GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

SUBJECTIVE EVALUATION OF SMART TELEVISION PLATFORM TECHNOLOGIES FOR ARCHITECTURAL AND PERFORMANCE COMPARISON

Druvika Pandita[*1], Swastik Gupta[2], Ashish Suri[3] & S.B Kotwal[4]
[*1,2,3&4]School of Electronics and Communication Engineering Shri Mata Vaishnodevi University, Katra,
Jammu and Kashmir, India

ABSTRACT

Consumer electronics has been recognized with advanced possibility with the advancement in computing trends and technology. Almost all the consumer electronics like smart television, stereo systems, telephones, mobiles now run on the smart operating systems that profiles internet access, multitasking, streaming media live, audio video playbacks and a broad range of applications. A smart television entitles an operating systems or platform that enables the user to access and view network based and online multimedia content eliminating the need to connect to an extra box hardware. There is a strict need for comparison for being acquainted with variations present in content access, possible privacy concerns and for developer concerns, the software development kit (SDK), the integrated development environment (IDE) framework and the supported codecs. Above all there is a need for balancing the physical attractiveness of a smart television platform, with other important factors like native applications, web support and playback. This paper recognizes several such smart television platforms for television and presents a subjective comparison among them by means of contrasted parameters of application framework, developer SDK, web support, multitasking, playback, schedulers and others. The results are noted and evaluated.[1]

Keywords: *operating systems, memory, multitasking, kernels, tizen, webOS, android, framework, software development kit, system manager, legacy television, native playback, device driver web support*

I. INTRODUCTION

For the scenario of consumer electronics “Smart Television” are on priority list these days. Technological innovations in the sector of multimedia, in particular, with the incoming of the digital televisions have made the television ecosystem stronger. [1], [19] Ignoring the missing productivity outcomes like e-mails and word management, a smart television is similar to the computer. This smart screen is not about the shape or the color picture or even the sound, but the concept of user experience also comes in as a principal element. The system can be connected to the web service to distribute popular over the top (OTT) services.[17] These television ecosystems are defined not by the channels tuned but by the software they are qualified to run. These new capabilities of the smart television are made to exist due to the presence of strong “application processors” originally designed for the use in smart phones. Mostly installed as an array of as many as fast processing units, these processors compete with the modern personal computers in performance.[10] In addition to the low basic incremental costs, the high performance computing has been added to televisions which can now support web browsing, streaming, multimedia options, video conferencing and usage of various applications and widgets. [11] As smart television ecosystem has turned out to be a handy appliance for vast range of applications, it also has now been a logical control for the automation of the house like security monitoring and smart homes. All these functions gain access by the networking technologies which can trustworthily interact with sensor devices of different types distributed throughout a home system. [1]

All the listed technologies constituting the pre-requisites of a smart television are made available and tested: processor design, architecture and integration, picture quality enhancement, media implementations with network technologies and so on. In this paper mere focus is upon several smart television ecosystems to provide a comprehensive review of their various features and research impacts along with an exploration via programming language, memory consumption, availability of the application store, native applications, schedulers and others.[12]

II. GENERAL SYSTEM ARCHITECTURE

The initial idea of the smart television began to add various functions like the internet and web specifications to normal legacy television. On the basis of the fundamental smart television concept, basic legacy system architecture has the server issuing contents and applications, network devices with internet connection and set top box clients for home appliances.[14] The figure 1 provides the legacy architecture of smart television system. It comprises of basic network or broadcasting function engine, user interface and overall management module, codecs for audios and videos, and a web based module. The architectural system can incorporate contents of only pictures and videos that were already defined or put up as a standard. Legacy smart television platform was generally designed on a private closed scenario and needed customization. To cover up the weaknesses, various approaches tried to get updated.[22]

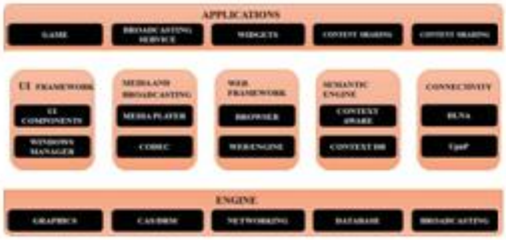


Fig.1. Legacy architecture of smart television

Legacy is mainly a platform that combines internet as well as web services in televisions and set top boxes, and provides a path for technological integration between those televisions and devices like smartphones, tablet and computers. Developing applications for legacy television platforms were somewhat different from developing the simple web pages for the internet as shown in Table 1.[14] This technology could increase management by sharing resources and by providing flexibility and scalability to this platform, it could have its on cloud space.

