Technical White Paper

TDA Family SDK: Comprehensive Tools to Address Development Challenges Across Industries

ABSTRACT

This paper examines the key market trends—autonomy, AI integration and edge computing—driving complexity in embedded software development and the unique challenges they pose for engineers. The paper explores the need for robust Software Development Kits (SDKs) and software offerings that address constraints related to power, safety, security, development timelines and scalability. Lastly, this paper highlights how Texas Instruments' high-end processor families' (TDA4 and TDA5) software offerings support successful development.

Table of Contents

1 Introduction

Software is a crucial aspect of embedded devices, which are specialized computing systems designed to perform specific functions in a product or machine. Embedded software is the programming that runs on the embedded system's processor, controlling behavior and interacting with peripherals to achieve the desired functionality of a system. This software is organized as a *software stack*, with embedded software developers typically writing high-level applications at the top and leveraging lower-level middleware and firmware at the bottom. The lower-level software interfaces with hardware and is often called a *Software Development Kit* (SDK). An SDK includes reference code, documentation, tools and support provided by a processor supplier to facilitate application development.

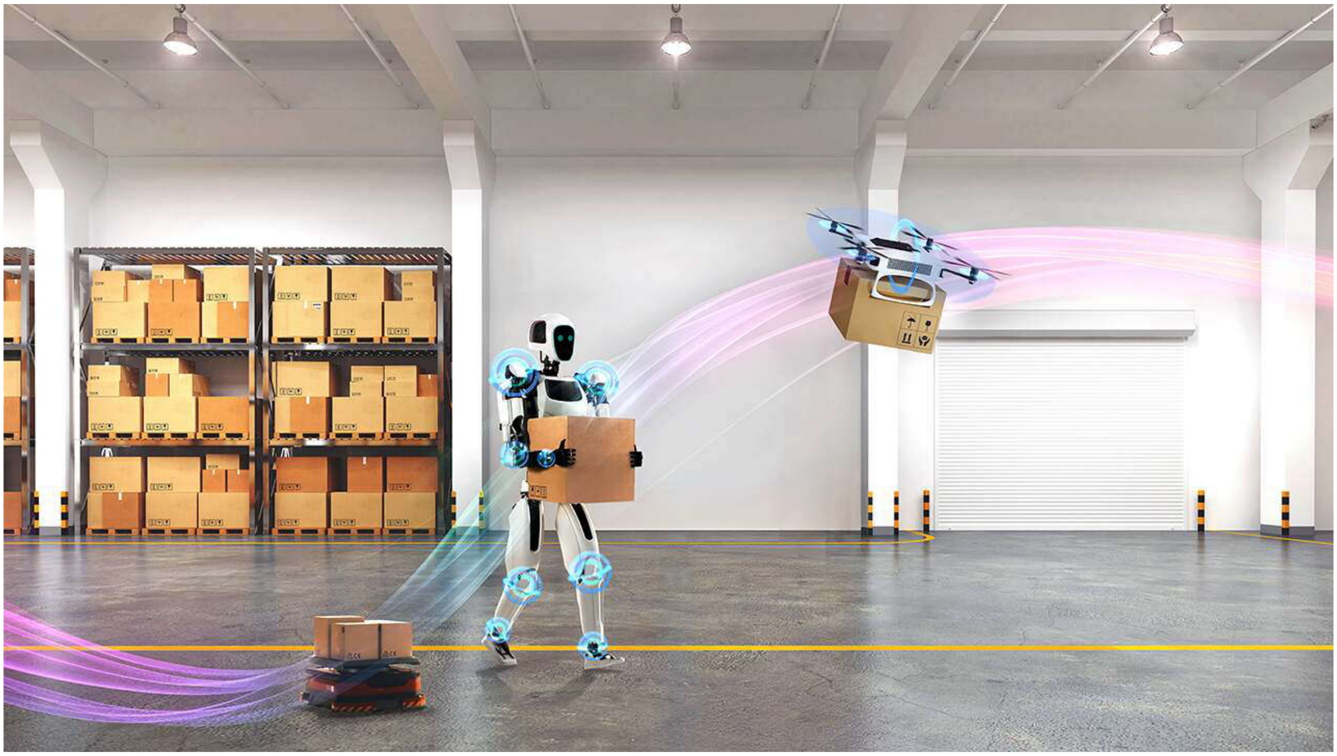

Figure 1-1. Embedded software controls processor behavior and peripheral interaction for system functionality

Across embedded system markets—from automotive safety systems to humanoid robots—engineers must invest considerable resources to innovate with high-quality embedded applications or risk falling behind competitors. A common observation among engineers is that software development represents the majority of project resources. Typically, 70–90% of the overall development effort and cost is allocated to software. Therefore, a successful product depends heavily on an embedded processor's software and tools. A robust SDK is a crucial factor in meeting the constraints and goals of embedded systems—often a complex mix of power, cost, safety, security, development timelines and scalability—to maintain a competitive edge.

2 Market Trends

Broadly speaking, there are three major embedded system market trends across industries creating unique development challenges for embedded software engineers. These trends—autonomy, AI integration and edge computing—are driving engineers to incorporate more sophisticated software features, raising the need for high-quality development tools.

2.1 Embedded Autonomy

The growth of autonomous systems, such as self-driving cars, drones and robots, is driving the development of embedded systems that can process complex sensor data and make decisions in real-time—usually with a tight power budget. As a result, software must be cutting-edge, safety-certified and efficient. For example, today's embedded developers need SDK support for robust machine learning (ML) and deep learning (DL) frameworks, such as TensorFlow, ONNX Runtime and Caffe. Libraries like these and support for optimizing these models for embedded hardware, are critical to drive product innovation.

