mVDI: A New Paradigm Shift for Mobile Cloud

Professor: Eui-Nam Huh
Innovative Cloud & Security Lab
Kyung Hee University – Global Campus


PC Era
Cloud Era


A World Where People Can Work
or Play from Anywhere

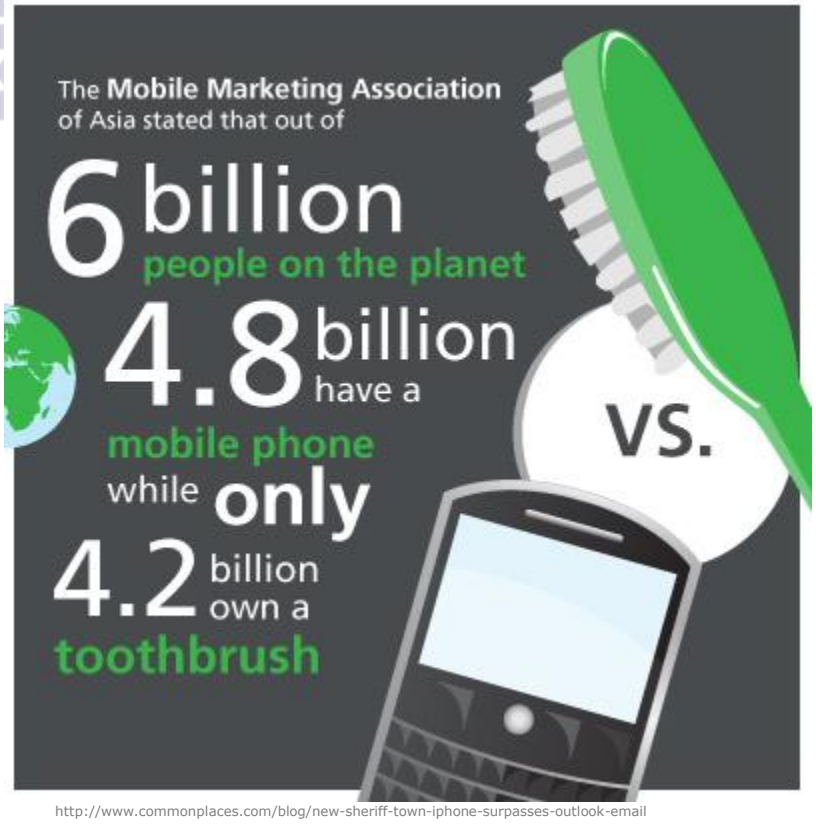
Presentation Overview

- Trends
- Concepts of DV/VDI/mVDI
- Issues – Solutions – Challenges
  - M-Virtualization
  - Network/CPU Consumption
  - NDK Apps Incompatibility
  - OpenGL Performance
  - Web-based mVDI
- Summary

❖TRENDS

The rage of mobile devices

2012



http://www.commonplaces.com/blog/new-sheriff-town-iphone-surpasses-outlook-email

10 Billion

Mobile devices by 2020
Gartner

8x Faster Adoption than internet

and if tablets included, even faster Morgan

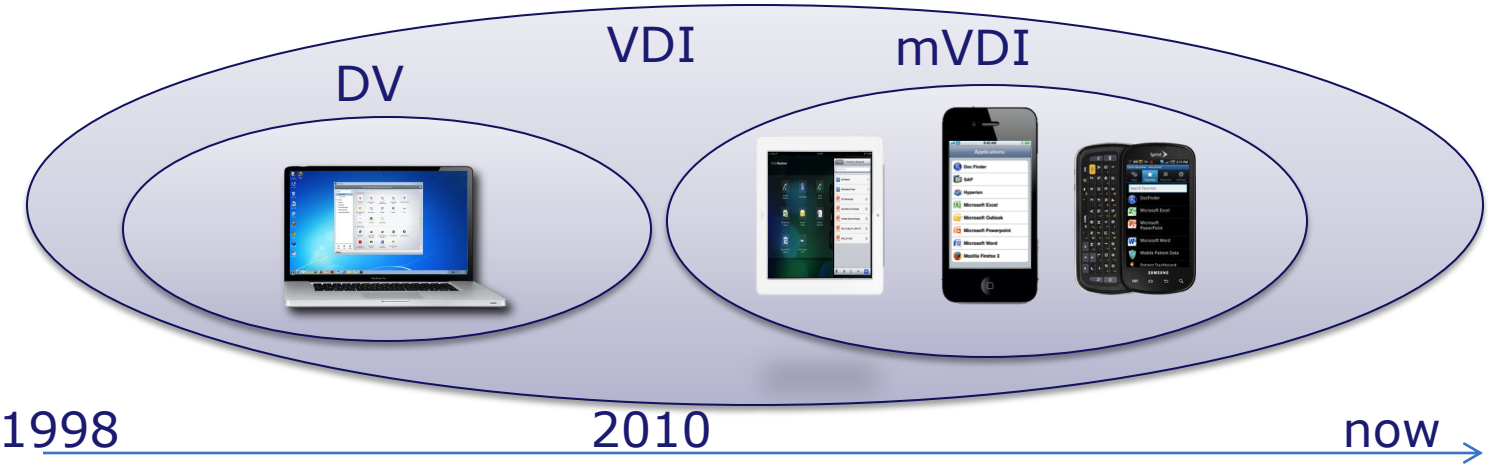
Trends

	Standard	Download	Upload
2.5G	GPRS	114 Kbps	20 Kbps
2.75G	EDGE	384 Kbps	60 Kbps
3G	UMTS	384 Kbps	64 Kbps
	W-CDMA	2 Mbps	153 Kbps
	HSPA 3.6	3.6 Mbps	348 Kbps
	HSPA 7.2	7.2 Mbps	2 Mbps
Pre-4G	HSPA 14	14 Mbps	5.7 Mbps
	HSPA+	56 Mbps	22 Mbps
	WiMAX	6 Mbps	1 Mbps
	LTE	100 Mbps	50 Mbps
4G	WiMAX 2	1 Gbps	500 Mbps
	LTE Advanced	1 Gbps	500 Mbps

Peak data rate of different mobile network

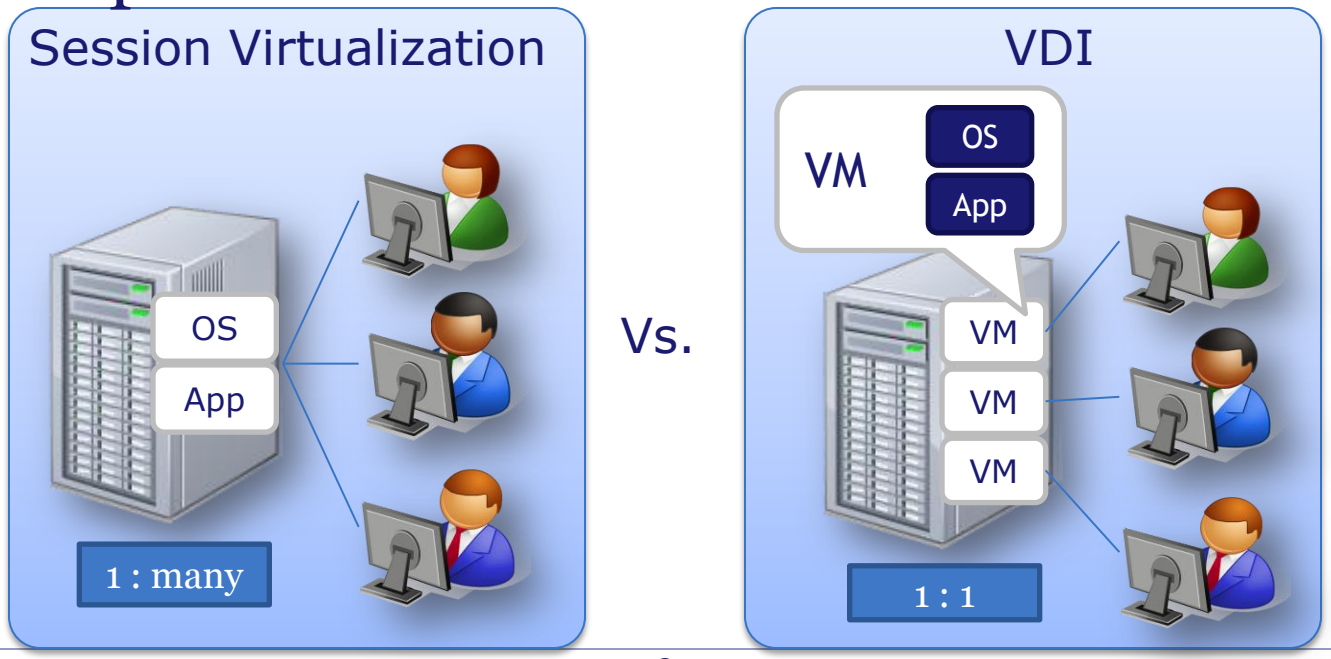
What is DV, VDI, and mVDI?