Table 1 legacy television development vs web development

Features	**Legacy Television Platform Development**	**Web development**
Three major resolutions supported	960 x 540 pixels 1280 x 720 pixels 1920 x 1080 pixels	Supports responsive layouts, and adjustable for different screen sizes and resolutions
Application display	Use all the advantages of its large display	Displayed often using a web browser on a computer or mobile
Engines	Gecko, for platforms and Webkit for newer Platform	done for many different browsers

III. TIZEN TV ECOSYSTEM

Tizen television ecosystem is an open and workable operating system built up to tackle all the needs of the mobile and internet connected devices ecosystem, all involving mobile manufacturers, mobile operators, software vendors and application development.[2] Tizen ecosystem is an open source platform and invites any members who are willing to participate. There are various Tizen OS profiles which have been serving different industry requirements generally involving Tizen IVI, Tizen Mobile, Tizen Wearables. Also for Tizen 3.0 version, all the profiles are constructed on top of a Tizen Common. These profiles can be used by a Tizen manufacturer for modifying the architecture to meet the needs or to develop a profile to interact with the memory, power requirement and processing of any device. Customization of these operating systems can be modified by device partners and mobile operators to meet the special need of demographics.[9] The Tizen ecosystem gifts the power of application development for native apps and also the potential to extend the reach to wearables, electronics, as well as smart devices. The basic platform for Tizen is based on Linux kernel and GNU (GNU not Uniplexed Information and Computing Service UNIX) library incorporating the Linux Application Programing interface (API).

A. System Architecture

This Tizen ecosystem provides a complete standard based platform for various device categories. Also in addition to the standard, a comprehensive description for the development of the web application is provided by the Tizen Web API. It gives the leverage to use the full ecosystem to handle web applications. The core possesses Tizen Core service and Tizen API. Its web application API is a group of newly defined API's. The kernel layer supports Linux kernel and different device drivers. Tizen gives application development tools supporting java script libraries jQuery Mobile and jQuery. Tizen software development kit enables the user to use HTML 5 and other related web technologies to develop applications that run on supported devices. Tizen comes up with standard based SDK for multiple categories of devices. Tizen web application program interface provides a complete description for web application development.[2] Figure 2 shows a complete Tizen architecture for smart phones and tablet devices.

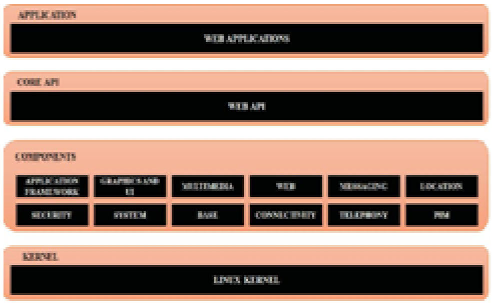


Fig.2. Tizen architecture for smart phones and tablet devices.

Tizen braces web applications using the application layer and Tizen web application use the full power of the platform similar to the native ones. The core layer possesses the core service and the tizen API. Tizen core has application framework, security, connectivity (for 3G, Wi-Fi, BT and HTTP), graphics and UI, location based services, messaging (SMS, MMS, Email and IM), multimedia (Gstreamer), Personal information Management, telephony and web services (Webkit).The kernel layer has the device drivers and Linux kernel.[9]

B. Programming SDK

A tizen application works on the standard technology such as Java script, HTML, CSS, and WSC widget packaging. The Tizen service API is very rich, and thus can help developers to create a vast number of applications which could run on multiple devices. Tizen software development kit is a complete set of tools for different applications: Rich templates for web applications, Java script, CSS syntax checker, HTML. The Tizen ecosystem supports the following Software Development Kit (SDK) interfaces:

Web SDK - supports HTML 5 and Java script

SDK

Native SDK /Knox Tizen SDK - C based SDK/ C and C++

C. Availability of application store

Tizen store is a store for applications designed specifically for Tizen applications. Tizen store provides applications optimized for tizen devices to users in over 180 countries. Also can be used for providing variety of service information like store introduction and recommend collection.[2] Applications in this store are featured in All Games, Social networking, entertainment, kids. In addition to these, selected applications and recommendations are featured.

