Cloud audio broadcasting synthesizer

(SunWave Broadcasting System)

This is a radio automation solution designed to provide feature rich broadcasting with the storage and processing of all necessary information in a secure «cloud», without additional complicated technical requirements and involved costs. Allows you to reach modern broadcast quality standards and automate the entire process of broadcasting from the source audio file pull to “on air” stage including seamless live-performances with a few clicks.

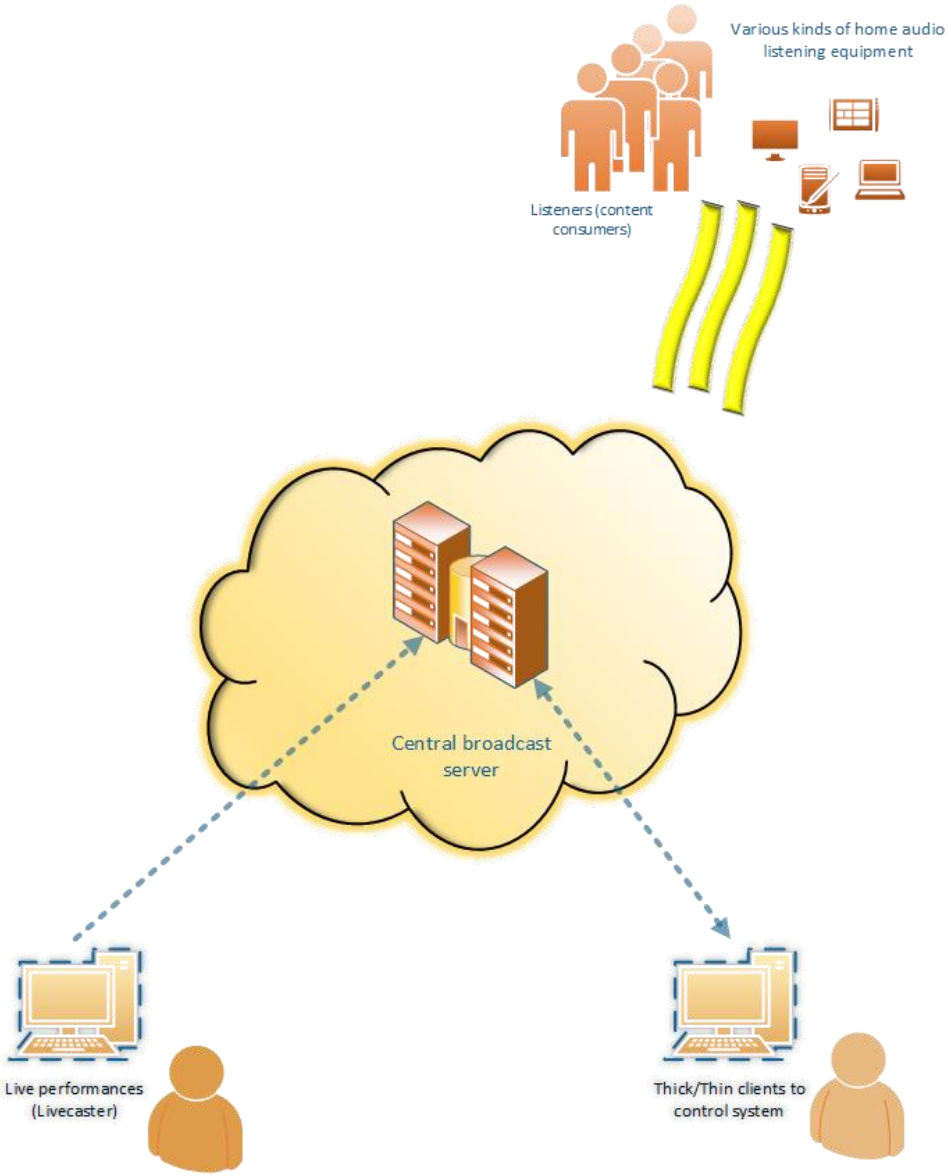
OR you can effectively and easy automate sound accompaniment and live performance in your café, bar, kiosk, mini-shop and other places all with a few clicks.

It is based on some innovative design principles that is different from what can be observed in other comparable software in this category.

Consists of:

1. **Backend broadcasting server (central, where SWBS is installed)**
2. **Control center workplace (where you are managing operation). Can be thin (Web) or thick clients.**
3. **Clients for different platforms to organize live realtime performances**