Developers working in the automotive and industrial markets often develop in environments where safety is paramount. To maintain a high standard, embedded software engineers implement operating systems across multiple hardware cores to manage tasks and verify predictable, low-latency responses to sensor data. Their entire software stack must be safety-rated and tested against international standards such as ISO 26262 and IEC 61508 to reach ASIL B/ASIL D (automotive) or SIL 2/SIL 3 (industrial) ratings. Embedded software engineers also need a strengthened platform to build safety-critical software, such as certified libraries, development processes and tooling.

Power constraints also present a constant challenge for embedded software engineers developing autonomous systems. For example, in automotive front camera applications, thermal constraints limit a processor's power budget to five or six watts because of the typical placement under windshields, which are subjected to sustained high temperatures. Running inefficient software results in excessive power draw, adding unnecessary heat to the system. Too much heat results in thermal noise and poor image quality. Dynamic software design, low-power modes and power optimization tools are essential for developers looking to keep their embedded designs running efficiently.

Figure 2-1. ADAS embedded software engineers require access to safety-critical software

2.2 AI Integration

System autonomy is built on effective AI integration. The increasing demand for AI and ML is driving the development of accessible, efficient and specialized SDKs—often in markets where expertise is scarce and turnaround must be quick to remain competitive. Take, for example, agricultural markets. Combines, skid steer loaders and implements are now competing with features such as real-time crop monitoring, thermal imaging and wildlife avoidance. Each application demands the development of high-powered, highly specialized perception software. Developers quickly become bogged down if there is a steep learning curve to an SDK lacking a powerful toolchain, sample applications and comprehensive demonstrations. On top of this, embedded AI applications are often developed by small teams faced with the challenge of deciding between outsourcing development, adopting open-source software, or a combination of the two. To keep pace with their markets, embedded software engineers need support for a variety of OS or RTOS choices, such as Linux or FreeRTOS. As a result, engineers purchase software licenses from middleware vendors to leverage expertise and reuse code. SDKs, therefore, must support high-quality operating systems with the flexibility to incorporate third-party software stacks.

Figure 2-2. Embedded system software enables features such as real-time crop monitoring and autonomous operation.

In the humanoid robotics industry, innovation is taking place so rapidly that there is no time to wait for new product releases to begin development. The software development experience becomes a make-or-break factor for hardware adoption and, ultimately, a successful product. Simple onboarding, debugging tools and DevOps practices are major considerations. DevOps is a set of principles, including CI/CD (continuous integration/continuous delivery), that are expected by embedded software engineers from SDK suppliers to keep up with the pace of product development in such a rapidly paced industry. CI/CD is a process in which a software provider delivers updates to the product immediately, without waiting for grouped releases. In this way, developers in need of bug fixes or new features experience faster turnaround and can stay on schedule.

2.3 Edge Compute

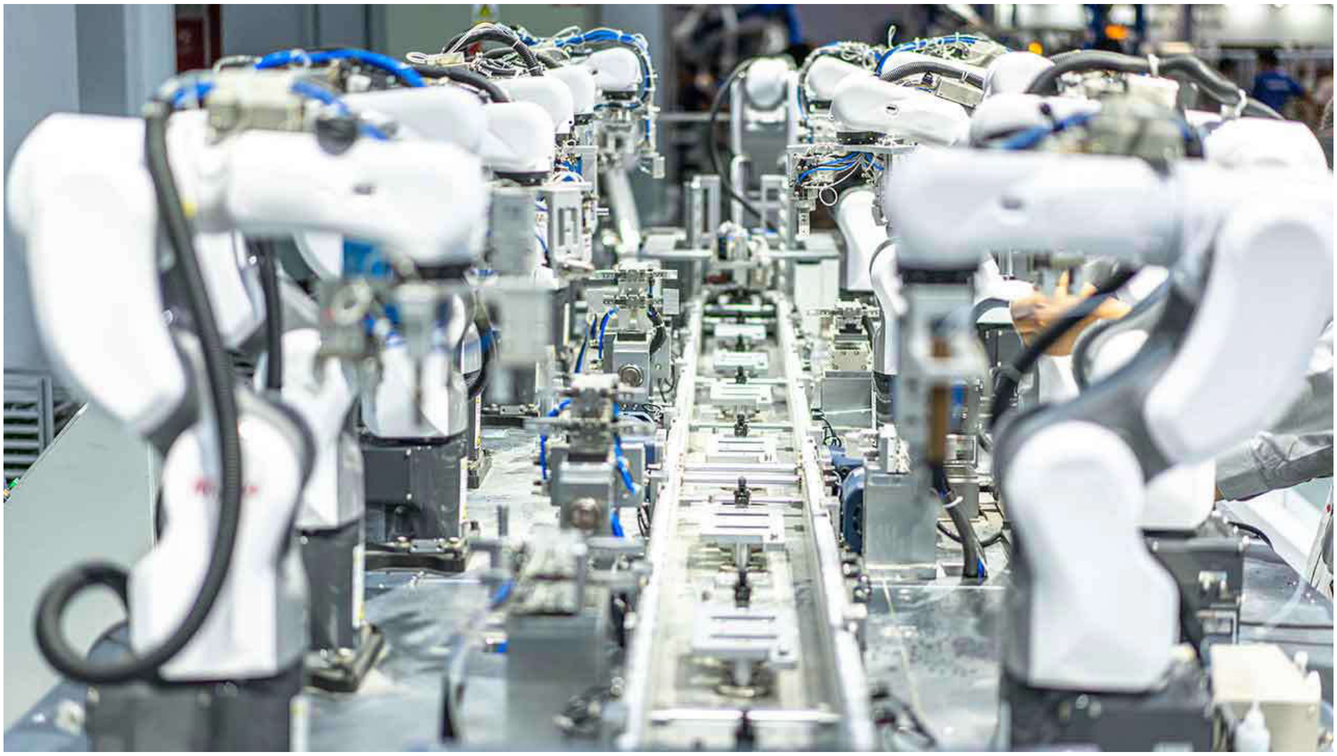

Figure 2-3. Industrial equipment developers design for 10+ year support cycles amid rapid hardware innovation

