

# 技术选型

## 涉及的关键技术

- PPT、key的动画展示
- 代码编辑器的展示
- 白板的展示如何实现
- 如何广播左侧屏幕（代码编辑器+白板+PPT）+视频
- 中途加入的学生获取当前直播间状态
- 录播如何展示

除了以上提到的方面，还有一些易于实现的部分，如评论功能的实现，账号系统的管理等。这些问题的解决方案比较成熟，不在调研范围之内（或仅简单提及）。

## 前端选型

### 1.Vue

- 前端架构，我们只学习了该架构。由于学习、时间成本等原因未对其做深入调研

### 2.其他

- 将采用webpack，gulp等工具，因为是主流方案，未进行过多调查

## 后端选型

### 1.Django

- 后端架构，同上，未对其他相似技术进行调研

### 2.Socket.io

- 用于通信，调研其文档，未调研其他技术

以下为组件的调研内容，我们作了相当广泛的调研。最终选型基于多方面确定，将在该部分提及。

## 组件选型（其最终选型在此部分最末）

### 白板

名称格式：作者/项目名

#### 1. literallycanvas/literallycanvas

描述：项目部署出错

*BSD协议*

- 迷之解决，服务器部署似乎需要使用魔法完成，暂时未复现
- 考虑到项目解决方案变动，不再需要网络通讯部分，我们觉定拆分该项目，仅保留本地模块
- 暂定

#### 2. erichonorez/whiteboard

描述：可以正常运行，但是只有画笔。单一的笔触绘图，功能太少

- 放弃

#### 3. bharatpatil/whiteboard

描述：npm无法正常部署

- 放弃

#### 4. katspaugh/Multiplayer Whiteboard

描述：npm 无法正常部署，功能无法满足需求

- 放弃

#### 5.cyrusinnovation/whiteboard

描述：npm无法正常部署,尝试多次后未解决

- 放弃

#### 6.声网 - 白板组件

描述：文档过于简单

- 放弃

#### 7.ramroll/blackboard

描述：部署成功且可以运行，其白板功能较多，自带一个代码编辑器（查看之后发现用的是codemirror），支持普通绘图功能以及录音功能，不支持几何图形功能，由于literallyCanvas功能较为齐全，所以此blackboard可以用来做代码的参考已改进白板的功能

**未注明LICENSE**

- 放弃
- 可用作参考

#### 8.streams-blackboard

描述：home.html打开之后显示 `renderPage`，怀疑是配置或者是代码问题，github的README中给的网页打不开，看示例图只有普通的笔触功能

- 放弃

#### 9. vrobby/whiteboard

描述：可以部署与运行，具有笔触画图，调用摄像头，聊天室的功能。对聊天室的实现可以提供代码参考。

**MIT协议**

- 放弃
- 可作参考

10.upsuper/white 描述：README写的十分详细，看起来功能十分完善，部署没问题，运行时出错，尝试过多种方法后未能解决

- 放弃

11.vinsonchuong/whiteboard 描述：部署没问题，运行报错 Path must be a string. Received null，尝试多次过后未解决

- 放弃

### 代码编辑器

#### 1.ace

描述：配置非常方便，高亮，代码补全，空格等属性设置方便，用js编写，易懂易配置，可根据自己需求配置后嵌入网页。

**BSD协议**

- 选用

#### 2.codemirror

描述：配置比起ace麻烦一些，使用js编写，demo效果不错 **MIT协议**

- 备选

#### 3.EditArea

描述：配置资料很少，比起codemirror还要麻烦，没有继续深入调研。

- 放弃

# 聊天室

## 1.thesadboy/Chatroom

描述：通过node.js和socket.io实现的一个简单的聊天室，可以实现上下线广播，广播聊天和点对点聊天。该项目由js编写，还包含登陆注册界面，如果使用，需要去掉该功能。 **未标明LICENSE**

- 作为参考

# 视频分享

## 1.声网

描述：效果满足要求，部署过程简单易操作。但是官方文档过于简单且书写混乱，理解起来十分困难；对于API的描述过于简略，同时SDK中js文件经过压缩导致可读性差，且与最终的解决方案不符。

- 放弃

## 2.nile.js

描述：npm无法正确部署与运行，多人尝试多次均失败

- 放弃

## 3.RecordRTC

描述：可以进行HTML的元素录制操作，支持使用socket.IO进行通讯 **MIT协议**

- 选用

# PPT播放

在网页上播放PPT一般有以下几种实现方法：

### 1.PPT → PDF

- 如果不要求PPT的动画效果，这种方案可行性很高；但在询问后得知客户要求实现动画效果，因此放弃这种解决方案

### 2.PPT → swf

- flash格式会显得网站破旧，我们需要其他解决方案 (°\_°ノ)┐┐ノ

### 3.PPT → HTML

- 使用网络上通用的转换方式将会使ppt变为静态网页，无法保留动画。在github上也没有找到对应的项目
- 我们参考老师提供的解决思路，使用zamzar将keynote文件转为HTML文件，动画有较好保留，但后退功能的支持并不完善，每次只能回到首页。我们阅读转换后的网页源码，发现其使用了一个名为KeynoteDHTMLplayer的技术。js文件可读性较强，但代码量较大，通过深入js脚本运作方式控制HTML播放并不可行。另外网上并无相应文档。通过对比Mac笔记本转换得到的HTML文件，我们猜测该转换是苹果公司提供，**但苹果公司并没有为开发者提供文档或公开其API。**

