REAL-TIME MEDIA ANALYSIS ARCHITECTURE & PROTOCOL

INVENTORS:

DARYL WILLIAM MALAS

CHRISTOPHER J. CORCIMIGLIA

DEBORAH S. FITZGERALD

ERIC WINKELMAN

JACOB B. MALONE

SCOTT L. CARUSO

Description

The following anecdotal story continues to the use case of AR glasses, but it is intended to illustrate a protocol and system architecture for delivering content to devices in real-time based on varying inquiries from those devices.

Let us go back in time a bit and imagine you are attending a CableLabs Conference located in Keystone, CO at the Keystone Conference Center. You dress nicely, but not too nicely as the conference has stepped down its formality over the years. The fresh mountain air hits you as you walk out of your hotel room and head over to the conference center. Before you step through the doors of the conference center, you slip on your new company provided, lightweight AR glasses. Touching the side of the glasses, you ask clearly, "Load my SalesForce app." (Or, the SalesForce app could load automatically the first time you look at someone and say, "who is that?") You walk through the doors and pan the busy hallway located outside of the main conference rooms. You're not looking for anyone in particular, but you would love to connect with current and past contacts from across the cable industry. Someone catches your eye, and you think you remember meeting them from some event before. As you look at them for just a second or two, your glasses immediately display their name, "John Doe at Cable Tech." "That's right, now I remember their name," you think to yourself. However, you still can't place where you last saw them. Touching the side of the glasses again, you ask clearly, "Give me more information about John Doe at Cable Tech." The glasses immediately display John Doe's title, contact information and a note regarding the time you met at CES in 2017 at the Samsung Booth. Knowing this information, you walk up and introduce yourself. "Hi John, it's great to see you here. Didn't we meet at CES a few years back? How are things going?"

The above AR scenario is one example of devices and applications enabled by this new CableLabs innovation. Imagine devices and applications developed for many different interests both for personal hobbies and business. Here are a list of example use cases supported by this architecture and protocol:

- Autonomous car requesting traffic changes and updates in real-time
- Cloud gaming device requesting and receiving content analysis in real-time.
- Manufacturing robot controls
- Farm implements responding to conditions
- Factory automation
- Warehouse robotic pickers

The device and application use cases supported by this architecture are incredibly extensive.

This idea revolutionizes how media is delivered to devices. It solves the following problems:

• Delivers application content in a way that leverages the future network's capabilities providing a near real-time experience for the consumer, making devices, such as AR, an interesting technology for consumers.
• Provides an ability for users to not require "every application" for viewing data, while offering operators the ability to offer enriched content through business revenue relationships
• Improved dynamic experiences
• Platform to offload the compute and memory demands of current generation of devices while enabling the development of future devices with smaller form factors and less resource demanding
• Open platform that encourages broader development and lowers the cost of entry by applications and service providers

This architecture provides the framework and interface for real-time devices, and other next generation applications, to interact with local edge locations for improved user experiences. A primary objective of this entire solution is to dramatically expedite the analysis of every media request and respond to the devices in sub-second time with a result to the media observed via the device.
• Referencing the attached diagram, it provides an overview of the innovative aspects of the end-to-end architecture.
• While some of the aspects of the architecture are not innovative, the following aspects are innovative within the architecture represented in the attached figure. Specifically, the functions in "red" in the attached figure are believed to be an innovative approach to this architecture. They are numbered and correlated in the following description.

The media request [1] is a new network-based protocol that transports the content, metadata and authorization credentials from the device or device's application to the cable network. This protocol could be standardized in an SDO, such as the IETF, if desirable to increase scalability of the protocol with the ability to interop with devices, applications, service providers and other cable operator networks.

This metadata could include, for example in an AR use case: location, time of day, the area they are looking at, any specific command, etc. From the location you can discover if they are in a store, or a wildlife sanctuary. And the eye tracking information allows you to determine if they are looking at the bird, or boat.

The Media/Metadata Analysis and Decision server [5] unpacks the request. It sends authorization requests [2] to the local cable operator's authorization server [3], it validates the media contained in the payload, it sends the captured media to the application for analysis to determine what the media represents, receives the analysis resolution from the application, and it retrieves the relevant result from the CDN for transmission back to the device.

The app authorization (AAA) [3] proxies requests on-behalf of the user and the application. The purpose is to avoid having to send authorization requests to the application for every device request. This authorization process continues in intervals [4] on-behalf of the user to ensure their authorization has not expired.

In order to ensure scalability for many requests from many devices, the content is retrieved and stored in the cable operators CDN through a CDNI interface [6] with the application. While the first request to the application may take a few seconds, subsequent requests will be very fast while using the cable operators CDN to store the content.

