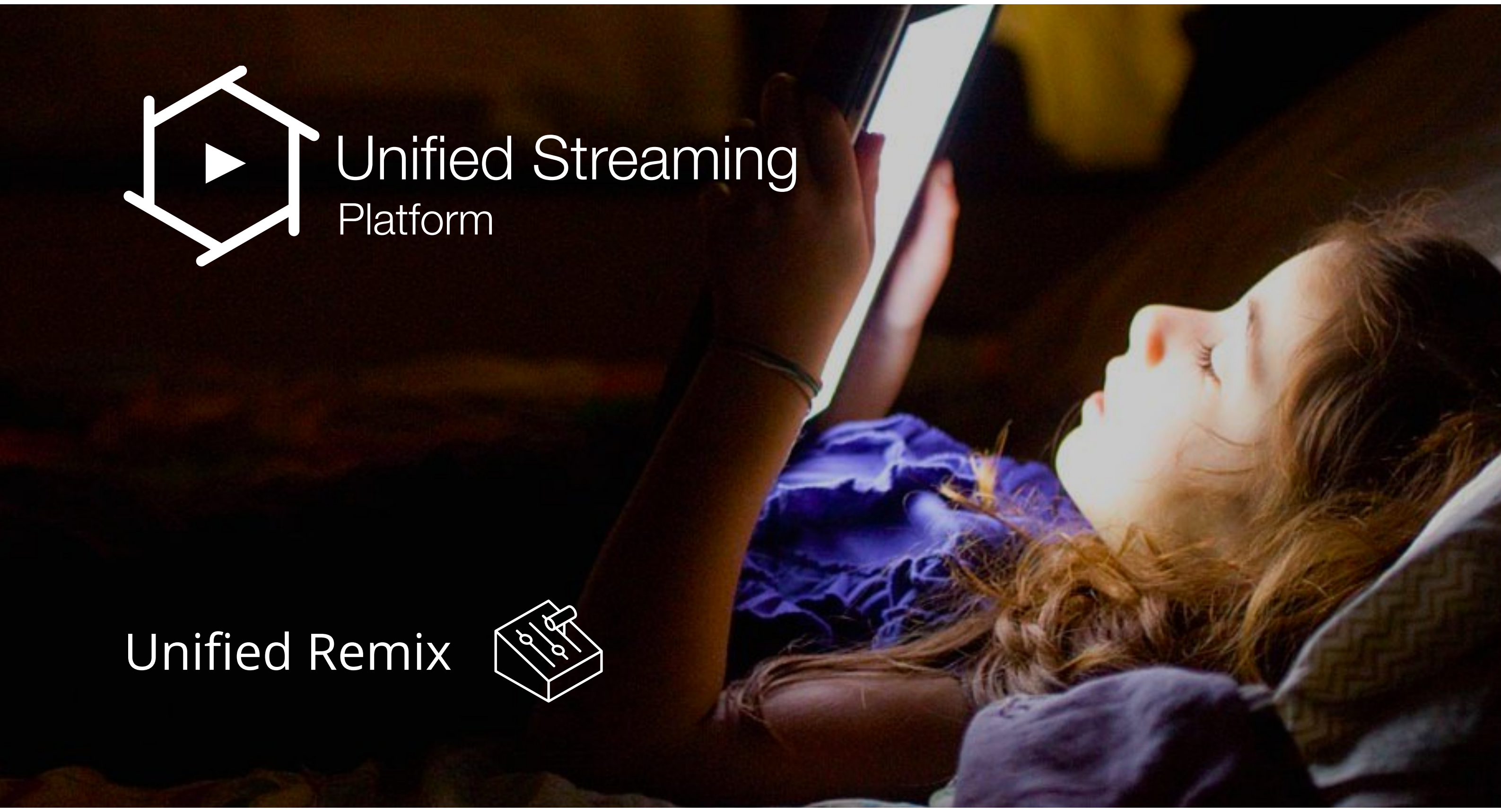


Unified Streaming
Platform

Unified Remix

A new and innovative approach for server side remixing of content

The basic idea of Unified Remix is that it 'mixes' clips from various origins into a single stream. This means that players do not see any discontinuity, the stream has a single origin and a single timeline.

Viewer info (e.g. session id or cookie) can be used to personalize streams on any level. From a personalized stream for every viewer to streams to groups of viewers (e.g. based on geolocation, subscription model) or even a single stream for all viewers (e.g. a bumper).

Creating a remixed stream opens new doors for content owners to monetize content but also offers new options for creating personalized streams of for instance live archives.

Best of all, Unified Remix will create a stream that plays on all platforms in all formats and devices and as such it provides a great multiplatform experience.

Use Cases:

- Bumper for Everyone (VoD)
- Pre-, Mid-, Post-Roll (VoD)
- Archive-to-VoD (Infinite Live archiving)
- Live-to-VoD (Separating Live ingest from playout)
- Live Scheduling (Rotating 24/7 Playlists)
- Bumper for Everyone (Live)
- Dynamic Ad Insertion (VoD)
- Dynamic Ad Replacement (Live)

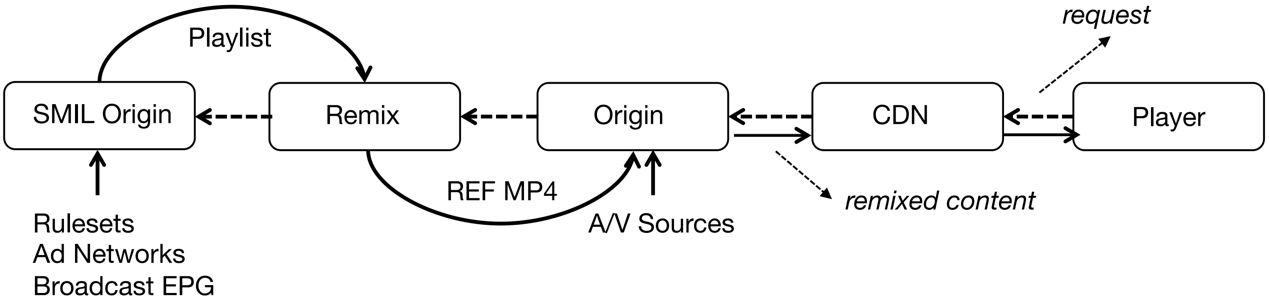
Unified Remix
Make it personal

The workflow to implement these use cases contains the following steps.

First, the SMIL origin creates a SMIL playlist and for instance could call into an Ad network, use a predefined rule (in the CMS) or even co-operate with the broadcast automation to get EPG info to understand at which time the clip should be placed.

Secondly, Unified Remix creates an MP4 file that references the media listed in the SMIL playlist. Unified Remix is a webserver module that handles HTTP request on-the-fly, dynamically. Requests can be forwarded and cached as it is all HTTP.

Lastly, the Unified Origin ingests the reference MP4 and creates a single stream, so players do not see any discontinuity: to a player the stream has a single origin and a single timeline. Schematically this looks like the following; the player does a GET request for the (personalized) URL received from the CMS, this translates into GET requests upstream until the content is produced:



For Unified Remix demo's and more details on the several use cases, please visit demo.unified-streaming.com/remix.html