OF白皮书

之

OFBANK

V1.3

版权声明

望周知。

OFBANK项目理事会

2017年01月03日

目录

术语表

OF：是Order & Freedom的缩写。

OFBANK：支持区块链技术落地的技术平台。

OFCOIN：OFBANK生成的虚拟货币。

OFAPPs：接入OFBANK的第三方应用。

前言

随着移动互联网和社交网络的普及，网络交易已经从网上渗透到线下，从网络商城蔓延至人际社交网络，交易和结算变得高频且无处不在。在新的开放社会中，旧市场固守着一个个封闭的系统，各个封闭系统之间难以建立信任，安全互通的成本高昂。传统产业和传统的交易开始无法适应新的开放社会的高流通需求。

除此之外，封闭市场的衍生增值价值将随着用户离开封闭市场或市场关闭而消失。而随着开放社会不断加速发展，资产的流动和市场的更替变得更快。这样就给越来越多的用户带来了不便和损失。

与此同时，随着数字化社会的到来，交易价值之外的一种巨大的群体性价值正在频繁产生。这种用户溢价价值由于其特有的群体属性，常常因为无人维权和监管而被侵占。生产价值的用户平日里没有得到溢价的回报，却在某些被错误使用的情况下，合法权益被侵犯。

区块链技术既能降低信任的成本、促进资产的安全互通，又能将用户资产保护起来、减少资产被盗风险，还能帮助用户获取更多的群体性价值。然而，在区块链技术的落地过程中却存在一些难以解决的问题：

1、门槛高：区块链资产通常通过私钥来管理，然而有些私钥管理的系统非常薄弱，这致使数字货币私钥丢失或被盗案件时有发生。一旦密钥丢失或被盗，用户将会丧失全部资产。然而由于此问题解决方案的技术门槛过高，用户常常手持毫无安全性可言的区块链资产。

2、落地难：区块链时常见诸媒体却鲜有落地产品。这是因为，当前的区块链系统是一个匿名的去中心化系统，而现实世界是一个实名的中心化世界。如果没有切实可行的方案、没有可靠的技术支持，区块链系统就非常难以与现实世界融合。

基于以上的社会需求和技术问题，我们设计了**OFBANK**虚拟资产管理和结算平台。它基于分布式账户系统，与区块链账本强关联，通过可信计算与区块链技术的结合，使实名制与隐私保护并存；同时，OFBANK拥有完善的激励体系，让社会化力量参与分布式计算和社区建设，保障区块链时代用户自有数字资产的安全流通。**OFBANK**的诞生使得以上的问题都可以得到系统性的解决，并且整个系统能够在内生力量的驱动下自循环、自进化下去。

简介

本文提出了一个虚拟资产管理和流通平台OFBANK，基于一个分布式的账户和账本基础设施(OF基础设施)，以此保障区块链技术在现实业务的落地。

OF基础设施的目的是提供一个隐私保护的分布式账户和账本系统，基于可信计算技术(Trusted Computing)和区块链技术(Blockchain)实现。用户的身份将被严格审核和记录，然后这些账户信息将被分块存储在不同的存储节点上，攻击单独的几个存储节点无法获得任何有效信息，用户隐私极难被泄露，从而解决了传统账户系统容易被攻陷和拖库的弊端。OFBANK通过“手机挖矿”算法，将OFAPPs的用户贡献，通过区块链公正记录并公平奖励，这样，就使得用户在整个区块链生态中的贡献打通，让“天生我才必有用”有了一种数学表达和金融分配算法。

OFBANK采用灵活的用户私钥管理制度，保证用户私钥的安全生成，安全保存，安全使用。用户可以自己保存私钥，做成冷钱包，保障虚拟货币的安全，也可以托管给OFBANK和第三方密钥管理机构，私钥不会丢失。

OFBANK将首先应用于数字化、娱乐化的轻应用，能够形成业务闭环的场景，目前已经有十多个区块链落地场景在对接开发中。应用场景还包括了资产公益基金监管，虚拟资产的分发、管理，投资监管，积分兑换等方面。

1 问题的提出

传统的密码学货币有其天然优势，也有其不安全的一面，比如：

2014年，比特币[5]世界的Mt.Gox事件导致当时价值超过3亿美元的比特币被盗；2016年，以太坊[6]世界的the DAO事件导致360万以太币被劫持，以太坊分叉为ETH和ETC；2017年5月，Wannacry勒索病毒大规模爆发，该病毒感染计算机后会导致电脑大量文件被加密，受害者电脑被黑客锁定后，病毒会提示支付价值相当于300美元的比特币才可解锁。

另一方面，对于非IT行业没有良好的计算技术功底的用户来说，驾驭和使用数字资产、数字货币也有非常高的门槛。

目前虚拟资产领域亟需一种普通用户可以理解和使用的方式，来控制虚拟资产的产生和流通。对于开发者来说，普通的中小企业、个体创业者、大学生试验，也可以通过简便API接口，可以开发他们自己的区块链应用。

因此我们设计了OFBANK系统，来解决这些问题。

1.1 为什么需要实名制

密码学货币逐渐成为计算机犯罪分子获得收益的重要方式，传统密码学货币的匿名性助长了计算机犯罪，因为这让他们犯罪的收益更容易获得。这些发动攻击的人至今也无人可知，受害者的权益至今也没有追回。

那么，虚拟资产的投资者、参与者的权益由谁来保护？如果投资者权益受损了，谁偷走了他们的权益？他们应该去找谁？

匿名虽然带来了部分个体行为的便利性，但是对于群体行为的秩序、规则和互相协作方面也同时带来了隐患。人类社会的大部分行为是实名制度或者基于隐性的实名制的保障基础的。

所以，在数字世界构造信用社会、契约社会的过程中，实名制必不可少，这样，区块链技术才能真正投入到日常生活中，广大实体经济企业才能充分信任区块链技术，区块链才能真正落地。

1.2 为什么需要隐私保护的分布式账户系统

传统的中心账户系统，用户账户的查阅权限是掌握在中心服务器的，用户信息容易被滥用、泄露。用户自己不能控制自己信息的保密性，而是被动的把这种权限放在了第三方。用户的用户名和密码hash存储在单一的文件系统中，第三方需要花费大量资源保护账户系统安全，而且如果系统被攻破，通过拖库撞库，获得用户各种隐私只是时间问题。这类问题每年都层出不穷，特别是各种信用卡信息泄露事件，给用户带来了巨大的经济损失。

而在隐私保护的分布式账户系统中，账户的查阅权限将掌握在用户自己手中，没有用户的允许，单个存储节点无法获得用户信息。

目前市面上的去中心化分布式账户系统，只能做到防篡改，做不到用户自己控制查阅权限，也做不到隐私保护，因为他们是把同一份数据分别存到不同的节点上，这样，用户很难放心把实名相关的敏感身份信息放进账户系统，比如身份证、手机、姓名、住址、信用卡信息等等。用户希望可以放心把这些信息存储在分布式节点，第三方不经授权无法读

取，不用担心私密信息被窃取、滥用、盗用。

1.3 为什么需要智能合约标准化

传统的区块链系统，是一个完全去中心化的架构，而我们认为，中心化和去中心化各有它们的意义。对于公众来说，他们的利益需要得到保护，特别是在智能合约领域，公众不一定了解智能合约的相关信息。而同时，因为业务本身的分类，大量的业务事实上是可以归类到一定的范围内，所以，我们将对智能合约地址进行分类，并标准化，以适应不同的应用场景。

