**基于多个云存储的网盘设计与实现**

|  |  |  |
| --- | --- | --- |
| 开发设计人员 | 丁智渊  1130372089  zhiyuan.ding@gmail.com | 崔盛斌  1130372086  cuishengbin0212@163.com |
| 郑富德  1130372125  food0517@163.com | 毛慧鎏  1130372103  tianguohut@163.com |

上海交通大学软件学院

2013年 11月 10 日

**变更记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 变更日期 | 变更类型 | 变更人 | 摘要 |
| 0.1 | 2013.11.10 | 新增 | 崔盛斌 | 新增文档 |
| 0.2 | 2013.11.12 | 修改 | 丁智渊 | 修改文档 |
| 0.3 | 2013.11.13 | 修改 | 郑富德 | 修改文档 |
| 0.4 | 2013.11.16 | 修改 | 丁智渊 | 增加系统用例图 |
| 0.5 | 2013.11.30 | 修改 | 毛慧鎏 | 补全文档，更改文档性质从概要文档变为实现文档 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[第一章 绪论 1](#_Toc373584980)

[1.1 背景概述 1](#_Toc373584981)

[1.2 系统目标 1](#_Toc373584982)

[第二章 架构设计 2](#_Toc373584983)

[2.1 架构设计 2](#_Toc373584984)

[2.2 系统运行环境 4](#_Toc373584985)

[2.3 系统部署 4](#_Toc373584986)

[2.4 开发语言及开发工具 4](#_Toc373584987)

[2.5 开发使用到的类库 4](#_Toc373584988)

[第三章 功能概述 5](#_Toc373584989)

[3.1 系统主要功能 5](#_Toc373584990)

[3.2 文件系统 5](#_Toc373584991)

[3.3 网盘API整合 6](#_Toc373584992)

[3.4 客户端应用程序 8](#_Toc373584993)

[第四章 任务分工 9](#_Toc373584994)

# 绪论

## 背景概述

随着云计算的发展，市场上出现很多有关云方面的产品，云盘服务是其中的一种。

云盘是互联网云技术的产物，它通过互联网为企业和个人提供信息的存储，读取，下载等服务，云盘相对于传统的实体磁盘来说，更方便，用户不需要把存储重要资料的实体磁盘带在身上，只要能访问互联网，就能下载到本地字盘。云盘不仅可以存储大容量的文件，而且可以通过好友分享，轻松与其他人进行分享,云盘也能提供备份的功能，可以完全映射一个本地磁盘，并通过一定的策略机制来进行同步。

虽然各大公司提供了网盘服务，并且提供了不小的空间，但是毕竟受限于各公司提供的空间大小，如果需要更大的空间，还是需要花钱购买，本文主要针对这个问题进行研究，如何能不通过购买的方式同样可以享受超大网盘的方法。

## 系统目标

本系统主要的目的是通过技术方法来解决如何能通过免费的方式享受到更大的网盘空间,通过在本地虚拟一个磁盘，让用户觉得所有的云盘合并在一起，得到了一个大容量的云盘，对用户来说，与云盘之间操作是透明的。

当前各大公司都提供了一定容量的网盘空间，并提供了相应的OpenAPI对网盘进行操作，那么我们可以利用这些API来整合各大公司的免费网盘成一个超大容量的网盘，通过客户端软件（映射本地一个磁盘）来操作这些网盘，对于用户来说还是一样的上传，下载操作，但是享受的免费容量将是以前的好几倍。

本次系统我们将整合百度、金山、新浪网盘成一个网盘。

# 架构设计

## 架构设计

本系统主要由三个主要模块组成：

1. 文件系统。系统将通过虚拟磁盘的方式提供给用户进行方便的操作，用户可以通过对本地映射的虚拟磁盘做新增、复制、删除、剪切等操作来对云盘进行操作。
2. 网盘API整合。系统将整合百度、新浪、金山的OpenAPI为一个API供客户端调用，通过一个调度算法，将用户需要执行的操作路由到对各个实际网盘的操作。
3. 客户端应用程序。本系统将提供一个可视化界面供用户进行相关的设置，如设置百度、新浪、金山网盘的帐号，设置本地文件的存储路径等。



