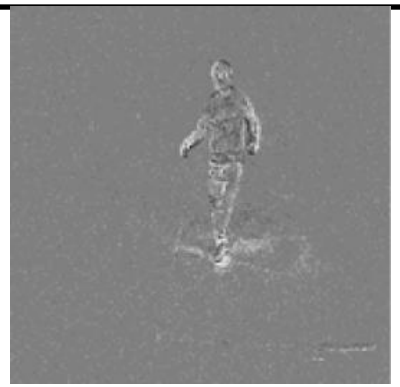
Images from: https://pixabay.com/ko/%EC%8B%A4%EB%A3%A8%EC%97%A3-%EA%B0%80-%EB%96%A8%EC%96%B4%EC%A7%80%EB%8A%94-%EC%B6%95%EC%86%8C-%EC%84%9C%EC%82%AC%EC%8B%9C-%EB%94%94%EC%9E%90%EC%9D%B8-%EC%95%84%EC%9D%B4%EC%BD%98-%EA%B8%88%EC%9C%B5-3172280/

Fall Detection Device, Elderly Care

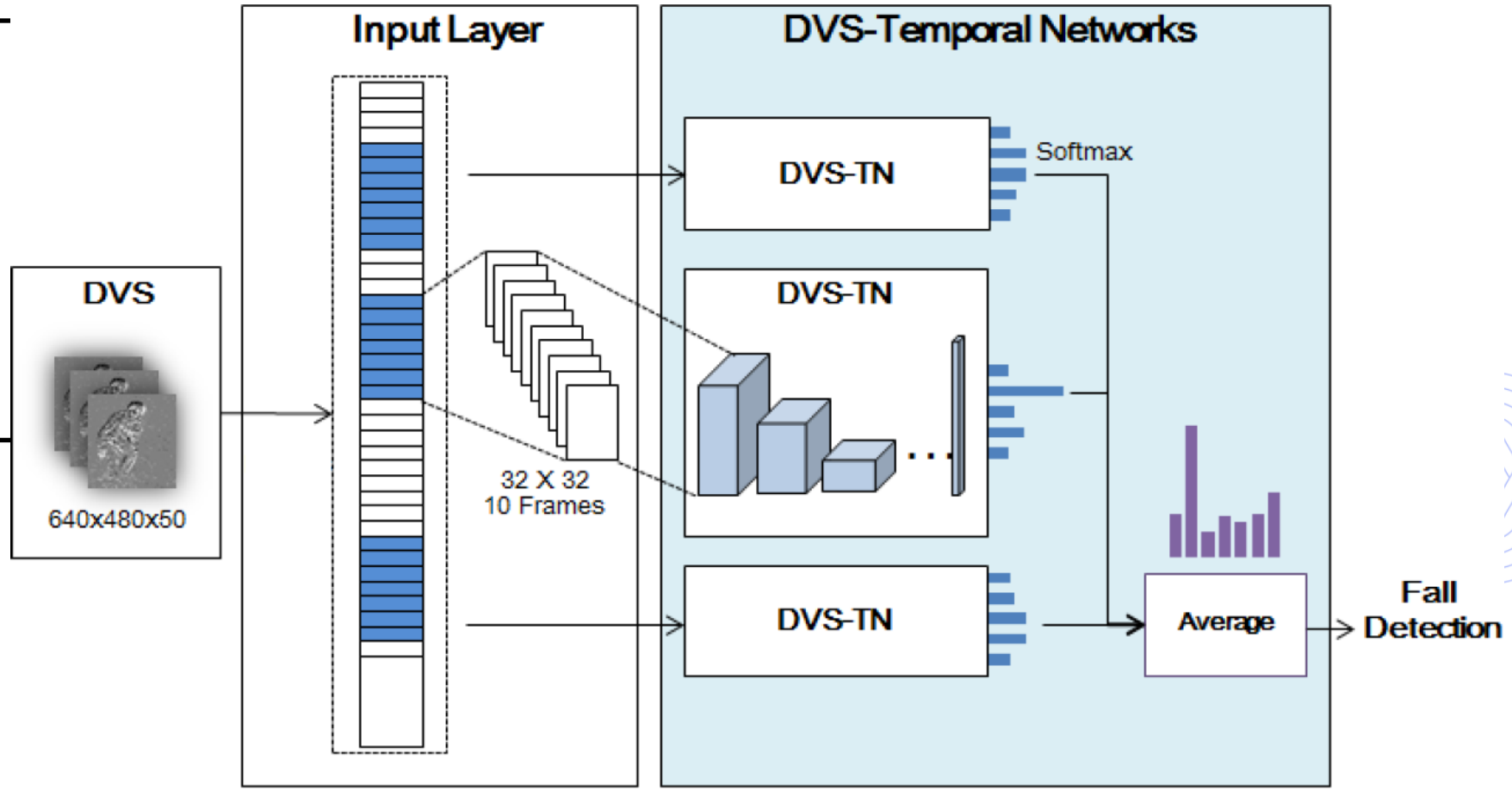
SOSCON 2018

실 적용 사례

DVS 센서



(Samsung Artik 5 IoT)

