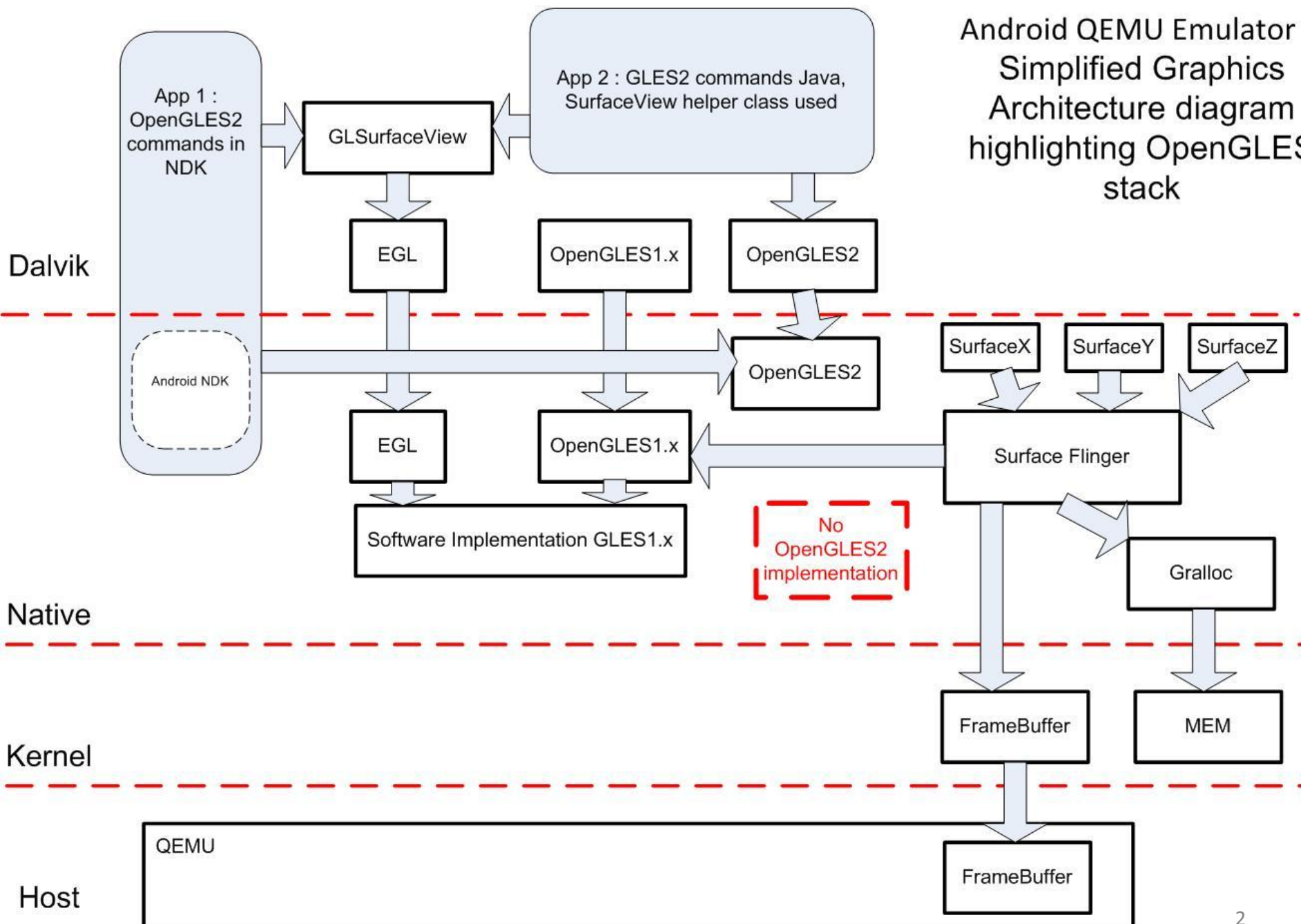


Accenture - OpenGL[®] ES 2 & Android[™] QEMU Emulator

Droidcon - London

OpenGL is a registered trademark of Silicon Graphics Inc. used by permission by Khronos.
Android is a trademark of Google Inc.”

