



EM- 685 (102): SOFTWARE ARCHITECTURE

2016 SPRING SEMESTER

MILESTONE 2

SYSTEM ARCHITECTURE REPORT

Submitted By:

Team Acoustic

Introduction

Product Family: Music Players

Team Members and their Products :

- a. Krupa Merupu – VLC Media Player
- b. Maithili Parikh– Songbird
- c. Praveen Kumar Medapati – Banshee

Music has become a fundamental part of life, with it being available from many different sources, in various formats and multiple access mechanisms. With the advance in technology and cheap storage space, digital music has become a much more common occurrence. We can download and store music in players as well.

As a result, which Music player preferred is subject to the various features it offers. To explore this, our team has decided to explore VLC, Songbird and Banshee. The common features offered by these are they are cross platform compatible (Work across multiple operating systems), Search and Play local music library, create playlists and edit them, play various formats of Music and download Music from external websites. Built with codes from Firefox and VLC, Songbird is a desktop music player with web browsing capabilities. It is also compatible with most, or all of VLC's features, which makes it a versatile audio/video player that can play almost every file format you throw at it. Songbird has a unique feature of browsing the internet which the other two players lack. Songbird manages some of the clutter in their default views with tabs, but Banshee has a series of permanent panes that feel badly in need of cleanup.

Music Player: VLC Media Player

Overview:

VLC media player (commonly known as VLC) is a portable, free and open source cross-platform multimedia player and streaming media server written by the VideoLAN project. VLC plays most multimedia files as well as, Audio CDs, VCDs, and various streaming protocols.

It is one of the most platform-independent players available, with versions for Microsoft Windows, Mac OS X, iOS, Android, Windows Phone, Linux, BeOS, Syllable, BSD, MorphOS, Solaris and Sharp Zaurus. The programming language used is C, C++(with Qt), Objective-C. The User Interface of VLC is in more than 75 languages available and its space demand on hard disk: 110 MB.

VLC media player supports many audio and video compression methods and file formats, including DVD-Video, video CD and streaming protocols. It is able to stream media over computer networks and to transcode multimedia file.

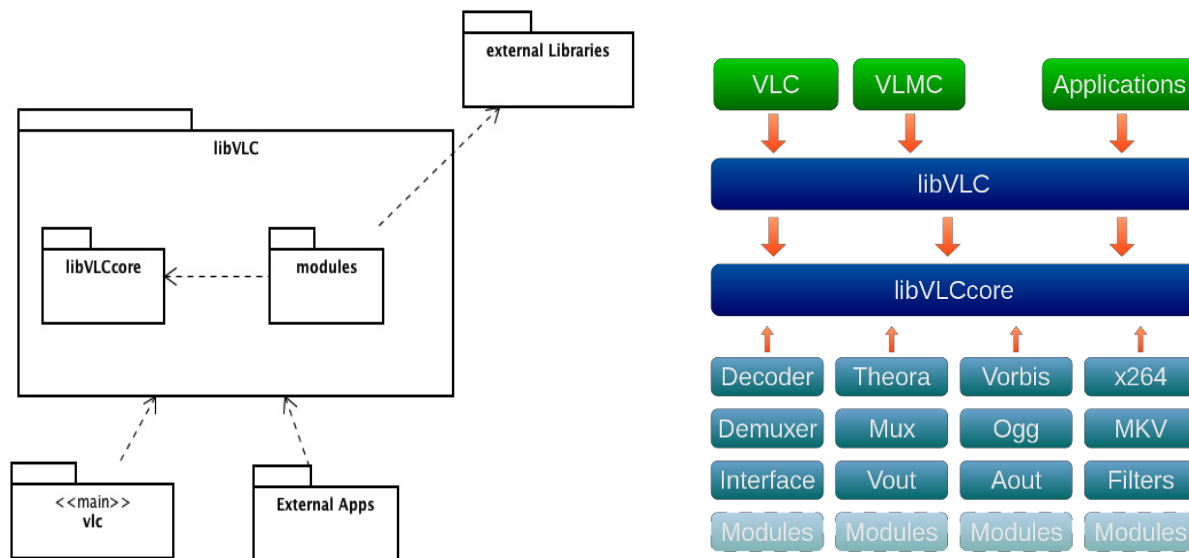
Because VLC is a packet-based media player it plays almost all video content. It can play some, even if they're damaged, incomplete, or unfinished, such as files that are still downloading via peer-to-peer (P2P) network. It can also play from external sources such as Satellite, Cable, Digital TV cards (DVB-S, DVB-T), and several types of network streams: UDP/RTP Unicast, UDP/RTP Multicast, HTTP, RTSP, MMS, etc.; Acquisition or encoding card, Webcams and other devices.

The most observable type of architecture used in the VLC Media Player is the Multilayered Architecture. Each layer in the architecture interacts with the layers both above and below it in order to allow a seamless and robust flow of data needed between the various layers. They can be most broadly classified into.

- Presentation Layer (The interface of the player)
- Application Layer (Allows various API's to interact with the player)
- Business Layer (Determines how data can be created, displayed, stored, and changed.)
- Network Layer(Logging and networking required to support a specific process)

Module View:

VLC, like most multimedia frameworks, has a very modular design which makes it easier to include modules/plugins for new file formats, codecs, or streaming methods. VLC has more than 380 modules. The VLC core creates its own graph of modules dynamically, depending on the situation: input protocol, input file format, input codec, video card capabilities and other parameters. In VLC, almost everything is a module, like interfaces, video and audio outputs, controls, scalars, codecs, and audio/video filters. In the source code, modules are usually located inside the modules/ subdirectory.

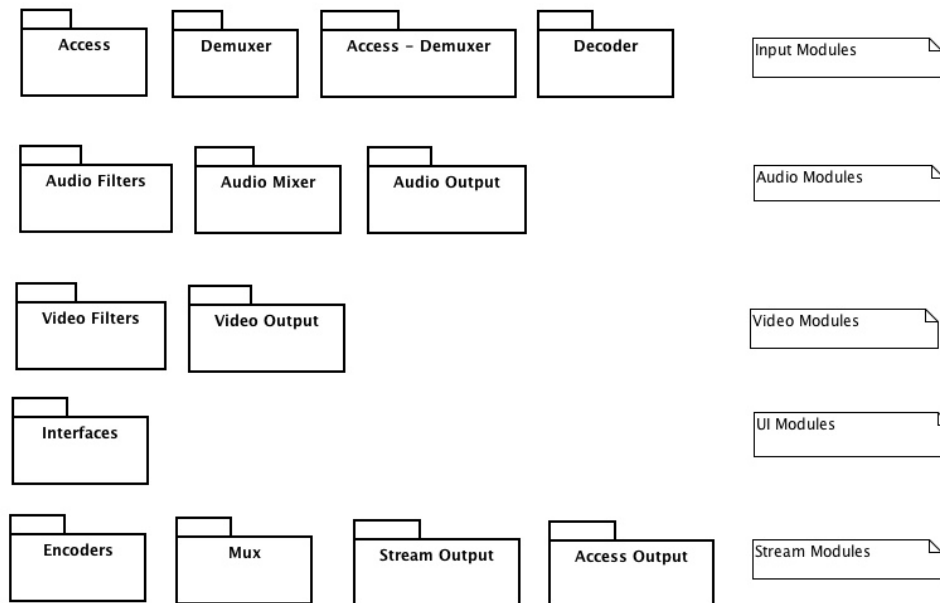


VLC has a core and a lot of modules (between 200 and 400 depending on the build). There are following modules:

- **libVLCcore:** it is the central core of the framework. It provides an object-oriented layer to C, module loading/unloading and a set of abstract functionalities regarding multimedia: input, multiplexing, demultiplexing, audio output, video output.
- **Modules:** They provide concrete functionalities of the framework. Modules are categorized according to their capabilities. There are modules for input management (file, network, cd), modules for codecs (mp3, divx,...), modules for gui (textual, web, telnet, qt-based, macosx native)
- **External libraries:** As there are lots of modules, there are lots of external dependencies.
- **vlc (main):** It is the main of the player. It initializes **libVLC** and launch user interface.

As previously said VLC without modules is useless. Modules can be categorized according to their capabilities. A capability should be thought as something similar to an Object Oriented Interface. Module loading is simple to explain: when VLC needs a module with a particular capability, it opens all modules for that capability, until one matches. It opens them in decreasing

score order (bigger score first, smaller ones at the end), and runs the `Open()` function of the module. When one module returns OK, VLC uses this module.



Input Modules:

These modules allow VLC to read its streams from different sources. Input modules consist of access, demuxer, access- demuxer and decoder. Some of the examples are dvdplay, dvd, dvdread, vcd, cdda, etc. In a video stream, the video signal and the audio one are always into "containers" formats. Demuxers extract the streams from it and pass it to the decoders. It allows you to read .avi files, .asf files, AAC files, .ogg files and DV files allows to demux streams from a satellite card.

Audio Modules:

These modules consist of audio filters, audio mixer and audio output. Audio Output allows you to choose the way the sound will be output to your audio system.

Video Modules:

Video outputs are the modules that enable the support of some systems to display the video on your screen. They consist of Video Filters and Video Output. Video Filters allow you to perform modifications on the rendered image. They allow you to have the video cut in pieces in several windows, which you can order as you wish. It can be used to generate image walls with several sources. They allow you to distort, rotate the video in several ways, inverses colors, image contrast, hue, saturation and brightness Video Outputs are the modules that enable the support of some systems to display the video on your screen video output uses Microsoft Direct X libraries.

UI Modules:

Interfaces are the way you interact with VLC media player. Like anything else in VLC, they are modules, which allows for their interchangeability.

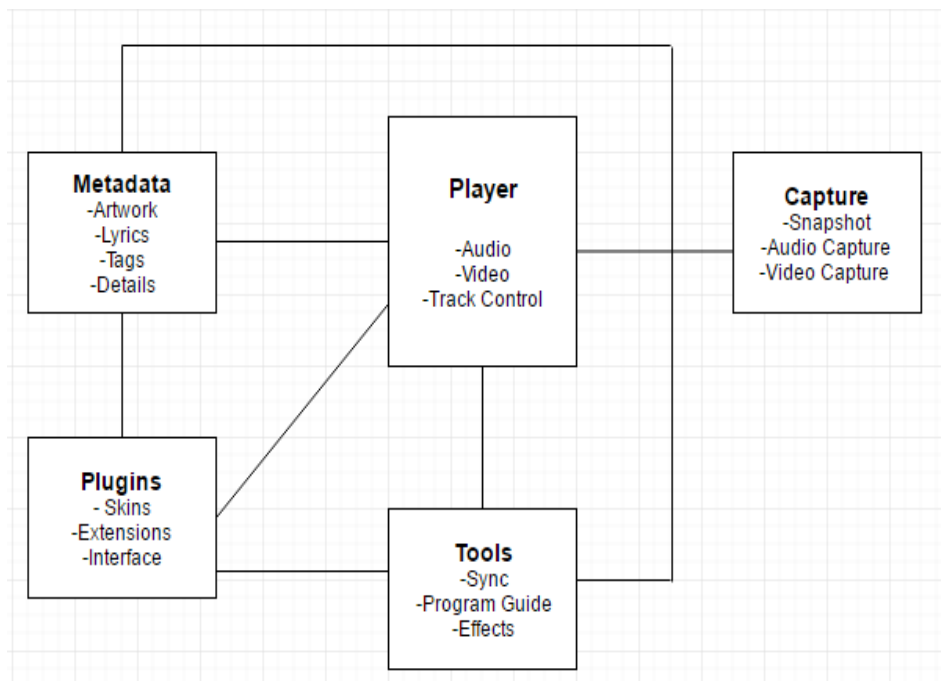
VLC has four main graphical interfaces:

- The Qt Interface (qt) is the default interface on Linux and Windows starting with version 0.9.0.
- The skins2 Interface is an interface where you can customize VLC's look (works on Linux and Windows).
- The Mac OS X Interface is the default (and only) graphical interface on Mac OS X.
- The BeOS Interface is the default (and only) graphical interface on BeOS.

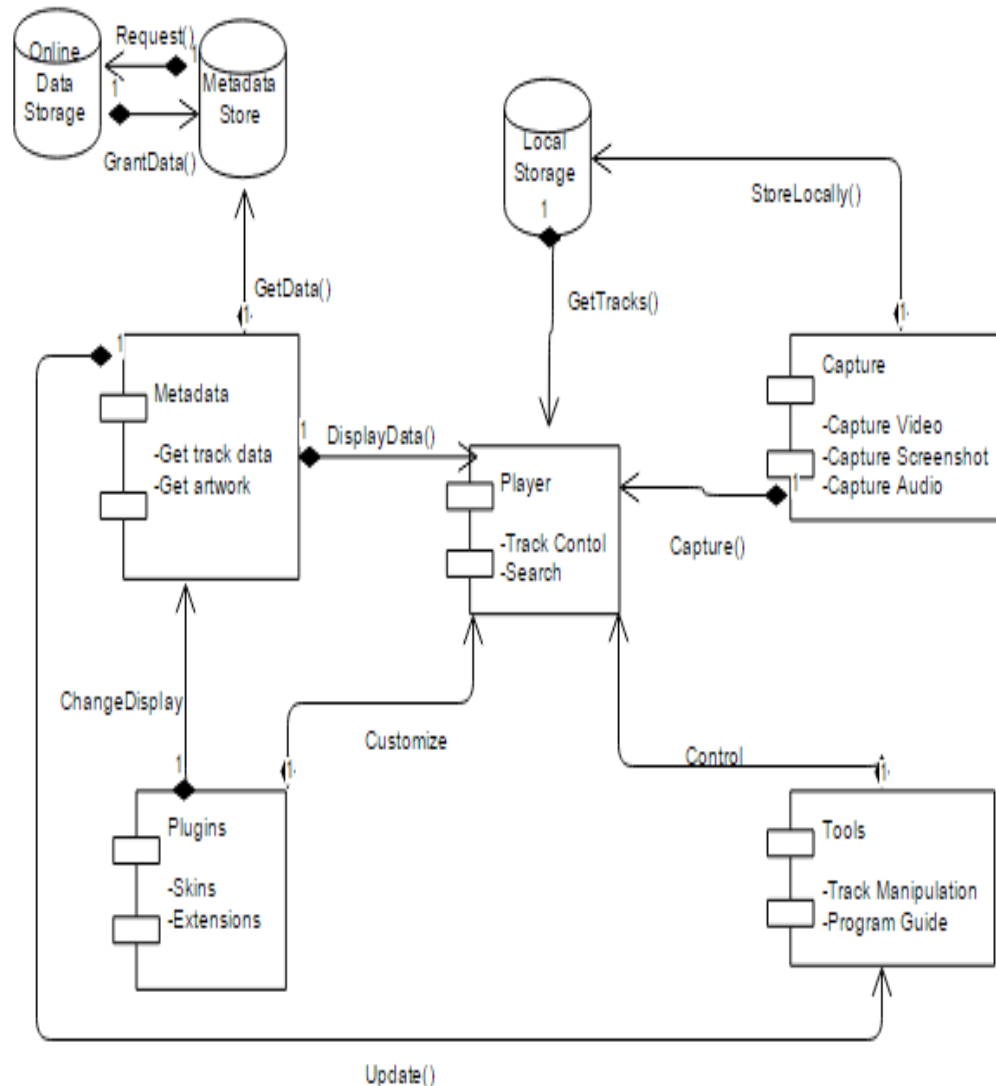
Stream Modules:

VLC's stream output allows VLC to be used as a streaming server instead of a client. It has very extended capabilities:

- Stream in unicast and multicast on an IPv4 or IPv6 network everything that VLC is able to read, via UDP, RTP or HTTP ;
- Save the input stream to a file in AVI, PS, TS or OGG format.
- Transcode an input stream, and then, send it, to the network or to a file.



Components & Connector View:



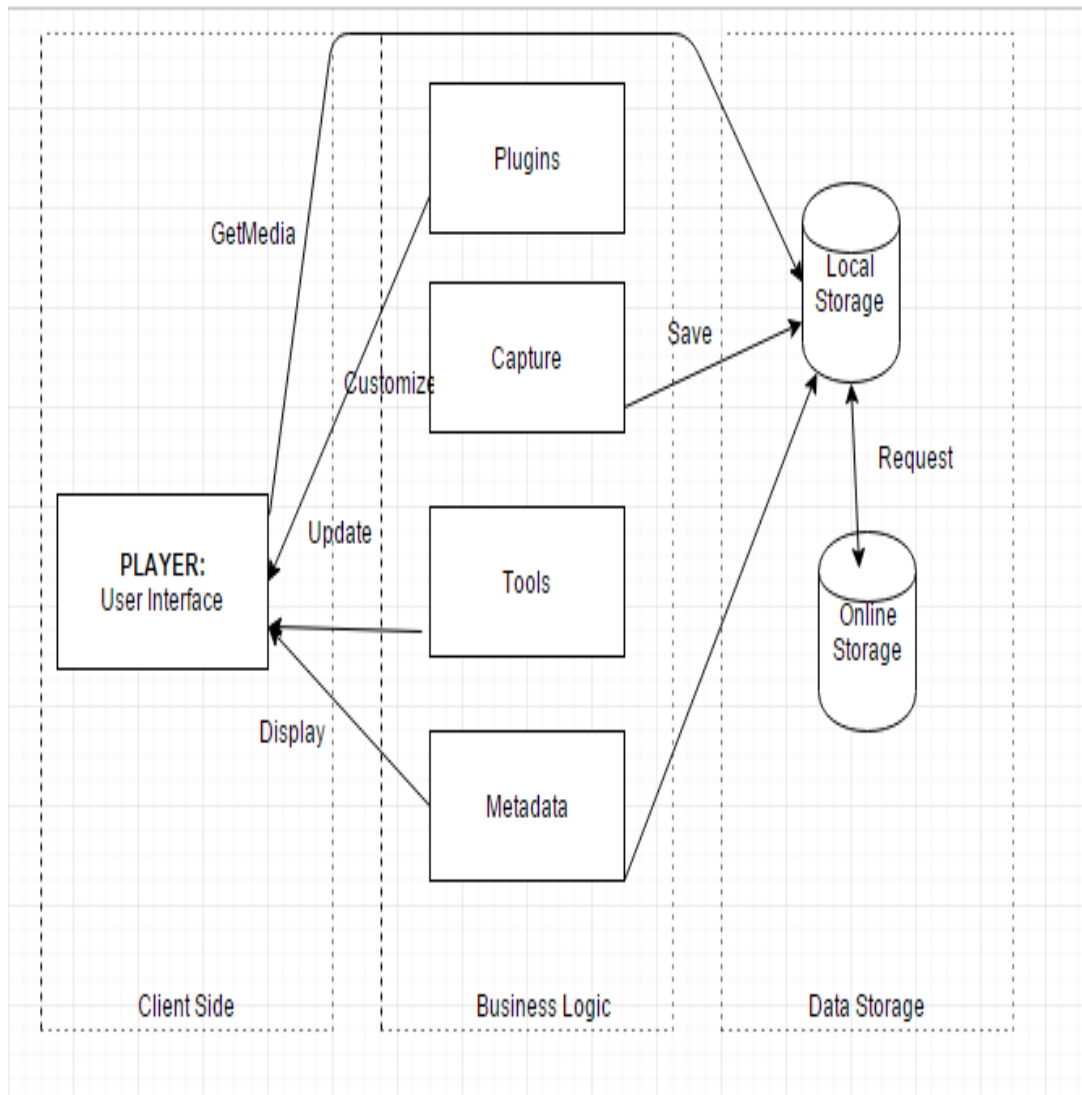
The flow of data usually starts from the player module which includes the user interface from which user can control and manipulate various aspects of the player and play media. From the metadata module the player is updated with information about the track being played as well as other tracks in the queue. The metadata module also interacts with the tools module and plugins module, providing updated data (after any changes are made) and updated layouts (in case of

skins/extensions change) to these modules respectively. The metadata module may not have the information required to publish to the UI as per users' request. In this case, a request is made to an online server from where it retrieves the data, if not present locally on the system.

The Tools module allows the track being played to be manipulated and controlled such track synchronization etc. It interacts directly with the player. The plugins module allows user friendly customization with the player and thus interacts directly as well. The capture module captures screenshots, audio or video files and saves them locally on the system.

