



# 开源的企业文件云存储平台

潘凌涛 [freeplant@gmail.com](mailto:freeplant@gmail.com)

[@freeplant](#)



# 内容提纲

- 产品介绍, Seafile 是什么
- 项目理由, 为什么要做 Seafile
- 工作原理, Seafile是怎么工作的
- 经验分享, 从 Seafile项目从学到的经验





# Seafile 是什么

**Seafile** 是一个面向企业、团队的云盘，解决文件管理和共享的问题。

在此基础上，**Seafile** 提供群组协作的功能，形成一个**以文档为中心的协作平台**。



Seafile

我的页面

群组

机构

个人页面



我拥有的 共享给我的 群组

资料库

★ 星标文件

Activities

个人维基

消息

客户端

通讯录

名称

描述



seafile-dev

Seafile 开发共享资料



quick

quick log



sale

Sales



spec

seafile, seahub 开发规



个人维基

个人维基页面

资料库和个人页面



Seafile

我的页面

群组

机构

 海文互知



 资料库

 讨论

 维基

 成员

 管理



添加一个讨论...



**freeplant** 4 小时前

@林帅, 客户端帮助文档内容:

## 资料库

## 群组页面和讨论



Seafile

我的页面

群组

机构

机构

资料库

资料库

群组

成员

名称

描述



test\_readonly\_rep  
o

test read only repo

机构页面



Seafile

[我的页面](#)

[群组](#)

[机构](#)

## seafile-dev

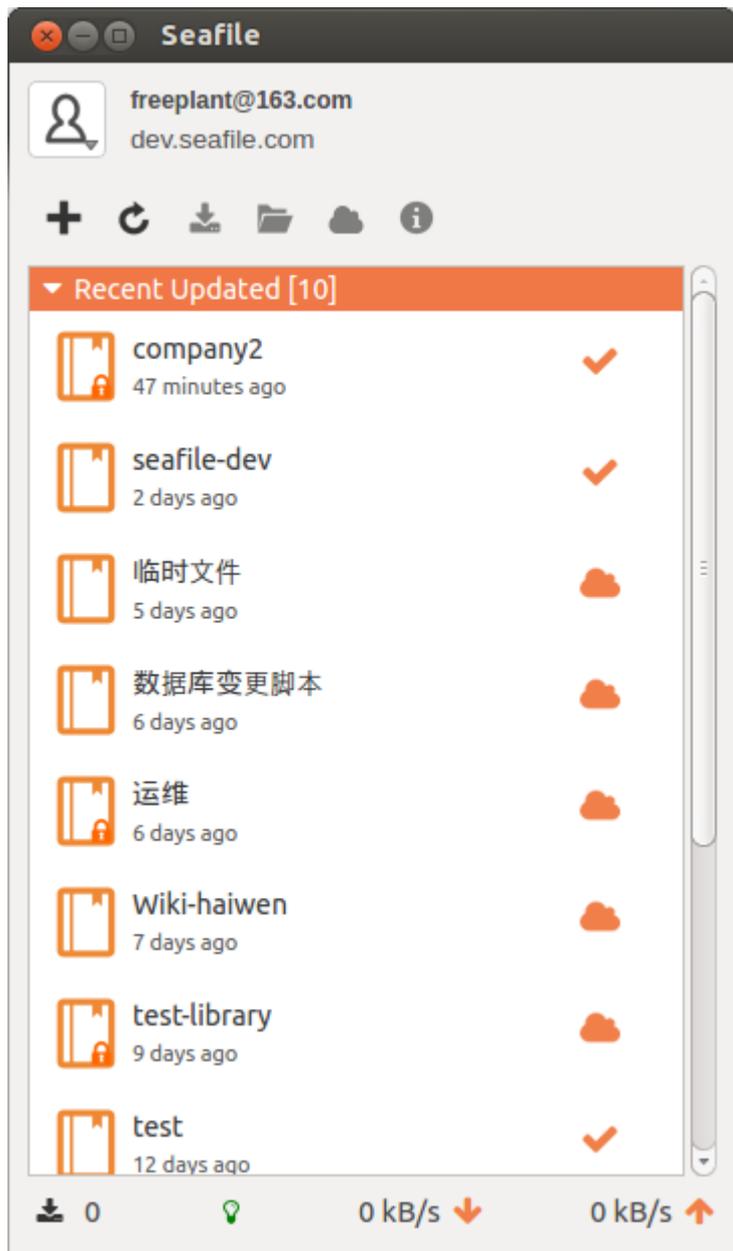
Seafile 开发共享资料 | 大小: 158.4 MB

添加了 "rotate-1-win.ico"以及另外1个文件. [详情](#) |  [freeplant](#) 1 小时前 | [历史](#)

