YottaChain

# 重定义区块链存储

YottaChain 基金会 2018 年 11 月

V0.98

# 目　录

# 摘　要

存储是区块链最佳落地应用场景，YottaChain 是区块链存储的平台型公链。

YottaChain 利用独有的技术、商业模型、经济模型、生态策略和治理结构重定义了区块链存储行业，使得区块链存储可以突破桎梏发展到全新格局，并为其它区块链存储系统的发展提供关键作用。

在技术层面上，YottaChain 拥有对区块链存储至关重要的“加密后去重”独家专利技术、确保用户数据主权不受侵犯的数据安全专业能力和确保数据永不丢失的存储专业能力，使存储个人与企业数据所需的安全性和将数据存储空间扩大 5-10 倍的去重能力能同时满足，还将区块链存储应用领域扩大扩大了数倍，在规模达数百亿美元的市场上比现有中心化存储具有压倒性优势( 可靠性提升万倍、成本降低数倍，并额外增加抗 DDoS 和容灾等奢侈特性），从而颠覆了区块链存储的格局。

在经济模型上，YottaChain 采用贡献资源挖矿的模式，以及由资产背书的资源通证与总数固定的流通币构成的双层通证模式。通过巧妙的设计保证商业用户购买存储空间价格稳定在低价，但流通币价值会长期不断推高；

在治理结构上，YottaChain 第一次提出了一个完备的去中心化治理结构，解决彻底去中心化时“谁定规则”、“规则如何执行”、“出现任何一个人作恶或不作为时由谁来管理”等问题。

YottaChain 独有的激励模式使得存储资源拥有者将其硬盘空间贡献给 YottaChain 后，反而可以获得更多的存储空间，并得到额外的数字货币奖励，而且该模式无需任何补贴，是长期可无限持续的；

在商业模型上，YottaChain 具有强大的与企业用户对接的专业能力，包括技术上与中心化存储的应用无缝对接的专业能力、深刻了解市场需求和用户痛点的专业能力、与商业用户销售模式打通的专业能力和调动企业存储行业资源切入市场的专业能力，能够形成商业闭环，可直接无缝迁移现有中心化存储市场，兑现区块链存储的强大优势；

在生态发展上，YottaChain 一方面可直接迁移现有数百万种 IT 应用，另一方面提供开放平台将自有核心能力开放出来，并且让开放平台上的区块链存储系统可以共享去重红利，第三方区块链存储系统加入 YottaChain 生态即可获得关键技术能力，还能马上实现收入倍增。目前已经获得 IPFS 生态的主要力量支持。

# 1、背景

## 1.1.存储是区块链的最佳应用场景

### 1.1.1 什么是区块链存储？

区块链存储不是将数据存储在区块链上，而是去中心化存储加上区块链的激励，利用区块链的激励让更多的节点和用户加入到系统中，从而构建更可靠、更低成本和更大规模的存储系统。

### 1.1.2 存储自身有去中心化的需求

中心化存储的可靠性已经达到一个极致，光靠技术改良已经难以解决一些技术以外的因素对数据可靠性和服务稳定性的影响。例如 2018 年 8 月爆出的腾讯云彻底丢失用户数据事件（硬件故障加运维人员操作失误），2018 年 9 月 Microsoft 的 Texas 数据中心停止服务长达 20 多个小时（因为雷电击坏了制冷设备），以及 2017 年 AWS 的对象存储服务故障（运维人员操作失误）和支付宝停止服务事件（光纤被挖断），更不用说 911 直接摧毁了很多大型公司的总部。

为了提高持久化存储的可靠性，需要建立分散在全球各地的存储节点，数量越多越分散则数据可靠性更高；

为了网络加速，也需要去中心化，建立分散在全球各地的 CDN 节点，数量越多离终端用户越近则网络加速效果越好。

### 1.1.3 数据去重的放大效应

数据去重( 本白皮书后续章节对该技术有详细论述 )是存储的一项关键技术，该技术可以放大数据存储空间，其特点是用户越多数据越多则放大效应越大，即同样的存储空间可以存储更多的数据。

### 1.1.4 存储可以直接 TOKENIZE 上链

被在区块链其它应用场景中，例如溯源，区块链只能保证上链后的数据不会被篡改，却无法保证上链的数据是真实的。相反，存储既是物理世界的（指在现实社会中被大量使用的）又是数字世界的（指无需第三方机构即可直接被区块链程序所处理），可以直接通证化（Tokenize）后上链。

### 1.1.5 区块链+存储的化学反应

利用区块链的激励作用，可以无需巨额投资即可迅速在全世界招募众多矿工节点加入区块链存储系统并吸引大量用户使用，很快形成规模，并且节点数量多、地理位置分散、离终端用户近、用户多、数据多，从而提高存储的品质、增大存储空间、降低成本。

反过来说，由于存储自身的需求和特性，可以更加突出地体现出区块链的价值。因此，区块链存储不仅有实际的应用场景、刚性的市场需求和巨大的市场空间（每年将近 1000 亿美元），而且也是区块链的最佳应用场景。

例如，Airbnb 作为去中心化的酒店，很快就成为了世界上最大的酒店，超过了希尔顿等历史悠久的中心化酒店。Airbnb 是花费了巨额的营销费用实现这一点的，如果借助区块链的激励，去中心化存储也有望超过 AWS/Google 成为全球最大的存储池。

### 1.1.6 区块链存储的用户价值

在去中心化应用(DApp)发展起来之前，区块链存储可以从现有的中心化存储争夺市场，在区块链应用中是非常罕见的可以靠品质和价格与中心化应用争夺市场的。

对用户来说，区块链存储作为持久化存储使用的话，在数据可靠性和服务稳定性方面都比中心化存储有万倍以上的提升，还包含了非常昂贵的容灾和抗 DDos 特性，价格只有不带容灾和抗 DDos 特性的中心化存储的几分之一。换句话说，品质超过奢侈品，价格低于山寨货。

区块链存储作为持久化存储使用也有一个缺点，就是性能较差，主要体现在延时(Latency)指标上（系统总吞吐量指标可以靠节点数量堆砌），这是由于存在较大的网络传输延迟造成的。尽管如此，并不妨碍区块链存储在持久化存储领域具有压倒性的优势，这是因为存储本来就是分层的，每一层作为下一层的缓存，越往上性能越好、单价越贵、容量越小，越往下性能越差、单价越低、容量越大。区块链就作为最下面一层的持久化存储层，往上还可以有好几层中心化存储作为本地缓存（硬盘、SSD、3DXPoint、内存、L3 缓存、L2 缓存、L1 缓存等）。这种分层存储体系本来就是已有的，有区块链存储后无非就是多增加最可靠、最便宜、容量最大也最慢的一层，例如从 7 层增加到 8 层。

在网络加速(CDN)领域，区块链存储由于节点数量多、离用户近也具有无可比拟的性能优势，而成本也同样更低。

目前，以 Dell-EMC/NetApp/HDS/IBM/HP 为代表的企业级存储，以及以 AWS/Google/Microsoft 为代表的云存储构成的中心化存储的市场是每年 600 多亿美元，而且全球数据总量是每 27 个月翻倍（以上数据来源 Gartner）。区块链存储在持久化存储和网络加速两个领域都比现有中心化存储具有压倒性的优势， 这两个领域的市场规模达到数百亿美元。

## 1.2 IPFS

星际文件系统 IPFS（InterPlanetary File System）是区块链存储的明星项目。IPFS 是一个去中心化存储系统，2015 年发布，其口号是“取代 HTTP”，其对应的区块链激励层 FileCoin 在 2017 年 ICO 时募资 2.57 亿美元，是当时最大的 ICO，预计将在 2019 年上线。

### 1.2.1 IPFS 解决的问题

#### 1.2.1.1 去中心化的存储

IPFS 提供了一个非常出色的去中心化存储机制，将无数个不可信任的节点连接起来，却形成了非常可靠的存储系统，这就像比特币将不可靠的节点连接起来构成了比银行更可靠的金融系统。目前 IPFS 还缺乏冗余编码机制，因此还存在数据丢失的问题，但按计划将在 FileCoin 开发时补上这个缺陷。

#### 1.2.1.2 健康可持续的挖矿方式

IPFS/FileCoin 的“挖矿”方法就是为生态系统奉献存储空间：谁提供的存储空间大、服务稳定、带宽高速、离中心城市近，谁就获得最多的 FileCoin 作为奖励。这种新颖的共识机制为社会的贡献者提供了奖励，并且贡献越多的人，拿的奖励也越多，从而解决了传统区块链的过渡消耗资源挖矿的问题，形成一种健康可持续的模式。

### 1.2.2 IPFS 的不足

#### 1.2.2.1 缺乏数据安全机制

IPFS 的底层没有提供数据的安全机制，任何人只要知道了文件的 Hash 就能任意访问该文件。这样的设计方式更适合存储网页等公开信息，而不适合存储个人数据和企业数据，因为个人数据和企业数据都希望以更安全的方式进行存储，而非公之于众。

实际上，IPFS 标志性的“取代 HTTP”口号也体现了这个无奈，即 IPFS 的设计是适宜存储网页等公开数据，而非个人和企业数据。

IPFS 建议在应用层通过文件加密解决部分数据的安全性问题，但这并不是解决数据安全性问题的根本方法。数据安全是高度专业的，很难将应用层做好，而且在应用层做文件加密也无法解决文件去重的问题，从而影响了整个系统的效率和成本。

#### 1.2.2.2 不支持动态网页

IPFS 被设计作为 HTTP 协议的取代者，通过去中心化的方式存储静态文件，但是目前互联网绝大多数网站大都采用动态网页技术，在缺乏计算能力的情况下，IPFS 协议就不能被用作网站访问的直接入口，而只能作为动态网站的底层文件存储协议，但这与 IPFS 协议的设计初衷并不一致。如果浏览器把 IPFS 协议作为访问网站的入口方式，IPFS 需要在底层支持动态网页的处理机制，这就必须有计算能力作为 IPFS 的支撑。

#### 1.2.2.3 数据可靠性不够

由于数据不加密，为了伦理问题，IPFS 设计成每个存储节点只有主动 ping 才能获取该文件的副本( 以免暴力色情违反宗教信仰等该存储节点所有者不愿意接受的文件进入该节点），也就是说一个文件上传后如果没有其它节点 ping 的

话其实全网仍然只有一个副本，很容易丢失。这个机制可以保障热点文件（例如热门音乐）有很多副本不会丢失，但冷门文件就可能会丢失，从而失去了做持久化存储的可能性。

#### 1.2.2.4 服务稳定性不够

IPFS/FileCoin 对所有节点不加区别地按照统一的激励算法进行激励，导致大量的无法保证稳定服务的个人节点混杂其中，这将会拖累整个 FileCoin 体系的服务质量。FileCoin 为了应对这些问题，采用了抵押惩罚机制，并且另其他节点可以在节点离线时可以重建丢失数据，但是这也会势必影响 FileCoin 的商业级交付质量。

## 1.3.数据加密与数据去重

### 1.3.1 数据加密

对于大公司（例如 AWS、Google、Dropbox）提供的中心化存储来说，数据加密是一个亮点，但不是必须的。因为用户可以信赖大公司的品牌、内控体系，指望大公司不会作恶。虽然事实上这一点并不是那么可信，不管是百度的诸般作恶还是 Facebook 的剑桥门事件，都暴露了其中的风险。

对于去中心化存储来说，数据加密就成为存储个人和企业数据的必备要求。因为去中心化存储的节点本身不可被信任的，此外，源代码是开放的，而且每个存储节点都可以自由访问。如果不对数据进行加密，那么去中心化储存只适合存储网页这种公开数据，不适合存储涉及隐私/商业秘密的个人或企业数据。所以通用的区块链存储都必须做数据加密，而且是“零知识”的数据加密，即除了数

据所有者或其授权者外，其他任何人（包括存储节点的拥有者、系统的设计者和开发者）对该数据都一无所知，即使作恶也无法窥视数据。

### 1.3.2 数据去重

如果多个人拥有相同的数据，不重复存储而是合并共用同一份空间，称为去重（即去掉重复数据），也称为重删（即重复数据删除）。

数据去重和冗余存储是不同层面的概念。即使是去重后只存一份数据，这份数据也必须用冗余编码分成很多碎片，分别保存在多个不同节点上，即使其中有部分节点数据丢失也不影响数据的完整性。这么多个节点上存储的碎片合起来称为一份数据。

这两个概念之所以有时候会产生误导或混淆，是因为有一种最简单的冗余算法是多副本存储，例如 IPFS。这种情况下，多个用户拥有的相同数据会通过去重而只保存一份，但这一份是有多个副本。

数据重复率与用户数和数据量呈正相关：若用户数越多，数据量越大，则重复率越高。根据一个可参考的数据，360 云盘的平均数据重复率是 5 倍。360 云盘只是单一应用，并且不是规模最大的应用，那么整个区块链存储的数据重复率将远远超过这个倍数。

数据重复率越高，平均存储成本就越低。如果平均数据重复率是 10 倍，则 1GB 空间平均可以存储 10GB 的数据，平均存储成本降低 10 倍，从而构成了区块链存储的强大竞争力。

除了大幅度降低成本外，区块链存储还能利用数据去重特性构建强大的激励模型。假设，一个拥有 100GB 存储空间的人，如果用来存自己的数据只能存

100GB，但如果将该存储资源贡献用于挖矿，再利用挖到的数字货币购买存储空间，将可以储存 200GB 的数据，并且富余很多数字货币。这种方法可以储存更多的数据且让人获得很多额外的数字货币，可以有效地激励存储资源的拥有者加入系统挖矿。而整个过程无需补贴，系统甚至还可以收税，是长期可持续的。这种“魔法效应”的奥妙就在于 100GB 的空间平均可以存 500GB 甚至更多的数据。

### 1.3.3 数据加密与数据去重是“鱼与熊掌不可兼得”吗？

如前所述，零知识数据加密和数据去重都对区块链存储起到决定性的作用。但在行业中却存在一个“公知常识”：数据加密后不能去重。即零知识数据加密与数据去重二者不可兼得，最多只能选一个。

有人认为这是因为数据加密后就变成乱码，无法识别数据重复。这其实并非问题的关键，完全可以保存数据明文的 hash 值，通过比较 hash 值来识别重复数据，这并不会对数据安全性有任何的风险。

