AdTonos integration for streaming providers

Version	Description	Author	Date
1.0	Initial version	Wojtek Lichota	05/17/2019
1.1	Description of startup parameters Upgrade instructions v1.1.0 release notes	Wojtek Lichota	09/11/2019
1.2	VAST link configuration v1.2.0 release notes	Wojtek Lichota	12/09/2019
1.3	Support for additional ad break markers Authentication configuration v1.3.0 release notes	Wojtek Lichota	01/30/2020
1.3.1	v1.3.1 release notes	Wojtek Lichota	05/20/2020
2.0	Detect intro and outro jingles using fingerprinting algorithm v2.0.0 release notes	Wojtek Lichota	10/15/2020
2.1	Logs endpoint v2.1.0 release notes	Wojtek Lichota	12/09/2020
2.2	Requirements versions bump Installation instruction update v2.2.x release notes	Robert Kawecki Wojtek Lichota	10/01/2021

Version	Description	Author	Date
2.3.0	● Bump requirements versions ● Update installation instruction ● Add new stream configuration parameter ● Update ICY Metadata ad break detection description ● Fix grammar mistakes in the documentation ● Add V2.3.0 release notes	Radosław Malec	01/05/2024
2.4.0	API endpoints similar to icecast v2.4.x release notes	Wojtek Lichota	01/16/2024

Introduction

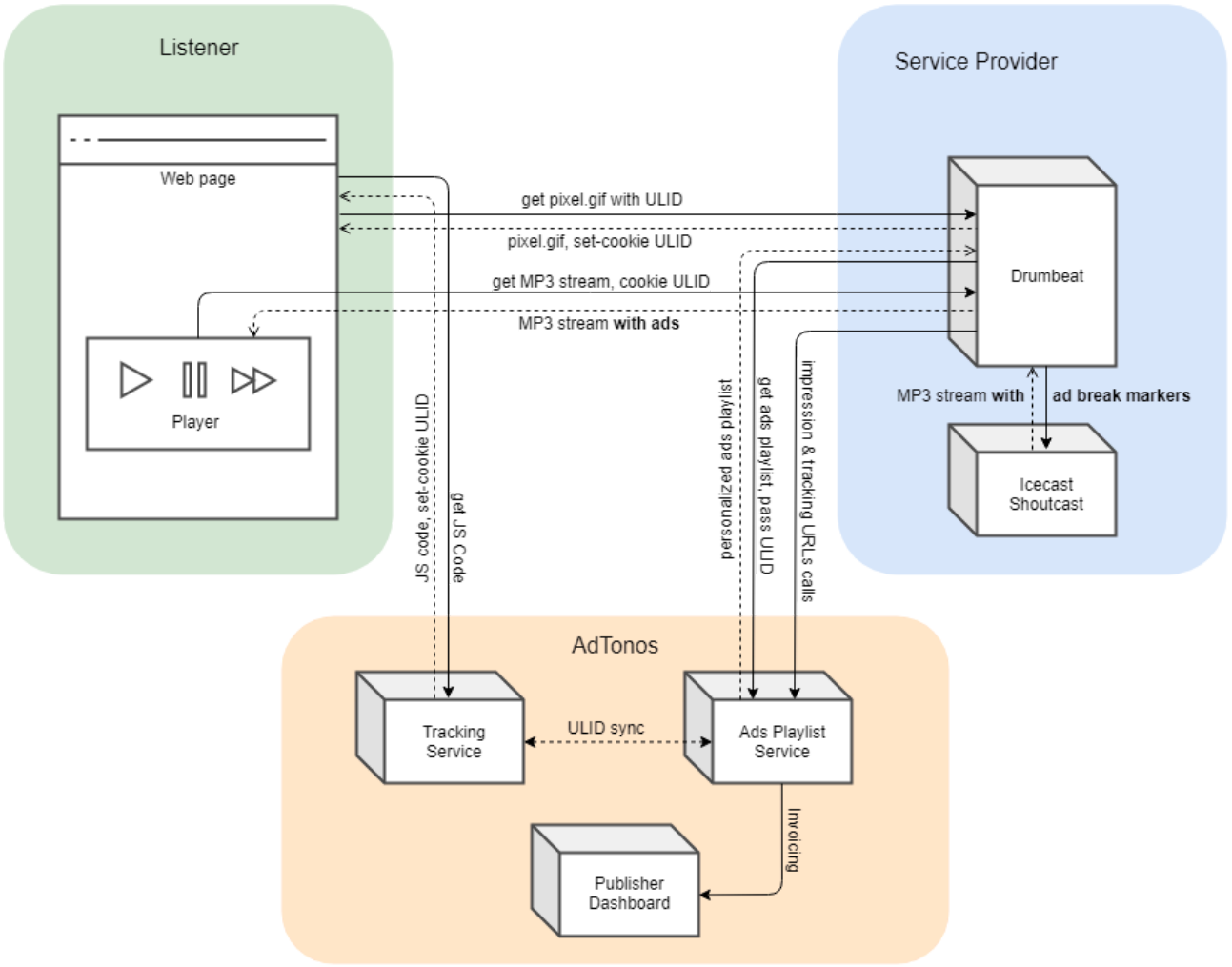
AdTonos solution for online radio broadcast replaces on-air commercial break content with targeted advertisements changing spray-and-pray ATL ads into pay-per-play performance marketing. AdTonos uses server-side ad-stitching to put targeted ads into MP3 stream allowing integration without interfering with existing players. Although server-side ad-stitching service cannot access cookies used in traditional client-side tracking, AdTonos puts JavaScript code on web pages with players to take over this task. This JS code may be used to sync cookies, unique user identifiers or GDPR consents.

