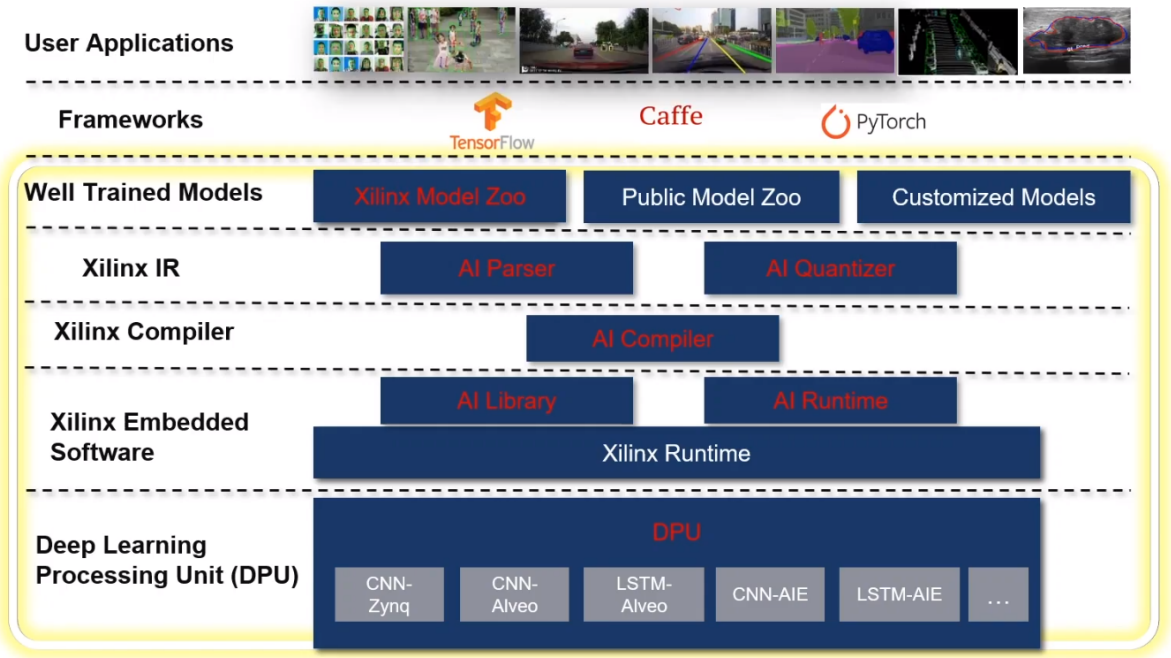
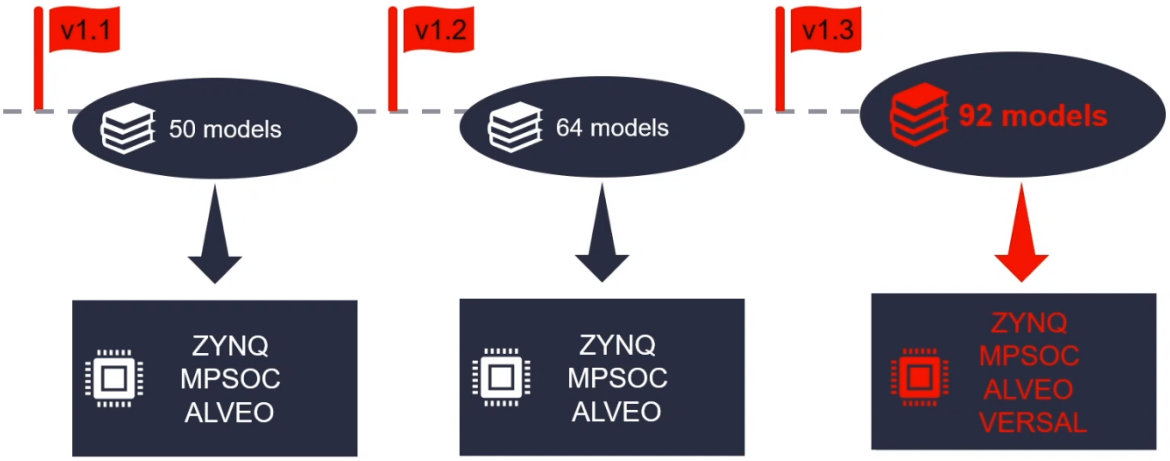
**Vitis AI: Unified AI Inference Solution Stack**




**Model Zoo**

- Provides state-of-art models.
- Yaml file for each model
- Link for different overlays
- Readable from AI Library



**AI Parser & Quantizer: Workflow**

In Vitis AI, the AI parser and quantizer are key components of the workflow for deploying and optimizing deep learning models on Xilinx FPGA devices. Here is an overview of the workflow:

1. Model Conversion: The workflow typically begins with converting a trained deep learning model from popular frameworks such as TensorFlow, PyTorch, or Caffe into an intermediate representation format supported by Vitis AI, such as ONNX (Open Neural Network Exchange) or Caffe prototxt.