加密后去重的核心问题在于数据的授权。即 A 存储的数据，当 B 也要存储相同的数据时，如何将 A 的数据授权给 B 使用而且还不影响 A 的数据安全性。通常这个问题被认为是无解的，所以零知识数据加密与数据去重二者只能选一个。

在这种情况下，IPFS 选择了数据去重，牺牲了数据安全性，这就是 IPFS 设计用于存储网页等公开数据的真正原因。IPFS 提出在应用层做数据加密，实际上是让应用来承担数据不能去重的后果。还有一些区块链存储项目选择了数据加密，牺牲了数据去重，虽然保证了数据安全性，但存储成本大幅度上升，而且牺牲了一种极其有效的激励模型。

YottaChain 的创始人是一名知名的密码学和存储科学家，突破了常规思维，从数据安全和存储的最基本原理出发用严谨的科学研究方法发明了能实现“加密后去重”的 TruPrivacy 技术，从而颠覆了区块链存储的格局。

# 2、YOTTACHAIN 介绍

## 2.1. YOTTACHAIN 简介

YottaChain 是基于颠覆性的技术和深厚的行业资源打造的区块链存储公链，突破了 IPFS 的诸多局限，不仅为矿工提供强大的激励方案，为原中心化存储的用户提供端到端无缝衔接的高品质低成本持久化存储和网络加速解决方案，还制定区块链存储协议 BSP，打造区块链存储开放平台，为 DAPP 提供可靠、廉价、大容量、高性能的去中心化存储，为其它区块链存储系统提供核心能力并共享去重放大效应。

相对于 IPFS/FileCoin，YottaChain 的改进之处包括：

### 2.1.1 不影响去重的数据安全机制

TruPrivacy 是世界上唯一能实现“加密后去重”的技术，从而实现零知识数据加密与数据去重的“鱼与熊掌兼得”。2015 年，TruPrivacy 技术在全球最大黑客大会 DefCon 上公开悬赏验证，在敞开服务器任黑客自由出入并给黑客提供管理账户权限的前提下，全球顶级黑客联手都未能偷走服务器上存储的用户数据，无人领取高额现金奖励。

TruPrivacy 的全球专利均已正式授权生效，其技术细节可查阅各国相关专利文献（2015 年 10 月 20 日授权的 9164926B2 号美国专利，2016 年 4 月 13 日

授权的 2024272 号中国专利，2016 年 9 月 19 日授权的 2830282 号欧洲专利），在本白皮书的 5.2 节也有简要描述。

YottaChain 独家拥有 TruPrivacy 技术，将在继承 IPFS 现有存储设计的基础上，增加数据安全方面的机制，主要体现在三个方面：

1）对数据进行零知识加密，然后做数据去重，使得最终存储的是不重复的加密数据，没有权限的人（包括所在存储节点的拥有者、系统设计者/维护者）是绝对不能获知数据内容的；

2）实现文件的权限系统，按照文件的 Owner、所在 Group 和 Everyone 三个维度定义文件的访问权限；

3）在数据级实现文件的授权机制，使得文件只能被授权人打开，不管各个节点如何作恶（包括恶意修改代码）都无法突破授权机制。这种机制的可靠性和区块链一样，是用密码学为基础的数学公式来保障的。

采用 TruPrivacy 技术后，在保障数据安全的前提下可以实现数据去重，YottaChain 相当于成为一个“空间魔方”，即矿工投入 1GB 的空间，YottaChain 可以产生 5-10GB 的存储容量，这样就产生了资源供应者获得的数字货币的购买力超过其供应的资源的奇迹效应。拥有存储资源的人与其用来存自己数据，不如用 YottaChain 挖矿，用挖矿所得的数字货币再来买存储空间存数据，不仅可以存更多的数据还能富余一些数字货币。这种机制可以激励更多的人来参与挖矿，贡献自己的存储资源。

### 2.1.2 远超中心化存储的数据可靠性

- YottaChain 的持久化存储服务统一采用冗余编码，任何数据自动编码成分成 N（例如 100，具体数字将来由社区治理委员会确定）个碎片，其中只要有任意 M(例如 70)个碎片即可恢复出数据，然后将这 N 个碎片分别存储到 N 个存储节点中，每个节点保存一

个碎片，这样只要不同时有 N-M+1（本例中为 31）个节点失效就能保证数据完整不丢失

- 任意一个节点失效的时候，系统将会立即另选其它节点重建失效节点的数据，在本例中只要重建第一个失效节点完成之前不会有另外 30 个节点也相继失效，就可以保证数据永远不丢
- 各节点之间会互相监控、互相校验，任何节点一旦失效都能被快速发现
- 重建一个失效节点的数据时，会分成很多个节点同时重建以加快重建速度。例如失效节点上存储了 1 万个文件的各一个碎片，重建一个碎片平均需要 0.5 秒钟时间（主要是网络传输花时间），选 100 个节点参与重建，每个节点只需要重建 100 个碎片，平均 50 秒钟完成全部重建工作。只要 50 秒钟内不会有同一个文件的另外 30 个节点同时失效，该文件数据就不会丢
- 由于冗余性好且地理位置分散，不用担心因为硬盘损坏、个别节点运维失误（2018 年 8 月曝光腾讯云因为运维失误丢失用户数据）、雷电天气（2018 年 9 月，Microsoft 因为雷击导致部分地区的 Azure 服务停机 20 多小时）、停电、光纤被挖断、地震火灾等原因而数据失效
- 由于节点分散而且冗余性好，不怕 DDOS 攻击

### 2.1.3 更低成本

相对中心化存储，YottaChain 存储的成本要明显低很多，原因是：

- YottaChain 采用了 TruPrivacy 技术，可以安全实现数据去重，存储相同数据占据的硬盘空间减少 5-10 倍

- 绝大部分存储节点只有很少的存储设备，不需要专门的制冷系统（占数据中心耗电的三分之一甚至一半），靠自然通风即可散热，CapEx 和 OpEx 都大幅下降
- 家用存储矿机无需额外花费带宽费用，无需支付租房成本，家用电费也比工业用电更便宜
- YottaChain 绝大多数存储节点无需专业运维工程师驻场，每个节点都自动化运行而且一旦有意外故障失效会有其它节点自动顶上，节省了昂贵的运维费用
- YottaChain 大量存储节点都是利用闲置硬盘空间，属于沉没成本，边际成本接近零

### 2.1.4 无缝迁移中心化存储的应用的机制

YottaChain 创始团队是存储行业老兵，将提供与中心化存储二进制兼容的接口，包括但不限于块存储、NAS 存储和对象存储，使得中心化存储的应用无需重新开发、无需修改代码、无需重新编译，可以无缝迁移直接使用 YottaChain 存储。对这些应用来说，以为仍然在使用 AWS/EMC 等传统存储，虽然实际上已经切换到了 YottaChain 存储。

这样，现有所有中心化存储的应用都是 YottaChain 生态上的应用，其存储市场规模达到每年数百亿美元。

### 2.1.5 区块链存储的开放平台

YottaChain 秉承同行(hang)是同行（xing）的理念，视所有区块链存储系统（包括但不限于 IPFS）为同行者，共同打造区块链存储生态，既为各 DApp 提供强大技术支撑，也共同抢夺中心化存储的近千亿美元市场。

为此，YottaChain 建立一个区块链存储开放平台，将自己独有的核心技术开放给 IPFS 在内的业内同行。其它区块链存储系统通过一个区块链存储协议 BSP 对接 YottaChain 区块链存储开放平台，所有基于 BSP 协议的区块链存储系统获得如下价值：

1. 享有 YottaChain 的核心技术赋能，包括所有区块链存储系统都非常渴望的数据加密去重技术在内
2. 对接近千亿美元的中心化存储的应用
3. 共享数据去重的红利。支持 BSP 协议的区块链存储系统越多，数据去重对存储空间的放大倍数就越多，所有区块链存储系统都可以享受到该放大倍数。

YottaChain 的 BSP 协议将成为一个国际工业标准。YottaChain 的创始人是国际权威的工业标准组织 OASIS 的一个技术委员会主席，曾一手打造了中国第一个得到国际认可的软件国际标准，另外 OASIS 的 CEO 也是 YottaChain 的顾问团成员，将具备足够的能力和丰富的经验来打造该标准。

第三方区块链存储系统加入 YottaChain 生态后不仅获得缺失的关键能力，还可以共享“数据去重”的红利（数据去重的放大倍数遵循“用户数越多数据量越大则放大倍数越高”的规律），立即多挣几倍的钱。例如一个区块链存储系统有 1 万台矿机，10PB 存储空间（冗余之后的容量），自身的数据重复率是 2，

在利用 YottaChain 开放平台的加密去重能力后可以销售 20PB 的数据空间，与 YottaChain 生态共享“数据去重”红利后去重的放大系数提高到 5 倍，即可以销售 50PB 的数据空间。

### 2.1.6 确保数字货币长期增值的经济模型

为保证币值稳定，同时也提供市场化机制便于通过市场发现价格，YottaChain 采用双层货币模式。其中一层是市场化的流通币，另一层是资产背书的资源通证。资源通证严格锚定矿工贡献的资源，绝不超发，采用系统定价和去重系数自动增长的机制，保证资源通证稳定增值，是非常好的稳定币。流通币的价格以及与资源通证的兑换比例都是完全市场化的，允许适度的投机性以保持流动性，但总额固定的流通币对应越来越多的加入系统的资源，从机制设计上也能保证其长期价值是增值的。

### 2.1.7 去中心化的治理结构

YottaChain 第一个提出了完善的去中心化的治理结构。YottaChain 提出的 YottaChain 宪法将汲取人类社会经济学和政治学的理论和实践成果，从根本上解决区块链去中心化治理的问题，打造一个民主制衡、透明、自动化的平行世界，兼顾公平和效率。

YottaChain 宪法和规则的执行是由代码实现的。“Code is law”的准确表达应该是“Law is implemented by code”。确保新开发的代码完美地执行制定出来的规则，是 YottaChain 治理结构的重点之一。

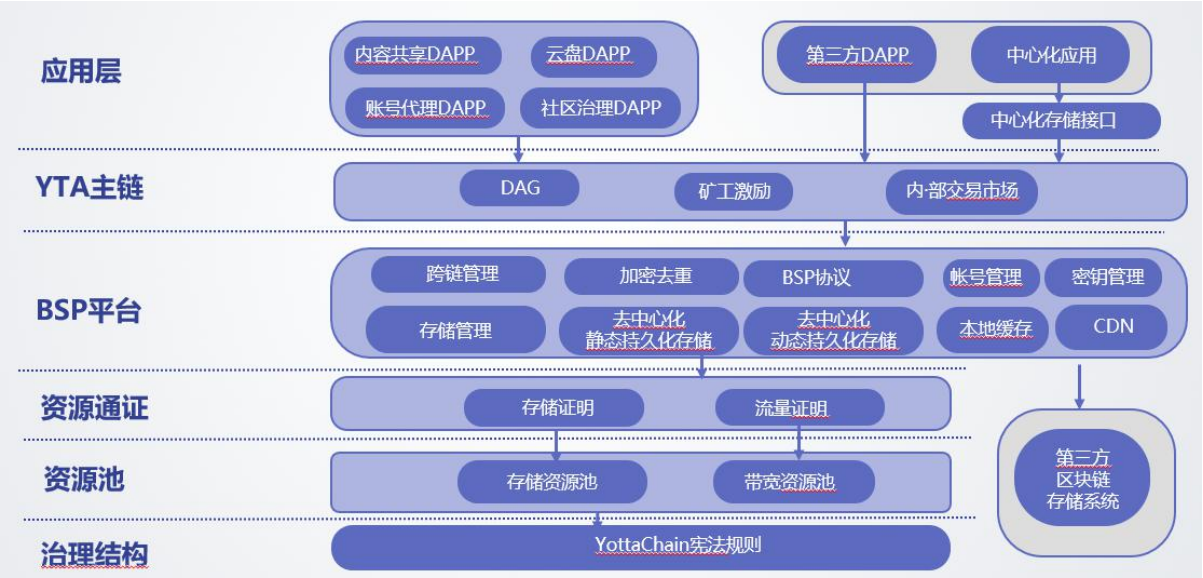
### 2.1.8 其它重要改进

除了存储服务服务外，YottaChain 还自带内容共享、云盘、账号代理和社区治理四个 DAPP。这四个 DAPP 既是 YottaChain 上的示范应用，同时也能对 YottaChain 的运营推广发挥作用。其中内容共享 DAPP 可以通过内容运营手段扩张 YottaChain 的规模；云盘是实用的 DAPP，可以让 C 端用户对 YottaChain 产生粘性和依赖性；账号代理 DAPP 提供不同安全等级和便捷程度的账号体系，并可以与其它账号体系（例如银行 U 盾、企业员工身份认证）打通，让用户可以单点登录；社区治理 DAPP 实现彻底去中心化的社区治理（见 2.1.7 节和第 8 章），使得 YottaChain 成为真正社区自治的公链，可以吸引生态相关各方加入。

YottaChain 将节点分为商业节点和普通节点。对商业节点有一定的门槛要求，例如必须 7x24 小时在线，节点规模、性能、可靠性、稳定性和行为规范达到基本要求，由商业节点提供存储和计算服务，从而保证服务的可靠性和稳定性；对普通节点则尽量降低门槛，保证普及性。

YottaChain 将用户分为个人用户、家庭用户和企业用户，提供企业用户的授权审核机制和家庭用户的透明机制。

## 2.2. YOTTACHAIN 系统结构




如图，YottaChain 商业节点包含六层。其中蓝底的是 YottaChain 自身要实现的，灰底的是纳入 YottaChain 生态的第三方系统。

最底层为 YottaChain 宪法规则，为整个系统提供基于代码实现的治理规则，主要功能是控制其它所有软件的更新。

资源池是矿工用于挖矿的各种类型的资源，包括硬盘、闪存、内存等存储资源池，以及带宽资源池。

资源通证层是将各类资源通证化（Tokenize）后发放资源通证的机制，能够验证矿工是否真实提供了相应类型的资源。对存储资源用存储证明（FileCoin 的 PoSt 算法的等效形式）机制发放存储类的资源通证，对带宽资源用流量证明机制。

BSP 开放平台为各区块链存储系统提供核心技术赋能，并为上层应用提供各类存储服务，包括至关重要的加密去重技术，去中心化静态持久存储、去中心化动态持久存储、本地缓存、CDN 等存储服务，存储管理、跨链调度等管理功能，还包括账号管理、密钥管理等数据安全机制。所有这些都通过 BSP 区块链存储协议开放给所有区块链存储系统和上层应用。

