

# Live and Post Production for Sports Broadcasting

He Zhi  
MIGU, ChinaMobile



# Agenda

- Overview
- Live Production and Streaming
- Post Production and Editing
- AI Enhanced Production
- Requirements and Experience
- Expectation for Web Production

# Overview

EURO  
2020

TOKYO  
2020

Beijing  
2022

Qatar  
2022

## Sub-licensed rights to air live and on-demand coverage of UEFA Euro2020, Tokyo2020, Beijing 2022 and the Qatar 2022 FIFA World Cup

In the past sports broadcasting events, web-based technologies have been largely deployed for live and post production

### Live broadcast and Streaming

Receive the live stream and split a short video around the start and end time of the video hotspot

- ★ Rtmp\hls\http-flv
- ★ H264 \H265\vp9
- ★ Html5\hlsjs\videojs
- ★ CDN\m3u8\section

### Post Production and Editing

Obtain multiple media contents, split and combine them, and add titles, endings, special effects, graphics, subtitles, music, etc

- ★ Web Assembly(editing before upload)
- ★ Canvas\fabricjs\WebGL
- ★ Ffmpeg\Opencv\Opengl
- ★ Cloud services\GPU

### AI Enhanced Production

AI analysis is used to identify characters, scenes, actions, etc, provide information, support splitting and content processing

- ★ Action->3D CNN
- ★ Face->MTCNN
- ★ Ocr->CTPN
- ★ Cloud services\GPU

# Live Production and Streaming

## Live Broadcast Editing Tool:

- Receive the live stream and split around the start and end time of the hotspot

## Tech:

- Html5\hlsjs\videojs

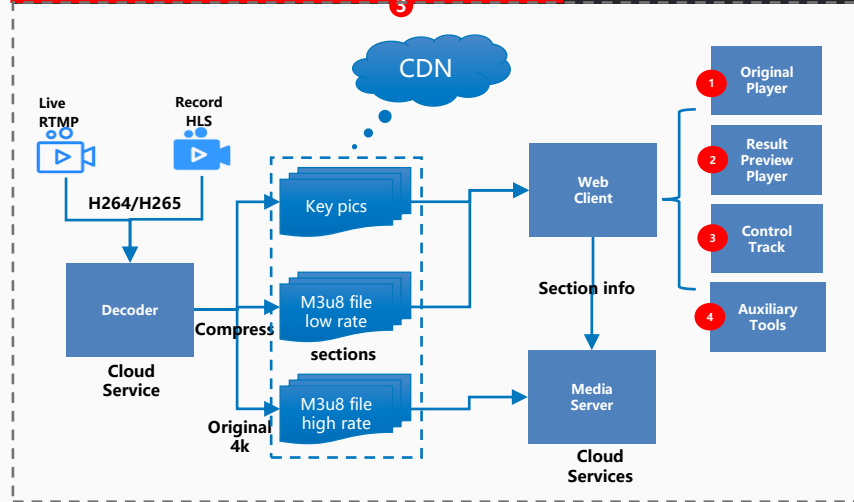
## Feature:

- CutToPublish、CutToGif、Screenshot、AICut

## Target: publish within 30s

- Decode as HLS with 2s slicing
- Provide 720P compressed data
- Preview key frame pics in track
- Preview editing result
- Share load by media server
- CDN cache

Expect: videojs could share stream with another one



# Post Production and Editing

Post Editing Tool: professionalism

- Support kinds of media, split and combine them, and edit with titles, graphics, subtitles, music, special effects, etc

