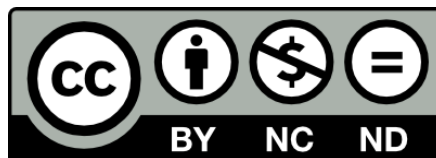




Application Framework WindowManagerService

David Lau • China



本作品采用知识共享 署名-非商业性使用-禁止演绎 3.0 中国大陆 许可协议进行许可。
要查看该许可协议，可访问<http://creativecommons.org/licenses/by-nc-nd/3.0/cn/>

您可以自由：

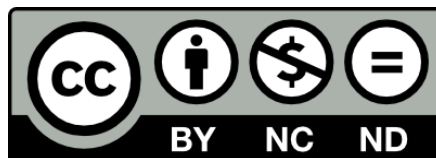
复制、发行、展览、表演、放映、广播或通过信息网络传播本作品

惟须遵守下列条件：

- 署名 — 您必须按照作者或者许可人指定的方式对作品进行署名。
- 非商业性使用 — 您不得将本作品用于商业目的。
- 禁止演绎 — 您不得修改、转换或者以本作品为基础进行创作。

© Copyright 2013 These slides created by :刘智勇(David Lau)

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



This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/3.0/> or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

You are free:

to Share — to copy, distribute and transmit the work

Under the following conditions:

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Noncommercial — You may not use this work for commercial purposes.

No Derivative Works — You may not alter, transform, or build upon this work.

© Copyright 2013 These slides created by :刘智勇(David Lau)

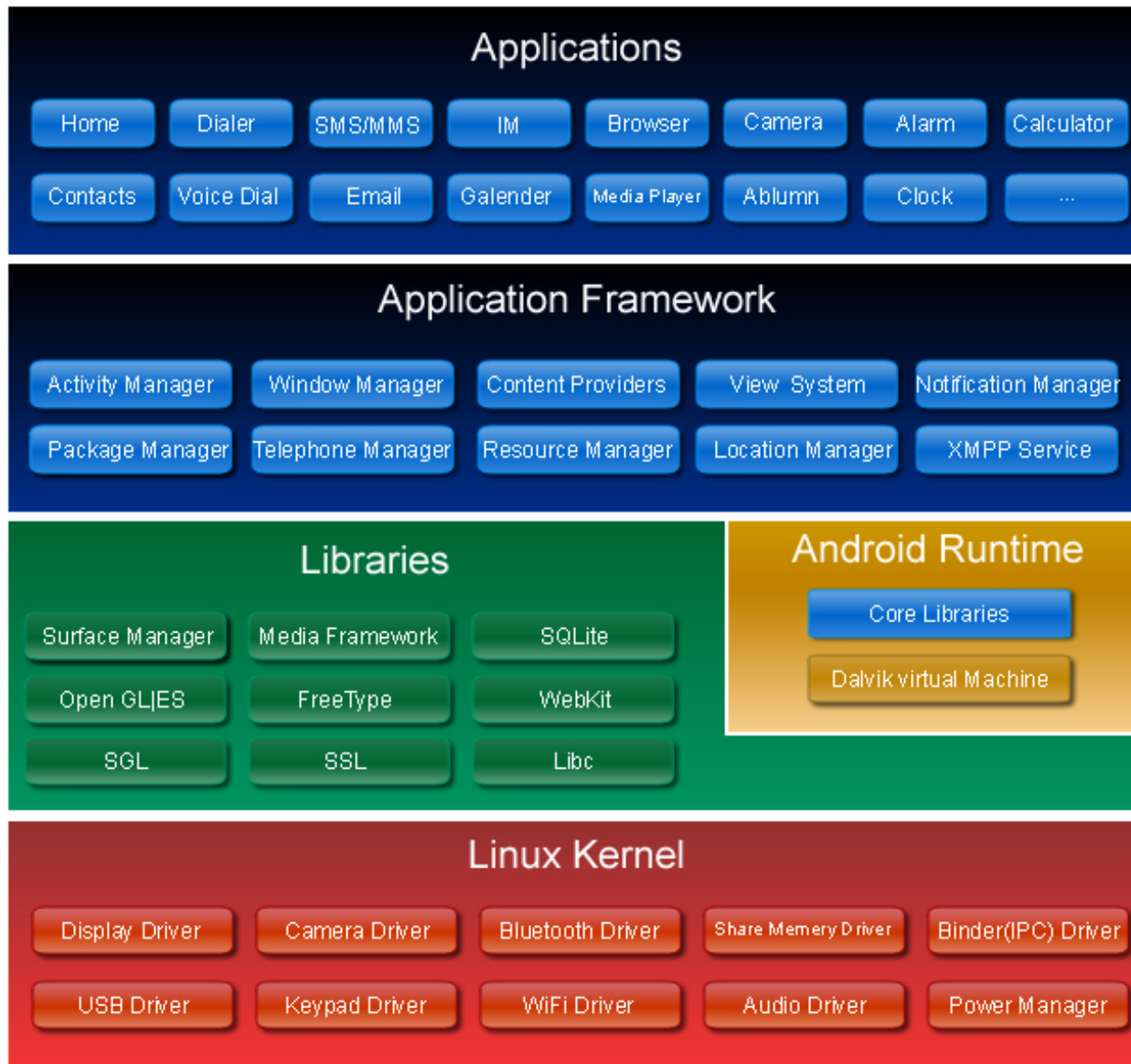
Email: zhiyong.liu@aliyun.com Latest Update: 2013-09-08

