# Design of Triple Play Terminal Middleware for Service Convergence

Chun-Yun Hsieh
Industrial Technology Research Institute
*cyshieh@itri.org.tw*

***Abstract****-This article presents Triple Play terminal Middleware architecture to support Triple Play Service Convergence. The Triple Play terminal architecture, including IPTV and V²oIP, is presented in detail. With significant improvement of the silicon technology nowadays, the proposed architecture can be implemented on single terminal, i.e. Set-Top-Box. The implemented system enables IPTV users to easily handle video phone calls by using remote control while watching TV Program.*

**Keywords:** IPTV, V²oIP, Triple Play, Service Convergence, Middleware

## 1. Introduction

Triple Play is the marketing term used in the telecommunications for provisioning the bundled offering of three services: Television, Telephony and Internet Access. As broadband infrastructure is well constructed, All-IP multimedia service through All-IP networks can come true sooner or later. Moreover, with significant improved media coding algorithms and modern access technologies, i.e. WiMAX, the deployment of Internet Protocol-based television (IPTV) over different broadband access network is made possible [1].

While the evaluation of IPTV Service over Next Generation Networks [1], [2] continuously progress, adoption of NGN-based IPTV remains in early stage. Even so, enabling service convergence over NGN brings many issues and business opportunity [3], [4], [5]. These requests to design and develop an open Triple Play Middleware, and attempt to try user's affinity.

This article presents Triple Play terminal Middleware architecture to support Triple Play Service Convergence. The Triple Play terminal Middleware architecture, including IPTV and V²oIP, is presented in detail. With this Triple Play Middleware, more interaction between IPTV and Telephony service can be archived.

The outline of this paper is as follows: section 2 introduces the proposed architecture of the Triple Play Middleware.

Section 3 introduces application scenarios. Section 4 describes the reference implementation for application scenarios. The last section presents conclusions and future outlook.

## 2. Triple Play Middleware Architecture

This section provides overview of the Middleware architecture and corresponding classification of Middleware API. The terminal software is composed of four layers, including Triple Play Application layer, Triple Play Middleware layer, Terminal RAL layer, and OS layer.

The OS layer provides supporting operation system and hardware resource for upper layers. The operations system is responsible for scheduling the task or process, controlling the signal, memory management, handling system functions, etc. The hardware resources, includes computing devices, CPU, RAM, storage devices, firmware (e.g. codec), rendering devices (e.g. display, speaker), IO devices, etc.

The Terminal Resource Abstract Layer (RAL) is the interface of OS Layer and the Middleware. This layer abstracts the interface of the Drives/OS, and then it brings the advantage that the services and the applications are independent of the Drive/ OS layer. It is designed so that a manufacturer or chip vendor can write their respective device drivers without ever worrying about the middleware software above.

The Triple Play Middleware layer is defined as the layer of software between applications and resources, which consists of a set of service enablers that allow multiple functionalities running on one or more devices in Triple Play service system to interact across a network. It brings an abstract view of OS and resources. It isolates the application from the hardware, enabling portability of the application. The Middleware provides Triple Play application core function modules to enable quick implementation of Triple Play services. It provides APIs for upper layers. They are used and

enriched by the application services in order to simplify the development of service components and applications above it.

The Presentation layer provides required functionality for presenting use experience. It may be a HTML Browser. An application can either have full or restricted access to the presentation engine layer features.

The Applications layer includes the applications of the Triple Play Service, including IPTV, such as VoD, EPG, Linear TV, PPV, PVR, etc, $V^2$oIP, Internet applications, such as IM, Maps, etc, as well as other value-added services. The applications are either downloaded or resident. In particular, an application is powered by a presentation engine (e.g. HTML Browser). An application can either have full or restricted access to the presentation engine layer features. Moreover some applications may directly access the service logic adaptation layer without using a presentation engine.

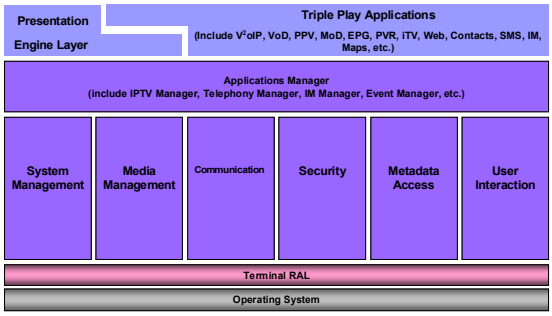


Figure 1 Triple Play Middleware architecture

## 2.1. Components in Triple Play Middleware

This session describes the functionalities of the Triple Middleware components and identifies their APIs. The Middleware is made of an Application Manager and underlying six service components. Those service components are pure native components that offer functionalities common to all Triple Play Middleware implementations (e.g. service discovery & selection, service presentation, service information management, PVR, security system, etc.). They are used and enriched by the application services in order to simplify the development of service components and applications above it.

