INTRODUCING AN ABSTRACTION LAYER TO CONTROL THE USER EXPERIENCE, APPS AND INTEGRATE CONTENT ON ALL SCREENS

Albert Dahan, Founder and CTO, Metrological

Abstract

Multichannel video programming distributors (MVPDs) are facing increasing complexity as they seek to deliver pay-TV services, over-the-top (OTT) content, widgets, and third party applications that are integrated into a single experience and accessible across a growing number of devices. MVPDs want to accomplish this with the ability to instantly control the user interface (UI) and user experience across devices. The challenge is that each device behaves differently and requires device-specific solutions if MVPDs decide to solve this natively. A unified approach that enables MVPDs to simultaneously introduce new services and applications across devices is needed to simplify the process and reduce operational expenditures.

Resolving this issue demands a controversial approach. Rather than continuing to develop big, closed, monolithic software solutions, MVPDs can now use small, modular software components that connect new web-based and open source technologies with existing legacy ecosystems, to avoid being locked into proprietary systems. One way to resolve this issue is to use an abstraction layer that separates the MVPD backend, device specific middleware and operating system from the user interface, presentation and navigation layers on each device. This abstraction layer or framework should operate independently from the device and have its own mechanism in place to manage services and applications (apps) in real-time within the framework. This framework should also be able to support features that bridge the gap separating live TV and OTT content from features such as contextual apps and widgets, and merge all content on to one screen.

This paper will outline how to implement a managed framework approach that uses the browser as a generic integration point to run and control the user interface, apps and client services across connected devices. It will discuss how a cloud-based framework can be optimized and accelerated within a specific MVPD ecosystem and applicable set-top box (STB)/browser configuration to improve the user experience and minimize CPU and memory usage of the end device. Apps only need to be developed once to run on all devices and can be developed by anyone using an open Software Development Kit (SDK). This enables MVPDs and content providers to control their services and user experience in real-time, reduce costs, and rapidly introduce new apps and services.

BACKGROUND

Change is underway in the pay-TV industry as proprietary and hardware-based platforms are being replaced with more flexible IP delivery and software-driven solutions that are increasingly based on open source. The industry's evolution from offering video on a single TV screen to multi-screen delivery of live TV, OTT content and apps requires more agile solutions that can rapidly deliver new services to emerging devices.

With STB apps projected to reach 40 percent of pay-TV households by 2019 (source: ABI Research), TV apps represent the next logical extension to the app-driven world that mobile subscribers have embraced. Bringing app content to the TV transitions the experience from a passive, lean-back experience into a lean-forward, proactive experience that offers more dynamic engagement, such as games, quizzes, voting, and personalized advertising.

An application framework or abstraction layer approach can bring integrated viewing experiences to the entire device ecosystem by using the browser as a generic integration point to run and control the user interface, apps and client services across connected devices.

With a device and software agnostic platform approach, MVPDs and content owners only have to develop an app once to deploy across all devices. This platform operates independently from the device itself, ensuring that all relevant hardware and software applications can be controlled and managed centrally. App platforms offer the flexibility to quickly configure and launch integrated app services, and control the user experience across devices in real-time. This abstraction layer enables new levels of personalized experiences, such as apps that present contextual content specific to the viewer's interests across devices.

By processing video data on this framework and on very low-powered hardware, MVPDs can yield the benefits of minimal CPU usage with accelerated memory performance. Using this technology, MVPDs with legacy STBs can now launch robust TV app platforms without needing to deploy next-generation STBs.

In the last few years, this type of framework or abstraction layer approach has become more prevalent, as an alternative to streamline app development and deployment for MVPDs. This technology lays the groundwork for MVPDs to develop a successful TV app strategy that goes beyond offering a collection of apps. Operators can deploy their own branded app store that complements the TV experience, including a relevant catalog of apps and widgets, with an underlying technical platform to configure, launch and operate it.

TRENDS DICTATING A NEW APPROACH

Technology drivers

Today's consumers demand delivery of video whenever and wherever they are, in a way that is seamless to access and consume. This puts pressure on MVPDs to combine live TV, video on demand and OTT/app services into a single user experience across devices.

MVPDs typically have a separate backend management structure for content source. Managing multiple, separate structures becomes more challenging as services are distributed to an increasing number of devices.

