

The background is a dark, stylized illustration of a cyberpunk city street at night. In the center, a large building features a prominent, glowing Bilibili logo. The scene is filled with neon lights, rain, and a sense of depth, with a silhouette of a person visible on the right side.

WebAssembly

在多媒体场景的实践与思考

bilibili 主站技术中心 B端技术部 高裕轩



成为UP主的一小步

文件上传

(推荐采用mp4、flv格式，可有效缩短审核转码耗时)

P1

BBC.Horizon.2014.Inside.the.Dark.Web.720p.HDTV.x264.AAC.MVGroup.org

0%

已经上传：4.1MB/1.3GB

当前速度：18.2MB/s

剩余时间：1.3分钟

采用mp4或flv格式可以更快转码噢！

+ 添加视频

基本信息

使用投稿模版

|

⚙

视频封面设置

(格式jpeg、png，文件大小≤5MB，建议尺寸≥1146*717，最低尺寸≥960*600)

上传封面

正在生成可选封面，系统会默认选中第一张作为视频封面。

优化用户体验时遇到的问题

有问题的文件需要等上传完成之后在后端检测

- 非视频文件或无法解码的文件
- 视频轨道缺失
- 特定类型的投稿需要分辨率限制

文件上传 (推荐采用mp4、flv格式, 可有效缩短审核转码耗时)



BBC.Horizon.2014.Inside.the.Dark.Web.720p.HDTV.x264.AAC.MVGroup.org

0%

已经上传: 4.1MB/1.3GB 当前速度: 18.2MB/s 剩余时间: 1.3分钟 采用mp4或flv格式可以更快转码噢!

优化用户体验时遇到的问题

有问题的文件需要等上传完成之后在后端检测

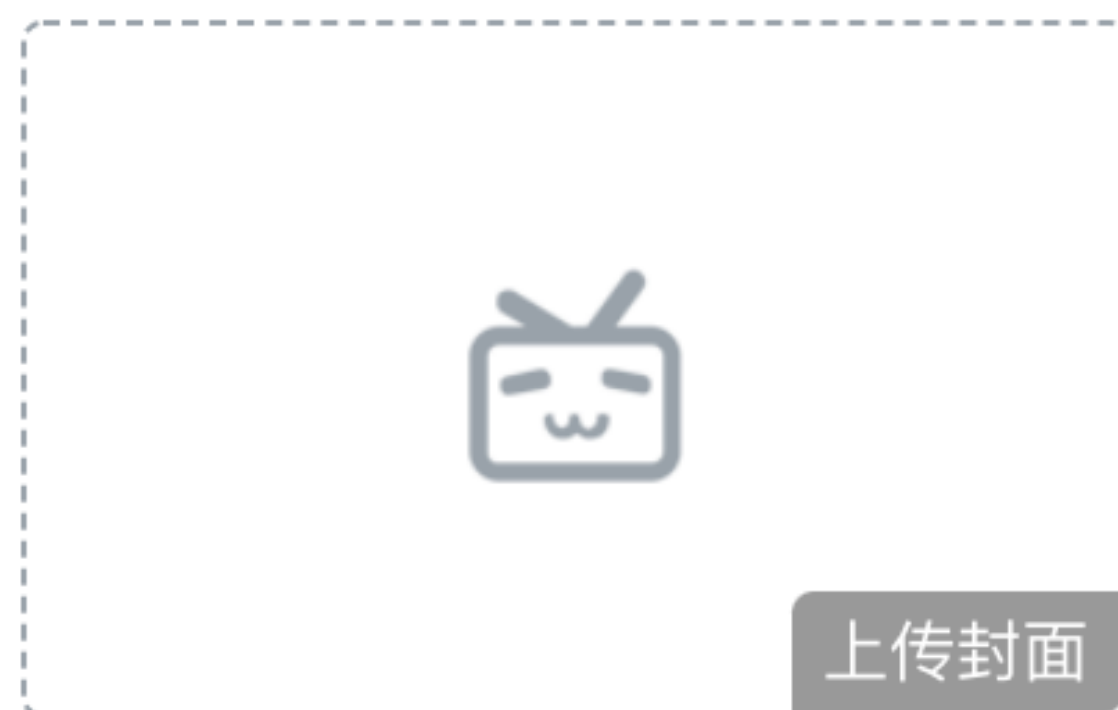
- 用户后知后觉
- 云端资源被白白浪费

优化用户体验时遇到的问题

稿件的推荐封面需要等到稿件转码完成后才能获得

- 推荐封面： 视频截图 + AI打分
- 用户在提交前可能还看不到推荐封面

视频封面设置 （格式jpeg、png，文件大小 $\leq 5\text{MB}$ ，建议尺寸 $\geq 1146 \times 717$ ，最低尺寸 $\geq 960 \times 600$ ）



正在生成可选封面，系统会默认选中第一张作为视频封面。



优化用户体验时遇到的问题

前端处理？

- 在文件上传之前检测文件信息并拦截
 - 用户可以得到及时的反馈
 - 节省了云端资源
- 在本地截取视频截图并通过AI模型进行封面推荐
 - 即时展现推荐封面
 - 减轻云端的计算压力

