Android's Handler

David Lau • China

Email: zhiyong.liu@aliyun.com Latest Update：2013-09-08

Email: zhiyong.liu@aliyun.com Latest Update：2013-09-08

http://www.workercn.cn2012/11/9 15:30 来源： 深圳晚报

What's the Handler

- A Handler allows you to send and process message and Runnable objects associated with a thread's MessageQueue.
- Each Handler instance is associated with a single thread and that thread's message queue. When you create a new Handler, it is bound to the thread/message queue of the thread that is creating it—from that point on, it will deliver messages and runnables to that message queue and execute them as they come out of the message queue.

http://developer.android.com/reference/android/os/Handler.html

Handle

Native

Handle

JAVA

Java's Thread

Two main ways of having a Thread execute application code :

1. Providing a new class that extends Thread and overriding its run() method.

2. Providing a new Thread instance with a Runnable object during its creation. In both cases, the start() method must be called to actually execute the new Thread.

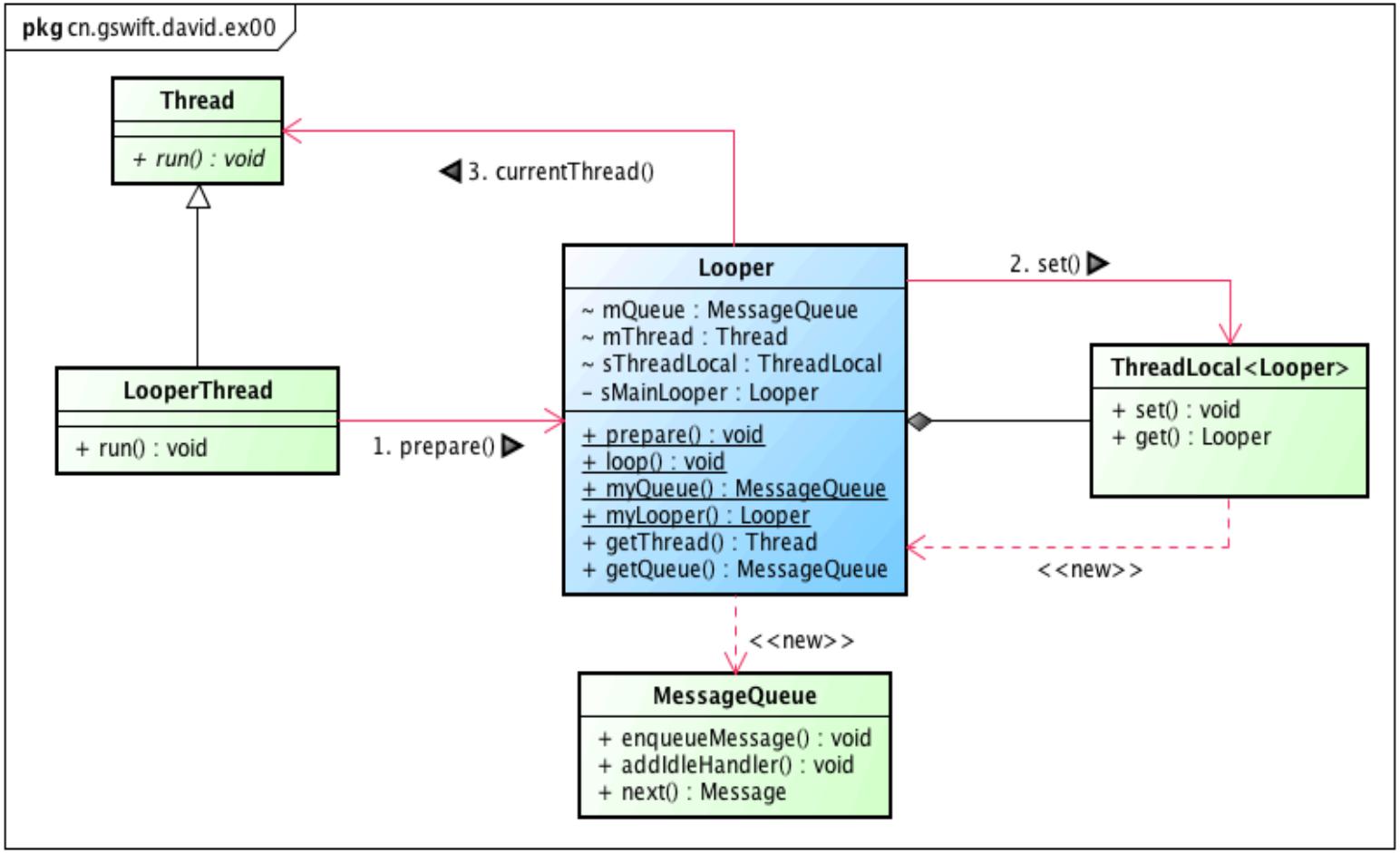
http://www.slideshare.net/HoangNgoBuu/android-thread-handler-and-asynctask