Offering OTT/app content in addition to live TV and on demand viewing increases the CPU and memory demands on the STB. This directly impacts overall STB costs and complexity for MVPDs with older generation STBs in their footprints.

Business drivers

A number of business drivers are causing the industry to look for a new way to develop and deploy TV apps including:

- The development costs of TV app stores include the number of apps, devices and customizations required. These costs are increasing dramatically due to the volume of devices and networks on which the MVPD needs to have an app available.

- Time to market is a big cost determinant in the overall development of TV apps. The number of apps, device and network customizations required and back office complexities are creating costly time delays in getting apps and TV app stores deployed. The faster an app or app store can be deployed, the sooner the MVPD can monetize the service and see benefits such as stickiness and consumer satisfaction.

- MVPDs need to be able to productize all video offerings, including apps. But because MVPD backend systems aren't able to manage different assets agnostic to its heritage or distribution (live TV, on demand and OTT/apps), MVPDs are often unable to productize, distribute and deliver video in a way that meets consumer expectations.

EVOLUTION OF TV APP PLATFORMS

Historically, MVPDs have approached TV app deployment by developing native apps that run on STB middleware clients. This approach usually requires an MVPD to develop an app for each STB platform and multi-screen device that a consumer will use to access the service. This approach does not scale and is often too costly, requiring device specific deployment and maintenance.

In the past, some MVPDs used web-based apps for TV app development. However, using web apps instead of native apps causes each device to behave differently, requiring specific browser optimizations. This approach also lacks proper integration with the device middleware and MVPD backend components.

Today, each MVPD offers its own proprietary environment to support app publication, causing app publishers, such as content providers, to struggle to keep pace with app development. This means that each app needs to be specifically developed to run within this environment. This results in huge development costs for content providers that want to have a broad reach across all MVPD environments. It also causes some content providers to offer apps to a limited number of MVPDs.

MVPDs face the same issues as content providers. The development and maintenance costs required for several different versions of apps increases the operational costs, making it difficult to develop an attractive business case for TV apps.

TV app platforms have been moving back and forth between native and web-based technologies without great success. From a technology point of view, HTML5 provides the best abstraction level. HTML5-based app platforms are attractive from a development perspective, but pose a host of complexities, which will be reviewed in the next section of this paper.

Overcoming HTML5 drawbacks

HTML5**,** the fifth revision of the HTML standard, is a markup language used for structuring and presenting content for the Internet (source: Wikipedia). The HTML5 standard is emerging rapidly for tablets, phones and desktop environments. As is the case with evolving standards, MVPDs need to address a few key drawbacks in order to overcome issues that can arise in using HTML5.

HTML5 complexities can be minimized by first ensuring that the browser is up to date with the HTML5 features implemented correctly on the device. Using HTML5 combined with CSS and JavaScript gives developers an easy way to develop applications. HTML5 tools for profiling, debugging and rendering flaws are evolving to a level that can now be used on embedded devices.

Second, MVPDs need to ensure that the app framework addresses aspects such as Window Management, focus behavior, application lifecycle, security, compatibility layers or polyfills.

Lastly, using a unified abstraction between the browser and the app can help to further reduce complications. MVPDs need to move Application Program Interfaces (APIs) to the cloud as much as possible. This way APIs can be updated independently and without causing changes to the framework or the applications. Selecting a user interface framework that abstracts rendering across browsers and device types will simplify rendering across devices.

MANAGED FRAMEWORK APPROACH

Simply deciding to quit offering native apps does not solve the complexity. The device browser requires a mechanism that accelerates and optimizes the support of browser-based apps. Secondly, there's a need to create a controlled environment that separates the backend, device specific operating system and native client from the browser and the platform on which web apps can be developed in one unified way.

This mechanism or framework also needs to handle the integration with web apps and native device aspects such as device navigation, scaling and display, and other advanced features that require connection between the web app and the consumer device.

MVPDs are under competing pressure to lower the cost of service and to increase the OTT and app content and app capabilities on the STB. Today's more prevalent approach, which handles everything locally on the device, is not equipped to solve these two mutually exclusive demands on the STB. The way to solve this roadblock is to ease the demands on the STB by handling these new features outside the device.

Taking steps to make sure that technical platforms are open and easy to integrate will help make scaling and operating easier. Additionally, it will also help MVPDs to remain attractive to content providers and developers to expand app content and libraries to subscribers.

