

Who Am I?



Keyur Paralkar

Front-end developer 🧑💻; Book enthusiasts 📖

 @keurplkar

 keyurparalkar

 keyurparalkar

freeCodeCamp (🔥)

 keyurparalkar



Enhancing video experience with video frame snapshot tooltips

Keyur Paralkar

The What ?

- We are building a component that displays the video's frame at that specific second. Something like below:



The Why ?

- Explore real-world scenarios and complexities.
- Implementation without hacks
- Enhance system design mindset.

Some Backstory

- This talk's topic is a part of this project here: [react-youtube-player-clone](#)
- In this project, I am trying to create a clone of the youtube's video player.
- You guys can read more about the project in this [blog post series](#).

Some Backstory: Project Architecture

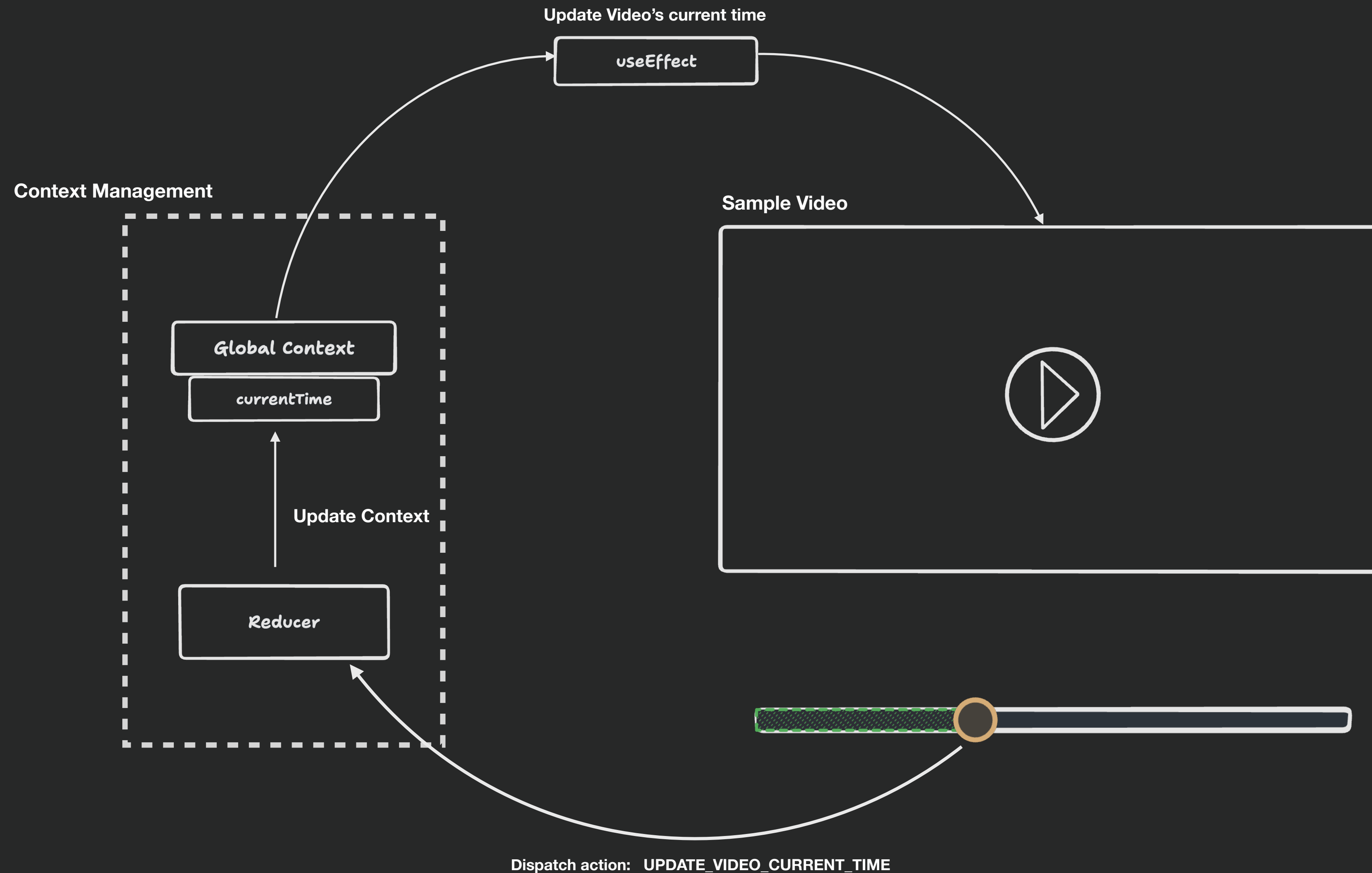
Context Management



Sample Video

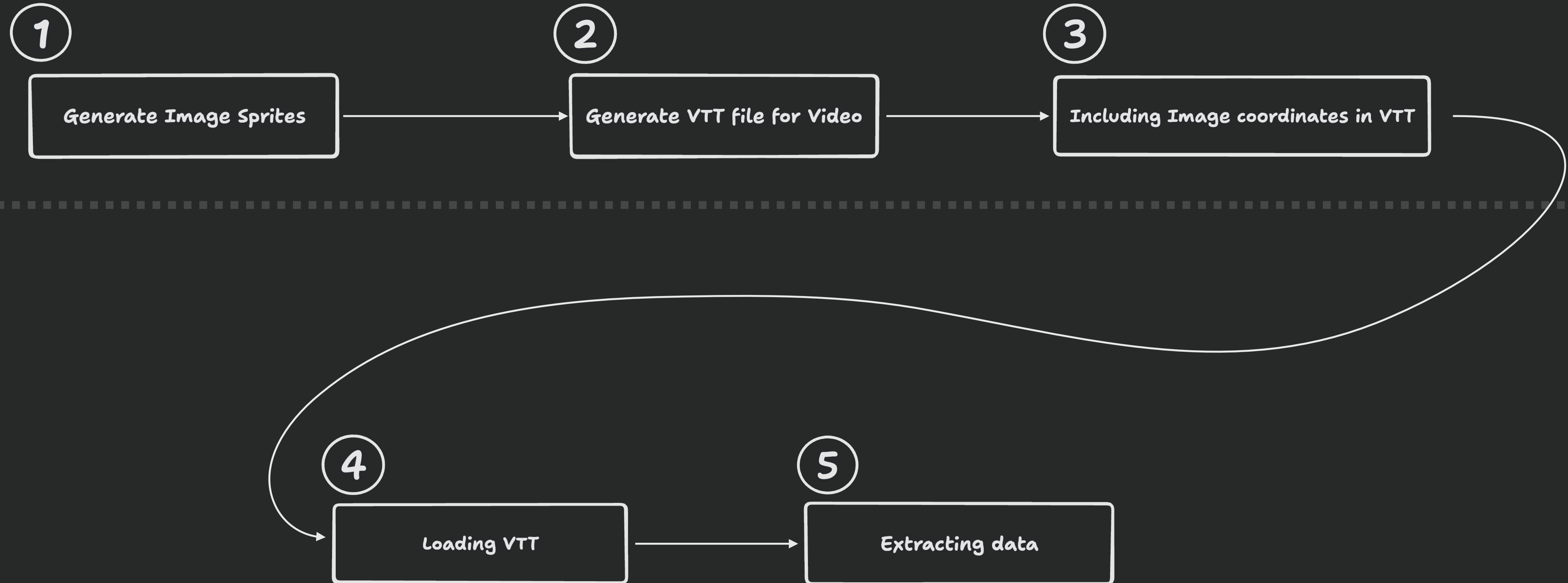


Some Backstory: Project Architecture



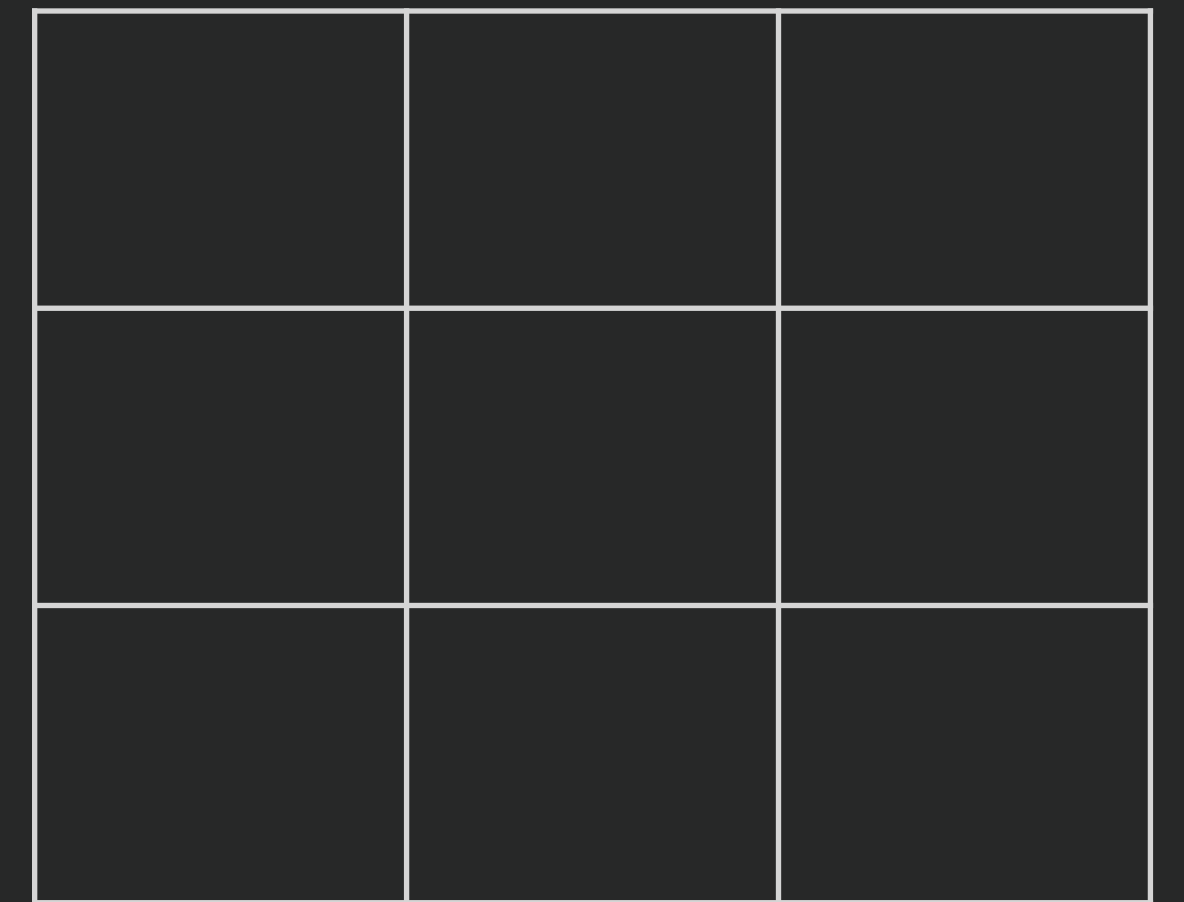
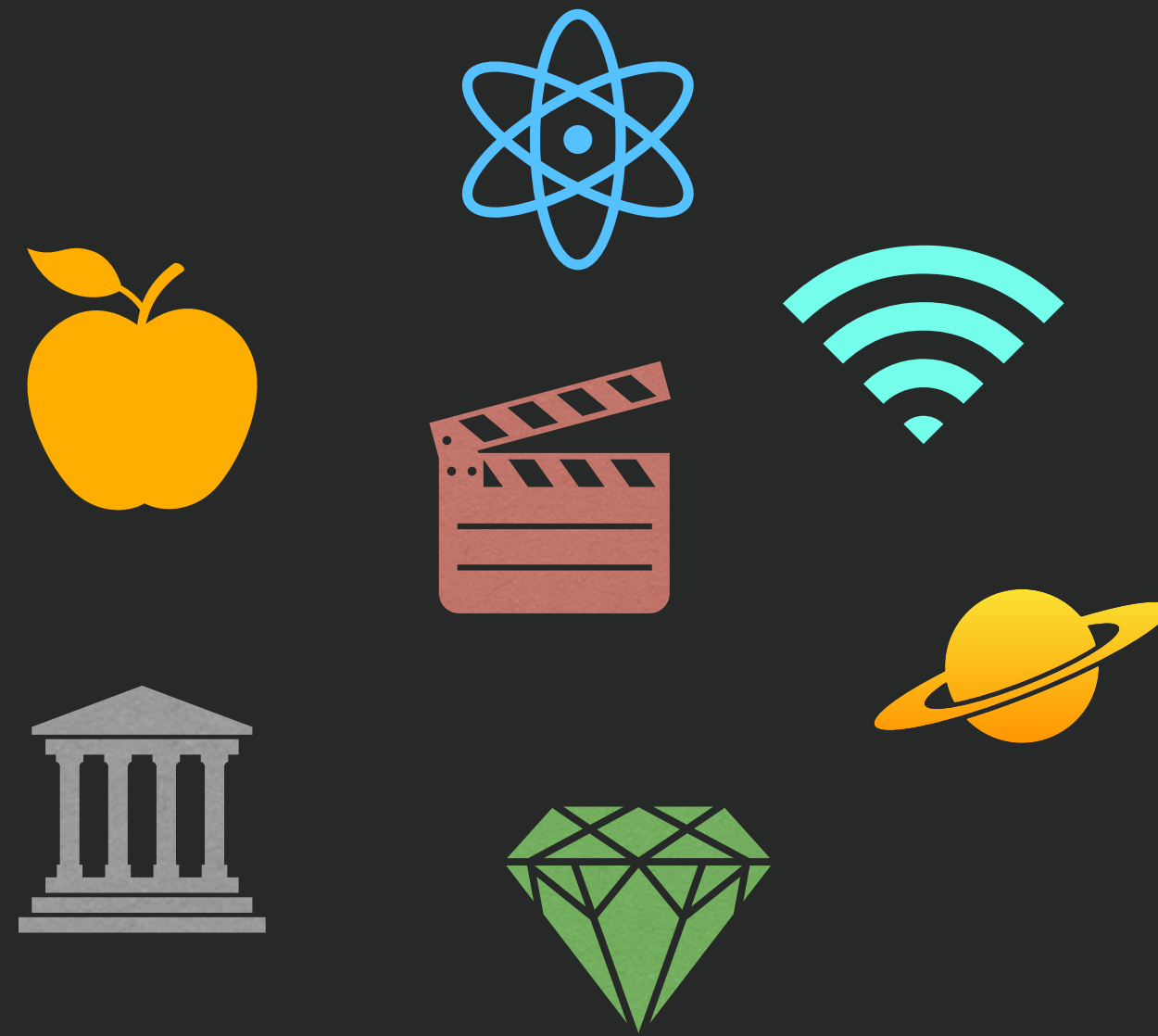
How?

