

Interactive Mobile Musical Application using faust2smartphone

WENG Ruolun

allen1991shcm@gmail.com

Shanghai Conservatory of Music, Shanghai, CHINA



上海音乐学院
SHANGHAI CONSERVATORY OF MUSIC

ABSTRACT

We introduce faust2smartphone, a tool to generate an edit-ready project for musical mobile application, which connects Faust programming language and mobile application's development. It is an extended implementation of faust2api. Faust DSP objects can be easily embedded as a high level API so that the developers can access various functions and elements across different mobile platforms. This paper provides several modes and technical details on the structures and implementation of this system as well as some applications and future directions for this tool.

BACKGROUND

Mobile devices are increasingly used as musical instruments in the context of interactive performances and installations. Current real-time audio or DSP (Digital Signal Processing) API (Application Programming Interface) provided by common development environments are written in different programming languages and not easily approachable by composers and sound engineers of interactive electronic music.

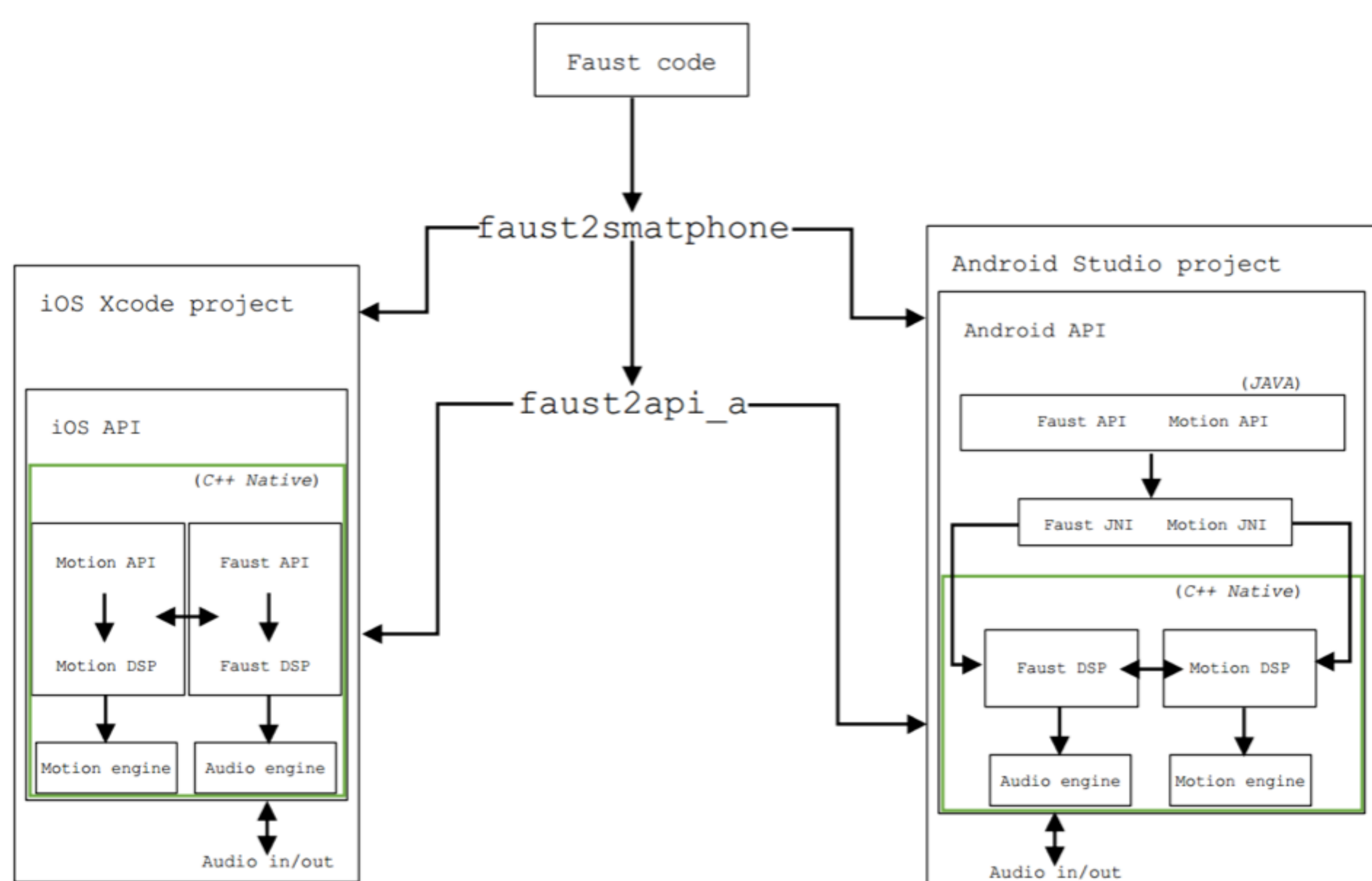
Faust^[1] (Functional Audio Stream) is a functional programming language for sound synthesis and audio processing with a strong focus on the design of synthesizers, musical instruments, audio effects, etc. Faust targets high-performance signal processing applications and audio plug-ins for a variety of platforms and standards. The goal of the faust2api is to provide a tool to easily generate custom APIs based on one or several Faust objects.