Connectors:

- *GetData()*, *Request()*, *GrantData()*: These queries are passed from the metadata module in order to retrieve data and publish it to the metadata module from an online server in the event that the requested data is not available locally in the data store.
- *Update()*: When the metadata is updated, this requires other modules to update their layout (in the case of plugins, player and tools) and view in order to accommodate the changes made.
- *Customization()*: This connectors allows the plugins to make the necessary UI changes to the player.
- *Capture()*, *StoreLocally()*: This captures and then saves the captured image, audio or video file locally on the database.
- *DisplayData()*: The metadata allows the player to display data to the user and give him the necessary information about the current track.
- *GetTracks()*: Retrieves the tracks that the user requests from the local system and plays it in the player.

Allocation View:

In the allocation view above, the client side includes the player which consists of the user interface. In the business logic tier all the key processes take place. The various modules interact with other kernels in order to either publish or retrieve data for the required operations. The data storage kernel contains both the local database store as well as the online data store which is essentially a server which can be requested access to in the event that certain required information is not present at the local DB.

Conclusion:

The architecture used by the developers of VLC Media Player is highly efficient in order to carry out the processes and functions required by the user. VLC media player has a highly portable framework and works across different platforms. It is available for Windows, Macintosh, and many Linux flavors.

The interaction and connectors as seen in the connector view enable us to gain a certain level of understanding of the internal workings and relationships between the various modules. While the tools and capture modules allow the middle tier to access the database in order to provide the data to the client side front end. The user inputs his desired process into the interface which then accesses the middle layer. This middle layer in turn may access the backend for data which may either be present locally or online.

VLC media player in simple terms is the Swiss army knife of media playback software that marvels with compatibilities and capabilities and its open source foundation and community makes sure that the player keeps improving and finally it's a robust piece of software that lets a techie do all kinds of arcane things

Music Player: Songbird

Overview:

Songbird is a discontinued music player originally released in early 2006 with the stated mission to incubate Songbird, the first Web player, to catalyze and champion a diverse, open Media Web. Songbird utilizes the cross-platform frameworks Mozilla XULRunner and GStreamer media framework. Songbird currently runs on Windows and OS X. In 2012, an Android version and an iOS version were released. Songbird at one point also supported Solaris and Linux. As a result, users have forked Songbird and created a Windows, Mac, and Linux compatible derivative under the name Nightingale.

The programming language used was java. It is available in 98 languages. It has multi –platform compatibility with Windows XP, Vista and Mac OS X v10.5 (x86, x86-64). Songbird has ability to play multiple audio formats, such as MP3, AAC, FLAC, Apple Lossless and WMA. It has ability to play Apple FairPlay -encoded audio on Windows and Mac platforms via hooks into QuickTime and also Windows Media DRM audio on Windows platforms.

Media files stored on pages viewed in the browser show up as playable files in Songbird. It has a skinnable interface, with skins called "feathers". Songbird has unique ability to build custom mixes and subscribe to MP3 blogs as playlists. It has ability to scan the user's computer for all audio files and add them to a local library.

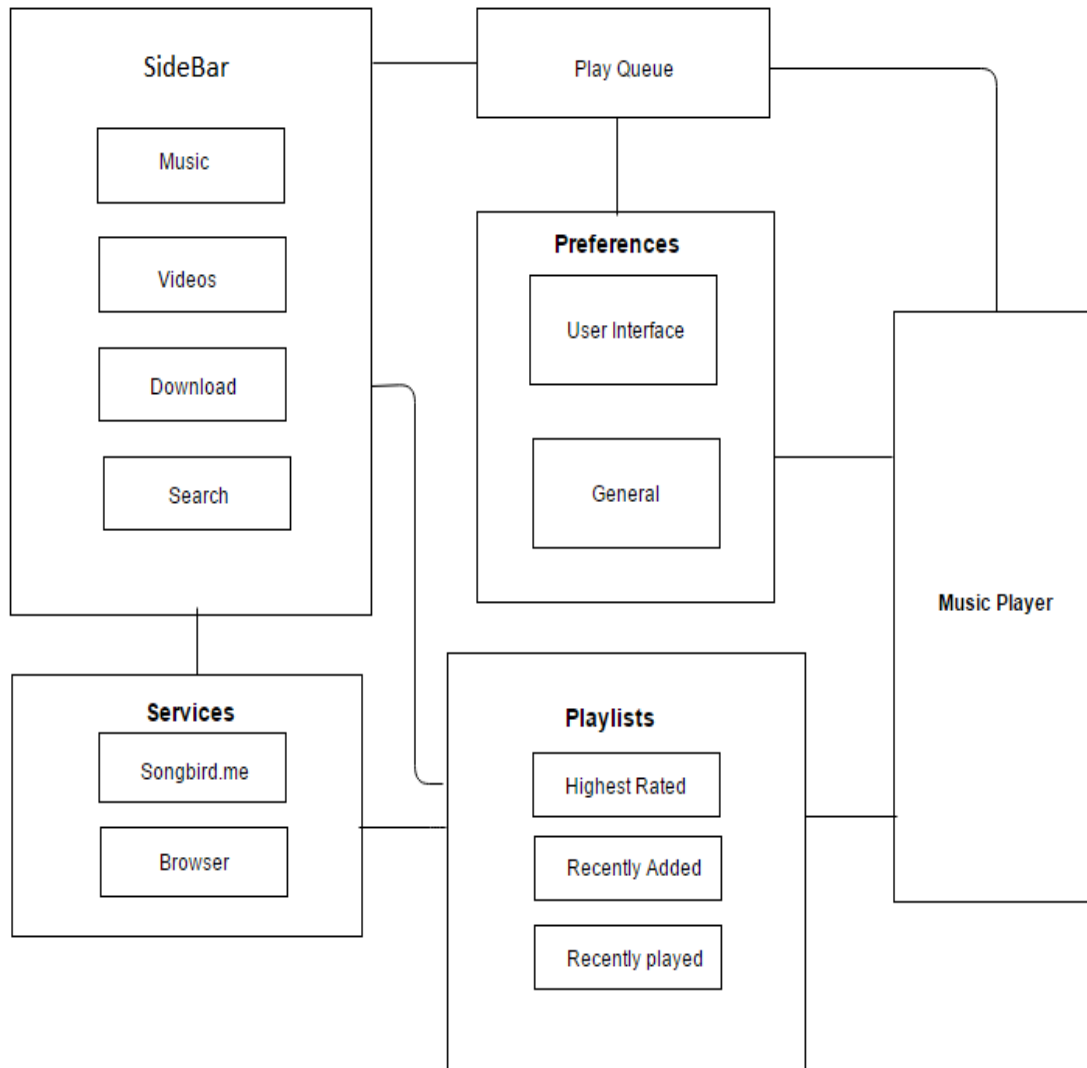
Songbird also works as a media server, streaming your media collection to UPnP or DLNA devices on the same local network. It's also available as a Web app and an Android app that sync with the desktop version.

Songbird's plug-in architecture has allowed developers to create add-ons ranging from lyric-fetching tools to plug-ins that scan your music library to let you know which of your favorite bands are playing in your area.

Songbird trumps iTunes with a number of very cool features — a full-fledged web browser, RSS and MP3 discovery tools, integrated lyric searching, related artist add-ons and much more.

