BRINGING A CENTRALIZED SERVICE DELIVERY PLATFORM FOR ADVANCED NAVIGATION AND SEARCH TO COMPANION DEVICES AND THE EXISTING STB

Steve Tranter
NDS

Abstract

Cable technologists are facing pressure to ensure their networks and in-home experiences include compelling and intuitive access to the most content possible on as many devices as possible, while relying on the limiting capabilities of deployed set-top boxes (STBs). Guides for cable systems are generally restricted to the abilities of the least capable STB deployed, due to the increased effort it takes to integrate and support variants of the user interface (UI) across many STB types. Thus the guide is often overlooked in favor of limiting the integration effort.

This paper will present technology considerations for decoupling the guide appearance (skin) from the underlying STB architecture, such that updates and re-skins of the user experience can be quickly achieved and optimized for the STBs on which they are running. Adopting this service layer approach also enables the operator to move some of the complex UI features, such as search, from the limiting processing environment of the STB to the headend, simplifying the STB integration further. Migrating the guide functionality from the STB to the headend not only extends the capabilities of the STB, but also provides a centralized architecture for supporting companion guides and services on consumer purchased devices, such as PCs, iPads, etc.

This paper will review the various service layer approaches that can be adopted within the STB software architecture, such as EBIF (XML) and OnRamp (Java), highlighting the advantages and disadvantages of each. Furthermore, the data transport architecture will be discussed to show the infrastructure and related standards required to centralize guide functionality in the headend (DOCSIS and DSG implementations, etc.).

Finally, considerations of the centralized metadata headend will be explored to highlight the technology options that can be adopted to extend the guide and MSO service offering to companion CE devices on which the MSO wishes to have a presence. This includes architectural decisions regarding the application server environment and the inclusion of a service delivery platform to provide standard, pre-certified application programming interfaces (APIs) into the TV ecosystem for application developers, such that applications can be contextual to and control the TV viewing experience (channel change, DVR recording, etc.) as well as

taking advantage of in-home networking standards, such as DLNA.

INTRODUCTION

Since the first deployment of digital STBs the guide and user interface have been tied to, and hence limited by, the STB on which they are running. The user interface has evolved since its first deployment to cater to new services being deployed by MSOs, such as video on demand (VOD), digital video recording (DVR) and simple interactivity through add-on user-agents (EBIF). Every new service and corresponding user interface enhancement has had to run on the same STBs that were originally deployed. As such, each new user interface enhancement has been implemented as a disconnected application that independently provides an interface to the STB's new feature and not to the overall service being offered by the MSO. For example, this is seen in separate user interfaces for linear, VOD and DVR content, even though the same content can be found across all three.

Further, guide user interfaces have been limited by the hardware and networking capabilities of the STB platforms. Because of limited memory, processing and graphics capabilities, most legacy guides are built in native C code using 640x480 resolution screens, with little to no animation, limited color palette, and basic video scaling capabilities. These traditional guides have limited networking capabilities for providing real-time data updates and therefore limit the amount and frequency of guide data updates that can be accessed by the STB. Adding new features to the guide requires lengthy development and integration cycles. As a result, a great deal of integration work is required for every new guide feature and new STB model that is deployed. This is necessary to ensure that the guide and supporting data run correctly on every STB and with each disconnected application that is resident on these STB platforms.

The legacy STB dependency that exists today has resulted in:

- Restricted user experiences, dated by 10 year old technology; and
- Complex development, integration and testing cycles, requiring excessive manpower and long delays in getting even minimum guide changes to deployment.

Now, with the proliferation of broadband connectivity (DOCSIS 2.0 and 3.0) to the home, the STB restrictions described above are being eroded away. Having a managed broadband connection means that much of the data storage and processing that has traditionally been performed on the STB can now be performed in a centralized data center and the resulting information delivered into the home when required. In addition, not only can the data center support guide services for DOCSIS-enabled STBs, but also any broadband IP connected device, such as PCs, tablets or mobile devices. Thus, enhancing the way an MSO promotes their services and providing control to viewers through companion devices.

