SIMPLE PRACTICE BASED ON THE PLATFORM DESIGN FOR SMART TV & N-SCREEN SERVICES IN OPEN CLOUD ENVIRONMENT

JuByoung Oh[1] and Ohseok Kwon[2]

[1]Koino, Inc., Seoul, Korea (South)
[2]Computer Science Engineering Department, Chungnam National University, Daejeon, Korea (South)

ABSTRACT

The cloud virtualization and N-Screen technologies related to open cloud environment are promoting to change legacy IT service architecture. The technologies are very useful to transform old service architecture to new one. With the technologies, Smart TV and its service architecture can be improved. Smart TV has been discussed as a promising device of Post PC category to handle various user needs by adding computing power to general TV. Smart TV is already commercialized and used in web-surfing, on-demand requests on movies combined with Internet enabled set-top box device. There has been specific approach to increase its usability by adding TV apps for specific Smart TV hardware. However, as Post PC perspective, current Smart TV system and architecture are lack of flexibility and need new paradigm. The architecture should provide office-work friendly environment, cover various OS-dependent users and apps based on Android OS & iOS together, and support legacy IT resources. In this paper, we suggest new platform design to achieve the goal to make Smart TV as Post PC device based on emerging cloud virtualization & N-Screen technologies and develop a simple set to test main functions of the platform.

KEYWORDS

Cloud Virtualization, Smart TV platform, Post PC, N-Screen

1. INTRODUCTION

Smart TV is an advanced form of legacy TV and has been discussed as one of promising devices for Post PC. Up to now, Smart TV is gradually changing its system architecture by adding functions to increase its usage and coverage. However, previous approaches were insufficient because they were lying on the legacy broadcasting paradigm or dependent on hardware. We suggest new platform design to add more flexibility and to cover weak points of the previous systems.

1.1. Legacy Smart TV & its limited architecture

The original concept of Smart TV was started to add functions like Internet and Web2.0 specification to legacy TV and it was believed that it would take the role of PC. [1][2][3] Based on the fundamental Smart TV concept, legacy Smart TV system architecture consists of the server providing contents and applications, set-top box clients for home appliances, and reasonable network devices with Internet connection. Even though it had been improved its system and functions continuously, the independent Smart TV system was requested to upgrade its overall system architecture because of lack of applications, device-oriented set-top-box, inflexible UI inconvenience, etc.

The following Figure 1 shows the legacy architecture of Smart TV system. It consisted of basic network / broadcasting function controlling engine, UI & overall management module for user interface, codec modules for videos, and web-browsing module to read simple documents and pictures (might be limited). The system can process contents of only video and image which are already pre-defined or set as a standard. Legacy Smart TV platform was usually designed on a closed private environment and needed customization for each company. It was hard to add functions and difficult to change its structure. To cover the weakness of the legacy system, several approaches were being introduced. [4][5][6]

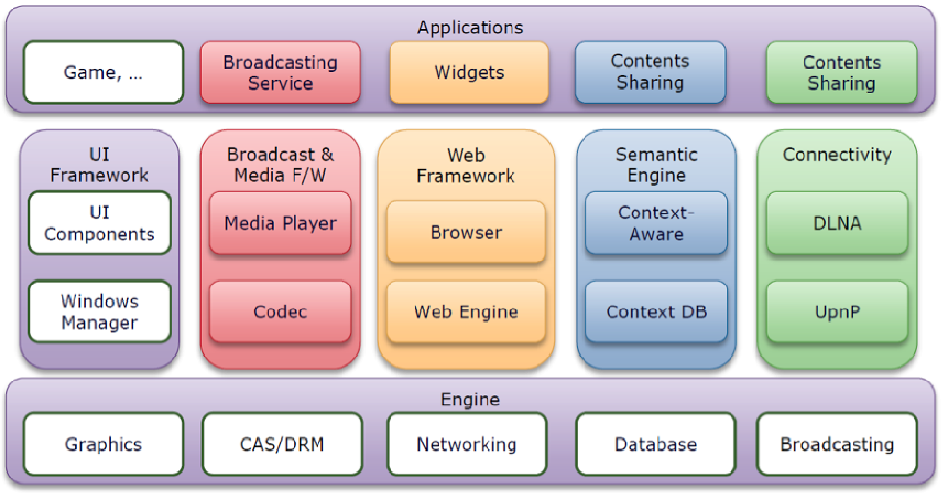


Figure 1. Legacy architecture of Smart TV

1.2 Smart TV's new trend of N-Screen & apps deployment

Recently new types of Smart TV approaches were introduced by renowned IT companies like Apple, Google, and Samsung to overcome weakness and restriction of legacy Smart TV system.[7][8][9] Those were iTV of Apple, Android TV 2.0 of Google, and SmartHUB of Samsung. According to the advent of these brand new system architectures and infrastructures with cloud computing environment, they anticipated that Smart TV would be a core element of killer contents & applications in IT resources with gradual increase of smart devices.

Android OS and iOS smart devices are very common personal devices and also have steadily growing numbers of apps of covering various genres and versatile subjects. Regarding mobile apps, iOS apps were exceeded 700 thousand in 2013 and Android apps were also exceeded 700 thousand at the same time. Each new Smart TV system targeted to be a rich-content Smart TV and to give a strong impact to the industry by providing a lot of apps for customers to feel much more added values compared to that of the legacy Smart TV system of having simple broadcasting capability.

With use-ready apps, contents-transferring cloud platform, and its own brand set-top box, it was tremendous paradigm change of providing rich customer experience and additional side effects compared to legacy Smart TV. Two big software companies' approaches were very similar in that they utilized their own apps and their own smart devices. Unlike two big software companies, Samsung's Smart TV approach was rather TV device oriented approach. Samsung gave additional value to only the buyers of their Smart TVs by providing their apps only working on theirs. Samsung's approach was not a fundamental change but to give a value to its hardware. They have about 1,500 apps for the TV as of year 2013. Samsung's approach is a trend but it is not the main trend at this moment.

