

基于云存储的分布式文件系统

同步与加密

PRESENTED BY WQ

指导老师：邢 凯

组员：王若晖 朱一铭 钟立 韦清

目录

『CONTENT』

- ▷ **第一部分** 『课题背景』
- ▷ **第二部分** 『设计思路』
- ▷ **第三部分** 『实现方法』
- ▷ **第四部分** 『创新点』
- ▷ **第五部分** 『成果展示』

第一部分

『课题背景』



课题背景

▷ 云存储



1

课题背景

▷ 立项依据



速度



隐私



可用性



成本

1

课题背景

▷ 速度

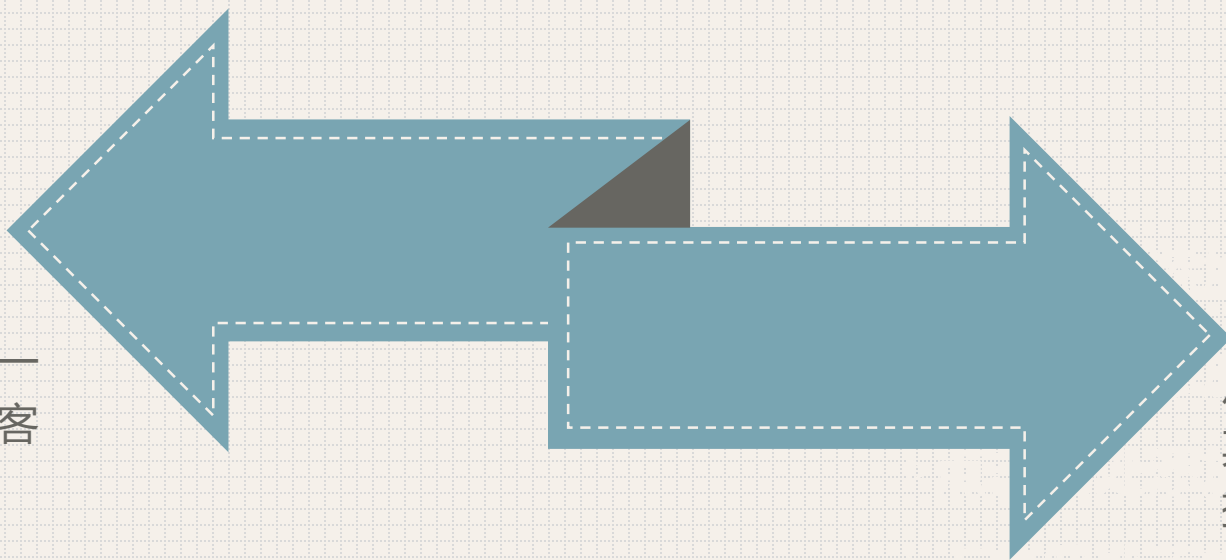
现状

国内网盘存在限速情况。

国外网盘由于网络问题在一般情况下也难以达到较为客观的速度。

目标

从多个服务端获取数据，则其速度将比单服务器更容易提升。



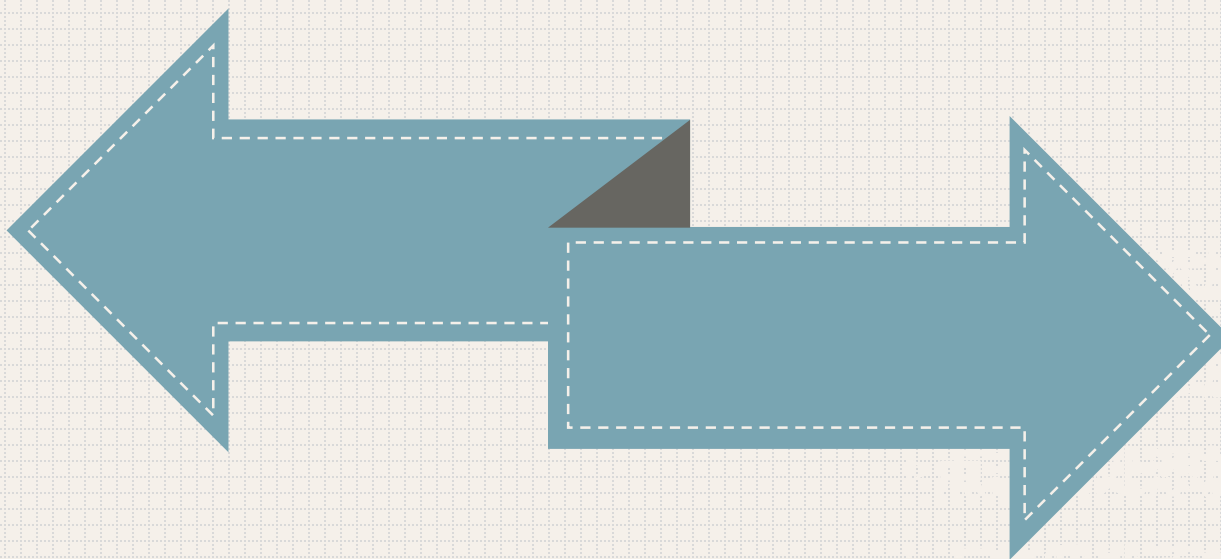
1

课题背景

▷ 隐私

现状

提供存储的公司事实上拥有你所有的存储内容，无论那是否涉及个人隐私，网盘公司都保留查看的权利。



目标

数据存于多个服务端，使得每个网盘提供商都无法获得完整的数据。

上传前加密。