Figure 1: Example Guides for Broadband IP Connected Devices Supported by a Centralized Data Center

This new cloud-based architecture provides greater efficiencies for guide deployment – five months to deploy an iPad guide application compared to 18 months for a legacy STB guide. These dramatic development cycle reductions are as a result of eliminating much of the regression testing that is required for legacy STB guides. As a result, the development staff needed is also reduced considerably; typically only 10 developers are used on an iPad application compared to 300 for a legacy guide across multiple STB models.

ARCHITECTURE

MSOs are gravitating towards cloud-based architectures for video navigation and delivery, which enables them to keep up with the accelerated product cycles required to stay competitive. As more and more functionality is moved from the STB to the cloud, Service Delivery Platform (SDP) infrastructure is being utilized to enable these cloud-based services. SDP architecture provides the benefits of delivering advanced guides and services to companion devices as well as integrating with the existing MSO headend to provide improvements to the guide on DOCSIS enabled STBs.

The SDP architecture is a multi-tier, web service-based platform architecture that integrates with MSO's TV headend infrastructure, as well as new Internet-based social networking, enhanced data, and content providers. Internally, SDP architecture consists of a web-services presentation tier, an aggregation tier to Internet-based services and an adaptation tier to existing operator backend capabilities, which includes bridging components for certain legacy environments.

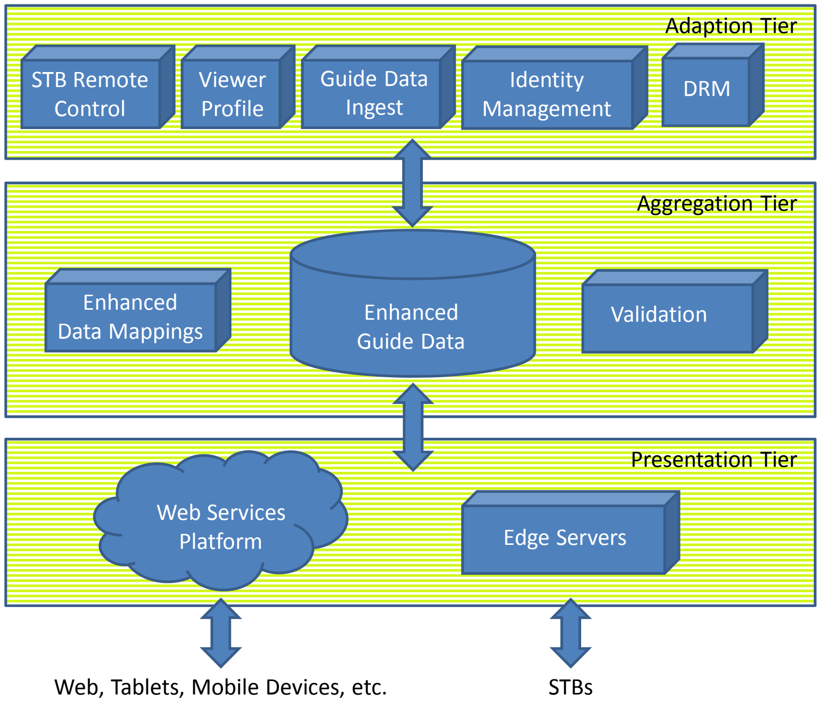

Figure 2: Centralized Service Delivery Platform Architecture

PRESENTATION TIER

The presentation tier is a standard Web services platform (WSP) that can be deployed in the cloud or via edge servers in an operator network. It includes security via standards such as HTTP and OAuth. The user authentication and authorization is tightly integrated with the operator's identity management systems via the adaptation tier, permitting operators to further control which web services are available to which applications. The presentation tier also includes monitoring and analytics such that operators can be aware of the usage of their services across different applications, devices and demographics.