### 4.使用Google Docs/Onedrive等服务商提供的文档嵌入网页接口

- 文档最齐全、API最丰富、对开发者最友好的google docs在国内无法正常使用，如果要解决这个问题有可能需要付出非常多的精力。
- OneDrive对PPT的显示效果非常好，但是我们并没有找到微软提供的API，我们猜测微软并没有将相应接口开放。如果是这样，文档上传的自动化工作将难以进行
- 除了以上两家公司外，我们还发现了另一家服务提供商zoho。该公司对开发者相当友好，未付费账号在前三个月内可以上传3000个文档，而且该公司提供了较为详尽的开发者文档。但其提供的嵌入链接有时不能很好地工作，我们在调研期间尚未找到解决方法。同时该公司不支持.key格式的文件展示，因此需要提前将其进行转换。

以下是该部分调研的具体内容（对上面的补充）：

#### Google doc

描述：效果极佳，但是国内无法使用

- 放弃

#### zoho

描述：嵌入效果不稳定

- 需要寻找iframe引起无法加载问题的解决方案，我们决定必要时联系zoho的客户服务人员
- 暂定

#### onedrive

描述：个人免费空间有限，播放相关API未找到，考虑到综合效果，暂时放弃

- 如果使用，可能需要借助爬虫
- 备选

#### Zamzar

描述：格式转换效果较好，但是存在以下问题

- 最终获取的HTML文件使用时无法精确控制动画
- 页面转换效果佳，API仅限用于1M以内大小的文件转换
- 放弃

#### CloudeConvert (key转PPTX)

描述：转换较为成功，这一步转换主要是为了兼容zoho的播放类型

- 采用

## 网络通讯

### 广播功能

对于教师端，基本的功能有：PPT、白板、代码编辑器、视频、发言。在以上的调研中，这些问题都已解决。但目前还没有解决广播功能广播功能的实现问题。目前为止考虑过以下两种解决思路：

#### 1. 利用socket传递相应组件的状态，从而使学生端与教师端同步。

- 视频的同步较容易解决
- 代码编辑器的同步并不容易解决。比较直接的思路是在老师的每次变更后直接将整个代码编辑器的内容从教师端传至服务器，然后由服务器广播至学生端
- 聊天室的同步较易解决，对于后面加入的同学，只对其推送加入后发生的聊天内容即可，不需要加载全部的对话记录。这种设计也比较符合主流的聊天室规则
- PPT的同步。考虑学生和老师是同步播放的，而且没有画笔或激光笔的需求，所以教师端只有向前翻页与向后翻页两种选项。当教师端执行相应动作，我们将其广播至学生端即可。但如果考虑中途加入的学生，我们可能需要更为复杂的API，例如**定位到某一页的某一个动画**。而这一点是在目前调研的技术中很难实现的
- 白板同步。目前的技术中有很多都自带广播功能，我们在这里没有作过多调研

#### 2. 利用屏幕录制技术录制网页，通过广播视频使学生端与教师端同步

- 这种思路的优势是不需要处理各个组件复杂的状态，从而不必对各组件提供复杂的API，尤其是摆脱PPT播放对组件的依赖
- 另一个优势是有成熟的技术可以采用，即之前提到过的RecordRTC。该项目在github上有过千的star，提供了详尽的文档，大量的demo与源代码。此外该项目的开源协议是MIT License，使得我们的开发可以自由进行

### 录播功能

录播的思路有三种，分为使用时间轴、分三部分录制视频、录制全部屏幕内容

#### 1. 使用时间轴

该解决方案一旦实现，可能会达到较好的效果，能使得直播过程真实复现，且占用较少的宽带资源。然而考虑到多个组件的复杂交互，难以在短时间内凭借我们的技术水平较完美地实现，因此在客户做出进一步要求之前暂时不考虑。

#### 2. 分三部分录制屏幕

考虑到学生在观看直播的过程中可能出现切换左右两个视屏框区的需要，考虑了分多部分录制的方案，一个附加的效果是：这样做或许可以节约少量的宽带资源。但是之后考虑到多视频窗同时播放难以控制视频完全同步，一旦三个框区的内容出现延迟，将产生较差的用户体验，因此这种实现也被放弃。

#### 3. 录制全屏

这种实现是相对简单直接的，符合多数直播平台的展示效果，同时学生拥有回放、暂停画面的自由。

**最终选型** 调研过程中改变了广播方案的思路：



这种思路大幅度扩展了方案技术的可选范围。

- 例如视频部分，我们原定采用声网的解决方案，该方案能非常简单地部署视频直播;但缺点在于其缺少详尽的文档，具体实现对我们而言类似黑箱。我们甚至没能找到录制视频的接口。
- 再如PPT播放，该部分是调研中耗费时间最多的一部分。如果使用“组件·状态同步”的方式进行广播，势必要研究所采用模块（如zoho/onedrive/google docs/直接使用转换的HTML）的API。但由于种种限制，甚至可能无法获得对应接口。这将为开发带来很多麻烦

而另一方面，元素录制这一方案更贴合于“录播”需求。可以想见在录播模块与直播模块，会有大量可重用的代码，有利于减少工作量。

采用录制元素的方案，也使直播更贴近于传统直播模式。如果今后项目遇到延迟/同步/并发等问题，也易于参考现有平台的优化方案。

**最后对于组件模块，选型如下：**

- 白板：literallycanvas，同时有vrobby/whiteboard备选
- 代码编辑器: ace
- 聊天室：自己实现，thesadboy/Chatroom作为参考
- 视频：RecordRTC相应模块
- PPT：使用zoho公司的服务在网页中嵌入PPT
- 广播：使用RecordRTC录制教师页面，进行广播
- 录制：使用RecordRTC录制教师页面，进行保存