D. Multitasking

This feature allows the user to save the application when the user launches a different application and is started back when application is restarted. In Tizen Television ecosystems, when the user shifts from one application to another or a channel is changed, Java Script execution is halted and current application must preserve its current state to RAM and keep it in the background. This state is recovered back when the same is resumed. All Tizen ecosystems support multitasking with addition to the TV emulator.

E. Multi video playback

Different audio and video formats can be played in this application. The system allows to play audio and manage the playback in addition to the handling of video formats. The Tizen can accurately control the player and audio video sound mode with sending particular URLs for the streaming playback. The Tizen OS multimedia structure is based on G streamer, an open sourced project.[2] It can assist the user to play and interpret video, audio, as well as VoIP. Tizen OS platform enables support for HTML5 audio/video tags in addition to embedded playback. It gives support to media information as well as metadata extraction.[9] All the codecs supported by tizen OS are mentioned in Table 2.

Table 2- different codecs supported by tizen

Type	**Formats**
Audio encoder	AMR-NB Vorbis
Video decoder	MPEG-1, H.263, H.264, WMV3, VC1, MPEG-4 part 2 MS v1/2/3, Theora
Audio decoder	WAV, Vorbis, AAC, MP3, WMA 7/8, AMR- NB / AMR-WB
Video encoder	MPEG-4 part 2,H.263, Theora

The Tizen schedulers have specific scheduler API's and are specific to service applications. With the help of the scheduler API, applications can enroll a background task with particular triggered conditions. When the triggered condition occurs, the service application is launched. A native application mainly could be defined to be an application program that has been made for use on that platform only.[9]

This uses the native Application program interface, which provides variety of interfaces to the device hardware enabling the user to take advantage of different capabilities which can run with limited device resources. Tizen (Samsung) is always quite active in the market of smart televisions.[2] Tizen platform is developed for the future purposes. Tizen has enabled to create a bright user experience for mobile devices and devices they connect to.

1. WEBOS TV ECOSYSTEM

This television ecosystem is often known as the Open webOS or LG webOS, is a Linux kernel operated multitasking operating systems mainly incorporated for Smart television. This platform was made open source by HP. Many devices have webOS featuring in their devices including Pre, Pixi, and Veer smartphones, Touch Pad tablet and others. Within the basic graphical user interface, webOS has many similarities with the mainstream Linux distributions.[21]

A. System Architecture

The Application environment is the system user experience provided along with the feature that is entitled to the application developer as illustrated by Mojo Framework and Palm services. The core OS covers: Linux kernel, drivers through the OS services, subsystems, middle wares, wireless systems. The UI system manager is mainly responsible for everything that is user visible. The application run time is given by the individual applications, and built in framework and some system applications.[1] The application Manager executes in one process schedules and manages every running application and handles all through interfaces to graphic system and on device storage. Applications are based upon the framework for their UI trait set and for services access. The UI features are made into the framework and are taken over by the application manager directly to the service handler. [21]No direct interaction with the end user and Core OS takes place. Here users communicate to the applications and UI System Manager mainly headed towards System UI. All together this forms the application environment. Figure 3 shows a simplified vie of the web OS architecture.

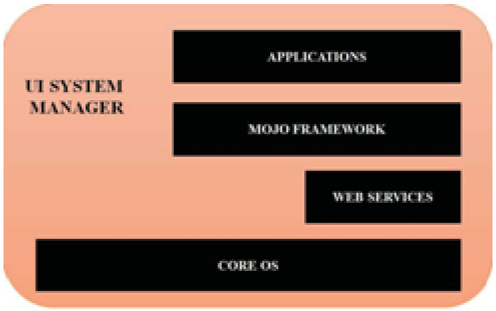


Fig.3. Simplified webOS architecture

The webOS environment uses the Synergy architecture to enable users to synchronize personal contacts, tasks, calendars or emails during segregation of that information without settling with enterprise data integrity. The initial Palm OS has a typical native model for application scenario.[20]