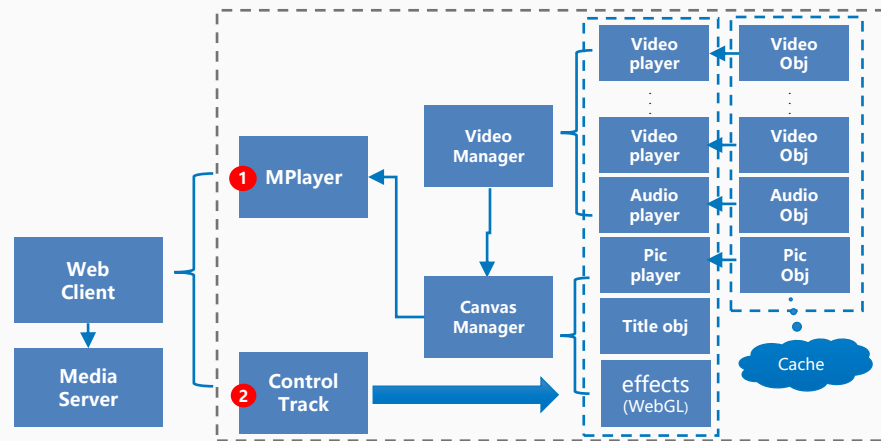
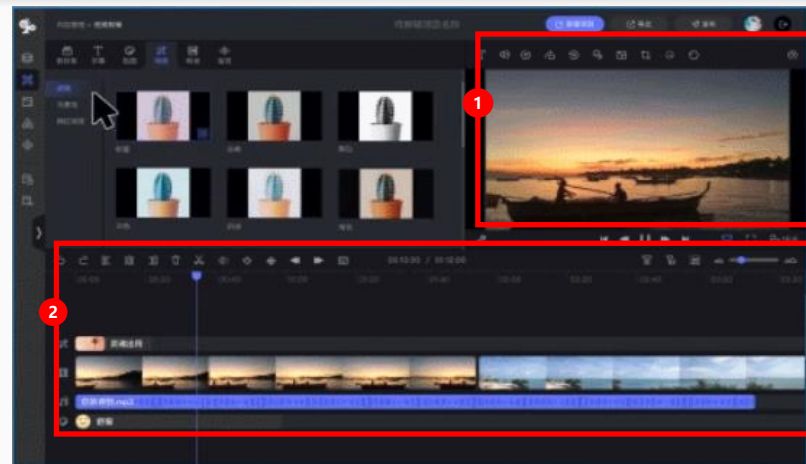
Tech:

- Canvas\Fabric.js\WebGL
- HTML5\videojs\WebAssembly

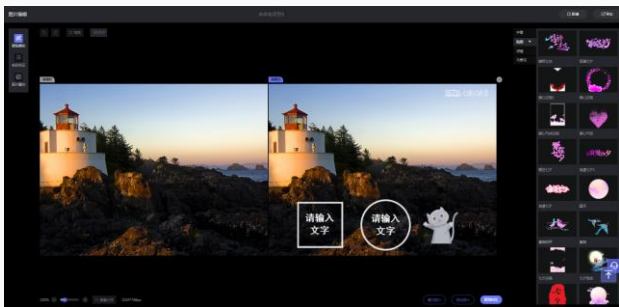
Target: high performance and efficiency

- Simulate result by canvas and implement by mediaserver
- Simulate the effects by WebGL with 480\*270(cost laptop 40% cpu)
- Recode as HLS for fast preview
- Preprocess video while uploading by Webassembly

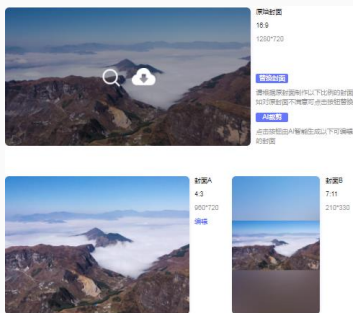
Expect: WebCodecs could provide more functions as ffmpeg, and improve the performance of WebGL



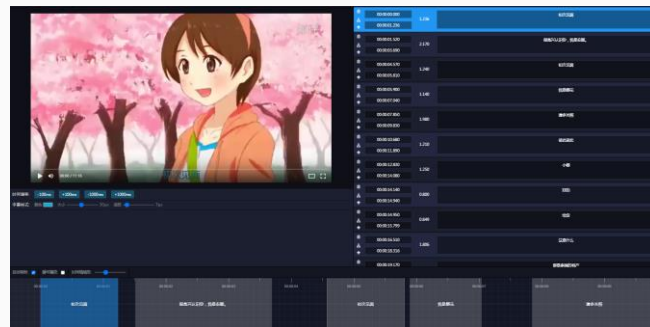
# Additional Features with Web Production