ELEMENTS OF A FRAMEWORK / ABSTRACTION LAYER APPROACH

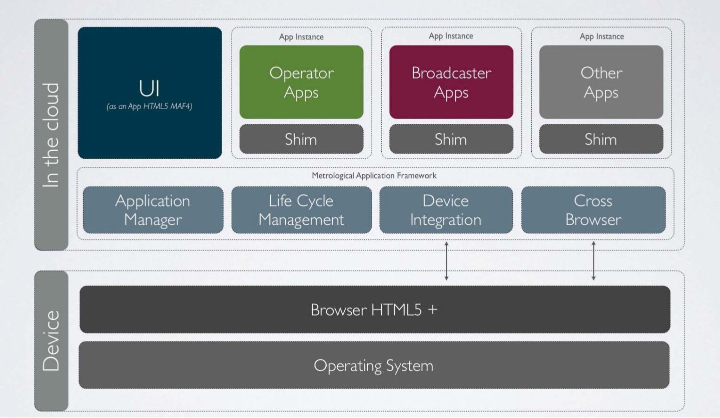


Figure 1: Sample Architecture for TV App Abstraction Layer

Definition and approach

This approach uses an abstraction layer that separates the MVPD backend, device specific middleware and operating system from the user interface, presentation and navigation layers on each device. This abstraction layer operates independently from the consumer device, including a STB or any second-screen device, has its own browser onboard, and has its own mechanism in place to manage services and apps in real-time within the framework. This framework is also able to support features such as contextual apps and widgets, and merge all content onto one screen.

This modular approach is unlike today's self-contained apps that have key functionality built directly into the app itself. All applications are isolated in their own sandbox from a development and security process. This separates the user interface, EPGs, channel bar, notification, VOD , etc. into a shared resource, thus simplifying the lifecycle management of each app and speeding up the development process. Now key functionality can be used in a more modular way rather than being confined to single app. For example, for a self-contained app a channel lineup change means that the entire app needs to be updated. An abstraction layer approach enables MVPDs to create apps with specific content but to share these key functionalities such that updates are shared and can be performed automatically

Benefits over other approaches

The implementation of a managed framework approach that uses the browser as a generic integration point to run and control the user interface, apps and client services across connected devices offers several benefits:

- The cloud-based framework can be optimized and accelerated within a specific MVPD ecosystem and applicable STB/browser configuration to improve the user experience and minimize CPU and memory usage of the end device.

- Apps only need to be developed once to run on all devices.

- Apps can be developed by anyone using an open SDK. This enables MVPDs and content providers to control their services and user experience in real-time, reduce costs, and rapidly introduce new apps and services.

- The framework backend can support live TV, video on demand and OTT/app content with an integrated experience on the TV screen.

SYSTEM ARCHITECTURE

Technology requirements

The main requirement for implementing an abstraction layer is to ensure that devices have a secured/trusted browser environment that provides maximum support for HTML5 specifications. This includes specifications that are still in draft format. HTML5 is evolving rapidly and the browser on the device should be able to adapt to updates and changes accordingly. Most of the middleware capabilities can be mapped to HTML5 specifications or moved to the cloud.

Integration process

By enabling an application framework, device integration is simplified, lifecycle management is controlled and runtime is securely isolated. The key to leveraging the application framework is to (1) isolate features within an application and (2) expose services between applications. For example, video on demand and EPG are separate applications but search and recommendations functions should be able to leverage both types of content within the app. The basics of an application framework should contain window and focus management, lifecycle management of applications, resource management, security, and cross application communication.

Furthermore, it’s important for MVPDs to select a user interface framework that runs correctly and efficiently on embedded devices. Web application development should use the correct development tools to isolate memory leaks, hardware acceleration bottlenecks, and avoid high CPU and memory usage.

This approach uses the device browser. Therefore, the framework settings need to be optimized and the browser needs to be accelerated.

Additional points of integration that are required include:

- STB integration: enables app store navigation via the remote control unit, provisioning of channel tune information, exchange of channel change commands, etc.

- EPG integration: enables apps to run contextual to the live TV or IP video content played out.

- User management integration: handles user access, user profile information and other criteria.

- Billing integration: processes and handles app and in-app purchases via the MVPD invoice.