Video Preprocessing



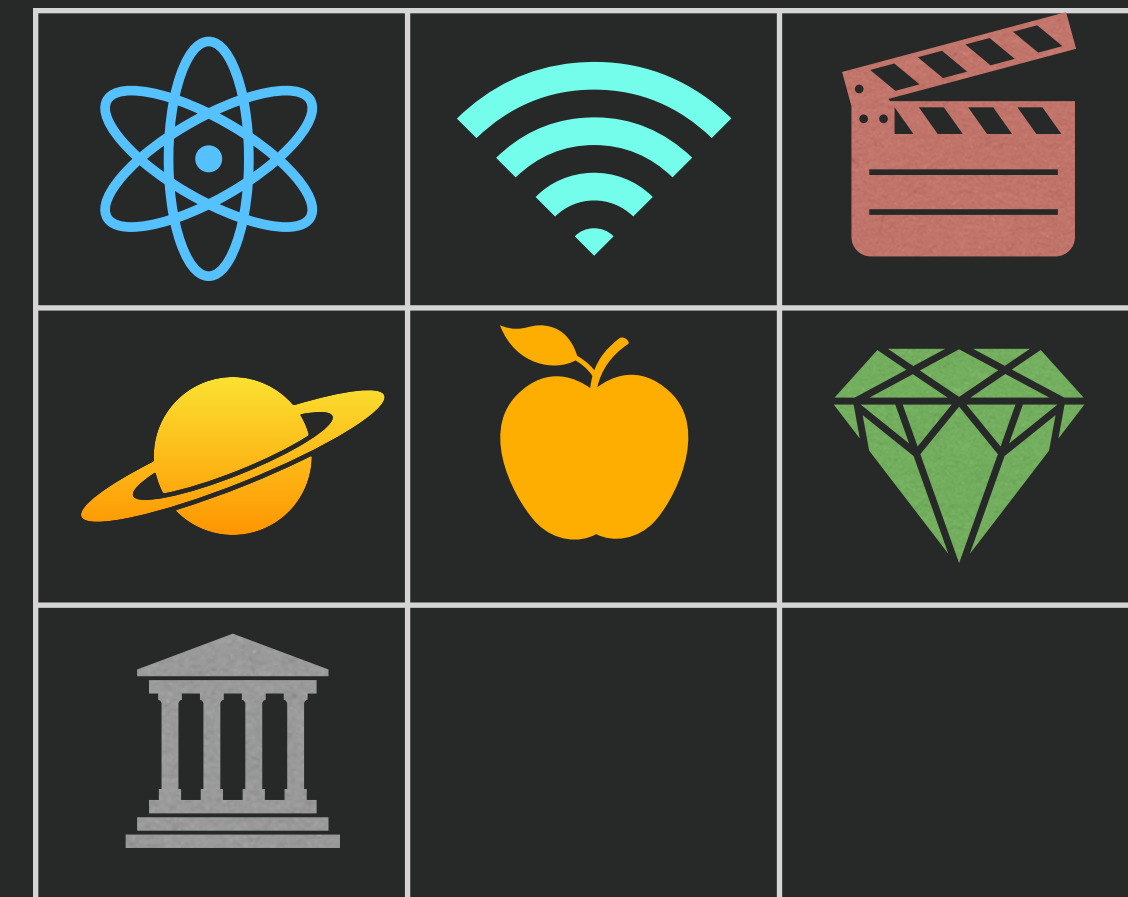
Preprocessing Step: Generating Image Sprites