Integration overview

This document aims to describe the integration method of AdTonos that streaming providers can use to monetize streams that they already serve. In this document, you can find what on-premise and external components are required to install or configure and from which steps this process consists.

Architecture

The system contains several interoperating components. Some of them are located on the Streaming Provider side (e.g. Drumbeat), others on AdTonos infrastructure.



Streaming provider side

Drumbeat

This on-premise component provided by AdTonos is responsible for:

1. Proxying MP3 stream and inject ads in marked places,
2. Limit maximum concurrent connections to stream,
3. Provided basic listener metadata (e.g. listener IP address, user-agent),
4. Store synced ULID (Unique Listener ID).

The source of the MP3 stream is Icecast or Shoutcast that is already installed on the streaming provider side. Drumbeat is installed just before Icecast/Shoutcast and acts as a multiplying proxy, it means that when multiple listeners will connect to Drumbeat only one connection will be established between Drumbeat and Icecast/Shoutcast. Because of that Drumbeat will be responsible for limiting maximum concurrent connections allowing streaming providers to handle service tiers the same way as before.

The Drumbeat to get advertisement playlist (list of ads to play to the listener) from AdTonos system needs to know ULID (Unique Listener ID) set by AdTonos' JS code. JS code sends this identifier in pixel redirect to the domain on which steam is provided. Drumbeat stores this ULID and uses it later in communication with AdTonos systems.

Icecast/Shoutcast

Streaming media server which provided MP3 streams. One of Drumbeat's ways to recognize where to put ads is analyzing ICY metadata (currently played title). Because of that, it is important to turn on ICY metadata in Icecast/Shoutcast configuration.

AdTonos side

AdTonos Tracking Service

AdTonos Tracking Service is responsible for profiling listeners so that it selects the most suited ads to play. To perform this task AdTonos Tracking Service uses JavaScript code, embedded in a web page where the audio player is located. This JS code mainly synchronizes unique identifiers with third-party providers but can be also used to pass GDPR consents granted by a listener.

AdTonos Tracking Service as a final step sets in cookie ULID (Unique Listener ID) and starts the cookie syncing process with Drumbeat.

User tracking and cookie syncing are fully compliant with GDPR regulations. Collected data is anonymized (we do not store any contact information of a listener) and used only for ads profiling. A listener can opt-out from tracking using https://app.adtonos.com/opt-out.

AdTonos Ads Playlist Service

This service provides Drumbeat a list of ads that should be played to the listener during the next ad break. This is also a central point to collect acknowledgments that a given ad was played to the listener. Data exchange between AdTonos Ads Playlist Service and Drumbeat uses VAST format. VAST (Video Ad Serving Template) is XML-based, industry standard for data exchange regarding advertisement.

