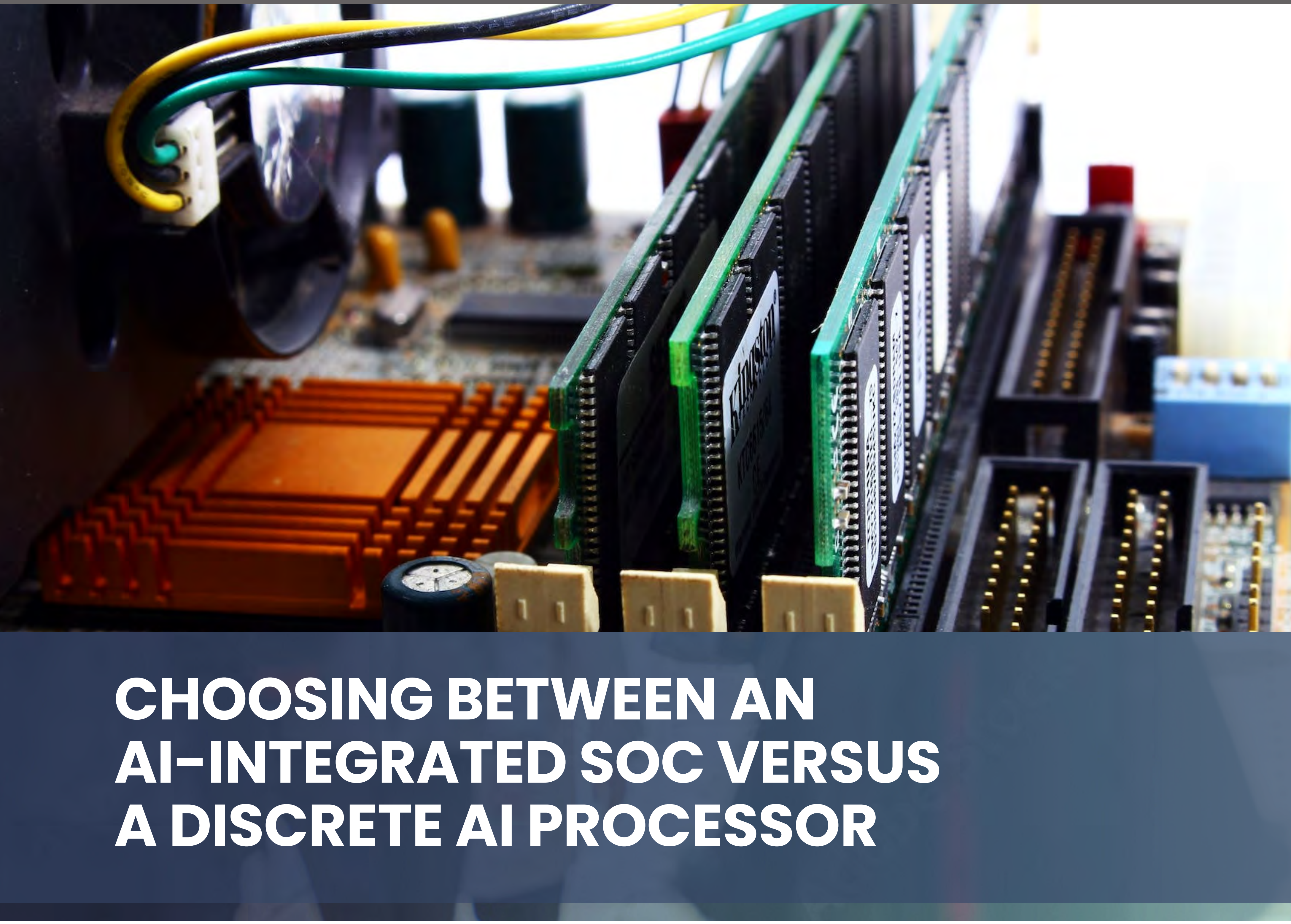
kinara



CHOOSING BETWEEN AN AI-INTEGRATED SOC VERSUS A DISCRETE AI PROCESSOR

WHITE PAPER

Edge AI is being used in such a wide variety of use cases, there is no one-size-fits-all solution. To run AI models in edge devices, system developers can use everything from microcontrollers, CPUs, GPUs, and integrated and discrete AI processors. The right compute unit to run AI models depends on factors such as performance, power, and cost. For example, in many edge AI applications such as smart retail, smart city, and industrial, a high level of performance and accuracy is required. This