A ROBUST CLIENT ARCHITECTURE ON ANDROID TO CATER END-2-END REAL-TIME CONTENT MANAGEMENT AND PERSONALIZED IPTV SERVICES TO MOBILE INTERNET DEVICES

Suman Kumar S.P[1] and Vijay Anand[2]

[1]Centre of Excellence (CoE) Division, Aricent Communications, Sigma Tech Park
Suman.Prasanna@aricent.com
[2]Whitefield Main Road, Bangalore-560066.Karnataka, India
Vijay.Anand@aricent.com

ABSTRACT

Digital Video Content is now pervasive and services like IPTV are gaining traction on hand held devices too. Android, a software platform from Google, pioneering middleware and applications for handheld devices does not offer support for content management services by design.

*Motivated by these observations, we propose an aggregate model software platform to offer content management and value added services (VAS) like IPTV on android. The outline of paper is as follows: First, we discuss few related works to investigate and analyze industry-wide content management and delivery practices. Second, we discuss few characteristics of android architecture and detail the constraints associated with middleware to offer such VAS. Consequently, we describe the proposed software platform which addresses key inherent challenges like Efficiency, Extendibility and Security while achieving efficient content delivery. The Module Interconnection Architecture can be visualized to be constituted by three basic building blocks, namely: **VAS Content Management framework, IPTV framework and Application Services framework.** The architecture employs a different perspective of classifying content based on applications. We point out the need and rationale behind such a classification to correlate in an end-2-end IPTV architecture. We show the benefits of using the Application Services framework to ensure smooth and swift portability onto various flavors of android. Finally, we conclude by detailing some use-cases indicating content management services like Electronic Service Guide (ESG) on Android..*

KEYWORDS

Content Framework, Android, MID, Middleware, IPTV, Value Added Services

1. INTRODUCTION

With the constantly growing mobile-subscriber base, existence of a vast variety of handsets and market-value of Value Added Services, the mobile industry has long felt the importance of providing content to the user that they actually want and that is the most appropriate to the device they have. The mobile devices, as compared to the usual PC systems, have smaller screens, limited primary and secondary memories and a limited computing power. To add to the complexity is the wireless bandwidth which is quite less as compared to its wired counterpart. On the other hand, android making an impressive impact amongst the early adopter consumers is ensuring mass market success precisely due to "The Cost Factor with Appealing Entertainment Experience" [1]. However, one challenge has remained i.e., the variety of content that end-user is interested in and in a form of his interest [2]. The content ranges from as trivial, static and small as ring-tones, mobile-applications to as novel, dynamic and huge as actual web-

DOI : 10.5121/ijngn.2010.2307

pages, content-feeds, multimedia content and streaming videos. Addressing these constraints not only requires efficient and systematic content management at the client, but also demands for an intelligent, efficient and most justified delivery of the same as shown in fig 1.

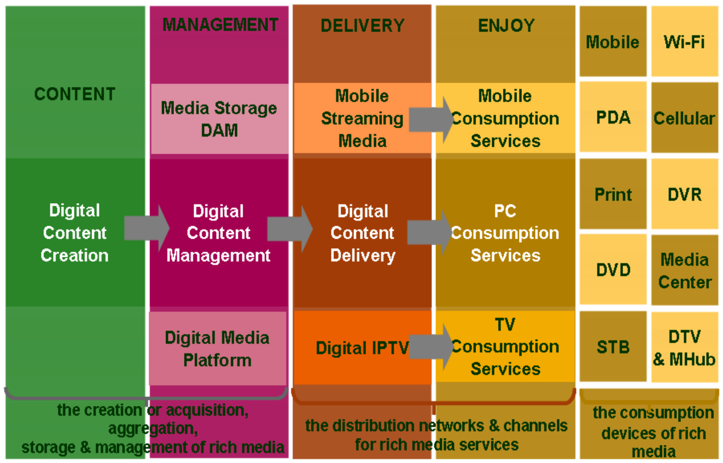

Figure 1: Digital Content Delivery Model

In this paper, we propose novel software architecture on Android to address above seen limitations. The architecture tends to be futuristic by using a “Module Interconnection Architecture” design which provides the required flexibility to add new features like Video Voice Call. Ex: This can be achieved with slight extension in Content Management Framework. This paper also provides different architectural views like “Meta Architecture”, “Conceptual View”, “Logical View” to stimulate further research in this direction. In an attempt to demonstrate the evaluation of this architecture, we have demonstrated few common user scenarios along with implementation guidelines like MSC (Message State Chart). In an effort to aid designers considering this architecture as reference, this paper also suggests few design pattern methodologies which is best suited for the implementation to ensure robustness and acceptable performance in the end product developed.