An extension of the cable operators CDN could be placed on the device, represented as a "Deep Edge" cache. This is a cache located on the user's device, such as a smartphone. This cache is controlled by the cable operators CDN to ensure content policies are enforced by the application. In the future, AI/ML could be used to pre-cache analysis data based on user patterns, geofencing awareness and other use cases.

Finally, an optional component is the Dynamic Media Analysis and Application Director [7]. This allows for devices to be operate in a "general mode" (i.e., application provided by the cable operator or in partnership with the device manufacturer). The general mode allows the devices to capture media not associated with any application on the device. The media is forwarded to the Director, and the Director does some basic analysis and provides suggestions of applications with additional information about the captured media. The application can be run from the cloud or give the user an option to download the application to the device. This application suggestion can be a revenue opportunity for cable operators.

Problem Being Solved

Tomorrow's applications will require real-time responses to device queries. This goes beyond simply the network layers. Low latency and higher throughput are necessary building blocks, but these devices require a protocol that adds to the lower-level improvements to ensure queries and the application responses are also delivered expeditiously. The types of devices and the applications in use will vary. Examples of these devices will be smartphones, intelligent vehicles, autonomous robots and augmented reality glasses. For the purposes of understanding the types of applications requiring real-time responses, this idea will focus on Augmented Reality (AR) as a specific use-case. However, any of the example devices can leverage this architecture.

AR is not a new concept. Large, clumsy glasses have been produced by vendors for years now with notables being MagicLeap and Microsoft HoloLens. In general, AR glasses have fallen flat with consumers. The need to store and process application data on the glasses adds to the bulky size. Even if AR glasses manufacturers achieve a more ideal form factor, the technology still lacks the cloud-based app ecosystem to really compel mainstream consumers to justify the cost and integrate a regular use of AR technology in their daily routines. Here are the main problems of the current closed ecosystems that a cloud-based architecture can improve on:

- Applications are too closely tied to the company's own applications that they restrict 3rd party software developers from taking advantage of the glasses
- Applications are so loosely developed they lack reliable functionality across diverse AR glasses
- Current AR glasses force users to utilize local storage, which means that the correct application may not be installed for a given situation, leading to poor UX
- No known network architecture to support a common and scalable AR experience
- AR applications are co-located on a device and/or with the AR glasses
- AR application data requires a large amount of device and/or AR glasses memory space
- No AR application dynamic data access
- Little improvement over just using the AR applications on your phone