Two main uses for a Handler

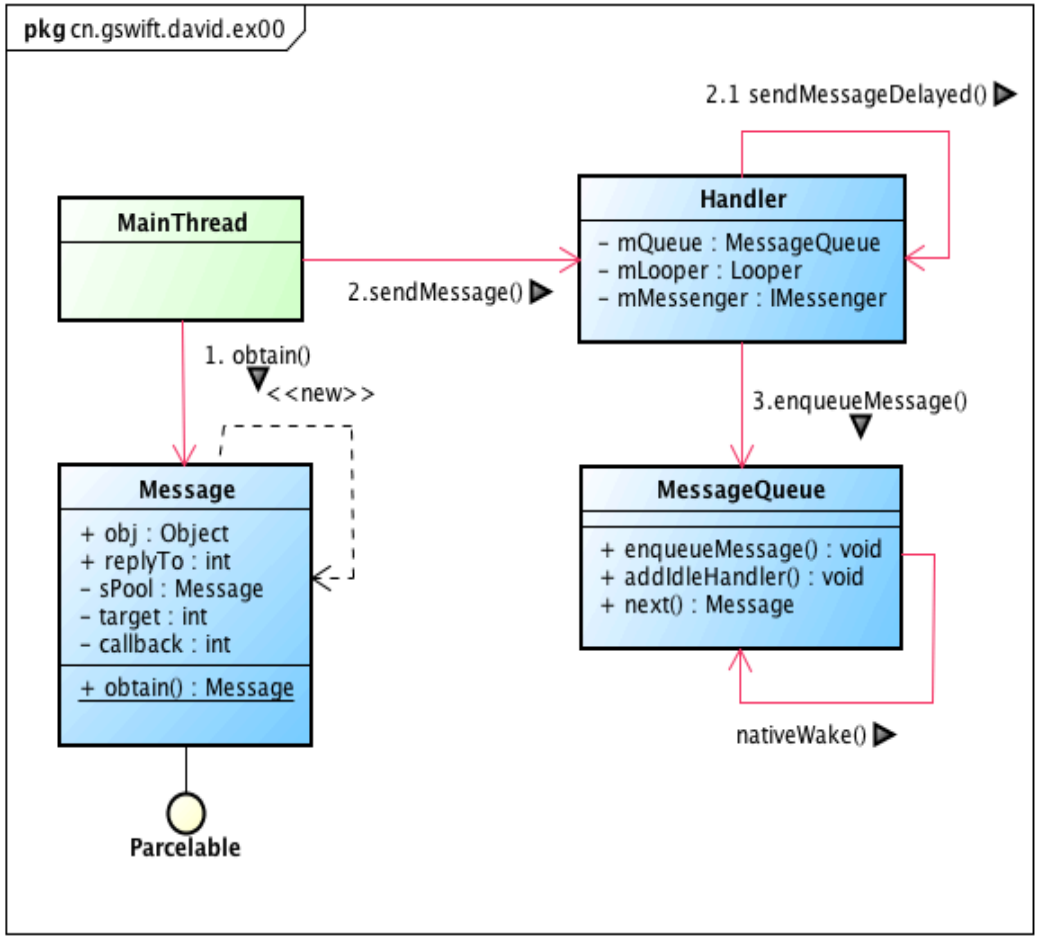
1. Schedule messages and runnables to be executed as some point in the future;
2. Enqueue an action to be performed on a different thread than your own.

http://developer.android.com/reference/android/os/Handler.html

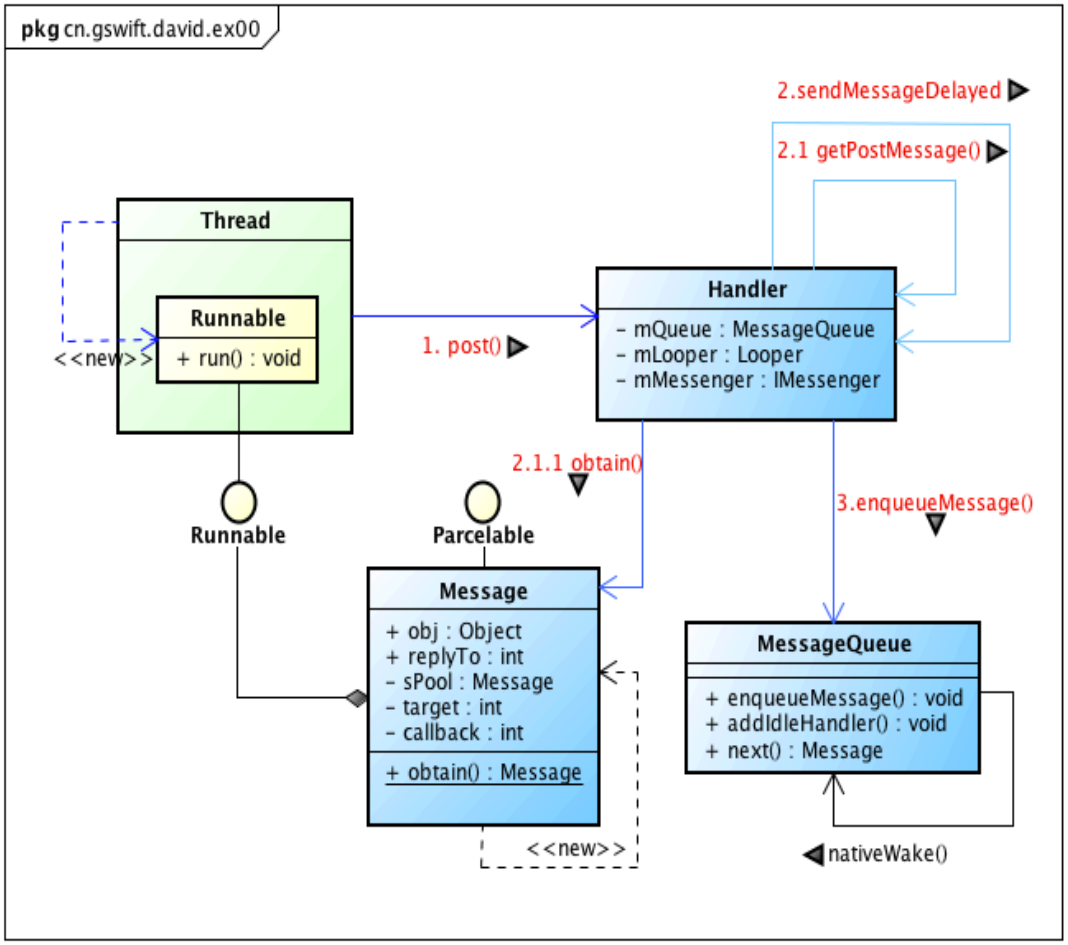
Looper.prepare()



