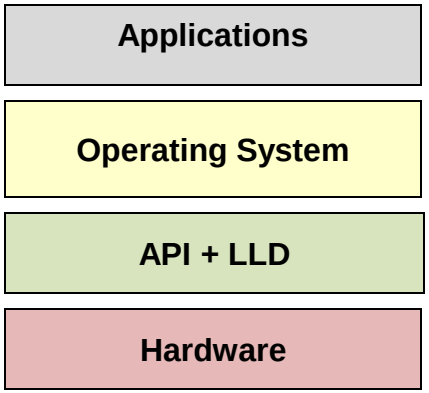
# KeyStone Multicore
# Software Development Ecosystem

# Agenda

- Operating System Basics
- TI's Traditional Development Support
  - DSP SYS BIOS
  - Framework, Utilities, and Drivers
- Multicore Development Ecosystem
  - Code Composer Studio (CCS)
  - Multicore Software Development Kit

# Operating System Basics

- On the surface, the OS provides:
  - Management, scheduling, and prioritization of system-level memory, processors, and input/output devices to enable multiple processes/threads.
  - Standard API for utilities:
    - Print
    - Networking (sockets)
    - Monitoring
  - Mediates to hide hardware and driver details from the application
- Looking under the hood, the OS also provides:
  - Low level drivers (LLD) to abstract communication with the hardware layer, including device-specific modules/interfaces (from one or more manufacturers)
  - One or more layers of utility APIs (routines, structures, variables, etc.) that connect the application to the LLD
- code development Environment:
  - Integrated Development Environment (IDE) tools:
    - Source code editing
    - Build automation
    - Debugging
  - Libraries (code, sub-routines, values, etc.)



# Traditional TI DSP Software Support

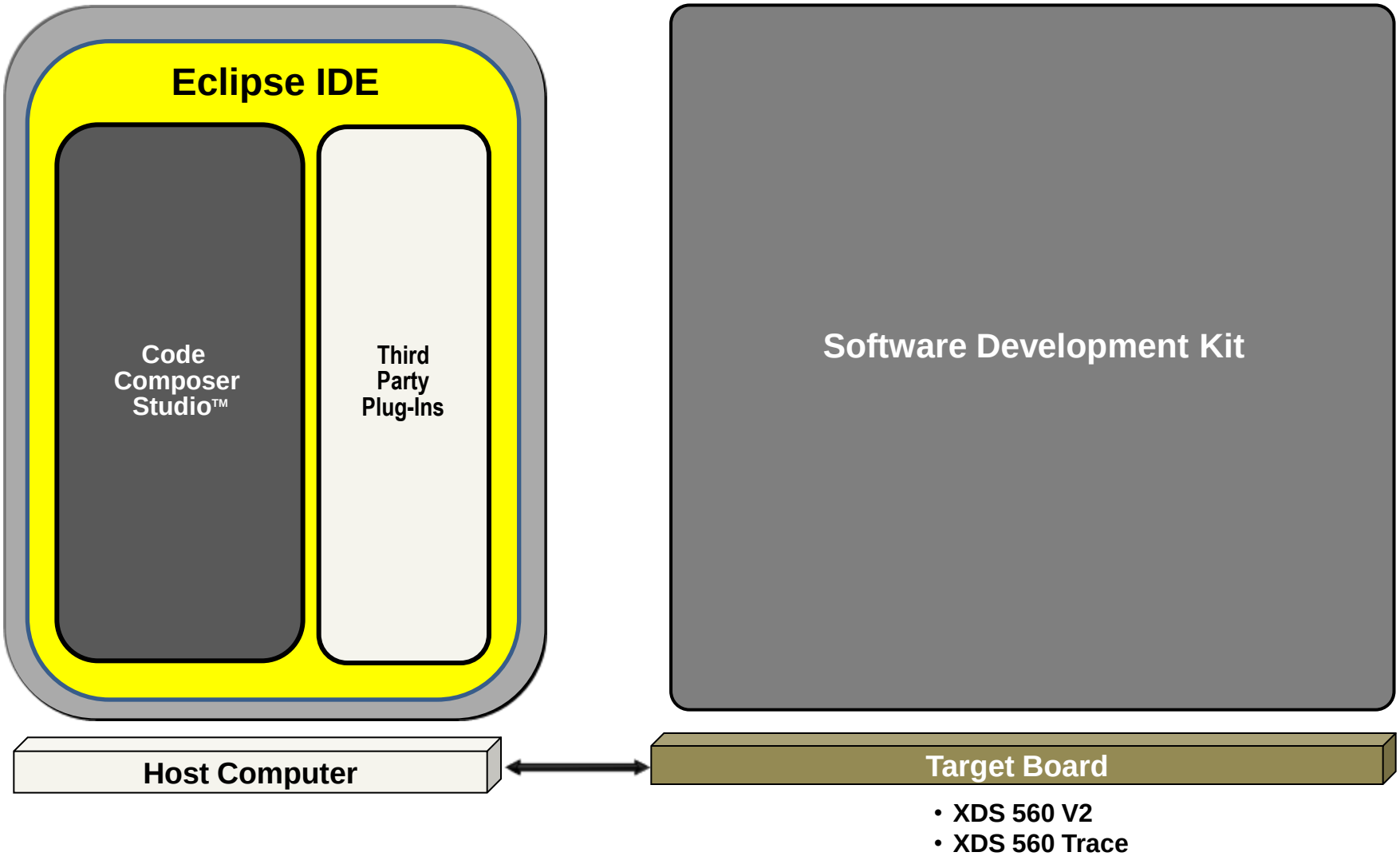
- SYS/BIOS (formerly DSP/BIOS) Real-time Operating System (RTOS) support for DSP-only and DSP+ARM devices, including:
  - Scheduling (hardware/software interrupts, tasks)
  - Memory management
  - Communications
  - Performance monitoring
- Optimized algorithm libraries
- Framework components
  - Easy access to peripherals and other resources (DMA)
  - Abstracts hardware from the application (CSL)
- IDE for single core and multicore devices
  - Code Composer Studio (CCS)
  - Code Generation (CodeGen)
  - Debug, emulation, monitoring, profiling

