

The Kazuhm Transcoding Application is a multi-codec/format video transcoding solution that lowers the cost of transcoding for those already transcoding in the cloud or offers a cost-lowering alternative to the cloud for brownfield sites with considerable legacy infrastructure issues now wrestling with a multitude of formats, consumption devices, and delivery scenarios.

It allows Telecommunications, Media, and Entertainment content providers to deliver video to consumers demanding immediate playback start, uninterrupted smooth viewing with no buffering or frame drop, at the highest quality possible on their device, whatever it may be.

## Features

**Standalone software-only solution** with no proprietary or complementary expensive hardware systems that are often difficult to easily modify or significantly upgrade once built and deployed.

**100% OPEX**

**Comprehensive format support** with the default built-in FFmpeg video and audio processing engine.

**Multi Bit Rate (MBR) support;** 1 input to n outputs in a single file.

**Utilize existing IT infrastructure** as a heterogeneous distributed network comprised of any combination of desktop, server, or cloud resources.

**Flexibility to add and change** available resources as required.

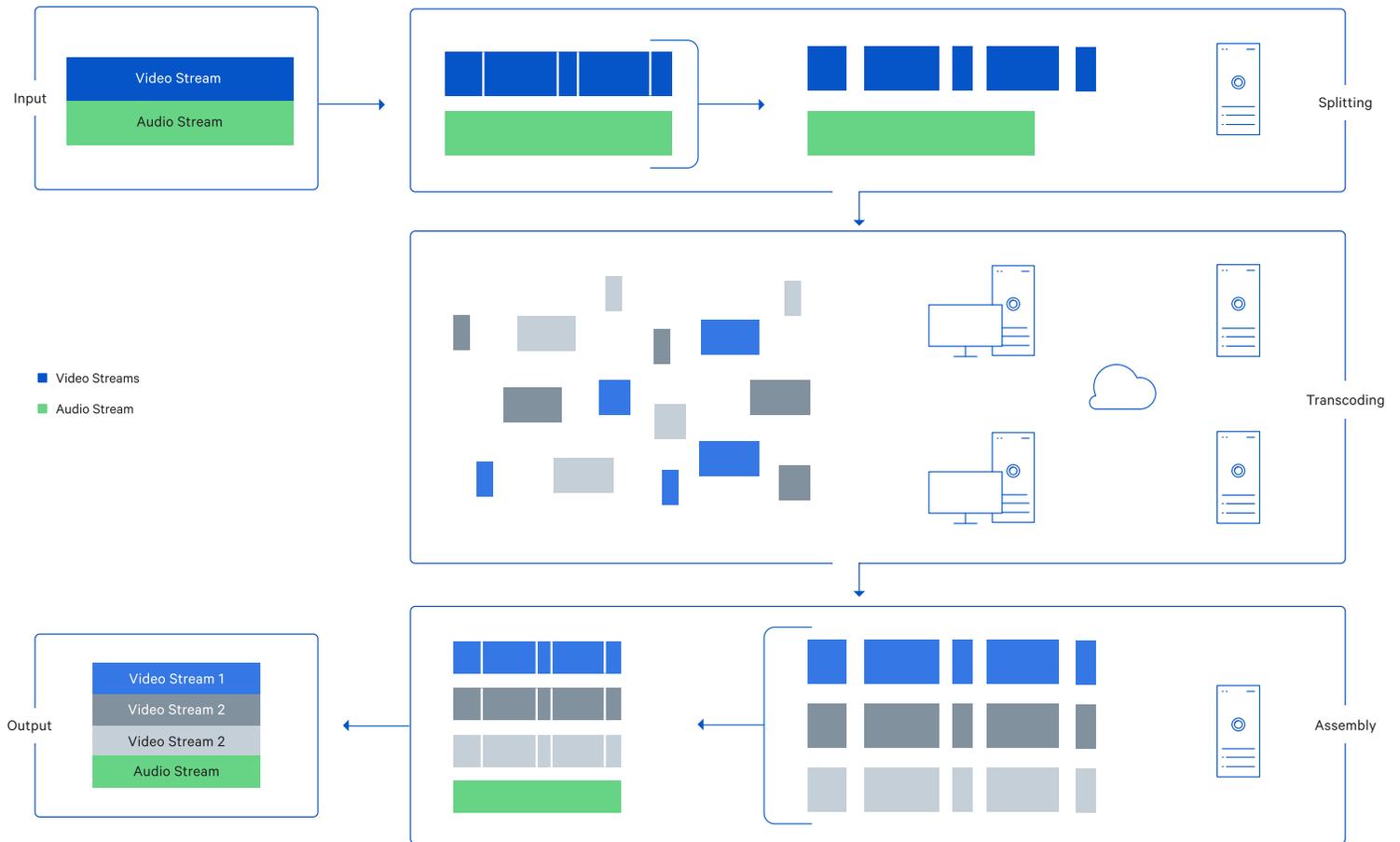
**Instant on-demand scale-up** through cloud-burst to address utilization peaks, while paying for that extra compute resource only when it is needed.

**Easy deployment** without the need for specialist support.

**Intuitive user interface** to allow quick video and audio parameter configuration together with ingest and output locations.

**Visibility** across the entire transcoding process.





## Process Overview

1. **Input:** Once uploaded and transcoding begins the input file(s) are first separated into audio and video streams. Note - if the file contains multiple video streams then the stream with the greatest resolution is used.
2. **Splitting:** The video stream is then split into chunks using key-frame boundaries; each keyframe (or i-frame) is a full frame of the image in a video while subsequent frames only contain the information that has changed.
3. **Transcoding:** Chunks are variable size sets of frames with associated metadata. If an input file is split into 5 chunks, with a selected output of 3 streams, then 15 chunks total are individually assigned to the allocated resources for processing.
4. **Assembly:** As chunks are processed they are then reassembled, with individual output video streams being reassembled independently.
5. **Output:** The audio stream, processed separately, and all video streams are finally recombined into a single output file.

Coming Soon

Automated batch processing.

Adaptive Bit Rate (ABR) support.