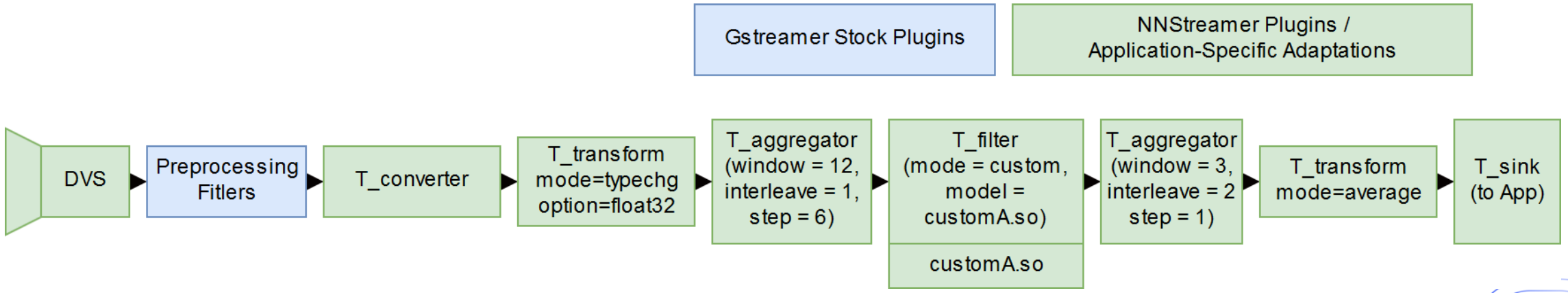


H. Lee, et. al., “Embedded Real-Time Fall Detection Using Deep Learning For Elderly Care”

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%B8%88%EC%9C%B5-3172280/

Fall Detection Device, Elderly Care

실 적용 사례

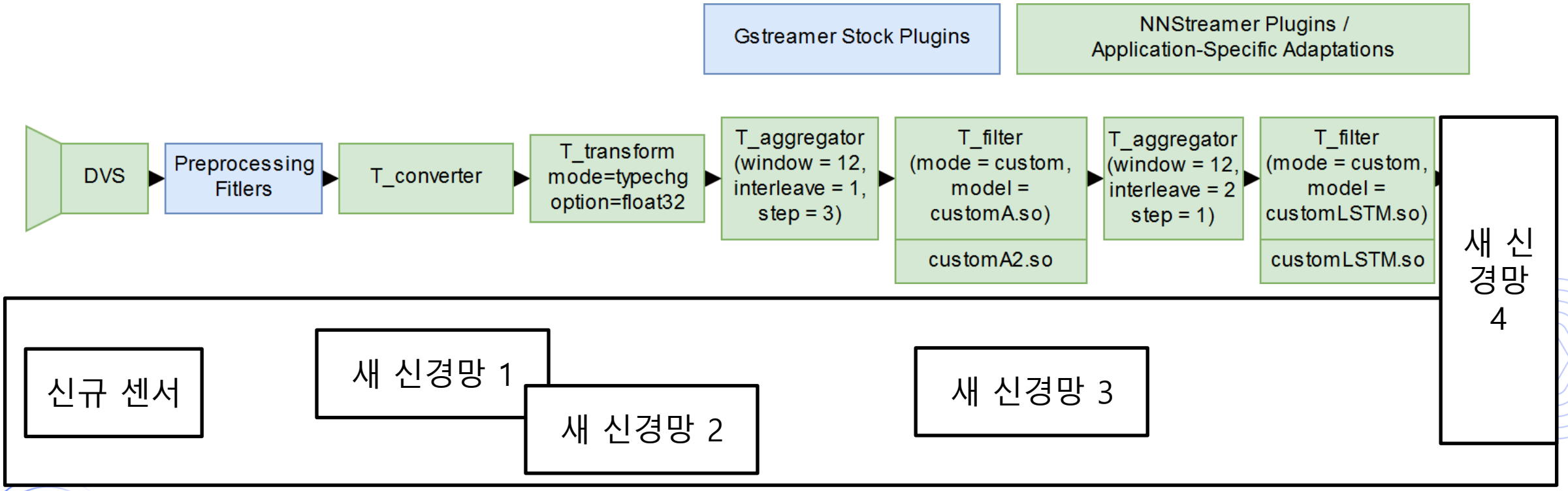


