YottaChain

去中心化的加密存储和算力共享系统

YottaChain 基金会 2018 年 8 月

V0.90

目　录

摘　要

YottaChain 是一个区块链基础公链，它用独有的技术手段、经济模型和治理结构将全球的计算资源和存储资源连接起来，构成一个规模浩瀚的星际计算机，使得普通人可以拥有今天的超级巨头才能享有的计算能力和存储能力，并可为其它区块链提供便捷、可靠、廉价的存储和计算服务。

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，而且这个模式无需任何补贴，是长期可无限持续的。

YottaChain 构建在 IPFS 的基础之上并克服了 IPFS 的一些局限性，相当于升级版的 IPFS。IPFS 在设计之初作为 HTTP 协议的取代者，因此更适合保存用于被公开访问的网页数据。YottaChain 除了兼容 IPFS，还实现了对文件的加密机制和授权机制，利用独有的“加密后去重”技术保证加密后不增加系统存储成本，并增加了算力共享机制，这样不仅可以用于保存个人或者企业非公开的文件，支持动态网页（从而真正取代 http），而且从 IPFS 的“存储+网络”升级成了“存储+加密+计算+网络”，从而具备了完整的基础架构能力，不管是应用场景还是商业模式上都具备了无限的发展空间。

1、背景

1.1.传统区块链存在的问题

区块链可以改变世界，但作为早期的开拓者，比特币、以太坊等传统区块链存在以下的不足：

1.1.1 传统区块链过度耗费资源挖矿的问题

比特币、以太坊等传统区块链都有一个特点，就是需要空耗算力来挖矿。这种工作量证明（POW）有一定的公平性，这是数字货币整个的商业逻辑能够自洽的重要因素，但却造成了社会资源的大量浪费。仅比特大陆销售挖矿芯片的收入就达到 100 多亿元，每年挖矿消耗的电力就超过了世界上 159 个国家的用电总量。这么多的矿机和电力没有对社会有任何的贡献和价值，全都被虚耗掉了。我们所宣扬的区块链新时代，不应该是这样一个空耗资源的时代，这样的时代也不能代表社会的进步。从社会进步文明发展的角度看，应该以“谁对社会贡献资源就给谁奖励”的正向机制，取代现行的过度耗费资源挖矿机制，从而构建真正能推动社会进步的区块链新时代。

1.1.2.传统区块链治理结构的问题

传统区块链缺乏治理结构，导致事实上被基金会或者少数人操纵。在 code is law 的区块链世界，决定 code 的核心开发人员能够决定所有的法则。虽然区块链项目是开源的，每个人都可以贡献代码，编码本身是去中心化的，编码和贡献代码是可以做成去中心化的。但谁的代码被采纳、合并进主干，这就是中心化的了。更为关键的是，产品的需求规格是谁定义，这就是核心问题了。

技术人员的编码都是按照产品需求规格的要求来做的。在传统的软件项目中，产品需求规格都是通过中心化的方式来决定，掌握这个权力的人就拥有决定一切的权力。产品需求规格是由谁定义的、按照什么规则定义的、如何保证编码严格符合产品需求规格、如何保证最终发行的版本是符合需求规格定义的代码等，这些问题是区块链项目治理结构的核心问题。如果没有治理结构，或者没有去中心化的治理结构方案，那一个区块链项目就不能称之为真正的“去中心化”。

EOS 在治理结构方面迈进了一大步，由 0 发展到了 0.1。创始人放弃对社区的主导管理权，有了 EOS 宪法，选出了 21 个超级节点。与缺乏治理结构的比特币和以太坊相比，EOS 毫无疑问是有治理结构的，只是这个治理结构还不够完善。

1.2 IPFS 没有解决的问题

星际文件系统 IPFS（InterPlanetary File System）是一个面向全球的、点对点的分布式版本文件系统，目标是为了补充（甚至是取代）目前统治互联网的超文本传输协议（HTTP），将所有具有相同文件系统的计算设备连接在一起。2015 年初，正式发布开源协议 IPFS，全世界有很多的区块链项目都是基于 IPFS 作为底层存储技术，采用 IPFS 作为存储机制已经成为越来越多区块链项目的首选。

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，甚至比你自己的存储还可靠得多，这就像比特币同样用不可靠的节点却构成了比银行更可靠的金融系统一样。

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，谁就得到最多的 FileCoin 奖励。这种新颖的共识机制奖励为社会做贡献者，谁的贡献大谁拿的奖励多，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 没有解决的问题

1.2.2.1 数据安全

IPFS 底层并没有实现数据的安全机制，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，而不适合用于存储个人数据和企业数据。因为个人数据和企业数据都是希望以更安全的方式存储，而非公之于众。

IPFS 建议在应用层通过文件加密实现部分的安全性问题，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，应用层很难做好，而且应用层做加密也无法解决文件去重问题，从而影响了整个系统的效率和成本。

1.2.2.2 计算能力

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 服务稳定性

IPFS/FileCoin 对所有节点不加区别的按照统一的激励算法进行激励，大量无法保证稳定服务的个人节点混杂其中，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，并且在节点离线时其他节点可以重建丢失数据，但是这也会势必影响 FileCoin 的商业级交付质量。

1.2.2.4 应对监管

IPFS 的推广预期会面临各国政策和法律的风险，而 IPFS 本身体系的设计并没有考虑应对此方面的风险。如果监管层通过屏蔽特定协议和网络端口，将会直接使整个 IPFS 网络瘫痪；而如果试图通过更改协议和端口的方式绕开监管，将会变成一场猫捉老鼠的游戏，并且会形成一个一个孤立的 IPFS 网络孤岛。