فج



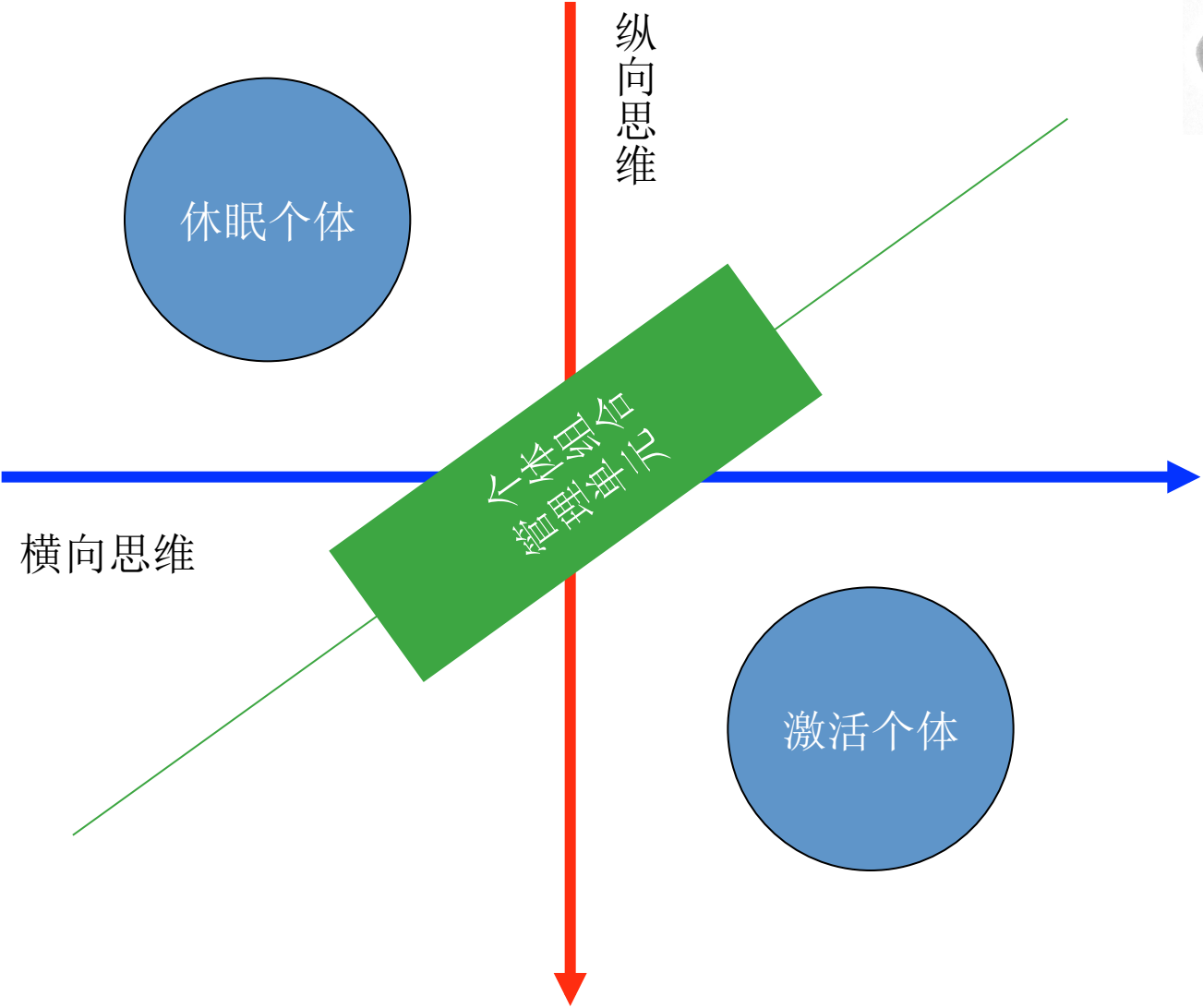


Android System Framework





影随形生，影随形动，如影随形



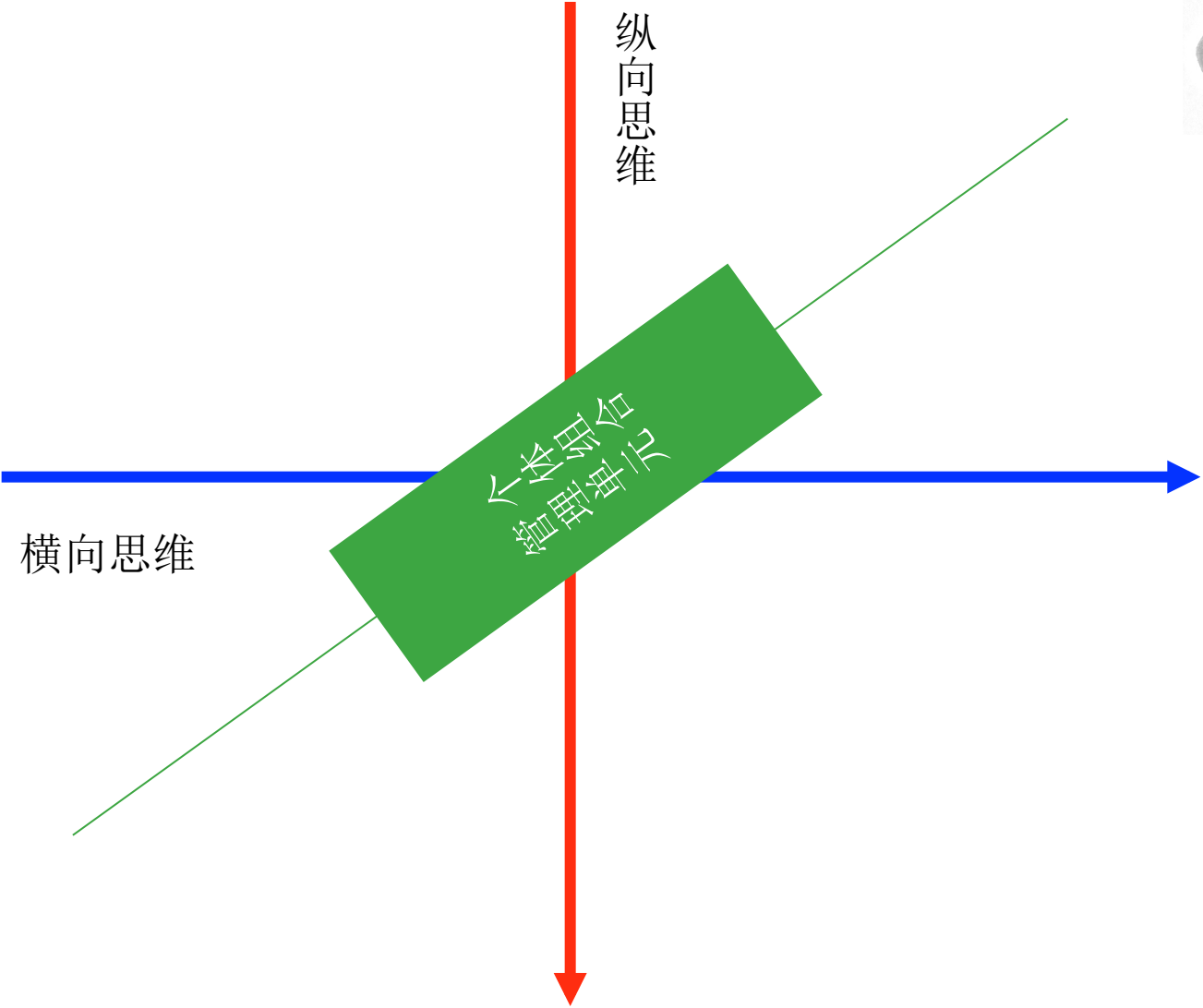
休眠个体

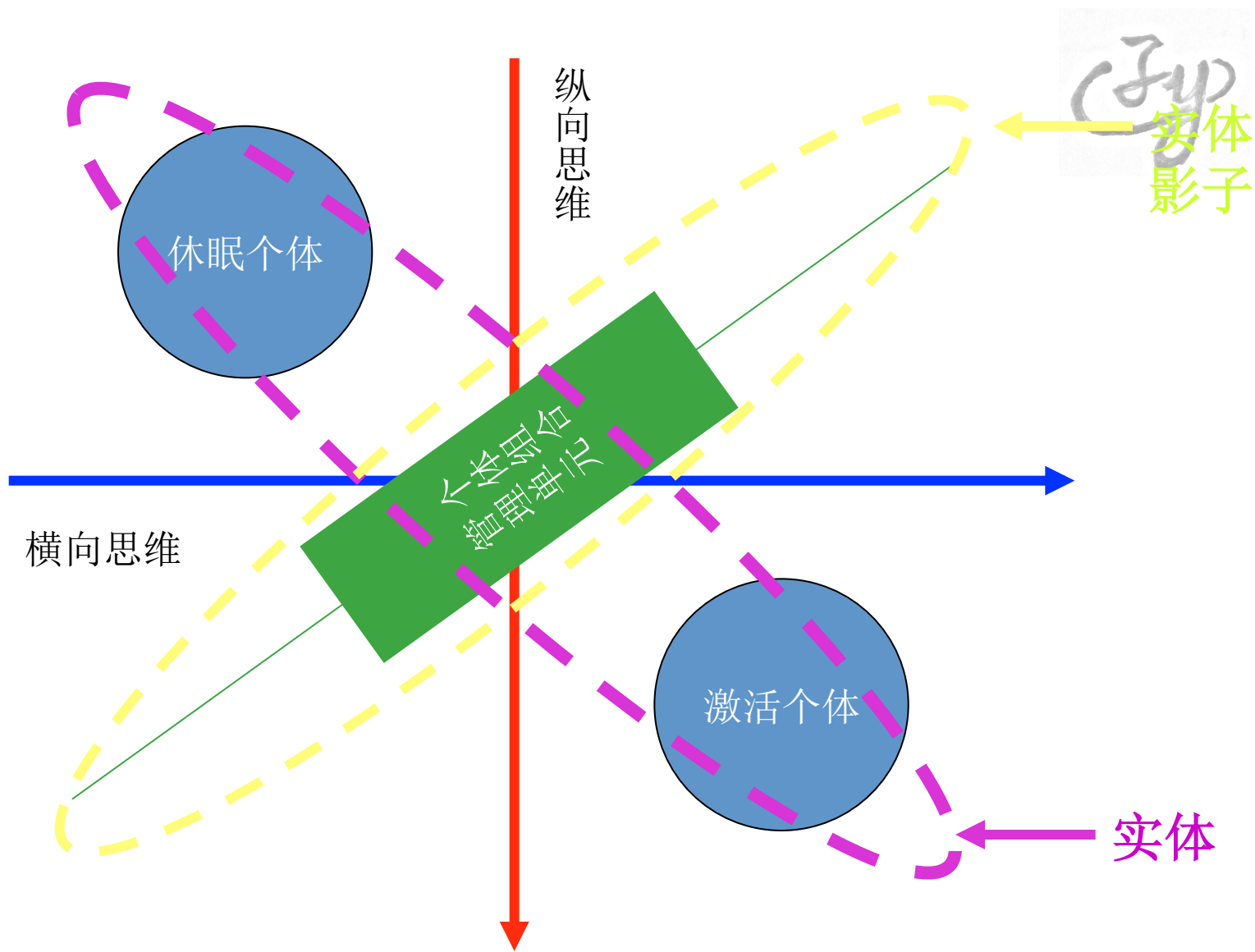
纵向思维

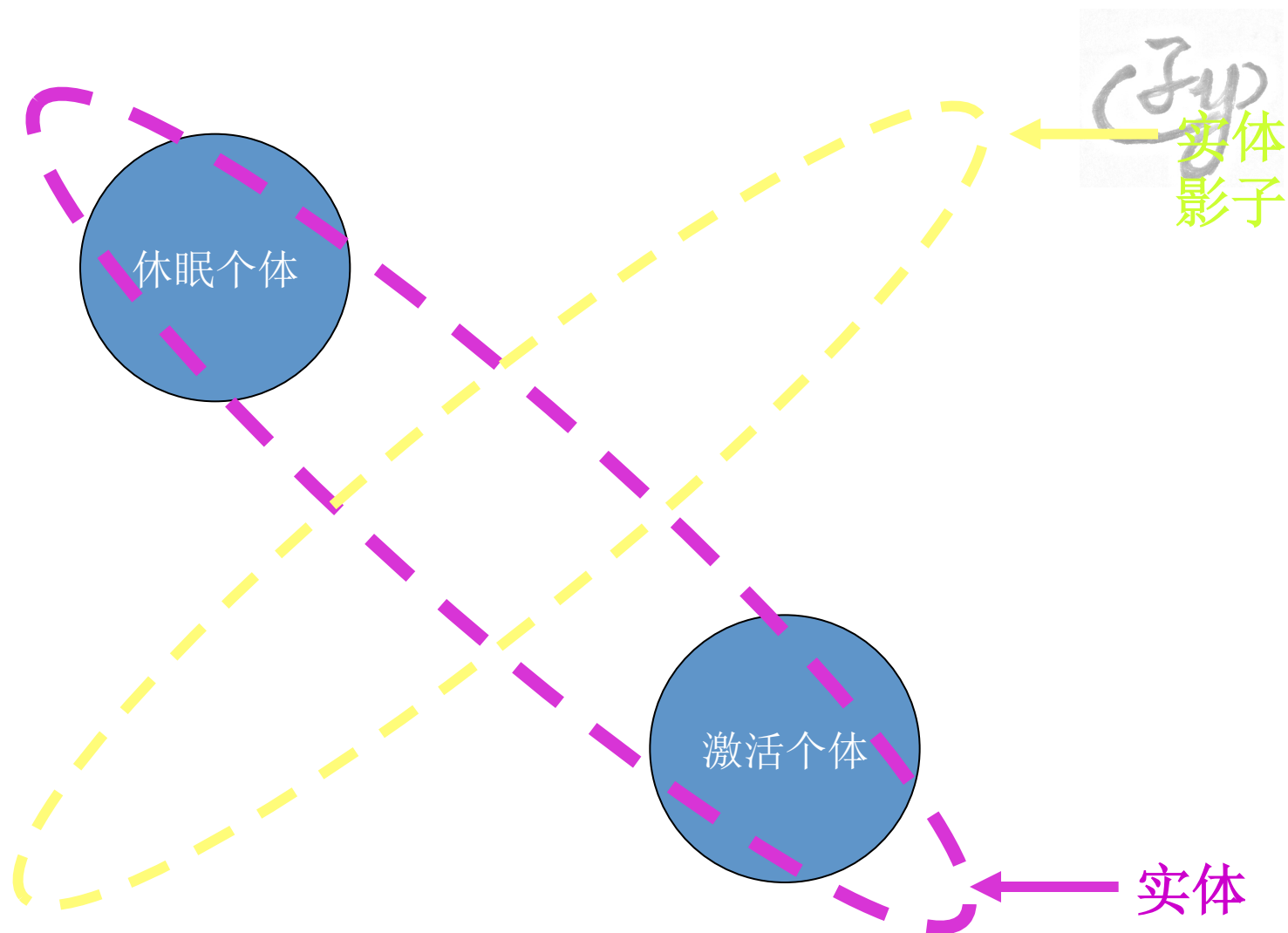
横向思维

← 思维如
逆时针

激活个体







窗口服务中的元素



- Activity
- Window
- InputEvent
- Display
- PhoneWindow's View

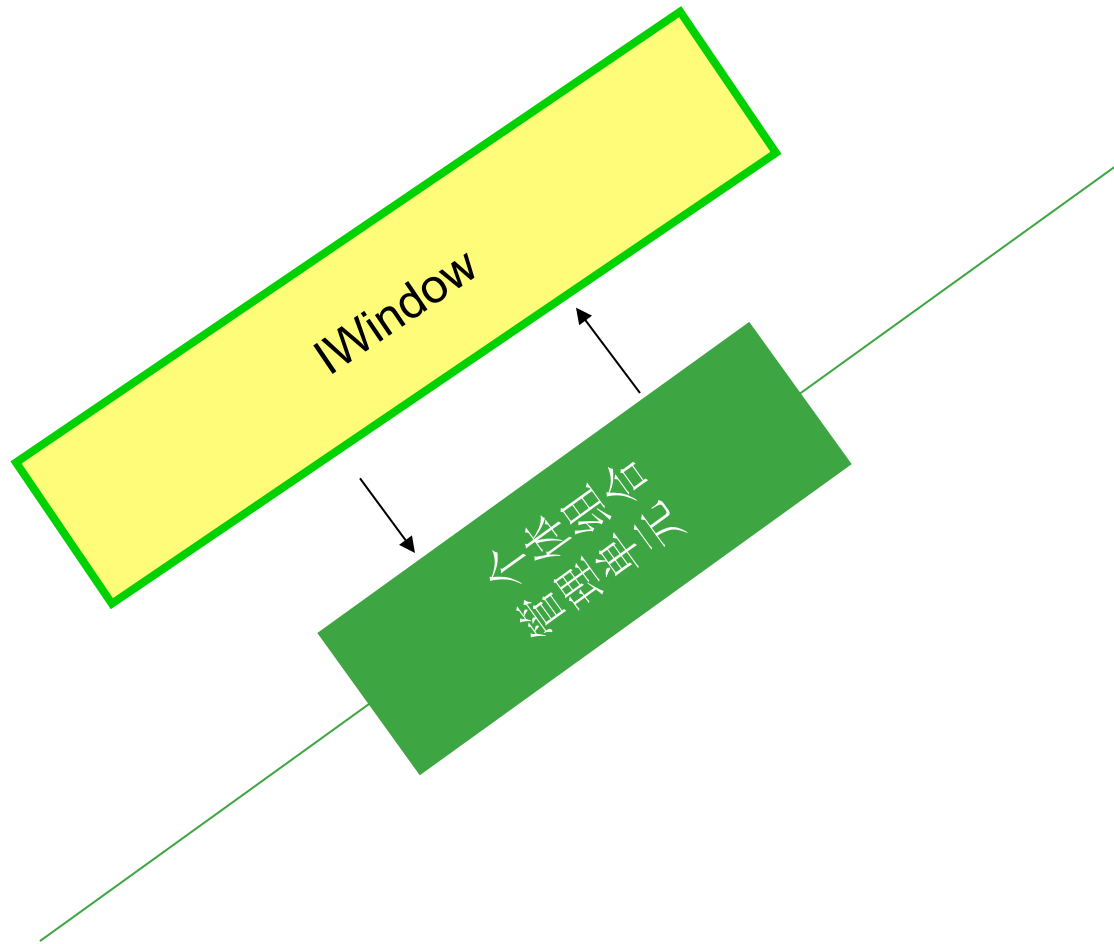
WindowManagerService

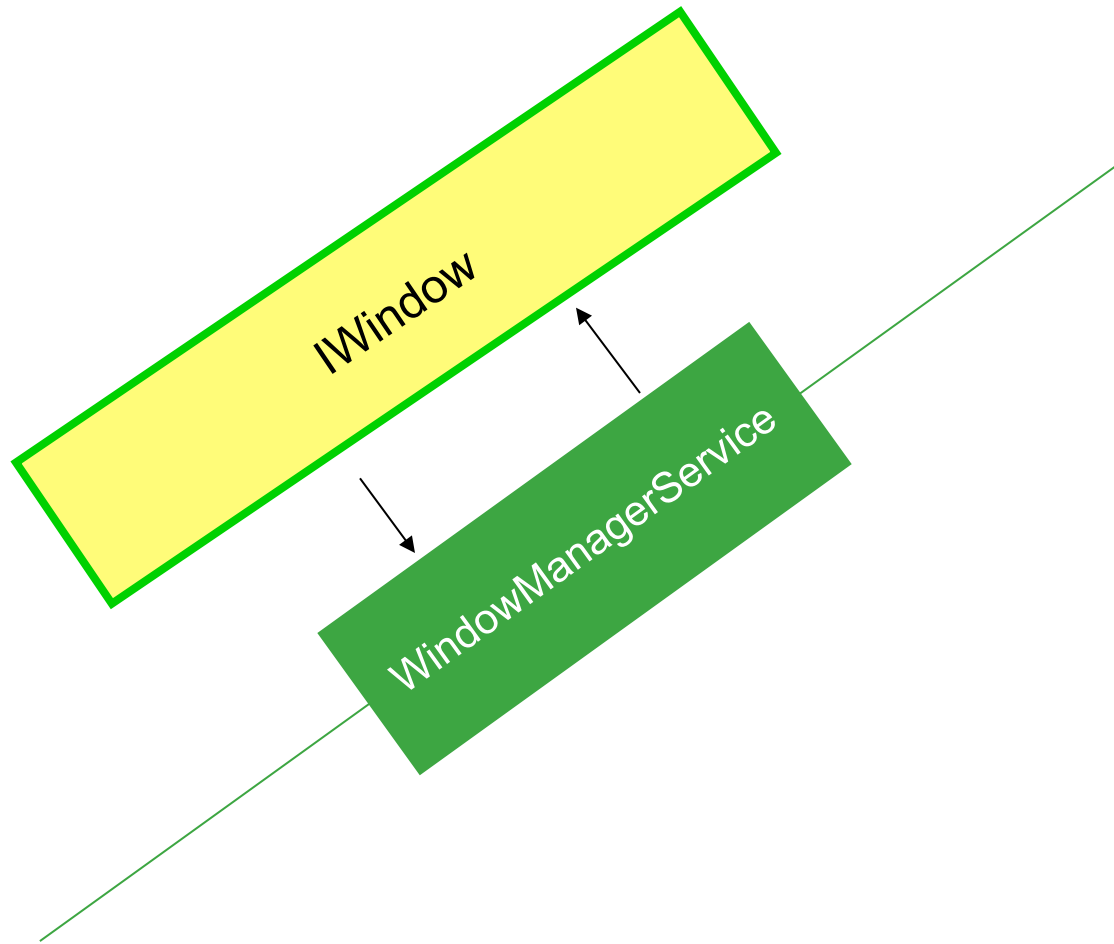


- Create a Session for the ViewRootImpl and open session by the IWindowSession interface
- Add the IWindow to the WMS
- Create a WindowState for the Activity's Window
- Attach WindowState to the Session
- Add the IWindow to WindowManagerService