- nnstreamer 적용으로 생산성 (수천라인 → 수십라인) / 성능 (x ~4) 대폭 향상

➔ 신규 기능 / 신경망 추가할 여력 발생

Fall Detection Device, Elderly Care, “Next-Gen”

실 적용 사례



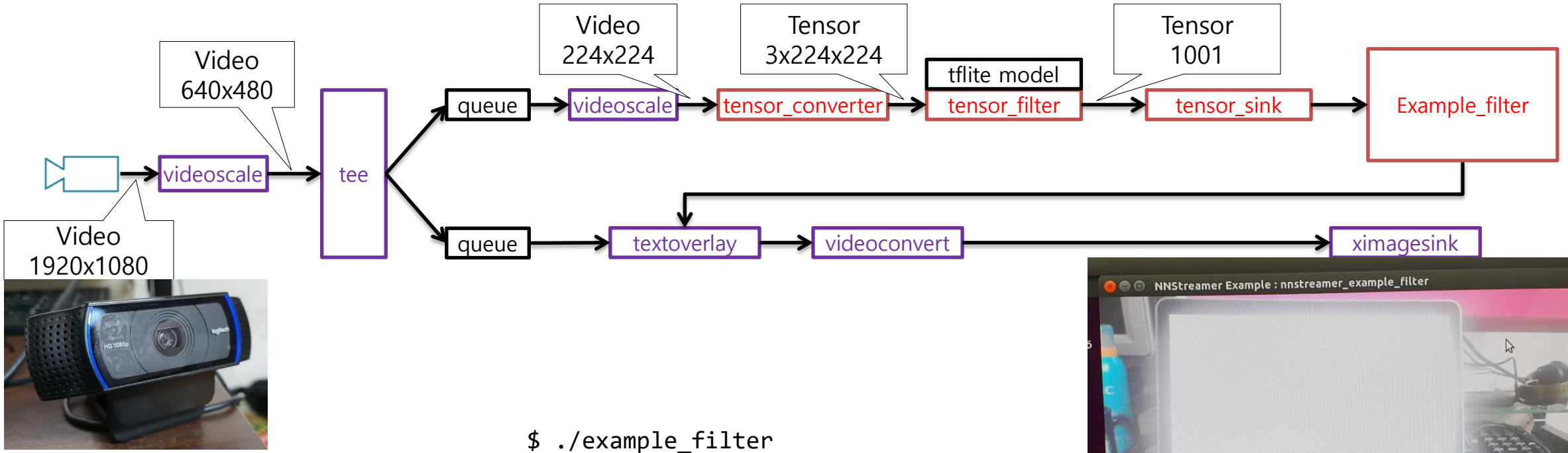
- LSTM 적용, 신경망 상호 작용 구조 변경, 신규 센서 및 신경망 추가 등.
- 매우 빠른 시간 내에 고성능/저비용 Solution 제작.

