**关于文档管理和即时通讯系统的立项报告**

## 背景

计划将文档管理和即时通讯做成两个独立的系统。

文档管理模块参考网盘，除了提供网盘的基本功能外，还会提供版本控制、网络分享、秒传、协同开发等功能。

及时通信模块参考网页版微信，提供基本文字交流、组群管理、定时通知等。

每个模块都能独立工作。也能合并为一个系统服务，提供一些特殊功能.(详见技术实现)

项目设想主要来源于目前的几个问题：云桌面属于在系统层面提供服务，无论是VIO、VDI、IDV提供的都是一整个操作系统。UAM正好可以弥补这一点，UAM可以提供应用级别的服务，相对于云桌面来说更加轻量级。更加适合远程办公、云应用等场景。UAM的侧重点是应用，因此我们还需要另外一套系统来提供文件管理服务，这样才能算一套完整的产品解决方案。

武大版UAM虽然可以通过FTP管理文件，但是不具备版本控制、文件分享等功能。稍微有点规模的项目几乎都要多人协同开发，FTP肯定不能胜任的。

龙青版UAM使用nextCloud承担文件管理，确实是一个完美的解决方案。但是nextCloud太过庞大，我和陈明聊个这个问题，重构它的可能性几乎为零，要完全掌控它不是一朝一夕的事。

刚好现在有需求，希望我们开发一套文件管理和即时通讯系统。因此我建议在nextCloud之外再写一套我们自己的文件管理系统。要完成这个系统，知识储备是完全没有问题的，理论模型测试已经完成。

## 思路

1. 在文件服务器上安装Git，通过Git来实现文件管理。我们在写一个java服务，将文件呈现在web页面上。当用户在页面操作文件时，我们将操作转换为对应的Git命令。这样用户根本感受不到Git的存在。

2. GitLib提供的fork、push、marge等功能稍微修改一下就能变为文件分享、协同开发等功能。

3. Java生态中的Jodconverter、lucene、ikanalyzer 、poi、jgit等，配合OpenOffice能快速实现全文检索、在线预览、在线编辑。这些基本都是以前研究过但是一直没机会用上的技术。

4. 自己维护一张文件Hash表，和自己开发的即时通讯系统配合，实现文件秒传。

## 难点

1. 文档管理模块中文件锁我大概知道原理，但是没做过具体编码，有点心虚，问题应该不大。第一期可以只做版本控制，这样就不会有冲突存在了。
2. 即时通讯模块，这个可大可小，难度要根据规模和功能决定。但是如果要做我就想把它做好，毕竟公司现在没有这个方向的产品，模块化后可以和任何产品搭配使用。
3. 前端

图1：系统拓扑图

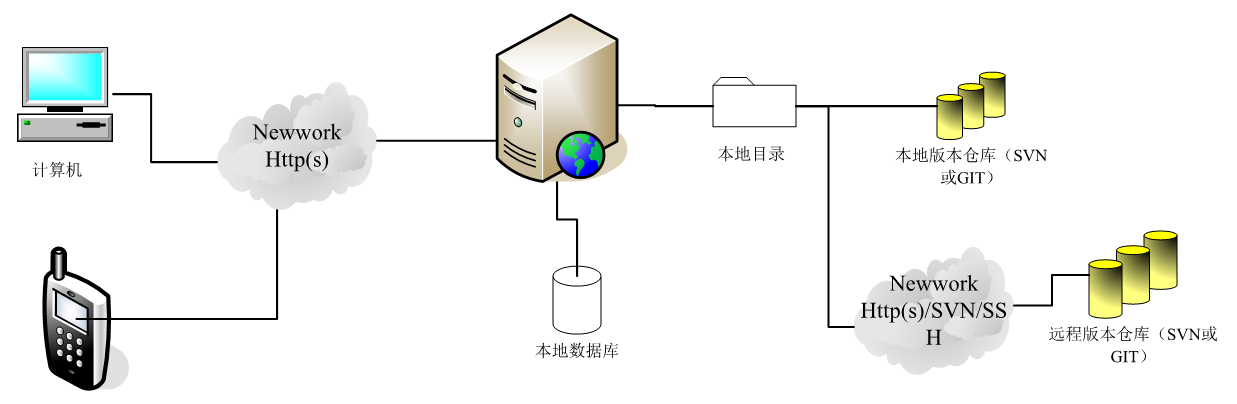


图2：系统操作流程图

