Whitepaper

# Five Key Criteria for Choosing the Right Audio Codec Implementation

**By Reshma Rai, Senior Manager, Media Client Technologies, Ittiam and**

**Saketh Sathuvalli, Senior Engineer, Media Client Technologies, Ittiam**

October 2017

With the growing emphasis on outstanding audio experiences across electronic applications, the need for deploying high performance software audio codecs gains more importance than ever. Popular codec formats are often deployed across multiple products with varying processor architectures within a single organization. System engineers thus face the overwhelming challenge of identifying the right implementation for their applications.

Drawing upon our extensive experience in customer deployments of a wide array of codecs across a range of applications including mobile, broadcast, automotive, defense, industrial and wearables, we bring to you this comprehensive Assessment Guide. This document provides a high level overview of codecs and outlines the factors to be considered during the selection of audio codec implementations or the codec vendor. By leveraging this guide, engineers can ensure ease of integration and faster development and deployment of a product. The whitepaper also explains the impact of these factors with suitable examples to help engineers customize the assessment process to their product needs.

# Choosing the Right Audio Codec

**Software audio codec,** as you must be aware of, is a piece of program that implements an algorithm to achieve compression and decompression of audio data. While a wide range of algorithms and their corresponding implementations are available, some of the most popular ones include MP3, AAC, WMA, FLAC, SBC, OPUS, Dolby AC3, Vorbis, and ALAC. Furthermore, the audio codec implementations are available for a wide range of platforms including ARM, X86, MIPS and DSPs. This explains why effective evaluation of audio codecs is not a simple and straightforward exercise.

> High performance software audio codecs are like the 'engine of a car', propelling the creation of reliable applications that deliver outstanding audio experiences.

While the choice of a codec will be governed by the larger product ecosystem, the choice of the specific codec implementation is in your hands. As system/application engineers, your best way forward is to weigh a combination of multiple influencing factors in the light of your unique application and platform requirements and priorities. This will help you arrive at the 'single best audio codec' that fulfills all your requirements and thus qualifies to power your product.

The following are some of the chief criteria we have identified for you to consider, while selecting a codec for deployment in a particular audio application or use case.

1. **Features and platforms**: The range of audio codec formats and platforms
2. **Performance**: Impact of performance on product specifications and functionality
3. **Quality and compliance**: Assessment of quality and relevant measurement tools, metrics, and compliance
4. **Integration**: Ease of integration, readiness for frameworks, OS platforms, etc.
5. **Reliability and support**: Factors that affect the product during development and deployment

# Key Selection Criteria

## 1. Features and Platforms

Most audio codec implementers support all the standard features of codecs. However, some of the specialized features their codec library is expected to support typically depends upon the **end application where the library is to be deployed.** For instance, an AAC codec library implementation by a vendor, who caters to the consumer market segment, might not support Low Overhead Audio Transport Multiplex (LATM) streams. It might also not comply with the Association of Radio Industries and Businesses (ARIB) standards. But these two features are essential for you if you are working on broadcast applications.

The second important factor is the availability of codec implementations on **popular hardware platforms.** Consider the use case where you are a manufacturer of wearable devices for the consumer market. Although your product may use a low power ARM Cortex M processor, you are likely to design a host application on mobile, PC or tablet that connects to it, enabling music streaming from the host to the wearable. In addition to selecting the audio codec on the wearable device, you also need to use complimentary codecs on ARM and x86 processors that run the host application. Therefore, the codec vendor that you plan to work with should offer **wide platform coverage.**

Furthermore, your requirement might be to build a family of products, comprising a larger ecosystem of software and running on multiple processors. Since the diversity in content ecosystems also translates in to diversity in the audio codec formats that they support, you may need to support more than one codec format. Your best approach therefore is to partner with a vendor who can support a range of codecs on all the target platforms. Bundling a larger set of libraries together also helps reduce the total cost of licensing from most vendors.

**Quick Checklist – Features and Platforms**

1. **What are the codecs that you need to support in your products or services?** Consider the wider ecosystem and a slightly longer term view in arriving at the list.
2. **Have you identified the unique codec features specific to your application?** Industry specific compliance requirements are a good starting point to explore the feature list.
3. **Are you working on a single processor, a processor family, or multiple processor families?** Review the product roadmap of those processors to be aware of upcoming architectural changes.
4. **Do the vendors support the formats and features on your platforms of interest?** Also consider the readiness and projected timelines for any gaps in features, formats or platform support.