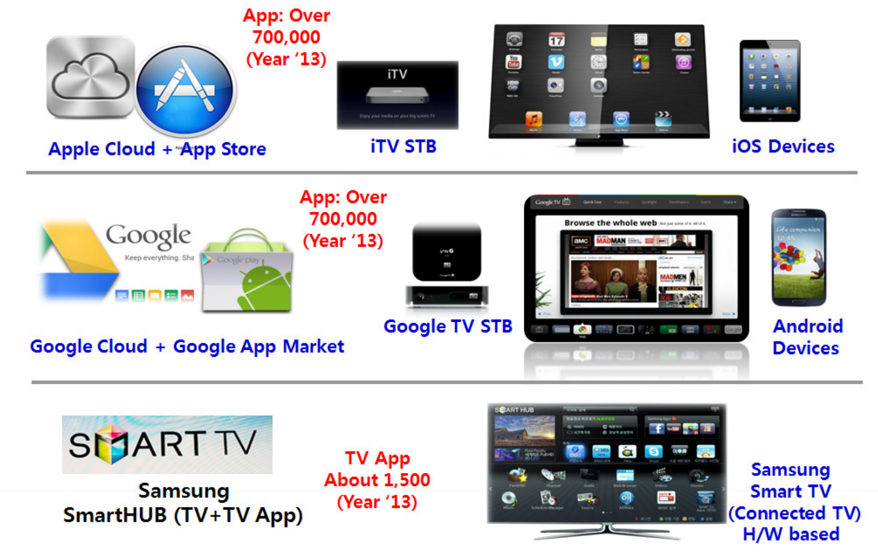

App: Over 700,000 (Year '13)
Apple Cloud + App Store
iTV STB
iOS Devices
App: Over 700,000 (Year '13)
Google Cloud + Google App Market
Google TV STB
Android Devices
SMART TV
Samsung SmartHUB (TV+TV App)
TV App About 1,500 (Year '13)
Samsung Smart TV (Connected TV) H/W based

Figure 2. Current Smart TV approaches: Android TV, Apple TV, and Samsung TV

1.3 Cloud computing reflected on new trendy approaches

Cloud computing is an architecture to provide IT functions as service as like people can use ATM easily even though they have no knowledge on internal solution and used technology. The definition of IEEE is abstracted in one phrase: a paradigm of data stored permanently in a server residing on the network and temporarily in client devices like desktop, tablet, wall-mountable computer, and portable device. Figure 3 shows the concept of cloud computing.

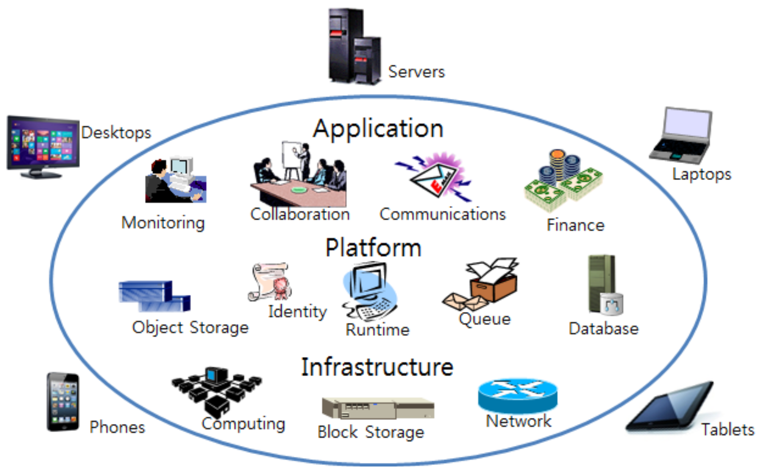


Figure 3. The concept of cloud computing

Comparing to the architecture of independent server and desktop, cloud computing service can reduce the initial purchase cost and provide mobility to users. It is also a good approach to Green

IT by increasing the effectiveness of resource usage. As a client perspective in a concept of N-Screen, it is possible to use multiple devices on the same contents without such limitation of OS and location even though there is cross-platform issue. It is also much safer to keep all the user data to the server, not to his own carry devices.

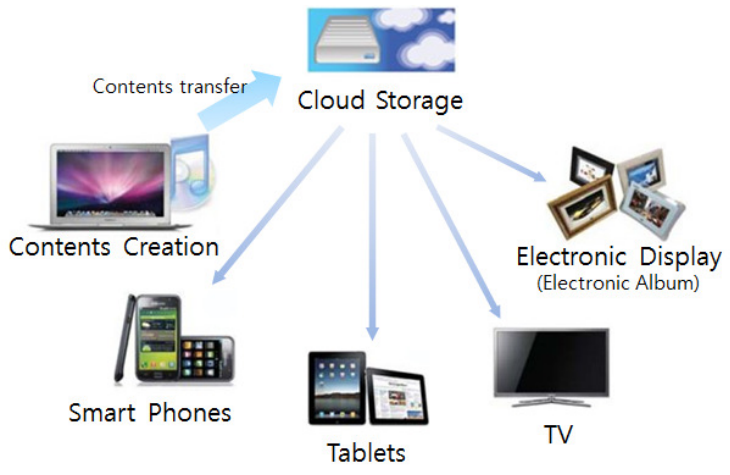


Figure 4. N-Screen concept diagram

By applying the cloud computing technology to Smart TV system, it surely is able to increase the effectiveness of legacy server-client Smart TV architecture. Recently, cloud computing integrated N-Screen infrastructure is getting more popular according to the spread of smart devices and ubiquitous network environment. N-Screen is a good starting point of consideration to apply it to Smart TV platform for user experience enhancement.

As a back-end server side of the Smart TV platform should be definitely cloud based computing. Because Smart TV platform has a lot of apps and contents (videos, etc.), the server system must have capability of effective management. The cloud technology could enhance management effectiveness by sharing resources. The cloud technology also could easily provide flexibility and scalability to the platform. For actual example, Apple has its own cloud space named iCloud and Google has the same kind of cloud space named gCloud.

1.4 Weakness of new approaches: iTV and Android TV

- *Hardware and OS dependency, Limitation of document processing and individual OS*

Though they are very good platforms to apply for Smart TV, each has both weakness and shortcoming. Each solution is based on its own ecosystem and its own specific hardware devices. iTV set-top box is packed iOS device and is able to share content with iPhone, iPad with iCloud, and to communicate with iStore environment. It cannot use Android OS based smart devices and Google's cloud environment. Though iTV can increase easily user experiences with their apps, it is limited to their own apps and cannot use Android's apps. It is the same situation in Android TV. They cannot use iOS smart devices and iStore's apps, too. As a customer perspective, if he has an Android tablet and wants to see iTV, he should buy forcefully iPad.

Both iTV and Android TV have another weak point if a customer wants to extend it to desktop environment. The two solutions are not interoperable with Microsoft applications on desktop PC working environment. Even though there are some alternative apps to cover it, they cannot cover the most widely used desktop document applications like Excel, Word, and PowerPoint.

Furthermore, Android OS and iOS were originally designed for each individual and were not targeting multi-user or multi-processing. It might be a severe limitation when the OS has to work on simultaneous tasks in Smart TV. When we consider cloud desktop virtualization, it might be also reluctant to adopt mobile OS as one of virtualization guests in that lack of resource management and multi-processing. To increase effectiveness, we must consider deploying multi-user OS as a guest. Well-known multi-user operating systems are Microsoft Windows, Linux, and UNIX. Considering the performance, environment, and cost for it, it should be an effective approach to deploy Windows OS or Linux on x86 hardware.