Figure 1 系统整体框架图

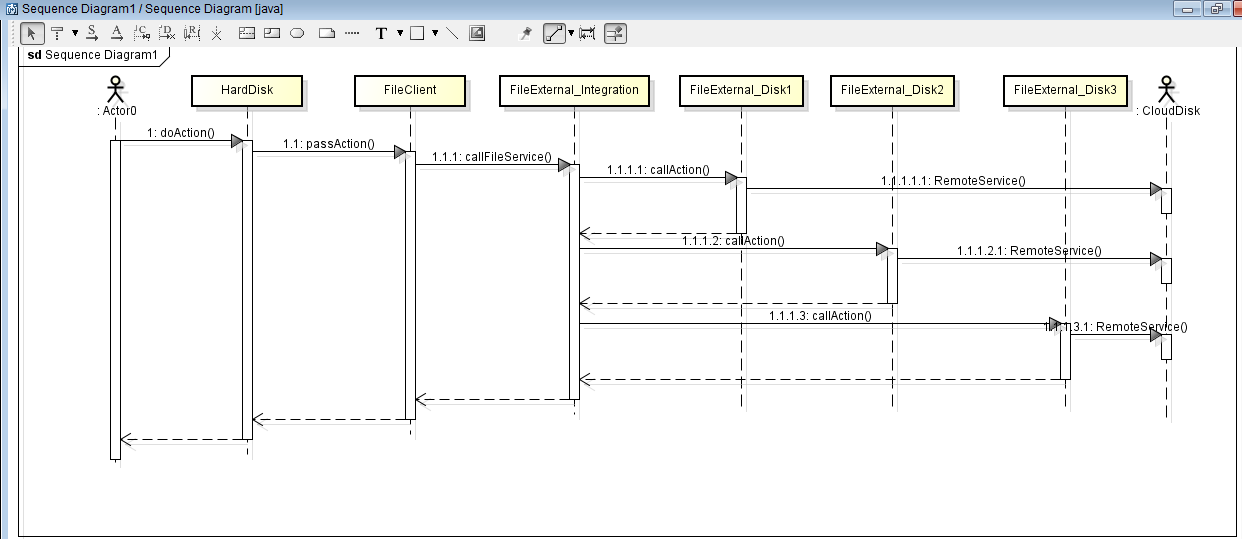


Figure 2 系统总体顺序图

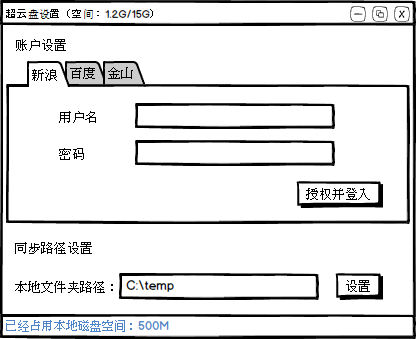


Figure 3 配置界面线框图

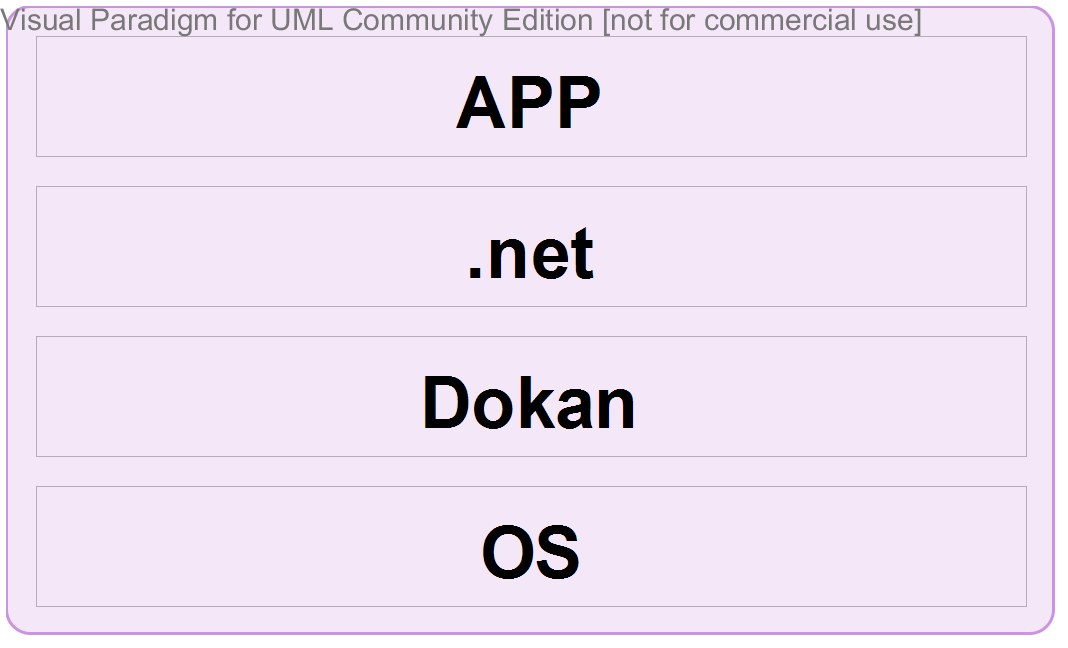


Figure 4 本地文件系统详细图

## 系统运行环境

本系统暂时只支持Windows下运行。

## 系统部署

本系统将提供一个exe安装文件包供用户进行一键式安装。

## 开发语言及开发工具

开发语言：C#.net

开发工具：Visual Studio 2010

## 开发使用到的类库

本地文件系统：Dokan

开发主体使用的第三方类库：.net framework 2.x

# 功能概述

## 系统主要功能

系统通过一个可视化界面，与逻辑控制模块进行交互。而逻辑控制模块则负责在相应条件下（例如检测到本地本件或者远程文件的变化时）与本地文件系统接口、网盘接口实现交互。主要场景用例图描述如下图所示：