It has multilayered architecture. The layers interact in a way that is both efficient and user friendly. By maintaining the front end module as a separate player module, we are able to divide the architecture into tiers.

Module View:



i. Sidebar

The Library of the music player located on the left side of the application is used for browsing through music files in the drive. It consists of various features such as, Downloads which is used to access downloaded music, videos which are used for accessing videos from the drive, Search which is used for searching music within the library and Music for browsing through the added music.

ii. Services

The services of the Music player is used for using online features such as to purchase music or concerts through 7Digital Store, Radio which is used for using online radio.

iii. Playlists

This Tab in the music player is used for creating playlists which is a list of music files to be played in sequence. These playlists can be made automatically such as by Highest rating or recently added or recently played and can be made by the user according to the needs.

iv. Preferences

The preferences window is used for user customization of the music player. It is divided into two parts, User interface and General.

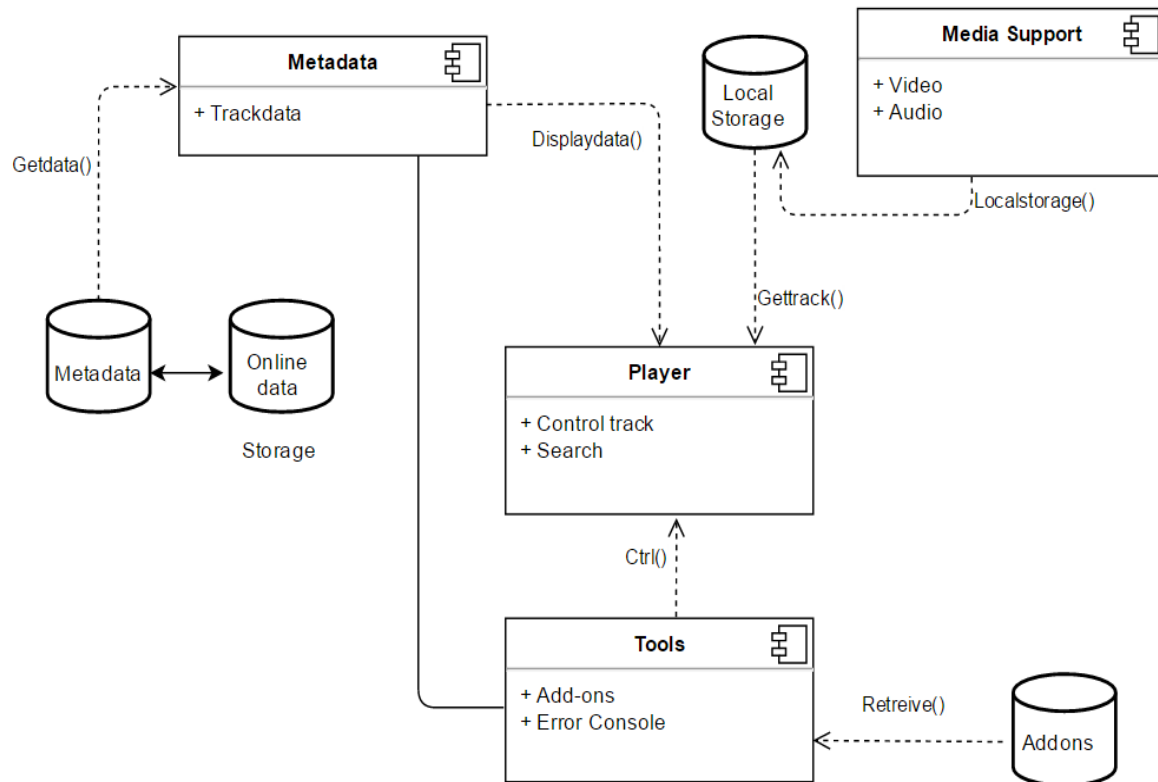
v. Play Queue

Play Queue is used for creating queues of songs which the user wants to hear. This is similar to a playlist, but a playlist is permanent while a play queue is temporary.

vi. Music Player

The Music Player is an essential module of the application which retrieves data from the source and plays it based on how the user wants it.

Components & Connector View:



COMPONENTS:

Metadata

This is a module where all the data is stored. The player will solve any query given by the user using this data. In simple words it is a set of data that describes and gives information about other data pertaining to the music player. It contains details about all the tracks (songs) in the music player.

Online Data

This is an online database accessed while the player is streaming from the internet. It is connected to the metadata module with the help of `getdata()` connector.

Player

This is the module where all the functions are displayed for the user to access. It contains various features like control track, search, etc. User can choose any function according to his or her requirements.

Tools

This module gives the player access to some add-on features. It also contains the error console which is shown if something goes wrong.

Add-ons

All the Add-ons are stored here and can be retrieved in the tools module using the retrieve() connector. Most of the add-ons are already present here and additional add-ons can be added manually by the user.

Media Support

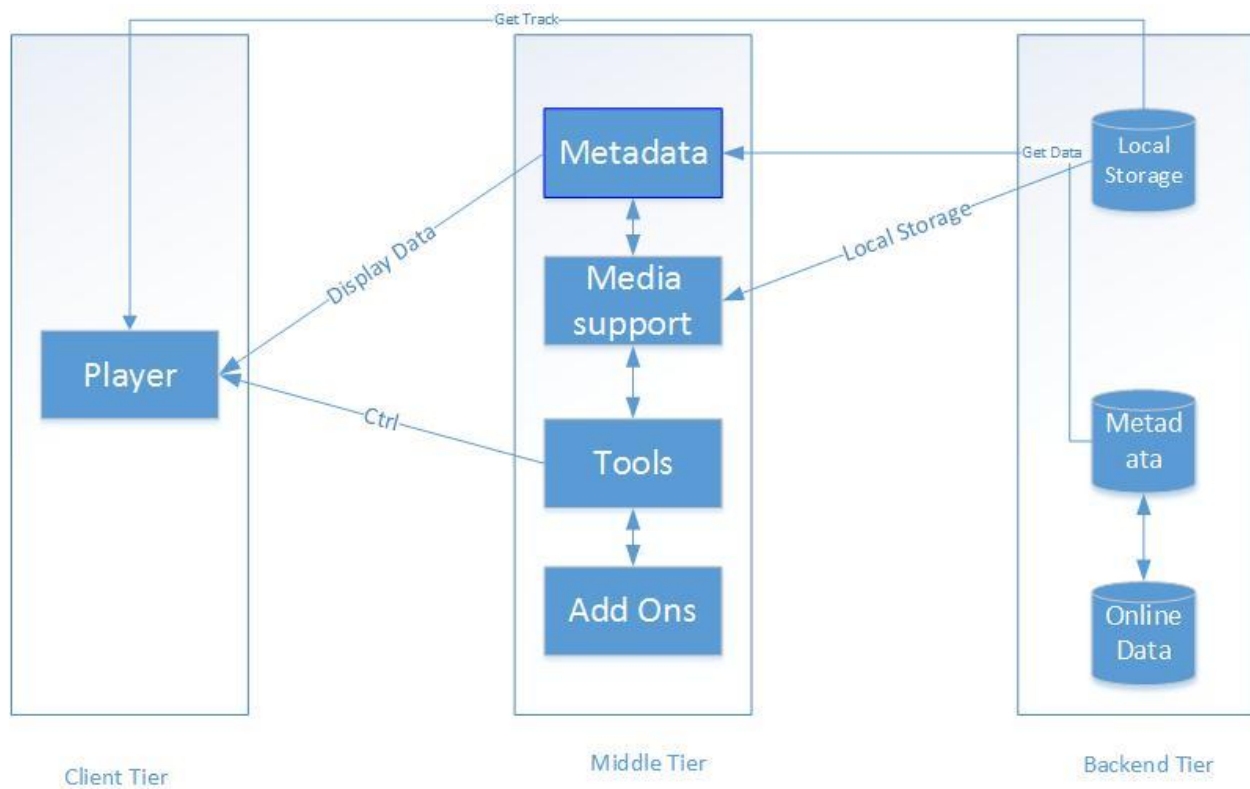
In this module all the files necessary to execute the videos and audios are stored. This is one of the most important modules in the player.