## 2. Performance

The performance of a software audio codec is often the most critical parameter to evaluate different implementations. The performance parameters typically used for comparison include the following:

- Number of cycles consumed by the codec library to process one second of audio data.
- Usually expressed in Million Cycles Per Second/Million Instructions Per Second (MCPS/MIPS) and often equivalent to the processor clock units in Mega Hertz (MHz).
- The amount of system memory that the codec libraries require for processing.

These parameters are usually listed in a codec datasheet. However, numbers in datasheets can be misleading if the following secondary factors are not considered:

- Whether the performance numbers have been measured using a simulator for the platform or on the actual device. A simulator environment has the advantage of 0 wait state for memory access, thus enabling 50-100% better performance numbers than the actual numbers on a given platform.
- The audio content of the stream for which the numbers are presented by the vendor. For instance, a stream with more music content and less silence will need more processing power than a stream with less music content and more silence.
- I-Cache and D-Cache values of the hardware platform on which the performance of the implementation is evaluated.
- MCPS measurements method used in a cache configuration. Since audio, video and other file parsing operations are interweaved in a typical system configuration, the cache is thrashed across audio, video and other system calls. However, vendors might not accurately reflect this scenario

in their datasheets. It is therefore essential that vendors account for audio codec numbers based on a setup where the cache configuration is thrashed or cleaned up after every process call.

While the average MCPS determines the steady state performance, very high peak to average ratios can often make the systems non-real time, leading to audio artifacts and system failures. Higher peak MCPS numbers will also require system designers to introduce higher latency and larger buffers in the audio pipeline to smoothen fluctuations – leading to additional latency and memory requirements that might be undesirable in several applications.

The performance advantage also varies across target platforms. Consider codecs from two vendors with performance numbers of 13 and 26 MCPS. The difference in performance may not be very significant on a multi-core processor that is clocked at 1.5 GHz. But, on a low power wearable device with Cortex-M4 processor running at 96MHz, the difference results in significant extension of battery life. We therefore recommend that you directly evaluate the performance of codecs on your platform of choice, instead of solely relying on vendors' datasheet figures.

**Quick Checklist - Performance**

1. **What is the codec performance on the actual target platform with complex audio inputs?** If you cannot measure this, request the vendor to provide performance measurements on a platform closest to the processor and memory configuration of your target platform.
2. **What is the application headroom required in your product?** Ensure that the codec performance syncs comfortably with rest of the system.
3. **What is the peak to average performance ratio across a variety of complex content?** Consider the peak MCPS (and not average) for decisions on system design.

# 3. Quality and Compliance

Most popular codecs are specified by standards bodies like ISO/IEC, ITU, MPEG and the specifications also cover detailed compliance requirements. So, a standards compliance report covers several bases in assessing a codec implementation. Even proprietary codecs from corporations like Dolby and Microsoft clearly specify compliance requirements. However, you also need to consider a few additional factors beyond the standard compliance tests:

- Typical compliance for decoders rely on the Root Mean Square (RMS) error – any implementation should be within a certain threshold of error when compared with the reference implementation. Although all compliant implementations may be under this threshold, minimizing this error can provide additional differentiation in quality.

- Compliance tests specify a comprehensive set of test content that assesses the codec's handling of varied test patterns. Compliance streams also get updated on a regular basis for any standard driven addendums or corrigendum. It would be prudent to check for the appropriate compliance streams as well as the version of compliance streams tested.

- While standard compliance tests cover the implementation for its completeness, they typically do not assess error behavior. Additional test patterns that include multiple erroneous test streams will help validate if the codec library provides a graceful exit with appropriate error codes.
- Encoder implementations like AAC use EAQUAL for measuring the quality of custom implementations for test vectors provided by SQAM. Similarly, Dolby that owns codecs like AC3 and EAC3 specifies a standard set of tests to consider an implementation acceptable, including a certification procedure for custom implementations of its proprietary codecs. Although thresholds are specified, the absolute measurements obtained using such quality tools can be used to differentiate an implementation that delivers average quality from one that delivers outstanding quality. Subjective tools like MUSHRA can be added to the quality evaluation process to provide further insights into the audio quality of various implementations. Additionally, commercially available tools can be used to measure Total Harmonic Distortion (THD), which reflects the quality degradation or distortion in encoding compared to the input samples.

**Understanding the quality components**

**Losses:** The losses introduced in audio compression are mainly of two types – algorithmic losses and precision losses. While reducing precision losses is well within the realm of a codec implementation, developers and implementers have little control over algorithmic losses, as these are determined by the codec format and standard. Algorithm losses are unavoidable in the case of a decoder as it is governed by the encoder behavior. Similarly, in an encoder, the algorithmic losses arising out of the normative elements are beyond our control since they are governed by the compression tools used in the specifications. However, we have more control over the losses in an encoder, originating from non-normative elements.