Demonstration

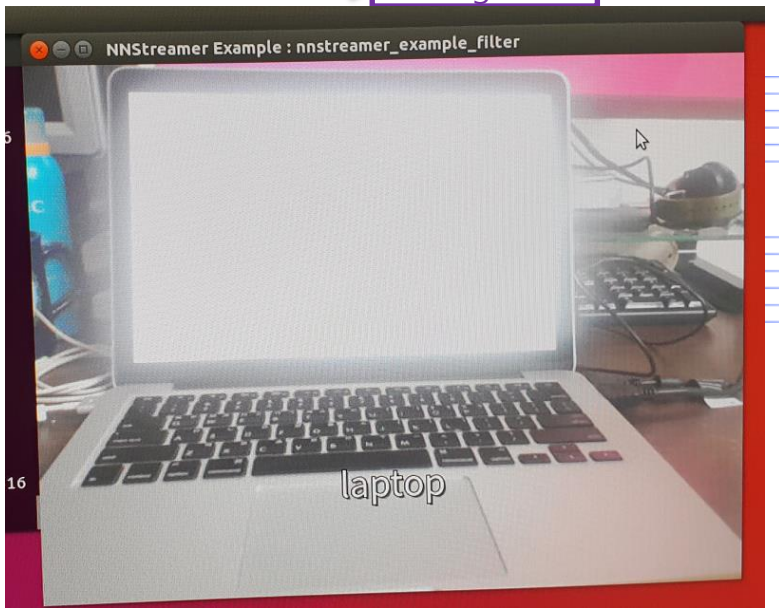
nnstreamer  live action

Demonstration

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

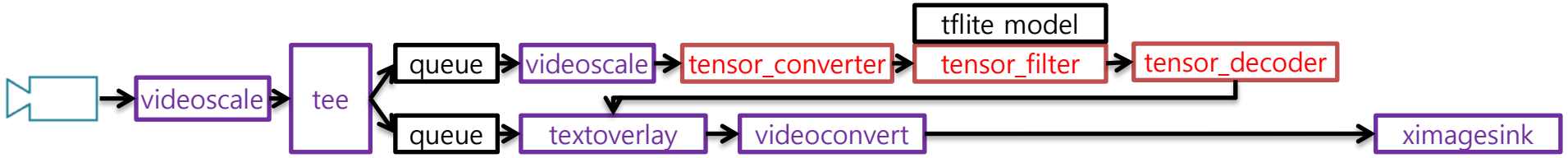



```
$ ./example_filter

or

$ python example_filter.py
```



Demonstration

nnstreamer @ Ubuntu Demo




```
$ gst-launch-1.0    v4l2src name=cam_src ! videoscale ! \   # "cam_src"에서 영상 받아서 scale
      video/x-raw,width=640,height=640,format=RGB ! \       # 3x640x480의 RGB Stream
      tee name_t_raw \                                      # Data Stream 분기
      t_raw. ! queue ! videoscale ! video/x-raw,width=224,height=224 ! \   # 224x224 RGB 로 전환
            tensor_converter ! \                                           # "tensor" 포맷으로 전환
            tensor_filter framework=tensorflow-lite model=mobilenetssd.tflite ! \ # tflite 모델 부름
            tensor_decoder mode=labeling option1=labels.txt ! txt. \       # 결과값 해석 방법 지정 ➔ txt
      t_raw. ! queue ! textoverlay name=txt ! \                            # txt로 들어온 문자열을 합성
            videoconvert ! ximagesink                                      # 합성결과를 화면에 출력
```

CI / CD & How to Contribute

How nnstreamer  is developed
How you can contribute

nnstreamer CI: TAOS-CI

Continuous Integration (CI)

CI system을 통해 모든 개별 Commit은 자동으로 검수

- Regression 방지
  - Build & Unit Test
- Coding Style/Rule 강제
- Integration 확인
  - Tizen/Ubuntu Packaging & UI App Test

- 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)

CD system을 통해 설치용 Package 상시 릴리즈

- Ubuntu: launchpad PPA
- Tizen: OBS
- 향후 지속 추가

How to Contribute?

You are welcomed to contribute!

Github (github.com/nnsuite/nnstreamer) 에서 개발 진행

- Issue 를 통해 개발할 기능, 해결할 버그를 등록.
- Pull Request를 통해 해당 Issue를 해결하는 Code를 등록.
  - Coding Style/Rule 및 모든 Build/Unit/Integration Test를 통과해야 함.
    - Submit 전에 미리 빌드/테스트 해주세요!
  - 신규 기능은 해당 기능을 위한 Unit Test를 포함하는 Commit도 추가해야 함.
  - Bug Fix는 해당 Bug를 확인하는 Unit Test를 추가해야 함.
  - Maintainer/Reviewer와의 적극적인 소통 권장

