/\* pkg-config --cflags --libs gstreamer-1.0 这个命令可以自动找到对应的头文件和库文件。

alientek@alientek-virtual-machine:~/桌面$ pkg-config --cflags --libs gstreamer-1.0

-pthread -I/usr/include/gstreamer-1.0 -I/usr/include/glib-2.0 -I/usr/lib/x86\_64-linux-gnu/glib-2.0/include -lgstreamer-1.0 -lgobject-2.0 -lglib-2.0

alientek@alientek-virtual-machine:~/桌面$

\*/

/\*

所以编译的时候，后面加这句话就表示编译的时候使用这些对应的头文件和库文件。

‘`’这个键就是Tab上面的键

gcc hello-1.c -o hello-1 `pkg-config --cflags --libs gstreamer-1.0`

\*/

/\*

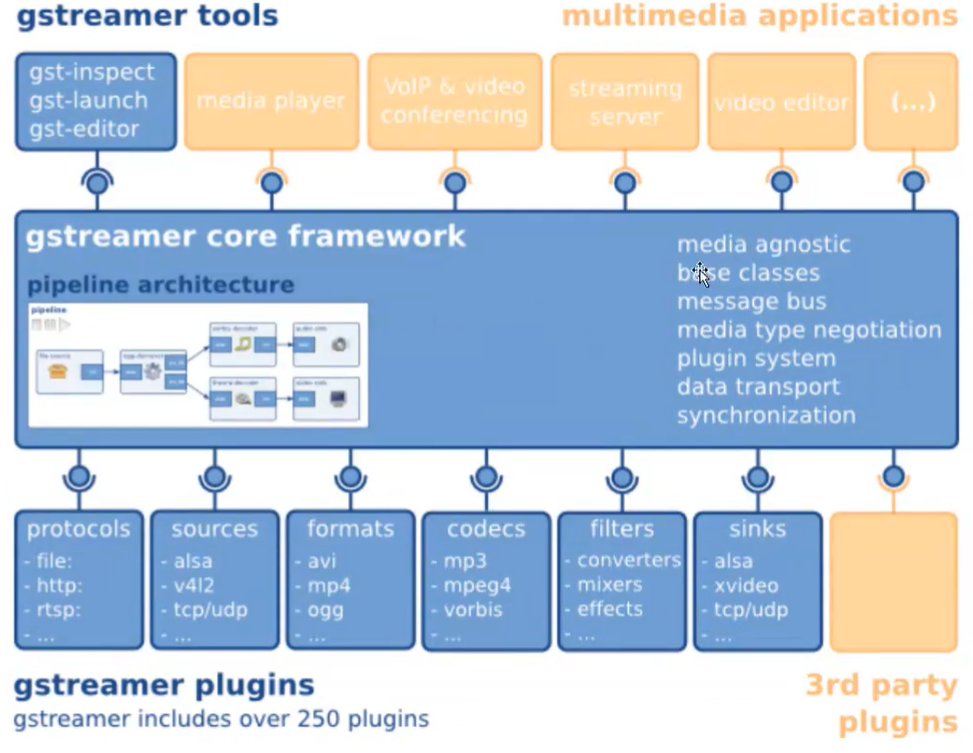


HelloWorld之前先把重要的数据结构和API了解一下：



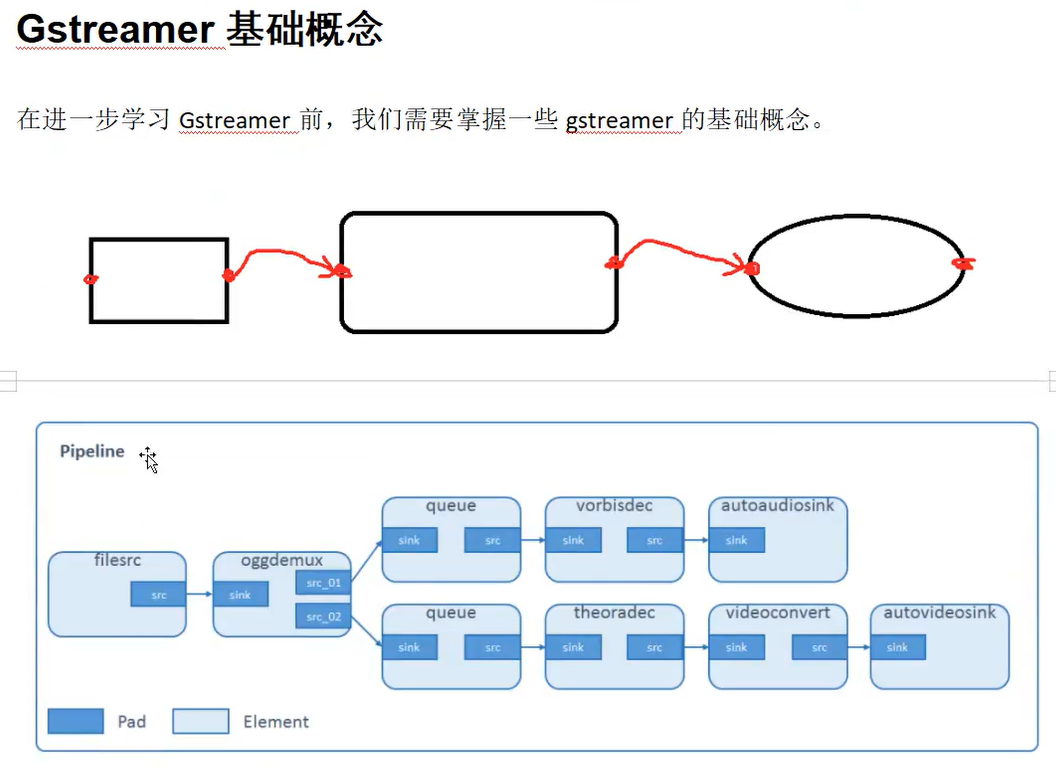
/\*

第一章里面的框架：