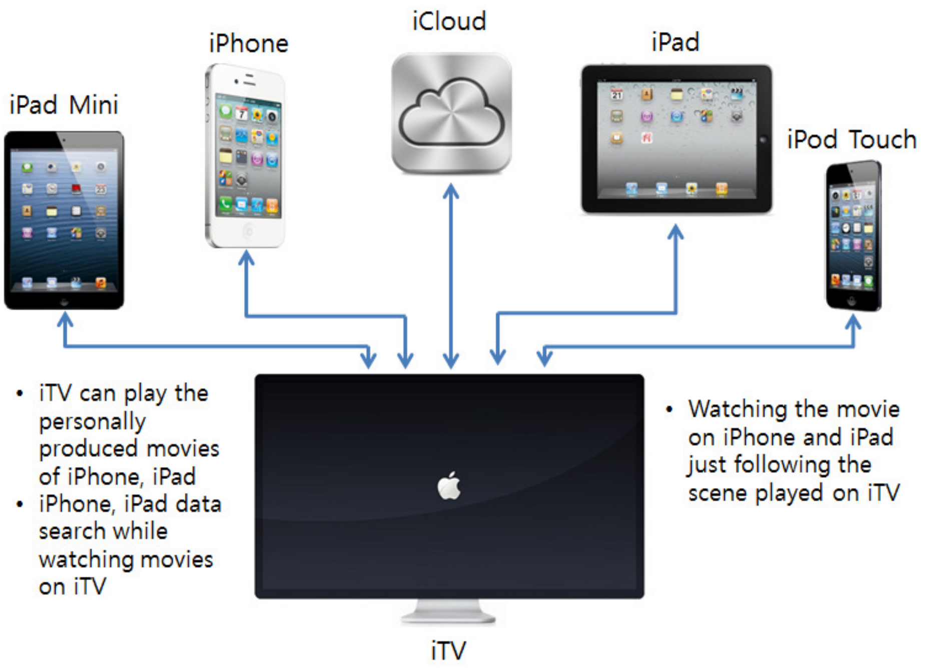

iCloud
iPhone
iPad
iPad Mini
iPod Touch
• iTV can play the personally produced movies of iPhone, iPad
• iPhone, iPad data search while watching movies on iTV
• Watching the movie on iPhone and iPad just following the scene played on iTV
iTV

Figure 5. Apple's Smart TV - iTV

2. CONSIDERATIONS ON FLEXIBLE SMART TV ARCHITECTURE

Upon the introduction of Android TV and iTV owing to the combination of cloud and smart device technology, Smart TV's capability and user experiences are improved much further comparing that of traditional legacy Smart TV system. However, the approaches have weakness of OS & hardware dependency, lack of document work functionality, and limitation of cross-platform cooperative work functionality. To overcome weaknesses and shortcomings, we considered to deploy the cloud virtualization technology for getting rid of OS barriers, to design multi-purpose VDI (Virtual Desktop Infrastructure) protocol for effectiveness, and N-Screen technology for user accessibility.

2.1 Considerations on Cloud Virtualization technology

Cloud computing itself is already a common terminology to people and is usually considered as a representative example of paradigm change. Recently there are a few approaches to deploy VDI (Virtual Desktop Infrastructure) with cloud computing platform. [10][11][12][13] VDI concept is to integrate desktop computers into the cloud platform. When they apply VDI to cloud computing system, they are able to not only use resource effectively but also utilize zero client or thin client as a client device. It also gives big advantages to operate and manage desktops and to keep desktop data in secure. There are several trials to upgrade effectiveness of using virtualization resources: CPU, memory, HDD space, and processes. To apply VDI to cloud computing environment, there needs a hypervisor which can control virtualization guest OS. Most well-known approaches in OSS (Open Source Software) are OpenXen and KVM. [14][15]

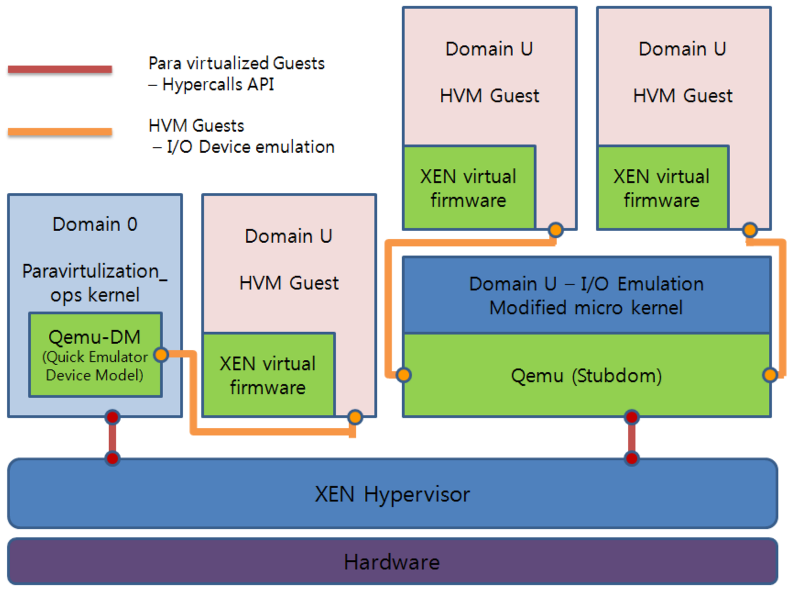


Figure 6. OpenXen Architecture and modules

OpenXen is an open source version of commercial Xen developed by Citrix Systems. OpenXen hypervisor should be installed on Linux system called Domain0 can control other guest OS called Domain1 ~ DomainU. The OpenXen hypervisor can control hardware directly even though it is installed on Linux system. However, sudden type of guest OS which should control BiOS directly like Microsoft's windows OS needs binary emulation called HVM (Hardware Virtual Machine). Qemu (Quick Emulator) is widely known binary translator for HVM and it has many variations according to its functional differences. [16][17]

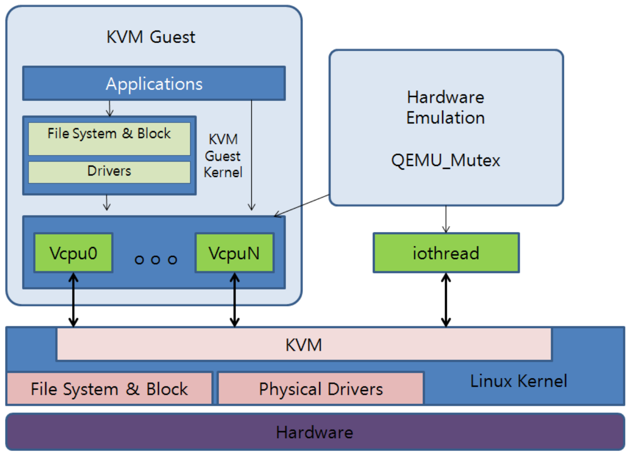


Figure 7. KVM hypervisor Architecture

KVM is open source hypervisor developed by Redhat, Inc. It is hybrid type hypervisor and controls guest OS at the same position level as that of Linux kernel. It also needs Qemu variations to install Microsoft windows system. KVM has rather simple architecture by integrating advantages of traditional hypervisor and Kernel and is strong to handle multimedia and VDI.