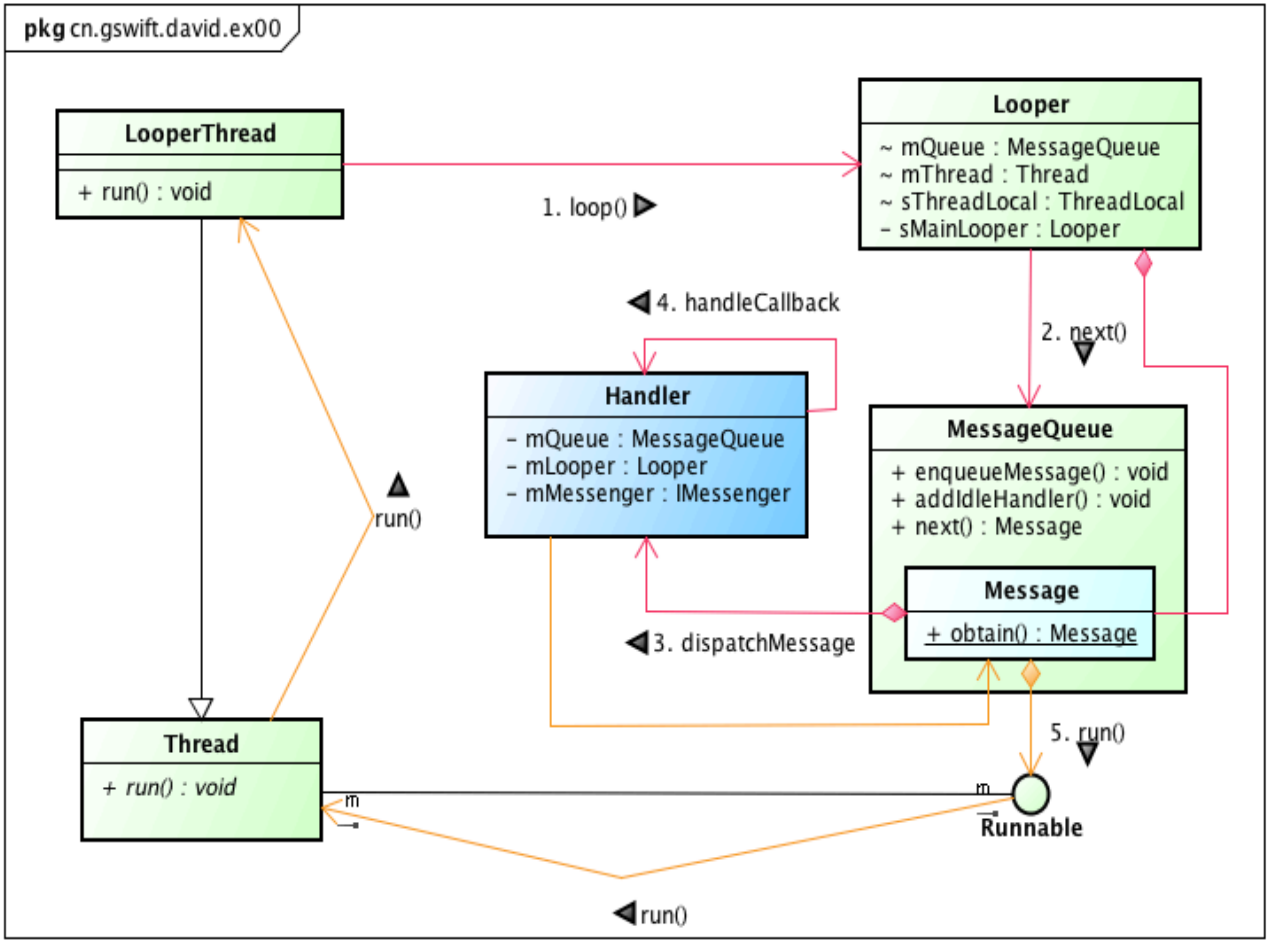
Thread&Handler



Mult-Thread&Handler



Looper.looper()