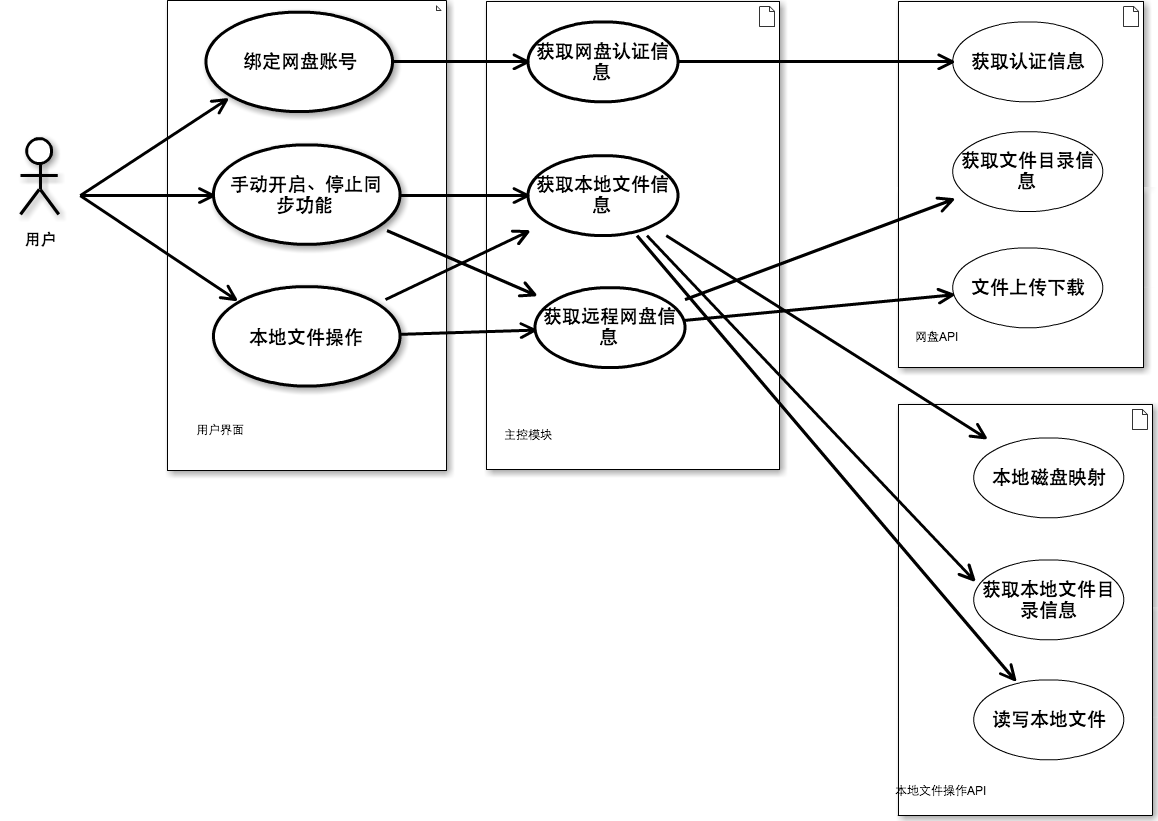


Figure 5 系统用例图

## 文件系统

本模块主要在用户系统中创建虚拟磁盘，用户可以通过此磁盘上传下载云盘的内容。此模块的功能总体来讲就是同步云盘的功能，本系统会对这个磁盘进行监控，用户对此磁盘的操作就相当于对云盘做了操作，对于用户而言，本系统的虚拟盘只有一个，总的容量大小是已经授权的所有网盘的综合，而就本地磁盘内的文件映射到网盘上哪个网盘中，本系统可以通过配置的方法给到用户选择，但是在实际操作时，只要配置完成了，那对用户而言是透明的。

如复制文件或是文件夹到此磁盘就相当于上传文件或是文件夹到云盘；用户删除此磁盘上的文件或是文件夹就相当于对云盘上的内容做了删除。系统会事实监控此磁盘的内容，如果本地文件或是云盘上的内容发生变化，那么将同步到此磁盘中。本文件系统将通过开源Dokan来实现相关的功能。

