Multimedia Framework Products

Nick Lethaby
Manager, Software Partners Network

Agenda

- What are TI Multimedia Framework Products?
- XDAIS & XDM overview
- Framework Components overview
- Codec Engine overview
- Block diagrams for select TI devices
- Linux Utilities overview
- Local Power Manager overview
- WinCE summary

What are Multimedia Framework Products?

- **Multimedia Framework Products (MFP) provide a low-level framework that provides remote or local instantiation of XDAIS-compliant algorithms and XDM-compliant codecs**
  - It can execute any XDAIS-compliant algorithm and is not limited to multimedia applications
- **MFP (in combination with 1-2 other products) are often referred to as the 'TI Multimedia Stack'**
- **MFP do not provide a full multimedia framework with services such as stream parsing or AV synchronization**
  - Such a framework (e.g. Gstreamer) calls the MFP to execute a particular codec or algorithm
- **MFP consist of several different software products that will be explained in a largely 'bottom-up' order**
  - A basic understanding of XDAIS/XDM is essential
  - These products may be used standalone to increase developer flexibility

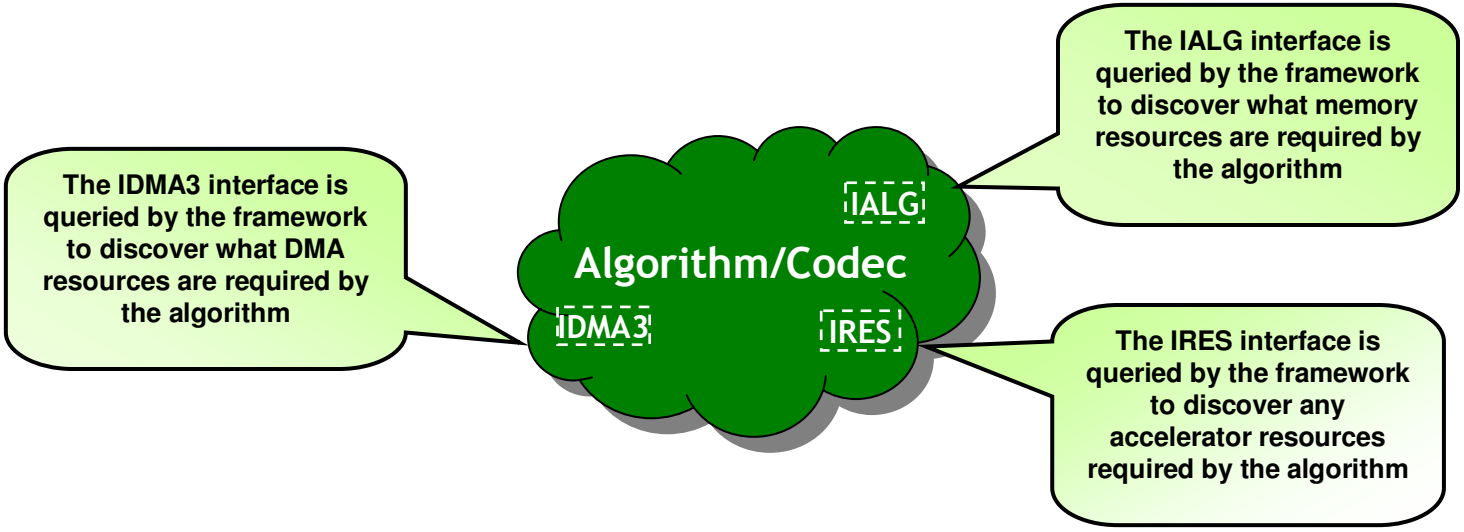
Overview of MFP Products

Component	Description
Codec Engine (CE)	The framework that builds on the low-level components to provide transparent remote and local execution of XDAIS- and XDM-compliant codecs. In addition, CE also performs some basic power management of a remote DSP core. CE bundles its dependent components (see below) in a directory called cetools.
Linux Utilities	These modules provide services that require Linux kernel components to perform memory allocation, DMA transfers, or interrupt handling.
Local Power Manager (LPM)	This module provides a Linux interface for power management of a DSP core from an ARM on ARM-DSP devices.
Framework Components (FC)	These modules provide off-the-shelf components for building a framework that correctly interacts with XDAIS-compliant aglorithms.
XDAIS	This SDK contains the documentation, header files, examples, and validation tools required to develop a XDAIS or XDM-compliant algorithm or codec.

