



HOW TO INNOVATE WITH ANDROID™ TV: A PATH TO BETTER PRODUCTS

Authors

Fabien Battini, Technicolor & Patrice Silvant, 3SS



How to innovate with Android™ TV: A Path to Better Products

3 Screen Solutions (3SS) and Technicolor are at the forefront of our industry in Android TV™ deployments, having jointly delivered several major Android TV Set-Top-Box (STB) service platforms for leading Network Service Providers, featuring Operator Tier with Custom Launcher integrations. The partners have together delivered four major projects, based on Technicolor's hardware platform and middleware combined with 3SS' 3READY Android TV Front End and UI Framework. Several further collaborative projects are to be announced shortly.

The underlying recipe for success in these deployments has proven to be “**clean separation between application and OEM layers**”.

As a consequence of the proliferation of OTT services, several trends have emerged in our industry:

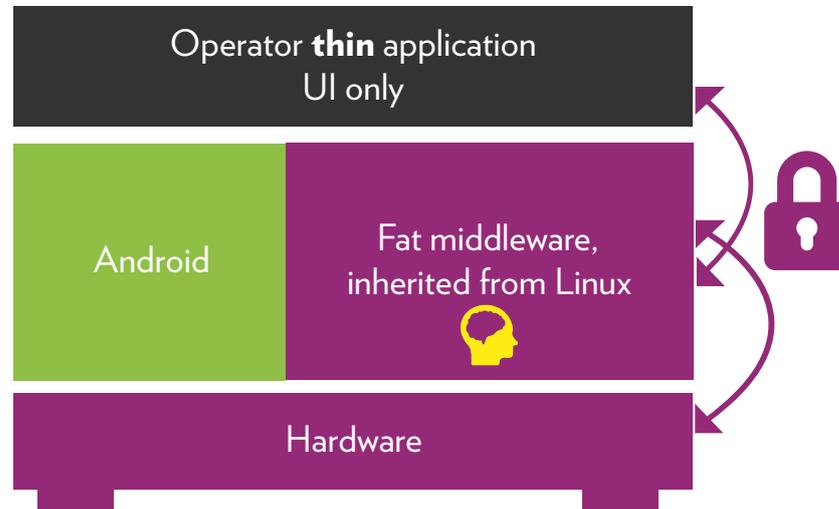
Pay TV operators are also becoming **content aggregators**, delivering broadcast, broadband and third-party OTT services and applications; guiding, curating and advising end-users in the increasingly fragmented content jungle.

As a result of this trend, TV applications today require a **dynamic features and services roadmap** which can be updated throughout the lifetime of the STB; meanwhile subscriber retention increasingly depends on deep user experience customization.

With today's rapid pace of technological innovation, and growing competition, service providers need to be nimble, able to adapt and optimize all their delivery devices, with a **much accelerated time-to-market**.

Linux is no longer able to meet today's market needs:

The old Linux-based model, with its inflexible, 'locked-in' interplay between User Interface (UI), middleware and STB platform, is no longer meeting the needs of today's dynamic market.



Several OEMs have attempted to replicate the traditional Linux side-by-side execution paradigm and apply it to Android™ TV projects. However, these attempts are doomed, and are beset with inefficiencies, due to restrictive and limiting technology interlocks. As a result, these projects often suffer delayed integration. Furthermore, application portability is poor. Legacy code is incapable of smoothly adapting to today's regular necessary enhancements of internal security (SE-Linux) and dealing with the introduction of new features, therefore maintenance is prohibitively expensive. With such an approach, introducing

new features requires updates to both middleware and applications, thereby slowing down roadmap rollout. Such deep and heavy integration effort creates a technology "vendor lock", which is highly disadvantageous for operators.

For example, if the full PVR stack is implemented within the middleware, then the integration with the TV APK is painful because of specific APIs. Furthermore, success in porting to a device made by a different manufacturer will likely be limited, and evolutions would continue to require updates to the middleware.

In today's market, with Android™ TV's growing popularity, Android™ TV STBs **deserve a new and better development model**, based on **Lean** (Lean Software Development: An Agile Toolkit. Addison-Wesley Professional. ISBN 978-0-321-15078-3.) and **Agile** (<https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>).

There are multiple benefits to adopting these methodologies.

To successfully achieve this adoption, we advise the following:

Rule 1: Whatever can be in the TV App should be in the TV App

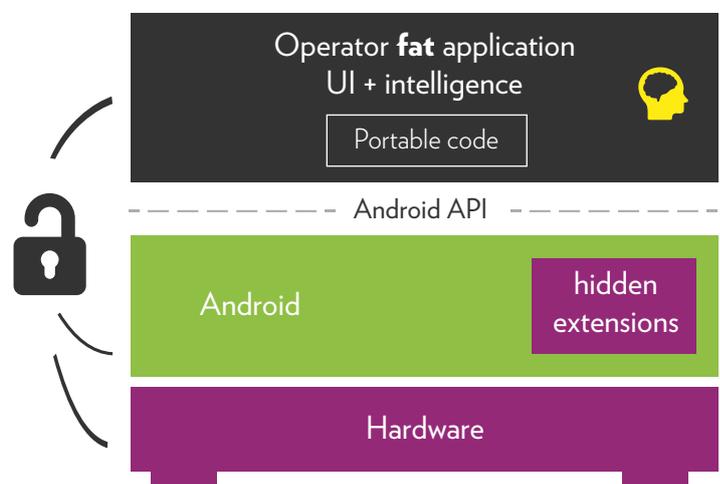
To optimize operability, security, and performance of the hardware, only a **limited set of features is required of the middleware**. The operator has the freedom to implement and enable additional features and intelligence in the TV or Launcher application:

As a result, the operator benefits from easier and greater control over the service and feature roadmap, the operator can perform updates more often and more efficiently, and can more smoothly and efficiently optimize the User Experience with little or no need to modify the middleware.