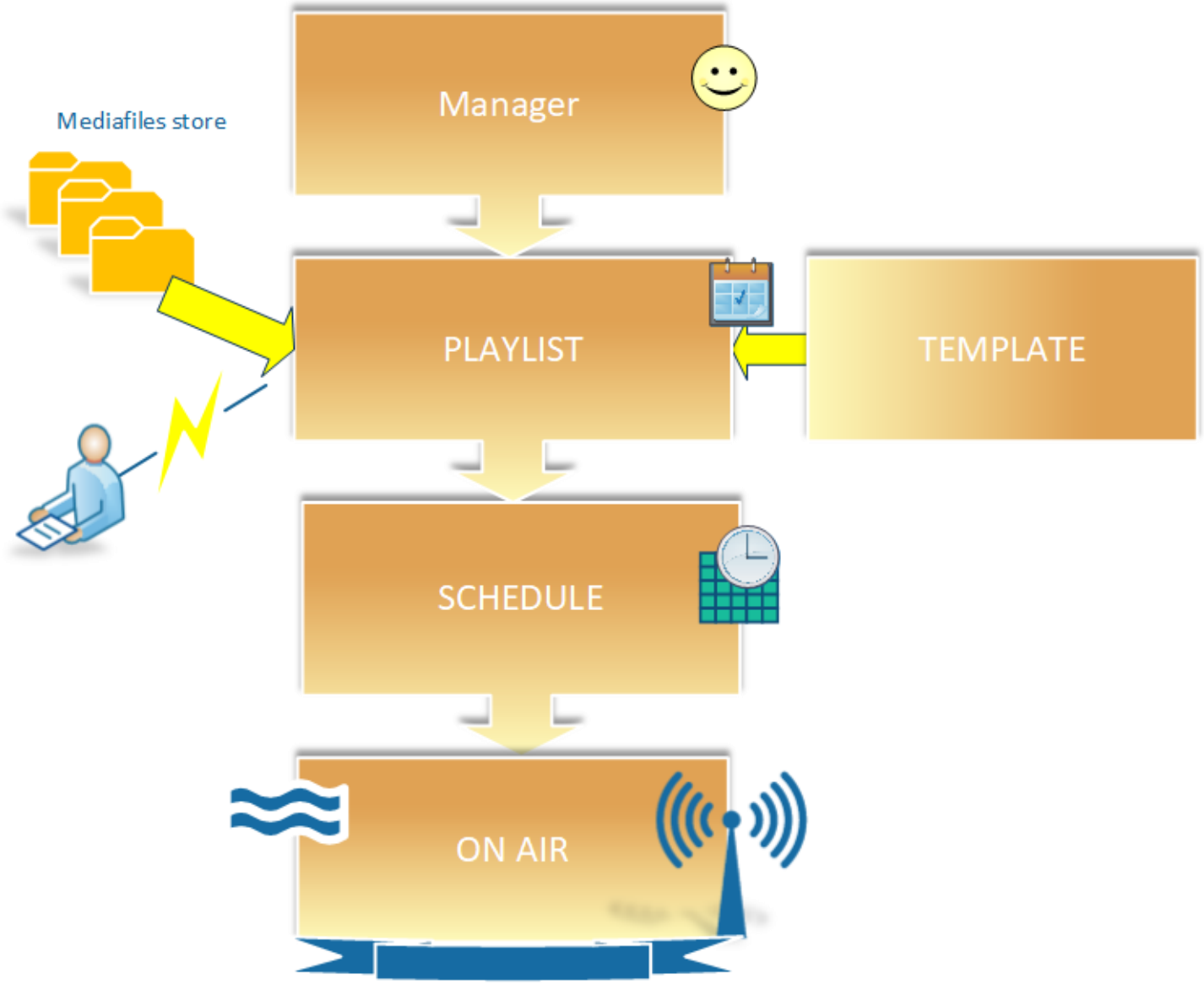
General plan of entire system



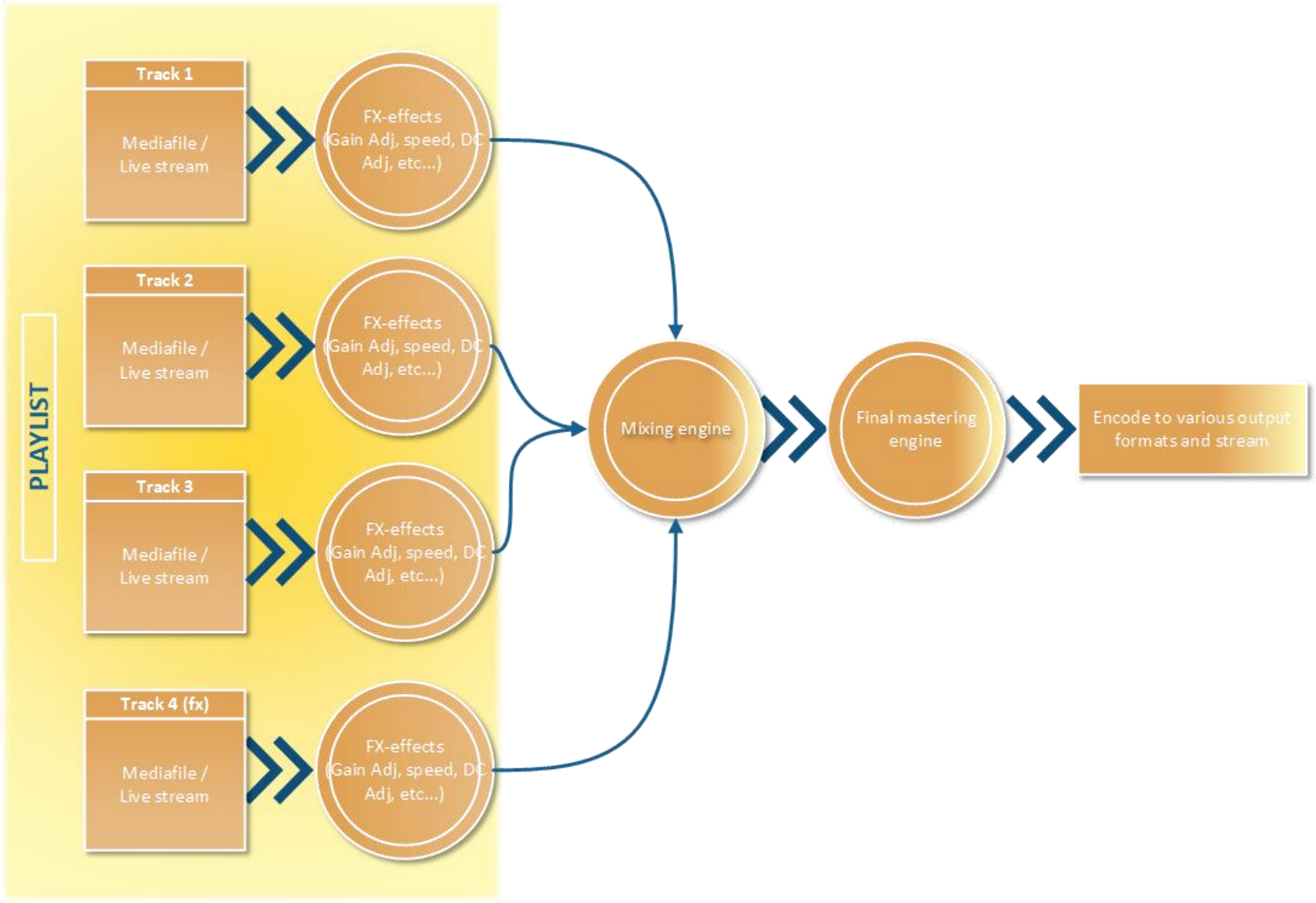
Major features:

- Fully cloud-based, scalable solution that contains all the necessary functionality for broadcasting and planning while maintaining low running costs
- Interactive, always real-time operation, secure access from anywhere, content pre-moderation
- Support for all major audio files formats - *mp3*, *vorbis*, *flac*, *opus*, *wav*, *aac* (MPEG-4 Audio). 8-32bits, 11khz-96khz
- Single-pass full 32-bit processing in real time, providing a minimum signal distortion and no delays
- Automatic: tagging, cutin, cutout (to exclude silence), fades, DC and GAIN adjust, BPM metering and adjusting
- Built-in cloud audio editor for basic editing of the audio material and the pre-listen it without need for offline audio editors
- Maximum flexibility – various administration clients (Web-based, require only internet browser, or feature rich - MS Windows native client)
- Intelligent multifunctional playlist generator based on a simple-to-use templates (just a few clicks with mouse and playlist is ready)
- Publicly accessible extensive interface (API) for cooperation with third-party solutions and control entire system programmatically
- Lightweight, flexible and concise language for describing playlists and templates allows almost unlimited opportunities to expand
- Logging of all system activities, operations. Integrated analytical reporting and statistics generator
- Granular system objects access restrictions (RBAC)
- Quoting system. (by duration, size, number of items) per-groups audio materials in media store
- Built-in a variety of effects at start and end playback of each audio materials
- A built-in final mix output DSP mastering: limiter, equalizer, dynamic compressor and maximizer ensures smooth sound
- Receive multiple "live broadcast" streams for real time processing (3 unique modes: *mix it over playlist content*, *full replace* or do *“intelligent”* injection)
- Output (master-out) signal at the same time in most popular formats (MP3, Vorbis, Opus, AAC) in WebM/OGG containers. RAW soundcard output
- Special automatic seamless dumping to disk file selected radioshows for various demands. And attaching them to system mediastore

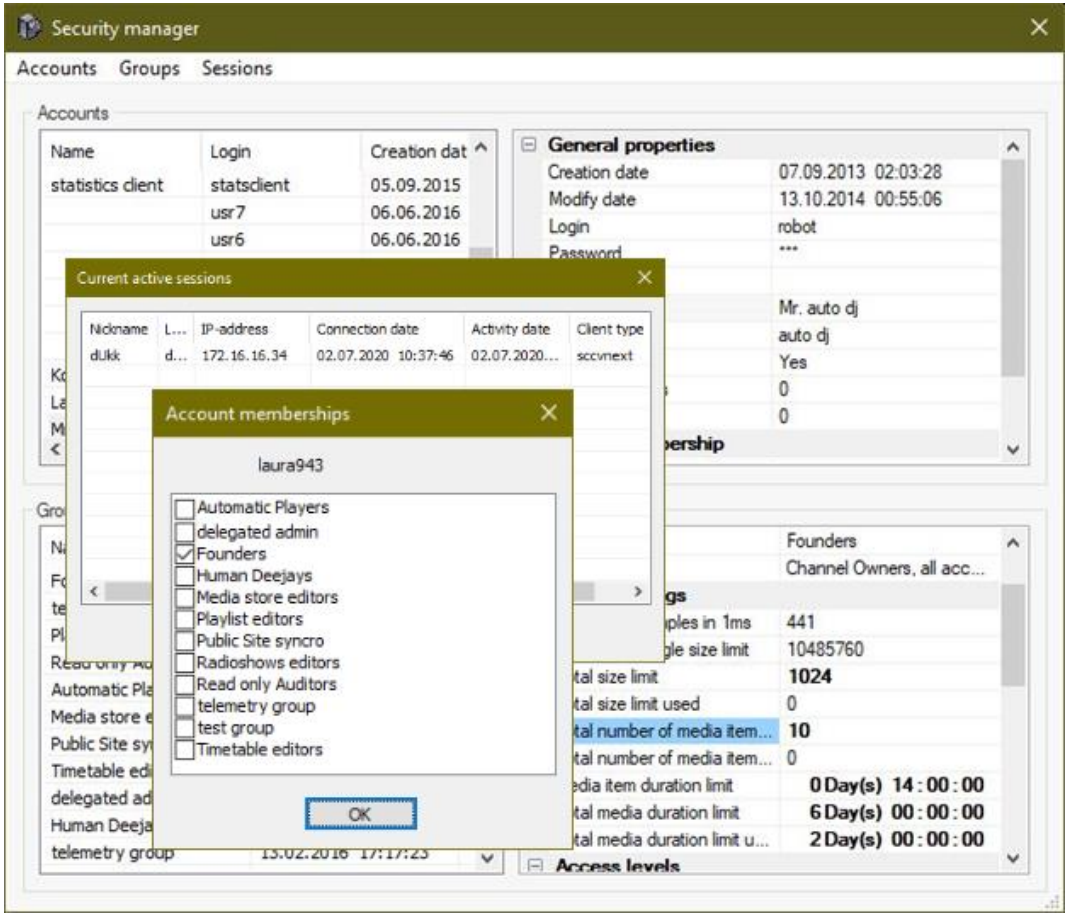
Entire system workflow diagram



General diagram of a broadcaster engine



Accounts, groups and security permissions



Permissions is flexible demarcated on per-group basis, which simplifies the process of entire system administration and running overhead. Thus, in order to assign certain rights to the user for working with the system, it should be included in a specific group, has these rights had.

