
项目：一种基于分布式技术的永久云存储实现

终稿

Version 4.3

2022.01.20

项目成员	
高歌	2030416018@stu.suda.edu.cn
**	*****
**	*****
**	*****
**	*****

历史版本

版本号	修改人	修改日期	备注
1.0	高歌	2021/10/08	初稿
1.1	**	2021/10/10	增加了部分核心概念
1.11	**	2021/10/10	对一些概念进行修改
1.12	**	2021/10/16	补充了一部分项目背景
1.13	**	2021/10/17	增加了部分核心概念
1.2	高歌	2021/10/17	增加引言，修改问题描述，修改核心概念排版
2.0	高歌	2021/11/16	撰写二稿，并修改一稿中部分描述
3.0	高歌	2021/12/14	撰写三稿
4.0	**	2022/01/19	增加活动图，修改原稿中部分措辞
4.1	**	2022/01/19	修改了部分系统概述
4.2	**	2022/01/20	修改了部分概念的描述
4.3	高歌	2022/01/20	撰写终稿

成员工作内容

姓名	工作内容	备注
高歌	撰写文稿	组长
**	修改文稿	组员
**	修改文稿	组员
**	修改文稿	组员
**	修改文稿	组员

目录

0	引言.....	3
1	问题描述.....	4
1.1	分布式存储行业现状分析	4
1.2	传统集中式云存储方式的缺陷	4
1.3	其他云存储方案的缺陷	5
1.4	目标系统概述	5
1.5	预期实现目标	5
2	核心概念.....	6
2.1	分布式存储系统	6
2.2	CAP 原则	6
2.3	分层及分片	6
2.4	冗余及版本控制	6
3	领域和需求概要分析.....	7
3.1	利益攸关者	7
3.2	基本用户需求	8
3.3	领域描述	8
3.4	问题框架	9
3.5	非功能性需求	9
4	系统需求分析与业务建模.....	11
4.1	项目范围	11
4.2	用例图	12
4.3	核心用例及用例描述	12
4.3.1	文件交互	12
4.3.2	积分转账	13
4.4	业务对象模型	14
4.5	分析类模型	15
4.6	系统架构	15
4.6.1	用户层	15
4.6.2	积分交互层	16
4.6.3	文件交互层（子网/主网）	16
4.6.4	服务器层	16

0 引言

当前随着各类互联网服务体量规模的增大，以及数据访问频率的增加，对高效可靠数据存储服务的需求日趋增加。二十一世纪以来，随着大数据与云计算的发展，作为一种新型且较为可靠的存储模式的分布式存储领域也在不断成熟。近年来，以Dynamo、Cassandra、BigTable等为代表的分布式数据存储系统已经相当成熟，在低时延、高度一致性、分区容忍度等方面均有不同程度的突破。然而目前这些技术多用于大企业的数据中心，并且由于这些技术设计时并未过多考虑分散零碎的数据存储的安全性，而是更多从提高数据交互效率方面考虑，因此并不适用于面向个人及小企业的小体量数据的可靠存储。此外，相比于企业，个人往往对云存储中的隐私安全及数据可靠性有更多需求，而目前市面上存在的大多数云存储服务的安全性也很难令用户满意。这带来了个人及小型企业对更加安全可靠的云存储服务的需求。本项目将尝试提出一种基于分布式存储的新型云存储技术来解决这一困境。

1 问题描述

1.1 分布式存储行业现状分析

如图 1-1 所示，近年来随着以大数据及云计算为代表的互联网技术的快速发展，我国对分布式存储行业的需求正越来越大，并且目前为止仍保持着很高的增速。从可预测的未来来看，由于大数据、云计算等新兴互联网技术仍具有较高热度，对于分布式存储的需求仍会不断增长。