In the PVR example, a standard middleware simply implements the regular recording commands: Start/Stop/Play/Record. The TV APK manages scheduling, conflicts as well as logic for season and series recording. Also, thanks to correct and appropriate implementation of the Android™ TV architecture, the application's PVR feature can evolve freely, e.g. splitting and migrating between local and cloud storage as needed.

Rule 2: Use standard APIs whenever possible

Based on the experience derived from hundreds of Android™ TV projects, we know that Android™ TV APIs enable the appropriate interfaces and enable rigorous testing. Our approach is that the STB vendor should preferably use Android™ TV APIs over custom code, and that necessary extensions should use Android design patterns as much as possible. When the STB manufacturer uses standard APIs, integration of the TV applications is much simpler and faster, and, since the TV app is more generic, it can also be shared more efficiently.



For instance, Technicolor delivers a single, standard TV Input Framework (TIF) for all linear video content (DVB-C/S/T, ISDBT/Tb, IPTV, etc). Additional information is stored as part of internal provider data in the TIF database. Validation can be done via the un-modified Google Live-Channels App, independent of the operator's TV app. Therefore, 3SS is enabled to independently develop its TV APK on different hardware with similar characteristics, thus, integration of the whole broadcast element on the final STB is straightforward.

Porting a custom TV stack from a legacy Linux implementation into Android™ TV software architecture yields poor results. By following this path, the STB manufacturer might accelerate its product development and TIF implementation. But ultimately, as it requires lower level integration with Android™ TV (e.g. memory management, security) the resulting product is not completely proven/refined and will have higher maintenance cost; these factors will negate any short-term benefit.

If the semantic of the legacy Linux API is similar to Android™ TV TIF, the STB manufacturer will greatly benefit from moving to the standard Android™ TV API, as this would greatly simplify integration with the TV launcher. If the STB manufacturer retains a radically different API semantic, then the resulting integration will be project- or STB manufacturer-specific, time-consuming, not portable, not reusable and cannot effectively be tested on different devices. Therefore, when a third-party TV launcher is desired, using standard Android™ TV APIs is much better and much faster. Legacy interfaces are meaningful only when the STB manufacturer plans to offer its own TV launcher.

Pierre Donath, Chief Product Officer, 3SS:

“As industry leaders, 3SS and Technicolor have collaborated on many major Android TV projects and together we’ve developed a now proven and successful integration methodology. **From our standpoint as frontend engineers, system integrator and user experience experts**, our shared approach yields not only the traditional benefits of well-designed APIs - easy prototyping on standard devices, high level of portability, low interoperability issues - it also **results in faster delivery of projects, higher quality, and it minimizes the impact of Android™ TV evolutions, delivering smoother ongoing Android™ TV version migrations.**

Our Custom Launcher based on 3READY has been evolving since its first deployment at Com Hem (Tele2) in 2018, independent of choice of STB, while adding features and intelligence without the effects of vendor-lock and enabling the possibility of use in other form-factors (e.g. dongles and connected TVs). **This generates a hugely positive scalability effect, highly beneficial to all stakeholders in any project’s ecosystem**, and is certainly one of the key success factors of the projects we have developed and delivered together.”

**Brian Jentz, Senior Director Video Product Management,
Technicolor Connected Home**

“By working closely with Google and 3SS, we have paved the way to **a more industrial approach to Set-Top-Boxes**, enabling larger scalability of Technicolor’s software R&D effort. For example, **Technicolor’s DVB-C/S/T stack is reused in over 20 products**. In addition, Technicolor has performed over **30 Android letter upgrades** with deployed devices. This is an order of magnitude higher than most OEMs. The adherence to Android design rules has also simplified the integration of new Android low-level features such as improved containerization and treble, as well as enabling optimized memory and power management. **Early integration and automated testing reduce overall time-to-market.**

Technicolor made a large software investment to implement this strategy, which generated key competitive advantages with clear business benefits. We are committed to continuous improvement and we are already developing the next stages of this approach, leading to **additional scalability and increased customization.**”

Conclusion

The Android™ TV deployment strategy used collaboratively by Technicolor and 3SS leads to **simpler, faster, more successful project delivery**, with far fewer integration and maintenance risks.

The initial launcher/middleware/STB/backend integration occurs very early in the deployment process.

Additionally, a **custom launcher can be developed independently** from a given hardware integration and deployment schedule, thereby allowing more time for design and customization.

This approach also simplifies and streamlines the development process in new collaborative projects; this was a key success factor in the Com Hem Tv Hub project and has led to the customer winning multiple prestigious industry awards in 2018 and 2019:

CSI Awards 2018

Com Hem Tv Hub named “Best TV user experience - Operator deployment”

Connies Awards 2019

Com Hem Tv Hub wins two prizes, is named “Best User Experience” and “International Grand Prix Winner”

Should you want additional information about Technicolor or 3SS commercial offer, please write to : contactsales@technicolor.com or contact@3ss.tv

technicolor