DSP Systems
VOICE • VIDEO • AUDIO • BROADBAND • DSP

The XDAIS Standard

- **TI's XDAIS (eXpress Dsp Algorithm Integration Standard) standard simplifies integration of multiple algorithms**
  - Algorithms are not allowed to hardcode access to hardware resources
  - In lieu, they provide interfaces that may be called to determine which resources are needed
  - The application framework calls these interfaces and grants resources



The XDM Standard

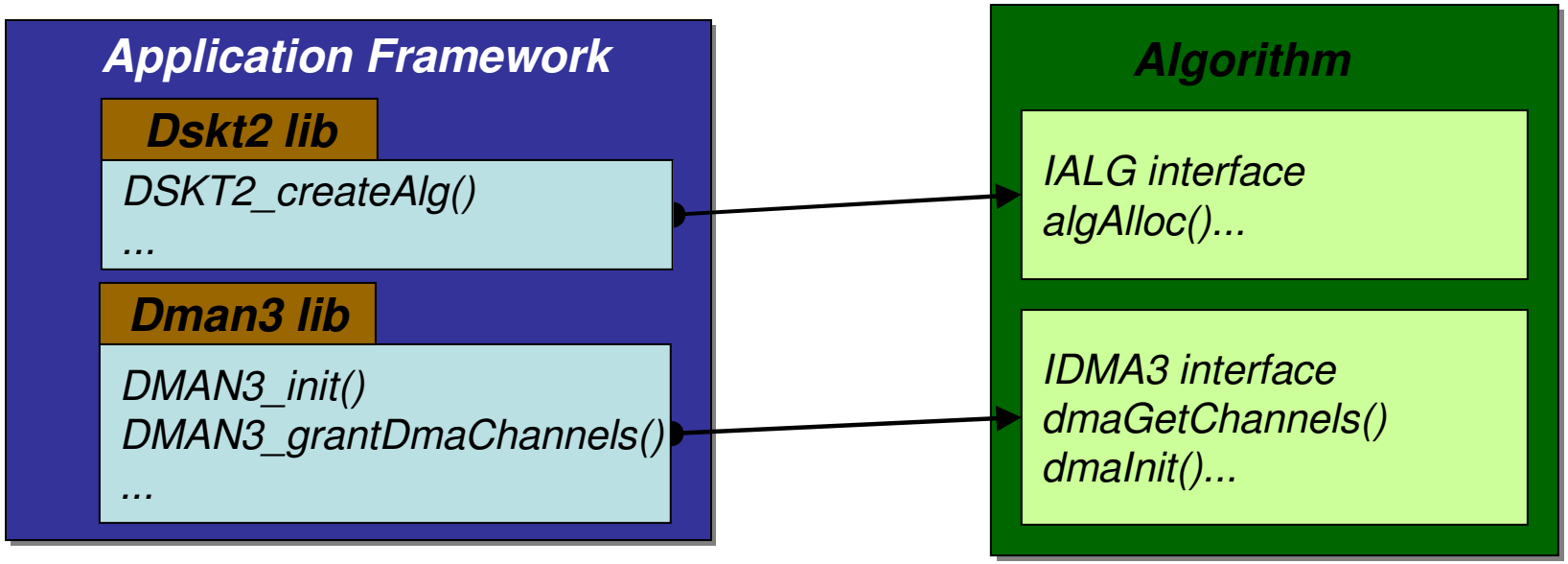
- **TI's XDM (eXpress Dsp Multimedia) standard standardizes APIs for common 'classes' of multimedia codecs**
  - Enables different XDM-compliant implementations to be easily swapped in or out
  - Enables the same application code to support multiple different codec standards (for example a digital music player that needs to play MP3 and WMA-encoded songs)
  - Included in XDAIS 5.x and above
  - These APIs are also referred to as the VISA (Video/Imaging/Speech/Audio) APIs

Multimedia Class	XDM (VISA) Interfaces
Video	IVIDDECx (decode), IVIDENCx (encode)
Imaging	IIMGDECx (decode), IIMGENCx (encode)
Speech (Voice)	ISPHDECx (decode), ISPHENCx (encode)
Audio	IAUDDECx (decode), IAUDENCx (encode)
Video Analytics	IVIDEANALYTICSx
Video Transcoding	IVIDTRANSCODEx
Arbitrary algorithm	IGENERICx