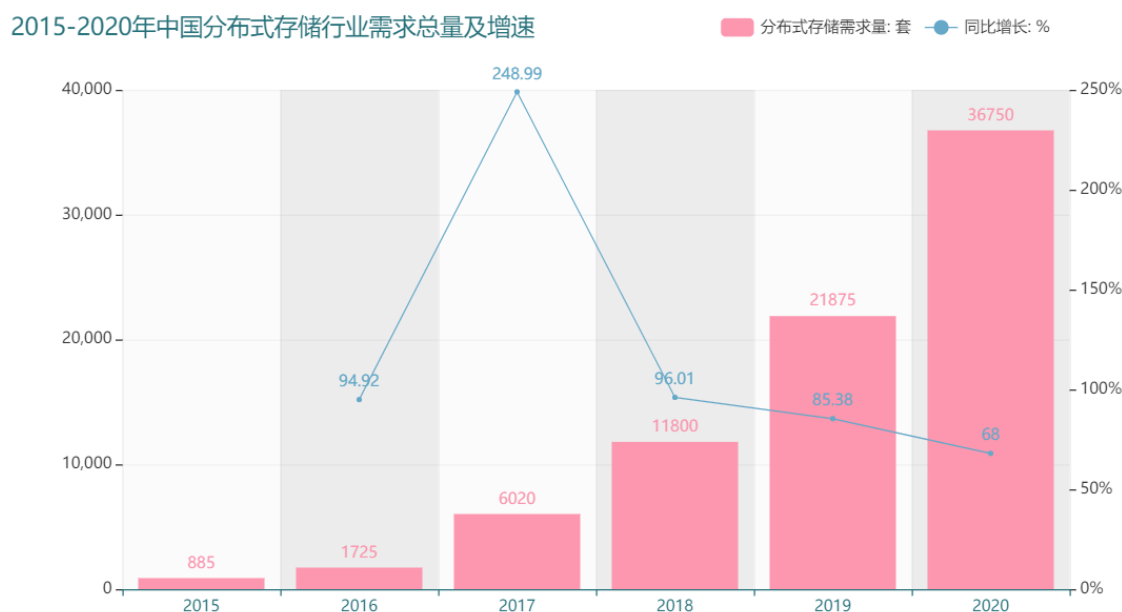


图 1-1 2015-2020 年中国分布式存储行业需求总量及增速

相比于传统集中式存储方案，分布式存储具有高性能、易拓展与数据可靠等多项优势。尽管具有诸多优势，分布式存储的主要应用领域目前仍主要局限大型数据中心，抗干扰能力较弱，难以应对高度复杂的数据存储需求，难以向小型企业及个人提供云存储解决方案。

1.2 传统集中式云存储方式的缺陷

当下来说，互联网面向个人的云存储服务多采用传统的集中式存储方案，例如百度网盘、阿里云盘、Dropbox、Google Drive 等。尽管适用于基于服务器集群的数据中心的分布式存储方式已在华为等公司大规模运用，但由于大多数云服务的数据交互频率不高，因此主流云服务仍选择更加成熟的集中式存储方案。

然而集中式存储方案的数据安全性并不令人满意。近年来云存储安全事故频发，例如百度贴吧近年来发生的大规模数据丢失事件与 Instagram 的数据泄露事件。这些事件的发生既有偶然因素也有人为因素掺杂其中，然而不论事件发生的具体原因如何，都可以说明目前大多数云存储服务难以保证用户数据的隐私安全与长期存储安全。

此外，传统集中式存储方案也对集中式的服务器集群存在高度依赖，数据密度很大，数据安全取决于服务器的稳定性以及服务提供商的信誉，若服务器出现故障或是服务提供商进行人为干涉，极可能出现大规模的数据丢失，数据安全难以保障。

1.3 其他云存储方案的缺陷

当然，除传统集中式存储之外，也存在其他云存储解决方案，例如并不依赖于大型服务提供商的 P2P 协议及 BitTorrent 协议。

然而这些技术仍对服务端的稳定性有极高的依赖，一旦服务端网络掉线，极有可能出现数据的永久丢失。以 BitTorrent 协议为例，若无人持续提供服务端以供下载者下载文件，BT 链接将会失效，最终下载量低的文件将会永久从网络中丢失。

此外，这些技术也普遍存在内容审查难以进行的问题，本项目将同时对如何解决内容审查问题进行一些尝试。

1.4 目标系统概述

由于当前缺少合适的云存储方案能够保证数据存储的长期安全，而个人及小企业又往往在数据安全方面对存在隐私安全以及存储安全存在更多需求，因此建立一套完备的永久存储云体系是十分必要的。

本项目主要面向个人及小型企业，以永久存储作为核心需求，利用分布式技术，结合网络分层、区块链、P2P 等其他技术作为辅助，建立一个高效的云数据存储系统，并以其高度拓展性为众多互联网服务提供较为可靠的平台。

系统应当以数据的长期稳定存储安全作为前提，尽可能保证数据的隐私安全与网络一致性，并保证合理的传输速度与可用性。同时需要保证系统具有极高的抗干扰能力，最大程度上降低节点欺诈以及其他第三方攻击对系统安全稳定运行带来的风险。

为了系统的长久稳定运行，本项目还会实现激励方式以保证较为充足的服务端供应。此外，为了适应各国家的法规政策，本项目还将引入内容审查系统。

最终系统将以高度去中心化的方式在互联网中长久安全稳定运行。

1.5 预期实现目标

优先级从上到下依次递减。

- 建立分布式存储网络，最大程度上消除对主干网的依赖，优先保证数据安全。
- 保证网络稳定性。
- 保证数据一致性。
- 在最大程度上保证隐私安全。
- 引入分层，提高网络的易用性及传输速度。
- 在分层网络的基础上，尽可能提高网络速度。
- 增强网络对第三方攻击的抗性。
- 实现激励层
- 建立适应各国法规的内容审查体系
- 在服务器充足的前提下扩充网络，增强数据流动性（可选）

2 核心概念

2.1 分布式存储系统

分布式存储系统指将数据分散存储在多台独立的设备上，采用可扩展的系统结构，利用多台存储服务器分担存储负荷，利用位置服务器定位存储信息，它不但提高了系统的可靠性、可用性和存取效率，还易于扩展。

该项目所使用的分布式系统采用**完全无中心架构**。在这种架构中，客户端通过设备映射关系计算出其读写数据的数据节点位置，从而可以直接访问数据存储节点。

2.2 CAP 原则

CAP 原则又称 CAP 定理，指的是在一个分布式系统中，一致性（Consistency）、可用性（Availability）、分区容错性（Partition tolerance）。CAP 原则指的是，这三个要素最多只能同时实现两点，不可能三者兼顾。