### 系统功能罗列

1. 虚拟磁盘的创建。系统将在安装时在用户系统中创建相关的虚拟磁盘并于实际的存储路径相关联。
2. 文件或文件夹的创建。在虚拟磁盘中创建相应的文件或是文件夹。
3. 文件或文件夹的删除。在虚拟磁盘中删除相应的文件或文件夹。
4. 文件或文件夹信息的获取。获取虚拟磁盘中文件或文件夹的信息。
5. 文件覆盖更新。覆盖更新虚拟磁盘中的文件或文件夹。
6. 系统提供右键快捷操作，可讲用户系统中的任意文件直接拷贝到文件系统中来。

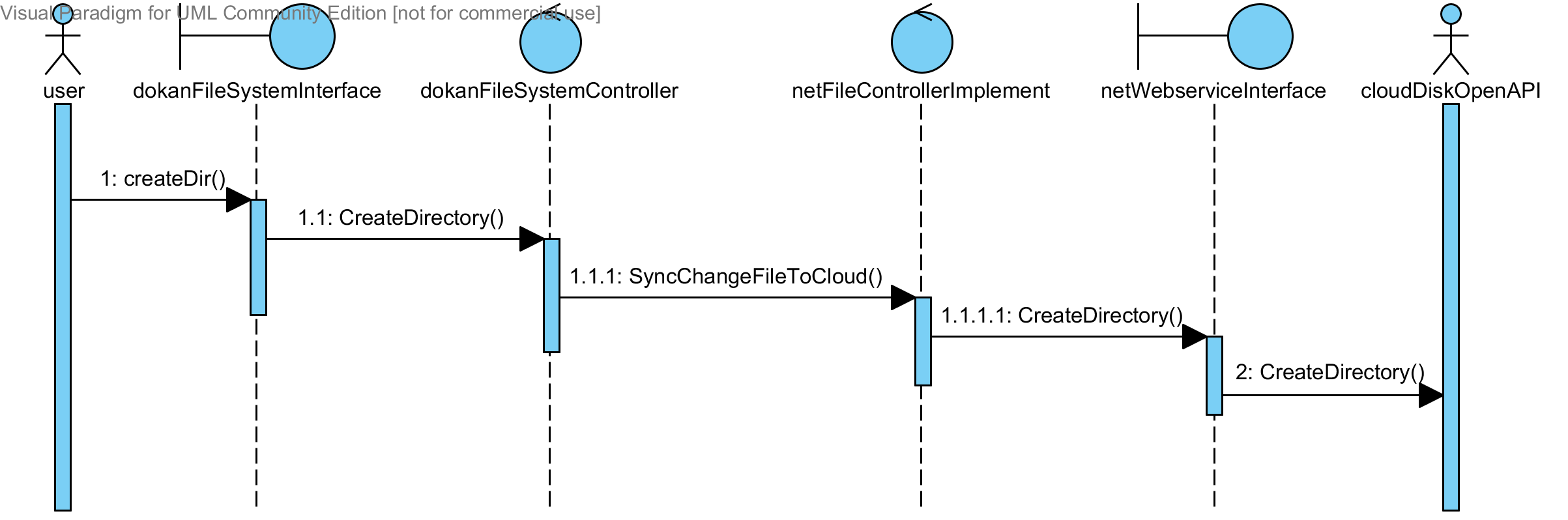


Figure Use case example: Create Directory

## 网盘API整合

本模块将整个各个网盘的API后提供一个统一的操作API供客户端应用程序使用本API，客户端程序不用关心用户的文件是如何到达网盘，或是本次操作实际是对哪个网盘进行操作的，它只需要调用此API完成相关的操作即可。主要功能有：

