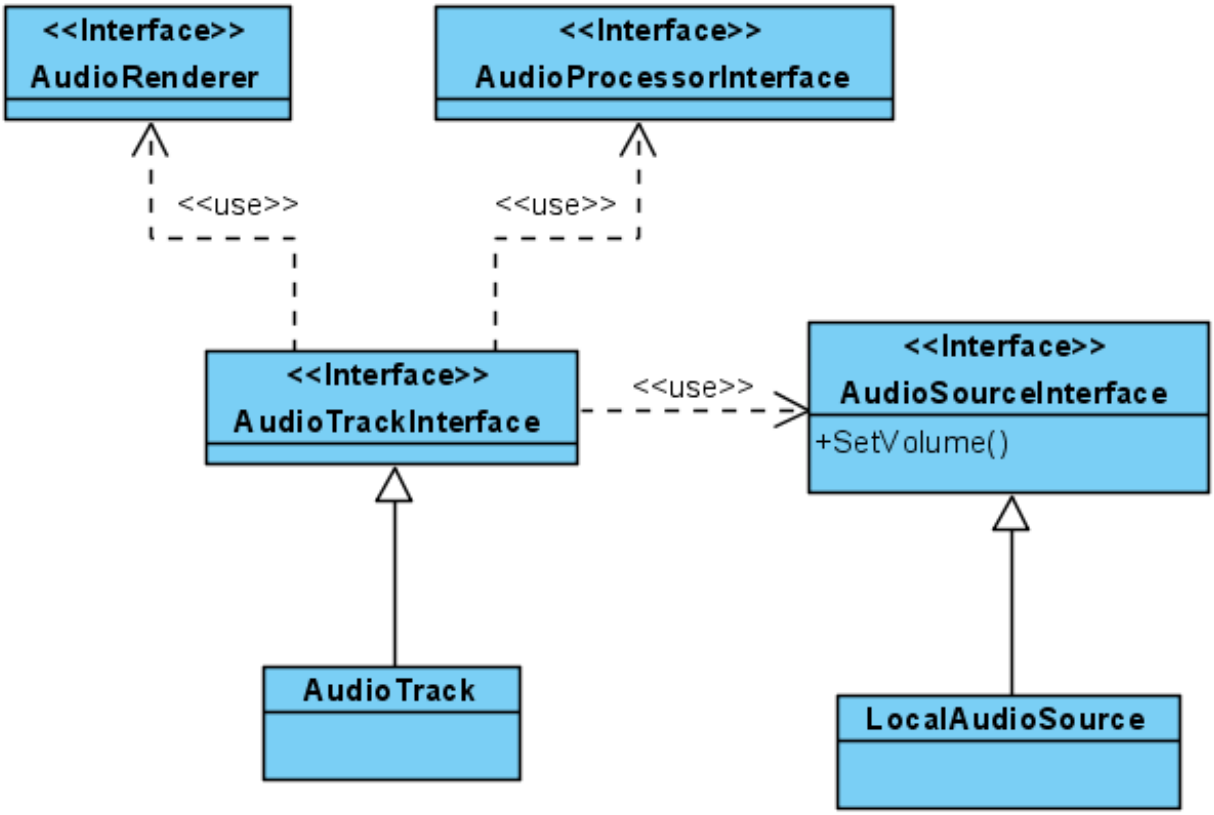
WebRTC手记之本地音频采集 - 孤竹君 - 博客园

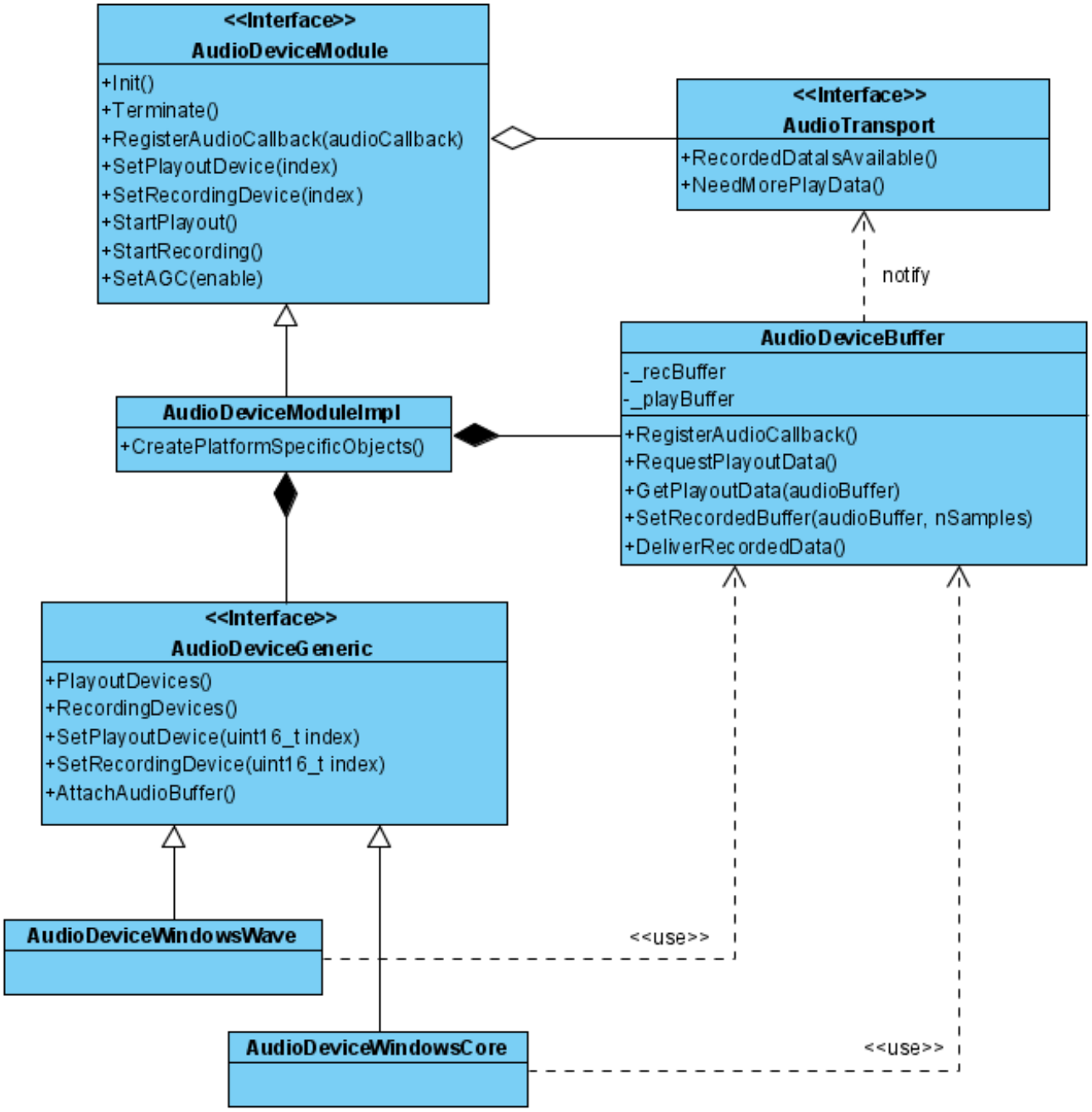
上一篇博文介绍了本地视频采集，这一篇就介绍下音频采集流程，也是先介绍WebRTC原生的音频采集，再介绍Chromium源码对它的定制。

1. WebRTC原生音频采集

先介绍一下WebRTC中与音频采集貌似相关的接口概念：



结构上看起来是不是和视频Track的结构类似？不过前面提过，如果你以对称的思维，在此结构中找出与视频track相似的采集源和输出源，那就肯定无功而返了，LocalAudioSource对AudioSourceInterface的实现就是一个空实现，没有了音频源，那音频处理接口AudioProcessorInterface和输出接口AudioRenderer都成了无米之炊了。这些接口先摆在这，可能类似于AudioCapturer的框架正在实现的途中，也可能这些接口有别的用处，比如远程音频流的抽象等，这里就暂且搁置，先记下有这回事吧。这里只谈WebRTC本地音频的采集处理。前面介绍音视频接口的时候也提到的，本地音频的采集由AudioDeviceModule接口统一封装:



AudioDeviceModule是个大而全的接口，恨不得将所有音频相关的接口都封装在里面（实际也差不多了），具体包括：枚举音频采集设备（Record）和播放设备（Playout）、设置当前的采集设备/播放设备、开始/停止音频的采集/播放、设置音频增益控制开关（AGC）等。AudioTransport是个关键的对外接口，负责音频数据的传入（调用NeedMorePlayData方法，供Playout使用）和输出（调用RecordedDataIsAvailable方法，数据由Record采集操作产生）。

AudioDeviceModuleImpl实现了AudioDeviceModule接口，创建的时候调用CreatePlatformSpecificObjects方法创建平台相关的AudioDeviceGeneric接口实现。该接口抽象了音频的采集和播放逻辑，在Windows平台下有两种实现方案：

- AudioDeviceWindowsWave实现的是传统的Windows Wave APIs方案。
- AudioDeviceWindowsCore实现的是Vista之后才支持的Windows Core Audio APIs方案。

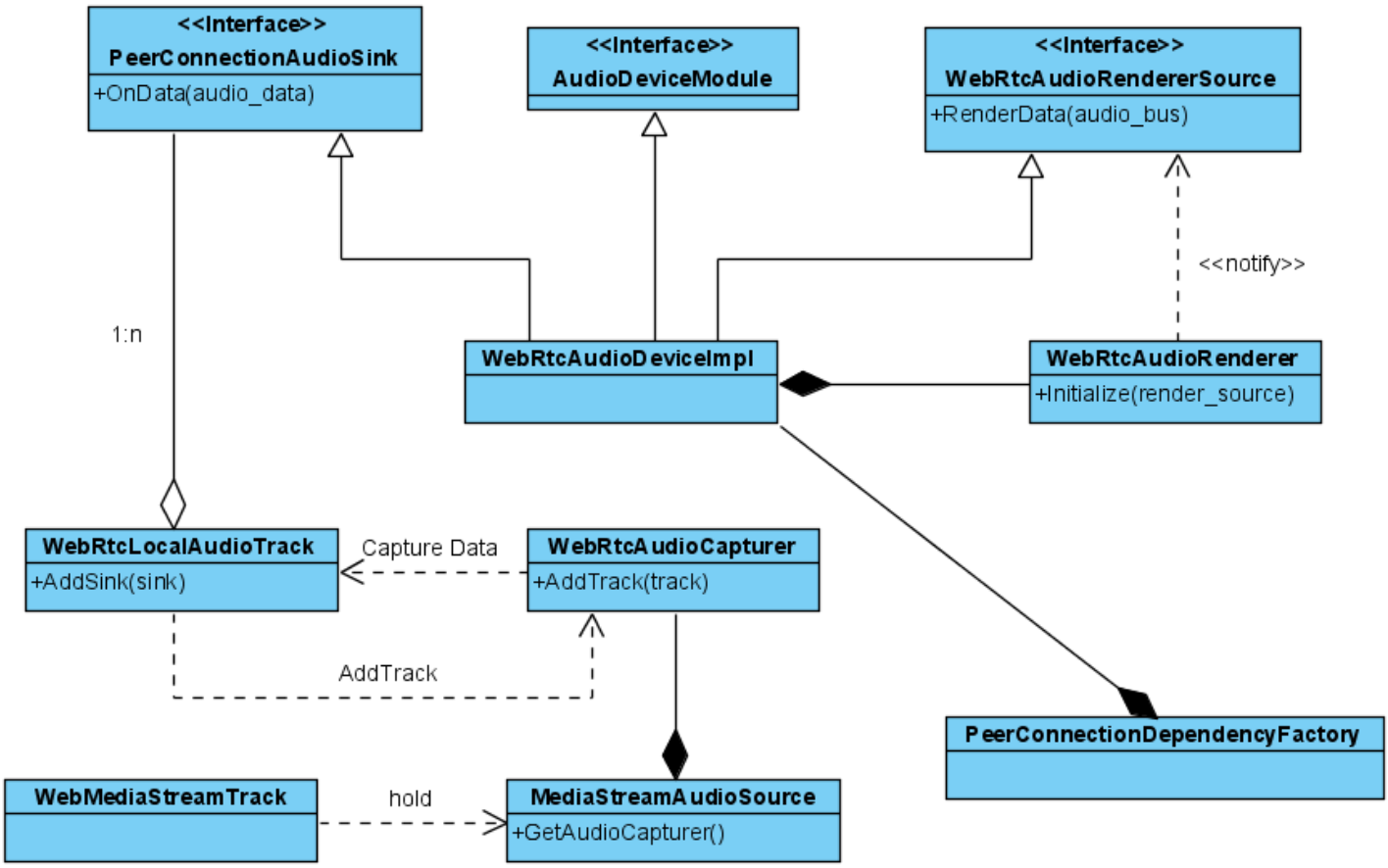
此外，AudioDeviceModuleImpl还维护了一个AudioDeviceBuffer对象来管理音频数据的缓冲区，由它直接与对外接口AudioTransport交互。比如：