- Virtual Desktop Infrastructure (VDI) is an alternative desktop delivery model that allows users to run an OS/application inside a Virtual Machine (VM) running in the data center.
- mVDI is the evolution of server virtualization supported for mobile devices.
- mVDI has the same look and feel of a traditional mobile device, but all actions are happening remotely in the data center

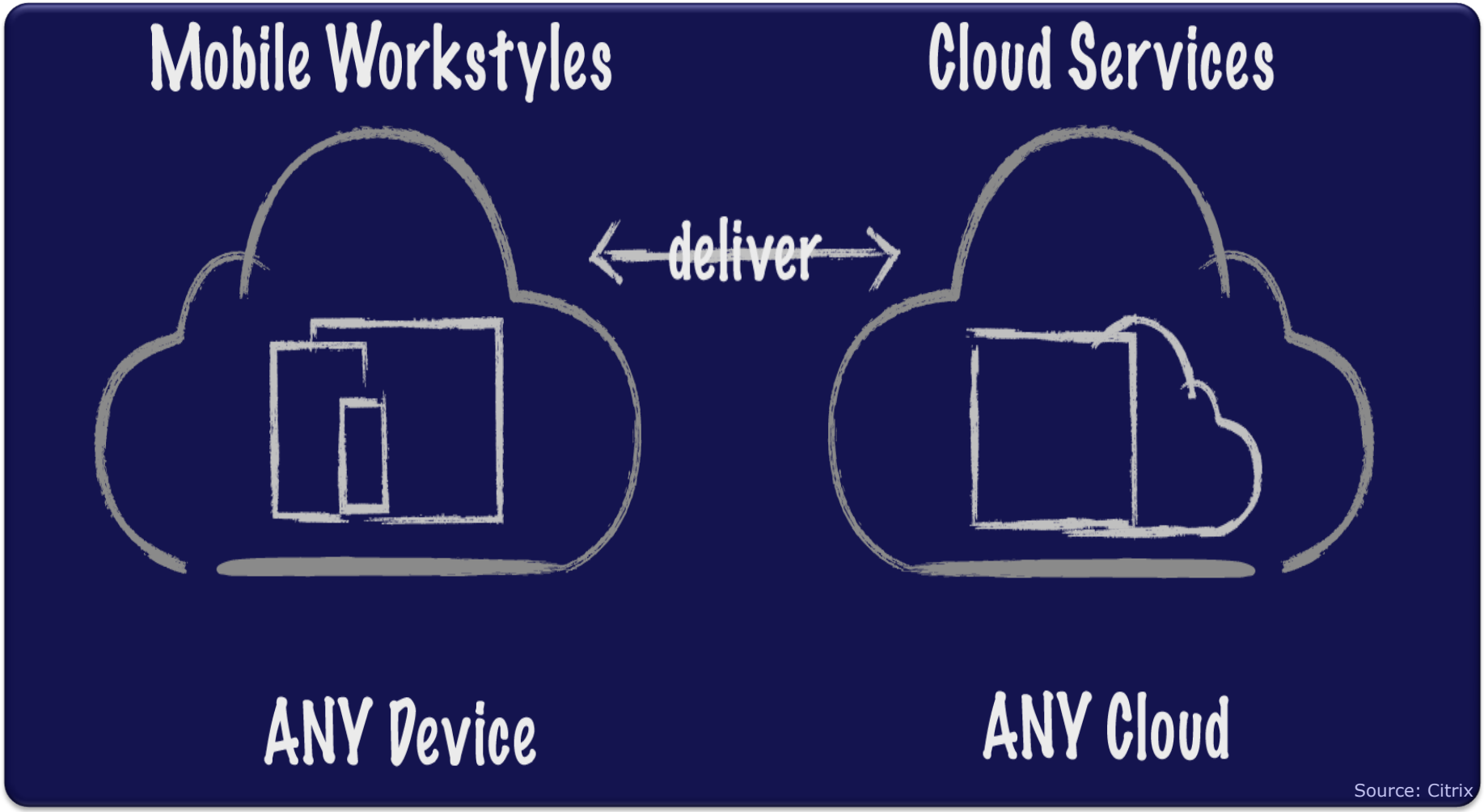


Session Virtualization vs. VDI

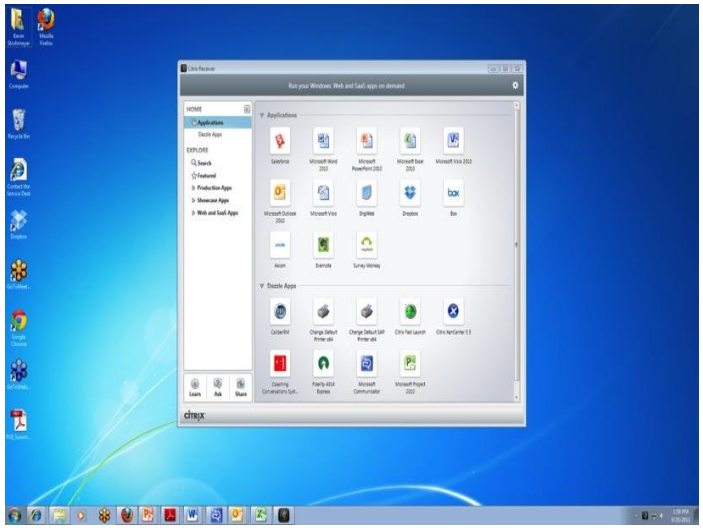
- Previous DV generations – like Session Virtualization (formerly Terminal Services) – only deliver one or more bundled applications presented in a desktop window, not a true desktop.