NEW MIDDLEWARE APPROACHES

Most new middleware architectures have a browser approach, but enabling HTML5 development can pose challenges for middleware testing. It is important to leverage HTML5 correctly. For example, in creating a rich user experience for devices that have OpenGL capabilities, MVPDs and developers should take a look at new standards, such as WebGL.

LEVERAGING OPEN SOURCE TECHNOLOGY

The use of open source components will help speed development on both apps and devices. Open source offers a faster time to market and enables MVPDs to tap into a big developer community to bring more apps and innovation to TV apps.

There are, however, downsides to using open source in the TV environment. Open source can pose a security risk because of its inherent transparent approach to the developer community. MVPDs should also be aware of the restrictions and legal risks associated with an open source approach. For example, the basic philosophy of making software source code available to developers and providing software to others with limited or no restrictions on its use can have legal impacts on intellectual properties, patents and revenue streams.

ABSTRACTION LAYER IMPLICATIONS FOR THE APP DEVELOPER COMMUNITY

Benefits of an open SDK

Using an open SDK is crucial in simplifying the development of apps. The SDK should come equipped with an open source and lightweight user interface framework, to ensure performance and stability on any device. By developing on the SDK, MVPDs can ensure the application will run on any type of device.

An open SDK also circumvents the need to redevelop or 'reinvent the wheel' every time a new operating system is launched, and with pre-packaged app store content available, MVPDs can, in effect, hit the ground running.

And finally, an open SDK also offers a more cost effective means for distributing niche content. Using an SDK, developers can reach target audiences more effectively, rather than using a wider broadcast route.

NEW FUNCTIONALITY

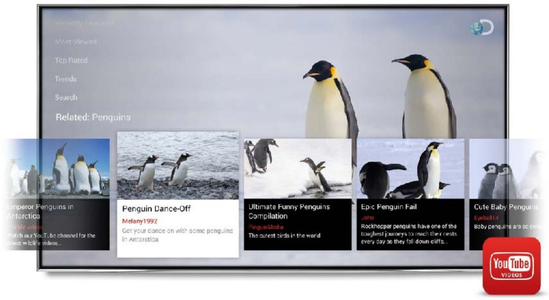
This framework approach is able to merge all content sources and support new features that enhance the overall consumer experience. A key advantage of using an app framework approach is that it enables MVPDs to bring these new features and the same user experience to legacy STBs without touching the device. Examples of this type of functionality include:

- Device pairing: enables viewers to connect their mobile phone and use it as a remote control, a video game controller, a payment device, or an authorization mechanism that enables consumers to simply to swipe videos to their TVs for instant viewing.

- Contextuality: uses APIs that allow apps to recognize what the customer is watching and suggest contextually relevant internet content across devices. For example, while watching sports or a wildlife movie, contextual apps can display related content such as real-time scores or animal clips, integrating app or OTT content with the live TV experience. A contextual app can deliver "Red Button" functionality as well. The overlay or side-by-side app supports interactive program formats, such as live voting and sing-alongs, enriching the viewer's experience of a live broadcast.

- Social Media Integration: personalizes the TV experience, enabling consumers to share TV and app content with family and friends.

- Cloud based UX: supports centrally managed, cloud-based user interfaces that can work across multiple devices. This offers the potential for MVPDs to be able to perform a single update of the user experience that leads to an automatic, real-time update of the user interface across all devices.



mperor Penguins in Antarctica
Watch our YouTube channel for the test wildlife videos..

Penguin Dance-Off
Melany1992
Get your dance on with some penguins in Antarctica

Ultimate Funny Penguins Compilation
PenguinMedia
The cutest birds in the world

Epic Penguin Fail
John
Rockhopper penguins have one of the toughest journeys to reach their nests every day as they fall down cliffs...

Cute Baby Penguins
Eyeball14
Baby penguins are so

Figure 2: An Example of Contextual Viewing Experiences

Benefits for TV MVPDs

The framework or abstraction layer approach can be productized, as a fully customized TV app store with the ability to manage the user experience across devices. This enables MVPDs to easily add a suite of business management and business intelligence tools. A web-based CMS enables MVPDs to manage their own app stores and user experiences across devices in real-time.

The approach can be used to deploy an app store and user experience on legacy devices. The framework can deploy an app store using a cloud rendering service even if the device does not have a browser. In this case the app content will be delivered to the legacy device "as a live channel".

