

在 Chrome、Firefox 等高版本浏览器中

实现在线编辑、审阅 Office 文档新方案

一、背景

在 IE、Chrome 45 及 Firefox 53 以下版本的浏览器网页中，可直接或间接调用微软免费开源的 ActiveX 控件 DsoFramer 即可嵌入本地安装的 Office 软件实现在线编辑或审阅 Office 文档。然而好景不长，从 2015 年开始 Chrome、Firefox 等新生代浏览器先后取消了对 NPAPI 插件的支持，在这些浏览器的后续版本中如何实现就成为了技术难题。IE 浏览器又因各种问题导致其不断被用户抛弃，其市场份额已降到前端主流框架直接放弃兼容的地步。而随着 Windows 系统及 Office 软件的不断升级，DsoFramer 的稳定性和兼容性问题越来越多，不支持 64 位浏览器不说，也不支持金山 WPS 及开源的 LibreOffice 等办公软件，要命的是 2008 年以来微软再无升级版发布，导致其用于商业用途具有很大风险，除非自己有研发能力解决这些问题。虽然随后有点聚 WebOffice 控件模仿 DsoFramer 接力举起免费的大旗，为其收费的电子印章产品引流用户，然而依然无法克服 ActiveX 控件固有的缺陷，只能在 32 位的 IE 浏览器中使用不说，往往还需要修改终端电脑 IE 的默认安全设置，实施成本高、加载运行时弹警告导致用户体验差，这不现在点聚 WebOffice 的官方网页和交流论坛也都没了踪影，不再继续维护升级，因为没开放源代码，第三方也无法修改其缺陷，其用户再次被抛弃了。

随着 SAAS 模式的兴起，纯网页实现的在线编辑或审阅 Office 文档成为了大家重点发力的方向，包括微软、金山、谷歌、阿里巴巴、腾讯、字节跳动等巨头在内的诸多公司纷纷发布了自己的在线 Office 服务并不断拓展其功能，对传统 OA 厂家都造成了一定威胁。经过多年的发展，虽然这些在线文档服务在多人协作分享、跨平台、使用成本及便捷性、维护升级等方面具有优势，但存在的问题也着实不少，遇到系统崩溃时用户就得抓狂。相对于桌面版 Office 来说，在线 Office 服务存在一些功能缺失，需网络随时在线方可用，并且和桌面版制作的文档格式不兼容、内容排版错误或表格公式及函数无法使用，文档内容复杂或表格记录大时打开缓慢甚至直接卡死，多文档同时编辑或嵌套内容难，私密性、安全性不够，在线服务器压力大，最关键的是其开放能力往往不足，导致很难整合到 OA、CRM、ERP、企业网盘等系统作为子模块使用。另外，不少企业由于历史的原因都已购置了桌面版 Office 的长期使用授权，也整合到了自己的各种业务系统中使用，也就很难再每年付费切换到在线 Office 版使用，毕竟还面临要改变用户使用习惯的难题。云端在线协作编辑一时爽，调整格式真酸爽，所以最好的解决办法还是在原来的桌面版 Office 软件之上，通过技术手段让其可顺畅运行在 Chrome、Firefox、Edge、360、Opera、QQ 等主流版本浏览器中。

二、现有方案

实现微软 Office、金山 WPS 等桌面软件嵌入浏览器网页实现在线编辑或审阅 doc、docx、xls、xlsx、ppt、pptx、wps、et 等办公文档，大体上有如下几个方案：

1、 浏览器插件方案

此方案目前只适用于 IE 浏览器，网页中直接运行 ActiveX 控件调用桌面 Office 软件的自动化接口来实现。前述章节提到的免费 DsoFramer 及点聚 WebOffice 控件如此，收费的北京卓正 PageOffice、广州华尔太 WebOffice 控件也是如此。而在 Chrome 45、Firefox 53 以下版本的浏览器中，收费的重庆软航 NTKO Office 文档控件、江西金格 iWebOffice 智能文档控件在 NPAPI 插件中再调用 ActiveX 控件来请求桌面 Office 软件的自动化接口。这些商业收费控件，自然也为满足 OA 等系统的需求做了一些兼容性处理，如兼容更高版本的 Windows 系统、微软 Office、金山 WPS 及 64 位浏览器，功能扩展如智能填报内容，文档加密、版本管理、拆分合并，痕迹保留等。2015 年开始在非 IE 浏览器取消 NPAPI 插件支持后的技术突围过程中，江西金格 iWebOffice 智能文档控件表现相对突出，其基于 Chrome 浏览器 PPAPI 可信插件技术发布的新版可用于兼容 44 以上版本，并通过在系统驱动层拦截并修改浏览器启动参数解决了 PPAPI 可信插件必须通过命令行启动浏览器才支持的技术难题，因此其销售价格非常昂贵。然而采用底层的系统驱动技术方案可能导致操作系统蓝屏死机不说，微软对系统驱动程序管理越来越严格，每年的数字签名证书费用也是一笔不小的开支，况且不少单位的等级保护要求因为安全原因是不允许用系统服务或驱动程序的。此外，PPAPI 插件并未得到 Firefox 等浏览器的支持，目前看只是一个过渡技术方案，这不 Chrome 在 2021 年初已经取消了对 Flash Player PPAPI 插件的支持，也宣布了 2021 年 6 月终止对 NaCl、PNaCl 和 PPAPI 插件的支持，这就彻底宣告了浏览器插件模式即将寿终正寝。