2. AI Quantizer: The AI quantizer is used to quantize the model to lower precision, typically from floating-point (FP32) to fixed-point (INT8) or reduced-precision floating-point (FP16). Quantization reduces the memory footprint and computational requirements of the model, making it more suitable for deployment on FPGA devices. The AI quantizer performs calibration and quantization-aware training techniques to minimize the loss of accuracy caused by quantization.

3. AI Compiler: Once the model is quantized, the AI compiler is used to compile the model for the target FPGA device. The AI compiler performs various optimizations, such as layer fusion, memory optimization, and kernel customization, to improve the performance and efficiency of the model on the FPGA.

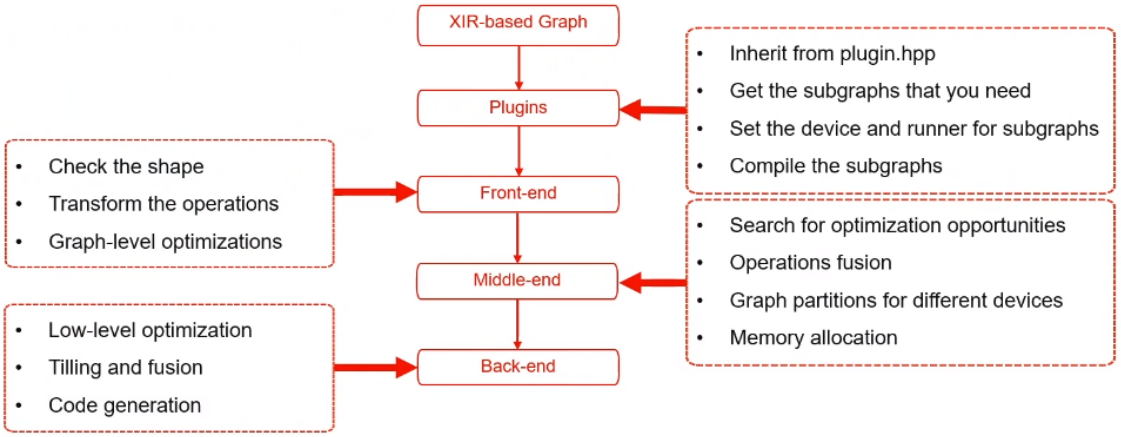
4. AI Library: The AI library provides a collection of pre-optimized FPGA-accelerated functions, known as AI kernels, for common deep learning operations. These kernels are specifically designed and optimized for the target FPGA architecture and can be used to accelerate the execution of the quantized model.

5. FPGA Deployment: After compilation, the generated bitstream along with the quantized model and AI kernels are deployed onto the target FPGA device. The FPGA device, with its parallel processing capabilities, allows for efficient execution of the quantized model, providing low-latency and high-throughput performance.

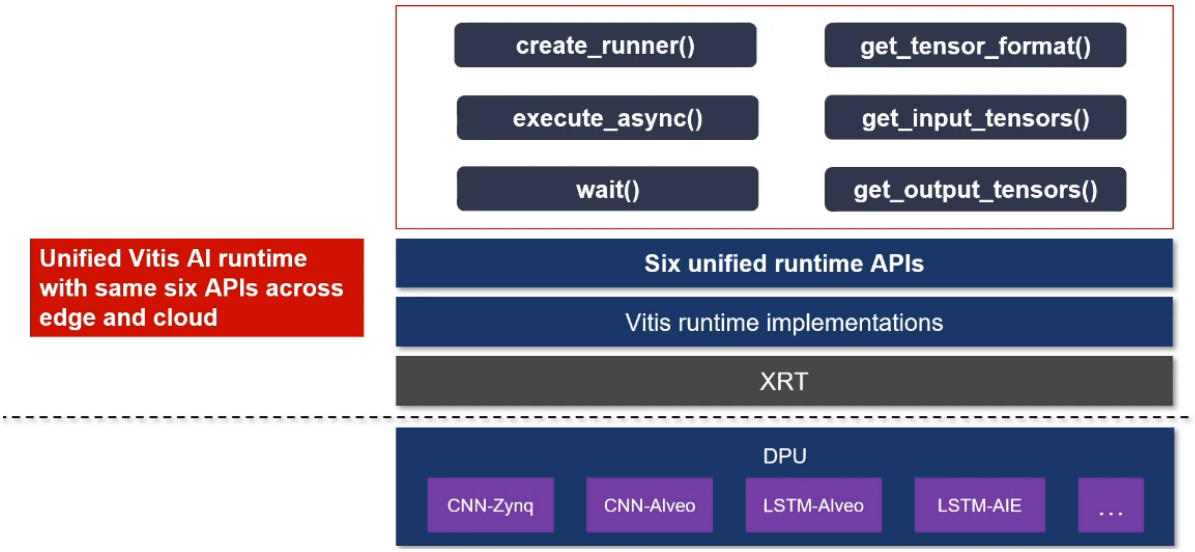
The Vitis AI workflow leverages the Xilinx Vitis unified software platform, which integrates tools, libraries, and runtime environments for FPGA development. It enables developers to optimize deep learning models for FPGA acceleration and deploy them in real-world applications that require low latency and high performance.