This model comprises of the application data, user interface, logic interface all integrated on the executable with addition to direct access to OS data and services. The standard web applications are the simple HTML based applications which leave a HTTP request to the web server at the end of every user action and then halt and wait for a response like Ajax applications which handle various user interactions and make web requests asynchronously. Also it may include event handling, multitasking models, and the notification services. Palm webOS provides support to traditional web content quite competitively. [17]

B. Programming SDK

A webOS application is similar to the web application based entirely upon standard CSS, HTML, but the lifecycle of the application is different. All the applications are executable within the UI system manager, a runtime built upon the standard browser technology, for rendering display, assisting with the events and handling java script. Here the API’s are delivered in the form of Java Script called Mojo, which enables common application level functions, access to built in applications, native services etc.[21] LG webOS smart television platform, is the fast, smartly designed and well stocked in entity of the smart applications. Mainly languages including HTML, CSS and Javascript can be used to develop webOS. Palm has stretched the card based multitasking abilities of webOS including a new card stacking feature that will bring together related tasks. This would create ease in the task management and for users to navigate between individual cards and applications.

C. Availability of application store

LG webOS has a provision of application store but with a very limited number of applications. The applications are available on the basis of licensing and the local TV providers. Applications are mostly free but the content is not. In group applications and games there would be 75+ applications while in the premium scenario there would be 18+applications. [8]

D. Multitasking

webOS is a multitasking operating system for devices such as smart televisions and it has been mainly used for mobile operating systems. The navigation in webOS is based on multi-touch gestures for touch screen. This user interface makes use of 'cards' for managing multitasking and representing applications.[20] The user switches and transfers between running applications with swipe from left and right on the screen. Applications are closed by swiping a 'card' up. The application 'cards' can be changed for an organization.

E. Multivideo playback

All the supported AV codecs are mentioned in table 3.Using the webOS software development kit and Mojo API, any conventional web application can be build, downloaded and installed. This can be supported on any webOS device. Also various widgets can be added to the applications to leverage the rich UI inbuilt in Palm webOS. Smart television with webOS platform can help achieve a perfectly consistent home entertainment experience.

Table 3 different audio video codecs for webos

Extension	**Codec**	**Standards**
.mkv/.mp4/.ts	Video	HEVC, H.264/AVC
	Audio	Dolby Digital/ plus, AAC,
.mpg/.mpeg/ .dat	Video	MPEG-1 MPEG-2
	Audio	MPEG-1 Layer I II MPEG Layer III
.vob	Video	MPEG-1 MPEG-2
	Audio	Dolby Digital, MPEG-1 Layer I II DVD-LPCM
.rm/.rmvb	Video	RV 30, RV 40
	Audio	Real audio 6 AAC LC, HE-AAC

IV. ANDROID TV ECOSYSTEM

Android Television is a smart television platform designed and developed by Google. On the basis of Android operating system, an interactive experience of television is created through a user interface. The Android Television Framework reduces the complexity of delivery of the live media to the Android Television. [23]The Android Television Input Framework (TIF) offers the standard API to the manufacturers in order to create various input modules for the control of Android television and to allow live television search as well as recommendations published by TV media via metadata. The framework has no intention to implement television standards for requirements, but it can help to meet various television broadcast standards without reimplementation. This feature set would increase with the latest versions in which the platform TIF API are stretched. *A. System Architecture*

The Android television input Framework consists of a TV input manager.[9] This input framework works with the TV application, a system application that cannot be replaced by any third party application, for accessing the built in and tuner IP channels. The TV application communicates with the television input modules given by the manufacturer via the Television input manager.

The Television input framework (TIF) mainly has: a TV provider which is a database of programs and channels, a TV App which is the application that handles the user interaction, Android Television Input Manager which enables the TV inputs to interact with the TV application, TV Input which is an application indicating virtual or physical tuners with input ports, also, TV input HAL which is a hardware allowing the system TV inputs to use TV specific hardware on implementation, HDMI-CEC which allows the remote handling of different devices over HDMI.[23] Figure 4 shows the detailed architecture of the Android TIF architecture. The architecture is exercised in the following way: The user interprets and interacts with the television application, a system application which cannot be changed by a third party application.