On the whole, the functionality of those six service components can be defined to full fill deploy Triple Play Service:

i. System and resource management component.
ii. Media management component.
iii. Communication component.
iv. Security component.
v. Metadata access component.
vi. User interaction component.

All this components can optionally use the functions available on the network.

The System and resource management component is used to manage Triple Play terminal device resources, and provide software Initialization API, upgrade and download API, system resources management API, and terminal management API to the upper application layer. It provides terminal device configuration, task initialization, such as application management, DRM task, subscriber authentication process, etc., network connection initialization, such as, to get network parameter, configure access mode, and etc., resource initialization, such as memory, timer, I/O equipment, A/V decoder and other system resource, and coordinating with other Middleware APIs, such as, to work with Security and Authentication API to start user authentication procedure, to work together with Upgrade and Download API to check software version and perform software upgrade procedure, and diagnosis for terminal device.

The Media management component is responsible for managing media streaming, media presentation and media storage, and provides media service API to the up layer application. It provides media streaming management, such as media streaming session set up and control for VoD, multicast linear TV, unicast linear TV, and time shift, media decoding management, media rendering and trick mode management, such as play, stop, pause, and resume, Caption presentation, trigger of DRM process, media buffer management, and PVR and storage management.

The Communication component is responsible for Triple Play terminal device communications, which including the functionalities such as communication management, communication protocol management (HTTP, RTSP, SIP etc.), and Internet access. It provides all the necessary communications between terminal device and service platform. For Internet access, it also controls and manages the basic operations of open Internet resident clients including Internet client management, such as web browser, e-mail and news client, Internet client control, etc.

The Security component is responsible for security mechanism of whole system, including subscriber authentication, media authorization, network security, software upgrade security, and

service application security. This middleware service component provides security and authentication API and DRM API to the upper application layer. It provides management functions such as subscriber authentication and identification information management, service application authorization, software upgrade and download authentication, network security policy management, key, token and registration information management, parental control related information management, content purchasing information management.

The Metadata access component is responsible for metadata system access, the metadata presentation and service selection management, and provides metadata access API to the upper application layer. It provide management functions such as metadata system access, metadata delivery management, metadata presentation, such as program scheduler and event catalogue, etc., and service selection management.

The User interaction component is responsible for interacting with end-users, and dispatch events originated by end user to application layer user interaction API to the upper application layer. It provides functionalities such as interaction with end user, via keyboard, or mouse, or remote controller, etc., receiving and dispatching event from end user to application layer, service provider's terms and conditions notification to end user.

The Application management component is responsible for the management of the life cycle of the applications and interaction operations between them. It provides management functionalities such as manage the lifecycle of application, maintain application status, manage application according to control information from user or application providers, manage the resources for applications, etc.

## 2.2. Workflow for IPTV Service

This section will give an overview on basic workflows to support IPTV service. The Figure 2 depicts a typical signaling flow of session setup procedure for initializing IPTV service. This may be accomplished by Communication component as described above. For some commercial IPTV system, there would be slightly modifications on corresponding RTSP headers in order to protect well developed IPTV System. The Figure 3 depicts a typical data flow of media consuming procedure after successfully initializing IPTV service. This may be accomplished by Media Management component as described above. For most deployed IPTV System, the data is carried using MPEG 2 Transport Stream over UDP. Within managed IP core network, the packet loss rate is considerably low.