- What is an Image Sprite?
 - It's an image that consists of multiple images.
 - It is efficient to use a single image that consists of bunch of images rather than fetching every image.
 - A single image in a sprite is extracted with CSS



Preprocessing Step: Generating Image Sprites

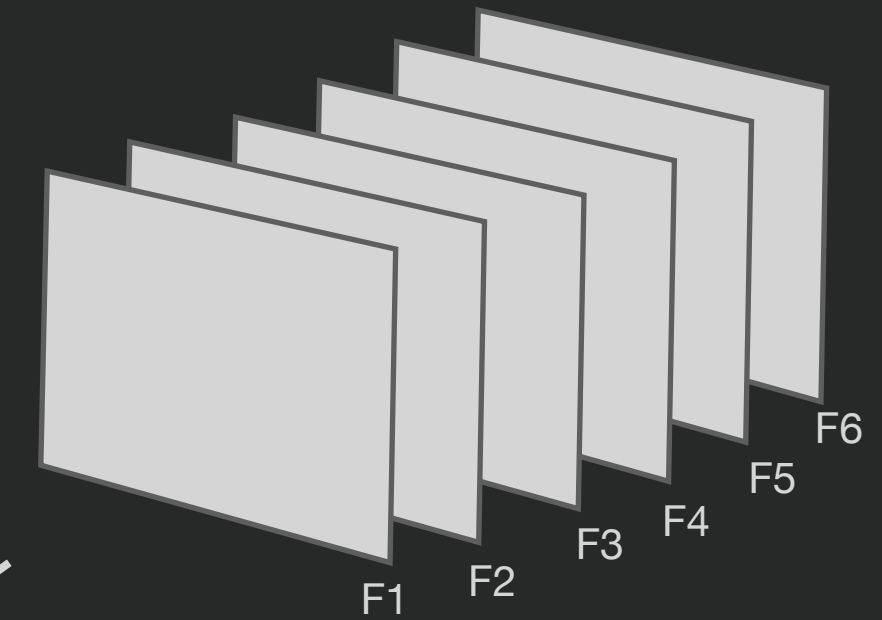
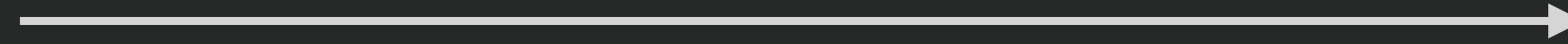
- What is an Image Sprite?
 - It's an image that consists of multiple images.
 - It is efficient to use a single image that consists of bunch of images rather than fetching every image.
 - A single image in a sprite is extracted with CSS
 - It can be horizontal, vertical or grid based.



