



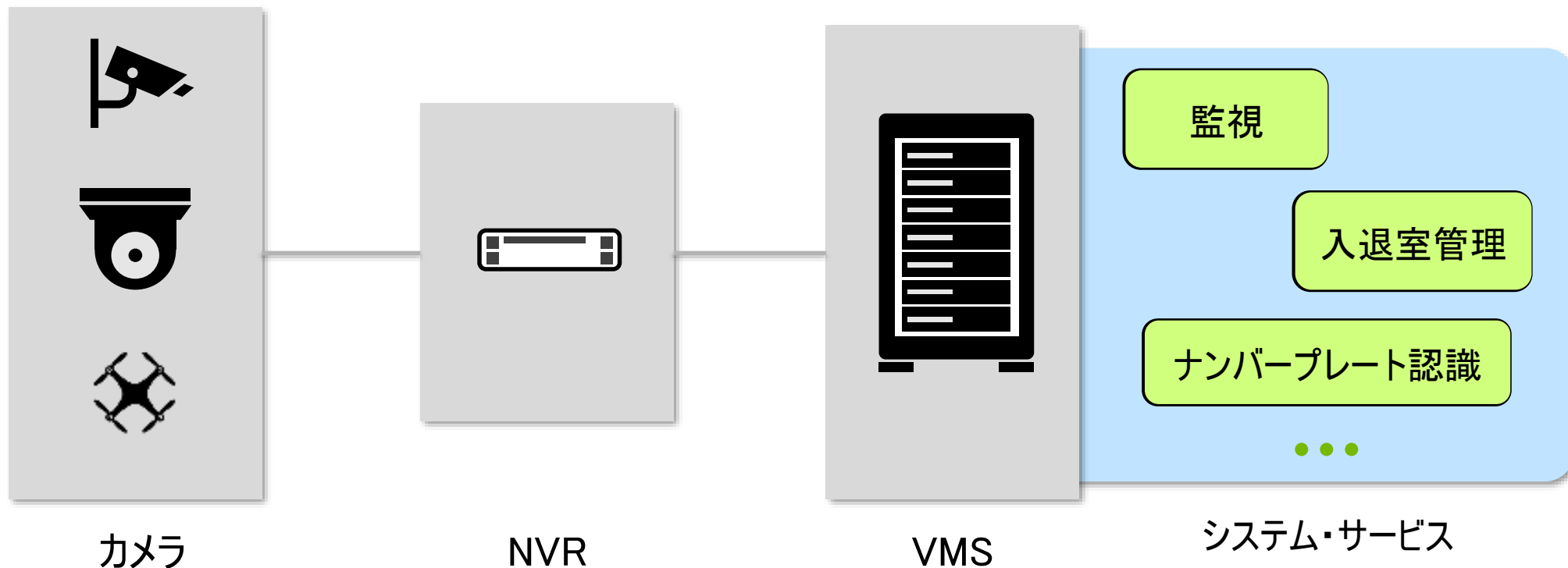
IVA と DeepStream SDK

NVIDIA Corporation, Technical Product Marketing Manager 矢戸 知得

アジェンダ

- インテリジェント・ビデオ解析
- DeepStream 紹介
- リファレンス・アプリケーション
- GStreamer
- メタデータ
- パイプライン・アーキテクチャ
- プラグイン
- メモリ管理
- パフォーマンス計測

「監視カメラ」→ インテリジェント・ビデオ解析 (IVA)



事例 - MotionLoft

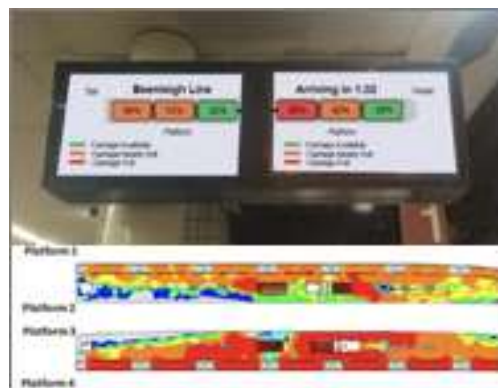
With Jetson



IVA とスマートシティー



入退場ゲート管理



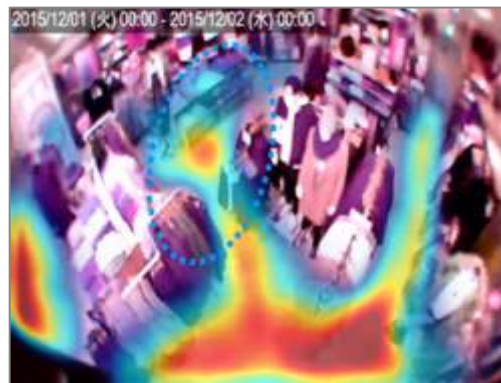
公共交通機関の利用



スマート パーキング



交通工学



小売り分析



インフラ

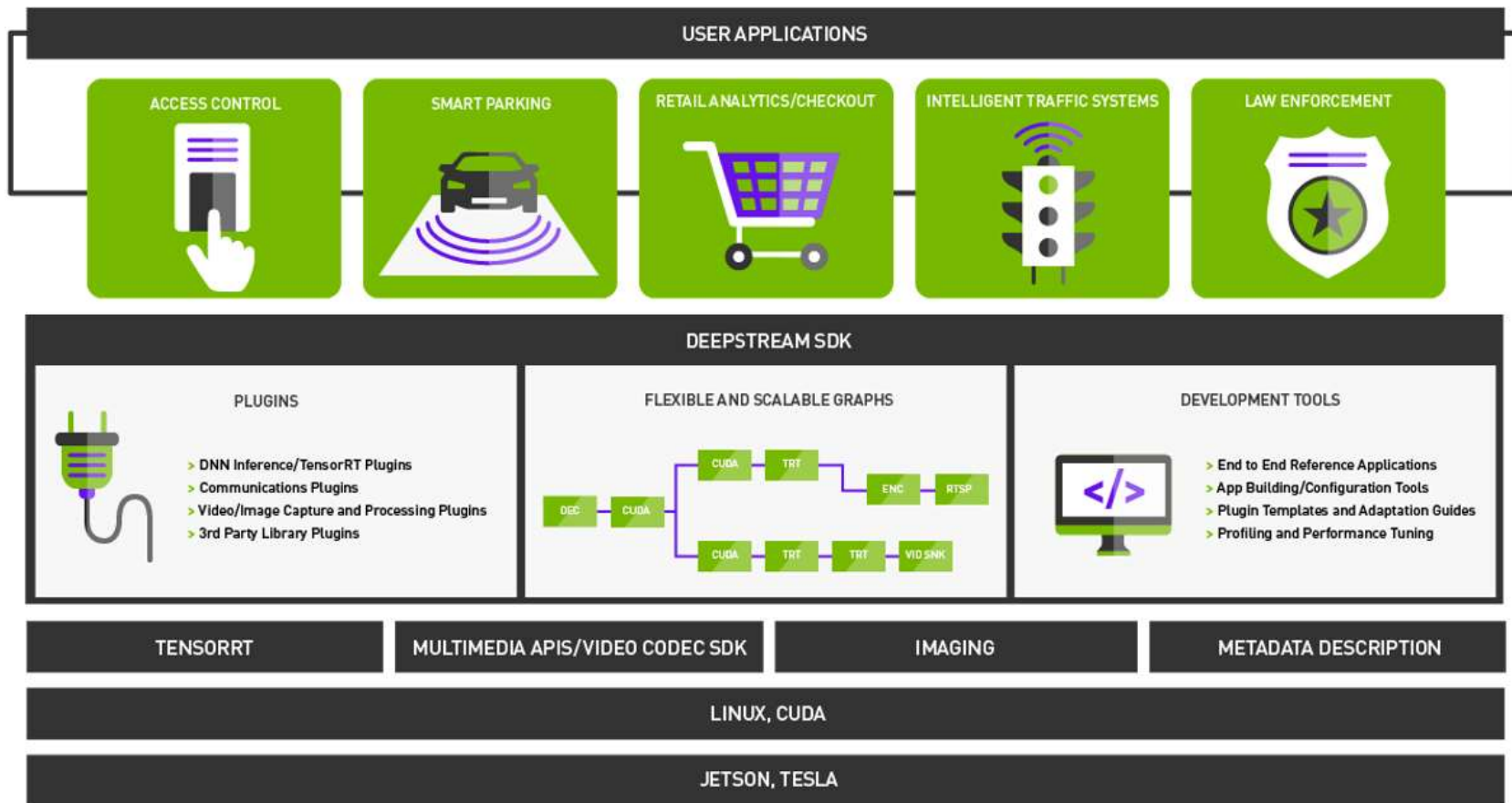


物流



ビデオ分析

DeepStream ソフトウェア スタック



Deepstream 2.0

DeepStream on Tesla v1.5



Tesla P40, P4

API	C++
Streams	Multi
Graph	Fixed function
DNNs	Single
Examples	DNNs

DeepStream on Jetson v1.5



Jetson TX1, TX2

API	Modular plugins. Gstreamer based
Streams	Single, Multi thru multi-app
Graph	Custom
DNNs	Multi
Examples	Full app, multi-DNNs, tracking

DeepStream v2.0



Tesla



Jetson *

API	Unified edge-cloud
Streams	Multi
Graph	Custom
DNNs	Multi
Examples	Multiple apps, more plugins, multi-DNNs