2.2 Considerations on Multi-purpose VDI Protocol Design

Desktop sharing and control are main aspects of functions to enable virtualization. Virtual desktop running on a server can send screens to client device. There are several protocols used in desktop sharing. RDP, ICA, and RFB are widely deployed protocols. RDP is Microsoft's protocol and used in Microsoft RDP server and client. [18] ICA protocol is used in Citrix Systems'. [19] The two protocols are proprietary used in their own products and not published their structure in public. Otherwise, RFB protocol is opened its structure to the public. Many applications and products are using RFB as default protocol and it is also widely used. [20] The protocols stated above are 1st generation protocols and have limitation to implement desktop sharing.

Recently Redhat, Inc. takes an important role in Open Source Software by providing Redhat Linux OS, substituting legacy UNIX system. They published KVM hypervisor for cloud virtualization and desktop VDI client for effective VDI. Redhat's streaming protocol is classified as 2nd generation protocol, running on KVM hypervisor, is showing good performance with seamless playing on multimedia contents. However it is not supporting N-Screen devices and only working on its own cloud virtualization platform. Besides, current KVM hypervisor supports just one session between VDI client & server and is so limited.

2.3 Overall considerations on new platform concept

We consider new Smart TV platform having advanced system architecture. Firstly, it should have capability to connect numbers of N-Screen devices into one virtual OS guest residing on cloud virtualization server and users can see same screen via various N-Screen devices. Secondly, it should be able to use 2nd generation VDI protocols to support both document mode and streaming mode to cover office and individual requirements. It should also support smart device apps running with N-Screen features. Thirdly, the platform should be able to use N-Screen user's device as VDI client without client hardware dependency. There should be no request to prepare additional device for Smart TV. To use apps, if he who uses Android device, he could play Android apps on his Android device as he did. For iOS device user, he also could run iOS apps as he did. The user might get computing resources from any guests (i.e.: Windows, Linux, Android, etc.) of the virtualization server except the case of the closed OS like iOS which is not be able to be invited as a guest to the virtualization server.

Considering as a VDI client device, the user must utilize legacy desktop PC resources besides Android and iOS devices. With reflecting the features stated above, we design advanced flexible Smart TV platform & architecture to be able to use Smart TV system as Post PC.

3. NEW PLATFORM DESIGN AND ITS ARCHITECTURE

3.1 Functionality and coverage of the platform

To overcome the weakness & shortcoming of previous approaches, reflecting overall considerations on new platform stated above, we suggest flexible architecture concept of designing cooperative document work functionality, N-Screen capability, and multi-user resource sharing based on existing cloud virtualization and VDI architecture. Following requirements are covered by new platform proposal.

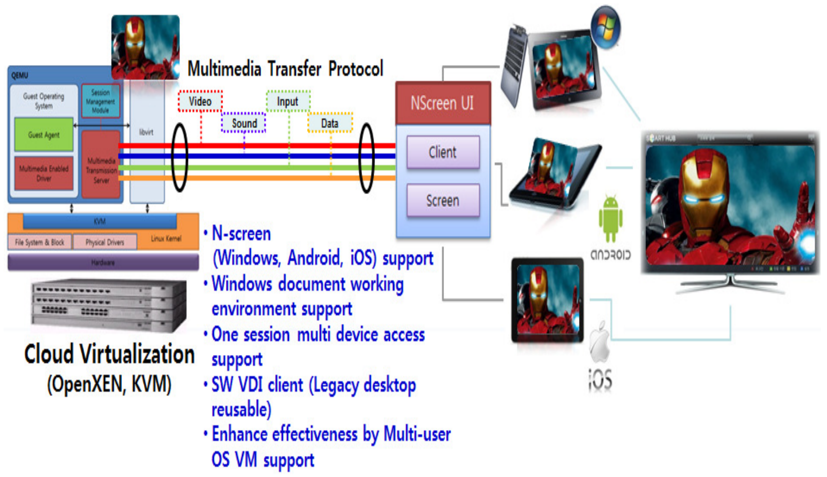

Multimedia Transfer Protocol
QEMU
Guest Operating System
Session Management Module
Guest Agent
Multimedia Enabled Driver
Multimedia Transmission Server
libvirt
KVM
File System & Block
Physical Drivers
Linux Kernel
Hardware
Video
Input
Sound
Data
NScreen UI
Client
Screen
• N-screen (Windows, Android, iOS) support
• Windows document working environment support
• One session multi device access support
• SW VDI client (Legacy desktop reusable)
• Enhance effectiveness by Multi-user OS VM support
Cloud Virtualization
(OpenXEN, KVM)
ANDROID
iOS

Figure 8. Module design diagram of new Smart TV platform

- Real-time multimedia support from VDI server for Smart TV server platform with Multimedia Transfer Protocol featuring 2nd generation VDI protocol
- Virtual Guest (VDI server for Smart TV) based desktop sharing & control performance providing document work functionality (office and individual work environment support)
- VDI S/W clients for N-Screen devices including legacy desktop PCs and thin client
- One server session with multi VDI streams to support group-watch or switching N-Screen devices
- Any network environment support using network tunneling server

3.2 Multimedia support VDI protocol and related module design

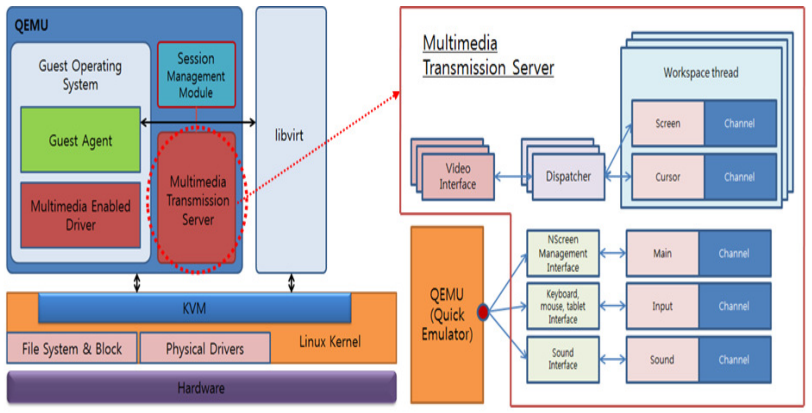


Figure 9. Multimedia Transfer Protocol diagram for the Smart TV Server and the Client

As we acknowledged from the previous approaches, there were cross-platform problems aroused from the provider of OS and hardware specification. Thus, new platform is basically designed to support OSS (Open Source Software) and to have flexibility and rather to be free from license issue. To solve the problem of OS-dependent cross-platform issue, our new Smart TV platform delivers full screen from the server to the client using VDI technology. Multimedia transmission server of the server side could handle this request of transferring steady high-resolution N-Screen delivery on both multimedia contents and document screen. The protocol is capable of delivering distinguished channels to carry screen delivery and access control separately. Its architecture is reducing conflicts and showing good performance especially on multimedia mode. Figure 9 shows that the structure of multimedia transmission server to handling each channel of delivery data and access & control signals.