1. 登录验证。验证用户设置的用户名和密码是否正确，如果正确从实际网盘获取相应的token。
2. 获取实际网盘的容量信息。可以获取用户相应的网盘总的容量以及当前可用的容量。
3. 创建目录。在实际网盘上创建目录。
4. 删除目录。在实际网盘上删除目录。
5. 获取目录信息。获取实际网盘的目录信息，包括目录的相关信息，如创建时间，修改时间等，以及目录下的文件列表信息。
6. 上传文件。上传文件到相应的网盘。
7. 下载文件。从网盘上下载文件。
8. 删除文件。从网盘上删除文件。
9. 移动/复制文件。

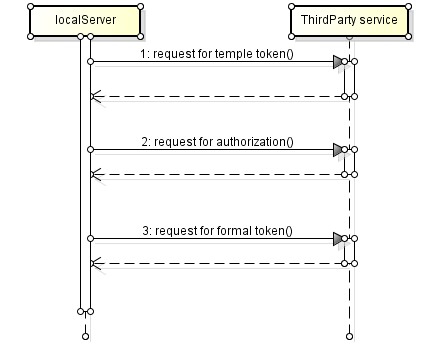


Figure 用户授权

## 客户端应用程序

本系统将提供一个简要的可视化界面，供用户进行相关的一些系统参数设置。主要功能有：

1. 账户设置。用于用户进行设置百度、新浪、金山三个网盘的用户名和密码。在用户设置时需要验证用户名密码是否正确，如果不正确需要提示用户重新进行设置。
2. 设置本地磁盘位置。用户设置云盘的文件内容具体放到本次的哪个磁盘路径上。
3. 同步功能。用户可以手动点击同步按钮进行本地磁盘与网盘内容的同步。

# 任务分工

本系统的实现由我们四人共同实现，具体分工如下：

丁智渊、崔胜斌完成第三方网盘API的整合。

郑富德、毛慧鎏完成文件系统的相关功能，以及客户端应用程序。

# 详细设计

## 系统使用策略

### 文件系统队列轮询策略

由于本地映射的磁盘对用户来说是一个与物理磁盘相同的磁盘，所以理论上普通磁盘能够进行的操作该磁盘也需要支持，那么，用户会自然而然的将一些体积比较大的内容或者大批量的文件移动或复制到该磁盘中，如果我们实时的去请求对云盘的操作，那会受限于网络条件和文件体积的大小。在极端情况下，可能导致磁盘的无响应，给用户提供了不好的使用体验。

为了解决这样的情况，我们采取了内存Cache中间层的策略，即在用户直接对磁盘进行操作后，系统将用户的操作和操作对象组成一条OP指令，将指令存放在内存Cache中（该Cache由系统自身创建和维护），而并不是直接与云盘接口操作，这样可以快速的得到操作的返回结果，给到用户快速的反应。

在系统启动时，会创建出一个线程，该线程会每10秒扫描一下该内存Cache中是否存在需要执行的OP指令，如果存在，即立即进行云盘操作，等到操作结果返回后即可将该条OP指令从内存Cache中撤出。

