



在Qt程序中集成多媒体功能

Integrating multimedia into your Qt application

胡岭 (ling.hu@nokia.com)



主要内容

- Qt多媒体系统架构
- 媒体播放器
- 媒体录制
- 相机
- 低延迟音频
- Qt多媒体对QML的支持
- 扩展Qt多媒体

你对多媒体有什么期待？



玩游戏



看电影



照相



听音乐

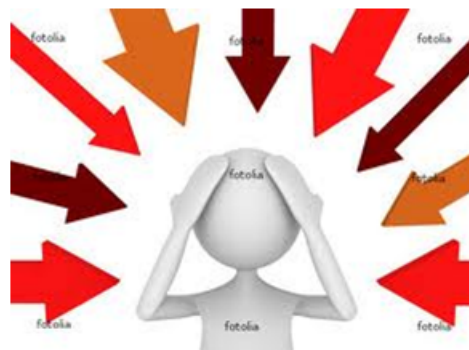


视频聊天



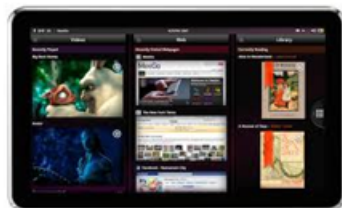
对媒体对用户而言是
奇妙多彩的体验😊

对开发者来说却是
很大的挑战!



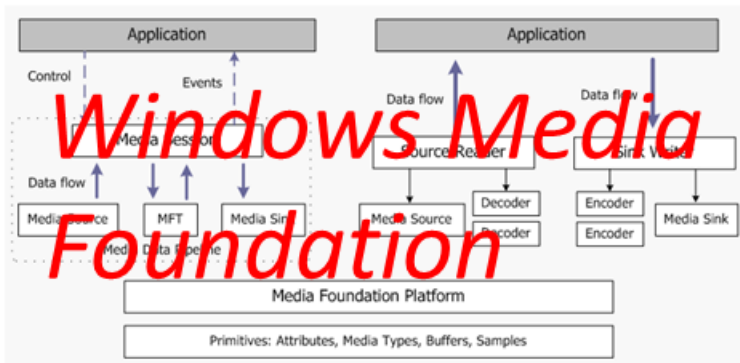
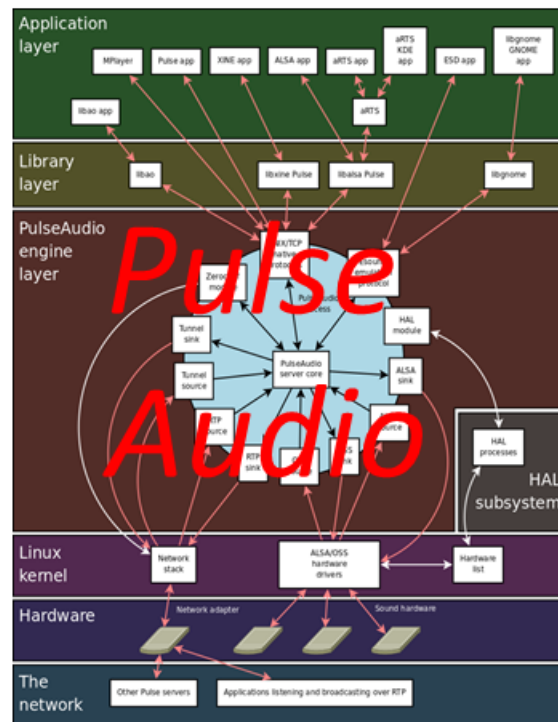
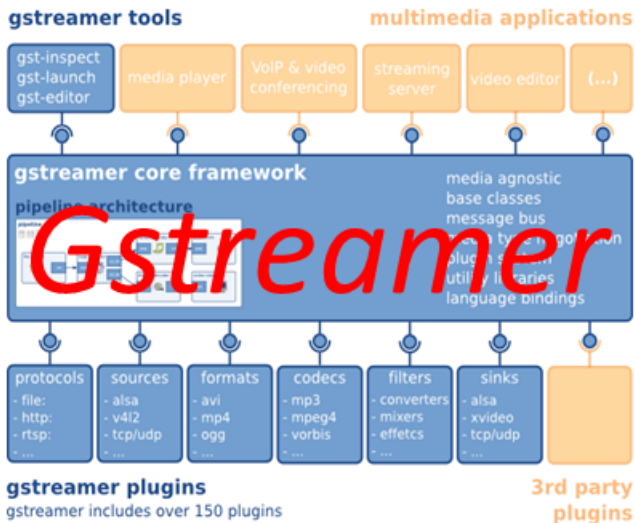


