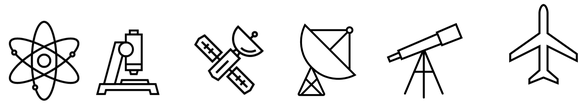
HPC at the Edge: Enabling Real Time Streaming Sensor Analytics

Adam Thompson | adamt@nvidia.com | CLSAC ‘22

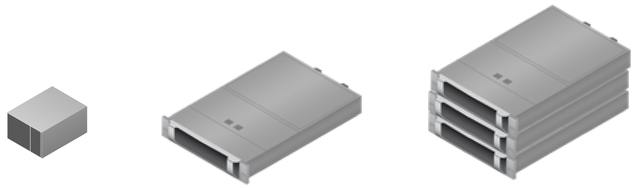
Defining the Edge

Bringing Compute Closer to the Sensor for Real Time Insights and Control

Cameras, Environment Sensors, Microscopes, Light Source, Telescope, Satellites



Data Collection, Aggregation, Reduction Filtering, Analytics, Distribution



EDGE

Form Factor used close/near sensor depends on compute needs and space
Limitation: ARM, ARM+GPU, x86+GPU

Designed for pre-defined set of function to meet response latency, space, power and form factor constraints

Designed for 1000s of jobs, 1000s of users, 1000s job-hours, Batch processing

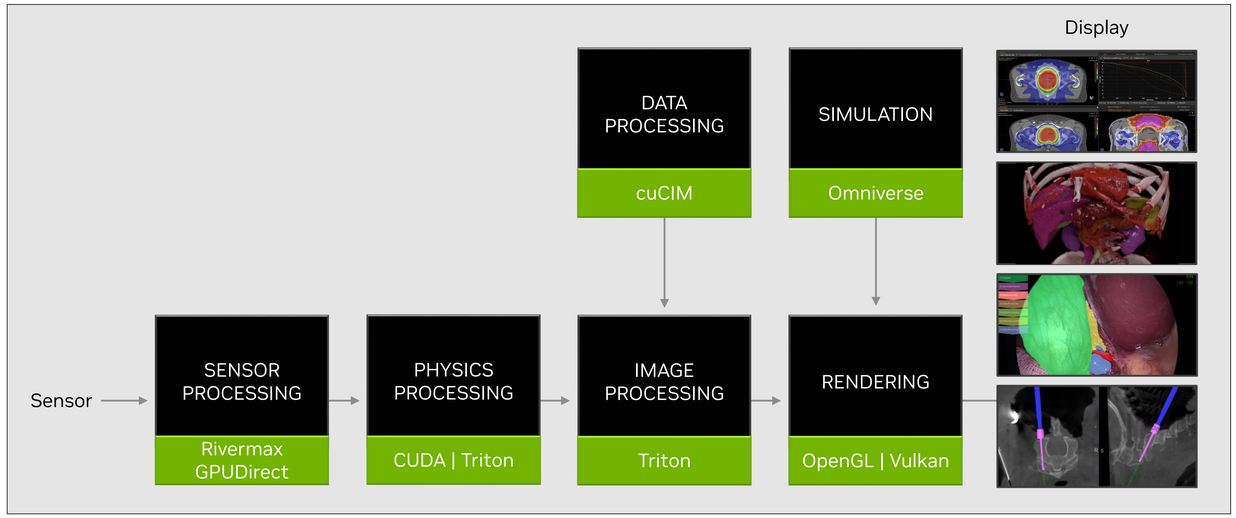
Simulation, Training, Big Data Analytics

DATACENTER

NVIDIA Clara Holoscan Platform

AI Computing Platform for Medical Devices

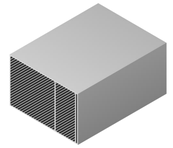
- Optimized for Streaming AI
  - Accelerated AI and HPC Libraries
  - Rivermax for GPUDirect RDMA Data Streaming
  - Pre-trained Models, Sample Applications (C++, Python)
- Safety, Security and Manageability Built In
  - Safety Extension Package
  - Functional Safety Island
  - sMCU
- Built for Medical Certification (IEC 60601, 62304)
- Long life Hardware & Long-term Software Support
- Rich sensor partner ecosystem



