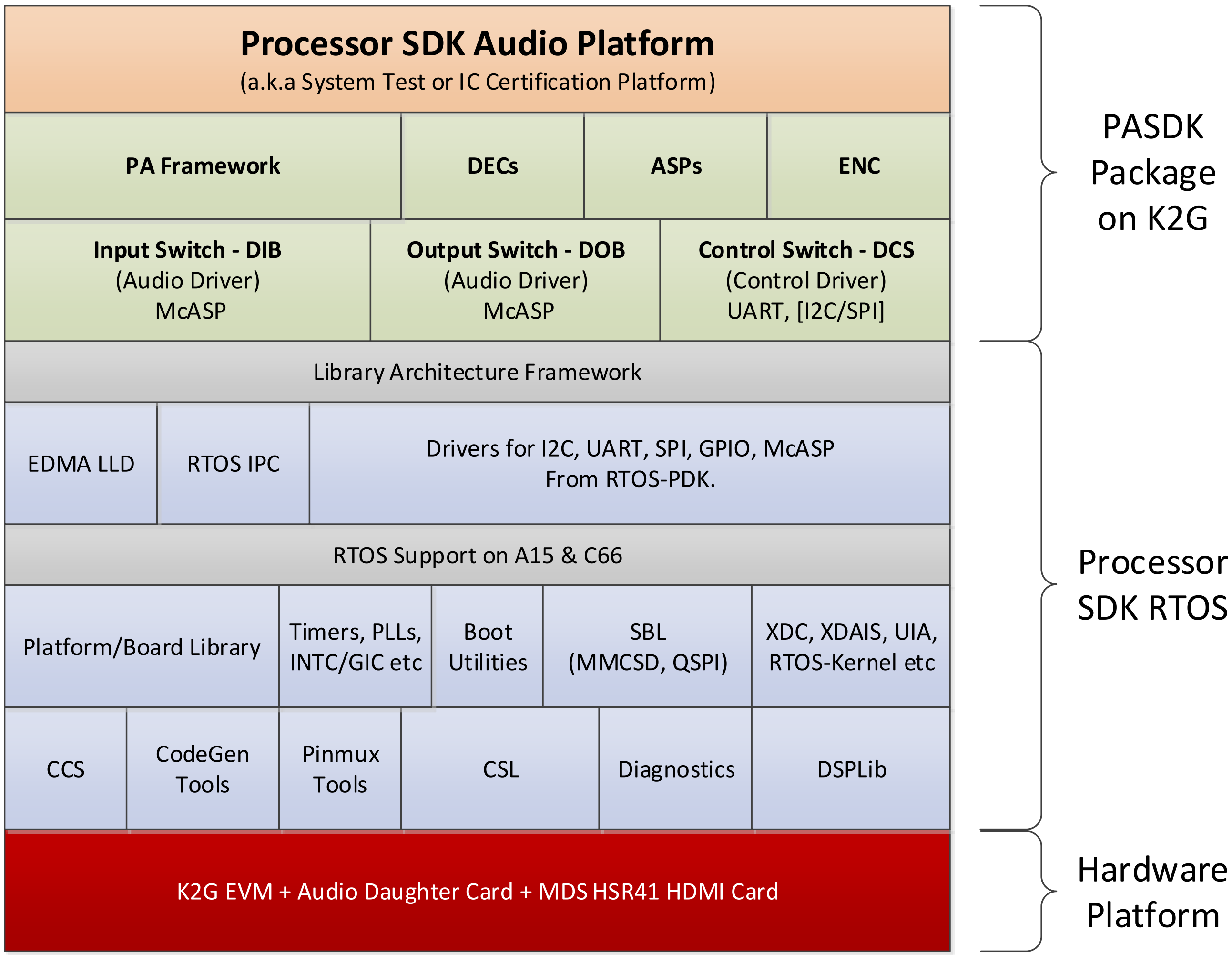
Processor SDK Audio

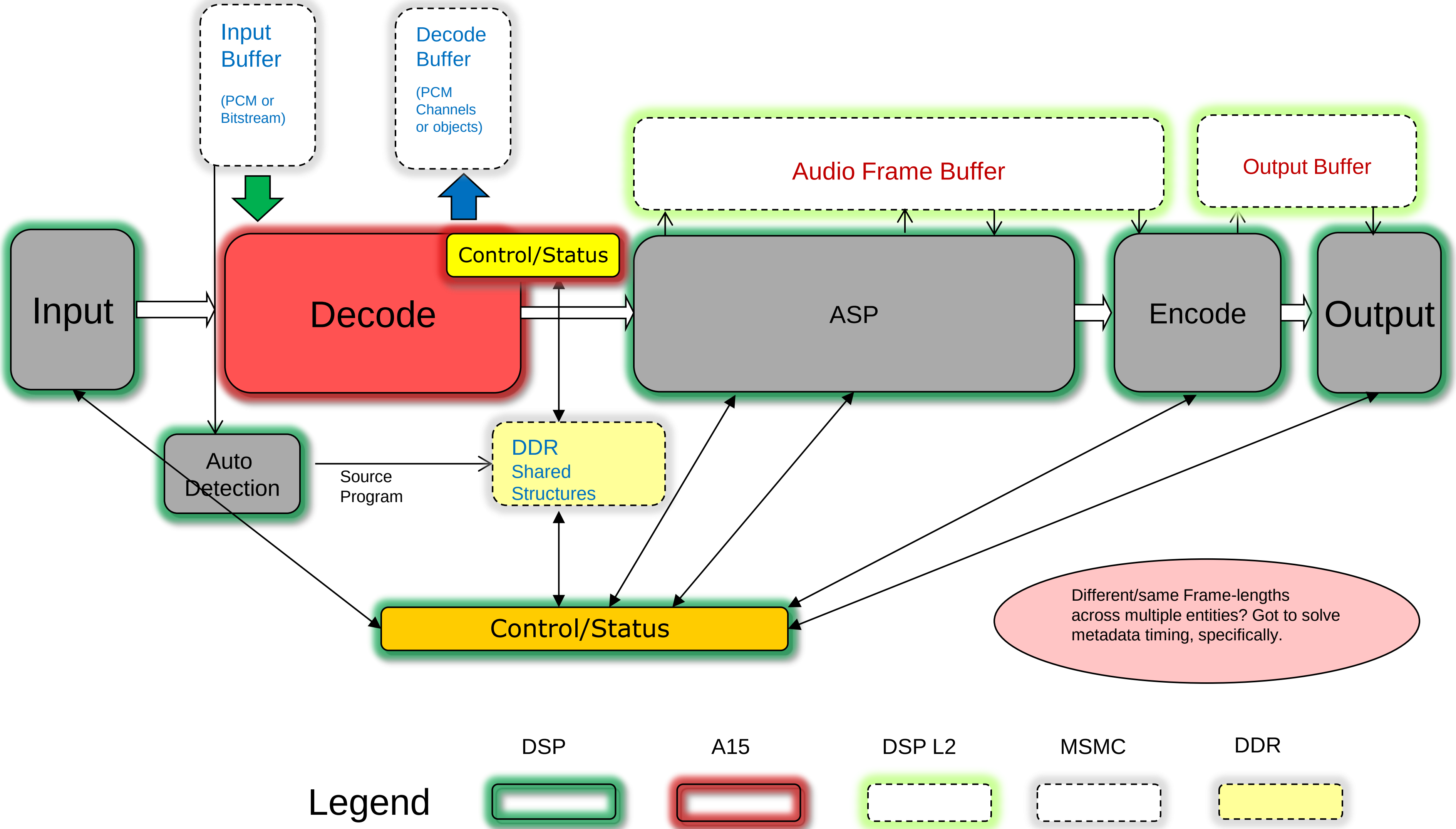
Getting Started Guide

9/8/2017

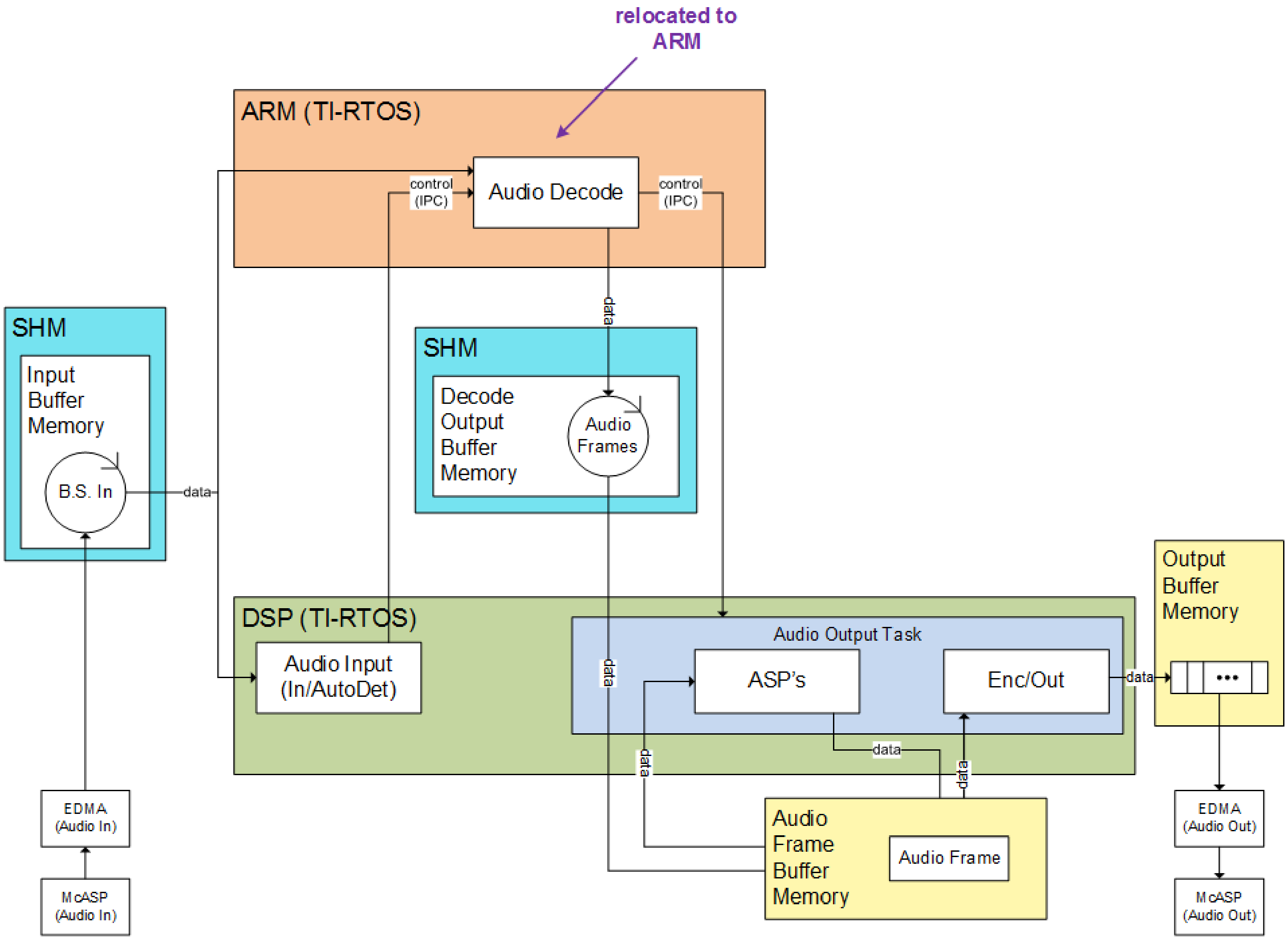
Processor SDK Audio Software



K2G: ARM+DSP Layout



Audio Tasks Layout



Adding New Components to PASDK (K2G)

- Decoders
  - Execute on ARM Core under TI SYS/BIOS (RTOS)
  - Must provide auto detection method definition and/or implementation for identifying the content type within an incoming bit stream which can be integrated into the real-time system.
- Audio Stream Processing (ASP) Components (e.g. audio rendering, filtering, etc.)
  - Execute on DSP Core under TI SYS/BIOS (RTOS)
- All components must provide interface that complies with the XDAIS API Specification
  - Standard specification has been extended for use in PASDK
  - IALG interface handles memory management and component creation
- All components must use Audio Frame Structure where required
  - Described in documentation and example ASP

Typical Development Steps

- Step#1: Implement component using standard C language