DSP Systems
VOICE • VIDEO • AUDIO • BROADBAND • DSP

Framework Components

- **Framework Components (FC) provide pre-tested components to facilitate development of frameworks using XDAIS-compliant algorithms**
  - Each XDAIS-interface that can be exported by an algorithm (IALG, IDMA3, IRES) has an equivalent FC module
  - The primary role of the FC modules is to query the algorithm about its resources and then grant them

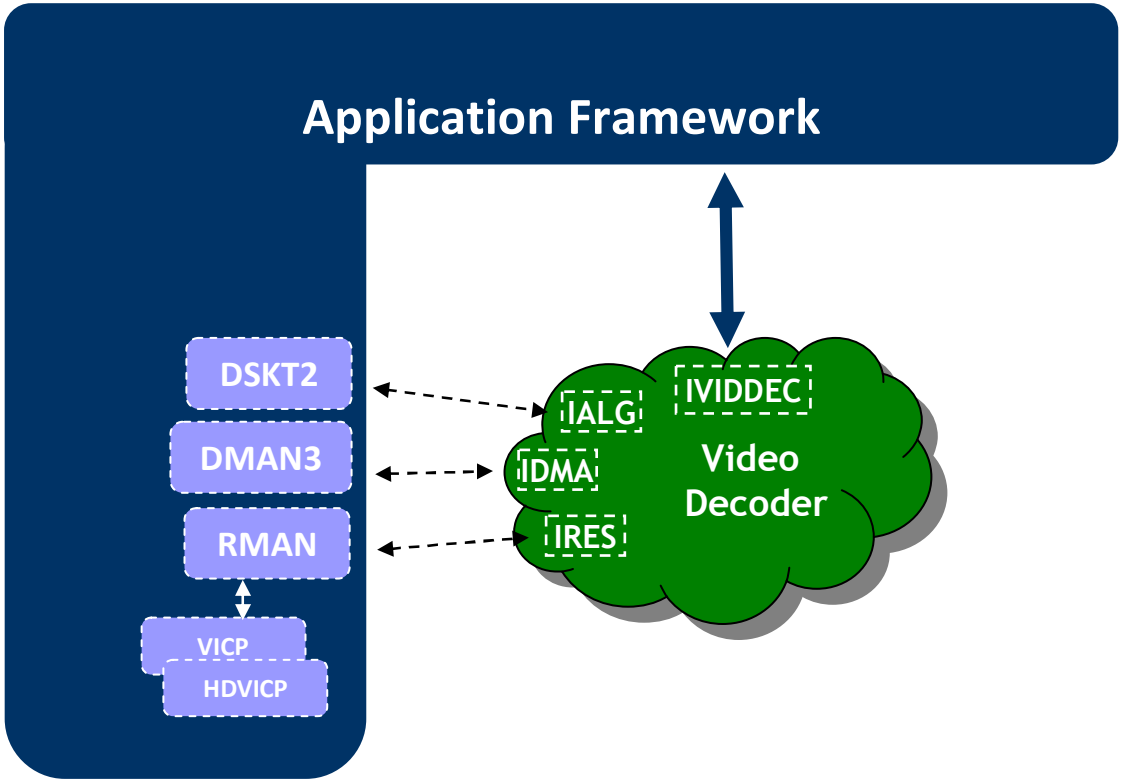


Framework Components

Component	Description
DSKT2	DSKT2 allocates memory resources algorithms. DSKT2 queries an algorithm's IALG interface. It ensures algorithms only acquire free or scratch memory resources. DSKT2 support different memory types, such as scratch and persistent, and enables scratch memory to be efficiently reused by multiple algorithms.
DMAN3	DMAN3 is the manages DMA resources for the C64x+ EDMA3 hardware. DMAN3 queries an algorithm's IDMA3 interface. It ensures that algorithms only acquire free DMA resources. Note that DMAN3 primarily deals with allocation of QDMA channels. For sharing EMDA3 channels directly, use RMAN/IRESMAN. DMAN3 is compatible with the EDMA3 driver delivered in the Platform Support Packages for DSPs.
RMAN	RMAN is an abstraction layer that is required because many devices have multiple resource types that need to be managed. RMAN redirects specific resource management to IRESMAN components.