2. Design Approach

In this paper, we envision to provide “The Big Picture” of the content management system extending android. We detail few aspects of content classification, IPTV terminologies and architecture aspects of Android. A literature survey on content management reveals content classification methods for handheld devices can be based on a number of factors ranging from size to format of codecs to pricing-model to DRM applied and so on. However, early investigations suggest best practices in content classification to be based on a combination of two significant factors. Firstly, the subscriber-segment the data addresses. Every kind of data is not meant for everyone. Secondly, the kind of transformation the content may require for efficient and logical delivery. In view of this, we propose the classification of content into four categories: Trivial static web-pages, Text-based Information Rich Content (like RSS), Multimedia content playback and Multimedia content delivered over network As most VAS will be delivered over network, this paper primarily focuses on the last category. Our VAS Content Management Framework deals with Content Reception, Content Adaptation and transforming the content to user desired format. Also, services like IPTV, ESG based on well accepted and proven standards like DVB-H, we have defined IPTV framework with components addressing issues like data acquisition, transcoding and so on to ensure robust media management. On other hand, android is based upon an opaque IPC model. Applications, Services expose to the system their functionalities, and at runtime, other applications can request these functionalities. Essentially, the platform provides a managed and secured late code binding. This model is specifically useful in the interaction between our software platform and Application Services Framework within Application framework on Android to ensure easier portability. The associated software architecture is proposed in the section 3 along with the brief description of the software modules.

3. Proposed Content Management Platform on Android

Literature survey indicates the following critical factor to be considered while offering VAS in mobile devices [3]. As voice revenues decrease, MNOs are increasing and looking towards data and Value Added Services for increased revenue and as a key differentiator. Following are key factors that operators are considering while offering VAS to end consumers.

- **User Experience:** A compelling user experience is the key to increasing the usage of VAS.
- **Personalization:** Allowing the user to personalize the service increases usage.
- **Location Based Services:** Understanding where the user is located can allow for increased personalization, meaning, and usage. Some services are only effective when location-aware.
- **Cost / Payment Methods**: Premium SMS, pre-paid and post-paid will be supplemented with advertisement supported and opt-in (giving up personal information).
- **Business Model (Carrier):** The carriers will drive low prices / margins from the VAS providers – too much competition from other providers and too few carriers.
- **Business Model (Content Providers):** Another model is to go after the various content providers (Times of India, etc.) and bring them a new channel for their content.

Mobile operators tend to offer a wide spectrum of data applications. These can be divided into categories reflecting their nature of being communication-centric, transaction-centric or content-centric. Communication-centric applications mainly cover person-to-person communications over mobile networks. Transaction-centric applications cover financial and monetary transactions over mobile networks. Content-centric applications cover the delivery of content over mobile networks. In line with the objectives of defining the Mobile Content Framework principally covers content-centric services and applications. The Content includes vast, rich catalogue of static and dynamic content for all types of mobile content applications, including content for music, news, games, sports and more. Considering these key essential factors for a content management platform, we propose a software platform as shown in Fig 2:

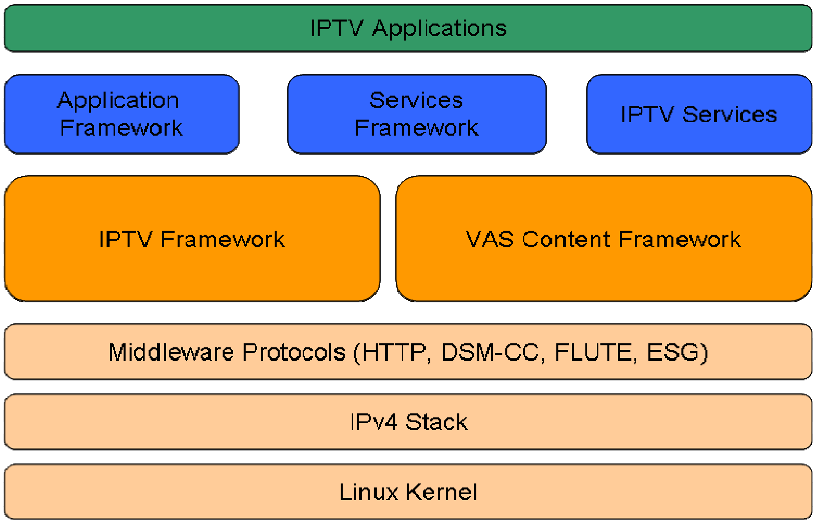

Figure 2: Proposed Content Management Software Platform

a. **IPTV Applications:** Electronic Service Guide (ESG), Personal Video Recording (PVR), Internet radio, Photo gallery, Web Browser, Media Player and so on.

b. **Application Services Framework(ASF):**The Application Services Framework is responsible for responding to events posted by middleware and interacting with Android Application framework components. The ASF reacts with input and external events, which includes User input, Connection & application status for IPTV applications.