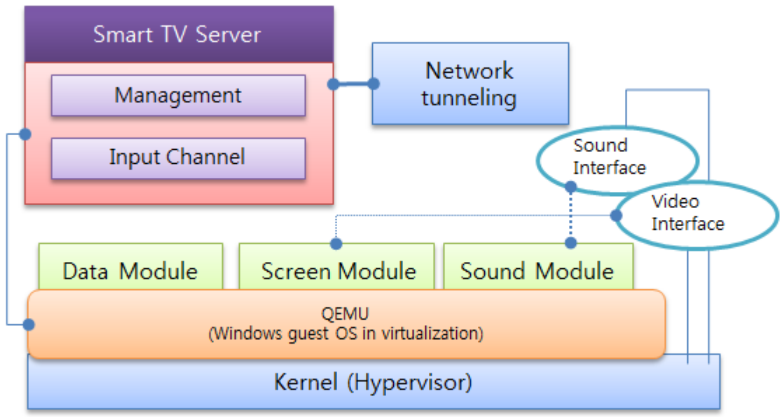


Figure 10. Smart TV server modules and stack structure

Even though the design was based on OSS, it is inevitable to support Microsoft Windows which is most commonly used and the user usually has a bundled license. To support MS Windows as a guest on the virtualization environment, there is a need to deploy the method of RDVH (Remote Desk Virtual machine Host) using Qemu (Quick emulator). Regarding Smart TV client, we consider legacy desktop PCs which are mostly using MS Windows OS and emerging smart devices running under Android OS or iOS.
Smart TV server consisted of several modules: the management module to process the request of the client by channels, the input channel to process keyboard and mouse value, the screen channel to transfer screens, the sound channel to process sounds, the data channel to process data between the server and the client, and networking module to process network handshaking and advanced pier to pier network tunnelling.

The sound module or the screen module needs to communicate with hardware should interface via VGA driver or the sound driver installed in the OS kernel. Figure 10 shows Smart TV server modules and structure of the server channel by channel.

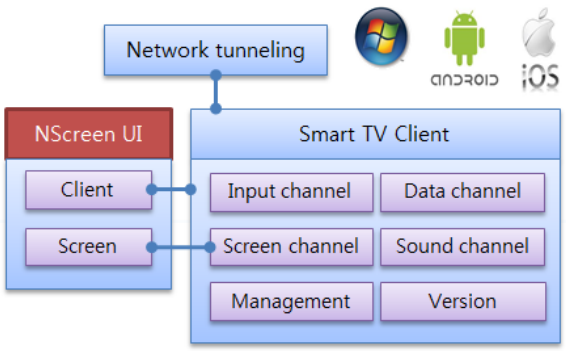


Figure 11. Modules and stack structure of the Client

The Smart TV client consisted of the almost same modules that have to cooperate with those of the server. The client consisted of the input channel, the sound channel, the screen channel, the data channel, and the network channel. N-Screen UI is designed to support N-Screen device and the client management module is to manage sessions and versions. Figure 11 shows the client module structure of channels and functions.

The data transfer protocol design for the server and the client are targeting to transfer each data using multi-channels and adjusting the requests of each other by handshaking traffic, reducing buffering, keeping steady packet transfer, and aiming stable communications. The user friendly protocol should be considered to deploy easily on the server and the client. It is also considered to design adding more smart device or OS in the future.

3.3 Screen Share to watch the same screen to N-Screen devices (Group watch)

The previous Smart TV approaches have not Screen Share function to watch the same screen with N-Screen devices. Following figure shows the concept.

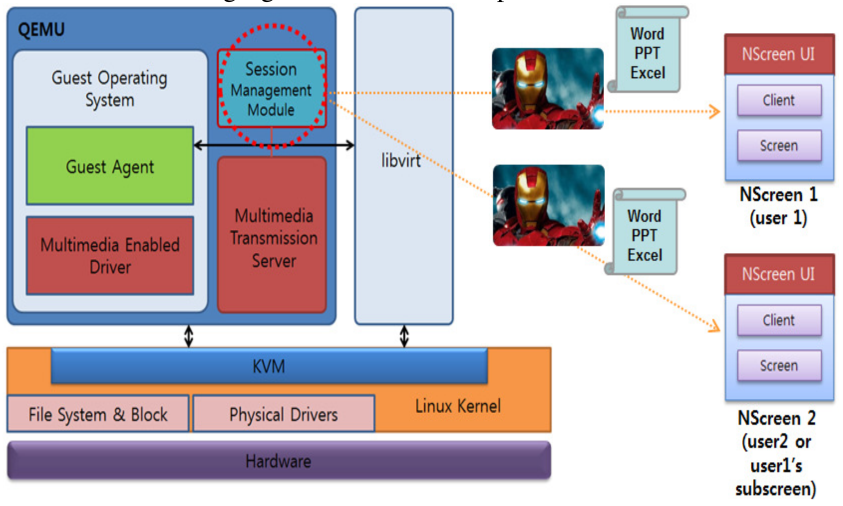


Figure 12. Smart TV to support N-Screen

The Session Management Module of the server could provide the function of Screen Share to send the screen on the main N-Screen display and many other sub screens of smart devices. It could be provided as a form of replicating stream to share the same handle.

3.4 N-Screen extended pack design and network tunnelling module

To utilize the user owned devices, it should include Android and iOS smart devices as basic client owned hardware. VDI client should be carefully designed because mobile OS like Android OS or iOS have limitations on multi process handling, memory handling, etc. Figure 13 shows Android modular stack architecture and virtual machine structure.

Considering N-Screen devices, even though mainly Android OS, iOS should be considered, Windows OS installed desktop PC should be included as a good computing resource in customer side. Additionally new mobile OS might be added on the list in the future. It depends on the market needs and it should be also considered. Moreover, it should be reminded that the mobile OS periodically published new major and minor version much more frequently than that of desktop PC. To support N-Screen devices, it is also considered additional co-working features and functions: file transfer, system information, process information, screen capture, etc.

For seamless network connection, the network tunnelling server is inevitable for the clients residing on local private network. The network tunnelling server should be designed to process high stress of simultaneous session establishment requests. The server also should be considered active-active or active-standby backup server to support high availability.

