What is MAX

The Modular Accelerated Xecution (MAX) platform is a unified set of APIs and tools that help you build and deploy high-performance AI pipelines. MAX is built from the ground up, using a first-principles methodology and modern compiler technologies to ensure that it's programmable and scalable for all future AI models and hardware accelerators.

We created MAX to solve the fragmented and confusing array of AI tools that plague the industry today. Our unified toolkit is designed to help the world build high-performance AI pipelines and deploy them to any hardware, with the best possible cost-performance ratio.

Preview release

We're excited to share this preview version of MAX! For details about what's included, see the MAX changelog, and for details about what's yet to come, see the roadmap and known issues.

What's included

MAX includes everything you need to deploy low-latency, high-throughput, real-time inference pipelines into production:

✓ **MAX Engine** 🏎️

A state-of-the-art graph compiler and runtime library that executes models from PyTorch, ONNX, and TensorFlow[1] with incredible inference speed on a wide range of hardware. More about MAX Engine.

✓ **MAX Serve** ⚡

A serving wrapper for MAX Engine that provides full interoperability with existing AI serving systems (such as Triton) and that seamlessly deploys within existing container infrastructure (such as Kubernetes). More about MAX Serve.

✓ **Mojo** 🔥

The world's first programming language built from the ground-up for AI developers, with cutting-edge compiler technology that delivers unparalleled performance and programmability for any hardware. More about Mojo.

✓ **MAX Graph** 🤘

A low-level API to build high-speed graphs for inference with MAX Engine. It combines the performance and programmability of Mojo with the cutting-edge graph compiler technologies of MAX Engine. More about MAX Graph.

Preview release

There's still a lot to come, but the MAX SDK is available now as a preview. Get started now.

For details about what's still in the works, see the roadmap and known issues.

How to use MAX

MAX doesn't require that you migrate your entire AI pipeline and serving infrastructure to something new. It meets you where you are now and allows you to incrementally upgrade.

You can use the same models, libraries, and serving infrastructure that you use today, and capture immediate value from MAX with minimal migration. Then, when you're ready, you can migrate other parts of your AI pipeline to MAX for even more performance, programmability, and hardware portability.

Add performance & portability

You can start by using our Python or C API to replace your current PyTorch, ONNX, or TensorFlow inference calls with MAX Engine inference calls. This simple change executes your models up to 5x faster (thus reducing your compute costs), compared to stock PyTorch, ONNX, or TensorFlow runtimes. For example, if you execute your models from Python, you can upgrade to MAX Engine with just 3 lines of code.

MAX also makes your pipeline portable across a wide range of CPU architectures (Intel, AMD, ARM), and GPU support is coming soon. You can select the best backend for the job without rewriting or recompiling your models. This allows you to take advantage of the breadth and depth of different cloud instances at the best price, and always get the best inference cost-performance ratio.

Additionally, you can upgrade your production inference performance by using MAX Engine as a drop-in replacement for the backend in your NVIDIA Triton Inference Server.

Extend & optimize your models

Once you're executing a model with MAX Engine, you can optimize its performance further with our platform's unrivaled programmability.

MAX Engine is built with the same compiler infrastructure as Mojo, which makes MAX Engine fully extensible with Mojo. That means you can do more than just run an inference with MAX Engine—you can extend its capabilities. For example, you can write custom ops for your model the compiler can analyze, optimize, and fuse into the graph.

Going further, you can use the MAX Graph API to build your whole model in Mojo, allowing you to customize the low-level graph representation for the MAX Engine compiler. And the performance gains don't have to end there,

because Mojo can provide significant speed-ups for any compute workload—such as pre/post-processing of your model data—as we've demonstrated in a series of blog posts.

How MAX works

When we began the effort to unify the world's AI infrastructure, we realized that programming across the entire AI stack—from the graph kernels up to the application layer—was too complicated. We wanted a programming model that could target heterogeneous hardware and also deliver state-of-the-art performance in the application. That's why we created Mojo.

As illustrated in figure 1, Mojo is the core technology for the rest of the MAX platform, including our next-generation graph compiler and runtime system, called MAX Engine. You can load any model into MAX Engine and achieve low-latency inference on a wide range of hardware.