优化用户体验时遇到的问题

面临的挑战

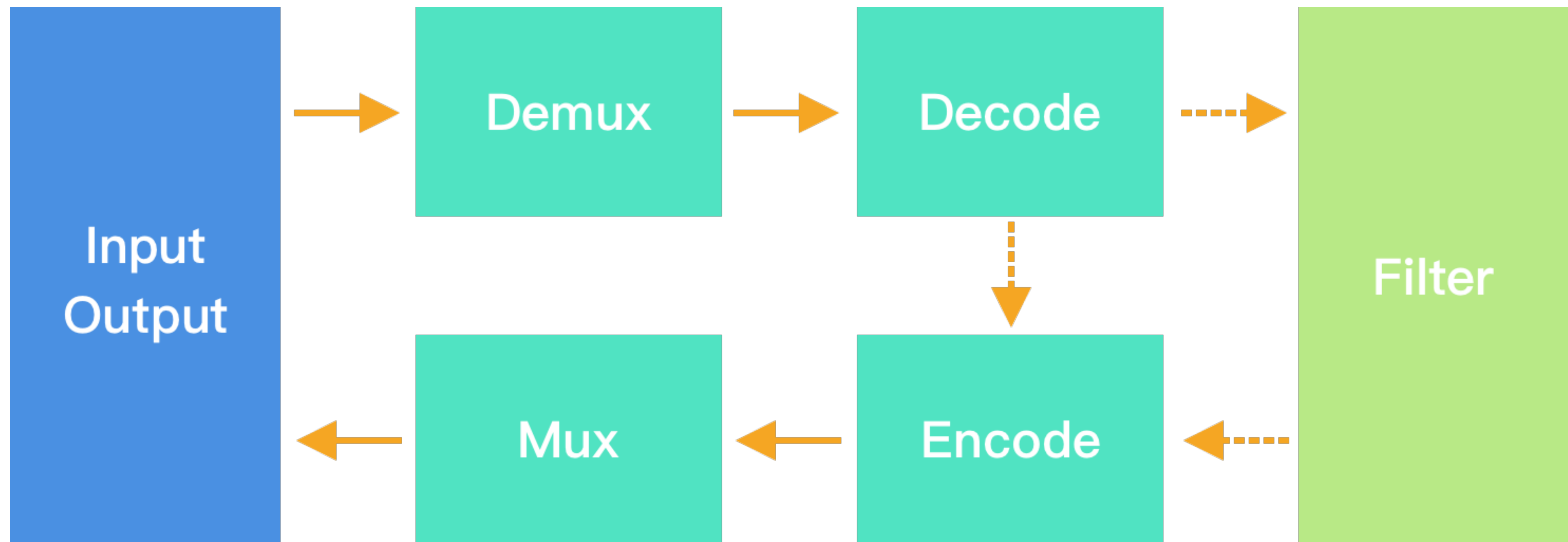
- 需要面对各种封装和编码的视频文件
- 需要有分析和处理多媒体文件的能力

引入WebAssembly

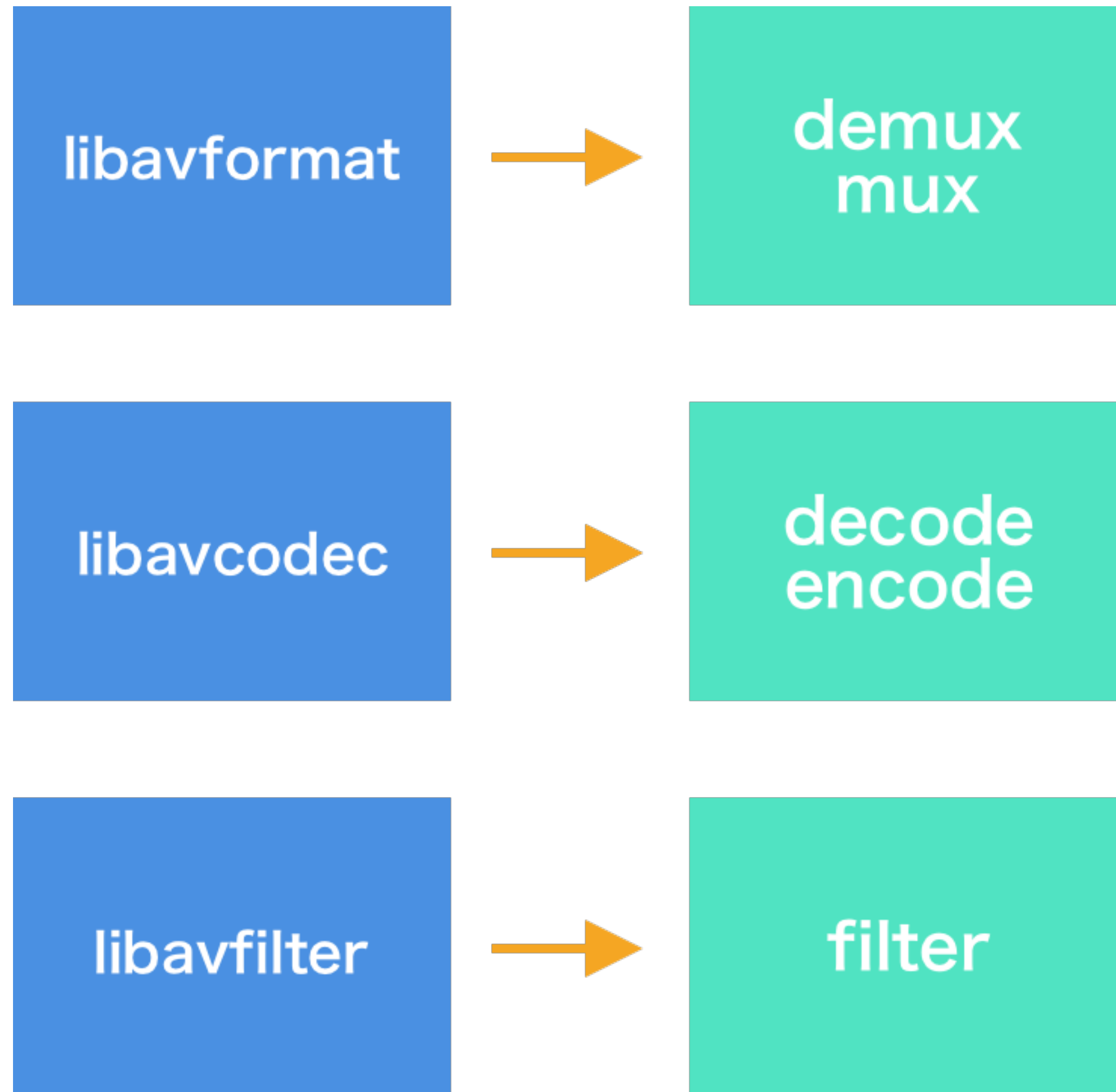
最初的梦想

- 多媒体处理场景在C/C++生态中已有相对成熟的实现
- WebAssembly标准得到广泛的支持
- C/C++到WebAssembly的工具链相对完善

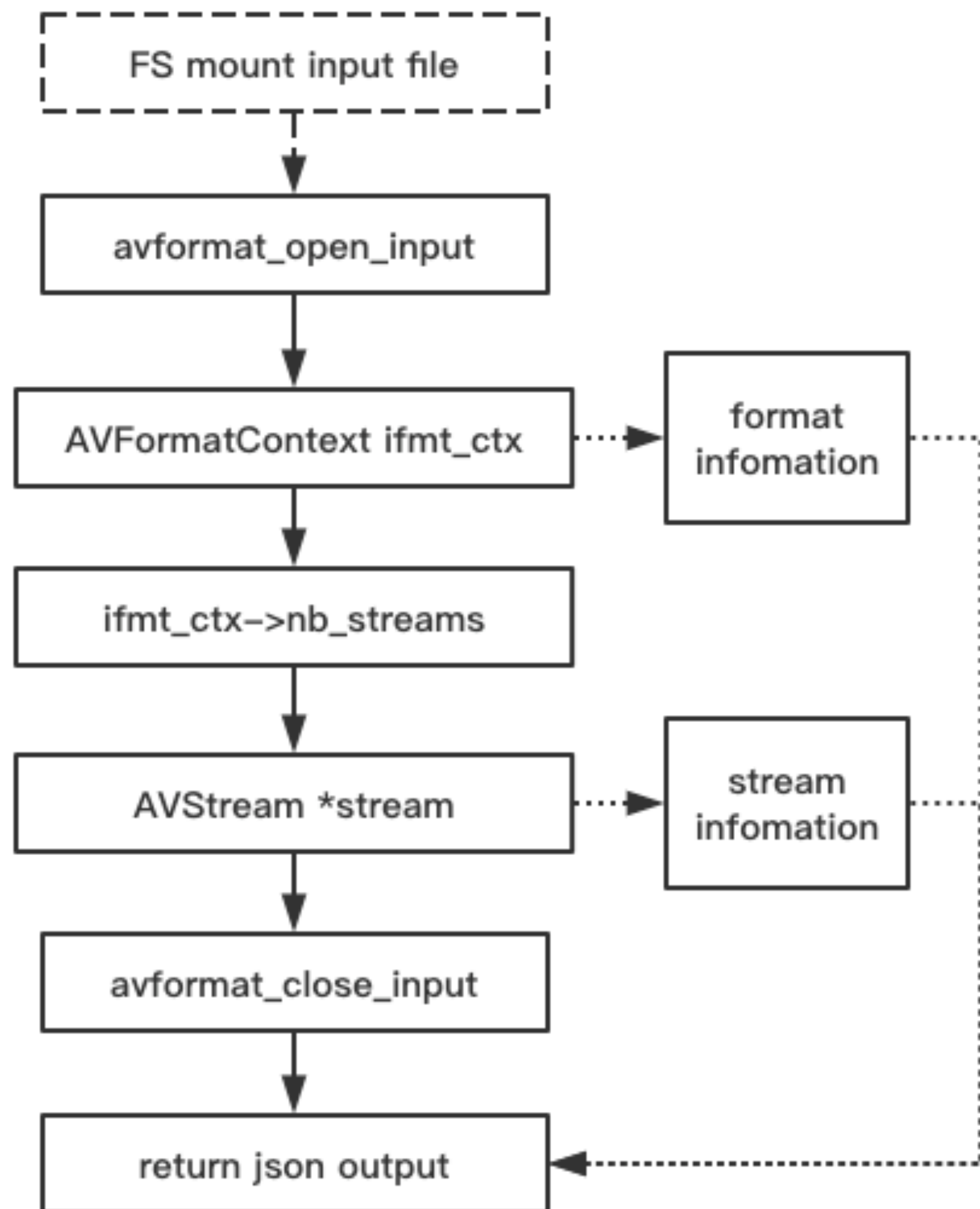
使用C++处理多媒体文件



使用C++处理多媒体文件

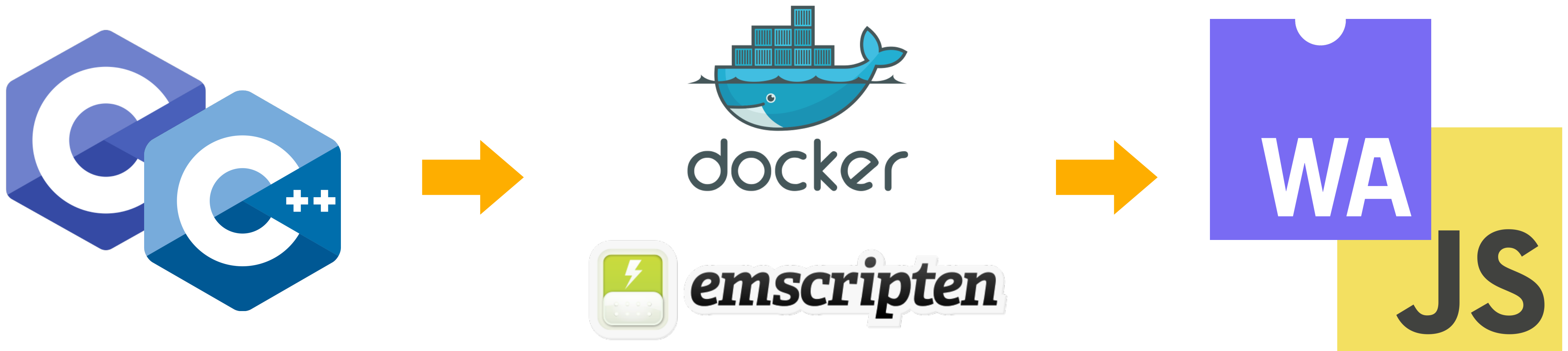


使用C++处理多媒体文件



```
▼ format:
  nb_streams: 2
  nb_programs: 0
  format_name: "mov,mp4,m4a,3gp,3g2,mj2"