去中心化静态持久存储适宜存储静态数据，存储的数据采用如 2.1.2 所述的数据可靠性保障机制，可靠性远高于最好的中心化存储，是通过数据的 hash 值来访问的；

去中心化动态存储的数据可靠性与去中心化静态存储相似 ,但是需要通过 ID 来访问，一个文件无论内容发生多少变化，其 ID 都不会发生改变。同一 ID 的文件，新内容会覆盖旧内容；

本地缓存使用本地存储资源，性能较高但不能用于持久化存储；

CDN 适于网络加速，只保存最热门的内容，对没有命中的内容回源；

存储管理根据市场供需状况自动分配各存储服务所占用的存储资源 ,以保证矿工利益和用户利益的最大化 ,并根据各资源通证和第三方区块链存储系统的市场行情、当前平均数据重复率和税率自动计算各存储服务的报价。

跨链调度根据用户需求分配存储流量，评估各存储系统的可靠性，并在用户许可的前提条件下尽量做到跨链存储 ,以最大程度隔离故障域、提高系统冗余性，从而提高数据可靠性。

数据安全机制基于 TruPrivacy 技术，用于对持久化存储的数据进行加密和密钥管理，确保只有数据的 owner 及其授权者才能使用数据，其他人都无法看到数据；而且读写权限分离，可以分开单独授权；

BSP 协议模块通过标准的 BSP 协议对接各区块链存储系统，必要时还进行数字货币兑换（例如用 YTA 兑换 FIL 用于购买 FileCoin 的存储空间）；

YTA 主链实现基于 DAG 的区块链系统，通过 DPOS 共识算法（详见 3.3 节）实现对矿工的激励，并提供 YTA 双向兑换各类资源通证以及用户使用资源通证购买各类存储服务或计算服务的内部交易市场。

YottaChain 提供内容共享、云盘、账号代理和社区治理四个示范 DAPP 应用，同时这四个应用也是个人和企业使用 YottaChain 的基础和通用需求。第三方可以基于 YottaChain 开发独立的 DAPP 应用。尤其重要的是，YottaChain 通过中心化存储接口模块提供兼容中心化存储的块存储、NAS 存储和对接存储等存储接口以及相应的存储管理功能，使得现有中心化存储的应用可以不用修改代码、不用编译二进制兼容，可无缝迁移到区块链存储上。

## 2.3 BSP 协议与 BSP 开放平台

YottaChain 将自身独有的、对区块链存储起到决定性作用的核心技术通过 BSP（Blockchain Storage Protocol）协议开放出来，通过 BSP 协议为整个区块链存储行业构建一个开放平台，称为 BSP 开放平台。

一个区块链存储系统，通过对接 BSP 协议，可以享有 BSP 开放平台的全部功能，包括加密去重等对区块链存储起到决定性作用的独家技术，直接带来市场收益的对中心化应用的无缝对接，也包括非常专业的各种类型的存储服务，专业的密钥管理和账号管理功能等。除此之外，所有运行在 BSP 开放平台之上的区块链存储系统还都可以共享“去重效应”的红利，获得远大于自身的存储放大系数（这是因为用户越多数据越多则放大系数越高），从而可以销售更多的数据存储空间，多挣几倍的钱。例如，假设某第三方区块链存储系统有 1 万个存储节点，合计 10PB 存储空间，单独自己的去重放大效应是 1.8 倍，可销售 18PB 存储空间，加入 YottaChain 生态可获得 5 倍放大效应，销售 50PB 空间。

不管是 YottaChain 自身的存储系统，还是第三方区块链存储系统，对 BSP 开放平台来说都是一视同仁的，这一点可以通过开源代码来进行验证。而且作为一个彻底去中心化治理的公链，YottaChain 的功能定义、算法、代码实现和部署的全部过程都是公开透明的、社区自治的，第三方区块链存储系统甚至可以通过竞选 YottaChain 社区治理委员会参与管理。

调用 BSP 开放平台需要消耗一定的 YTA，但数量是很少的，相当于各种 ERC20 代币在以太坊平台上需要消耗一定的 ETH 作为 Gas。与 BSP 开放平台带来的收益相比，其比例是很低的，从而达到生态共同发展共同收益的目标。具体的比例系数在主链上线前由社区治理委员会制定。

# 3、通证和代币设计

## 3.1 概述

为保证币值稳定，同时也提供市场化机制便于通过市场发现价格，YottaChain 采用双层货币模式。其中一层是市场化的流通币，另一层是资产背书的资源通证。

YottaChain 只有一种流通币 YTA，这也是上各大交易所交易的数字货币（Cryptocurrency）。

YottaChain 为每一种资源（例如硬盘存储资源、带宽资源、x86 CPU 资源）都发行一种类型的通证(Token)，YottaChain 有内部交易所提供系统内各类资源通证与 YTA 的交易服务。各资源通证与流通币的交易价格是完全由市场的浮动来决定的。

YottaChain 的矿工贡献资源挖矿获得相应的资源通证，然后再兑换成 YTA。需要使用 YottaChain 系统内资源的用户购买 YTA，然后再兑换成相应的资源通证，购买相应的资源。

## 3.2 资源通证

YottaChain 针对每种类型的资源都发行一种通证，称为资源通证。例如，针对硬盘存储资源、闪存存储资源、内存存储资源、带宽资源种类型的资源，分别发行 4 种类型的资源通证。具体发行多少种资源通证，由社区治理委员会决定。

所有资源通证的发行量都取决于矿工贡献的该类型的资源数量。矿工用于挖矿所贡献的资源越多，则该种类型的资源通证发行量就越大，绝不超发。资源通证的发行量和用于挖矿的资源数量的对应关系是固定的。

对资源使用者来说，可以用资源通证来购买相应的资源。由于数据去重的因素，可供用户使用的资源数量是矿工贡献的资源的很多倍，从而大大降低了用户购买使用资源的成本，也构成了一个人人赢利的经济模型。

以下以硬盘存储为例说明 YottaChain 的资源通证机制。假设硬盘存储资源的通证是 YTA-HDD（具体名称在该通证创建时由社区治理委员会最终确定），一个矿工存储 1GB 数据一年可以获得 1YTA-HDD，

如果不考虑数据去重因素的话，那 1GB 数据存 1 年需要 1YTA-HDD。但加上数据去重因素后，就会产生非常特别的模型。

根据我们的调研，一个中等规模的云盘应用的数据平均重复率是 3 倍左右，一个大型云盘应用的数据平均重复率是 5 倍左右，使用的人越多数据量越大则重复率越高，因此我们可以预估 YottaChain 的平均重复率大约是 7-10 倍。也就是

说，如果整个系统存了 1EB 的数据的话，实际占用的物理存储空间大约在 100P-200P 之间。考虑到数据存储还需要做冗余编码，我们以 5 倍平均重复率为例说明（这是抵消了冗余编码带来的数据冗余率之后的数字）。

在 5 倍平均重复率的情况下，1GB 数据平均只需要 0.2GB 空间，即使加上交易费用，也只需要大约 0.25YTA-HDD 就可以买到 1GB 数据 1 年的存储空间。这就创造了一个魔幻般的激励效果：一个用户拥有 1GB 的硬盘空间，如果自用，则只能存 1GB 的数据，但如果用来挖矿，则将这部分硬盘空间贡献出来帮助他人存一年时间的数据，这样可以换来 1YTA-HDD 的资源通证，然后再利用其中的 0.5YTA-HDD 可以购买到 2GB 数据的 1 年存储服务，并且还剩余 0.5YTA-HDD。这不仅可以帮助他人，自己也会从中受益。这种模式是长期可持续的，系统运营方不仅不补贴还可以从中收点交易费用用于长期生态建设，充分体现了区块链模式的优越性。

一个矿工在贡献资源挖矿得到 YTA-HDD 之后，他可以用如下方式处理这些 YTA-HDD：

1. 用这些 YTA-HDD 购买更多的存储空间，储存个人的数据

2. 将 YTA-HDD 兑换成其他资源通证（以 YTA 作为中介），购买其它类型的资源（例如 CPU 资源）

3. 兑换成 YTA，再兑换成法币

4. 兑换成 YTA，持有 YTA，通过投票或竞选等方式参与 YottaChain 社区治理

以上是以 YTA-HDD 为例说明。无论是哪种资源通证，其价值都与对应的资源直接相关，保证能购买到相应的资源，永远不用担心价格归零，但也不会出现

短期内十倍百倍的增值，属于长期稳健增值的通证，随着时间的推移其购买力会越来越强大。

## 3.3 流通币

### 3.3.1 YTA 概述

YTA 是流通币，既是在各大交易所交易的加密数字货币，也是 YottaChain 体系内各种不同的资源通证之间互相兑换的中介。所有资源通证都可以在 YottaChain 系统内与 YTA 自由兑换 ,除了与 YTA 兑换外不允许在不同账户间转账。资源通证与 YTA 的汇率则是浮动的，完全由市场来决定。当购买某种资源通证的需求大于出售该资源通证的数量时，此时供不应求，则该资源通证的价格会上涨，反之则会下跌。不同类型的资源通证之间也存在类似情况，当某种资源通证过剩而另外的资源一种资源短缺时，二者之间的兑换汇率（以 YTA 为中介）也会发生变化。

资源通证的价值是相对稳定的，但由于 YTA 与资源通证的汇率是浮动的，所以 YTA 的价值也是浮动的。但是，YTA 的价格与 YTA 兑换资源通证的汇率是呈正相关的，也就是说，YTA 能买到的资源通证越多，其价值就越大，价格也会随价值的增大而相应上涨。

### 3.3.2 发行数量和锁仓

YTA 总共发行 50 亿，按照四个部分分配：

- 创始团队分配 10 亿 YTA
- YottaChain 基金会分配 15 亿 YTA

- 投资者分配 15 亿 YTA
- 系统上线后区块链记账节点奖励 10 亿 YTA

其中，YottaChain 基金会 15 亿 YTA 暂定按如下比例分配：

- 生态建设奖励 10 亿，包含初始矿工奖励，生态投资，以及必要时的市值管理（例如 YTA 持有者惜售导致矿工无法兑换 YTA 时）等用途
- YottaChain 系统开发与维护 1.5 亿，主要包含系统长期开发维护时给全球开源社区贡献者的奖励
- 社区治理 1.5 亿，主要包含社区各委员会成员的薪酬、持币者投票奖励等，这也是长期开支
- YottaChain 营销推广 1.5 亿，主要包含对外的营销推广，例如在 Google 买关键字销售 YottaChain 的优质服务，吸引更多用户加入和使用
- 社群维护 0.5 亿

在 YottaChain 主网上线前，YTA 会用以太坊 ERC20 智能合约的方式上线交易，待 YottaChain 主网上线再通过映射方式将 YTA 的 ERC20 代币映射到主网。

以 YTA 的 ERC20 代币上第一个交易所交易记为代币上市日。

创始团队锁仓：代币上市 12 个月后解锁 25%，之后每个月解锁 1/48，总共 48 个月全部解锁。在代币上市前，将从该部分中拿出总量为 0.5 亿 YTA 用于创始人以外的前期团队成员工资等用途，团队成员在领到该部分 YTA 后也要锁仓，锁仓方式为代币上市前解锁 20%，代币上市后每个月解锁 20%，4 个月全部解锁。

基石轮锁仓：基石轮共约 3.5 亿 YTA，其中的 3 亿 YTA 的锁仓方式为代币上市 3 个月后解锁 20%，代币上市 12 个月后解锁 80%；其余约 0.5 亿部分的锁仓方式为代币上市前解锁 20%，代币上市后每个月解锁 20%，4 个月全部解锁。

早鸟 IbO 轮：早鸟 IbO 轮共释放约 1.5 亿 YTA，其中的约 0.3 亿 YTA 的锁仓方式为代币上市前解锁 20%，代币上市后每个月解锁 20%，4 个月全部解锁；其余约 1.2 亿 YTA 的锁仓方式为代币上市后每个月解锁 1/6，6 个月全部解锁。

私募轮：私募轮锁仓方式为代币上市后每个月解锁 1/6，6 个月全部解锁。

YottaChain 基金会锁仓：其中不超过 0.5 亿 YTA 用于前期的社群建设等用途，该部分 YTA 的锁仓方式为代币上市前解锁 20%，代币上市后每个月解锁 20%，4 个月全部解锁；其余用于系统开发与维护、社区治理、YottaChain 营销推广和社群维护的约 4.5 亿 YTA 将按照代币上市 3 个月后解锁 12.5%，之后每 3 个月解锁 12.5%，24 个月全部解锁；用于生态建设的 10 亿 YTA 中初始矿工奖励部分将在系统主网上线后逐渐释放，释放数量主要取决于主网上线后存储矿工的总算力能力、用户总需求量和币价等多方因素；生态投资部分在可以预期的将来尚无规划，保留未来需要时使用。

投资者锁仓：代币上市前解锁 20%，自代币上市日开始，每个月解锁 20%，4 个月全部解锁。对于投资额度高的战略投资人，采用与团队或基金会相同的锁仓期。

YottaChain 团队已经开发出 YTA 自动锁仓的智能合约，前述锁仓机制通过智能合约强制执行。

### 3.3.3 共识机制和挖矿速率

YTA 采用 DPOS 共识机制，每年增发一定的 YTA 用于给记账的节点，其中超级节点 21 个，备用节点不超过 100 个。

YTA 上线后 60 年内的挖矿速率见下表：

| 时间区间 | 挖矿速率 | 本时段挖矿量 | 本时段结束时流通量 |
|---|---|---|---|
| 第 1 年 | 1 亿/年 | 1 亿 | 41 亿 |
| 第 2 年 | 0.9 亿/年 | 0.9 亿 | 41.9 亿 |
| 第 3 年 | 0.8 亿/年 | 0.8 亿 | 42.7 亿 |
| 第 4 年 | 0.7 亿/年 | 0.7 亿 | 43.4 亿 |
| 第 5-6 年 | 0.6 亿/年 | 1.2 亿 | 44.6 亿 |
| 第 7-8 年 | 0.5 亿/年 | 1 亿 | 45.6 亿 |
| 第 9-10 年 | 0.4 亿/年 | 0.8 亿 | 46.4 亿 |
| 第 11-12 年 | 0.3 亿/年 | 0.6 亿 | 47 亿 |
| 第 13-15 年 | 0.2 亿/年 | 0.6 亿 | 47.6 亿 |
| 第 16-18 年 | 0.1 亿/年 | 0.3 亿 | 47.9 亿 |
| 第 19-21 年 | 900 万/年 | 2700 万 | 48.17 亿 |
| 第 22-24 年 | 800 万/年 | 2400 万 | 48.41 亿 |
| 第 25-28 年 | 700 万/年 | 2800 万 | 48.69 亿 |
| 第 29-32 年 | 600 万/年 | 2400 万 | 48.93 亿 |
| 第 33-37 年 | 500 万/年 | 2500 万 | 49.18 亿 |
| 第 38-42 年 | 400 万/年 | 2000 万 | 49.38 亿 |
| 第 43-47 年 | 300 万/年 | 1500 万 | 49.53 亿 |
| 第 48-52 年 | 200 万/年 | 1000 万 | 49.63 亿 |

| | | | |
|---|---|---|---|
| 第 53-57 年 | 100 万/年 | 500 万 | 49.68 亿 |
| 第 58-62 年 | 90 万/年 | 450 万 | 49.725 亿 |