c. **VAS Content Framework:** The VAS Content framework provides API's that are easy to adopt, yet capable of creating advanced applications. The framework is responsible to provide purely network services and transparently manage the detection of new services.

d. **IPTV Framework:** This middleware entity is responsible for providing all IPTV services onto Application Service Framework. In-turn, ASF handles the requests, processes them and sends it to corresponding application through the android application framework. It consists of ESG Engine, Download Manager, protocol stacks like FLUTE etc.,

4. Architectural Design Aspects

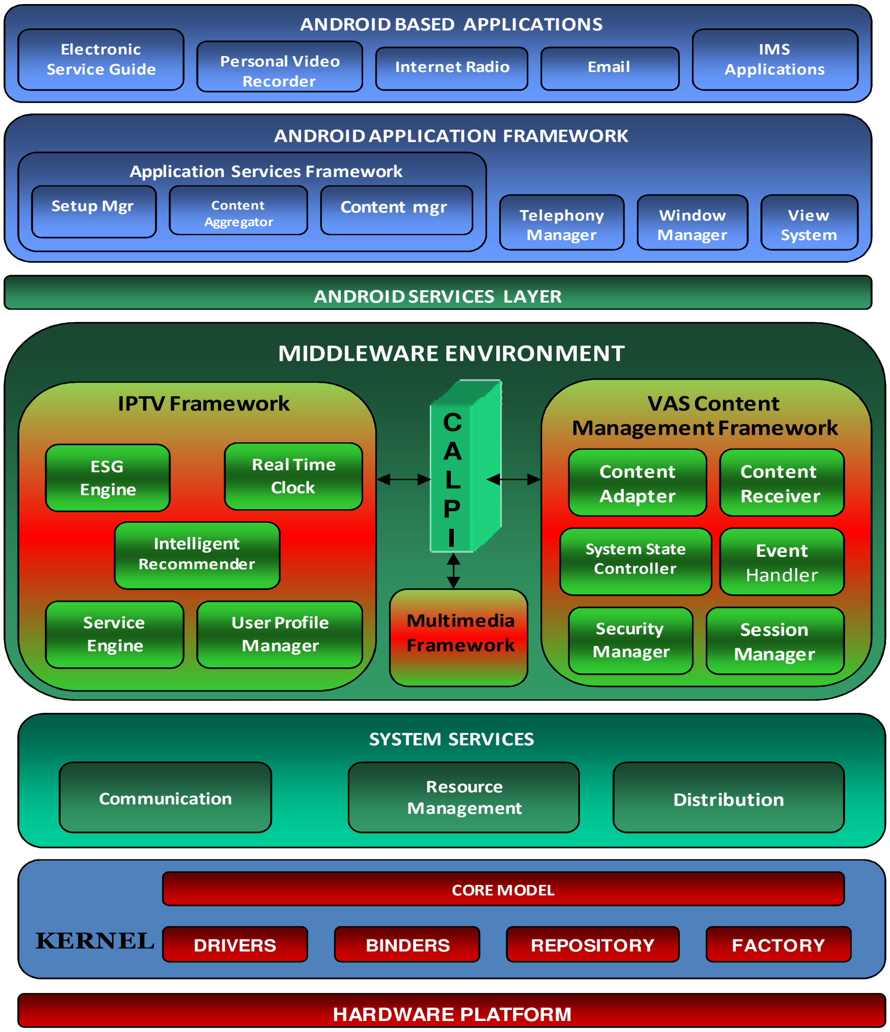

Figure 3: Conceptual Architecture View

The rationale of the android platform is “breaking down the barriers to building new and innovative services and applications”, particularly in the field of content management and peer-to-peer networks. Technologies like IPTV have specifications in order to support rich multimedia content in a very expressive manner on complex communications. It allows us to give semantics to the communication without taking into account the message content and the underlying platform. Moreover, a software platform like android allows us to build both simple interaction (i.e. request-response) and more complex ones (i.e. services and binder interaction useful for providing service in a client-server fashion) [4, 5]. Hence, leveraging the flexibility offered by DVB-SH standard for IPTV services, with the power of the ANDROID platform, in our opinion, brings a strong value in the development of innovative applications based on content management and peer-to-peer paradigm. Furthermore, it is worth to mention a very important feature of Android: the application process’s lifetime is not directly controlled by the application itself. As a matter of fact it is determined by the system by analyzing what parts of the application the system knows are running, how important they are for the user and how much total memory is still available. This is particularly important for our Content Abstraction Layer API (CALPI) which interacts with IPTV, VAS and Packet Video Multimedia framework of android and the Android Services framework to decide on the resources to be allocated for a particular application. With such interesting features offered by Android and exciting features of IPTV, the design of the proposed software platform (as shown in fig. 3) intensively focuses on extendibility, security and efficient content delivery as three key parameters impacting the overall system the most.

