

Architecture

- [Evolution of RDK](#)
- [Architecture Details](#)
- [Application Scenario](#)
- [Useful Links](#)

RDK Video Architecture is designed to enable service providers and device manufacturers to develop and deploy innovative video applications, services, and user experiences. It consists of several key components that work together seamlessly to provide a robust video platform.

By leveraging the pluggable architecture of RDK, a variety of target device profiles can be supported, ranging from a basic IP streaming-only STB to a full-fledged RDK TV.

- [RDK Video \(RDK-V\) IP](#) provides a common method to manage video playback functions. The IP client device serves as an interface and receives video content from an in-home media gateway device or from an external media server. It will have only IP based data streaming without any display of its own Tuner capabilities.
- [RDK-V Hybrid STB](#) is an IP STB device along with capabilities such as tuning, conditional access, and stream management with which we can manage complex video functions .
- [RDK-V IP TV](#) is a smart TV profile powered by RDK Video stack that brings all your favorite apps, live channels, and On Demand contents together in one place.
- RDK Hybrid TV is a combination of [RDK TV](#) plus [Hybrid](#) capabilities such as tuning, conditional access etc.

Evolution of RDK

From RDK6, RDK shifted from quarterly to annual release cycles, prioritizing quality through meticulous planning and comprehensive certification for improved user experience. This annual RDK Video release aims to synchronize RDK-V with standard industry release practices while comprehensively addressing shared challenges within the community. This approach facilitates the smoother and more consistent adoption of newly contributed features, utilizing the latest releases from technology partners. By aligning with SoC partners, the release enables better resource planning to support core RDK platforms. Furthermore, the RDK video release aligns with SoC, OEM, and App releases, fostering a more cohesive and efficient ecosystem. The first annual release is RDK6 and its release notes can be accessed from [here](#) .

Quarterly Releases



Video Accelerator
Pre-integrated Apps
Pre-integrated App Store
Voice Integration

Quarterly Releases



TV Profile
Reference UI
Test & Certification Suites

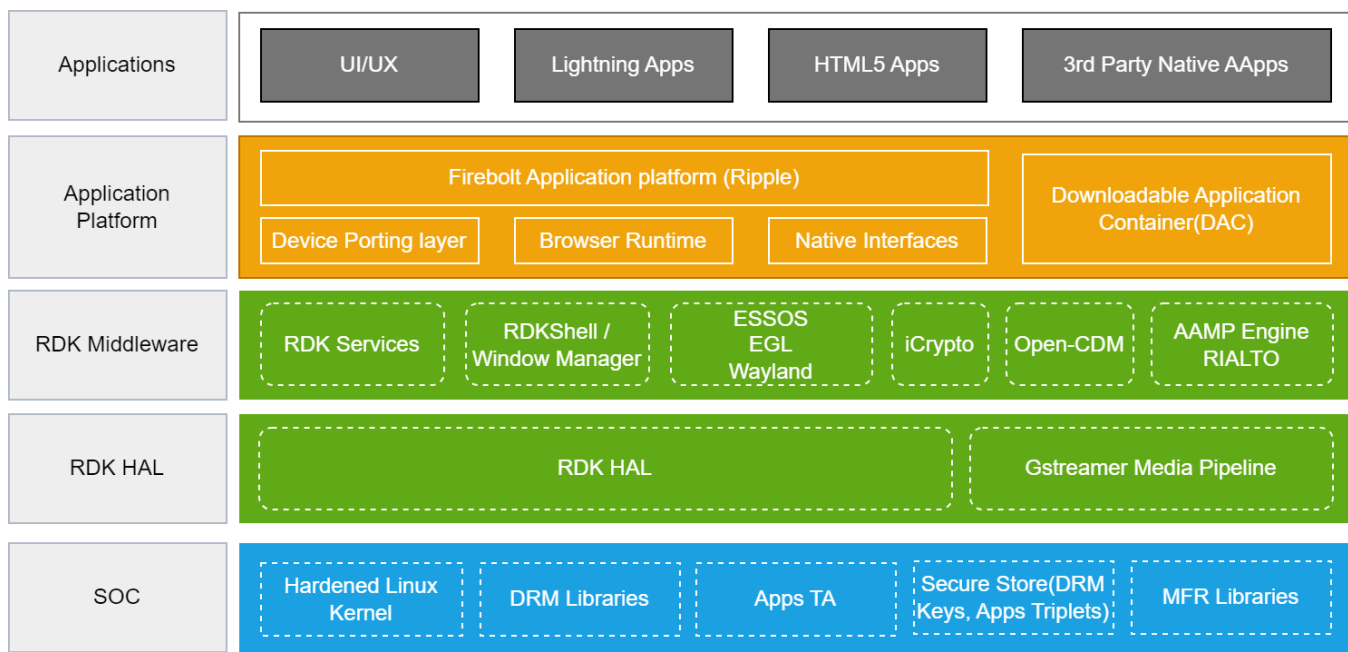
1st Annual Release



Certified Annual Releases
Security & Containerization
Firebolt 1.0 Certified
WPE Browser Upgrade
Android Common Kernel
Device Management

Architecture Details

Below is an illustrative representation of the RDK Video software stack, depicting the various components and their interactions.



At its core, RDK consists of five main stack levels, each serving a specific purpose in the overall architecture. These levels are as follows:

Applications

The Application Layer primarily focuses on the end-user experience. This layer contains applications that provide various services, content, and features to the users. While the RDK ecosystem is continuously evolving, supported applications typically include popular **OTT services** like Netflix, Amazon Prime Video, and YouTube, alongside native broadcaster applications and other services.