来自平台的挑战





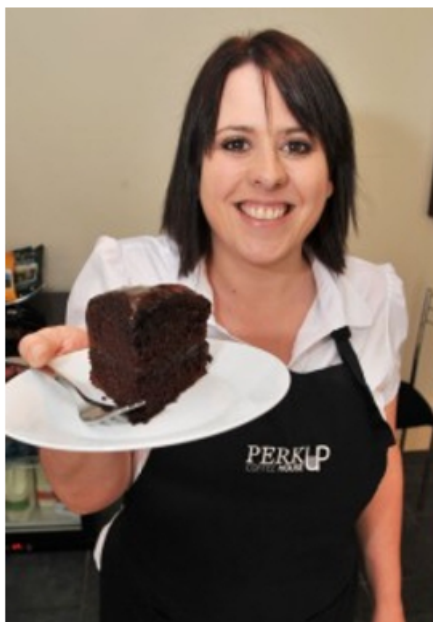
来自软件系统的挑战



但是我们有选择么？

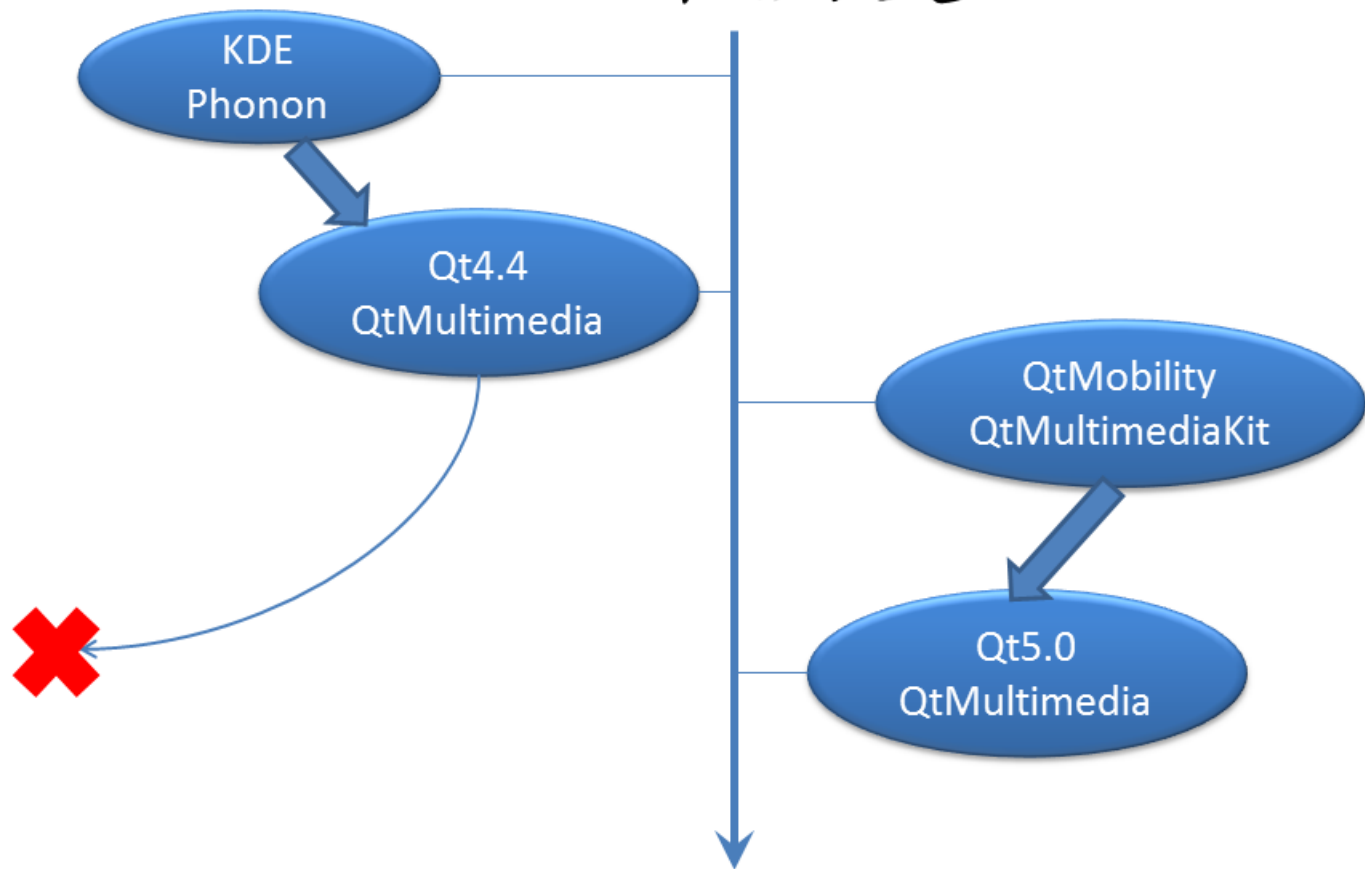


QtMultimediaKit 正是为你准备的甜蜜方案😊





一点历史





QtMultimediaKit如何帮助你？



跨平台支持

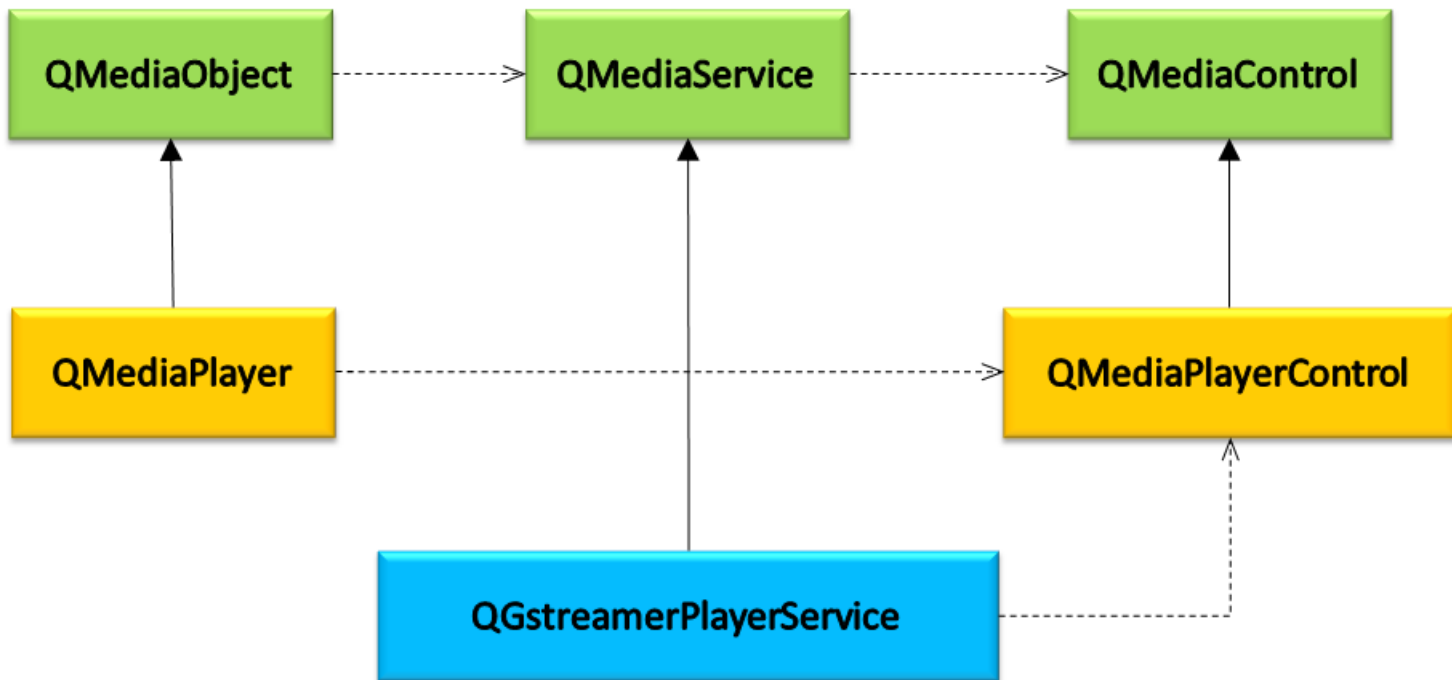


使用简单且功能灵活



通过支持QML提供快速开发

服务，控制和媒体对象





- 媒体对象 (QMediaPlayer, QCamera, ...)
 - 为应用程序服务

- 服务和控制 (QMediaService, QMediaControl)
 - 新的平台实现
 - 扩展新功能
 - 复杂的应用



QtMultimediaKit的架构

QML API

Video

Audio

Camera

SoundEffect

High Level Media API

QMediaPlayer

QMediaRecorder

QCamera

QRadioTuner

Services

Q_MEDIASERVICE_MEDIAPLAYER

Q_MEDIASERVICE_AUDIOSOURCE

Q_MEDIASERVICE_CAMERA

Q_MEDIASERVICE_RADIO

Low Latency Audio

QAudioOutput

QAudioInput