4.1 VAS Content Management Framework

This framework is responsible for providing the middleware functionalities like establishing the session, notifying about events and detection of new service like Weather Stock updates and so on. It supports both “push-model” and “pull-model” to handle the content with the CALPI. Fig 4 and Fig 5 shows the working execution model of various components constituting the VAS content management framework.

- **Ex (push-model):** On detection of update in existing service channel (s) a notification screen should be shown on UI with options to select/deselect the feature. Ex: A Remind on a particular program may be triggered from this framework to the application. In this case this framework triggers a push notification to the application. Another use-case could be Dynamic update of content like Stocks, Weather and so on.
- **Ex (pull-model):** Upon selecting the respective menu (Say ESG Menu), the contents of channel/program information should be shown to the user. This case the application triggers a pull for data from this framework.

Event Handler (EH): is responsible for handling events in the system. It handles both internal and external events.

- Contains a priority queue to ensure all the simultaneous events are captured. Upon receiving any event the corresponding components registering for the event will receive a notification.
- A Mediator pattern may be used.

Use Case: An event of low battery then irrespective of component all the components will be intimated about the event. Ex: All application generated/ internally generated events like alarm are routed to CAPLI which decides which UI screen to display. All the system generated events are either processed within the service layer or the critical events are alerted directly to the CAPLI with highest priority.

Security Manager (SM): To enable security, the Security Manager gets interfaced with the system state controller for providing user authentication, data encryption and authentication.

Content Receiver Manager (CRM): This component is primarily responsible for downloading the content, accessing the content from the specified location and storing in persistent storage.

- An observer pattern may be used with this as source.

Use Case: It interacts with the underlying System Services Layer to establish the transport data connection (Ex: HTTP) and start receiving the data (Ex: in the form of XML or RSS feed). Upon successful completion, a notification will be generated to the Content adapter indicating the data is ready. This component will need to register for the hardware found event with EH to start triggering the download of data to the storage medium.

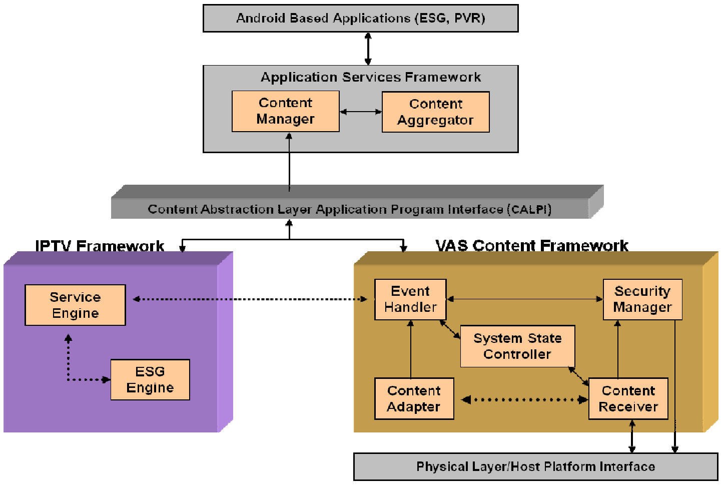

Figure 4: Execution Architecture on detecting new VAS Client

Content Adapter (CA): is responsible for providing the requested data to the CALPI and the ASF.

- An observer behaviour pattern can be used to get the notifications from the CRM. Within the implementation, a bridge or decorator pattern can be used to map the interfaces with the CALPI layer.

Use Case: However, in case of dynamic update like stock information, the application can also request CALPI to get the data. CAPLI being a Command behaviour pattern will automatically get the update of the requested data event.

Session Manager (SM): maintains information pertaining to currently available and next sessions. Initially it finds out the location of hierarchy file for selected IP platform and gets details of services available on the IP Platform. It regularly checks for any updates to the details of the sessions.

System State Controller (SSC): is the heart of this framework maintaining the overall state machine of the system.

- Any events received from the IPTV framework/underlying layers will be sent to respective components depending on current state of the system.
- A mediator pattern can be used to forward the event to respective module to handle the same.

Use Case: A power-off should trigger all modules irrespective of system state to exit gracefully as it's a system event.

- A Fly weight design pattern to implement this module. An access control list is a concept to enforce privilege separation i.e. to determine appropriate access rights to given resource depending on the source identification.

Use Case: When the IPTV App starts up (say Child lock feature in Media Guide), the user is automatically requested to authenticate himself by choosing the desired level of security. If authentication fails, the user is informed of the result and the application exits. With Security Manager, user identity is established and encryption on the audio and video streams can be performed with the defined software modules and according to the user's security level preference. The SM module provides a facility that could be integrated within a DRM Agent.