2、YOTTACHAIN 介绍

2.1. YOTTACHAIN 简介

YottaChain 相当于升级版 IPFS，在 IPFS 的基础上了进行了诸多的优化工作。

YottaChain 在 IPFS 基础上的优化之处包括：

2.1.1 提供了数据安全机制

YottaChain 得到 TruPrivacy 专利持有者（主要技术贡献者书生星际的母公司）的独家授权，在区块链项目中独家集成了 TruPrivacy 云数据安全技术，保证存储网络上的数据只有被授权者才能看到。

YottaChain 的 TruPrivacy 云数据安全技术是全球唯一能实现加密后去重的技术，并持有全球专利。YottaChain 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）TruPrivacy 将采用独特的数据加密技术，使加密后存入 YottaChain 后的文件可以去重；

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

3）TruPrivacy 在数据级实现文件的授权机制，使得文件仅能被授权人打开，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剩下一些数字货币。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 提供了“存储+计算+网络+加密”的完整能力

YottaChain 不仅对贡献存储空间有数字货币的奖励，只要贡献对生态系统有价值的资源就可以奖励数字货币，包括但不限于存储能力、计算能力等。

众所周知，IT 基础架构由存储、计算和网络三大件组成，其中所有的区块链项目都天然带有网络能力。IPFS 提供了存储能力，而 YottaChain 可以提供“存储+计算”能力，从而凑齐 IT 基础架构三大件，构成完整的 IT 赋能，使得 YottaChain 可适用场景和商业模式将大为扩张。YottaChain 不仅可以支持动态网页，为 DAPP 提供完整的 IT 基础架构能力，甚至还支持去中心化的 IaaS 公有云商业模式，这将改变 AWS/阿里云等传统云计算服务商的商业模式。

2.1.3 去中心化的治理结构

YottaChain 提出了完善的去中心化的治理结构。YottaChain 提出的 YottaChain 宪法将从根本上解决区块链去中心化治理的问题，汲取人类社会经济学和政治学的理论和实践成果，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执行制定出来的规则，是 YottaChain 治理结构的一个重点。

2.1.4 其它重要改进

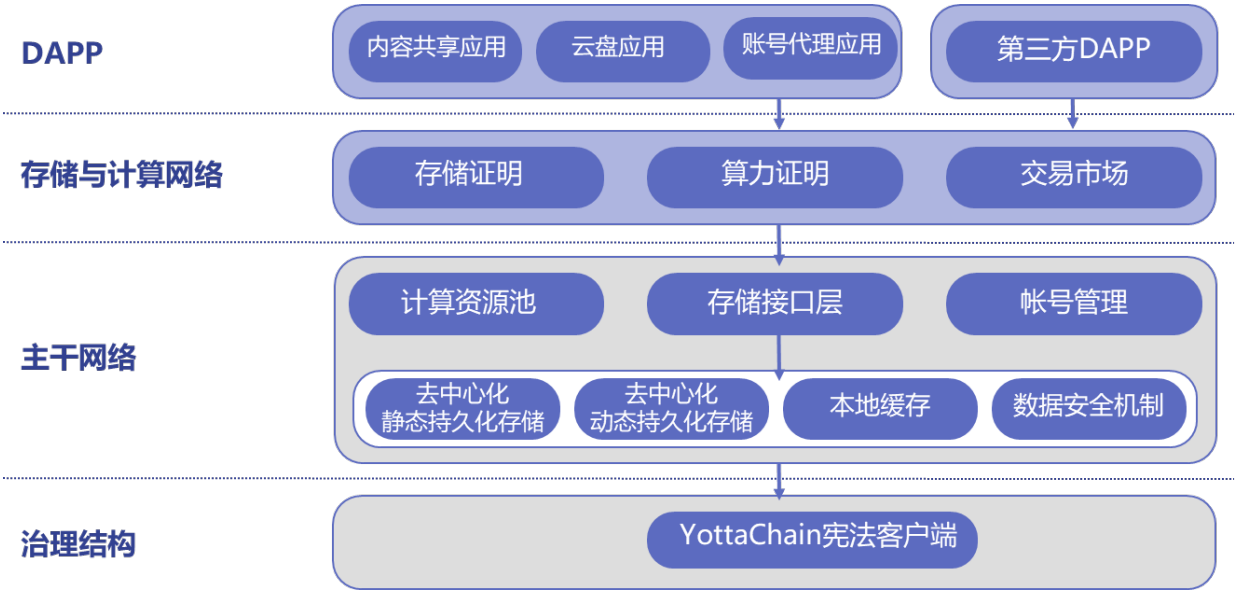
除了存储服务、计算服务外，YottaChain 还自带内容共享、云盘和账号代理 DAPP。

将节点分为商业节点和普通节点，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。

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

YottaChain 运行在一个能提供存储、计算的区块链之上，并提供内容共享应用和云盘应用， YTA 作为 YottaChain 的通证，奖励给对 YottaChain 生态有贡献的用户。



YottaChain 商业节点包含四层。

最底层为 YottaChain 宪法客户端，为整个系统提供基于代码实现的治理规则。

YottaChain 主干网络作为区块链的运行网络，包括去中心化静态持久存储、去中心化动态持久存储、本地缓存、数据安全机制、计算资源池、账号管理。

去中心化静态持久存储基于 IPFS，适宜存储静态数据，存储的数据是全球分布、非常可靠的，是通过数据的 hash 值来访问的；

去中心化动态存储与 IPFS 相似，但是通过 ID 来访问，一个文件不管内容发生多少变化，其 ID 都不变。同一 ID 的文件，新内容覆盖旧内容。动态存储数据先保存在本地缓存，每过一个间隔时间（例如 1s）自动更新到去中心化存储中，也可以用 flush 命令强制更新；

