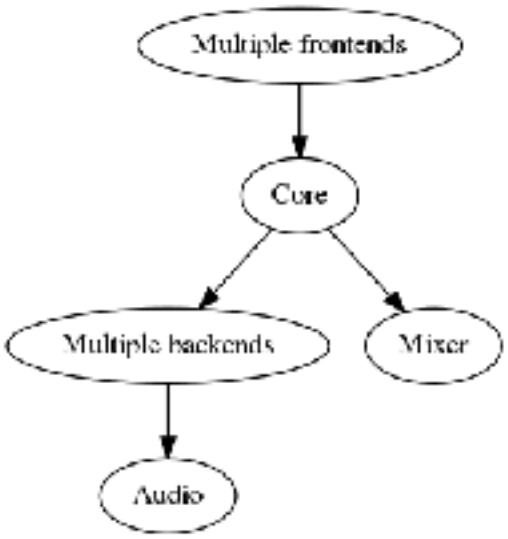
Liyi Cao
Email: caoliyi@bu.edu
Github:

Mopidy Issue #1599

What us Mopidy?

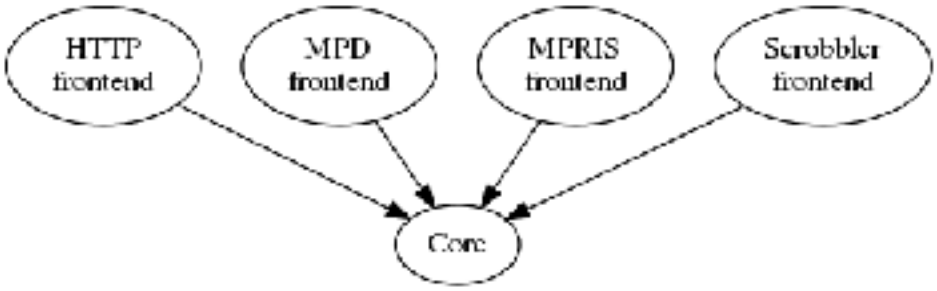
Mopidy is an extensible music sever written in Python. Mopidy plays music from local disk, Spotify, SoundCloud, Google Play Music, and more. You edit the playlist from any phone, tablet, or computer using a range of MPD and web clients. With Mopidy's extension supports, banckends for new music sources can be easily added. Mopidy is a python application that rus in a terminal or in the background on Linux computers or Macs that have network connectivity and audio output.

How Mopidy Work?

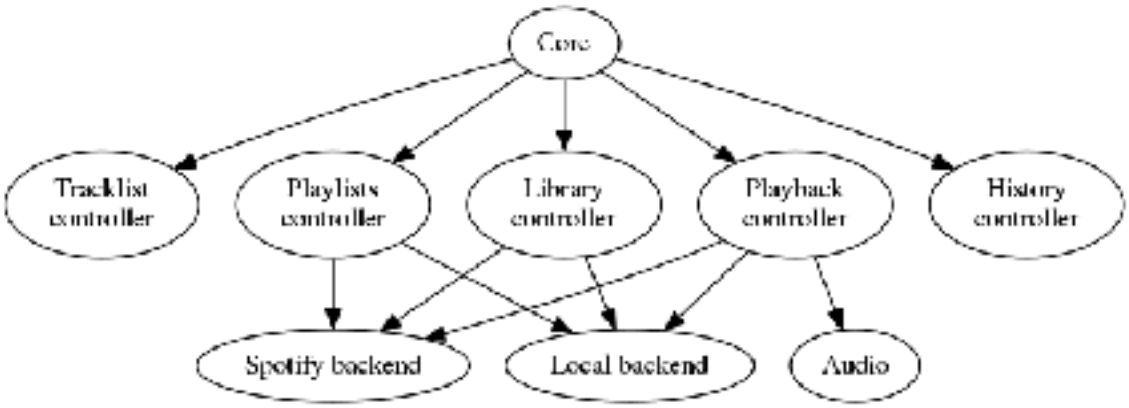


As this picture show, the architeture of Mpoidy is to connect the frontends and backends. Frontend is used for creating threads opening TCP ports, communicating with clients, and more, through the core and backend API connect with backends. The core not only controls the backend resources, but also controls some controls when