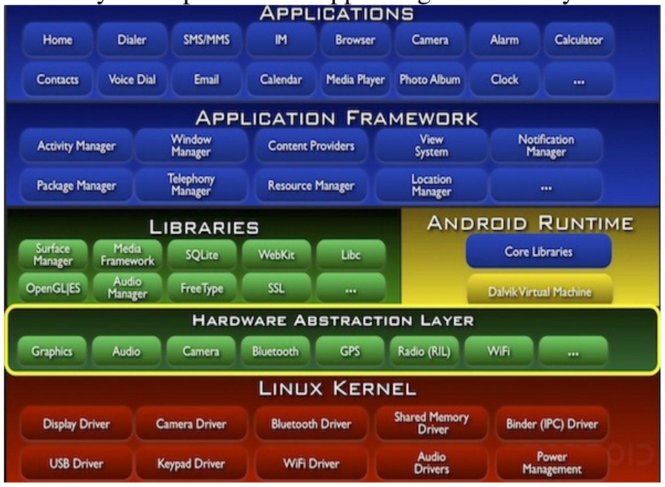


Figure 13. Android modular stack architecture

4. SIMPLE PRACTICE

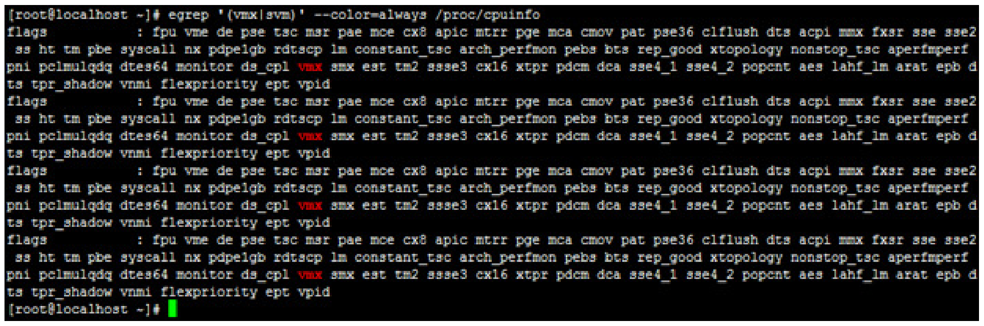
Based on the architecture design of new Smart TV platform stated above, we developed a simple set consisted of a cloud virtualization server, a hypervisor, and N-Screen clients to test the main functions of the design. Its purpose was to test basic functionalities of each module and to make which combination might show good performance.

4.1 Hardware preparation for cloud virtualization

The base hardware platform is generally x.86 (or x.86-64) blade server with adequate numbers of CPUs and memories to be used on virtual machines. To install a hypervisor on the general x.86 based hardware, the CPU of the hardware should support 'hardware virtualization' to handle full virtualization type guest OS installation. Using a command on a Linux shell, it is easy to find out whether the hardware supports it or not.

Command: ***egrep '(vmx|svm)' --color=always /proc/cpuinfo***

If the CPU supports hardware virtualization, following messages are displayed with the command. There is no message if the CPU does not support it.



```
[root@localhost ~]# egrep '(vmx|svm)' --color=always /proc/cpuinfo
flags           : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2
 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good xtopology nonstop_tsc aperfmperf
pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm dca sse4_1 sse4_2 popcnt aes lahf_lm arat epb d
ts tpr_shadow vnmi flexpriority ept vpid
flags           : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2
 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good xtopology nonstop_tsc aperfmperf
pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm dca sse4_1 sse4_2 popcnt aes lahf_lm arat epb d
ts tpr_shadow vnmi flexpriority ept vpid
flags           : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2
 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good xtopology nonstop_tsc aperfmperf
pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm dca sse4_1 sse4_2 popcnt aes lahf_lm arat epb d
ts tpr_shadow vnmi flexpriority ept vpid
flags           : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2
 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good xtopology nonstop_tsc aperfmperf
pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm dca sse4_1 sse4_2 popcnt aes lahf_lm arat epb d
ts tpr_shadow vnmi flexpriority ept vpid
[root@localhost ~]#
```

Figure 14. Messages of the CPU supporting hardware virtualization

4.2 Hypervisor environments

Server virtualization has three types: Hardware emulation type, Full virtualization type, and Para-virtualization. Firstly, Hardware emulation type is most complex structure and especially used in game emulators like MAME, PCSX, etc. It is also used in phone emulators like Android or iOS phone emulators. It is mainly targeting independent device emulation for specific purpose.

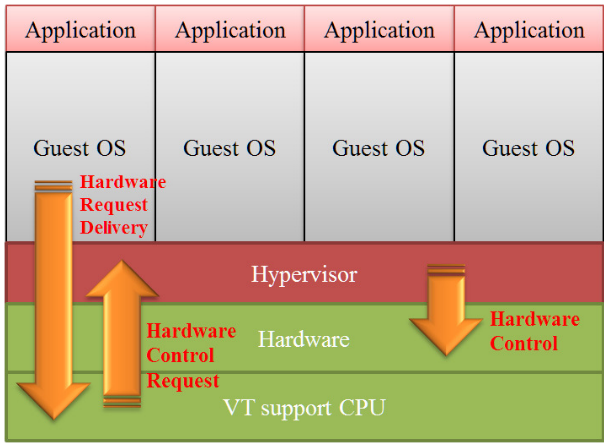


Figure 15. Full virtualization control flow

Secondly, Full virtualization type uses a virtual machine module called 'hypervisor' to control over hardware between the hardware and multiple guest OS. It does not need to amend or revise guest OS because hypervisor emulates hardware commands. It is getting popular because deployment is rather easy and has merits of using a guest OS without modification and supporting most renowned operation systems including Windows. Para-virtualization type was proposed to enhance performance compared to Full virtualization. It is almost similar to use its own hypervisor to control hardware but a guest OS should be revised and recompiled to have control code inside. It cannot deploy Windows as guest OS because it needs open source code.

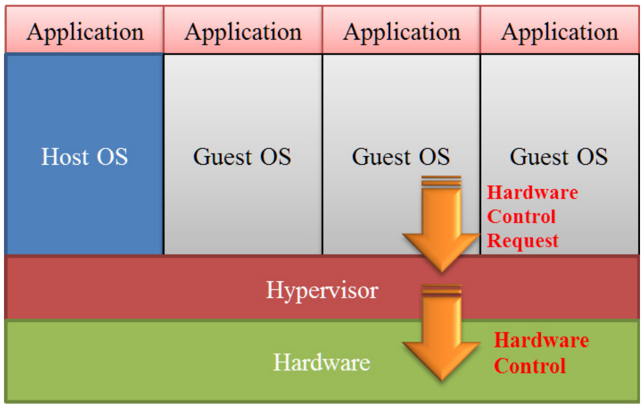


Figure 16. Para-virtualization control flow

To accomplish the main goal to test functions of new Smart TV platform as Post PC, it is necessary to deploy Windows as a guest OS for user workspace environment. Thus, we tested widely used and renowned full virtualization hypervisors in open cloud environment: OpenXen and KVM. Following figure shows the brief overview of two hypervisors.

Table 1. Environment review on OpenXen and KVM

Hypervisor	Company	CPU	Host OS	Guest OS	License
OpenXen	Citrix Systems	X.86 X.86-64 IA-64	NetBSD Linux Solaris	Linux Solaris FreeBSD NetBSD Windows	GPL
KVM	Redhat, Inc. (formerly Qumranet)	VT support Intel/AMD processor,	Linux	Linux FreeBSD Solaris Windows	GPL2

The detailed specifications of two hypervisors are very similar. Following table shows two hypervisor's comparison on each item. The most different thing is that OpenXen support both full virtualization and para-virtualization.