For most industrial and automotive applications, AI integration and computation in embedded systems is trending toward hardware closest to the system sensors, rather than being outsourced to cloud services. This trend, *edge computing*, has a variety of benefits, including cost savings, reduced latency and enhanced security for the consumer. However, this raises unique challenges for software developers. Unlike data centers, edge computing hardware requires extended software lifecycles. In industrial end equipment, such as programmable logic controllers (PLCs) and robotic arms, developers design their applications with 10 years of support or more in mind, often clashing with rapid hardware innovation. Forced migration is a legitimate concern for embedded software developers, and they often consider their suppliers' software longevity policy with the same weight as the hardware longevity policy. Also, embedded software engineers must design with appropriate security measures. They need in-depth tools for boot-time security, runtime protection and data-at-rest protection to offer comprehensive coverage. Without a robust development kit, these protections are not feasible to implement. On top of this, engineers must protect all layers of the stack, including hardware, firmware and software. Therefore, engineers need an SDK with industry-standard compliance and hardware-backed security, or risk safety issues in the embedded system.

In the automotive industry, edge computing trends are shifting vehicles toward increasingly complex architectures. These structures are revolutionizing the way vehicles compute AI models for safety systems. Rather than a disjointed approach to computing, vehicles are pushing AI models closer to respective sensors, leaving room for a centralized computer to make informed and unified decisions for the entire vehicle. As a result, multiple processors must communicate, creating the requirement for a scalable SDK. Embedded software engineers must verify that they can reuse drivers and toolchains across the vehicle to keep software development on track.

3 Addressing These Challenges

The embedded software industry is responding to the growing AI design challenges through a combination of technological innovations, collaborative efforts and strategic investments. Below is an overview of how embedded software engineers are keeping up and how Texas Instruments' TDA family SDK supports those efforts.

3.1 Operating Systems and Ecosystem Diversity

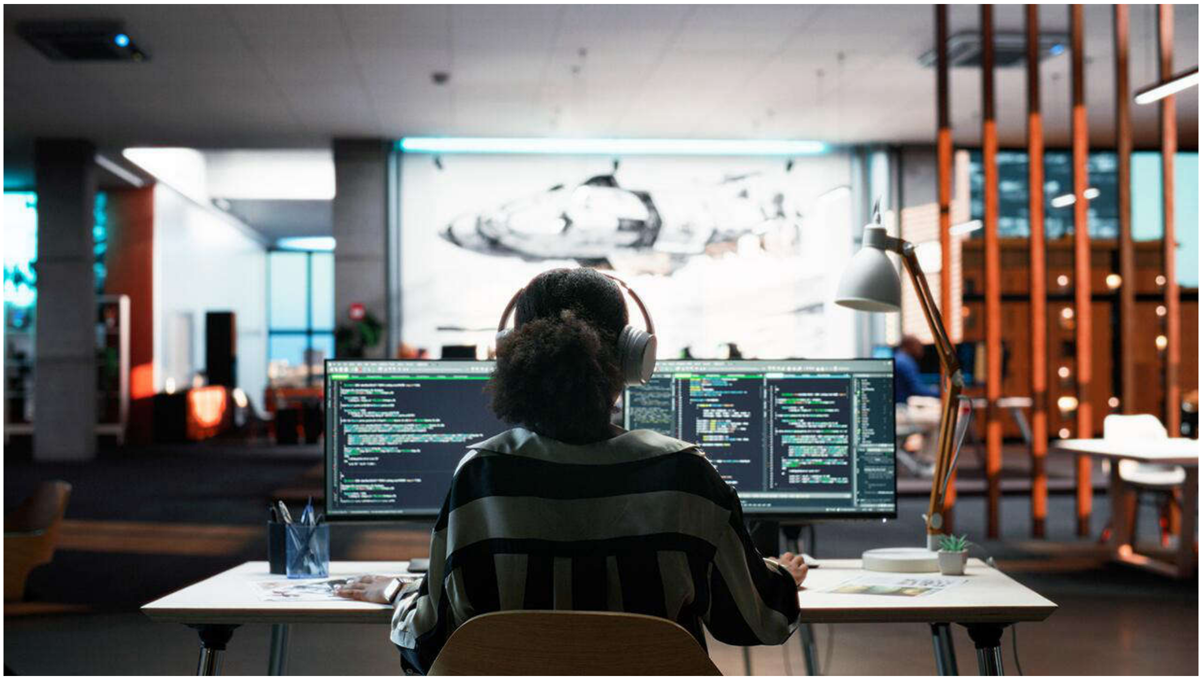

Figure 3-1. TI's partner ecosystem connects developers with regional experts and resources across the software stack

Embedded software developers look for a strong software ecosystem from silicon suppliers, as this presents a major advantage for accelerating development and maintaining product quality. Texas Instruments' (TI's) SDK enables developers with open-source software such as FreeRTOS, Linux, U-Boot, Yocto and more. As a top contributor to many of these projects, TI can provide and collaborate on architectural improvements, bug fixes, drivers and core technologies. TI hosts a well-connected partner ecosystem for developers in all regions, enabling teams to collaborate with local experts and gain access to valuable resources and services at all levels of the software stack. TI's SDK focuses on open-source OSs where engineers can leverage public community support for cutting-edge software, all while having the technical expertise of TI's engineers. Today, a comprehensive partner directory is available on TI.com.