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；

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；

存储接口层，基于静态持久化存储、动态持久化存储、本地缓存和数据安全机制，提供块存储、NAS 存储、对象存储等常用存储接口，便于直接移植现有应用；

计算资源池包括 CPU 资源和内存资源（将来扩展到 GPU、FPGA 等），为本地计算以虚拟机或容器的形式提供计算能力 ；

账号管理为使用 YottaChain 系统的用户提供账号创建、登录、私钥托管、分组、消息管理等机制。

YottaChain 存储和计算网络为用户提供存储和计算能力，YottaChain 将综合 Filecoin 等项目的优点，通过存储证明和算力证明让提供方证明提供了服务且不能伪造，并提供了交易市场进行存储和计算服务的交易和任务调度管理。

YottaChain 提供内容共享和云盘两个示范 DAPP 应用，同时这两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。

3、通证和代币设计

3.1 概述

YottaChain 的原生本位加密数字货币（Cryptocurrency）为 YottaCoin，代号是 YTA。YTA 是系统权益币，主要用于为系统内其它通证提供价值抵押。YTA 也是流通币，是上各大交易所交易的数字货币。

YottaChain 内部的每一种资源（例如去中心化的静态持久化存储、去中心化的动态持久化存储、x86CPU 虚拟机资源、Java 虚拟机资源）都使用自己的通证(Token)进行挖矿和交易，YottaChain 提供系统内各类通证与 YTA 的交易服务。

YottaChain 的矿工提供资源获得相应的通证，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，针对静态持久化硬盘存储，静态持久化闪存存储，动态持久化硬盘存储，动态持久化闪存存储，本地硬盘缓存，本地闪存缓存，本地内存缓存，x86 虚拟机，ARM 虚拟机，Java 虚拟机等十种类型的资源，分别发行十种类型的资源通证。具体发行多少种资源通证，由立法会决定。

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相对固定，但每年有 50%的涨幅，这样越早期的矿工用同样数量的资源可以挖到的资源通证就越多。

对资源使用者来说，可以用资源通证来购买相应的资源。由于数据去重和 CPU 虚拟化的因素，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。

以下以静态持久化硬盘存储为例说明 YottaChain 的资源通证机制。假设静态持久化硬盘存储的通证是 SPH（具体名称在该通证创建时最终确定），在主链上线的第一年，一个矿工存储 1GB 数据一年可以获得 1SPH，第二年则需要存储 1.5GB 数据一年才能获得 1SPH，第三年是存储 2.25GB 数据一年获得 1SPH，以此类推，每年恒定增值 50%。

如果不考虑数据去重因素的话，那 1GB 数据存 10 年大约需要 2.95（即 1+1/1.5+1/1.52...+1/1.59）SPH。但加上数据去重因素后，就产生了很有意思的模型。

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 5-10 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 0.6SPH 就可以买到 1GB 数据 10 年的存储空间。这就创造了一个魔幻般的激励效果：一个用户拿 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帮别人存数据一年时间可以换来 1SPH 的资源通证，然后用其中的 0.6SPH 可以买到 1GB 数据的 10 年存储服务，手里还剩 0.4SPH。这就是人人做雷锋，人人帮助别人，自己还受益，而且还长期可持续，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 SPH 之后，他可以用如下方式处理这些 SPH：

1. 用这些 SPH 购买更多的存储空间，存他自己的数据

2. 将 SPH 兑换成其他资源通证（以 YTA 作为中介），购买其它类型的资源（例如虚拟机资源）

3. 持有 SPH，坐等每年 50%的增值

4. 兑换成 YTA，再兑换成法币落袋为安

5. 兑换成 YTA，持有 YTA，从而可以从零次分配中获得更多的 YTA

以上是以 SPH 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，永远不用担心价格跌到零，但也不会短期内十倍百倍增值，属于长期稳健增值，随着时间的推移其购买力越来越强大。

3.3 流通币

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，汇率则是浮动的，完全市场化的。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。总体说来，YTA 的价格与 YTA 兑换资源通证的汇率是正相关的。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。

YTA 总共发行 100 亿，其中 40 亿个用做私募、团队激励、社区建设、市场营销等，60 亿个通过挖矿产生。

YTA 采用 POS(Proof of Stake)共识机制，依据持币量来挖矿。上线后 100 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 1-4 年	4 亿/年	16 亿	56 亿
第 5-8 年	3 亿/年	12 亿	68 亿
第 9-12 年	2 亿/年	8 亿	76 亿
第 13-16 年	1.5 亿/年	6 亿	82 亿
第 17-20 年	1 亿/年	4 亿	86 亿
第 21-25 年	0.75 亿	3.75 亿	89.75 亿
第 26-30 年	0.5 亿	2.5 亿	92.25 亿
第 31-35 年	0.4 亿	2 亿	94.25 亿
第 36-40 年	0.3 亿	1.5 亿	95.75 亿
第 41-45 年	0.2 亿	1 亿	96.75 亿

第 46-50 年	0.15 亿	0.75 亿	97.5 亿
第 51-55 年	0.1 亿	0.5 亿	98 亿
第 56-61 年	750 万	4500 万	98.45 亿
第 62-67 年	500 万	3000 万	98.75 亿
第 68-73 年	400 万	2400 万	98.99 亿
第 74-79 年	300 万	1800 万	99.17 亿
第 80-85 年	200 万	1200 万	99.29 亿
第 86-91 年	150 万	900 万	99.38 亿
第 92-97 年	100 万	600 万	99.44 亿
第 98-104 年	75 万	525 万	99.4925 亿