C++ & Python APIs | Pre-Trained Models | Sample Applications

Accelerated AI & HPC Libraries, Rivermax

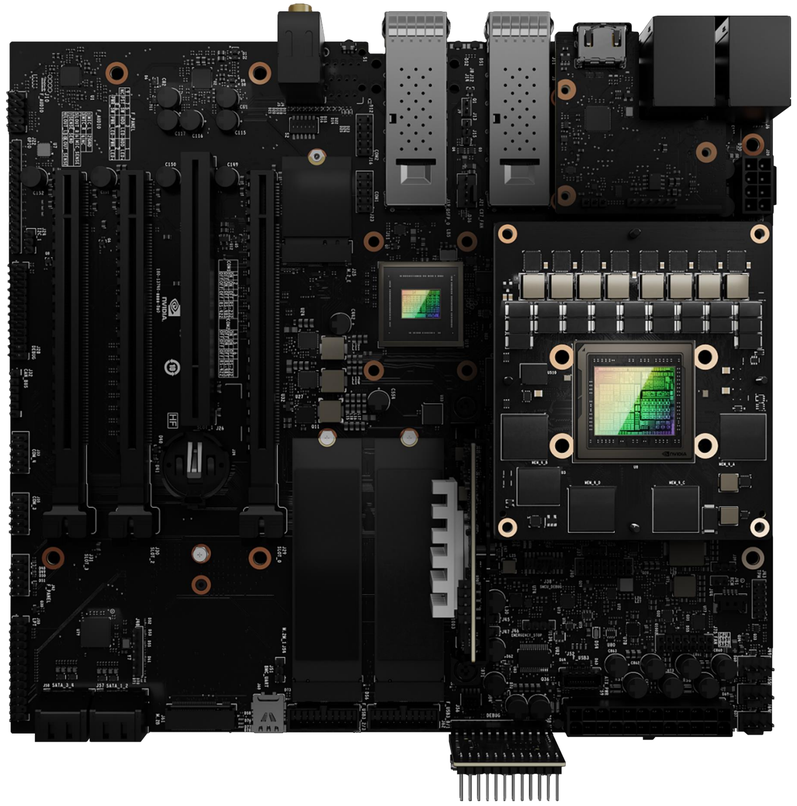
Commercial OS | Drivers | Safety Extension Package



IGX
Orin, ConnectX-7, A6000
Functional Safety Island, Safety MCU

NVIDIA IGX Orin

- Full platform including industrial-grade hardware and software with long term commercial support
- Secure by design with encrypted memory, IP protection from CPU to GPU, security engines for key management
- Enable functional safety with Orin SoC Safety Extensions, Orin Safety Island and dedicated safety microcontroller unit (sMCU)



Orin
12-Core Arm
64GB
275 TOPS

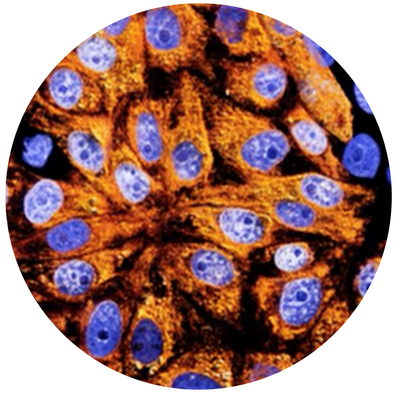
CX7
200GbE
Rivermax
RDMA

RTX
RT Coers
Tensor Cores
CUDA Cores

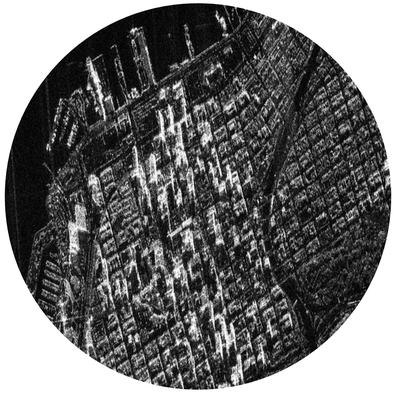
Safety
Safety Extension Package
Orin Safety Island
Safety MCU