When a device requests information through an SDP architecture web services API, the SDP can use service tokens to identify device type and user identity, which are used to filter service metadata and content references appropriately to the application, user and/or device type. Service accessibility rights are exposed to the presentation tier either through the adaptation tier, which utilizes the operator's own identification management systems to determine application or user profile information, or through the aggregation tier to access similar third party account information. Applications using the presentation layer may leverage this service accessibility information to modify the user experience. For instance, by determining IP address and geo-location, an application may act differently when accessed within the home or outside the home.

AGGREGATION TIER

The aggregation tier provides the majority of the improvements to the SDP-delivered user interface, compared to that of legacy STB guides. The process is predominantly focused on aggregating the metadata for all of an MSO's services, enhancing the metadata relationships and validating the resulting relational data structure. The result is a superset of metadata that can be queried, filtered and utilized by guide applications on the various client devices.

In the aggregation tier, operator-specific metadata, user profile and identity information are aggregated with third party enhancements such as additional content references or social network linkages. This enhanced metadata may be available from public sources on the Internet or through federations between network operators and other service providers. By aggregating metadata, the third party information may be used in a contextually appropriate manner, for instance by providing references to critical reviews on Rotten Tomatoes, to a YouTube video for a live linear program, or a viewer's Internet identity, on sites such as Facebook.

The aggregation tier enables the MSO's guide application to aggregate access to content from a variety of sources (linear, VOD, DVR, over-the-top (OTT), etc.) into a single, coherent guide experience. This means that the guide can now present a wealth of information for content from a variety of sources, providing greater choice and satisfaction to viewers. Also, by storing this guide information in the headend, STBs no longer need to store program information beyond the next day or two, which frees up memory in the STB, for caching other information. One way this can be utilized is for the STB guide to dynamically request and cache images and data in advance, related to the current guide view, such that guide screens can be quickly populated, with no delays as the viewer navigates the guide.

The aggregation tier performs much of the data manipulation, searching and filtering that are traditionally performed by legacy STB guides. In addition, the aggregation tier generates relational links between the various sources of content, such as linear, VOD and DVR as well as enhancing the relational links beyond those defined in the ingest process. For example, relating content by genre, cast and crew, etc. In addition, the aggregation tier provides the ability to associate content metadata with external web sources, linking the ingested, structured content identification mappings to external metadata sources, such as poster art, reviews, biographies, etc. As long as the source for this supplemental data is trusted, this provides a controlled method by which to enhance the information available in the guide, while maintaining a structured format that maps to the source of the content.

The validation role of the aggregation tier is important to ensure the guide data always directs viewers to the correctly associated content source and that any supplemental metadata is correctly associated with the content. For example, Jack Black is assigned as part of the cast for the 2005 version of King Kong and not the 1933 version. This is in addition to ensuring regulatory information is present, such as parental ratings and audio descriptions. This role is predominantly automated, based on pre-defined rules and checks, which ensure that the resulting guide metadata being offered to client applications is complete and accurate.

By centralizing this process in an SDP architecture, it makes the task of adding new data sources and mappings to new content sources much more efficient and less problematic than if it were being done within the guide application on the STB.

ADAPTATION TIER

The adaptation tier provides the bridge from the operator's TV headend into the SDP architecture. Functionally, the adaptation tier includes silos such as the following:

- Content metadata for linear, VOD;
- DVR recording information and management;
- User identity, profile and personalization;
- STB remote control;
- STB viewing history; and
- Recommendations.

The means of adaptation depend on the capabilities available in a particular operator headend. If an operator has existing components for a given functionality, the adaptation tier includes simply an adapter from the interfaces provided by the operator's existing components into an SDP. However, if the function does not exist in the operator's headend, an SDP deployment must include new components to perform that functionality. In other cases, the operator's existing components provide a basic functionality, but the SDP requires more advanced capability, so a hybrid architecture for adaptation is employed.

