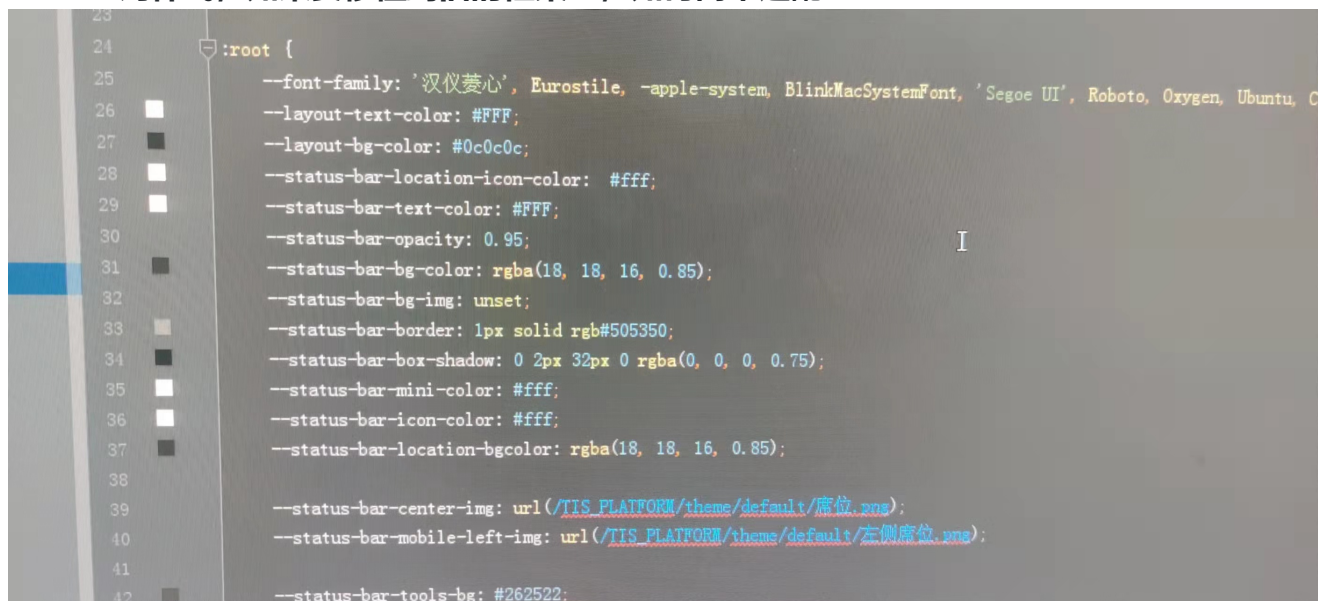


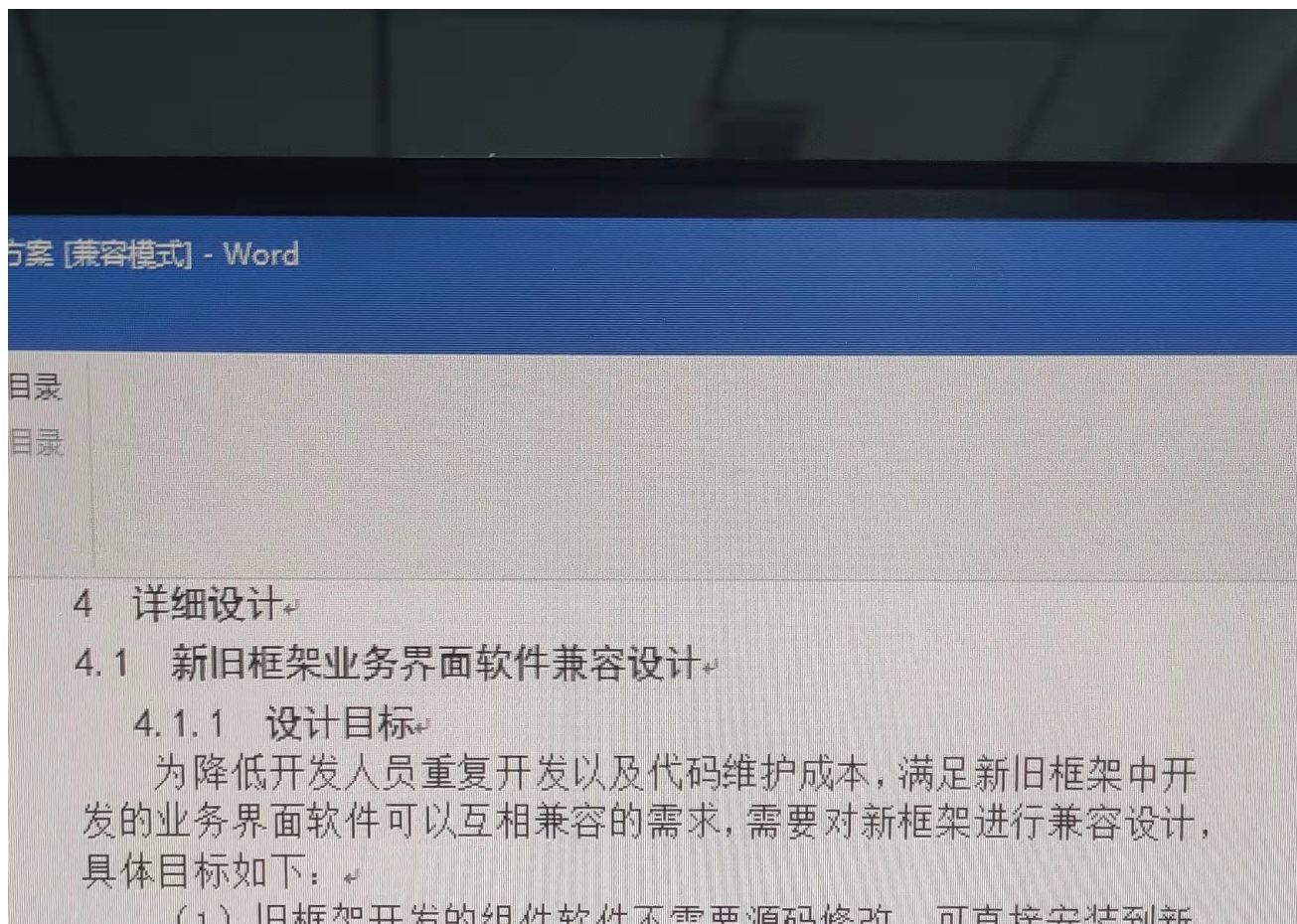
1. 后续移植组件样式适配方案；在新框架中后续开发的组件使用了在新框架中定义的全局样式 root 全局样式，如果要移植到旧的框架上，如何简单适配！



2. 客户担心后续组件移植再出问题

- 1). 需要兼容旧框架暴露出来的接口；
- 2). 新旧框架样式兼容，可自动切换；

主要表现为



- (1) 旧框架开发的组件软件不需要源码修改，可直接安装到新框架中使用；
- (2) 旧框架开发的组件软件可以直接移植到新框架中进行开发和编译；
- (3) 新框架开发的组件软件不需要源码修改，可直接安装到旧框架中使用，主题和旧框架保持一致。

4.1.2 设计方案

自上而下分析，新旧框架组件开发模式一致，理论上讲已具备兼容性；具体分析，为保证良好兼容性，需要主要考虑如下几个方面：

(1) 新旧框架接口兼容：

新框架兼容全部的旧框架接口，内部实现机制可能略有不同，对基于旧框架开发的开发者暴露的接口保持一致。例如在旧框架中提供的 `$EsOpenComponent` 等接口全部支持。组件可以直接使用此接口，而基于新框架开发的组件则可以沿用新框架发布的升级接口。与商业框架的升级模式保持一致，在新增接口或变动接口的同时，对原有接口保持完好的向下兼容。