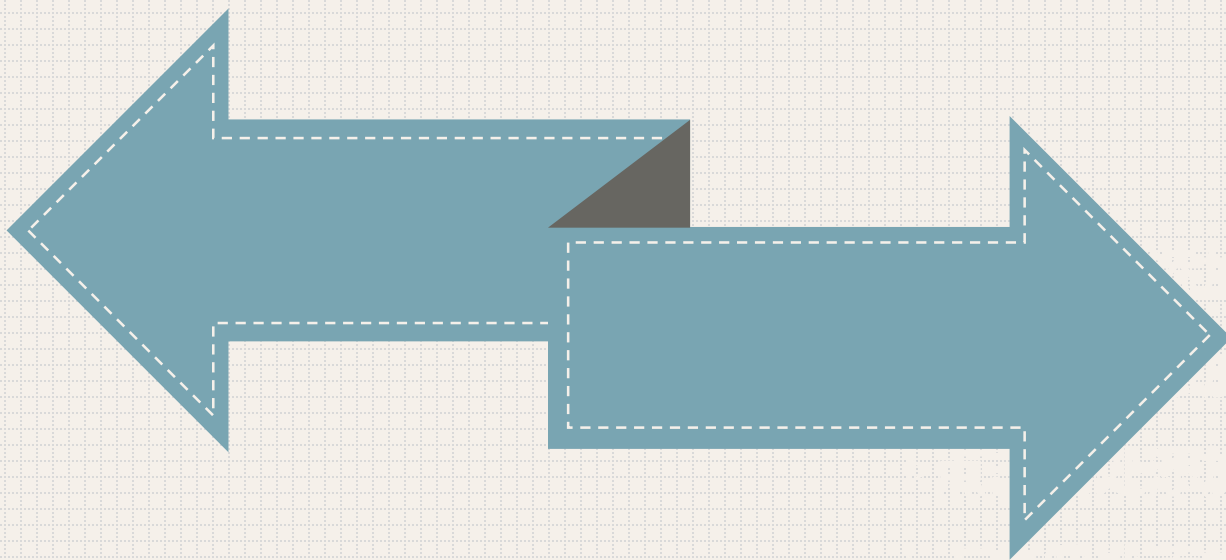
1

课题背景

▷ 可用性

现状

虽然互联网公司通常有一定程度的保证，但这个保证并不是硬性的。通常互联网公司可以随时停止服务。



目标

使用冗余存储和分布式存储以保证数据完整性和可用性。

1

课题背景

▸ 成本

现状

网盘免费存储空间有限。

对非会员设置限速。

非会员无法上传、下载大文件

目标

通过分布式存储，有效利用
多余空间和闲置带宽。

1

课题背景

►相关工作：工业界的传统的云存储加密方案

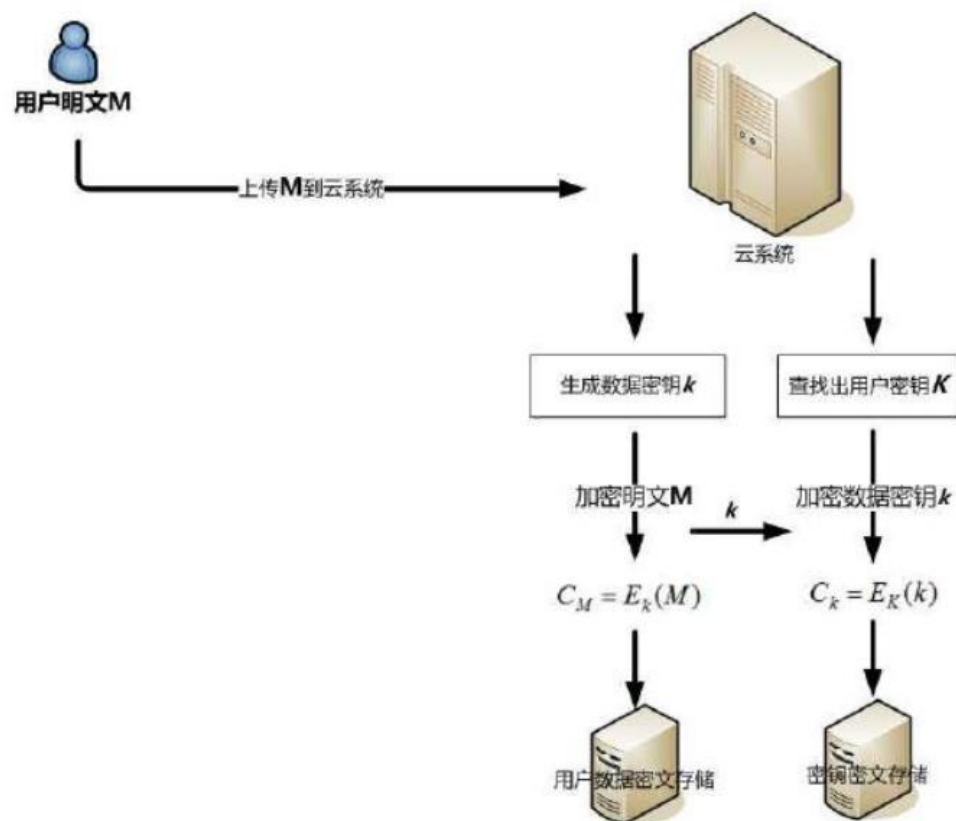


图 1 传统分布式存储加密方法

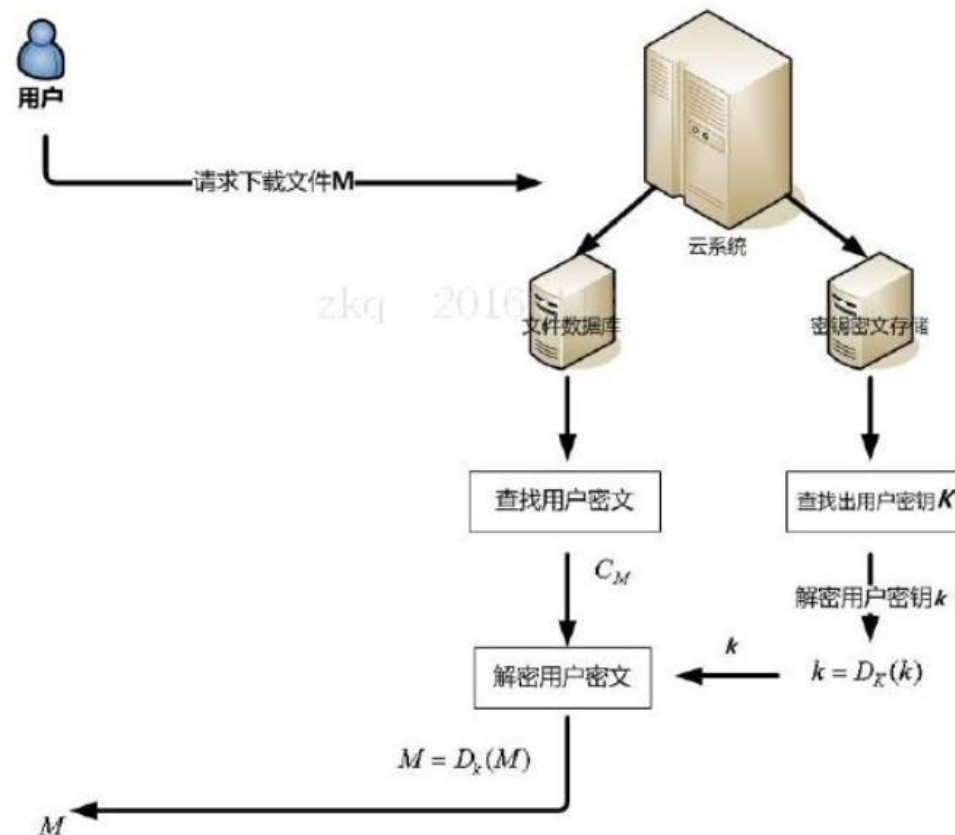


图 2 传统分布式存储解密方法

1 课题背景

▷ 相关工作：部分网盘供应商提供的文件夹自动同步功能

1 mac版百度网盘同步盘能干什么？

百度网盘同步盘的基本功能是自动同步。您不必手动进行上传下载，只需在登录后选择需要同步的文件夹即可，同步盘将对这个文件夹里的文件和云端文件进行自动同步。您不必在每次更新文件后都再把文件上传一次，只要开着同步盘，这些操作都会被同步盘自动完成，省时又省心。

2 百度网盘同步盘的同步文件夹是怎么回事？如何设置？