QtMultimediaKit的架构

QML API

Video

Audio

Camera

SoundEffect

High Level Media API

QMediaPlayer

QMediaRecorder

QCamera

QRadioTuner

Low Latency Audio

QAudioOutput

QAudioInput

Services

Q_MEDIASERVICE_MEDIAPLAYER

Q_MEDIASERVICE_AUDIOSOURCE

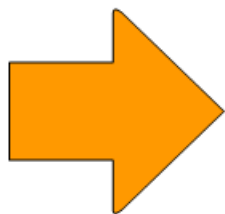
Q_MEDIASERVICE_CAMERA

Q_MEDIASERVICE_RADIO

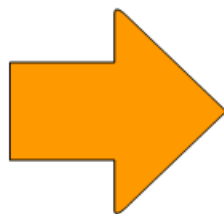
媒体播放概览



媒体来源



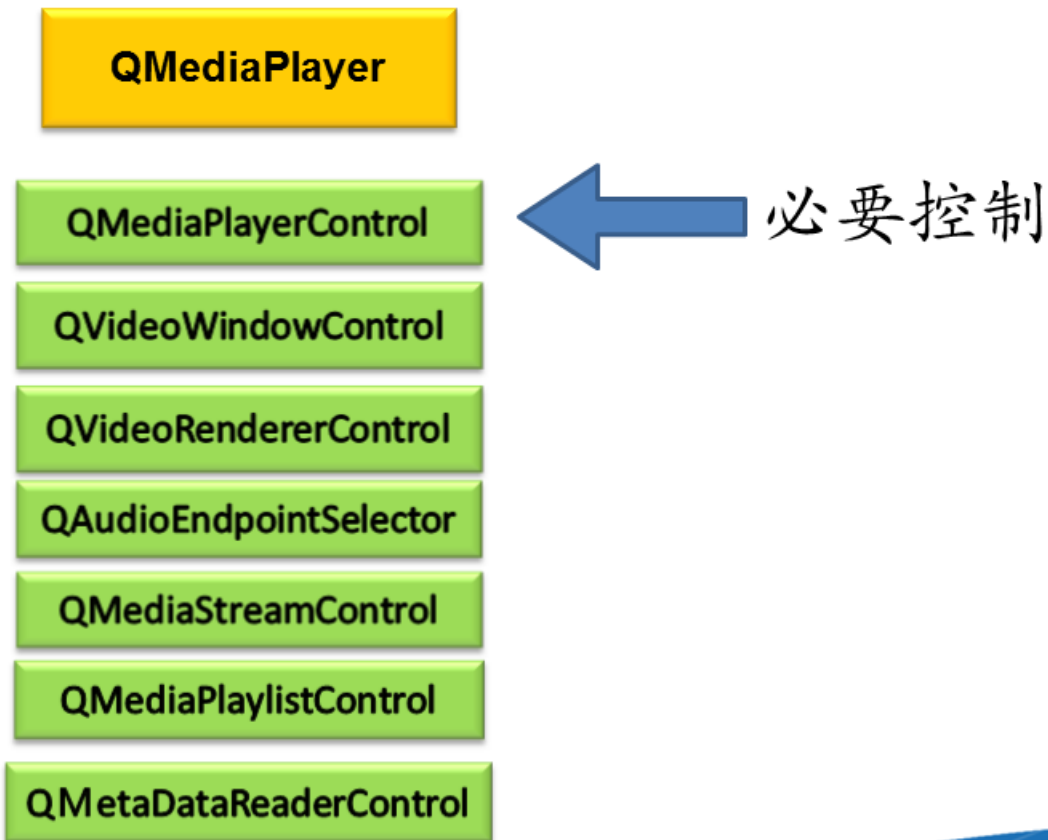
媒体播放器



图像输出



媒体播放架构





媒体播放代码示例

```
player = new QMediaPlayer;  
  
player->setMedia(QUrl("http://example.com/movie.mp4"));  
  
videoWidget = new QVideoWidget;  
player->setVideoOutput(videoWidget);  
videoWidget->show();  
  
player->play();
```