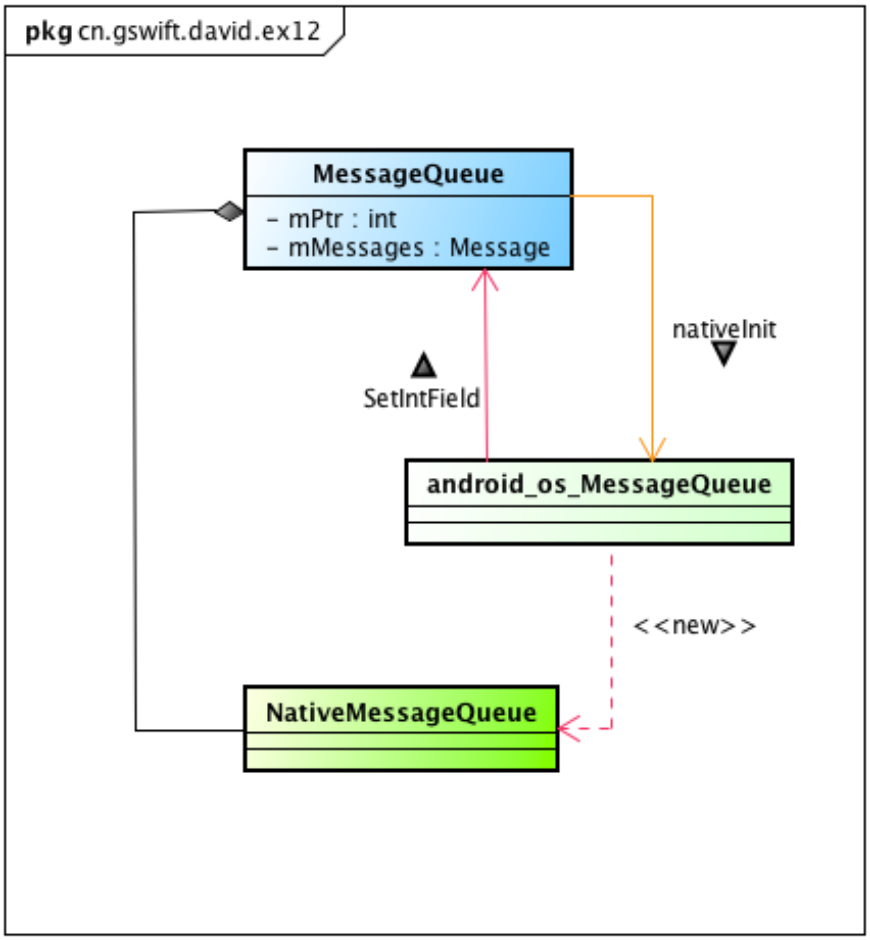
Handle

Native

Handle

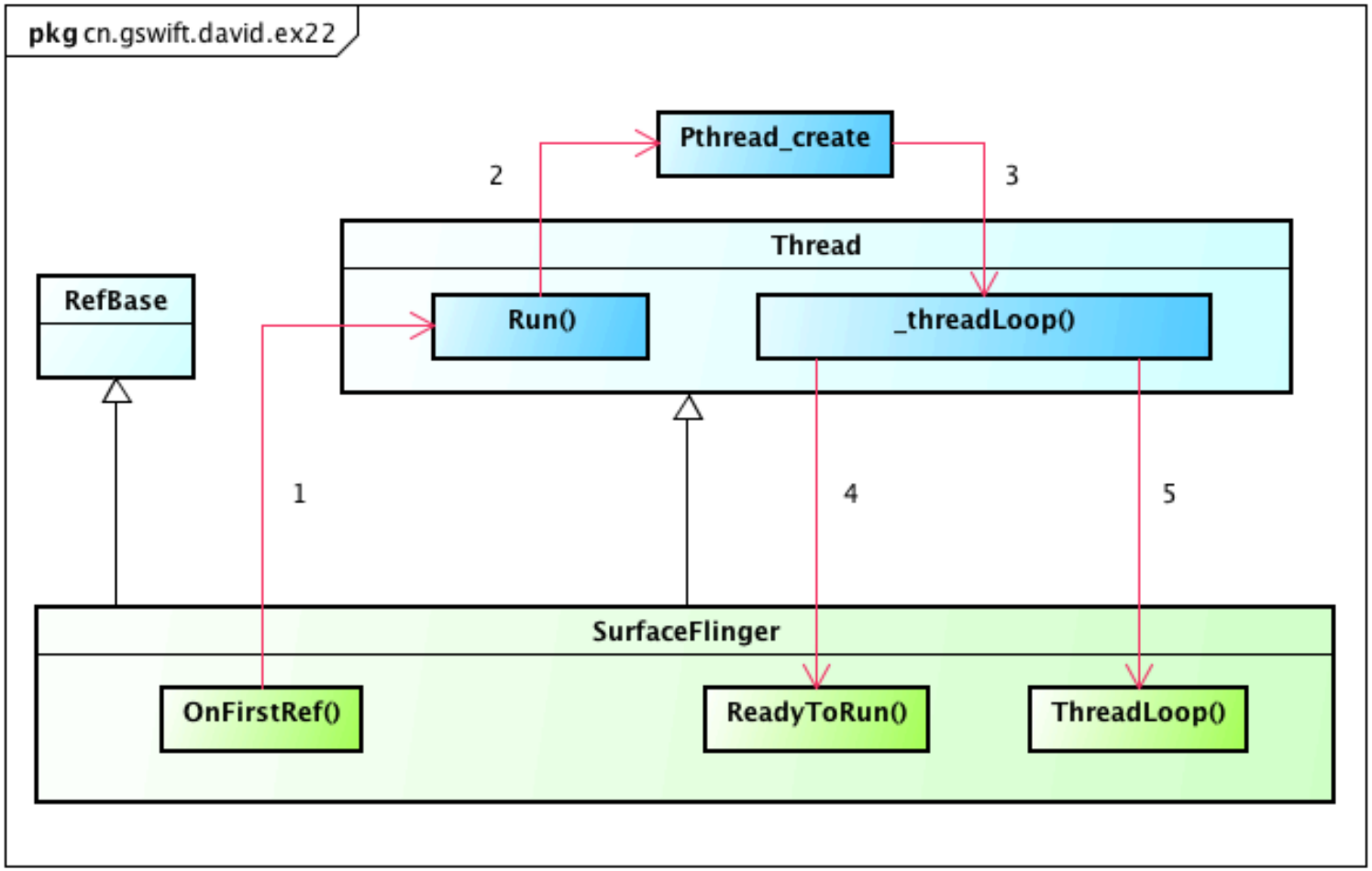
JAVA

NativeMessageQueue

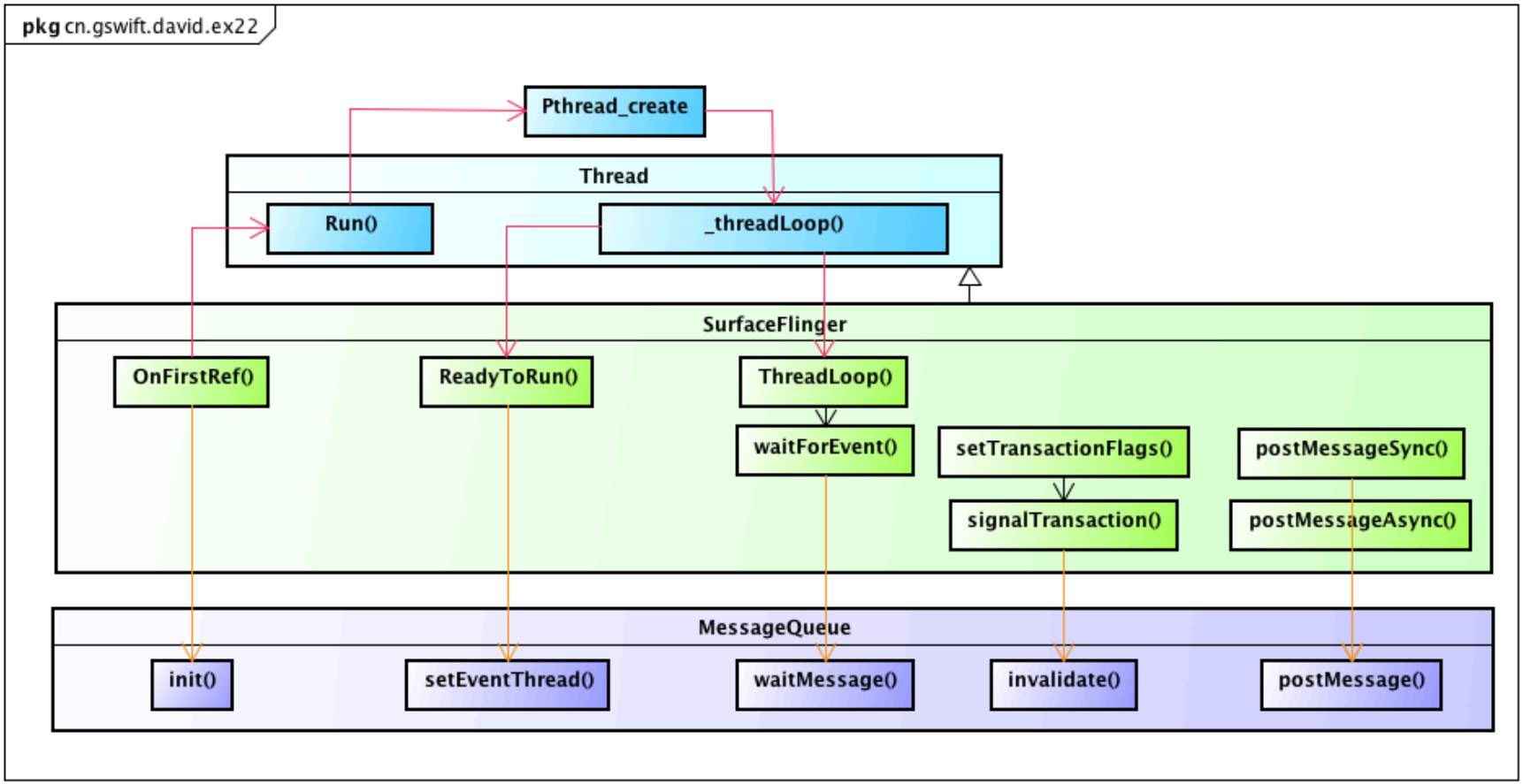