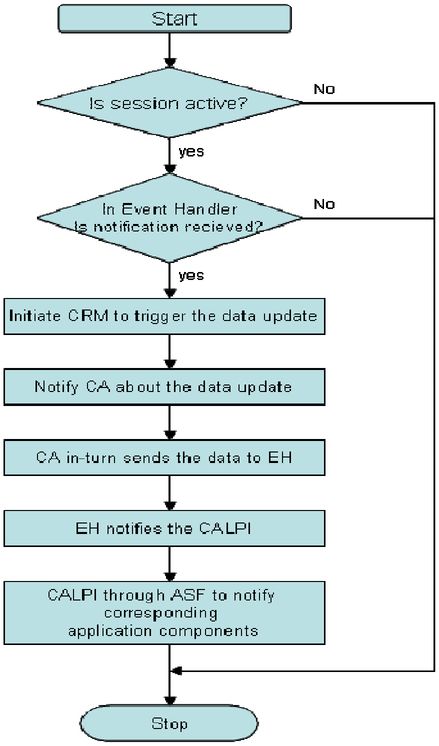
Start
Is session active?
No
yes
In Event Handler
Is notification recieved?
No
yes
Initiate CRM to trigger the data update
Notify CA about the data update
CA in-turn sends the data to EH
EH notifies the CALPI
CALPI through ASF to notify corresponding application components
Stop

Figure 5: Dynamic Content Update Execution Mechanism

4.2 IPTV Framework

An electronic service guide (ESG) is an on-screen guide to schedule broadcast television programs, typically with functions allowing a viewer to navigate, select, record, remind, and discover content by time, title, channel, genre and so on. ESG also provides acquisition information, which is needed for rendering live streaming services and downloading broadcast services [6, 7]. IPTV framework is responsible for providing all the middleware functionalities mentioned above to the Application Services Framework. It is composed of the following modules described below

ESG Engine: This module will maintain all up-to-date ESG (in XML format) information. ESG engine tracks the version information of the ESG files and whenever an update is found, it will initiate for a download. A persistent storage mechanism will be used to store ESG information. It is also responsible to maintain information related to:

- Available Services, their schedule and related content
- Service Provisioning information for purchasing an individual or bundled services.
- Service Acquisition related information to access the service.

It also contains XML Parser to parse ESG files and extract information as per XML Schema given as per the Operator specification.

User Profile Manager (UPM): UPM is responsible for storing and retrieving user profile information in the setup table of persistent storage. UPM is also responsible to update the rating of each programmes list stored in content table in case the user profile is modified.

Real Time Clock Component (RTC): is an intermediate component in multi-media flow graph. RTC is responsible to provide a real time clock synchronized with the host.

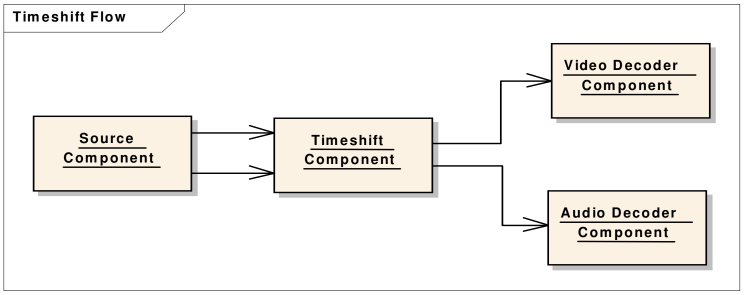

Figure 6: Real Time Clock Component Data Flow

- Real Time Clock Component works in two modes. When time shift is disabled it receives data from source component and forwards it to downstream components. When time shift is enabled, the received data goes through temporary backing files to enable delayed play.
- Real Time IO agent, a part of Real Time Clock component, implements file buffer operations. The time shift component provides optimized IO operations by using a Continuous Block Buffer (CBB) module, which optimizes the file IO operations, based on the fact that Block-Reading and Block-Writing are more efficient.

Intelligent Recommender (IRM): IRM is responsible to provide a list of recommended programmes to CALPI. For features like automatic recording it calculates the next automatic recording taking into account the already scheduled recordings and the space on the media.

Service Engine (SE): is responsible for IPTV Service Detection and configuration. When the application starts, it first creates Service manager which in-turn creates other modules of the middleware. Service Manager is responsible for getting the details of the platforms available.

- Provide interface to application for displaying Service List (Program channel List).
- Upon service selection, it requests SDP from ESG Engine and parses it and sends it to Multi Media Framework for playback. The information includes format of a/v data, configuration data to initialize decoders, IP channel details and so on.

The Service Controller (SC) component within SE engine shall work upon the service context block for the call to determine the list of active services, current active service and the trigger received etc. to determine whether it is a new service or not. If it is a new service, then it shall check to see the action taken based on the list of the active services. It shall indicate one of the actions below:

- New service not allowed.
- New service is queued till existing service is terminated.
- New service is allowed to be executed, with certain additional information.