As an example, the adaptation tier relies on an ingest process, in much the same way as is done with legacy guides today. TV listings data for linear and VOD content is imported from known, trusted sources to ensure there is a structured data set based on unique identifiers for the content, channels, program series, etc., each mapped to supplemental information such as descriptions, cast and crew, etc. This results in a structured set of metadata that is mapped to content sources and their delivery/access parameters, which describe how the client application accesses the referenced content over the MSO's delivery plant. This ensures that the source of the content is mapped correctly to the program metadata and avoids broken links or attempting to access incompatible content formats.

The ingest process is an offline process, occurring periodically. However, the retrieval of data from the resulting database of metadata is a continuous process, with higher rates of concurrency than experienced with legacy guides, due to the greater real-time demand for data with the cloud-based architecture. For this reason, it is important to separate the ingest process from the dynamic query process to ensure that data access from client applications remain unaffected.

VIEWER PROFILE MANAGEMENT

A Viewer Profile Management component can store profile information entered by viewers through the guide application, such as preferences, parental controls, favorite channels, etc. The profile information can also be enhanced with subscriber information (billing location, subscription entitlements, etc.) to enable applications to make logical decisions about how the content metadata is to be displayed on a viewer's guide, based on their profile. Through integration of the viewer profile management with the digital rights management (DRM) domain management system, MSOs can tailor the content choices presented to viewers based on an MSO's carriage rights for the content (such as geographic blackouts or in-home/out of home viewing). Another obvious benefit of storing the viewer profiles centrally is that each viewer's preferences are accessible from whichever STB or device they are using and do not have to be managed independently per device. In addition, it provides a gateway for external applications, such as remote DVR recording and TV Everywhere services.

STB REMOTE CONTROL

One of the new features provided by guides on companion devices is the ability to control the TV experience being watched on the STB. The STB remote control component is required to adapt the commands received via an SDP API into protocols that can be passed over the operator TV network to the STB inside the home. To deploy this function to legacy STBs requires either modification or enhancement of the existing resident guide (not likely to be achieved very easily), or the creation of an unbound EBIF control application deployed on the STB. Through this mechanism the SDP can change channels, manage DVR recordings and synchronize to the channels and programs being viewed on the STB.

COMMON BRAND EXPERIENCE

Today, all MSO guides are developed as applications that reside and run on their respective devices. This means that multiple versions of an application are required (even for different STB models), each integrated and tested on their respective device. This also means that to ensure common branding and usability, the guide application feature set is based on the least capable device being supported. This means that capabilities of some devices are not used to their full advantage.

NEW UI TECHNOLOGIES

Consumer electronic devices in the home can now take advantage of new UI technologies. These technologies are more feature rich, easier to develop and perform better than the traditional technologies used to implement guides on legacy STBs. New UI technologies available today support both browser-based applications and native applications.

UI technologies that appear to be making the most traction today include the following:

- HTML5
  An open and evolving W3C standard, which, while still maturing as a standard, promises to offer rich graphical functionality and is becoming widely implemented on many CE devices via web browsers.
- Adobe Flash
  A mature, proprietary technology that is widely available on IP STBs and PCs and available as a plugin to a web browser or as a standalone environment.
- Qt
  A proprietary framework technology that enables comprehensive, rich UI implementations. Although Qt requires a greater depth of integration on client devices, the results are superior to that of browser-based rendering. Qt can also be used in conjunction with browser technologies, such as HTML5, to provide acceleration enhancements while minimizing integration effort.
- Android
  A java-based and relatively feature rich environment, which is becoming widely deployed on mobile platforms. It supports native Android applications which run on a Dalvik VM.
- iOS
  The operating system on Apple devices, such as iPad, iPhone, and iPod, that provides a simple and pleasing user experience, given the controlled UI design requirements, widget set, and UI effects. A native application environment using Objective-C.