在安装完百度网盘同步盘后，会在您电脑中自动创建一个同步文件夹，同步盘会实时监测这个文件夹里的文件，一旦发现用户对这此文件进行了操作，会马上对网络硬盘中也进行同样的操作，保持网络硬盘中的文件与本地文件是一致的。同样，如果您对网络硬盘中的文件进行了操作，同步盘也会马上对电脑上同步文件夹中的文件进行同样的操作。

也就是说，在本地新建一个文件，网络硬盘中就自动新建一个完全一样的文件；本地修改、删除，网络硬盘中就自动修改、删除。反之亦然，在网络硬盘中所做的文件操作，也会被同步盘自动同步到本地电脑上。

目前版本暂不支持用户设置同步文件夹的位置，之后的版本会增加此功能，敬请期待~

3 我怎样进行文件同步？

您只需通将文件拖入百度网盘同步文件夹即可实现同步到云端。

第二部分

『设计思路』

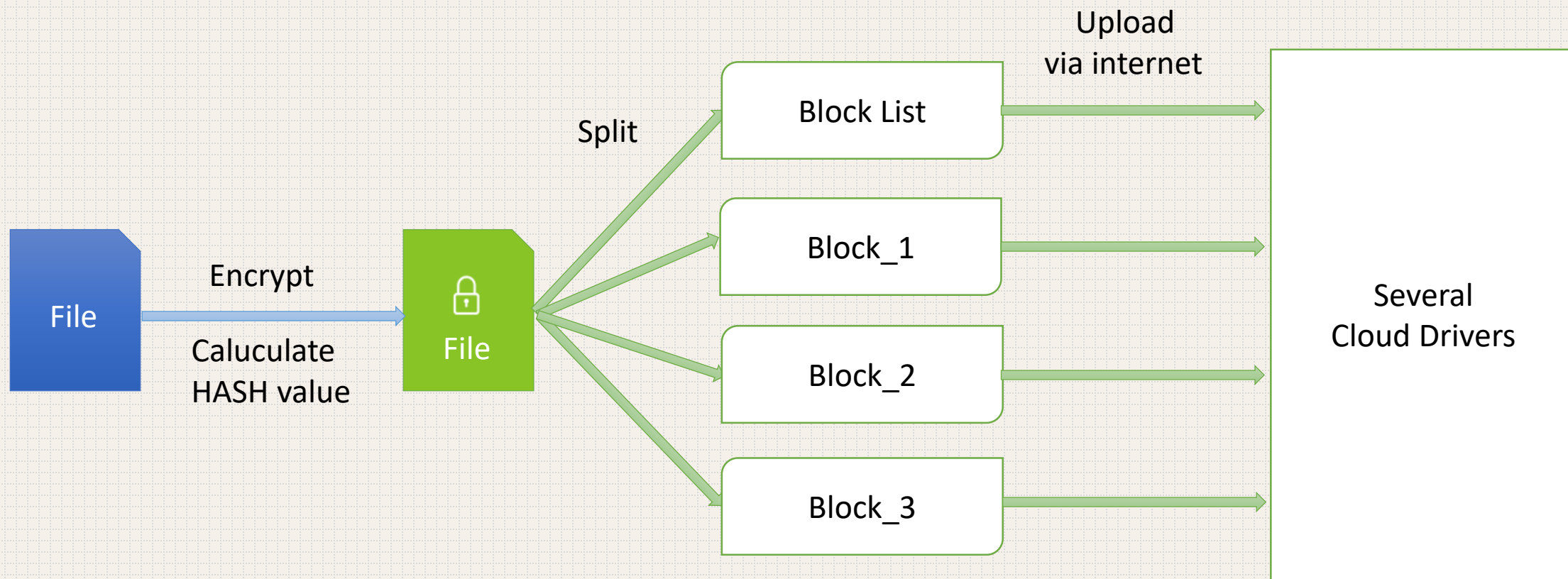


▸ 研究内容

- 实现一个具有Web前端以及本地客户端的同步系统
- 借助Python对本地文件操作
- 应用HASH码实现数字签名，AES-CBC算法在本地加密，确保上传的文件不会被损坏或篡改。
- 在存储时，文件按块分割(当作二进制数据流)，分别加密上传，考虑冗余以增加可靠性，另一方面，防止运营商对私人数据的泄露。