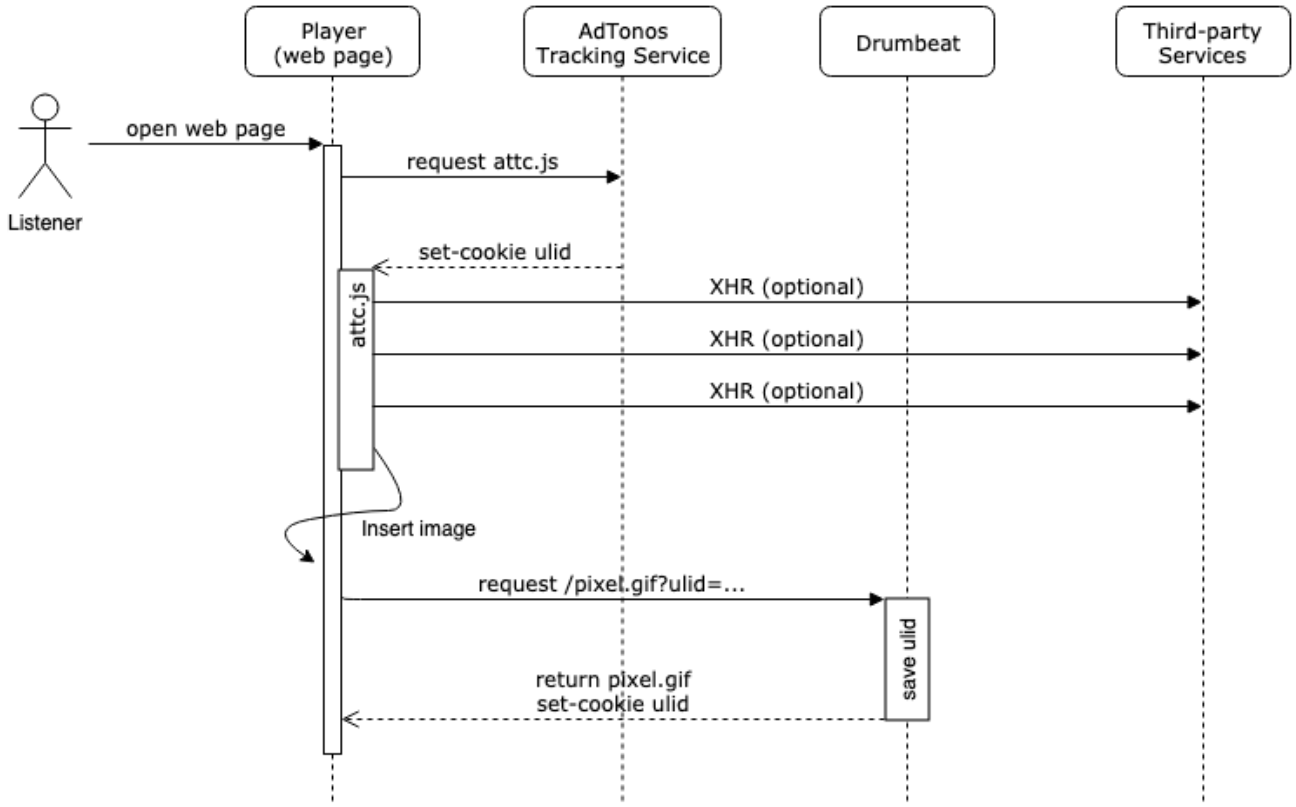
VAST contains Impression and Tracking URLs. Drumbeat as soon as an ad is inserted into the output stream to the listener opens provided urls so that AdTonos can bill advertisers and remunerate publishers.

AdTonos Publisher Dashboard

In AdTonos Publisher Dashboard representatives of streaming providers can check basic statistics and track revenues from ads played in streams. Here they can also download invoices.