## 3.4 经济模型

资源通证的发行数量与矿工贡献的资源数量以及用户实际存储的数据总量相关。具体来说，对 YottaChain 的每个新矿工，YottaChain 都新发行少量的存储资源通证购买其存储空间作为库存，当该空间被最终用户购买并保存数据后，系统又新发行资源通证继续向该矿工购买空间，直到该矿工的所有空间都存满了数据。

之所以要设计一定的系统库存，是因为在主链刚启动的时候，还没有用户购买存储空间等资源，矿工手里就没有资源通证，这时在交易市场上就没有资源通证流通，用户也就不能用 YTA 来换取资源通证。为了打破这个怪圈，YottaChain 采取系统采购部分库存的方式解决该问题，即系统向所有矿工预先采购部分存储空间等资源作为向用户出售的库存（库存大小由社区治理委员会确定），这样矿工手里就有了部分资源通证，用户就能用 YTA 换到资源通证了，整个经济体系就运转起来了。

即便加上库存，YottaChain 发行的资源通证数量也是低于矿工所拥有的资源总量的，这样可以保证用户以资源通证永远可以买到相应的存储空间。

用户采用资源通证购买存储计算服务的价格是由系统来统一定价的，系统根据该服务需要消耗的资源数量加上适当的税收实时计算该价格。以持久化存储服务为例，其单位价格为(1+税率) / (挖矿难度*平均去重系数)，由于挖矿难度是逐年上升的，去重系数则是随着用户数和数据量的增加而逐渐增大的，该机制保证了价格的相对稳定性，相同数量的资源通证可以买到的存储空间是稳定上升的

（即单位存储空间的价格是稳定下降的）,有利于用户对采购价格有稳定的预期，有利于用户单位的预算管理，同时社区还能收取适当的税收保证社区有足够的资源继续发展。其中税率是由社区治理委员会来规定的，比例将远小于用户所获得的收益。

通过以上机制设计，既能让矿工不断挣钱，又能让用户以稳定低廉的价格采购存储和计算服务。

从经济模型来说，YottaChain 存储提供了远超中心化存储的品质（数据可靠性、容灾、抗 DDos），成本还大大降低，从而构成了矿工和用户赚取的利润空间来源。

## 3.5 稳定性与流动性

YottaChain 自有资源通证的发行数量与矿工贡献的资源数量呈硬性关联，绝不超发，价值相对稳定，抵消硬件成本下降的因素后依然拥有可观的收益，适于需要使用对应资源的用户，也适用于风险厌恶型的稳健投资人，比理财基金的收益率更高，是可靠稳定的数字货币。

流通币与实用资源并不直接挂钩，可以通过调节流通币与资源通证的汇率来调节流通币对应的价值。从短期来看，该机制会带来较多的投机性，但从长期来看，也能保证增值。这是因为 YTA 的总量相对恒定（除了每年奖励记账节点新发行的少量 YTA 外），随着时间的推移，系统中的资源越来越多，平均每个 YTA 兑换的资源通证就越来越多，价值也会相应地增长。因此，从短期来看，YTA 的价格受供求关系、市场操纵炒作等因素影响较大，但从长期来看，一定会大幅度增值，永远没有归零的风险。

为保证 YTA 和资源通证的双向交易的流动性，必要时 YottaChain 基金会将在内部交易所做市，保证矿工挖矿得到的资源通证可以换成 YTA，用户购买 YTA 后可以换成资源通证。YottaChain 基金会可以利用所掌握的 YTA 和收取交易费获得的资源通证来实现这一点。

### 3.6 区块链存储生态

对于在加入 BSP 开放平台的区块链存储，其发行的币种相当于 YottaChain 的一种资源通证，与 YTA 有一定的汇率关系，相互之间的资源调剂都是通过 YTA 作为中介进行。

对于 IPFS 或其它的大型区块链存储系统，可以开发一个符合 YottaChain 平台协议的对标系统，在 YottaChain 上发行相应的资源通证，该资源通证与对标的系统的数字货币采用侧链等方式 1:1 锚定。

通过以上方式，可以在 YottaChain 上形成庞大的区块链存储生态，囊括市场上所有的区块链存储系统。

## 4、YOTTACHAIN 账号管理

### 4.1.概述

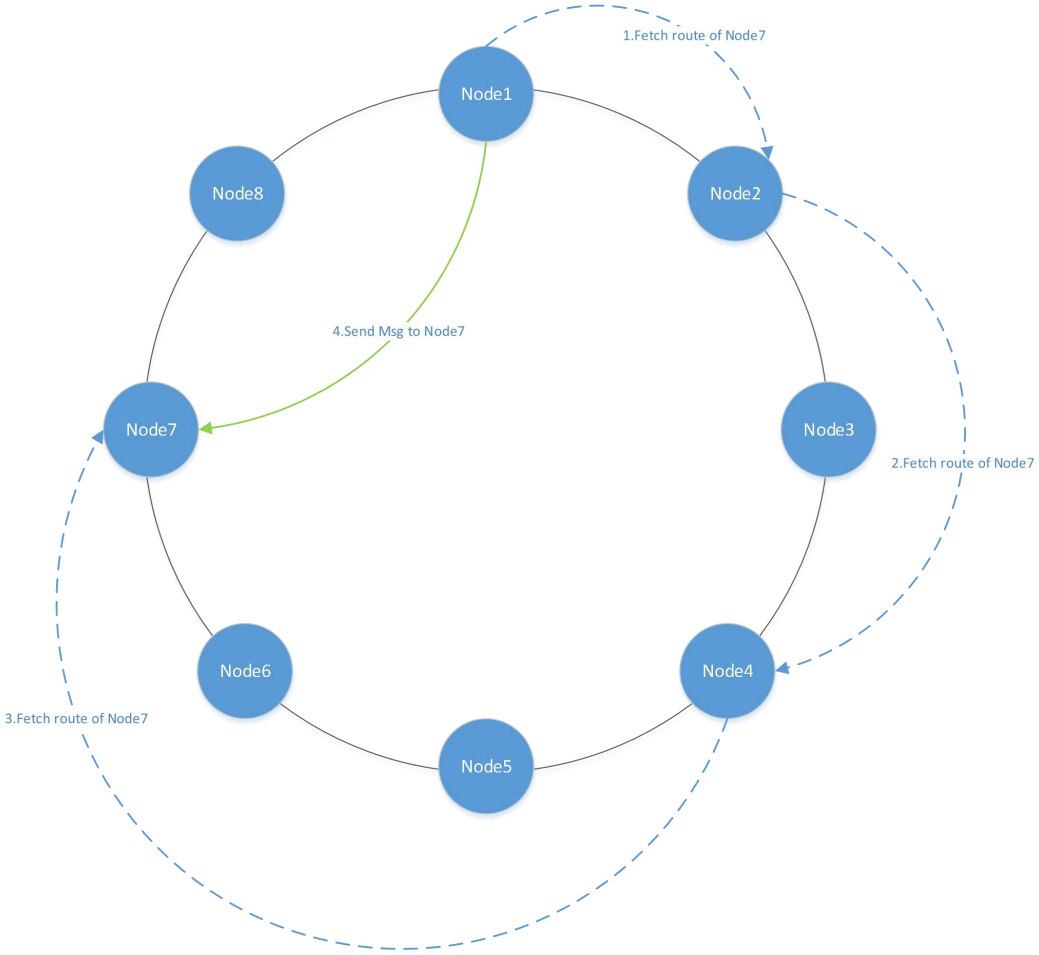
其他的区块链应用中往往使用公钥或其变体来唯一标识用户，但其可读性并不强。在 YottaChain 中使用可读性较强的账户名来对用户进行唯一标识。账户名由 10-32 个字符组成，账户关联着用户的 YTA 账户余额，除此之外还关联着用户的公私钥对，以及 P2P 网络中用户的路由信息。

### 4.2.账户创建

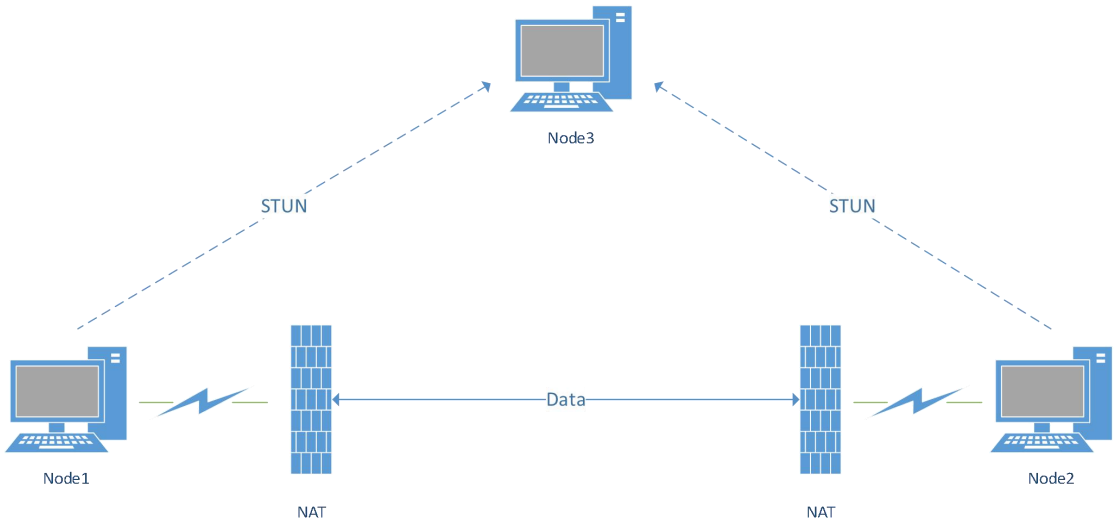
账户通过 YottaChain 账号代理服务创建。账号代理服务可以运行在普通节点上（此时只管理本机用户使用的账号），也可以运行在商业节点上（此时对公众提供服务）。创建账号时用户需提供 YottaChain 全局唯一的账户名，同时生成三对随机的公私钥对（一对登录密钥，一对签名密钥，一对加密密钥），然后将账号名和公钥写入 YottaChain 区块链。主干网络不负责保管私钥，由账号代理应用层负责。对于账号密码登录方式的，可以用密码加密私钥后存在账号代理应用自己的存储区，例如 YottaChain 的持久存储层。用户下次登陆时，账号代理应用验证密码后用密码解密出登录私钥，然后连接到任意一个商业节点，用挑战应答方式验证，在不出示私钥的前提下证明自己拥有该私钥。

## 4.3.消息机制

账户可以发送结构化消息给其他账户 ,消息分为预定义消息和用户自定义消息。预定义消息由 YottaChain 节点的内部机制处理（比如分享文件的消息），自定义消息可以由用户定义的处理器代码处理。消息在基于 YottaChain 节点构成的 Kademlia DHT 网络间进行路由，路由信息保存在 DHT 网络中的全局路由表中。由于 Kademlia 的查询高效性，一次路由查询的平均复杂度为 log2(n)，n 为 YottaChain 节点数量。下图中即为节点 1 向节点 7 发送消息的全过程，节点 1 在本地路由表中查询据节点 7 最近的节点，得到节点 2，节点 2 按同样的规则将消息转发给节点 4，最后到达节点 7，节点 7 将自己的地址通过之前的节点返回到节点 1，最后节点 1 和节点 7 建立连接并交换信息。




由于当前互联网中 NAT 设备的普遍存在，NAT 穿透技术是实现 P2P 网络重要的技术支撑。YottaChain 采用 ICE NAT traversal 框架以保证 YottaChain 节点在各种类型的 NAT 设备后方都可以正常进行相互连接。下图展现了 YottaChain 节点间在存在 NAT 情况下的连接方式。



节点 3 是一个位于公网上的 YottaChain 节点，节点 1 和节点 2 均位于 NAT 设备后，节点 1 想与节点 2 进行通信需要在节点 3 的帮助下使用 STUN 协议交换映射到公网上的地址和端口信息，并使用这些信息在 NAT 设备上打洞，打洞成功后节点 1 和节点 2 可以直接建立连接并交换数据。若打洞失败则需要通过 TURN 协议利用节点 3 中转节点 1 和节点 2 之间的通信。

## 4.4.群组管理

YottaChain 账号体系中有群组的概念，其类似于 Linux 操作系统中的用户组，每个用户可以创建多个群组，新建账号的默认群组为 Everyone。

每个群组在创建时同时生成该群组对应的公私钥对用于群组中的文件共享。群组信息和群组公钥保存在 YottaChain 区块链中，群组私钥保存在群组创建者的代理应用层中，群组成员信息以分布式形式存储在 DHT 网络中以便于查找。

当其他账号想加入该群组时则向群组创建者请求群组私钥，同时将账号和群组对应关系写入 DHT 网络中。

## 4.5.权限机制

数据的访问权限是通过密钥来控制的。采用一文一密的方式，数据用随机生成的对称密钥加密的（随机密钥的作用是保证该密钥没有其它任何人知道），该密钥称为该数据的存储密钥。对该数据有访问权限的人用自己的加密公钥来加密其存储密钥，然后保存在 YottaChain 系统区域。以后该用户需要访问该数据时，用自己的加密私钥解密出存储密钥，就可以解密该数据得到数据明文了。对于共享给群组的数据也是类似的机制，只是使用群组的密钥代替前述的用户密钥，属于该群组的用户是可以拿到该群组的密钥的( 方法是在一个用户加入一个群组时以用户的加密公钥加密群组的加密私钥，然后保存起来，以后该用户可以用自己