Edge Sensors For Every Domain

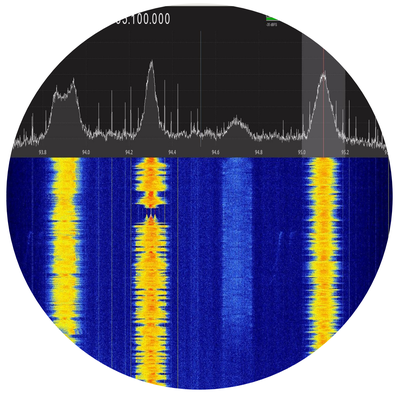
Delivering Streaming, Real Time, and Low Latency Data Insights from Space to Underseas



HPC Experiments

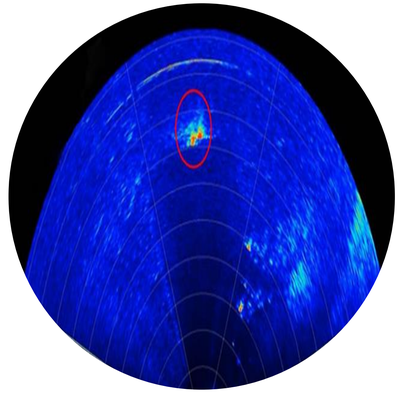


Remote Sensing



RF Signal Processing

Computer Vision



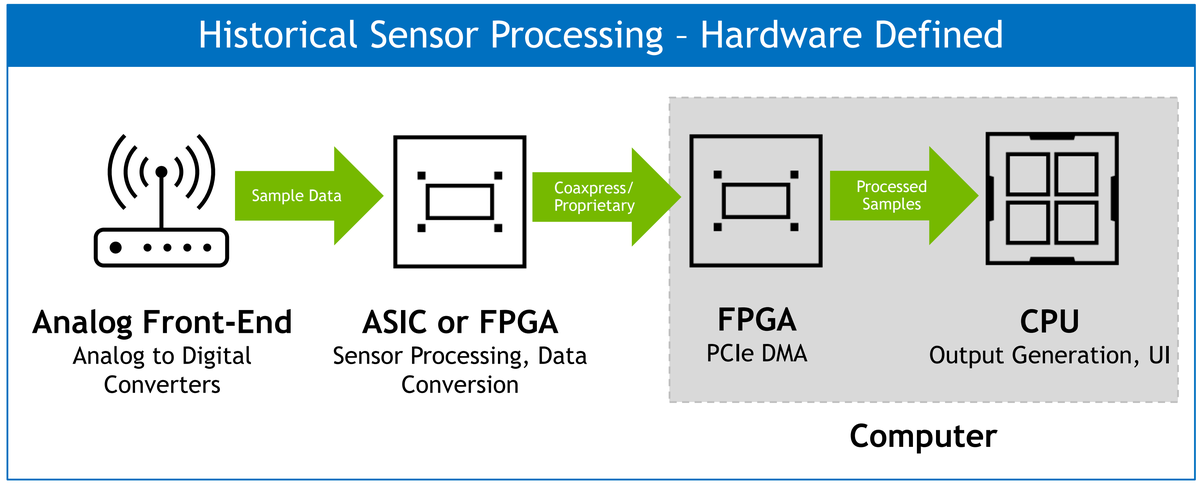
Sonar / Acoustics

Multiple Sensor Data Types
Line Rate Processing Requirements
Scalable Compute at the Cloud, Edge, and Data Center
Combining HPC and AI to Fuel New Data Insights

One Platform

Generic Sensor Processing Architectures

Connecting GPU Compute to Front End Sensors and Increasing Developer Productivity



FPGA is challenging and time-consuming to program, adapt to new processing workflows, and reconfigure