在主链上线运营的早期，YTA 的挖矿速率相对较高，而这时系统的资源总量较小，导致发行的资源通证数量较少（资源通证的发行数量是与系统的资源总量直接相关的），YTA 能兑换到的资源通证也相对较少。随着时间的推移，YTA 的挖矿速率越来越慢，而系统中的资源越来越多，YTA 兑换的资源通证就越来越多，价值就会相应增长。以存储资源举例，在系统上线第一年可能新增了 100PB 的存储空间，但在第 5 年可能新增了 100EB 的存储空间，增长了 1000 倍，但新增的 YTA 数量反而还更少了，这就必然会导致每个 YTA 能换到的资源通证数量相应增加，YTA 自身的价值相应增长。因此，从 YTA 的价格短期看受供求关系、市场操纵炒作等因素影响较大，但长期看一定是大幅度增值的。

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，但其可读性并不好。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。

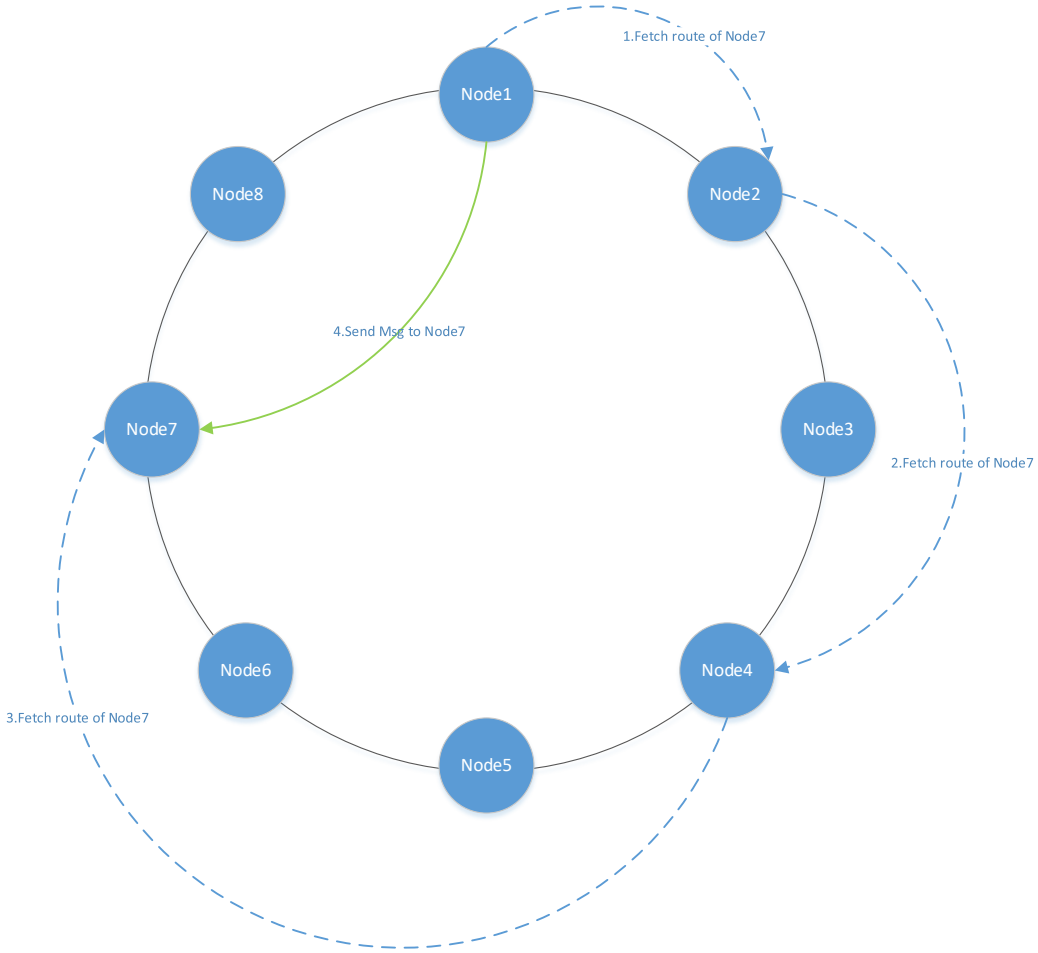
4.2.账户创建

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。

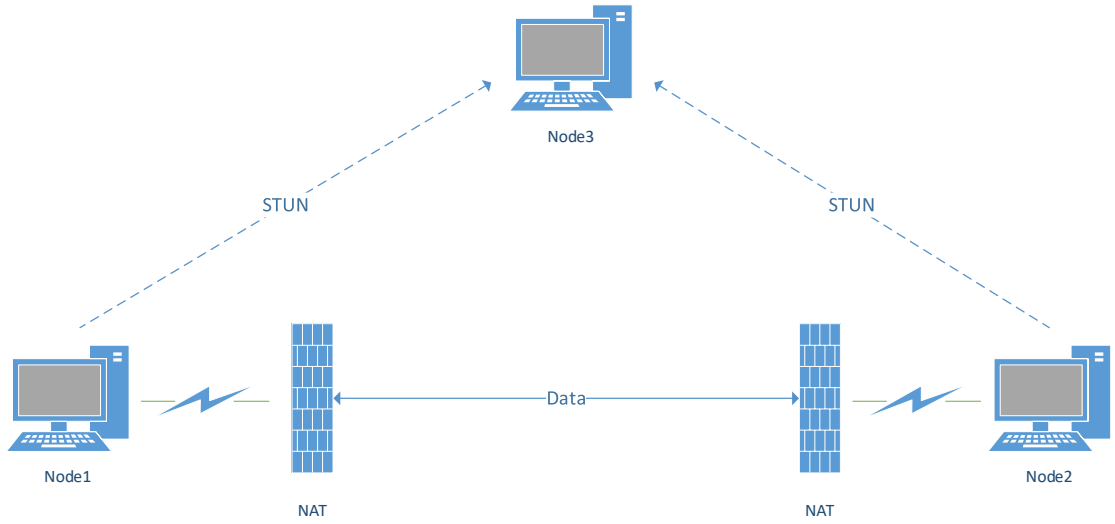
4.3.消息机制

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便改别人的文件，就乱套了。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（Stk），通过存储密钥 Stk 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。

