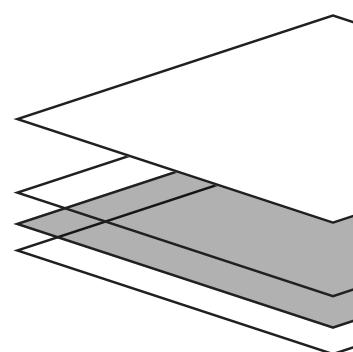
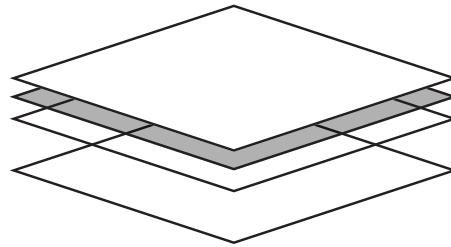
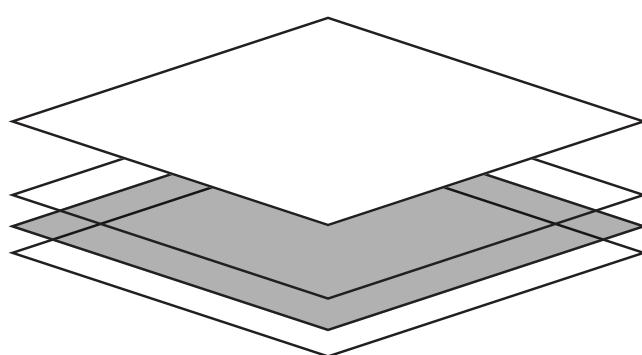
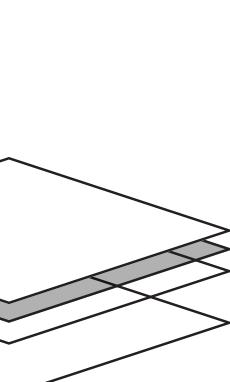




基于隐私计算的新一代联盟区块链平台



| | | |
|-----------|--------------------------------|-----------|
| 1. | 引言 | 1 |
| 1.1. | 区块链当前发展阶段 | 1 |
| 1.2. | 行业现状..... | 1 |
| 1.2.1. | 市场概况..... | 1 |
| 1.2.2. | 小结 | 1 |
| 1.3. | 区块链的企业级方案 | 2 |
| 1.3.1. | 区块链分类 | 2 |
| 1.3.2. | 企业选型..... | 2 |
| 1.4. | 联盟链实践中的挑战 | 2 |
| 1.4.1. | 合理共存的诉求：共享与隐私 | 2 |
| 1.4.2. | 企业应用规模发展痛点 | 3 |
| 2. | PlatONE的简单介绍及应对思路 | 5 |
| 3. | PlatONE技术解决方案..... | 6 |
| 3.1. | PlatONE计算 | 6 |
| 3.1.1. | 可验证计算 | 6 |
| 3.1.2. | 隐私计算..... | 7 |
| 3.1.3. | 国密支持..... | 11 |
| 3.1.4. | 并行计算..... | 13 |
| 3.2. | PlatONE中的电路 | 14 |
| 3.3. | 专用计算硬件 | 14 |
| 3.4. | 权限模型..... | 14 |
| 3.5. | 支持多语言的WASM虚拟机..... | 15 |
| 3.6. | 企业级合约管理..... | 16 |
| 3.6.1. | 一键合约数据迁移 | 16 |
| 3.6.2. | CNS（合约命名服务） | 17 |
| 3.7. | 高度优化的BFT共识算法..... | 17 |
| 3.8. | 形式化验证 | 18 |
| 3.9. | 企业级部署与运维工具集 | 18 |
| 4. | PlatONE技术架构 | 20 |
| 4.1. | PlatONE中的基本概念..... | 20 |
| 4.2. | 高度优化的BFT算法 | 20 |
| 4.2.1. | 概述 | 20 |
| 4.2.2. | PlatONE的共识算法详细介绍 | 21 |
| 4.3. | 权限模型..... | 24 |
| 4.4. | WASM创新与优化..... | 25 |
| 4.4.1. | WASM/EVM合约互调用 | 26 |
| 4.5. | 合约数据迁移协议 | 26 |
| 4.6. | CNS（合约命名服务）方案..... | 27 |
| 5. | 技术路线图..... | 29 |
| 6. | 应用场景 | 30 |
| 6.1. | 供应链金融 | 30 |

| | | |
|-----------|-------------------|-----------|
| 6.2. | 防伪溯源..... | 31 |
| 6.3. | 积分管理..... | 32 |
| 6.4. | 股权登记..... | 33 |
| 6.5. | 物流 | 35 |
| 6.6. | 慈善行业..... | 35 |
| 7. | 术语表 | 37 |
| 8. | 参考文献 | 39 |