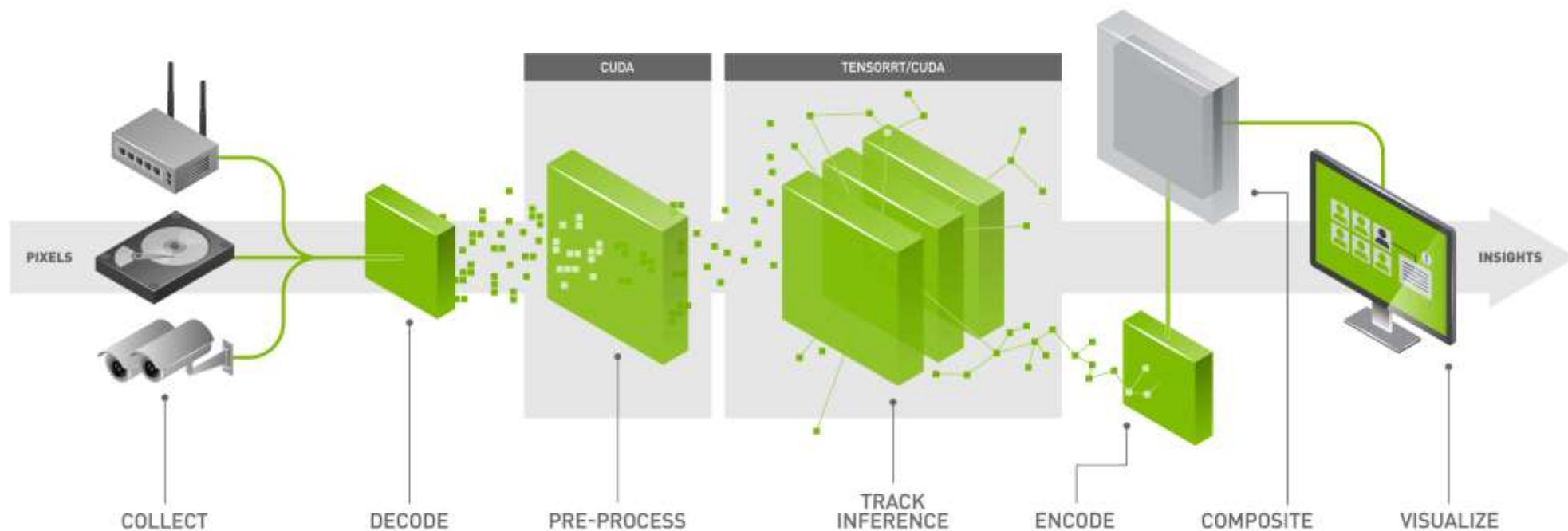
New modular framework and APIs →

Previous

Tesla - Now
Jetson - 2H'18

例 ; リファレンス・アプリケーション

インテリジェント ビデオ分析の典型例



DeepStream シングル ストリーム 出力



DeepStream リファレンス サンプル アプリケーション

車両検出、トラッキング、識別



DeepStream リファレンス アプリケーション 出力



DeepStream リファレンス アプリケーション

25x 720p ストリーム時の システム構成 と パフォーマンス

CPU - Intel® Xeon(R) CPU E5-2620 v4 @ 2.10GHz × 2

GPU - Tesla P4

システム メモリ - 256 GB DDR4, 2400MHz

Ubuntu 16.04

GPU Driver - 396.26

CUDA - 9.2

TensorRT - 4.0

GPU クロック周波数 - 1113 MHz

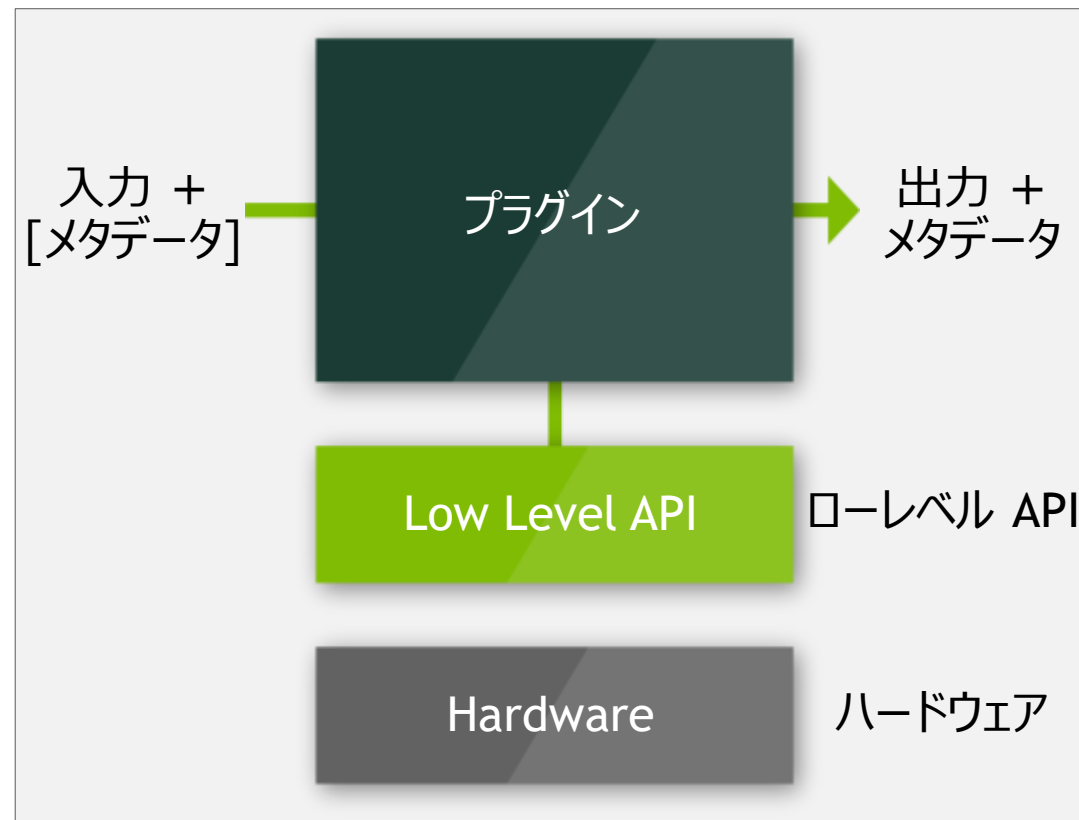
```
GPU 00000000:05:00.0
Utilization
  Gpu           : 59 %
  Memory        : 30 %
  Encoder        : 0 %
  Decoder        : 88 %
GPU Utilization Samples
  Duration      : 16.44 sec
  Number of Samples : 99
  Max           : 83 %
  Min           : 54 %
  Avg           : 65 %
Memory Utilization Samples
  Duration      : 16.44 sec
  Number of Samples : 99
  Max           : 39 %
  Min           : 28 %
  Avg           : 32 %
ENC Utilization Samples
  Duration      : 16.44 sec
  Number of Samples : 99
  Max           : 0 %
  Min           : 0 %
  Avg           : 0 %
DEC Utilization Samples
  Duration      : 16.44 sec
  Number of Samples : 99
  Max           : 100 %
  Min           : 63 %
  Avg           : 87 %
```

GStreamer

DeepStream の 構成要素

GStreamer プラグイン

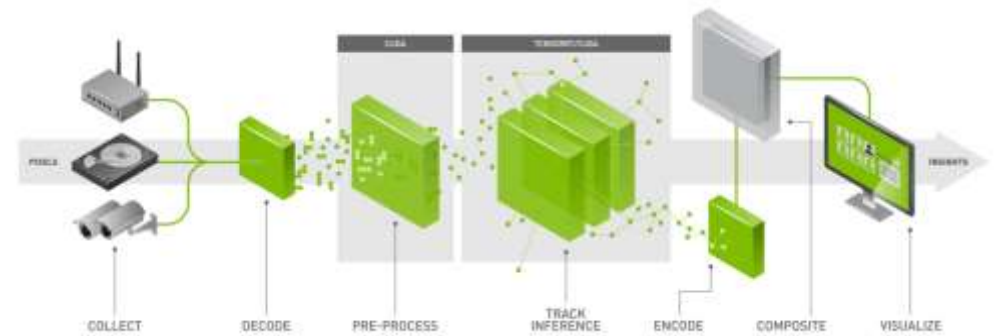
- オープンソースの GStreamer フレームワーク
- パイプライン アーキテクチャによる プラグイン
- 高水準のコンポーネント相互接続を可能にする
グラフ構造ベースのパイプラインインターフェイス
- GPU と CPU のヘテロジニアス並列処理
- 並列化と同期処理を隠蔽
- マルチスレッド



GStreamer

プラグイン

- ソース エLEMENT
... データを生成 (出力)
- フィルター エLEMENT
... データの加工 (入力 と 出力)
- シンク エLEMENT
... データのエンドポイント (入力)