Permission system has two global types:

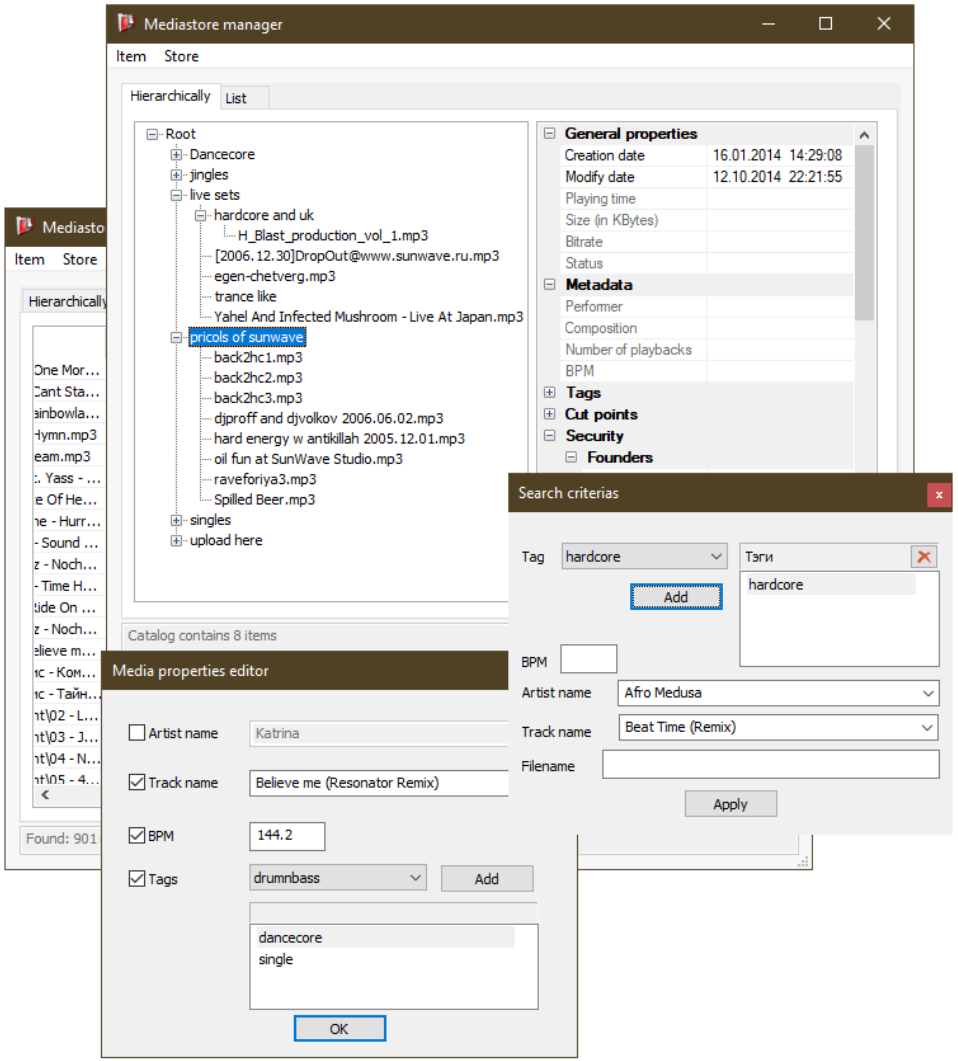
- *Class right* (top level). Maximum available access right
- *Instance right*. It is a further separation of class right to multiple small different rights

Each group has distinct rights in wide sections, each of which relates to a specific part of the system.

The differentiation of rights with great granularity allows you to build an effective workflow for broadcast management.

All of permissions-related activity strongly logged in audit journal (like the rest of operations in system)

Media store



All audio files, with which the system work live here.

Supported MP3, FLAC, VORBIS (ogg), OPUS (ogg), WAV, MP4 (AAC), with sampling frequency from *11khz* to *96khz*. Sound bit depth - *8-32* bits. When uploading new file, it will be:

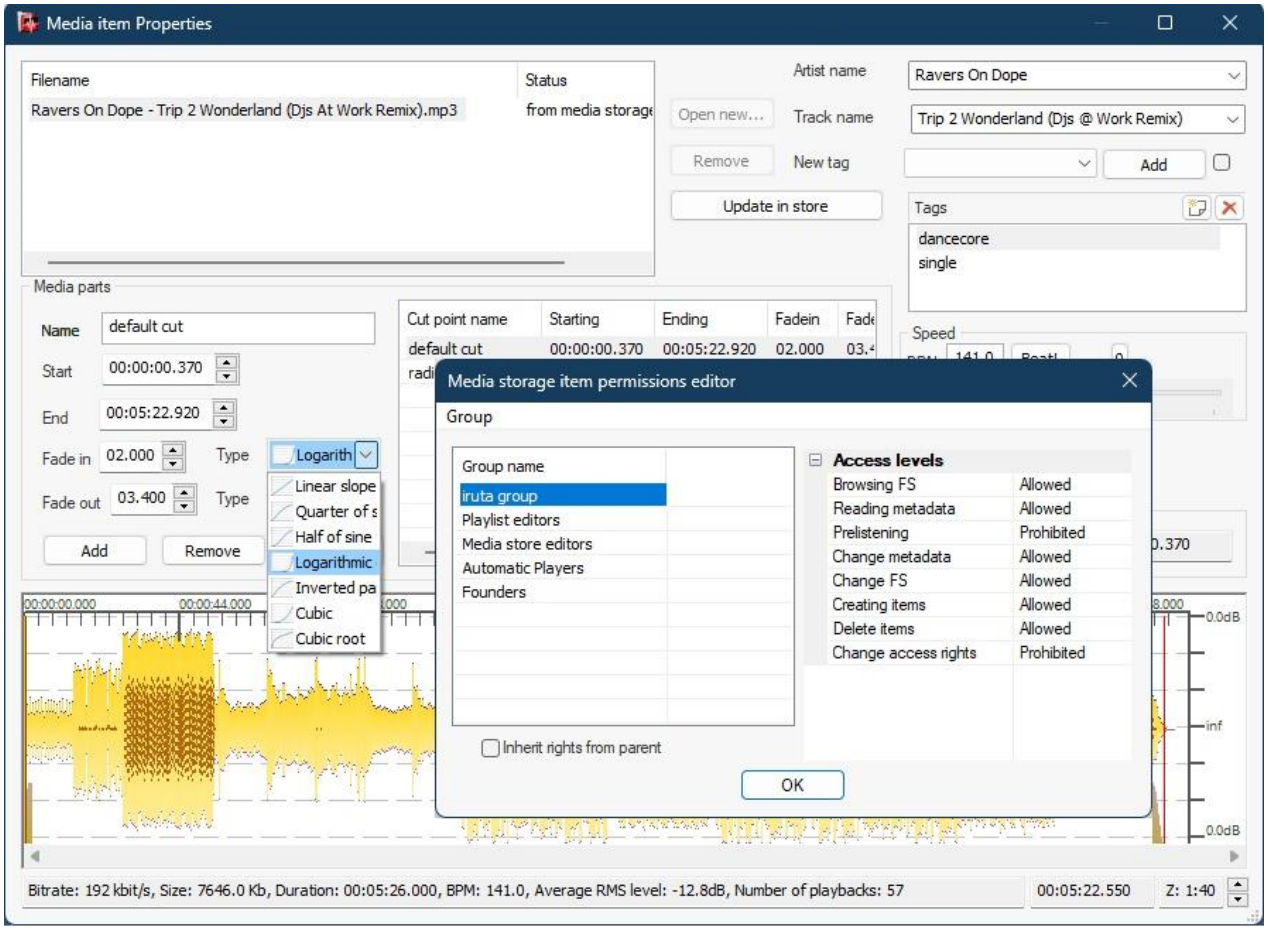
- checked for correctness,
- prevent duplicated audio,
- normalize average listening level of volume (if needed),
- perform *DC*-adjustment (if needed),
- detect *BPM* (Beats per minutes), beat grid,
- creation of cut in/out marks for first fragment

