Scope Document

Video Ad Server Integration

1. Introduction

This document details the scope of the video ad server integration, based on the **AntPath Ads** solution, into the mobile application for Android and iOS platforms. The integration includes the insertion of ads in **pre-roll, mid-roll, post-roll, and banner** formats, managed dynamically through the API and SDKs of **AntPath Ads**. Additionally, the requirements regarding events, tracking, compatibility with VAST/VPAID players, and segmentation and frequency control rules (capping, geolocation, and content segmentation) are defined.

2. Scope for Product and Development Teams

2.1. General Objective

Define the functional scope of the video ad server integration in the application, covering the insertion and dynamic management of ads in video format (pre-roll, mid-roll, post-roll) and banners, as well as the handling of events, tracking, and segmentation to optimize the user experience.

2.2. Integration Description

Ad Formats:

- **Pre-roll**: Ads shown before the main content.
- **Mid-roll**: Insertion of ads during the playback of long videos.
- **Post-roll**: Ads at the end of the content.
- **Banners**: Integrated into the app interface in defined areas (e.g., in menus or during pauses).

Dynamic Management:

- The app will dynamically connect to the **AntPath Ads** ad server to request ads in real-time, adapting to the user's context and the app section.
- Predefined segmentation and frequency control rules will be configured and sent with each request to ensure an optimal experience.

User Flow:

- When starting the app and accessing video content, a pre-roll ad will be requested.
- During lengthy content, mid-rolls will be inserted at predefined points.
- At the end, a post-roll ad may be shown.
- Banners will be distributed in strategic areas of the interface without interrupting the user experience.

2.3. Tracking and Metrics

- The app will automatically register the following events:
- **Impressions**: Every time an ad is loaded.
- **Clicks**: User interaction with the ad.
- **Complete View**: Confirmation that the video ad was viewed in its entirety.
- **Abandonment**: Record of cancellation or interruption during viewing.
- This data will be sent to the **AntPath Ads** API for analysis and optimization, allowing adjustment of segmentation and frequency control strategies.

2.4. Recommendations on Video Players

- It is recommended to evaluate and integrate a VAST/VPAID compatible player for the delivery of video ads, ensuring compatibility with high-quality video campaigns.

Suggested Options:

- **JWPlayer**: Ideal for robust integrations with broad compatibility.
- **Video.js**: Flexible and open-source solution that can be extended with plugins.
- The development team should determine which option best fits the app's architecture and requirements.

2.5. Segmentation and Frequency Control Rules

- **Capping**: Set limits to prevent a user from receiving more than a certain number of ads in a specific period, protecting the user experience.
- **Geographic Segmentation**: Configure the system so that ads are targeted according to the user's location, using geolocation data.
- **Content Segmentation**: Allow the delivery of ads according to the type of content being viewed, optimizing relevance for the user.

2.6. Delivery and Testing

Development and Testing Environment:

- An integration environment will be configured using the IPs and servers provided for **AntPath Ads**.
- End-to-end tests will be performed to verify the correct dynamic loading of ads, the operation of event tracking, and integration with the video player.

Documentation and Support:

- The **AntPath Ads** technical guides and documentation will be provided, along with implementation examples.

Final Validation:

- Once the SDKs are integrated and the segmentation rules are configured, a validation phase with users and data analysis will be carried out to ensure that the

experience meets the requirements.

3. Structured Technical Version (API Spec Type)

3.1. Objective

Provide detailed technical specifications for the integration of the **AntPath Ads** SDKs and the interaction with its API for the dynamic management of ads in the app.

3.2. General Architecture

AntPath Ads Endpoints:

- The IPs and URLs of the production environment will be provided to make HTTPS requests.

SDKs Used:

- **iOS**: *AntPath Ads iOS MRAID SDK*
  - Integrated via CocoaPods or manual framework import.
  - Allows loading MRAID ads and managing callbacks (impressions, clicks, errors, etc.).
- **Android**: *AntPath Ads Android MRAID SDK*
  - Integrated via Gradle (AAR) or manual import.
  - Supports MRAID ads and interaction events.
- **AntPath Ads API:**
  - A RESTful API will be used for requests, reports, and tracking.
  - Responses will be received in JSON format, containing the creative data, segmentation parameters, and tracking URLs.

3.3. Ad Formats

Video:

- **Pre-roll**: Requested and shown before the content.
- **Mid-roll**: Inserted during longer videos.
- **Post-roll**: Shown at the end of the content.

Banners:

- Display ads that can be placed in designated areas within the app.

3.4. Integration Flow and Ad Management

Request and Response:

- The app will send HTTP requests to the **AntPath Ads** API specifying parameters such as location, content type, and segmentation data.
- The response will include the creative in JSON/XML format, with attributes such as

duration, dimensions, and tracking URLs.

Rendering and Display:

- Video ads will be integrated using a VAST/VPAID compatible player (see section 3.6).
- Banners will be rendered in native views or WebViews as appropriate.

Global Configuration:

- Global parameters for capping, segmentation, and frequency rules will be established and included in each API request.

3.5. Events and Tracking

Impressions:

- An event will be sent to the API when an ad is visible.

Clicks:

- Clicks will be recorded and sent to the API via tracking URLs.

Completion and Abandonment:

- Events of complete viewing and ad abandonment will be captured.

Other Events:

- Pause, resume, and error events will be included, as specified in the technical documentation.

3.6. Player Compatibility Requirements

VAST/VPAID Support:

- It is recommended to use players that support these standards to ensure the correct playback of video ads.