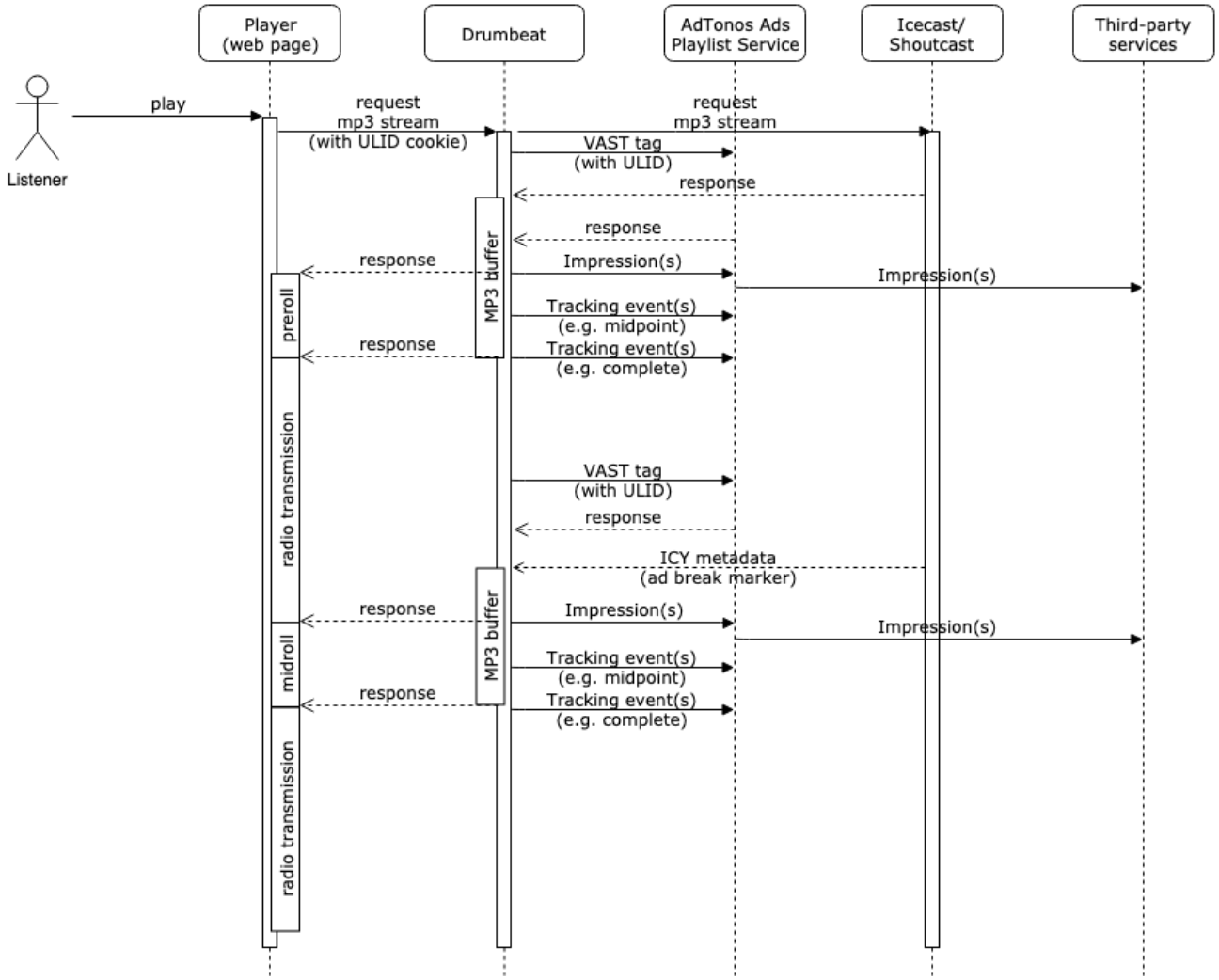
Processes

Listener tracking & ULID syncing



1. Listener lands on the webpage where the publisher's web player is located.
2. Web player page sends JS code request to AdTonos tracking service.
3. AdTonos tracking service sends JS code and sets set-cookie ULID.
4. JS code may send some additional XHR requests based on user actions (e.g. for deep targeting).
5. JS code inserts tracking pixel image into the publisher's webpage. URL to this image contains ULID so that Drumbeat can extract passed ULID and save it for later use
6. Drumbeat returns a pixel image and sets a ULID cookie on Drumbeat's domain.

Playing MP3 stream



1. Listener clicks play on the web player.
2. Web player sends a request for mp3 stream (attaching previously generated ULID cookie) to Drumbeat.
3. Drumbeat passes the request to the streaming provider's Shoutcast/Icecast along with a VAST tag to AdTonos Playlist Ads Service.
4. Shoutcast/Icecast streaming server responds with the stream, Drumbeat sends it to the mp3 buffer.
5. If there is a pre-roll ad available, Drumbeat requests the ad, sends it to the buffer and sends a notification to Playlist Ads Service about the impression. The ad is sent to a web player from Drumbeat's buffer.
6. Information about the impression is passed further to third-party services (DSPs).
7. In the middle of pre-roll playout information about it is updated in Playlist Ads Service for accurate billing.
8. When pre-roll ad playout is finished, appropriate information is sent to Playlist Ads Service to mark the ad as fully played. Along with that, the stream is released from the buffer and sent to the player.
9. During the broadcast VAST tag request is re-requested to prepare an ad playlist during mid-roll and response is sent back to Drumbeat from Adtonos Playlist Ads Service.

10. When an ad break marker (based on ICY metadata) is sent from Shoutcast/Icecast to Drumbeat and a new ad playlist is sent to Drumbeat's buffer.
11. Ads are delivered from Drumbeat to the web player.
12. Information about playout is sent to Drumbeat the same way as for pre-roll.

Ad break detection

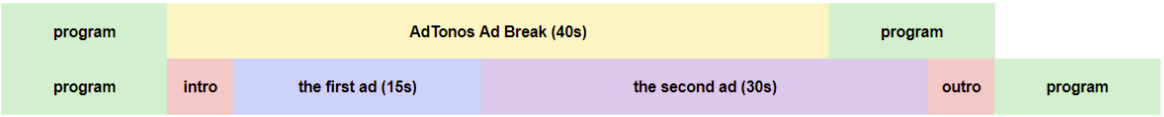
Drumbeat can detect where to put ads using two mechanisms:

- ICY Metadata analysis
- Fingerprinting algorithm

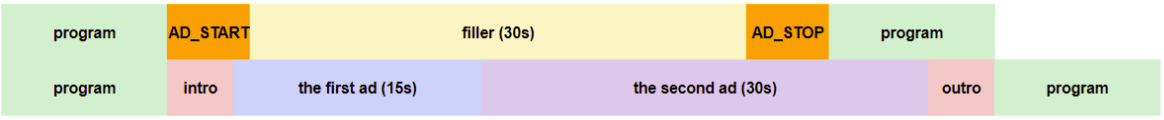
Ad break detection using ICY Metadata

Drumbeat recognizes where to insert ads by analyzing the stream's currently played song. So to mark where an ad break is DJs or stream owners have to add to the playlist special ad break marker files with MP3 ID3 title tag set. When Icecast/Shoutcast streams MP3 they pass ID3 tags in ICY metadata.

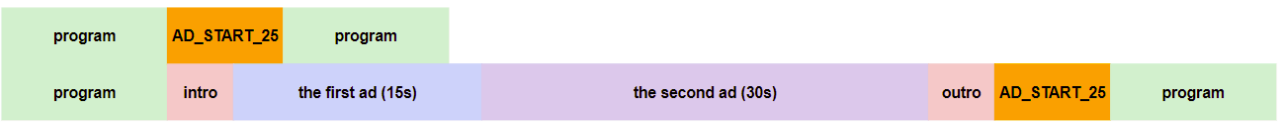
Drumbeat detects a file titled 'AdTonos Ad Break' and replaces the whole file with personalized ads. The marker file can't be longer than `maxAdsBreakLength` parameter.



Drumbeat also recognizes ad breaks marked with files titled AD_START and AD_STOP. In such a case it will replace any content between marker files including them. If AD_STOP will not be detected in time configured in `maxAdsBreakLength` parameter, Drumbeat will stop ad break after the currently played ad.



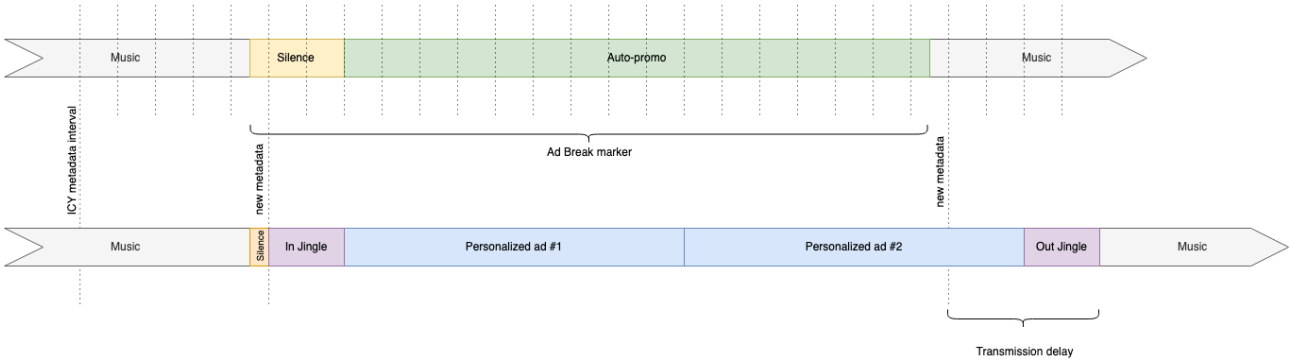
Drumbeat supports also AD_START_XX files, where XX is a two digit number (eg. AD_START_60) describing the length (in seconds) of the ad break. In such a case Drumbeat will insert (not replace like in other cases) an ad break as soon as the title AD_START_XX is detected. Rest of the file AD_START_XX will be played after the ad break. This approach delays the following stream by the length of ad break so it's not recommended for live streams. This mode is enabled only when `insertMode` parameter is set to true in stream configuration. Support of such files is also **mutually exclusive** with support of other markers (`insertMode` disables support of two other markers).



Ad break detection delay

ICY metadata protocol is MP3 stream extension which adds additional packets to MP3 stream which contains information such as the currently played tune. ICY metadata are added to MP3 stream in regular intervals of 8kb or 16kb (for example Icecast uses fixed 16kb value, see: https://github.com/karlheyes/icecast-kh/blob/master/src/format_mp3.c#L52). Because ICY metadata is not sent immediately after a new song is played but in the next interval, there can be some delay between played song and metadata. This delay depends on MP3 stream quality. For example in the stream with quality 64kbps maximum delay can be 2s (because in 1s 8kb are sent), in 128kbs - 1s of delay, in 196kbps - 650ms and in 320kbps - 400ms.