We introduce faust2smartphone, a tool to generate editable musical mobile application projects using the Faust programming language. faust2smartphone works as an extension of faust2api. Faust DSP objects can be easily embedded as a high level API so that developers can access various functions and elements across different mobile platforms. We wanted to extend the capabilities of faust2api by adding more specific functions to facilitate the development of musical mobile applications. In this paper, we present faust2smartphone which provides the same features on iOS and Android (Windows phones are not supported yet). For now, faust2smartphone is a separate branch and maintained on Github. Normally it should work with the latest version of the Faust official branch.

You can find all the source of this project on

<https://github.com/RuolunWeng/faust2smartphone.git>

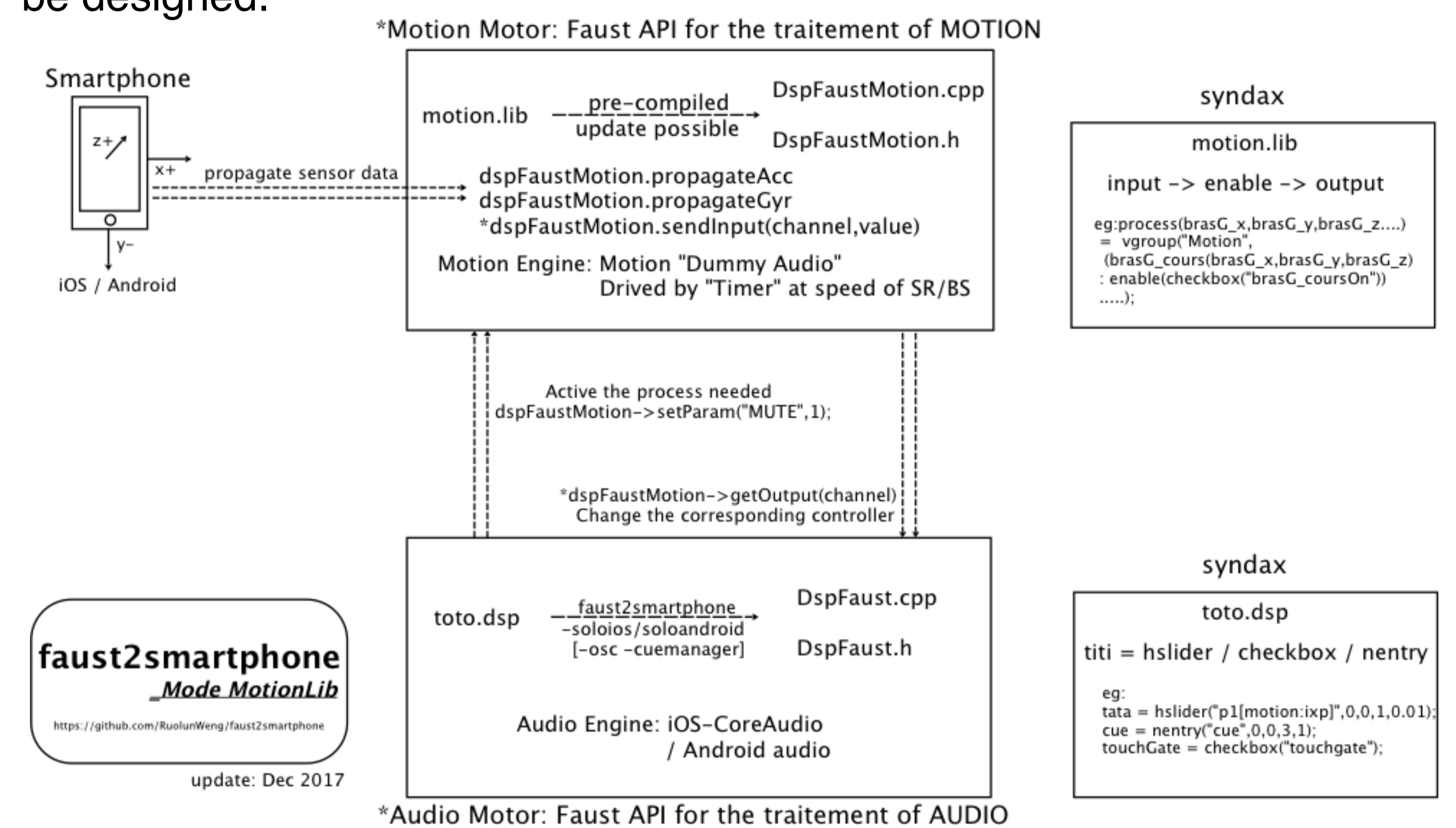
OVERVIEW



Followed by the installation instruction of Faust and faust2smatphone, you are ready to explore the function by simply taping in your terminal "faust2smartphone -help" for the details. As faust2smartphone is designed for iOS and Android, "-ios, -iosmotion, -iosplugin" and "-android, -androidmotion, -androidplugin" will guide you to the target.

faust2smartphone inherits from faust2api, so almost all the options for mobile systems are ready to be called, including: "-oscall/-oscalias" will activate the OSC (Open Sound Control) interface; "-soundfile" to active libsndfile support.

When simple mode is used, faust2api is automatically called and copies the generated files (e.g., DspFaust.cpp and DspFaust.h) to a template XCode or Android Studio project. That is what we call an "edit-ready" project, which bears the same name as the Faust code, embeds the Faust audio DSP engine and is ready to be used. This project is just a workplace to start, all the faust2api functions can be used and custom interfaces can be designed.