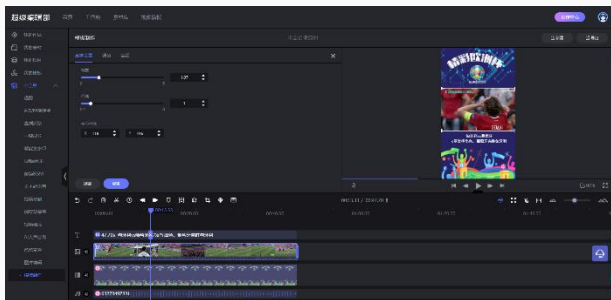
Editing Picture



Making Cover



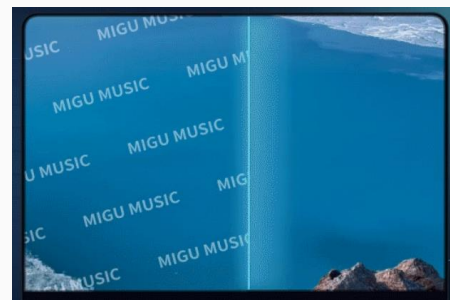
Making subtitles



Creating&Using Template



Horizontal to vertical

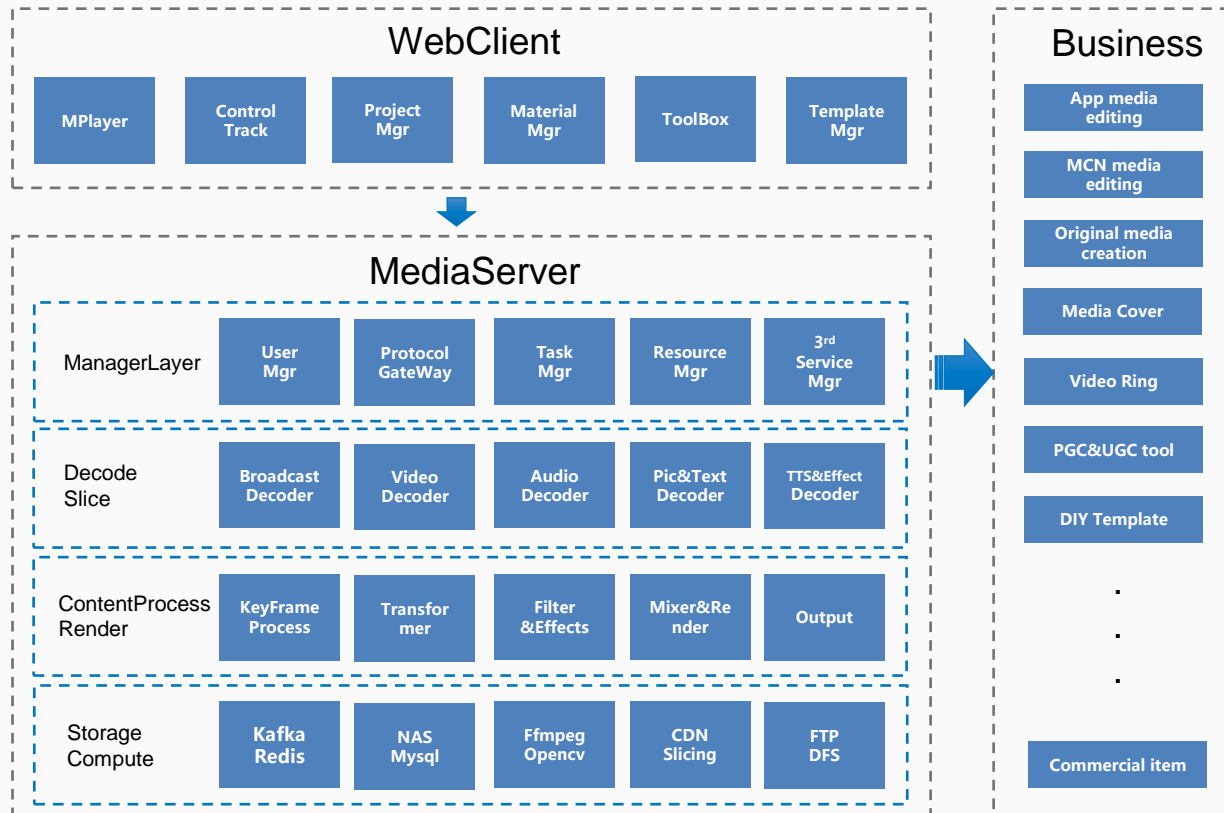


Remove watermark

# Leveraging frontend and backend

## Key Reason:

- Performance. Most high CPU/GPU consumption jobs depends on server
- Flexibility. Light client could be easily integrated in business application
- Compatibility. MediaServer could provide API for kinds of client, as cloud services