2

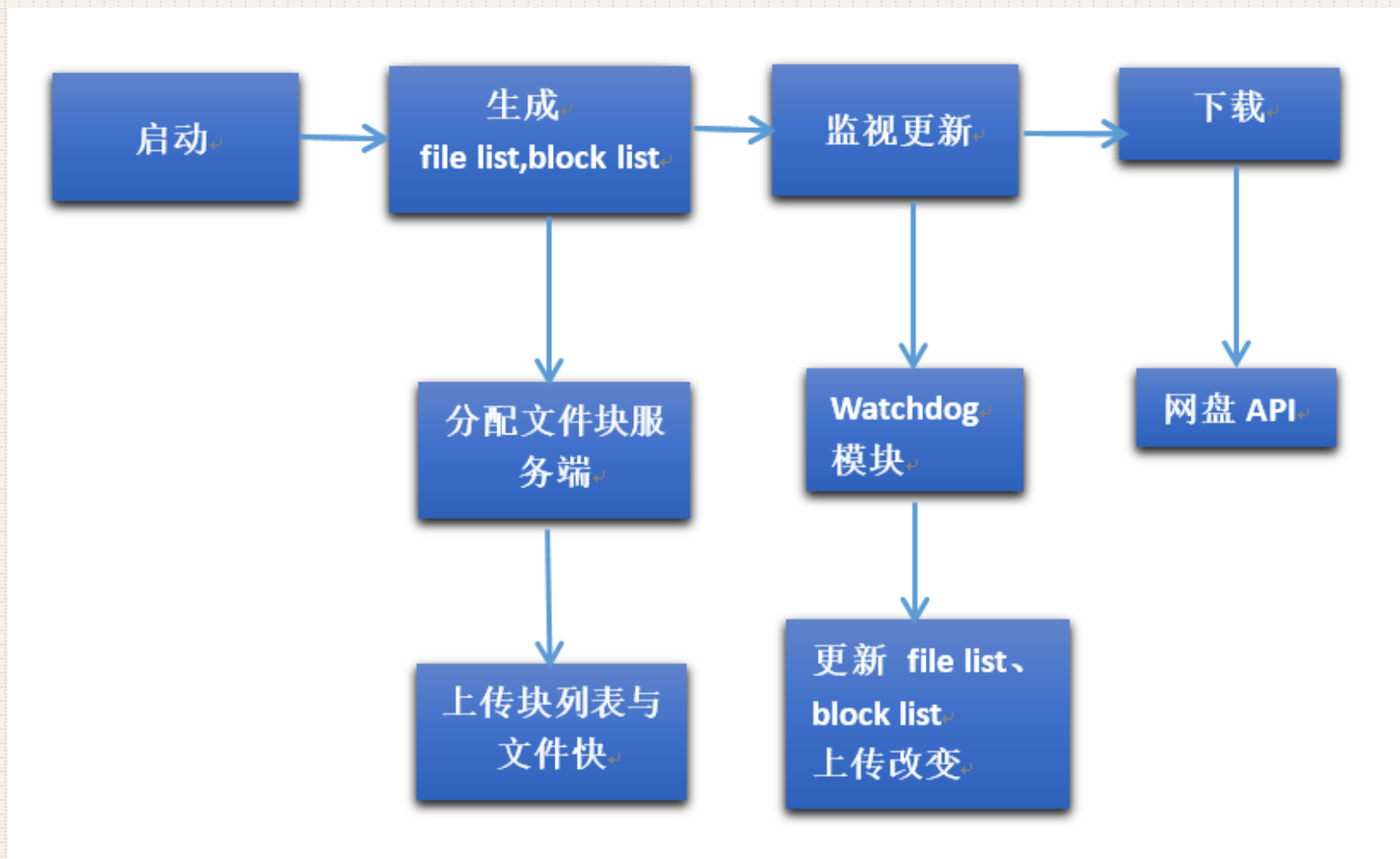
设计思路



2

设计思路

技术路线



▷ 面临的问题

- 如何实现本地和服务器端的上传和下载
- 如何实现文件同步
- 如何判断下载下来的单个分块是否完整
- 如何有效又节省空间地加冗余
- 如何管理云端网盘账户

etc.