与过去中心化的标准化不同的是，区块链时代的智能合约标准化，因为其开源和自进化机制，可以更快的速度迭代、分支，适应与不同的业务场景和个性化需求。

这种技术和应用进化的方式，跟进化树的模型极其相似，具备天然的强生命力。

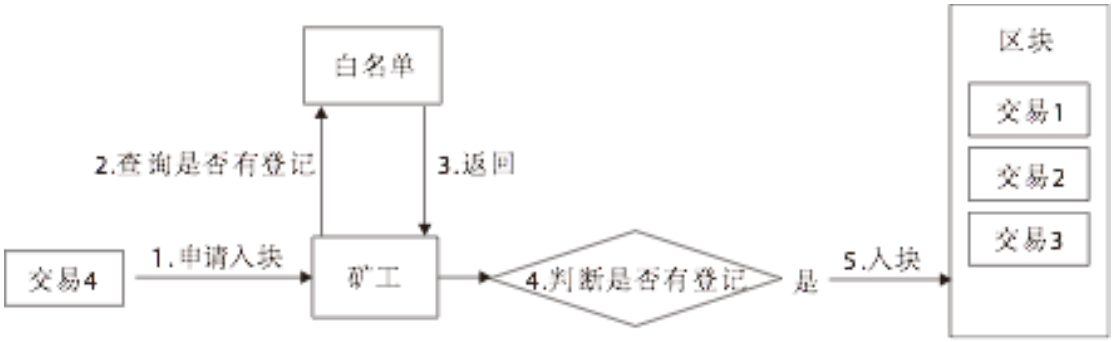
2 设计思路和解决方案

我们已经考察了建立实名制可监管分布式账户和账本系统的三种方法。第一种是建立一种完全重新开发的分布式账户和账本系统，这需要长期的开发迭代。第二种是利用现有的开源的带智能合约的账本系统进行安全增强，然后自己开发新的账户系统。第三种是改造现有的开源分布式账户和账本系统，如hyperledger fabric[7]。这类系统一方面都是联盟链，没有挖矿奖励机制，系统得不到足够的社会支持，另一方面这类系统无法抗击账户服务器被拖库，信息被滥用。目前以太坊的框架是经历了实践检验的，这对我们很有吸引力。所以，我们决定用第二种方法，自己开发一套账户系统，大幅改进以太坊源码，对其进行安全增强（类似selinux之于linux），并将两者对接融合。

OFBANK基于秘密分割技术[8]和可信计算技术[4]构建了一个分布式的、用户隐私保护的、可监管的实名制账户系统，同时将以太坊加以修改，使其总币量固定，gaslimit（最大单位交易费用）无上限，共识协议替换为POT，形成一个抗拥堵、能源友好的账本系统，再将账户系统和账本系统融合，对用户私钥进行管理，形成一个高可用性的虚拟资产管理流通平台。如果说分布式账本系统关注的是资产，是财物，那么分布式账户系统关注的就是人，是用户。OFBANK使得用户在进行数字资产操作和交易时，拥有自己的现实身份，从而使得区块链的“可追溯”有真实的人可追，“不可篡改”的内容有内容的相关人。

在OF基础设施中，所有交易相关人都必须经过账户系统的实名登记和审核，没有登记审核的交易不能入块，也就无法生效和被认可。用户还可以要求交易是被监管的，这时每个验证节点都会将交易提交监管方，得到其认可，监管方确认相关人合法性并签名后，方可入块。

同时，不同于比特股等携带账户信息的区块链系统，OF基础设施在区块上看不到任何相关人的隐私信息，除了公钥hash地址，没有任何其他账户信息，所有相关人的其他账户信息，必须在取得相关人本人许可后，才可以通过地址在账户系统查询。因此，区块链并不会泄露任何账户隐私，公众能看到的，只是一行包含公钥hash地址的消息。



实名制在OFBANK系统中是“隐身实名制”和“授权实名制”的结合，达到系统的可信性。就像在社交网络中，朋友带来的朋友，虽然是不知道名字的，但是我们可以知道他是“实名的”可信的。所以，这种不知道名字和匿名是不同的。每一个用户，都可以通过分布式账号系统，掌控自己的个人隐私信息，并有偿授权给对方。

中本聪的创世论文旨在解决互联网贸易中碎片化交易、不可逆交易的可信性等问题而设计了一个点对点的交易系统，为了让这个系统运转下去，设计了矿工激励体系，为系统提供算力支撑。历史上曾经有很多人设计过类似的结算系统，我们认为中本聪的进步在于，他设计了一个挖矿激励机制，通过矿工之间的算力竞争，支撑了这个系统生生不息的运转。区块链本身有很多特性，我们设计的系统，重在挖掘现实业务中那些需要奖励和需要将奖励价值流通的环节，让整个系统流动起来，这是我们聚焦的突破点。

我们专注于不便用法币衡量的奖励部分，其中包括：1）ofcoin用于奖励的那些不方便用法币来结算（会导致商家赔钱）的资产，比如积分，所以通常会用延期兑换的方法，比如积分够3000可以兑换相应的礼品，而这对用户就有风险和不确定性，当商户倒闭，用户遗忘或不方便使用，或者政策改变时，用户就会产生损失；2）如果用法币结算一些未来有增值空间的物品，过一段时间后，物品升值，卖家往往会觉得吃亏了，如果用ofcoin兑换，相当于用一个有增值空间的资产兑换另一个有增值空间资产，卖家会预期有增长收益。这样，我们是事实上将资产分三种：价值衰减，价值趋于稳定，价值有增值空间。三种资产分别可对应手机电子产品等快消品、黄金、算法/密码学货币/论文/艺术品等。

3 账户

实名制分布式账户系统的开发难点在于，它不是一个简单的数据库系统，每份数据都是被秘密分割的，每个注册管理节点都是满足可信计算标准，可以被远程验证的。我们设计和实现了详细缜密的注册，分割，保存，合并，查阅申请，许可的流程，并维持这个过程的高可用性与保密性。

3.1 注册和账户存储管理服务

整个OFBANK中信息最集中的环节，在于注册和账户管理。因为账户存储的时候是秘密分割的，攻破单独一个节点没有意义，但是注册和管理的时候信息是集中而敏感的，如果注册服务被监听，一段时间内提交注册的账户信息是可以被记录的。这时就要求注册和管理的服务是高度安全可靠而保护用户隐私的，我们认为可信计算技术最能契合这种需求。

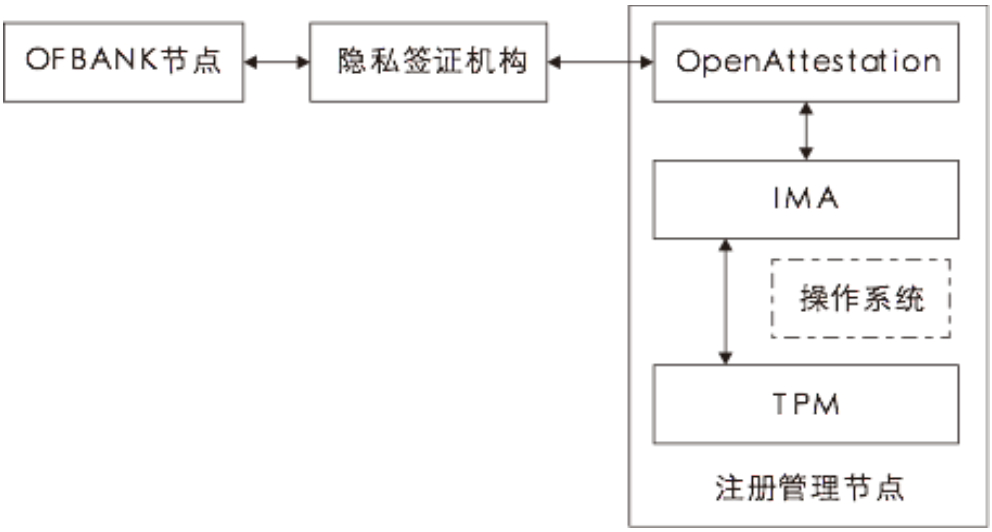
可信计算技术和区块链技术的共同点在于，他们都提供了一个信任环境，不同的是，区块链是通过大多数人的共识实现的，这就要求大多数人知道具体的信息并作出决策，而