2、 特定浏览器方案

为了应对 Chrome、Firefox 等浏览器关闭插件接口后的技术挑战，又希望保证原有业务系统的正常运行，一些实力较弱的公司只好推荐客户继续使用面临安全漏洞威胁的低版本 Chrome 或 Firefox 浏览器，或者推荐使用双内核的国产 360 浏览器并要求切换到 IE 内核使用。实力稍强一点的公司基于 Chromium 的低版本内核开发定制版浏览器来确保 NPAPI 插件可以继续使用，用户无法享受高版本浏览器的功能不说，低版本内核存在的安全漏洞和程序 BUG 也无法得到修复。然而企业信息化系统往往是非常复杂的，对浏览器的要求比较高，很可能不支持这些特定版的浏览器，所以不能因为 Office 文档控件只能在这些特定浏览器上运行而导致 OA 等系统也跟着受到限制。

3、 外接程序方案

重庆软航 NTKO Office 文档控件、北京卓正 PageOffice、广州华尔太 WebOffice 控件等在 NPAPI 插件禁用后，纷纷采用浏览器外部协议(URL Protocol)方式来启动独立的 EXE 外接程序，而这个独立程序会内嵌一个 IE 内核用来支持 ActiveX 控件的加载运行，也号称兼容 Chrome 和 Firefox 等高版本

浏览器运行了。这种技术实现导致在网页中每次启动 Office 控件时，都会弹出一个让最终用户不胜其烦的对话框来确定，如下图所示：

要打开 WebOfficeProtocol 吗？

<http://www.officectl.com> 想打开此应用。

打开 WebOfficeProtocol

取消

要打开 POBrowse 应用程序 吗？

<http://www.pageoffice.cn:8080> 想打开此应用。

打开 POBrowse 应用程序

取消

用户使用体验很差，这些弹出的独立进程窗口和浏览器原来的网页窗口是完全分离的，还涉及到两个窗口之间的数据传递和控制问题，增加了系统复杂性。如果用户提前启动了 Word、Excel 等程序使用，往往还会导致其控件在打开文档时异常，对商业用途的客户来说这种技术方案基本上是不可接受的。

