

GEEKS
SOLUTIONS

Managed IT Services You Can Trust

Case Study:

JavaScript based Video Processor

Background:

Clients company deals with the Virtual reality and Augmented Reality application and its related products.

Requirements:

The primary requirement for this project was –

1. The web-application should be able to process the 3dimensional video and image assets on the browser with the help of the latest technologies available in the market.
2. The web-application should be able to take multiple format images as an input and should process VR supported 3D video as an output.
3. The web-application should be able to perform the moderate level of video processing, such as – cut, split, duplicate frames, merge audio and video together as well as image processing.

Technologies used:

The selection of appropriate development technologies was mandatory, as the application must process the heavy tasks on both ends (backend and frontend), so the technologies must be reliable.

So, our developers have come up with the solution which balances both – Latest reliable frameworks and to manage heavy workloads.

The application platform was basically a MERN stack except for ‘M (MongoDB)’ as we are not using any database to store any metadata. So, it was ERN stack application – combination of - Express.js, ReactJS and NodeJS.

To process the videos in the backend we have used FFMPEG alongside with the MELT Framework, as they are the one who are responding to the NodeJS orders.

Frontend –

1. ReactJS

Backend –

1. NodeJS
2. Video Processing frameworks –
 - a. FFMPEG
 - b. MELT

Workflow:

Backend –

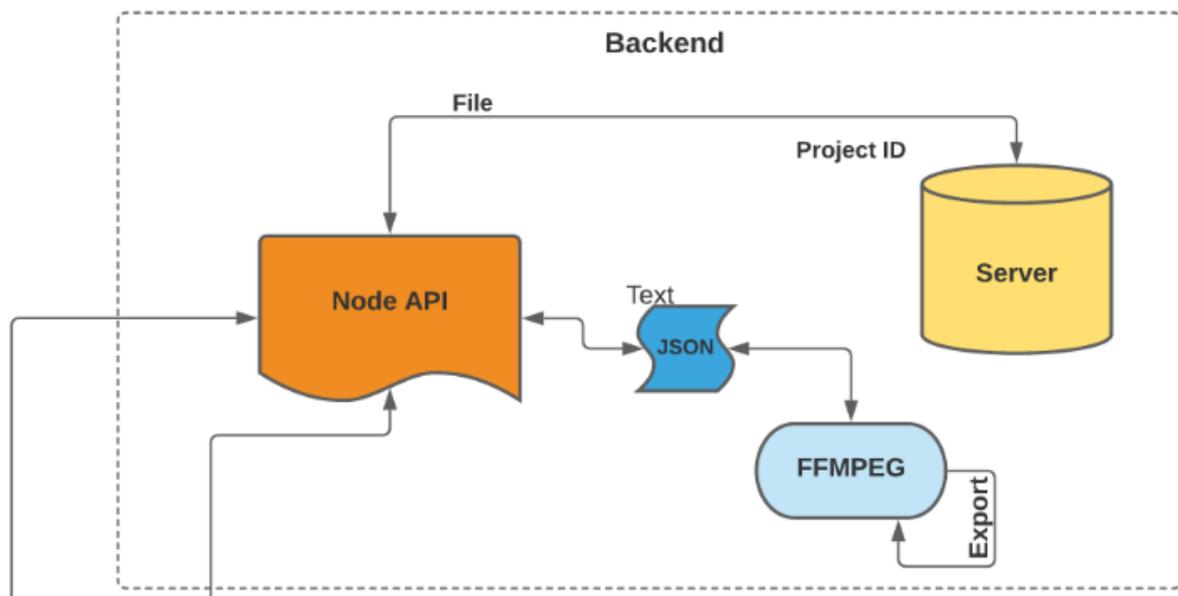


Diagram explanation –

Backend uses NodeJS to handle all the API requests incoming from the React (Front end).

NodeJS also maintains a file (in JSON format) where it stores all the tasks performed in the frontend and once the user is ready to export the data then NodeJS orders FFMPEG/Melt to perform the processing action.