The concept of VDI

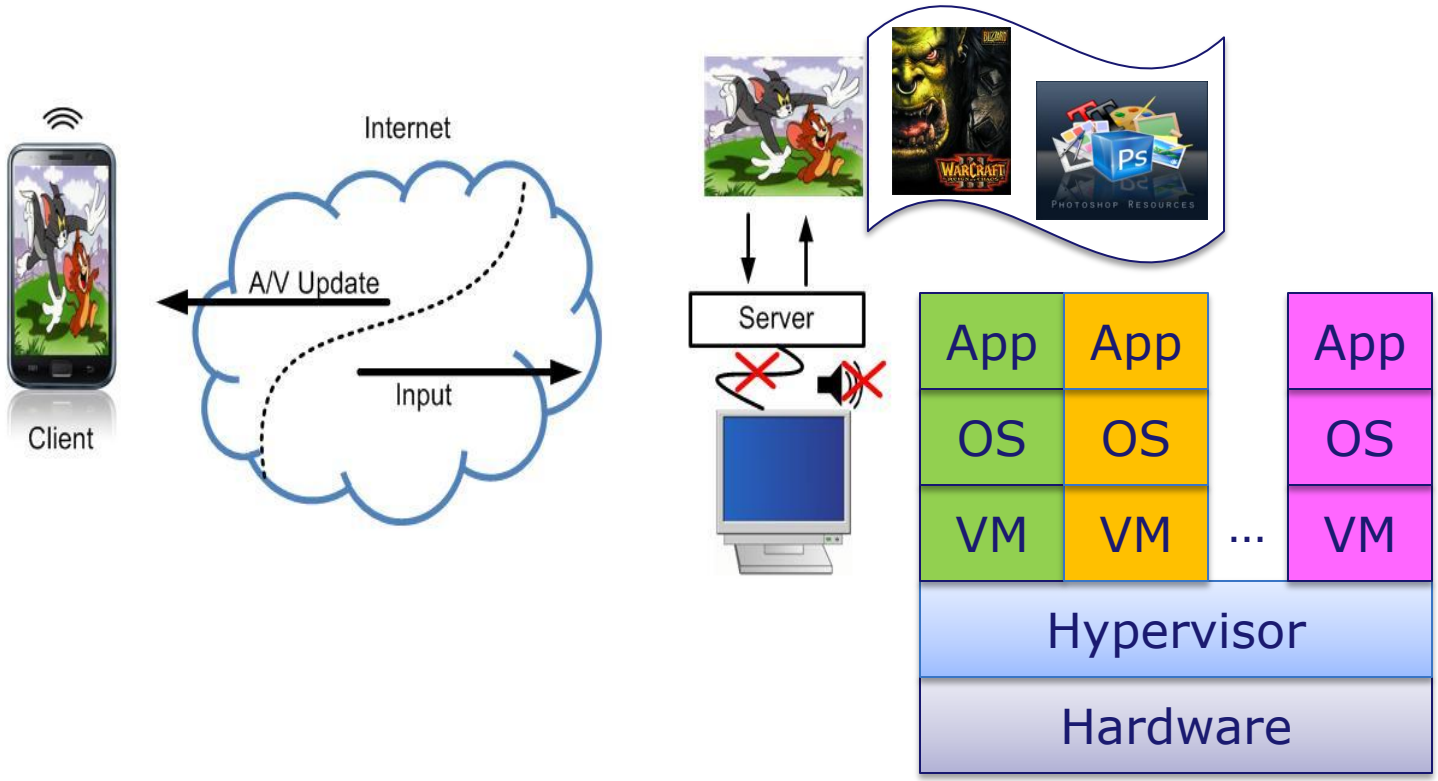


Desktop as a Service



Source: Citrix

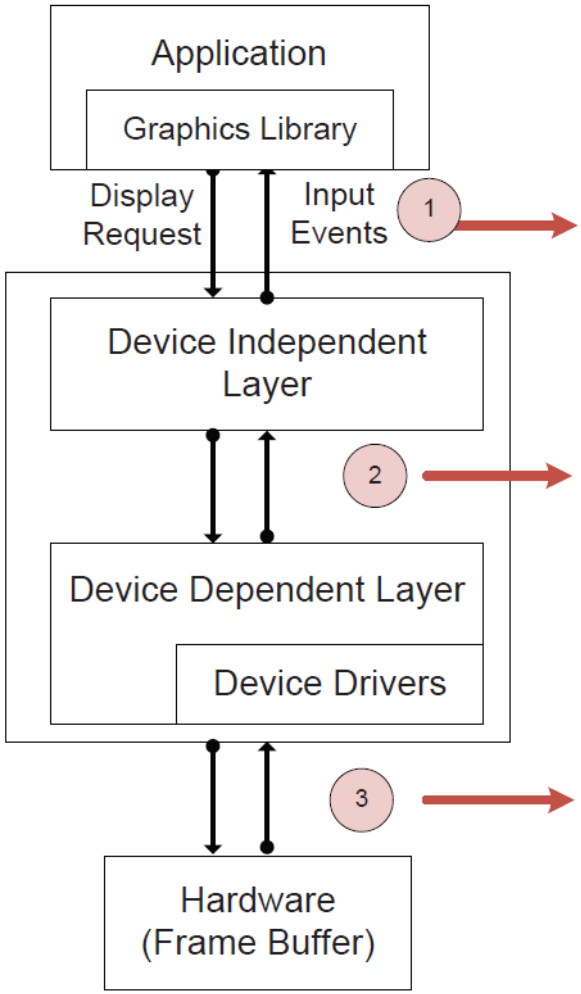
Key Technology of mVDI



❖ Audio/Video (A/V) output from server would not be delivered to locally attached devices (as monitor and speaker),
❖but redirected over the network to client to be displayed.

Key Technology of mVDI

Display Interception Layer



Remote Desktop Protocol (RDP)

Pros: low data bandwidth network

Cons: - execute the windows system on the client results.
- require continuous synchronization over the network.

THINC

Pros: efficient network usage

Cons: - translation is difficult and depends on client's graphic hardware.
- The translator must be increased set of commands supporting 3D and high definition quality video.

VNC

Pros: - deliver the graphical screen (raw data) to the client → do not execute heavy task for displaying on client side. (suitable to thin client).
- support 3D display easily.

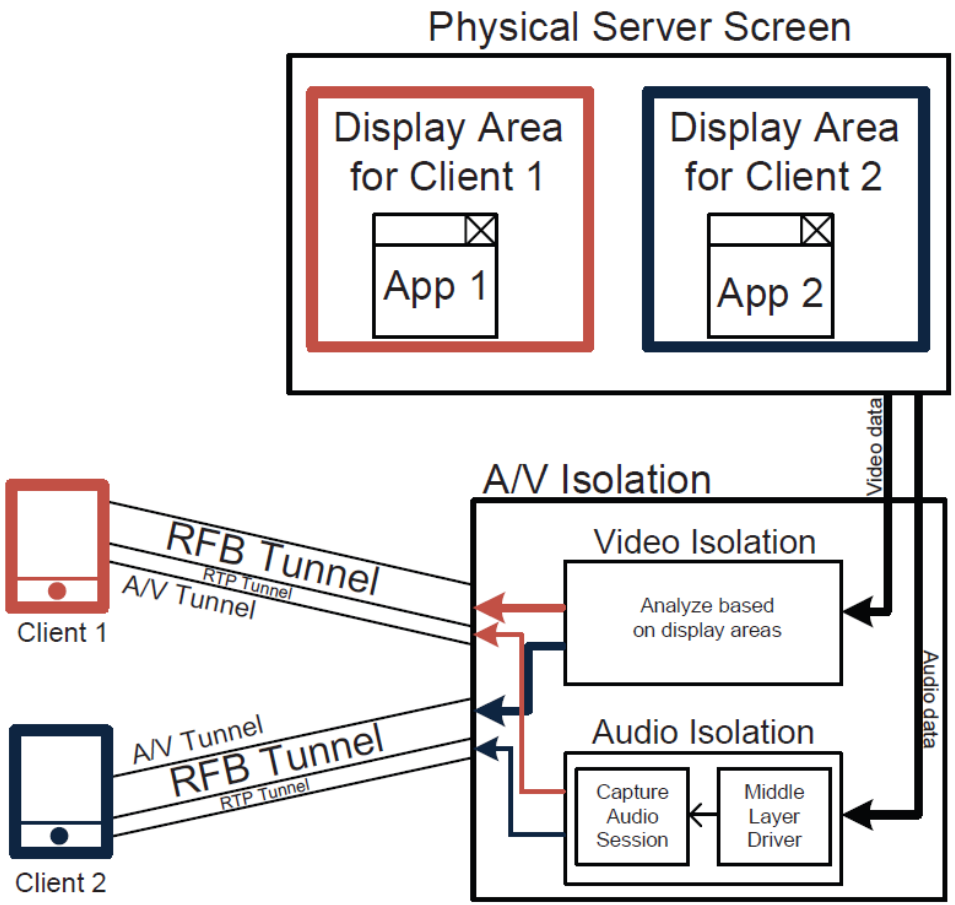
Cons: - require high bandwidth network.