通过在 `Vue` 原型上挂载对应的接口来实现对旧框架接口的兼容。内部实现细节通过 `tis` 对象上的接口来实现接口一致性。

需兼容接口如下：

接口名称	接口
消息订阅	<code>\$EsDataMsg</code>
消息注销	<code>\$EsData.getInstance().disconnect('test')</code>
消息订阅	<code>\$EsDataMsg</code>
隐藏/显示弹出窗	<code>this.\$EsShowWnd(false)</code>
关闭当前弹窗	<code>this.\$emit('EsCloseWnd')</code>
最小化当前弹窗	<code>this.\$emit('EsMinimumWnd')</code>
最小化恢复当前弹窗	<code>this.\$emit('EsRecoverWnd')</code>
大化当前弹窗	<code>this.\$emit('EsMaxmunWnd')</code>
关闭指定弹窗	<code>this.\$EsCloseOneWnd('弹窗组件名称')</code>
动态设置当前弹窗大小和位置	<code>this.\$emit('changeSizePos', width, height, x, y)</code>
打开指定的组件	<code>this.\$EsOpenComponent(['PlotPanel1', 'test12'])</code>
当前分屏空闲屏幕状态	<code>this.\$EsGetScreenStatus()</code>
打开组件到指定分屏	<code>\$EsSetSplitScreen(screenInfo)</code>
快捷键定制	<code>this.\$EsAddShortcutkey</code>

(2) 新旧样式兼容，可自动切换：

基于旧框架开发的组件，可以在新框架上正常加载，并且样式正常显示。通过 `cdn` 的方式进行资源引入，保持框架与组件在主题切换后能够保持样式不冲突。

存储、处理涉密信息

目录

目录

(2) 新旧样式兼容，可自动切换；✧

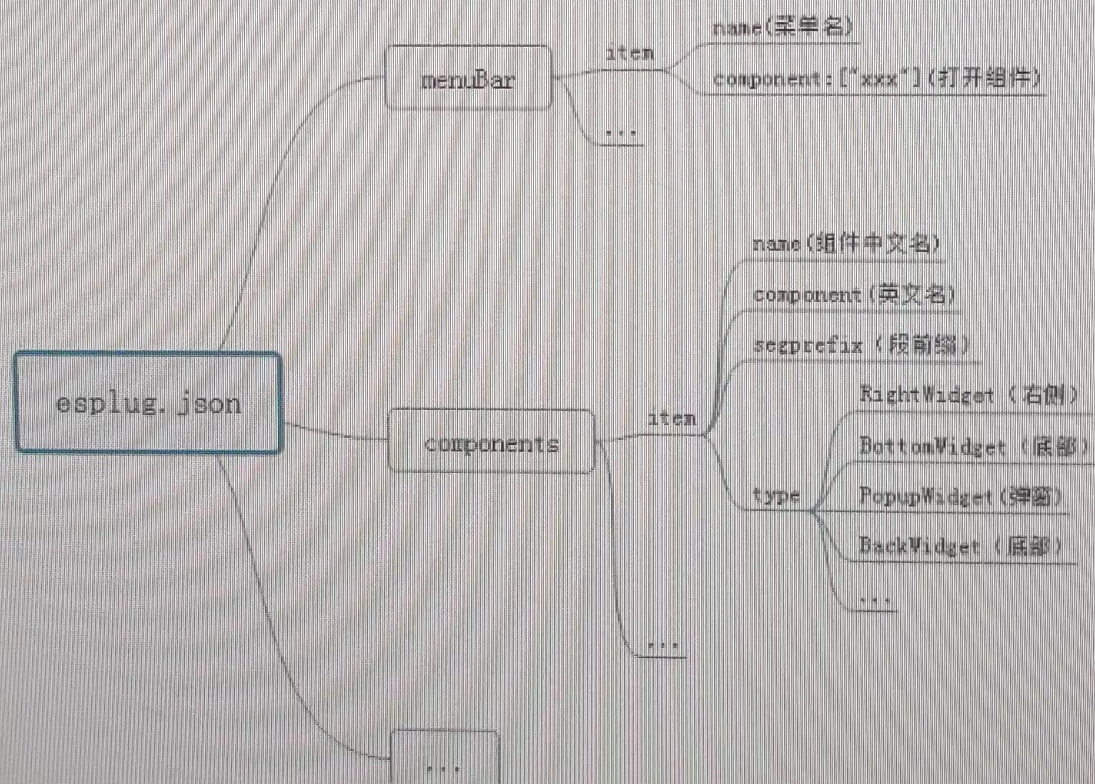
基于旧框架开发的组件，可以在新框架上正常加载，并且样式正常显示。通过 cdn 的方式进行资源引入，保持框架与组件在主题切换后能够保持样式不冲突、不混乱、不污染✧