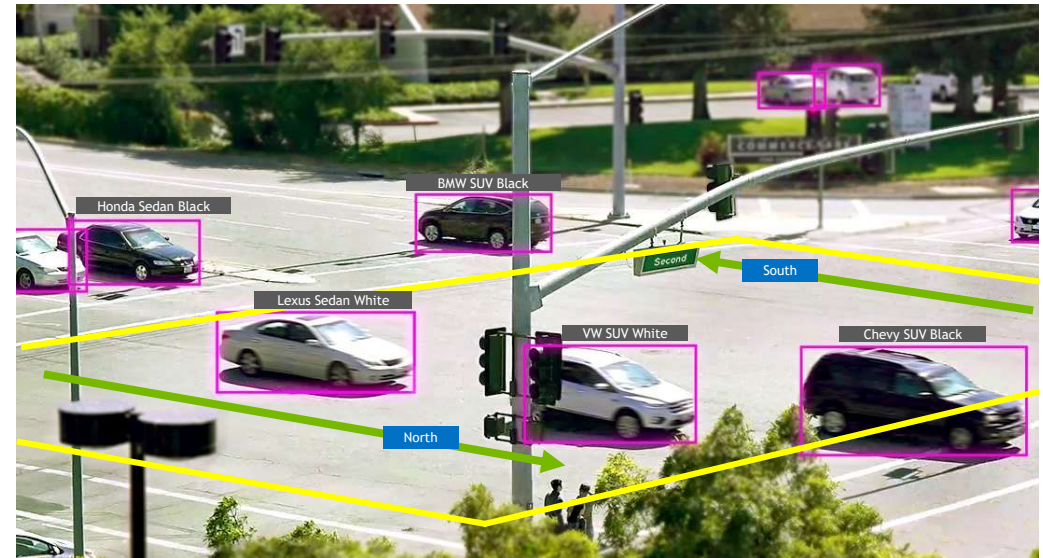


<https://gstreamer.freedesktop.org/>

メタデータ

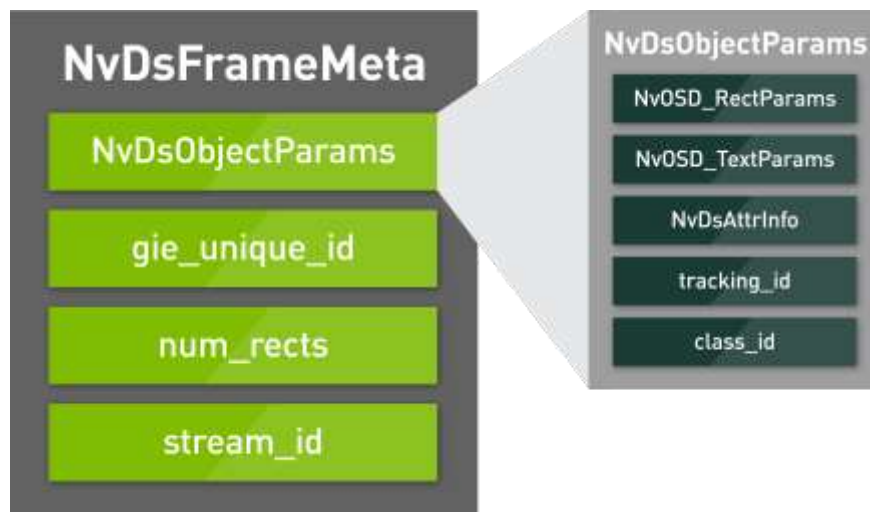
DeepStream における メタデータの取り扱い

- ▶ メタデータはグラフ内のプラグインによって生成される
- ▶ プラグインはメタデータを順次追加可能
- ▶ 生成後のメタデータは任意のタイミングで使用可能
- ▶ メタデータの例
 - ▶ 検出された物体の座標や種類、属性
 - ▶ ユニーク ID (トラッカー ID)
 - ▶ ROI 表示座標 (Regions of interest)
 - ▶ ソース ID、GPU ID
 - ▶ レンダリング情報、他



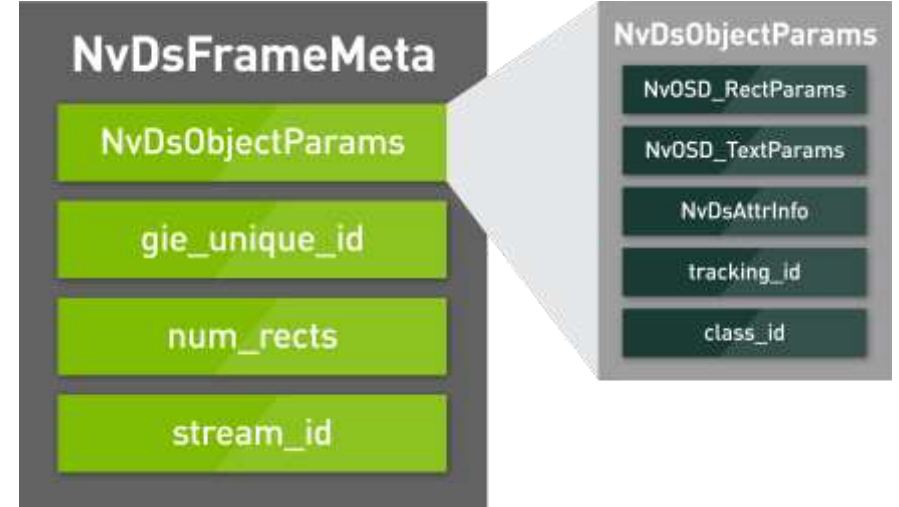
メタデータのデータ構造 (1/2)

- ▶ **NvDSObjectParams** - フレーム内で検出された物体のメタデータのサブセット
- ▶ **GIE_Unique_ID** - 各ニューラルネットワークに割り当てられるユニーク ID
- ▶ **Num_rects** - フレーム内で検出された物体数
- ▶ **Stream_Id** - メタデータが属するストリームの ID (マルチストリームの場合)



メタデータのデータ構造(2/2)

- ▶ **NvOSD_RectParams** - バウンディング・ボックス座標
- ▶ **NvOSD_TextParams** - 画面表示に必要なラベル情報
(例：白い車、メルセデス、セダン)
- ▶ **NvDSAttribinfo** - 物体の属性 (タイプ、色、メーカー)
- ▶ **Tracking_ID** - 物体のトラッカー・ユニーク ID
- ▶ **Class_ID** - 物体の種類 (人、車、自転車、道路標識)