Example of SurfaceFlinger

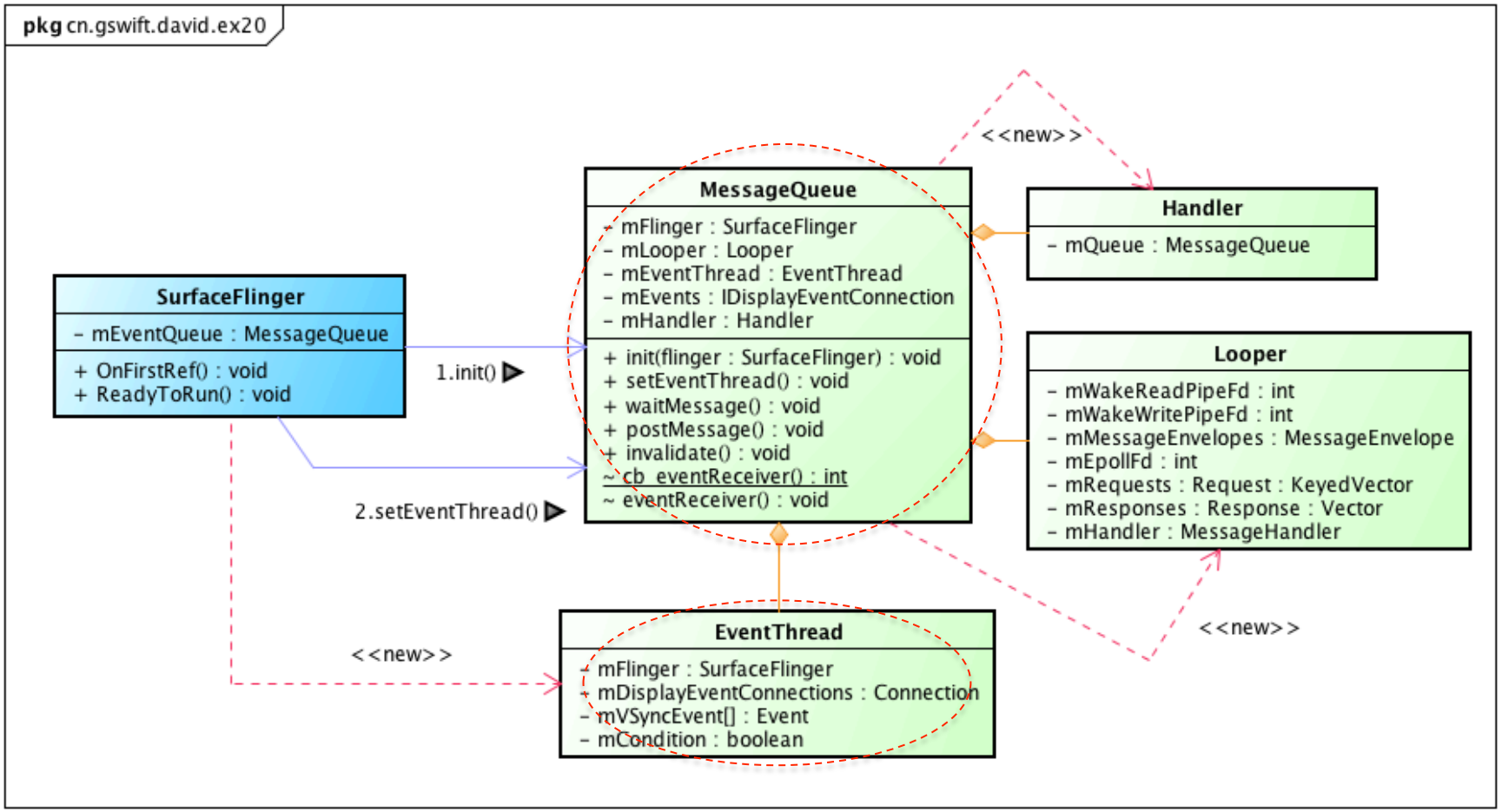
SurfaceFlinger Thread



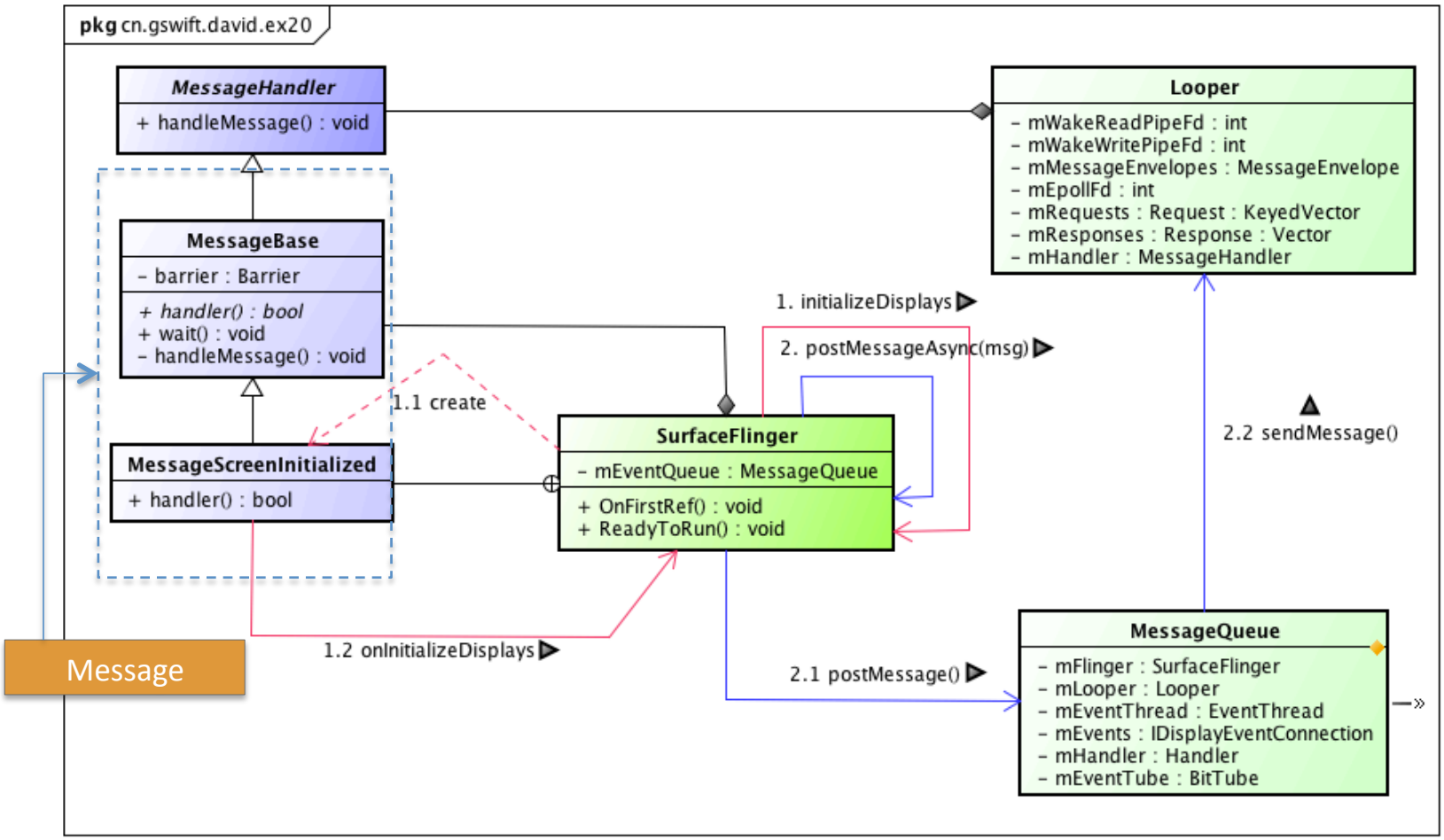