的加密私钥解密出群组加密私钥，就可以访问所有该群组有访问权限的数据了）。对于共享给 Everyone 的数据也是同样机制，将 Everyone 定义为一个系统设定的特殊群组（所有用户都在创建账号时自动加入该群组）即可。

对于动态存储，需要对写权限进行限制，否则任何人都可以去随意修改他人的文件，就会造成混乱的局面。该机制是这样的：在创建动态持久存储数据时，随机生成一对公私钥，分别称为写权限私钥和写权限公钥。拥有写权限私钥即可有权对该数据进行修改。为了验证这一点，写权限公钥作为该数据的元数据保存起来。在修改该数据时，必须用写权限私钥对新数据进行签名，所有保存了该数据的碎片的节点在接受到写请求时都要用写权限公钥来验证签名，验证通过才修改该数据。

# 5、YOTTACHAIN 存储系统

## 5.1.文件安全

### 5.1.1 需求

从用户需求角度，当用户选择一个存储介质存储自己的文件时，希望的是自己的文件是保密的，而不是完全公开的。存储系统本身也应该考虑到对自己存放的文件进行数据保护。从社会需求角度，对于宣扬极端恐怖主义等违反人类社会共同价值观的文件，也应该有手段屏蔽。

### 5.1.2 问题

目前在 IPFS 存储网络中文件的唯一索引标识 Hash，可以通过 Get(Hash) 方式获取文件的全部内容，而且无须任何的认证，且文件难以被销毁。既无法满足个人需求，也无法满足社会需求。

### 5.1.3.解决方式

YottaChain 采用数据源端对文件进行加密后进行上传，当文件开始进入 DSN 网络时候就已经是经过加密的了，而且除了 owner 或其授权者，其他人是无法解密的。

## 5.2.加密 DSN

加密的 DSN 方案协议：

在 Put 数据之前的处理，随机产生文件的存储密钥（$St_k$），通过存储密钥 $St_k$ 对文件进行加密，生成加密文件，以用户的加密公钥来加密存储密钥，分别计算数据明文的 Hash 值和密文的 Hash 值，然后将密文、加密后的存储密钥、明文 Hash 和密文 Hash 都保存起来。

1. Hash(Data) →$H_{data}$　计算明文 Hash
2. RandomSym()→$St_k$ 随机生成对称密钥作为文件的存储密钥
3. Enc($St_k$, Data)→EncData 用存储密钥加密文件
4. Hash(EncData) →$H_{enc}$　计算密文 Hash
5. Enc($S_{pub}$, $St_k$) →$EncSt_k$ 以用户的加密公钥加密存储密钥
6. PutStatic(EncData)　将加密数据存入到持久化静态存储中
7. PutPri($H_{data}$，$H_{enc}$, $EncSt_k$) 将加密后的存储密钥，密文 Hash 存入到用户权限列表，记录在明文 Hash 项下

在 Get 数据的时候，从明文 Hash 取出对应的密文 Hash，通过密文 Hash 从持久化静态存储中取出密文和加密后的存储密钥，以用户的加密私钥对加密后

的存储密钥进行解密获得存储密钥，用存储密钥对加密数据进行解密，获得数据明文。

1.GetPri($H_{data}$)→$H_{enc}$,$EncSt_k$ 从权限列表中通过明文 Hash 获取到密文 Hash 和加密的存储密钥

2. GetStatic($H_{enc}$)→EncData 用密文 Hash 从持久化静态存储中取出密文

3.Dec（$S_{prv}$, $EncSt_k$）→$St_k$ 以用户的加密私钥对加密后的存储密钥进行解密获得存储密钥

4.Dec($St_k$, EncData)→Data 解密获得文件数据

优化的 DSN 方案有效地保证了数据的安全性。

安全性：在数据源之外只以密文出现，只有利用用户的加密私钥才能获取到数据明文，用户只要保管好自己的加密私钥就不用担心数据被泄密。

完整性：Hash 对应的数据 D，不会存在通过 Get（hash）获取到 D1，其中 D1≠D。

数据可恢复性：Put 成功的数据 D，一定存在一个成功的 Get 请求获取到数据。

以上方案不能解决数据去重问题。传统上行业内都公认加密后不能去重，服务器端要想零知识加密的话，重复的数据就只能重复保存，这是因为相同的数据在加密后就变得不一样了。这就是所有的大型云存储服务商都不提供零知识加密存储的原因，IPFS 为此干脆都不提供加密机制。

YottaChain 提供了一种特殊的机制，既能防止存储重复数据，同时还能保证同样的安全性，打破行业“公知常识”实现鱼与熊掌兼得。采用这种机制时，

除了用户权限表外 ,还要维护一个全局的元数据表 ,记录明文 Hash 和密文 Hash 的对应关系，在写入数据时要先查询是否存在相同 Hash 的数据，如果没有该项再存：

1. Hash(Data) →$H_{data}$ 计算明文 Hash
2. If CheckDup($H_{data}$) = TRUE goto 11 如果已经存在相同的数据 ,转到第 11 步
3. RandomSym()→$St_k$ 随机生成对称密钥作为文件的存储密钥
4. Enc($St_k$, Data)→EncData 用存储密钥加密文件
5. Hash(EncData) →$H_{enc}$ 计算密文 Hash
6. GenKey(Data) →$S_{data}$ 从数据明文生成对称密钥 ,可以用数据明文加盐之后计算 Hash 值的方式生成。之所以要加盐是因为明文 Hash 是一个公开的值，不加盐的话不拥有数据明文的人也能获得该密钥。为了保证一致性，盐值可以是一个固定的算法生成，例如先做第一次 Hash 作为盐值，然后加盐后再计算第二次 Hash 作为对称密钥，两次 Hash 可以采用不同的算法。
7. Enc($S_{data}$, $St_k$) →EncSt$_k$’ 以数据明文生成的对称密钥来加密存储密钥。这是非常“诡异”的一步，以明文作为密钥，密钥作为明文来加密，大多数人看这个算法的时候都以为写反了，实际上就是专门这么设计的，而且这一步可是 TruPrivacy 的核心步骤。
8. PutStatic(EncData) 将加密数据存入到持久化静态存储

9. PutMeta($H_{data}$, $H_{enc}$ ,$EncSt_k$') 将密文 Hash 和明文加密的存储密钥记录在全局元数据表中，记录在明文 Hash 项下
10. Goto 14
11. GetMeta($H_{data}$) →$H_{enc}$ ,$EncSt_k$' 从全局元数据表中取出密文 Hash 和明文加密后的存储密钥
12. GenKey(Data) →$S_{data}$ 以同样算法从数据明文生成对称密钥
13. Dec($S_{data}$, $EncSt_k$') →$St_k$ 用该对称密钥解密出存储密钥
14. Enc($S_{pub}$, $St_k$) →$EncSt_k$ 以用户的加密公钥加密存储密钥
15. PutPri($H_{data}$， $EncSt_k$) 将加密公钥加密后的存储密钥存入到用户权限列表， 记录在明文 Hash 项下

在 Get 数据的时候 ,从全局元数据表中从明文 Hash 取出对应的密文 Hash，通过密文 Hash 中从持久化静态存储中取出密文，从权限列表中取出加密后的存储密钥，以用户的加密私钥对加密后的存储密钥进行解密获得存储密钥，用存储密钥对加密数据进行解密，获得数据明文。

1.GetMeta($H_{data}$)→$H_{enc}$ 从全局元数据表中通过明文 Hash 获取到密文 Hash

2. GetPri($H_{data}$)→$EncSt_k$ 从权限列表中通过明文 Hash 获取到加密公钥加密的存储密钥

3. GetStatic($H_{enc}$)→EncData 用密文 Hash 从持久化静态存储中取出密文

4.Dec（ $S_{prv}$, $EncSt_k$ ）→$St_k$ 以用户的加密私玥对用户加密公钥加密后的存储密钥进行解密获得存储密钥

5.Dec($St_k$, EncData)→Data 解密获得文件数据

优化的 DSN 方案不仅有效地保证了数据的安全性，而且还能实现加密去重。

为了更好地实现去重效果，可以将数据按固定长度分块，每块分别去重。

上述方案只能用于存储静态数据。当存储动态数据时，不仅要用不随内容变化的 ID 代替 Hash 作为数据的标识，而且还要加上写权限的验证以防止数据被其它人覆盖篡改。这时创建流程如下：

1. RandomAsym()→$Sw_{pub}$ ,$Sw_{prv}$ 随机生成非对称密钥作为写权限密钥
2. Create($Sw_{pub}$) →ID 创建一个动态数据，获得一个唯一的 ID，并记录该 ID 对应的写权限公钥
3. RandomSym()→$St_k$ 随机生成对称密钥作为存储密钥
4. Enc($S_{pub}$, $St_k$) →$EncSt_k$ 以用户的加密公钥加密存储密钥
5. PutPri(ID，$EncSt_k$) 将加密公钥加密后的存储密钥存入到用户权限列表，记录在 ID 项下

每次写数据时的流程如下：

1. GetPri(ID) →$EncSt_k$ 从用户权限表中取出加密公钥加密后的存储密钥
2. Dec（$S_{prv}$, $EncSt_k$）→$St_k$ 以用户的加密私钥对用户加密公钥加密后的存储密钥进行解密获得存储密钥
3. Enc($St_k$, Data)→EncData 用存储密钥加密文件
4. Hash(EncData) →$H_{enc}$ 计算密文 Hash

5. Enc($Sw_{prv}$, $H_{enc}$) →$EncH_{enc}$ 用该 ID 的写权限私钥对密文 Hash 进行签名

6. PutDyn(ID, EncData, $EncH_{enc}$) 写入加密后的动态数据，以签名数据代表写授权。

存储动态数据各碎片的节点在写入动态数据时，需要先验证写权限：

1. GetKey(ID) →$Sw_{pub}$ 取出该 ID 对应的写权限公钥

2. Hash(EncData) →$H_{enc}$ 计算密文 Hash

3. If Dec（$Sw_{pub}$, $EncH_{enc}$）= $H_{enc}$ Write(EncData) 如果签名验证通过，写入数据

动态数据的读流程如下：

1. GetPri(ID)→$EncSt_k$ 从权限列表中获取 ID 对应的用加密公钥加密的存储密钥

2. GetDyn (ID)→EncData 用 ID 从动态数据存储区读出密文

3. Dec（$S_{prv}$, $EncSt_k$）→$St_k$ 以用户的加密私钥对用户加密公钥加密后的存储密钥进行解密获得存储密钥

4. Dec($St_k$, EncData)→Data 解密获得文件数据

## 5.3.拜占庭容错

拜占庭将军问题出现的主要背景是：拜占庭罗马帝国地域广阔，各个军队相隔很远，军队之间的通讯只能通过信差，任何的战略部署都需达成统一后才能展

开行动。如果军队中将军或者信差有不可信的人存在就会扰乱作战计划，无法达成共识。因此在已知有叛军存在的情况下，如何意见达成统一就成为了拜占庭将军问题。

存储故障即称为拜占庭故障，即有不诚实不可信的矿工丢失了他们的数据，从而让文件无法获取成功。

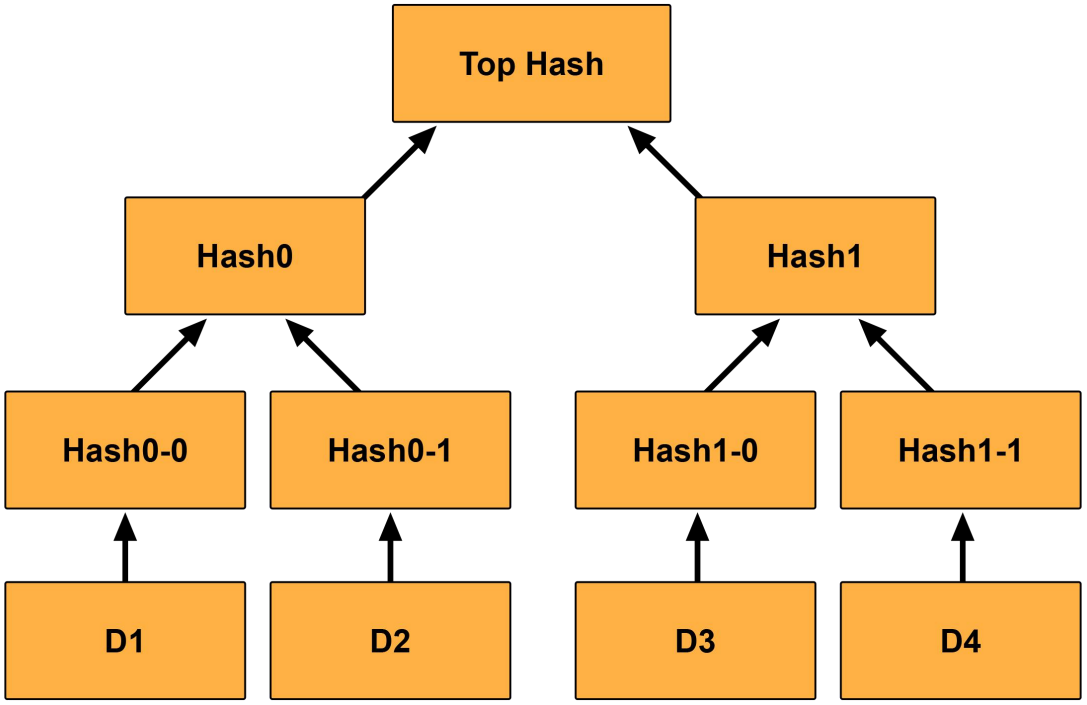
拜占庭容错方案：Put(D,n,m)，当数据上传的时候使用冗余编码将数据分为 n 个碎片，并允许最多 m 个碎片失效（即只要能任意获取 n-m 个碎片即可完整读取数据）指定 n 个存储节点来存储这部分数据，每个节点存储一个碎片，这样可以容忍 m 个节点故障。当故障节点<m 时，文件是可以 Get 成功的，这时候通过修复机制将重新选择新节点代替故障节点存储数据。