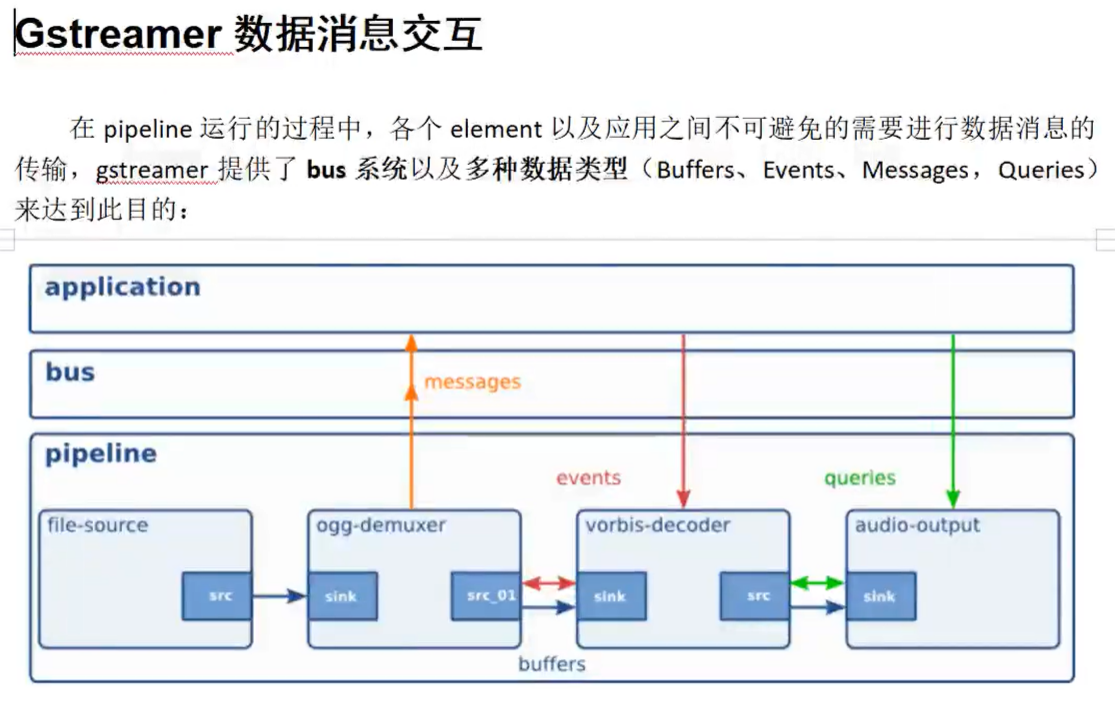
中间是framework，上层是应用，下层是插件。

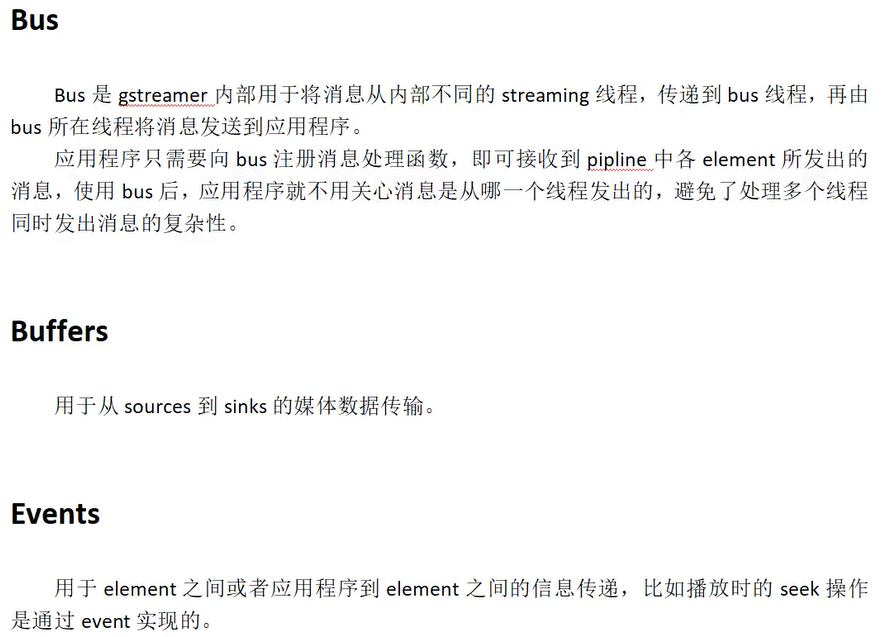
后续讲到了基本概念：



Element、Pad、bin

然后是数据交互：





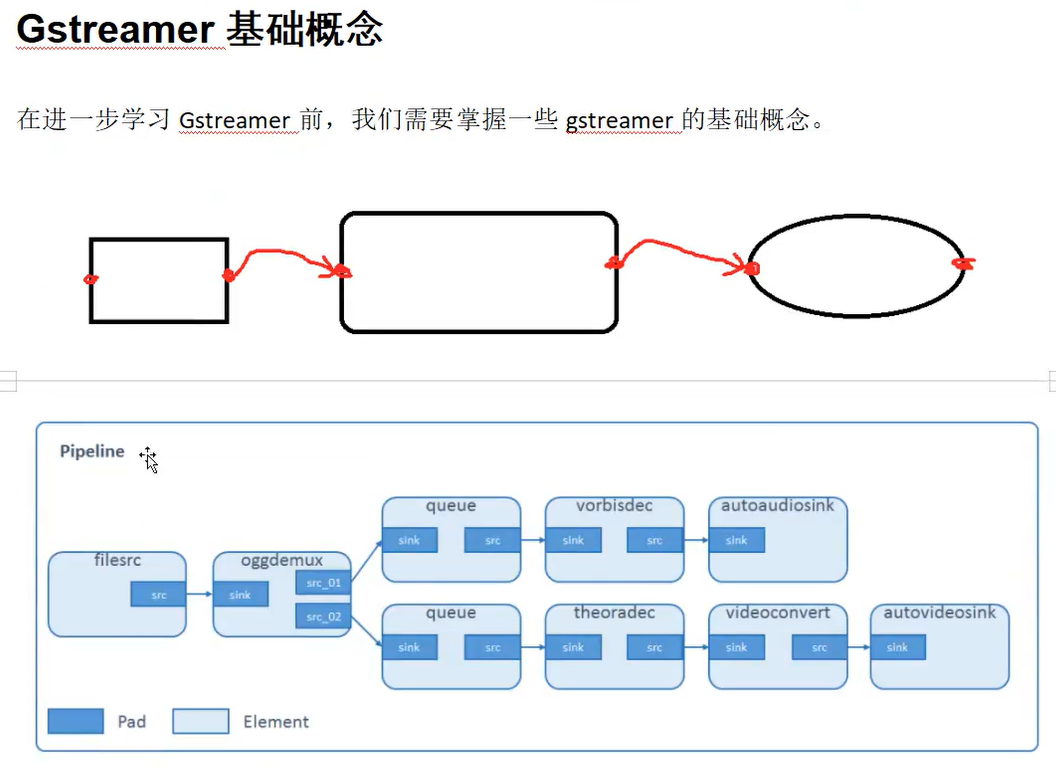


\*/

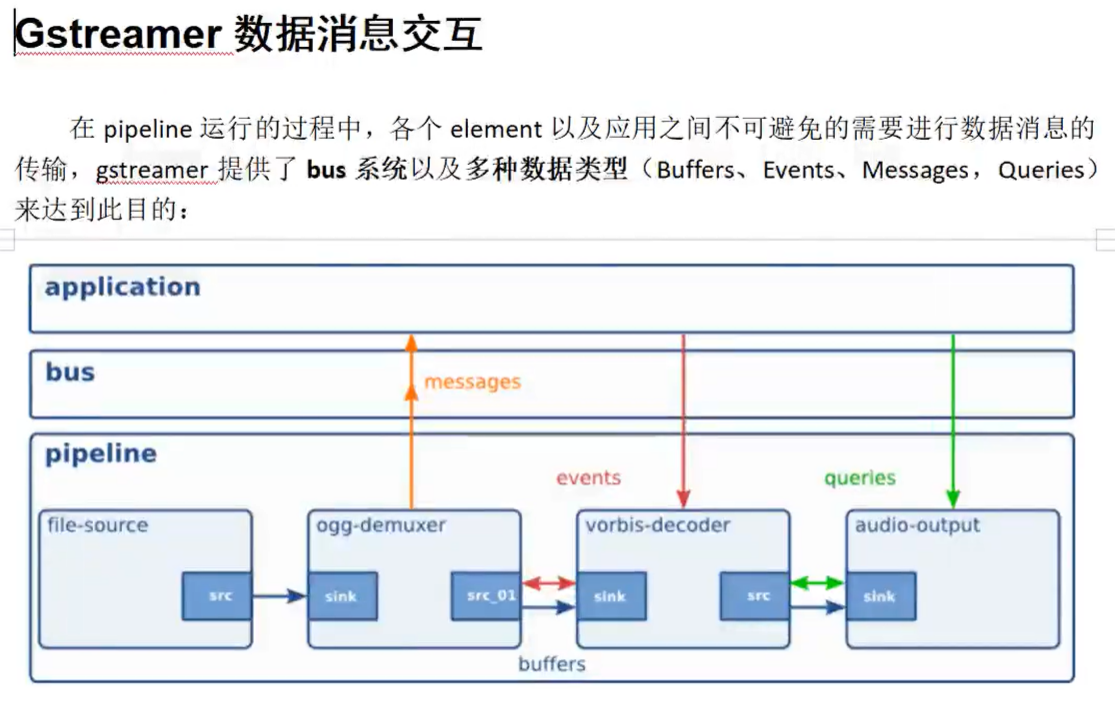
这里先看数据结构：

如图：

基本的Pipline:这些一个个的（浅蓝色）是Element，中间连接的地方（深蓝色）就是Pad



这张图就显示了对应的消息机制：我们写的应用可以注册相关的消息函数，通过相关的消息函数进行发消息等。



GstElement：就是对应上图浅蓝色的部分。

GstBus：消息管理模块,是Gstreamer注册好的运行时框架里面的。

GstMessage：消息。

函数api：

gst\_init

gst\_parse\_launch

gst\_element\_set\_state

gst\_element\_get\_bus //拿到bus管理器，进行管理消息

gst\_bus\_timed\_pop\_filtered //这个就会返回相关的消息

gst\_message\_unref //释放

gst\_object\_unref

/\*



\*/

\*/