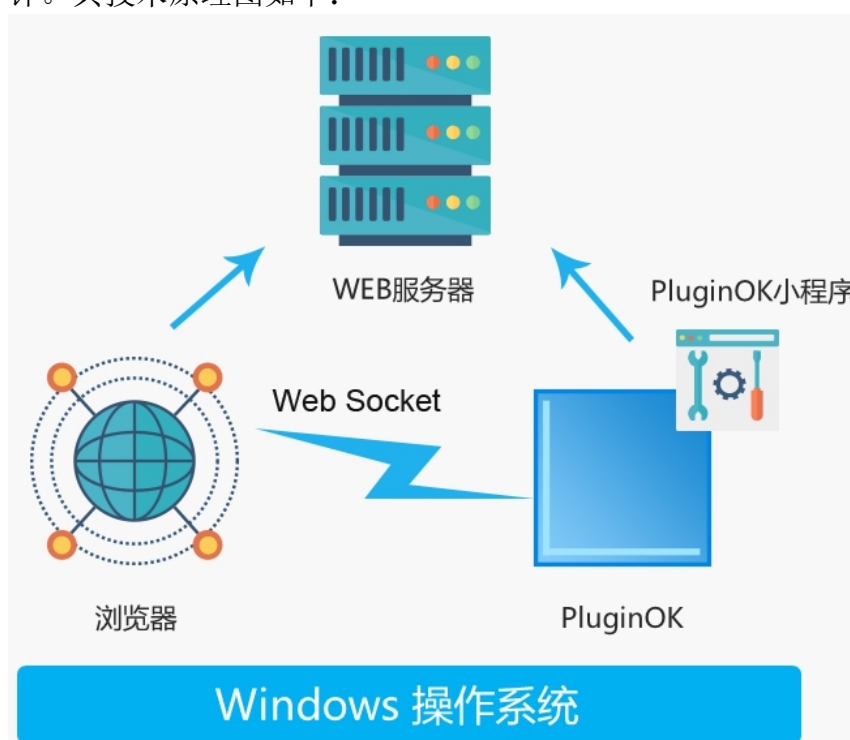
4、 双内核方案

此方案主要是依托 Chrome 等浏览器上的扩展程序 IETab 来实现，商业用途是需要付费使用的。其原理是通过在 Chrome 标签页中加载一个 IE 内核渲染的整个网页进行覆盖显示，这个网页再调用比如 DsoFramer 控件实现 Office 文档的编辑。由于 Chrome 的扩展程序必须通过谷歌或微软应用商店进行安装，导致此方案存在大规模自主可控部署难问题。而且 IETab 实现的效果是对整个网页生效，无法在 Chrome 内核网页中再实现局部加载 IE 内核网页来显示，况且 IE 对运行和下载 ActiveX 控件常弹出警告，用户体验很差，即使通过修改 IE 的默认安全设置来减少弹窗，然而这个修改设置过程也需要专业的技术人员才能完成，大幅度增加了实施成本。

三、改进方案

通过上述总结的现有技术方案可以看出，想要在当前主流版本的 Chrome、Firefox、Edge 等浏览器网页中内嵌桌面 Office 软件实现文档或表格在线编辑等功能，核心就在于如何在各浏览器中实现一个统一的不依赖浏览器自身扩展技术的外接系统，同时必须对各品牌及各版本的浏览器有比较好的兼容能力才具有较大的实用价值。所以改进方案就是在网页中指定位置和大小，模拟实现一个内嵌到网页中显示的窗口，在这个窗口中再调用桌面 Office 软件的自动化接口实现 doc、xls、ppt 等文档的操作。前端还必须可对这个窗口进行实时控制，而且窗口必须跟随浏览器的移动和缩放、网页滚动、标签页切换、关闭等操作进行自动联动。

这个窗口的宿主进程同时提供 Web Socket 的服务端和 JSON 打包命令的解析执行模块，前端就可以通过 Web Socket 连接后发送 JSON 打包的控制命令实现控制此窗口的动作。此方案可以说是上述外接程序方案的升级版，关键差异在于此方案可实现内嵌 Word、Excel、PowerPoint、WPS 文字、WPS 表格等程序窗口到网页指定区域运行的效果，而且抛弃了通过 IE 内核来加载 ActiveX 控件的方案，解决了用户体验差、加载缓慢和内存占用高的问题。另外启动这个外接程序是通过 Web Socket 连接实现的，也解决了每次启动都会弹提示的烦恼问题，还有就是提供了类似 ActiveX 控件的自动升级方案，可在网页中实现静默自动升级，并额外增加了调用验证机制确保外接程序的安全启动。此技术方案在 2019 年申请了软件发明专利保护，且已有实现的成熟软件产品发布，这就是跨浏览器的原生小程序软件系统—PluginOK 中间件(<https://codechina.csdn.net/zorrosoft/pluginok>)，自发布以来成功应用到电力、医疗、交通、安防、保险、工业设计等行业客户，获得了一致好评。其技术原理图如下：



微软 Office 网页小程序(<http://zorrosoft.com/Files/msofficeapplet.txt>)和金山 WPS 网页小程序(<http://zorrosoft.com/Files/WPSOfficeapplet.txt>)都提供了丰富的二次开发接口，并可根据自己需要进行二次开发。由于这个改进方案提供的 Office 网页控件接口和老方案的差异比较大，所以比较适合应用在新项目和新产品的中。那么对于哪些已经调用了微软 DsoFramer，点聚 WebOffice、NTKO Office 等文档控件的信息化系统，是否有代价比较小的改造方案的升级方案呢？答案自然是肯定的，可以借助 PluginOK 中间件之上的 IE 控件小程序，动态实现双内核的运行效果，类似现有方案中的双内核方式，但解决了分发和在线升级的难题，同时可实现在 Chrome 网页中局部加载显示的效果，用户体验也得到了大幅度提升。无需改变这些文档控件的调用方式，只需要在网页中判断如果遇到的是非 IE 浏览器时，请求启动 IE 控件小程序加载即可。这里是实现的效果展示视频：<https://www.bilibili.com/video/BV1zp4y1n7R6/>

四、总结

一个好的技术实施方案，首先是要满足客户的刚性需求，其次是尽量降低采购、开发、实施及维护的总成本，再次是要有良好的兼容性和稳定性，最后需尽量确保技术方案不会因为浏览器的升级而失效。本文基于当前最新的技术信息和实践经验，给大家提供了 PluginOK 中间件搭配 Office 网页小程序这样一个稳定可靠、体验好、兼容性佳的桌面 Office 文档控件技术方案，尤其适合在内网处理复杂文档和数据量大的表格场景，以供大家技术选型参考。如还有技术问题或系统开通权限体验，可以加微信 ZorroSoft 沟通。