1. 引言

1.1. 区块链当前发展阶段

自2009年比特币的创世区块诞生，到2014年区块链概念的提出，区块链技术经历了启蒙、认知、探索阶段，近些年已在金融、商业、组织协作与治理等各方面逐渐呈现出了其价值，受到各实体行业乃至国家战略层面的重视。区块链在2015世界经济论坛被列为未来六大趋势之一，也被Gartner列入企业组织在2019年需要探究的十大战略性技术。2016年12月15日，在国务院印发的《“十三五”国家信息化规划》中，强调了需加强区块链等新技术的创新、试验和应用，来抢占新一代信息技术主导权。

1.2. 行业现状

1.2.1. 市场概况

据IDC研究，2018年中国区块链的市场支出规模达到1.6亿美元，并预计这一增长态势将在未来三年延续。2022年全球区块链解决方案支出将达到117亿美元。在预测期内，区块链支出将以强劲的速度增长，2017-2022年复合年增长率（CAGR）为73.2%。

在市场规模上，预测期内美国仍是全球区块链投资最大的区域，占全球支出的比重为36%。分列二到五位的是西欧、中国、亚太（不含中国和日本）和加拿大。

区块链应用的行业分布十分集中，根据最新的全球区块链支出排名，排名前五的行业分别为：银行、离散制造、流程监控、零售和专业服务行业，其总支出占全部支出的比重高达58.4%。

2018年，全球市场支出均在1亿美元以上的场景主要为：跨境支付和结算、产品溯源、贸易金融及交易（后）管理、资产和货物管理、身份认证等。

从技术角度看，IT服务和商业服务是预测期内最大的支出类别，占比在70%以上。区块链平台软件将成为除服务类别之外最大的支出类别，也是整体增长最快的类别之一。

1.2.2. 小结

区块链已在多个实体行业中落地，且呈现较大的实际应用价值。从总体的趋势来看，区块链技术正逐渐引起更多行业的重视，大量基于区块链的应用场景有待发掘，市场潜力巨大。

1.3. 区块链的企业级方案

1.3.1. 区块链分类

按照许可和权限维度划分，当前区块链可分为公有链、联盟链和专有链。

公有链即无须许可的区块链系统，任何个体都可以自由进入和退出，也可以在其中写入、读取、参与交易，如比特币、以太坊等。

联盟链指由联盟成员维护的区块链，通常设计有线下认可的节点准入、用户管理和权限控制机制。联盟链可大幅降低异地结算成本和时间，比现有系统更简单，效率更高，同时继承去中心化优点减轻垄断压力。

专有链指各个节点的写入权限收归内部控制，而读取权限可视需求有选择性地对外开放。专有链具备区块链多节点运行的通用结构，适用于含多级机构的组织进行内部数据管理与审计。

1.3.2. 企业选型

联盟链因其成员具有组成可控、权限管理可控、同时由多个相关机构共同参与和管理的特点，成为了当前符合企业间协作需求的一种区块链选型。在近年来的企业区块链实践中，联盟链已应用到各垂直领域，如金融、供应链、物流、物联网、溯源等。相对于传统协作模式中明显的上下游企业属性划分，联盟链提供了一种对等多方协作范式，为减少机构间协作摩擦、提高业务水平提供了一种新的思路。

1.4. 联盟链实践中的挑战

1.4.1. 合理共存的诉求：共享与隐私

区块链的价值基础之一来自于其提供数据公开、共享、透明的机制，使得参与多方在开放互信的基础上开展业务合作。应用该特性，企业基于区块链尝试去解决边界摩擦、数据共享等传统非区块链解决方案的场景，使相关的企业业务水平及内部效率得到了充足提升。例如，基于透

明可信的特性，银行通过区块链来承载同业间跨境人民币清算业务，可实现安全、高效、快速的清算流程。

然而，随着企业实践的深入，越来越多与公开对立的隐私保护需求逐渐出现。根据IDC调研报告显示，数据安全是金融企业在应用区块链时最担忧的问题。区块链可理解为一种多方共识、共享的账本，方便各方对账本内的信息的直接检索、分析和应用。但这在很大程度上给平台各参与方的隐私带来了威胁，使得企业更愿意在一个较小的安全边界施展业务，从而局限了区块链的应用价值。

例如，在医疗场景中，由于医疗数据隐私问题，医疗保健中的预测分析可能难以利用，但如果预测分析服务提供商可以对加密数据进行操作，则这些隐私问题会减少；在业务流程管理的场景下，通常存在关联耦合的子流程，如税收、货币兑换、运输，数据暴露给每个参与方是很难在实践中被接受的。