Note that the workflow may vary depending on the specific requirements of the application and the target FPGA device.

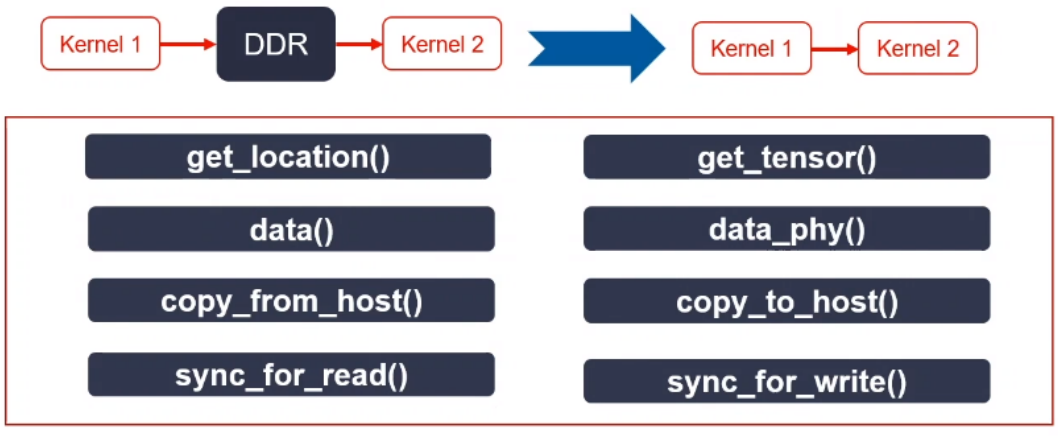
## AI Compiler




## VART: Unified runtime APIs



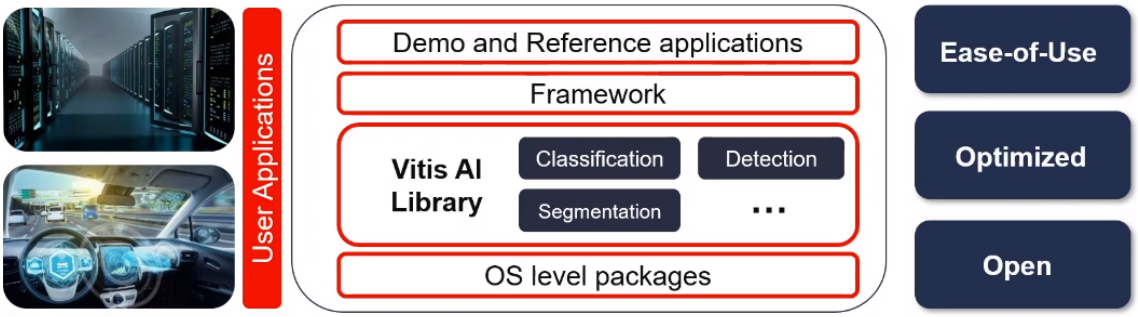

## VART: Zero Copy




New added APIs to achieve zero copy

## Vitis AI Library: the What?

- **Vitis AI Library provides high-level API based libraries across different vision tasks: classification, detection, segmentation and etc.**
  - **Reference applications to help customers' fast prototyping**
  - **Optimized codes used in AI applications and products**

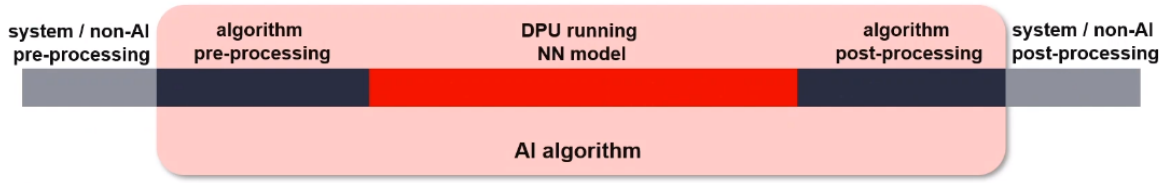



## AI Application General Processing Flow

- **A typical abstraction of processing flow:**


- **Algorithm-level processing**
  - » Data normalization before sending to DPU
  - » Post processing (e.g. bounding boxes decoding in detection)
- **Additional system-level workloads for AI inference**
  - » Color conversion / resizing
  - » Path planning / control / status update

## What Vitis AI Library Provides




- **AI Library offers libraries for**
  - Algorithm-level optimization
  - Open and easy to extend
  - Directly support models in AI Model Zoo

## AI Library Samples

- The Vitis AI Library provides image test samples ,video test samples, performance test samples  for all the above networks. Each sample has the following four kinds of test sample.
  - test_jpeg_[model type]
  - test_video_[model type]
  - test_performance_[model type]
  - test_accuracy_[model type]
- In addition, the kit provides the corresponding performance test program. For video based testing, we recommend to use raw video for evaluation. Because decoding by software libraries on Arm® CPU may have inconsistent decoding time, which may affect the accuracy of evaluation.

## Easy-to-Use APIs to Deploy Full Algorithm