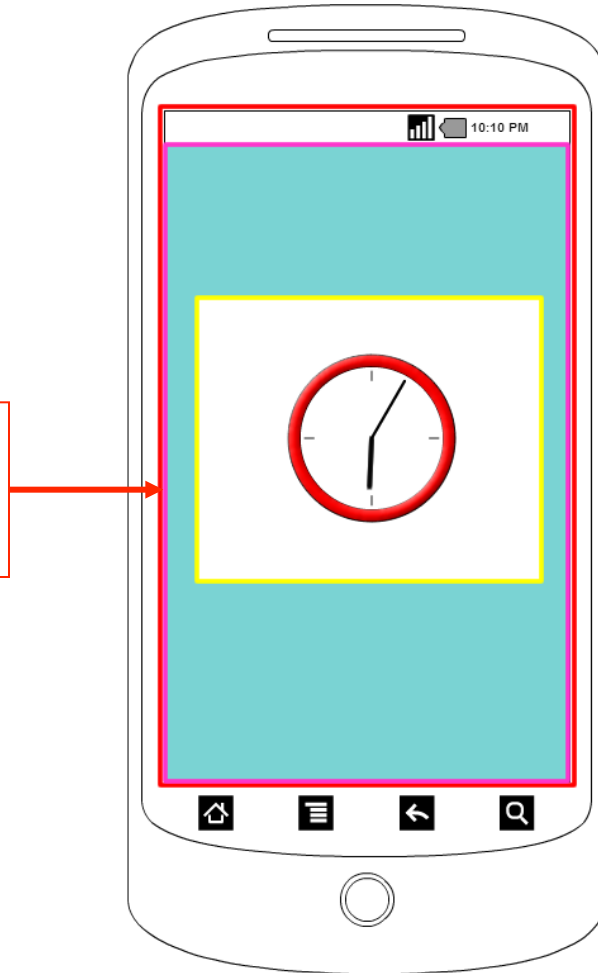
影生于形



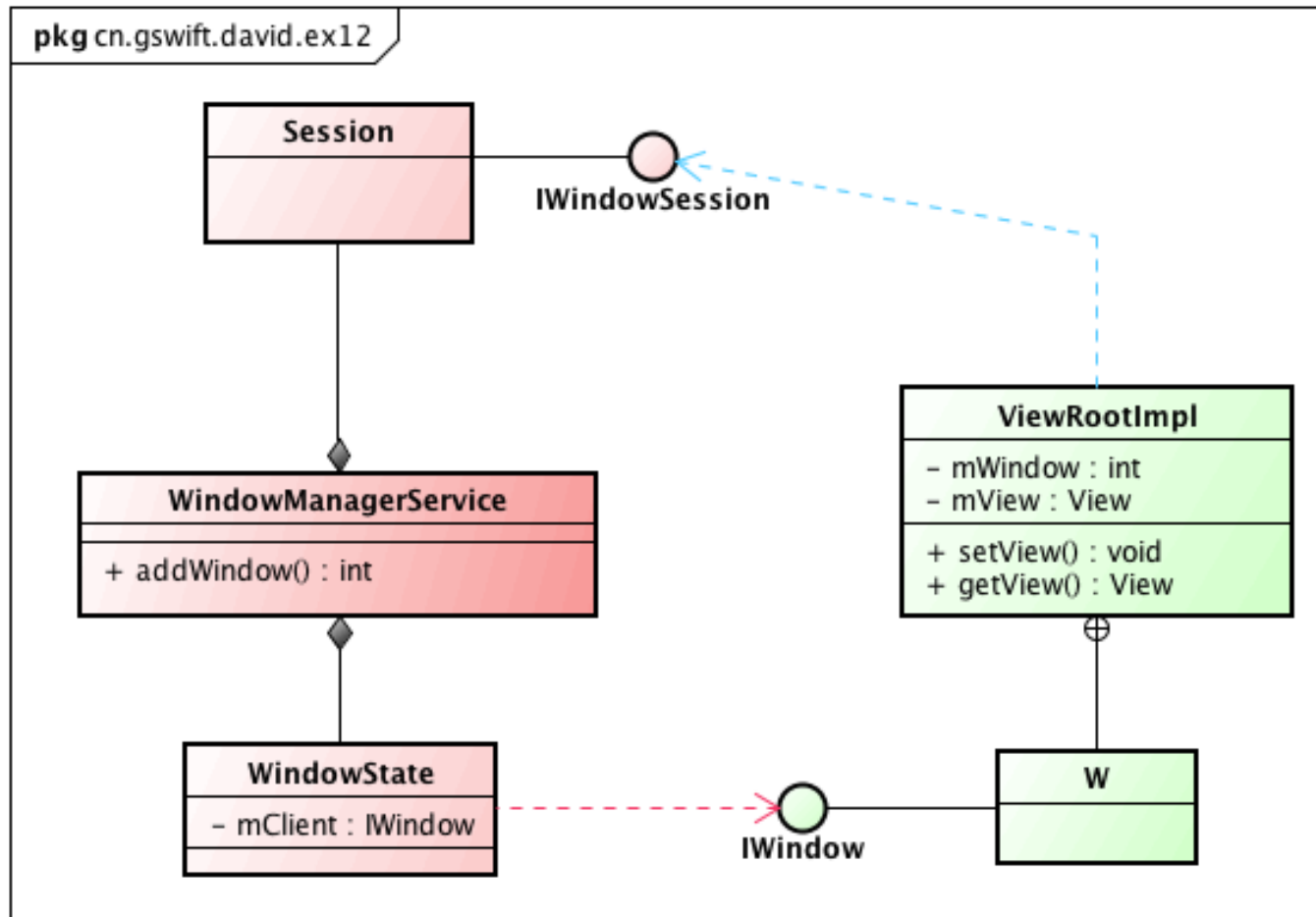




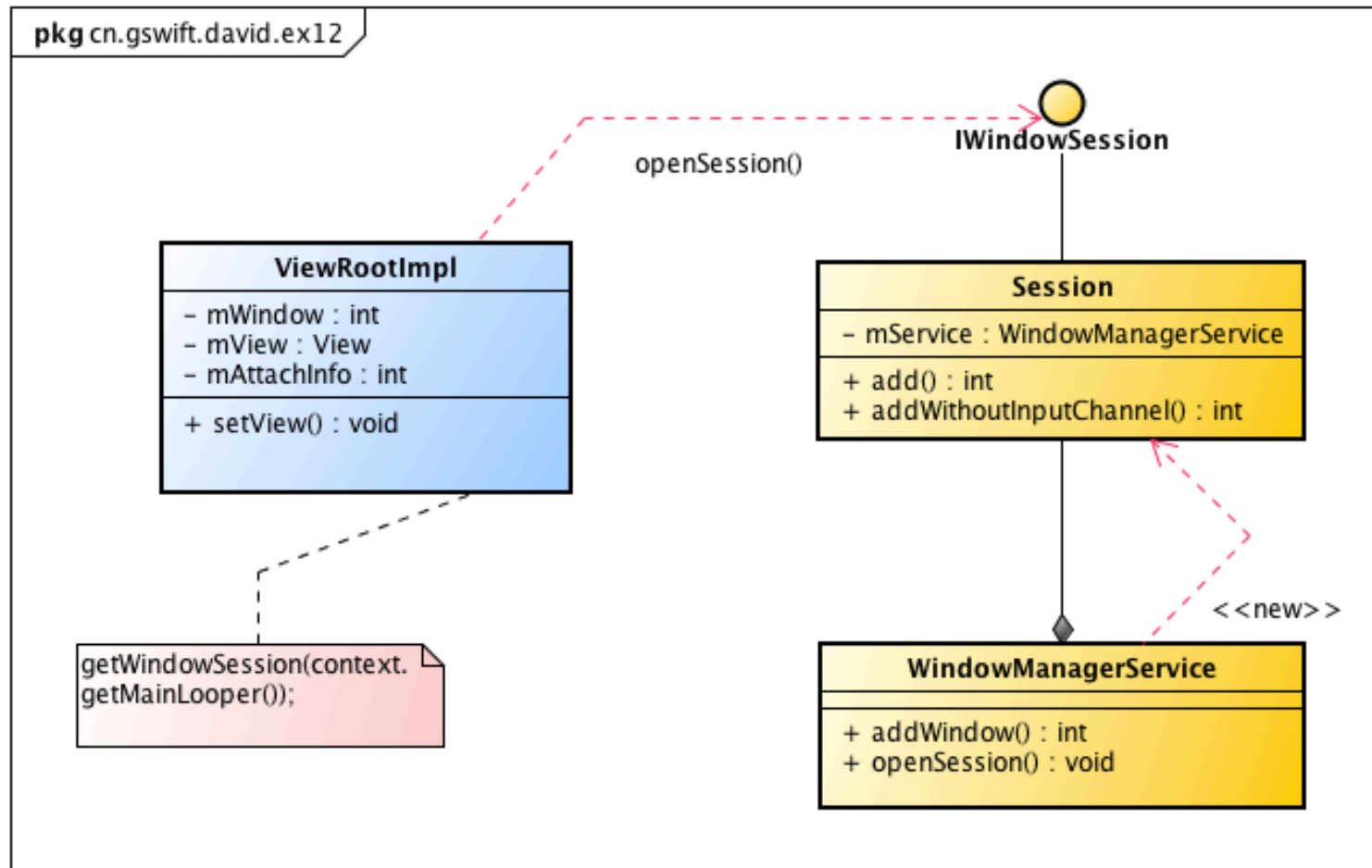
**PhoneWindow's
DecorView**



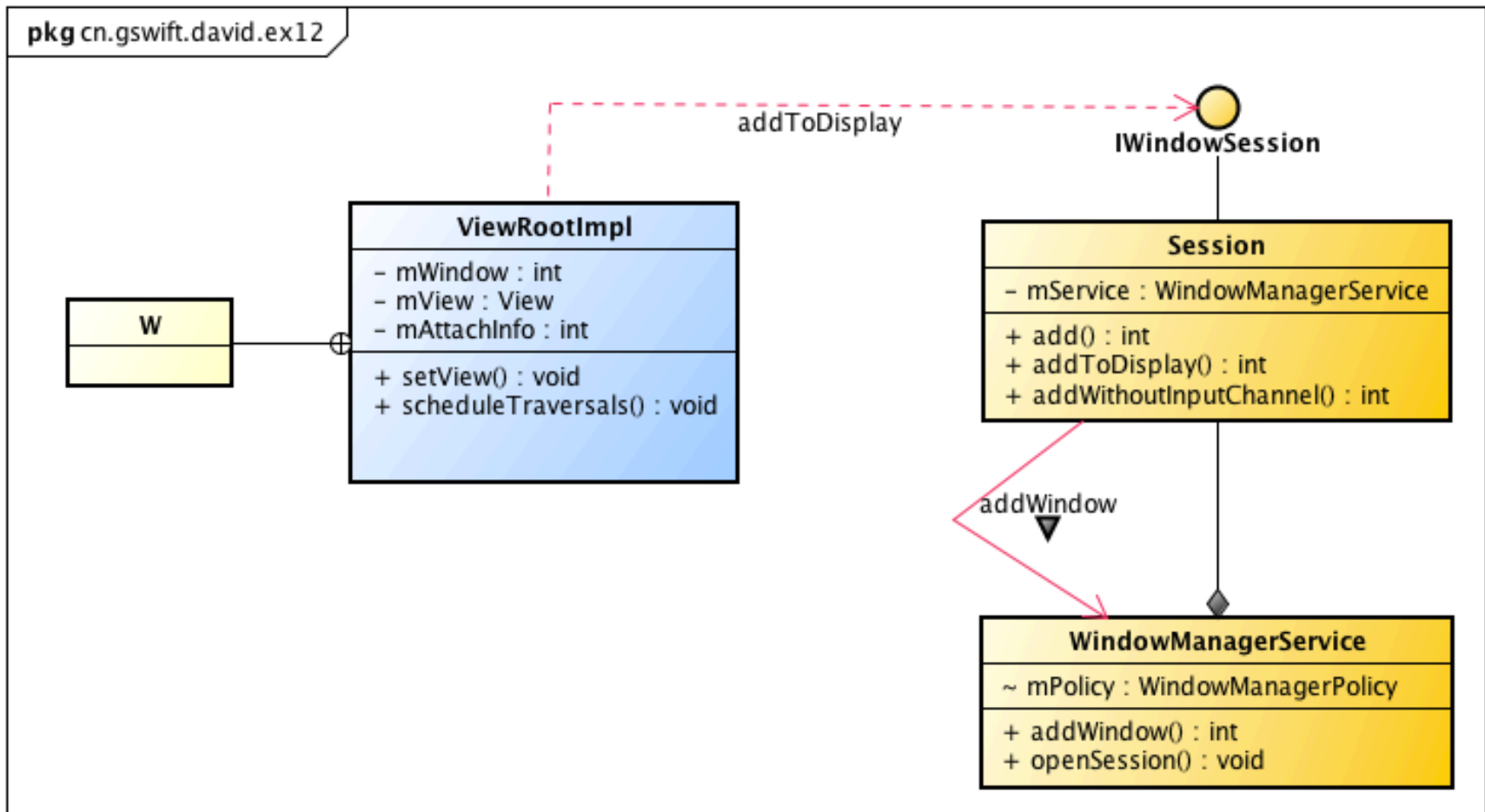
WMS | ViewRootImpl



Create a Session