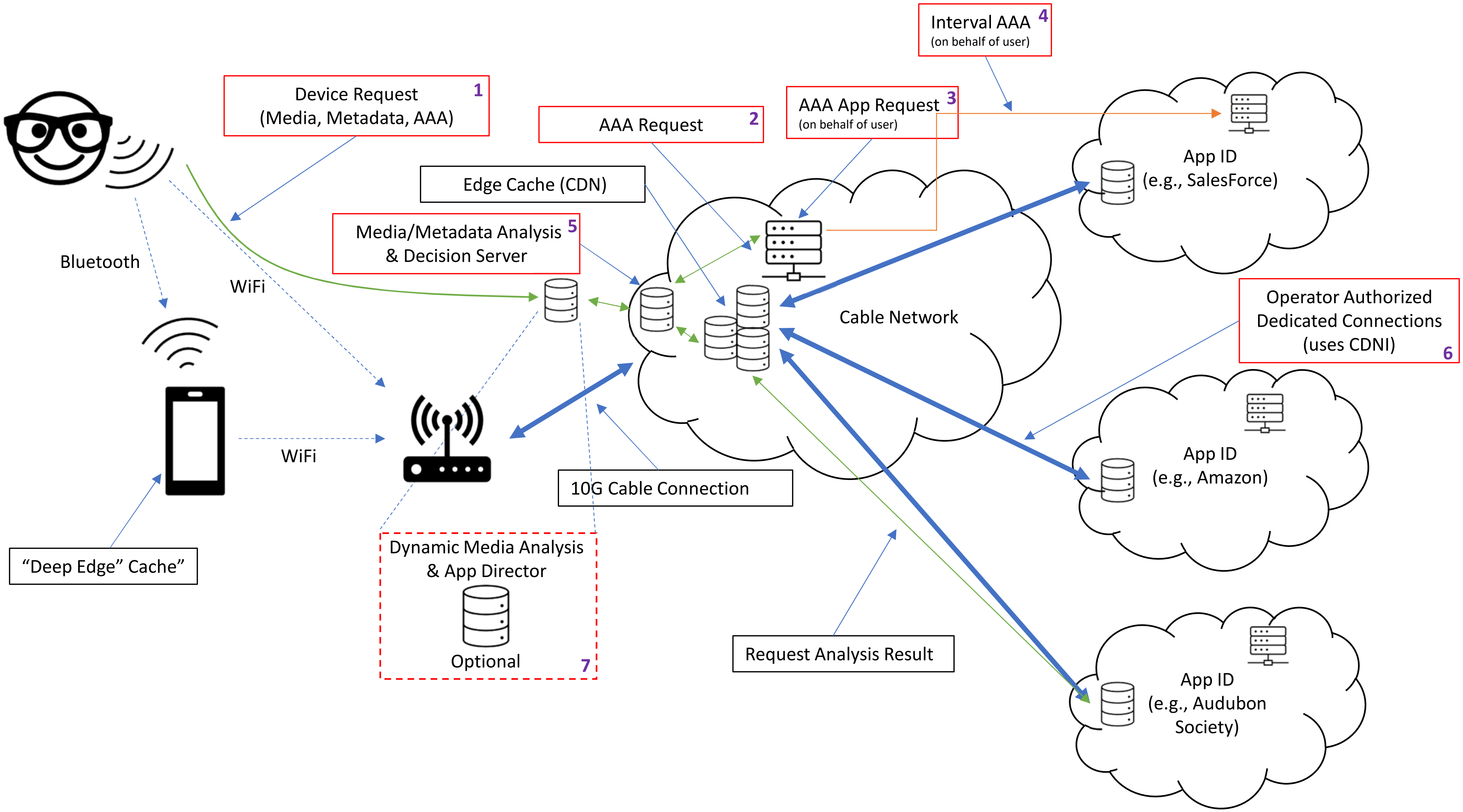
Value Proposition: The future of devices requiring real-time application interaction will be enabled by an architecture based on a consistent set of protocols, high speed, low latency connections, and edge caching. This innovation positions the cable industry to enable and realize this future for these devices and users by building a standardized set of protocols and architecture for any application to work with any device over a highspeed network. An architecture such as this will allow for off-loading processing from the devices allowing device manufacturers to create innovative devices in the market. This will increase competition and ultimately benefit the subscribers. In addition, this architecture opens a wide range of new business strategies and opportunities for revenue generation to cable operators by creating the ability to sell preferences to apps through an innovative media analysis server and app cache.

Hypothesis and Supporting Assertions

The cable industry is well positioned to provide an open scalable media analysis and caching architecture, APIs, and network to provide consumers and application vendors an interoperable platform for devices and applications.

- Primary Assertions
  - Market/Supply Side
    - Device manufacturers will eventually create devices with form factors that are small, lightweight, and dependent on dynamic cloud analysis.
    - Application providers desire to provide smart devices dynamically changing data and information in real-time
    - Architecture provides off-loading processing from the device, enabling a simplified form factor
    - Ecosystems developed and managed by companies such as Google, Apple, Microsoft, and Meta will encourage platform lock-in that discourages open and portable development.

- Device manufacturers and related application vendors prefer a consistent protocol for scalability and interoperability
  - User Behavior and Preferences
    - Users of smart devices will desire the ability to analyze objects in real-time as demonstrated by Android phones and the Google Lens app
    - Users will desire interoperability between applications and their preferred devices
    - Users prefer dynamically updated data available based on real-time object analysis
  - Network Performance
    - The future deployment and use of this system architecture will intersect with the wide scale availability of 10G HFC, 5G UW (or 6G) and Wi-Fi 7 public networks
    - A lot of data will eventually traverse networks to/from these smart devices, and this will ensure cable operators are an integral part of the ecosystem
    - Architecture provides increased speed and low latency for device requests and responses
- Secondary Assertions
  - The APIs could be adapted to support applications, such as AR, VR and MR.
  - An example market opportunity, the AR market* is expected to dramatically grow in the next 3 – 5 years



Device Request
(Media, Metadata, AAA)
1
AAA Request
2
AAA App Request
(on behalf of user)
3
Interval AAA
(on behalf of user)
4
Edge Cache (CDN)
Media/Metadata Analysis
& Decision Server
5
Bluetooth
WiFi
WiFi
Cable Network
App ID
(e.g., SalesForce)
Operator Authorized
Dedicated Connections
(uses CDNI)
6
App ID
(e.g., Amazon)
10G Cable Connection
"Deep Edge" Cache
Dynamic Media Analysis
& App Director
Optional
7
Request Analysis Result
App ID
(e.g., Audubon
Society)