Drumbeat as soon as it receives ICY metadata with the title set to ‘adtonos ad break’ checks for pending ads playlist. If there is at least one ad to play to the listener Drumbeat immediately interrupts proxying of mp3 frames, opens the in-jingle file from the configuration, puts it into the stream and then inserts cached ads. After the ad break (when there is new ICY metadata with a title other than ‘adtonos ad break’) it puts out-jingle and starts normal transmission. In case of empty ads playlist, Drumbeat does not insert any ad and plays original marker file content.



Because of these two behaviours (ICY metadata delay and marker file playout in case of no ads) marker file should be specially crafted to avoid glitches (e.g. suddenly interrupted sounds) or listener confusion. Marker file should:

- starts with silence (400ms for 320kbps streams up to 2s for 64kbps streams). This will help to avoid situations that Drumbeat interrupts any sound during switch do ad break,
- contain some meaningful sounds inside (e.g. radio jingle, auto-promo, generic ad contracted directly with the broadcaster or any song or tune),
- have MP3 ID3 title tag set to ‘adtonos ad break’,
- have a length equal to desired ad break length (min 15s, max 5 min).

Ad break detection using fingerprinting algorithm

Audio fingerprinting is a way of preparing a fingerprint - an individual, digital representation of an audio sample. By comparing two fingerprints of two audio samples we can find out if those samples are similar to each other. Drumbeat at startup prepares fingerprints of intro and outro jingles. Next starts to proxy the stream and constantly prepares fingerprints of incoming frames. When the incoming fingerprint matches with intro fingerprints it starts to insert ads.

So to mark where an ad break is DJs or stream owners have to add to the playlist special ad break markers (intro and outro jingles). The same markers have to be put into Drumbeat’s configuration.

Installation

JS code installation

It is necessary to embed JavaScript code on the website containing the player. This code is used to profile ads and synchronise ULID.

Embedding involves inserting the JS code at the end of the page, just before the closing tag `</body>`.

```
<script async defer src="https://play.adtonos.com/attc-{name}.min.js"></script>
```

where `{name}` can be found in AdTonos Dashboard.

JS code supports TCFv1 and TCFv2 standards (GDPR popups). It can extract listeners' consents automatically if the publisher's website uses TCFv1 or TCFv2. Otherwise the publisher should pass the listener's disagreement on profiling. To do this it should be done before (above) mentioned JS code:

```
<script>
window.adtonosOptOuts = ['*'];
</script>
<script async defer src="https://play.adtonos.com/attc-{name}.min.js"></script>
```

Example JS code when another variable specifying the consent for profiling is available:

```
<script>
if (!gdprAllowPersonalizedAds) {
  window.adtonosOptOuts = ['*'];
}
</script>
<script async defer src="https://play.adtonos.com/attc-{name}.min.js"></script>
```

Drumbeat installation

Examples shown in this section assume that Drumbeat is installed on Ubuntu 18.04 LTS system. Drumbeat can be installed on other Linux machines. In case of any installation problems we offer our help, from adjusting installation steps to your environment to remote installation by our experienced DevOps.

Pre-requirements

Drumbeat to install requires:

- Linux environment (also Windows in the future),
- Node.js 18.x (`v18.12.1` is recommended),
- NPM 5.x+ (`6.4` is recommended),
- FFmpeg 3.x or 4.x (`4.2.4` is recommended).

Installation

We recommend configuring a separate user on which Drumbeat will be installed and run:

```
sudo adduser drumbeat
```

Ubuntu 20.04 (LTS) installation example with `nvm` used for obtaining a supported Node.js version:

```
sudo apt update
sudo apt install ffmpeg
sudo apt install build-essential
sudo su - drumbeat
wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
source ~/.bashrc
nvm install v18.12.1
npm install -g npm@6.4
```

Drumbeat package includes all necessary dependencies, so installation is as simple as downloading package and running install script:

```
cd /home/drumbeat
mkdir config fx logs
wget https://adtonos-drumbeat.s3.amazonaws.com/adtonos-drumbeat-x.x.x.tgz
npm install adtonos-drumbeat-x.x.x.tgz
```

To verify installation you can use `npm start` command:

```
cd node_modules/adtonos-drumbeat
npm start
```

After this step Drumbeat will listen on port 8500 for connections, so to test it you can open http://host_or_ip:8500/pixel.gif in your browser.

Drumbeat should be started as a system service. On Ubuntu 20.04 you can use systemd script:

```
sudo cp ./config/adtonos-drumbeat.service /etc/systemd/system
sudo systemctl daemon-reload
sudo systemctl enable adtonos-drumbeat
sudo systemctl start adtonos-drumbeat
```

Installation you can check using by opening http://host_or_ip:8500/pixel.gif in your browser.

Configuration

Startup parameters

Basic parameters needed to start Drumbeat are located in `adtonos-drumbeat.service` file.