❖ISSUES

Issue 1: M-Virtualization

- With current Hypervisor (VMWare, VirtualBox…), one server (Intel Pentium Dual-Core CPU E5500 2.80GHz) can only allow 2 VMs (Windows OS) to run simultaneously.
  - Too expensive to provide mobile service (each mobile device controls one VM)
- Current VNC only supports a single user in remote accessing to a single physical server for single service.
- Not support audio properly.

Solution1: Platform Isolation Technology



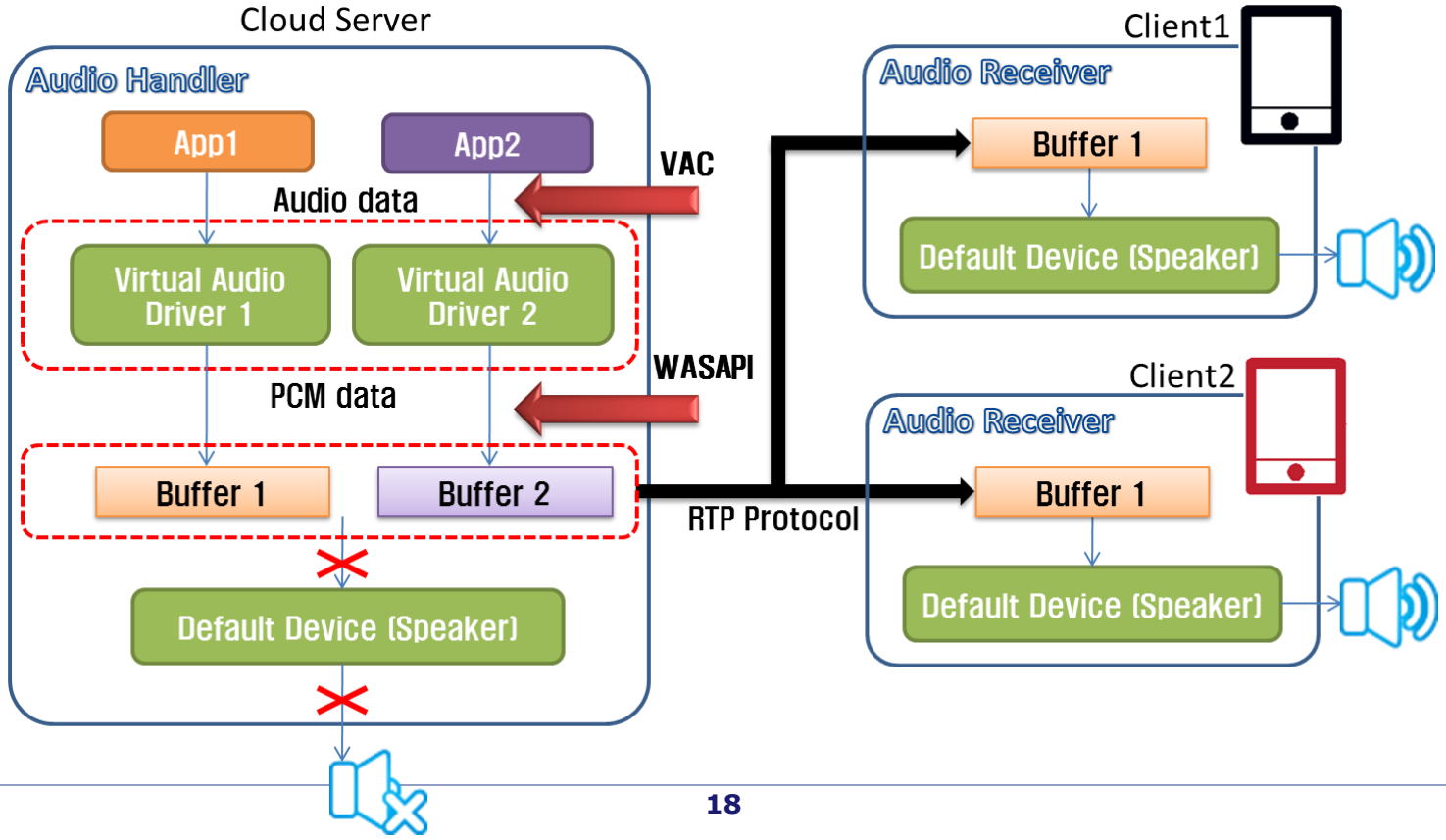
- Scenario:
  - 2 clients access to one server by using VNC, execute App1 and App2 respectively.
- We call this technology is A/V isolation because
  - this approach intercepts at layer (3), video and audio of all applications are normally mixed to one data stream.
  - So, it needs to be separated before streaming to user.
- Consists of:
  - Video isolation technology
  - Audio isolation technology

Video isolation technology

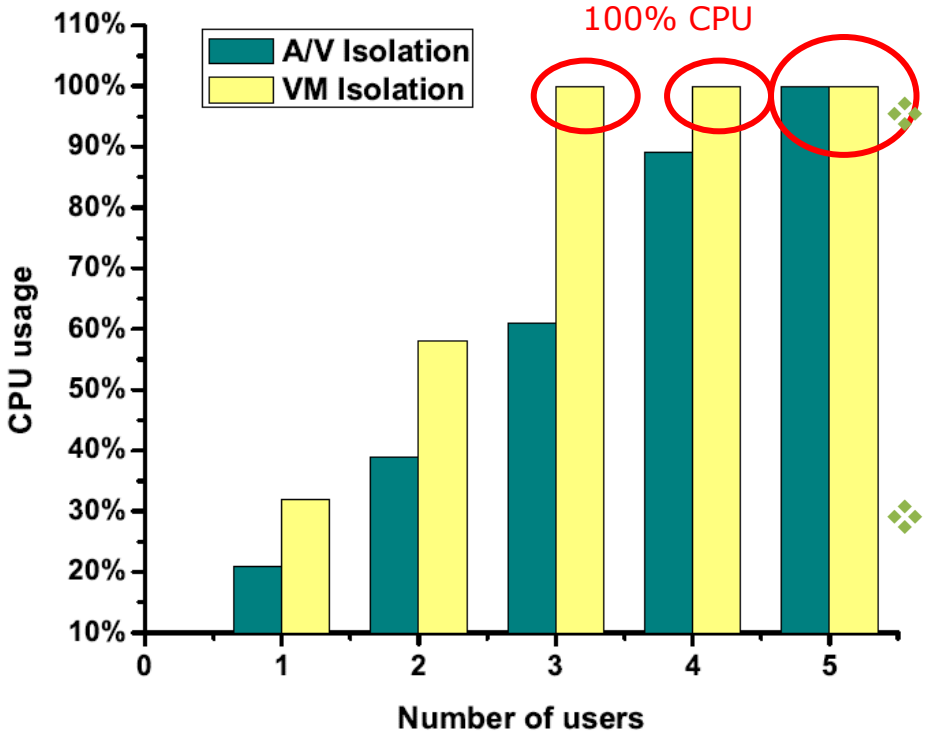
- In order to isolate the graphical output of one user session from others, the server:
  - **assigns** a non-overlapping rectangle area for each user session.
  - **Extracts** the specific user session from the coordinates of the corresponding rectangle.
- Then using RFB protocol to deliver to corresponding users.

Audio Isolation Technology

- **Audio Isolation**
  - VAC (Virtual Audio Cable) : Using Virtual Audio Driver to store data buffer
  - WASAPI(Windows Audio Session API) : Hook audio buffer from VAC.