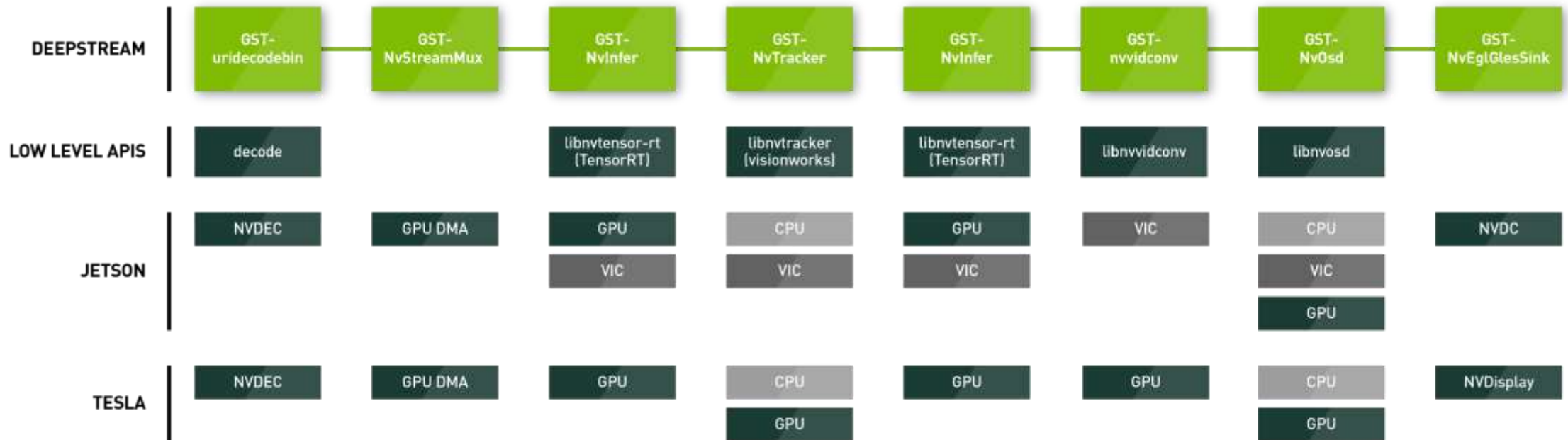
パイプライン アーキテクチャ

DeepStream リファレンス サンプル アプリケーション

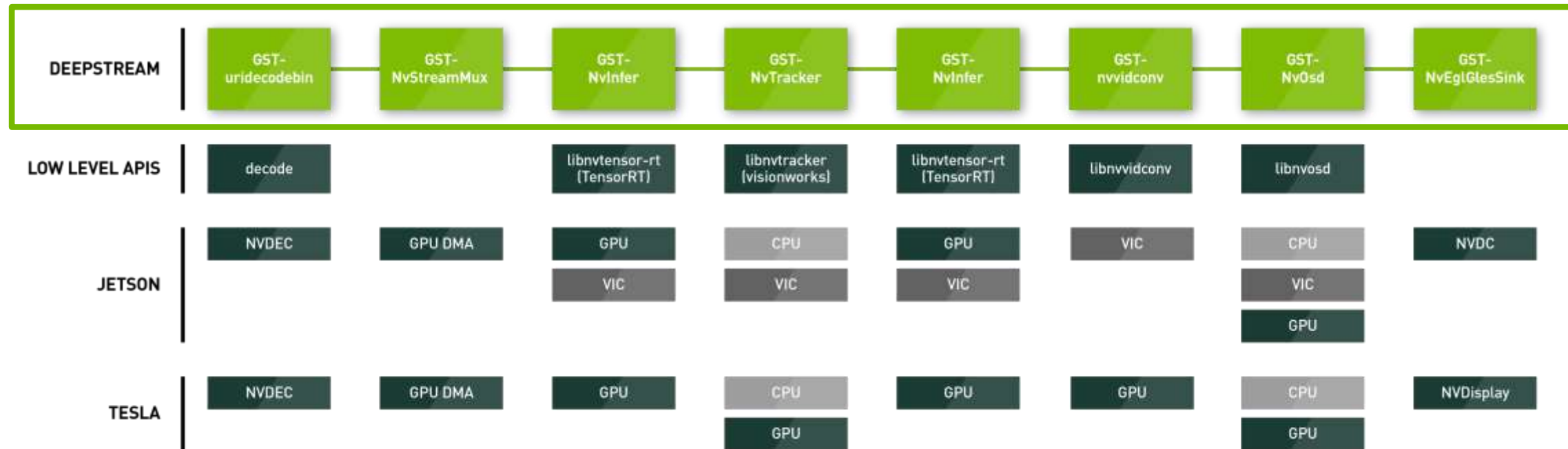
車両検出、トラッキング、識別



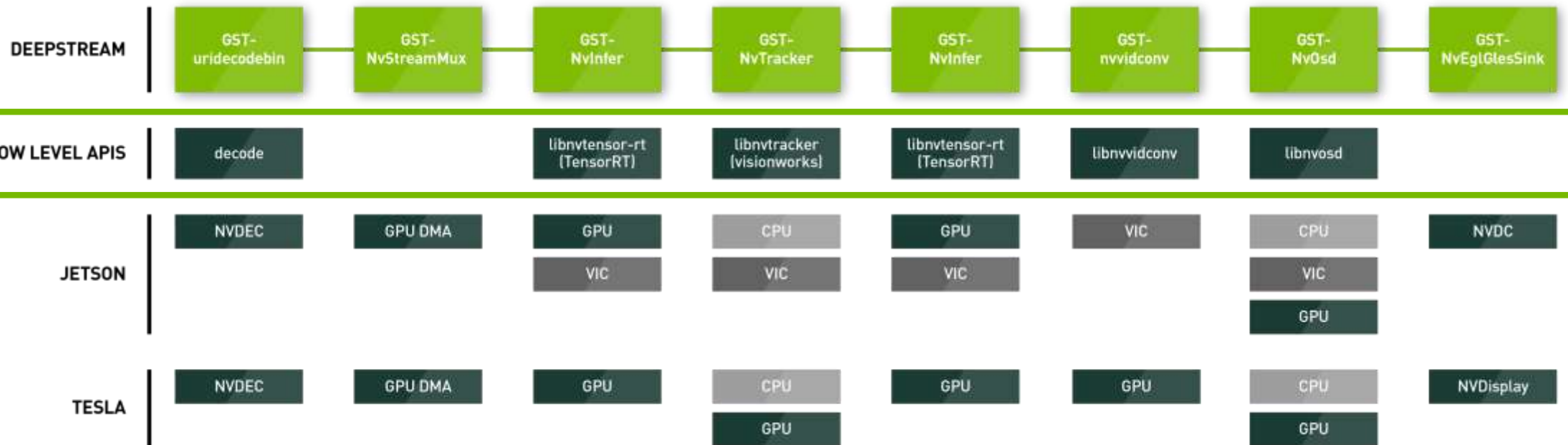
DeepStream パイプライン アーキテクチャ



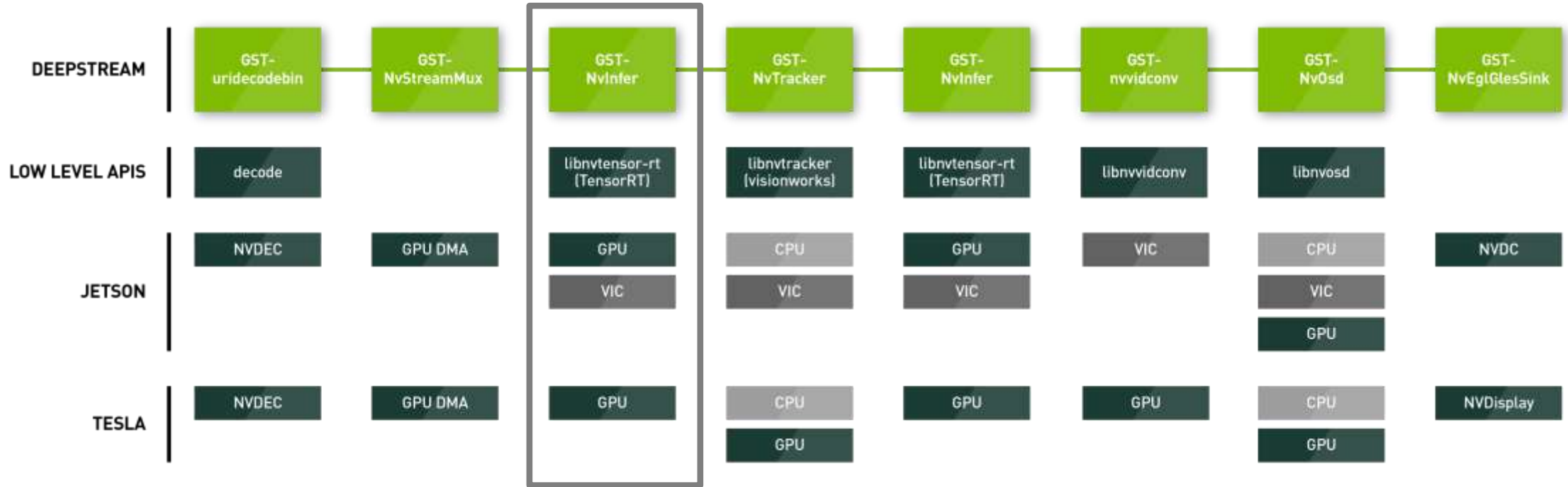
DeepStream パイプライン アーキテクチャ



DeepStream パイプライン アーキテクチャ



DeepStream パイプライン アーキテクチャ



プラグイン

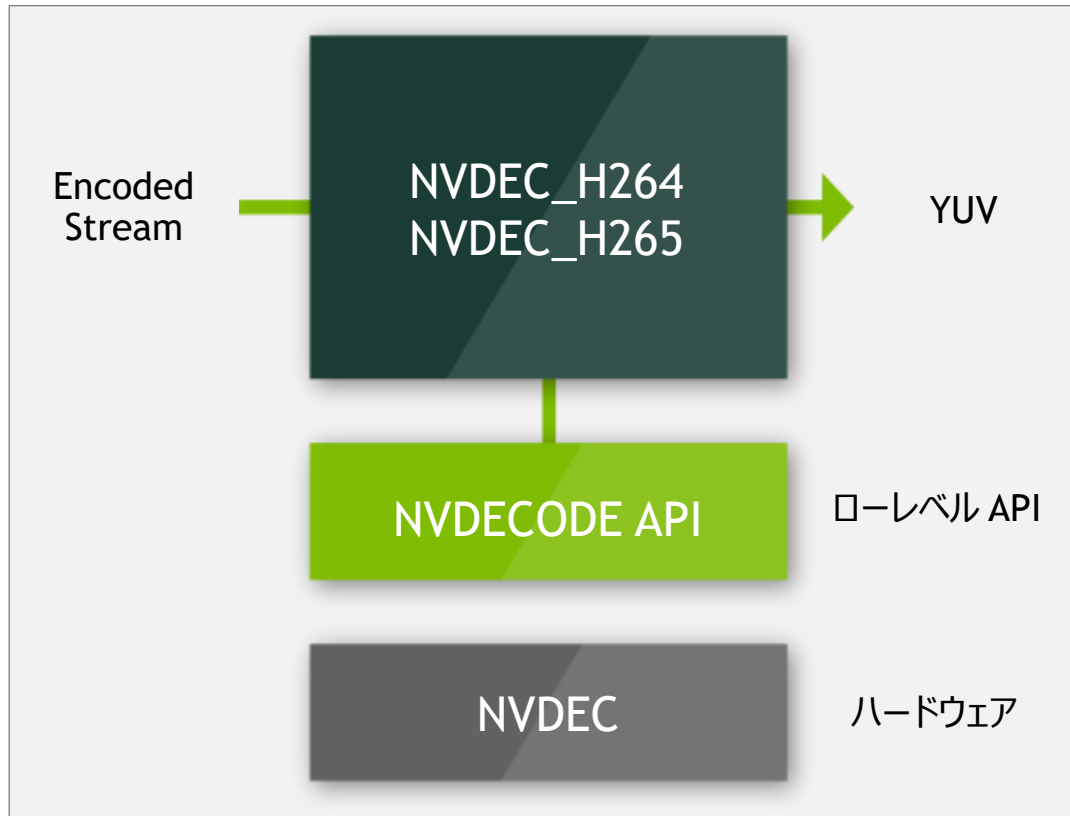
NVIDIA ハードウェア対応プラグイン

DeepStream プラグイン

プラグイン名称	機能
gst-nvvideocodecs	H.265 & H.264 ビデオ デコーダー
gst-nvstreammux	ストリーム・アグリゲーター - マルチプレクサ と バッチ処理
gst-nvinfer	物体検出、及び 識別 の為の TensorRT ベースの推論
gst-nvtracker	リファレンス KLT トラッカーの実装
gst-nvosd	バウンディング・ボックス や オーバーレイ・テキスト表示 の為の OSD API
gst-tiler	複数のソース映像を 1画面に表示する タイリング表示機能
gst-eglglessink	NVIDIA ハードウェア対応 X11 / EGL ベースのレンダラー プラグイン
gst-nvvidconv	映像の スケーリング、フォーマット変換、ローテーション

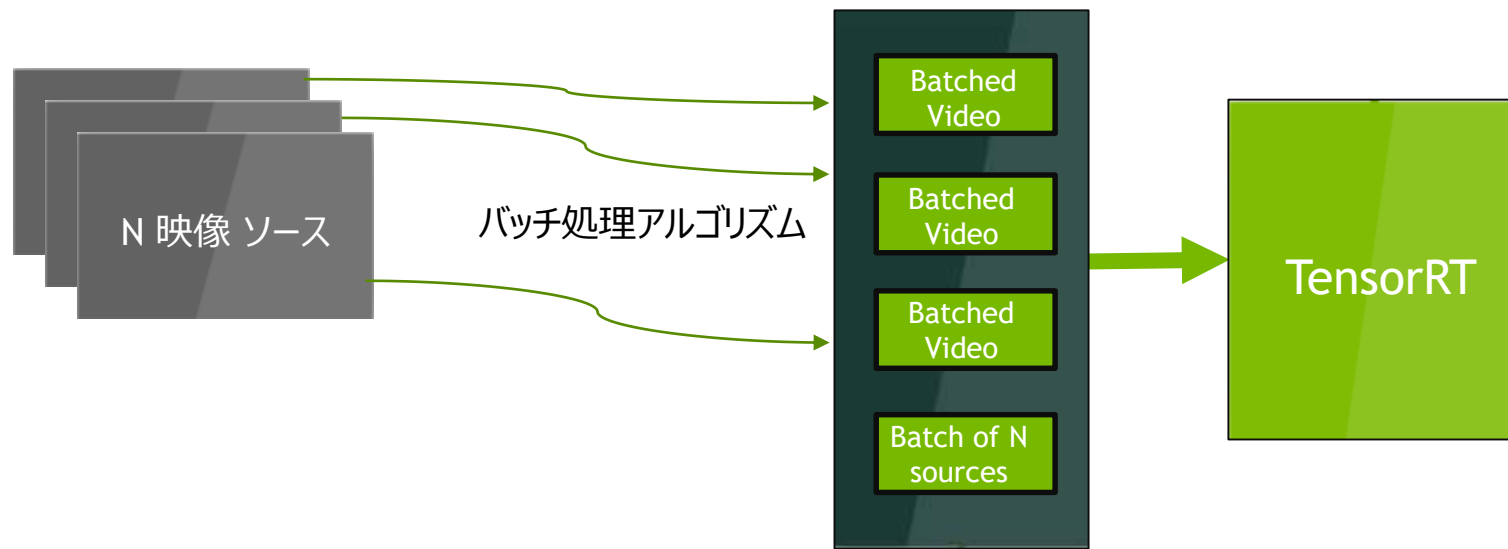
ビデオ デコーダー プラグイン

プラグイン名 - `nvdec_h264`, `nvdec_H265`



- ▶ マルチストリーム、並列デコードをサポート
- ▶ NVDECODE API (旧 NVCUVID API) を使用
 - ▶ H.264
 - ▶ H.265
- ▶ ビット深度 - 各 HW プラットフォームのデコード依存
- ▶ 解像度 - 各 HW プラットフォームのデコード依存
- ▶ YUVデータ プラグイン との互換性
 - ▶ Nvinfer
 - ▶ Nvvidconv