Android QEMU Emulator Simplified Graphics Architecture diagram highlighting OpenGL ES stack

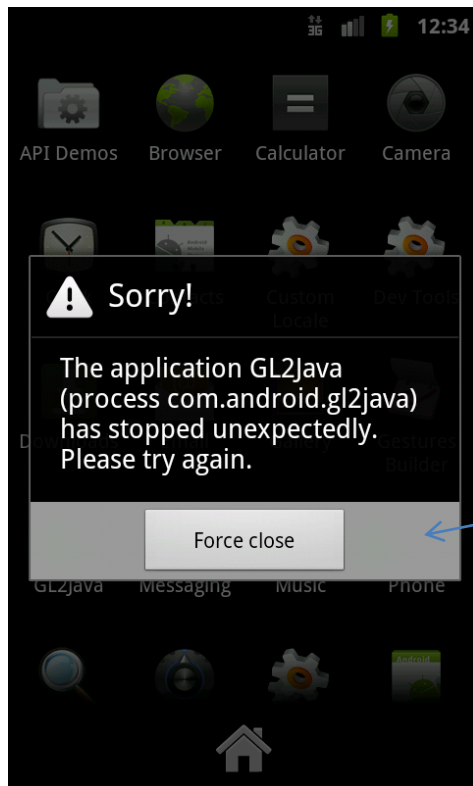


Android™ QEMU Emulator – Running OpenGL®ES 2 Applications

Problem Statement : As there is no OpenGL®ES 2 implementation can't run OpenGL®ES 2 apps in Android™ QEMU emulator

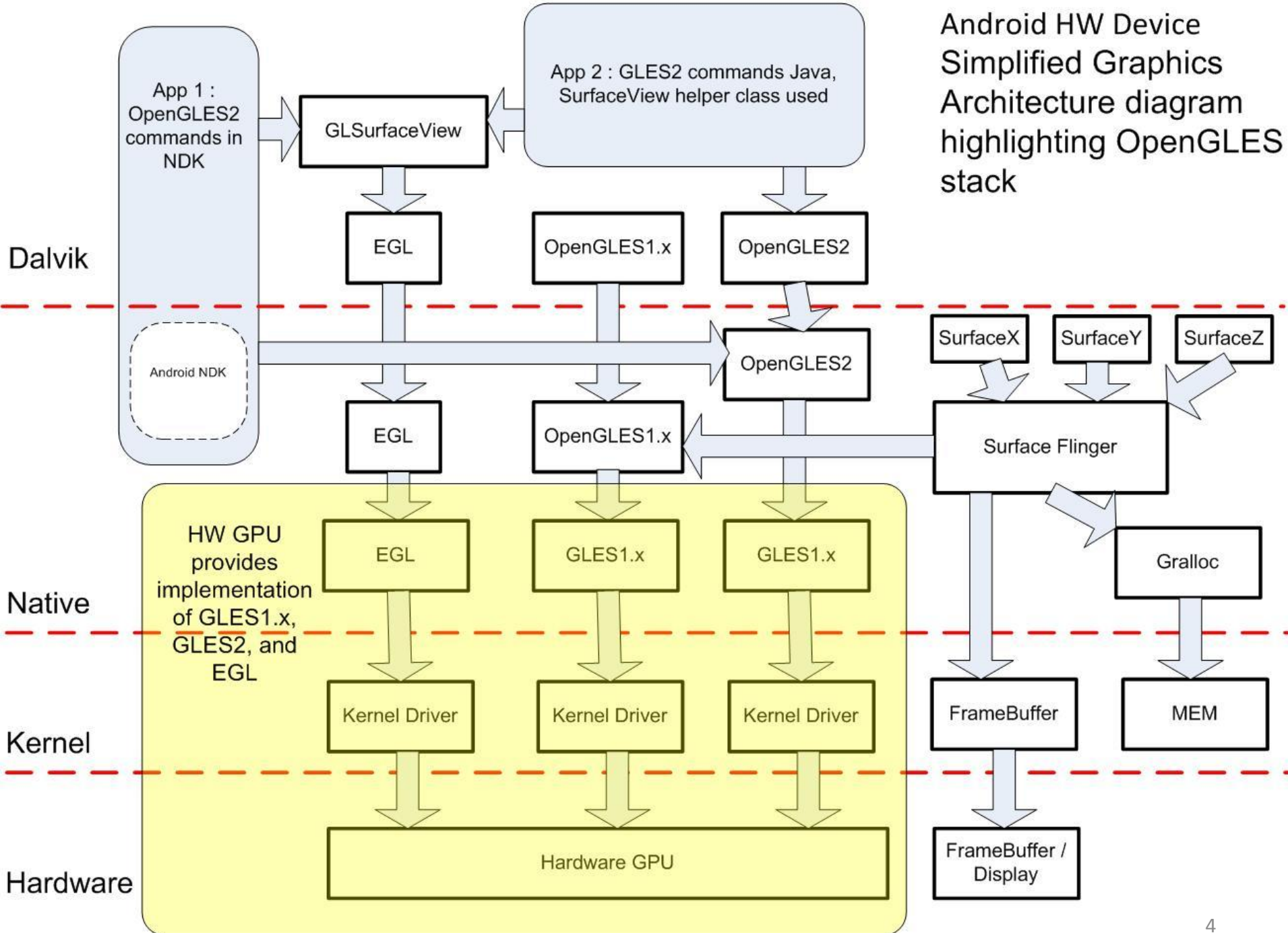
Implication : Need Android™ device(s) to develop / debug / test (App developers)