Frontend –

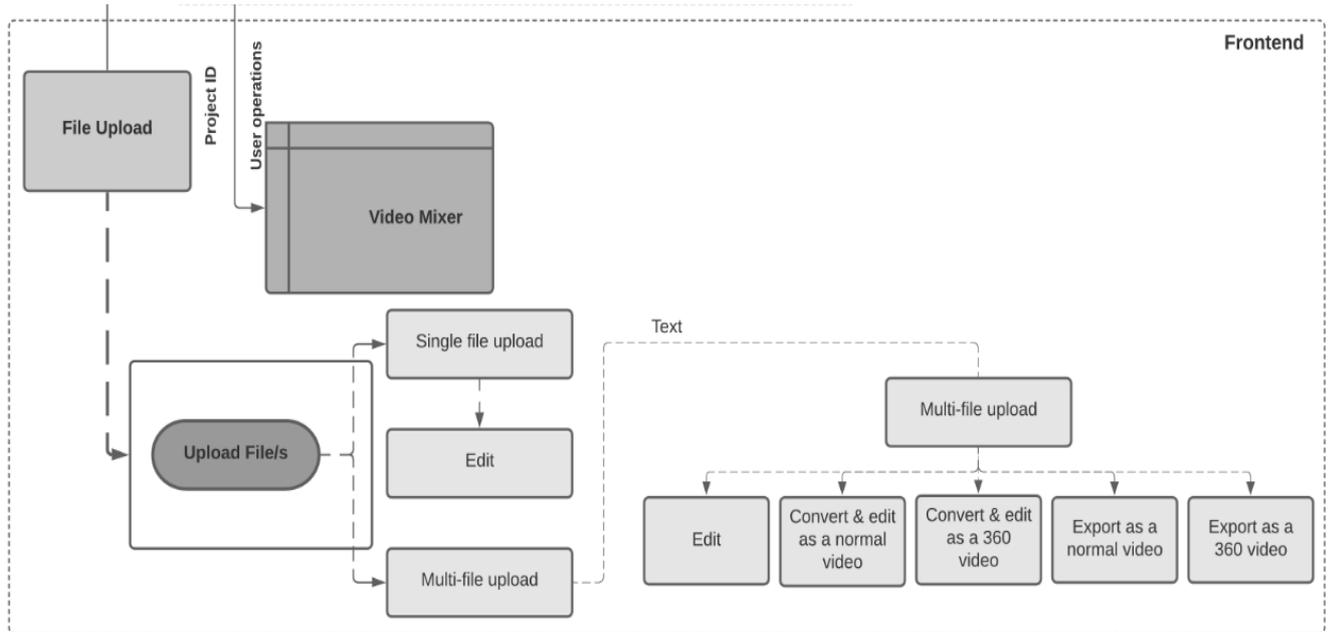


Diagram explanation –

We have implemented the frontend with the help of ReactJS, where ReactJS handles all the user inputs with the help of in/out API calls to NodeJS.

React also reads the user inputs from the JSON file which was earlier maintained by the NodeJS.

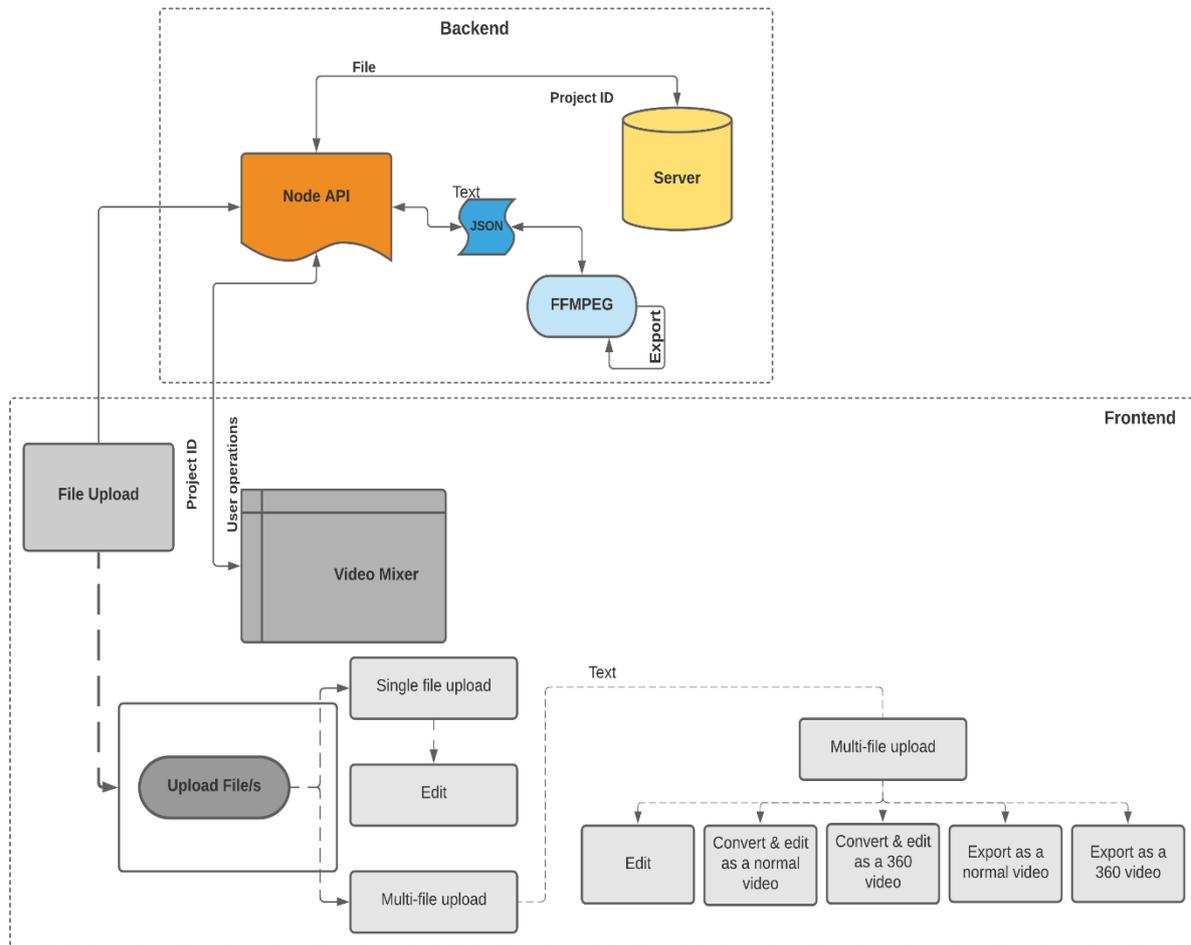
Our Approach:

Client's requirement was thoroughly evaluated and various components in the requirement were explored in depth to get a best possible architecture for optimum benefit. However major setback was server load while processing the heavy user inputs –

So, we had to optimize the code by considering the server configuration and process management. So, we have developed model that was prepared to be feasible with all possibilities explored.

There was an option to use the client browser/machine to process all the required tasks but there are chances that browser might get crash, so in the end backend is the one who must process all the inputs with the expected output as a result.

Frontend & Backend in action –



Benefits:

1. The web application can process any format of videos.
2. The web application can process any dimensional of videos, whether its 2D or 3D.
3. The web application can render all types of video resolutions.
4. The web application can generate .mp4 extension file as an output which works on all the devices that supports mp4 format.

For more information and queries, feel free to contact us. We are available at info@geekssolutions.in

Thank you!