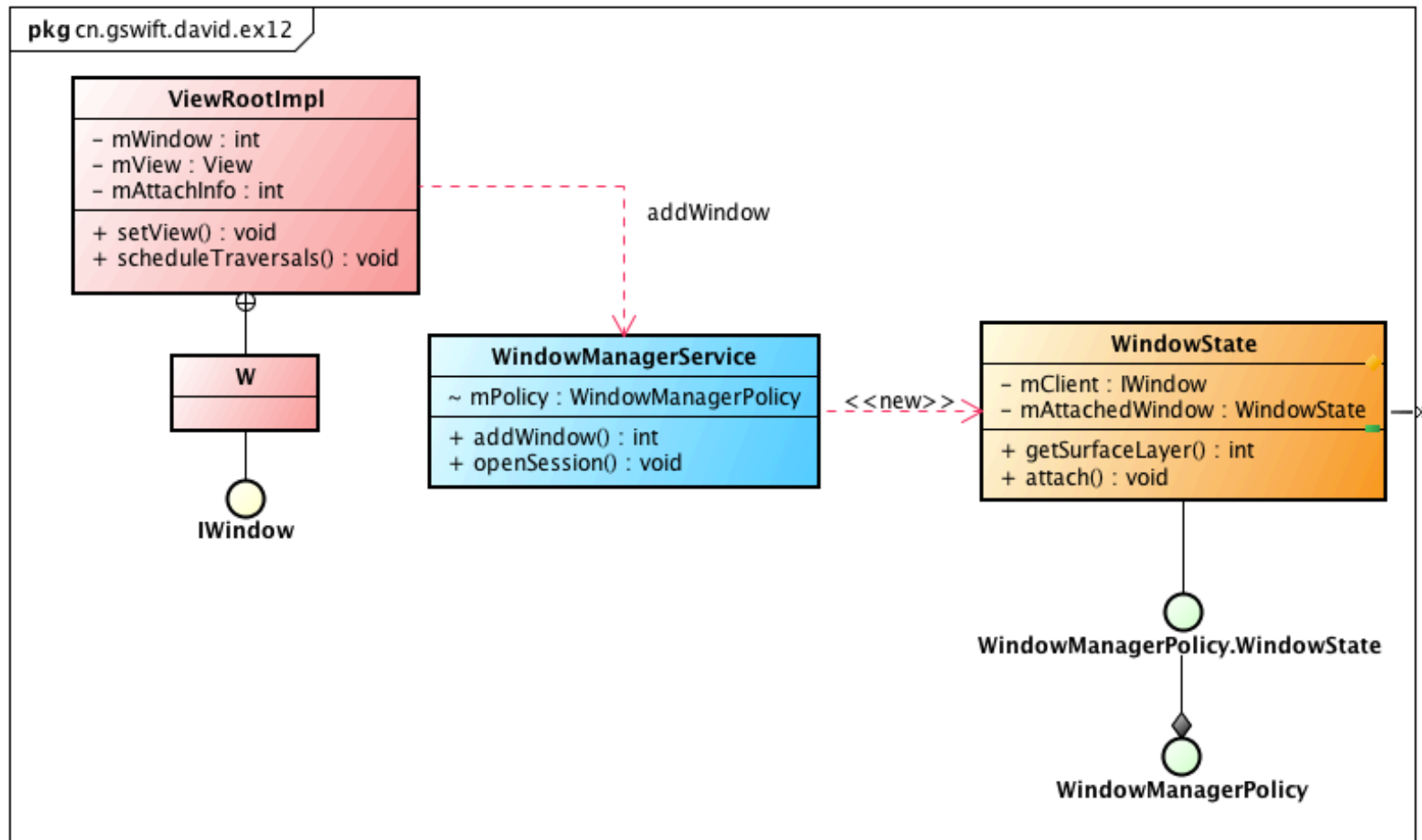
Window - Session





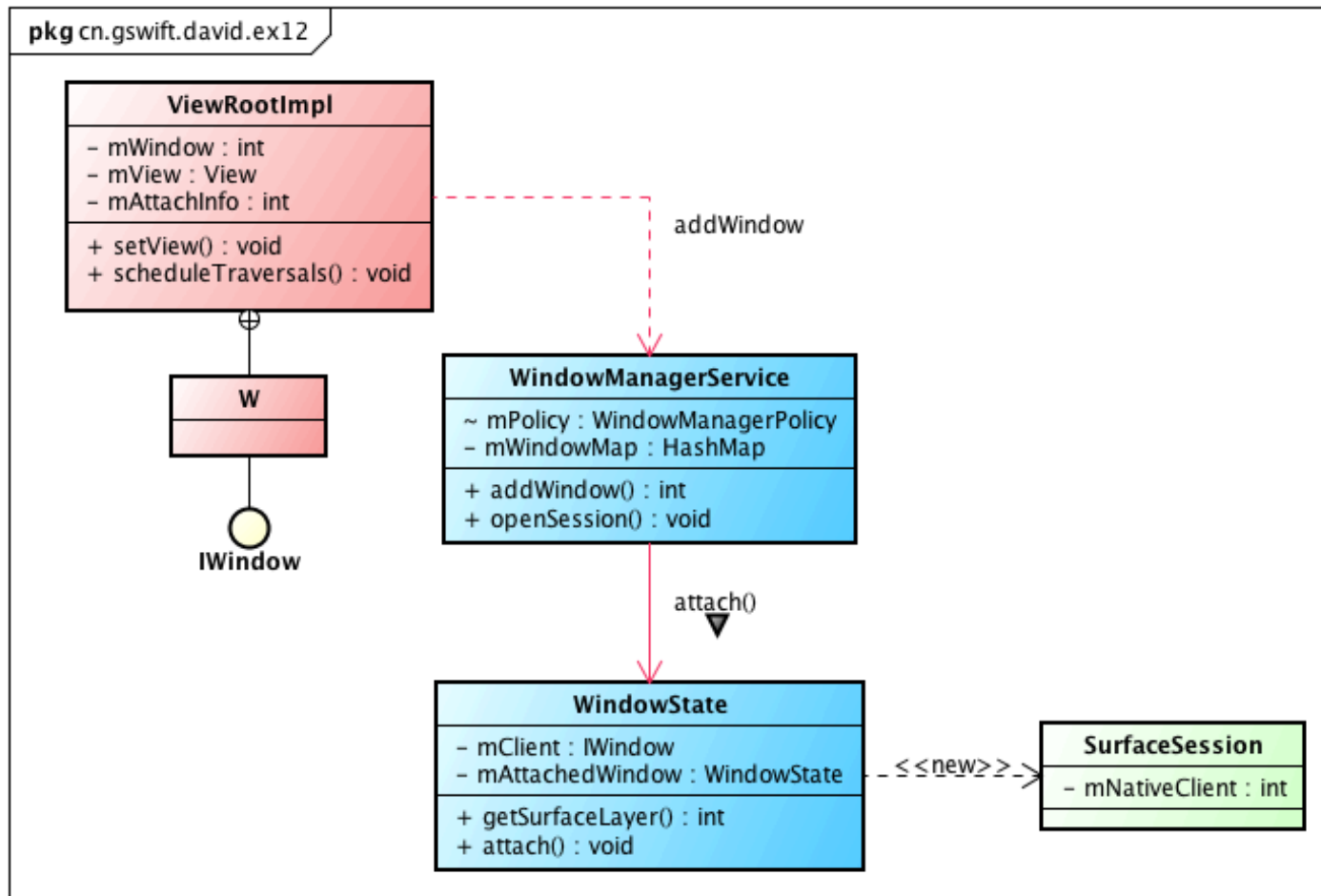
影随形动

W – Create a WindowState

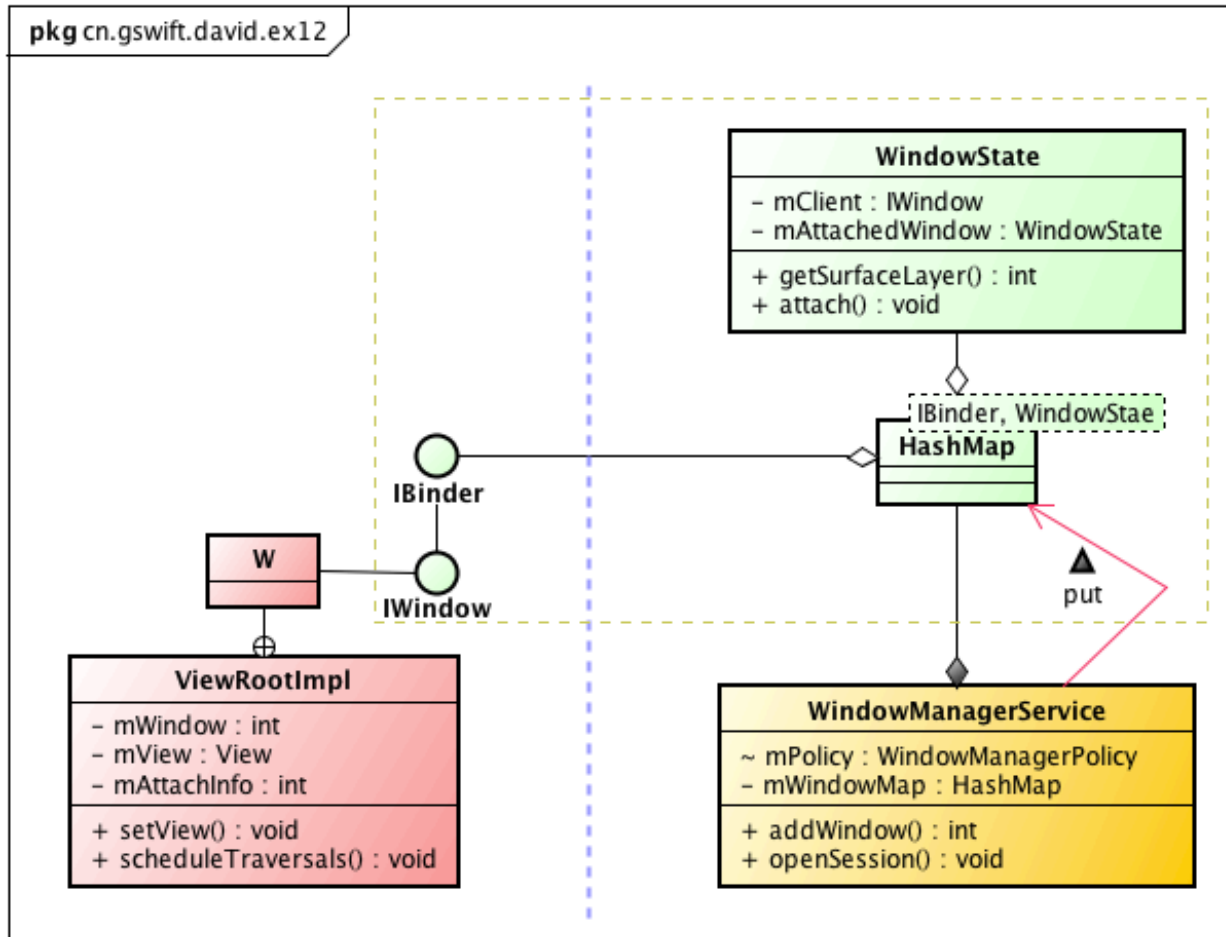


creates the window's `WindowManagerService.WindowState` instance for managing the Activity's remote window