There are two ways of uploading files in the archive which affect the speed of the appearance of the file in the availability:

1. Via full native-platform client. Thick client
2. Via web-interface. Thin client

Mediastore represented in two distinct visual apperances: *as tree, as list*. Each is convenient for specific scenarios.

Important thing to note: Media store only allows you to upload new materials, it does not allow you to download the original (master) of the added materials "back" in any way.



The system provides 7 types of the volume envelope curve (rise and fall). Fade out / Fade in Fx effect on media start and stop.

Optionally all uploaded content must be pre-approved by the group of special moderator role (or legit). Moderators who access the material, give permission or prohibit use in system for some reasons.

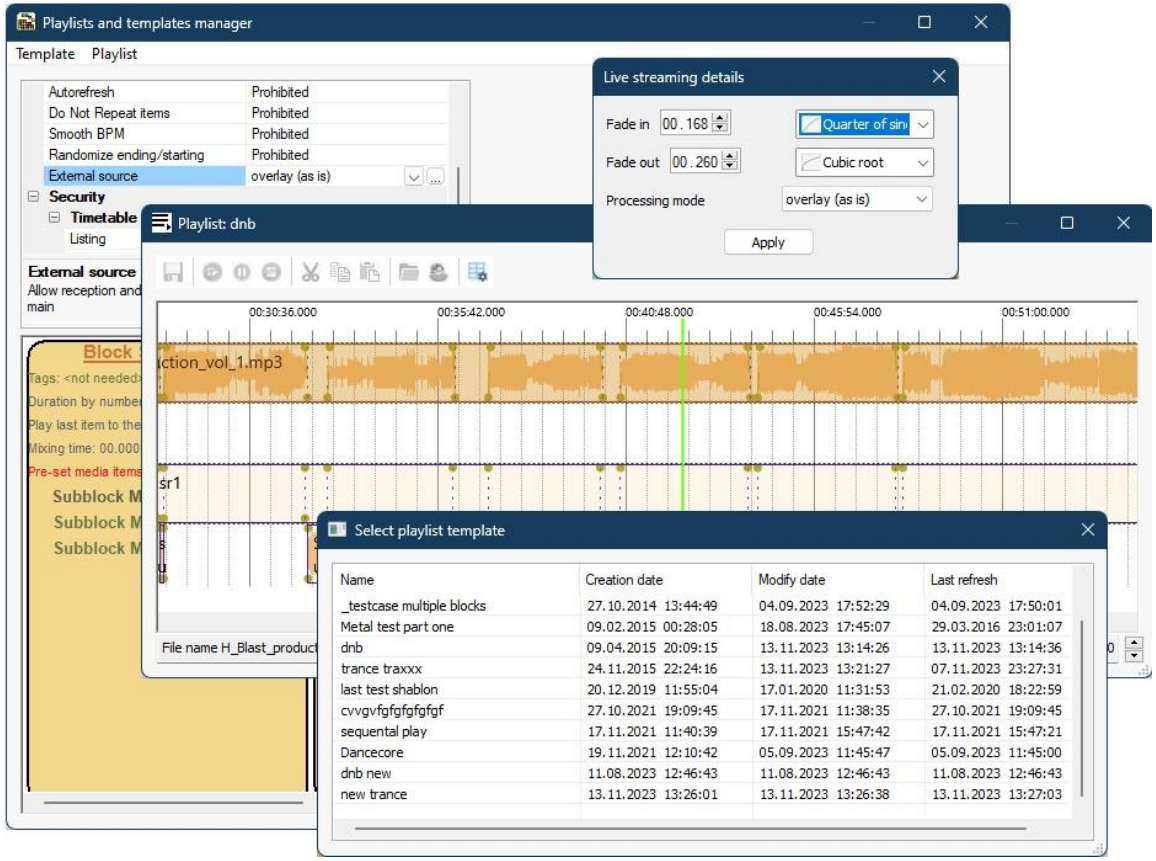
If the moderator does not permit its use, it does not deleted, and is still in the store, so that later it was possible to examine the reasons for the ban and, in case of an error, as quickly as possible to open the access track in the ether.

To perform pre-moderation function, you do not need to fully download this audio for yourself from somewhere. It is available in a simplified quality, quickly, via the web-console from any device (tablets, mobile phones, laptops, etc.)

Dividing media files into cutpoints provide a convenient feature when you can make several smaller fragments from one large and use them separately. For example, cut off the beginning, leaving only the content of the radio show itself. And for all this, you don’t need to use third-party programs or have this phonogram locally and upload it again after cutting. Everything happens instantly within the system!