第三部分

『实现方法』



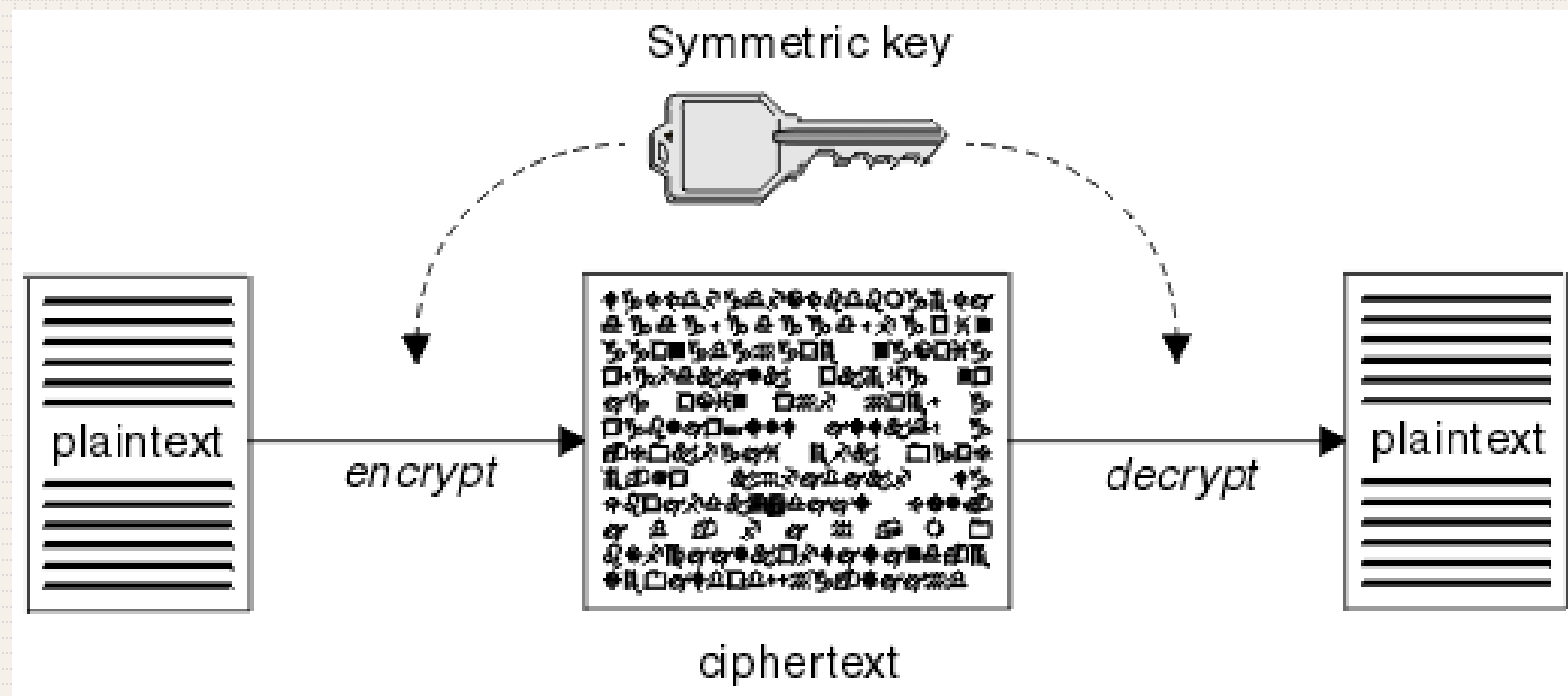
3

实现方法

- 本地加密
- 分块、冗余
- 闲时同步
- 上传、下载

➤ AES

➤ Crypto



3

实现方法

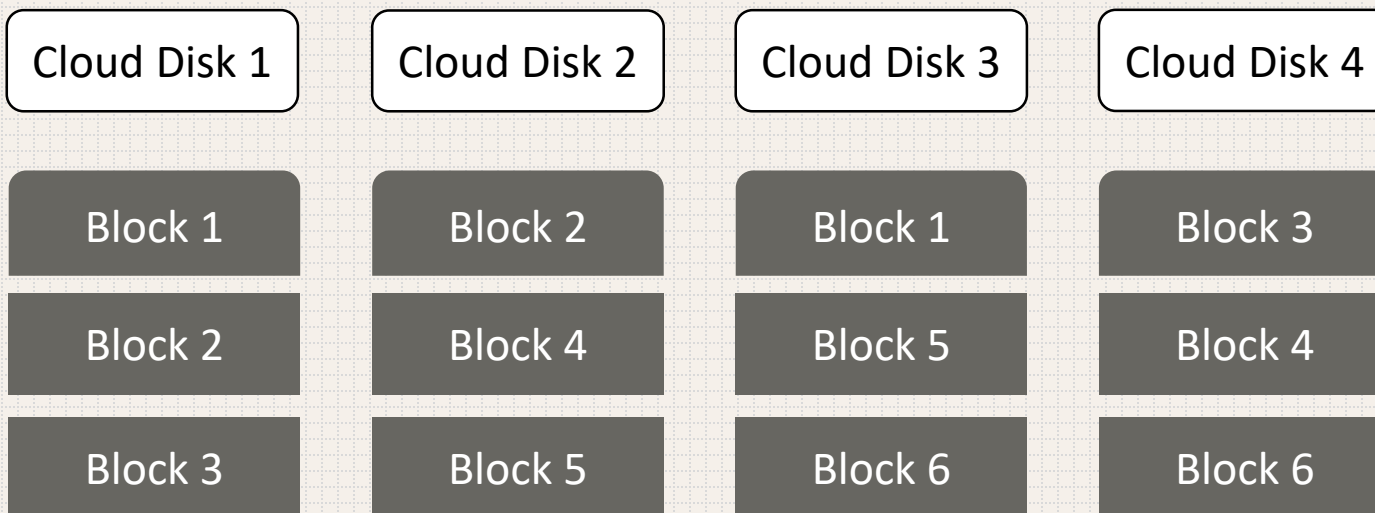
- 本地加密
- 分块、冗余
- 闲时同步
- 上传、下载

➤ 类似 RAID1

➤ 将分块信息存在块索引表中

- 分块信息包括文件的时间戳，HASH值，文件名，块数，及每个分块存储的位置