REMOTE UI

By using an SDP architecture, the guide appearance and functionality can be managed by the edge servers in the presentation tier in the same way that the metadata is tailored for each device. By implementing remote UI technology, a design framework for the guide that defines the layout, navigation and functionality can be delivered in an appropriate format and resolution to a connected device without needing to have an integrated application on the device. Various remote user interface technologies exist today that rely on a browser or rendering application on the device, which renders the guide framework to the screen. HTML5, Flash and Qt can be used as remote UI technologies. These can also be supported by edge servers in conjunction with native applications running on other client devices, where typically a native application still provides optimum performance, such as the iPad.

Whether utilizing remote UI technologies or native applications on the various client devices, MSOs now have the opportunity to optimize the look and feel of the guide based on the device on which it is running. A common guide look and feel based on the lowest common denominator of functionality is no longer necessary, especially when the guide application will be one of many other applications on a CE device, to which it will be compared. Therefore, the guide design is evolving to become a design framework that takes into account differentiating functionality across devices, while maintaining a consistent brand and user experience to the viewer. The UI design framework must take into account a number of varying attributes, which may be implemented differently on devices, such as aspect ratio, human interface controls (remote control, touch screen, mouse/keyboard, etc.), screen rotation (portrait/landscape), resolution, etc. Support for these unique features can be built into native applications, developed for specific devices or managed by the edge servers along with the metadata filtering.

DEVELOPMENT TOOLS

A key advantage of utilizing an SDP architecture is the decoupling of the UI application on the client device from a good portion of the guide engine, which now resides in the cloud. SDP architecture also opens up the possibility of developing the application as a native application for a specific device or using a remote UI technology, which can be rendered in an optimum manner by a number of devices. This decoupling means that development of the client application no longer requires specialist knowledge of the cable infrastructure and associated complex integration. This opens up the development task to a wide resource of developers familiar with the respective devices and/or remote UI environments.

Although development tools exist for popular environments such as HTML5, Flash, Qt, Android, and iOS, certain aspects of the guide application require creating features and functions specific to the TV environment. Some of these features control the TV viewing experience, such as changing channel, managing DVR recordings, changing settings, etc. Other features provide context about what the user is watching on their TV or has watched recently. Finally, enhanced metadata about the TV programming or related to buddies in the social network complement the TV experience.

These TV-control and context features all require integration and testing with the cable system and present a level of control to the application. It is important that these functions are separated from the main UI application development as a set of pre-developed, integrated and tested APIs, which can be included as part of the SDP software developer kit (SDK) made available to developers. This lets the MSO maintain control and integration of the service, leaving the developer community to concentrate on the application development.