Playlists and Templates

Playlist - a strict sequence of media elements (one-after-another), placed in chronological order with the set playback effects. To add the playlist "on the air" it is placed in the broadcast schedule. To facilitate the compilation of playlists a technology called templates are invented.



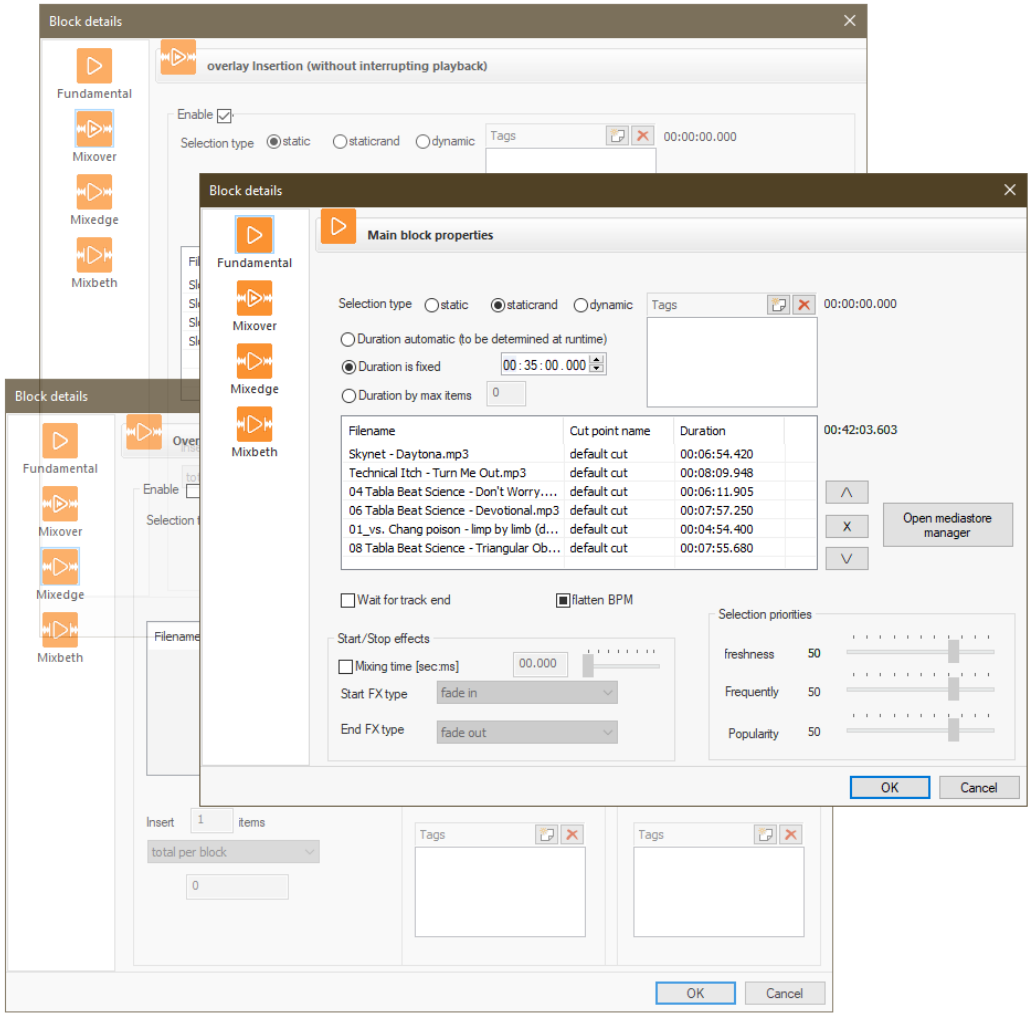
The system allows the user to audit what is happening on the air in real time visualizing the playback position in the playlist.

Each playlist using 4 (virtually unlimited) separate turntables. Three of them - universal, so can be used to accommodate any type of audio. The fourth channel is a special one. Dedicated to placing short accompanying media - such as jingles, spots, advertising, callsigns and etc.

Template defines a sets of characteristics that identified by the user. In accordance with these characteristics the system automatically creates a playlist from the available media archive material. Each template consists of blocks.

Each template has a large number of properties that the user can adjust at its discretion.

System can display used realtime playlist with playposition marker in it.



Property "Type of sampling" defines how the tracks will be placed inside the block:

1. **STATIC**. Specifies the list of audio files and the sequence in which they are played.
2. **STATICRAND**. Specifies a list of audio files to be played in random order.
3. **DYNAMIC**. It specifies only tags, which the system chooses to play audio files with user-defined preferences (Popularity,Frequency, Freshness).