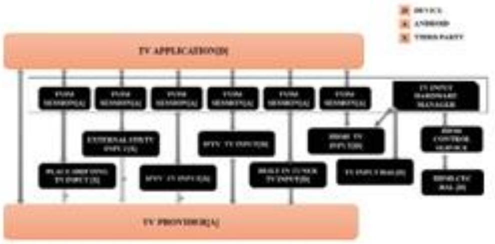


Fig.4. Android Television Input Framework

The TV Provider database saves the channels and the programs from the android TV Inputs. The android TV provider manages the permissions for TV inputs. The android TV input manager gives a central hub system API to the android TIF. It judges interaction between TV inputs and applications and also additional parental control functionality. Its sessions should be created one to one with TV inputs for listing and checking statuses and creating sessions for managing listeners. The TV inputs are the applications in the sense that they already have a preinstalled manifest.xml. Some of the inputs just like the HDMI or built in tuner input, is provided by the manufacturer only with the fact to speak with underlying hardware directly.[17] Others like IPTV, external STB and place shifting can be provided by the third parties on store. After the downloading and installation the new input would be checked within the application.

B. Programming SDK

Android TV applications are developed as a mixture of components that could be invoked independently. It enables the user to provide the various resources for different devices. Android offers a bright user experience that could be optimized for running applications on screen devices, like High definition televisions (HDTV's). Android television ecosystems can be developed in many programming languages including: Java, Kotlin, C#, Python, HTML5, Lua etc.

C. Availability of App store

The android TV provides a portal for selection of applications which can be accessed in the play store on Android TV. All the applications like NETFLIX, BBC iPlayer are featured in android TV store. [17] All the applications in the category of gaming, entertainment, and multimedia are available. Android TV permits single, multi, online as well as offline gaming modes.

D. Multitasking

All of the released Android TV devices are much capable of multitasking and a remote friendly selection menu can be easily developed. Android TV enables the applications to run in the background similar to any other android device. This implies that the play audio or any other tasks can run in the background. Also relaunching an application from the launcher to its paused previous state will be done if the active application is not destroyed. [23]

E. Multi video playback

This section describes the codecs supported by the Android platform. For the development any media codec can be used which is available on any Android powered device including those which are device-specific. Usually device specific media encoding profiles are best in practice. Table 4 describes the few media format support for Android TV platform.[23] The android television ecosystem feature set can be viewed as a system which would be able to: access and navigate all the channels, access information bar, access electronic programming guide data, support multiple audio video, provide parental control PIN challenge, displaying application linking cards, supporting time shifting API, handling functionality of DVR and support recording API's.

Table 4 media codecs supported by android tv

Supported Codecs	**File type/formats**	**Details**
Video H.263	.3GPP, MPEG-4	
Video H.264 AVC	.3GPP, MPEG-4, MPEG-TS	Baseline profile
Video MPEG-4 SP	.3GPP	
Video VP8	.WebM, .mkv	Streamable in 4.0 and above
Audio AAC LC	.3GPP, MPEG-4,.ADTS(.aac decode in 3.1+)	Support dor stereo, mono, 5.0, 5.1 content
Audio AMR NB	.3GPP(.3gp)	4.75 to 12.2 kbps sampled
Audio MIDI	Type 0 and 1(.mid, .xmf), .OTA, RTTTL(.rtx)	Support for ringtone formats OTA, RTTTL/RTX and iMelody
Audio MP3	.mp3	Mono/stereo CBR or VBR

V. KODI TV ECOSYSTEM

Kodi was formerly known as XBMC and is a free and open source smart platform developed by XBMC Foundation. Kodi platform is available for multiple OS and hardware platforms. The Kodi platform allows the users to play and view the streaming media like podcasts, music, videos and common digital media files from network or local media. Kodi ecosystem is multi-platform home entertainment PC application. This ecosystem is richly customizable due to the fact that various skins can change its appearance, and different plugins can help users to access streaming media content like prime, YouTube, Pandora internet, crackle etc. Kodi can be made to run on Linux, windows, android, iOS,

Mac OS X. The further versions have a PVR (Personal Video Recorder) and graphical front end to receive live television with program guide.[20] Various add-ons can be created by third party developers using python. The libraries in Kodi are used to store the contents mostly entertainment.[20] These can be stored anywhere on user computer, phone, or any external device