Metadata

- Application ID (What app are you querying?)
  - Amazon
  - Audubon Society
  - SalesForce
- Geo-location Coordinates
- Media Type (e.g., Audio, Video, Picture)
- Time of Day

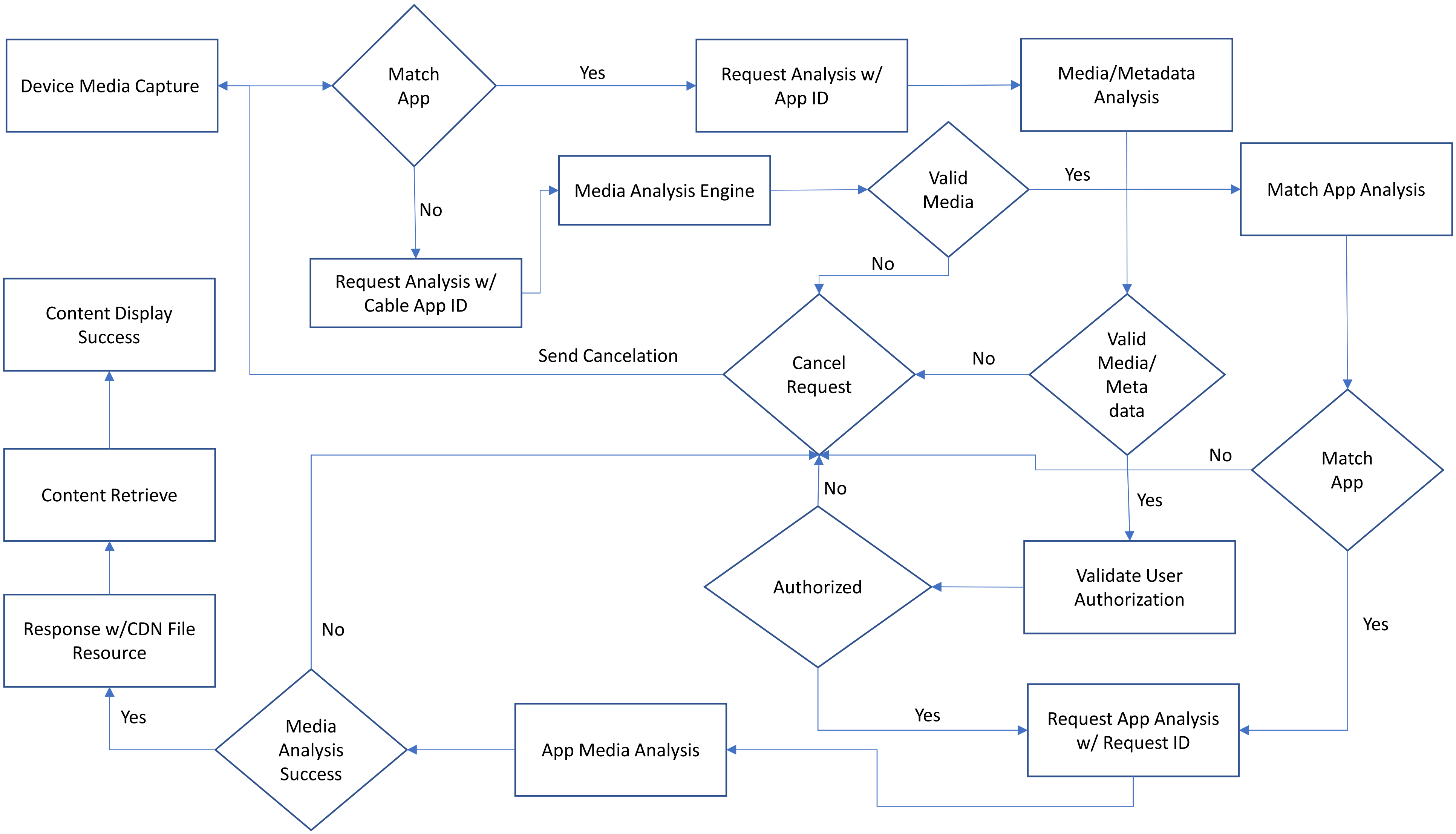
Analysis & Decision Server

- Verify media is valid
- Verify metadata is valid
- Extract App ID
- Extract AAA & Validates AAA
- Sends media to App for AI/ML analysis (verify transfer complete)
- App sends media to operator CDN via CDNI
- App sends a result (metadata)
- Server requests CDN for result