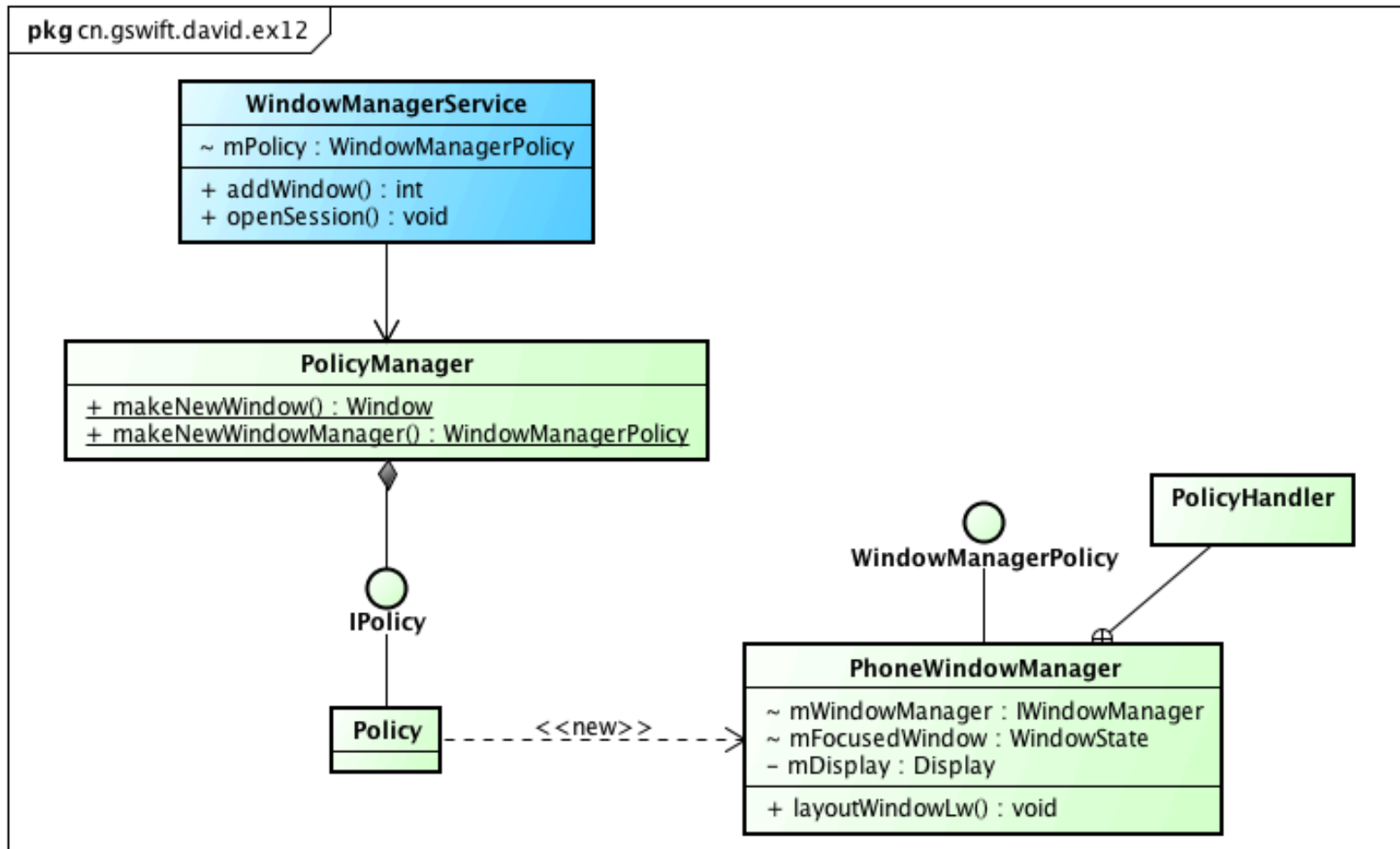
Attach WindowState to the Session



Add the IWindow to WindowManagerService



PhoneWindowManager

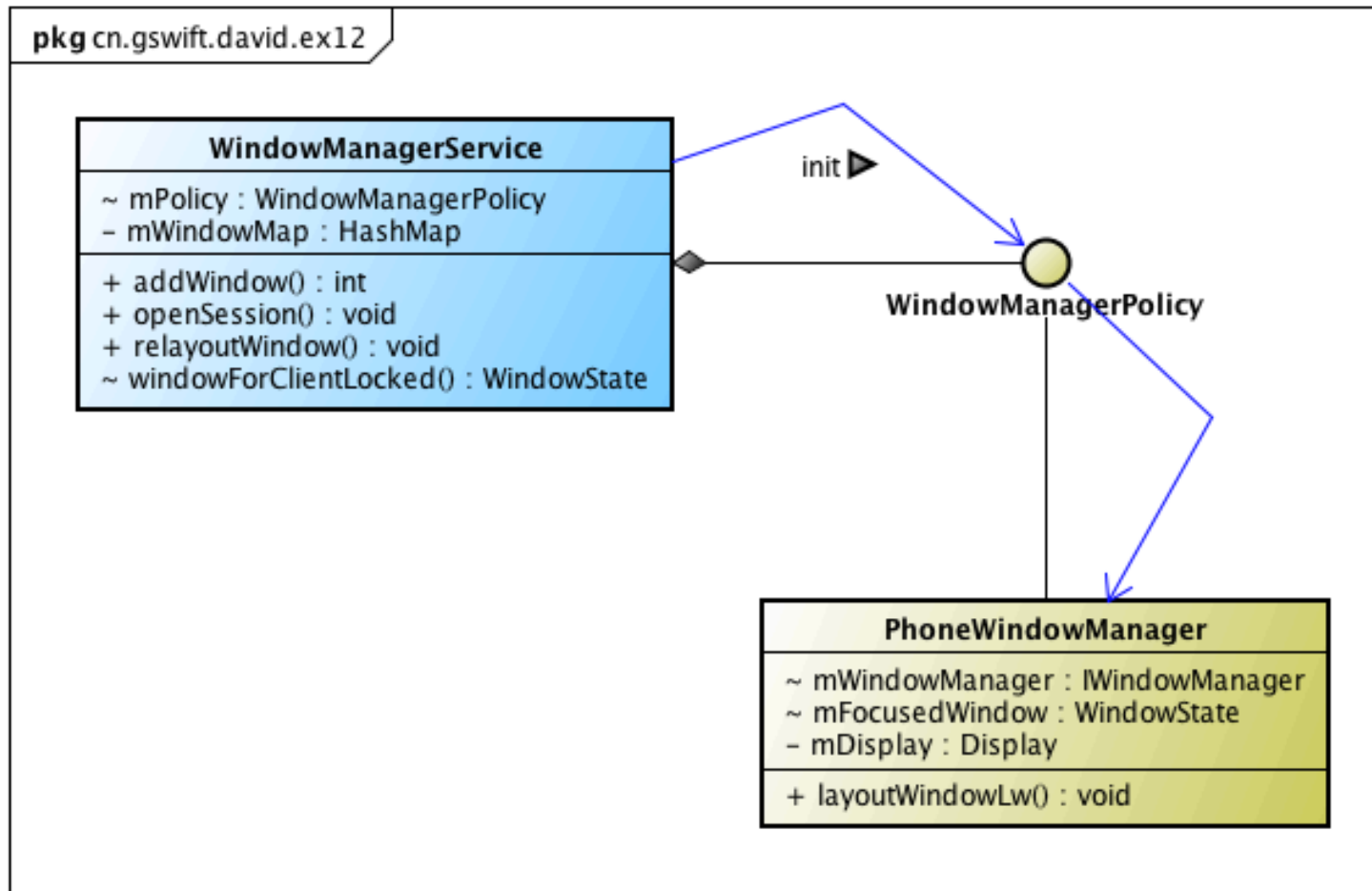




如影随形

WMS - PhoneWindowManager

Copy

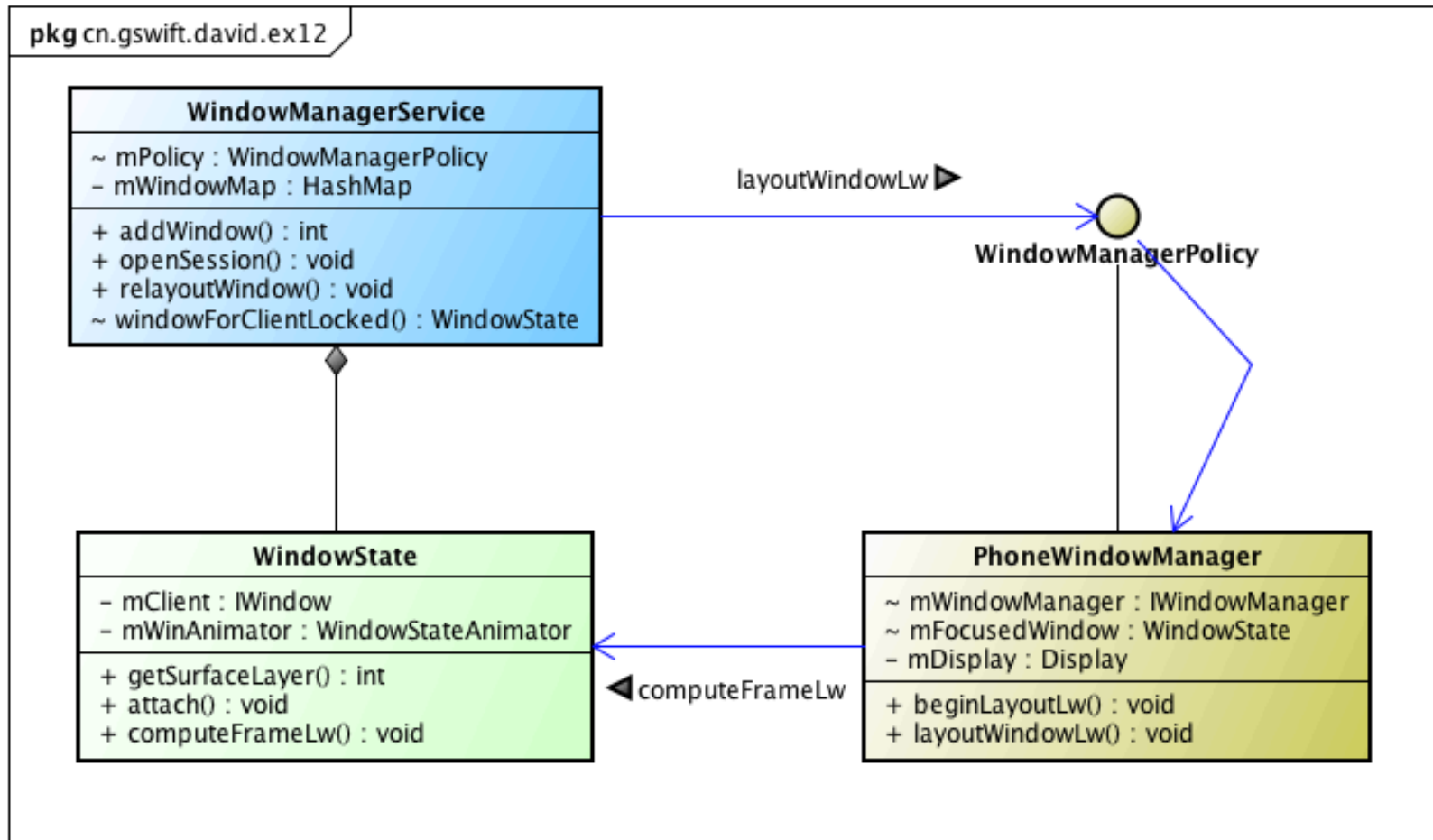


UIThread

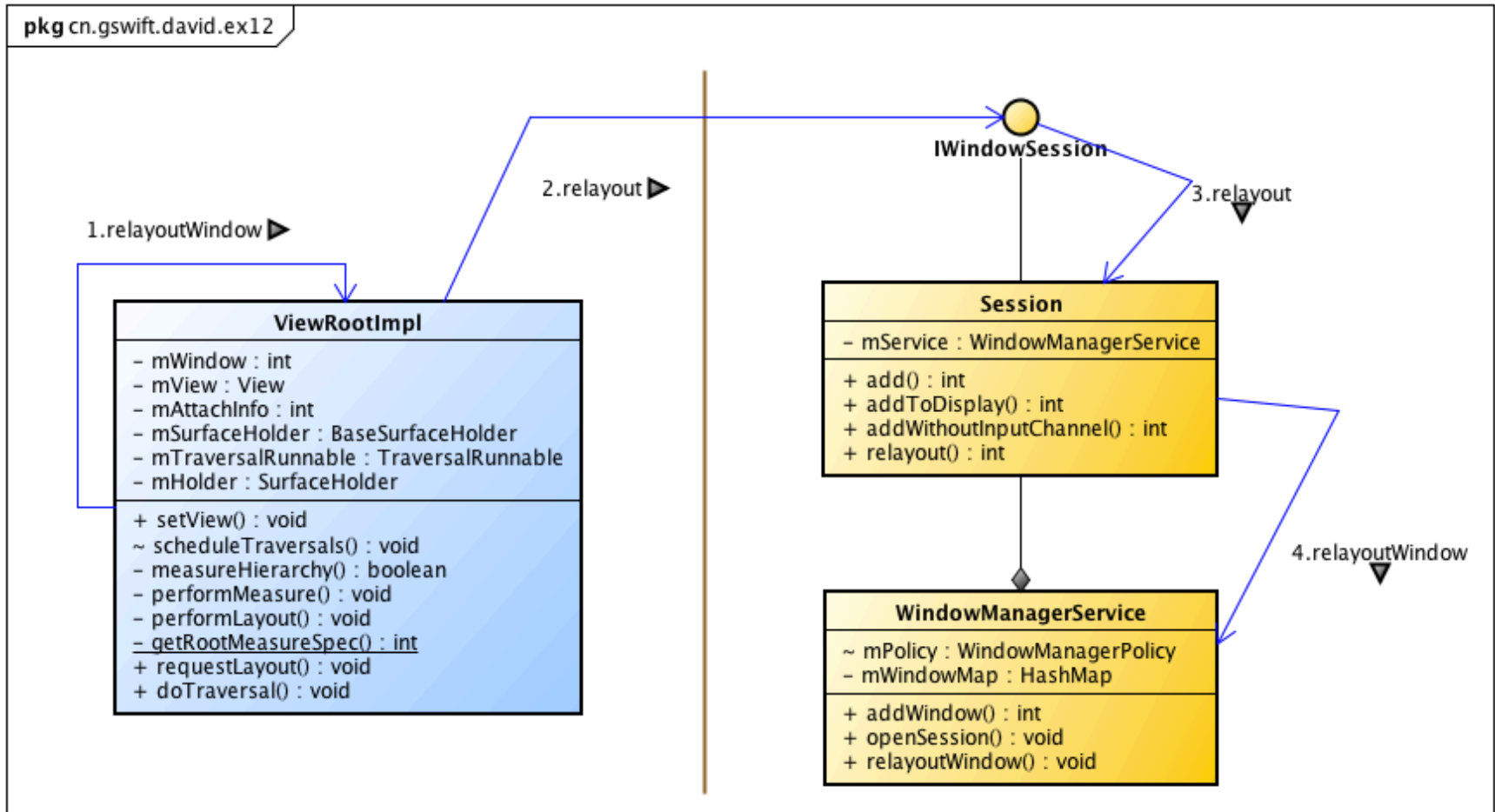


```
private void initPolicy(Handler uiHandler) {  
    uiHandler.runWithScissors(new Runnable() {  
        @Override  
        public void run() {  
            WindowManagerPolicyThread.set(Thread.currentThread(), Looper.myLooper());  
  