4.3 Application Services Framework

Content Manager (CM): This module has the interface implementation for accessing and managing contents. For content specific functionality, respective components in the IPTV framework will be invoked. This module also keeps track of interface for various content providers and aggregators.

Setup Manager (SEM): This component is mainly responsible for the hand-shaking mechanism with the Host Abstraction Interface Layer (HAPI) to the requested hardware device. Upon success, the HAPI generates a notification to the event handler which in-turn passes back to the Setup Manager itself for further actions. If EH is implemented using a Chain of Responsibility design pattern then this event notification will be handled by the Content Receiver Manager. If EH is implemented using Command pattern internally with Composite pattern then a separate request should be triggered within the composite class implementation to trigger the download to CRM. In any case, upon success of the hand-shaking mechanism, the CRM will be notified and a content download will start.

Content Aggregator (CA): The main responsibility of content aggregator is to gather the content from various sources and aggregate the various input. Whenever it receives a request from an application like A/V editor, the content from various sources is received from VAS, IPTV framework will be transformed to the form expected by the user. It interacts with CALPI to cater to application services

5. USECASE SCENARIOS – IPTV SERVICES ON ANDROID

5.1 Push Model - Dynamic update of Stocks/Weather onto ESG

In order to demonstrate various content management services, we have created the test bed to simulate our architecture. If the user has set a periodic update of information about stock, weather and other value added services on his device, typically as shown in fig 7, the content receiver manager will be invoked to push the updated data onto the UI. The use case was simulated to verify on Android based device. We have performed an experiment implementing our architecture on ARM based hand held device running on Android. The objective of our software platform is to deliver converged services of IPTV on android based hand held devices. Our architecture augments that significantly by adding personalized IPTV services onto IPTV framework. Being such a modular and easily portable architecture encompassing VAS content management, IPTV and ASF will almost certainly result in multiple vendors.

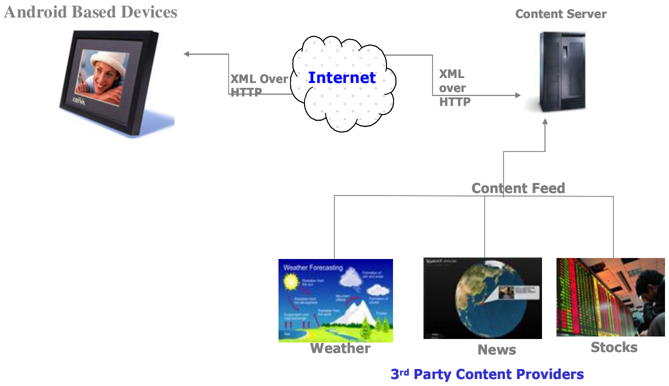

Figure 7: Dynamic Content updation on android based devices

A scenario for personalized IPTV Service used in this test is given shown below:

"When Clock strikes 8:00:00, Content Framework is turned on. When content Framework is on, one of the new stored photos replaces the old one periodically." The execution procedure of the test scenario can be described as:

1. If a new scenario is delivered, it is analyzed.

2. An event of the analyzed scenario is registered.

3. At 8:00, the registered events (Content Framework) are triggered.

4. When an execution command is delivered from content framework, it finds a service in Common Description Repository and requests it's execution to an appropriate s/w module.
5. Content framework parses a command transmitted from and requests it's execution.
6. The content framework device notifies its status changes as a result of the execution.
7. Content Framework notifies that it has received the content from the specific network server and parses the XML based content. As the status is changed, the next event which triggers an automatic change of the personalized picture frames.

5.2 Pull Model – User Selects ESG Menu on Android

If a user wants to watch a list of programmes that will be broadcasted in next week, then on selecting ESG guide, following are the sequence of actions triggered in the system. Fig 8 shown below illustrates the same.