in turn demands a dedicated AI processor - whether it be discrete or integrated into an SoC.

Furthermore, when selecting an AI processor, you must consider factors such as the overall system design, system utilization, and most importantly compiler support for AI models. When considering the system design, a discrete AI processor will perform better than an AI subsystem integrated into an SoC because the latter's performance will be heavily influenced by system-level overhead and a shared memory bus.

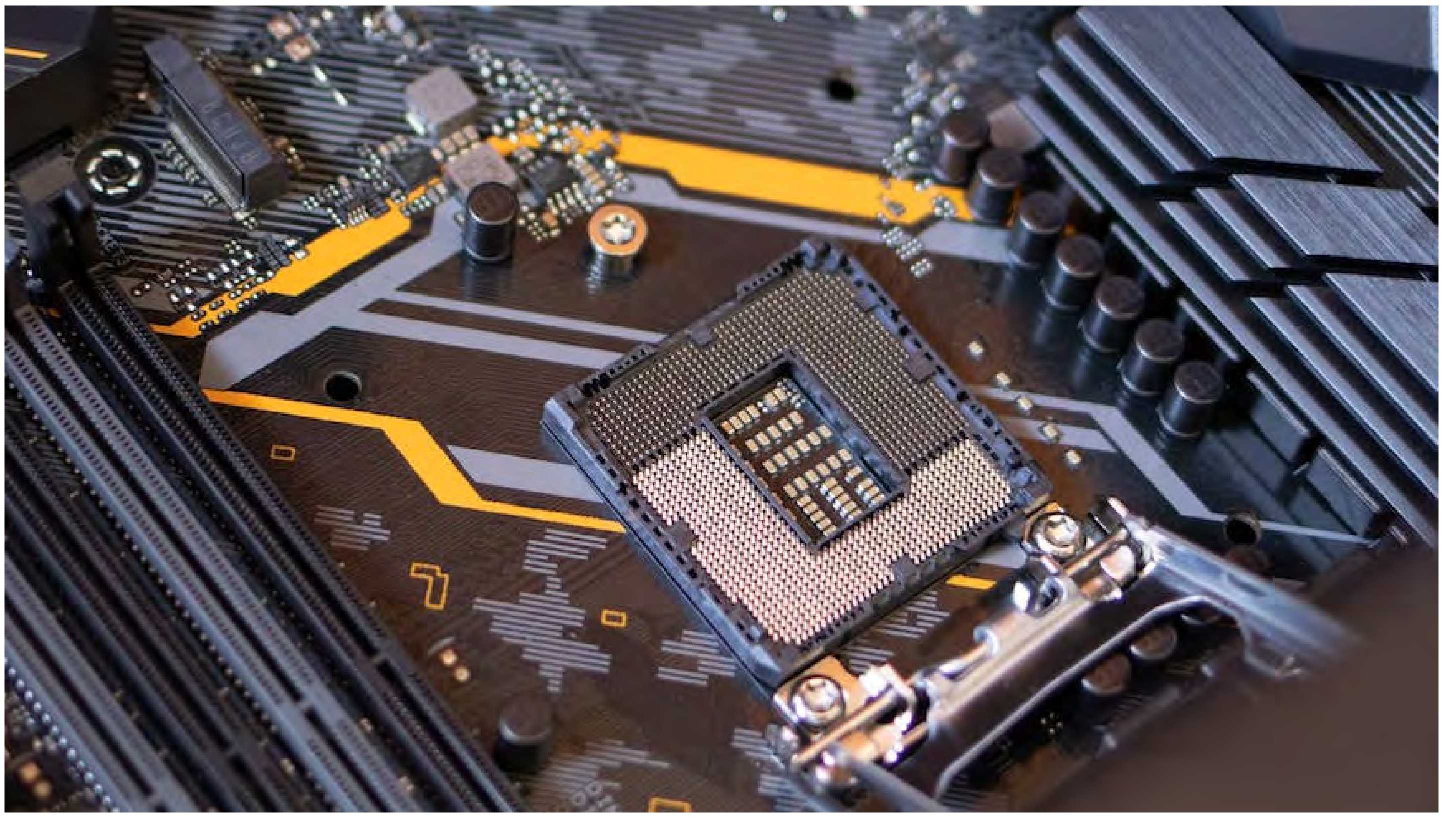
In this white paper, we'll explore the benefits of a discrete processor:

- Augment AI performance while leveraging your existing embedded system design.
- Scale up performance by attaching one or more devices.
- Performance is not limited by system-level overhead.

Don't fix it if it's not broken

Even before there was such a thing as edge AI, in the days where applications were referred to as embedded systems, there had to be (and still is) a main processor to run the main application, an operating system, drivers, and a variety of other functions. Some of these SoCs also include integrated GPUs and other specialized hardware accelerators (e.g., encryption, video codecs). With the addition of computer vision and AI, embedded systems also must include even more functionality for supporting the pre- and post-processing associated with the inference pipeline (Figure 1).

A dedicated AI processor is required for many edge AI applications that demand a high level of performance and accuracy.



These pre-processing functions take in the raw data and perform tasks such as resize and color conversion, before feeding the input into the model running on the AI processor. And on post-processing, the output of the inference is fed into tasks such as non-maximum suppression (NMS) and image blending.

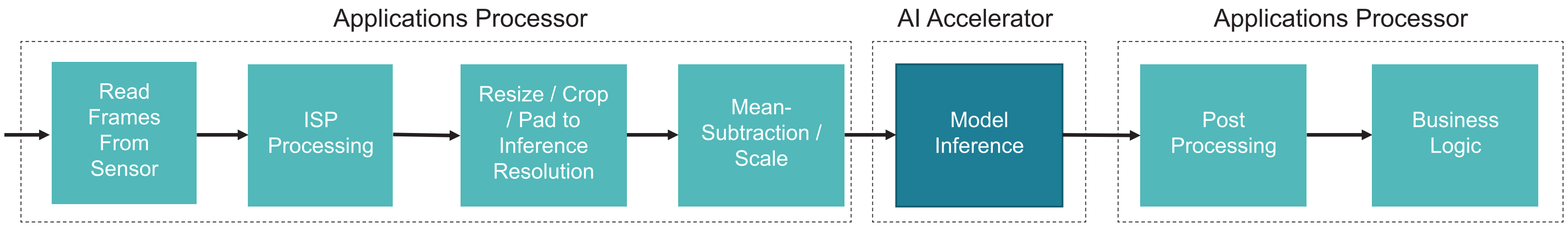


Figure 1. For AI model inferencing, the pre- and post-processing functions are typically performed on an applications processor.

A key takeaway from the preceding paragraph is that putting the software pieces together for an embedded and/or edge AI system is no small feat. Going from the bench to full commercial deployment represents a significant investment in time and resources. Large, embedded processor companies like NXP and Qualcomm recognize the importance of this and have hundreds of engineers in their organizations to support their software infrastructures (e.g., for Linux, drivers, libraries). Therefore, unless you're building an embedded system from scratch, switching to an entirely new SoC, whether in the same product family or from another vendor (especially one that is new to the market with a relatively small software team) should be avoided. This is especially true if your main goal in switching is to supplement the platform's AI capability, which can be accomplished more easily and safely by adding a discrete AI processor, like Kinara's Ara-1.