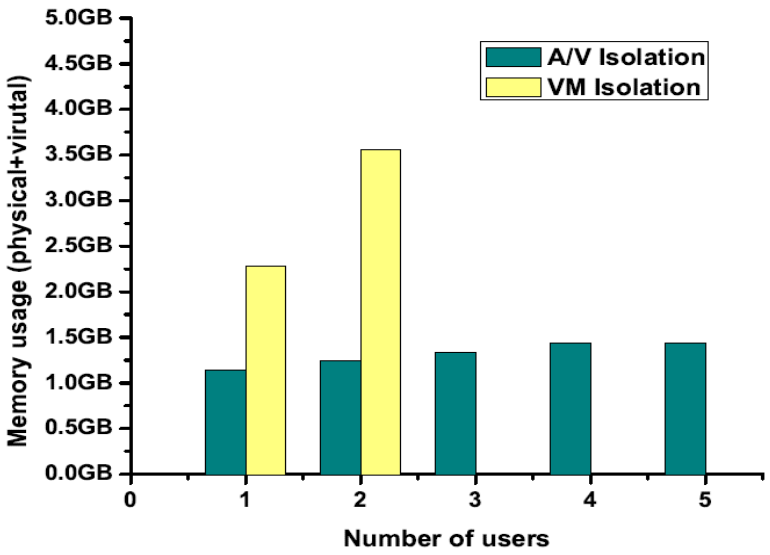


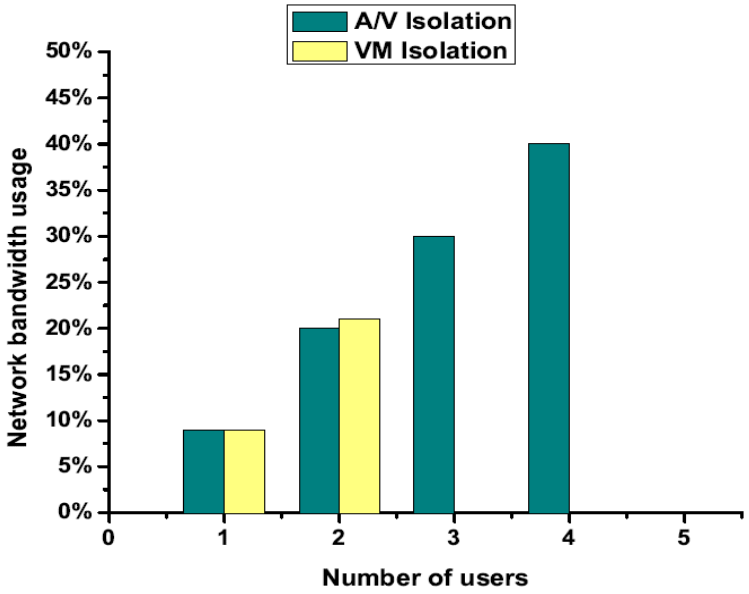
Performance Result

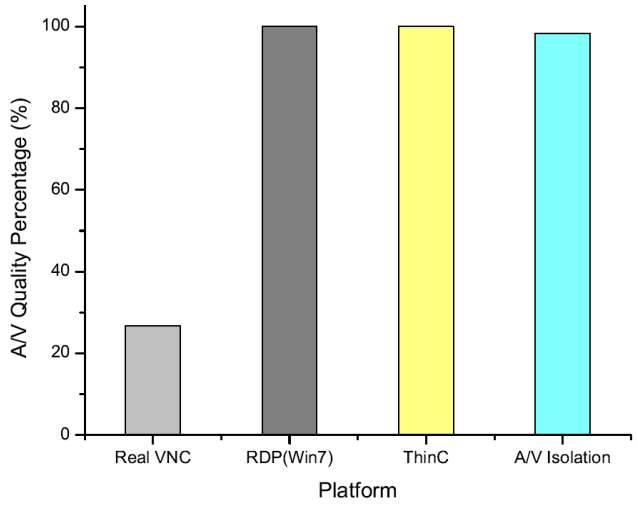


- The result shows that our approach (A/V isolation) outperforms VM isolation.
  - VM isolation is method that supports multi-sessions by creating multiple VMs on single physical server by using VMWare tool.
- With the same hardware condition:
  - VM isolation supports maximum 2 clients while our approach can support maximum 4 clients (based on CPU usage)

Performance Result







Challenges

- How to increase number of mobile devices which can be provided by one cloud server?
- How to schedule in cloud server to guarantee Real-Time and SLA of each mobile device?

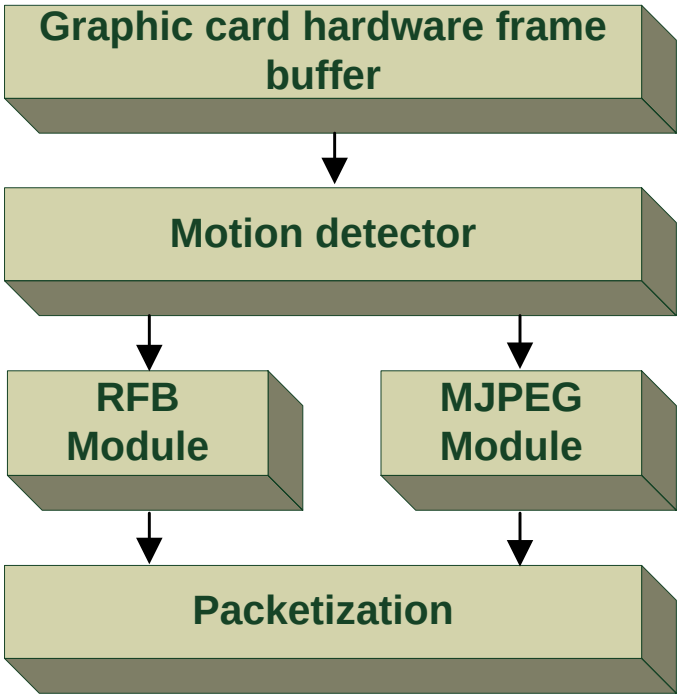
Issue 2a: Network Consumption

- Application can be categorized into:
  - Slow motion (Website)
  - High motion (Video)
  - Network consumption

Scenario	Metrics	VNC-Tight	VNC-Auto	H.264-Streaming
Website	Bandwidth	58KBps	370KBps	200KBps
	Client CPU	1.3%	2.3%	50%
Video(640*480)	bandwidth	1.69MBps	5.19MBps	800KBps
	Client CPU	7%	5.7%	54%
	QoE	Bad	Good	Good

Solution 2a: MJPEG Encoding

- Motion Detector : Video Encoding Algorithm
  - Low Motion : RFB (raw data)
  - High Motion : MJPEG encoding



Issue 2b: CPU Consumption

- CPU consumption for MJPEG encoding in cloud server.