Implication : Need separate product configuration (Device manufacturers)



GLES2 application doesn't run on emulator

Android HW Device Simplified Graphics Architecture diagram highlighting OpenGL ES stack

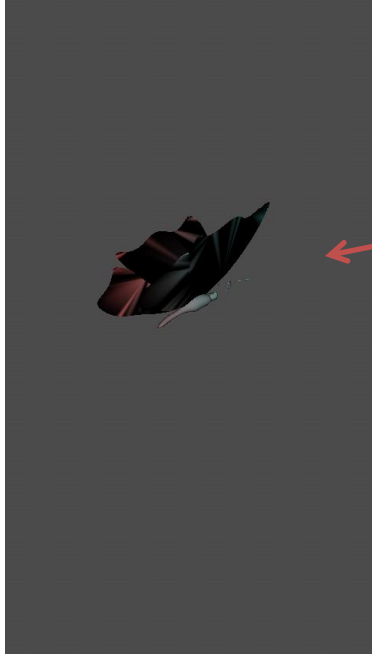


Android™ HW Device – Running OpenGL®ES Applications

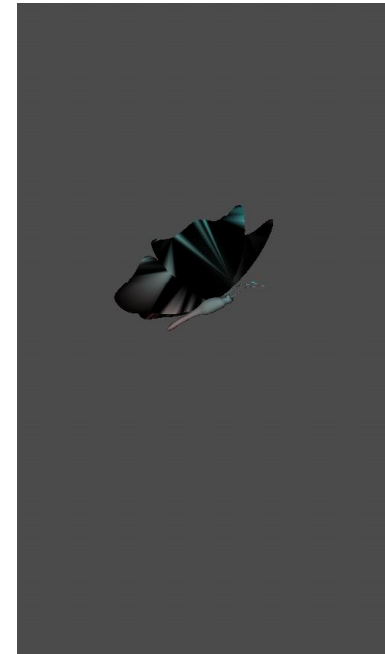
Android™ HW Devices contain GPU's which provide implementation of GLES1.x, 2.0 and EGL. Examples are Qualcomm Adreno, Imagination SGX, Nvidia, ARM Mali