当前路径: seafile-dev

<input type="checkbox"/>	名称 ▾	大小	更新时间 ▾
<input type="checkbox"/>	 <a href="#">bizspark模板</a>		7 天前
<input type="checkbox"/>	 <a href="#">ceph-config</a>		10 天前
<input type="checkbox"/>	 <a href="#">docs</a>		10 天前

# 资料库页面



# 客户端



# 为什么要做 Seafile

文件管理和共享和企业/团队的最基本的需求。

**Seafile** 就是要做一个高质量的开源的企业云存储平台。同时加入协作功能，形成一个创新的产品。



# Seafile 特色

- 群组协作功能
- 端到端加密
- 小巧/高性能，可部署于非常多的平台



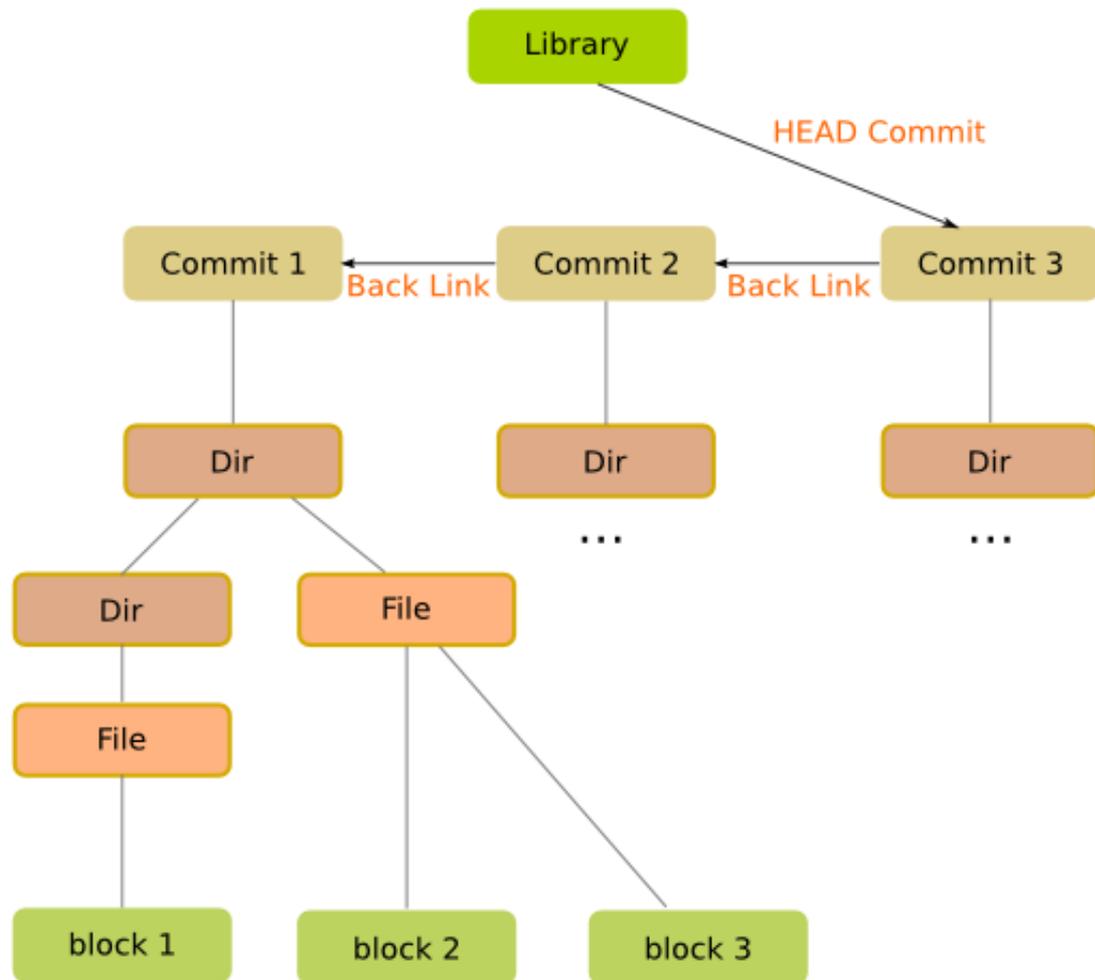
10000 次/月的服务器软件下载

1300+ Github Star



# Seafile 怎么工作 - 资数据库模型

一个资料库是一个带历史的去重的文件系统





# Seafile 怎么工作 – 数据存储

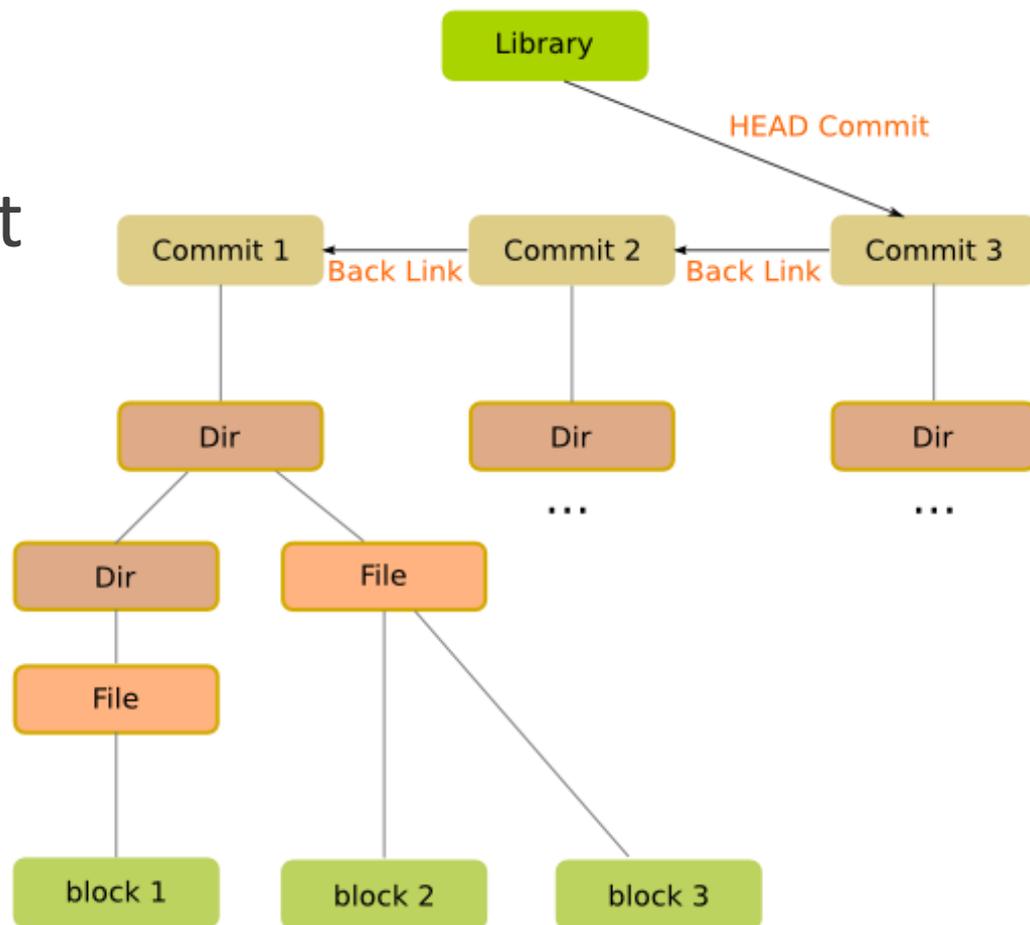
文件对象，目录对象，文件块存在操作系统的文件系统上，或者 S3/Swift/Ceph 这样的外部存储中。