PlayerDemo



支持平台



- Linux (Gstreamer)



- Windows (DirectShow / MediaFoundation)



Mac

- Mac OS (QuickTime7)

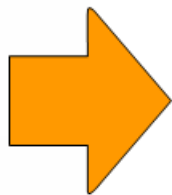


媒体播放更为灵活的接口

```
setMeida(QIODevice *stream)
```

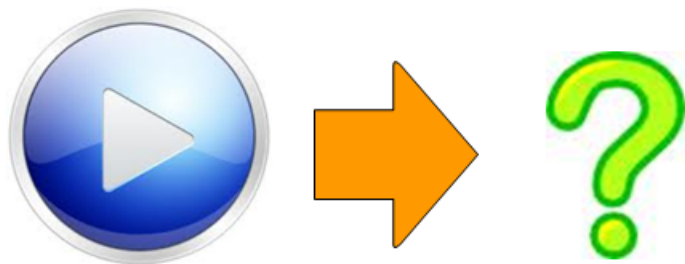


DRM



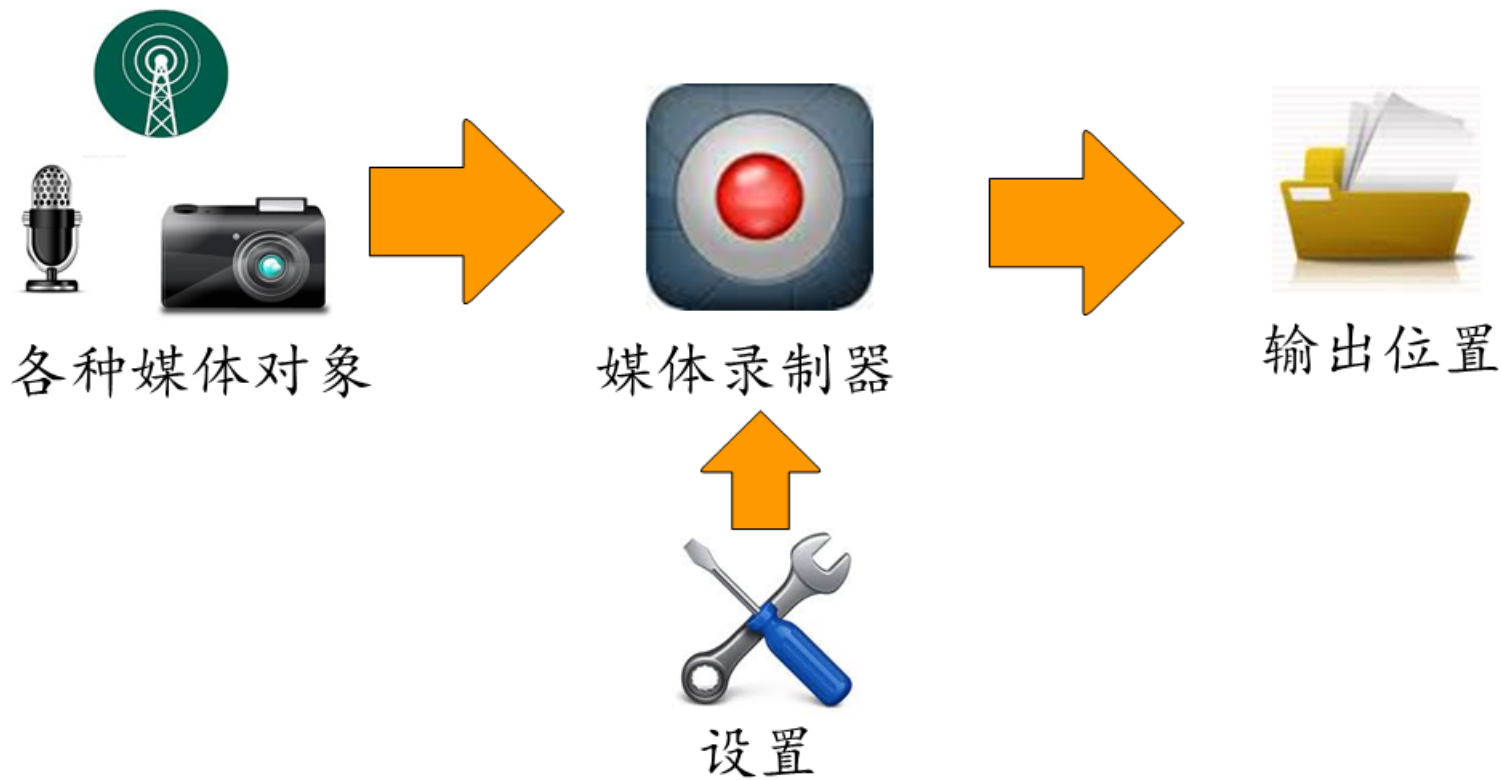


媒体播放更为灵活的接口



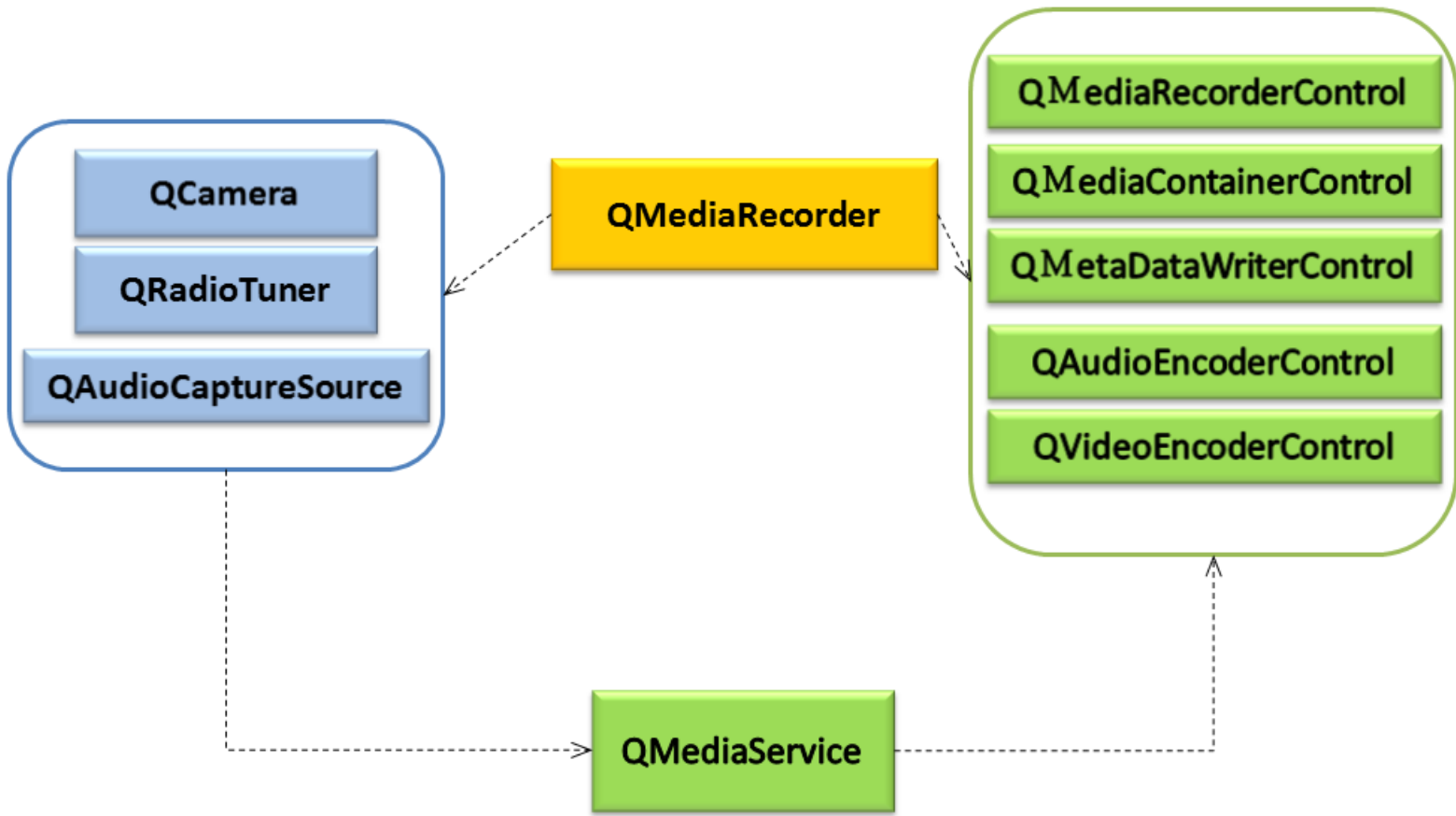
`setVideoOutput(QAbstractVideoSurface*)`