Therefore you can run OpenGL®ES 2 Apps on Android™ HW devices

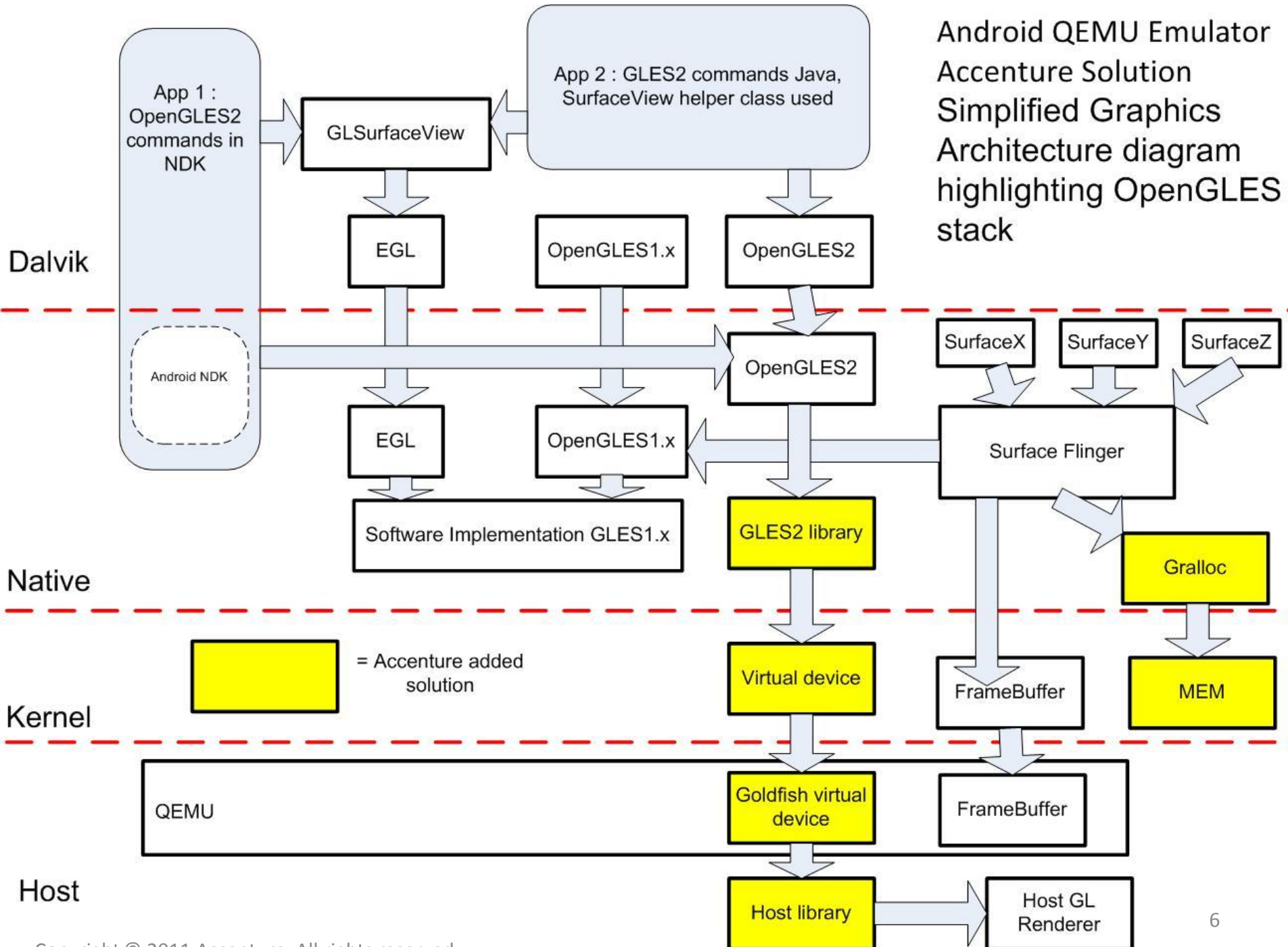
**Some low end Android™ devices don't support GLES2. Architecture can vary, for example client libraries which express OpenGL®ES 1.1 as GLES2 commands



