

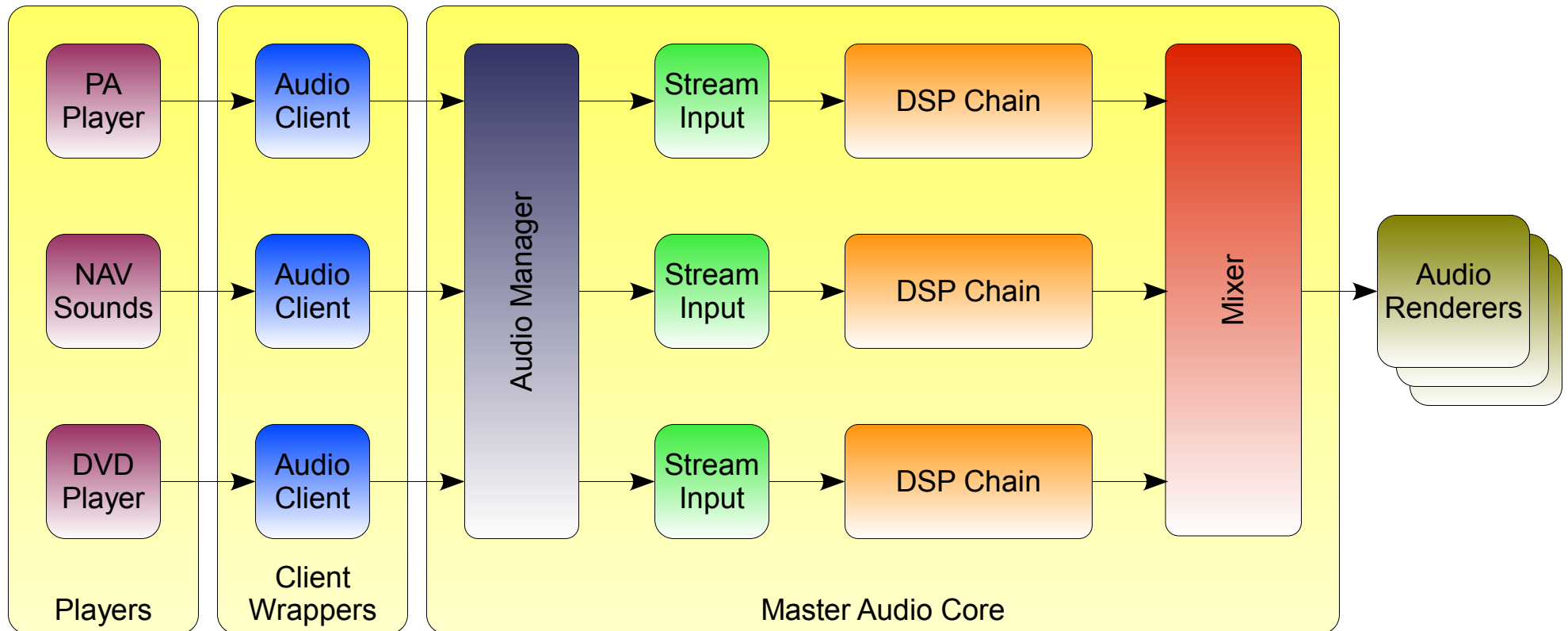
## MasterAudio Framework: Overview

The *MasterAudio Framework* is an audio abstraction and processing subsystem that allows clients to deliver data in a wide range of formats, while ensuring that the data provided to the underlying audio provider is in a format that can be rendered properly and in a manner that suits the content. It also provides the ability to dynamically configure any number of data processing routines that are to be performed on the provided data before it is delivered to the audio provider. Modules that use the *MasterAudio* framework are referred to as clients.

*MasterAudio* is stream-based, meaning that clients provided data to the framework in a sequence of consecutive data frames. These frames are then processed by *MasterAudio* and passed-on to the platform-specific audio provider. Clients may also specify that data is to be passed to the audio provider untouched. This allows encoded, encrypted, and protected streams to be processed by the framework without compromising their content.

The *MasterAudio* processing engine is extensible through the use of plugins. This provides the ability for clients to extend the built-in features of the framework, including the addition of new data formats, filters, and transforms. It also creates the possibility of support for existing audio-processing libraries through the use of wrapper-plugins.

# MasterAudio Framework: High-Level Architecture



**Audio Clients** are wrappers around the Audio Manager interface that simplify dealing with a specific stream format (i.e. PCM, float, Encoded). They are not required to interface with the framework, but require less code when handling common stream formats.

The **Audio Manager** handles requests from clients and manages the interactions between the elements that make up the core of Master Audio. It is also responsible for the creation and destruction of streams within the core.

A **Stream Input** acts as a buffer and converter between the client input data flow and the data flow of a specific MasterAudio stream. A Stream Input exists and is specifically configured for each client stream.

A **DSP Chain** is a processing container for data filters and transforms. It handles the creation and teardown of a filter graph for each client stream, as well as the processing flow-control for the graph.

The **Mixer** is the aggregation point for client streams and may represent either a software-based stream mixer or simply a hardware passthrough container. It is also responsible for buffering and converting stream data for the renderer. This may include format conversion or encoding, depending on the hardware configuration.