- **一致性**：分布式存储系统需要使用多台服务器共同存储数据，为了保证在有服务器出现故障的情况下系统仍然可用，一般做法是 把一个数据分成多份存储在不同的服务器中。这里称保证多个副本的数据完全一致的性质为一致性。
- **可用性**：在系统中的一部分节点出现故障之后，系统的整体不影响客服端的读/写请求称为可用性。
- **分区容错性**：当一个网络因为故障而分解为多个部分的时候，分布式系统需要具有一定的容错性来处理网络故障带来的问题。

2.3 分层及分片

分层及分片是提高分布式存储系统效率的两种方案。

- **分层**：网络分层即将网络节点所要完成的数据的发送或转发、打包或拆包，控制信息的加载或拆出等工作，分别由不同的模块去完成。这样可以将往来通信和网络互连这一复杂的问题变得较为简单。
- **分片**：分片（partition）即通过将数据分别存放在不同节点中以提高操作效率。在数据规模较大时，分片是唯一的选择。

2.4 冗余及版本控制

冗余及版本控制是保证分布式存储系统可靠性的两种思想。

- **冗余**：即多个节点负责同一个任务。在分布式存储中，多个节点复杂存储同一份数据，以此增强可用性与可靠性。同时，冗余也会带来性能的提升。
- **版本控制**：版本控制是指对软件开发过程中各种程序代码、配置文件及说明文档等文件变更的管理，是软件配置管理的核心思想之一。

3 领域和需求概要分析

3.1 利益攸关者

1. 用户：
 - a) 用户可以在网络中自由上传与下载文件，并可选择是否加密。
 - b) 用户可以将文件通过指定链接共享。
 - c) 用户可以通过积分系统提高请求响应速度，但仍可选择不对请求附加积分。
 - d) 请求处理的平均响应时间、平均需支付积分、网络易用性等都将直接影响用户体验。
 - e) 用户账号信息及文件的安全性将得到极其严格的保障。
2. 服务器提供者：
 - a) 服务器提供者将以更高的基础比率获得定时发放的积分。此举用于鼓励任何用户在提供带宽以促进网络运行，即使他们计算机的性能不足以同专业服务器竞争，但仍能够增加网络的安全稳定性。
 - b) 主动响应附带积分的请求并成功处理后，服务器将获得积分奖励。
 - c) 共有两大类服务器：专业服务器与业余服务器。专业服务器将获得更多积分奖励，然而这类服务器运行之前必须先将一定量的积分锁入服务器才能被网络识别，并将在正常运行十二个月后才能取出，专业服务器所获得积分也必须要六个月后才能提取。此外若专业服务器处理请求时出现延迟及错误，将会扣除一定量的初始积分，初始积分归零后服务器将不再被网络识别。此举是为了保证网络的稳定性。
 - d) 在网络中仍存在未附加积分请求的情况下，服务器必须提供一部分带宽以响应未附加积分的请求。该比例将根据网络情况自动调整。
3. 政府组织：
 - a) 网络的运行将遵循各地法规进行。在政府向基金会提交请求后，可以开通专用子网络，该网络中一切信息将对其公开，并且网络自身会屏蔽其他一切子网络在该地区的访问。
 - b) 政府组织将在不影响网络整体运行的前提下拥有对所拥有子网络的最高权限，可以进行屏蔽匿名用户、查看信息发送者信息等操作。
4. 开发者：
 - a) 该项目将提供大量底层接口，以供任何感兴趣的开发者使用。
 - b) 开发者既可将应用运行于公用网络之上，也可专门将自己的服务器接入网络作为私有服务提供者。这类特殊服务器将不能获得积分奖励，但可以选择仅处理与自身利益有关的请求。
 - c) 开发者必须遵循各国法规。若开发者的应用需要进驻某一已经申请独立子网络的地区，可以选择将应用信息直接公开于目标政府或在目标子网络建立新的应用。
5. 基金会：
 - a) 基金会负责网络的基本治理，但部分最高权限不被任何人所有，例如关闭网络。
 - b) 基金会将由大量积分持有者、高活跃用户以及优秀开发者、贡献者组成，将完全遵循开源宗旨，一切决定将由投票进行并且完全公开。
 - c) 基金会的部分重大决议将下放至全体用户进行投票，每个用户根据其持有积分的数量以及活跃程度、贡献值将具有不同的投票比重。在十四天之内若决议获得了三分之二以上投票者的同意将会通过。

3.2 基本用户需求

1. 登录及身份认证：

用户分为两种，认证用户及匿名用户。

- a) 匿名用户：仅能提交最基本的上传/下载文件请求，并且不能够使用积分。匿名用户发送请求的处理优先级将低于认证用户。
- b) 认证用户：匿名用户可以通过受基金会认可的第三方 DID（去中心化身份）服务进行身份认证，以后也将通过认证使用的 DID 服务进行登录。认证用户将可以使用积分处理请求，并且会定期获得一定的积分奖励。