From a hardware perspective, the Ara-1 can be attached via PCIe or USB to a host SoC (common examples are the NXP i.MX 8M Plus and Qualcomm QCS610 application processors). From a software perspective, if you've already taken advantage of the GStreamer or OpenCV support or C++ or Python direct API support provided by an SoC vendor for the inference pipeline, you can easily plug in the Ara-1. Furthermore, if more AI performance is required, you can add in multiple Ara-1 devices and utilize Kinara's load balancing software to distribute the inference load across devices.

Kinara's load balancing software automatically distributes the inference load across multiple Ara-1 devices

Be sure to consider system-level overhead

In a smart camera or edge appliance, the integrated SoC (i.e., host processor) acquires the video frames and performs pre-processing steps to transform this sensor data into a format that matches the input requirements of the AI inferencing performed on the AI processor (integrated or discrete). Specifically, the host processor performs normalization, mean-subtraction, scale-type activities (e.g., resize, crop), and other functions. These functions can be done using either the CPU cores or the GPU, but they can also be performed by hardware processors if they are available on the SoC (e.g., image signal processor).

After these pre-processing steps are completed, the AI subsystem that is integrated into the SoC can then directly access this quantized input from system memory, or in the case of a discrete AI processor, the input is then delivered for inference over USB or PCIe interface. Following the inference, any post-processing must run. An example of post processing is non-maximal suppression (NMS) logic used to interpret the output of most object detection models. Post processing is typically performed on the SoC, though in some cases it can also be offloaded to the AI processor (i.e., Ara-1). In either case, the post-processed results are then available to the application to implement the desired business logic.

Kinara Ara-1

As we've alluded to, an integrated SoC can contain a range of computation units, including CPUs, GPUs, AI subsystem, vision processors, video encoders/ decoders, image signal processor (ISP), and more. Most of these computation units have one thing in common – they all share the same memory bus and consequently access to the same memory. Furthermore, the CPU and GPU might also have to play a role in the inference and these units will be busy with other tasks in a fully running system. This is what we mean by system-level overhead.