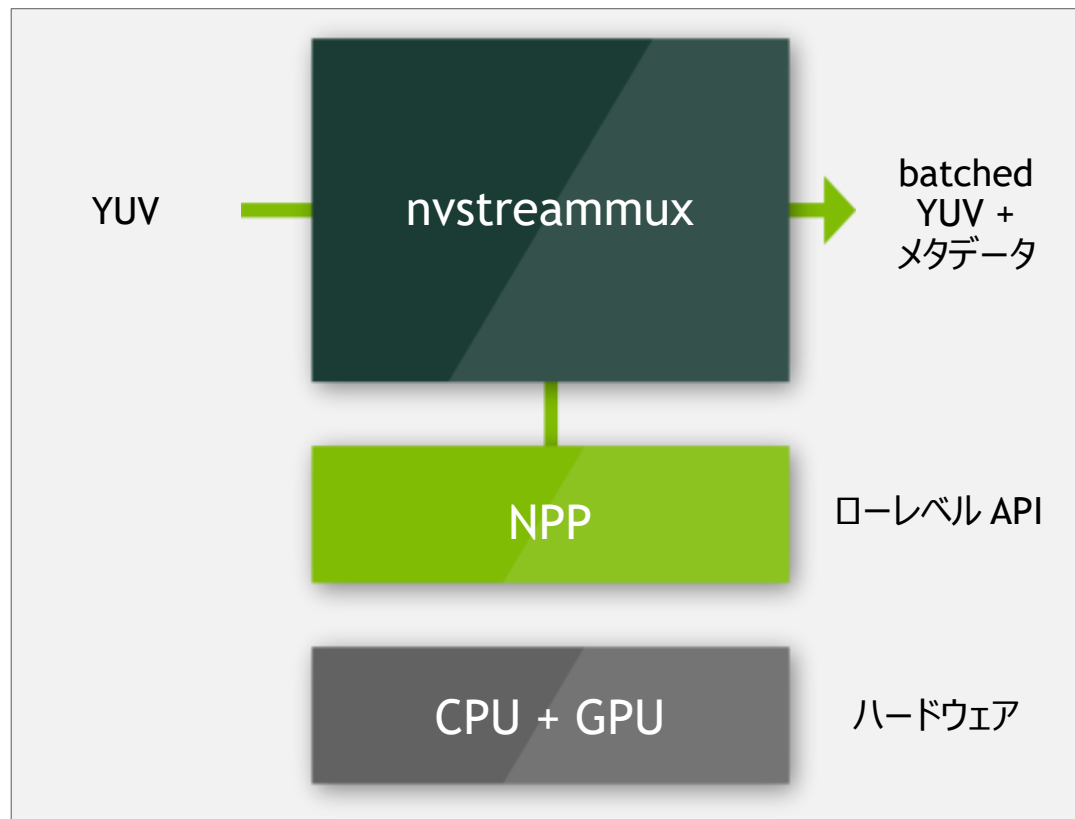
マルチストリーム バッチ処理



- バッチ入力に最適化された TensorRT
- 複数ソースをまとめてバッチ処理作成
- 複数ソースのバッファを区別するためのメタデータの添付

ビデオ アグリゲーター プラグイン

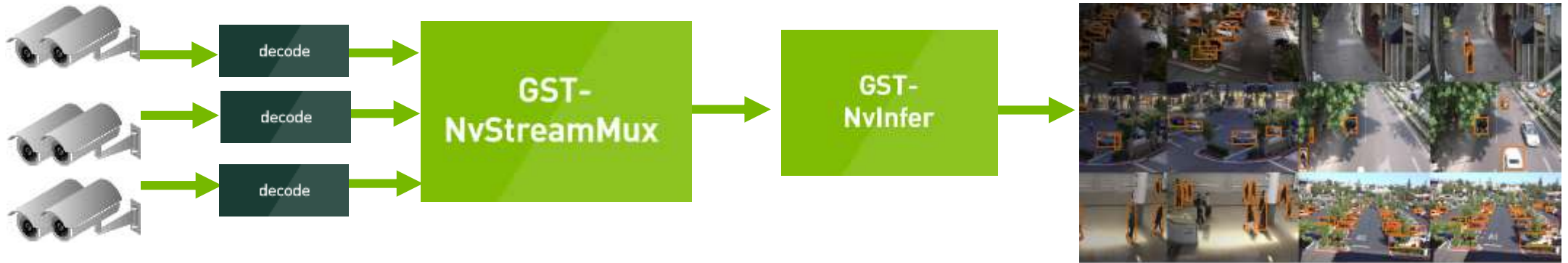
プラグイン名 - nvstreammux



- ▶ N個のストリーム を 連続したバッチフレーム へ変換
- ▶ スケーリング サポート - DLネットワークモデル解像度と異なる解像度の映像入力 をコンバート

モチベーション

スループットの向上による フレームレートとストリーム数 の増加



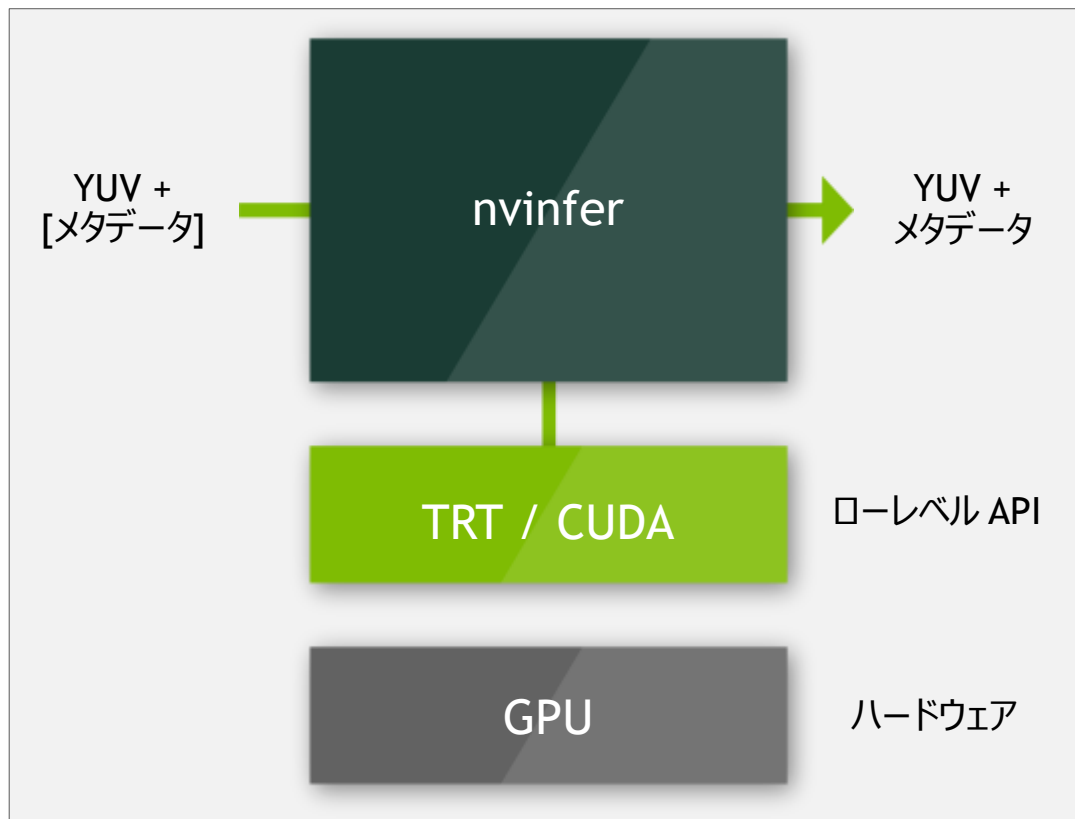
パイプライン遅延の最適化による イベントへの迅速な応答



大規模なパイプラインをサポートすることで、より洗練された分析（カスケードネットワーク、トラッキングなど）を可能にする

推論 プラグイン

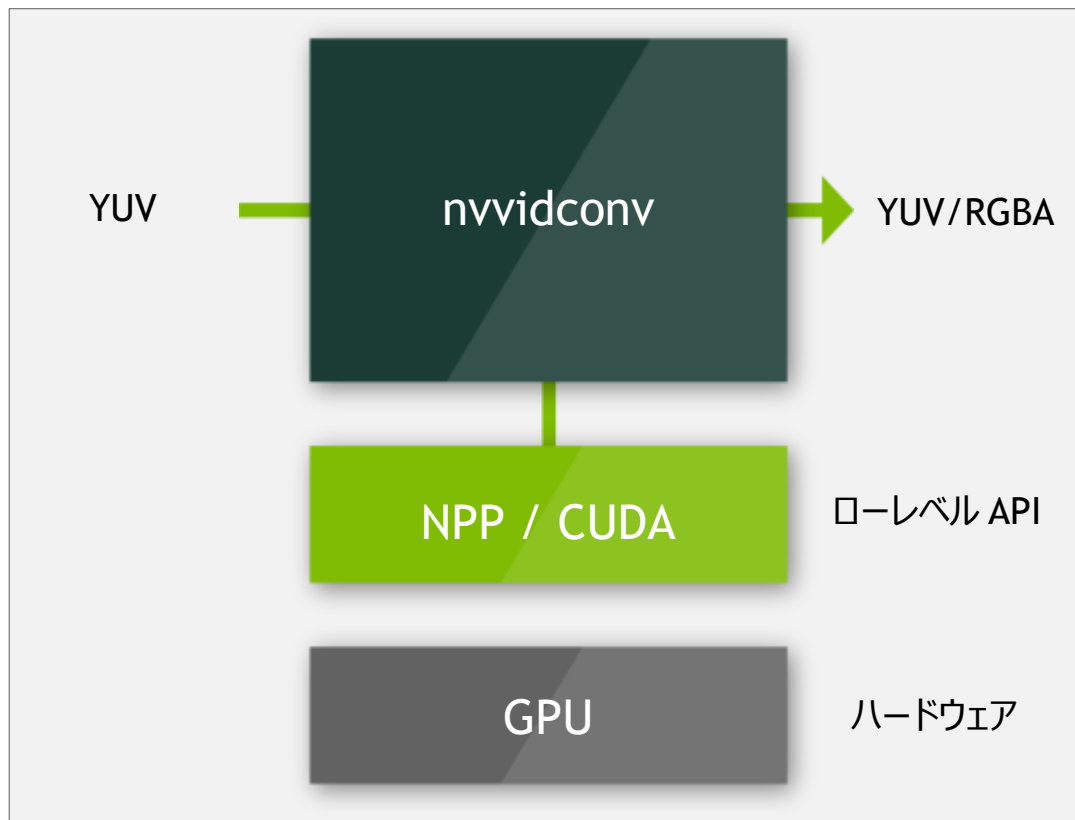
プラグイン名 - nvinfer



- ▶ 物体検出、及び 識別 の為のプラグイン
- ▶ プライマリー と セカンダリー モードをサポート
- ▶ TensorRT をサポート
- ▶ バッチ処理 をサポート
- ▶ カスタム ネットワークモデルの追加をサポート
- ▶ クラスタリングのためのグループ矩形アルゴリズム