- 当AudioDeviceWindowsWave或者AudioDeviceWindowsCore需要播放音频数据的时候，会调用AudioDeviceBuffer的RequestPlayoutData方法请求播放数据，然后通过GetPlayoutData方法来获取刚请求到的数据。AudioDeviceBuffer的RequestPlayoutData就是调用AudioTransport接口的NeedMorePlayData方法来请求待播放的音频流数据。
- 当AudioDeviceWindowsWave或者AudioDeviceWindowsCore采集到音频数据后，会调用AudioDeviceBuffer的SetRecordedBuffer方法将采集到的音频数据传递进去，然后调用DeliverRecordedData方法来派发出去，该派发方法就是通过调用AudioTransport接口的RecordedDataIsAvailable来实现。

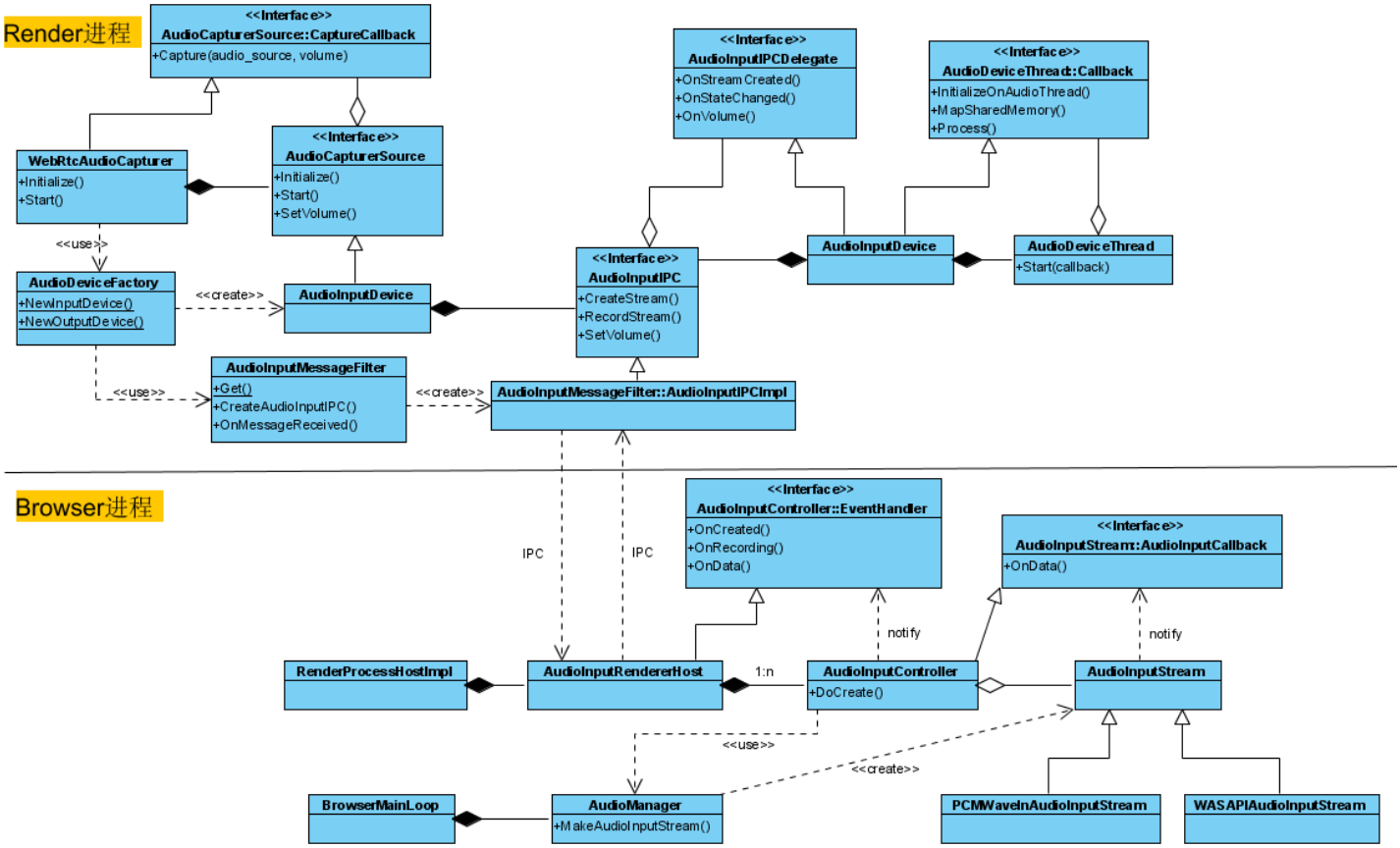
总之，音频采集模块处处都透露出大而全的结构设计。如果可以，真的应该细化一下概念设计，比如将音频采集和音频播放逻辑分离、音频输入和输出的接口拆分等等，那样才能谈得上结构设计。