# Developing on KeyStone SoC

- TI's KeyStone SoC devices offer many advantages:
  - Multicores
  - Advanced core performance
  - Distributed memory architecture
  - Multiple peripherals and co-processors
  - High-speed transport mechanisms
- The challenge: How to simplify programming and development of optimized applications on KeyStone devices?

# TI Software Development Ecosystem

## Multicore Performance, Single-core Simplicity



# Eclipse-based IDE Advantages

- Eclipse-based CCS supports application development on multiple cores/devices:
  - Multiple perspectives (default and custom)
  - Three debug operation modes
  - Advance debugging, monitoring, and profiling
- Optimized compiler/optimizer/assembler/linker
- Support for OpenMP
- Enables addition of third-party plug-ins
- Multiple configurations allow a single executable or multiple executables to be generated for the same project.
- GUI interface for SYS/BIOS, project-based system to build drivers and utilities for developer's target platform.

# Multicore Development Requirements

- Standard set of APIs to configure and utilize peripherals, accelerators and other hardware resources
- Compliant implementation of packet- and network-based protocols
- Utilities to boot, test, debug, and monitor execution
- Individual core-based real-time operating system
- Communication and facilitated cooperation between processes, cores, and devices, as well as between peripherals and cores
- Optimized, ready-to-use algorithm libraries, example code, and demonstration applications

**TI's solution:**

**The multi-layered software system known as MCSDK.**

# What is MCSDK?

TI's Multicore Software Development Kit (MCSDK):