To keep up with challenging safety standards, developers create multicore programs to run tasks in real-time, often with redundancy. To meet this demand, the TDA family SDK takes advantage of multicore hardware, allowing embedded software developers to develop across the SoC with seamless software partitioning. For example, engineers who work in safety-critical markets have access to a QNX-based SDK, which can run in parallel with the SDK RTOS. Customers receive an out-of-the-box, real-time solution adhering to ISO 26262 and IEC 61508 safety standards, aiding their software certification process. At the same time, they may leverage in-house or partner RTOS software to enable real-time software applications. Other SDK multicore investments include dedicated security and safety monitors, AI accelerators, boot, power and device management software for MCUs. This offering is essential for developers building complex, real-time and safe systems.

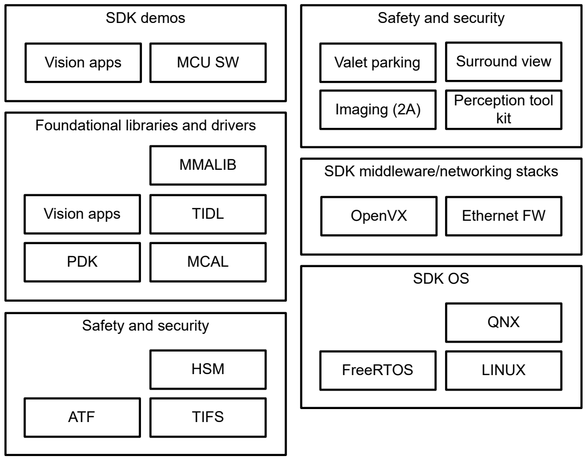

Figure 3-2. TDA family SDK example stack supported by TI

The example in Figure 3-2 does not include partner ecosystem configurations and pre-integrated designs.

3.2 Security and Functional Safety

As edge computing system architectures move into various markets, engineers face complex security and safety challenges. To tackle these challenges, developers are implementing security features such as secure boot, encryption and secure firmware. They are also obtaining functional safety certifications, such as ISO 26262 or IEC 61508, to verify systems meet rigorous safety standards. TI software tooling offers protection across security domains, from hardware up to the software layer to protect against a wide range of attacks. The TDA product family SDK categorizes security into three domains: boot-time security, runtime protection and data-at-rest protection.

Boot-time security covers security upon device start-up. Authentication and key management are critical to ensure that the device remains secure. This allows developers to implement a secure boot process with minimal effort, ensuring that only authorized software can run on the device. At runtime, TI processors enable systems with hardware driver support for cryptographic algorithms and trusted execution environments (TEEs). TI's driver support reduces the complexity and time required for security implementation. As a result, developers can get to market safer and faster. Finally, data-at-rest protection covers the protection of data while the device is not in use. Features such as secure storage are designed to provide a secure and reliable way to store sensitive data, ensuring that the data is protected from unauthorized access, tampering and other security threats.

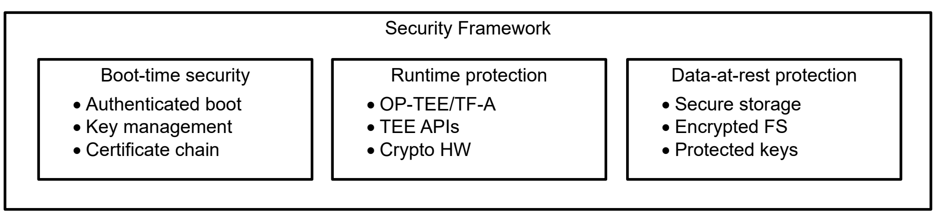

Figure 3-3. Security domains create a chain of trust from boot through runtime, ensuring system integrity and data confidentiality

Functional safety is another crucial element in embedded software design, as developers must rely on a certified SDK to keep their safety-critical software robust and eligible for sale in many regions. TÜV SÜD is a partner of choice for many developers seeking safety-certified software in embedded systems, as TÜV SÜD is a leading certifier for software tools. TI's TDA4 SDK has achieved recognition by TUV SUD for functional safety. This means that TI verifies certification of the Safety Element out of context, enabling developers to benefit from using software and certificates in building their safety- and mission-critical systems. The software development process adheres to automotive ISO26262 and IEC61508, enabling ASIL D and SIL 3 eligibility for software developers. To keep developers current with dynamic markets, TI refreshes the audits and certificates for the latest features. In this way, developers can rely on safety certifications across SDK versions..