Difficult to scale with new collection requirements

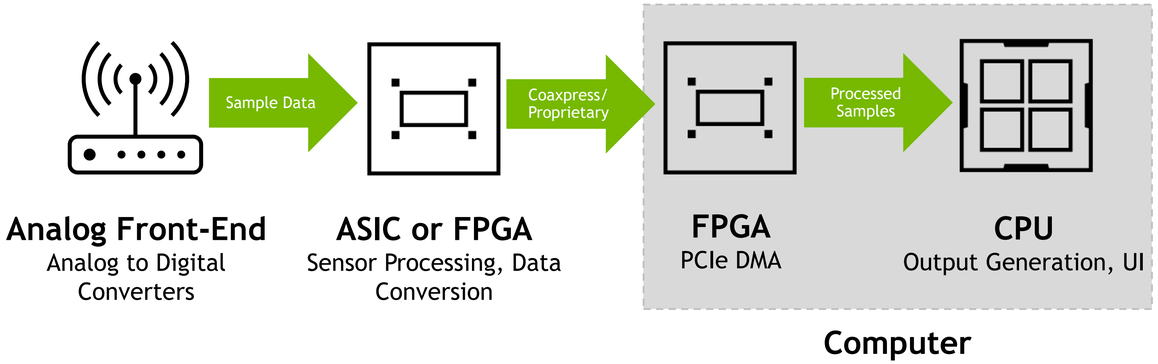
Increasing sensor requirements are exceeding power and memory available in FPGA architectures

FPGA IP Core licensing and software costs inflate overall cost of the solution

Generic Sensor Processing Architectures

Connecting GPU Compute to Front End Sensors and Increasing Developer Productivity

Historical Sensor Processing – Hardware Defined



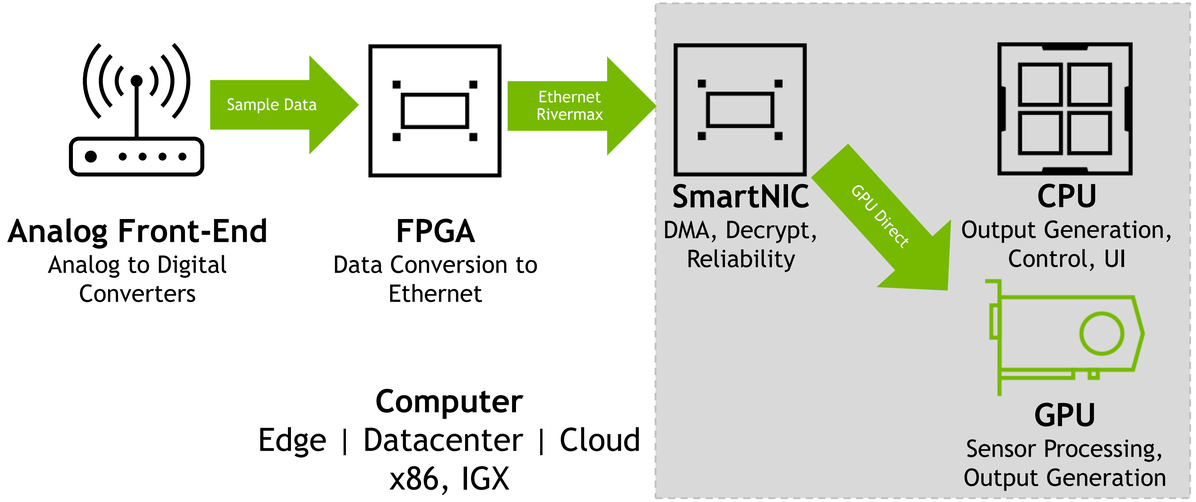
FPGA is challenging and time-consuming to program, adapt to new processing workflows, and reconfigure

Difficult to scale with new collection requirements

Increasing sensor requirements are exceeding power and memory available in FPGA architectures