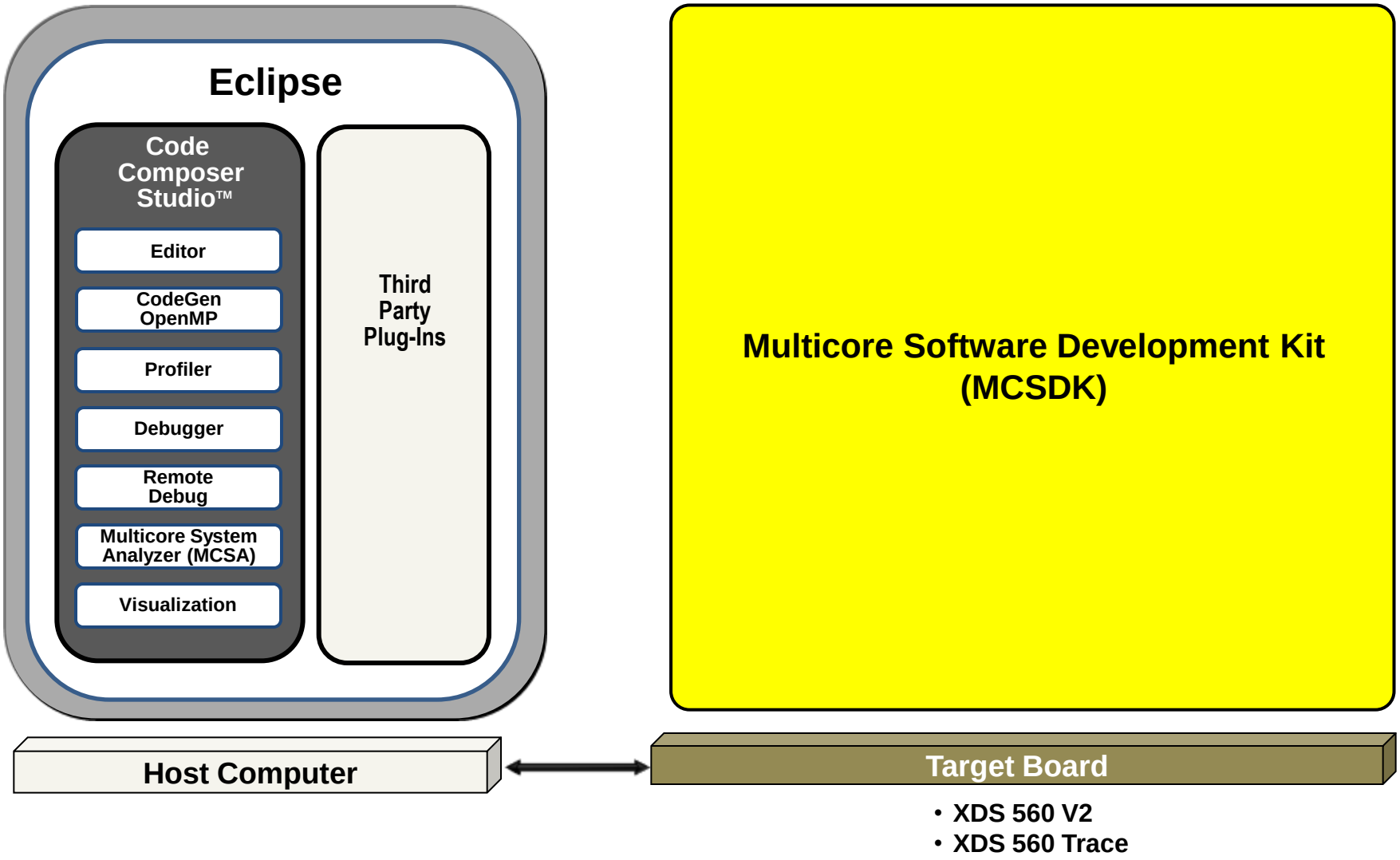
- Provides the core foundational building blocks for customers to quickly start developing embedded applications on TI high-performance multicore DSPs:
  - Uses ether the SYS/BIOS or Linux real-time operating system
  - Accelerates time-to-market by focusing on ease of use and performance
  - Provides multicore programming methodologies and utilities
- Simplifies porting of applications:
  - To a standard evaluation platform
  - From a standard evaluation platform to customer's target platform
  - To next generation platform hardware
- Available as a free download on TI.com, bundled in one installer as source code along with pre-built libraries