节点 n 的选择，以及容错节点 m 的选择用户可以自己选择，系统会默认设定一个值，其中 n>3m+1。

## 5.4.标准格式文件 STDFILE

具备自我描述功能，通过获取文件的头信息即可获得文件的相关信息。

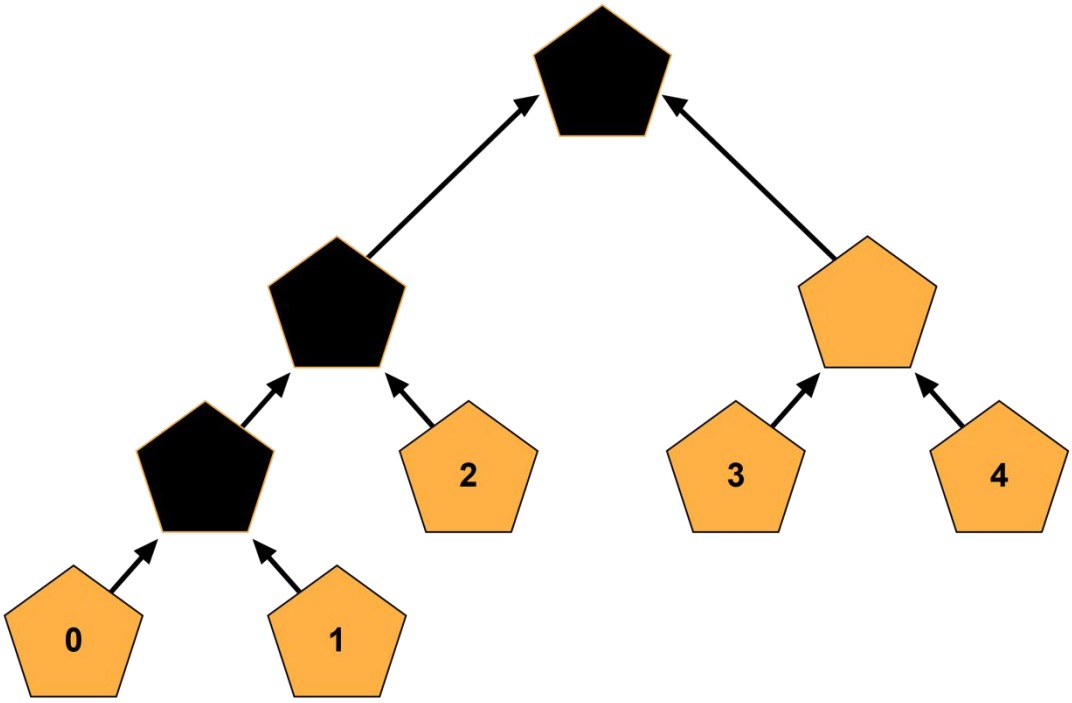
什么是文件头信息（HeadHash），先简单了解一下默克尔树（Merkle Tree）



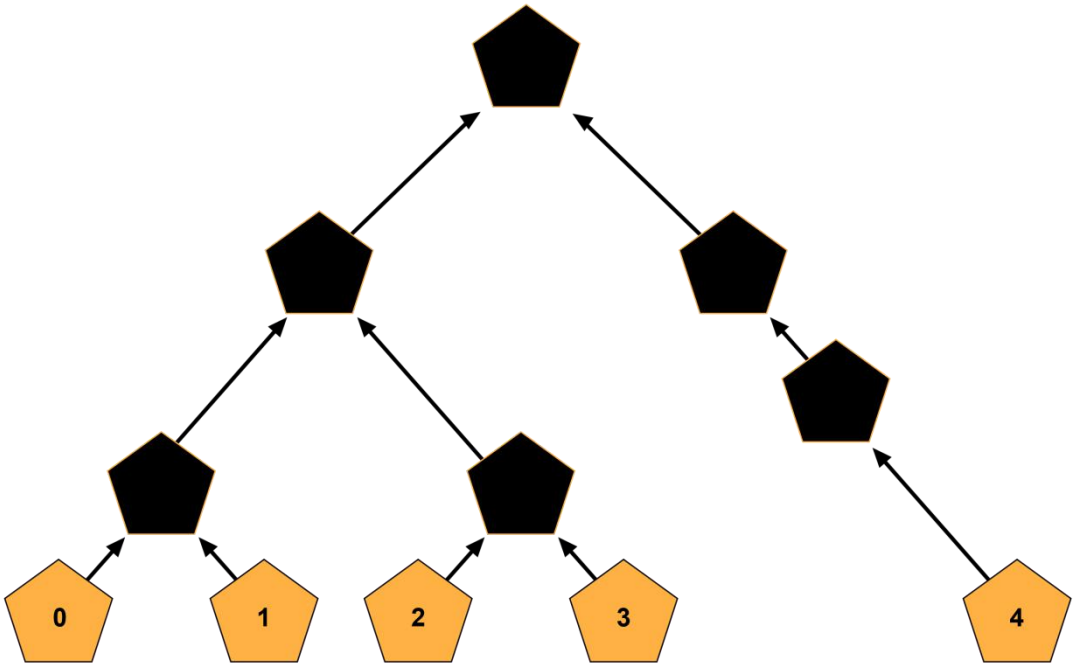

它是装载 Hash 列表的一个树形结构。在树的最底层是已经被切割的具有固定大小的数据小块（除去最右侧小块），　有相应的 Hash 与其对应 ,相邻两个小块合并再做 Hash，以此上推，至最上层的 Top Hash 也就是默克尔根。

文件索引：通过获取文件的 Hash（top Hash），来获取到叶 Hash，获取到叶子 Hash，最后获取到最小的数据小块。

Merkle Tree 插入




通过插入一个具有固定格式固定大小的数据块0( Data Block )来改变Merkle Tree，但是基本不改变结构关系。因为 1,2,3,4,5 的数据块内容未发生改变。经过插入后的 Merkle Tree 变为:




这里要研究的就是数据块 0 的内容。

因为在获取文件数据的时候，可以将 0 作为文件头 Head，>0 的数据块作为文件内容 Data，而数据块 0 就是文件的自我描述，它可能会包含类似文件编码格式、创建时间、文件格式、文件名的元数据。以固定的格式表述这些元数据并通过特定的转换方法 Convert（HeadInfo）将其转换为固定数据块大小的文件头信息（HeadBlock），连同文件内容（Data）一并被存储。

SFILE ： Hash（HeadBlock（Finalsize）+Data）-->Top Hash

HeadBlock（Finalsize），不会影响到数据在 Merkle Tree 中的整体结构，如果最底层数据块是偶数个，那个当 0 被插入的时候，会产生孤儿，但是孤儿永远是最右边（end）的一个。

SFILE 文件索引：从 Top Hash 开始查询叶节点，当查询到最底层数据块 Hash 时，你永远都知道第一个块儿是头信息块，它用来描述文件，并非文件内容的一部分，所以在组织数据时可以忽略掉。

SFILE 旨在创建一种具有自我描述功能的标准格式文件，从而实现文件的一些信息交换。

## 5.5.文件分享

在 YottaChain 系统中文件是加密存储的，是安全的，获取文件必须拥有文件的存储密钥才能解密。A 用户想要将自己的文件分享给 B 文件，则 A 文件需要进行的操作就是 YottaChain.share(EncD, $St_k$, ObjectB)将存储密钥分享给 B，B 才能获取到存储密钥 Stk 从而解密文件 EncD，但是这样就会存在在传递过程中泄露的风险。而且在 YottaChain 的密钥管理系统中，不存在文件的明文存储密钥 stk，而都是加密后的加密存储密钥 EncStk。这里存储密钥的交换用的是非对称加密算法交换存储密钥。

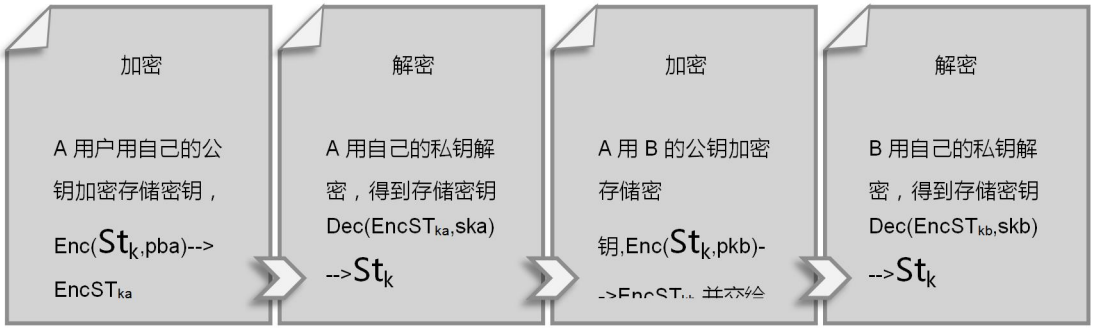
具体的构建过程如图

pka---A 用户公钥

ska---A 用户私钥

pkb---B 用户公钥

skb---B 用户私钥




通过账号管理的消息传递机制，A 将格式化的消息

YottaChain.setMes(EncHashD,EncStkb)-->Mes

发送给 B，

YottaChain.sendMes(A,Mes,B)-->MesID，

B 收到消息并利用自己的处理脚本

YottaChain.getMes(MesID)

获得消息内容，从而获得通过 B 的公钥加密的加密存储秘钥,B 用户通过自己的私钥解密获得明文存储密钥。

# 6、YOTTACHAIN 存储网络与交易市场

## 6.1.YOTTACHAIN 存储网络

### 6.1.1.概述

在传统存储提供商中心化的市场中，用户选择存储提供商并将数据付费存储，当用户需要获取数据时，通过存储提供商直接获取数据。存储提供商不需要实时向用户证明自己存储了数据（他只能且必须存储用户数据），而是当用户获取数据时，将存储的数据提供给用户。在这个市场中，中心化的存储提供商对于用户是公开的，用户可以自己选择相信存储提供商。

YottaChain 是去中心化的存储网络，在 YottaChain 的存储交易市场中，用户付费将数据存储在存储提供商的存储区域。用户和存储提供商之间是“匿名”的，这就需要存储提供商提供有效证明，供 YottaChain 网络进行验证，证明自己确实安全的存储了数据。

### 6.1.2.需求

存储证明（POS）必须能够防止作恶矿工产生的三种攻击：女巫攻击，外部攻击，世代攻击。作恶矿工可能通过这三种攻击来谎称自己提供了存储从而获得奖励。

女巫攻击：作恶矿工通过产生多个女巫身份来宣称自己存储了多个副本，实际只存储一个副本。

外部攻击：作恶矿工通过从其他存储提供商获取验证数据，来宣称自己存储了并没有存储的数据，从而获得比实际存储更高的奖励。

世代攻击：又称代攻击，指作恶矿工通过小程序来快速的生产处理请求，迎合验证，谎称自己存储了大量的数据来获得更多的奖励。

### 6.1.3.基于安全的复制证明和时空证明

这里主要参考了 FileCoin 的复制证明( PoRep )和时空证明( Post ),PoRep 改善了 PDP 和 PoR 方案，有效的防止了三种攻击。

1. SEAL 封箱操作

存储矿工存储数据时通过 SealτAES-256 方法存储数据并生成副本 ,目的是让存储矿工可以诚实的存储数据 D 独立的 N 份副本数据，并保证有足够的时间允许验证者 V 生成随机验证挑战 RC。

2. 复制证明

定义：复制证明（PoRep）允许存储提供商通过提供副本证明（π）来说服验证者，在验证者发出随机挑战时，提供证明，证明数据 D 相对于证明者的特定副本 R 已经存储在唯一的专用物理存储区了。该方案是一种交互式协议。

复制证明（PoRep）的三个构建阶段：

PoRep.setup() -->副本 R，副本 Hash 树根 Merkel root of R，封装证明πSEAL

PoRep.prove() -->存储证明πPOS

PoRep.Verify() -->bit b （存储有效性证明 b1（πPOS）^封装有效性证明 b2（πSEAL））

3. 具体构建实践

PoRep.setup()

inputs：

--prover key pair (pkP , skP )

--prover SEAL key (pkSEAL)

--data D

outputs:R，Merkel root of R，πSEAL

处理过程：

· 计算 hD=CHR(D)

· 封装计算生成副本 R=SEALτ（D，skP）

· 通过散列函数输出树根 rt=MerkelCRH(R)

· 设置参数 $\vec{x}$ =（pkP，hD，rt）

· 设置参数 $\vec{w}$ =（skP，D）

· 计算副本封装证明πSEAL=SCIP.Prove(pkSEAL, $\vec{x}$ , $\vec{w}$ )

· 输出 R，rt，πSEAL

PoRep.Prove()

inputs：

--prover Proof-Of-Storage key pkPOS

--replica R

--random challenge c

outputs: a proof πPOS

处理过程：

· 计算 Merkel 树根 rt=MerkelCRH(R)

· 计算从根 rt 到叶子 Rc 的路径 path

· 设置 $\vec{x}$ =（rt, c）

· 设置 $\vec{w}$ =(path, Rc)

· 计算存储证明πPOS=SCIP.Prove（pkPOS，$\vec{x}$ ，$\vec{w}$ ）

· 输出存储证明πPOS

PoRep.Verfy()

inputs：

--prover public key ，pkP

--verifier SEAL and POS keys vkSEAL ， vkPOS

--hash of data D, hD

--Merkel root of R， rt

--random challenge ，c

--tuple of proofs，（πSEAL，πPOS）

outputs: bit b =1 表示有效

处理过程

· 设置 $\vec{x}$ =（pkP，hD，rt）

· 计算 b1=SCIP.Verify(vkSEAL, $\vec{x}$ ,πSEAL）

· 设置 $\vec{w}$ =（rt，c）

· 计算 b2=SCIP.Verify(vkPOS, $\vec{w}$ ,πPOS)

· 计算 b1^b2

4. 时空证明

允许存储提供商能够提供证明在某一时间段(t)内，都有效的存储了数据。采用时空证明（PoSt）审核存储提供商提供的存储，没有指定的验证者，任何的网络成员（有权限的网络成员）都能够进行验证，该方案是非交互式的协议。

Post 的构建方案：

PoRep.setup()　-->副本 R，副本 R 的 Merkel 树根 R，封装证明πSEAL

PoRep.prove()　-->t 时间内生成顺序的存储证明πPOST

PoRep.Verify()　-->bit b （存储有效性证明 b1（πPOS）^封装有效性证明 b2（πSEAL））

在 Post 的构建方案中，setup()和 Verify()跟 PoRep 复制证明一样，在 Prove()中，证明人接受验证者的随机挑战生成复制证明，并将复制证明作为输入迭代 t 次后输出顺序的存储证明πPOST