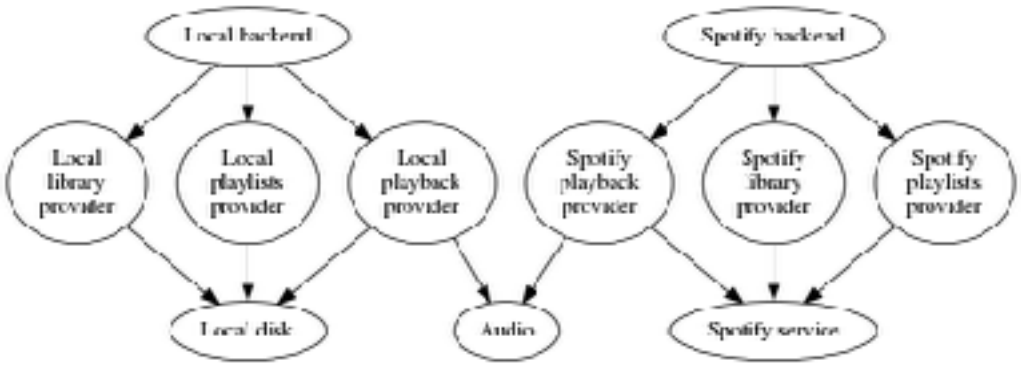
the music plays, such as volume, song switching, and so on. Multiple backends is audio from different platforms.



For the multiple frontends, Mopidy has a lot different source such as HTTP, MPD, MPRIS, Scrobbler. The user can go through the frontends give a request and signal to core. Then the core finds a corresponding backend and responds.



In the backends, the core can do tracklist, playlists and library playback history in the mixer. When connected to the backend and controlled, a local backend will be generated. These records or data will be stored in spotify and local backend. Tracklist is used to record your multiple backends and the history is to record your history record.



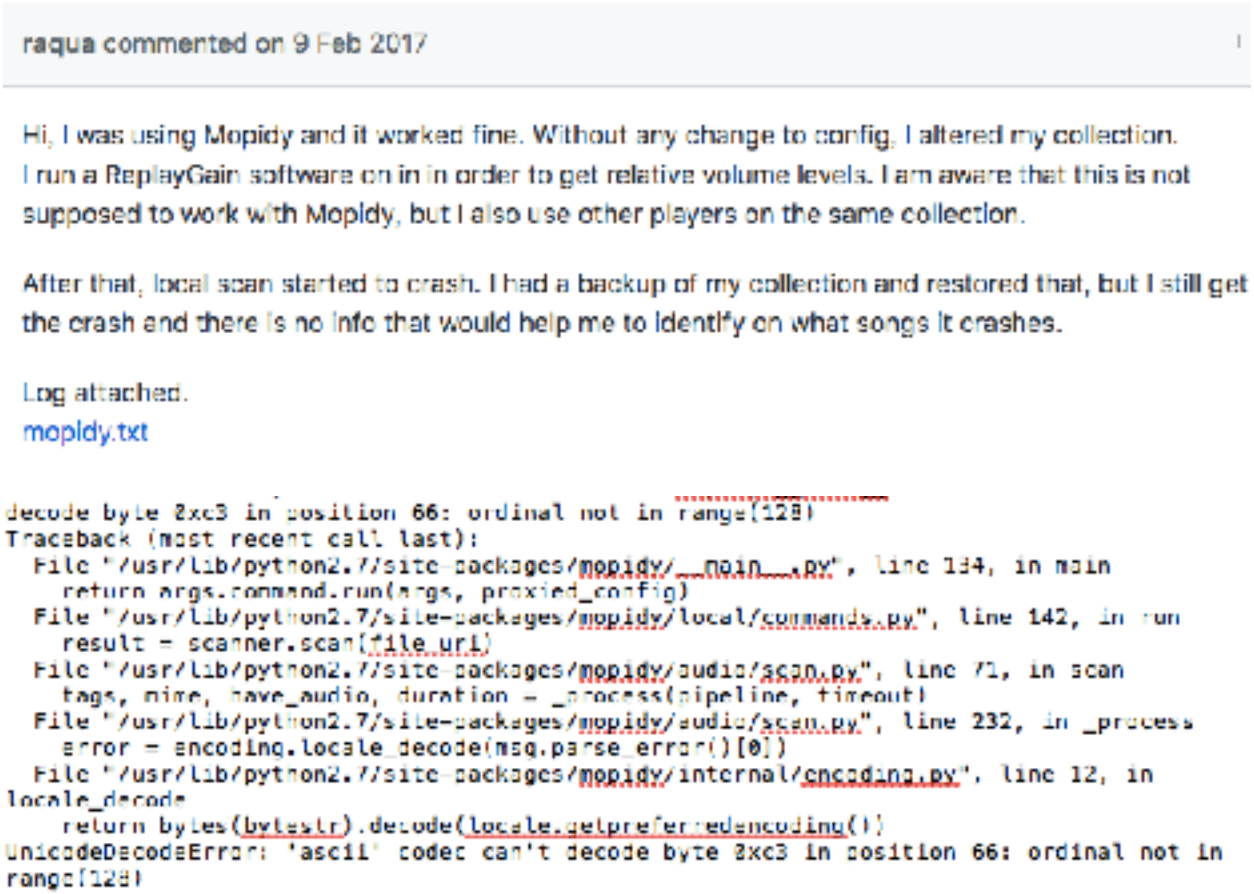
Local and Spotify backend are re two sets of similar records. They will record together the library, playlists, playback, stored separately in local disk and spotify service.