# Developer Challenges > MCSDK Solutions

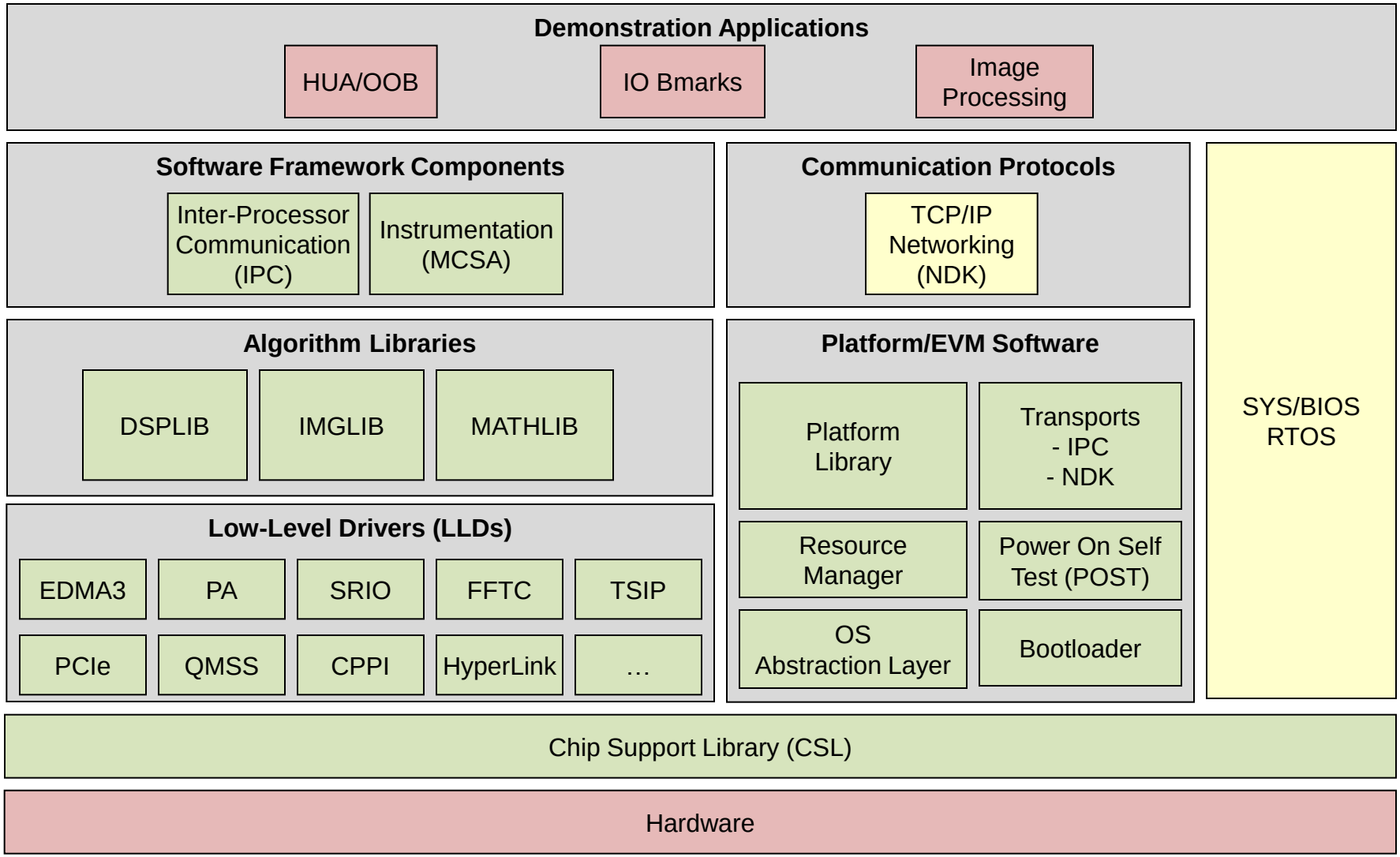
| Development Requirement | MCSDK Solution |
|---|---|
| Hide hardware details from the user to simplify process of porting to new hardware. | Chip Support Library (CSL) is the only MCSDK layer that depends on the hardware. This layer is completely transparent to the user/application. |
| Standard API to talk to peripherals, accelerators, and other resources | Low Level Drivers (LLD) provide standard API to initialize, configure, and utilize peripherals and other resources. LLD blocks include SRIO, PCIe, PA, CPPI, QMSS, FFTC, and many more. |
| Utilities to facilitate system operations | Platform/EVM Software provides platform-level utilities such as bootloader, Power On Self Test (POST), resource manager, and platform utilities. |
| Efficient real-time individual core operating system | SYS/BIOS provides an efficient, mature, real-time operating system with a low memory footprint. |
| Reliable interface to external networks | Network Development Kit (NDK) provides a standardized interface for common packet- and network-based communication protocols (e.g., IPV4 and IPV6-compliant TCP/IP). |
| Coordination of tasks/process across multiple cores | Inter-Processor Communication (IPC) provides several-high level utilities and libraries to communicate between cores and enable multiple cores to work together. |
| Facilitate application development | Optimized algorithm libraries with standard APIs. |
| A starting point for multicore application development | Demonstration applications (e.g., Image Processing) show how to build and run a complete multicore application. |

