First of all,

It is possible to download the video and check all its features without any problem. The profile is fine and there is already jurisprudence around this type of cases #565, #691, #762, etc.

Secondly,

The standard media type is **video/mp4**.

The standard mp4 container format is commonly used for both AAC audio, and H.264 video + AAC audio. These have different media types, audio/mp4 and video/mp4, however often you want different applications for audio and video and on some systems it is only possible to associate a default application with a file extension. Therefore it has become popular in some circles to use the extensions .m4a and .m4v for audio and video(+audio), respectively, **in an mp4 container format**. However **this does not affect the media type**, which already distinguishes these using the audio or video prefix.

Apple, who is the developer of the m4v, started using their own media type, **which are in an mp4 container** and use a .m4v extension.

And,

According to RFC 4337 § 2, **video/mp4** is indeed the **correct Content-Type for MPEG-4 video**. And since the video in question is an **MPEG-4 video**, then although its extension is m4v, **its format is video/mp4**. And this complies with the guidelines.

Sources:
https://stackoverflow.com/questions/15277147/m4v-mimetype-video-mp4-or-video-m4v

https://stackoverflow.com/questions/9929940/correct-mime-type-for-mp4

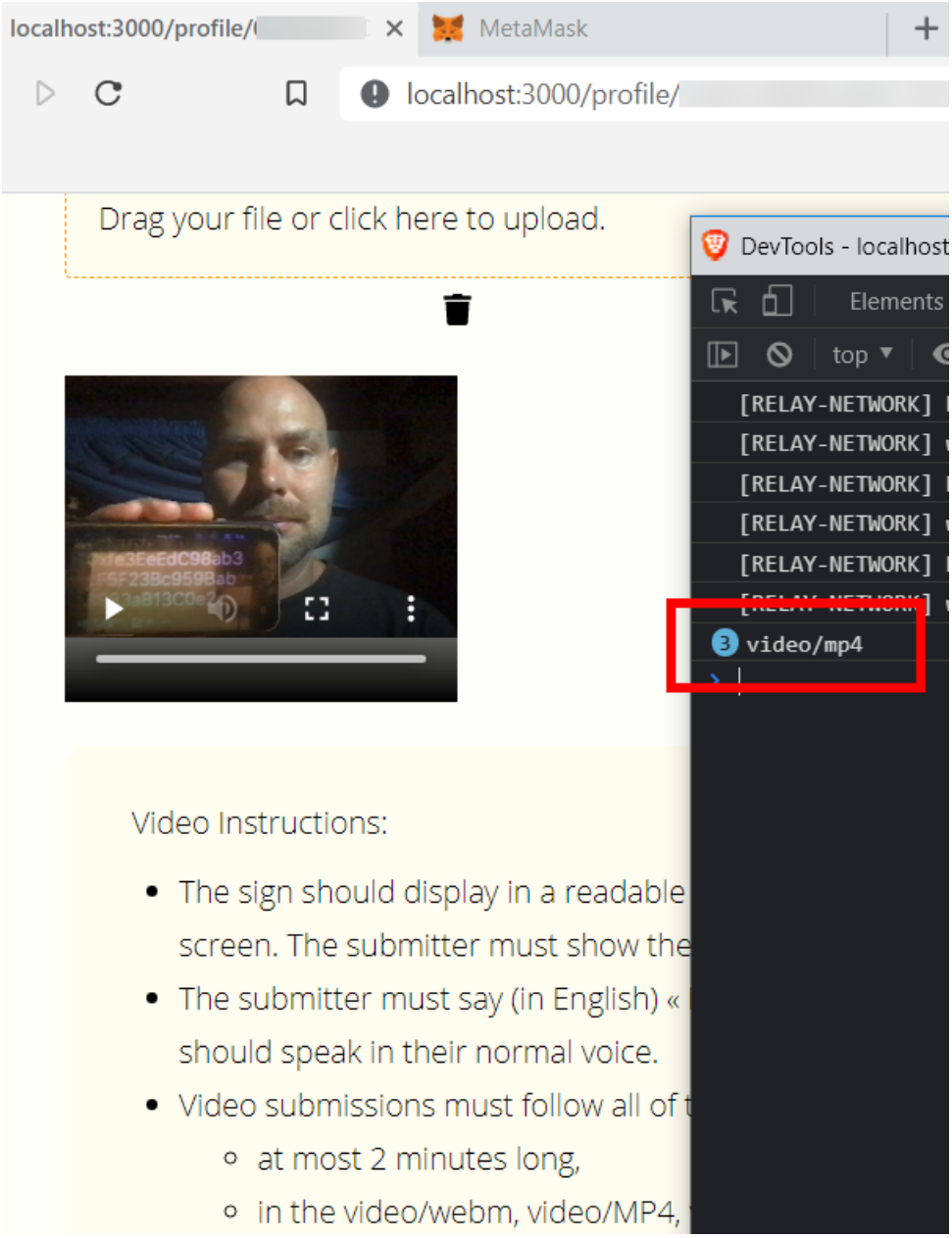
On the other hand, the challenger has a misconception of the format used by the video (No media type with the name video/m4v has been standardized. You can see http://www.iana.org/assignments/media-types),

so I tested this myself.

I deployed the project locally and added a line that would output the format of the video in question to the console.

```
.test(
  "fileType",
  `Video should be one of the following types: ${VIDEO_OPTIONS.types.
  label}`,
  (value) =>
    !value
      ? true
      : VIDEO_OPTIONS.types.value.some((allowedMimeType) => {
          const [mimeType] = String(value.type)
            .toLowerCase()
            .split(";");
          // eslint-disable-next-line no-console
          console.log(mimeType);
          return mimeType === allowedMimeType;
        })
)
```

And when I add the video with the m4v extension, the console outputs the **video/mp4 format** as the one added by the user.



As you can see, even the validation done by the YUP component allows the video to be added without any problem, since its format is video/mp4.

Then, based on the fact that the video can be checked without problems, that it also complies with the requirements, and that there is jurisprudence on the type of case,

I am completely sure that it is a clear yes.

Thank you for your time.