  start_time: 0
  duration: 8
  bit_rate: 5195621
  probe_score: 100
  size: 5195621
  ► __proto__: Object
▼ streams: Array(2)
  ▼ 0:
    codec_type: "video"
    codec_name: "h264"
    duration: 8
    width: 1920
    height: 1080
    rotation: 0
    sample_aspect_ratio: ""
    display_aspect_ratio: ""
    ► __proto__: Object
  ▼ 1:
    codec_type: "audio"
    codec_name: "aac"
    duration: 8
    ► __proto__: Object
  length: 2
```

C++编译到WebAssembly



C++编译到WebAssembly

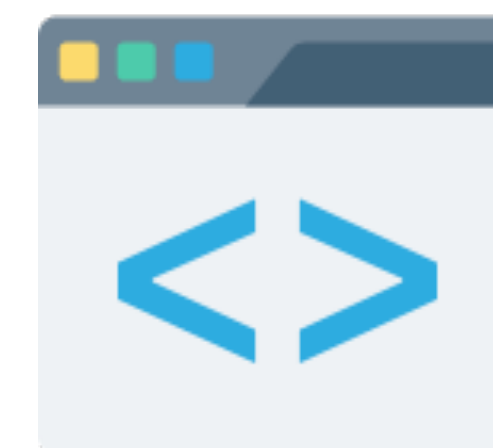
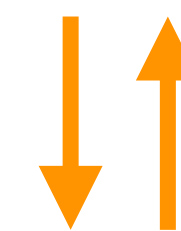


使用WebAssembly

Worker + WebAssembly

数据通信

- Transferable
- WORKERFS (FileReaderSync)
- IndexedDB

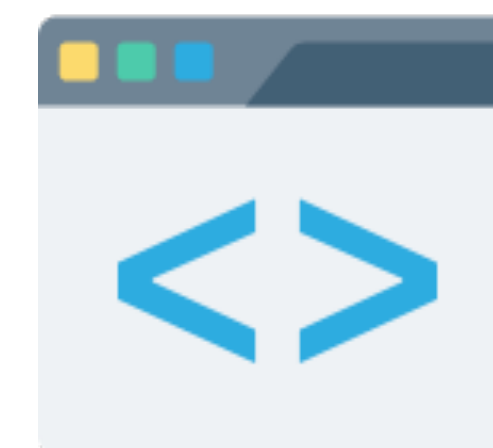
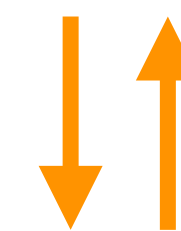


使用WebAssembly

Worker + WebAssembly

数据通信

- Transferable
- WORKERFS (FileReaderSync)
- IndexedDB
- SharedArrayBuffer ?