フォーマット変換 & スケーリング プラグイン

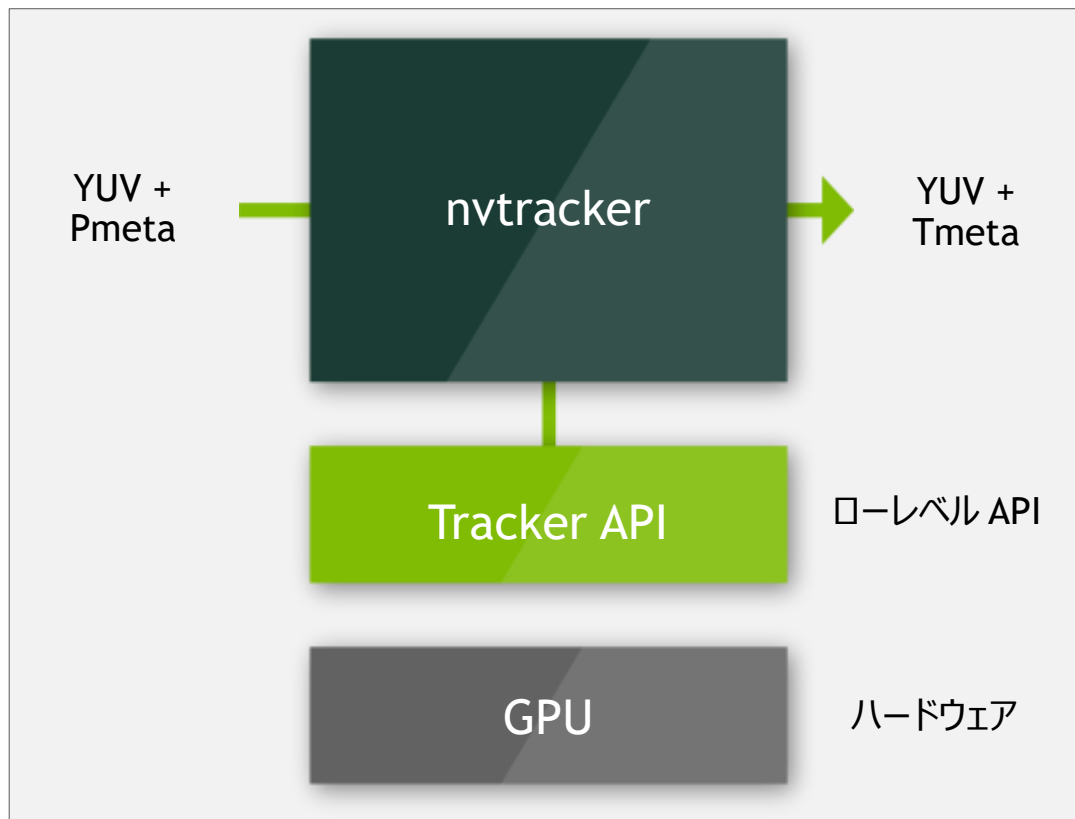
プラグイン名 - **nvvidconv**



- ▶ NPP を使用 (NV performance primitives)
- ▶ フォーマット変換
 - ▶ YUV > RGBA
 - ▶ YUV > BRGA
- ▶ 解像度スケーリング
- ▶ イメージの回転

オブジェクト トラッカー プラグイン

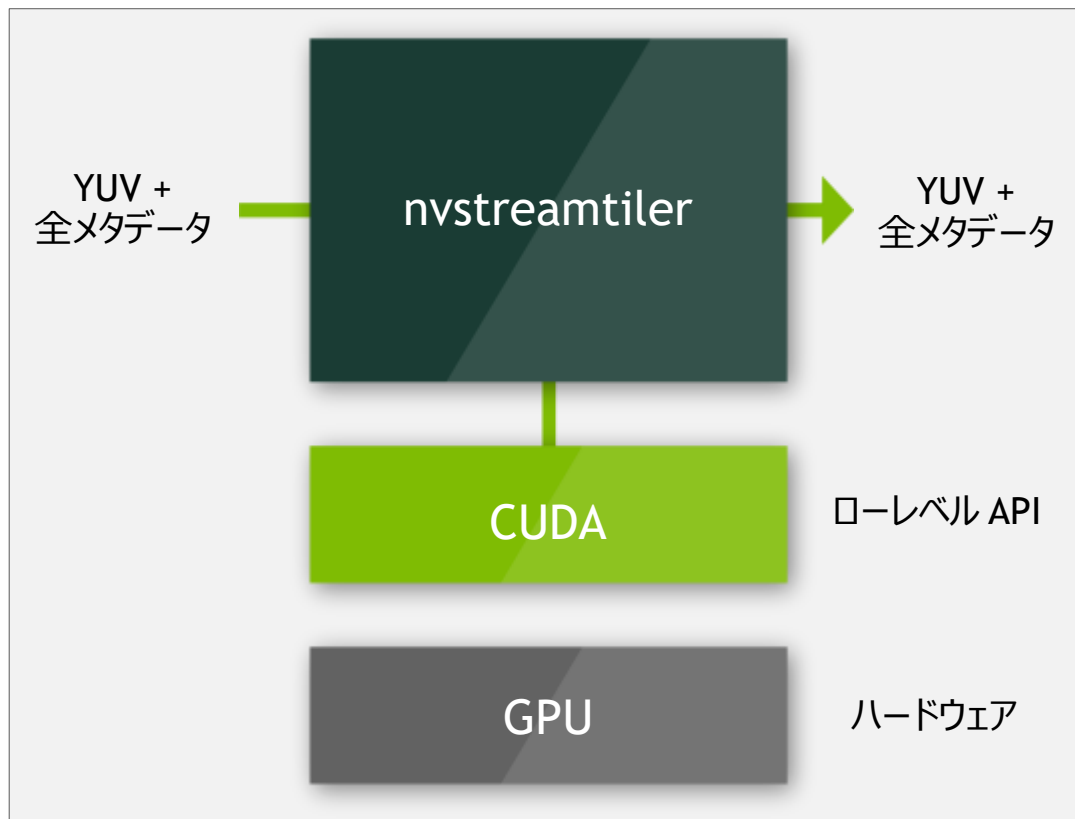
プラグイン名 - nvtracker



- ▶ KLT リファレンスの実装
- ▶ スケーリングとフォーマット変換に NPP / CUDA カーネルを使用
- ▶ 他のより進んだトラッカーにアップグレード可能