Application Platform

The Application Platform Layer in the RDK ecosystem offers essential tools for developers to create applications. It includes components like a **UI framework**, **HTML5** rendering engine, and a **JavaScript runtime**. Acting as middleware, this layer facilitates communication between applications and core RDK services. In the RDK Video framework, **Firebolt®** handles UI rendering and user input, enabling extensive customization. **Lightning™**, an open-source JavaScript platform, manages the application lifecycle and integrates components using WebGL for rendering. Together, Firebolt® and Lightning™ form a robust foundation for seamless and efficient application development in the RDK Video ecosystem.

Firebolt® 1.0(Ripple) - Firebolt® 1.0 (Ripple) streamlines RDK app integration with standardized rules. Ripple, its open-source Rust-based Application gateway, facilitates dynamic extensions and serves as a Firebolt® Gateway. RDK 6 is Firebolt® 1.0-certified, with a comprehensive test suite for compliance.

Security - The Application Platform Layer ensures robust security with Dobby-managed **containerization**, leveraging Linux kernel features for process isolation. **Downloadable Application Containers (DAC)** enable secure running of binary applications on STBs without modification, ensuring compatibility across RDK 6 devices. Access Control is enforced through **AppArmor**, a proactive Linux security system. RDKM's open-sourced AppArmor profile generator tool for RDK 6 provides fine-grained control over process resources, contributing to a secure environment.

RDK Middleware

Serving as a vital bridge between the Application Platform Layer and the hardware(HAL), the RDK Middleware Layer incorporates essential components that are pivotal for the seamless operation of the RDK platform. Core to this layer are **RDK services**, providing JSON-RPC services for interactive applications. In the realm of security, **iCrypto** handles critical cryptographic operations, ensuring secure communication and data protection. **Rialto** offers a secure solution for AV pipelines in containerized applications, and the **Window Manager** orchestrates GUI layout. **Device management** enables streamlined operations in RDK deployments, including bulk operations and firmware downloads. XCONF integration revolutionizes code downloads for a smoother deployment experience. Log uploads aid comprehensive debugging, offering insights into system performance. RDK Feature Control (RFC) enables dynamic feature management for enhanced flexibility. Telemetry systematically collects essential data insights, while WebPA ensures secure communication between cloud servers and RDK devices. The Media Player, crucial for local rendering devices, manages various pipeline functions, supporting IP and QAM playback. The **Open Content Decryption Module(OCDM)** enforces Digital Rights Management (DRM) policies. Together with other RDK elements, these components ensure the efficient and secure functioning of the RDK platform.

RDK HAL (Hardware Abstraction Layer)

In the RDK video stack, the HAL layer plays a vital role in facilitating communication between the video application software and hardware components like the GPU, video encoding/decoding hardware, and audio devices. It provides a standardized framework for functions, data structures, and protocols, enabling efficient hardware resource utilization. The HAL layer manages **hardware initialization**, **memory allocation**, **input/output operations**, and **hardware-specific events**, shielding software developers from hardware complexities and allowing them to prioritize user experience and functionality.

RDK provides a set of HAL APIs that abstract the platform from RDK. Vendors need to implement the HAL APIs to meet the HAL Specifications. With the help of the HAL API Specification for different RDK Components, vendors can successfully port RDK to their platform. Depending on the device profile ([IP STB](#), [IP TV](#), [Hybrid STB](#) etc.), vendors may choose the relevant components and perform the port by implementing the HAL layer. Detailed information about vendor porting guide will be available soon.

SOC

The System on Chip (SOC) Layer forms the foundational interface between hardware components, ensuring system security and reliability. It incorporates various crucial elements such as **DRM Libraries** , which manages digital rights and secure content delivery to prevent unauthorized access and distribution. **Trusted Applications (Apps TA)** guarantee the secure execution of sensitive operations and protect critical data from unauthorized access. The **Secure Store** oversees the storage of DRM keys and Apps triplets, maintaining the confidentiality and integrity of vital data. Additionally, **MFR Libraries** manage hardware functionality, providing access to specific hardware features and capabilities, thereby contributing to the overall security and functionality of the system.

Application Scenario

Consider the use case of a user accessing a streaming application like Youtube on an RDK Video-supported device. The user interacts with the application through the Application Layer, selecting content and initiating playback. The Application Platform Layer, utilizing the Firebolt® and Lightning™ frameworks, manages the user interface and application lifecycle. The RDK Layer ensures seamless communication between the application and the hardware, managing services, cryptographic operations, inter-component communication, window management, and content decryption through OpenCDM. The RDK HAL Layer then utilizes the Gstreamer media pipeline to decode and render the video content, ensuring a smooth and high-quality viewing experience. Finally, the SOC Layer provides a secure environment for the entire system, safeguarding the hardware, managing DRM policies, and securing sensitive data, ensuring a secure and reliable video streaming experience for the user.

Useful Links

RDK Video:

You can find an overview of the RDK Video platform, detailing its key features and functionalities at [RDK Video Documentation](#)

Application Development:

Developers interested in RDK application development using Firebolt® can refer [Firebolt® Overview](#) and

Developers interested in RDK application development using Lightning™ can refer [Lightning™ Framework](#)

Security:

To understand the security features in the RDK, please refer [RDK Security](#).