媒体录制概览





媒体录制架构

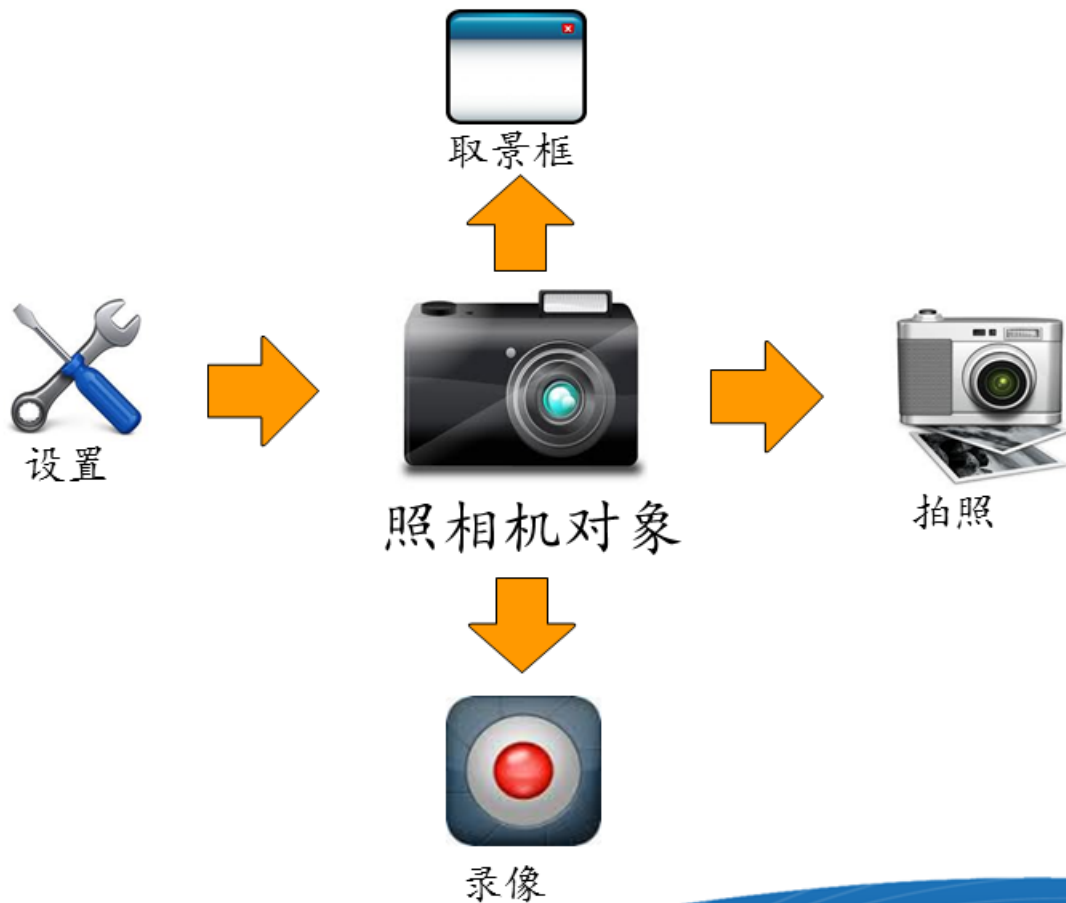




媒体录制代码示例

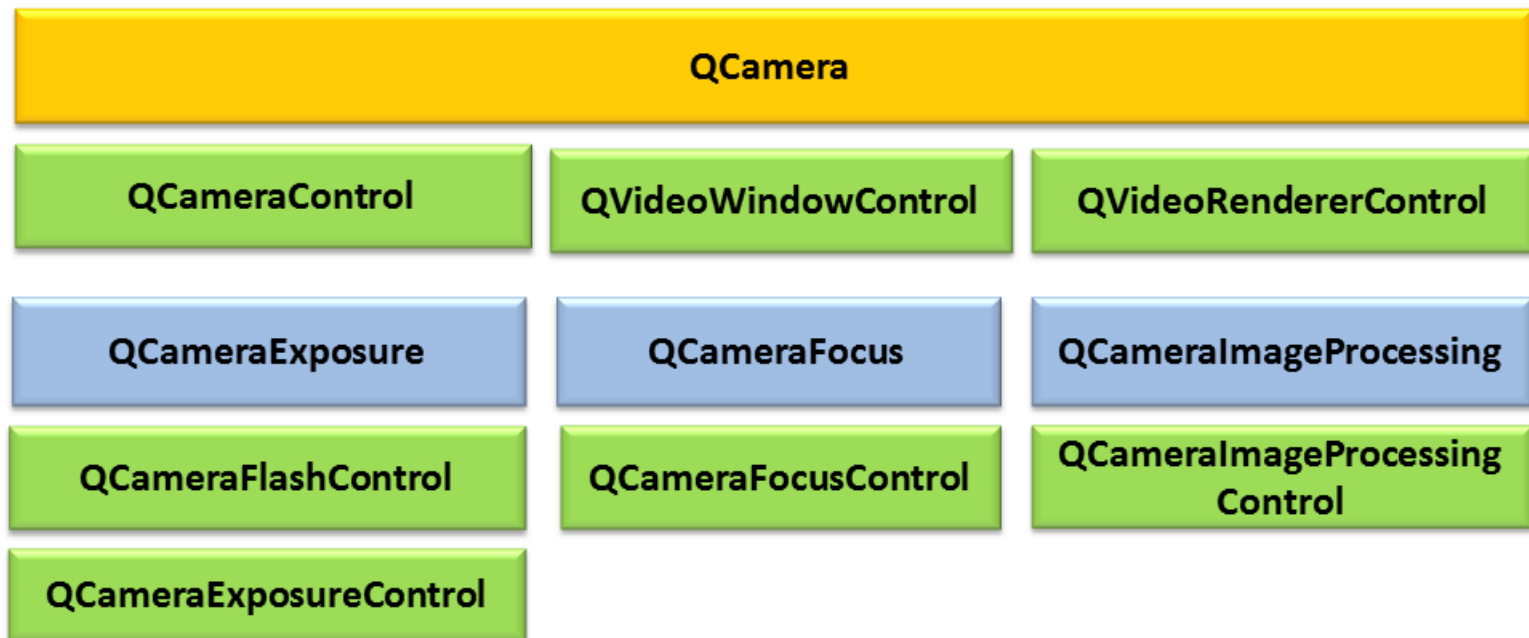
```
audioSource = new QAudioCaptureSource;  
  
recorder = new QMediaRecorder(audioSource);  
  
QAudioEncoderSettings audioSettings;  
audioSettings.setCodec("audio/vorbis");  
audioSettings.setQuality(QtMedia::HighQuality);  
recorder->setEncodingSettings(audioSettings);  
  
recorder->setOutputLocation(QUrl::fromLocalFile(fileName));  
  
recorder->record();
```