A. System Architecture

Kodi makes use of OpenGL graphics under Kodi for mac OS and Linux. While kodi for windows makes use of Direct 3D and Microsoft multimedia DirectX framework. Also kodi makes use of Simple Direct Media Layer framework partially. The kodi software is based on six different operating systems varying from high specification to low specification computers for the sole purpose of only running the media player. For the context view, the behavior of the environmental scenario for kodi is shown in figure 5. It can be seen that multiple operating systems are supported by Kodi.[6]

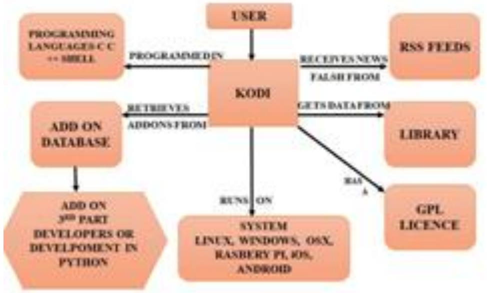
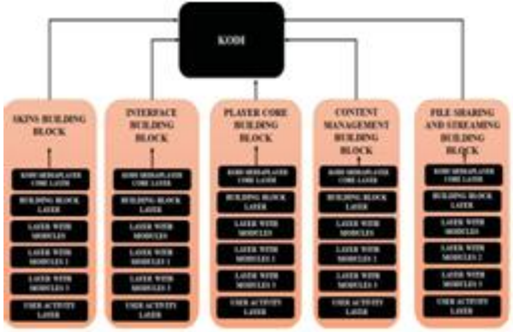

USER
PROGRAMMING LANGUAGES C C ++ SHELL
PROGRAMMED IN
KODI
RECEIVES NEWS FALSH FROM
RSS FEEDS
RETRIEVES ADDONS FROM
ADD ON DATABASE
GETS DATA FROM
LIBRARY
HAS A
GPL LICENCE
RUNS ON
ADD ON 3RD PART DEVELOPERS OR DEVELPOMENT IN PYTHON
SYSTEM LINUX, WINDOWS, OSX, RASBERY PI, IOS, ANDROID

Fig.5. Context view for Kodi television ecosystem

Kodi platform is developed in flux with standard for supporting operating systems as well as different hardwares. For extending the functionality of Kodi, at the run time addons are used by kodi. Kodi had external dependencies or certain limitations put on by certain hardware that are to be considered while developing as they mainly impose effects to the architectural design. Figure 6 shows the modular design depicting all the building blocks. From the figure, it can be seen that the Skin building block contains all the packages and files regarding the fonts, translations, skins, graphical user interface of the Kodi platform.[20] It forms the most popular modules for development processes and also a great scenario of visualization of module working with software architecture.


KODI
SKINS BUILDING BLOCK
INTERFACE BUILDING BLOCK
PLAYER CORE BUILDING BLOCK
CONTENT MANAGEMENT BUILDING BLOCK
FILE SHARING AND STREAMING BUILDING BLOCK
KODI MEDIAPLAYER CORE LAYER
BUILDING BLOCK LAYER
LAYER WITH MODULES
USER ACTIVITY LAYER

\

Fig.6. Building blocks of Kodi

The interface building block contains all the packages and files regarding the add-ons, event servers, streaming clients and all the external libraries. The ad-ons are mainly developed by the external developer. The content management building block has the functions, packages and files about handling multimedia content. This block works on rather a difficult side of media files which involves hard disk, disk drives external server streaming, collecting data from FTP, HTTP, RSS and other sources. The player core building block has the task of reading and displaying the AV files with the help of codecs which are important for the conversion of rough data files to the clear view for user. Finally the file sharing and streaming building block is concerned with the sharing of files. This is connected to the add-on that enables downloading, streaming, sharing and scraping just like FTP service providers who publish software, movies, and audio that could be downloaded.[17]

B. Programming SDK and Store

Kodi contains a built in Python interpreter which enables the users to develop scripts and plugins (add-ons) which interface very easily and clearly with the Kodi dashboard. These add-ons increase the functionality of Kodi with the ease of no extensive programming. The kodi software development kit has many features as: it provides the event support for xmbc.Monitor, the python interfaces (variable management, value escaping, function calls, console logging), C# bindings for Kodi modules (having xbmc, xbmcgui), URL routing and support for add-ons. Python scripts allow normal users to add new functionality and meaning to Kodi.[20]

The present plugin scripts include functions as that the movie- trailer browsers, Internet TV, cinema guides, weather forecast, and OTT video streaming services including BBC iplayer, Hulu, Netflix, You Tube, Pandora Radio (internet radio station browser), Flickr, email clients, IM, home automation scripts for Front end control PVR software and hardware, IRC, Bit torrent (P2P file sharing downloaders) and also games like Tetris.