2. 文件机制：

- a) 上传：用户可以提交文件上传请求，在被服务器处理后就将存在于网络之中。
- b) 下载：用户可以提交文件下载请求，并直接从最近且存有该文件的服务器中下载文件。
- c) 加密：用户在上传文件时可以选择加密/不加密，也可标记为私有/公有。私有文件将不会直接加入公共文件索引之中，并仅会出现在用户的私有文件索引之中。
- d) 共享：用户可以将私有文件通过指定链接进行共享。公有文件则可从网络中直接访问，不需要共享。
- e) 冻结：长期未接收到下载请求的文件将会暂时冻结。冻结后的文件将以高度压缩的形式分块冗余存在于网络之中。冻结后的文件仍可下载，但冻结后第一次下载该文件需要额外提交一次合成文件的请求，合成之后文件将以原有形式存在于网络之中。

3. 激励系统（积分）：

- a) 用户提交请求时可以为请求附加积分，附带积分的请求将被服务器更快处理，而这些积分大部分将被服务器赚取，仅有 0.5% 将被基金会收取以维持其正常运行。
- b) 用户可以为文件本身附加积分，并设定每次下载时将会扣除多少积分。这种形式的文件下载时会扣除一定自身积分并发送下载请求，相当于提前支付了下载请求所需积分。在文件自身积分扣除完毕后，文件将重新成为普通文件。
- c) 为了防止服务器只处理附加积分的请求而忽视未附加积分的请求，服务器将永远保留一定带宽用来处理未附加积分的请求。该比例将由网络拥堵情况自动分配。
- d) 积分有三种获取途径：认证用户定期领取一定积分，积分发放将以年通胀率 2.5% 的方式进行；用户也可作为服务器提供者在处理请求时获取请求附加的积分；用户还可接受其他用户主动发送的积分或是向其他用户发送积分。