            mPolicy.init(mContext, WindowManagerService.this, WindowManagerService.this);  
            mAnimator.mAboveUniverseLayer = mPolicy.getAboveUniverseLayer()  
                * TYPE_LAYER_MULTIPLIER  
                + TYPE_LAYER_OFFSET;  
        }  
    }, 0);  
}
```

Get the layoutWindowLw



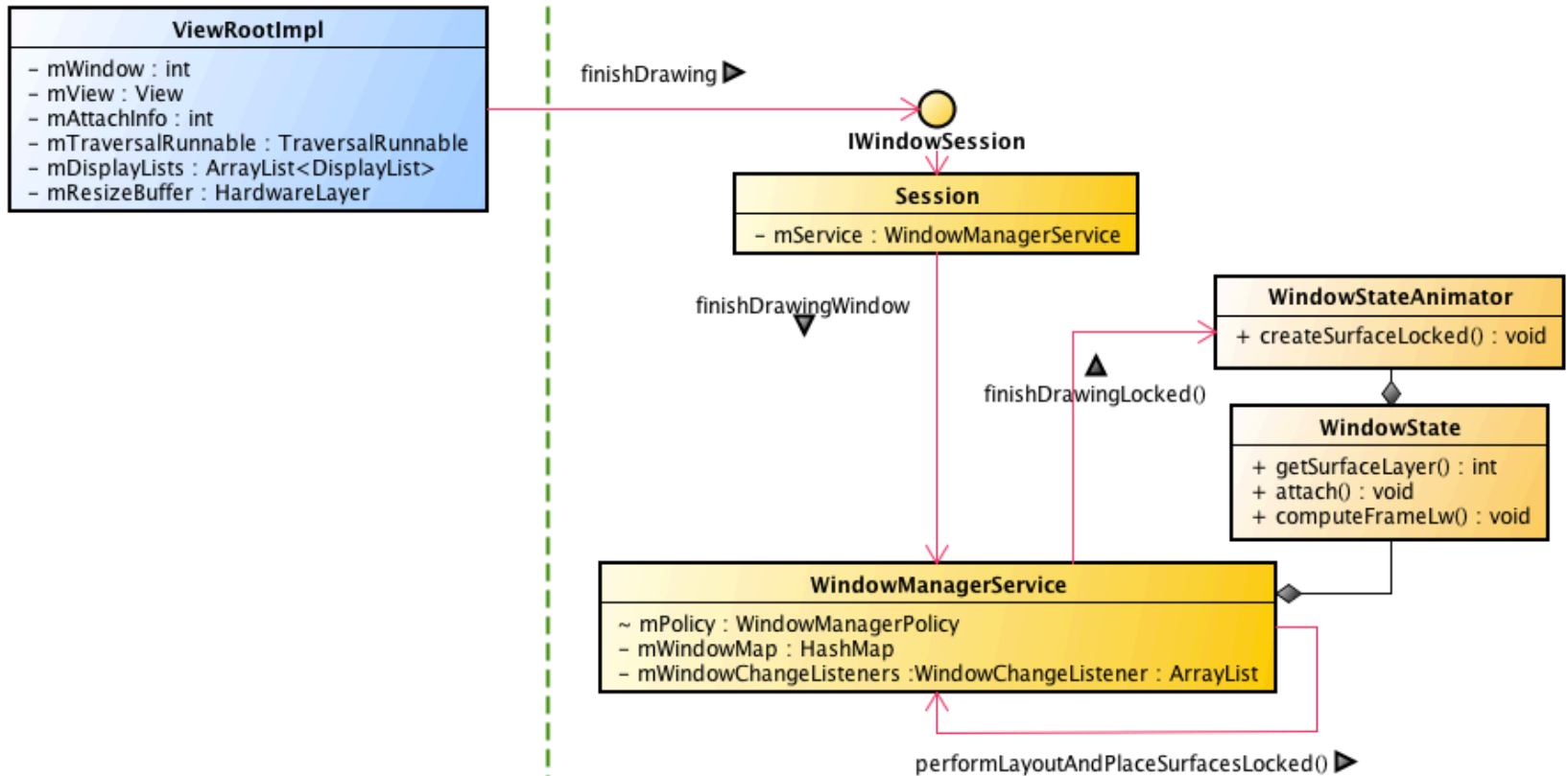
Get the Activity's window size via WMS- relayoutWindow



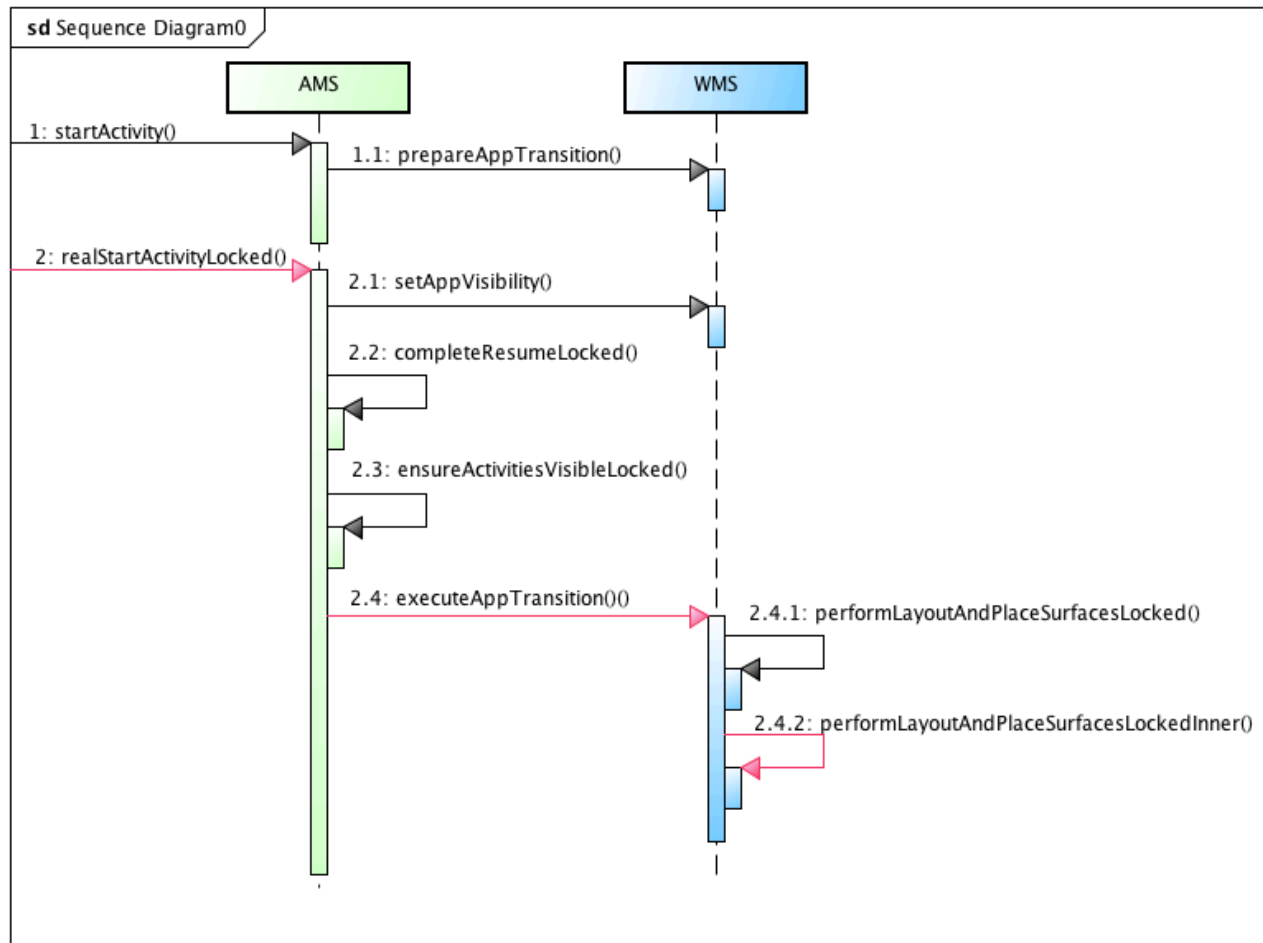
finishDrawing



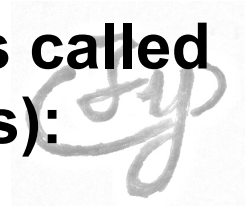
pkg cn.gswift.david.ex12



AMS-WMS



WMS.performLayoutAndPlaceSurfacesLocked is called by one of the following methods (conditions):



