

# Processor SDK Linux Overview

# Other Processor SDK Training

The [Processor SDK Training Series](#), which includes:

- [Processor SDK Overview](#)
- [Processor SDK RTOS](#)
- Processor SDK Linux:
  - [Processor SDK Linux Components](#)
  - [Processor SDK Linux Matrix](#)
  - [Processor SDK Linux Installation](#)

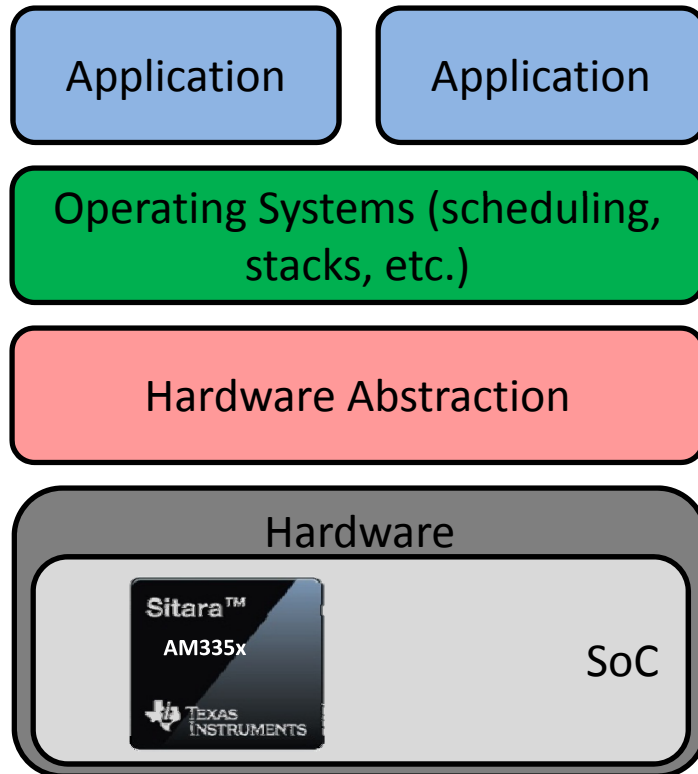
# Agenda

- Why Processor SDK Linux?
- Defining an SDK for Linux
- Linux SDK Experience

# Why Processor SDK Linux?

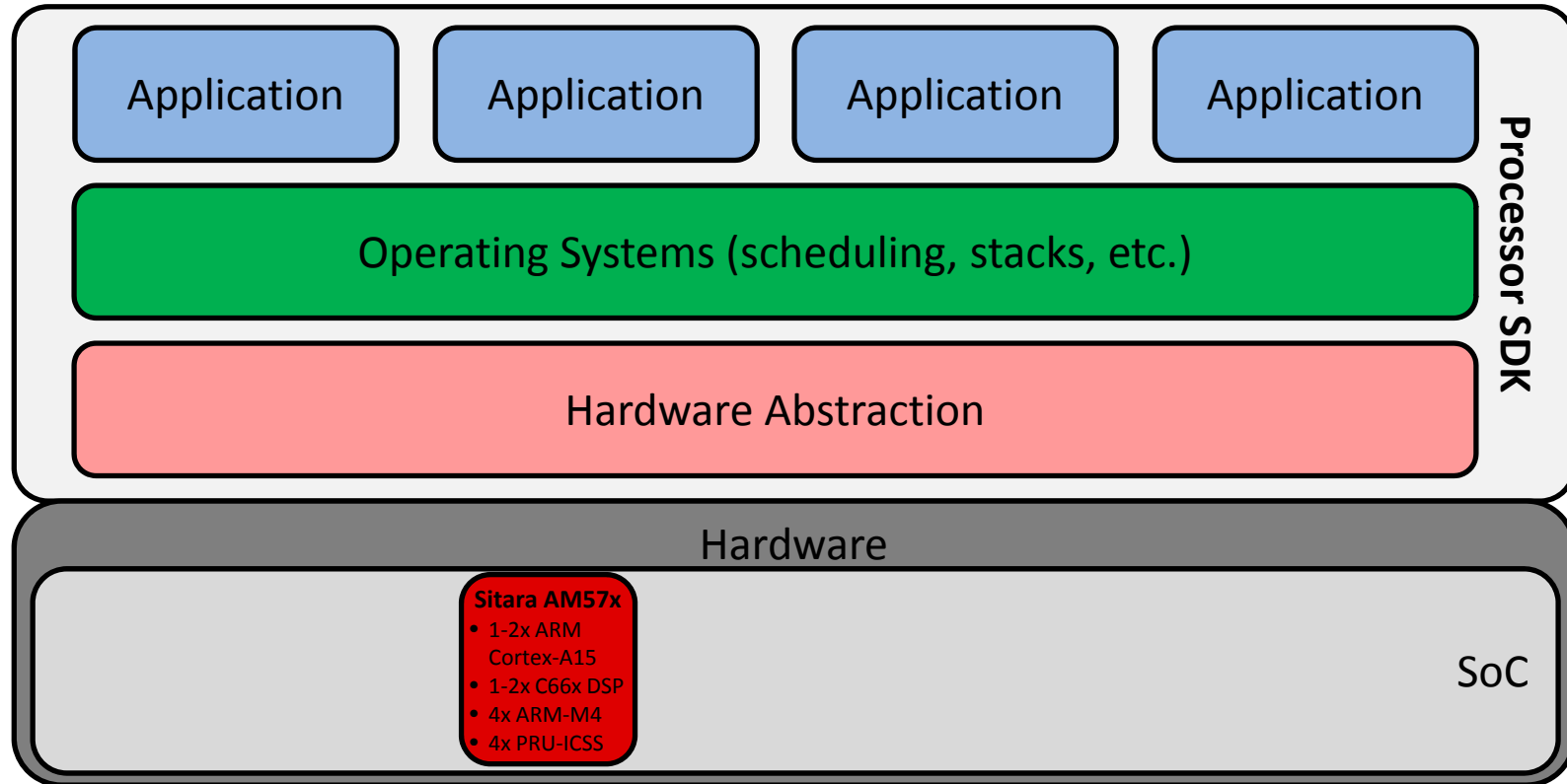
Processor SDK Linux Overview

# Problems Need Abstraction

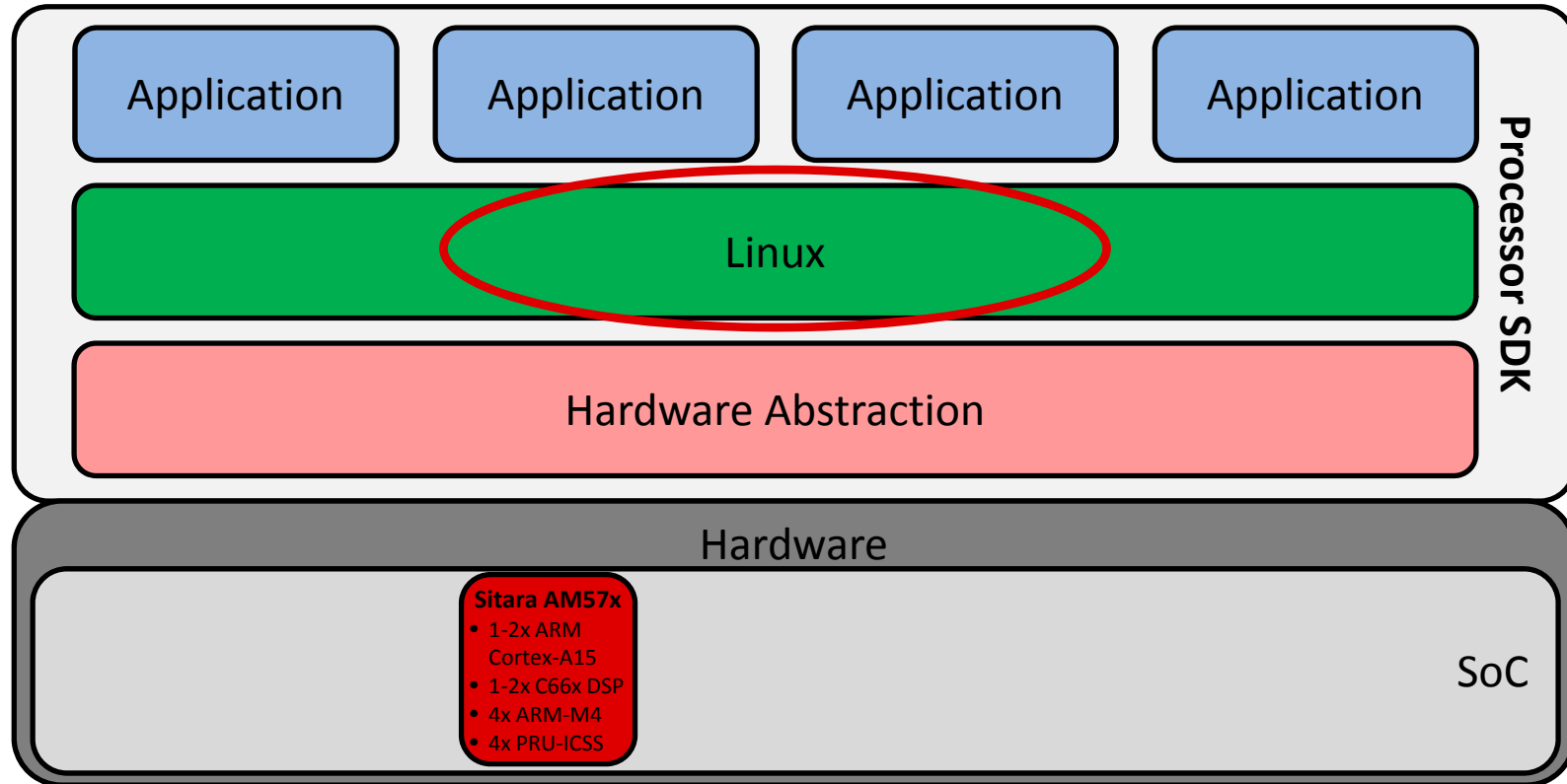


- Applications solve problems.
- Time-to-market and efficiency demand that application software be reused as much as possible.
- Lower-level software should enable this reuse.
- Developer experience and lessons learned should translate from project to project.
- Ideally, software reuse should scale across SoCs.

# Processor SDK



# Processor SDK Linux



# The Design Goals

Processor SDK Linux Overview



# What is an SDK for Linux?

Definitions and solutions vary. But we think an SDK should:

- Be more than just a board support package (BSP) or a demo, and therefore should also contain:
  - Tools for developing on TI parts
  - Pre-built libraries that customer applications can utilize