Table 2. Performance factor comparison on OpenXen and KVM

Hypervisor	SMP support	New OS deployment	Virtualization type support	Performance	Commercialization
OpenXen	O	Possible	Full/Para virtualization	Fast/Good	Possible
KVM	O	Possible	Full virtualization	Fast/Good	Possible

The more detailed hardware control functions for two hypervisors are listed below. For desktop virtualization, two hypervisors support partition, USB, real-time migration, real-time memory allocation, 3D acceleration, etc.

Table 3. Main function comparison on OpenXen and KVM

Hypervisor	Partition Support	USB support	Real-time Migration	Real-time Memory Allocation	3D Acceleration
OpenXen	O	O	O	O	VMGL
KVM	O	O	O	O	Depends on hardware spec.

We tested OpenXen and KVM as hypervisors whether it was adequate approach to handle desktop virtualization for new Smart TV platform design. The hypervisor and open platform showed good performance to meet our expecting performance level.

4.3 Multimedia transaction module and QEMU

To test the adequate design for the multimedia transaction module shown in Figure 9, we developed basic multimedia transaction module and linked with QEMU. The multimedia transaction module was developed as library module to combining to other modules. It supports two channels: screen-audio channel and control channel. It can be extendable to add more channels to the needs of multi-channel. The module is executed and linked with QEMU upon the hypervisor. The module process is executed in pair with each guest OS. QEMU is originally emulating a target system and its architecture upon a host system. If the target guest system and the host system is the same OS, it is not effective. However, if the architecture of host and guest systems are different each other and the target system is slower than the host system, it is expected to enhance the performance of the target system to the extent. QEMU is converting the target system's code via TCG (Tiny Code Generator) operation and change it to the host code to be understood in the host system. Following figure shows the conversion process.

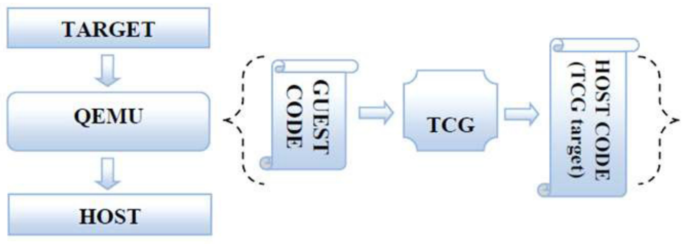


Figure 17. QEMU code conversion process

4.4 Network module & N-Screen development

Network end-to-end connectivity is very important factor for user access capability. To access for users where inside the private network or a firewall protected, we designed the session transaction server called network tunnelling server and developed it simple.

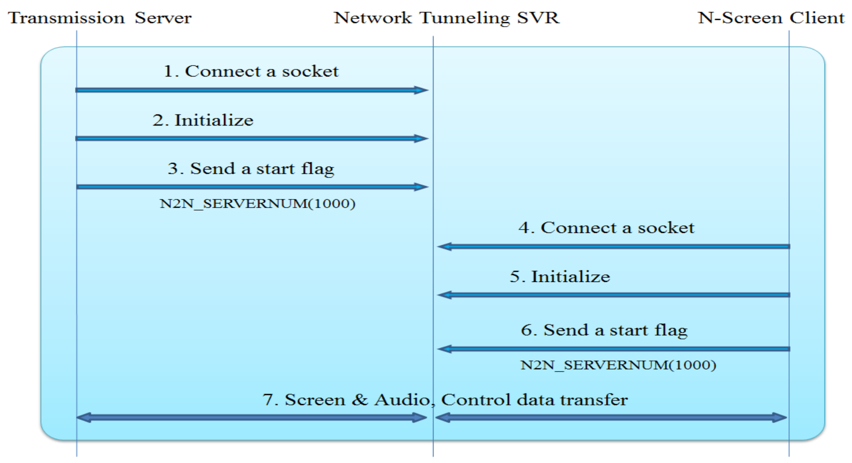


Figure 18. The role of network tunneling server and data flow

Above procedure shows the data flow among the transmission server, network tunnelling server, and clients. A start flag is important factor for successful session transfer because the flag of the server and the flag of the client should be the same as it is used as key value to transfer its session.

Table 4. Start flag related structure on session transfer

Transfer a start flag (N2N_SERVERNUM)		
Parameter	Type	Description
Command	Unsigned char (2)	N2N_SERVERNUM(1000)
servernum	Unsigned int (4)	Session identification ID

We developed N-Screen clients for Windows, Android OS, and iOS for the practice. Though the programming languages are different on each smart device OS platform, each basic class structure or data flow is almost the same.

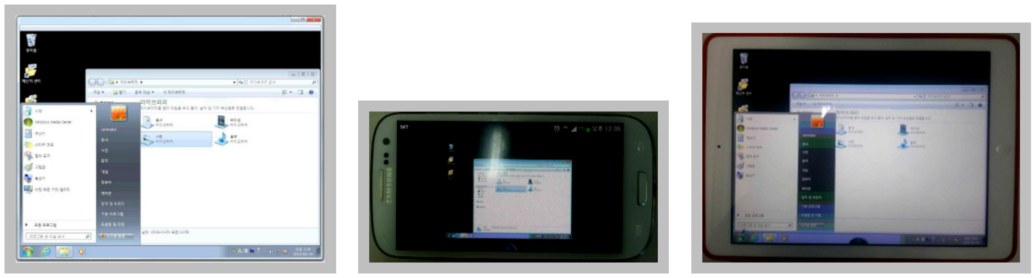


Figure 19. N-Screen clients of legacy Windows OS, Android OS, iOS device connected to the virtulization platform

4.5 Overall test result

Upon the simple practice following the new platform design, overall results were summarized as below. Cloud virtualization & N-Screen technology with open system environment showed good performance.

Table 5. Simple practice result summary

Specification	Requested function	Legacy system (1st Generation)	Test result
Multi-channel design	Multi-channel deployment	One channel in all (Screen, Control)	Two channels support
Network tunnelling server design	session establishment in private network	Network configuration support	Public & private network support
Cross-network design	Cross-network support	LAN based	LAN, Wifi, 3G, 4G support
Speedy remote access & control design	Both document and multimedia contents	Lack of multimedia support	Buffering time minimization
N-Screen design	Multiple smart device support	Limited	More than 3 screens (Win, Android, iOS)

5. Conclusion & Further Considerations

Reviewing architecture comparison focused on functions provided to Smart TV users, the new platform provides more flexibility compared to the previous approaches. New Smart TV platform is effective in that it consists of pure software based server and thin client to support N-Screen devices even including legacy desktop resources. It also provides both multimedia mode and document mode to support office work and individual needs. Following table will show the benefits of proposed new Smart TV platform.