Hash(Data) →Hdata　计算明文 Hash

RandomSym()→Stk 随机生成对称密钥作为文件的存储密钥

Enc(Stk, Data)→EncData 用存储密钥加密文件

Hash(EncData) →Henc　计算密文 Hash

Enc(Spub, Stk) →EncStk 以用户的加密公钥加密存储密钥

PutIPFS(EncData) 将加密数据存入到 IPFS

PutPri(Hdata，Henc, EncStk) 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 IPFS 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(Hdata)→Henc,EncStk 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. GetIPFS(Henc)→EncData 用密文 Hash 从 IPFS 中取出密文

3.Dec（Sprv, EncStk）→Stk 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(Stk, 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 ZeroKnowledge 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 ZeroKnowledge 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：

Hash(Data) →Hdata　计算明文 Hash

If CheckDup(Hdata) = TRUE goto 11　如果已经存在相同的数据，转到第 11 步

RandomSym()→Stk 随机生成对称密钥作为文件的存储密钥

Enc(Stk, Data)→EncData 用存储密钥加密文件

Hash(EncData) →Henc　计算密文 Hash

GenKey(Data) →Sdata 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值。

Enc(HData, Stk) →EncStk’ 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。

PutIPFS(EncData)　将加密数据存入到 IPFS

PutMeta(Hdata, Henv，EncStk) 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

Goto 14

GetMeta(Hdata) →Henv，EncStk’ 从全局元数据表中取出密文 Hash 和明文加密后的存储密钥

GenKey(Data) →Sdata 以同样算法从数据明文生成对称密钥

Dec(Sdata, EncStk’) →Stk用该对称密钥解密出存储密钥

Enc(Spub, Stk) →EncStk 以用户的加密公钥加密存储密钥

PutPri(Hdata， EncStk) 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 IPFS 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(Hdata)→Henc 从全局元数据表中通过明文 Hash 获取到密文 Hash

2. GetPri(Hdata)→EncStk 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. GetIPFS(Henc)→EncData 用密文 Hash 从 IPFS 中取出密文

4.Dec（Sprv, EncStk）→Stk 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(Stk, 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：

RandomAsym()→Swpub ,Swprv 随机生成非对称密钥作为写权限密钥

Create(Swpub) →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

RandomSym()→Stk 随机生成对称密钥作为存储密钥

Enc(Spub, Stk) →EncStk 以用户的加密公钥加密存储密钥

PutPri(ID， EncStk) 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：

GetPri(ID) →EncStk 从用户权限表中取出加密公钥加密后的存储密钥

Dec（Sprv, EncStk）→Stk 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

Enc(Stk, Data)→EncData 用存储密钥加密文件

Hash(EncData) →Henc 计算密文 Hash

Enc(Swprv, Henc) →EncHenc 用该 ID 的写权限公钥对密文 Hash 进行签名

PutDyn(ID, EncData, EncHenc) 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：

GetKey(ID) →Swpub 取出该 ID 对应的写权限公钥

Hash(EncData) →Henc 计算密文 Hash

If Dec（Swpub, EncHenc）= Henc Write(EncData) 如果签名验证通过，写入数据

动态数据的读流程如下：

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

GetDyn (ID)→EncData 用 ID 从动态数据存储区读出密文

Dec（Sprv, EncStk）→Stk 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

Dec(Stk, EncData)→Data 解密获得文件数据

5.3.拜占庭容错

拜占庭将军问题当时的主要场景是：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。

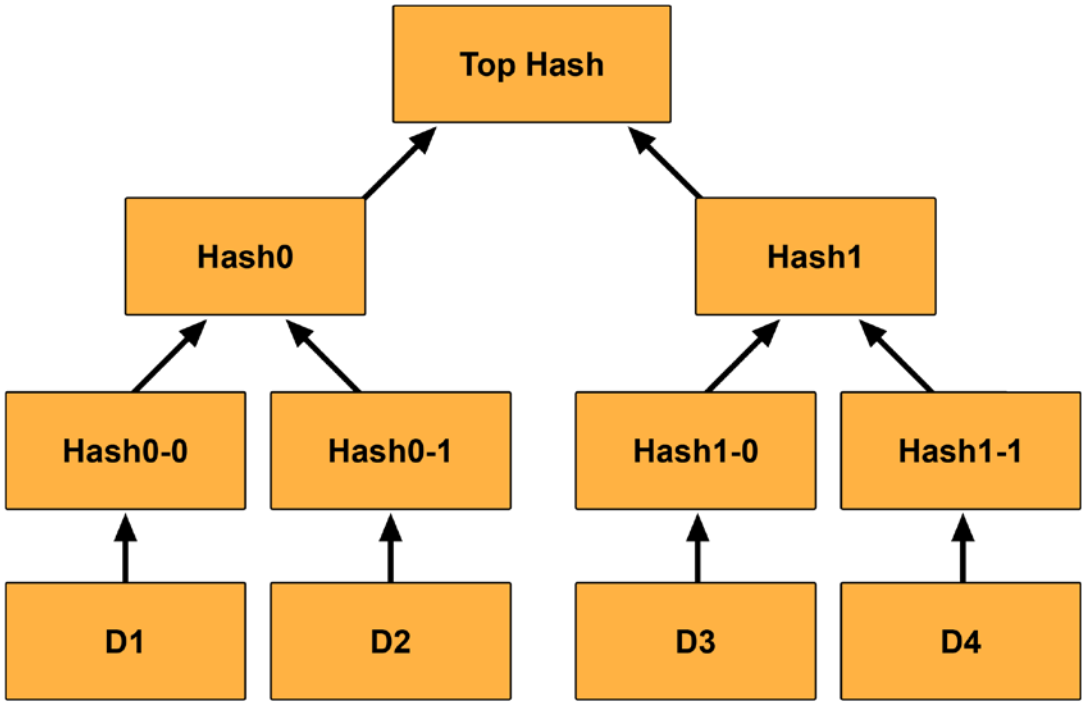
拜占庭容错方案：Put(D,n,m)，当数据上传的时候使用冗余编码将数据分为 n 个碎片，并允许最多 m 个碎片失效（即只要能任意获取 n-m 个碎片即可完整读取数据）指定 n 个存储节点来存储这部分数据，每个节点存储一个碎片，这样可以容忍 m 个节点故障。当故障节点<m 时，文件是可以 Get 成功的，这时候通过修复机制将重新选择新节点代替故障节点存储数据。