FPGA IP Core licensing and software costs inflate overall cost of the solution

Modern Sensor Processing – Software Defined and GPU Accelerated



Variety of software to translate sensor data to Ethernet on FPGA

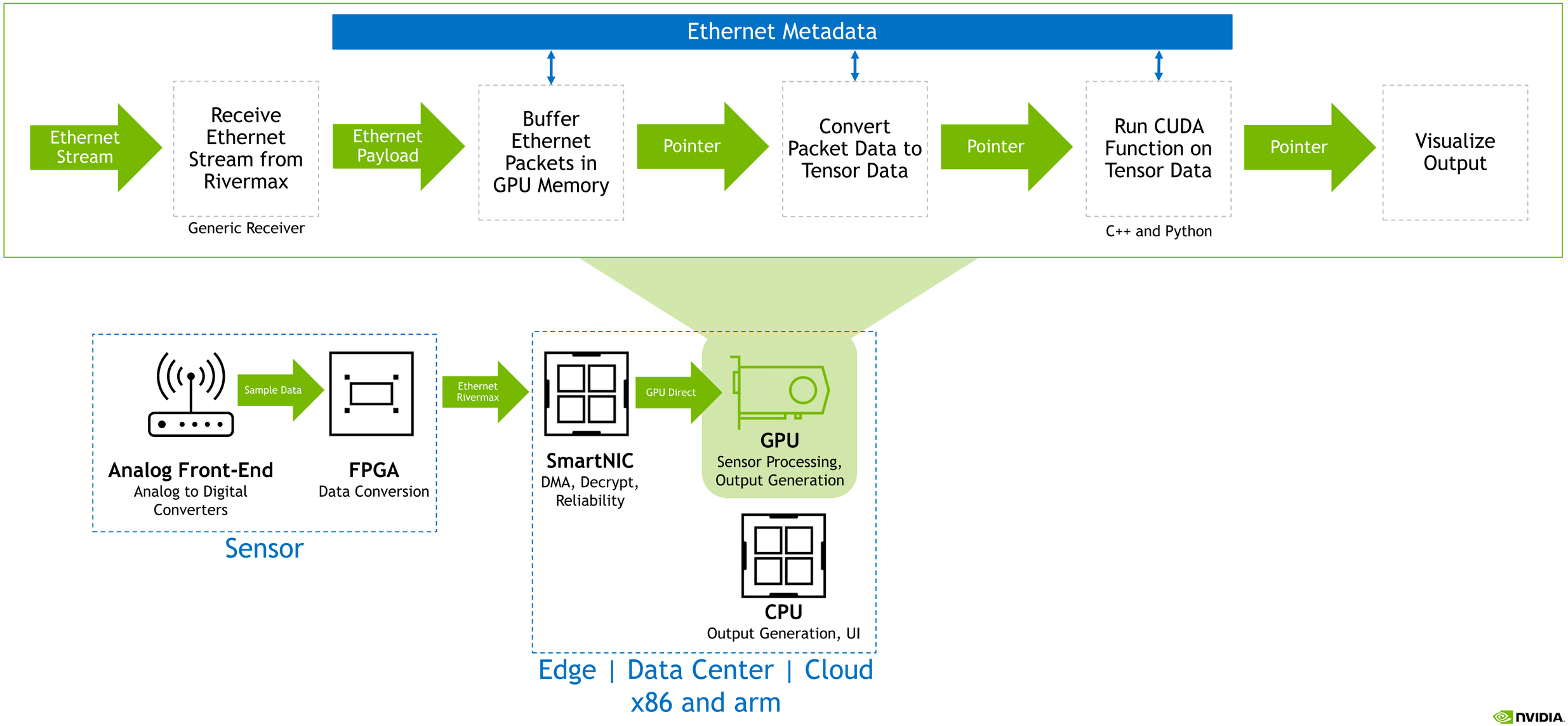
Scales as collection requirements increase and supports Cloud Native architecture

Supports a variety of GPU platforms, both at the Edge and in the Datacenter; Matched PCIe roadmap

DMA from NIC to GPU comes for free with Rivermax

Generic Holoscan Sensor Processing Workflow

Collect and Process Streaming Sensors Anywhere



Streaming Sensor Deployment Challenges at Scale

Bridging the Gap Between Experiments and Production

Data Scientists and Researchers

Focused on developing ML/DL models or accelerated-computing codes to advance the state-of-the-art

Typically more comfortable in higher-level languages like Python, MATLAB

Performance on streaming data isn't typically optimized and isn't designed for scale