目前大多数联盟链基础设施，不提供加密计算的框架和工具集，降低了企业的数据上链意愿。因此，在安全多方计算、隐私保护等方面加强安全防护，并在保证隐私的情况下实现区块链上数据的高效管理利用，是推动区块链大规模应用的有效路径。

1.4.2. 企业应用规模发展痛点

除上之外，经过多年在行业应用中探索，我们认为导致目前区块链应用规模进一步扩大的痛点有：

- 交易性能、扩展性不足。IDC调研还显示，有22.1%的用户认为区块链发展不成熟，存在技术瓶颈是限制其应用的最重要的三个因素之一。目前，行业内正在优化区块链共识算法，从而提高区块链系统性能这一方向上有不少努力，然而，在优化提高共识算法性能的同时论证算法的完备性、保障共识的可靠性仍在缺失。我们认为提供高效可靠的共识机制，满足大规模商业化交易需求，是区块链发展面临的重要挑战。
- 缺乏完备的、面向企业级业务的工具箱。区块链的实际落地离不开产品化，需要配套提供易用的管理组件，以及方便企业开展实施部署、运维和

业务人员使用的工具集。当前的联盟链产品更多地关注在底层技术框架，尚未有一套经由多个行业应用实践和反馈形成的高可用系统工具箱，这也提高了企业应用区块链的使用和管理成本，进而阻碍了区块链应用的规模化以及在领域内的深度拓展。

- 缺乏灵活的权限管理设计。传统软件经由多年实施应用，已逐渐形成稳定、成熟的权限模型，从通用的设计架构提供一种相对普适的权限系统。然而基于区块链的权限模型仍然是一个新的课题。需要在弱中心化、公平对等的理念下，弱化超级权限的设计、同时保留灵活可管控的权限分配、权限的生命周期管理，是保障基于区块链的业务流程安全、提高业务精细化管理的重要挑战。
- 缺乏自动化运维节点防控违法有害信息的机制。区块链系统节点分散，容易衍生诸如违法有害信息、且上链后无法有效处置等问题。如何在区块链中防控这些违法有害信息，如何为广大使用者及厂商提供真实唯一、合法便捷的信息服务，如何实现对节点查询、上链等相关信息进行自动、有效的拦截和处置，显得十分重要。
- 针对区块链应用开发的配套成熟度低。传统的机构和企业的技术栈储备以主流编程语言为主，如C++、Java等。而目前主流的联盟链应用开发语言，或者为从以太坊继承的Solidity，或者仅提供单一的开发语言，缺乏扩展性和多样性，对开发技术栈提出了较高的迁移学习成本，阻碍了应用生态的繁荣，也弱化了基础平台承载的价值。

因此，我们希望通过从解决实际业务实施中痛点的角度出发，设计和优化联盟链技术架构和治理工具水平，并将其沉淀和内化到一个通用链基础设施，来为大规模生产级的区块链应用提供支撑。

2. PlatONE的简单介绍及应对思路

PlatONE是基于隐私计算的新一代联盟区块链平台，以支持企业级应用。平台提出了一种以隐私计算为特色的企业级联盟链基础设施，可满足金融商业等多种需求场景。

目前PlatONE提供了多种创新性技术和功能，包括：安全多方计算、同态加密等密码学技术植入、优化的高效共识、高TPS、完备、易用的企业级工具链和组件、优化的用户/权限模型、多开发语言支持等特性，旨在解决当前联盟链发展中存在的困境。

3. PlatONE技术解决方案

3.1. PlatONE计算

3.1.1. 可验证计算

可验证计算 (Verifiable Computation, 简称VC) 方案允许计算资源有限的客户端将函数 F 的计算外包给一个或者多个服务端。其中函数 F 的输入 x_1, \dots, x_k 是客户端动态选择的。然后服务端会将该函数的计算结果 $y_i = F(x_i)$ 和一个证明 π_i 返回给客户端。简而言之，可验证计算是一个两方的协议，在该协议中客户端选择一个函数，并将该函数和相关的输入发送给服务端。服务端根据收到的函数和函数的输入计算出输出结果，并将该输出结果返回给客户端。客户端根据收到的计算结果验证该计算结果的合理性。通常计算过程的复杂度要高于验证过程的复杂度。