Motion mode is a special mode based on motion.lib and can be used as a platform to prototype musical applications involving motion gestures. motion.lib written in Faust uses the accelerometer, gyroscope, and rotation matrix signals provided by smartphones as an input. The output is the result of sensor's processing.

For the syntax, we need to do is a declaration in the metadata of the controller like:

```
toto=hslder("toto[motion:ixp]",0,0,1,0.01);
```

Where "motion" is the keyword, followed by which function you want to call in the motion.lib. By default, all the processes in motion.lib are muted to save CPU consumption; only if the program detects that you call the function, it will activate the corresponding process and affect this controller with the result calculated. We have some other reserved keywords declarations:

```
toto=checkbox("touchgate");
```

```
tata=nentry("cue",0,0,5,1);
```

```
titi=hslder("screenx/screeny",0,0,1,0.01);
```

This suite works with a sub-mode of motion mode, we call it cueManager. We provide a simple interface for this mode to deal with the code composed with different cues. To active cueManager, you just need to add -cuemanager in the command line.



APPLICATIONS

faust2smartphone has already been used in these productions:

"Audio Guide" is an application designed by Christophe Lebreton and me for blind person to experience a special sound map in the project created by GRAME and La Maison des Aveugles in Lyon. Check the introduction online: <http://www.game.fr/events/carte-sonore-de-traces-en-traces>.

"Virtual Rhizome" at 2018 Biennale of Music in Lyon, created by Vincent-Raphaël Carinola and Christophe Lebreton, is performed by a solo artist armed by two smartphones. You can check a video clip online: <https://www.youtube.com/watch?v=cGZB44KI9Y0>.

"sfPivoine" is a mobile application which I created for a participative "Pivone". The spectators could have an immersive and augmented experience with their participations. The application is both available at App Store and Google Play.