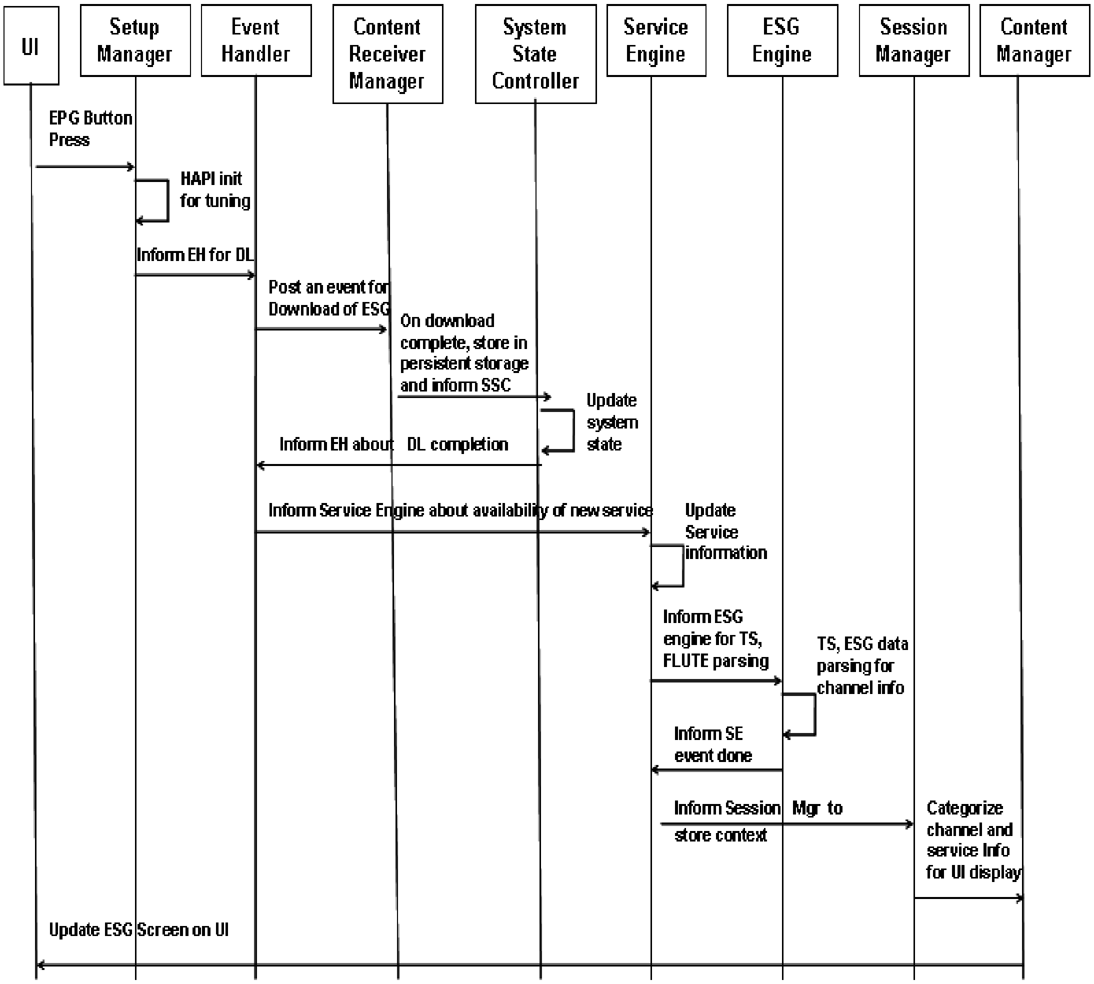
UI
Setup Manager
Event Handler
Content Receiver Manager
System State Controller
Service Engine
ESG Engine
Session Manager
Content Manager
EPG Button Press
HAPI init for tuning
Inform EH for DL
Post an event for Download of ESG
On download complete, store in persistent storage and inform SSC
Update system state
Inform EH about DL completion
Inform Service Engine about availability of new service
Update Service information
Inform ESG engine for TS, FLUTE parsing
TS, ESG data parsing for channel info
Inform SE event done
Inform Session Mgr to store context
Categorize channel and service Info for UI display
Update ESG Screen on UI

Figure 8: Logical Architectural View

5.3 IPTV Content with DRM

Digital Rights Management (DRM) [8] is to enable distribution and consumption of the digital content in a controlled manner. Using DRM on Android will help the user in preventing unauthorized users from accessing the shared contents. For instance, if the user wants the share the media file to another android device, the user selects the particular file from the media gallery on the client and the client sends the image captured by the device to other application.

To prevent this content from being accessed by any others than the intended user and this is encrypted by a forward delivery feature of DRM. The security manager of VAS content framework handles this feature efficiently.

