# Project Architecture



### MediaPlayer

Thread/Activity?

Plays a list of files from a current playlist.

Allows control via an interface (or events?):

- Shuffle

- Volume control

- Repeat (one/all/none)

- Get playlist

- Set playlist

## Playlist

Stores a list of FileDescriptors or paths to media files.

Allows random access, iteration, shuffle and set as favorite/priority playlist (will show this somehow or put them at the top of the playlists list) - maybe allow also tagging media this way?

## MediaProperties

Represents a playable entity - usually an audio file, but in the future might be video, picture or something else.

For audio contains all the possible information:

- Artist, album, track number, track name etc.

- Length, sample rate etc.

- Other tags.

## SettingsProfile

A profile contains a list of settings.

That can be used to set different settings for different activities the user participates in. For example, when driving the user might want a tap to pause/play and while running he might want the tap to play the next song.

Maybe it will allow you to assign a default playlist, file or priority playlists for this profile (for example, if you'd like a specific playlist for Jogging and another for Driving).

## SettingsContainer

Contains various settings:

- last playback options (shuffle, repeat etc.)

- tap configurations - mapping of physical configurations to MediaPlayer actions.

- voice configurations

\* Must allow to set each input, that works when device is locked, to make it not work when the device is locked - sometimes you really want the device to be locked...

Some settings might be set to "unspecified" meaning it will use the default settings (this can also be configured but in the default there is no "unspecified" option)

## SettingsActivity

Allows setting the settings in the SettingsContainer :)

Also allows managing profiles (saved settings - presets)

I see the mapping of actions to taps/gestures like a controls mapping screen in computer games where you could choose from a list of supported taps/gestures/keys and maybe map several input ways for the same function.

## MediaBrowser

An activity that allows browsing the media files in the device (or maybe on the user's profile on our online store via the internet).

Might be an abstract class to allow different browsing modes - records browsing (a la iPhone), list browsing or online shop browsing (like iTunes or something).

## MediaPlayerUI

The interface of the MediaPlayer - will allow controling it using buttons and other GUI elements.

## TapInterceptor

Uses the sensors to identify taps (even when the device is locked).

Will raise an event when it registers a tap, but how will it communicate with other elements?

## DirectionTapInterceptor

Uses the sensors to identify taps on specific sides/areas of the device (even when the device is locked).

Will raise an event when it registers a tap, but how will it communicate with other elements?

Will pass the direction/location information in the event somehow.

## FlickInterceptor

Uses the touch event or any other way to identify flicks and their direction (even when the device is locked? Maybe with using the sensors...).

Will raise an event when it registers a flick, but how will it communicate with other elements?

Will pass the direction/location information in the event somehow.

## VoiceInterceptor

Uses the speaker and voice recognition(?) to identify special user recorded commands to control the device (even when the device is locked?).

Will raise an event when it registers a command, but how will it communicate with other elements?

Will pass the command information in the event somehow.

## InputEventTranslator

Receives all the input from the interceptors (also keyboard events or other available device buttons) and translates them using the settings to MediaPlayer actions and communicates those requests to the MediaPlayer.

All interceptors will implement the same interface and will be registered somehow to communicate with the input translator. They will declare what kind of input they intercept so it could appear in the settings.

# Development Plan