GLES2 application running on
SEMC Xperia Arc which uses QC
Adreno 205 GPU

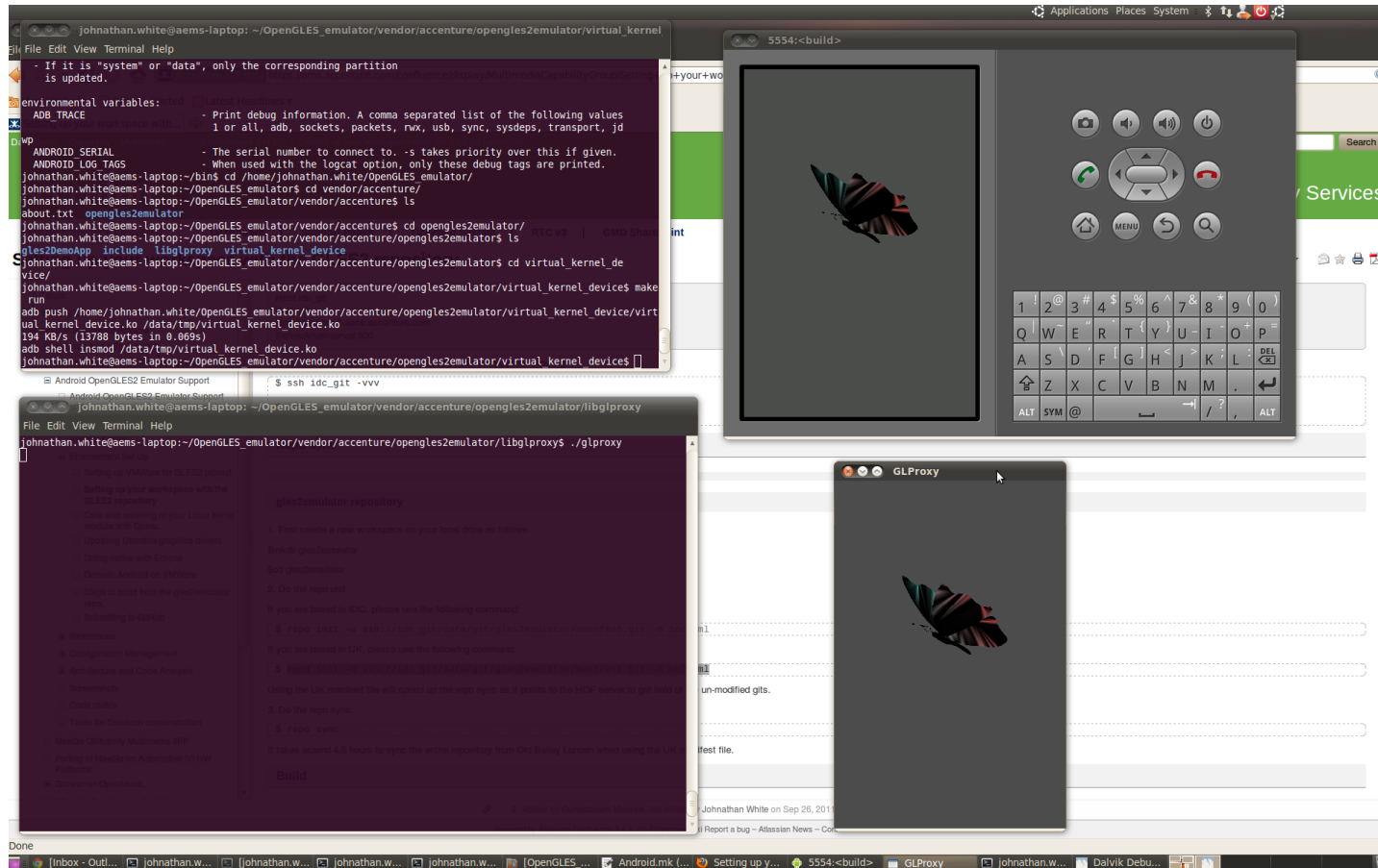


Android QEMU Emulator
 Accenture Solution
 Simplified Graphics
 Architecture diagram
 highlighting OpenGL ES
 stack



Accenture Solution Android™ QEMU Emulator – Running OpenGL®ES 2 Applications

Using the Accenture solution it is possible to run OpenGL®ES 2 application in QEMU emulator!!!



We wanted to allow you all to play with our solution so we have open sourced it !

We have used GitHub to open source the code

<http://bit.ly/AMShub> (<https://github.com/AccentureMobilityServices>)

Details of source code, building and usage instructions can be found on wiki -

<http://bit.ly/AMShubWiki>
(<https://github.com/AccentureMobilityServices/gles2emulator/wiki>)

Due to problems with android.git.kernel.org and AOSP we have provided each GIT Repo we have modified, we have also provided a default repository manifest XML. This should be edited to point at a repository of your choice.

`git://github.com/AccentureMobilityServices/kernel.git`
`git://github.com/AccentureMobilityServices/vendor_accenture.git`
`git://github.com/AccentureMobilityServices/system_core.git`
`git://github.com/AccentureMobilityServices/prebuilt.git`
`git://github.com/AccentureMobilityServices/hardware_libhardware.git`
`git://github.com/AccentureMobilityServices/frameworks_base.git`
`git://github.com/AccentureMobilityServices/external_qemu.git`