➤ 每个分块存储在至少两个云盘中



3

实现方法

- ▷ 本地加密
- ▷ 分块、冗余
- ▷ **闲时同步**
- ▷ 上传、下载

➤ Watchdog模块监测指定目录下文件修改和增删

3

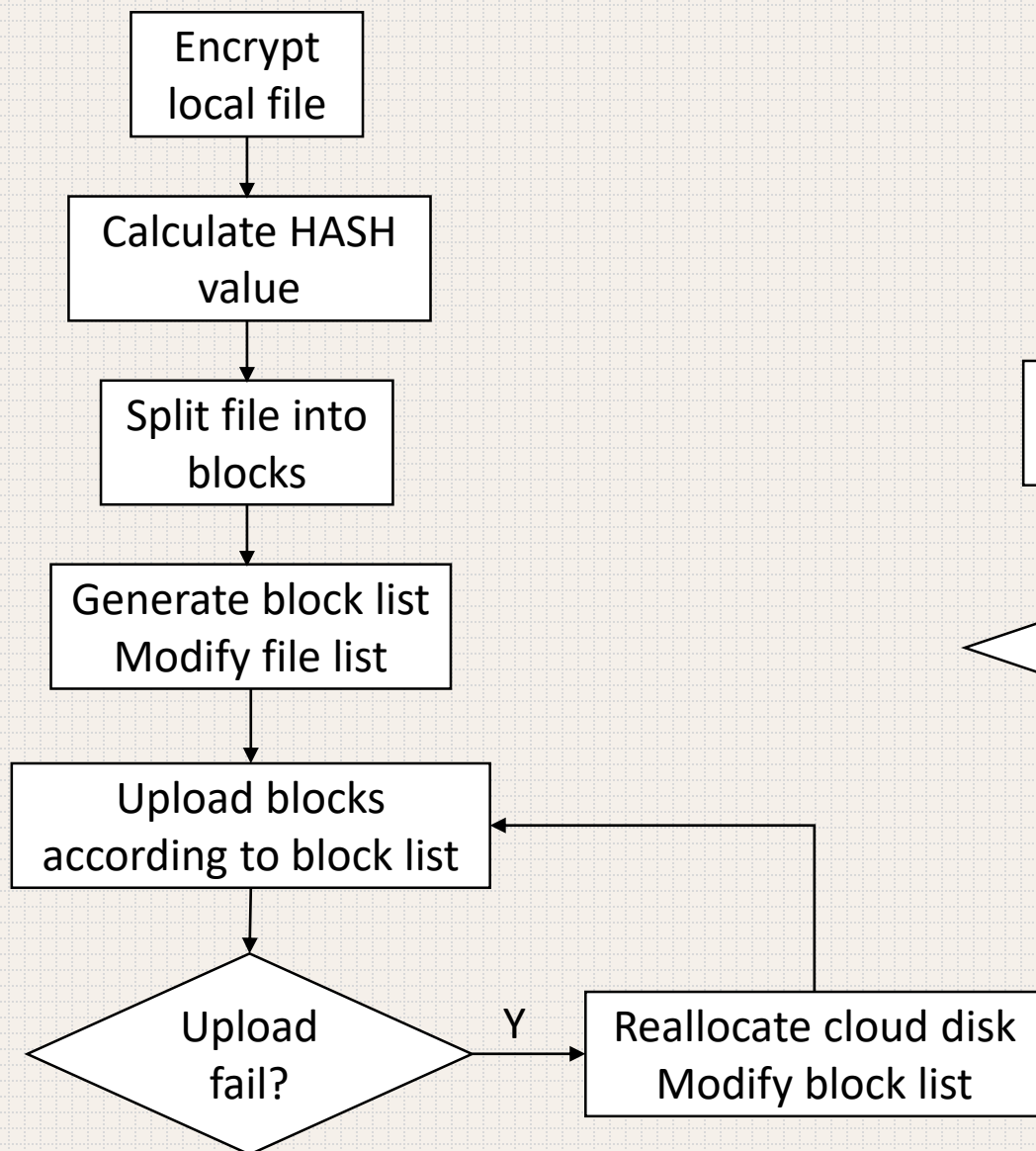
实现方法

- 本地加密
 - 分块、冗余
 - 闲时同步
 - 上传、下载
- 本地存储文件索引表，存放文件名和其对应块索引表存放位置
 - 每个文件对应一个块索引表，上传至多个网盘
 - Rclone
 - Multiprocessing