Configuration parameter	Description

`PORT`	Port on which Drumbeat will listen for connections.
`CONFIG_DIR`	Directory in which streams configuration is located.
`LOG_DIR`	Directory in which access log (same format as Icecast-kh) is saved.
`ADS_DIR`	Directory used as a cache for ads audio files.

Streams configuration

At first copy sample configuration

```
cd /home/drumbeat
cp -r node_modules/adtonos-drumbeat/config/* /home/drumbeat/config
cp -r node_modules/adtonos-drumbeat/jingles/* /home/drumbeat/fx
```

In the config directory (`/home/drumbeat/config`) you should place one config JSON file per stream.

After saving the config file, Drumbeat will automatically detect changes and reload it without any further user actions. During reload all listeners will be disconnected.

Configuration parameter	Description
`streamOutput`	Name used in output URL (e.g. when `streamOutput` is set to “demo” then you can listen to proxied stream on http://host:8500/demo).
`streamSource1`	URL of source stream.
`fallback1Url`	Backup stream source URL.
`fallback2Url`	Backup stream source URL.
`vastUrl`	URL to AdTonos VAST Tag generated for the owner of a stream.
`maxAdsBreakLength`	Maximum length of ad break.
`hasPreroll`	Boolean parameter that activates pre-roll ads.
`prerollLength`	Amount of ads in the pre-roll.
`maxListeners`	Maximum number of concurrent listeners.
`playJingles`	Turn on/off jingles inclusion during ad break.
`useFingerprinter`	Turn on/off ad break detection using fingerprinting algorithm.
`jingles`	File paths to jingle files. This parameter is required when `useFingerprinter` or `playJingles` is turned on. See *Jingles Configuration* section.
`authentication`	Optional parameter. See *Listener Authentication* section.

`insertMode`	Turn on/off support of AD_START_X files. NOTE: This parameter also turns off/on support of other ad break markers. Defaults to false (AD_START_X is not supported)

Jingles Configuration

Intro and outro jingles are configured in *in* and *out* sub-objects or *jingles*. Each object consists of:

Configuration parameter	Description
`files`	This parameter is required when `playJingles` is turned on. One of the files configured in this object will be inserted into the stream when ad break starts or ends. Names in this object define the quality of the file - using template bitrate_samplingrate_channels (e.g. `64_44100_m`, `320_48000_s`). Values are filenames of files from `jingles` directory. Use `transcode.sh` script to prepare jingle versions in all qualities.
`file`	This parameter is required when `useFingerprinter` is turned on. File used to prepare fingerprints for intro/outro detection. File should be located in the jingles directory.
`threshold`	This parameter is required when `useFingerprinter` is turned on. Value of threshold found by `autotune.sh` script. Take the value around 80% of the range between min and max.
`customDelay`	This parameter is required when `useFingerprinter` is turned on. Value of delay found by `autotune.sh` script. Take the value around 80% of the range between min and max.

Listener Authentication

Listener authentication is a feature of Drumbeat which allows you to secure a certain stream such that in order to listen, a listener must pass some verification test. Configuration is very similar to Icecast2 implementation (https://icecast.org/docs/icecast-latest/auth.html).

There are two types of authentications available:

1. htpasswd

```
"authentication": {
  "type": "htpasswd",
  "filename": "~/drumbeat/config/teststream.htpasswd",
  "allow_duplicate_users": false
},
```

Parameters:

- `filename` - specifies the name of the file in which logins and passwords are stored. This file can be generated using the `htpasswd` command. Note that this file needs to exist adding this line to configuration.
- `allow_duplicate_users` - when set to `false` will prevent multiple connections using the same username.

2. URL

```
"authentication": {
  "type": "url",
  "listener_add": "http://example.org/",
  "listener_remove": "http://example.org/",
  "username": "user",
  "password": "pass",
  "auth_header": "HTTP/1.1 200 OK",
  "headers": ["x-forwarded-for", "Accept-Language"],
  "header_prefix": "ClientHeader."
},
```

Parameters:

- `listener_add` - when a listener connects, before anything is sent back to them, this request is processed. The default action is to reject a listener unless the auth server sends back a response header which may be stated in the `header` option.
- `listener_remove` - when a listener disconnects (or is kicked), just after closing the connection a GET request is sent to the URL configured in this parameter. Parameter is optional.
- `username` and `password` - credentials added as basic auth when calling `listener_add` and `listener_remove` URLs.
- `auth_header` - expected response header to be returned that allows the authentication to take place (default is: `icecast-auth-user: 1`, but it could be anything you like, for instance `HTTP/1.1 200 OK`).
- `headers` - list of HTTP headers provided by Drumbeat which should be passed to the authentication service. Those headers are prepended by the value of `header_prefix` and sent as POST parameters.
- `header_prefix` - prefix used for passing client headers.