可信计算技术是通过绝对的隐私信息（无法被计算机软件读取的根密钥），不向外披露任何隐私信息，仅通过密码学向外证明实现的信任。

可信计算的难点在于对计算环境的要求是严苛的，普通用户难以构建这种计算环境，这也是可信计算推广多年却不被公众所了解和应用的原因。但如果一个计算环境作为专用的服务，不进行其他活动，对普通用户透明，那么对计算环境的严苛要求就不再是一个问题。

所以区块链技术更适合应用在常规的计算环境中，用来处理那些需要众人参与，需要达成共识的事件，比如合约，只有大家都看到了、认可了，这个合约才有意义。而那些商业机密的讨论，关键相关人员的社交网络，谈判中的冲突，并不一定适合为公众所知，而处理这些隐私信息正是可信计算的优势领域。

我们目前使用了基于TPM2.0的可信计算技术构建账户注册管理系统，通过IMA完整性度量和OpenAttestation 远程证明来对节点进行度量和报告，公开验证注册系统的完整性。



这要求账户注册管理节点拥有符合TPM2.0标准的硬件芯片并加载IMA和OpenAttestation，保持操作系统和软件版本符合严格的规范，可以被识别和度量。需要注册服务的用户可以通过隐私签证机构（PCA）验证证实平台TPM的有效性，此后证实平台收集平台摘要信息并度量和报告。

存储信息分发到各个节点之后，注册管理节点需要维护这些节点，测试他们的可用性，当存储某个账户信息的一定数量节点不可用，不安全，或超过一定限期之后，要求存储节点重新向注册管理节点汇总此账户信息，删除存储节点的原有信息，重新分配存储方案并实施。

3.2账户存储

基于秘密分割的账户存储系统是OFBANK账户系统的基础。注册管理节点收集和验证注册信息之后，将信息切割成固定长度的分组。通过shamir秘密分割算法分割这些分组。基本的shamir秘密分享算法如下：

首先选择有限域F（q），q>=n。设参与者集合为P={P1，P2，……，Pn}，k为门限值，秘密信息s。选择F（q）上的n个互不相同的非零元素x1，x2，……，xn，公开这些元

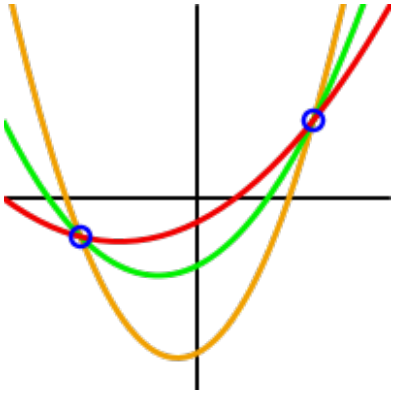
素。

随机选择F（q）上的k-1次多项式g(x) =a0 + a1x + … + ak-1x^(k-1)，其中a0=s，也就是秘密信息，其余的ai随机的选择自F（q）。分别计算si= g(xi)，i=1，2，…，n，将(xi，si)作为子秘密分发给成员Pi。

任意k个成员可以将其持有的子秘密共享，从而通过拉格朗日插值公式恢复出子秘密s。设k个成员的子秘密为{(xi1，si1)，…，(xk1，sk1)}，拉格朗日插值公式如下：

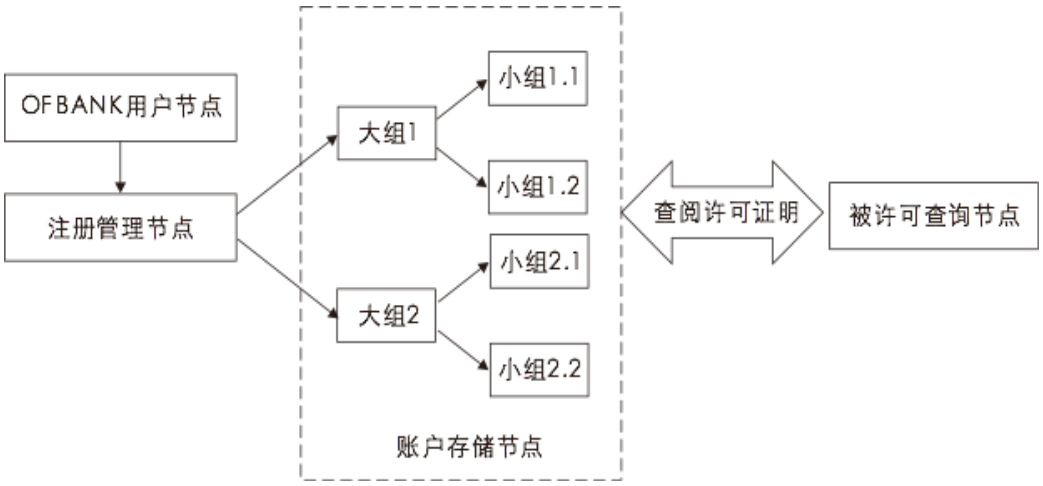
$$g(x)=\sum_{r=1}^{k} s_{i_r} \prod_{\substack{t=1 \\ t\neq r}}^{k} \frac{x-x_{i_t}}{x_{i_r}-x_{i_t}}$$

由多项式理论可知，若两个k-1次多项式在变量的k个不同取值处得到的函数值相等，则这两个多项式必定相等，于是上式成立，由此计算出s=a0=g(0)。



我们将信息分为M份(M>0)，其中P份可以还原(0<P<M)，分给M个大组，然后再把每份分成N(N>0)份，其中Q份可以还原(0<Q<N)，分成N个小组，指定M*N组账户存储节点保存这些信息，对每个节点，发送对应的秘密分块时，通过节点公钥进行加密，以防止监听。

账户存储节点收到信息之后存入数据库，当收到合法的账户查阅许可证明后，将对应账户信息分块用请求方公钥加密，传输给请求方。



当此情形，只要诚实节点控制了网络，账户存储机制就是可靠的，只有当大部分大组中的大部分小组都被攻击者占领，账户存储网络将变得脆弱。用户可以要求自己的账户存储平台拥有和注册管理节点类似的可信平台并做远程证明，来保证自己账户信息的高隐私性。

3.3 账户查阅许可

在大型的p2p网络中，授信查阅账户信息的流程，需要考虑诸多问题：首先，在p2p网络中，如果一个账户想查阅另一个账户的信息，任意两节点之间如何通信？然后，B如何才能得知，确实是A请求获得自己的信息而不是有人假冒A来骗取信息？还有，B需不需要先知道A的具体信息，才能放心的给A自己的信息？还有，如果B允许A查看自己的信息，是要永远都让A可以查看自己的信息吗？如果B未来的变化不想让A知道怎么办？