ZERTIFIKAT ◆ CERTIFICATE ◆ 認證證書 ◆ СЕРТИФИКАТ ◆ CERTIFICADO ◆ CERTIFICAT

Product Service

CERTIFICATE
No. Z10 088989 0034 Rev. 01

Holder of Certificate: Texas Instruments
12500 TI Boulevard
Dallas TX 75243
USA

Certification Mark:

Product: AUTOSAR basic software
Jacinto AUTOSAR MCAL Drivers

Model(s): J721E AUTOSAR MCAL Drivers
J721S2 AUTOSAR MCAL Drivers
J784S4 AUTOSAR MCAL Drivers
J7200 AUTOSAR MCAL Drivers
J742S2 AUTOSAR MCAL Drivers
J722S AUTOSAR MCAL Drivers

Parameters: ASIL D
SIL 3
The Jacinto AUTOSAR MCAL includes the following drivers:
ADC, DIO, WDG, CAN, SPI, GPT, PWM, EPWM, ICU, FLS,
CDD IPC, MCU

Tested according to: ISO 26262-2:2018
ISO 26262-6:2018
ISO 26262-8:2018
ISO 26262-9:2018
IEC 61508-1:2010
IEC 61508-3:2010

The product was tested on a voluntary basis and complies with the essential requirements. The certification mark shown above can be affixed on the product. It is not permitted to alter the certification mark in any way. In addition the certification holder must not transfer the certificate to third parties. This certificate is valid until the listed date, unless it is cancelled earlier. All applicable requirements of the Testing, Certification, Validation and Verification Regulations of TÜV SÜD Group have to be complied. For details see: www.tuvsud.com/ps-cert

Test report no.: TD101327C

Valid until: 2028-09-14

Date, 2025-07-02

(Peter Weiß)

Page 1 of 1
TÜV SÜD Product Service GmbH • Certification Body • Ridlerstraße 65 • 80339 Munich • Germany

Figure 3-4. TUV SUD certification for processor selection

3.3 AI Compute

To keep pace with the demand for strong AI applications, software developers are choosing to invest in SDKs with high-quality libraries and features. TI offers robust enablement for autonomous systems, AI integration and edge computing with a powerful set of deep learning libraries, tools and SDKs. When developers install the TDA product family's SDK, they download complimentary access to Vision Apps, which allows embedded system engineers to interact with the TI Deep Learning Library (TIDL). TIDL provides the fundamental building blocks for training and implementing robust AI models on TI hardware. It supports industry-leading training frameworks, including PyTorch and TensorFlow and model formats such as ONNX and TensorFlow Lite. These industry-standard frameworks and formats are essential for today's ML training process. TIDL provides tools to compile and optimize the application for the processor, develop on OpenVX and feed data through TIDL Inference. OpenVX is an interface to abstract embedded development on TI processors and the TIDL inference library enables developers to optimize the input of data to the model. These tools offer software developers improved access to TI's powerful IP, such as the C7TM NPU, which comprise TI processors' AI engines. This way, embedded system engineers are seamlessly enabled on the latest DL and ML models to get the most out of TI's TDA processors.

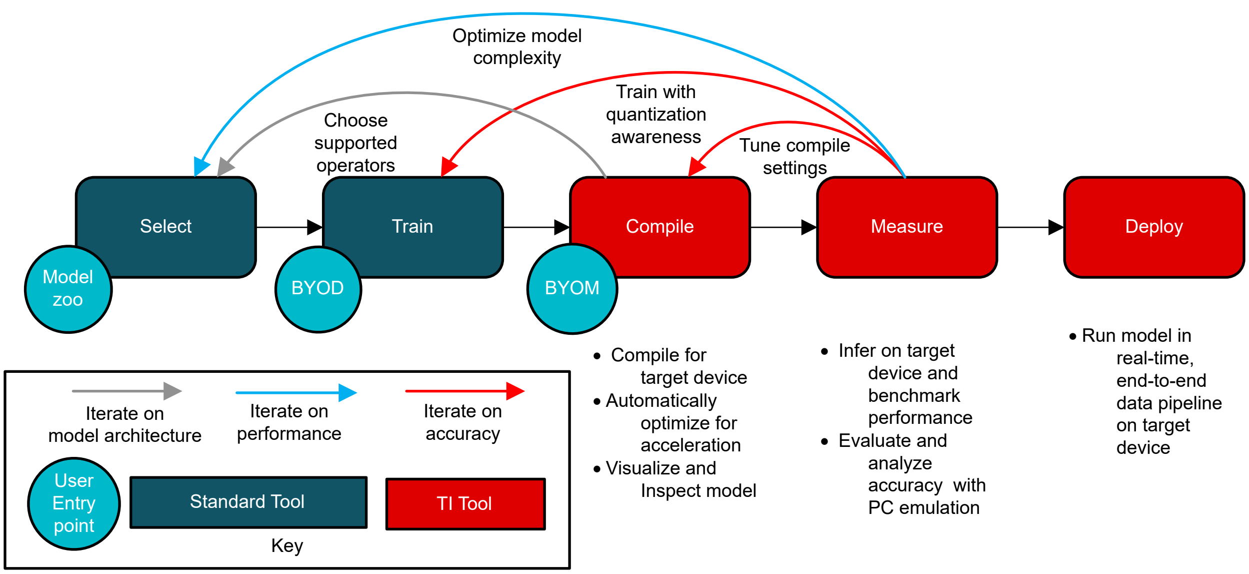

Figure 3-5. Example TIDL model implementation process

Additionally, the TI Processor SDK offers copious design and optimization tools. Libraries for graphics, display and video codecs are all available for engineers looking to develop a competitive embedded system. These features are available across TI's high-end SoC processor portfolio, including the TDA4 and TDA5 product lines. The SDK provides engineers with all the library tools and features needed to execute innovative software on their embedded system.