可验证计算方案 $\mathcal{VC} = (KeyGen, ProbGen, Compute, Verify)$ **主要包含如下4个算法：**

- $KeyGen(F, \lambda) \rightarrow (PK, SK)$: 输入为函数 F 和安全参数 λ , 随机密钥生成算法生成一对密钥(PK, SK), 其中公钥 PK 为公开计算密钥, 服务端会使用该密钥来计算函数 F , 私钥 SK 由客户端秘密保存;
- $ProbGen_{SK}(x) \rightarrow (\sigma_x, \tau_x)$: 输入为值 x , 客户端使用私钥 SK 将函数 F 的输入 x 编码成公开值 σ_x , 并将该 σ_x 发送给服务端用于函数 F 的计算, τ_x 由客户端秘密保存;
- $Compute_{PK}(\sigma_x) \rightarrow \sigma_y$: 使用客户端的公钥, 编码后的输入 σ_x , 服务端计算编码后的函数 F 的输出值 $y = F(x)$;
- $Verify_{SK}(\tau_x, \sigma_y) \rightarrow y \cup \perp$: 使用私钥 SK 和 τ_x , 验证算法将服务端的编码输出转换成函数的输出, 如 $y = F(x)$ 或者直接输出终止符 \perp , 表示 σ_y 不是函数 F 对 x 的一个合理输出。

可验证计算需满足以下性质：

- **正确性**: $ProbGen$ 算法产生的值被诚实的服务端用于计算, 并且服务端产生的对应的输出值能够成功通过验证;
- **安全性**: 恶意的服务端都无法使验证算法 $Verify$ 接受一个不正确的输出;
- **有效性**: 验证算法 $Verify$ 的复杂度或运行时间需要比直接计算 F 开销要小很多。

公开可验证计算与传统的可验证计算方案的区别在于，它要求服务端在返回计算的函数值的同时要给出计算输出值的正确性的证明。一个公开可验证计算方案 $\mathcal{VC} = (KeyGen, Compute, Verify)$ 主要包含如下3个算法：

- $KeyGen(F, \lambda) \rightarrow (EK, VK)$ ：输入为函数 F 和安全参数 λ ，输出为一对密钥 (EK, VK) ，其中 EK 为公开计算密钥， VK 为公开验证密钥；
- $Compute(EK, x) \rightarrow (y, \pi_y)$ ：输入为计算密钥 EK 和 x ，输出为计算值 y （预期为 $F(x)$ ）和一个关于 y 的计算正确性证明 π_y ；
- $Verify(VK, x, y, \pi_y) \rightarrow 1/0$ ：输入为验证密钥 VK ，输入 x ，函数 F 的计算结果 y 和证明 π_y ，如果 $y = F(x)$ ，则输出 1；否则输出 0。

在PlatONE中，我们的公开可验证计算方案还具备如下特性：

- **不可伪造性**：对于任意恶意的服务端，如果 $y \neq F(x)$ ，那么产生可接受的证明 π 在计算上是不可行的；
- **计算有效性**：针对一些函数，生成证明是相对有效且低成本的；
- 不涉及可信第三方初始化过程。

从可验证计算方案特性我们知道，验证算法的开销比计算算法的开销要小，因此在PlatONE中我们使用公开可验证计算方案能显著降低计算节点的开销，并且高开销的计算可以转移到链下进行处理，能进一步提高可扩展性且降低链上计算开销。

3.1.2. 隐私计算

PlatONE通过安全多方计算和同态加密算法实现真正的隐私计算，实现对计算代码和数据的隐私保护。与其他基于TEE/SGX的方案不同，PlatONE全流程保证安全，不存在任何安全边界。

安全多方计算

安全多方计算（Multi-party Computation）主要用于解决在多方协同计算任务中用户数据的隐私保护问题。在传统密码学方案中，如对称加密、非对称加密等，方案针对系统合法用户外的恶意攻击者，提供了数据在传输或存储过程中的机密性、完整性的保护。而在安全多方计算的方案中，要求保护每个合法参与者各自的数据隐私。

一般来说，在安全多方计算的场景中，假设有 N 个参与者，每个参与者拥有自己的私密数据(d_1, \dots, d_N)。他们使用各自的私密数据作为输入，合作完成一个计算任务，记为 $F(d_1, \dots, d_N)$ ，使得每个参与者都可以得到计算任务的输出，同时参与者无法得知其他参与者的私密数据。

由此看出，安全多方计算方案需要满足：

- **输入隐私保护（Input privacy）**：在协同计算的交互过程中，用户无法获得除计算输出结果外，任何其他参与者的私密数据信息。
- **正确性（Correctness）**：假设存在若干恶意用户（小于方案的安全阈值），诚实用户在执行协议后，依然可以得到正确的计算结果。

安全多方计算可以被广泛应用在电子选举、门限签名以及电子拍卖等应用中。安全两方计算是安全多方计算的基础和特例，有着重要的理论价值和广泛的应用价值，目前，安全两方计算协议的主流设计框架仍基于姚期智先生最早提出的基于加密电路的两方计算通用协议，并由Beaver、Micali和Rogaway进一步扩展到多方计算。