**Fixed or floating point implementation?** Another critical quality factor is whether the codec implementation is fixed or floating point. Fixed point implementations have been very popular in embedded platforms. Although these deliver lower precision when compared to floating point implementations, they typically deliver higher speed. But in recent times, we have seen the rise of next-generation processors with more efficient floating point arithmetic capabilities. This has made floating point implementations more viable in embedded systems, enabling higher precision and therefore better quality.

**Quick Checklist – Quality & Compliance**

1. **Have you reviewed the compliance report in detail?** Pay attention to the absolute RMS errors as well as objective and subjective quality measurements to differentiate between implementations.

2. **Do your vendor's test reports include results with erroneous streams?** Explore how these streams were generated and perform additional testing at your end by introducing random errors in the streams.

3. **Does your hardware have floating point support?** If quality is the most critical factor, consider evaluating a floating point implementation to assess the speed-quality trade-off.

# 4. Integration

Audio codecs do not operate in isolation and are typically used in multimedia frameworks to realize systems and applications. Whether it is used with popular frameworks like Gstreamer and libstagefright, or proprietary in-house frameworks, three factors are critical in assessing the ease of integration of a codec implementation.

- Well defined APIs – with readily available plug-ins for popular frameworks
- High quality API documentation for integration with proprietary frameworks
- APIs that offer higher degree of flexibility and control to accomplish specialized functions beyond what is supported natively in popular frameworks

With plug-ins, implementers can limit validation to a stand-alone or unit test. But as system designers would realize, a thorough integration test or even a full system test along with video and other functions goes a long way in eliminating functionality and performance related issues. So, we recommend that you partner with vendors who have proactively validated the plug-ins at a system level, thereby significantly reducing problems for integrators.

**Quick Checklist - Integration**

1. **Do your vendors have ready plug-ins for your multimedia framework(s) of choice?** Consider a vendor with wider framework options to accommodate future changes in your design.
2. **How well are the plug-ins validated at a system level?** Pick vendors who have extensively validated their plug-ins either in-house or in customer deployed systems.

# 5. Reliability and Support

There are many applications where a single product or a product line uses the same codec implementation over several years and sometimes decades. In addition to performance, quality and integration – reliability of the product and the vendor and extended support from the vendor become critical factors for continued success of the product or product line. Being soft factors, these are sometimes the hardest to assess, and hence ignored during the evaluation process.

A sure-shot way to assess reliability of the codec is through very rigorous testing. It is often possible to do this during the evaluation phase or by provisioning suitable acceptance clauses in your contracts. However, this requires extensive knowledge about testing audio features. If you have inadequate in-house expertise, you can still evaluate reliability by checking if the vendor's codecs have been deployed in products similar to yours or in industries where reliability requirements are more stringent. These could include automotive applications where a single product is designed to outlast the car in which it is deployed, or mobile processors with very large deployment volumes.

In addition to product reliability, consider the following scenarios where vendor reliability and support are critical:

- Two years into your product development, a new processor or processor architecture is added to the mix and the codecs need an upgrade.

- Even after thorough testing, you encounter a specific issue in field deployments and need help in identifying and fixing the issue in quick time.
- There is a new codec format that your product now needs to support. Or a new profile or feature of a current codec that you now need to support.

**Quick Checklist - Reliability & Support**

1. **Are you satisfied with the vendor's quality of documentation and support during evaluation?** Consider the product evaluation as a trial engagement with the vendor and pay attention to the non-technical aspects of engagement.
2. **Have you assessed the vendor's track record, customer success stories and references?** Although a very traditional model of vendor selection, this still works.

# Case Studies

Now that we have established a framework to assess implementations, we can gain better understanding of how it works through a couple of real world examples.

## Evaluating an SBC Codec for Wearable Products

Here is a look at how a leading wearables brand evaluated and chose us for our SBC codec implementation, based on the key evaluation criteria we have defined in this paper. The competition offered an Open Source implementation.

A simple codec with low algorithmic complexity, SBC is a popular choice for low power wearable devices.

### 1. Features and Platforms

The customer required an SBC codec only for the wearable device based on Cortex-M4 processor. However, the eco-system for the product demanded the codec to be present on multiple host applications:

- SBC decoder Linux build on Cortex-A8 (Host)
- SBC encoder for Windows and MAC OsX on x86-64bit platforms (Host)

In addition to the standard SBC features, the customer also needed to support specific features for Universal Windows Platform (UWP). This was the only missing feature from us, which we plugged and delivered in a very short span of time.