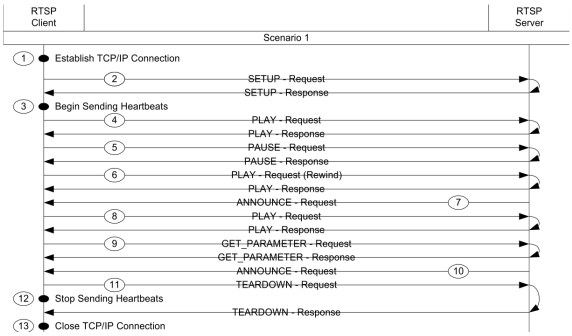

Figure 2 Signaling flow for IPTV

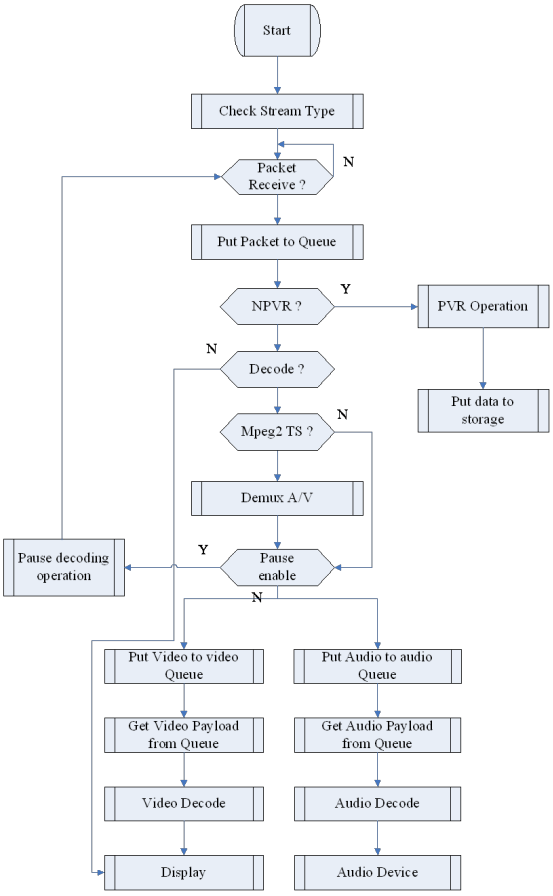

Figure 3 Data flow for IPTV

In order to smoothly control interactions between components within Middleware and provide an unified control interface for IPTV application development, an IPTV Manager is response for providing all necessary functionalities

such as VCR-like operations, user interaction handling, DRM information exchange, etc.

### 2.3. Workflow for VoIP Service

This section will give an overview on basic workflows to support VoIP service. The Figure 4 depicts a typical signaling flow of multimedia communication session setup procedure for initializing VoIP service via SIP protocol. The Communication component is response to provide all required SIP stacks. The Figure 5 depicts a typical data flow of bidirectional media consuming procedure. This may be accomplished by Media Management component.

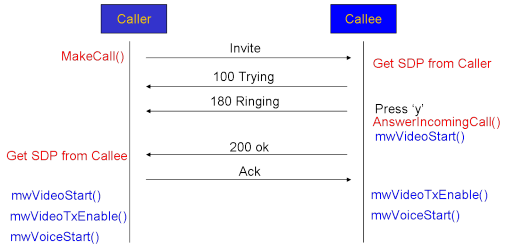


Figure 4 Signaling flow for VoIP

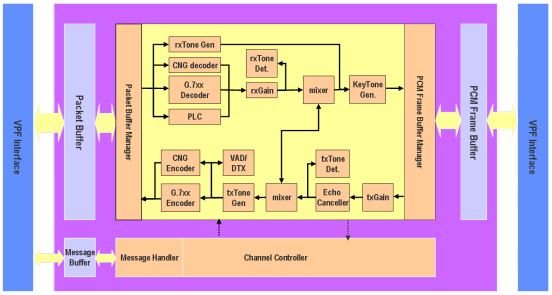


Figure 5 Data flow for VoIP

## 3. Application Scenarios for Triple Play Service Convergence

This section presents a new service scenario which blending IPTV and VoIP Service. The converged service described below is so called TVPhone service, which provides click-to-call and click-to-answer functionality. This enables service convergence between IPTV and VoIP Service, which are two disjointed services. The new service scenario not only provides Caller-ID on TV feature, but also provides unified user experience through single TV screen. It allows services inter work and control one another, e.g. call disposition via TV Remote Control, Live TV content is automatically recorded during a call via PVR.


Figure 6 Interactive Program Guide


Figure 7 Telephony click-to-call

During watching television programs, users can make a VoIP call to friends to talk about plot. Users also can immediately inquire about product information and then make an order by calling the hot line on the advertisement.