3.3 领域描述

认证服务领域：登录、DID 身份认证。

文件系统领域：响应请求、提供文件下载、文件存储/共享/冻结、自我冗余

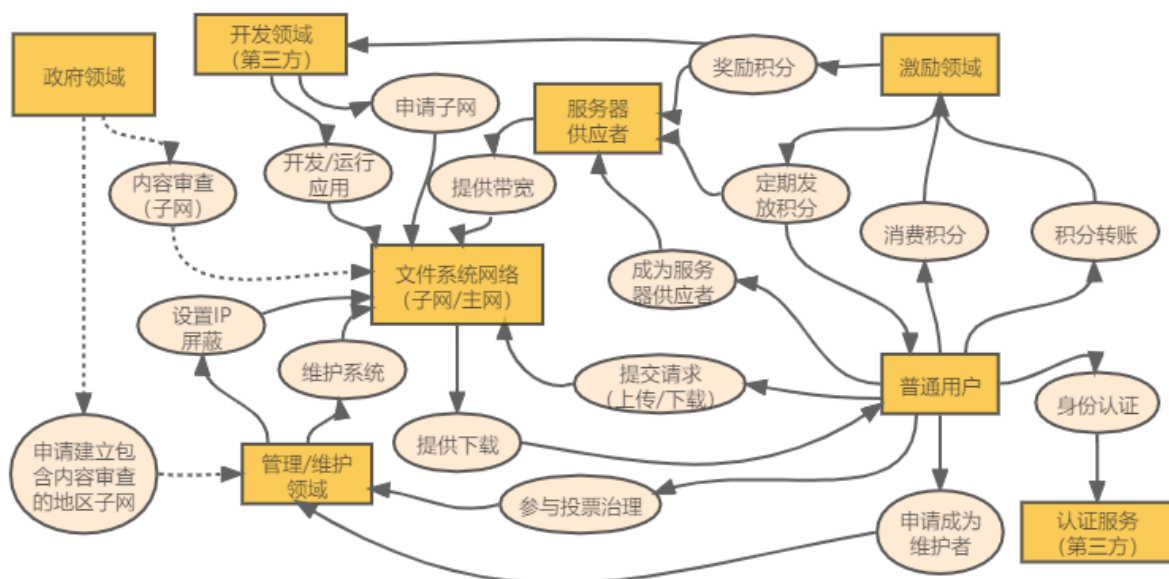
激励领域：积分获取、积分转移与交易、积分消费。

开发者领域：开发/运行应用、开发者子网。

管理/维护领域：社区治理、代码维护及更新、IP 地区屏蔽。

政府领域：内容审查（子网）。

3.4 问题框架



3.5 非功能性需求

1. 隐私安全性:
 - a) 一切上传文件在指定加密时都将通过格密码进行高度加密，以对抗量子解密算法。
 - b) 一切文件都将以分块形式在网络中存储，以防止服务器提供者直接查看文件内容。
 - c) 一切账户都必须通过严格的 DID 方式登录。尽管账户本身的认证全部由第三方 DID 服务处理，网络本身并不存储账户信息。但一旦某个被认可的第三方 DID 服务出现安全事件，基金会将会立刻冻结所有通过该方式认证的账户，并强制要求通过其他方式重新认证。
 - d) 积分转移记录将全部存储在一条独立的区块链公链之上，以保证其不可篡改。
2. 存储安全性:
 - a) 一切文件将以冗余形式同时存在于多个专业服务器之中，并且会有大量索引同时存放在专业服务器与业余服务器之中。任何服务器都将保有一定带宽用于进行文件冗余。
 - b) 在极端网络条件下，网络也会优先确保文件的永久存储。
 - c) 在极端 DDoS 攻击下，网络也不会丢失任何文件。
3. 一致性:
 - a) 网络中以冗余形式存放的任何同一文件都将尽可能保持一致。在进行删除操作后网络将快速将该文件从网络之中清除，以保证网络中的文件系统一致。
 - b) 在文件上传后该文件会以高优先级迅速复制到整个网络。
 - c) 在极端 DDoS 攻击下，网络也能做到对文件索引的快速更新，以保持文件系统一致。
4. 可用性:
 - a) 网络将在保证隐私安全与存储安全的前提下尽可能提高响应速度。
 - b) 网络将尽可能减少未附加积分请求的等待时间，并尽可能减少附加积分请求的平均积分需求。

-
5. 可拓展性及开发人员友好：
- a) 网络将提供大量底层接口供应用开发者使用，并且这些接口的调用将是易用的。
 - b) 网络将尽可能适应一切网络应用的底层云存储需求，从文件分享网站到文字论坛，从视频网站到综合网站，甚至交易应用。
 - c) 在保证高度拓展性的同时网络同时会保证其安全可靠，尽可能减少底层云存储基于该网络的应用在该方面产生的安全问题。
 - d) 以该网络作为应用的底层云存储服务将不会消耗开发者太多成本，并且将能够以较低成本长久维持运行。
 - e) 积分系统也将提供一定数量的接口，为部署智能合约提供可能。
6. 可维护性：
- a) 网络本身将是易于维护的，并且代码中将尽可能减少继承，转而大量使用组合以减少过耦合，提高代码可读性与性能。大部分代码将使用面向对象的标准设计模式，部分代码则会采用函数式方式编写。
 - b) 网络的底层处理方式，例如加密和传输，将是易于更新与替换的。基金会将会定期对这些逻辑结构进行修复及更新。此外基金会还会定期对加密模块进行重大更新，以适应密码学的发展。