# TI Software Development Ecosystem

## Multicore Performance, Single-core Simplicity

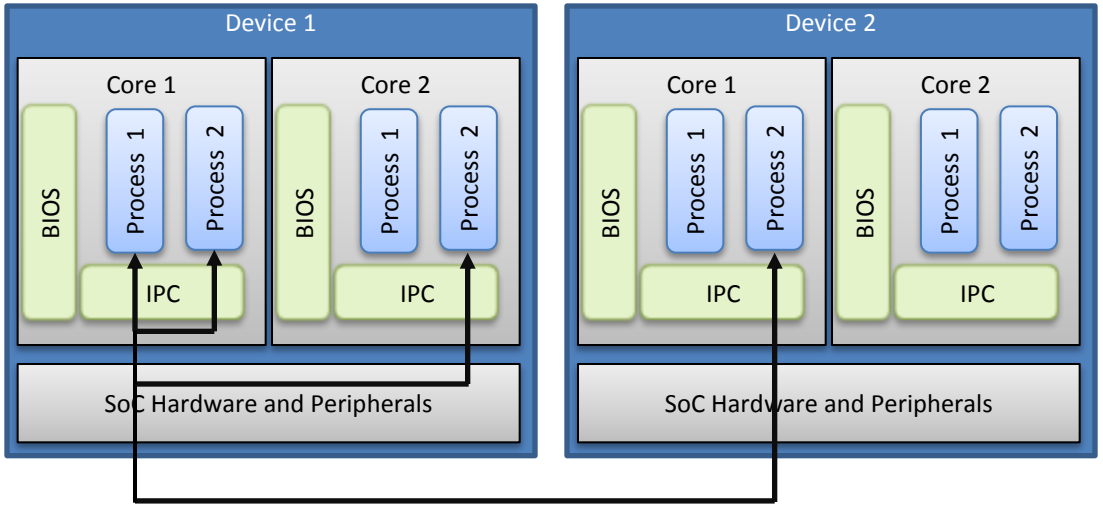


# BIOS-MCSDK Software Layers

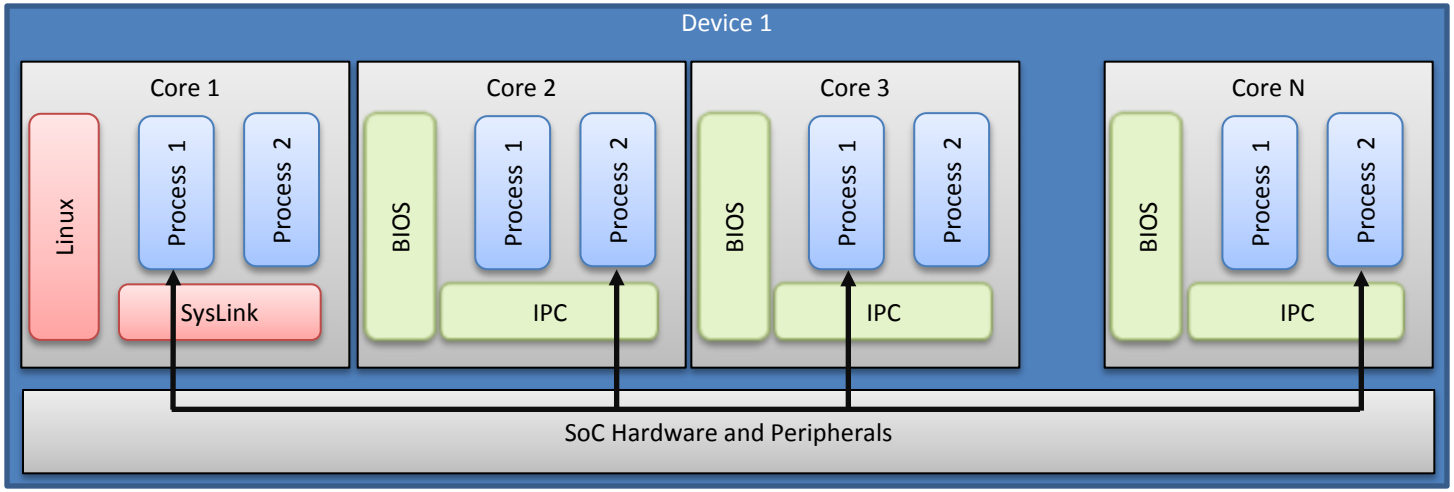


# Inter-Processor Communication (IPC)

IPC provides a standard interface between processes/threads in the same core, between two cores, and between two devices.




| IPC Transports | Task to Task | Core to Core | Device to Device |
|---|---|---|---|
| Shared Memory | x | x | |
| Navigator/QMSS | x | x | |
| Serial Rapid IO (SRIO) | x | x | x |