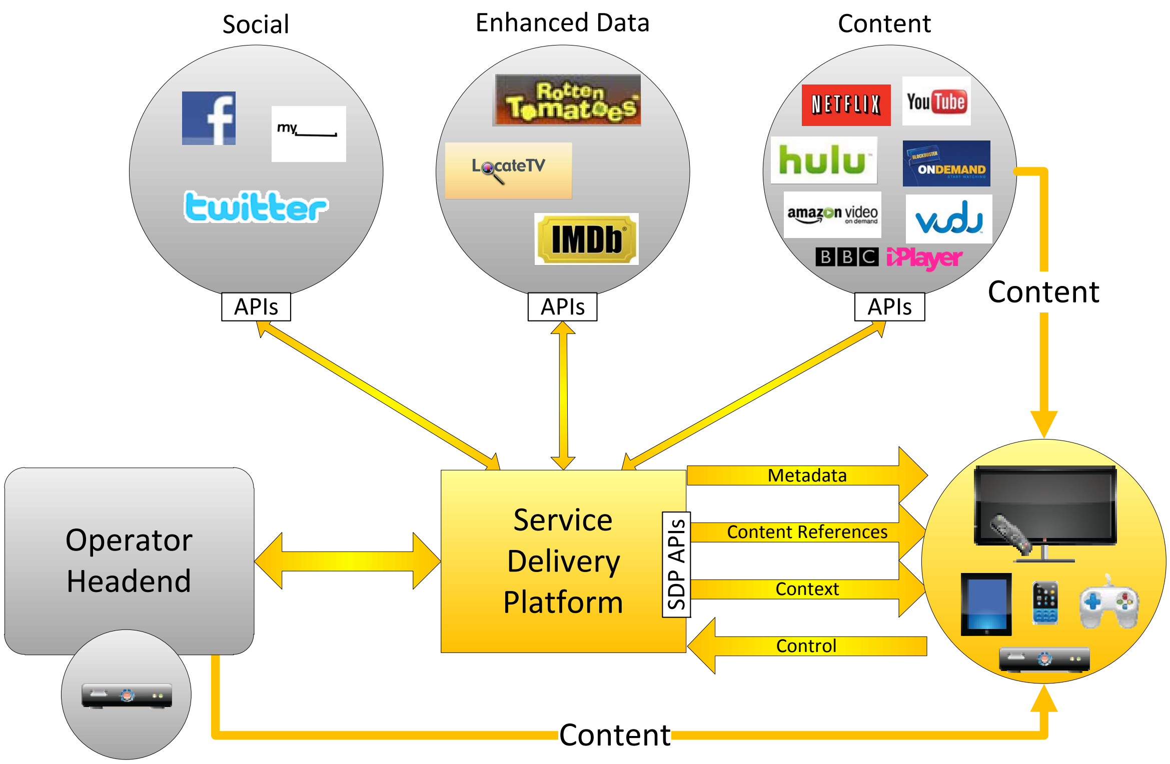

Figure 3: Service Delivery Architecture: TV-control and Context Features

The SDK APIs made available to the developer community are at the discretion of the MSO with full access controls in place, but there is obvious opportunity to standardize at least a core set of these APIs to ensure commonality of control, context and metadata features and speed of integration across platforms. For example, APIs to EBIF servers in the headend for the control and monitoring of TV viewing functions on legacy STBs from companion guide applications, such as channel changes and DVR recordings, is one such area being looked at by CableLabs®.

The SDP architecture also lends itself to convenient and controlled testing without the need for expensive lab systems. As the edge servers provide authenticated access to the aggregation tier, dedicated access can be defined on a per developer basis, which provides a filtered set of test data designed to exercise the guide application to its fullest and would not be available to regular client devices.

STANDARDS

By pushing much of the guide functionality into the cloud, the standards related to an SDP-architected guide solution are focused mainly on communication and not on hardware or implementation. The following are key standards bodies that are essential for an interoperable SDP-based guide solution:

- **uPnP** - Essential for companion device discovery on the home network.
- **DLNA** – Enables monitoring of the STB status and to send commands to control the ST for the communication between companion devices and STBs within the home.
- **HTML5** – A remote UI environment that can render the guide in an optimized format across a variety of devices. Needs to be included as the UI technology in an updated **CEA-2014**.
- **MPEG7** – For the standardization of program metadata and unique identities, by which programs can be identified across different systems and avoid re-mapping of unique IDs within the Aggregation tier.

CONCLUSION

Companion devices, such as tablets, are positioned to revolutionize the way TV is experienced by consumers. One way that operators can enable these new devices and capabilities is via a service delivery platform architecture that allows new features and functionality well beyond the capabilities of legacy guides. It can also be developed in a time frame significantly reduced from the legacy guides development and deployment cycle.

Furthermore, an SDP architecture can service DOCSIS-enable STBs to enhance the legacy guide experience by delivering the enhanced features and services over the broadband connection to the STB either for interpretation by the resident guide or through an EBIF user agent.

Additionally, building a standardized service delivery platform API and infrastructure across multiple operator systems can also create a more economically viable application development environment with the ability to deploy applications across operator networks and tie together users with different service providers.

The fast paced world of application development and social media mobility lead to challenges for MSOs to keep pace with the consumer electronics domain. Providing a controlled means of accessing platform services, protecting customer identity and confidentiality while enabling thousands of potential third party developers will ensure that an MSO’s services are accessible from the widest variety of devices. The most successful applications will engage customers and ensure that the services are always relevant on the current generation of devices.