Work In Progress / Plans

Future of nnstreamer

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

Conclusion

nnstreamer 

Recommended Applications

혹시 이런 것을 만드신다면…

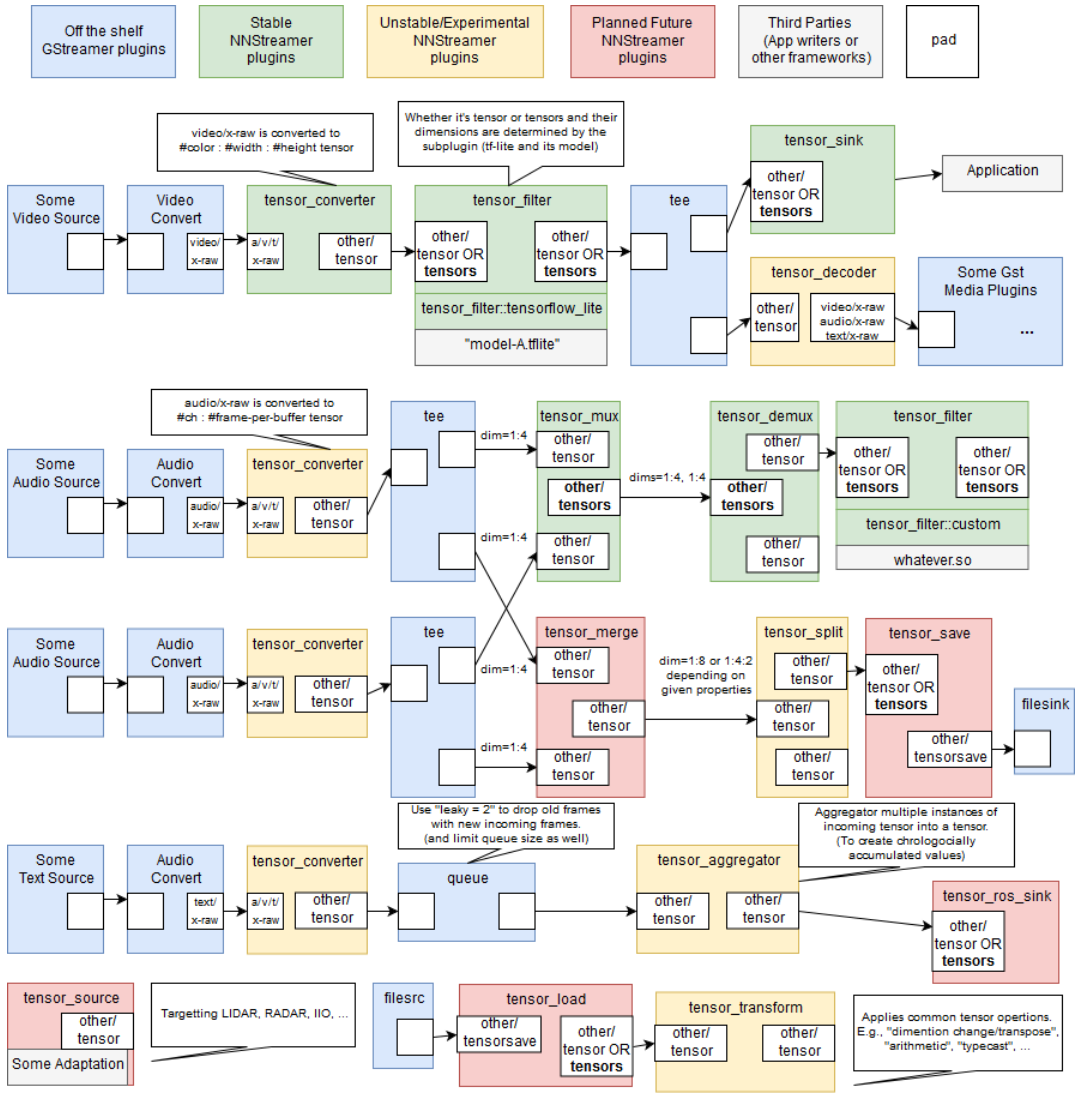
- Cloud/Server가 아닌 Device에서 직접 신경망 처리
- 실시간으로 생성되는 Audio/Video Stream애 신경망 처리
- 한 Device 혹은 한 Stream Pipeline 내에 여러 신경망을 혼합 사용
- 기존 Audio/Video Media Filter들의 재사용
- Stream Topology가 복잡하거나 동적임
- 고성능 Data Stream 처리 필요
- …

물론, 그 모든 것을 다 잘 직접 구현해서 쓰실 수 있으시면 상관 없습니다. ^^
그런데… “Don’t Invent the Wheel!”

nnstreamer Cheat Sheet

SOSCON 2018

그 플러긴들 어디다 어떻게 쓰지요??



https://github.com/nnsuite/nnstreamer/wiki/AD%3A-Overall-Design

nnstreamer github 내 Wiki page 참조

THANK YOU

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018