Local Storage

All the media support files along with the songs and videos are stored here.

CONNECTORS:

- **Retrieve()** : It is used to retrieve add-ons into tools module.
- **Displaydata()**: It is used to display the data in the player module.
- **Localstorage()**: It is used to store data back and forth in the local storages and media support.
- **gettrack()**: It is used to get the tracks from the local storages to the player module.
- **getdata()**: It is used to fetch data from online storages and other storages to the metadata module. Metadata module uses this same data to solve queries.
- **ctrl()**: It is used to control the player using the tools available.

Allocation View:

In the allocation view above, the Client Tier includes the Music Player which is the interface for the user to interact. The Middle Tier, which has various components such as Metadata, Media support Tools and Add Ons. The Backend Tier is used for storage of data and has components such as the Local Storage, Metadata and Online data.

Conclusion:

Songbird is unlike so many other music players on the various Linux desktop environments. It does not mimic the look and feel of other music app interfaces. Songbird does not fit squarely into one particular category. Instead, it is a desktop Web player, a digital jukebox and Web browser mash-up all nested in one cozy application.

Songbird's customization extends far beyond the large collection of plugins. Songbird lets users dock the menu bar at the top or the bottom of the application's window. Mini Mode hides the Songbird window and reduces the running app to what is essentially a floating toolbar. From it you can access player controls or return to its Main Player mode. Songbird uses the Gstreamer playback framework. This ensures a seamless playback for a variety of audio formats and radio stream sources. Overall, it is a good choice for managing and playing media collection.

Other features make it very suitable for organizing music library files with its Watch Folders option. Media importing and exporting is easy to do as well. Its extensive plug-in arsenal makes it very difficult to walk away from this music playing app.

Lastly, the fact that the architecture only allows the User to see the User Interface, while hiding all other vital functions allows for a large amount of data integrity, while at the same time providing a simple usage experience. In conclusion, therefore, one must say that the Software Architecture of the Songbird music player is based in simplicity, while still being robust enough to deliver the functionality promised.

Music Player: Banshee

Overview:

Banshee is a cross-platform open-source media player, called Sonance until 2005. Built upon Mono and Gtk#, it uses the GStreamer multimedia platform for encoding and decoding various media formats, including Ogg Vorbis, MP3 and FLAC. Banshee is a Linux-inspired player that helps you to organize your music and interact with other users via scrobbling to Last.fm.

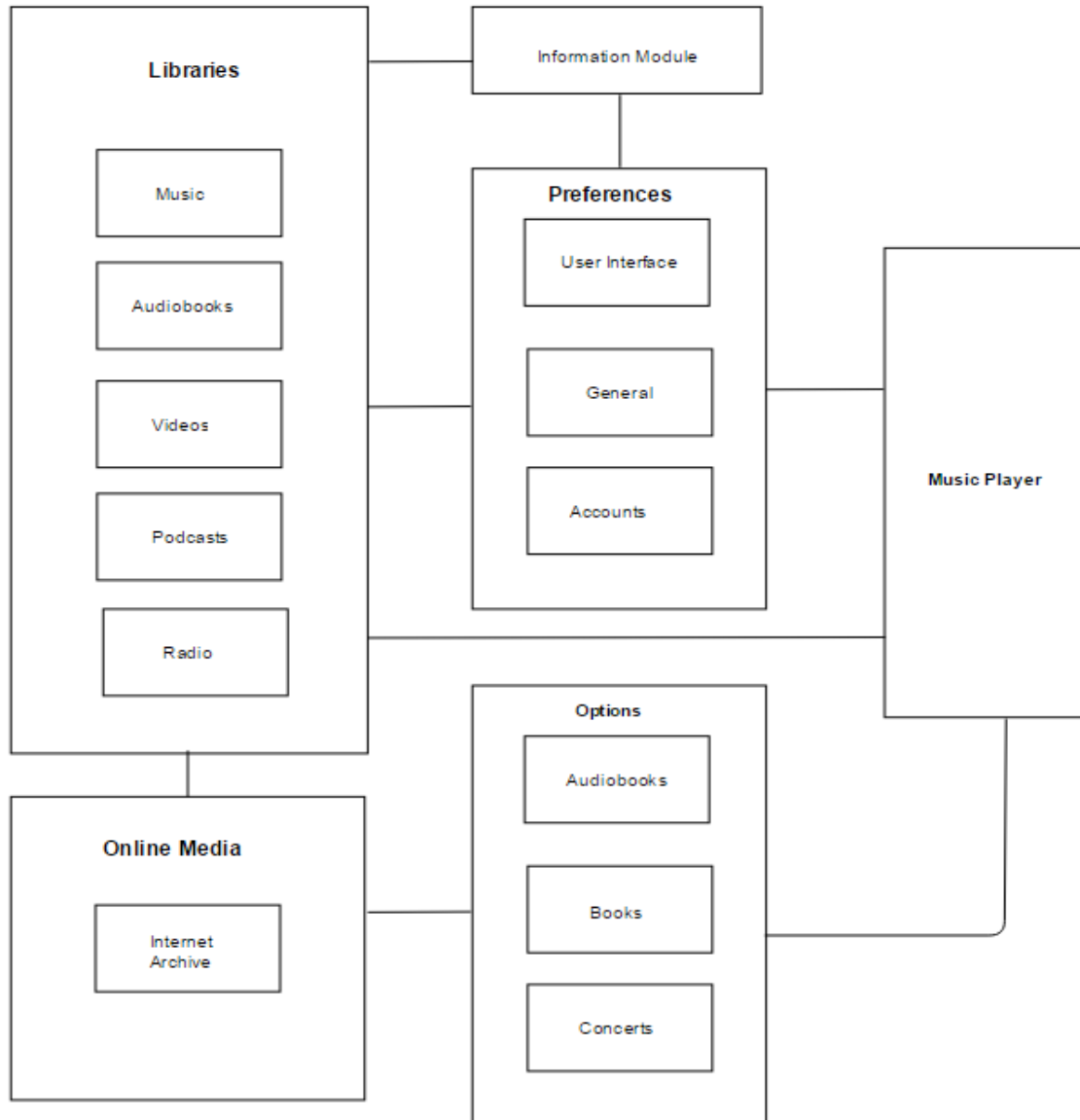
Banshee plays most popular formats, allow you to rip music from your CDs and create your own audio CDs and MP3s for listening to music stations over the Internet. You can also subscribe to podcasts and download them automatically using Banshee.