DevOps Engineers

Focused on application deployments, scaling, and health to meet performance and scale objectives

Typically fluent in higher level languages like Python but also comfortable with C/C++

Streaming data implementation needs to easily scale to meet diverse performance requirements

Performance Engineers

Focused on software engineering to optimize speed-of-light performance in both compute and networking

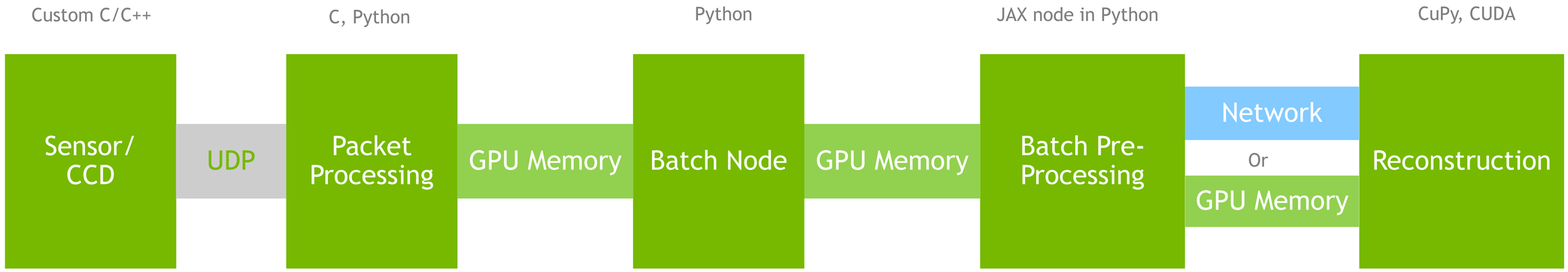
Typically experts in C/C++ with the ability to explore and debug hardware-level instructions

Performance on streaming data isn't typically optimized for developer ease-of use

Standardize on Streaming Data Framework that Enables Developer Productivity and Speed-of-Light Performance for Every User

A Reference Streaming Application

X-Ray Detector



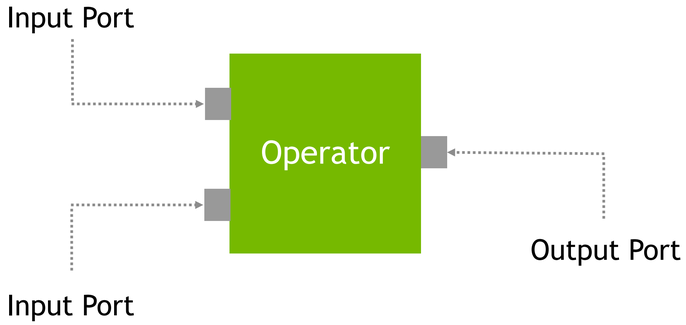
Many AI and HPC applications can be represented as pipelines of sequential compute stages with data or message movement between each compute node