Yao基本协议的核心技术是加密电路（Garbled Circuit，简称GC）和不经意传输（Oblivious Transfer，简称OT）。

- **加密电路：**将需要计算的函数 f 表示为布尔电路，其基本单元为逻辑门，门的输入线路可以是函数 f 的输入变量，也可以是其它门电路的输出线路。发送方以电路作为输入，记录每个门的真值表，并选择两个随机字符串作为加密密钥对门加密，对电路中的每条线路都选取一对加密密钥，所构成的电路称为加密电路。
- **不经意传输：**是一种可以保护隐私的双方通信协议。假设发送方提供多个字符串，接收方提供一个索引，选取其中的一个或多个字符串，协议完成后，接收方获得其索引所对应的字符串，对其他字符串一无所知，发送方没有输出，且不知道接收方获得了哪些字符串。

同态加密

同态加密是一种允许在密文上进行计算的加密方式。除了传统加密方案的原始组件之外，还有另一种计算算法，它将目标函数和加密数据作为输入。同态加密会生成一个加密的结果，当解密此结果时，获得的消息就像是在加密数据的明文上执行函数。同态加密的目的是允许对加密数据进行计算，通常用于安全外包计算，如云计算服务等。

同态加密主要包含如下几个算法：

- $KeyGen(\lambda)$: 密钥生成算法，输入为安全参数 λ ，输出为一对密钥(pk, sk)，其中 pk 为公钥， sk 为私钥；
- $Enc(pk, m)$: 加密算法，输入为公钥 pk 和消息 m ，输出为密文 c ；
- $Dec(sk, c)$: 解密算法，输入为私钥 sk 和密文 c ，输出为消息 m ；
- $Eval(pk, F, c_1, \dots, c_n)$: 输入为公钥 pk ，函数 F 以及密文 c_1, \dots, c_n ，输出为一个密文 c_{F^o}

同态加密算法除了满足传统公钥加密方案的性质外，它还具有同态属性，即：

- 如果 $c_1 = Enc(pk, m_1), \dots, c_n = Enc(pk, m_n); m_F = F(m_1, m_2, \dots, m_n)$
- 并且 $c_F = Eval(pk, F, c_1, \dots, c_n)$ ，则有 $m_F = Dec(sk, c_F)$ 。

同态加密主要有部分同态和全同态。对一个密码系统而言，部分同态只能实现对密文的部分运算，如加法同态和乘法同态。而全同态能够对密文支持任意的计算，其功能非常强大，目前共有3代全同态。

部分同态主要有加法同态和乘法同态：

- **加法同态**：在不需要知道 x, y 值的条件下，从 $Enc(x)$ 和 $Enc(y)$ 通过运算计算出 $Enc(x + y)$ ，满足该属性的方案有Paillier、Benaloh方案。
- **乘法同态**：在不需要知道 x, y 值的条件下，从 $Enc(x)$ 和 $Enc(y)$ 通过运算计算出 $Enc(x * y)$ ，满足该属性的方案有RSA加密方案、ElGamal加密。

全同态 (Fully homomorphic encryption, 简称FHE) 是一种加密密码系统，它对密文支持任意的计算，而无需解密。在云计算和分布式计算中，这是一种非常强大的算法。在2009年由Craig Gentry在《Fully Homomorphic Encryption Using Ideal Lattices》这篇文章中提出的，已经有非常多的全同态加密方案和实现，如HElib（IBM开发）、SEAL（微软开发）等。

在PlatONE中我们使用加法同态加密系统来实现相关功能，实现个人数据以及交易数据的完全保密，且可以在无需公开发送者、接收者以及交

易数量的条件下验证交易的合理性。PlatONE中主要采用的是Paillier加密算法，该加密算法具有加法同态属性，具体方案包含如下几个算法：

- 密钥生成算法 $KeyGen$ ：
 - 独立随机生成两个大的素数 p, q ，且两个素数长度相同；
 - 计算 $n = pq$, $\lambda = lcm(p - 1, q - 1)$;
 - 随机选择整数 g , 满足 $g \in \mathbb{Z}_n^*$, 且 g 的阶为 n 的倍数；
 - 设置公钥 $PK = (n, g)$, 私钥 $SK = \lambda$ 。
- 加密算法 $Enc(Pk, m)$ ：待加密的消息 $m \in \mathbb{Z}_n$ ，随机选择 $r \in \mathbb{Z}_n^*$ ，计算密文 $c = g^m \cdot r^n \bmod n^2$ 。
- 解密算法 $Dec(SK, c)$ ：待解密密文 $c \in \mathbb{Z}_{n^2}$, 令 L 函数为 $L(x) = \frac{x - 1}{n}$, 其中 $x \in \mathbb{Z}_{n^2}$
且 $x \equiv 1 \bmod n$, 明文 $m = \frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n$ 。