SurfaceFlinger Thread & MessageQueue



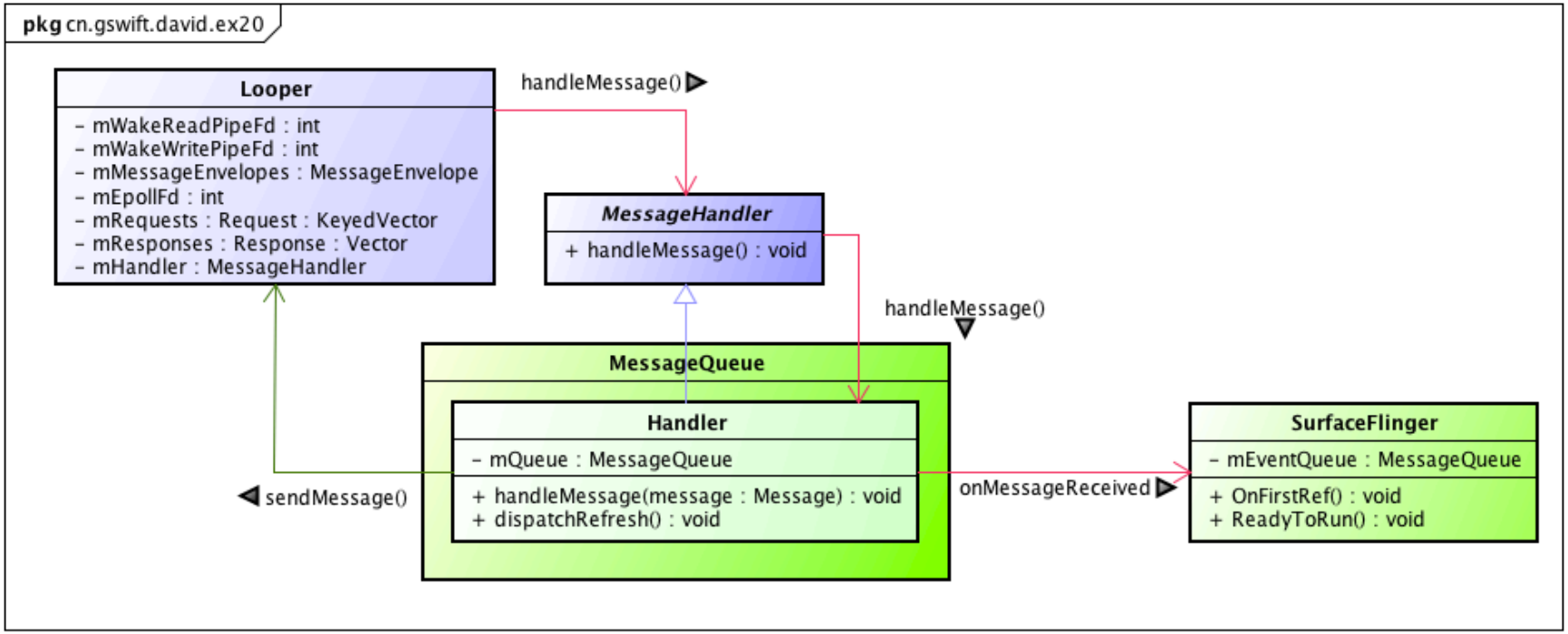
EventThread/ Messagequeue/Hanlder/Looper



SurfaceFlinger postMessageAsync()

