

The Sixteenth Annual Interactive Audio Conference

PROJECT BAR-B-Q 2011



Group Report: What's Next for HD Audio?

Participants: A.K.A. "Screaming Monkeys"

Kevin Brennan, Dolby

David Roach, Optimal Sound

Devon Worrell, Intel

Josh McMahon, Realtek

John Howley, Conexant

Dan Bogard, IDT

Doug Peeler, Dell

Jens Puchert, Cirrus

Facilitator: Linda Law, Fat Labs

The Problem

The audio architecture that is emerging to address the convergence of portable computing devices borrows from different legacy architectures from the PC and Mobile handset industries and is largely market driven. These architectures are not necessarily built in a technically optimal and efficient way but are rather an amalgamation of a legacy PC audio architecture (pre-HD Audio) as well as a legacy mobile architecture, resulting from the convergence of portable PCs and media-tablets. Some of the challenges of this emerging architecture include:

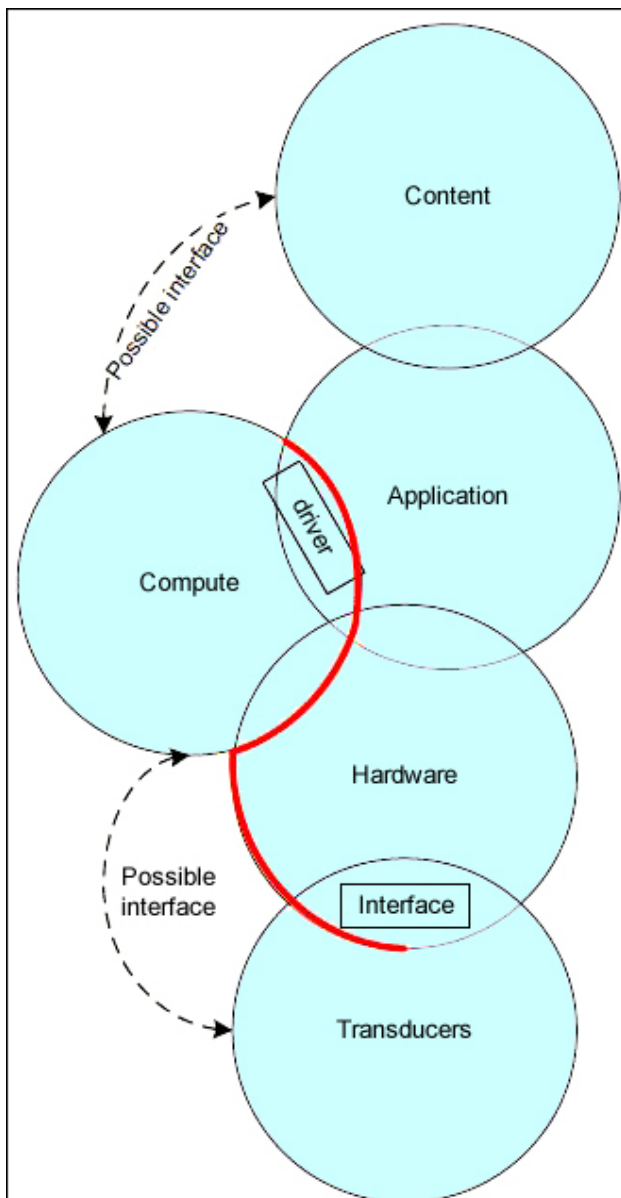
1. There is no standardized framework, recommendations, and little documentation on how to implement the audio subsystem. This has the following implications:
 - a. The entire ecosystem that had been established is now discarded along with the value that was created
 - b. Each system design has to be conceived as a custom solution from top to bottom
 - c. It will be much more difficult for players to differentiate and scale their solution and it's also much more difficult for multiple entities to cooperate in an efficient way in providing modularized components
2. Stringent power requirements, with a strong focus on idle power and lesser focus on active power states.
3. Increased costs (per unit component and one time development costs), which is likely to be passed to the end consumer.
4. In order to support legacy and the next-gen OS, OEMs will be required to maintain and support different architectures.
5. System effects are replaced with a vendor-specific DSP compute node. The DSP will contain the same attributes that existed in the existing HD Audio PC architecture.
6. Unlike in HD Audio, there is no standardized hardware interface and this has led to fragmentation and increased complexity of the hardware codec and bus interconnect.
 - a. Driver support is largely undefined and is specific to each compute solution.
7. Multiple busses are required including separate busses for data transfer and control. An audio subsystem may run on three or more independent busses, some of which are not dedicated to audio.
 - a. Schematic information is captured in ACPI tables in the BIOS resulting in increased pin count and other complexity
 - b. Other potentially viable solutions such as PCIe and USB have been excluded in the favor of

antiquated solutions (such I2C, I2S, GPIO, UART, etc.)