Message

The message between frontends and backends is use URI(uniform resource identifier). This kind of identification allows the user to interact with any resource (including local and Internet) through a specific protocol. The URI is defined by a scheme including a determination syntax and related protocols.

Mopidy Issue #1599

This issue is Mopidy work is good, and without change any config run ReplayGain software. After that, the local scan starts to crash.



raqua commented on 9 Feb 2017

Hi, I was using Mopidy and it worked fine. Without any change to config, I altered my collection. I run a ReplayGain software on in in order to get relative volume levels. I am aware that this is not supposed to work with Mopidy, but I also use other players on the same collection.

After that, local scan started to crash. I had a backup of my collection and restored that, but I still get the crash and there is no info that would help me to identify on what songs it crashes.

Log attached.
mopidy.txt

```
decode byte 0xc3 in position 66: ordinal not in range(128)
Traceback (most recent call last):
  File "/usr/lib/python2.7/site-packages/mopidy/__main__.py", line 134, in main
    return args.command.run(args, proxied_config)
  File "/usr/lib/python2.7/site-packages/mopidy/local/commands.py", line 142, in run
    result = scanner.scan(file_uri)
  File "/usr/lib/python2.7/site-packages/mopidy/audio/scan.py", line 71, in scan
    tags, mime, have_audio, duration = _process(pipeline, timeout)
  File "/usr/lib/python2.7/site-packages/mopidy/audio/scan.py", line 232, in _process
    error = encoding.locale_decode(msg.parse_error()[0])
  File "/usr/lib/python2.7/site-packages/mopidy/internal/encoding.py", line 12, in
locale_decode
    return bytes(bytestr).decode(locale.getpreferredencoding())
UnicodeDecodeError: 'ascii' codec can't decode byte 0xc3 in position 66: ordinal not in
range(128)
```

In the attachment document(mopidy.txt), we saw the error. The error might because the LANG variable is unset.

One suggestion given is to add a local- decode() returns a string to scan.py.The error string could be "Codec can't decode content. Check your LANG setting ($ echo $LANG).". Then I assume that the scanner stops at the track that failed.

This preliminary answer just explains why it crashed and did not solve this error. The root cause of this problem lies in the settings of UTF-8 and Lang. Finally he solved the problem with the following code：

In this process I learned:

1 How to find similar or similar errors to solve the problem.

2. Determine the cause of the problem by trying some conditions or settings.