该加密方案满足加法同态属性，即：对任意的给定的 $c_1 = Enc(PK, m_1), c_2 = Enc(PK, m_2), m_F = m_1 + m_2$, 定义 $c_F = c_1 \cdot c_2$, 则有 $m_F = Dec(SK, c_F)$ 。

零知识证明

在密码学上一个典型的问题就是如何向互不信任的各方提供秘密信息片段。其中各方都拥有相关的秘密设置，他们希望揭示这些秘密的一部分。秘密通常由一些公开信息来决定，因此探讨如何正确地揭示这些秘密是有意义的。关键的问题在于能够有效地验证揭示的秘密信息而不泄露其它的秘密信息。

一般来说，问题是证明某个断言的合理性而不泄露其它任何秘密信息。如果这种证明存在，我们称为零知识，该种证明系统在构造密码学协议中起着核心作用。简而言之，零知识证明就是证明某个断言的合理性，与此同时不泄露任何其它的秘密信息。零知识证明是由S.Goldwasser、S.Micali及C.Rackoff在20世纪80年代初提出的，实现了证明者能够在不向验证者提供任何秘密信息（witness）的情况下，使验证者相信某个断言是正确的。

一个证明系统通常有两个重要的属性：

- **可靠性 Soundness:** 它体现的是验证者的能力，即如果该断言是错误的，验证者能够以绝对的优势拒绝该证明。
- **完备性 Completeness:** 它体现的是证明者的能力，即如果证明该断言是正确的，证明者将以绝对的优势使得验证者接受该证明。

这两个属性对于证明系统来说至关重要。

证明系统分为交互式证明系统和非交互式证明系统。其中交互式证明系统是指与证明系统相关的两个计算任务即“生成”证明并“验证”证明的有效性。这些任务由两个不同的参与方执行，称为证明者和验证者，它们彼此交互。一般来说，交互可能更为复杂，并且可能采取验证者询问证明者的形式。两方的交互以一种比较自然的方式来定义，唯一值得注意的一点是，交互双方有个公共参考串作为输入参数。而非交互式证明系统是指证明者将所有的计算任务完成之后，将证明一次性发给验证者。中间不存在验证者询问的过程。

零知识证明系统不仅满足证明系统的两个属性，还满足零知识性，即证明不会泄露任何秘密信息。目前零知识证明已经被广泛应用在如ZCash等区块链项目中，用以实现保护交易数据的隐私性。在很多商业应用场景中，交易数据往往是公司的核心商业机密，而在区块链系统中，为了保证分布式一致性，需要满足数据的公开可验证，因此在设计策略中，应该同时兼顾数据的可用性和隐私性。

为了实现隐私计算，PlatONE引入了zk-SNARK这项技术使得用户可以在智能合约平台中完成数据的隐私计算。zk-SNARK，简短零知识证明，该技术可用于证明计算过程的正确性，即给定输入 x ，输出 y ，它可用来证明 $y = F(x)$ ，其中函数 F 可以含括非常丰富的类型，如哈希函数、椭圆曲线运算等等。证明者 P 根据输入 x ，输出 y ，以及函数 F ，运行证明算法可以得到一个非常简短的证明（大约286字节），而验证者 V 只需要花费10毫秒左右的时间就可以验证该证明。由于SNARK具有证明非常简短的特点，也就意味着，我们可以将许多计算在链下完成，而将计算结果记录在链上，这样既可以减少智能合约的泄露，同时也可以减少链上计算。

3.1.3. 国密支持

国密算法即国家密码局认定的国产密码算法，即商用密码。商用密码，是指实现商用密码算法的加密、解密和认证等功能的技术。商用密码的应用领域十分广泛，主要用于对敏感的内部信息、行政事务信息、经济信息等进行加密保护。比如：商用密码可用于企业门禁管理、企业内部的各类敏感信息的传输加密、存储加密，防止非法第三方获取信息内容；也可用于各种安全认证、网上银行、数字签名等。

SM2

国密标准是我国自主设计的商用密码标准，实现了对称加密、非对称加密、消息摘要等密码算法功能，用以改变在以银行业等核心金融领域长期依赖3-DES、SHA-1、SA等国际通用密码算法体系及相关标准的现状，从而达到摆脱对国外密码技术的过度依赖，实现密码算法自主可控的目的。国密标准主要包括SM-1、SM-2、SM-3、SM-4、SM-9等算法。

SM2是基于椭圆曲线的公钥密码算法，在商用密码体系中，用于替换RSA算法，可用来实现非对称加密、数字签名、密钥交换等功能。作为一种基于椭圆曲线的公钥密码算法，SM2算法在相同安全程度下，较RSA算法在密钥规模上存在优势。