Recommended Options:

- **JWPlayer** and **Video.js**, among others, to ensure compatibility with creatives delivered by **AntPath Ads**.
- The integration should validate that the chosen player can correctly process the data and formats sent by the API.

3.7. Security and Authentication

- All communications will be made via HTTPS.
- Tokens or API keys will be used to authenticate requests, ensuring data integrity and security.

4. System Topology and Ad Delivery Flow

The ad delivery system in the **AntPath Ads** platform follows a structured flow to ensure optimized and targeted ad placements within a mobile application. The following topology outlines the key stages involved in the process, from user interaction to data tracking and analysis.

This structured ad delivery topology ensures **efficient, data-backed advertising** within the mobile ecosystem. The combination of **real-time decision-making, CDN optimization, and event tracking** enables seamless ad placements while maximizing user engagement and ad revenue.

---

1. User Interaction & App Request

The process begins when a **user opens the mobile application (Android/iOS) and starts consuming video content**. At this point, the application identifies the need for an ad and sends a **real-time API request** to the ad server. This request contains critical data, such as:

- **User metadata** (location, device type, session context).
- **Content information** (type of media, category).
- **Advertising parameters** (frequency capping, segmentation tags).

---

2. Ad Server Processing & Selection

The **AntPath Ads Server** receives the API request and processes it in real time using dynamic selection algorithms. The server applies **targeting rules** based on:

- **User segmentation** (age, location, behavior).
- **Campaign settings** (bid strategy, ad frequency).
- **Ad format compatibility** (banner, video, interactive).

Once an ad is selected, the server responds with a **JSON payload** containing:

- **Ad creative details** (media URL, tracking parameters).
- **Playback specifications** (duration, format).

---

3. Ad Playback & CDN Delivery

Upon receiving the ad response, the **mobile application processes the JSON payload** and requests the actual ad content via a **CDN (Content Delivery Network)**. The ad is then loaded and displayed to the user through an **integrated video player SDK** (e.g., **Video.js, VAST/VPAID** compatible).

Key Processes:

- **CDN ensures fast and reliable ad delivery.**
- **Playback occurs in a seamless manner without disrupting user experience.**

---

4. User Interaction & Event Tracking

Once the ad is displayed, user interactions trigger **event-based tracking**, capturing actions such as:

- **Impressions** (ad fully loaded and displayed).
- **Clicks** (user engages with the ad).
- **Abandonment** (user skips or exits early).

These **event logs** are immediately sent to the **AntPath Ads API** for real-time tracking and data storage.

---

5. Data Analysis & Dashboard Reporting

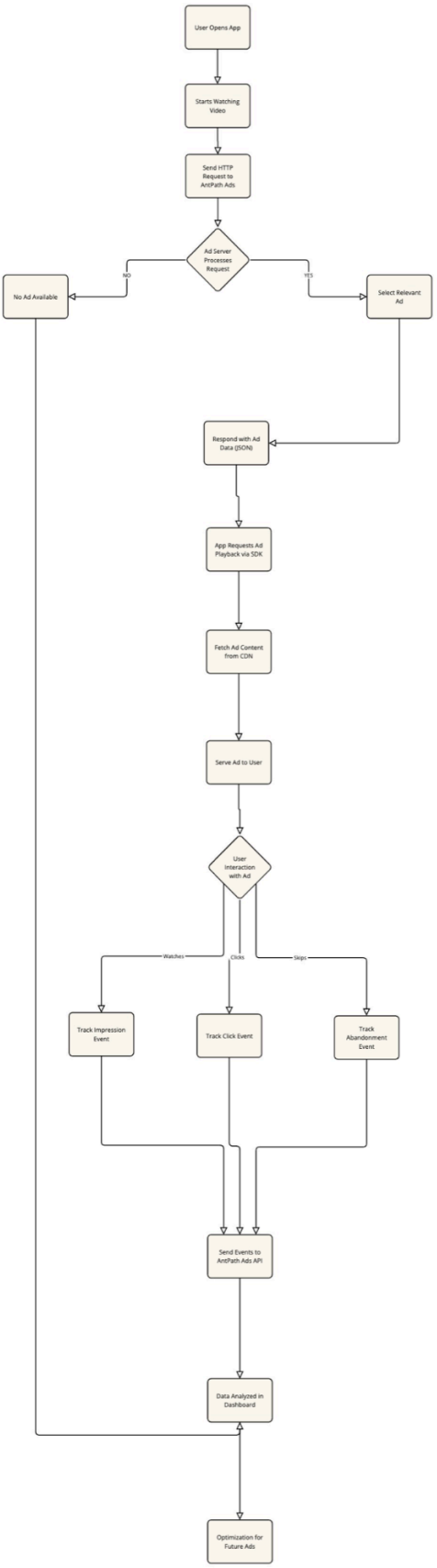
All collected interaction data is **stored and processed** for reporting and campaign optimization. Advertisers can access insights via the **analytics dashboard**, which provides:

- **Performance reports** (CTR, completion rates, engagement).
- **Optimization recommendations** (better targeting, budget allocation).

By leveraging **data-driven insights**, advertisers can refine their campaigns and enhance ad efficiency for better ROI.

ANTPATH Ads

Graphic Topology -> AntPath Ads - Flow



4. Conclusions and Next Steps

- The corresponding IPs and servers of the **AntPath Ads** environment will be provided upon contract advancement.
- This document will serve as a basis for both the technical team (with API specifications and integration details) and the product team, which will manage the user experience and functional requirements.
- It is recommended to start a pilot phase of integration and end-to-end testing to validate the correct dynamic loading, event tracking, and segmentation before going into production.