8. Known future OS certification requirement include a failover mode, which results in two separate audio systems on the product as well as a hardware loopback requirement.

The group's solution to this problem

In order to address the needs of modern low power, portable consumer computing devices (including portable PCs, media-tablets, etc.), a structured, scalable, and modular approach is required to audio. The working group investigated a framework and direction that was generic and centered around a discrete number of building blocks and interfaces. Each building block serves a specific function or group of functions within the overall framework and is connected to other blocks by a well understood interface.



The group believes that a modular framework will allow for more isolated development within the framework and enable a lightweight development process (for example, the PC OEMs will not have to busy themselves with unnecessary "oversight" activity to ensure each part of the system interoperates).

Each "block" within the framework is modular, self contained and serves a specific set of functions. Other characteristics of the framework "blocks" are:

- Independent
- Discoverable
- Scalable
- Provides power management
- The functionality of multiple blocks can be collapsed into a single "block"
- Blocks can be local or remote

Block Definitions

The group proposed the following "blocks" be part of the overall framework:

- **Content:** Consists of both the content itself as well as items that describe the content including metadata. It is not the traditional PCM.
- **Application:** In the emerging paradigm, the functionality of Application has been subsumed by the OS and Compute block. Application is simply the customer facing UI and in many ways is just a conduit between the content and the compute block.
- **Compute:** Compute is a new hardware offload that takes on the traditional role that was provided by system effects and drivers. It is not host software.
- **Hardware:** Hardware is the mixed-signal electrical interface between Compute and the transducers.
- **Transducers:** Transducers are the acoustical

elements in the system, and can serve as a bridge to a digital interface such as HDMI.

Interfaces

Connecting each "Block" in the framework is an Interface with well-defined characteristics, which allows Block developers to focus on their specific function within the overall framework and not concern themselves with the detail of the implementation of other Blocks. An interface exists between each adjacent Block in the framework. The key interfaces along with their characteristics (and other notes from the workgroup) are shown below:

- **Compute/Application Interface**
 - Format Discoverability (both directions)
 - The application can query for supported formats (optional). Otherwise, it will pass to the Compute/Application layer the audio stream. The Compute/Application layer will detect what the format is, treat it appropriately, and then, if the format can not be handled, pass back to the application that there is a problem.
 - The application should tell the Compute/Application node what its preferred rate for making content available (consumption rate) is
- **Compute/Hardware Interface**
 - Discoverability
 - Scalability (rates, number of streams, bit depth, simultaneous streams, number of devices)
 - Power management
 - Granular
 - Dynamic
 - Protectable
 - Monolithic
 - Clocks
 - Command and control
 - Data
- **Application/Hardware Interface**
 - This is the failsafe in the sense that it is for specialized hardware
 - Proprietary
 - Non-standard
 - For outliers
- **Hardware/Transducers Interface**
 - Jack detection
 - Interface
 - Analog
 - Digital
 - Standardized Building Blocks
 - Speakers
 - Headset Jack
 - 3.5mm Individual Jack
 - Analog Mic
 - Digital Mic
- **Transducers/Compute Interface (not main interface, but instead an example of combining interfaces)**
 - Useful when transducer is integrated with hardware
 - Useful when compute bypasses hardware.
 - Interface properties of both Compute and Hardware still exist in a single monolithic block
- **Content/Compute Interface**
 - The Compute Block tells the content what its preferences are
 - The Compute Block communicates directly with the content in encryption/decryption scenario.

if an application is present then it simply serves as a conduit or is bypassed entirely.

- **Content/Application Interface**

- Legacy Method

An Example Implementation

The workgroup started to define a potential implementation of the new framework that would be an extension of HD Audio. A first cut at a block diagram of this implementation is shown below.