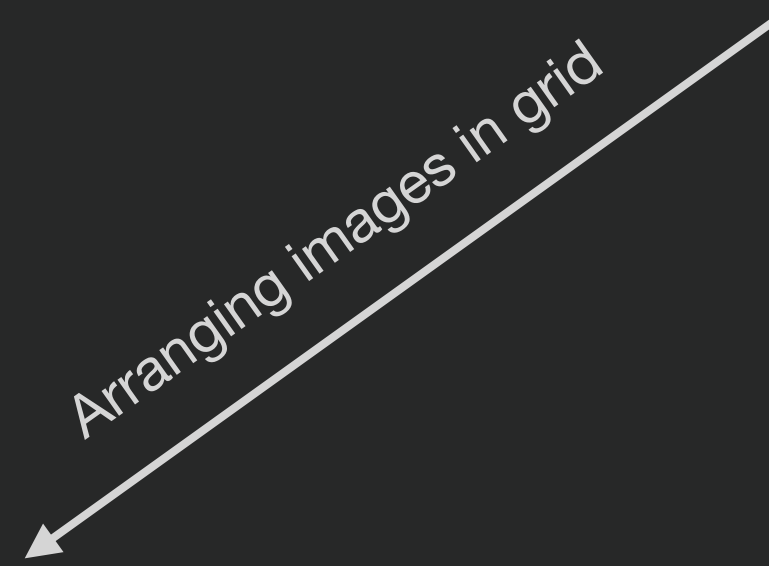
Preprocessing Step: Generating Image Sprites