我们对以上问题的解决方案如下：（1）节点如果想和其他任意节点通信，需要通过消息中转服务器主动推送。（2）A在请求获得B的信息时，必须对请求加以签名，这个签名是针对请求的hash的。（3）A可以在请求B的账户信息的消息中加入B对自己信息的查阅许可，并对整个消息的hash签名，这样，B在收到请求消息的时候，可以拿着带查阅许可的消息和A的签名，先去账户存储节点请求获得A的账户信息，当B得到A的账户信息并验证后，再发送给A自己的账户查阅许可签名。（4）A在发送账户信息查阅请求时，需要加入时戳和时长，即只能在一定时间内查看对方的账户信息，从而B颁发给A的许可是带有失效时间的。为了设计的简单，如果A的请求包含查阅许可，B查阅A的有效期和请求有效期相同。 而且为了防止协商具体内容导致的拥塞和攻击，如果B对请求的具体内容比如查阅时长有异议，B可以选择不应答，也可以简单的向中转服务器发送拒绝。

A发送的请求格式如下：

请求标记	A的地址和公钥	B的地址和公钥	请求的账户信息的范围	时戳	可查阅时长	是否允许对方查看自己信息	签名

B发送的许可证明格式如下：

许可标记	A的地址和公钥	B的地址和公钥	请求的账户信息的范围	时戳	可查阅时长	签名

有两种信息可以用来从账户信息存储节点获得账户信息，一种是提交带查阅许可的请求，另一种是提交许可证明。特殊的，如果用户想向全网公开自己的账户信息，可以主动提交被许可地址为0的许可证明。

因此如果账户A想请求获得账户B的信息并获得了许可，查阅了B的账户信息，计算机处理的完整流程如下：

1）账户A向中转服务器提交带自己签名的查阅请求。

2）中转服务器用A的公钥验证签名，如果验证通过，当B上线后转发给B。

3）B收到请求后再次验证签名，如果请求里有A的查阅许可，可以选择用A的查询许可向账户存储节点查阅A的账户信息。

4）B可以选择同意或拒绝A查阅自己的账户信息，如果拒绝，可以选择不应答或向中转服务器发送拒绝请求，如果同意，向中转服务器发送带签名的许可证明。

5）中转服务器用B的公钥验证B的签名，如果验证通过，当A上线后转发给A。

6）A收到B的许可证明后，验证签名，用B的许可证明向账户存储节点申请查阅B的账户信息。

7）账户存储节点收到A的查阅请求后，验证B的签名，时戳和时长，如果签名无误，许可未失效，验证通过后，向A发送B的账户信息分块。

这个流程看起来很复杂，不过用户不需要知道这些细节，所有细节流程都是计算机自动完成的。

申请查阅的实现如下：

```
func (of *OfStructureAPI) ApplyForThePremission(aAddress, bAddress common.Address, infoRange []string, share bool, useTime, password string) (bool, error) {


  account := &accounts.Account{
    Address: aAddress,
  }
  if password == "" {
    return false, &enterPasswordError{hit: "please enter the password"}
  }
  fmt.Println(infoRange)
  _, key, keyErr := fetchKeystore(of.am).ExportKey(*account, password)
  pubKey := crypto.FromECDSAPub(&key.PrivateKey.PublicKey)
  if keyErr != nil {
```

```
        return false, keyErr
    }
    timeStamp := time.Now().Unix()
    data := []interface{}{
        new(big.Int).SetInt64(timeStamp).Bytes(),
        aAddress,
        bAddress,
        infoRange,
        share,
        useTime,
        pubKey,
    }
    clearText, r, s, err := generateSign(data, key.PrivateKey)
    if err != nil {
        return false, err
    }
    httpClient, httpErr := rpc.DialHTTP(testapi)
    if httpErr != nil {
        return false, httpErr
    }
    var respErr error
    if err := httpClient.Call(&respErr, "ofworld_applyForThePremission", clearText, r.Bytes(),
s.Bytes()); err != nil {
        fmt.Println(err)
    }
    if respErr != nil {
        return false, respErr
    }
    httpClient.Close()
    return true, nil
}
```

查阅请求和生成许可是可以指定许可的域的（infoRange），也就是说，可以只申请和许可账户信息中的某一部分，比如，我只需要知道对方的姓名，而不求得到对方的身份证编号和手机号码。这样，许可方就不必担心这个许可的行为泄露了过多自己的隐私。许可生成的具体实现如下：

```
func (of * OfStructureAPI) GeneratePermission(bAddress common.Address, index int,
password string) (bool, error) {
    list := of.ApplicationList(bAddress)
    if list == nil {
      return false, &noApplicationsError{hit: "no applications"}
    }
    singleApplication := list[index]
    permissionTo := singleApplication.AAddress
    rangeInfo := singleApplication.InfoRange
    account := &accounts.Account{
      Address: bAddress,
    }
    if password == "" {
      return false, &enterPasswordError{hit: "please enter the password"}
    }
    _, key, keyErr := fetchKeystore(of.am).ExportKey(*account, password)
    pubKey := crypto.FromECDSAPub(&key.PrivateKey.PublicKey)
    if keyErr != nil {
      fmt.Println(keyErr)
      return false, keyErr
    }
    context := []interface{}{
      bAddress,
      permissionTo,
      singleApplication.TimeStamp,
      singleApplication.UserTime,
      rangeInfo,
      pubKey,
    }
    httpClient, httpErr := rpc.DialHTTP(testapi)
    if httpErr != nil {
      return false, httpErr
    }
    clearText, r, s, signErr := generateSign(context, key.PrivateKey)
```

```
        if signErr != nil {
          return false, signErr
        }
        var respErr error
        if err := httpClient.Call(&respErr, "ofworld_savePermission", permissionTo, clearText,
r.Bytes(), s.Bytes(), index); err != nil {
          fmt.Println(err)
        }
        if respErr != nil {
          return false, respErr
        }
        return true, nil
      }
```

得到许可的节点需要轮询那些账户信息存储节点来获得各个分片信息，使用许可获取用户信息的实现如下：

```
    func (of * OfStructureAPI) RequestInfo(aAddress common.Address, permissionIndex int)
(ofworldtype.UserInfo, error) {
        var userInfo ofworldtype.UserInfo
        httpClient, httpErr := rpc.DialHTTP(testapi)
        if httpErr != nil {
          fmt.Println(httpErr)
          return userInfo, httpErr
        }
        if err := httpClient.Call(&userInfo, "ofworld_requestInfo", aAddress, permissionIndex); err
!= nil {
          fmt.Println(err)
          return userInfo, err
        }
        fmt.Println(userInfo)
        return userInfo, nil
      }
```

各个存储节点为了保证传输安全，需要对传输的分片信息用接收方的公钥加密，使得只有接收方可以打开和拼接这些信息。

3.4 收费策略

注册管理节点收费

注册管理节点和账户存储节点可以收取一定的费用，这样一方面可以支付维持节点运行所需成本，也可以防止女巫攻击，即开启大量节点来访问从而拖垮服务器。用户可以多花一些费用，一次性提交很多地址用于注册，来提高隐私性。

账户查询收费

虽然交易双方原生是不知道对方信息的，但双方都知道，对方是有着实名身份的经过认证的人。用户如果想主动透露自己的信息并获得收益，也可以对自己的个人信息进行标价，通过主动和自动的授权，其他用户可以购买他的一些方便透露的个人信息。目前虚拟世界的现状是，用户信息被无授权的售卖和滥用，用户却得不到任何收益，我们希望用户可以主动控制自己的信息并获取应得的利益。我们会为用户提供一个算法平衡的定价区间，因为如果用户定价太高，他会失去很多本可获得的朋友，如果定价太低，他的个人信息就会大范围的泄露。