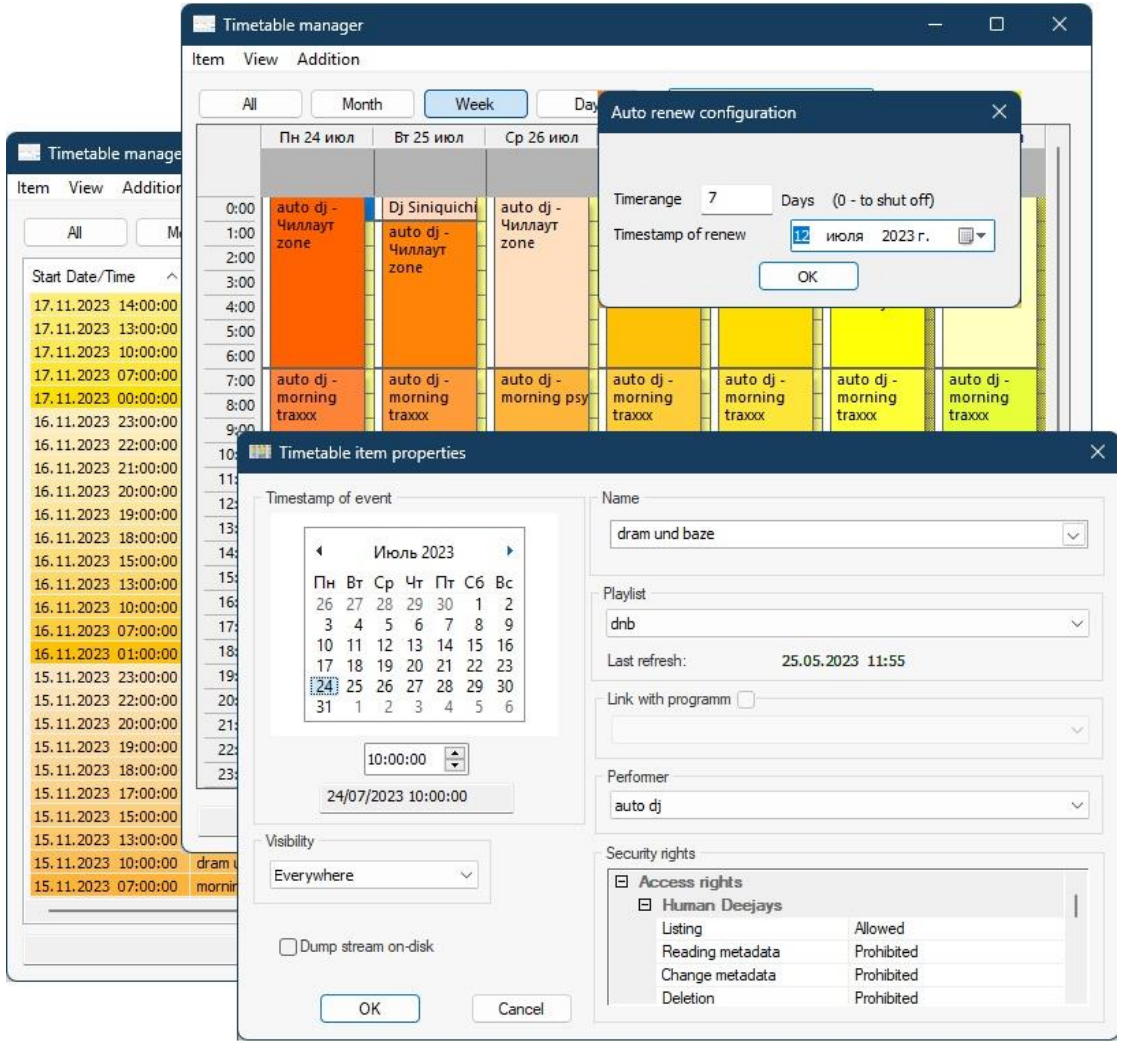
Within each block, there are special types called sub-blocks, special for - sign of the station, jingles, commercials, advertisements, promos, etc. These sub-blocks are of three types. Optionally, within one block you can use all three types of:

1. **Mixover**. audio files will be injected over-the-top of main block files. Placing the elements can be placed after each of them a certain number of tracks of the main assembly, or after a certain amount of time.
2. **Mixedge**. audio files are inserted into the gaps of the main block files. To place an insert FX-effects will be used. The rules are similar to the previous arrangement.
3. **Mixbeth**. audio files will be placed between the end and beginning of each N-th track. The number N may be set in the range from 1 to infinity.

For certain sub-blocks types you can use *Pre-roll* AND/OR *Post-roll* specials to inform listeners about coming in.

The variety of settings, options and playlist generator configuration templates enable compile various playlists even with a minimum of materials.

Broadcasting schedule



The system makes it possible to program broadcasting schedule in advance for any period (virtually unlimited) of time.
Each position in the grid painted in different colors depending on the time of day and day of the week. Days of the week are different primary color, and the evening hours are different from the morning a light by a tone.
Schedule view is grouped in four temporal criteria:

1. *All broadcasting time*
2. *Month*
3. *Week*
4. *Day*

The system provides the possibility of automatic extension of the existing air schedule at an arbitrary time to come. The system will automatically clones sequence of scheduled programs in a specified time range until you disable feature.

Schedule allow you to create different types of events (visible only in public site but ignored completely by system and hidden (from public site) entries.

Global system properties

Here is a various settings for entire broadcasting system.

The system supports concurrent output master signal in the 5 most popular formats:

- MPEG4 Audio (AAC ATDS),
- MPEG1-Layer III (MP3),
- Vorbis in OGG, WebM containers,
- Opus in OGG, WebM containers.
- RAW PCM to hardware ALSA-compatible soundcard (zero latency).

System have built-in limiter with dynamic maximizer. This ensures smooth sound output right under your control.

Supported output transmitter servers: *Icecast*, *Shoutcast* or sound-card.

All parameters applied in real-time without need for a software restart on server.

The separate role of the “sound engineer” in the term ‘access rights’ allows you to distinguish the final and pre-final DSP processing settings from the general system configuration.

Here you can set *preferred* codecs and containers for livecast sources.

Outputting the recording to a file allows you to implement scenarios with both “police broadcast recordings” and create copies for automatic replays of selected broadcasts. After configuring feature here set dump flag on needed scheduled shows and all is done.

So it is possible to seamless and fast integration into your already existing infrastructure.

Statistics and analytics

Allows almost in real time, evaluate and analyze the popularity of the use of resources of radio listeners as well as being to create reports based on historical data in the system.

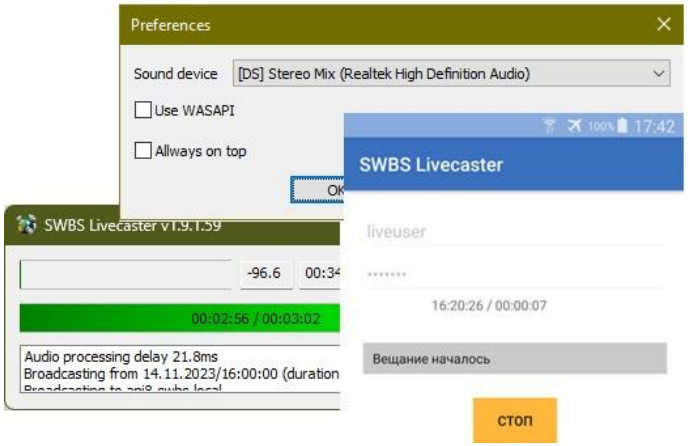
The system provides a dozen ready-made (pre-defined) reports. New reports are added as necessary, and if required – all data is available for the downloading via "raw".