Table 5: Basic Information of Server in experiment

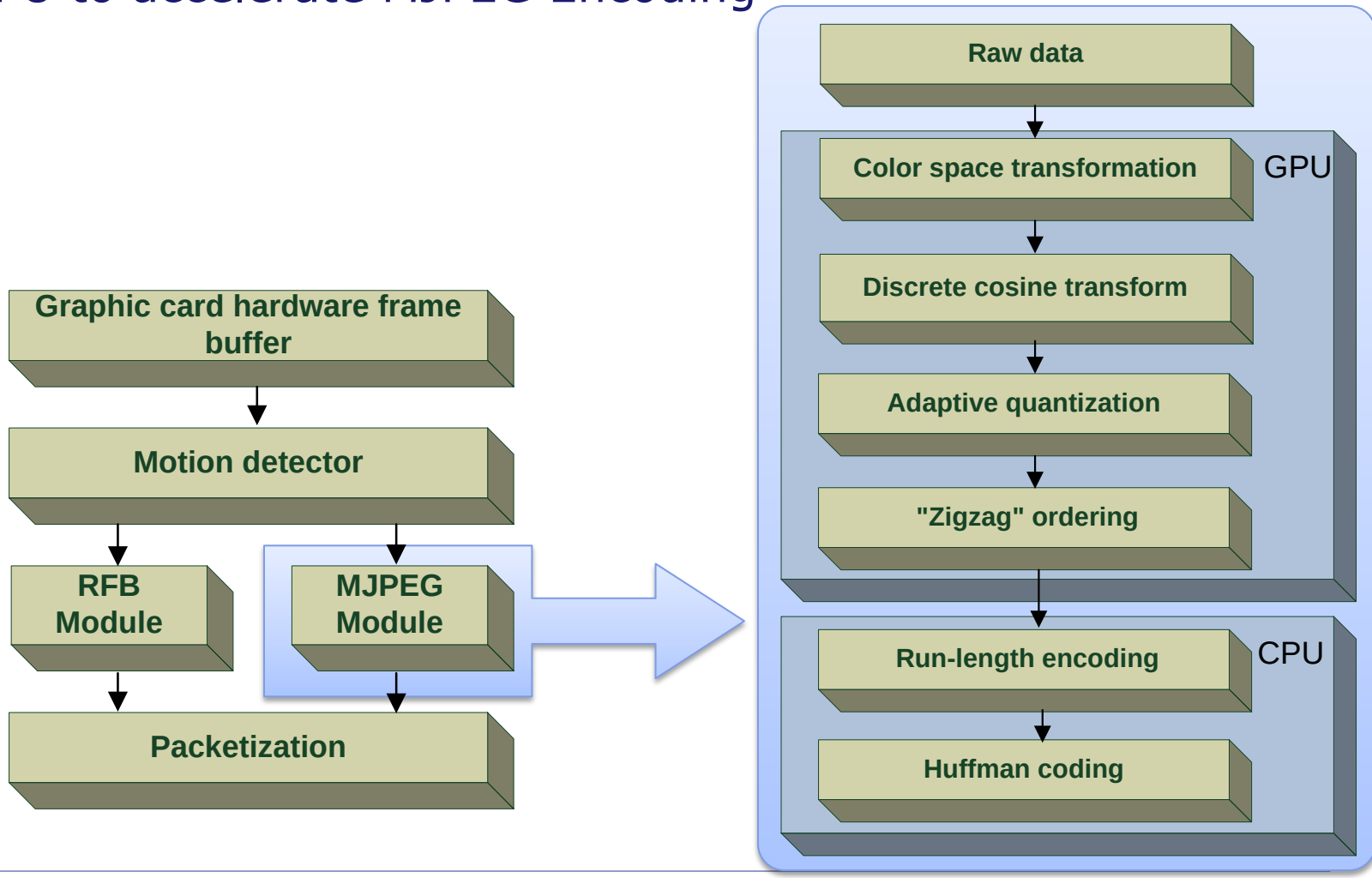
Processor	Inter Xeon CPU X3430 @ 2.40GHz, 2.39GHz
Graphic Card CUDA Cores Memory Size Total	NVIDIA Quadro FX 3800 192 1 GB GDDR3
Installed memory(RAM)	8.00GB
Operation System	Windows Server 2008 R2 Standard 64 bit

Table 6: The CPU consumption - Size relationship (MJPEG module)

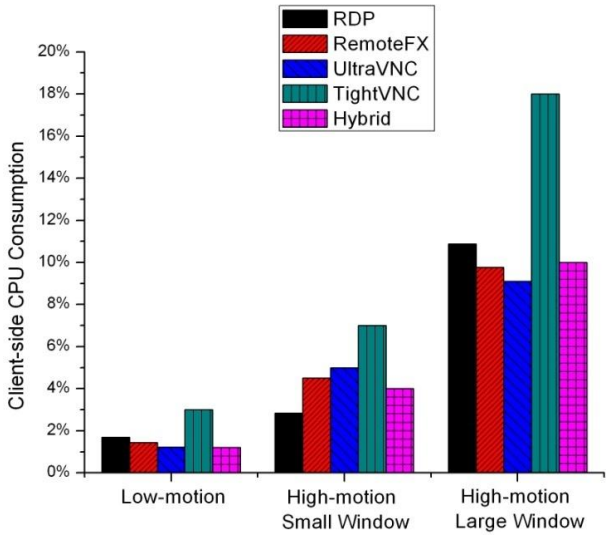
Size of Movie Clips(Pixels)	Server CPU Consumption(%)	Server CPU idle rate(%)
No data compression(VNC module)	2	98
120×80	6	94
120×120	9	91
160×120	11	89
240×160	16	84
240×240	23	77
320×240	26	74
320×320	30	70
480×320	38	62
480×480	44	56
640×480	55	45
640×640	60	40
720×640	62	38
720×720	62	38