Table 6. Legacy Smart TV system, Android TV, iTV, and new platform

TV Type / Coverage	Legacy Smart TV	Android TV	iTV	New Platform (Virtualization)
Internet	O	O	O	O
Cloud server (Movie files)	O	O	O	O (Multimedia mode)
Thin client	X (Specific device)	Android only (limited)	iOS only (limited)	O (Any OS on his own)
Legacy desktop PC as the client	X	X	X	O (S/W installation)
apps	X	Android only (limited)	iOS only (limited)	O (On his own device)
Document work (PowerPoint, Excel, Word, etc.)	X	Limited	Limited	O (Windows guest, Document mode)

Regarding functions to be developed or dispatched according to the design above, the new platform should utilize generic functions provided by cloud virtualization technology and related open technologies. Following figures show the new platform's coverage compared to legacy desktop sharing and pure hypervisor based open software platform.

Table 7. Screen share functionality comparison

Functions/ Types	Legacy Desktop sharing	Hypervisor based Open S/W	New Platform (Hypervisor based)
Desktop sharing	O	O	O
Multi-channel protocol & management	X	O (VDI protocol)	O (development or alteration needed)
Video streaming mode	X	O	O(same as above)
1:n multiuser view (N-Screen)	O	X	O(same as above)
Local network support	X	X	O(same as above)

As a further discussion, new platform should enhance effectiveness by deploying a multi-user OS for virtualization guest. Up to now, cloud virtualization server allows just one session between the server and a client. Though new platform design revises it to enable 1: n sessions and support N-Screen, original concept is based on 1:1 VDI concept. To support multi-user OS in virtualization server, it seemed that there need lot of efforts to modify it. However, if many users can access to a multi-user OS guest in VDI server for Smart TV, it becomes very effective approach to upgrade the capability of the platform as shown Figure 14.

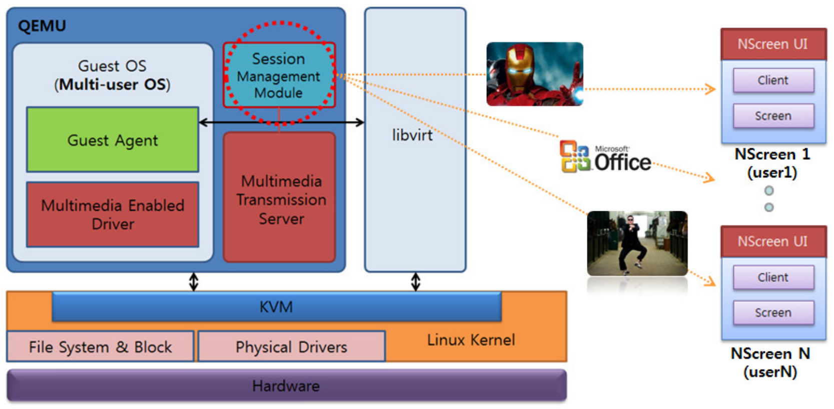


Figure 14. Consideration on the multi-user OS guest support

REFERENCES

[1] Steve Kovach, "What Is A Smart TV?", http://www.businessinsider.com/what-is-a-smart-tv-2010-12, Businessinsider.com, December 8, 2010

[2] Adrian Kingsley-Hughes,“PC, post-PC... what next?”, http://www.zdnet.com/pc-post-pc-what-next-7000018071/, zdnet.com, July 15, 2013

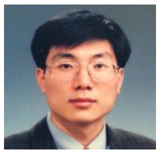
[3] “Collaboration in the PostPC era”, http://www.cisco.com/c/dam/en/us/products/collateral/unified-communications/jabber-android/cisco-collab-at-a-glance.pdf, Cisco.com, 2012

[4] Tobin, A., “A Sky in the Cloud: The UltraViolet initiative has given a boost to the notion of cloud-based digital lockers as the route to secure multiscreen on-demand availability. But does the concept make sense for pay TV operators?”, Digital TV Europe, Vol. No. 306, pp. 24-29, Informa Telecoms & Media, 2012

[5] JA Kim, DH Kim, "A Study on the Screen Evolution and Expansion of the Concept", The Journal of Korea Institute of Next Generation Computing, Vol.8 No.2, pp.87-98, Korea Institute of Next Generation Computing, 2012
[6] Samsung TV portal, http://www.samsung.com/global/article/articleList.do?articleMode=category&lndSiteCode=uk&selectedCtgry=3&selectedCtgryOrder=1&page=1, Samsung.com, 2013
[7] EJ. Choi, HW. Song, JW. Lee, CS. Bae, "Web-based Personal application Managements in Personal Cloud Computing Environments", The Journal of Korea Institute of Next Generation Computing, Vol.10 No.1, pp. 65-73, Korea Institute of Next Generation Computing, 2014
[8] Google TV, "How it works?", http://www.google.com/tv/features.html, 2014
[9] Chris Smith, "More evidence suggests Apple's 'iTV' plans are real", http://bgr.com/2014/03/06/apple-itv-release-date/, BRGMedia, LLC, 2014
[10] GW. Kim, WJ. Lee, CH. Jeon, "Virtualization technology for cloud computing", KSCI Review, v.18, no.1, pp.25-33, KSCI, 2010
[11] Tsai, H. Y. , Siebenhaar, M. , Miede, A. , Huang, Y. , Steinmetz, R.,"Threat as a Service?: Virtualization's Impact on Cloud Security", IT professional ,vol.14 no.1, IEEE, 2012
[12] Xing, Y. , Zhan, Y., "Virtualization and Cloud Computing", LECTURE NOTES IN ELECTRICAL ENGINEERING Vol.143, pp305-312, Springer Science + Business Media, 2012
[13] Hudic, A. , Weippl, E.,"Private Cloud Computing: Consolidation, Virtualization, and Service-Oriented Infrastructure", Computers & security Vol.31 No.4, Elsevier Science B.V., Amsterdam, 2012
[14] Yang, C.-T., Tseng, C.-H., Chou, K.-Y., Tsaur, S.-C., Hsu, C.-H., Chen, S.-C., A Xen-Based Paravirtualization System toward Efficient High Performance Computing Environments, Lecture Notes in Computer Science, Vol. No. 6083, pp126-135, SPRINGER-VERLAG, 2010
[15] "KVM and open virtualization: Who's using it, how and why?", IBM Systems and Technology, Thought Leadership White Paper, IBM, May 2013
[16] "Xen Architecture", http://en.wikipedia.org/wiki/Xen, Wikipedia.org, October 2012
[17] "QEMU", http://en.wikipedia.org/wiki/QEMU, Wikipegia.org, September 2011
[18] "Remote Desktop Protocol", http://en.wikipedia.org/wiki/Remote_Desktop_Protocol, Wikipedia.org, May 2014
[19] "Independent Computing Architecture", http://en.wikipedia.org/wiki/Independent_Computing_Architecture, Wikipedia.org, May 2014
[20] "RFB protocol", http://en.wikipedia.org/wiki/RFB_protocol, Wikipedia.org, March 2014

Authors

JuByoung Oh (Mr.)
CEO & Chairman, Koino, Inc.
Senior Researcher, ETRI
Ph.D. Completed doctoral course in Computer Science Engineering, Chungnam National University



Ohseok Kwon (Mr.)
Professor, Computer Science Engineering Department, Chungnam National University
M.EE, KAIST
B.EE, Seoul National University