4 账本

4.1 交易费用调节

我们认为以太坊的gaslimit机制导致了网络的严重拥塞，比特币简单的价高者入块的方法是简洁有效的交易费用定价方式。

同时，我们通过动态的价格调整来进行负载均衡，避免发生交易量的尖峰。账本系统会记录15分钟内的交易量，如果总交易量大于满载交易量的40%，则提高gas price（智能合约单位交易费用），如果总交易量低于满载交易量的5%，则降低gas price直到基值。

同时我们提高了智能合约的创建和运行费用，提高了创建智能合约的门槛。

4.2 特殊智能合约的设计

在OFBANK中，所有的交易与传统区块链系统一样，都不可更改、回溯和作废，因为这更多的和私人有关，变更和回溯会引起诸多问题。而一些面向公众的业务，比如公益性智能合约不一样，它是面向公众的，公众的利益必须得到保护，所以智能合约必须能够做到透明和公正。

在我们的系统中，公益性的智能合约必须披露相关信息，对合约内资产提交托管方案，配合监管方验证项目真实性，并取得相关资质后，方可将智能合约入块。

社区可以通过投票的方式对智能合约审核，让特殊智能合约暂停、恢复运行或作废，这对规范公益类智能合约的发布和运行是重要的。

4.3 会话密钥（Session Key）

用户需要减少账户私钥的使用频率，保证私钥使用的环境安全，比如在完全离线情况下签名，并在使用私钥后抹掉存储系统。日常的使用可以通过会话密钥（Session Key）来实现，这是一个业界通用的做法。用户可以通过账户私钥生成会话密钥并颁发一个带有效期的证书来向其他用户证明自己的身份，而减少使用账户私钥的频率。同时如果会话私钥泄露，也可以通过签发证书作废申明来作废会话密钥证书。

4.4 注册，标记与分组

首先，公钥hash生成的地址是不能直接在OF基础设施使用和入块的，所有地址必须经过审核注册，生成用户ID和填写个人信息。然后，根据不同的评审级别和应用环境，用户会被标记和分组，这些标记和分组，一方面用来供智能合约判断是否允许运行，另一方面供分布式账户的存储系统使用，同一分组中只能存储相同的一份账户信息秘密分割的数据。

4.5 消息与消息处理

OF基础设施的消息和以太坊在消息的构成和处理上有许多重大不同：

第一，OF基础设施的消息签名是用会话私钥签名而不是账户私钥。

第二，OF基础设施消息需要指定是否需要监管，监管方是谁，并在入块时，消息中需要包含监管方签名。特别的，如果是创建需要使用特定函数如收款的智能合约消息，则必须获得节点认可的监管方的签名。获得监管方签名意味着整个智能合约的创建和运行是被监管方监管的，智能合约的发起方必须向监管方提交申请，登记备案智能合约及相关产品，披露相关信息，对合约内资产提交托管方案，配合监管方验证项目真实性，并取得相关资质后，方可获得监管方签名并将智能合约入块。

第三，智能合约相关消息可以进行用户访问控制，使智能合约只能在特定的用户组中运行。所以即使是未来才会创建的账户和地址，一个智能合约也可以判定该账户是否可以执行本智能合约，进而矿工可以判定是否可以入块。

第四， 矿工在处理所有的消息时，需要经过白名单系统审核，确定当前消息的相关人是被登记在案的，不在案的一律禁止入块。

第五，OF基础设施增加了监管消息，包括对智能合约的暂停，恢复运行和作废。

故而OF基础设施相比于以太坊，更接近操作系统和安全操作系统[2][3]的概念，对于不同的用户组，可以定义不同的存储空间访问权限，实现操作系统安全中的自主访问控制(DAC)和强制访问控制(MAC)。

白名单录入的实现如下：

```
func (table *Table) PutWhiteList(aAddress common.Address) error {
  table.mutex.Lock()
  defer func() {
    table.mutex.Unlock()
    table.db.close()
  }()
```

```
    var existAddress []common.Address
    whiteList, err := table.db.getWhiteList()
    if err != nil {
      if err.Error() == "leveldb: not found" {
        existAddress = make([]common.Address, 1)
        existAddress[0] = aAddress
      }
    } else {
      if jsonErr := json.Unmarshal(whiteList, &existAddress); jsonErr != nil {
        log.Error("put white list json unmarshal err: ", jsonErr)
        return err
      }
      existAddress = append(existAddress, aAddress)
    }
    fmt.Println(existAddress)
    whiteListByte, jsonMarErr := json.Marshal(existAddress)
    if jsonMarErr != nil {
      log.Error("put white list json marshal err: ", jsonMarErr)
      return jsonMarErr
    }
    if putErr := table.db.putWhiteList(whiteListByte); putErr != nil {
      log.Error("put white list db put err", putErr)
      return putErr
    }
    return nil
  }
```

其中，账户存储节点不能提供给任意节点每个地址具体的账户信息，但是所有注册地址的列表，是可以提供给任意OFBANK节点的。

白名单检查和selinux系统的钩子函数（hook）非常类似，就是在实施前做权限的检查，具体实现如下：

```
  func (table *Table) CheckWhiteList(address common.Address) (bool, error) {
    table.mutex.Lock()
    defer func() {
      table.mutex.Lock()
```

```
    table.db.close()
  }()
  existWhiteListByte, getErr := table.db.getWhiteList()
  if getErr != nil {
    return false, getErr
  }
  var exsitWhiteList []common.Address
  if jsonErr := json.Unmarshal(existWhiteListByte, &exsitWhiteList); jsonErr != nil {
    log.Error("get white list json unmarshal err: ", jsonErr)
    return false, jsonErr
  }
  if find := findAddress(address, exsitWhiteList); find {
    return true, nil
  }
  return false, nil
}
```

5 OFBANK的作用

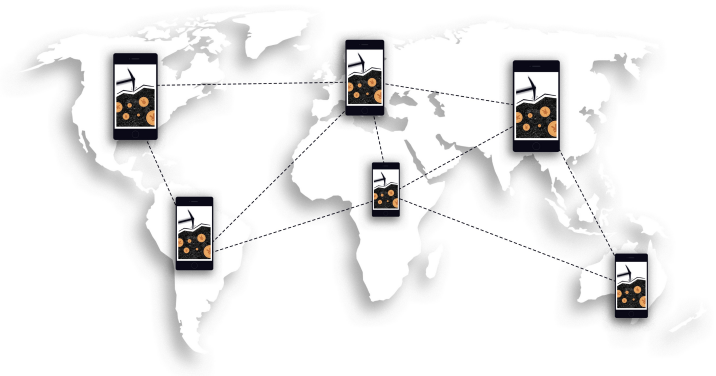
5.1 基础设施支持

OF基础设施为用户提供了安全完善的虚拟资产流通环境，如果说在比特币和以太坊世界，用户会担心自己私钥被盗，丢失，会担心自己参与智能合约时自身权益得不到保障，在OFBANK，这些担心会大大减少。私钥即使被盗，盗窃人也容易追查，即使丢失，也可以取回，智能合约都是经过自己信赖的监管方批准并时刻监督的。这样，用户才能放心的管理虚拟资产，虚拟世界才会更加繁荣。