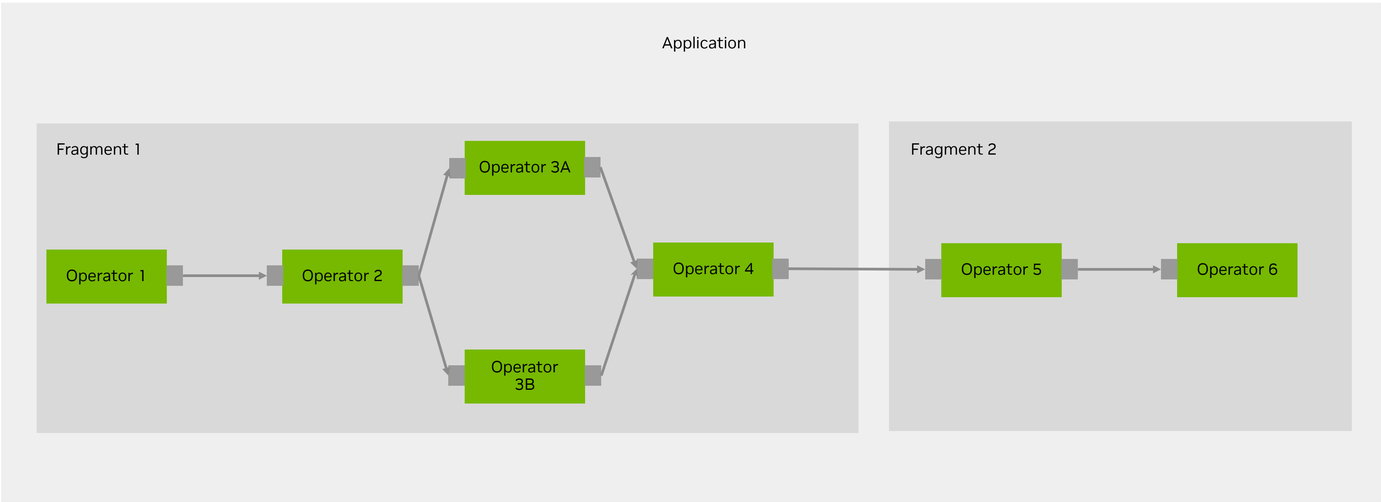
To speed up development, abstracting away data movement between nodes allows the developer to focus on the work being done at each compute step rather than moving data

Production deployments are simplified if this data framework allows for language/application agnostic compute nodes (regardless of locality) and flexible, network-aware message/data transactions between these nodes

Holoscan C++ APIs

Architecture For Building Composable and Scalable Streaming Pipelines





Operator is the most basic unit of work. It receives streaming data at an input port, processes it, and publishes it to one of its output ports.

Port is an interaction point between two operators. Operators ingest data at Input ports and publish data at Output ports.

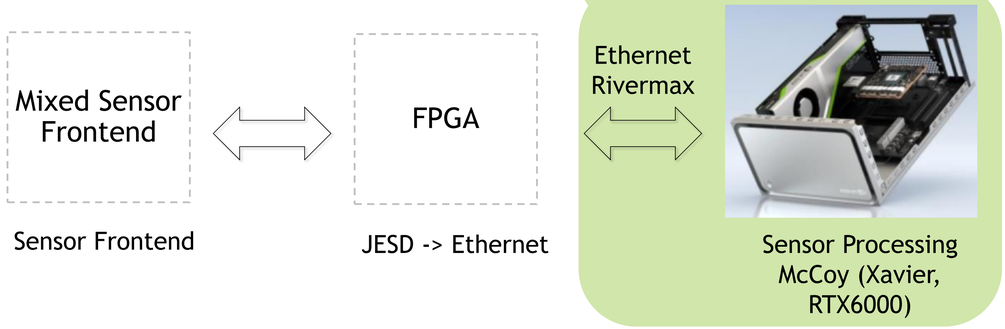
Fragment is a Directed Acyclic Graph (DAG) of operators. It can be assigned to a physical node of a Holoscan cluster during execution. The run-time execution manages communication across fragments. In a Fragment, Operators (Graph Nodes) are connected to each other by flows (Graph Edges).

Application acquires and processes streaming data. It's a collection of fragments where each fragment can be allocated to execute on a physical node of a Holoscan cluster.

5G Instrumentation Example with Holoscan

Holoscan Software Pipeline

Hardware Configuration



Software Enablement

Real Time Performance < 10ms

Algorithm Development MATLAB / Python (NumPy) - CPU	15,475 ms
Library Change, No Algorithm Optimization Python (CuPy, cuSignal) – RTX6000 GPU	686 ms
Algorithm Optimizations C++ (MatX, CUDA) – RTX6000 GPU	**2.4 ms**

30x

300x

Take-Aways and Final Thoughts

Summarizing Streaming Edge Processing on Holoscan

- Holoscan is a software framework that provides the building blocks to build streaming edge AI applications
- Holoscan will be sensor agnostic, allowing a plug and play approach to sensor processing
- C++ and Python Holoscan APIs are targeted to be released in December of 2022 – Consider "EA" release
- Currently soliciting feedback / suggestions and gathering additional requirements


NVIDIA