6. EVALUATION OF PROPOSED ARCHITECTURE: CONTENT PROVIDERS PERSPECTIVE

A user can select and view a desired program in Electronic Service Guide (ESG) for programs that are distributed live according to a time schedule. In existing broadcasting systems, the service information (SI) created by content provider (broadcast station) are collected at Service providers' SI Collection and Delivery Center. Service provider adds the ESG data and subsequently the content is returned to each broadcast station and distributed as ESG data. In this method, service provider requires a function that can compile SI of each content provider and distribute them to a user (terminal) as an ESG data. Unlike existing broadcasting systems, the content provider-specific SI uploaded to the SI Collection and Delivery Center do not have the physical constraints of service area and hence the maximum viewer target can be extended to the entire world. In such a use-case, it is certainly not practical to distribute ESG data covering all programs to all users as it would require vast volumes of SI data. Therefore, an optimal approach would be to embed a system to screen the user profile data (such as language and area), user preference data (such as content genres), and terminal attribute information (such as playback file format) and create ESG content in accordance with distribution area and individual user preferences. Such requirement of service provider to support user-specific or area-specific ESG is handled efficiently by User Profile Manager and Setup Manager Components in our architecture. This enables a user to view widely diverse content that was never provided by conventional broadcasting services. This feature is of significant advantage to service provider as there is no need to return the final form of created ESG data to a content provider. Instead, the ESG data can be directly distributed to a user from an SI Collection and Delivery Provider taking full advantage of the IP network.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we presented novel service architecture to cater content management and delivery to android based hand held devices. Though android has vast potential for content management, existing media management techniques are ineffective to cater services like ESG of IPTV. That said, the paper outlines and addresses efficient delivery of content management services on Android. The proposed architecture envisioned to address key parameters like Extendibility, Efficiency and Security by employing logical interfaces among different frameworks defined. A proof-of-concept extending android platform was illustrated to evaluate the feasibility of this architecture to translate into product deployment. Efficient content management and service delivery being essential to reconcile the characteristics among handheld devices, an efficient functionality based framework classification employed within our architecture also accommodates wide spectrum of applications like PVR and so on. In addition to showing the effectiveness of this approach, we have discussed and investigated several user scenarios to identify the characteristics and competencies required for successful product deployment. A related aim is to inform the scientific community about design in such new environments by providing MSC (Message State Chart), Flow diagrams and best fit design pattern. Finally, evaluation of the said architecture reveals that a successful debut has been made to cater revenue based content management services on Android.

Future work includes support for advertisements and operator specific content aggregation on ESG. It is yet another aim to incorporate gesture based program recommendation based on content information specified by the operator.

REFERENCES

[1] A. I. Wang, C.-F. Sørensen, and T. Fossum. Mobile Peer-to-Peer Technology used to Promote Spontaneous Collaboration. In The 2005 International Symposium on Collaborative Technologies and Systems (CTS 2005), page 8, Saint Louis, Missouri, USA, May 15-19 2005.

[2] T. Horozov, A. Grama, V. Vasudevan, and S. Landis. MOBY - A Mobile Peer-to-Peer Service and Data Network. In 2002 International Conference on Parallel Processing (ICPP'02), pages 437–444, 2002.

[3] Recommendation ITU-T Y.1910 (09/2008), "IPTV functional architecture," ITU-T, 2008.

[4] ETSI, "TISPAN; service layer requirements to integrate NGN services and IPTV," TS 181 016 V2.0.0 (2007-11), ETSI, Sophia Antipolis Cedex, France, 2007.

[5] Enck, W.; Ongtang, M.; McDaniel, P., "Understanding Android Security", IEEE Security & Privacy, 7(1), pp.50-57, 2009.

[6] Tao Wu, Sadhna Ahuja, Sudhir Dixit: "Efficient Mobile Content Delivery by Exploiting User Interest Correlation"; IEEE International Conference on Multimedia and Expo, Taipei, 2004.

[7] S. Long, R. Kooper, G. D. Abowd, and C. G. Atkeson. Rapid prototyping of mobile context-aware applications: Cyberguide case study. In Mobile Computing and Networking, pages 97–107, 1996.

[8] Kaijian Xu, Manli Zhu, Daqing Zhang, Tao Gu: "Context-Aware Content Filtering & Presentation for Pervasive & Mobile Information Systems"; Proceedings of the 1st international conference on Ambient media and systems, Quebec, Canada, 2008.

[9] V. Marques, R. Aguiar, C. Garcia, J. Moreno, C. Beaujean,E. Melin, and M. Liebsch. IP-based QoS architecture for 4G operator scenarios. IEEE Wireless Communications Magazine, June 03.

[10] Mobydick: Mobility and differentiated services in a future IP network – Final report," http://www.ist-mobydick.org, April 2004.

Authors

Mr. Suman Kumar is a part of Research Group in Aricent Communications involved in building proof-of-concepts from design to execution focusing primarily on NEXT GEN Architectures for Mobile handheld devices. He and his engineering team drive various technological roadmap activities, engages with academia, file patents and publish papers in competitive conferences and journals.

Mr. Vijay Anand is Director - Engineering in Mobile Terminal Products Division of ARICENT Technologies Limited. In this capacity, he directs research activities that drive the state-of-art in the design of next-generation wireless/Multimedia technologies. He has 2 software patents to his credit and has chaired several technical sessions. He has given tutorials on "Mobile Device Architecture: Present and Future", "Digital Living Network Alliance", "Fixed Mobile Convergence" at various International Conferences and symposia like NCC 10, NTMS 07, IEEE Wi-MAX. He is a sought after panelist, keynote speaker, and architect on next generation mobile terminals.