① 系统参数：

- F_q : 包含 q 个元素的素域
- 椭圆曲线方程的参数 a, b
- G : 椭圆曲线的基点 $G = (x_G, y_G)$
- n : 基点 G 的阶，由基点 G 生成的群的元素的个数
- 可选项: n 的余因子 $h = \left| E(F_q) \right| / n$
- 除了定义了椭圆曲线的参数和签名算法所需要的辅助函数（哈希函数和随机数发生器）外，标准引入了用户信息标识的概念。签名者拥有长度为 $entlen_A$ 比特的可辨别标识 ID_A ，在标准规定的签名算法中，需要在生成签名和验证签名前计算用户A的哈希值 Z_A 。 $Z_A = H_{256}(ENTL_A || ID_A || a || b || x_G || y_G || x_A || y_A)$ ，其中 $ENTL_A$ 是由 $entlen_A$ 转换而成的两个字节。

② 签名生成：

- 预处理：计算 $Z_A = H_{256}(ENTL_A || ID_A || a || b || x_G || y_G || x_A || y_A)$
- 生成签名：输入为 M ， Z_A ，私钥 d_A ，输出为签名 (r, s)
 - 消息处理： $\bar{M} = Z_A || M$ 并计算 $e = H_v(\bar{M})$
 - 产生随机数 $k \in [1, n - 1]$
 - 计算曲线点 $(x_1, y_1) = kG$

- 计算 $r = (e + x_1) \bmod n$, 若 $r = 0$ 或 $r + k = n$ 则返回重新选择随机数 k 这一步
 - 计算 $s = ((1 + d_A)^{-1}(k - r * d_A)) \bmod n$, 若 $s=0$ 则返回重新选择随机数这一步
 - 输出签名 (r, s)

③ 签名验证:

- 预处理: 计算 $Z_A = H_{256}(ENTL_A || ID_A || a || b || x_G || y_G || x_A || y_A)$
- 验证过程: 输入为消息 M' , 签名 (r', s') , 公钥 P_A , 输出为1或者0
 - 检验 $r' \in [1, n - 1]$ 是否成立, 若不成立则验证不通过;
 - 检验 $s' \in [1, n - 1]$ 是否成立, 若不成立则验证不通过;
 - 消息处理: $\bar{M}' = Z_A || M$ 并计算 $e' = H_v(\bar{M}')$
 - 计算 $t = (r' + s') \bmod n$
 - 计算椭圆曲线点 $(x'_1, y'_1) = s'G + tP_A$
 - 计算 $R = (e' + x'_1) \bmod n$, 检验 $R = r'$ 是否成立, 若成立则验证通过; 否则验证不通过

为了更好地满足在金融等领域的应用需求, PlatONE引入了对国密算法SM2的支持, 用户可以在智能合约中完成SM2签名的验证功能。

SM9

SM9是中华人民共和国政府采用的一种标识密码标准, 由国家密码管理局于2016年3月28日发布, 相关标准为“GM/T 0044-2016 SM9标识密码算法”。在商用密码体系中, SM9主要用于用户的身份认证。SM9主要包括三部分: 签名算法、密钥交换算法、公钥加密算法。SM9 算法是一种基于身份的密码系统, 用户的私钥是由密钥生成中心根据用户的ID为用户生成的, 而用户的公钥就是用户的公开ID。以SM9算法替代数字证书, 可以大大降低电子邮件加密等应用场景中的密钥管理开销。SM9在互联网、云计算、大数据等相关领域的数据安全方面展现了得天独厚的优势, 该算法可应用于金融支付、税务票据、移动终端安全管理等基础领域。

3.1.4. 并行计算

PlatONE中, 智能合约被编译成布尔电路 (Boolean Circuit), 布尔电路是由各种不同的门 (Gate) 构成的“复杂有向无环图”, 可分解为细粒度的计算任务, 并通过PlatONE网络将计算任务分发到多个计算节点并

行计算。为保证计算的可靠性，避免因节点掉线或超时导致计算失败，同一个子任务会同时分发给多个计算节点，保留一定的计算冗余度。

3.2. PlatONE中的电路

电路是由各种不同的门（Gate）通过输入输出线构成的“复杂有向无环网络”。由逻辑门（比如：与、或、非、异或等）构成的电路称为布尔电路（Boolean Circuit）；由算术门（比如加法、乘法等）构成的电路称为算术电路（Arithmetic Circuit）。任意形式的计算都可由电路表示，电路以有限种类的门构成各类复杂的计算形态。电路因为其基本组成部分的简易性，是在密码学中被广泛使用的计算模型。PlatONE通过电路来水平地连接各类算法和硬件。电路作为安全多方计算、零知识证明、可验证计算、全同态加密共同使用的通用计算模型，以其超强的普适性串联各类算法。电路表示的算法也天然适合专用硬件的实现。电路是PlatONE度量“计算”的基础。构成电路的基本单位为门，不同种类门的资源消耗不同，整个计算的度量可表示为电路中所有门的消耗的度量总和。电路为计算的度量和定价提供了理论基础。