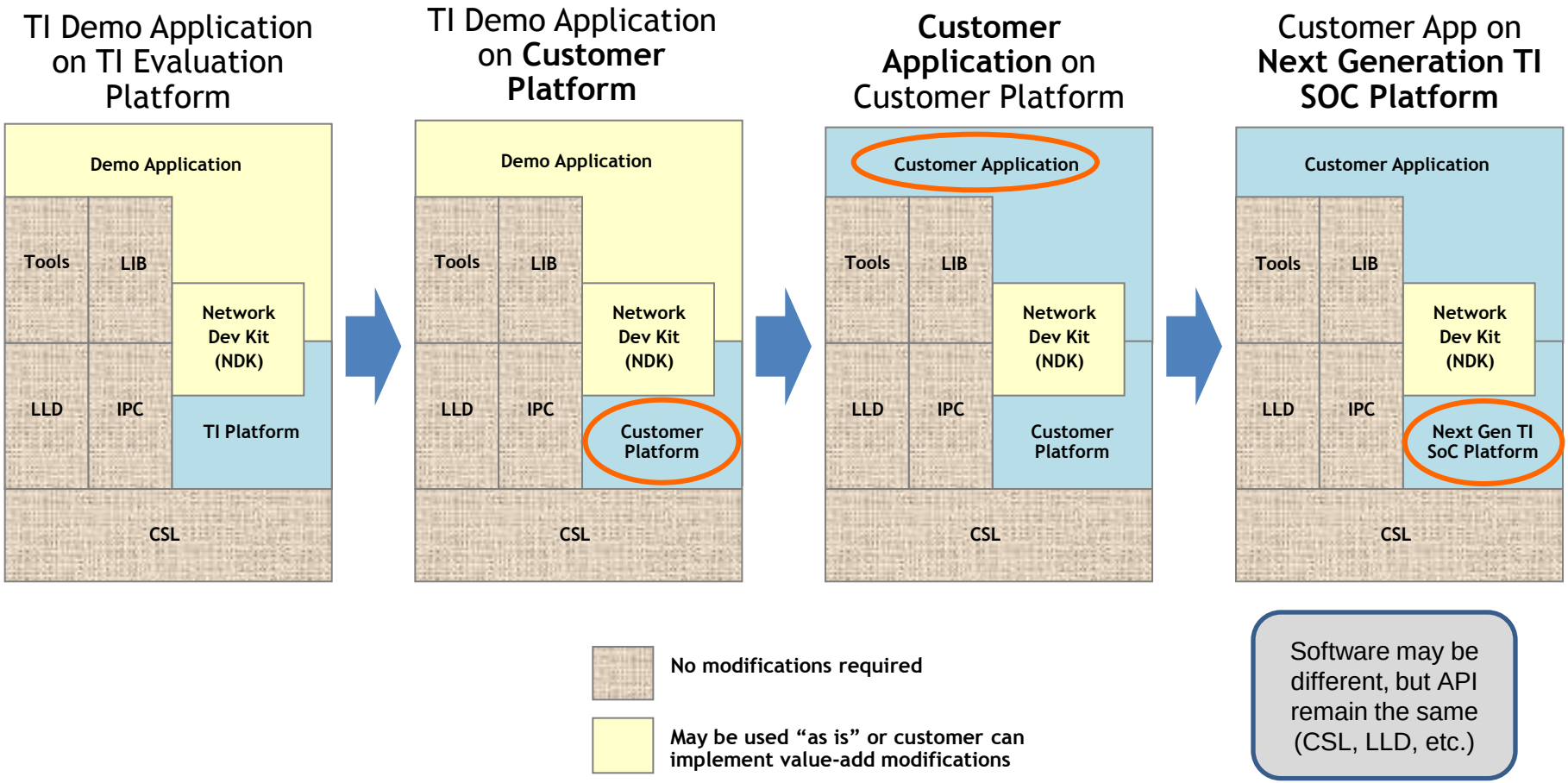


# MCSDK Algorithm Libraries

Algorithm libraries contain C66x C-callable, hand-coded, assembly-optimized functions for specific usage:

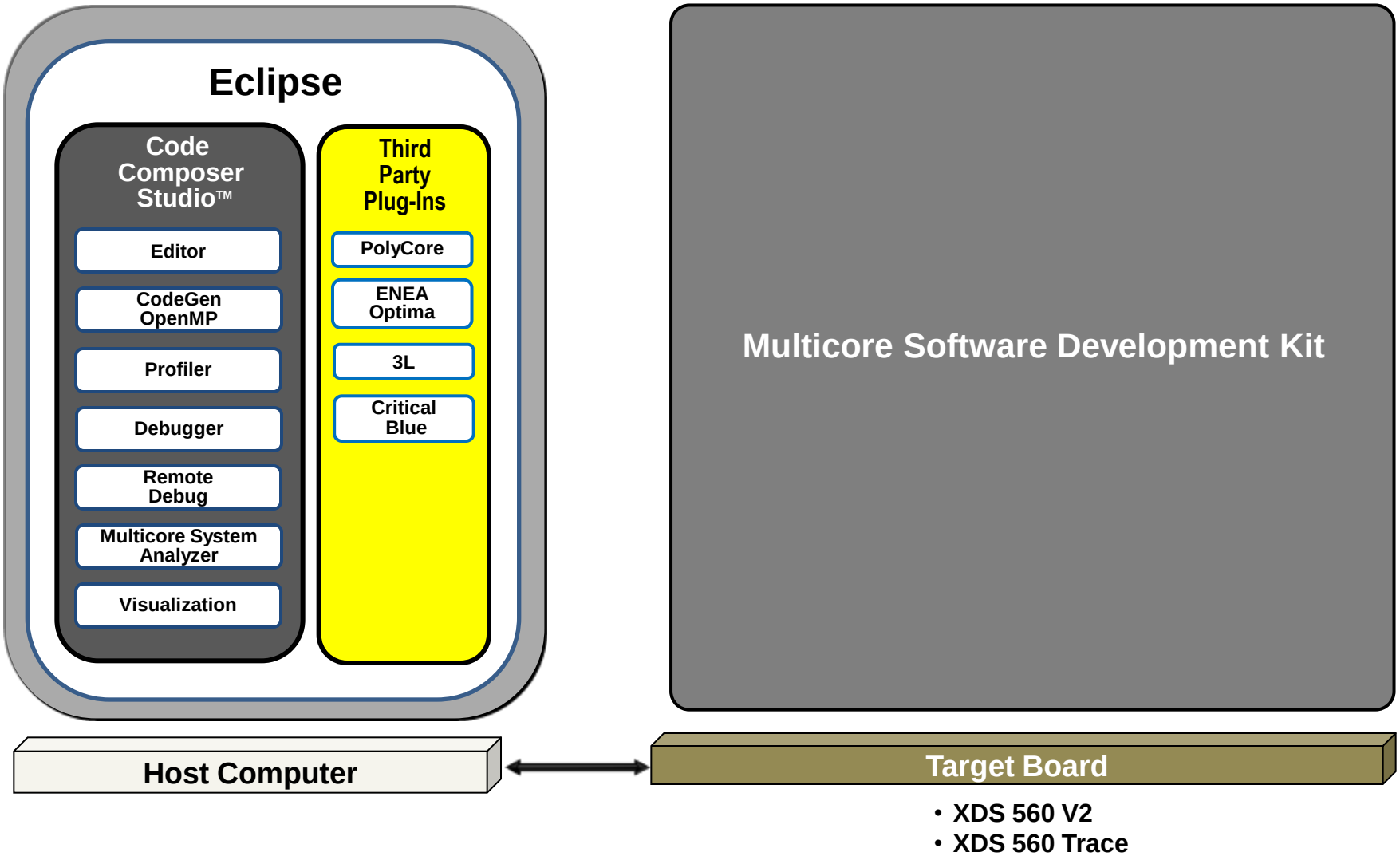
- **DSPLIB** provides signal-processing math and vector functions:
  - Adaptive filtering
  - Correlation
  - FFT (e.g. FFT functions for 'npoint' FFTs)
  - Filtering and Convolution (e.g., FIR, IIR filter functions, etc.)
  - Matrix (e.g., single and double precision matrix multiplication, etc.)
- **IMGLIB** provides image/video processing functions:
  - Compression & Decompression (e.g., forward and inverse DCT, motion estimation, quantization, etc.)
  - Image Analysis (e.g., edge detection, histogram, thresholding, etc.)
  - Image Filtering and Conversion (e.g., color space conversion, convolution, correlation, error diffusion, etc.)
- **MathLIB** provides floating-point math functions:
  - Single-precision (e.g., cosine/sine/tangent of a floating point number, etc.)
  - Double precision (e.g., similar functions as above with argument type and return values to be of type double)
- More info: http://processors.wiki.ti.com/index.php/Software_libraries

# Simplified Development & Migration



# Software Development Ecosystem

## Multicore Performance, Single-core Simplicity



# Third Party Plug-Ins

Eclipse allows developers to integrate third-party software tools that provide additional multicore programming, profiling and analysis capabilities:

- **Poly-Platform from PolyCore** http://polycoresoftware.com is a development framework consisting of tools and run-time software that provide a programming model for applications to scale from one to many cores.
- **Prism from CriticalBlue** http://www.criticalblue.com provides multicore analysis and exploration tools to evaluate parallelization strategies of existing software applications.
- **Optima from ENEA** http://www.enea.com includes overview and management tools for multicore systems, profiling tools showing resource usage, and debug tools that track execution of application and operating system events.
- **Diamond from 3L** http://www.3l.com is a tool-suite and model that provides a highly automated development flow from concept through to applications running in multiprocessor hardware.

# MCSDK Top-level Directory Folders

- /mcsdk
- /pdk
- /edma3
- /bios
- /dsplib
- /imglib
- /mathlib
- /ipc
- /ndk

# For More Information

- MCSDK Product Folder:
  http://www.ti.com/tool/bioslinuxmcsdk
- Download CCSv5 and MCSDK software:
  http://software-dl.ti.com/sdoemb/sdoemb_public_sw/bios_mcsdk/latest/index_FDS.html
- KeyStone C66x Multicore Wiki Resources
  http://processors.wiki.ti.com/index.php/Keystone_Device_Architecture
- For questions regarding topics covered in this training, visit the support forums at the TI E2E Community website http://e2e.ti.com