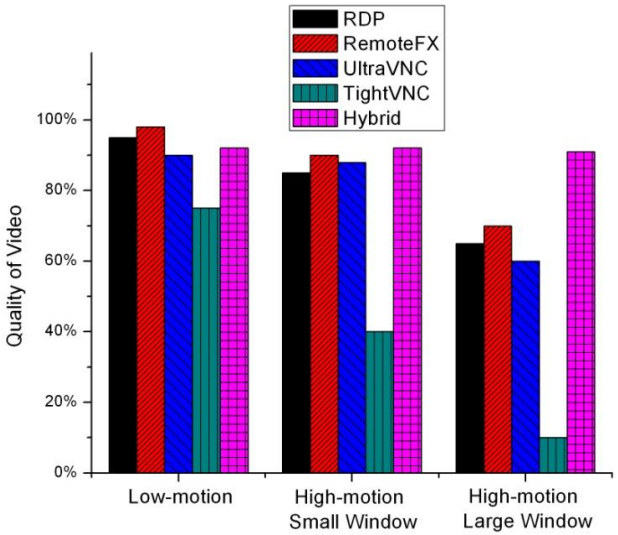
Solution 2b: GPU Acceleration

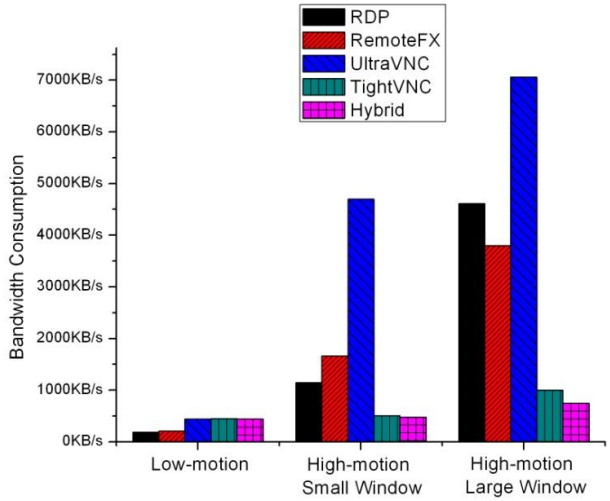
- Use GPU to accelerate MJPEG Encoding



Performance







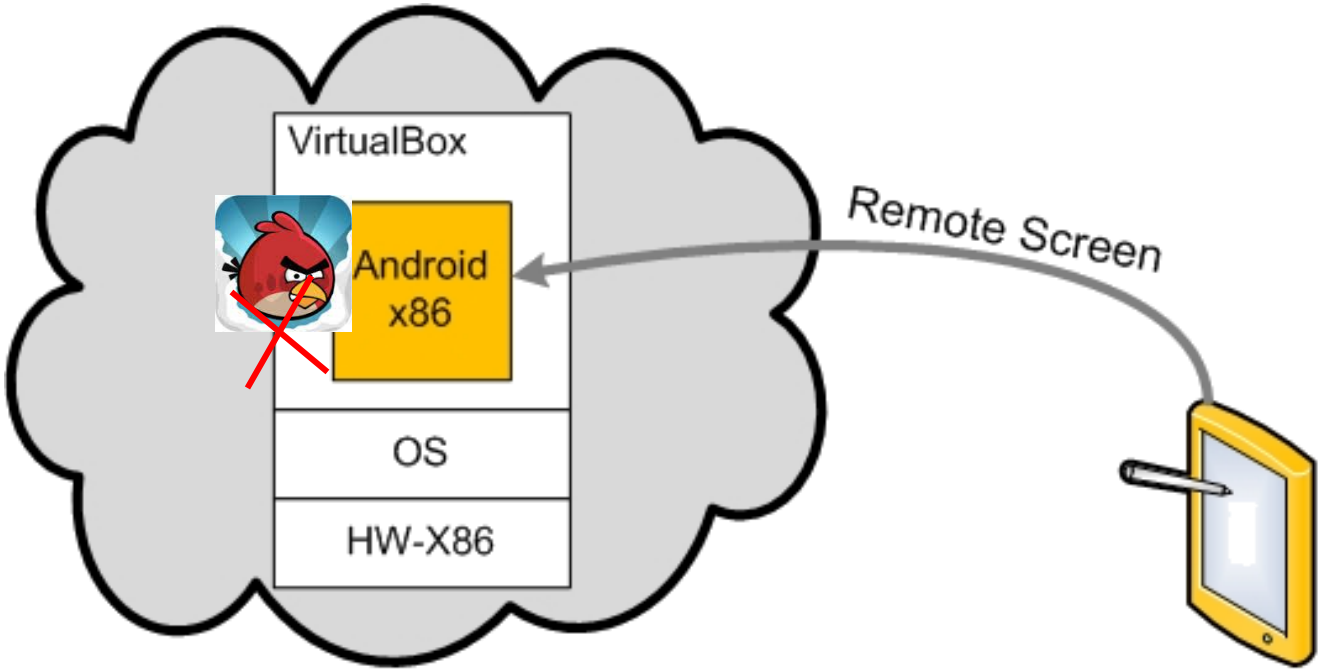
- Hybrid Display Protocol(Hybrid)
  - → Client-side CPU Consumption : Less CPU
  - → Quality of Video : Better Quality of Video
  - → Bandwidth Consumption : Less Bandwidth

Challenges

- How to detect content efficiently (picture vs. movie; slow motion movie vs. high motion movie)?
- How to establish parallel GPUs to accelerate computation?

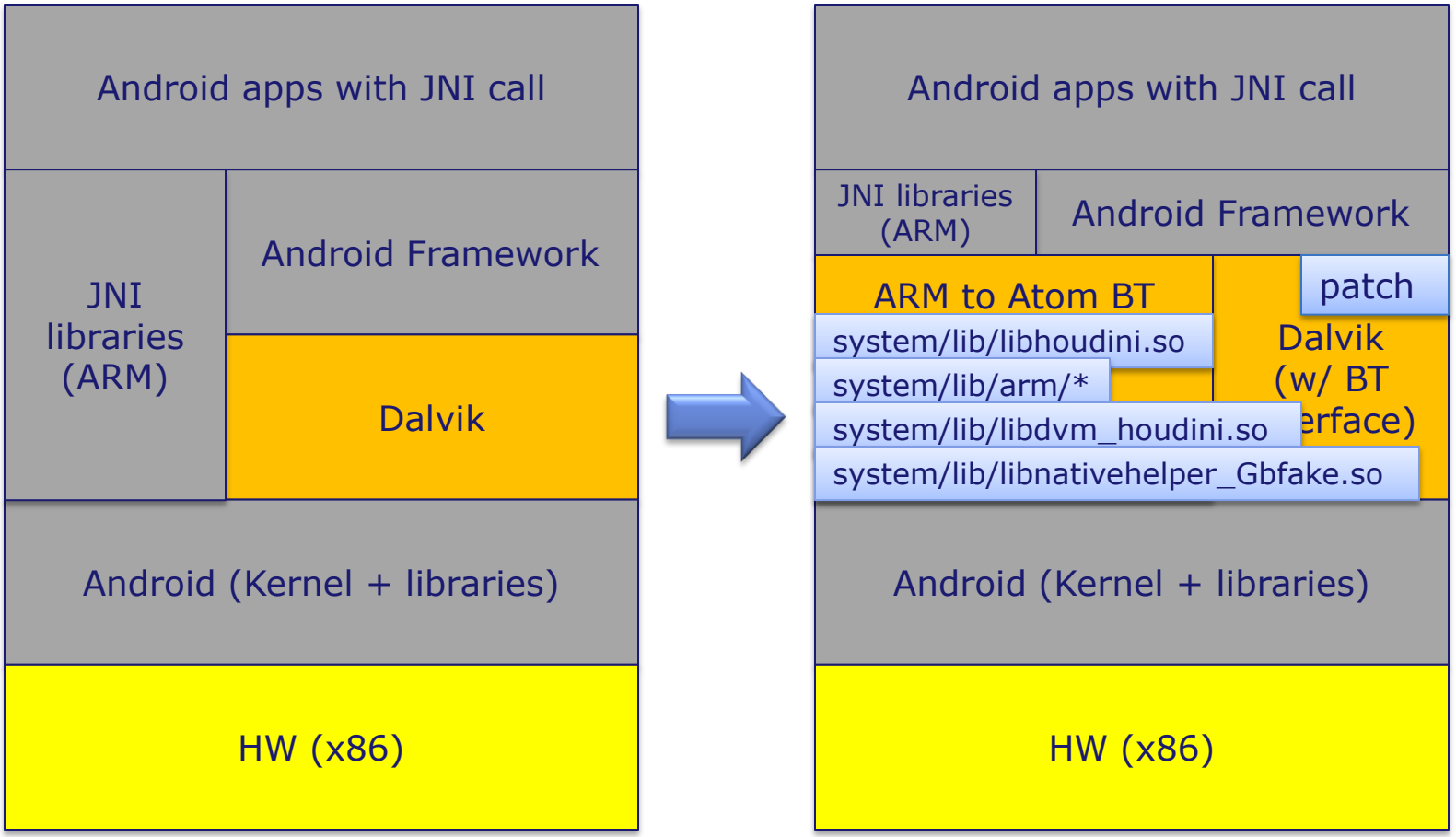
Issue 3a: NDK Apps Incompatibility

- mVDI integrated Androidx86
  - Android app are developed by:
    - **Java** language and be compiled to run on the platform's Dalvik runtime environment
    - **native** libraries coded in C or C++ within their Android applications.
- Incompatibility with NDK apps compiled only for ARM processors.



Solution 3a: ARM-x86 Binary Translation

- Problem with Android running on x86 : some applications (most games) run native ARM code.