Banshee has all the basic functions you would expect such as shuffle and repeat but also a few added bonuses such allowing you to see what others are listening to via scrobbling to Last.fm. The Banshee community is quite lively and you can exchange favorite tracks. It's also highly extendible via a comprehensive extensions section that covers everything from Karaoke to a Lyric-finding app.

Unlike the Mac version, Banshee for Windows and Linux also supports viewing of videos within the main interface. It also supports iPod syncing and playing of CDs. However, the codec support for videos is very limited meaning you might not be able to play everything you hoped for.

Banshee's plugin-capable architecture makes the software extensible and customizable. As of 2012 stable plugins include: Audioscrobbler, DAAP music sharing, iPod manager, Music Recommendations using Last.fm, Podcasting, Radio

Module View:



i. Libraries

Upon Launching Banshee the sidebar located at the left side allows the user to access various functions like audiobooks, videos, podcasts, radio etc. User can also queue his playlist by adding different songs to the file. Various features are included in Library module like Videos, Audiobooks, Podcasts, radio, music.

ii. Online Media

Online Media Module is used for accessing internet based music services such as Internet Radio, Concerts according to the user needs.

iii. Preferences

The preferences window is used for user customization of the music player. It is divided into three parts, User interface, Accounts and General. The User Interface is where the user interacts with the music player.

iv. Options

Books, concerts, audiobooks, lectures are a part of Internet Archives. Internet archive is building a digital library of internet sites and other cultural artifacts in digital form. Like a paper library internet archive provide free access to researchers, historians, scholars and general public.

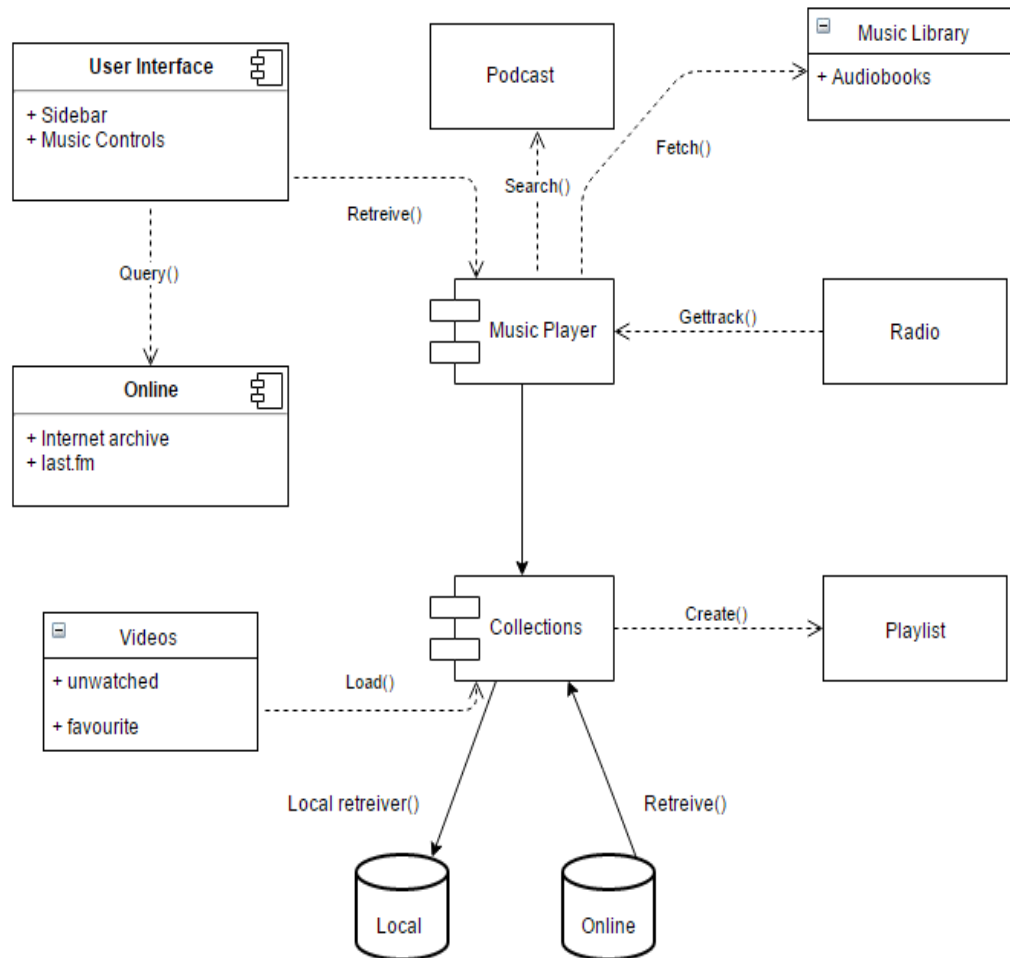
v. Information Module

This module is not actually used by the user itself. It is used for displaying information to the user such as lyrics or details of artists.

vi. Music Player

The Music Player is an essential module of the application which retrieves data from the source and plays it based on how the user wants it.

Components & Connector View:



COMPONENTS:

User Interface

All the interactions between the user and the music player are carried out through user interface. This includes the basic functions of the music player such as play, pause, fast-forward, stop, etc. These interactions take place via functions() connectors. Thus the UI allows the user to interact with the music player in real time.

Music Player

This is the component which is responsible for playing the music based on the inputs given by the user. It reacts to various inputs like volume, bass, treble, etc. It operates with the help of the data received from the search module.

Search

This is the core component of the music player. The user gives inputs or queries in the search module. Once the module receives a query, it retrieves it from the meta-data list, compiles it and displays it regardless of source. It displays this to the user and the user has a choice whether to select a particular song from the search results or select all the songs from the results. The meta-data is passed to the Music Player component which retrieves and plays the file.

Collection

This module stores all the metadata lists the user has in his or her local database. It also allows the player to stream songs from the internet. This means that give a search from the Search module, the Collection database creates a list of the appropriate meta-data from both the online databases.

- a. Local Database: Comprises of a list of the meta-data of all music files stored by User on local machine.
- b. Online Database: Comprises of a connection mechanism to online databases, which contain audio meta-data.

Online

This module gives the player an access to the online database of songs. The player searches the songs via this module during online streaming. It comprises of a connection mechanism to online databases, which contain audio meta-data.