Converting each video frame into an image



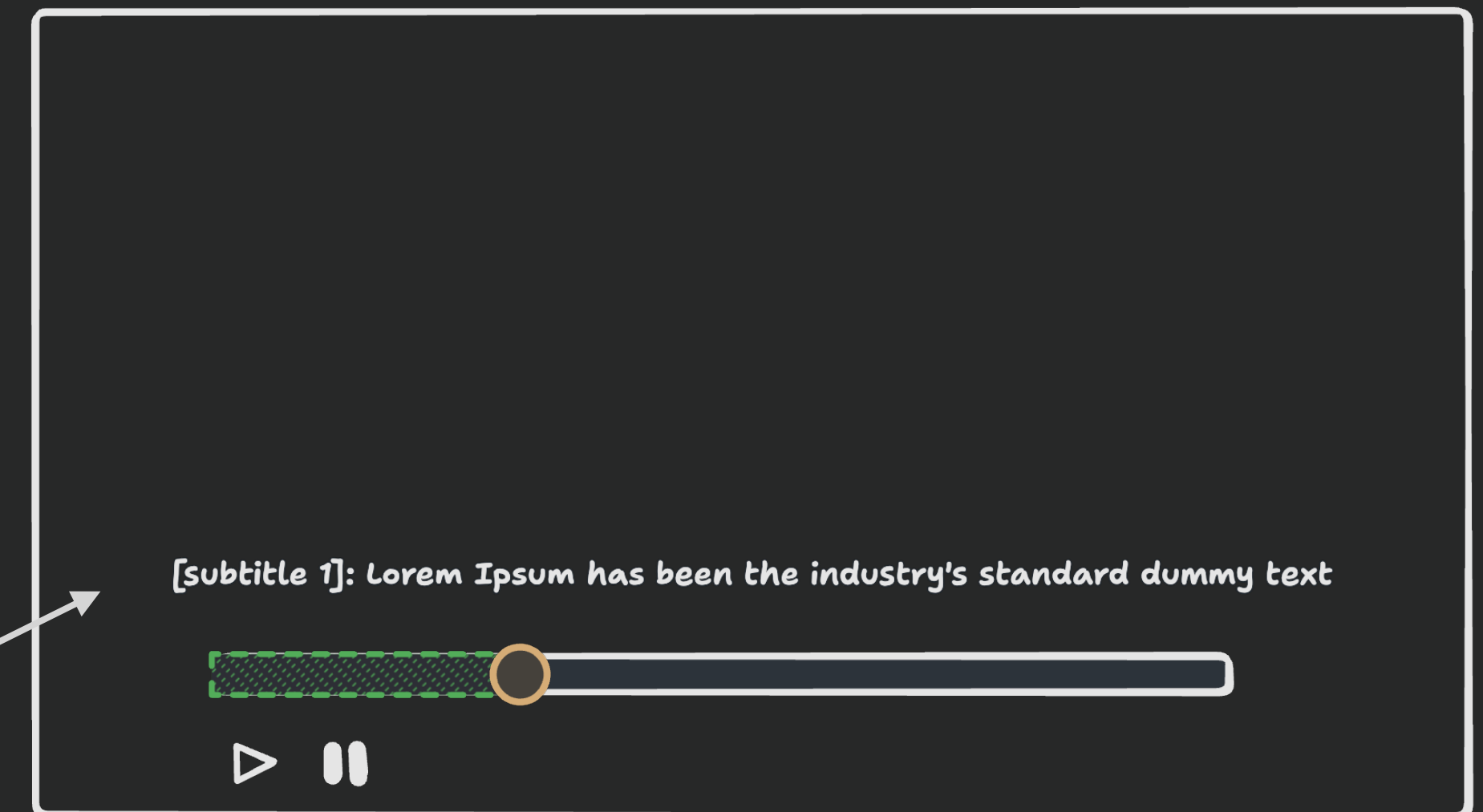
Arranging images in grid



F1	F2	F3	F4
F5	F6	F7	F8
F9	F10	F11	F12
F13	F14	F15	F16

Image Sprite of a video

Preprocessing Step: Generating VTT file for the Video



WEBVTT

00:01.000 --> 00:04.000

- [subtitle 1]: Lorem Ipsum has been the industry's standard dummy text

00:02.000 --> 00:03.000

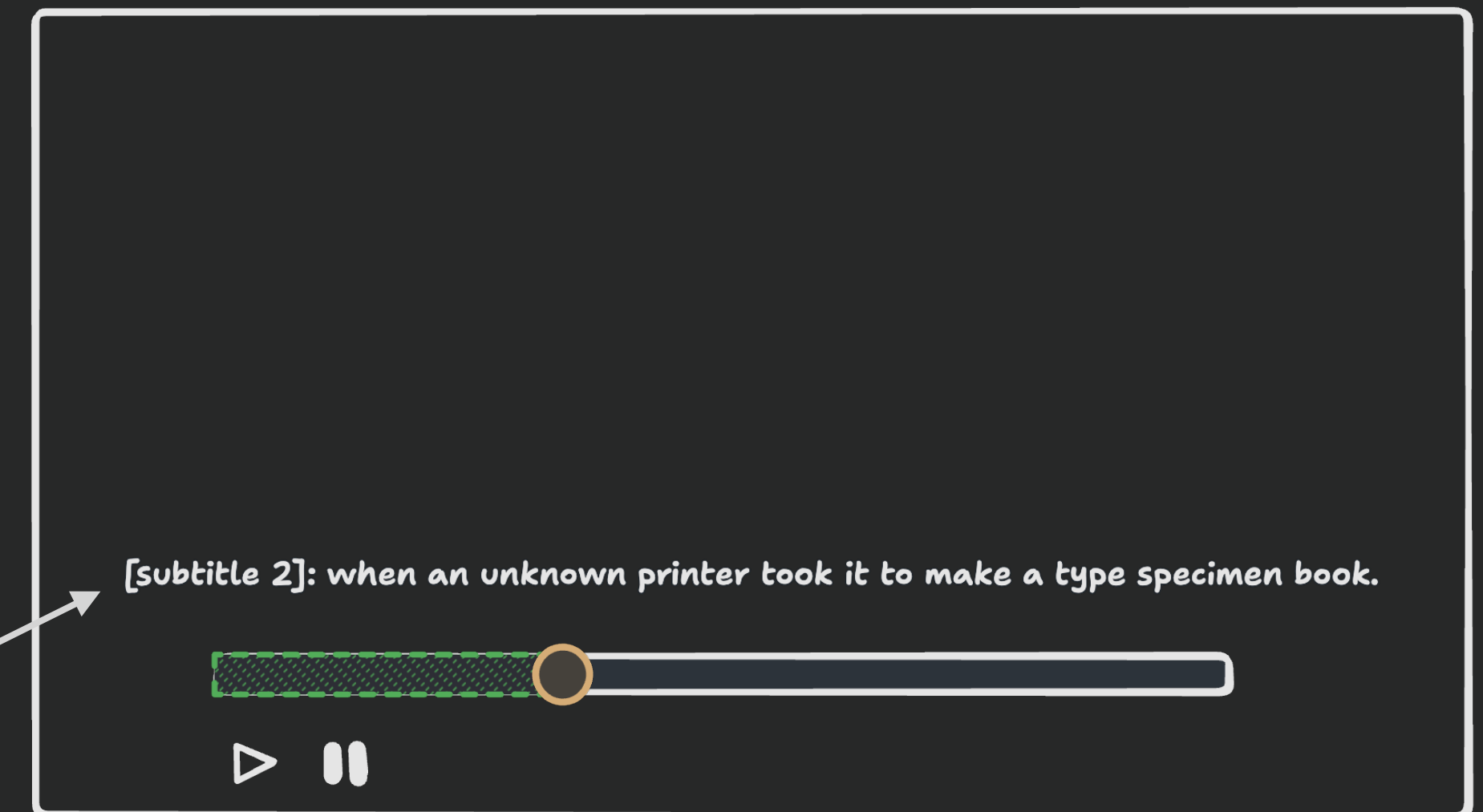
- [subtitle 2]: when an unknown printer took it to make a type specimen book.