3.4 Board Enablement

Note that software engineers must have a comprehensive understanding of end equipment and peripheral components to deploy successful applications. For example, engineers must use power modes to compensate for thermal limits, power optimization tools where battery life is limited and develop custom drivers for interfacing with board components from different suppliers. TI invests significant resources into high-performance SoCs to meet the many needs of successful PCBs. Complementary configuration and optimization tools are readily available for components such as DDR, ISP, pins (pinmuxing), clocks and power. In the TDA4 and TDA5 product lines, mainline drivers for PMICs, serializers, ADCs and analog technology are supported regardless of supplier, meaning engineers can incorporate non-TI components into a system.

3.5 User Experience

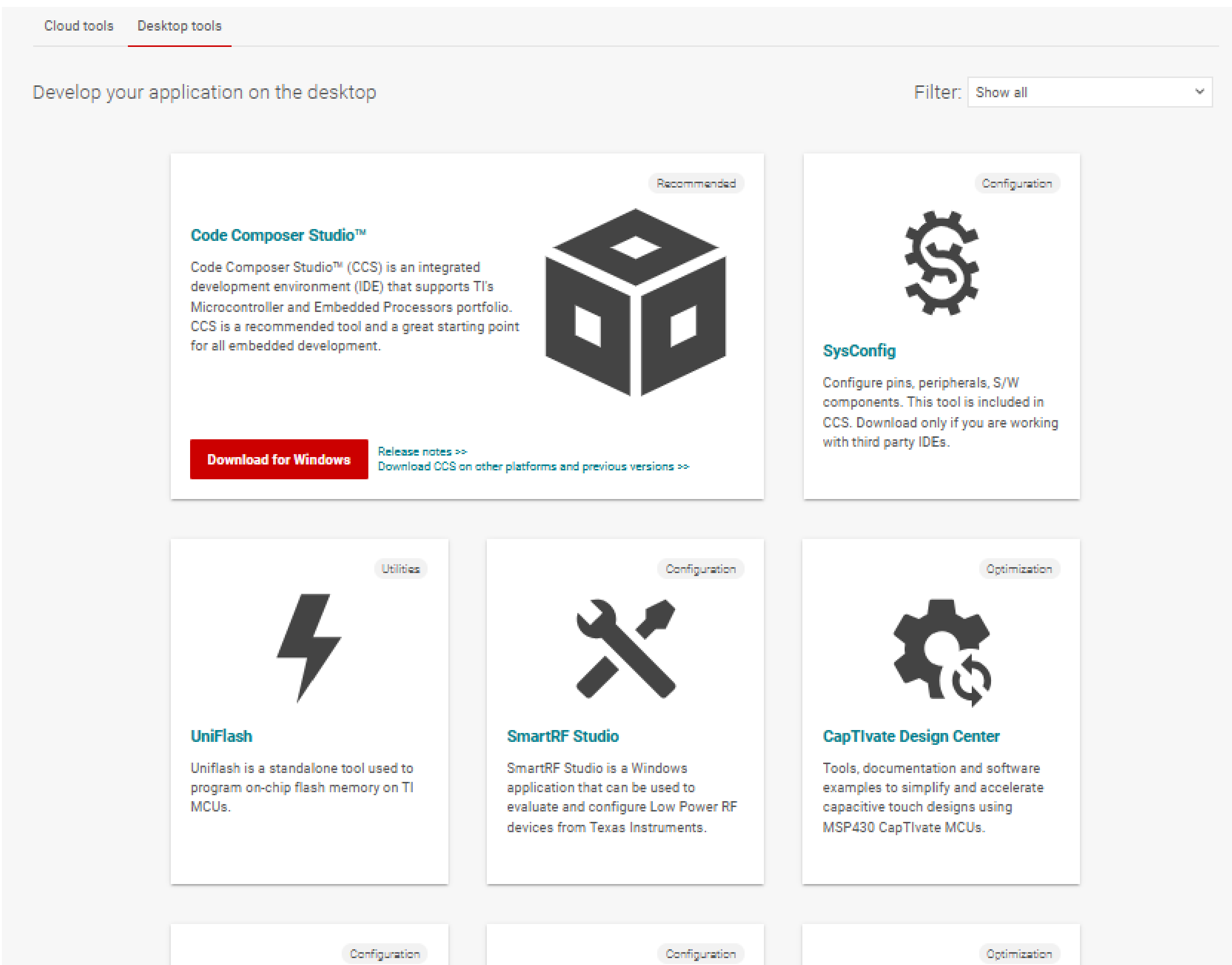

Figure 3-6. TI Developer Zone for embedded software evaluation

Regardless of an SDK's features and recognitions, the user experience for software developers can make or break the choice of processor for an embedded system. Processors, as complex as they are, not only play a crucial role in determining the overall performance, power consumption and cost but also carry the software to interact with all potential peripherals of the system. Therefore, developers are rightfully careful to pick an SDK that is quick to onboard with, carries the tools for each stage of the product's lifecycle and offers support when difficulties arise. To make sound investments, companies often have teams devoted to software research and evaluation. TI's SDK has copious resources needed to get started and jump into evaluation and development. Free resources such as demos, instructional videos and the TI developer Zone are available online today. The TI Developer Zone is a centralized platform to access all the development tools, software and training needed to develop, debug and analyze embedded software code on the desktop or in the cloud. Tools within the TI Developer Zone include Code Composer Studio (CCS), a web-based integrated development environment (or IDE) where embedded software developers may edit, build and even debug applications running on TI processors from within a web browser. With this tool, embedded software developers can evaluate and develop faster before committing to TI processors for their embedded system. On top of this, embedded software developers are encouraged to use the Resource Explorer tool to find demonstrations for CCS, libraries, documentation and package downloads to supplement their evaluation and development efforts. The Resource Explorer tool hosts training material for a wide array of topics such as inter-processor communication (IPC), one-time programmable (OTP) key-writing and tutorials on porting Linux to custom hardware.