- WMS.removeWindowLocked
- WMS.removeWindowInnerLocked
- WMS.setInsetsWindow
- WMS.setWindowWallpaperPositionLocked
- WMS.relayoutWindow → triggered by ViewRoot.performTraversals
- WMS.finishDrawingWindow → triggered by ViewRoot.performTraversals
- WMS.removeWindowToken
- WMS.updateOrientationFromAppTokensUnchecked
- WMS.executeAppTransition → called only by the ActivityManagerService
- WMS.setAppStartingWindow
- WMS.setTokenVisibilityLocked
- WMS.unsetAppFreezingScreenLocked
- WMS.moveAppToken
- WMS.moveAppWindowsLocked
- WMS.setRotationUnchecked
- WMS.H.handleMessage(ANIMATE) → called only by the WMS and WMS.WindowState
- WMS.H.handleMessage(WINDOW_FREEZE_TIMEOUT)
- WMS.H.handleMessage(APP_TRANSITION_TIMEOUT)

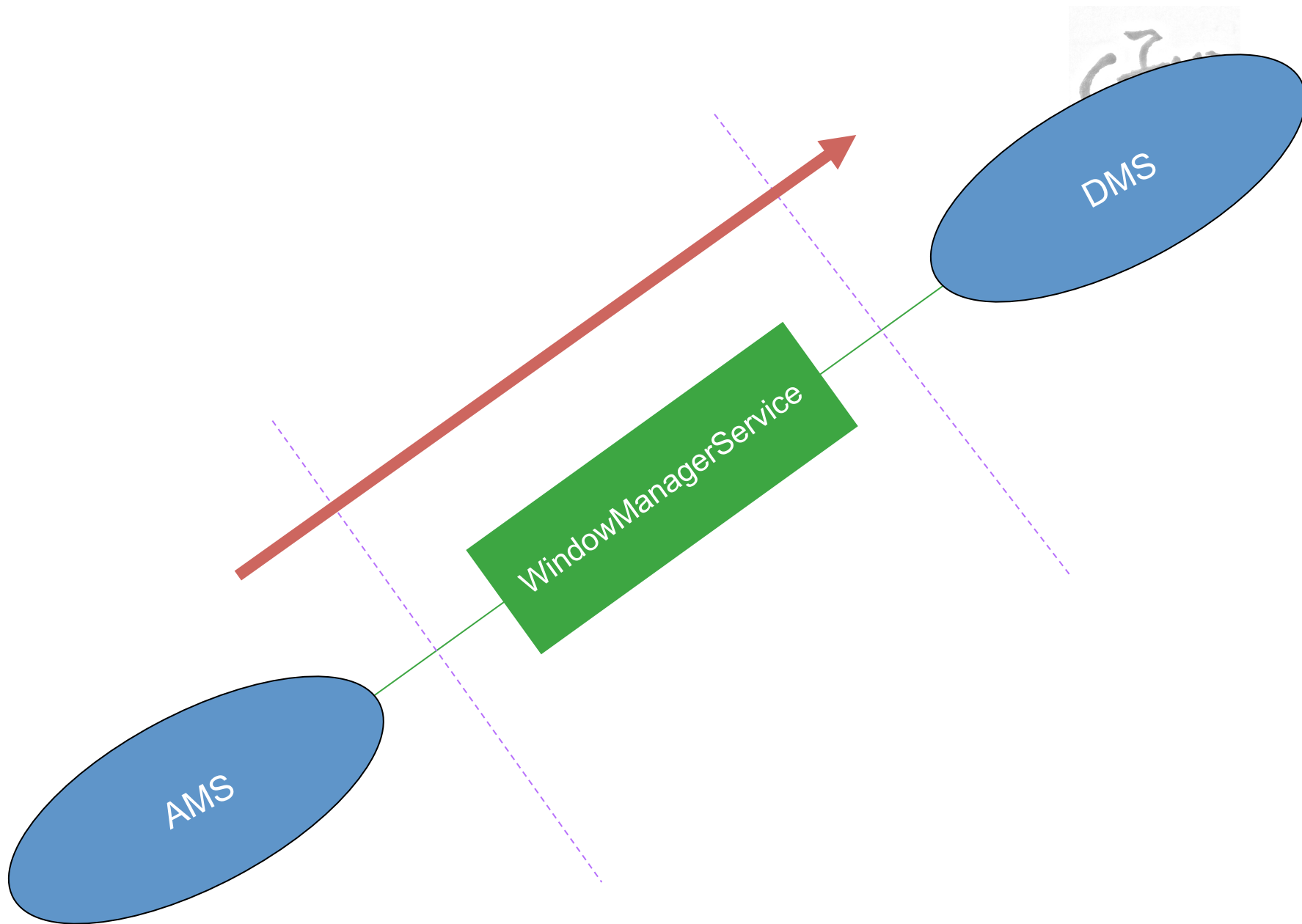
WMS.performLayoutAndPlaceSurfacesLocked pass



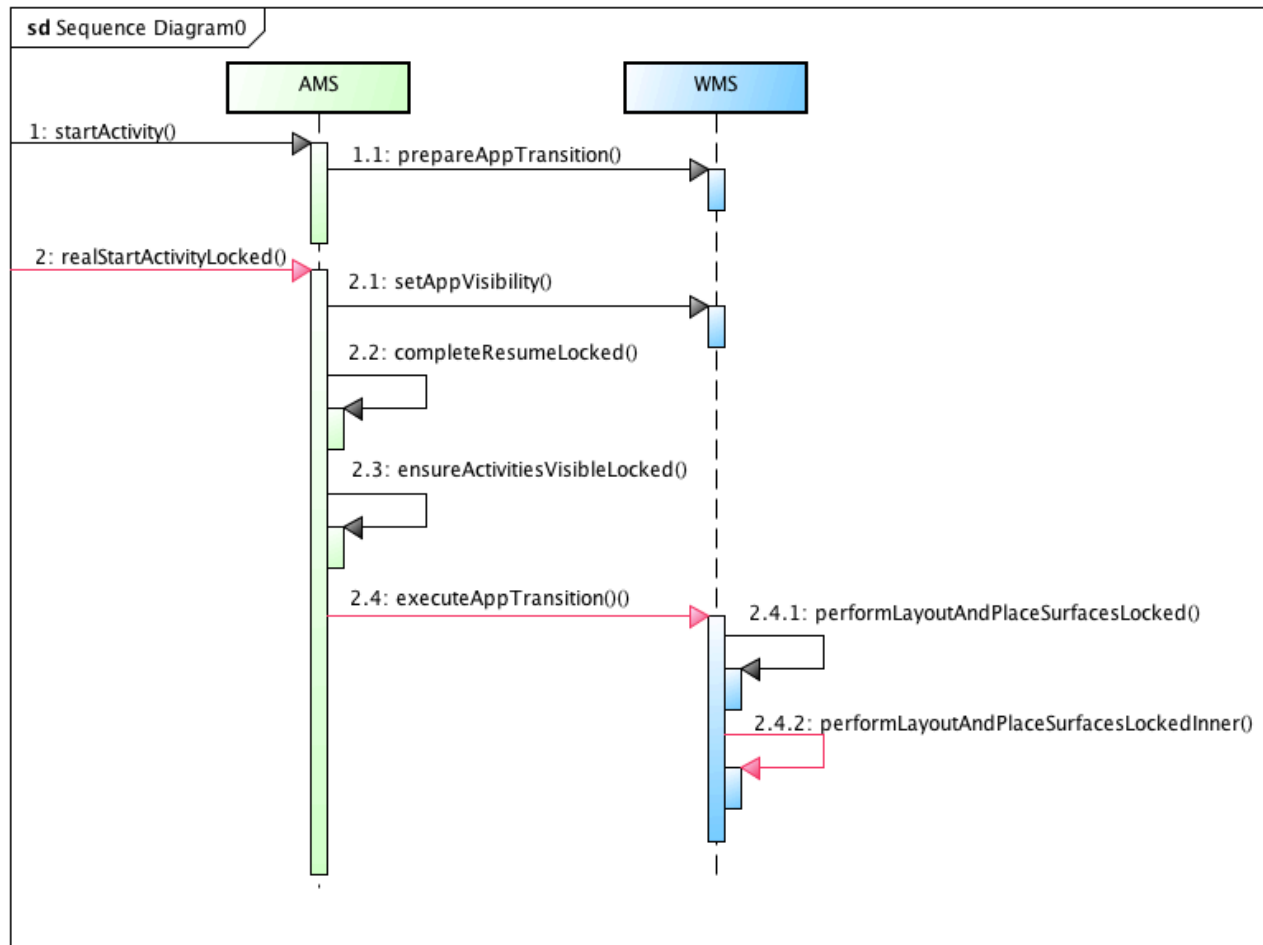
1. WindowManagerService.performLayoutLockedInner compute the window sizes and positions
2. Surface.openTransaction
3. During a WMS.performLayoutAndPlaceSurfacesLocked pass, the WMS.WindowState.stepAnimationLocked and WMS.AppWindowToken.stepAnimationLocked methods compute the transformation matrices for the desired animations (see Transformation.getTransformation->Transformation.applyTransformation).
4. Later during that WMS.performLayoutAndPlaceSurfacesLocked pass WMS.WindowState.computeShownFrameLocked computes the combined transformation matrix for that window. It also computes the size and position of the window.
5. Even later during that WMS.performLayoutAndPlaceSurfacesLocked pass Surface.setMatrix is called to transfer the computed transformation matrix to the SurfaceFlinger (LayerBase.setMatrix). Also Surface.setSize, Surface.setPosition and Surface.setAlpha are called to transfer this information into the Surface Flinger (LayerBase.cpp).
6. Surface.closeTransaction triggers the SurfaceFlinger to perform a SurfaceFlinger.threadLoop pass.
7. At the end of WMS.performLayoutAndPlaceSurfacesLocked this method triggers itself with a frequency of 60 Hz as long as some animation is running. This is done via WMS.requestAnimationLocked. WMS.requestAnimationLocked is called as long as WMS.WindowState.stepAnimationLocked or WMS.AppWindowToken.stepAnimationLocked return true.