Logs

Drumbeat creates two types of logs:

1. Access log (`LOG_DIR/access.log`)
   Drumbeat writes information about client requests in the access log right after the request is processed. In case of MP3 streaming log entry is added when client is disconnected from the stream. Log entry has the same format as Icecast-KH.
   Log entry in access log contains:
   - listener IP address
   - end of request timestamp (time when listener disconnected)
   - request method (e.g. GET)
   - request path
   - protocol (e.g. HTTP/1.1)
   - response status (e.g. 200 or 404)
   - size (bytes transferred to a listener)
   - referrer, if any
   - listener's browser user agent
   - session time (in seconds)
2. Operational log (`LOG_DIR/drumbeat.log`)
   Drumbeat writes debugging information on every interesting event happening in the

system (e.g. configuration changes, client connect/disconnect, VAST communication details).

API

Drumbeat returns usage statistics (how many listeners are connected to streams) when opening http://drumbeat-ip:8500/stats URL. Response is in JSON format. Example:

```
{
    "drumbeat-demo":{
        "listeners":7
    },
    "main-stream":{
        "listeners":69
    }
}
```

On the URL http://drumbeat-ip:8500/logs Drumbeat returns logs of currently connected listeners. The format is:
`#IP-address@start-time@user-agent@listener-id@@`

Sample response:

```
#192.168.0.100@7389483@WinampMPEG/5.80, Ultravox/2.1@01ES3YJS8Y0W2QGDAZ7FZM2C9K@@
#10.0.0.100@97369220@Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/87.0.4280.88
Safari/537.36@01EKDKE0HPV6AWHZKJKN7E2C9K@@
```

On the URL `http://drumbeat-ip:8500/admin/listclients?mount=/stream-name` Drumbeat returns logs of currently connected listeners. The format is based on XML and compatible with Icecast output.

It's possible to force listener to disconnect (kick) from the stream by calling URL `http://drumbeat-ip:8500/admin/killclient?mount=/stream-name&id=listenerId` Response is in XML and compatible with Icecast output.

API that shows listener details (e.g. IP address) so `/logs`, `/admin/listclient`s, `/admin/killclient` requires authentication taken from htpasswd file. To configure it please run command

```
htpasswd /home/drumbeat/config/admin.htpasswd admin
```

Drumbeat update

Upgrade procedure

```
cd /home/drumbeat
wget https://adtonos-drumbeat.s3.amazonaws.com/adtonos-drumbeat-x.x.x.tgz
npm install adtonos-drumbeat-x.x.x.tgz
# update configuration - check release notes below
sudo systemctl restart adtonos-drumbeat
```

Release Notes

1.0.0

Initial version.

1.1.0

During this update `adtonos-drumbeat.service` file has changed. This file should be copied again from `/home/drumbeat/config/adtonos-drumbeat.service` or upgraded manually (especially environment variables).

1.2.0

VAST Tag url is now configurable for each stream. Because of that `VAST_URL` from `adtonos-drumbeat.service` file is not used anymore but `vastUrl` parameters have to be added to config file for each stream.

1.3.0

Support for AD_START / AD_STOP and AD_START_XX marker files. Listener Authentication feature was added. Configuration doesn't have to be migrated. All new parameters are optional.

1.3.1

Bug Fixes. Configuration doesn't have to be migrated.

2.0.0

New configuration parameters for ad break detection using fingerprinting algorithm. Old configuration doesn't have to be migrated.

2.1.0

List of currently connected listeners available via API endpoint. Configuration doesn't have to be migrated.

2.2.x

Fallback URLs support, bug fixes, minor improvements. Configuration doesn't have to be migrated.

2.3.0

Added `insertMode`, which subsequently separates support of AD_START_X type of files (in ICY stream) from other markers. Adding the flag to the config and setting it to true is necessary if one wants to use AD_START_X files as ad break markers.
Fixed issues with streamed audio quality loss.

2.4.0

Copy initial admin.htpasswd file from the package to the configuration folder

```
cp node_modules/adtonos-drumbeat/config/admin.htpasswd /home/drumbeat/config
```

Known problems & solutions

Ad breaks are not injected at the right moment in ICY-metadata stream

Description: When ICY metadata interval is set to high value it is possible that the time of ad break injection may differ significantly from its real place causing the listener to hear marker/placeholder audio instead of ad for split second.
Solution: Decrease metadata interval. 4096 as ‘mp3-metadata-interval’ is known to work with 256kbps streams. Problem is inherent to the nature of metadata based ad break detection - Drumbeat cannot tell the exact time of ad break if data it has is not in sync with the stream source.

Help

In case of problems or questions regarding integration, please contact us by email: support@adtonos.com.