## 6.2.YOTTACHAIN 交易市场

YottaChain 上的各种资源都会在 YottaChain 交易市场上公开交易，交易市场会根据提供服务的节点的资源数量、带宽、网络延迟、报价等因素撮合交易。

每种不同的资源都有自己的通证，用户需要用对应的通证来购买所需的资源。所有的通证都可以与 YTA 进行自由兑换，当用户想要使用 YottaChain 上的资源时，需要用 YTA 来兑换相应的通证，交易市场会提供各资源节点的报价，并用集中竞价的方式自动撮合交易。

# 7、YOTTACHAIN 示范应用

## 7.1.YOTTACHAIN 内容共享应用

YottaChain 将构建一个内容交换应用，内容提供者将文件内容存储到 YottaChain 存储网络里，并将文件进行加密，内容访问者可以请求内容提供者进行授权，内容提供者授权后，内容访问者可以查看或播放相关内容。

内容提供者通过 TruPrivacy 技术将文件内容进行加密，并将文件保存到存储网络。内容提供者保存文件到存储网络时，将通过存储市场付费保存，为了鼓励更多的内容提供者上传内容，YottaChain 将拿出一定量的 YTA 对内容提供者进行激励，使内容提供者可以免费上传文件内容到存储网络。

内容访问者通过内容应用检索内容，并通过 YottaChain 主干网络发起访问请求交易，交易根据内容提供者设定的价格自动完成，并通过智能合约自动为内容访问者生成访问授权。内容访问者根据授权可以下载文件内容并使用。

TruPrivacy 将通过加密后去重机制解决相同文件内容被重复上传占用存储网络空间的问题。

内容上传者将会在内容共享应用里自动登记身份，当出现版权纠纷时可以根据登记信息进行确权。

通过 TruPrivacy 加密后的文件，如果出现违反法律法规的情况，将可以根据监管部门的要求对文件进行屏蔽访问。

## 7.2.YOTTACHAIN 云盘应用

### 7.2.1.云盘应用用户需求

1.数据文件共享的办公意义

中小企业和创业型公司有着很强的文件共享、数据备份等方面的办公需要，但由于自身规模的限制，这些企业难以像大企业一样，采购专业的存储和备份设施，更不可能安排专人进行日常的配置维护。

云盘应用基于以上挑战，为企业提供一个经济有效、且易于管理的解决方案，通过提供数据存储备份、安全分发、快速分享等重要功能，实现安全可靠、管理简便的企业数字资产管理平台，提高企业数字资产管理水平和利用效率，满足企业各部门无边界协同办公和信息共享与资源管理的需求，实现了办公的云化，帮助用户提高工作效率，降低运营成本。

企业云盘应用是针对企业用户而设计的产品，旨在满足企业协同办公的需求。同时也提供了个人使用的空间，保证了文件存储的私密性和文件安全性。多名员工可以创建一个协同办公文件夹，其中一名员工更改文档后，更新的文档会显示在协同的办公文件夹下。支持文档直接在线浏览，实现云办公。提高文件分发共享能力、增强协同性、提高办公效率。

2.无处不在的互联网环境趋势造就了虚拟移动存储

云存储服务正伴随着大数据和移动互联网的发展成为 IT 经济下一个新的增长点。根据 IDC 的统计，未来四年内，云服务的市场规模将从现在的 174 亿美元增长到 442 亿美元，其中，云存储服务的市场比例将从目前的 9%增长到 14%，也就是说云存储服务的市场规模将接近 62 亿美元。

许多互联网企业顺应市场发展，都推出了免费的大容量个人云盘业务，并获得了广泛的应用；对于企业级用户，也希望通过云存储服务来保护自己现有的数字资产、方便信息沟通，降低数据管理和维护成本，适应业务的快速发展。

目前，企业中重要的数据往往分散在员工的各种终端设备中，由于缺乏系统的数据备份保护，当文件不可用时，无法进行恢复，造成数据不必要的流失，随着企业规模的壮大和分支机构的开设，企业内部有很强的数据交换和分发需求，目前普遍采用邮件或 QQ 等方式传送，缺少监管流程，很容易造成数据泄露或丢失；对于大文件传送，受限于现有网络带宽，传输效率低下，成功率低，阻碍了企业日常业务的进行。

云盘应用基于以上挑战，为企业提供一个经济有效、且易于管理的解决方案，通过提供数据存储备份、安全分发、快速分享等重要功能，实现安全可靠、管理简便的企业数字资产管理平台，降低企业 IT 采购成本，提高企业数字资产管理水平和利用效率。

### 7.2.2.云盘应用特点

1. 共享性

快速分发

企业云盘可实现快速分发，上级部门上传文件后，子部门及下属分公司可立即获得文件；

通过分享功能，可将团队材料快速分享给同事成员，大幅度提高分发和管理的效率和质量。

通过外链功能，提取文件外链后可迅速通过邮件、QQ、微信等将文件分发给客户；

快速汇总

通过分享机制可迅速完成小组、团队人员对资料的收集和汇总。一人建立文件夹并分享团队人员。团队人员上传资料后，组长可以立即收集所有人资料并进行归纳统计。

快速收集客户资料，通过建立共享文件夹，分发文件夹连接后，可快速收集客户资料，实现与客户资料的手机更方便沟通，不丢失客户。

高效共享

企业云盘支持多种形式的共享方式。

组织内分享，云盘用户可快速将个人材料分享给伙伴、同事，方便沟通和协作。同时支持更灵活的小组机制。

外链分享，可通过外链方式将客户拉入交流团队，实现互动和共享。及时了解客户和合作伙伴的动向。

2. 易用性

轻松携带

通过设立自动同步，可随时将文件保存在云端。手机、电脑、pad 等多终端随时查看阅读。出差，回家，拜访客户，度假旅游再也不用携带 U 盘、硬盘、光盘、笔记本等。拜访客户用手机轻松搞定。旅途中也可以随时查看公司各种文件。

多终端

支持手机端查看材料，进行分享设定以及分发。使用手机随时随地调取网盘中文件，生成访问外链，无需任何上传时间，提高工作效率。

所有产品资料存储在网盘中，不用再打印、携带纸质材料，销售人员只需携带平板电脑或者使用手机，会议上投影、展示、分享给与会人员一键轻松搞定。

多接口

开放的 OpenAPI 可以兼容国际行业标准 Amazon S3 接口，面向开发者提供存储的 API，方便开发者在自己的应用中调用。

开放全面的 API 接口，满足企业所有集成需求。

多场景

在施工场地、项目现场使用手机实地拍照，直接上传网盘。施工现场使用平板电脑直接查看云盘中的施工图纸。

使用云盘 iOS、Android、PC 客户端，在旅途中也能及时处理工作任务。客户端支持多种格式图片、音视频的在线播放，及工作文档的在线预览。

3. 安全性

可控性

云盘应用在文件的使用、传播、存储等多方面采用了最高级的安全技术保障了公司的文件的可靠性。

加密安全

云盘应用旨在保护企业数字资产的完整可控，不会因为人员变动、电脑更换或丢失、重装系统、硬盘故障、病毒木马入侵等因素而丢失或泄露。通过对系统管理员权限的制约，让系统管理员也不能看到任何文件，无法偷窥企业秘密。

传输安全

云盘采用客户端加密，客户端解密，在云端不存储任何密钥和明文。全程传输采用密文传输，任何环节不泄露公司机密。

操作可控

云盘应用设置了角色权限管理系统，除保留角色外支持自定义角色，可以为不同人员分配不同的角色权限，从文件操作上保证文件的安全性。

设立了多重文件删除机制，即使员工恶意删除，文件也能安全找回。对员工离职设立了多种处理方案，保证文件的可传递性。

范围可控

云盘应用支持文件的分享，分享者可以自由选择可加入的人员与部门，以此控制访问人员的范围。

共享文件支持部门继承共享，同时，可控制子部门是否可以继承上级部门对文件夹的访问权限，从而进一步增强范围可控性。

梳理存储

建立共享视角、分享视角、外链视角等多种视角对文件进行组织。保证用户可以从不同角度梳理文件。

上级部门可以根据需要设立文件夹，并授予不同子部门相应权限。子部门递交相应材料后，父部门处自动按照子部门分类完成汇总，形成清晰目录结构。方便查阅。

历史版本功能，保证同一文件按照更新顺序进行记录，同时保证了所有文件都存在，可任意回退找到自己想要的版本。

### 7.2.3.云盘应用系统结构

YottaChain 云盘应用 DAPP 以 YottaChain 账号管理系统、YottaChain 存储网络为基础，支持以群组为单位开通云盘应用，相当于一个企业、部门或者组织开通一个共享空间的云盘应用。

云盘 DAPP 的开通需要三个步骤：1、设置云盘应用的群组，只有群组的管理者才可以开通云盘应用；2、使用 YottaChain 计算网络为云盘 DAPP 提供计算资源，云盘 DAPP 会通过 YottaChain 交易市场完成计算资源的购买；3、使用 YottaChain 存储网络为云盘 DAPP 提供存储服务，云盘 DAPP 会通过交易市场完成存储资源的购买。

# 8、YOTTACHAIN 治理结构

YottaChain 项目提出了一个去中心化的治理结构，解决了区块链的治理结构问题。

## 8.1 法律渊源

首先，我们参照法律学的研究成果，定义 YottaChain 的法律渊源。在人类社会中，集权体制（例如中国的封建皇朝）的法律渊源可以追溯到君主的个人意志，民主体制则是全民投票为大。在 YottaChain 中，作为去中心化的治理结构，法律渊源都可以追溯到全体持币者的投票。

全体持币者的投票是一币一票，而不是一账号一票。

## 8.2 社区治理委员会

YottaChain 实行代议制，由全体持币者投票选举的社区治理委员会来制定 YottaChain 的规则。这是因为：

出于效率考虑。若凡事都由全体持币者表决，将会缺乏可操作性。

出于专业性考虑。规则需要由具备专业能力的人来制定，且别说法律条款如何定义能达目的这种实体性问题，就算是在法律之间、各条款之间保持一致性就不是易事，堵住立法漏洞、实现罪罚相符等都很具有专业性。由具备专业能力的人来立法，这也是一个必然的选择。YottaChain 除宪法外，所有的规则都是由社区治理委员会来制定的，包括但不限于挖矿的算法、是否回滚、各委员会成员的任命、数字货币总量的增长规模等。

如果社区治理委员会制定的规则违背了大多数持币者的利益，那持币者可以罢免其权利。只要有一定数量的持币者提议，就可以自动举行全民投票。区块链保证了这种机制可以高效低成本公正地进行，从投票的发起、举行和结果履行都是程序自动执行的，和 EOS 选举超级节点相似，任何人都难以阻扰，也难以作弊。这种机制就可以保证社区治理委员会“大体上”是符合全体持币者的利益的。“大体上”是公平和效率之间妥协的结果，不能因为偶尔的非关键性的差错而改选社区治理委员会。

## 8.3 从规则到代码

在 YottaChain 看来，“code is law”的真正含义是“law is implemented by code”。我们将从规则到代码的转换过程分为三大环节，并用完善的制度保证每个环节都能正常履行职责。这三个环节分别是：从规则到产品需求规格的转换，从需求规格到编码的转换，以及编码的发行生效。

### 8.3.1 代码规格委员会

YottaChain 设立代码规格委员会，其职责是将社区治理委员会制定的规则转换为产品需求规格。代码规格委员会的成员由社区治理委员会任命，受社区治理委员会管制。代码规格委员会制定的需求规格必须严格遵守社区治理委员会制定的规则，除了改 bug 和性能提升等不破坏规则的改进外，代码规格委员会不能擅自提出新需求。

之所以单独设立代码规格委员会，是因为产品需求规格的撰写也是具有专业性的，其专业性要求与社区治理委员会的专业要求是不一样的。但如果代码规格委员会撰写的需求规格与规则不一致，社区治理委员会有权利撤换其成员。

鉴于整个过程是开放透明的，一旦发生代码规格委员会撰写的需求规格与社区治理委员会投票产生的规则不一致的情形，YottaChain 社区会向社区治理委员会举报。这种需求规格和规则的背离如果是有意为之，不管是否造成不良后果都是非常严重的违法行为，违背了其岗位职责、辜负了社区治理委员会的信任，社区治理委员会自然会严肃处理。万一社区治理委员会对此置之不理，则社区治理委员会就同样构成了非常严重的违法行为，违背了其岗位职责、辜负了持币者的信任，全体持币者自然会严肃处理。在这种一环扣一环的制度安排下，就可以保证每个人都不得不认真履行职责。

### 8.3.2 编码委员会

编码细分起来还可以分为架构设计、根据架构编码、测试、合并代码等环节。不过这些都遵循开源社区的规范来操作即可，全世界的程序员（甚至无需是 YottaChain 持币者）都可以贡献代码，实现去中心化的编码。有必要的话，也

可以组建编码委员会，牵头组织编码过程。与代码规格委员会相似，编码委员会也是由社区治理委员会任命，向社区治理委员会负责。

### 8.3.3 代码颁布委员会

最后一个环节是代码的发行生效。这是一个最后把关的环节，必须确认研发出来的代码符合需求规格而且没有严重的 bug 才能签字发行。YottaChain 的机制是所有的节点的最底层是 YottaChain 宪法的客户端程序，该程序的最主要的作用就是对特定签名的程序自动下载安装。为此，YottaChain 设立代码颁布委员会，当该委员会的成员投票表决同意颁布某一代码时，该代码就自动被赋予前述特定签名，则所有节点都将被自动更新，从表决通过到全部节点更新的过程都是全自动化执行的。同样，代码颁布委员会也是社区治理委员会任命，向社区治理委员会负责的。

### 8.3.4 小结

代码规格委员会、编码委员会、代码颁布委员会所要求的专业能力有相似之处，之所以分成不同的组织机构，主要是因为兹事体大，需要互相监督、制约均衡。代码规格委员会提交的需求规格如果与社区治理委员会制定的规则不符，不用等社区治理委员会采取行动，编码委员会就可以发现问题并提出异议。而如果编码委员会提交的代码不符合需求规格，则代码颁布委员会就不会批准，也就不会执行生效。反之，如果代码颁布委员会想做点手脚，但因为代码不是其编写的，无法批准发行带有其不轨意图的代码。

## 8.4 去创始人化

“去中心化”意味着“去创始人化”，但对于创始人来说，很难让其离开对一个倾注无数心血的从无到有的项目。这个问题就类似于集权制到民主制的转换路径。在项目启动时，项目由创始人来控制，等项目发展成熟却要“去创始人化”，这种转换不容易实现。从另一个角度来考虑，创始人彻底离场不一定是一件有益的事情。