IRESMAN	IRESMAN manages a specific resource such as EDMA3 or the VICP accelerator. Each resource will have its own IRESMAN implementation and a device will often have multiple IRESMAN components (for example IRESMAN_VICP and IRESMAN_EDMA3). IRESMAN for EDMA3 is compatible with the EDMA3 driver delivered in the Platform Support Packages for DSPs.
ACPY3	ACPY3 does not support any specific XDAIS-interface. It is a convenience function that provides fast memory-to-memory copies (as often required by video codecs) using DMA. It requires the DMA channels to be allocated via DMAN3.
SCPY	SCPY is a convenience function that provides Linux applications with fast memory-to-memory copies using DMA. It requires the SDMA Linux Utility.

DSP Systems
VOICE ▪ VIDEO ▪ AUDIO ▪ BROADBAND ▪ DSP

XDAIS, XDM & FC Summary



Codec Engine Overview

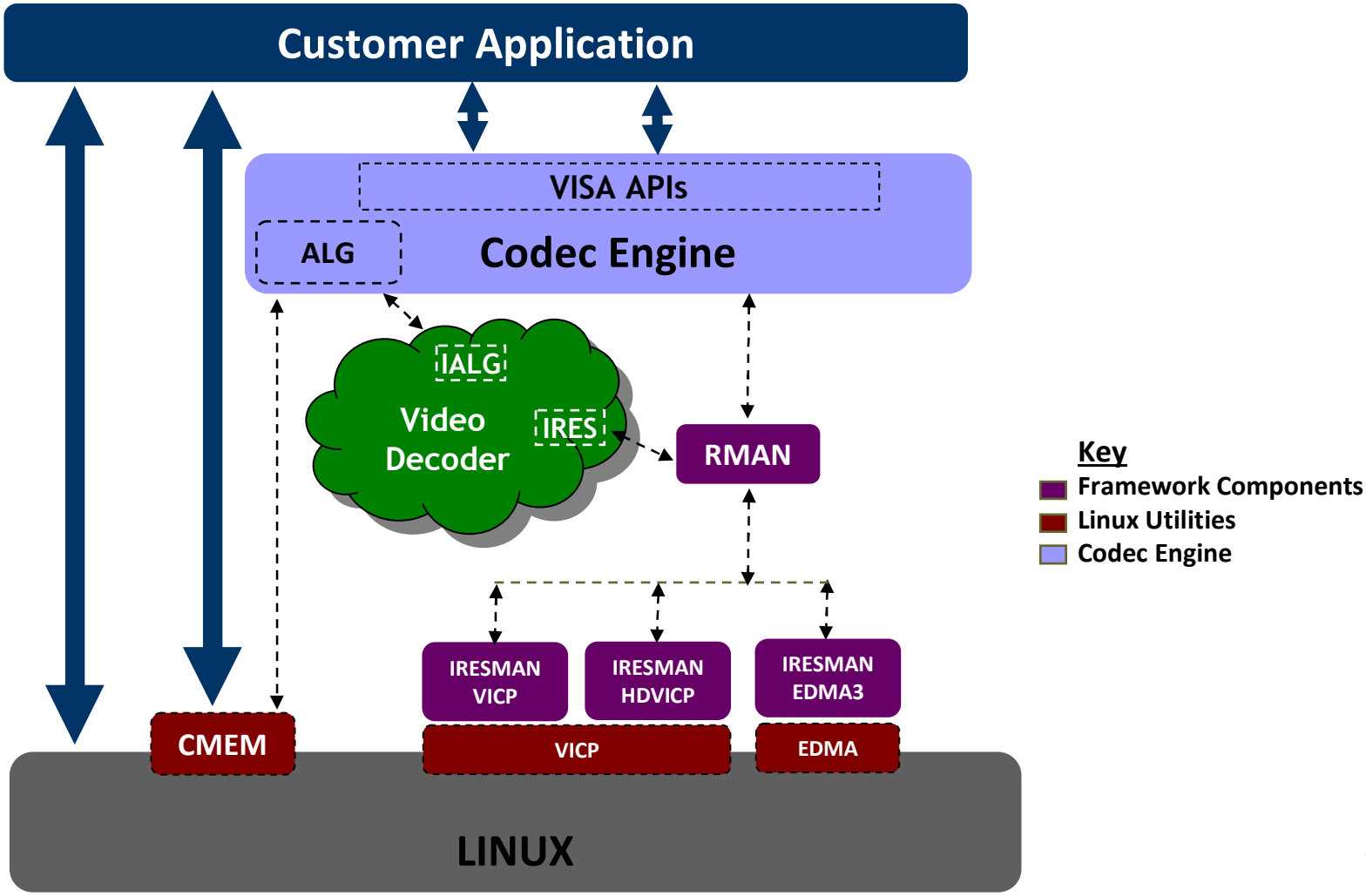
- **Codec Engine is a low-level framework designed to provide transparent execution of multimedia codecs and algorithms on TI devices**
  - It enables an application to create and execute a multimedia codec using the VISA APIs either locally or remotely
    - ARM-only, ARM+accelerator, ARM+DSP, and DSP-only devices are all supported
  - It automatically handles allocation of memory, accelerator, and DMA resources for the algorithm as well as any remote thread creation
  - Multimedia codecs must be XDAIS/XDM compliant and also conform to additional packaging standards to work in Codec Engine
  - Arbitrary algorithms that comply with XDAIS and the packaging standards are also supported through a generic interface
  - For ARM-DSP devices, it also provides power-on/off and hibernate/resume capabilities for the DSP core