Videos

This module is used when the user wants to watch videos on the player which are either stored in the local database or streamed from the online database. The user has an option to sort the videos into various categories such as favorites, unwatched, etc.

Radio

This module connects the player to the local radio channel. It allows the user to access radio channels at different frequencies. This module needs internet connectivity to function.

Podcast

This module uses a digital audio file made available on the Internet for downloading to a computer or portable media player, typically available as a series, new installments of which can be received by subscribers automatically. The user may have to subscribe to this service.

Playlist

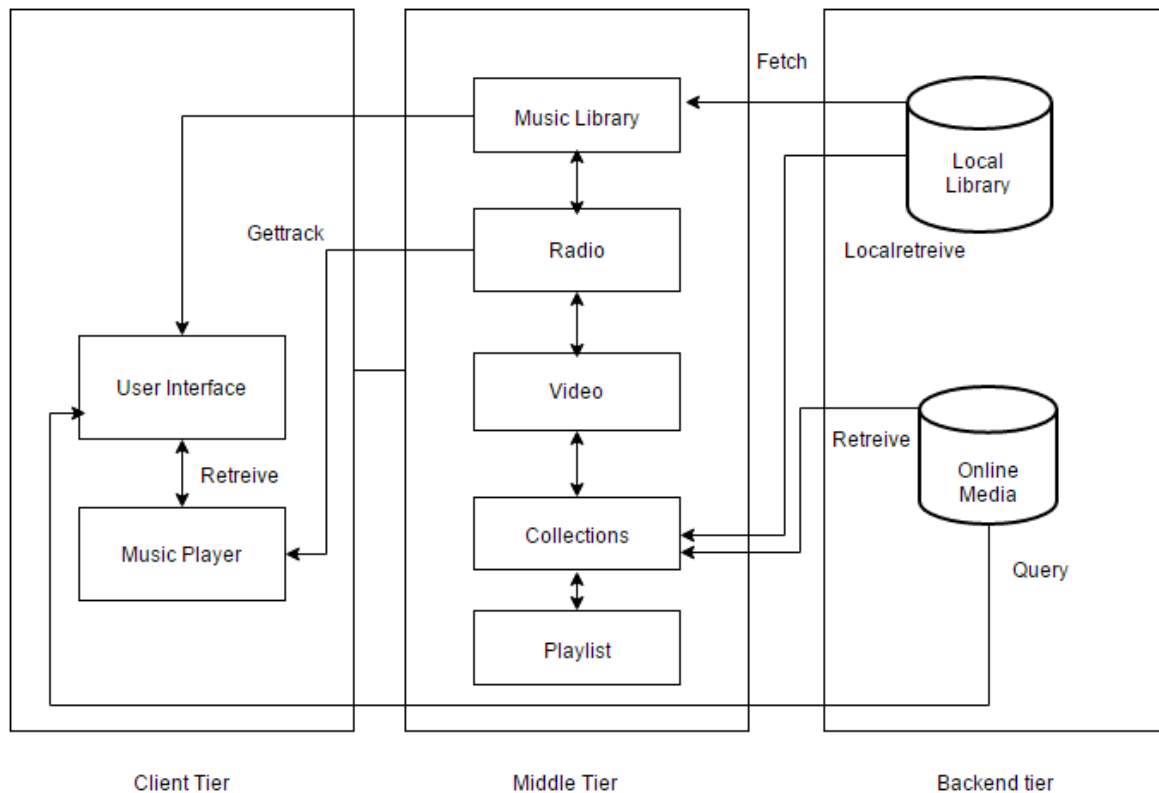
This module stores all the playlists created by the user. It will directly fetch the desired songs upon played. The playlists are generally created by the user manually, but there can be some suggested playlists pertaining to a specific genre already in the music player. This is nothing but a group of similar songs or favorite songs.

Music Library

This module has all the data saved at one place. The songs are organized under different categories. It is an advance collection module in a more organized way.

CONNECTORS:

- i. **Query():** This connector is a procedural call which passes the Search Query as entered by the User to the Search component. The connector is implemented. This is an EVENT connector, which is triggered when the User performs a search.
- ii. **Load():** This connector is used for transferring data in the music player. It is one of the DATA access connectors.
- iii. **Retrieve():** This connector is used to retrieve stored data from either the local databases or the online databases whenever necessary. It is also one of the DATA access connectors. It finds the meta-data which matches that defined in the Query.
- iv. **Create():** This connector is used to create playlists in the playlist module. It uses the data stored in the music player. It is a PROCEDURE call connector. It basically stores the metadata from the collections.
- v. **Search():** The Search module passes the Query entered by the User, to find the associated meta-data within the Collections. This can be categorized as a PROCEDURE CALL connector, which uses the Query parameters to perform a search through the Collection module.
- vi. **Fetch():** It fetches the audiobooks from the music library and plays it in the music player.
- vii. **gettrack():** It plays the track in the system after it retrieves the track requested by user.

Allocation View:

The Allocation view comprises of 3 tier: Client tier, Middle tier, Backend tier. The User Interface tier consists of the User Interface where the user interaction with the program takes place. The user either input some query or passes a command in the middle tier to play the music, play the video or radio, and create collections and playlist. These commands passed by the user in the Client tier are processed in the Middle tier. These requests are either to fetch video, turn on the radio, create collections, add music to the playlist and fetch music. The Middle tier identifies the request forwarded by the User Interface tier and further performs a search in Backend tier. The Backend tier consists of the local library and the online media. After a successful match is found in the Local library or online database for the requested query/command, this match is then returned back to the user.

Conclusion:

Banshee is a refreshing alternative to the other big names in Linux media apps. It is very customizable, thanks to the plug-in structure Banshee uses. Set up could be done way easily by accessing the Extensions options in Edit/Preferences. This will let you turn off/on the various services and components that Banshee offers. The same can be done for configuring source-specific settings and file policies within the Edit/Preferences options. For instance, it scans your hard drives and other attached storage media to catalog all existing music files, videos and podcasts. In addition, Banshee makes it simple to tune into the MiroGuide to locate music, HD Shows, video and audio collections for purchase and download.

The interface for Banshee is nice and easy to use, and very responsive. Search is instant, bringing up results as you type, and the way results are listed is conducive to finding the tracks, album or artist you're looking for. Banshee stands as a Linux media heavyweight. It does audio, video, Amazon music, Internet Archive, iPod, MTP, UPnP and USB external device support and it can rip CDs; it can even do variable bitrate MP3s.

Anything you stream (preview) is handled by Banshee, which means no Flash plugins and some lightning-fast buffering. Podcasts can be subscribed to with Banshee's in-built podcast tool and you can even add a shortcut link to your favorite shows on the main Banshee sidebar. Banshee gets things right in one package. While "right" varies among users' as much as personal preferences in music, some things stand out in ways that are not as apparent in other music players. Banshee is a great alternative for those who are tired of bloated music applications or just want something a little different. Banshee is the free and open source premier media player which allows you to play music, videos, keep up with podcasts, listen to internet radio, etc.

It is simple enough to enjoy and powerful enough to thrill.