通常说来，不存在任何一个比创始人对项目倾注感情更多的人，也不存在任何一个比创始人更了解该项目的人。没有任何人比创始人更想运营好该项目。创始人的离场，显然不是一个合理的方案。

一种比较合理的方案就是参照人类社会的君主立宪制，给创始人礼仪性的待遇和紧急情况下的有限权利，但正常情况下创始人拥有的权利和普通持币者是相同的。

在这种方案中，创始人将项目上线、选举出第一届社区治理委员会之后，就由社区治理委员会接管社区，原基金会解散，创始人只担任礼仪性的社区领袖，相当于君主立宪的王室。这种礼仪性的待遇包括在宪法中给一个名分，可以代表社区公开发言做报告（但不能替社区决定任何事或承诺任何事）等等。唯一的特殊权利就是在紧急情况下创始人有权发起全民投票（而其他人发起全民投票是需要达到特定持币量的账户联署的）。总之，创始人可以利用其影响力去号召大家，但无权替社区做任何决定。

## 8.5 治理结构总结

通过以上的制度化设计，可以保证 YottaChain 是不断发展完善的项目，所有的权力都归于全体持币者，可以高效专业地制定规则，所有的规则都能确保变

成代码执行，任何人犯的错误都有对应的救济纠正措施。这样可以构建一个彻底去中心化的区块链项目，同时其运营不失专业性和效率。

# 9、应用场景

## 9.1 兼容 IPFS 所有应用场景

YottaChain 本身就采用 IPFS 作为去中心化静态持久化存储模块的一部分，可以兼容 IPFS 的所有应用场景，包括静态网页、CDN 等：

- 挂载全球文件系统，实现去中心化的持久化存储
- 文件版本管理
- 可用于所有软件的带版本的包管理器（已经实现了：https://github.com/whyrusleeping/gx）
- 可以作为虚机的根文件系统
- 可以作为数据库：应用可以直接操作 Merkle DAG，拥有版本化、缓存以及分布式特性
- 可以做通讯平台
- 各种类型的 CDN
- 永久的静态网页访问，不存在不能访问的链接

## 9.2 为个人和企业数据提供安全、低成本存储

YottaChaintigg 提供完善的数据安全机制，保证不管数据保存在多么不可信任的节点上，都不用担心数据被泄露，即使由全球最厉害的黑客亲自出马也难以

攻破，在任何情况下都只有数据的拥有者或其授权者能看到数据，对任何其他人（包括 YottaChain 的设计者、实施者）来说都只是乱码，而且不存在被攻破的风险，从实践意义上来说可以视为绝对安全的。

所以，个人和企业的数据不管多么隐私保密，都可以放心保存到 YottaChain 上，绝对不用担心安全保障问题，相比于将数据保存到 AWS、Google、阿里云、百度云等，更加安全可靠，甚至比保存在个人电脑上更加安全可靠。

同时，由于 YottaChain 在做好加密安全保障的同时没有牺牲任何存储效率，尤其是同时还具备数据去重的能力，可以将存储的成本降低 5-10 倍，比在售的任何厂商的存储设备更加便宜，不管是云存储、企业级存储还是分布式存储，甚至包括桌面存储在内，不管是一线厂商还是低廉的劣质产品，都比 YottaChain 的存储成本更加高昂。

## 9.3 充分利用闲置资源，打造真正共享经济

共享经济是指以获得一定报酬为主要目的，基于陌生人且存在物品使用权转移的一种新的经济模式，其本质是整合闲置资源，是一种人们公平享有社会资源、各自以不同的方式付出和受益、共同获得经济红利的模式。此前，共享经济的模式通常是通过互联网平台来实现的。

共享经济牵扯到三大主体，即商品或服务的需求方、供给方和共享经济平台。共享经济平台作为连接供需双方的纽带，使得供给与需求方通过共享经济平台进行交易。

聚合全球闲置住宿资源的 Airbnb 是共享经济的代表性企业，试图聚合闲置交通工具的滴滴、摩拜也曾经被视为共享经济的代表，但在实践中，滴滴、摩拜等企业并非激活闲置的交通工具（除顺风车外），而是产生了专门为滴滴、摩拜

提供服务的交通工具，所以并不是真正的共享经济，没有起到激活闲置资源的作用。

YottaChain 利用区块链技术打造共享经济平台，聚合全球闲置的存储资源和计算资源，供有需求的用户使用，是真正的共享经济。

据 Gartner 的数据，全球现在总共有 300 多万个企业级数据中心，每个企业数据中心都有大量的存储资源和计算资源，如果加上个人家庭的资源（路由器、电视机等），更是不计其数了。这些存储和计算资源基本上都存在闲置的部分（将硬盘全部存满一点都不剩的情况是几乎没有的），如何合理地利用海量的闲置资源将是一个意义重大的课题。

YottaChain 利用独有的区块链激励模型能调动存储空间和计算能力的所有者将暂时闲置的资源贡献出来挖矿，为他人所用，充分共享社会资源，从而落地实现一个规模庞大的共享经济系统。

## 9.4 将自用存储空间用来挖矿

对 YottaChain 来说，除了将闲置资源用来挖矿外，正在使用和即将使用的存储资源也是可以用来挖矿换取奖励的，并且可以做到数据存储和挖矿换取奖励的两不误。

举例说明：一个用户如果有 1TB 的存储空间，本来是用来存储 1TB 的数据的，现在用这 1TB 的存储空间加入到 YottaChain 来挖矿，挖到的 YTA 反手再购买存储空间，可以存 2TB 数据，还能剩余一些 YTA。

这个奇迹般的魔法效果就是因为 YottaChain 的数据去重技术，使得 1TB 的物理空间可以存储至少 5TB 的数据，所以用 1TB 的存储空间挖矿得到的 YTA 币，要远远超过购买 2TB 数据存储空间所需要的 YTA 币。

所以对 YottaChain 来说，即使没有闲置存储空间也可以用存量资源来挖矿，以此来获得额外的数字货币作为奖励。

## 9.5 作其它区块链项目的基础架构

作为一条基础公链，YottaChain 将为其它区块链项目提供坚实、安全可靠、低成本的基础架构支撑，包括但不限于：

- 为其它区块链项目快速提供大量节点：每个新的链上线的时候，往往都需要大量的节点，节点越多区块链的分布式账本就越可靠。如果靠早期用户来建立区块链节点，需要很长时间才能达到比较多的节点数量；如果在 AWS 等公有云开虚拟机的方式快速建立区块链节点，则由于故障域不隔离，丧失了区块链的去中心化的一些重要价值。而如果在 YottaChain 上建立节点，则既能快速、低成本建立节点，而且这些节点本身就是建立在区块链节点之上的，天然符合区块链对去中心化、故障域分离等方面的所有要求。

- 区块链本身存储的分布式账本是所有全账本节点都要完整保存的，每个节点都存所有区块的信息，相当于是一种多副本冗余的存储模式。例如比特币有 1 万多个全账本节点，那相同的数据就存了 1 万多份。这种模式虽然可靠性非常好，但冗余度也非常高，对于存储空间非常小但重要性非常高的数字货币交易记录来说还可以接受，但如果要存储其它类型的的数据的话则成本将十分高昂。YottaChain 为其它区块链项目提供了

一种更有效率的存储模式，用冗余编码的方式存储数据，将数据存储的冗余度降低到经济合理的水平，而且不受区块容量的限制，可以存储近乎无限的数据。

- 区块链需要在区块中打包的交易记录，每个区块可以打包的交易记录数量是有限的，这就导致在交易高峰的时候发生拥堵阻塞，有时甚至要超过一天时间才能将所有待确认的交易打包到区块链中从而确认交易。而借助 YottaChain，可以将交易记录存储到 YottaChain 中，每个区块只需记录几十字节 hash 值即可，这样一个非常小的区块都可以存储无数笔交易，而且同样具备防作弊、防节点故障等特点，从此无需区块扩容。

## 9.6 作为低成本对象存储

对象存储是云存储服务商提供的一种基于 API 调用的存储模式，处理和解决了曾经被认为是棘手的存储问题：不间断可扩展性、弹性下降、限制数据持久性、无限技术更新和成本失控的。AWS 的 S3 对象存储服务的 API 协议是对象存储的事实标准。

YottaChain 将提供 S3 兼容的对象存储服务，而且存储成本更低，这样 AWS/S3 或其它云平台对象存储的用户无需修改代码即可马上降低每月的费用。

## 9.7 作为具备容灾能力的持久化存储

YottaChain 的去中心化存储是天然具备异地容灾能力的，YottaChai 将提供标准块存储接口和 NAS 存储接口，可以作为常规企业级存储（每年大约 600 亿美元市场）的低成本方案，而且自动具备容灾能力。

将来，所有的中心化存储（包括 AWS、阿里云、EMC、华为）都不会被视为持久化存储，而只能被被作为本地缓存使用。只有去中心化存储才能作为持久化存储使用。一个明显的例证是2018年8月腾讯云曾彻底丢失了用户数据，AWS和阿里云也曾经多次出现过全球性故障。

# 10、团队和顾问

YottaChain 是由新加坡 YottaChain 基金会负责运营的区块链项目，创始团队和顾问团都非常专业、资深。

## 10.1 核心团队

YottaChain 已经有数十人

1. 王东临 创始人

顶级 IT 科学家，知名企业家，同时有丰富的社会治理经验。

王东临具有 20 年+的密码学应用经验和将近 10 年的分布式存储经验，均达到世界顶级水平，先后发明了十多项国际领先的技术，被评为中国十大青年科学家（中国科协每年从全国各行各业中总共评十个，政治局委员颁奖的国家最高荣誉之一）、首届中国杰出工程师（中国科技部评选，软件互联网行业唯一入选）、中国软件十大杰出青年（中国工信部和团中央联合评选，唯一全票当选），先后发明十多项国际领先技术，创造多个中国 IT 业的里程碑，拥有 100 多项美国、欧洲、日本、中国专利。

作为连续创业企业家，王东临创办了国际知名的中国老牌 IT 领军企业书生集团，并曾经数亿元价格出售了旗下一家公司。

书生集团作为一家有技术基因的公司，历经 IT 业的多个时代始终屹立在技术创新的前沿，其基于密码学构建的数据安全产品（安全文档、安全存储、安全云盘、安全通讯）得到广泛应用，客户包括中国 100%的中央部委、100%的省级政府、100%的央企、100%的银行和若干顶级涉密机构，涉及数十亿份涉密文件从来没出现过安全责任事故。

王东临发明的 TruPrivacy 是世界上唯一能实现加密后数据去重的技术，在全球范围都有专利保护。TruPrivacy 技术依靠完善的密码体系，即使网络被攻破、服务器被控制、关键人员被收买，也能保证用户数据安全、黑客无法盗取。在 2015 年全球最大黑客大会 DefCon 上，书生云敞开服务器将其控制权交给黑客，高额现金悬赏，但无人能偷走用户数据，经受了最苛刻的公开验证。

王东临发明的 SurFS 分布式共享存储系统是分布式存储技术的重大创新，利用独特的技术大幅度缩短了数据路径、大幅度提升了节点之间的数据传输性能，并同时大幅度降低了系统成本，获美国《云计算》杂志“云存储卓越奖”。

王东临还担任 OASIS 国际工业标准组织 UOML-X 技术委员会主席，有丰富的按规则管理跨国组织的经验；多年深度参与立法的经验，对法律体系有深刻的认知。

2. 侯月文 联合创始人/CEO

侯月文是技术出身的企业家，曾任书生电子公司研发总监，精通密码学相关技术，多次承担国家顶级涉密机构的数据安全项目。作为连续创业者创办摩宝时代等多家创业公司，在互联网产品研发、社群运营等方面均有丰富的实战经验。

侯月文负责的云盘业务上线 2 年时间就在全球拥有 1000 多万用户，其中企业云盘产品是中国一线主流厂商之一。

3. Peter Junge 欧洲团队负责人

德国汉堡大学硕士，先后担任 Sun Microsystems 公司资深工程师和知名开源社区 OpenOffice.org 项目经理，在 IT 技术和开源社区管理都具有丰富的一线经验。

Peter 具有德国人特有的严谨认真，在合规性方面有突出特长。

4. Yvonne Li 美国团队负责人

美国休斯敦大学毕业，先后在洛克希德和 NASA 担任工程师，半导体巨头 KLA-Tencor 担任国际业务总监，后在硅谷做过移动互联网创业。

Yvonne 具有良好的技术背景，在硅谷具有广泛的人脉关系、

## 10.2 顾问团

- Laurent Liscia 国际开放标准组织 OASIS 主席

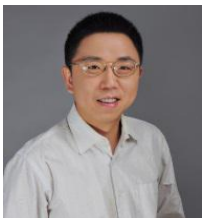
OASIS 是权威的国际工业标准组织，由全球 100 多个国家的主要 IT 厂商、用户、学术研究机构组成。

Laurent 曾担任法国外交部官员

- Louis Suárez-Potts OpenOffice 开源社区领袖

Louis 长期负责管理 OpenOffice 开源社区，曾任职 Oracle 社区运营总监。

- 蒋涛 CSDN 创始人



全球最大的开发者社区 CSDN 创始人，极客帮创投创始合伙人

- 王峰　火星财经/共识实验室创始人

区块链一线主流媒体火星财经创始人，上市公司蓝港互动创始人，极客帮创投合伙人。

- 赵晓 著名经济学家

曾担任国资委研究中心宏观战略部部长、中国经济学奖专家委员，著名经济学团体“博士咖啡”核心成员。

## 11、风险与免责声明

本文件是 YottaChain 项目阐述的概念性文件【白皮书】，并非出售或者征集招标相关公司的股份、证券或其他受管制产品。

根据本文件不能作为招股说明书或其他任何形式的标准化合约文件，也并不是构成任何司法管辖区内的证券或其他任何受管制产品的劝告或征集的投资建议。

本文件不能成为任何销售、订阅或邀请其他人去购买和订阅任何证券，以及基于此基础上形式的联系、合约或承诺。

在本文件中所呈现的任何信息或者分析，都不构成任何参与代币投资决定的建议，并且不会做出任何具有倾向性的具体推荐。

YottaChain 基金会不承担任何参与本项目造成的直接或间接的资产损失。

这份文件可能随时会被修改或者置换，然而我们没有任何义务更新此版本白皮书，或者提供读者额外资讯的渠道。