  - Documentation
  - Testing
- Provide a good, known starting point for product development:
  - Start with the SDK reference distributions.
  - Add and remove packages as needed.

# Processor SDK Linux Objectives

- The purpose of Processor SDK Linux is to provide a unique out-of-box experience and a quick path to application development.
- The Linux SDK accomplishes this by providing the following:
  - Example applications for key, high touch IP and peripherals
  - Tools for cross development and environment configuration
  - Host tools for device configuration
  - Documentation of SDK components
- Out-of-box in < 10 minutes and Development in < 1 hour
- A unified look and feel scaling across all TI Catalog Processor devices from ARM Cortex-A8 to Cortex-A15, with and without accelerators and co-processors, and beyond.

# Process SDK Linux Experience

Processor SDK Linux Overview

# Processor SDK Linux Experience

- **Matrix Application Launcher:**
  - GUI based on PHP and HTML5
  - Touchscreen, mouse and keyboard, web browser
- **Board Support Package (BSP):** Linux kernel, bootloaders, and file system
- **Documentation:** Quick Start Guide, Getting Started Guide, and Software Developer's Guide
- **Example Applications:** ARM benchmarks, power, multimedia, camera, Cryptography, USB, 3D Graphics
- SDK Installer to ease installation of all target and host software components and documentation
- **Scripts and Makefiles:**
  - Set up and configure system for development
  - Build an SD card image to run on a target board
- **Companion Tools:**
  - CCS
    - Linux-aware debug
    - Preconfigured projects for example applications
  - PinMux Utility
  - Flash Tools



# Summary

- Today's complex problems are demanding more efficiency and re-use from developers.
- Some of this efficiency is gained by using common lower-level components like operating systems (such as Linux).
- The Processor SDK Linux is designed to help new users get started – quickly and easily – by including everything needed for basic Linux development.
- For additional training on the Processor SDK, use the links on the next page.

# For More Information

- Processor SDK Training Series
  - Processor SDK Overview
  - Processor SDK RTOS Overview
  - Processor SDK Linux Components
  - Processor SDK Linux Matrix
  - Processor SDK Linux Installation
- Processor SDK Documentation
  - Processor SDK Linux Getting Started Guide
  - Processor SDK RTOS Getting Started Guide
- For questions about this training, refer to the E2E Community Forum:  
<https://e2e.ti.com/support>