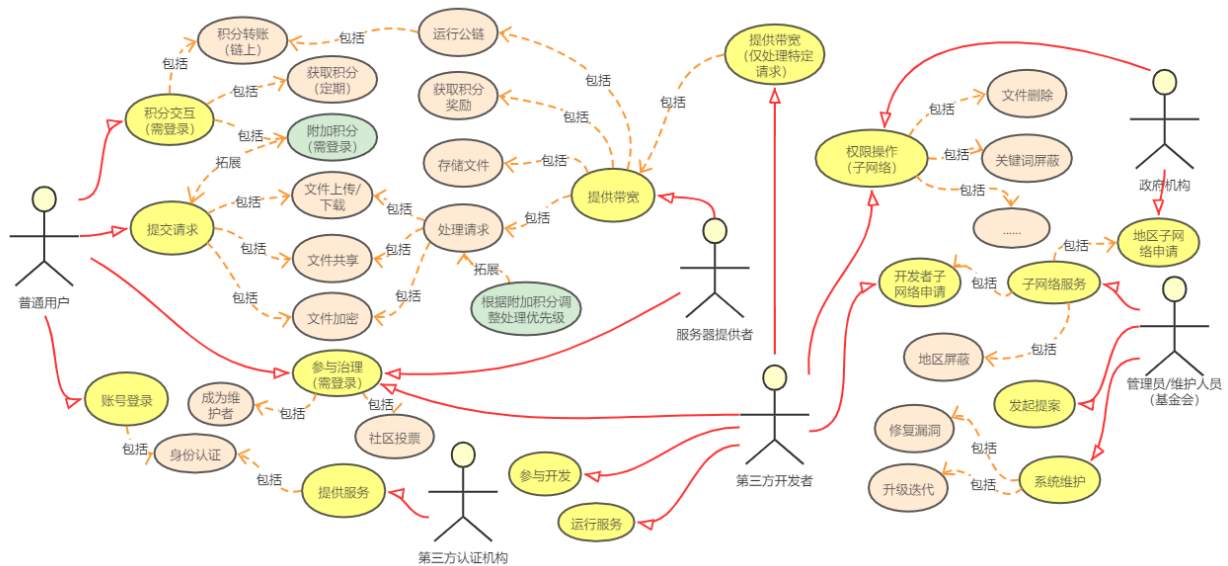
4 系统需求分析与业务建模

4.1 项目范围

1. 普通用户：
 - a) 所有用户可以在网络中自由上传与下载文件，并可选择是否加密。
 - b) 所有用户可以自行删除自己上传的文件。
 - c) 所有用户可以将文件通过指定链接共享。
 - d) 所有用户上传的文件将永久存储在网络之中。
 - e) 登录用户可以通过附加积分提高请求的处理速度。
 - f) 登录用户将定期获得一定的积分奖励。
 - g) 登录用户之间可以进行积分转账。
 - h) 登录用户可以参与社区投票和加入基金会参与治理。持有大量积分的用户将更容易参与基金会治理。
2. 服务器提供者：
 - a) 服务器提供者可以通过提供带宽获取更多的定期积分奖励，同时可以获得请求中附加的绝大部分积分。但处理请求错误、延时将受到扣除积分的惩罚。
 - b) 服务器提供者同样可以参与社区投票和加入基金会参与治理。提供带宽较大者将更容易参与基金会治理。
3. 第三方开发者：
 - a) 第三方开发者可以调用网络中提供的接口参与开发。
 - b) 第三方开发者可以将自己的应用服务运行于公共网络中。
 - c) 第三方开发者可以提交开发者子网络申请，开发者子网络将定期消耗一定的积分运行。开发者子网络可视为一个局域网，开发者在子网络中拥有最高权限（若所在地区存在地区子网络，则拥有政府组织以下的最高权限），并且可以将自己的应用服务运行于子网络中。
 - d) 第三方开发者可以提供私人服务器仅处理网络中与自身利益相关的请求，或是将其作为开发者子网络的载体，此时便无需定期支付积分。但私人服务器不会获得任何积分奖励，即使用户在请求中附加了积分。
 - e) 开发者同样可以参与社区投票和加入基金会参与治理。优秀开发者将更容易参与基金会治理。
4. 政府组织：