タイリング表示 プラグイン

プラグイン名 - nvstreamtiler

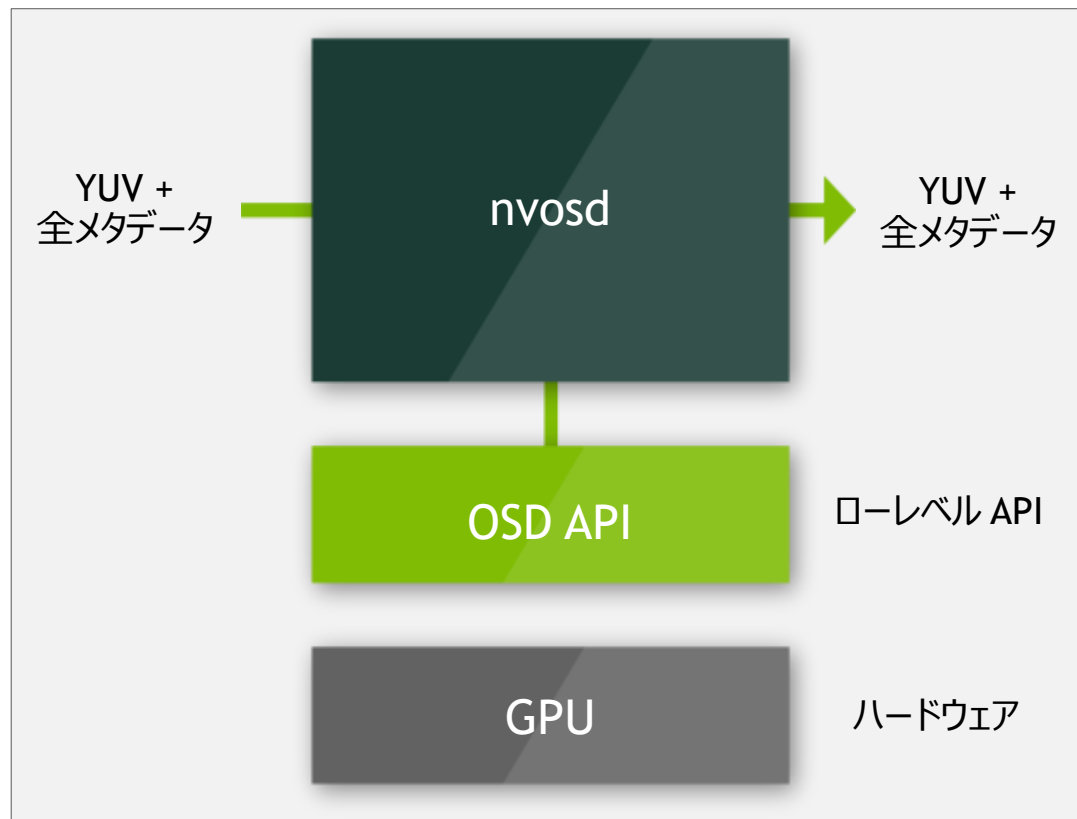


- ▶ ビデオ映像効果 の為のプラグイン
- ▶ 複数の映像ソースを 1画面に表示
- ▶ ウィンドウサイズは 設定可能



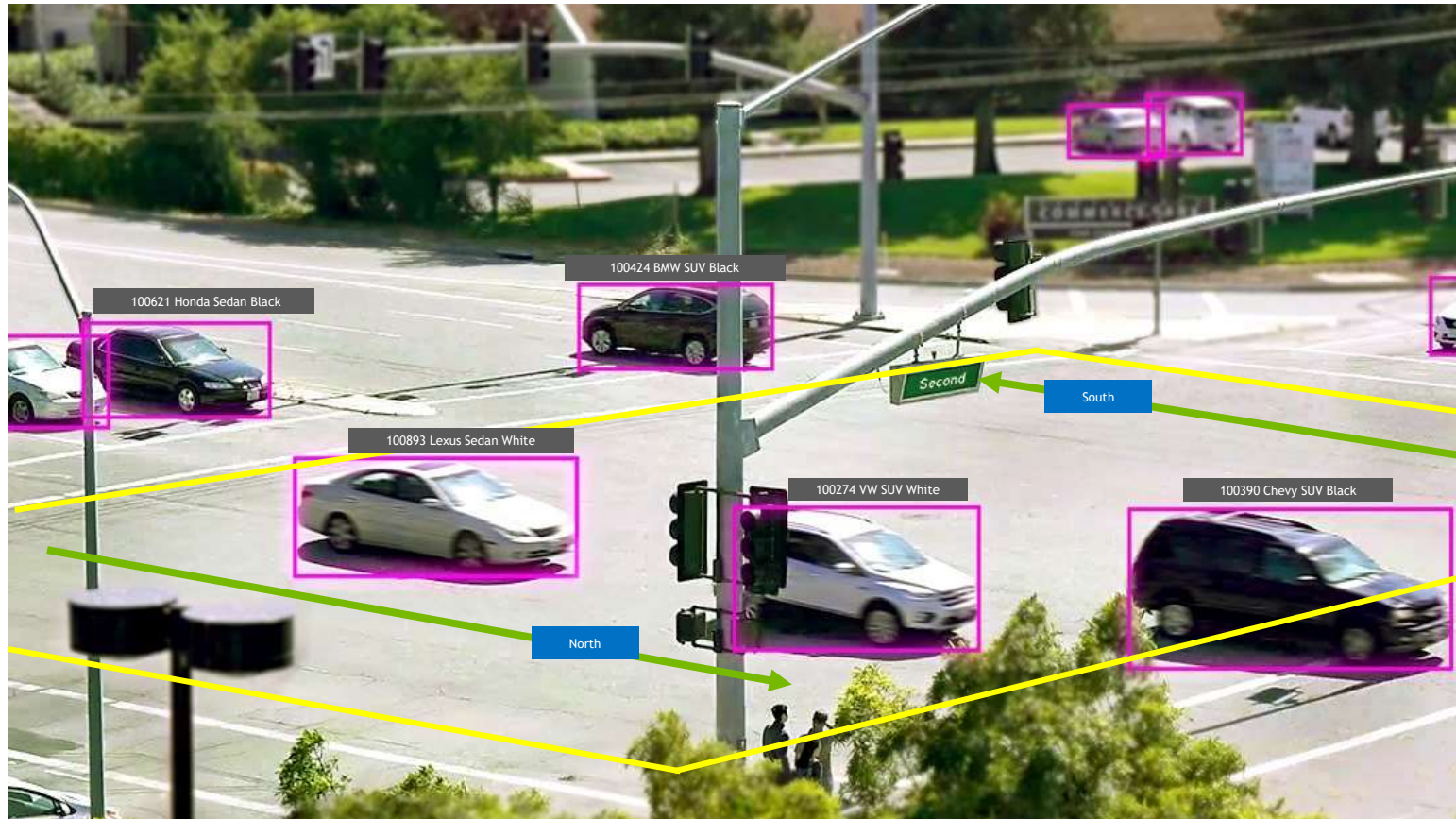
OSD プラグイン

プラグイン名 - nvosd



- ▶ 色付きの線と α ブレンド付きバウンディング・ボックス
- ▶ テキスト、ラベル表示
- ▶ 矢印、線、円、ROIエリア表示

OSD の例



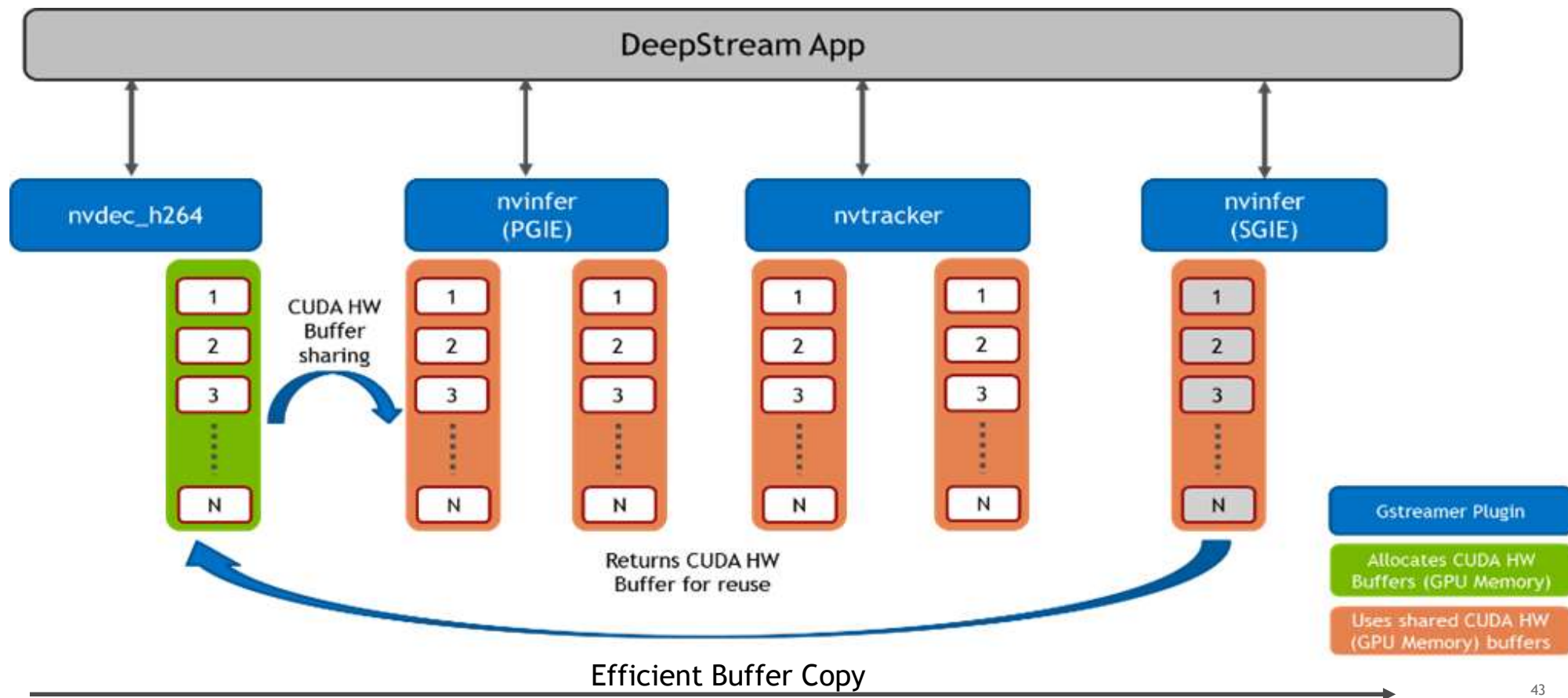
... そして さらに多くの プラグイン

プラグイン名称	機能
filesrc	ファイルの任意の位置からの読み込み
rtspsrc	RTSP プロトコルによるネットワークからのデータ受信
v4l2src	Video4Linux2 デバイスからの読み込み
xvimagesink	X11 ベースのビデオシンク
x264Enc	H264 エンコーダー
jpegdec/enc	JPEG イメージの エンコード/デコードを伴う 読み込み/書き出し
Dewarp	魚眼レンズビデオ
ALPR	3rd パーティー IP プラグイン （自動車のナンバープレート認識）