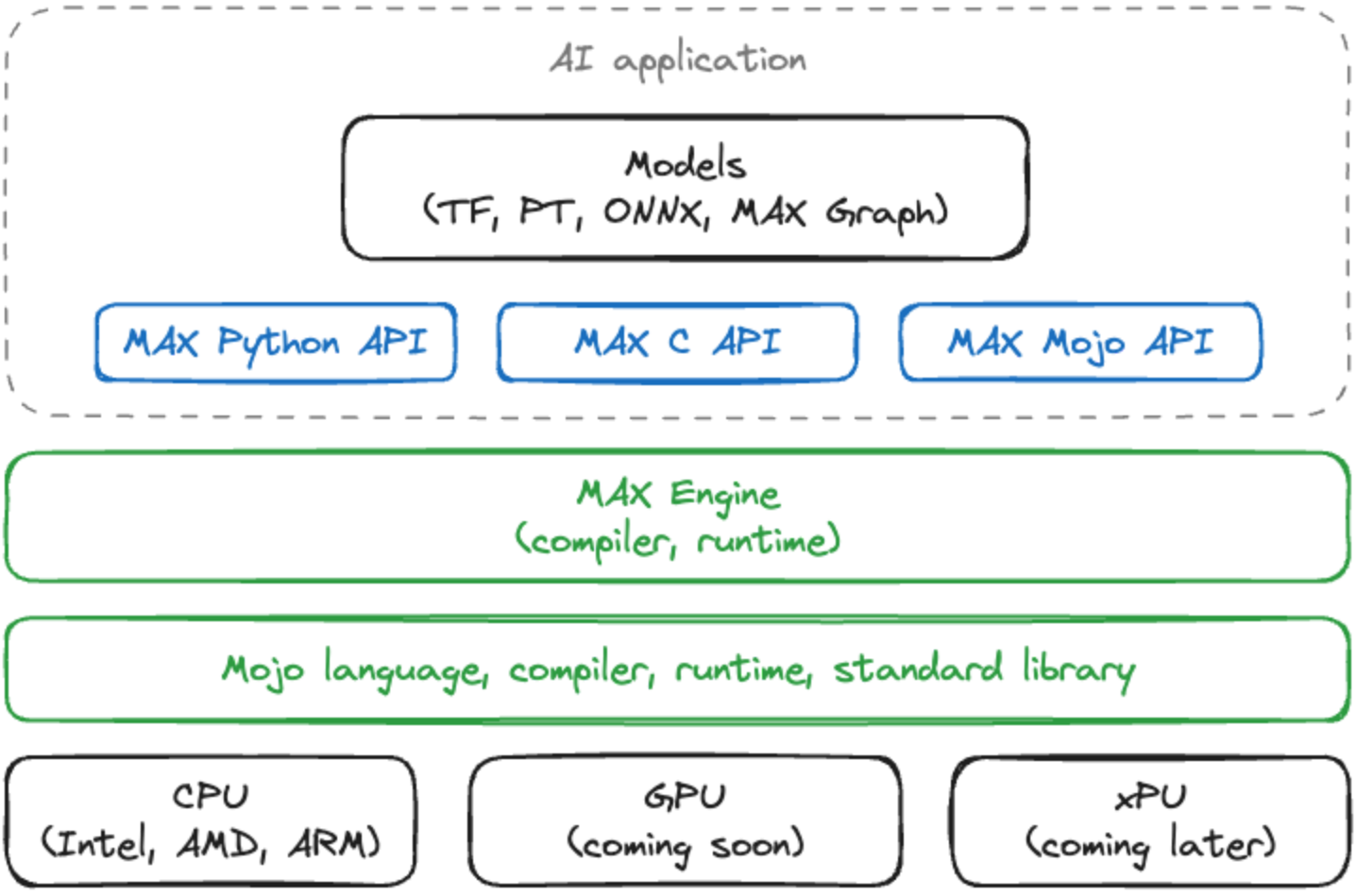

Figure 1. A simplified diagram that shows how MAX enables performance, programmability, and portability for your AI applications.

You **don't need to use Mojo** to use MAX. You can bring your existing models and execute them with MAX Engine using our API libraries in Python and C. However, using Mojo with MAX gives you superpowers. Mojo allows you to write custom ops for your model or write your inference graph for optimal performance in MAX Engine.

All of this is available for you to try today in the MAX SDK, and there's still much more to come.

A production AI pipeline requires much more than models and a runtime. It also needs data loading, input transformations, server-client communications, data monitoring, system monitoring, and more. We will add more tools and libraries to MAX that accelerate and simplify development for these other parts of your AI pipeline over time. For more details about what we're working on now, check out the MAX roadmap.

Get started

Share your ideas

Let us know what you think! What additional libraries do you need to streamline your AI development and deployment? Talk to us on Discord and GitHub.

1. Support for TensorFlow is available for enterprises only; it's not included in the MAX SDK. ↩