

# Overview of Possible Architectural Solutions for MeeGo Cloud Player

Oleksandr Kachur, Vladimir Sayenko  
KTURE  
Kharkiv, Ukraine  
{alexander.kachur, visank}@gmail.com

## Abstract

Main goal of this paper is to make brief overview of available architecture solutions and highlight possible implementation problems. Describe existing approaches and ways to retrieve media content through Internet network. Compare existing cloud services of third-party developer and propose architecture to implement.

**Index Terms:** MeeGo, GStreamer, Cloud services, Social integration, Music player.

## I. INTRODUCTION

An access and synchronization problem media libraries on different mobile and standalone devices has arose several years ago and some progress in this can be considered. Due to widely spreading of coverage area by various telecommunication providers and availability of mobile devices willingness to listen to music can be finally satisfied. Users with different internet devices can get access to the net almost twenty-four hours a day. It can be accessed by wireless access points or by cable. Particular technology doesn't really matter. What really matters is that fact that once been connected to the Internet it's possible to get private media library by accessing remote server. It gives an excited opportunity to listen favorite music while jogging, in car, on TV-set, netbook and personal computer. In fact mainly it's became possible due to cross platform technologies such as Qt, open protocols such as HTTP and brand new cloud-services [1] such as Google music.

## II. MAIN PART

Due to a fact that remote model of storage media library implies dedicated to this task remote server, it's possible to represent existed variants of solution on main three types, based on the type of resource retrieve process:

- personal server;
- cloud server;
- retrieving media content in other way.

Solution to choose must meet the requirements listed in this scenario.

User can authenticate to remote storage. This step prevents somebody to retrieve access to your personal media files. Authentication is very important due to fact that licensing music and unauthorized usage of media content can provide a reason to record

companies to take a look at your software or service. After successful login user gets play list from remote storage and when it's retrieved sends another request for desirable media file. When file or chunk of a file retrieved it's possible to play now. Since file is playing we want to scrobble it into the last.fm account, but before that we need to wait some pre-defined time interval, to prevent spamming last.fm server with scrobble messages. In this usage description login procedure to last.fm was omitted.

Visualization of this scenario is represented via UML sequence diagram on fig. 1.