C. Multitasking and native applications

Linux Kodi is mainly developed for Ubuntu Linux. The third party packages for almost other Linux distributions are available and, it is possible to incorporate XBMC media center from base for any type of Linux distribution only if required libraries are already installed.[20] Hardware accelerated decoding of video is achieved on NVidia's GPU via the VDPAU API and for AMD/ATI Radeon via the VAAPI API. Kodi for Android is a full port of Kodi C and C++ having all its dependencies to android with system that was designed to control and manage multiple processor architectures just as MIPS, ARM, and X86 in addition with Native development kit for android. APK runs natively within android as a native application. XBMC source code should be compiled with Googles official native development kit for android. XBMC for android does not enable the Google TV ad the NDK for android was not made available for earlier Google TV devices.

D. Multi video playback

The music and video library is one of the most important features of the Kodi. It enables the organization of audio video content by information of video files. Kodi makes use of CORE multimedia video player which is an in-house made media player cross platform in nature and is used for video playback. For the music, Kodi uses PA Player (Psycho- Acoustic Player) also an in-house built core player. This player handles various Audio file formats and enable various tagging standards. The different codecs for Kodi are mentioned in table 5. [20] There are some software limitations in the Kodi ecosystem source code that the DVD player and PA Player cannot officially play any audio or video which are encrypted with DRM technologies.[20] Such media can only be played using another media player. After capturing Kodi ecosystem from different view, it can be concluded that kodi's architecture has been designed well. By the use of python, new functionality could easily be added without recompiling the whole system.

Table 5 media codecs supported by kodi tv

Video formats	**Audio formats**	**Supported codecs**
MPEG-1, MPEG-2, H.263, MPEG-4, SP,	MIDI, WAV/WAVE, AIFF , MP2,	AAC, AC3-WAV, MP3, FLAC, CDDA, MOD,

ASP, MPEG-4 AVC, HEVC, INDE, MJPEG, RMVB, VP8, VP9, WMV, CINEPAK	MP3, AAC, AAC+, AC3, DTS, ALAC,AMR, FLAC, SHN, WMA, MOD, CD-DA, YM	DTS-CDDA, SHN,WMA, XM, MP2, IT, WAVPACK, AC3-CDDA, AIFF, ALAC

VI. CONCLUSION

Smart televisions in present era is getting hold of elements like live media streaming, navigation, on demand learning and various advanced features. The whole ecosystem of television scenarios is changed over the period of years. The decision pattern for the selection of a smart television platform or an ecosystem has got finer with the evolution of technology. Full study of middleware kernels are made. Table 6 shows the overall comparison of the contrasted ecosystems.

Table 6 feature comparison of different smart tv platforms

Tizen TV	**webOS TV**	**Android TV**	**Kodi TV**
Developer Samsung	Developer LG	Developer Google	Developer XBMC foundation
OS- unix like or Linux	OS- based on linux kernel	OS -android	OS- mac OS, iOS, android, Linux, Rasbian
Platform- ARM and x86	Platform- ARM	Platform- ARM, x86, x86-64, MIPS	Platform- ARM, x86-64, MIPS
Auto detection of sources	Multitasking interface using cards	Messaging and google cast with Auto correct and dictionary	Multi language support
Native application support using NDK	Over the air updates, Synergy	Multi tasking and Multi touch	Metadata extraction and skins with Addons and plugins
Application management and navigation	Wireless recharging	Multiple language support	Managers and emulators with Web interfaces
OTT services	Open source	Screen	Native

		capture facility	applcation support
HTML 5 and Java script SDK	Multitasking with HTML, CSS, and javascript SDK	Accessibility with Java, Kotlin, C#, Python, HTML5, Lua	Python scrpit and C# SDK