2. Chromium对WebRTC的音频采集适配

根据WebRTC的本地音频接口设计，Chromium提供了一个WebRtcAudioDeviceImpl类来实现AudioDeviceModule接口，该类对象由PeerConnectionDependencyFactory负责创建和维护，结构如下：



如图所示，WebRtcAudioDeviceImpl摒弃了原生的AudioDeviceModuleImpl实现中大而全的设计，而是将音频采集和音频渲染逻辑分开，分别对应于WebRtcAudioCapturer和WebRtcAudioRenderer。WebRtcAudioRenderer通过WebRtcAudioRendererSource接口的RenderData方法向WebRtcAudioDeviceImpl请求音频流数据来渲染，WebRtcAudioDeviceImpl将该请求转发给前面提到的对外交互接口AudioTransport。WebRtcAudioCapturer封装音频采集逻辑，它将采集到的数据通过WebRtcLocalAudioTrack对象所持有的PeerConnectionAudioSink接口派发出去，WebRtcAudioDeviceImpl正是实现了该接口来接收音频采集数据，然后也是通过AudioTransport接口往外传递。至于WebRtcAudioCapturer对象的持有者MediaStreamAudioSource和WebMediaStreamTrack，这里暂时有个概念就行，它们是Chromium对HTML5媒体流的实现接口。接下来仔细分析一下WebRtcAudioCapturer和WebRtcAudioRenderer两个关键类，毋庸置疑，它们都涉及到了特定平台实现，而且在Chromium中还跨越了Render和Browser进程。和介绍Chromium视频采集的模式一样，由于不是本文重点，这里只列出结构图，不打算详解，如果你有开发上的需要，可以照着该结构图细看源码。



这是WebRtcAudioCapturer采集音频数据的结构，牵涉到跨进程通信，结构还是非常复杂的。WebRtcAudioRenderer的结构就不准备介绍了，因为Chromium的这块设计非常具备对称性，基本上图中类命名中的Input改成Output就差不多是WebRtcAudioRenderer的架构了。