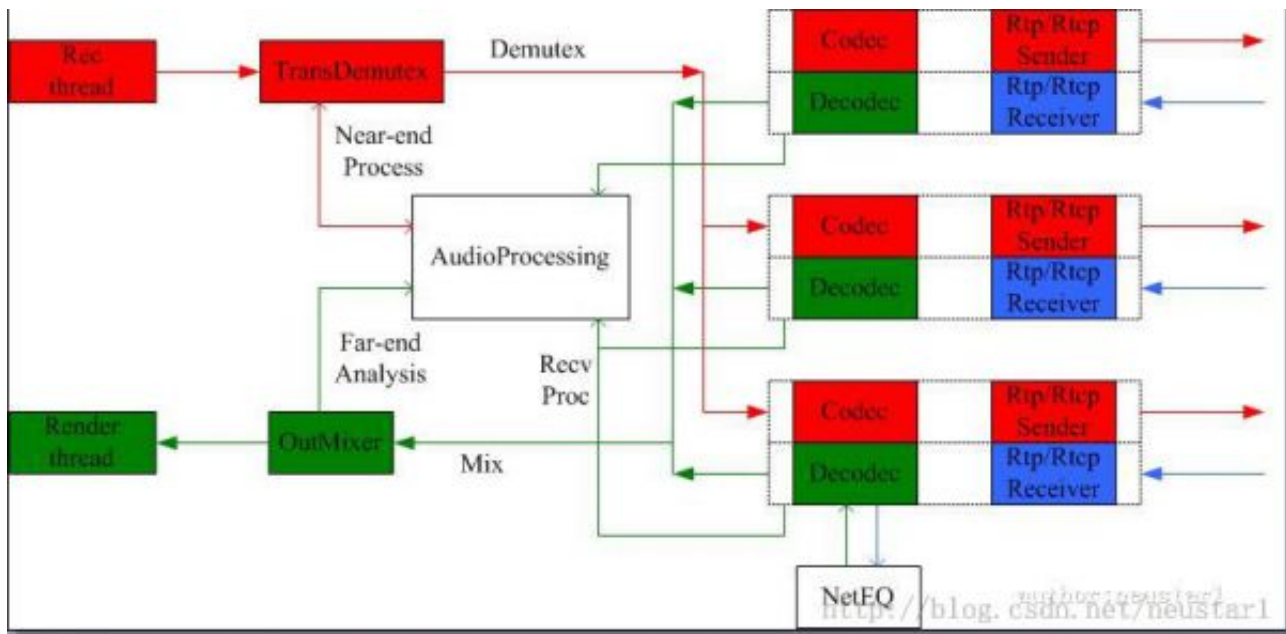


本文概要介绍WebRTC的音频处理流程，见下图：



webRTC将音频会话抽象为一个通道Channel，譬如A与B进行音频通话，则A需要建立一个Channel与B进行音频数据传输。上图中有三个Channel，每个Channel包含编解码和RTP/RTCP发送功能。

以一个Channel而言，应用程序中将包含三个活动线程，录音线程，音频接收线程和播放线程。

1) 录音线程：负责麦克风音频的采集，见图中红色路径，采集到音频后，缓存到一定长度，进行音频处理，主要包括EC，AGC和NS等。然后送到Channel，经过音频

Codec模块编码，封装成RTP包，通过Socket发送出去；

2) 接收线程：见蓝色路径，负责接收远端发送过来的音频包，解封RTP包，解码音频数据，送入NetEQ模块缓存。

3) 播放线程：负责耳机声音播放，见绿色路径。播放线程去OutMixer中获取要播放的音频数据，首先依次获取参与会话的Channel中NetEQ存储的音频帧，可以对其做AGC和NS处理；然后混合多个Channel的音频信号，得到混合音频，传递给AudioProcessing模块进行远端分析。最后播放出来。

如下为本地回环录音和播放代码:

```

VoiceEngine* ve = VoiceEngine::Create();
VoEBase* base = VoEBase::GetInterface(ve);
base->Init();
int chld = base->CreateChannel();
base->SetSendDestination(chld,3000,"127.0.0.1",4000);
base->SetLocalReceiver(chld,3000,3001,"127.0.0.1");
base->StartPlayout(chld);

```

```
base->StartReceive(chId);
```

```
base->StartSend(chId);
```

```
//....sleep...wait.....
```

```
base->StopSend(chId);
```

```
base->StopReveive(chId);
```

```
base->StopPlayout(chId);
```

```
base->Terminate();
```