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。

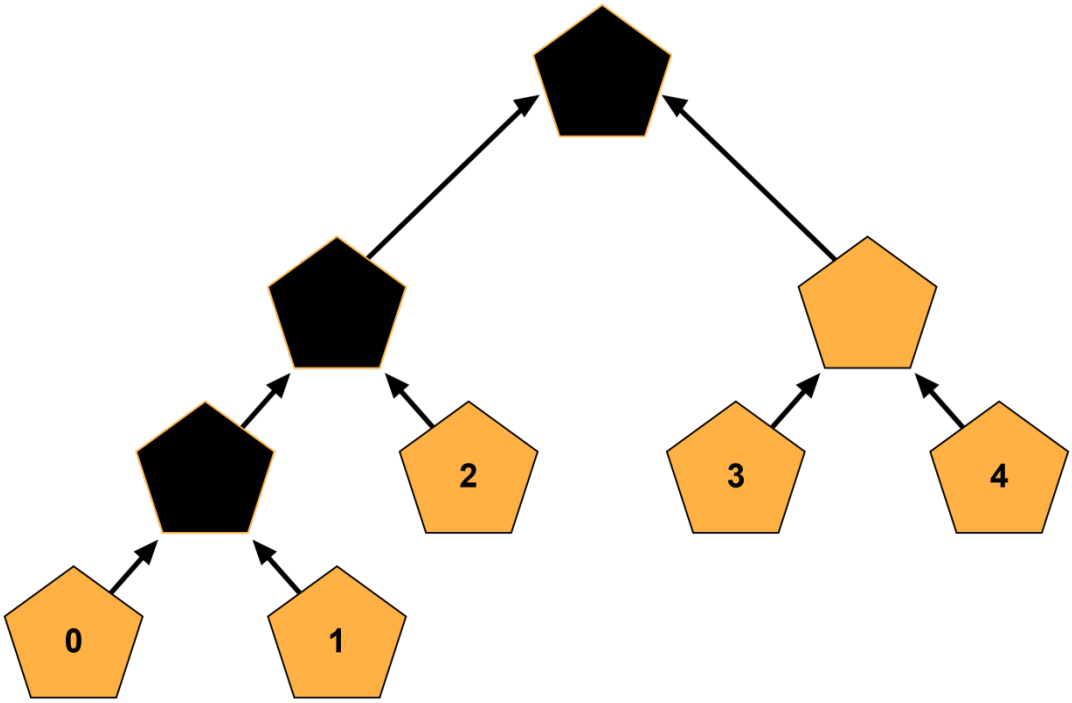
什么是文件头信息 (HeadHash)，先简单了解一下默克尔树 (Merkle Tree)



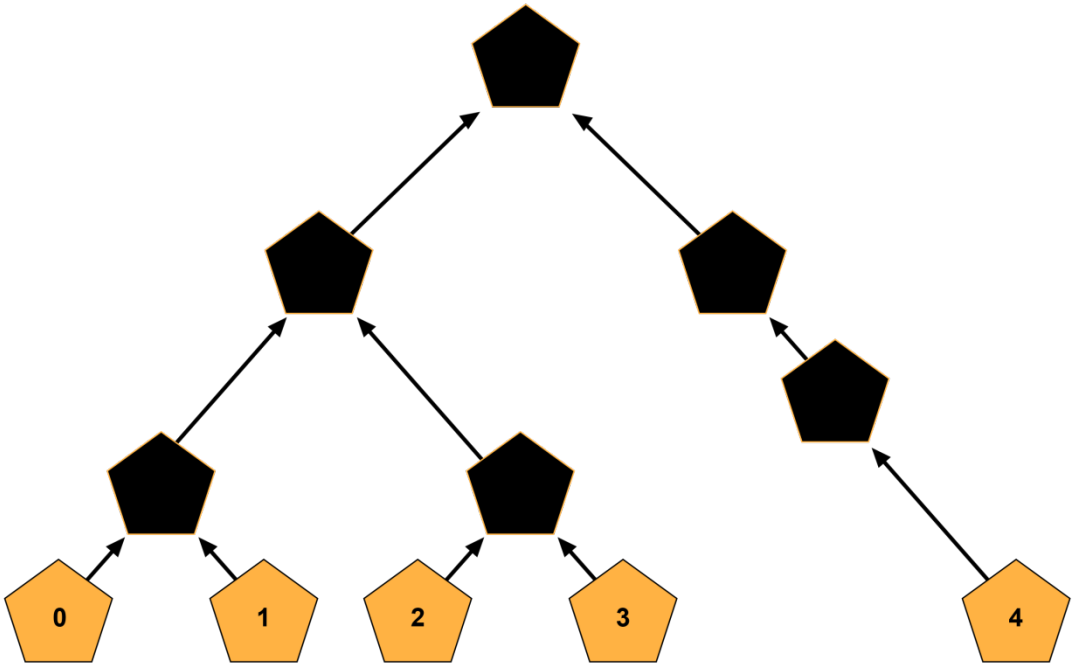
它是装载 Hash 列表的一个树形结构。在树的最底层是已经被切割的具有固定大小的数据小块（除去最右侧小块），　有相应的 Hash 与其对应，相邻两个小块合并再做 Hash，以此上推，至最上层的 Top Hash 也就是默克尔根。