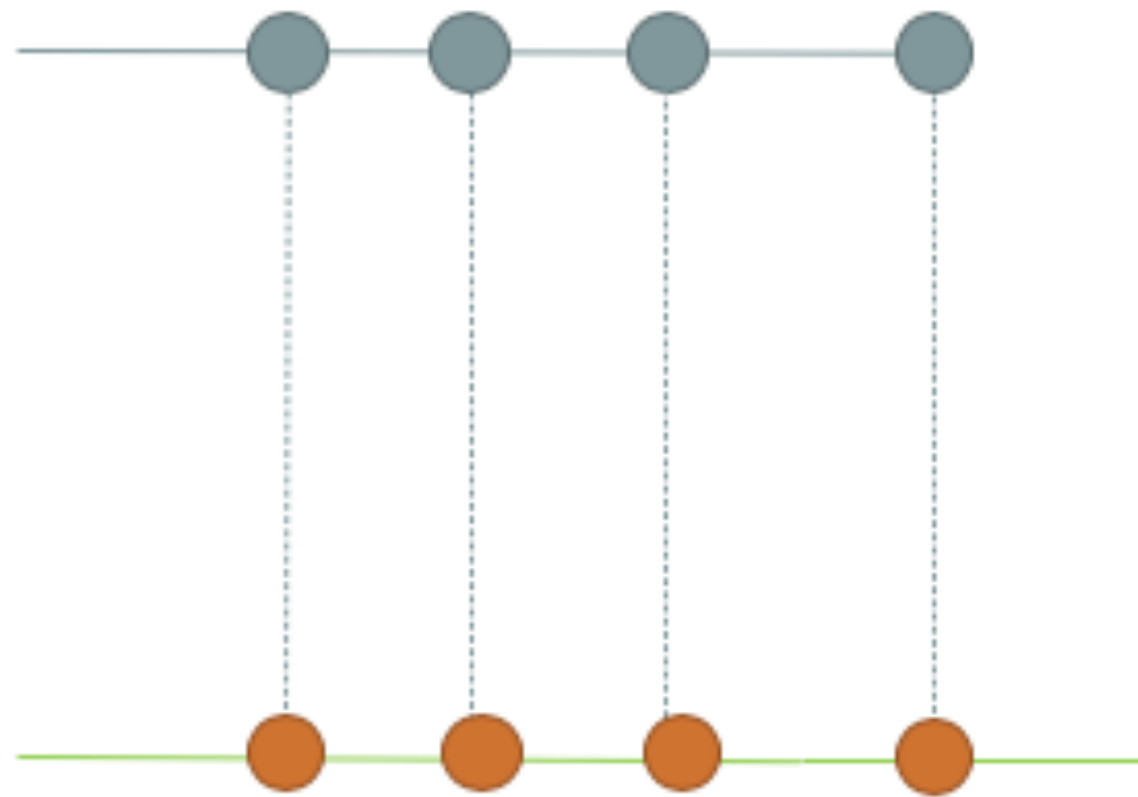


使用WebAssembly

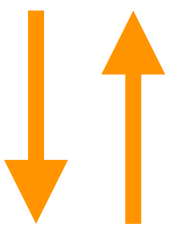
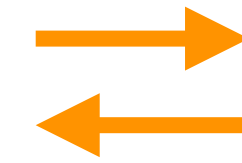
多个Worker协作