One notable component TI offers within the Developer Zone is called the Edge AI Studio, a collection of tools that enables development, benchmarking and deployment of AI applications. It supports Bring-Your-Own-Data (BYOD), enabling the re-training of models from the TI Model Zoo—a large collection of deep learning models validated to work on TI processors for Edge AI. For vision applications, engineers may evaluate accelerated DL models on remotely accessed development boards using trained or custom code. The TI Github hosts an even more extensive and flexible set of model-development tools. Developers may use these Linux-based PC tools to:

- Train TI models for a custom dataset (BYOD)
- Compile a custom or open-source model for TI's AI accelerators (BYOM)
- Analyze and optimize models for performance and accuracy

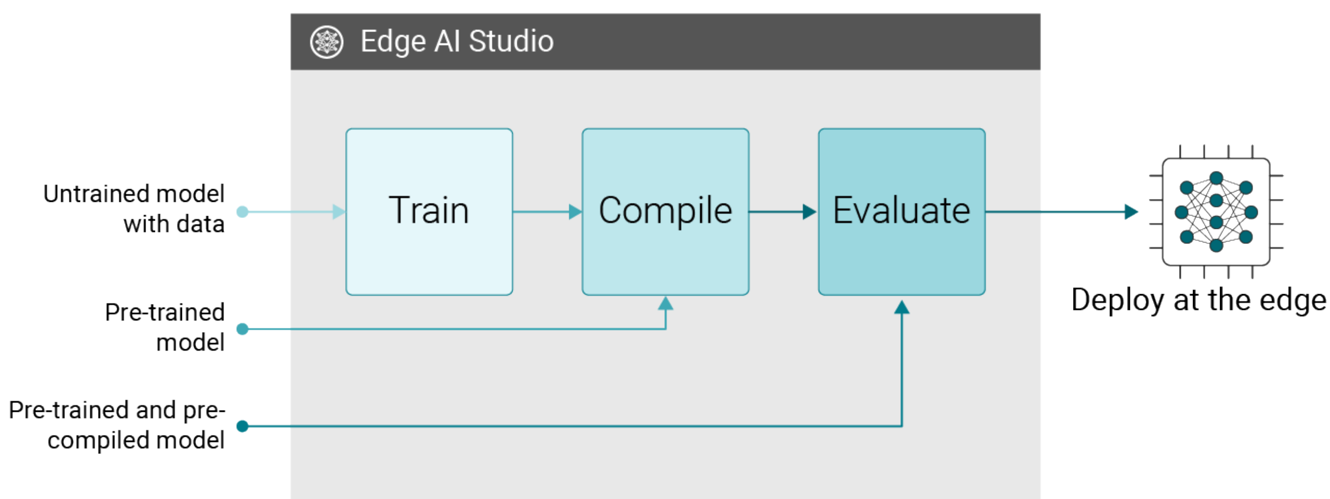

Figure 3-7. Edge AI Studio: tools for AI development, benchmarking and deployment..

As a result, engineers with diverse needs can implement AI applications faster and have safe resources to research and distinguish themselves within their markets.

3.6 Scalability

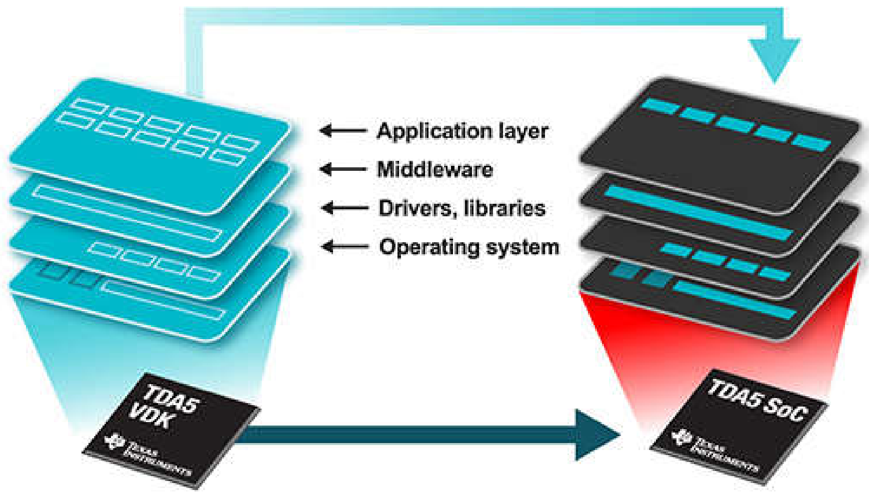

Figure 3-8. Synopsys Virtualizer™ enables pre-production silicon development

Embedded software engineers are challenged to continuously improve and scale their software applications, often across broad portfolios. In fast-paced markets, such as automotive ADAS and humanoid robotics, engineers must develop software prior to hardware availability to stay competitive. As a result, embedded software engineers must scale software on top of their current portfolio or develop atop virtual hardware simulators. The latter market trend has been nicknamed the digital twin phenomenon and appears everywhere from medical to manufacturing industries. Across TI's processor portfolio, software toolchains, drivers and firmware are reused, so software engineers can accelerate their scaled applications. Engineers may start their next-generation product development on their current TI hardware prior to porting their software to their new device. This is common with TI customers who develop across the TDA4 product family and must keep their software development at pace with their hardware roadmaps. Embedded software developers may develop with TI's next-generation processor today using the TDA5 Virtualizer™ Development Kit (VDK). Built with Synopsys Virtualizer™, the VDK enables the entire software stack to be developed on a register-accurate simulation of TDA5 SoCs. Embedded software developers have access to the TDA5 family prior to the release of silicon, able to execute unmodified binaries of production software. Therefore, they can stay on track with deployment and release products with a competitive timeline.