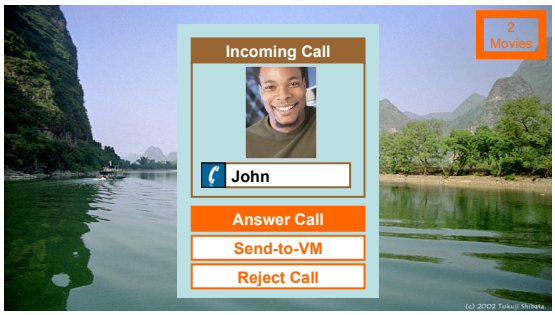


Figure 8 Telephony click-to-answer

When watching television programs, users can handle the incoming call in a more convenient way, i.e. just use the TV remote controller either to answer call, reject call. The OSD will also show the Caller-ID Information (text and/or picture-based) on TV helping users to quickly identify the caller. Live TV content is automatically recorded during a call so that users will not miss any burning plot of a play.

Figure 9 Video Conference

## 4. Reference Implementation

Figure 10 shows the reference architecture of the implementation deployed in the lab. The IPTV Middleware Server is a complete back-office that supports the business operations of modern On-Demand and Value-added television services. The IPTV Middleware System enables service providers to promote, price and bundle their products, and to manage the presentation and sale of product offerings according to a flexible set of business rules. The Video Server manages requests for on demand movies. The Movies On Demand service supports various types of transactional, free and subscription VOD services and provides an end-to-end solution for the service provider. It manages business models defined by the service provider and supports interfaces for multiple back-office billing systems. The Call Server is a SIP-based communications platform dealing with the setup of all SIP calls in the network. A SIP server is also referred to as a SIP Proxy or a Registrar.

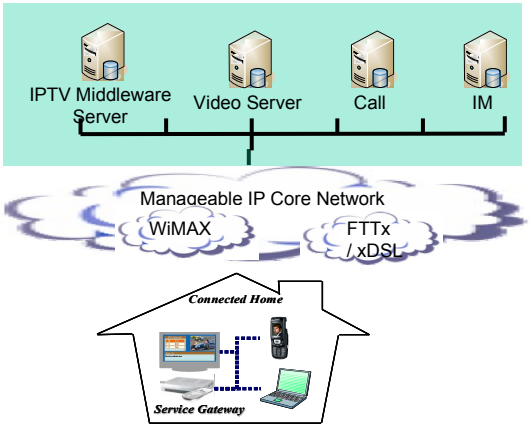


Figure 10 Triple Play Service Convergence System Architecture

Figure 11 Showroom Environment

To evaluate proposed Service Convergence scenario, e.g. TV Phone Service, and Triple Play Middleware, we use a SigmaDesign SMP8634 development board, which is popular Set-Top-Box Reference Design, acting as Triple Play terminal. Furthermore, the Servers are located in ITRI Campus. Media Stream is delivered via WiMAX Network. Figure 11 shows our prototype environment, which is located in the show room of M-Taiwan WiMAX Application Lab (MTWAL), a proof-of-concept laboratory by WiMAX Forum .

## 5. Conclusion

Triple Play is currently a hot topic with service providers. In this paper we presented the Triple Play Terminal Middleware architecture for service convergence. The corresponding logical as well as workflows is discussed. We also present an innovative service based on a Triple Play Terminal Middleware for providing IPTV users with click-to-call and click-to-answer telephony using an existing telephony system. We believe that the described application presents only the beginning in the era of convergent IPTV applications.

Future work will be dedicated to develop more interactive service including TV Chatting, TV Voting, IMS, and service provisioning. Moreover interactive advertising functions will be taken into consideration.

## References

[1] E. Mikoczy *et al.*, "IPTV Systems, Standards and Architectures: Part II - IPTV Services over IMS: Architecture and Standardization," *IEEE Commun. Mag.*, vol. 46, issue 5, pp 128 - 135, May 2008.

[2] C-S. Lee, "IPTV over Next Generation Networks in ITU-T," *2nd IEEE/IFIP International Workshop on Broadband Convergence Networks, 2007, BcN '07,* pp 1 – 18, May 2007.

[3] Adel Al-Hezmi *et al.*, "Enabling Triple Play Services over NGN," *ITI 5th International Conference on Information and Communications Technology, ICICT 2007*, pp 91 - 97, Dec. 2007.

[4] A. Al-Hezmi *et al.*, "Provisioning of an Open NGN/ Triple Play Toolkit and Testbed," *3rd International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities, 2007, TridentCom 2007,* pp 1 - 6, May 2007.

[5] Sohel Q. Khan, *et al.*, "Experiences with Blending HTTP, RTSP, and IMS," *IEEE Communications Magazine*, Volume 45, Issue 3, pp 122 - 128, March 2007.