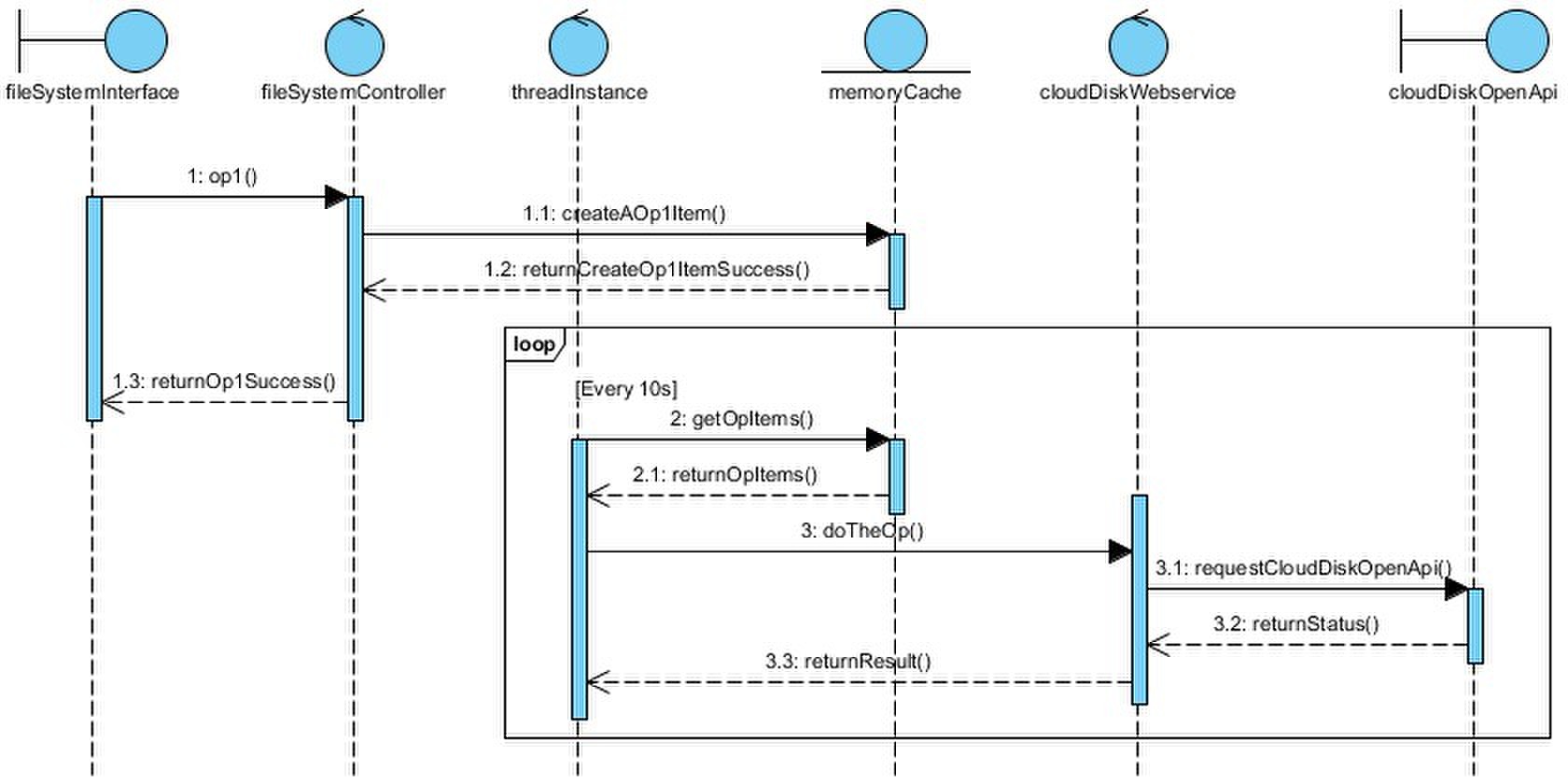


Figure 文件系统队列轮询策略

### 云盘文件分配策略

我们系统的最终的目标是使用户不用关心具体文件存储在云盘的具体位置，即多个网盘中无需关心到底存放在那个网盘上，从而对用户而言是通过授权多个网盘来达到总容量扩充的感觉，所以必须要有一个策略来决定用户放入磁盘的文件实际存储的位置。

本系统中我们可采用了配置文件方式来提供用户配置这个存放策略，而我们支持的策略包括：随机策略，最大容量优先策略，多副本策略。

#### 随机策略

需要上传的文件会被一个hash算法计算出实际需要存放的网盘位置（多个网盘随机获得一个），我们使用文件名作为种子计算出实际网盘序号，下次或许文件时也使用这个hash算法，由于hash算法可以保证每次计算得出结果相同，所以可以知道存储在哪个网盘上。

#### 最大容量优先策略

根据每个网盘剩余的空间大小倒序排列，将新的文件优先存储在空余空间大的网盘。

#### 多副本策略

将需要上传的文件上传至所有用户授权的网盘，知道该网盘空余的空间不够存储该文件为止。