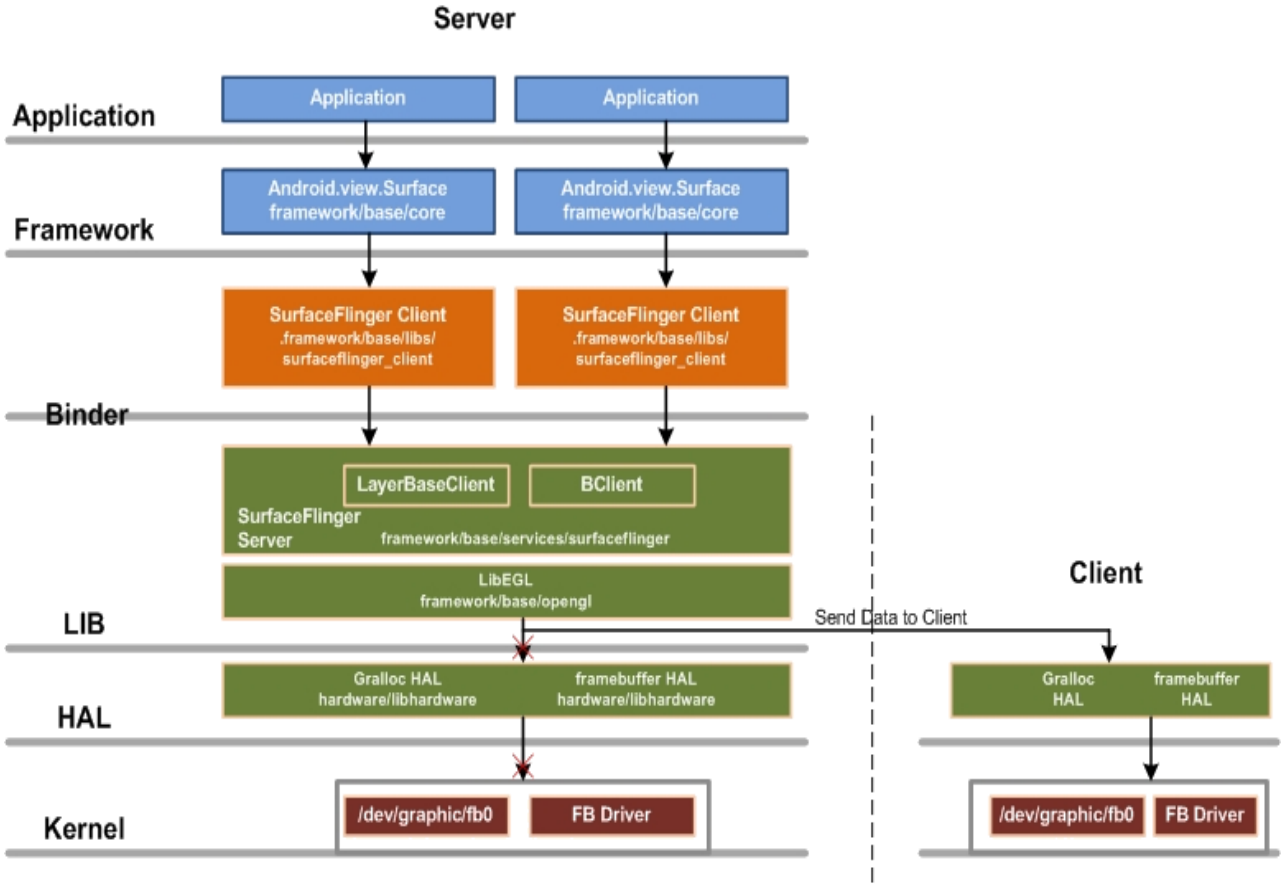


Issue 3b: OpenGL Performance

- Now, Angry Bird can run on Androidx86 but it's not very good – and very far from the performance as running on the real mobile device,
  - due to the GPU virtualization in Hypervisor (Virtual Box).

Solution 3b: OpenGL Layer Hooking

- We intercept at OpenGL library call, and send data to client for processing.



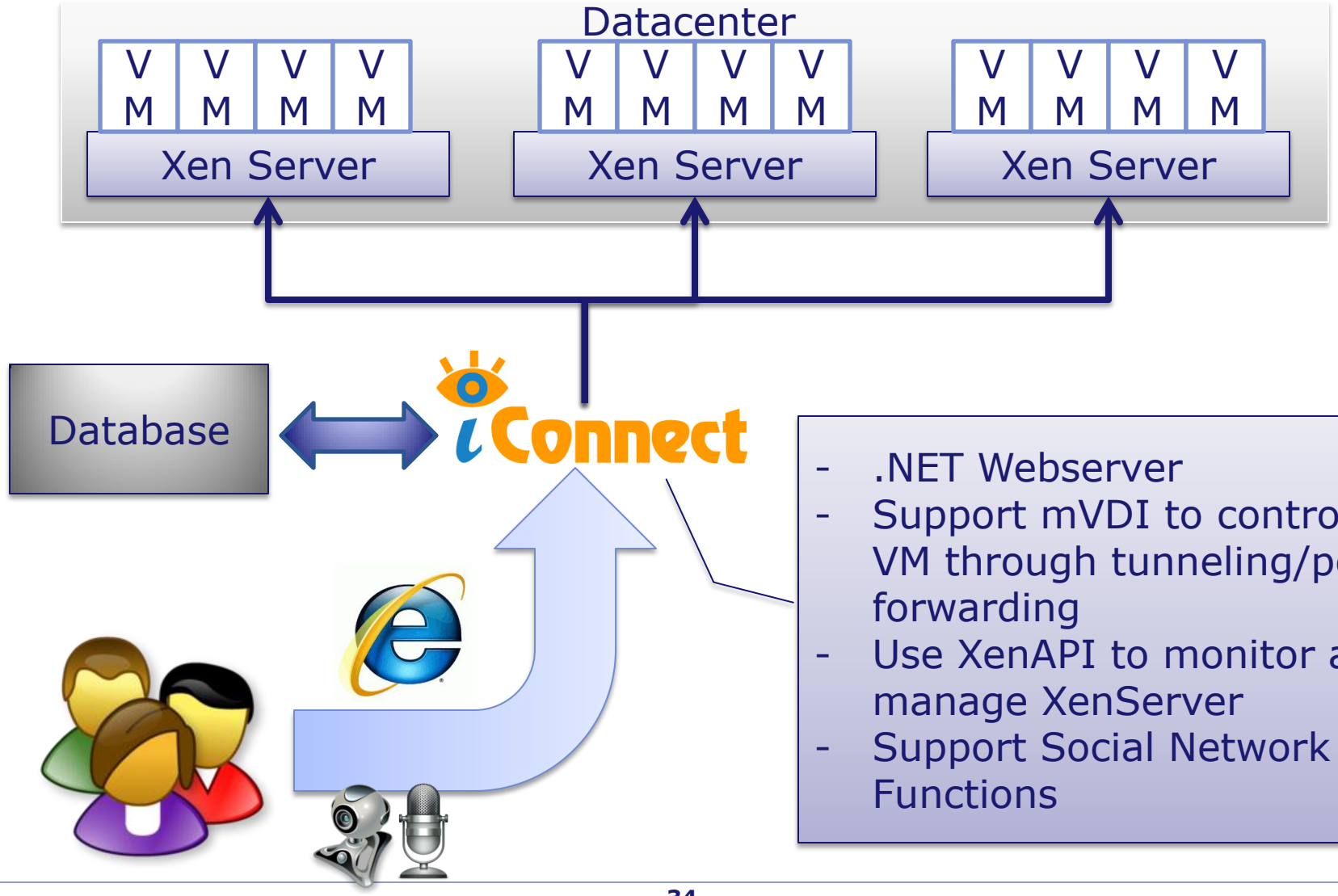
Challenges

- How to reduce data transmission from mobile to cloud server, and vice versa?
- How to decide which parts of application should be done in cloud server, which parts should be done in mobile device?

Issue 4: Web-based mVDI

- Current VDI/mVDI is implemented by application. It limits the seamless and mobility of mobile devices.
  - Web-based mVDI should be implemented, any mobile device can accesses mVDI easily.

Solution 4: iConnect



Challenges

- HTML5:
  - Support multimedia, hardware integration, user interactions, data storage, network enhancement.
- How to use HTML5 to implement mVDI?
- How to integrate Social Network into mVDI?
- How to utilize mobile's resource (sensor, camera, GPS…)?

Summary

- Issue:
  - M-Virtualization
  - Network Consumption
  - CPU Consumption
  - NDK Apps Incompatibility
  - Enhance OpenGL performance
  - Web-based mVDI
- Solution:
  - Video/Audio Isolation Technology
  - MJPEG Encoding for High Motion
  - GPU Acceleration
  - ARM-x86 Binary Translation
  - OpenGL Layer Hooking
  - iConnect

Summary

- Challenges:
  - How to increase number of mobile devices which can be provided by one cloud server?
  - How to schedule in cloud server to guarantee Real-Time and SLA of each mobile device?
  - How to detect content efficiently (picture vs. movie; slow motion movie vs. high motion movie)?
  - How to establish parallel GPUs to accelerate computation? How to reduce data transmission from mobile to cloud server, and vice versa?
  - How to decide which parts of application should be done in cloud server, which parts should be done in mobile device?
  - How to use HTML5 to implement mVDI?
  - How to integrate Social Network into mVDI?
  - How to utilize mobile's resource (sensor, camera, GPS…)?

THANK YOU !!!