- 流水线式处理

WebAssembly worker: 生产截图



TensorFlow worker: 为图片打分



TensorFlow

控制文件体积

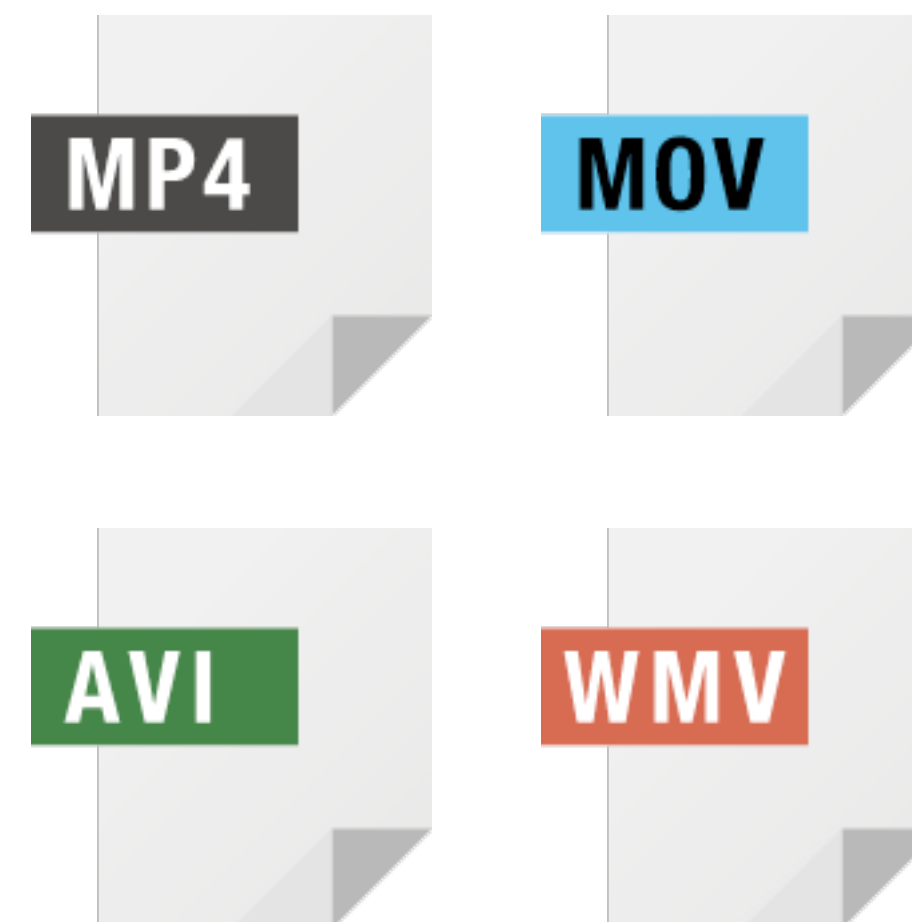
FFMpeg编译配置文件

- 去除不需要的工具库和依赖项
- 使用全部Demuxer + Decoder
 - 约26Mb
- 使用全部Demuxer + 部分Decoder
 - 分析以往用户投递的稿件选取了20+种decoder
 - 约6Mb

性能

检测多媒体文件信息

- 随机批量处理500个文件
- 单文件平均耗时 ~40ms

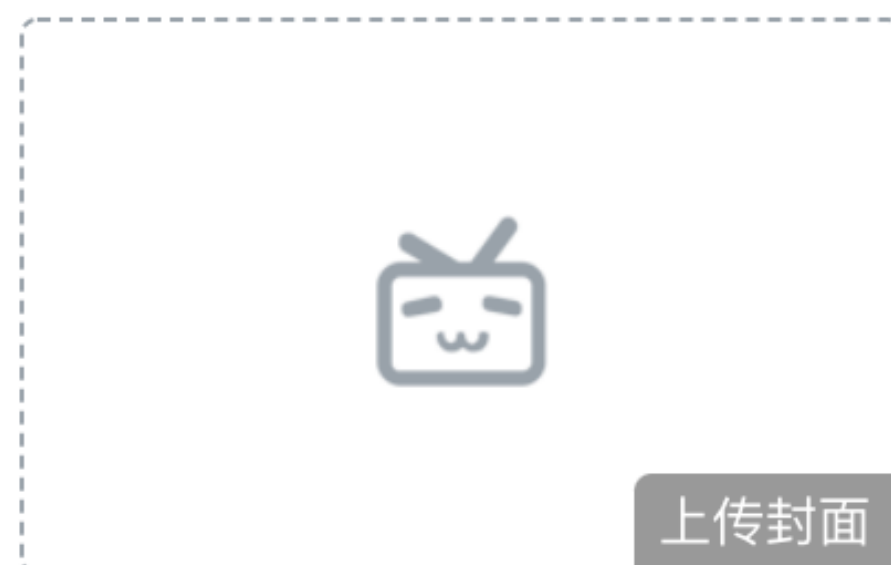


性能

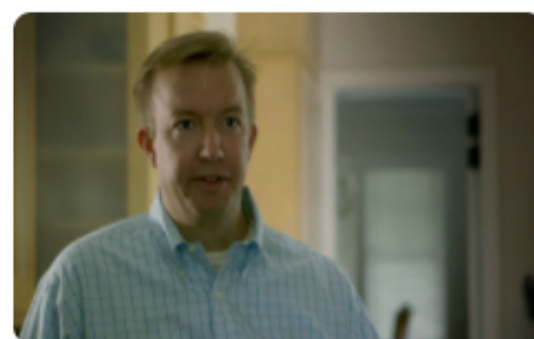
生成推荐封面

- 生成30张视频截图(略缩图) + tensorflow打分 1.2s
- 生成9张推荐封面 1s

视频封面设置 (格式jpeg、png, 文件大小 $\leq 5\text{MB}$, 建议尺寸 $\geq 1146 \times 717$, 最低尺寸 $\geq 960 \times 600$)



可选择以下封面。 [查看更多封面](#)