5.2 ofcoin的手机挖矿算法

OFBANK采用COW算法产生统一的虚拟货币ofcoin，ofcoin支持手机挖矿。拥有更多ofcoin的用户可以创建第三方应用并将挖矿所得的一部分分配给自己应用的用户，另一部分作为自身收益，我们的自有应用也会这样做。并且，我们将给接入的优秀第三方应用分配一定量的ofcoin，如果其他ofcoin的持有者看好这些应用，可以对这些应用投资ofcoin并享受收益。这样，整个OFBANK世界的人会维持系统的稳定运行，这和现实世界的资本市场非常相似，资本通过运营获得更多资本，也支付了员工的薪资。

今天人们的生活重度依赖手机，手机冗余的算力和网络带宽，可以支持用户边玩手机边挖矿。在OF系统中，用户可以通过手机挖矿随时随地赚取ofcoin，并通过ofcoin兑换各类资产，这样社会资源得到更好利用的同时，极大增加了OFAPPs用户的收益。



5.3 用户私钥保管

密码学货币的用户钱包私钥存储一直是一个难题，私钥丢失事件时常发生，OFBANK为用户提供了灵活可靠的私钥存储方案，保证了私钥的安全生成，安全保存，安全使用。

第一种方案是纯粹用户自己保存私钥，这种情况下，公私钥对是由用户生成的，最好是离线在可信环境下操作，保证私钥生成的安全。生成之后离线保存，离线签名，消除所有以后会在线的设备的数据。这对实现冷钱包这种极高安全性的方式是最有利的。

第二种方式是用户和OFBANK及若干第三方密钥托管机构共同管理私钥。这种方式和下一种方式都是服务端生成公私钥对，服务端通过可信环境生成私钥，通过HSM硬件加密技术保存私钥，客户端通过可信计算芯片保存私钥。通过秘密分享技术或多重签名技术，日常可以由用户参与多重签名或私钥的拼装，而如果用户私钥丢失，通过严格的验证流程，用户可以重新生成私钥。

第三种方式是纯线上托管私钥。首先绑定用户设备，以后通过传统的验证技术如指纹，安全码等验证用户身份。

5.4 应用

5.4.1 游戏及游戏化社交

我们认为游戏等虚拟世界活动的产出适合作为密码学虚拟货币的分配度量，相比于纯粹消耗电力和算力，这是更有趣而有意义的事情。同时，传统网络游戏卖道具的方式，很难让游戏的经济系统保持平衡，玩家在虚拟世界中的劳动得不到社会的认可，我们希望改变这个局面。

虚拟世界游戏

我们用UE4引擎创造了一个开放式虚拟世界。玩家投入的时间精力，在游戏内获得的成就，都会被度量和定价，玩家在游戏内的产出可以和OFCOIN互相兑换。

明星社群激励

将明星粉丝看作一个社群，我们会用游戏化和社交排行榜的方式，量化明星粉丝的社群贡献，用数字货币奖励。

5.4.2资产确权和透明流通

OFBANK的智能合约既拥有传统区块链“不可篡改”的属性，又因其后台实名制社群落地的问题，可应用于资产的确权和流通。而一些特殊的行业，比如基金会、拍卖行、公益募捐、资源分配等场景也可使用。

公益基金会使用OFBANK的智能合约，从发起、捐款到分配，每一笔的来源和出处都会被严格管理和记录，每一笔花销都会严格按照智能合约所规定的条款执行，使得公益基金的管理更安全、更简单、更阳光，如果分配流程有违规行为，可以追查出违规使用的资金的流向和相关人，解决了群众对公益基金分配的信任问题。

5.4.4 其他数字资产和OFCOIN之间的交换

OFBANK不是一个“商业银行”，而是一个结算系统，可以用来结算那些有价值而不方便定价或者不方便以货币进行结算的资产，通常这些资产会以奖励或者激励的方式出现，在线上对这些资产所有人进行传统的验证是相对耗时的。在OFBNANK系统中，我们用全球发行的数字货币OFCOIN作为载体流转这些资产，用隐私保护的账户系统验证资产所有人。我们用以下实例说明它的应用场景：

孤岛奖励的价值流动

还以积分为例，如某用户在A商城的积分奖励不能在B商城的兑换物品，而如果该积分是OFCOIN，则对用户来说，便捷性将会大大提高。表面上看起来，A商城的用户可能流失到B商城，但是B商城的用户也可能会流向A用户，这种用户之间的流动加速，会促使商城发展出更多特色，更好的服务用户的多样性需求。而那些不愿意参与这种流通竞争的商城，将会很快被淘汰，这也是社会学中所常说的“孤岛灭绝”规律。

传统交易模式中，用户的权益难以互通，常常造成社会资源的浪费，是商家和用户们的极大损失。OFBANK是开放的结算平台，无论是各种会员卡里的钱、消费次数、会员权益，还是酒店、商场、超市、电商、网站、论坛、航空公司、信用卡积分、APP内积分等都能与OFCOIN互通结算。最关键的是，OFBANK可以原生核实账户，从而用户不必跑到线下实体店去验证身份，也不用向第三方应用方再次证明自己的身份，直接简单的提交账户查询许可，第三方就可以远程核实账户并开展互通结算。

假设A太太因工作需求将从上海调到北京生活一年，在此期间A太太的美容卡中的次数因有时效、有地域限制将失去效用，而美容院也将因为A太太不在持续到店失去忠实用户和其他产品的潜在消费者，双方都蒙受损失。在传统区块链系统中，用户需要线下核实或者线上向第三方核实身份，而在OFBANK系统中，A太太可以直接向美容院提交账户查询

许可，美容院自动验证其身份，记录其账户地址，将其积分转化成OFCOIN并发送到对应账户地址，该店的美容权益也被B太太用OFCOIN买走，A太太和B太太都可以继续享受权益带来的美好体验，而美容院也发展了新的用户。

这是私有价值转为公有价值，再由公有价值转为私有价值的一种过程。这个过程将盘活大量被弃用的健身卡、家政卡、洗衣卡、各种商超、论坛、电商间的积分和优惠等用户权益，完成资源再分配。

更开放、更公平，积极参与竞争的商家有更大可能迭代升级，那些在传统互联网时代的商家，有机会借助区块链时代的新模式率先进化。

主体灭绝导致的价值湮灭

同样以商城为例，假设A商城因某种原因倒闭，那么在过去的模式中，用户在A商城的积分将无法在A商城以外的商城使用，用户的积分将会随着A商城的倒闭而湮灭，这对用户来说是不公平的。所以当有了一个统一数字货币ofcoin来替代积分之后，这种对用户的奖励，仍将有效。

大量的碎片化的不可逆或不方便用法定货币标价的交易

在实际的生活中，存在大量的不可逆的交易，或者在大部分用户的认知或者付费习惯中，会不愿意为某种无形的、专业经验类型的付出货币，比如询问律师”朋友“一个问题，比如请某人帮介绍认识一个他的朋友，这类帮助，从经济行为上来说，也是一次交易，然而这种”交易“在实际生活中付出法定货币的话，会给单方或者双方都带来一定的社交压力，而且，往往难以定价。如果此时用基于区块链的ofcoin来赠予报答给对方，既时尚，也不会给对方带来尴尬和压力。

以上的事例讲述了现实生活中奖励价值流动阻滞，给用户带来的损失。我们认为，用户正向的付出，都是对市场流通的贡献，这种贡献在不同的应用环境中，都会以不同的方式体现出来，不论是积分、游戏道具、排行榜，乃或者说是一句夸奖和称赞，这些东西都曾激励和加速社会的流转速度，今天数字化、移动化和全球化的浪潮将社会拉平，我们有机会将这些价值用一种数字资产的方式流动起来，同时，我们希望用户把时间精力投入在美好的事情上，而不是在每个小事上去证明自己的身份。