<video>

<track default src="subtitle.vtt" />

</video>

Preprocessing Step: Generating VTT file for the Video



WEBVTT

00:01.000 --> 00:04.000

- [subtitle 1]: Lorem Ipsum has been the industry's standard dummy text

00:02.000 --> 00:03.000

- [subtitle 2]: when an unknown printer took it to make a type specimen book.

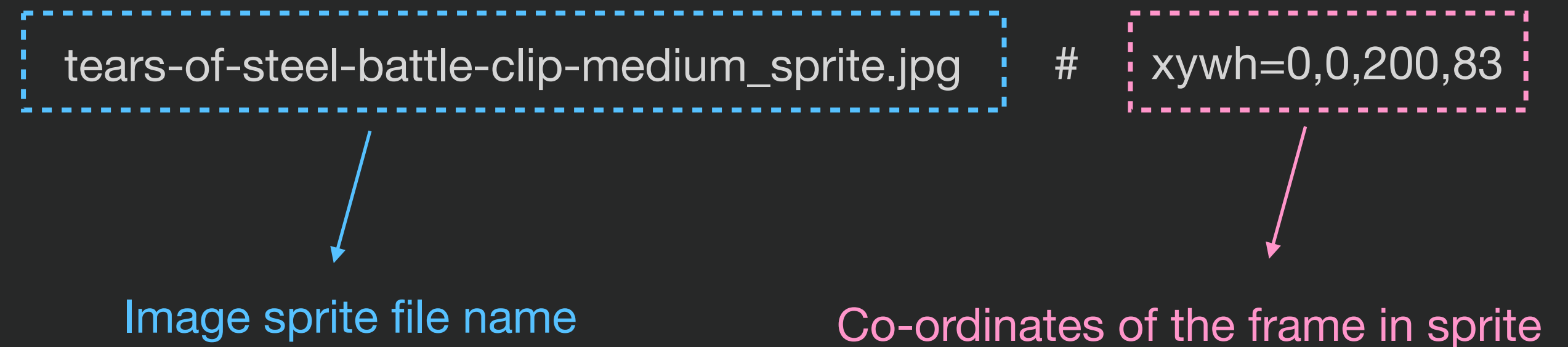
<video>

<track default src="subtitle.vtt" />

</video>

Preprocessing Step: Generating VTT file for the Video

- Our .VTT file will contain for each second the name of the image sprite file along with current seconds video frame co-ordinates
- We store this data in the cues because we can refer to the respective sprite file in that time span of the video
- We later prepend this cue with the host url that store the sprite file e.g. Dropbox



WEBVTT

Img 1

00:00:00.000 --> 00:00:01.000

tears-of-steel-battle-clip-medium_sprite.jpg#xywh=0,0,200,83

Img 2

00:00:01.000 --> 00:00:02.000

tears-of-steel-battle-clip-medium_sprite.jpg#xywh=200,0,200,83

Img 3

00:00:02.000 --> 00:00:03.000

tears-of-steel-battle-clip-medium_sprite.jpg#xywh=400,0,200,83

Img 4

00:00:03.000 --> 00:00:04.000

tears-of-steel-battle-clip-medium_sprite.jpg#xywh=600,0,200,83

Img 5

00:00:04.000 --> 00:00:05.000

tears-of-steel-battle-clip-medium_sprite.jpg#xywh=800,0,200,83

Loading and Extracting: Loading the VTT

- We make use of the track element as to load our VTT file
- Once the video is loaded we fetch each cue based on the current duration and store in the global context

Loading and Extracting: Extracting

- We access the stored cue in the previous step with the help of context APIs in our video frame tooltip component.
- We construct the URL from the cue and pass it to a styled component.
- Along with the URL, we also pass the offset values so that the video frame image can be extracted from the image sprite.

Let's Dive into the implementation

Thank you