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, Stk, 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。

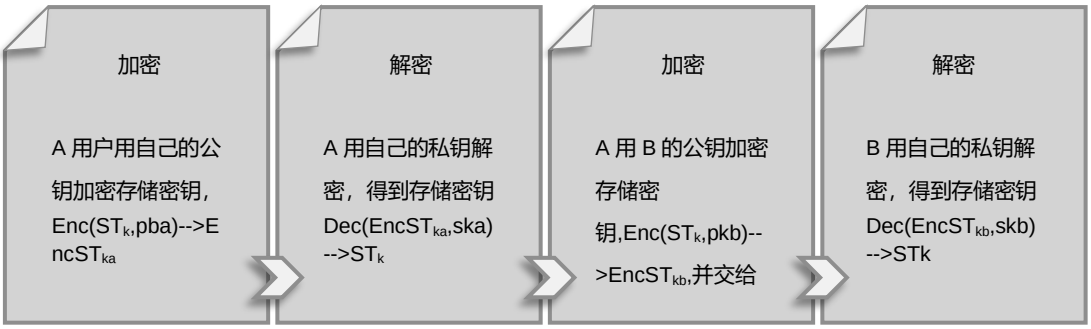
具体的构建过程如图

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 存储与计算网络

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)，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：或者代攻击，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 计算网络

YottaChain 计算网络使商业节点具备通过提供计算挖矿的能力。计算能力的提供以通过开通虚拟机或容器的方式提供，客户需要根据自己对计算能力的要求选择计算资源需求（CPU、内存、GPU 等），符合要求的商业节点开通相应的虚拟机，并以小时为单位租用。以后还会增加以容器方式提供计算能力的选项，以及 FPGA、TPU 等。

YottaChain 计算网络使用算力证明（PoCp）共识机制算力证明允许计算资源提供商证明提供了相应的计算资源池。

当一个 YottaChain 商业节点提供计算资源池时，验证者可随机发起挑战，被验证者采用 POW 方式证明确实提供了相应的计算能力。算力证明采用的

POW 算法与比特币、以太坊等区块链的 POW 算法不同，而且有多种 POW 算法可以随机选择，在社区发展过程中还会随时增加 POW 算法，以保证只有系统规定的 CPU 等计算资源才能完成 POW 任务，避免计算资源提供商采用 ASIC 矿机等方式进行作弊。

算力证明采用的 POW 算法包括验证 CPU 计算能力、验证 GPU 计算能力和验证内存，其中不同类型的 CPU 和 GPU 算法相同但验证的代码不同。在交易市场上，不同类型的 CPU、GPU 也是作为不同的 SKU 进行标价的。

YottaChain 计算网络提供计算市场，允许客户为矿工提供计算而付费。

6.3.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

对于中小企业和创业型公司，他们也有着很强的文件共享、数据备份等方面的需要，但由于自身规模的限制，他们不可能像大企业一样，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，大大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客户资料的手机方便沟通，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 存储网络和 YottaChain 计算网络为基础，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。

和 EOS 一样，全体持币者的投票只能一币一票，而不能一账号一票。

8.2 立法会

YottaChain 实行代议制，由全体持币者投票选举的立法会来制定 YottaChain 的规则。这是因为：

出于效率考虑，因为凡事都全体表决会是缺乏可操作性的。

出于专业性考虑。立法是需要专业性的，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，所有的规则都是由立法会来制定的，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。

如果立法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立法会大体上是符合全体持币者的利益的。之所以是“大体上”，这就是公平和效率之间的妥协，总不能因为偶尔的非关键性的差错而改选立法会。

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，其职责是将立法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立法会任命，受立法会管制。代码规格委员会制定的需求规格必须严格遵守立法会制定的规则，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，其专业性要求与立法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，立法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立法会投票产生的规则不一致的情形，YottaChain 社区会向立法会举报。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、辜负了立法会的信任，立法会自然会严肃处理。万一立法会对此置之不理，则立法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必须认真履行职责。

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，编码委员会也是由立法会任命，向立法会负责。

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立法会任命，向立法会负责的。

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立法会制定的规则不符，不用等立法会采取行动，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 君主立宪

“去中心化”意味着“去创始人化”，但一个创始人从无到有倾注无数心血的项目，凭什么要其撒手？这个问题就有点像集权制到民主制的转换路径。在项目启动时，是由创始人控制的，等项目成熟了就要“去创始人化”，这种转换不那么容易。更何况，创始人彻底离场对这个项目真的是好事吗？

一般说来，没有任何人比创始人对项目倾注的感情更多，没有任何人比创始人更了解这个项目，没有任何人比创始人更想运营好这个项目。将创始人赶走，显然不是一个好的方案。

一种比较好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、选举出第一届立法会之后，就由立法会接管社区，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，可以代表社区出去开会发言做报告（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。

还有一个细节需要注意。在社区刚上线时可能创始人持有最大数量的币，此时需要限制创始人的投票权，使得在投票时的权重以挖矿得到的数字货币为主。

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

IPFS 本身就采用 IPFS 作为 YottaChain 中的去中心化静态持久化存储模块，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 支持动态网页，真正取代 HTTP