メモリ管理

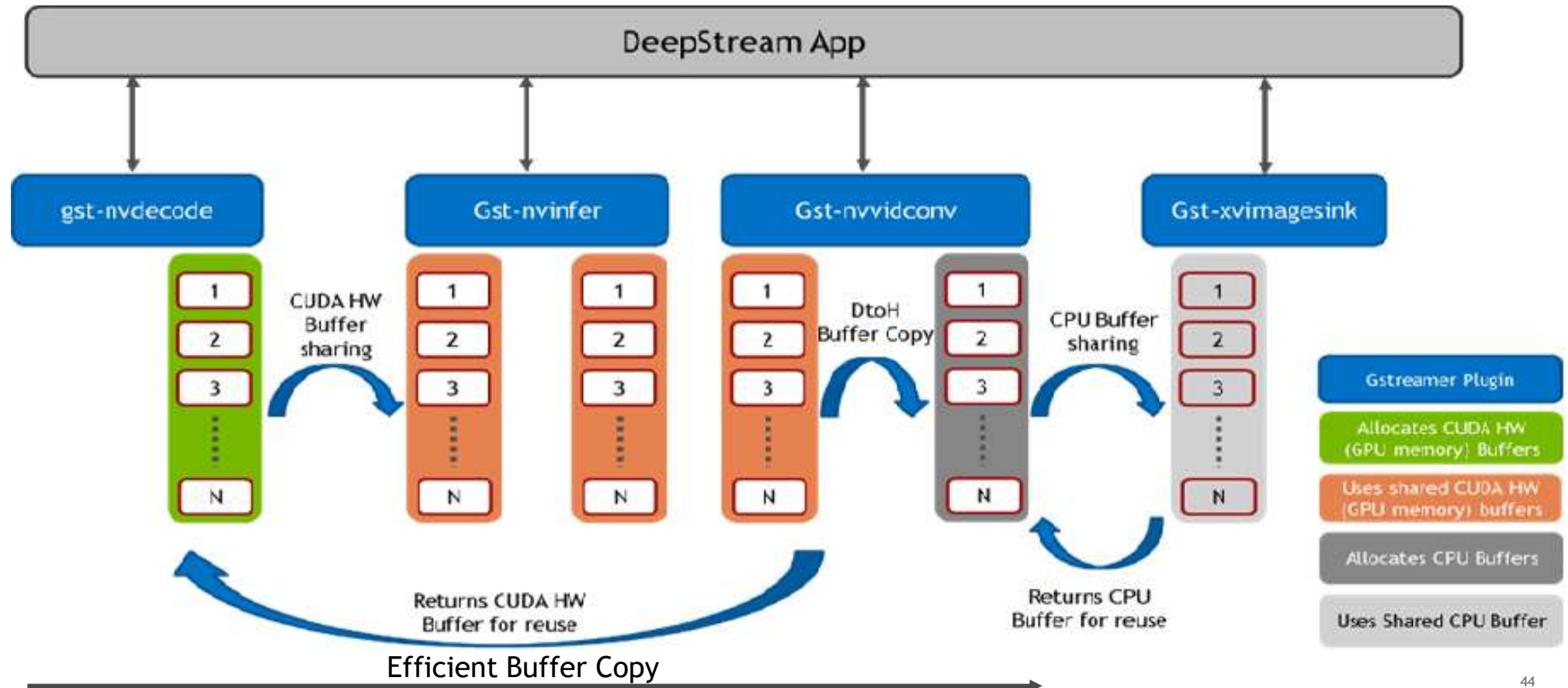
メモリ管理

効率的なメモリ管理



メモリ管理

GPU から CPU へのコピー



パフォーマンス分析

システムパフォーマンス計測のための KPI



1. スループット (fps) = 単位時間内に 処理されたフレームの数



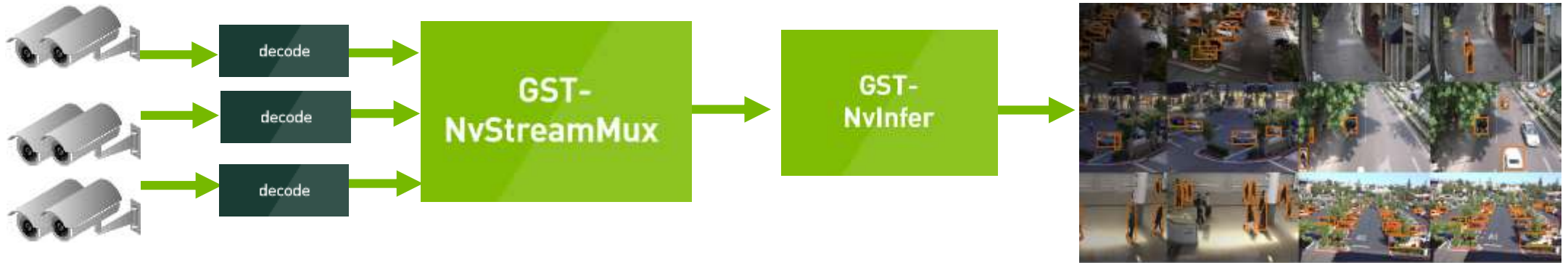
2. 遅延時間

3. ハードウェア利用率 (デコーダー, SM, メモリ, PCI-e バス)

4. 電力

モチベーション

スループットの向上による フレームレートとストリーム数 の増加



パイプライン遅延の最適化による イベントへの迅速な応答



大規模なパイプラインをサポートすることで、より洗練された分析（カスケードネットワーク、トラッキングなど）を可能にする

性能分析のための方法

トップ-ダウン アプローチ

- KPI を測定し、ギャップ（例えば、スループット、ストリーム数）を特定する
- nvidia-smi 等のツールを使いボトルネックを確認する
- gst-logs 等で遅延時間の測定し、フレームレートの上限を確認する
- カーネル実行プロファイリング（nsight、nvvpを使用）でさらに詳細な分析結果を入手する

スループットの計測

gst プロブはバッファがパッドを通過するとき呼び出される プロブ（コールバック）を登録する

```
gst_pad_add_probe (display_sink_pad, GST_PAD_PROBE_TYPE_BUFFER,  
    display_sink_probe, u_data, NULL);
```



プロブはフレームがパイプラインを通過するときの一時的な情報を格納する

プロブは様々なスループットに関連するメトリック（ストリーム毎の平均、最大、最小）を維持する

GST-LOGS を使った遅延時間測定

デコーダー遅延時間: 0:00:05.170122161 - 0:00:05.137672361 = 33ms (0:00:00.032449800)

```
0:00:05.137621066 <capsfilter1:sink> calling chainfunction &gst_base_transform_chain with buffer buffer:
0x7f9c02bc00, pts 0:00:35.719066666, dts 0:00:35.719033333, dur 0:00:00.016683333, size 56047, offset
172403525, offset_end none, flags 0x2400
```

```
0:00:05.137672361 <omxh264dec-omxh264dec0:sink> calling chainfunction &gst_video_decoder_chain with
buffer buffer: 0x7f9c02bc00, pts 0:00:35.719066666, dts 0:00:35.719033333, dur 0:00:00.016683333, size
56047, offset 172403525, offset_end none, flags 0x2400
```

```
0:00:05.170122161 <src_0:proxypad3> calling chainfunction &gst_proxy_pad_chain_default with buffer
buffer: 0x7f714c5020, pts 0:00:35.719066666, dts 99:99:99.999999999, dur 0:00:00.016683333, size 808,
offset none, offset_end none, flags 0x0
```

```
0:00:05.170196720 <nvvconv0:sink> calling chainfunction &gst_base_transform_chain with buffer buffer:
0x7f714c5020, pts 0:00:35.719066666, dts 99:99:99.999999999, dur 0:00:00.016683333, size 808, offset none,
offset_end none, flags 0x0
```

パフォーマンス向上のためのベストプラクティス

- 推論精度の最適化 (INT8/FP16)
- 推論時のバッチサイズの最適化
- 適切なフレームレートの映像入力を使用する
- システムメモリとデバイスメモリ間のデータ移動を最適化
- 最大限の並列性を得るために CUDA ストリームを使用する

developer.nvidia.com/deepstream-sdk

<http://developer.nvidia.com/deepstream-sdk>



The screenshot shows the NVIDIA Developer website for the DeepStream SDK. The header includes navigation links for RTX, BENCHMARKS, DESIGNWORKS, VIOWORKS, COMPUTEWORKS, JETPACK, DRIVE, and CLARA. The main banner features the text "NVIDIA DeepStream SDK" and "Accelerate your video analytics application development." Below this, a paragraph describes the SDK as ideal for developers creating AI-based solutions for video analytics at scale. A diagram illustrates the processing pipeline: COLLECT (camera) -> DECODE (GPU) -> PRE-PROCESS (GPU) -> TRACK INFERENCE (GPU) -> ENCODE (GPU) -> COMPOSITE (GPU) -> VISUALIZE (monitor). The text below the diagram explains that the SDK offers the capability to gain rich insights through a heterogeneous concurrent neural network architecture, allowing developers to leverage multiple neural networks to process each video stream. It also mentions that developers are increasingly turning to AI to process video because it has been shown to deliver significantly better performance than previous approaches for these use cases. Applications of Intelligent Video Analytics (IVA) include understanding consumer behavior in retail, intelligent traffic systems, web content filtering, and ad injection. Finally, it states that DeepStream makes use of:

- NVIDIA® TensorRT™ and NVIDIA CUDA® for AI and other GPU computing tasks
- Video CODEC SDK and multimedia APIs for accelerated encode and decode
- Imaging APIs for capture and processing
- A graph-based architecture and modular plug-ins to create configurable processing pipelines



The screenshot shows the NVIDIA DeepStream SDK download page. The header includes the text "DEEPSTREAM SDK". Below this, a diagram illustrates the SDK architecture, showing the flow from "VIDEO APPLICATIONS" to "DEEPSTREAM SDK" and then to "VIDEO OUTPUT". The "DEEPSTREAM SDK" section is divided into three columns: "Included" (listing 2D/3D Inference/TensorRT Plugins, OpenCL Acceleration Plugins, VideoFrame Capture and Processing Plugins, and 3D Pose Estimation Plugins), "FLYING & DOWNSIDE" (listing 2D/3D Inference/TensorRT Plugins, OpenCL Acceleration Plugins, VideoFrame Capture and Processing Plugins, and 3D Pose Estimation Plugins), and "DEEPSTREAM TOOLKIT" (listing SDK for Real-time Applications, App Acceleration/Performance Tools, Plugin Templates and Integration Guides, and Profiling and Performance Tuning). Below the diagram, there are four buttons: "DOWNLOAD", "MULTIMEDIA API/VIDEO CODEC SDK", "JETSON", and "DETAILED DESCRIPTION". The "JETSON" button is highlighted. Below the buttons, there are two sections: "Resources" and "Download for Jetson". The "Resources" section includes links for "Introductory Blog's" and "On-demand Webinar". The "Download for Jetson" section includes a button labeled "Download for Jetson".

DeepStream also comes with sample code and pre-trained deep learning models as examples to perform classification and object detection on video streams. DeepStream is available for NVIDIA [Tesla](#) and [Jetson](#). Download the DeepStream SDK today to get started.

[Download for Tesla](#) [Download for Jetson](#)

[Introductory Blog's](#) [On-demand Webinar](#)

NVIDIA DEVELOPER

COMPUTEWORKS
BENCHMARKS
JETPACK
DESIGNWORKS
DRIVE

Copyright © 2016 NVIDIA Corporation | [Legal Information](#) | [Privacy Policy](#)