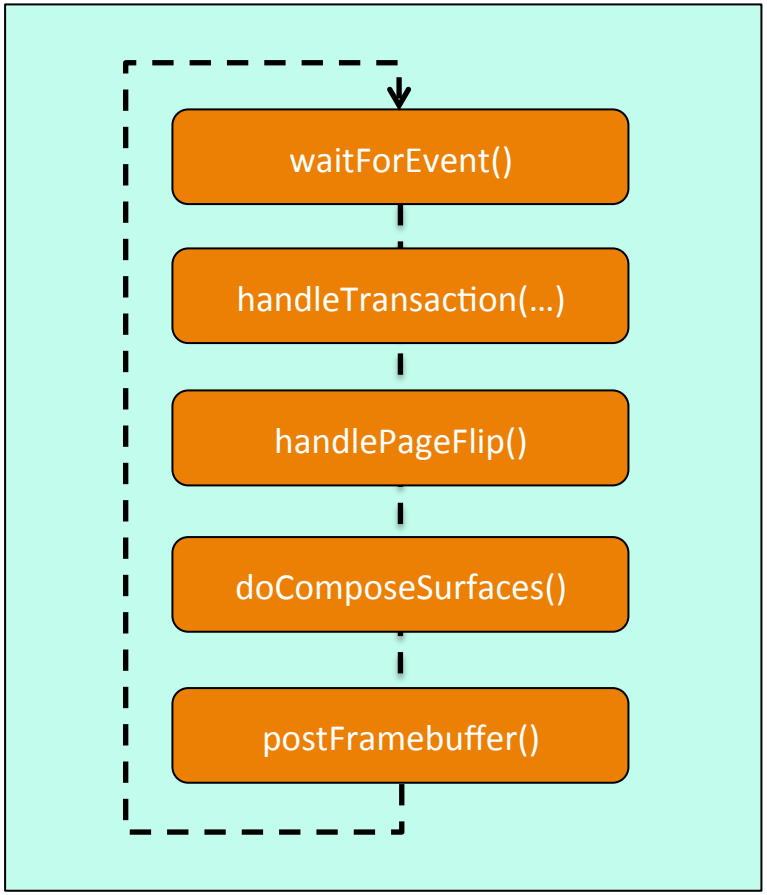


handleMessage

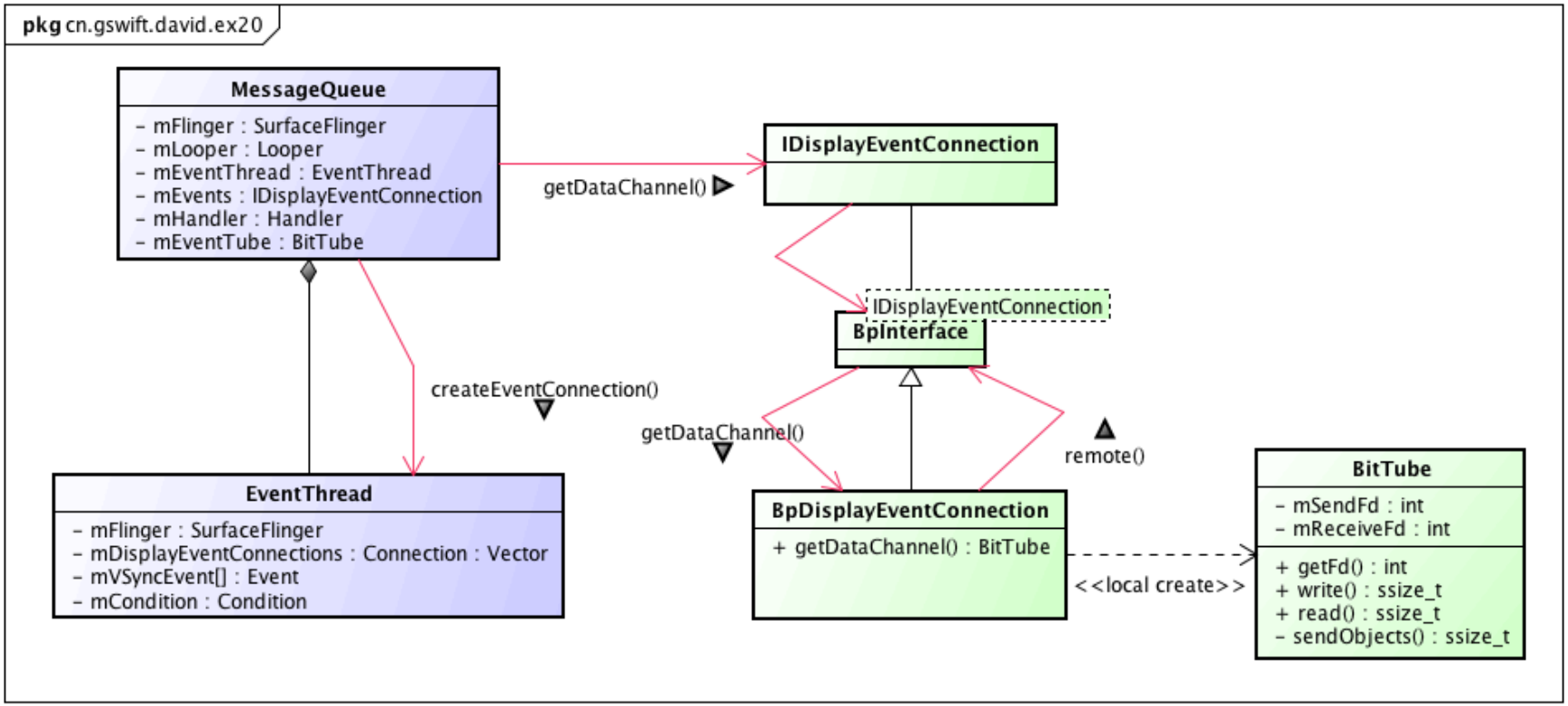


```
void SurfaceFlinger::handleMessageTransaction() {
    uint32_t transactionFlags = peekTransactionFlags(eTransactionMask);
    if (transactionFlags) {
        handleTransaction(transactionFlags);
    }
}

void SurfaceFlinger::handleMessageInvalidate() {
    ATRACE_CALL();
    handlePageFlip();
}

void SurfaceFlinger::handleMessageRefresh() {
    ATRACE_CALL();
    preComposition();
    rebuildLayerStacks();
    setUpHWComposer();
    doDebugFlashRegions();
    doComposition();
    postComposition();
}
```

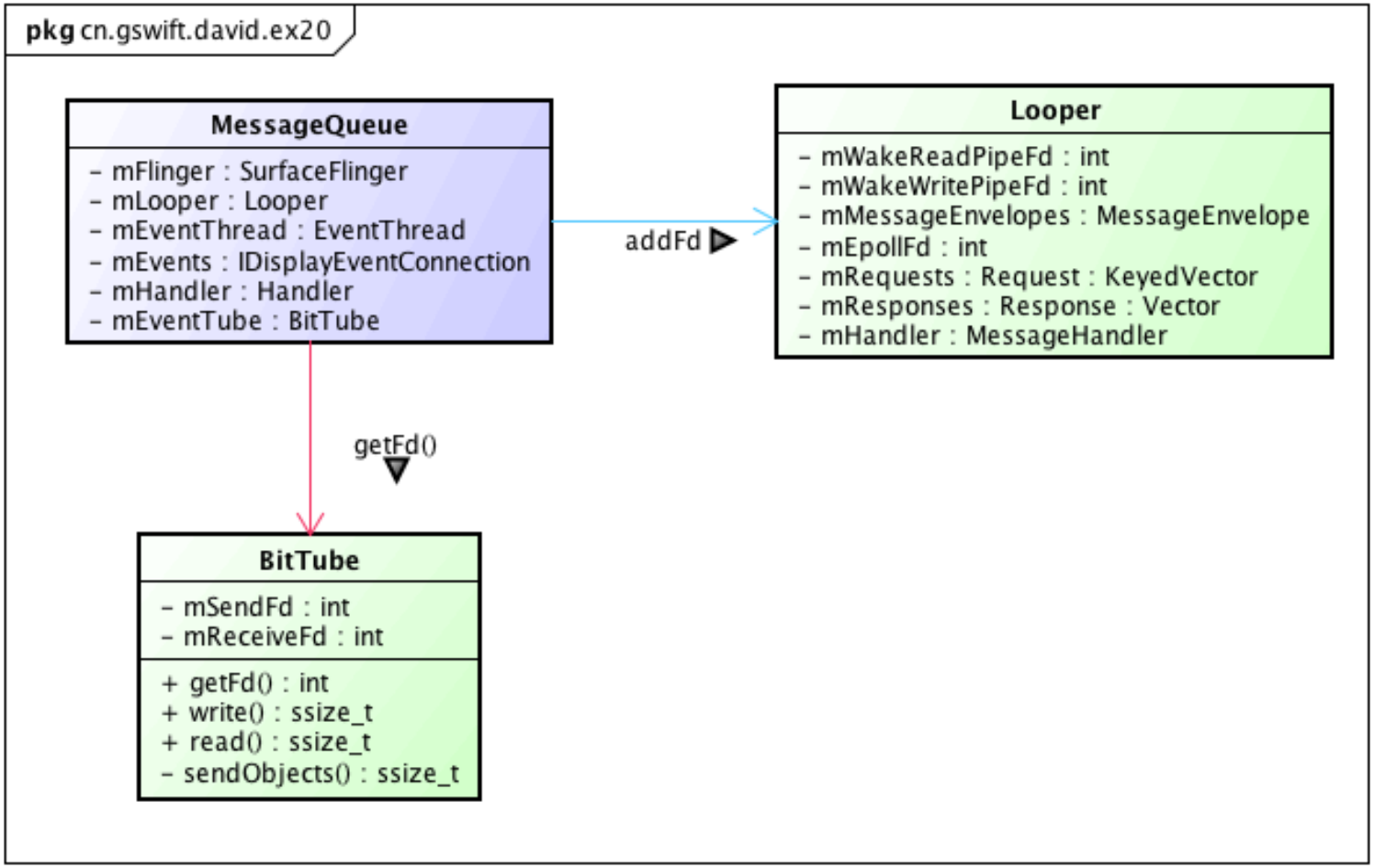
SurfaceFlinger::threadLoop()