This means that legacy devices and browserless STBs don't need to be replaced in order to support TV apps stores or offer more integrated viewing experiences. In essence, these STB platforms are "refitted" by putting the user interface and OTT/app services in the cloud. This approach has a number of other benefits for MVPDs including:

- Enables a quick launch of MVPD branded app stores with a unified consumer experience across screens.
- Offers a device agnostic approach across an expanding device ecosystem.
- Enables content monetization that helps MVPDs to increase ARPU, customer loyalty and reduce churn.
- Provides an integrated experience that combines live TV, OTT, social media content and capabilities into a single consumer experience.

BENEFITS FOR CONTENT PROVIDERS

In addition to benefits for MVPDs, this approach also has benefits for content providers. Using an abstraction layer approach, programmers and other content providers can develop an app once, rather than multiple times to reach multiple devices, in order to connect and interact with consumers. The technology also offers more reliable delivery with mechanisms that ensure that content is provided with quality of service to any device.

DEPLOYMENT EXAMPLES

Liberty Global uses an application abstraction/framework approach that is currently live across three of their platforms, including a cloud-based browser rendering solution that is deployed on its Horizon platform.

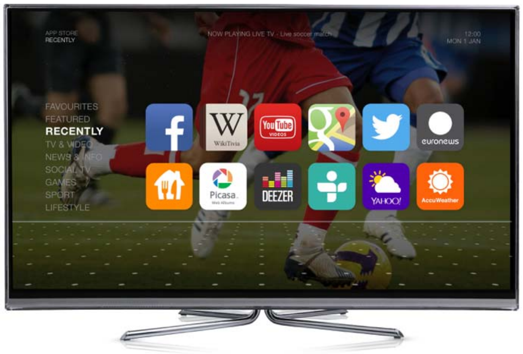


Figure 3: Selection of Liberty Global Horizon TV Apps

All applications are developed using an SDK that runs directly across all platforms and is integrated with the app framework. For example, a hackathon that was organized with 100 developers in collaboration with Liberty Global, created 20 applications during the event using the open SDK.

Market results

UPC Hungary, a Liberty Global property, used a variation of this technology to bring online content to its pay-TV subscribers. The MVPD partnered with YouTube and other technology companies to deliver 20-plus apps to legacy STBs. This was the first time that an MVPD offered an online video service to its entire subscriber base using legacy STBs.

In this deployment, UPC Hungary is using a cloud-based interface and TV app framework, including more than 20 apps, that is being delivered over the MSO's managed network. The MVPD is offering the TV apps to its 520,000 subscribers as part of its platform, providing additional value for existing subscribers.

UPC Hungary subscribers are streaming more than a million minutes per day of YouTube content. 68% of subscribers have tried the TV apps service, and of that number, 83% have returned for additional views.

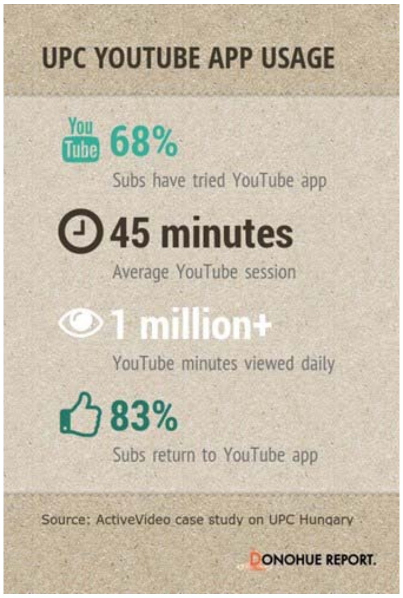


Figure 4: UPC Hungary YouTube App Usage

CONCLUSION

MVPDs need a mechanism that enables them to offer OTT/app content seemlessly integrated with the live TV and VOD offerings across a growing number of devices. An abstraction layer approach can be used to integrate these content sources with a single backend management system that streamlines app development and deployment. By separating the MVPD backend, device specific middleware and operating system from the user interface, presentation and navigation layers on each device, MVPDs can avoid having to develop native apps for each and every device.

This approach is deployable on legacy STBs to enable robust MVPD branded TV app store deployments. It also brings new functionality to TV apps, such as interactivity, contextual experiences, mobile device pairing, and social media integration that resonate with consumers. Global MVPDs are just beginning to reap the benefits of this technology that is engaging consumers and driving increased content and app usage on MVPD branded app stores.