Dynamic Media Analysis & Application Director

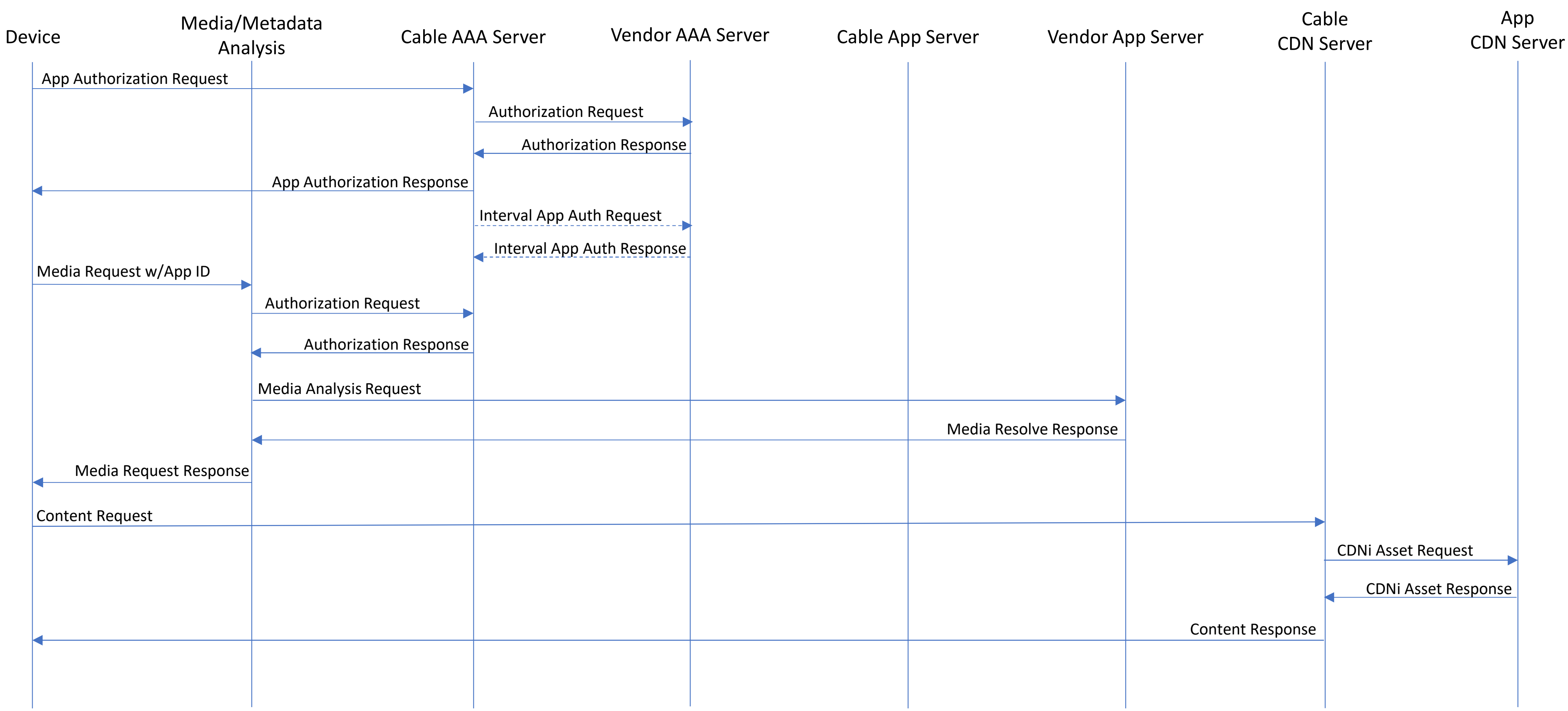
(In lieu of loading full application on device or completely cloud based)

- Skinny app "Generic Look-around" resides on device
  - Operator would need to do analysis on this media
- Full apps "specific app" exist on app cache
- Initial query goes to Decision Server/Specific App/CDN and initial response is rendered by virtual app
- Additional requests or preference loads full specific app to device
- Objective: Basic analysis and suggesting app for "more information", could sell "first app" awareness



Device Media Capture
Match App
Yes
Request Analysis w/ App ID
Media/Metadata Analysis
Media Analysis Engine
Valid Media
Yes
Match App Analysis
No
Request Analysis w/ Cable App ID
No
Content Display Success
Send Cancelation
Cancel Request
No
Valid Media/ Meta data
No
Match App
No
Yes
Content Retrieve
Authorized
Validate User Authorization
Yes
No
Response w/CDN File Resource
Yes
Yes
Request App Analysis w/ Request ID
Media Analysis Success
App Media Analysis

Sequence Diagram – App ID Known



Sequence Diagram – App ID Unknown