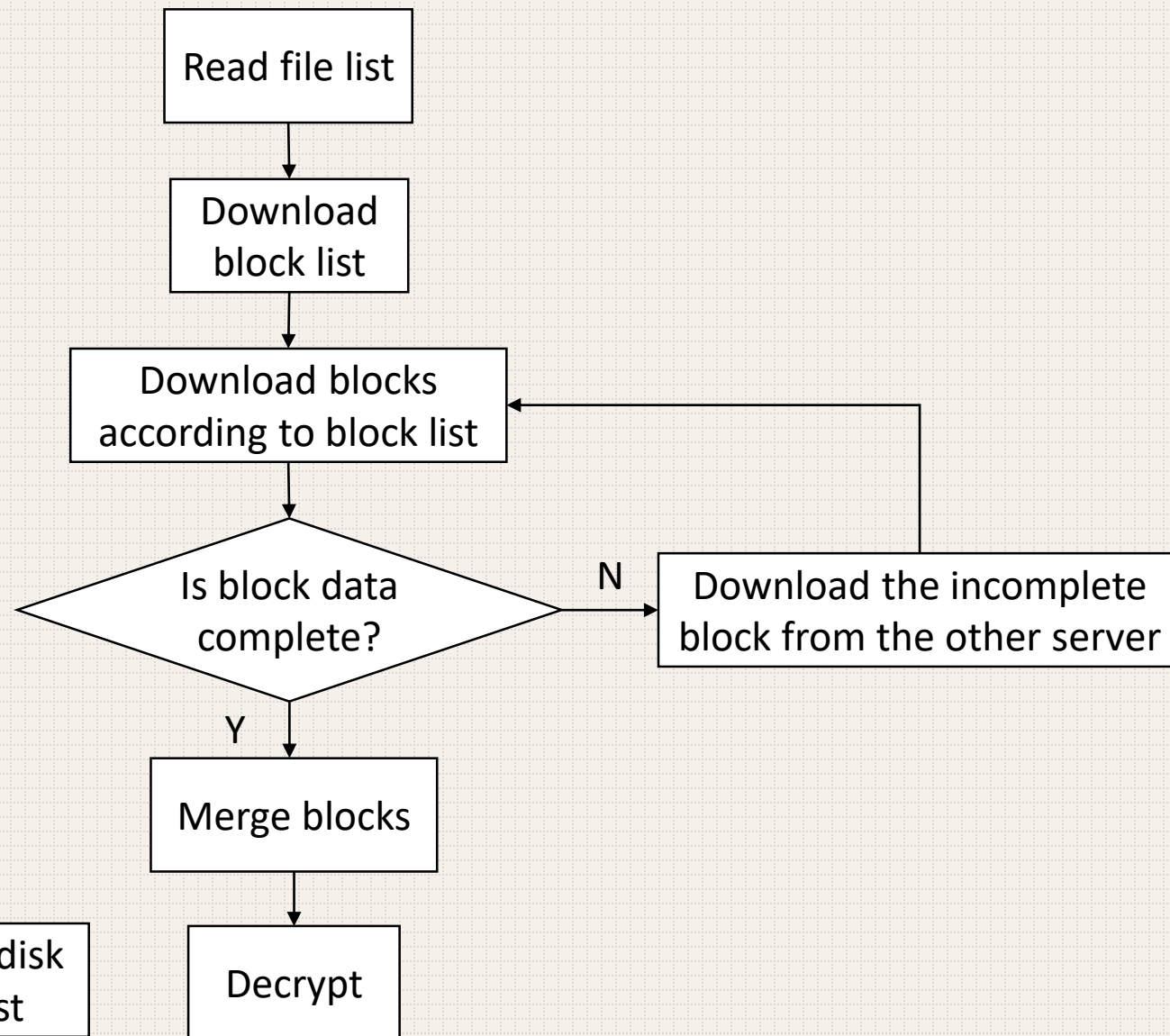
3

实现方法

Upload process



Download process



总体架构设计

➤ 前端

使用flask实现在本地运行的前端

➤ 依赖

rclone

python

python-flask

flask-bootstrap

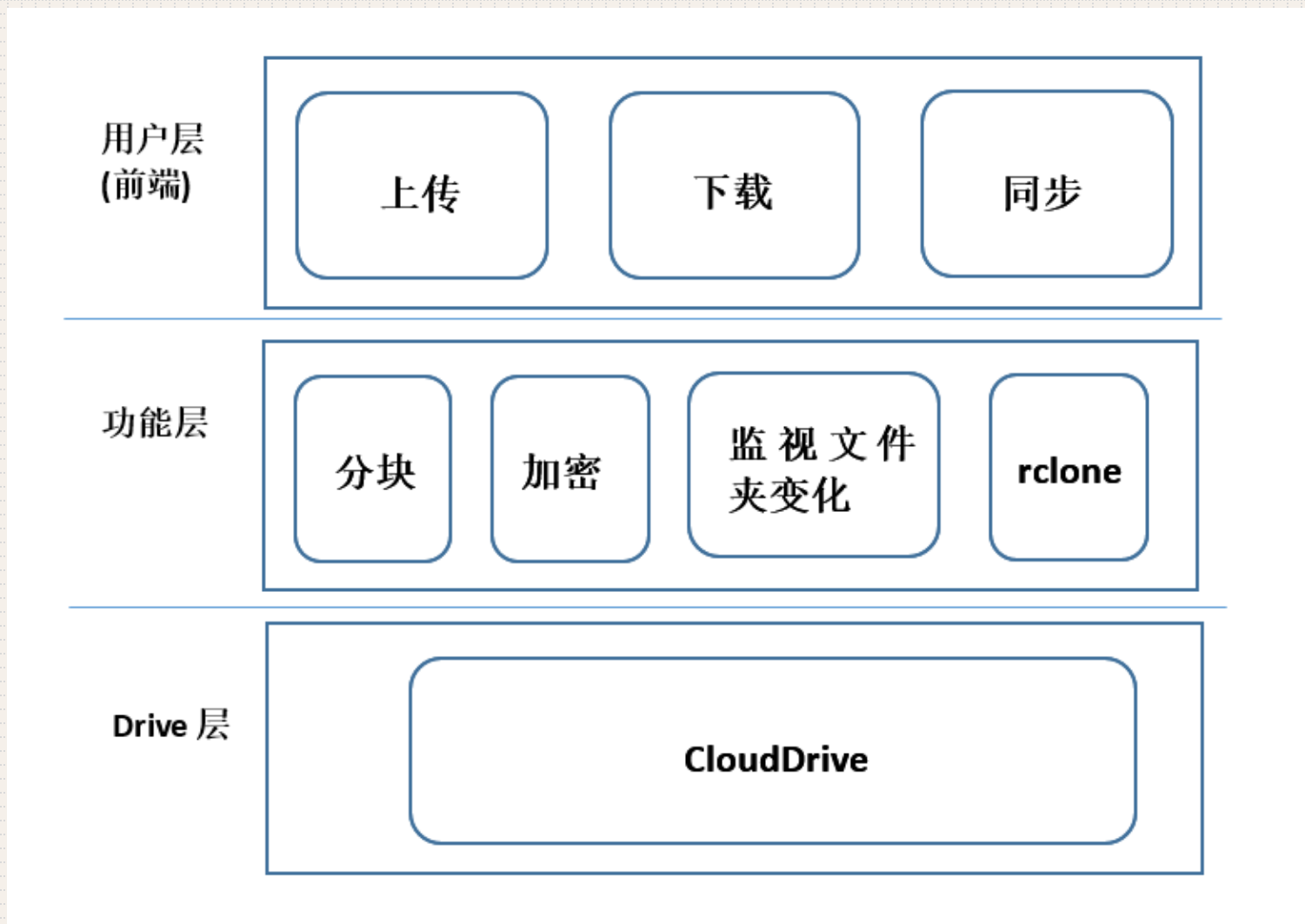
➤ 部署

python server.py

➤ 框架

1. (后台运行) 检查本地文件的更改(watchdog)
2. (后台运行) 向服务器发心跳包, 以及文件版本检查
3. (发现有更改) 上传至服务器
4. (发现服务器宕机) 调整分块列表, 向已有的服务器上传丢失分块
5. 加入新服务器
6. 空间满: 标记服务器为不可上传, 直到删除块