Codec Engine Dependencies

- **In addition to requiring XDAIS/XDM-compliant codecs and FC, Codec Engine may require all or some of the following:**
  - Linux Utilities: These provide user-space access to hardware resources managed by the Linux kernel
    - These are required by ARM-only, ARM+accelerator, or ARM+DSP devices
  - Local Power Manager: Enable power management of the DSP core from Linux
    - LPM is relevant for ARM+DSP devices only
  - DSP/BIOS Link: Enables ARM to bootload DSP core and provides communication services between the ARM and the DSP
    - DSP/BIOS Link is relevant for ARM+DSP devices only
  - DSP/BIOS: Provides basic RTOS services on the DSP core
    - DSP/BIOS is relevant for ARM+DSP and DSP-only devices only

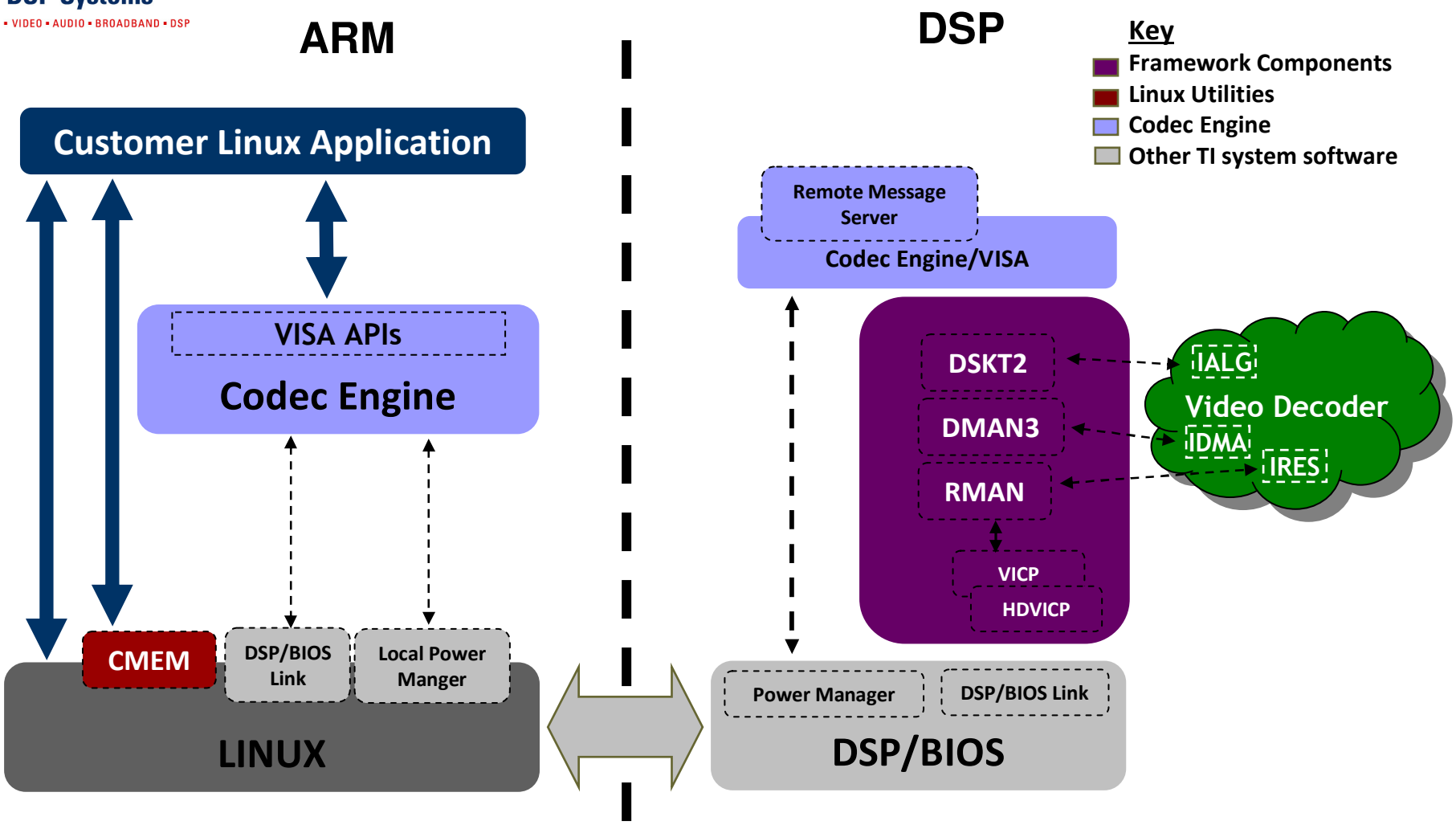
DSP Systems
VOICE ▪ VIDEO ▪ AUDIO ▪ BROADBAND ▪ DSP

DM365 MFP



DSP Systems
VOICE • VIDEO • AUDIO • BROADBAND • DSP

OMAP3530/DM6446/DM6467 MFP



Linux Utilities

Component	Description
CMEM	CMEM is a contiguous memory manager for Linux and ensures that large contiguous blocks of memory can be allocated (e.g. for video buffers). Contiguous memory is required because the DSP or accelerator processing the buffer lacks an MMU to handle a fragmented buffer.
SDMA	SDMA (System DMA) is a peripheral accessible from the ARM core on OMAP3x devices. The SDMA utility library provides the ability for user-mode applications to request SDMA channels and operate on a channel using direct, memory-mapped access to the channel's DMA registers. This library is used by the SCPY function (see FC slide) in ARM-resident multimedia codecs. Since TI does not directly provide ARM-side codecs, TI's DVSDK MFP stack configurations do not use SDMA.