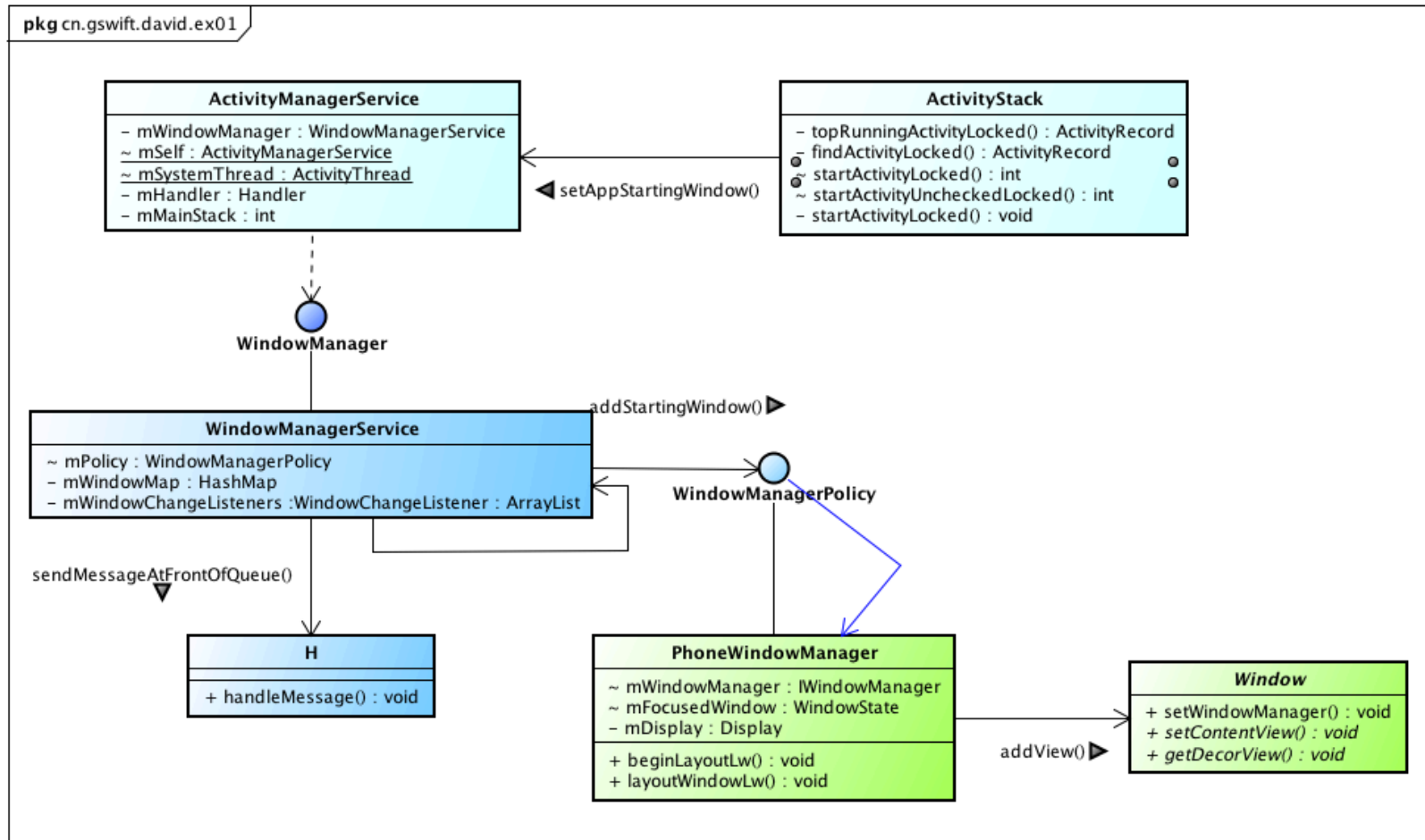
影生形



AMS-WMS

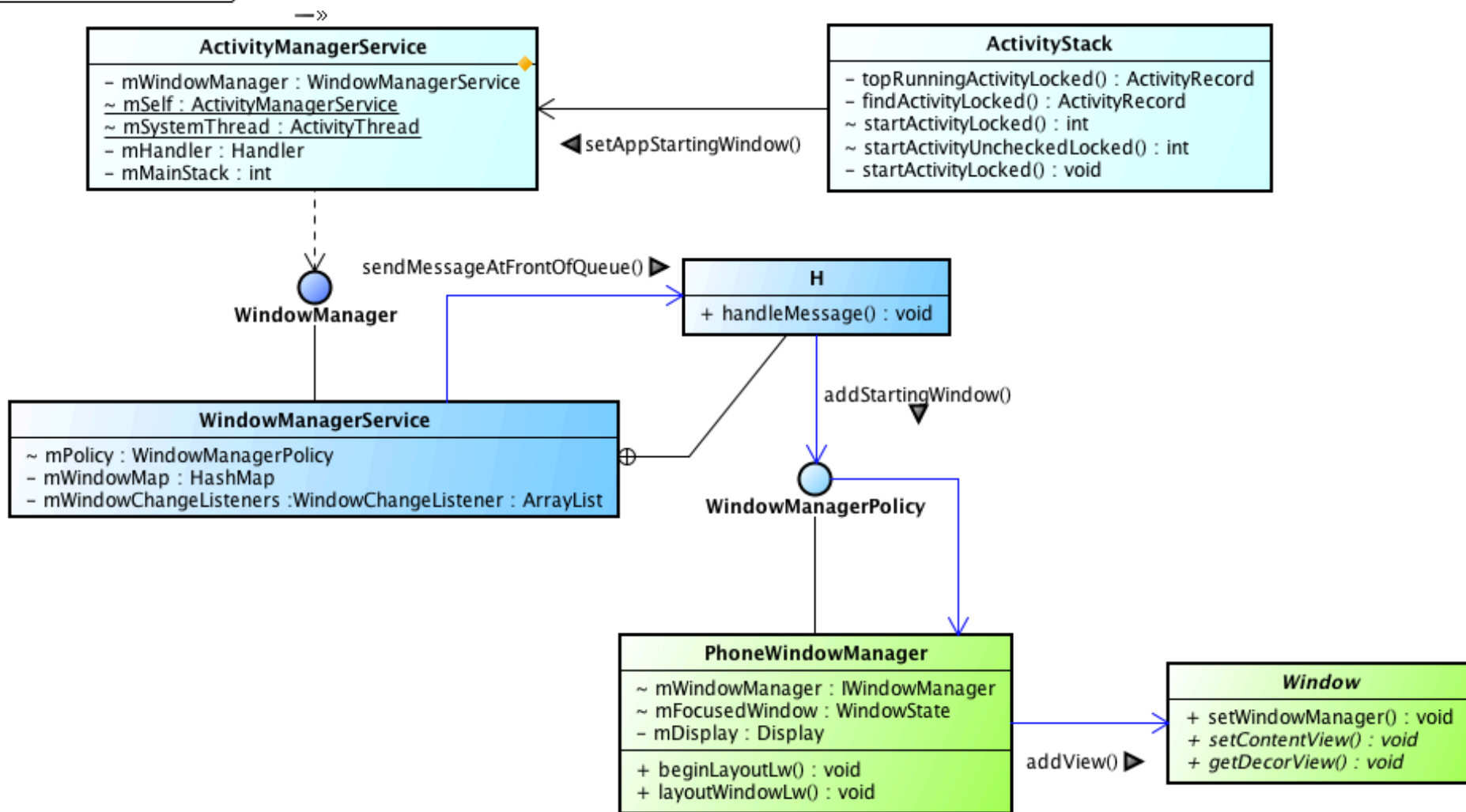


AMS-WMS

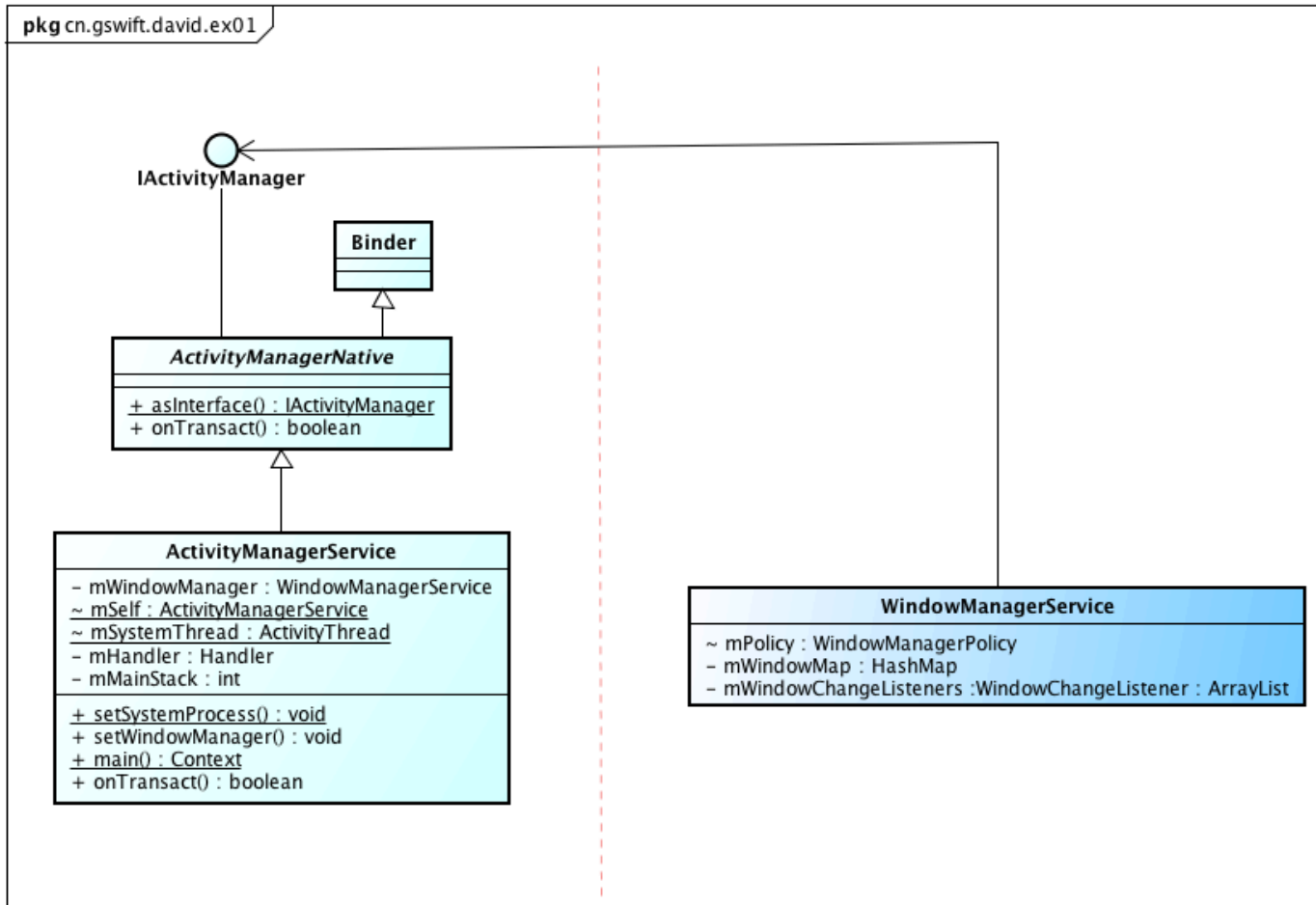


1711

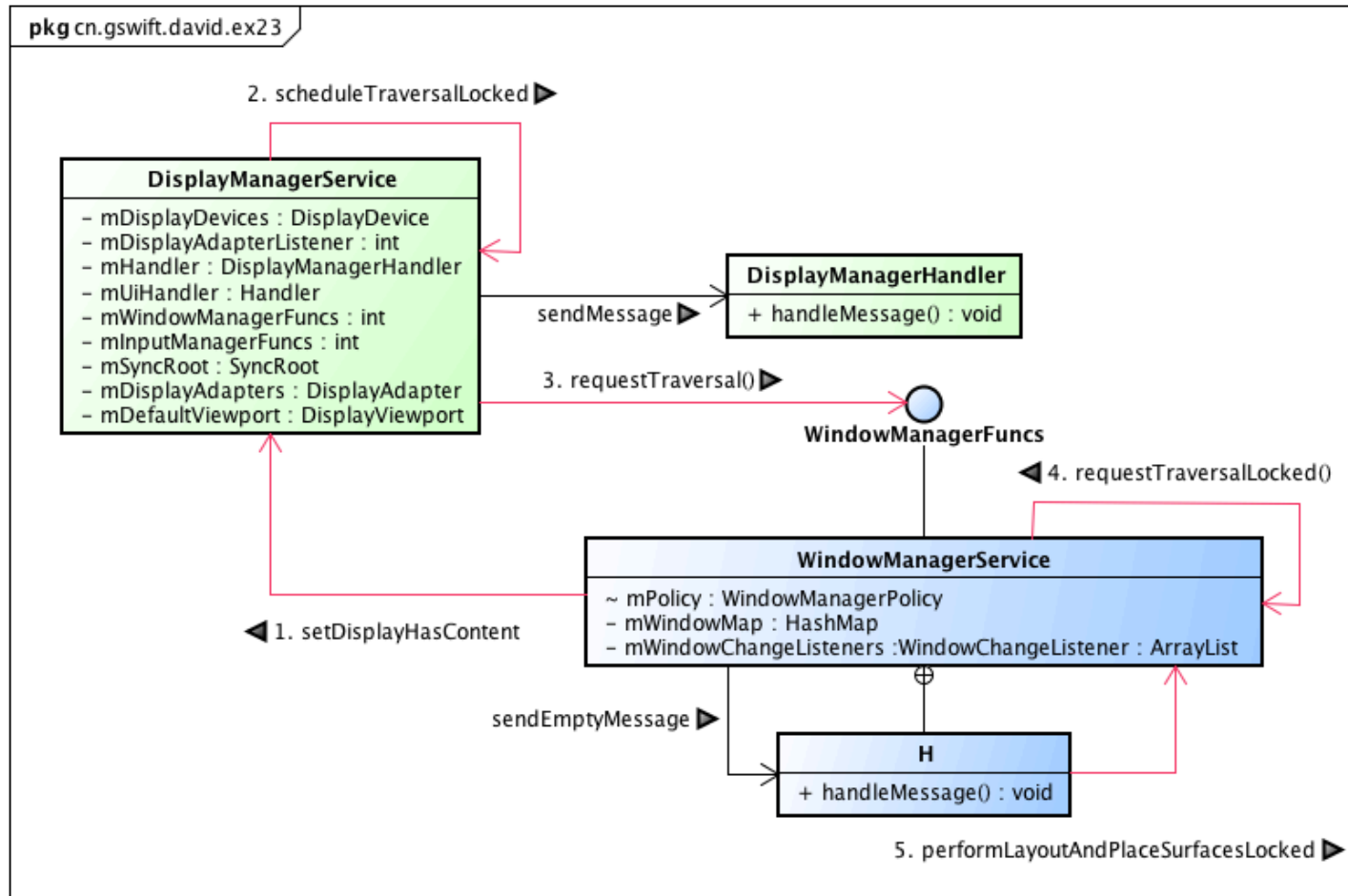
pkg.cn.gswift.david.ex01



WMS-AMS



SetDisplayHasContent in WMS





参考资料:

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

Android Graphics Architecture I ,himmele, <http://himmele.googlecode.com>

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 am 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>