  - To make it possible to comply with the XDAIS API standard and PASDK extension make sure the code is re-entrant and it does not use malloc() or similar dynamic memory allocation functions.
  - Code should not depend on any operating system or any target system drivers
  - Organize your code in a modular way minimizing module coupling and maximizing module cohesion
  - Your functions should be implemented for real-time system (using signal frame durations that can be configurable where applicable resulting in reasonable signal/buffering latencies once integrated into the system)
- Step#2: Optimize your code for ARM or DSP core where it may execute
  - This may be achieved through creation of a suitable "intrinsic" library with functions that can be called from the components
  - Port "intrinsic" library to ARM and/or DSP
  - ***Optimize the functions in the intrinsic library for ARM and/or DSP core*** (use ARM or DSP intrinsic functions provided by code generation tools; do NOT use assembly language – EVER! (unless standard assembly optimized libraries were provided by the ARM or TI)

Typical Development Steps (cont.)

- Step#3: Create XDAIS wrappers
  - Using IALG specification and PASDK extension create wrappers that can be used for integration into the PASDK framework
- Step#4: Update Auto detection methods
  - If creating decoder component, modify the appropriate PASDK software (drivers) to include new auto detection method for recognizing the particular decoder content (currently this runs on a DSP and informs ARM to switch to using appropriate decoder component)
- Step#5: Integrate new components on ARM and/or DSP
  - Familiarize yourself with the operation of the PASDK framework (tasks, drivers, memory management, buffer management)
  - Take special care of properly handling circular buffer between decoder and the renderers/ASP's
- Step#6: Implement alpha-commands that can be used for configuration and real-time control of new components
  - If necessary update apply() method of PASDK XDAIS extension to detect changes in configuration and apply them prior to processing incoming data