结构化数据存在 MySQL 或者 SQLite。



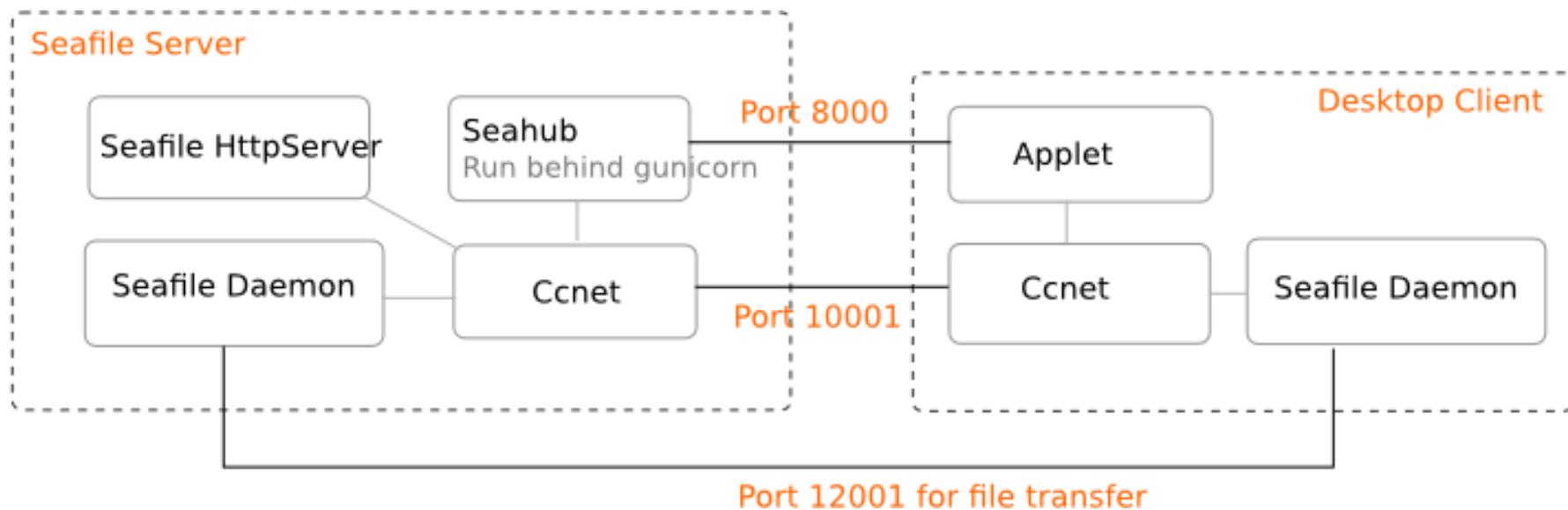
# 同步算法

1. 定期到服务器查看最新的 Commit
2. 下载文件和元信息，和本地内容归并
3. 本地有修改，生成新的 Commit，并增量上传到服务器





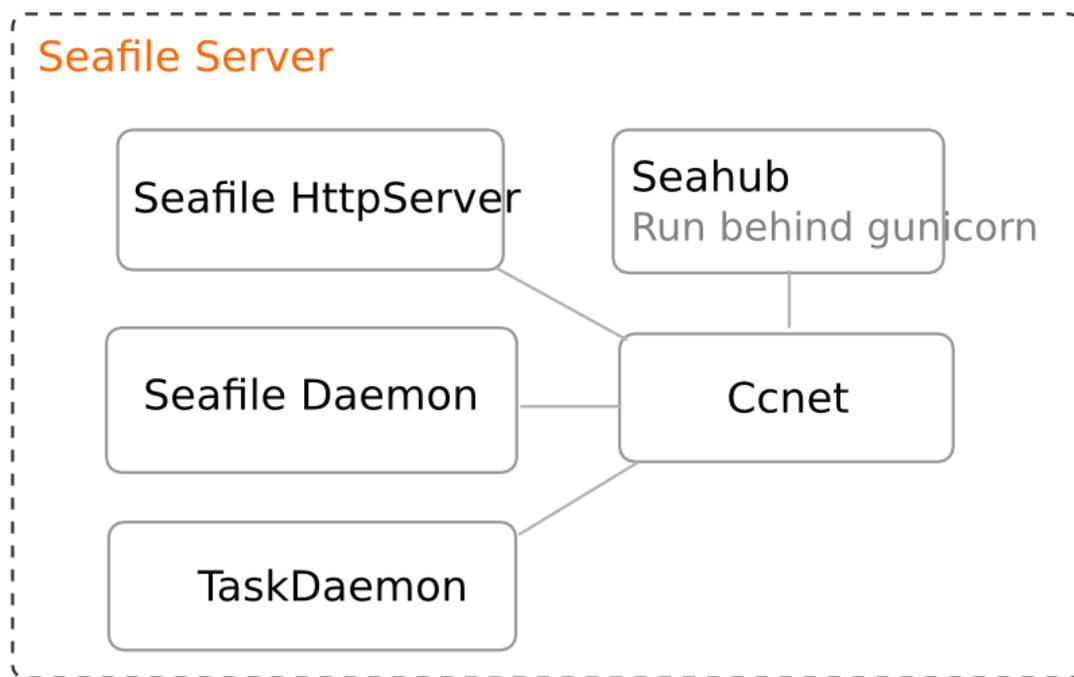
# Seafile 怎么工作 – 架构



Ccnet 负责通信，Seahub 网站界面，HttpServer 模块负责网页端的文件上传下载。Seafile 负责数据管理和同步。