Since the codec was available in open source, the platform support from both Ittiam and competition was not a highly critical factor for this customer.

### 2. Performance

High performance was a critical factor for the client as it enabled longer battery life of the wearable product. Further, the performance of libraries on the product device (Cortex-M4 based processor) was in focus, while the performance of the host applications was not very critical.

Table 1 shows almost a 2x performance advantage, which tilted the decision significantly in favor of Ittiam.

| | Ittiam | | Open Source | |
|---|---|---|---|---|
| **Stream Properties** | **Avg MCPS** | **Peak MCPS** | **Avg MCPS** | **Peak MCPS** |
| 4 sub bands 4 blocks per frame | 26.50 | 28.03 | 50.08 | 52.86 |
| 4 sub bands 16 blocks per frame | 16.37 | 17.47 | 35.12 | 37.8 |
| 8 sub bands 16 blocks per frame (the most popularly used configuration) | 15.23 | 16.32 | 34.18 | 37.8 |

*Table 1. Ittiam vs Open Source: Performance comparison*

## 3. Quality and compliance

Both vendors had a standard compliant version of SBC. However, our testing included several well designed erroneous streams in addition to the conformance streams to provide a higher degree of evidence of robustness.

## 4. Integration

The customer required a standalone UWP application to enable easier integration into their UWP application. In addition to building this quickly, our team also provided prompt support to resolve the build and integration issues faced by the customer during the process.

## 5. Reliability and support

By enabling the following during the evaluation engagement process, we were able to convince the customer of our claim of reliable support:

- Performed 64-bit migration of the code on request in rapid time.
- Guaranteed support for the customer's future requirements on the 64 bit ARM/X86 platform.
- Offered extensive support in terms of documentation, describing all the API calls, the build procedure for building binaries, user guide for standalone applications, and detailed reports of all the tests performed.

# Evaluating a Suite of Audio Codecs for an Entire Suite of Automotive Products

While the first example involved a single product with a single codec, the second gets more complex as it involves an entire suite of audio codecs optimized for a range of platforms powering the customer's automotive infotainment product line.

## 1. Features and Platforms

Since automotive infotainment products need to cover a much wider range of formats, and the customer was looking to design multiple product variants, the list of platforms and codecs was exhaustive.

| Codec | Processor Core | SOC | OS | Multimedia Framework |
|---|---|---|---|---|
| MP3 | ARM Cortex A8 | TI Jacinto 5 | QNX | QNX MME |
| | ARM Cortex A8 | TI Jacinto 4 | QNX | QNX MME |
| | ARM Cortex A15 | TI Jacinto 6 | QNX | QNX MME |
| | ARM Cortex A15 | TI OMAP5432 | QNX | QNX MME |
| | ARM Cortex A9 | Customer Proprietary | Linux | Gstreamer |
| WMA 9.2 Std | ARM Cortex A8 | TI Jacinto 5 | QNX | QNX MME |
| | ARM Cortex A8 | TI Jacinto 4 | QNX | QNX MME |
| | ARM Cortex A15 | TI Jacinto 6 | QNX | QNX MME |
| | ARM Cortex A15 | TI OMAP5432 | QNX | QNX MME |
| | ARM Cortex A9 | Customer Proprietary | Linux | Gstreamer |
| AAC-LC | ARM Cortex A8 | TI Jacinto 5 | QNX | QNX MME |
| | ARM Cortex A8 | TI Jacinto 4 | QNX | QNX MME |
| | ARM Cortex A15 | TI Jacinto 6 | QNX | QNX MME |
| | ARM Cortex A15 | TI OMAP5432 | QNX | QNX MME |
| | ARM Cortex A9 | Customer Proprietary | Linux | Gstreamer |
| HE-AAC v1/v2 | ARM Cortex A8 | TI Jacinto 5 | QNX | QNX MME |
| | ARM Cortex A8 | TI Jacinto 4 | QNX | QNX MME |
| | ARM Cortex A15 | TI Jacinto 6 | QNX | QNX MME |
| | ARM Cortex A15 | TI OMAP5432 | QNX | QNX MME |
| | ARM Cortex A9 | Customer Proprietary | Linux | Gstreamer |
| WMA 10 Pro | ARM Cortex A15 | TI Jacinto 6 | QNX | QNX MME |
| | ARM Cortex A15 | TI OMAP5432 | QNX | QNX MME |
| | ARM Cortex A9 | Customer Proprietary | Linux | Gstreamer |
| WMA Lossless | ARM Cortex A15 | TI OMAP5432 | QNX | QNX MME |
| | ARM Cortex A9 | Customer Proprietary | Linux | Gstreamer |
| WMA Voice | ARM Cortex A15 | TI OMAP5432 | QNX | QNX MME |
| PCM-WAV | ARM Cortex A15 | TI OMAP5432 | QNX | QNX MME |
| | ARM Cortex A9 | Customer Proprietary | Linux | Gstreamer |