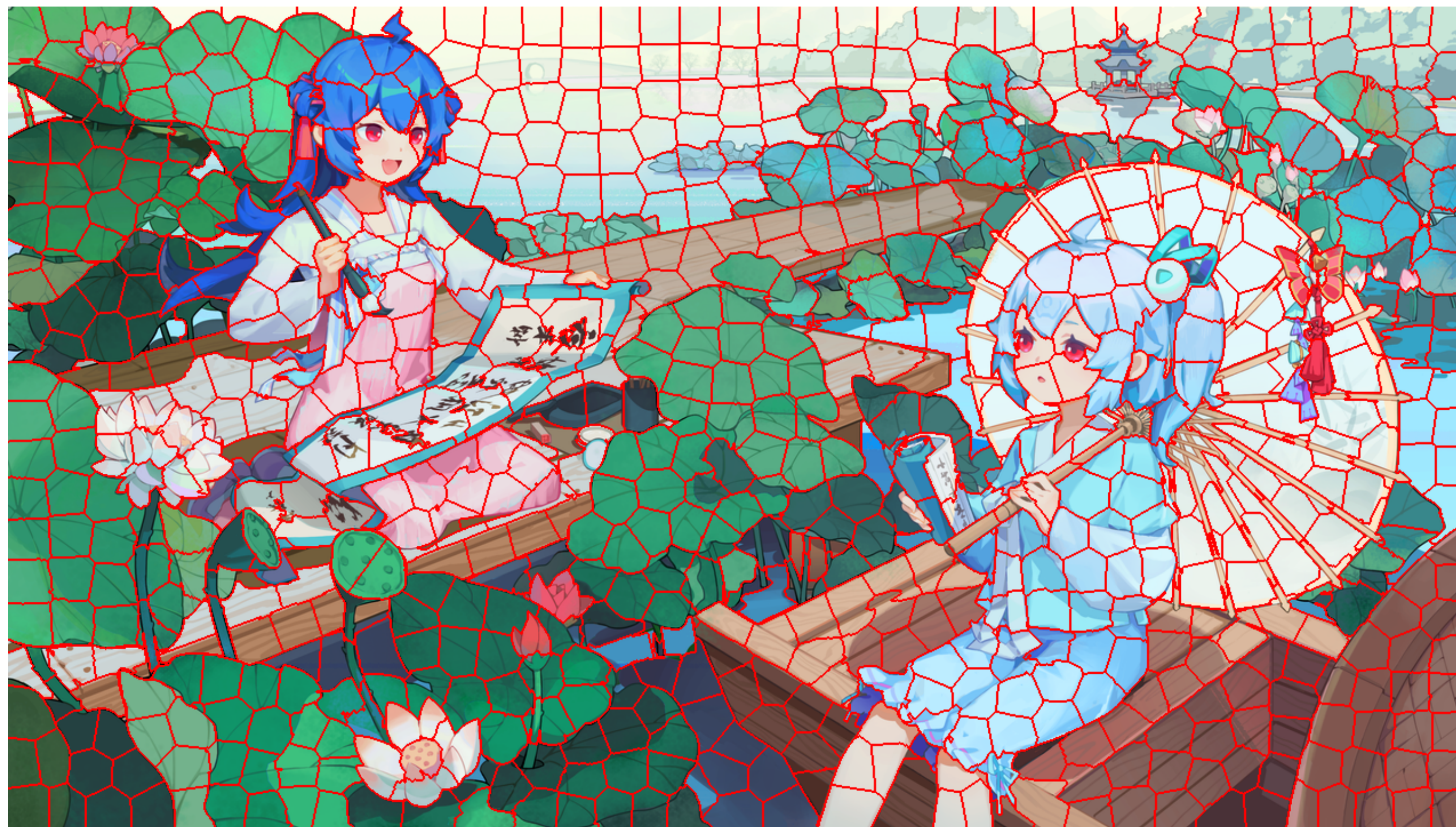


更多场景

图像

音频

视频



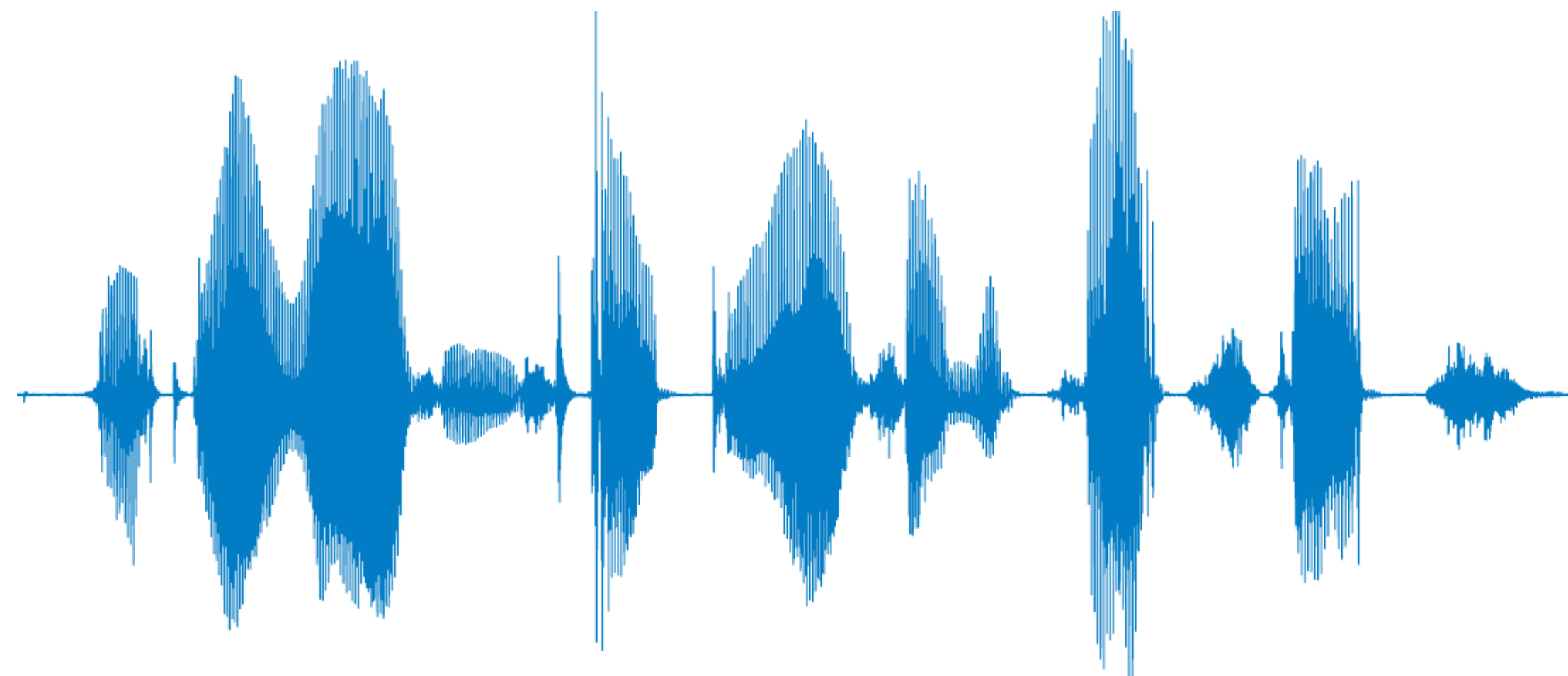
使用WebAssembly进行图像分割

更多场景

图像

音频

视频



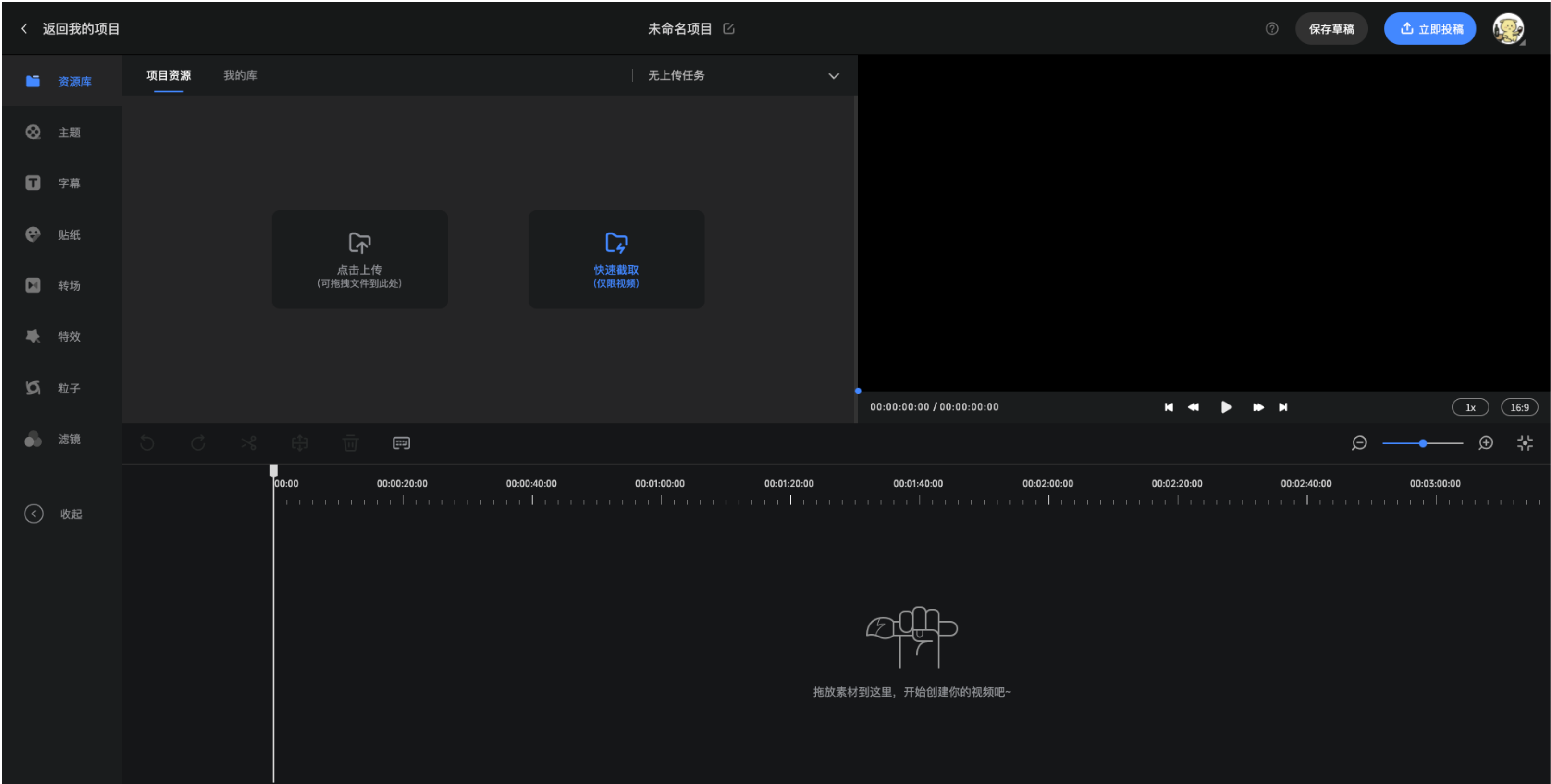
Web Audio API
AudioWorklet
+
WebAssembly

更多场景

图像

音频

视频



bilibili云剪辑界面



思考与提议

提供编/解码器、封装格式相关的标准API

结合WebAssembly扩展浏览器多媒体编解码与处理能力

- 现有场景：MSE API + WebAssembly
- 相关讨论：WICG/web-codecs

思考与提议

Web worker(Worklet)支持直接使用wasm文件

- API:
 - `new Worker(aURL, options)`
 - `worklet.addModule(moduleURL, options)`
- 现状:
 - Worklet内部不能使用fetch API wasm只能通过内联形式引入
- 相关提案: ES Module Integration Proposal for WebAssembly (Stage 2)



感谢观看

哔哩哔哩 (° - °) つ口 干杯~