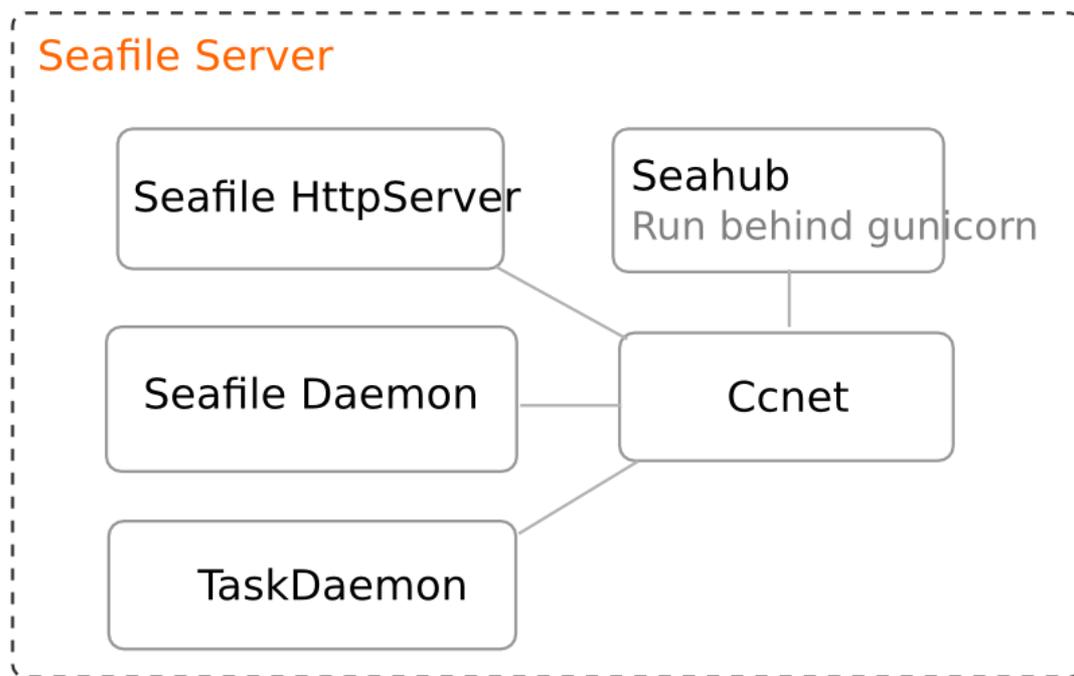
# 实现语言选择



Ccnet/Seafile/HttpServer 用 C 语言实现，使用 libevent 事件循环+线程池处理异步任务的架构。Seahub 使用 Python Django 框架。TaskDaemon 处理后台发邮件等任务，用 Python 语言。



# 实现语言选择



进程间通信使用自己设计的 RPC。用 python 和 c 都上实现了一遍 RPC Server/Client。用 json 来做数据交换的格式, 以支持跨语言的数据交换。