 - a) 政府组织可以向基金会提交建立地区子网络与进行地区屏蔽的申请。申请通过后该地区将仅有地区子网络与受认可的开发者子网络可以运行，而公共网络将被屏蔽。
 - b) 政府组织在地区子网络中拥有最高权限，可以进行屏蔽匿名用户、查看发送者信息、删除文件、关键词屏蔽等操作。
5. 基金会：
 - a) 基金会负责网络维护，包括但不限于漏洞修复、升级迭代等。但基金会无权查看、修改、删除公共网络中任何涉及用户个人隐私的内容，这些要求将固定于核心代码中，受所有用户共同监督。
 - b) 基金会对子网络服务进行维护。在开发者无力支付积分后收回开发者子网络的所有权，并且为子网络使用者（开发者/政府）提供技术支持。
 - c) 基金会根据政府组织的要求，对特定地区屏蔽公共网络。

d) 基金会负责收集并发起提案，在用户投票通过后对提案内容进行实现。

4.2 用例图



4.3 核心用例及用例描述

4.3.1 文件交互

用例简述：用户进入网络后，提交文件请求，并选择是否附加积分（需登录），等待处理完成。

行为者：普通用户。

前置条件：用户以登录或匿名状态进入网络，并提交请求。

后置条件：网络成功响应用户请求。

基本事件流：

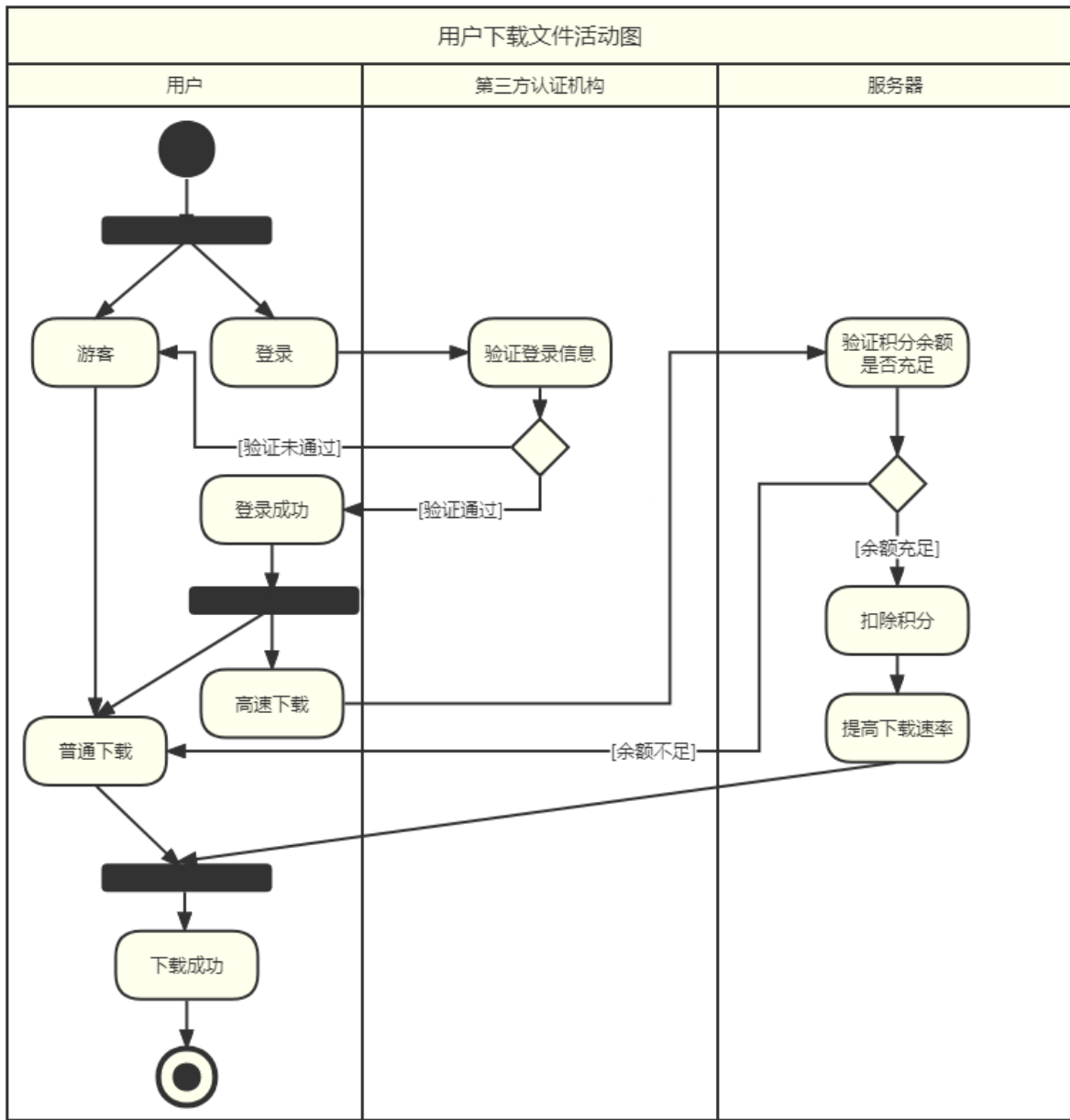
1. （用户需要附加积分时）用户选择使用的 DID 认证服务，点击确认，跳转到登录页面。
2. （用户需要附加积分时）用户输入账户信息，点击登录按钮。
3. （用户需要附加积分时）认证服务验证账户信息，正确后将提示网络跳转到主界面。
4. 用户进入主界面，选择请求类型（上传/下载/共享/加密），选择是否附加积分，点击确认。
5. 系统扣除用户积分（当附加积分时），并将请求广播到网络中。
6. 用户页面跳转到处理页面，实时显示处理进度。
7. 服务器接受并处理请求，经过多次确认后完成处理。
8. 用户页面跳转到成功页面。用户可查看详细信息以及获取返回值（例如下载请求将返回下载链接，共享请求则会返回共享链接）

