# Virtual Environment for Android development

The goal of this project is to set up software/hardware configuration to debug and test HCN products using remote access to virtual and physical devices.

### Definition of terms

Emulator - software that enables binary code from one architecture to run on a host having another architecture (QEMU versions for each pair of host/target architectures)

Simulator - software that enables to run hardware-independent applications (VirtualBox + Android-x86 + Java VM)

Server - Linux server with emulator/simulator, X window system, desktop and VNC server

Target - execution environment for applications, virtual machine running Android

Client - development workstation running remote GUI - VNC viewer

VNC - Virtual Network Computing, platform-independent protocol that allows a desktop to be viewed and controlled remotely over the network

### Virtualization options

Based on research on tools available for Android virtualization, 3 free products were identified:

**VirtualBox** (www.virtualbox.org) - x86 and AMD64/Intel64 virtualization product, Open Source from Oracle(GPL v2 license),

Extension Pack (USB 2.0 and USB 3.0 devices, VirtualBox RDP and PXE boot) binaries are released under the VirtualBox Personal Use and Evaluation License (PUEL)

Pros:

Good integration with OS, works out of the box

Allows to use disk images created by other means

Cons:

Works only on Intel architecture

Installation of Extension Pack requires Linux system on target, on Android it may be possible with re-work of installation script and adding required system components.

**QEMU** (www.qemu.org) is a generic and open source machine emulator and virtualizer (GPL v2 license).

**QEMUA** (qemu-android) - a fork from QEMU projects with Android-specific features, it is included in Google SDK.

Pros:

Support of multiple target architectures, including x86 and ARM

Integration with SDK

Cons:

As is, it's not configured to run custom images. Some integration work is needed to run HCN system image.

Works slowly because of run time translation of binary code between different processors

**AndroidScreencast** - a tool for remote GUI access to a tablet over adb

### Installation on server and start-up

To run VirtualBox on Ubuntu server, the following packages have to be installed:

xorg, openbox, openbox-xdgmenu, Xvfb, x11vnc, virtualbox

To run qemu-android, it has to be built from source

git clone <https://qemu-android.googlesource.com/qemu-android>

To start VNC server with remote openbox desktop:

Xvfb :1 -extension GLX -screen 0 800x600x16&

DISPLAY=:1 /usr/bin/openbox-session &

x11vnc -many -create -shared -display :1

This provides remote access to display 1 on server port 5900

For multiple displays, repeat this sequence for display 2,3 and so on.

Corresponding port number will be 5901,5902,...

### Virtual Machine configuration

VMs are created from VirtualBox control window with menus for options, also from command line using VBoxManage utility. For Android, select system type and version “Unknown”.

VM requires a disk to boot from. Here is an example of how to create a disk in VirtualBox format from raw disk image:

VBoxManage convertfromraw --format VDI image.dd image.vdi

The VDI disk has to be attached to VM, this is done from Virtualbox control window.

### Remote access from client

Install VNC viewer on client:

sudo apt-get install xtightvncviewer

Connect to server:

xtightvncviewer <server name or IP address>: <port number>

This brings up openbox minimal GUI window on client.

Right click on openbox desktop and from menu select “Terminal Emulator”

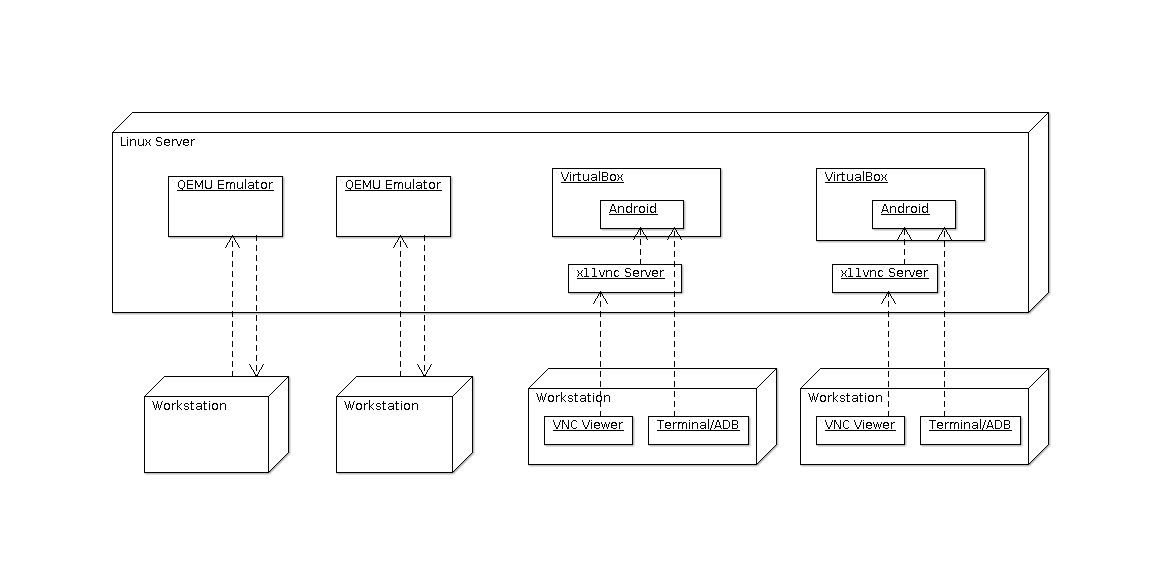
From terminal, start VirtualBox:

$ virtualbox &

In VirtualBox control window, select VM and click “Start”.