IPFS 虽然宣传可以取代 http，但是由于不具备计算能力，只能取代静态网页，不能取代动态网页，所以不具备真正取代 http 的能力，“取代 http”更多只是一个市场宣传口号。

动态网页的应用是非常广泛的，大多数网站或多或少都使用动态网页，要想真正取代 http，就必须解决动态网页的问题。

YottaChain 具备存储和计算能力，不仅可以存储静态网页，而且可以存储和运行动态网页，也就是说可以支持所有类型的网页，从而可以真正取代 http，用 yotta://<网站入口网页指针>可以访问完整的网站。

9.3 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，哪怕是全球最厉害的黑客亲自出马也不行，

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，比现在将数据保存到 AWS、Google、阿里云、百度云等要安全可靠多了，甚至比存到你自己的电脑上都要安全可靠。

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，比你自己买的任何厂商的存储设备都要便宜，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最烂的便宜货，都比 YottaChain 的存储成本要贵很多。

9.4 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。之前共享经济更多的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，但在实践滴滴、摩拜中并

不是利用闲置的交通工具（除顺风车外），而是专门为滴滴、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目的。

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），就更是不计其数了。这些存储和计算资源基本上都是有闲置的（将硬盘全部存满一点都不剩的情况是几乎没有的），如何利用这么多海量的闲置资源就是一个非常大的课题了。

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非常巨大的共享经济系统。

9.5 将自用存储空间用来挖矿

对 YottaChain 来说，除了将闲置资源可以用来挖矿外，正在使用和即将使用的存储资源也是可以用来挖矿挣钱的，而且可以做到数据存储和挖矿挣钱两不误。

举例说，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，完全是白赚。

9.6 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高不可攀。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，从此再也不需要区块扩容了。

9.7 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。

9.8 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。

9.9 去中心化的公有云服务

Airbnb 利用打造共享经济平台的方式短短几年就超过了几百年历史的老牌酒店，成为全球规模最大的连锁酒店。YottaChain 将提供存储、虚拟机、容器等 IaaS 公有云服务，也有望成为全球规模最大的公有云服务商。

与价值几千亿美元的 AWS 和阿里云等公有云服务商相比，YottanChain 有着巨大的价格优势，而且利用区块链的激励模型实现低成本扩张优势。

10、技术团队和顾问

10.1 书生星际与 YOTTACHAIN

书生集团是国际知名的中国老牌 IT 领军企业，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，十多项技术都达到国际领先水平。

基于密码学的数据安全技术和分布式存储是书生集团的主要技术优势，其数据安全产品（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。

TruPrivacy 数据安全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、黑客偷不走。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。

TruPrivacy 还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。依靠 TruPrivacy 技术，可以在所有数据都是密文存储、只有数据的

owner 或其授权者才能访问数据的前提下实现重复数据删除，将存储空间利用率提高了 5-10 倍，实现了贡献存储空间挖矿得到的数字货币可以购买更多的存储空间还能剩余数字货币的奇迹。

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”。

书生星际公司是书生集团中研究区块链技术的公司，负责将集团拥有的各项技术与区块链的结合，并研发新的区块链技术。

YottaChain 是由 YottaChain 基金会负责的区块链项目，书生星际将其拥有的技术和专利授权贡献给 YottaChain 基金会，并参与技术研发工作。

10.2 书生星际核心团队

1. 王东临 创始人/董事长

书生集团创始人、首席科学家，是国际顶级 IT 科学家，中国知名企业家，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，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（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（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 多项专利。

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、

10.3 书生星际顾问

- 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。

- 元道（陈升） 中国区块链主要布道者

中国最大民营 IDC、美国上市公司世纪互联的创始人，区块链、通证名词翻译者，致力于区块链的推广普及、理论研究和实践，通证经济、CoC 共识创始人，中关村区块链联盟理事长， FCoin 币改委员会主任

- 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

- 戴卫 原中关村管委会主任

多年担任中关村管委会主任，曾担任北京市政府副秘书长、北京政协常委，深谙在体制内为创新科技寻求发展空间。

- 杨天行 原电子部计算机司司长

原电子部计算机司司长，原全国信息技术标准化技术委员会主任，第二届国家信息化专家咨询委员会委员。曾担任电子部 15 所副总工、北京信息科技大学校长。

- 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。

- 卿斯汉 国家级密码学专家

国家保密局技术顾问，中科院计算机网络与信息安全管理领导协调小组成员，中国计算机学会计安全专委会副主任和保密专委会副主任，中国人民银行“网上银行发展与监管工作组”成员。

- 华平澜 原北京信息办主任

原北京信息办主任，原北京软件行业协会会长，曾担任北京电子工业办公室总工程师。

- 曾南山 微软亚洲互联网工程院副院长

在微软 20 多年，先后从事操作系统、搜索引擎、人工智能等 Microsoft 的主流技术的研究工作

- 贺卫东 天融信创始人

信息安全专家，创办的天融信公司是中国的信息安全知名企业

- 张向宁 万网创始人

中国第一代互联网创业企业家，创办了中国最大的域名注册和网站平台服务提供商中国万网（后被阿里巴巴收购），兼职担任团中央网络影视中心顾问，中国电子商务协会副理事长，中国互联网协会理事。

- 鲍岳桥 联众创始人

原 UCDOC 总工，创办中国最早的棋牌游戏网站联众，现中关村知名天使投资人

- 鲁瑞清 原海龙董事长

创办中关村电子市场行业的龙头海龙电子城，担任北京中关村电子产品贸易商会会长、中关村科技园区企业家咨询委员会委员。

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。