备选事件流：

1. 用户输入登录信息错误，提示重新登录。
2. 用户中断请求处理，按当前处理进度收回一定积分。

异常事件流：

1. 附加积分不足/文件网络交互请求过多，请求处理超时。



4.3.2 积分转账

用例简述：登录用户进入网络后，填写接收人账户与数额，提交转账申请，等待处理完成。

行为者：普通用户（已登录）。

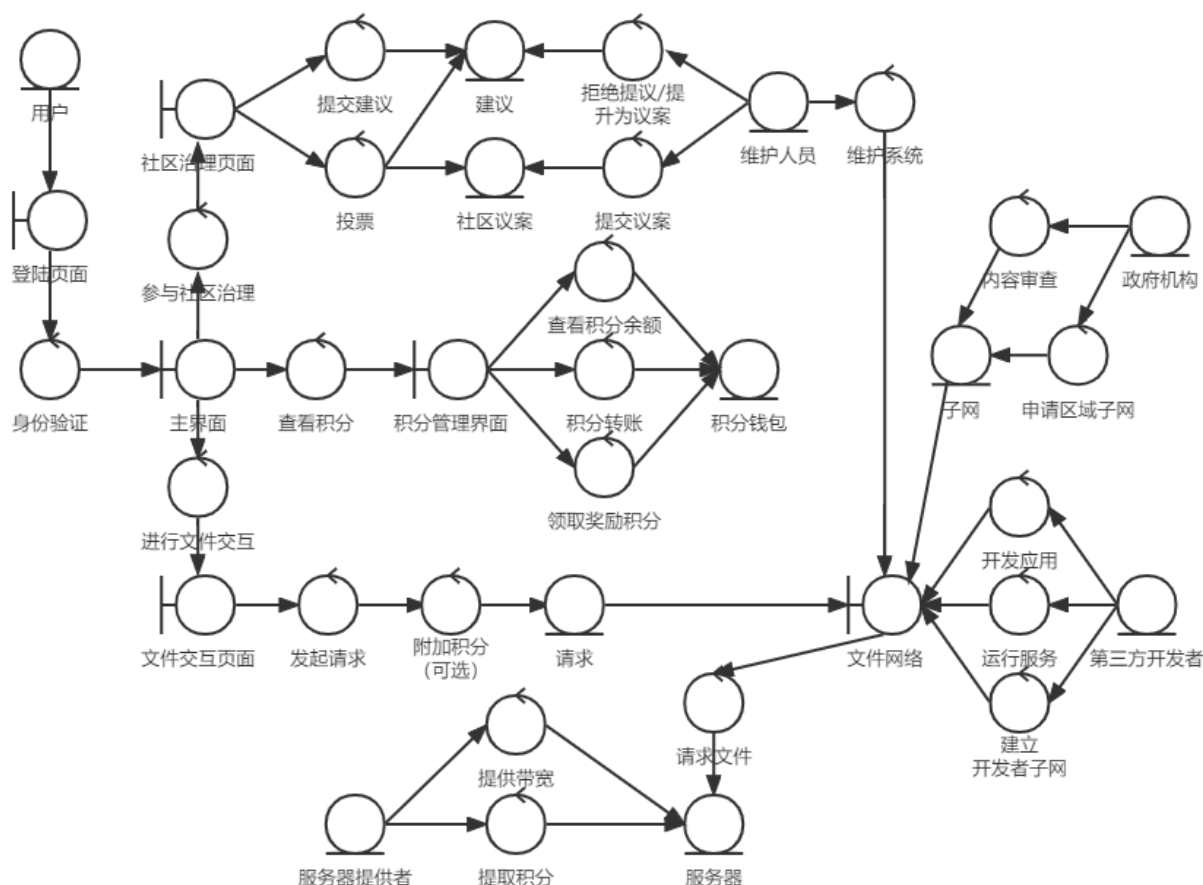
前置条件：用户以登录状态进入网络，进入转账页面。

后置条件：网络成功响应用户请求，转账成功。

基本事件流：

1. （已省略登陆操作）用户进入主界面，选择进入转账界面。
2. 用户填写接收人账户以及转账数额。
3. 系统验证接受账户是否存在以及转账数额是否超出账户余额，验证完毕后扣除相应积分及一定手续费，并在网络中广播转账请求。

4.5 分析类模型



4.6 系统架构

根据需求将系统分为四层：用户层、积分交互层、文件交互层、服务器层。

*注：项目本身仅提供底层的网络解决方案，因此不提供前端交互组件。

4.6.1 用户层

用户层负责用户与系统的基本交互

1. 账户组件：基于第三方认证服务提供的接口对用户信息进行验证，并进行登录。在账户不存在时自动创建账户。包含普通账户、维护人员账户、服务器提供者账户以及开发人员账户。
2. 积分钱包组件：查询积分信息，并负责处理积分领取、消费操作以及转账操作（向积分交互层提交请求）。
3. 文件管理组件：查询所有者为用户的文件信息（当用户权限为开发人员，请求网络为账户所有的子网时，文件范围为网络中存在的一切文件），并负责向文件交互层（主网/子网）提交请求。
4. 服务器管理组件：仅对服务器提供者开放，直接与服务器层进行有限的交互。查询账户所有的服务器信息，并负责向服务器层直接递交积分提取操作。
5. 子网管理组件：仅对开发人员及维护人员开放。可直接向服务器层提交请求创建子网。开发人员子网需绑定积分钱包，运行时会实时扣除积分，积分不足时将自动向服务器层递交删除子网请求。

4.6.2 积分交互层

1. 信息反馈组件：根据区块链信息响应来自用户层的信息查询请求。
2. 请求处理组件：验证用户层传来的积分交互请求，验证通过后提交至服务器层等待 12 次确认完成后录入区块链。
3. 积分区块链：将所有积分交易及消费信息保存在区块中。积分区块链本身存储于服务器节点上。

4.6.3 文件交互层 (子网/主网)

1. 信息反馈组件：根据用户层发送的权限信息以及所请求网络信息响应来自用户层的文件信息查询请求。
2. 请求处理组件：响应用户层发送的文件交互请求，根据请求类型的不同向服务器层递交不同请求，并返回处理进度。对于共享及下载请求，返回文件链接。
3. 自动冗余组件：当文件冗余数不足时，向服务器层提交复制文件请求。

4.6.4 服务器层

1. 文件索引组件：构建文件索引，标记同一文件在不同服务器中的备份。
2. 文件信息反馈组件：根据文件索引，返回来自文件交互层的信息查询请求。
3. 积分信息反馈组件：根据底层区块链信息，返回来自积分交互层的信息查询请求。
4. 服务器信息反馈组件：响应来自用户层的请求（服务器提供者），返回服务器信息。
5. 积分提取组件：响应来自用户层的积分提取请求（服务器提供者），并向积分交互层提交请求。
6. 服务器：保存文件，处理请求。每一个服务器同时还包含积分信息（即一种特殊的积分钱包），随处理请求的增多自动增长，可供服务器提供者提取至自己的积分钱包。
7. 积分请求响应组件：响应来自积分交互层的请求，修改区块链数据。
8. 文件上传组件：将文件存储在服务器上并更新文件索引。
9. 文件下载组件：在文件索引中找到标记为已同步的最近文件冗余，并返回下载链接。
10. 文件删除组件：在文件索引中找到指定文件，将其所有冗余标记为已删除。
11. 文件共享组件：在文件索引中找到指定文件，并将指定账户添加至文件所有人列表。
12. 文件加密组件：在文件索引中找到标记为已同步的最近文件冗余进行加密，并将其他冗余全部标记为待同步。
13. 文件请求响应组件：响应来自文件交互层的请求，查找最近的标记为已同步的文件冗余，并根据请求的不同对文件进行处理，实时返回处理进度。对于共享及下载请求，返回文件链接。请求处理完毕后立即更新文件索引，将其余文件备份标记为待同步。
14. 文件同步组件：实时查询文件索引，将标记为待同步的文件更新至标记为已同步的同一文件的其他冗余，并将标记为已删除的文件从服务器上删除并删除响应的文件索引。
15. 子网管理组件：根据来自用户层（开发人员/维护人员）的请求创建/删除子网。