# AI Enhanced Production



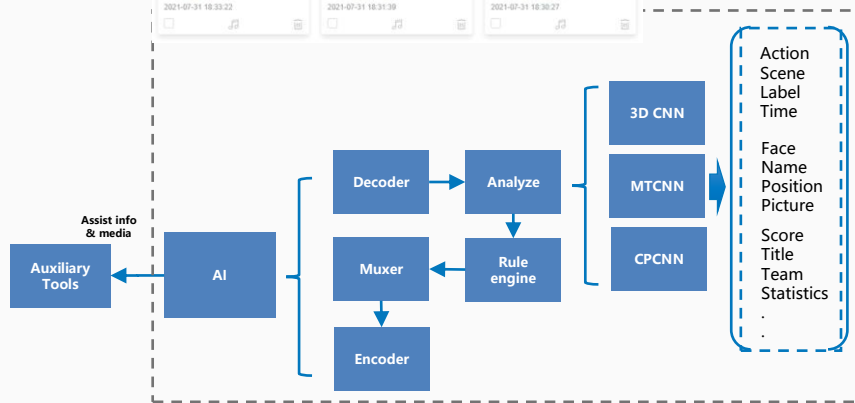
Scene: football match  
Score: 1:1  
Face: Marco Asensio  
Time: 32:20  
Label: football  
Team: ESP:CIV

AutoCut:



Target: More videos and lower cost

- Auto fill the label
- Auto add subtitles
- Auto remove mark
- Auto horizontal to vertical
- Auto cut video by action identification

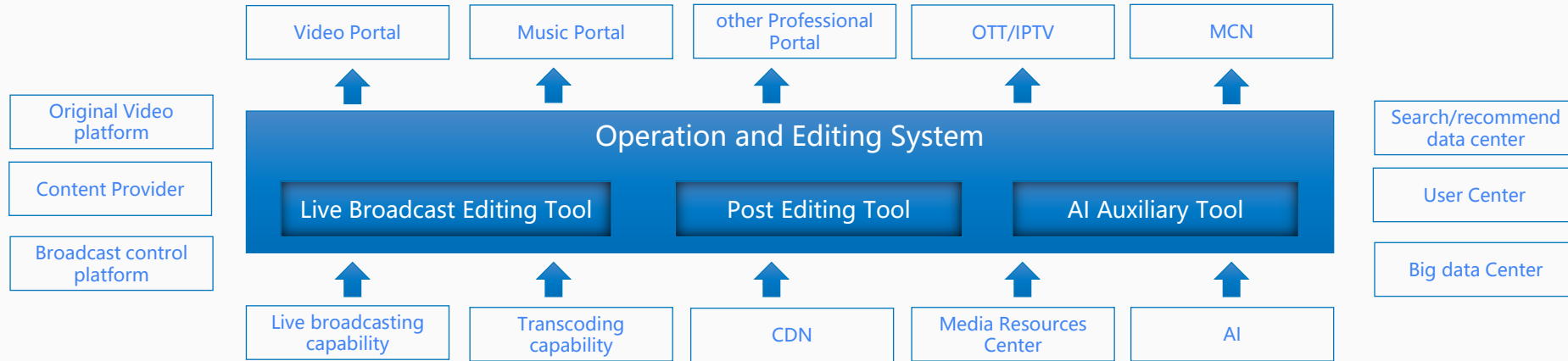




# Requirements and Experience

Why to choose web editing tech, take a media company with 200 employees as example

- Copyright requirement. Minimize the risk of content copy, download and disclosure.
- Operation requirement. Users could save a lot of time by online modification and resubmission the content.
- Product requirement. For better performance, flexibility and compatibility.
- Cost requirement. One graphic workstation need 5w, to support 200 operator it need 1000w. Replaced with shared cloud server, it could save 80% cost.
- Time requirement. Application normally need 1 year to build , and it took 3 months with web tech, it saved 70% time.



# Expectation for Web Production

Expect:

- WebCodecs could provide more high level API as ffmpeg to search, locate, cut, composite media;
- Through webcodecs, videojs could share stream with another one
- Improve the performance of WebGL to support high resolution rendering
- Support more format. e.g support rmvb\avi\mkv for network video, support h265\vp9 for 4k、 8k stream



Thanks