3.7 Lifecycle Flexibility

It is generally true that embedded systems must be developed rapidly to keep pace with new technology. It is also common, in markets such as off-highway vehicles, industrial transportation and material handling, for embedded software to remain in use for 10 or more years, with limited to no access to software updates once deployed. When silicon or third-party software suppliers retire a software product, embedded software engineers face unexpected software rewrites, which are expensive and time-consuming. Therefore, software engineers expect an SDK with quick turnaround for the latest features and bug fixes and stability to stay relevant for years to come. On top of this, suppliers must assure customers that the software is supported after deployment, until the end of life (EOL). TI has the unique advantage of possessing a powerful, mature SDK with an industry-leading longevity policy and the resources to implement lightning-fast turnaround through DevOps and CI/CD practices. Embedded software developers can develop on a next-generation SDK with less delay for bug fixes and new features, avoiding getting caught between expensive delays and dropping features. Developers can deploy with confidence, because their product has support for its extended lifecycle.

4 Maintaining a Competitive Advantage

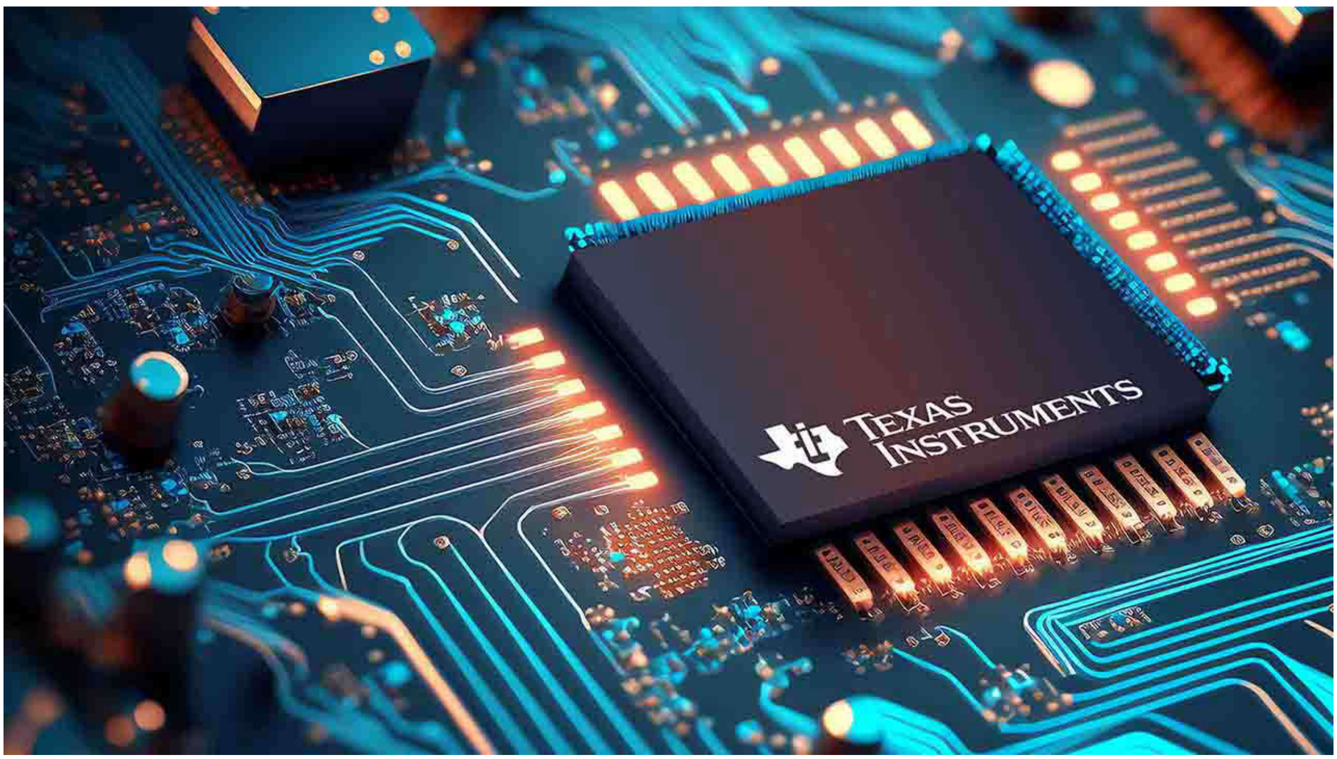

Figure 4-1. TI processor SDK and tooling empower embedded software engineers in their markets

Overall, TI's processor SDK and tooling empower embedded software engineers to maintain a foothold in their embedded system market. With powerful libraries, security features and applications within the SDK, developers are well positioned to innovate on TI hardware. TI software is made accessible through documentation and support tools, so engineers have direct and comprehensive support throughout their product's lifecycle. On top of this, the scalable attributes, virtualization tools and release strategy situate TI as a safe and competitive software choice for developers when choosing their processor. Only when embedded software engineers are supported with all these attributes, are they able to excel in their own markets. Visit TI.com to get started today or take a look through TDA4's flagship processor SDK offerings.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale, TI's General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.