*Table 2. Coverage of platforms and codecs for a customer's infotainment products*

Although the customer did not require any specialized features, it needed a wide coverage of formats and platforms that significantly reduced the number of vendors who were ready to fulfill the requirements.

## 2. Performance

Automotive products always try to manage the operational CPU clock to extend the life of the processor and system. Performance optimized codecs help this cause immensely. Since the requirement set spans many libraries across platforms, it is unusual to find a single vendor who has the best performance numbers on all codec and platform combinations. However, we presented a close enough case with the best performance for a majority of the use cases – especially for the most common ones like MP3 and AAC

## 3. Quality and compliance

In addition to standards compliance, automotive grade codecs need to exhibit the highest level of robustness. Our test reports that included error resilience testing, combined with our track record in the automotive industry made a convincing argument for our suitability for the customer's product line.

## 4. Integration

Table 2 lists two different frameworks on two different operating systems – QNX MME on QNX and Gstreamer on Linux. We provided ready plug-ins for both the frameworks for evaluation, and the integration tests at the customer's end worked seamlessly.

## 5. Reliability and support

The average lifespan of automotive products and the high expectations on reliability make automotive infotainment a challenging space for codec vendors. The following factors helped the customer in assessing if Ittiam meets these tough requirements for their products:

- Our deployment track record in automotive infotainment – spanning multi-million unit deployments across the world.
- Engagement experience and proactive support during evaluation.
- Financial health of the organization over the years and contractual provisions for extended support engagements.

# Conclusion

Consumers of electronic applications including enterprise applications, medical devices, MP3 players, gaming consoles and in-car infotainment systems increasingly demand best-in-class audio quality. No wonder audio codecs continue to enjoy overwhelming importance in the application ecosystem. However, choosing the best audio codec implementation for your applications can often turn out to be a challenging and complex exercise.

Many organizations consider 'pricing' as the dominant decisive factor, and pay inadequate attention to other major factors that play a bigger role in the success of a product. Whereas, by considering the key evaluation criteria in our assessment guide, you can take a comprehensive look at the larger picture to make the right decision that will determine the success of your applications.

# About Ittiam

Ittiam Systems is a global technology company with deep R&D driven solutions for media creation, management and consumption, and visual analytics. We provide advanced media codecs, software development platforms, systems and workflows for embedded and online applications to major mobility, cloud and media technology companies around the world. Our solutions are at the heart of over a hundred million lifestyle products that drive mobility, content access, networking and sharing.

**Explore our** Audio Codecs, Audio Post Processing and Loudness Metering & Leveling solutions

**Find out about our** Low Power Audio Solutions for Wearables

**Read our recent blog** on Ittiam's mSBC codec implementation for high performance Bluetooth devices

**Start a conversation** @ mkt@ittiam.com

# About the Authors

**Reshma Rai** brings in 17 years of extensive experience, of which she has spent about 12 years at Ittiam, with key focus on audio codec technology. She currently works as Senior Manager, Media Client Technologies, leading the development of audio codecs such as USAC, DSD, and HD-AAC. Reshma also leverages her expertise to enable audio transcoding system optimization of audio, speech and post processing components on ARM CAx, CM4 , ARM9E , Intel X86, TI C66x, and C64x. She has a B.E. degree from Dr. Ambedkar Institute of Technology, Bangalore.

**Saketh Sathuvalli** is a Senior Engineer in the Media Client Technologies business unit at Ittiam. He has been a part of the company for over four years, focusing primarily on integration maintenance and optimization of audio codecs and post processing components for ARM based micro-processors and micro-controllers. Saketh has a B.Tech degree in EEE from NITK, Surathkal.

# Disclaimer

This white paper is for informational purposes only. Ittiam makes no warranties, express, implied or statutory, as to the information in this document. The information contained in this document represents the current view of Ittiam Systems on the issues discussed as of the date of publication. It should not be interpreted to be a commitment on the part of Ittiam, and Ittiam cannot guarantee the accuracy of any information presented after the date of publication. Complying with all applicable copyright laws is the responsibility of the user.