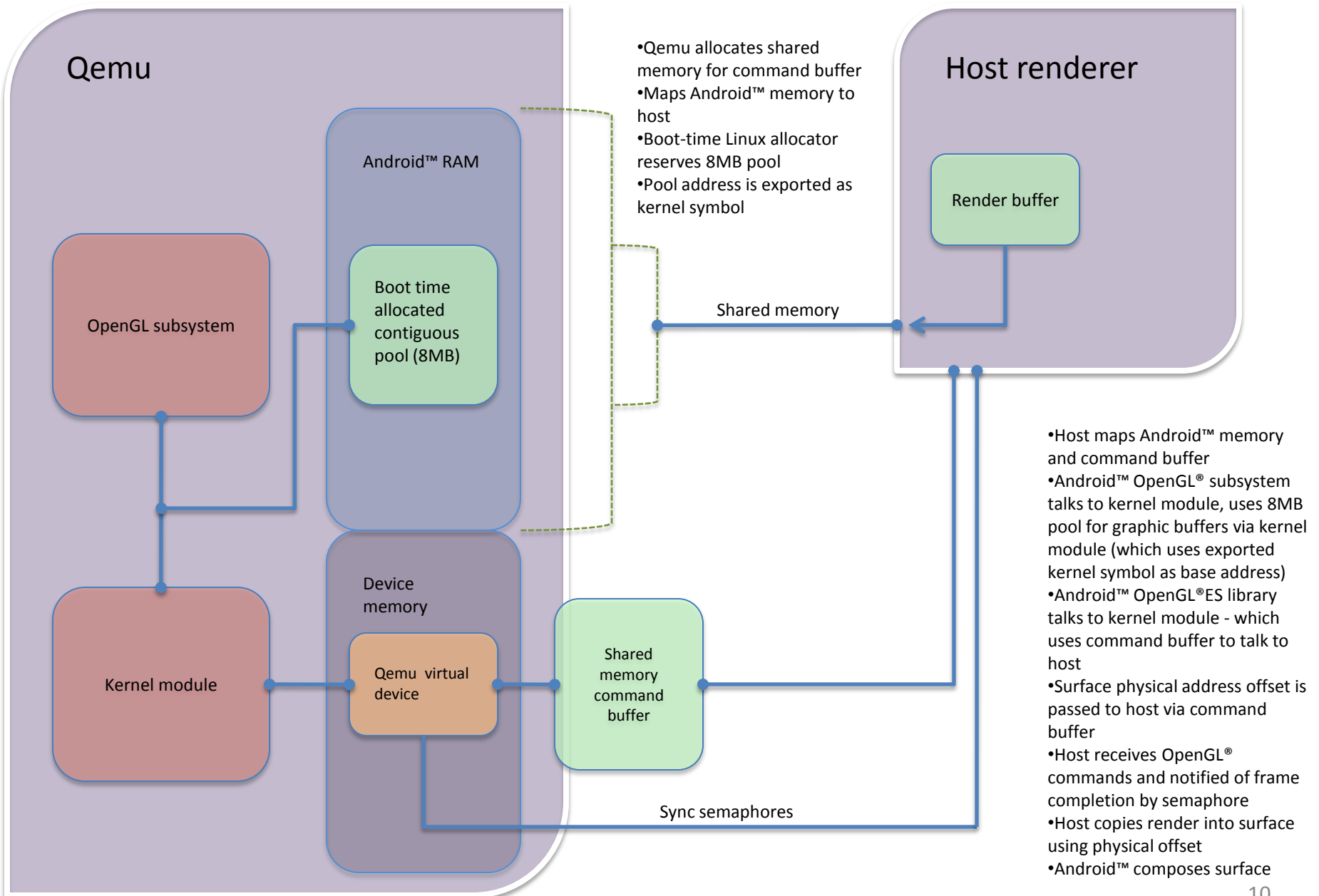
`git://github.com/AccentureMobilityServices/gles2emulator.git`

Our solution is based on Android™ 2.3.4, easy to integrate to other releases

Technical Solution – Android™ Application & OpenGL®ES 2 implementation

- OpenGL®ES 2 demo app created, GLES2 commands in JNI library (although Java API would work), object vertices loaded from file
- Normal and vertices passed to shaders, all positioning & colours & lighting done by shaders. This is very different to OpenGL®ES 1
- OpenGL®ES 1 still rendered by SW libagl. Composition done using Surface Flinger and OpenGL®ES 1
- Not complete support for OpenGL®ES 2 API yet, it is work in progress, shaders supported, no textures
- libvirtual2 functionality is to package and pass commands into buffers
- Split so libagl used for OpenGL®ES 1 and our solution for OpenGL®ES 2. Uses standard Android™ egl.cfg concept to load libraries.

Technical Solution – Memory Allocation, Buffer passing Host & QEMU



Technical Solution – Host Side OpenGL[®] rendering

- Renderer waits for packaged commands and executes as they arrive.
- Shader language similar but not the same. Can't write shader that works on both. GLES requires precision, although (at least some versions) of GL don't support this
- GLUT is used for host side windowing and integration to host system
- Developed and tested on Ubuntu

Questions ?

Will the solution work on Mac OS® operating system software or Microsoft® Windows® operating system ?
Will solution work on different versions of Android™ ?
Why have we open sourced in this way ?
Future plans ?
Why did we do it ?
Can you explain XXXX in more detail ?



Two URLs to remember –

<http://bit.ly/AMShub>

<http://bit.ly/AMShubWiki>

This document makes descriptive reference to trademarks that may be owned by others. The use of such trademarks herein is not an assertion of ownership of such trademarks by Accenture and is not intended to represent or imply the existence of an association between Accenture and the lawful owners of such trademarks.