照相机概览



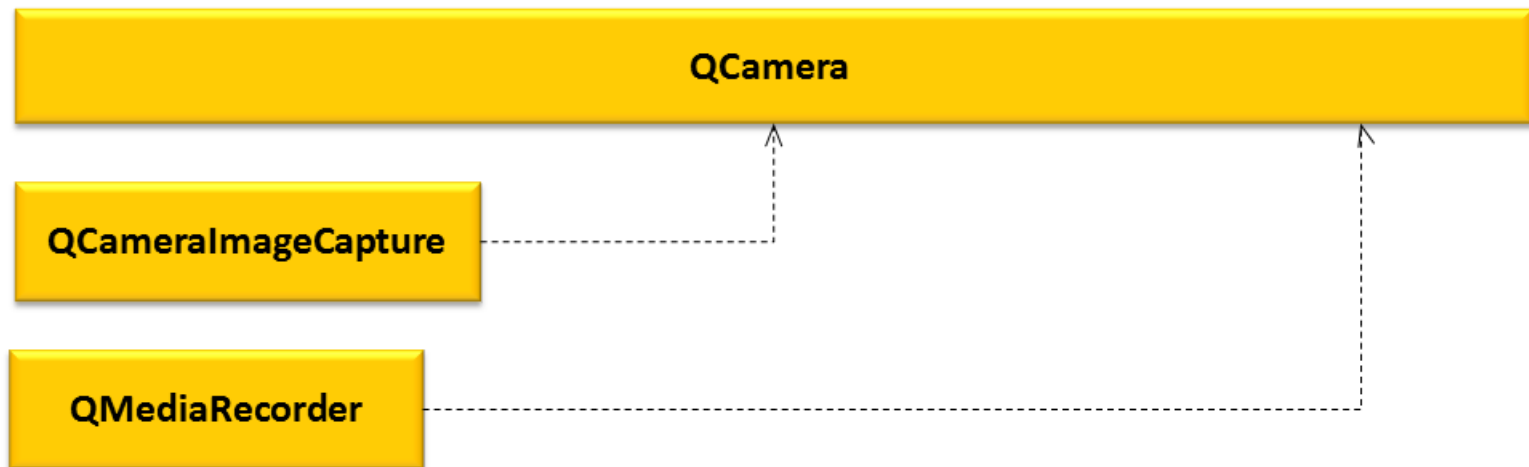


照相机架构





照相机架构





照相机代码(拍照)示例

```
camera = new QCamera;
```

```
viewfinder = new QCameraViewfinder();  
viewfinder->show();  
camera->setViewfinder(viewfinder);
```

```
imageCapture = new QCameraImageCapture(camera);
```

```
camera->setCaptureMode(QCamera::CaptureStillImage);  
camera->start();
```

```
imageCapture->capture();
```

通过
QMediaRecorder
实现摄像功能

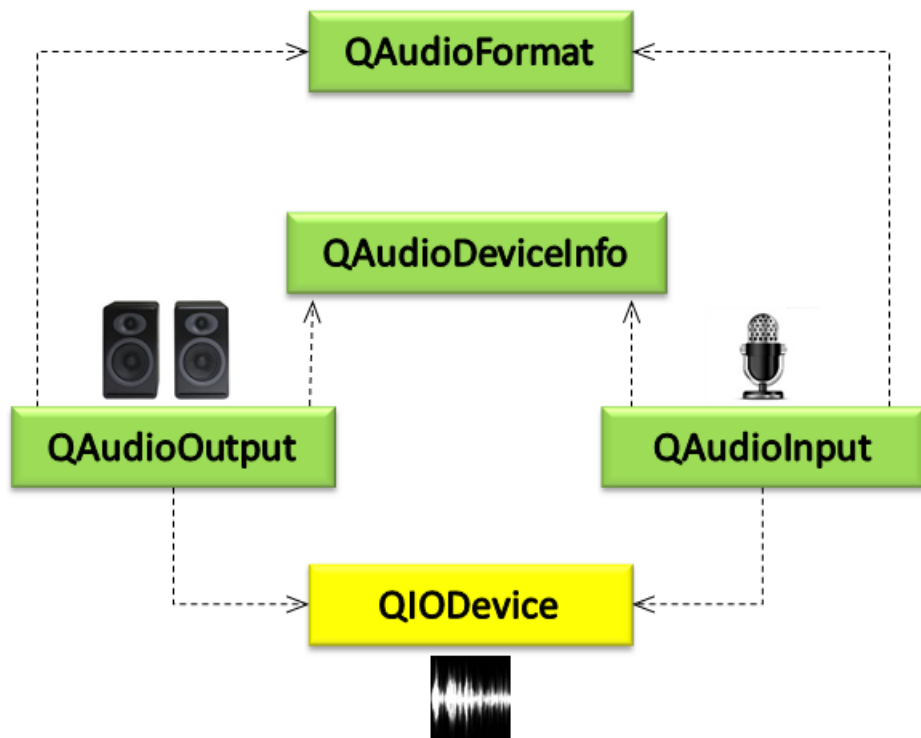
Camera

捕捉照片到内存

```
setCaptureDestinations(QCameraImageCapture::CaptureToBuffer);
```



低延迟音频概览





QAudioOutput 代码示例

```
sourceFile.setFileName("/tmp/test.raw");  
sourceFile.open(QIODevice::ReadOnly);
```

```
QAudioFormat format;  
// Set up the format, eg.  
format.setFrequency(8000);  
...  
format.setCodec("audio/pcm");
```

```
audio = new QAudioOutput(format, this);  
audio->setNotifyInterval(40);  
connect(audio, SIGNAL(notify()), SLOT(onNotify()));
```

```
output = audio->start(); // push mode
```

```
void onNotify()  
{  
    int bytesFree = audio->bytesFree();  
    while(bytesFree > 0) {  
        int l = output->  
write(myAudioData, bytesFree);  
        bytesFree -= l;  
    }  
}
```



手动提供数据



QAudioInput 代码示例

```
destinationFile.setFileName("/tmp/test.raw");  
destinationFile.open( QIODevice::WriteOnly | QIODevice::Truncate );
```

```
QAudioFormat format;  
format.setFrequency(8000);  
...  
format.setCodec("audio/pcm");
```

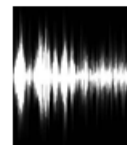
```
audio = new QAudioInput(format, this);  
QTimer::singleShot(3000, this, SLOT(stopRecording())); // Records  
audio for 3000ms
```

```
audio->start(&destinationFile); //pull mode
```



自动写入数据

用Push还是用Pull，
这是个值得考虑的问题。



QIODevice

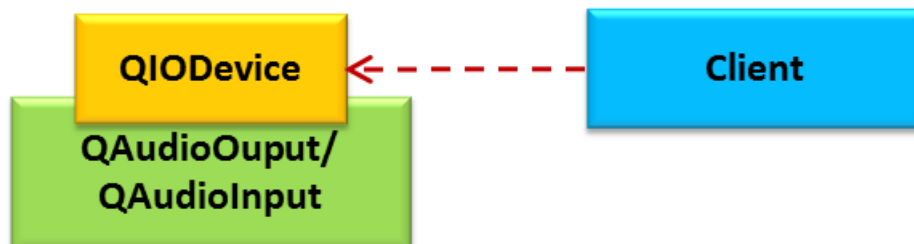




- *Push Mode*



```
QIODevice* start();
```





- *Pull Mode*



```
void start(QIODevice*);
```





Push/Pull 比较

- Pull比Push更方便
- Write比Read更高效

	Push	Pull (方便)
QAudioOutput	Write	Read
QAudioInput	Read	Write

Multimedia for QML

Code less, create more!