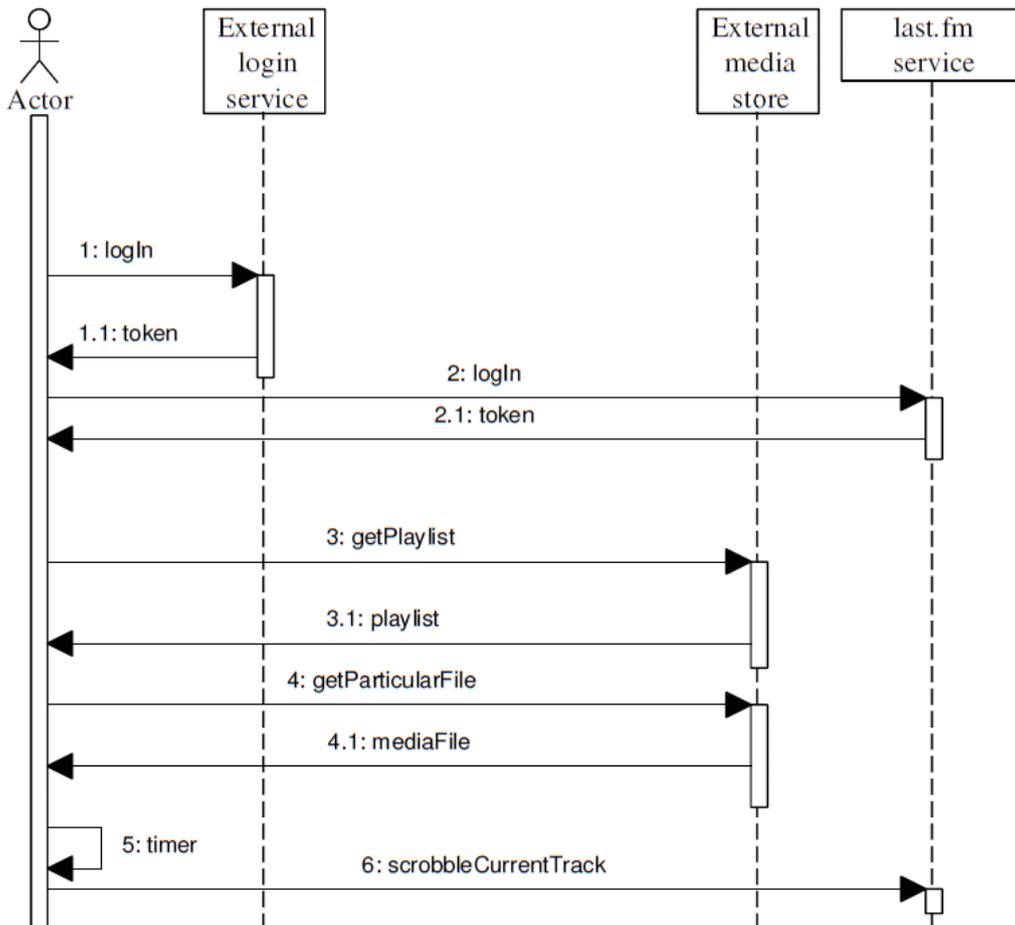


Fig. 1. Base usage sequence diagram

Let's consider each way in more details.

Personal server provides the widest range of available technologies and approaches to implement. Solution can be developed by using of httpfs, ftpfs, sshfs, webdav technologies or just by using already implemented possibility of different music players to stream current track and manage that player with client on mobile device. In fact this approach is definitely has a lot of advantages, but several disadvantages can be also highlighted. Among them security, because of necessity to open direct connection through Internet, licensing question, server availability and its maintenance.

Cloud service option implies storage and availability of streaming user's media files. That fact that streamed files was uploaded by user guarantees legality of streaming operation [2]. Due to importance of legality, this approach takes great advantage.

Among biggest service providers there are Amazon cloud player, Apple iCloud and Google music that currently on the beta testing stage. Comparison of services by these providers shown in the Table I [3].

TABLE I  
COMPARISON OF DIFFERENT CLOUD MUSIC SERVICES

	Amazon Cloud Player	Apple iCloud	Google Music Beta
Price/Availability	Free. Only for US users.	Available fall. Free for iTunes purchases.	Free. Private beta. Only for US users
Free storage allotment	5GB, plus songs purchased on Amazon	5GB	20,000 songs
Price for extra storage	\$20/year for 20 GB	Unlimited iTunes storage for \$24.95/year	unknown
Purchase Music Online	Y	Y	N
Stream music in Web browser	Y	N	Y
Upload music not purchased in the service	Y	\$25/year	Y
Sync music to iOS app/Android app	N/Y	Y/N	N/Y
Store data other than music	Y	Y	N
Require desktop app for music uploads	N	Y	Y
Number of devices you can download a song to	8	10	8
Create playlists/auto pls.	Y/N	Using iTunes	Y/Y
File types supported	MP3, AAC	MP3, AIFF, WAV, MPEG-4, AAC	MP3, AAC, WMA, FLAC

The biggest disadvantage of these kinds of services is a lack of open API for third party developers. So implementation can be done only via simulation of user interactions via HTTP or with reverse engineering of provider protocols.

The last available way to retrieve media content is to get it from Internet search. This solution implies presence of playlist so that all media files can be found via Internet search on special services (e.g. Vkontakte), torrent trackers or directly in user open folders. But this approach suffers from illegality in content and service usage.

### III. CONCLUSION

So now it's possible to conclude, that there are ways to implement cross-platform solution for music listening that shares single media content library among all clients. Difficulties of this process were considered such as copyright problems, lack of API from service providers and others. But nevertheless implementation of this concept is has to be done and major service providers such as Google, Amazon and Apple making considering steps in this direction.

### REFERENCES

- [1] "Gartner Says Cloud Computing Will Be As Influential As E-business," <http://www.gartner.com/it/page.jsp?id=707508>
- [2] "MP3Tunes 'Safe Harbor' Challenge Is Legal Test for Cloud Storage," <http://www.dailybinarynews.com/?p=6815>
- [3] "Apple iCloud vs. Amazon Cloud Player vs. Google Music Beta," <http://www.pcmag.com/article2/0,2817,2386491,00.asp>