- 1) Flexible.js 的引入，支持 rem 转化；✧
- 2) Sass 的支持；✧
- 3) Tis-ui 的支持✧
- 4) 组件样式白名单（不被 qiankun 隔离）。✧

(3) 相关配置尽量不需要修改。✧

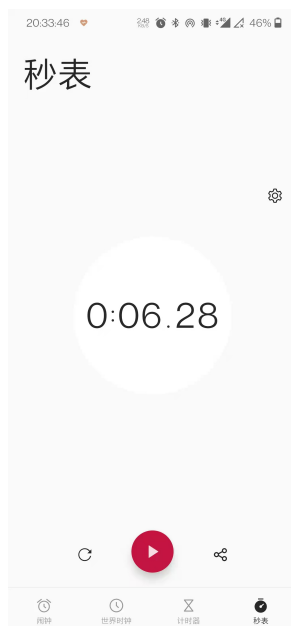
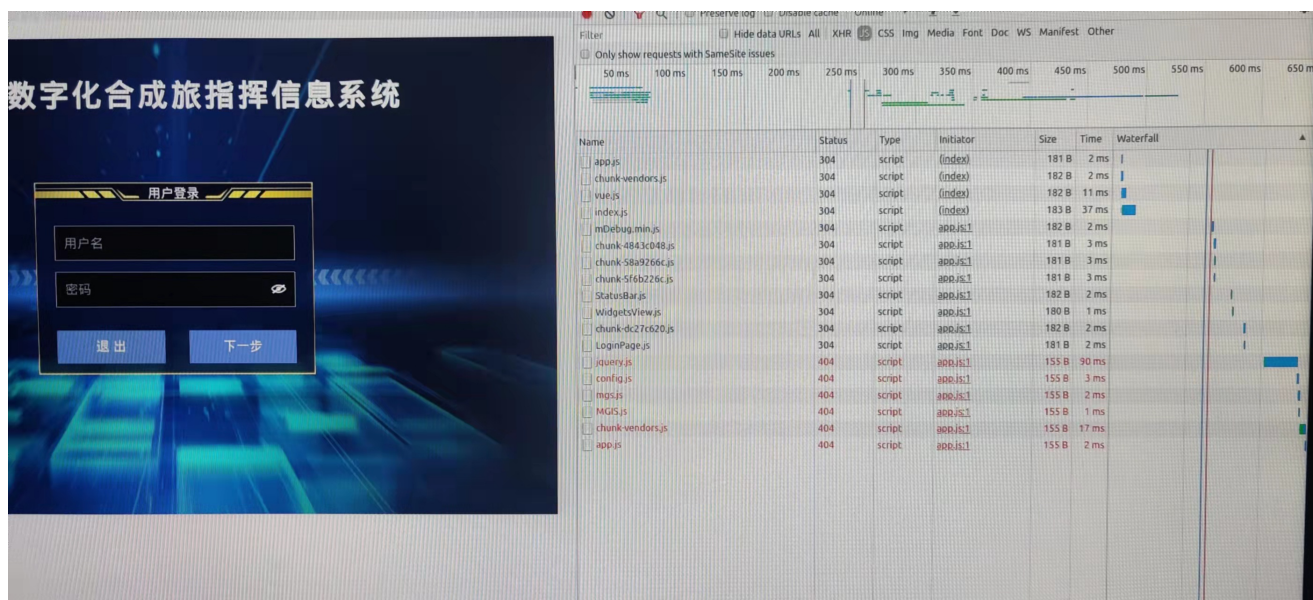
配置文件名称保持一致均为 esplug.json，配置文件内部字段保持一致，原先的 menuBar、components 字段均保留。并且组件内部的 type 属性与原先保持一致，例如弹窗组件为 PopupWidget、底部组件为 BottomWidget、右侧组件为 RightWidget 等，尽量保持字段一致，新增特性在原有结构上进行拓展。✧

下图中出现的 key 均能覆盖旧框架配置文件中出现的配置项。



本条可分条描述需要由软件产品完成的所有功能、工作模式、容错要求、特殊要求(如对某些意外的适应能力)及应急措施和可扩展要求。如适用，应标识清楚原有功能、改进功能、新增功能。
例如：

3. 新框架在平板运行卡顿的问题，移除所有组件，单独应用，加载时间6秒左右



4. 客户需要我们提供一份自动化测试用例