我们认为这些额外的奖励和激励，包括我们提供的服务，是人们感受小幸福的点点滴滴，故而，我们给of起了一个中文名字“福”，ofcoin就叫“福币”，“福币”是快乐和幸福的源泉，福币流到的地方，会给人们带来小小的幸福。这也是中国文化对世界和平深厚的祝福。

5.4.5 其他第三方应用OFAPPs

OFBANK将对外开放源代码，开发交互接口，并且给予应用开发者指导，帮助他们在OFBANK开发第三方应用程序。社区应用发行的数字资产与ofcoin的兑换，遵照市场波动，这和以太坊是一致的。

在OFBANK发布的应用将受到监管方的审核、管理和保护，如果发现有事件侵害了第三方应用的开发者和用户的权益，OFBANK将协助监管方，利用实名制账户账本系统，追溯和保护用户的应有权益。

我们会给优秀的第三方应用分配一定数量的ofcoin帮助他们成长，所有第三方应用都可以通过账户余额的ofcoin参与POT挖矿获得收益，并将收益奖励给自己的用户。

我们相信，广大第三方应用开发者会同他们的用户一道，维护OF系统的生态环境，促进OFBANK健康发展，同OFBANK一起成长壮大。

6 未来开发计划

6.1 DAG

随着交易量的不断增加，范围不断扩大，单链的区块链系统终将发生拥塞，为解决此问题，OFBANK将扩展为基于DAG的系统。DAG全称是“有向无环图”，在DAG系统中，每个用户都可以提交一个数据单元，数据单元里可以有交易、消息等信息。数据单元间通过引用关系链接起来，从而形成具有半序关系的DAG（有向无环图）。DAG的特点是把数据单元的写入操作异步化，大量的钱包客户端可以自主异步地把交易数据写入DAG，从而可以支持极大的并发量和极高的速度。

6.2 嫁接

OFBANK支持嫁接，即通过一一映射关系，将其他公链的地址和公钥对应到OFBANK地址上，从而其他公链的用户通过简单的地址登记和转换，就可以映射一个OFBANK地址并使用。

6.3 原生跨链支持

对于区块链通信的孤岛状态，目前已经出现了一些跨链概念验证并取得了实质性的进展。而另一方面，区块链系统本身需要为跨链技术提供更简便的对接，更好的支持，进而实现原生跨链。OFBANK将为跨链技术实现通用协议与接口，为链间通信和交易提供了原生支持。从而链与链之间简便的通信和链间协作的便捷开发成为可能。

6.4 Token升级为公链

基于DAG技术，OFBANK可以实现将智能合约Token升级为公链，从而满足更多的场景需求并支持代码升级。Token升级后的公链通过多hash指针与其他区块实现互认。矿工可以同时在多条链上并行工作以提高收益。

6.5 链群

通过DAG，嫁接，Token升级和原生跨链支持，OFBANK将实现进一步去中心化的链群结构，使各类区块链网络形成一个信息和价值互联互通，互相协作的大链群系统。区块链的孤岛形态将得以消融。

6.6 COW共识算法

传统的公链普遍采用POW或POS共识算法，通过用户工作量或权益来衡量用户应得收益。OFBANK将采用新的COW共识算法，实现工作量，资本和规则的有机融合。这种算法可以更好的反映现实世界的规律，从而更好的服务现实应用。

7 团队

团队现有开发人员30+人，精通区块链技术和信息安全架构。团队成员来自北京大学，清华大学，卡内基梅隆大学，复旦大学，哈尔滨工业大学，约克大学，Academy of Art University，SanJoseStateUniversity等国内外一流院校，成员曾任职于Google、腾讯、百度、IBM、AMD、微博等知名企业和组织。团队秉承“利他，用心”的价值观，致力为数字世界繁荣稳定贡献力量。

8 我们的哲学

分布式区块链系统对于过去已经发生的既成事实，进行公开的记录。这种公开的记录，对契约、秩序和法则是极大的推进和维护。而这些契约、秩序和法则，是人类文明的进化的成果，它们维护了社会的信用，使得交易能够在最低的成本下进行，是属于全世界人类的宝贵财富。区块链通过信息公开来维护社会信用，将契约，秩序和法则这些概念在网络空间进一步的发扬光大。

在大众认知中，公开既成事实代表了秩序、统一、规律，而隐私和变化代表了自由。有趣的是，区块链这种最公开最确定的技术，却成了过度的自由主义者的土壤。因为那些不公开不确定的信息没有存在的空间，反而更容易为所欲为。因此，对大量隐私的信息，以及不断变化的现存状态的保存，需要另外的技术和区块链技术相配合。只有公开信息和隐私信息，既成事实和现有状态互相影响，才能更好的描述整个世界。

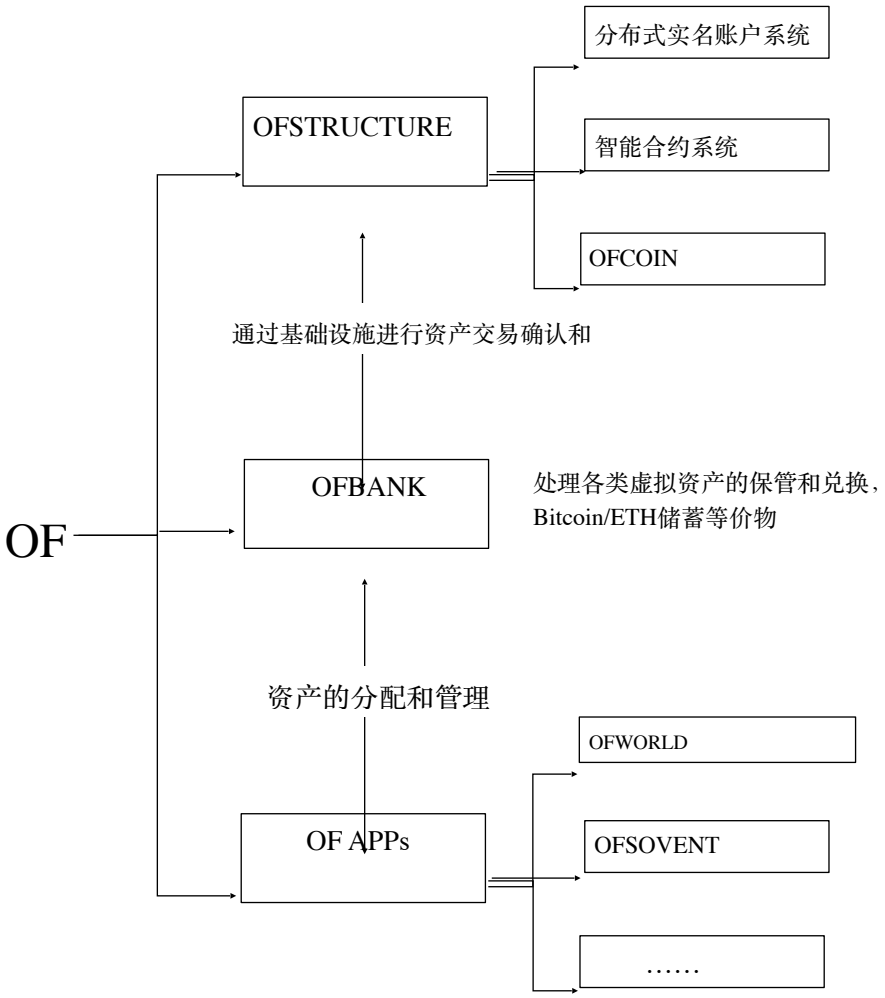
我们认为，自由需要一定的空间。我们希望构造出这个空间，让自由在这个空间跳舞而不逾界。 这也是区块链中智能合约的精神。