Selected VM will be shown in new window.

Fig 1. Client-server connection with x11vnc

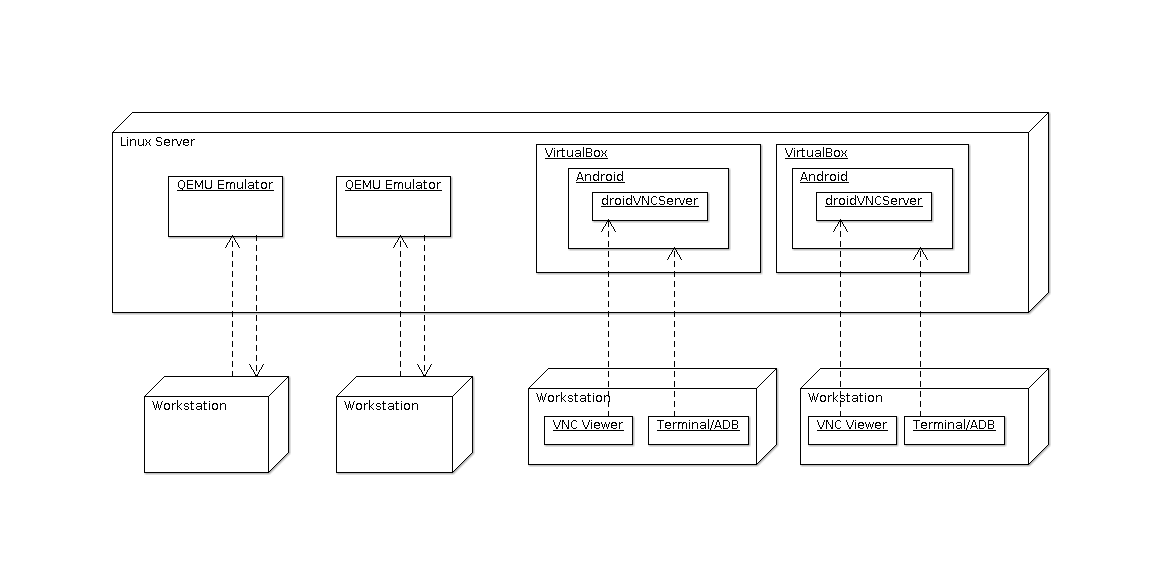


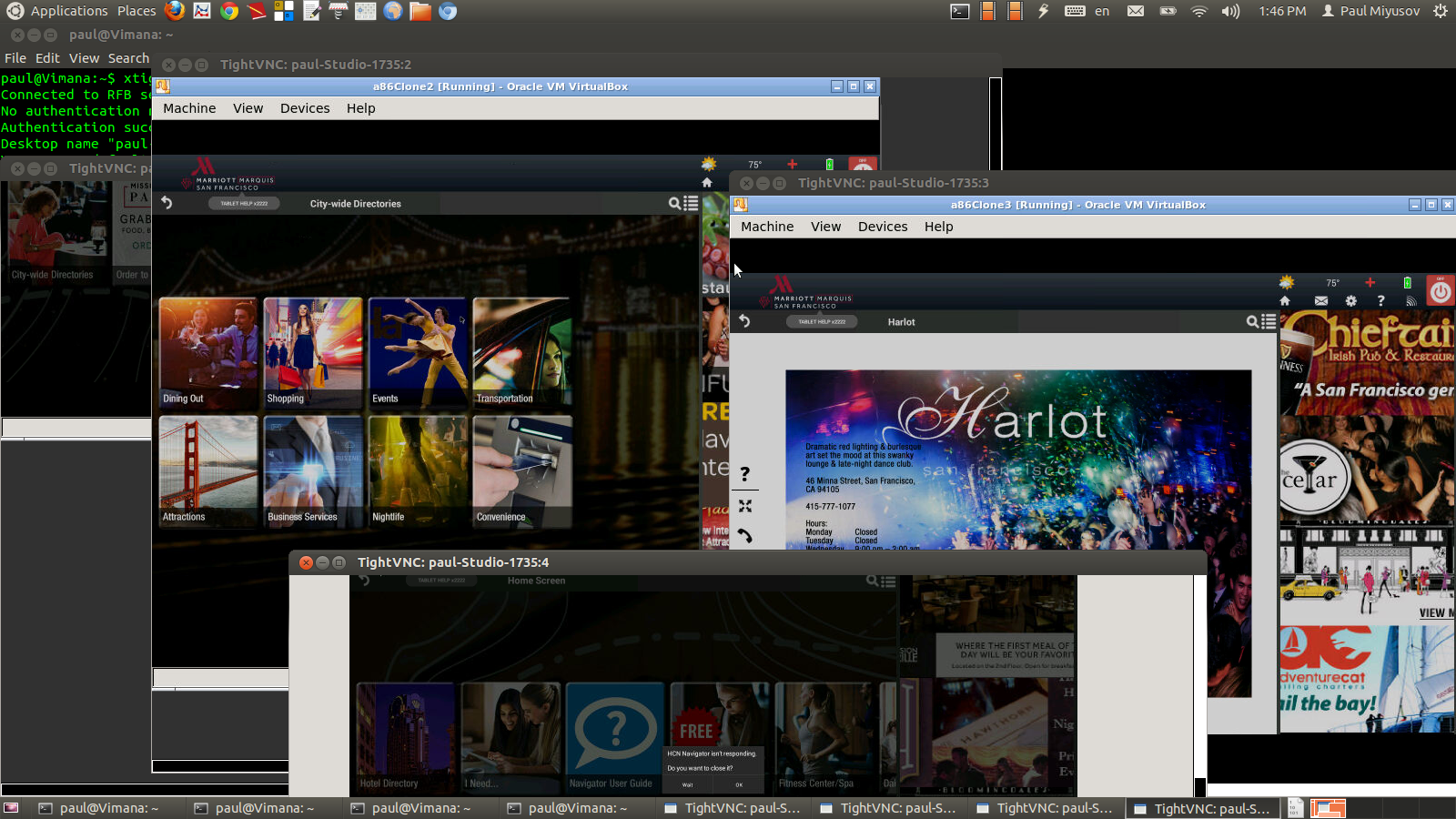
Virtual machine with VNC server

droidVNCServer is a software implementing VNC protocol on Android device, physical or virtual. It captures Android screen as opposed to desktop or server framebuffer.It is available on Google Play store as application package org.onaips.vnc (.apk) and in source code at github.com/oNaiPs/droidVncServer.

For use at HCN, it is installed on virtual machine along with HCN apps.

VM is packaged together with drive images into “virtual appliance”, .ova file that can be imported for use in Virtualbox installation on server.

Fig 2. Client-server connection with droidVNCServFig 3. Running 4 VMs with HCN Navigator application



### AndroidScreencast tool for remote GUI access to a tablet

This tool can be useful for development, support and demos.

It was taken from source repo on github, built and tested at HCN.

It can be downloaded from pub\_share:

wget <http://qc/pub_share/AndroidScreencast.zip>

Here is how to use it.

1. Find out IP address of a tablet from bluemonk or by other means.

Assume here it is 10.16.0.39.

2. Configure tablet to use adb over IP.

To do this, ssh to the tablet, set a property and restart adbd on tablet:

root@navtab:/ # setprop service.adb.tcp.port 5555 ;stop adbd; start adbd

This setting will be lost on reboot unless it is added to a startup file

3. On desktop, restart adb with IP address to connect:

export ADBHOST=10.16.0.39 ;adb kill-server; sleep 2; adb start-server

4. Start androidscreencast from command line:

java -jar androidscreencast.jar

If there are more than one devices visible to adb, select emulator-5554 in GUI menu.

5. If this is a first connection to this tablet, confirm connection on tablet's touch screen.

A screen, identical to tablet, is shown on desktop.

Use mouse emulating touches and desktop keyboard.

Note that there is about 5 seconds delay to synchronize screens. This is because adb takes screenshots of full screen.

Link: https://github.com/xSAVIKx/AndroidScreencast

### Current status

VirtualBox - configured to run HCN x86 image, remote GUI via VNC. Debugging over ethernet/adb

Separate Virtual appliance configured and is available for download and install on desktop

Two SDK emulators configured with HCN apps - for ARM and for x86

AndroidScreencast - available for download as a zip file with instructions

QEMUA - built from source for all supported architectures. Installed on U64 LXC

Download link from U64 LXC: http://10.0.0.47:8000/virtual/

### Current problems (TODO items)

Keyboard integration - special keys don’t work, no way to switch to console.

Mouse integration - cut&paste between desktop and VM window doesn’t work.

Screen size has to be coordinated between framebuffer, VNC server and viewer.

Need a convenient way to get IP address of VM to connect adb. One way is to show IP address in “About Tablet” window.

Setting for screen “Never sleep” in Android image, no way to wake it up.

droidVNCServer works on VMs but not on tablets or legacy Navigators