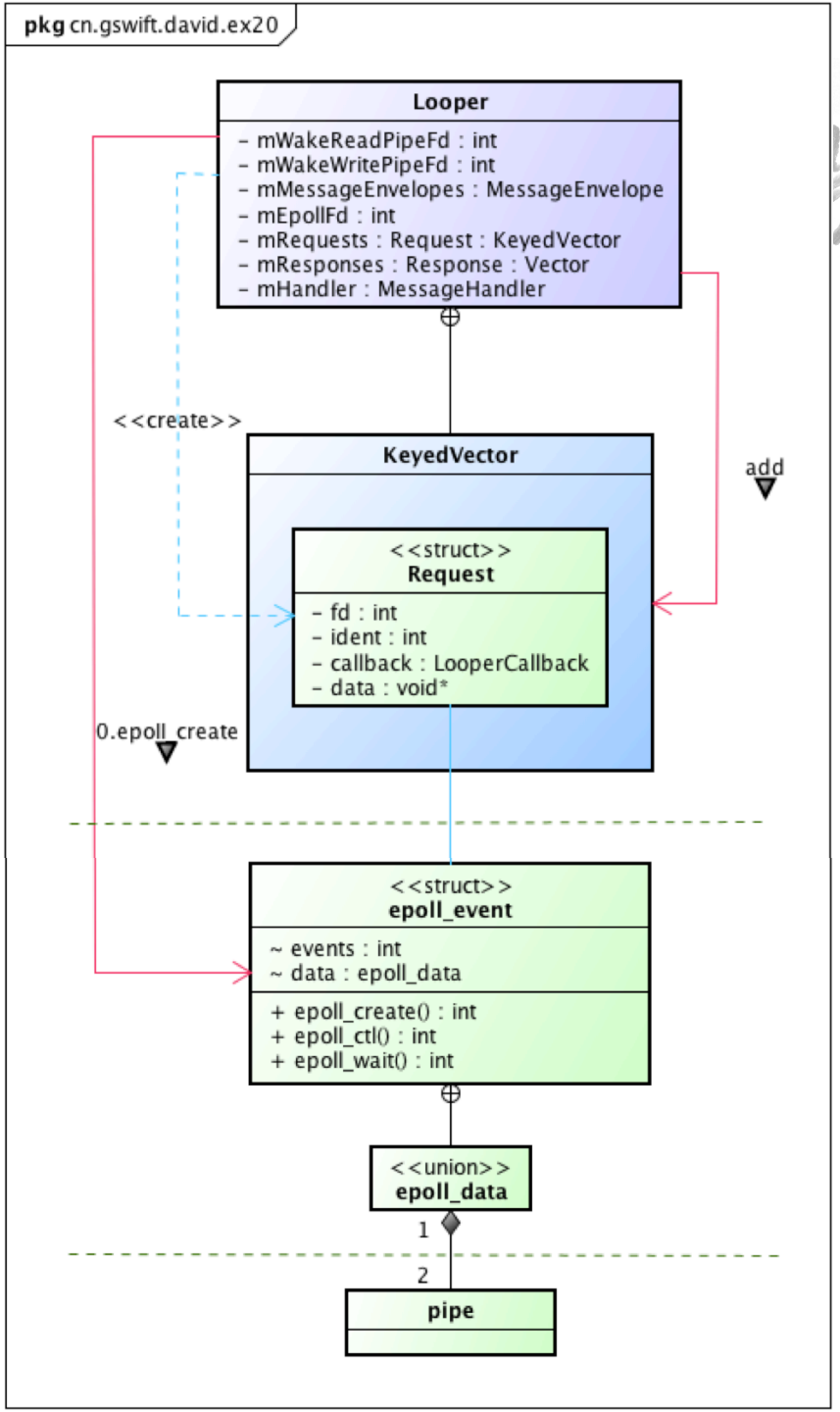
MessageQueue::setEventThread<1>



MessageQueue::setEventThread
Looper->addFd



Looper::addFd()



参考资料：

Android4.2 http://code.metager.de/source/xref/android/4.2/

http://www.slideshare.net/HoangNgoBuu/android-thread-handler-and-asynctask

About Me

- I have been working as a product-designer specializing in software/Web application design and development. I am passionate about mobile application development and became interested in Android programming when the platform was launched by Google. Thus I was not programming on Android projects, I spent spare time reading technical blogs, researching, analyzing, and testing mobile applications, as a software consultancy specialized in android technologies.
- In my product-design time, in the developing, I've encountered too many program manage troubles that suffer due to poor communication and code design, I know that help them to understand the system framework is very important. I amd experienced in system and application layers, my goal is simple: help someone who wishes to better understand the **Android framework** in java、JNI and C/C++ libraries.

- Please also check my article and slides on this http://blog.sina.com.cn/gswift

Contact: Zhiyong.liu@aliyun.com

http://weibo.com/gswift