The aim is not only to provide variant picture quality but also to provide increased user experience. By studying these different ecosystem platforms and by comparing these middleware platforms, increased viewership can be achieved, reach and brand experience can be boosted. The comparison is made layer wise using kernel platforms of architectural framework. Reviewing these infrastructures comparison is mainly focused upon function provided to smart television users, the new to mew platforms provide more flexibility compared to other approaches. These smart television platform are effective in the sense they contain entirely software based servers and thin clients to support different devices even including those of legacy desktop resources. [17]

As per the further discussion, all the smart television platforms should be enhancing effectiveness by installing a multi-user ecosystem OS for virtualization guest. Till now, cloud virtualization server enables a single session in between the server and a client. For the support of multi-user ecosystem OS in server, it is concluded that there are introduction of various new ecosystems for it like Amazon TV, Opera TV, Apple TV and others.

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. *"TIZEN developer," Samsung, [Online]. Available: https://developer.tizen.org/. [Accessed january 2018].*
3. *Adrian Kingsley-Hughes,"PC, post-PC... what next?", http://www.zdnet.com/pc-post-pc-what-next7000018071/, zdnet.com, July 15, 2013*
4. *"Collaboration in the PostPC era", http://www.cisco.com/c/dam/en/us/products/collateral/unifiedcommunications/jabber-android/cisco-collab-at-a-glance.pdf, Cisco.com, 2012*
5. *Solapure, Sharwari Satish, and Harish Kenchannavar. "Internet of Things: A survey related to various recent architectures and platforms available." Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on. IEEE, 2016.*
6. *Tobin, A., "A Sky in the Cloud: The UltraViolet initiative has given a boost to the notion of cloudbased 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*
7. *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*
8. *Bajpai, Vaibhav, and Jürgen Schönwälder. "A survey on internet performance measurement platforms and related standardization efforts." IEEE Communications Surveys & Tutorials 17.3 (2015): 1313-1341.*
9. *Samsung TV portal, http://www.samsung.com/global/article/articleList.do?articleMode=category&lndSiteCode=uk&selec tedCtgry=3&selectedCtgryOrder=1&page=1, Samsung.com, 2013*
10. *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*
11. *Ahvar, Ehsan, Gyu Myoung Lee, and Noel Crespi. "A cloud-based platform for watching same content on three-screen TV continuously in smart home." Information Science and Cloud Computing Companion (ISCC-C), 2013 International Conference on. IEEE, 2013.*

12. *Chris Smith, "More evidence suggests Apple's 'iTV' plans are real", http://bgr.com/2014/03/06/apple-itv-release-date/, BRGMedia, LLC, 2014*
13. *GW. Kim, WJ. Lee, CH. Jeon, "Virtualization technology for cloud computing", KSCI Review, v.18, no.1, pp.25-33, KSCI, 2010*
14. *Rostami, M., João Tavares, and Antonio Navarro. "Multi-decoder digital television platform." Euromicro Conference, 2002. Proceedings. 28th. IEEE, 2002.*
15. *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*
16. *Xing, Y. , Zhan, Y., "Virtualization and Cloud Computing", LECTURE NOTES IN ELECTRICAL ENGINEERING Vol.143, pp305-312, Springer Science + Business Media, 2012*
17. *Hudic, A. , Weippl, E.,"Private Cloud Computing: Consolidation, Virtualization, and ServiceOriented Infrastructure", Computers & security Vol.31 No.4, Elsevier Science B.V., Amsterdam, 2012*
18. *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*
19. *"KVM and open virtualization: Who's using it, how and why?", IBM Systems and Technology, Thought Leadership White Paper, IBM, May 2013*
20. *M. v. d. R. K. S. M. S. Kaj Dreef, "Architecting Software to Keep the Lazy Ones On the Couch," 23 APRIL 2015. [Online]. Available: https://delftswa.github.io/chapters/kodi/. [Accessed NOVEMBER 2017].*
21. *"webOS developer," [Online]. Available: http://webostv.developer.lge.com/.*
22. *Oh, JuByoung, and ohseok kwon. "multi-user service platform design for smart tv & n screen services in open cloud environment." Computer Science & Information Technology (CS & IT) Computer Science Conference Proceedings (CSCP) CS & IT-CSCP. 2014.*
23. *Android developers . (2018, february). Retrieved from Building apps for TV: https://developer.android.com/training/tv/index.html.*