Besides doing statistics collection on listeners, tracks, live events, system do intelligent telemetry collection while receiving livestreaming from network connected sources.

You can view all this data in realtime to help you identify quickly bottlenecks.

Live stream broadcasters

Livecaster - is an application that transmits an audio signal in real time to a broadcasting system for its further processing and broadcasting. The format of the transmitted signal can be either compressed with MP3/MP4/OPUS/FLAC/VORBIS codecs (embedded in WEBM,MP4,OGG) containers, or a PCM signal without compression at all.



In addition to “official” livecaster, the system mimics standard **LibShout** + **Icecast V2** software. This means that the live signal can be received from any program compatible with this software.

“Official” livecaster have an unique ability that shows in real time intervals when it is best to speak over and the moments when such will not be audible to listeners at all (overshadowed by jingles/call signs of the radio station, etc.) Visual indication - a red or green stripe with time counter.

Additionally broadcasting system has automatic mechanism to ensure signal continuity (at cost of increasing latency) when network conditions is not perfect (3G mobile networks for example). It will do dynamic buffering in this case. Of course you can manually tune its parameters.

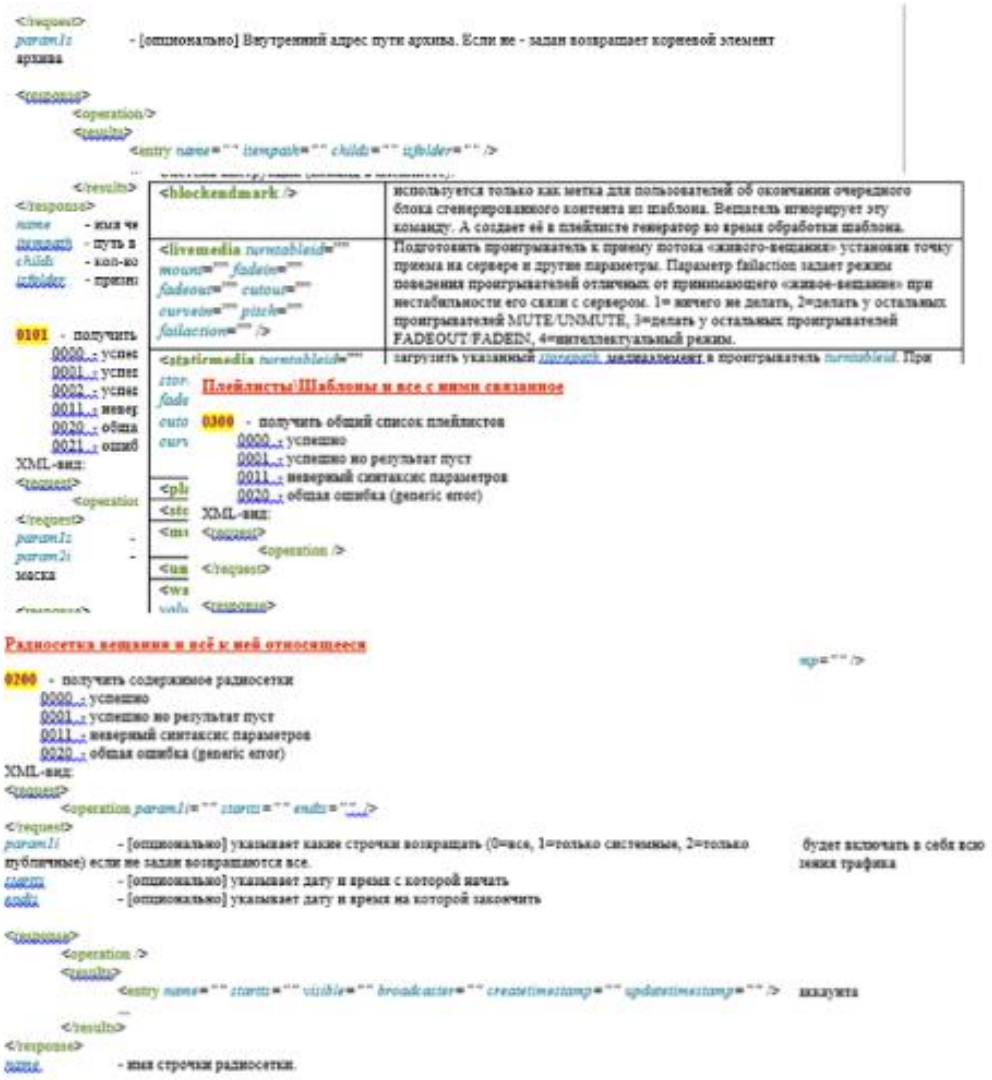
Available ready to use various platforms clients:

- Windows (version >= 7)
- Android (version >= 5.0)
- Linux (ALSA , Kernel >= 2.6)
- WebUI (HTML5 for modern browsers supporting latest technologies)

Live-streaming seamlessly works in 3 dedicated modes:

- **Overlay** live signal with already played in playlist. For example playing long set without voice and Deejay will overlay it with his real-time voice over from mobile
- **Replace** all playlist content with live signal except events with special Jingle/Advertisement types. This will allow non skippable by Deejay audio material
- **Completely replace** playlist content with live signal. This is a dedicated live streaming with failover (playlist) when network issues encounter

API for integration with various 3rd party systems and solutions



SWAPI – software interface (Web Service) to the system via simple request/response over HTTP-protocol.

Supports 2 popular internet information exchange formats: *JSON* or *XML*. All formats use of 2 encodings: *UTF-8* or *UTF-16*.

Technically, it consists of two parts: the authorization service to obtain a security *token*. And a various micro-services themselves where it is used once received *token*.

All external entities of system (thin and thick clients, Telemetry, Livecaster) using this open interface to interact with the system.

The simplicity, detailed description, and convenience of the interface allow you to quickly integrate it with your existing systems.