# Seafire 项目经验

- 产品设计
- 架构设计
- 实现技术选择



# 产品设计

在确定功能核心的基础上，  
列出最小功能集合，  
把易用性和符合大众习惯放在第一位  
来进行设计



# 产品设计

Seafile 的核心是团队文件共享，  
群组和**多资料库同步**是最基本的功能

**多资料库同步**造成了一定的入门难度，  
但是我们认为**是必要的**。



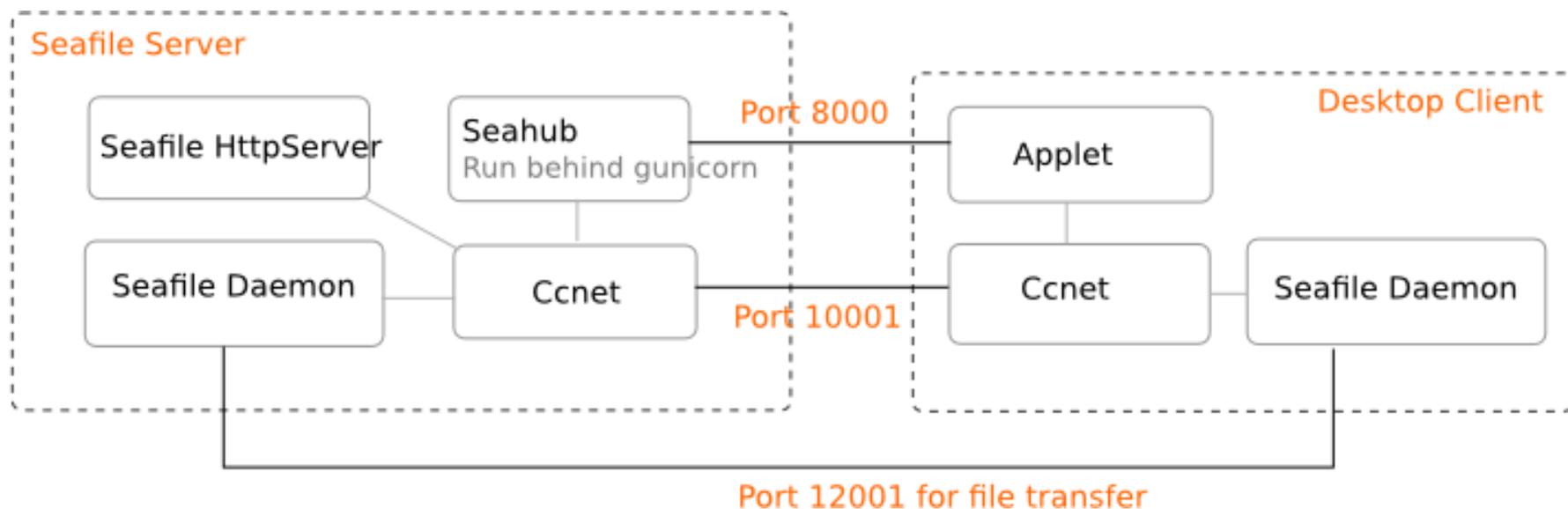
# 产品设计



原来为了方便，客户端的界面是用 Python 来实现了一个本地的 Web 界面。造成了不少混淆，这是不符合大众习惯的反面例子。后来我们改成了 Qt。



# 架构设计



客户端不应该采用复杂的多进程架构



# 通信协议选择

程序内部的多组件(可能分布在多个机器上)通信，**效率和便捷性**是第一位的，在 **TCP** 协议上直接设计一套 **RPC** 是最好的。

多程序间以及互联网的通信，**互操作性**才是最重要的。**HTTP**协议才是互联网标准的通信协议。



# 通信协议选择 – HTTP 协议

**HTTP** 的坏处是比较复杂，它不是一个为传输文件而设计的协议。在处理文件传输的时候，有太多的小陷阱。好处是一旦使用了 **Http** 协议，你的应用就能使用上 **Proxy, Cache, Load Balance** 这些基础服务。



# 实现语言选择

客户端：C/C++ 是最好，最成熟的跨平台语言

服务器：Python 写 Web 和脚本都是非常好的选择。追求性能的地方要用 C。



# 总结

难点：无论是服务器还是客户端都要运行在多个平台上。

功能选择还有架构设计上都要保持简单。使用最成熟的、支持最广泛的实现技术。



# 如果重新编写

客户端使用单进程，用 QT 框架, C++ 语言。

客户端和服务的通信使用 HTTP 协议。

服务器端保持目前的架构，使用 Python + C。

关于 Seafile 更多信息

<http://seafile.com>