这便是OF的涵义。

OF同时也是一个介词，表示从属，也有“连接”的隐含意义，of是一个很“轻”很“小”的词语，是英文中使用量第二大的词，无处不在，我们寄希望“Order & Freedom”伴随着人类

文明的进程，无处不在。

我们就是用这样的哲学，设计了OF系统，它包括OF基础设施，OFBANK和OFAPPs，整体架构和相互关系如下图：



如果说账本的不断积累，挖矿带来的数字货币的不断产生，是一个积分的过程，那么OF系统所做的就是让积分只在一定区域内进行，即有限曲面积分。我们据此设计了OF的logo，一个有限曲面积分的符号。

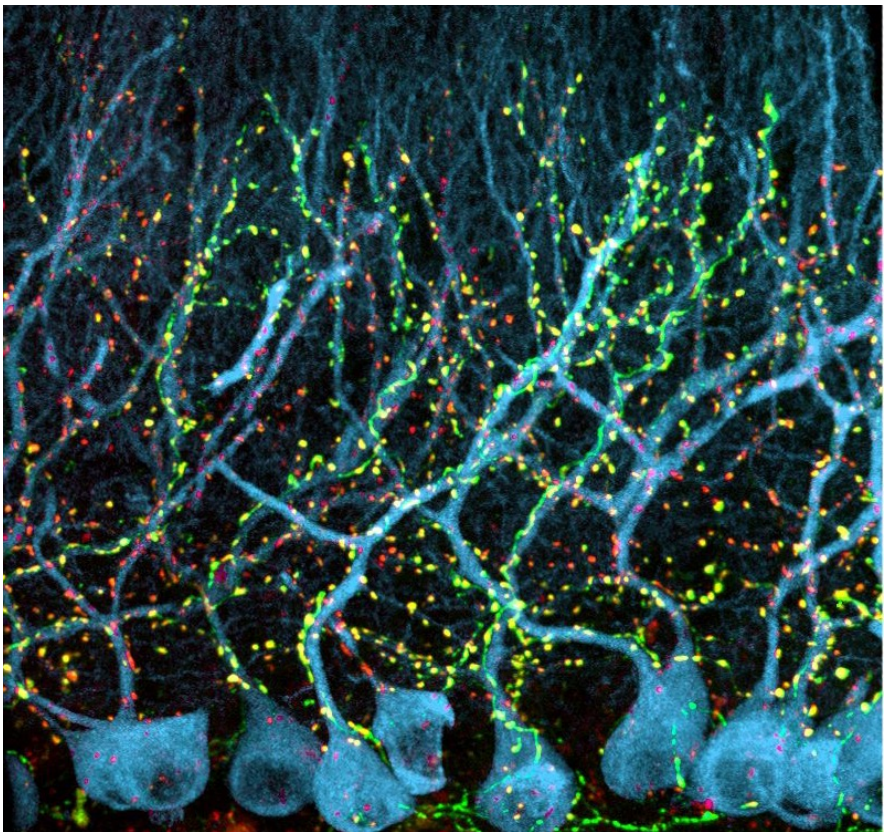
比特币的产生，相当于互联网有了自己的货币；以太坊的产生，相当于契约精神，贸易法和贸易协定的产生；而OFBANK的产生，相当于一个自生长的金融系统开始运转。

这个庞大的金融系统，是基于精巧的规则和数学法则。从中本聪的论文中可以看出，区块链的计算结构，和人脑的计算结构不谋而合的。不仅仅区块计算呈泊松分布和人脑惊人类似，并且多个诚实节点的冗余计算，也和人脑结构一致。多个输入的进行评判的方式，和每个接受多个树突的输入的计算模式也是一样[8][9][10][11][12]。

区块链技术中各个节点不断接收消息并记录的模式，和大脑的察觉能力相似。可信计算技术中，用另一个计算机观察记录和控制计算机运行的模式，和大脑元认知监控的能力也非常类似。

可以预见，区块链技术和可信计算技术的结合与不断发展，也是保障数字金融世界安全可靠的基石。这就是OFBANK设计的灵感来源。

人脑是一个承载百亿量级神经元的巨大的系统。这种数学上的类似表明，人脑强大而稳定的计算，也可以佐证新一代金融系统的产生和进化能力。



后记

我们认为区块链不仅仅是金融技术，也是下一代互联网应用的基础技术，它必然会在落地应用中发挥更多的价值。所以我们的目标在于将区块链技术普及到普通的用户手机中、电脑中。它的意义不亚于从DOS到Windows，从键盘到触屏。在研发的过程中，我们新增了分布式账户系统，并对账本系统进行了安全增强，通过引入监管方增加了中心化和去中心化的转换，在此过程中我们走过了无数的区块链之“Deep shit“，此处省略3500字。

后来我们发现，以应用落地为目标倒逼我们开发了这个系统，不仅可以更好的满足我们最初设想的需求，也可以解决目前区块链落地难的种种痛点。于是我们重新进行思考和梳理，对平台进行打磨和改进，让其更加系统和优美，并将之开放，服务于更多的开发团队。

不久的将来，我们将会完全开源整个源代码。

我们深信，这套可进化的开源代码和架构思想，将会吸引来自世界各地的优秀工程师，和我们一起迭代升级这个系统。在此基础上，围绕各种各样的应用，我们会共同维护一个基于共识的新一代的社交信用体系和交易系统，它会让虚拟和现实结合的更加紧密，让人类社会更加繁荣并充满乐趣。

参考文献

[1] 中国工业和信息化部. 中国区块链技术和应用发展白皮书（2016）

[2] United States Government Department of Defense. Trusted Computer System Evaluation Criteria; commonly called the "Orange Book"(1985)

[3] National Computer Security Center. The Rainbow Series.

[4] Trusted Computing Group: https://trustedcomputinggroup.org

[5] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System." In: Www.Bitcoin.Org (2008), p. 9. issn: 09254560. doi: 10.1007/s10838-008-9062-0. arXiv: 43543534534v343453. url: https://bitcoin.org/bitcoin.pdf.

[6] Vitalik Buterin. "Ethereum: A next-generation smart contract and decentralized application platform." In: URL https://github. com/ethereum/wiki/wiki/% 5BEnglish% 5D-White-Paper (2014).

[7] Hyperledger fabric： http://hyperledger-fabric.readthedocs.io/

[8] Shamir, Adi (1979), "How to share a secret", Communications of the ACM, 22 (11): 612 – 613, doi:10.1145/359168.359176

[9] Dayan, Peter, and Laurence F. Abbott. Theoretical neuroscience. Vol. 806. Cambridge, MA: MIT Press, 2001.

[10] Ma, W. J., Beck, J. M., Latham, P. E., & Pouget, A. (2006). Bayesian inference with probabilistic population codes. Nature neuroscience, 9(11), 1432-1438.

[11] Kass, R. E., & Ventura, V. (2001). A spike-train probability model. Neural computation, 13(8), 1713-1720.

[12] Bair, W., & Koch, C. (1996). Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey. Neural computation, 8(6), 1185-1202.

[13] David J. Wallin， Attachment in Psychotherapy. Guilford Press; 1 edition (26 April 2007). ISBN-10: 1593854560.