总体架构设计

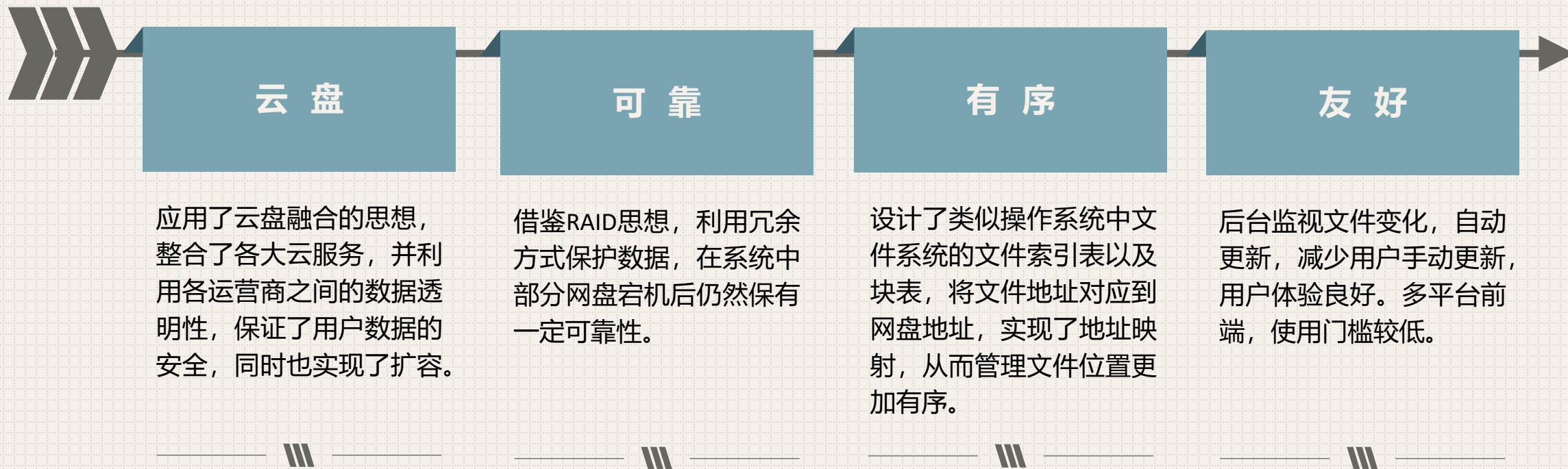


第四部分

『创新点』



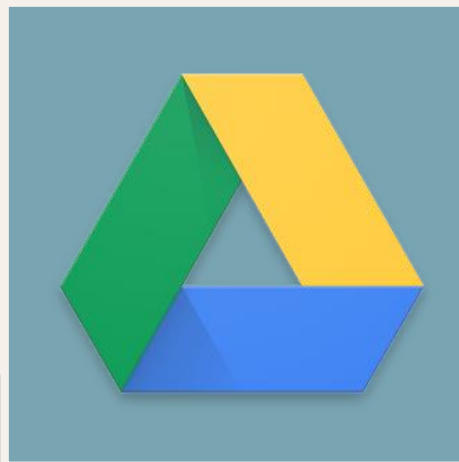
4 创新点



4

创新点

云
盘



4

创新点

▷ 可靠

-
- 通过加密与分块保证安全性
 - 通过冗余保证非易失性
-

4 创新点

▷ 有 序

设计了类似操作系统中文件系统的文件索引表以及块表，将文件地址对应到网盘地址，实现了地址映射，从而管理文件位置更加有序。

文件索引表 (file list) 格式

[filename1]

[blocklist地址(clouddrive代号)]

块索引表 (block list) 格式

[timestamp(使用大整数表示)]

[HASH]

[filename][block数]

#block 1

[clouddrive1][clouddrive2][...]

#block 2

[clouddrive1][clouddrive2][...]

#...

CloudDrive表

[name]

[download API URL]

[upload API URL]

其中，file list用来保存每个文件对应的块索引表网盘位置。块索引表存储在云端，通过取得块索引表，其中保存了文件的时间戳，HASH值，文件名和块数，接下来是每个块存储的云盘代号，为了保证文件可恢复性，每个块要求存在至少两个云端。

4 创新点

▷ 友好

-
- 后台运行，自动检测运行上传下载，减少用户操作
 - 利用python编写，保障跨平台性
-

第五部分

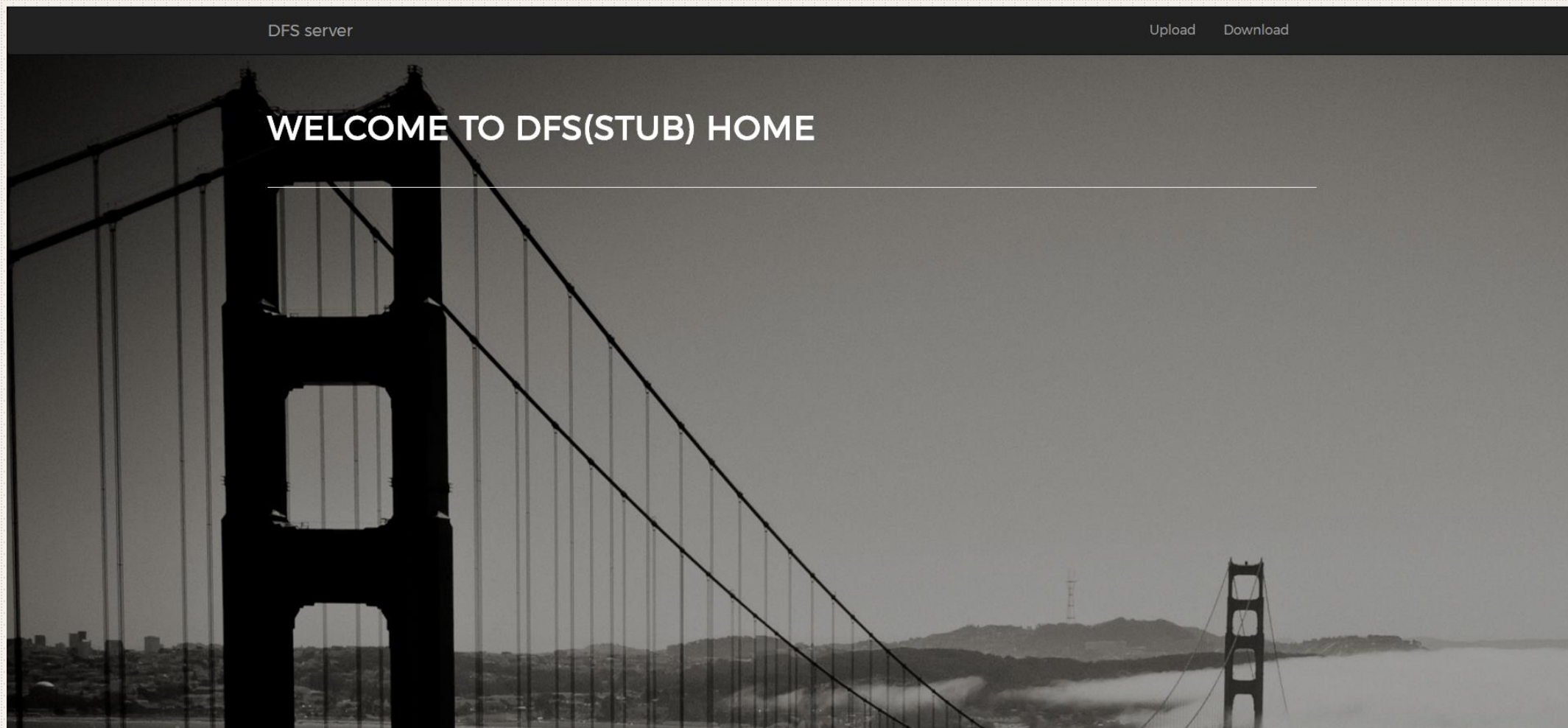
『成果展示』



5

成果展示

▷ 网页前端



5 成果展示

▷ 性能分析

- 上传100MB文件耗时20分钟
- 考虑到使用的国外网盘速度不稳定，以及网络情况较差，结果尚可以接受

谢谢！

PRESENTED BY WQ

指导老师：邢 凯
组员：王若晖 朱一铭 钟 立 韦 清