Audio

Video

Camera

SoundEffect



QML 多媒体代码示例

- Video

```
import QtQuick 1.1
import QtMultimediaKit 1.1
Video {
    id: video
    width : 800
    height : 600
    source: "video.avi"
    focus: true
    MouseArea {
        anchors.fill: parent
        onClicked: {
            video.play()
        }
    }
    Keys.onSpacePressed: video.paused = !video.paused
    Keys.onLeftPressed: video.position -= 5000
    Keys.onRightPressed: video.position += 5000
}
```




QML 多媒体代码示例

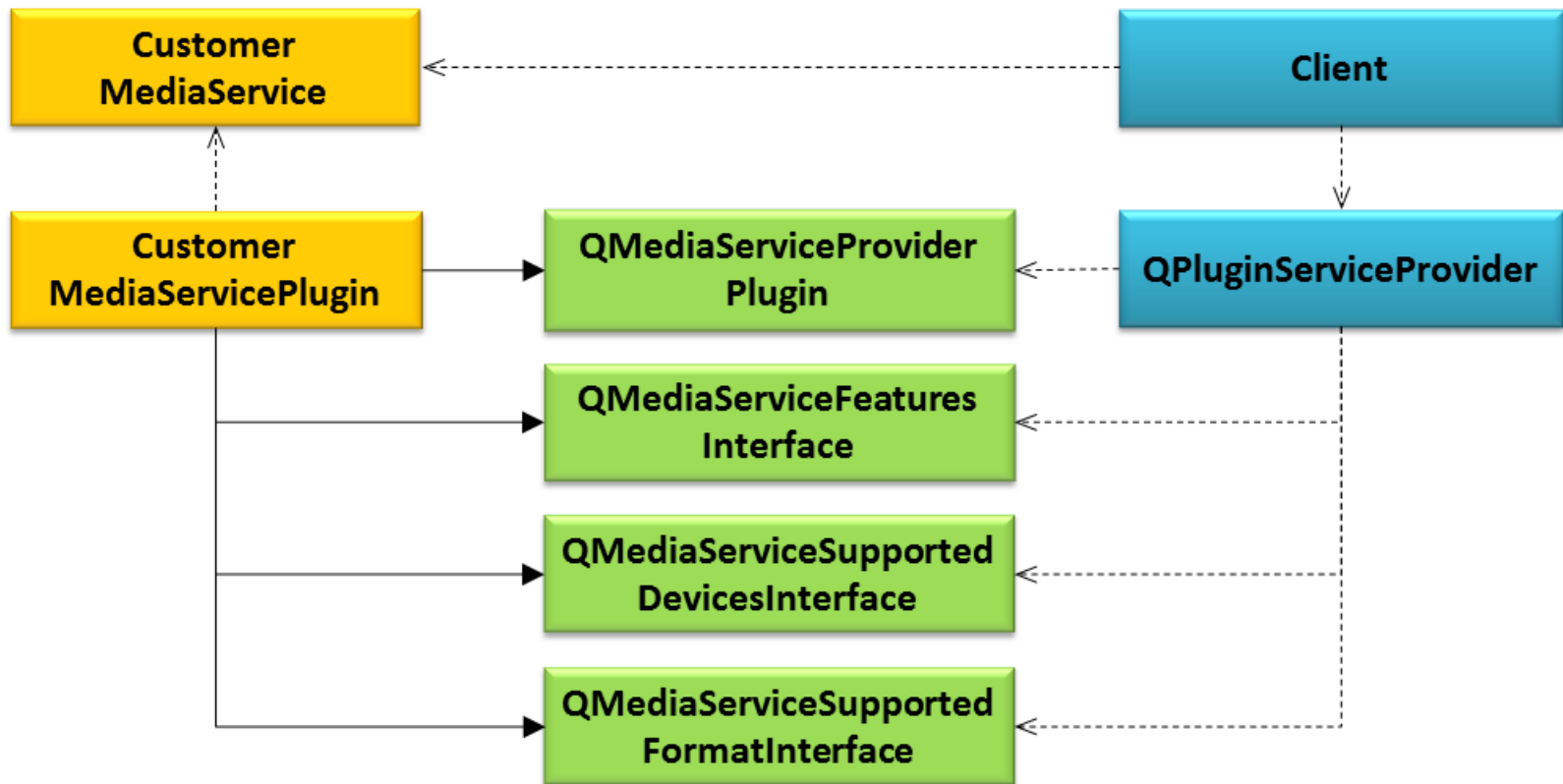
```
•SoundEffect import QtQuick 1.1
import QtMultimediaKit 1.1
Text {
    text: "Click Me!";
    font.pointSize: 24;
    width: 150;
    height: 50;
    SoundEffect {
        id: playSound
        source: "soundeffect.wav"
    }
    MouseArea {
        anchors.fill: parent
        onPressed: {
            playSound.play()
        }
    }
}
```



扩展QtMultimediaKit



MediaService插件系统



更多详细资料请参考：

<http://doc.qt.nokia.com/>



谢谢！