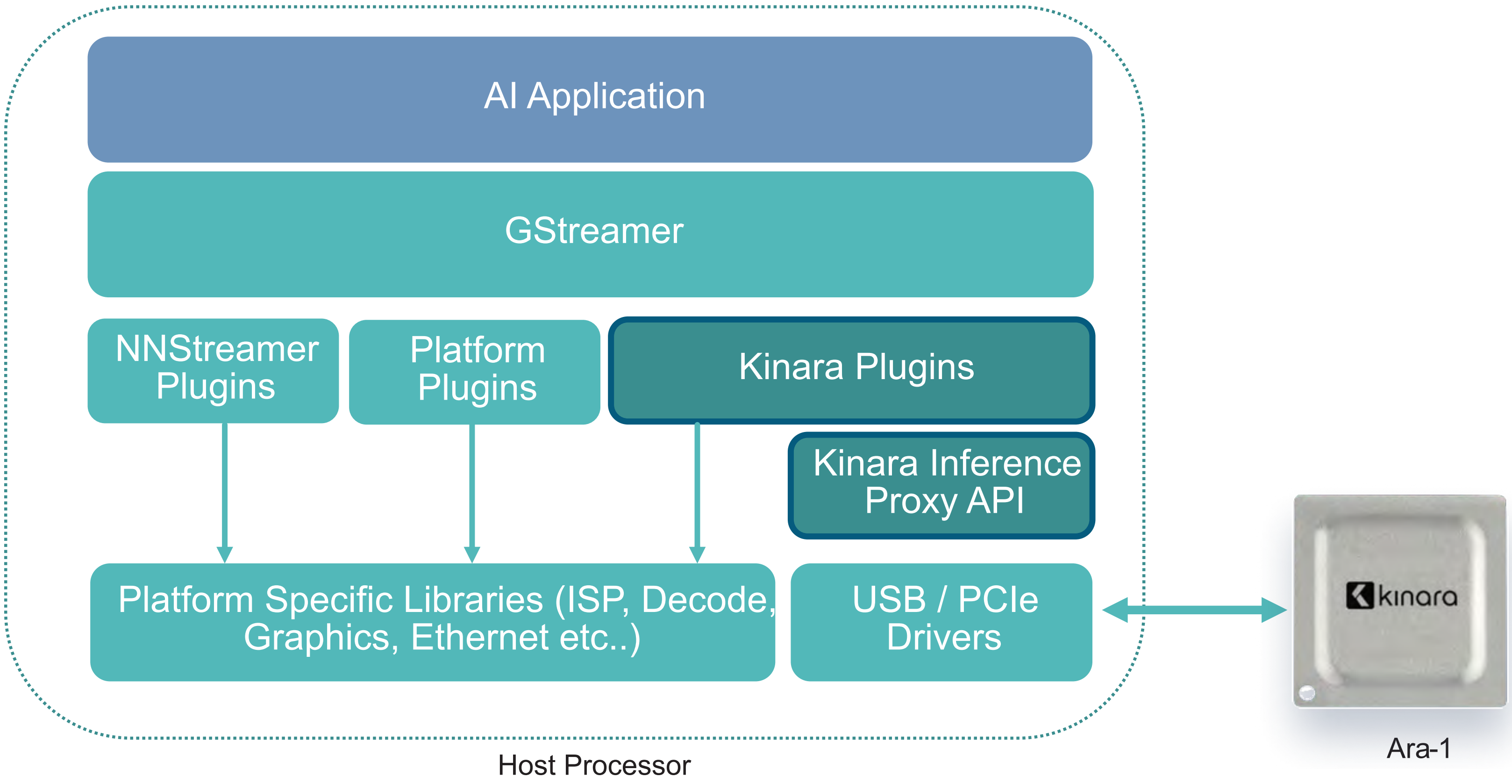


Figure 2. GStreamer and OpenCV support by an SoC vendor helps simplify the addition of Kinara's Ara-1 discrete processor.

This is the point where many developers go astray, spending inordinate amounts of time evaluating the AI processor's performance in the lab but overlooking the significance of system-level overhead. As an example, consider running a ResNet-50 benchmark on a 50 TOPS AI processor integrated in an SoC, which might be able to obtain a benchmark result of 1200 inferences/second (IPS). But in a fully functional system with all its other computational units active, those 50 TOPS effectively reduce to 12 TOPS and the real-world performance would only yield 300 IPS, assuming a generous 25% utilization factor. System overhead is always a factor if the platform is continuously processing video streams. Alternatively, with a discrete processor like the Ara-1, the utilization is effectively close to 100% because once the host SoC initiates the inference function and transfers the input data, the processor runs autonomously, utilizing its dedicated memory for accessing model weights and parameters.

Key Takeaways

It's a lot of effort to switch to a new SoC. So, instead of swapping out your existing non-AI SoC for a new AI-enabled SoC, you can keep using your existing SoC (and all the software infrastructure you've built up around it) and augment it with a discrete AI processor such as the Kinara Ara-1.

The true performance of an AI subsystem integrated in an SoC is typically significantly less than stated benchmarks. This is because in a fully operational system, there are shared resources and bottlenecks which limit AI performance. Since a discrete AI processor doesn't rely on SoC resources, other than for coordinating inferences and sending data, its performance is not impacted when the SoC is running under real-world workloads.

While an AI-integrated SoC is beneficial for reducing the system bill of materials, the discrete AI processor can easily scale up to even higher levels of performance by attaching additional Kinara Ara-1 AI processors with software support that automatically manages load balancing.