EDMA	The DM365 and DM355 use EDMA3 to provide DMA services to applications on the ARM core. The EDMA utility library provides the ability for user-mode applications to request EDMA3 channels and operate on a channel using direct, memory-mapped access to the channel's DMA registers. IRESMAN_EDMA3 uses this component when allocating EDMA3 resources.
VICP	The VICP utility library provides user mode access to the accelerators from ARM Linux codecs on the DM365 and DM355. This utility also handles the accelerator interrupts in the Linux kernel. The VICP utility library handles all the different accelerators on the DM3xx devices.

Local Power Manager Overview

- **In ARM-DSP devices, the ARM is often the focal point for power management:**
  - It usually best knows when the DSP will no longer be required. For example, if it has passed down the last AV buffers of a video clip.
  - It may have sole access to key power management hardware and therefore the DSP needs to request the ARM to assist any power downs
- **The Local Power Manager (LPM) provides user-space APIs to Linux applications (such as Codec Engine) that wish to manage the DSP core's power**
  - The DSP is treated as a peripheral by Linux and therefore cannot be directly accessed by an application
  - LPM typically supports turning the power on/off or hibernate/resume of the DSP core

WinCE Support

- The MFP is also available on WinCE 6.0
- WinCE versions of the products are BSD licensed and do not contain GPL code
- WinCE versions include support for platform builder