3.3. 专用计算硬件

PlatONE中，智能合约的计算逻辑被编译成布尔电路进行计算，整个计算回归到与、非、异或等处理。而布尔电路的操作，天然与FPGA的架构相匹配，通过将智能合约转换成FPGA的布尔电路并通过FPGA来执行这些逻辑单元，能够极大地提高运算效率和降低功耗/成本。PlatONE将在适当的阶段推出基于FPGA/ASIC的专用计算硬件，会极大提升整个区块链平台的交易性能，真正实践下一代计算架构当中的硬件部分。

3.4. 权限模型

现实场景中的商业模型往往是比较复杂的，它包含大量的商业元素及它们之间的关系，并且用来描述商业行为中的不同方面，如操作流程，组织结构及金融预测等。因此为了更好的满足不同企业级用户的需求，且保障节点间通信安全性，以及对节点数据访问的安全性，PlatONE采用系统合约的方式实现了一整套完善的权限模型，包括节点准入机制，用户角色管理，合约防火墙等功能，充分满足不同商业需求，为产业赋能，且在网络和存储层面上做了严格的安全控制，提升系统安全性。其中：

- 节点准入机制：PlatONE通过节点管理合约对节点进行管理，包括节点是否能够接入网络，节点是否能够参与共识以及节点信息的维护等功能。
- 用户角色管理：PlatONE根据不同的权限，设定了不同的用户角色，并通过系统合约的方式对用户的角色进行管理。根据不同的角色，用户在系统中被赋予不同的权限。
- 合约防火墙：PlatONE中合约的调用权限由合约防火墙控制，只有合约的创建者才可以设置该合约的防火墙。

3.5. 支持多语言的WASM虚拟机

WASM是一种基于堆栈式虚拟机的二进制指令格式。被设计为可以使用高级语言（例如：C/C++/Rust）直接编译成WASM中间字节码。其已被Google, Facebook, Microsoft等世界顶级互联网公司同时支持，同时也可以在所有流行的浏览器中运行。其一开始设计的目的是用于解决Web程序日益严峻的性能问题，因其有以下优越的特性，被越来越多的非Web项目所采用。

- **快速、高效、可移植：**通过利用常见的硬件能力，WASM代码在不同平台上能够以接近本地速度运行。
- **可读、可调试：**WASM是一门低阶语言，但是它有一种人类可读的文本格式，这允许通过人工来写代码，看代码以及调试代码。
- **保持安全：**WASM被限制运行在一个安全的沙箱执行环境中。像其他网络代码一样，它遵循浏览器的同源策略和授权策略。
- **不破坏网络：**WASM的设计原则是与其他网络技术和谐共处并保持向后兼容。

PlatONE支持WASM虚拟机意味着开发智能合约不再局限于Solidity一门语言，同时可以使用多种高级语言，例如：C/C++/Rust等来进行编写智能合约，最后编译成WASM字节码就可以在PlatONE上运行，极大的降低了入门门槛和开发成本，同时也提高了智能合约的安全性。

PlatONE同时支持WASM虚拟机与EVM虚拟机，且允许WASM合约与EVM合约之间的互调用，充分降低开发者学习成本，其中WASM合约支持多种高级语言开发，编译成.WASM格式文件执行。触发WASM合约的交易由共识节点打包，全网节点重复执行验证。WASM合约的状态保存在公共账本中。

可验证合约的开发和发布跟WASM合约没有区别，最终也是编译成.WASM格式文件执行。可验证合约的状态转换在链下由计算节点异步

执行，计算完成后新的状态和状态转换证明提交到链上，全网节点可快速验证正确性并将新的状态更新到公共账本中。可验证合约可支持复杂、繁重的计算逻辑而不影响整条链的性能。

隐私合约同样支持高级语言开发，编译成 llvm ir 中间语言执行。隐私合约的输入数据保存在数据节点本地，由数据节点在链下以安全多方计算方式进行隐私计算，并提交计算结果到链上。

3.6. 企业级合约管理

通过系统合约的方式，实现了系统参数动态调整CNS等个性化定制服务：

- 系统配置参数统一通过合约进行管理，支持技术升级和治理；
- 节点准入管理采用上传公钥模式，避免传统CA证书过期问题和证书传递过程中的泄漏风险；
- 支持优化的合约权限控制、角色支持和管理；
- 支持CNS (Contract Name Service) 服务，发送交易不再通过传统的十六进制格式的合约地址调用，而改为通过合约名称调用，减少合约升级带来的数据兼容问题；
- 支持区块链运维态势感知、威胁警告。动态监控区块链和智能合约的运行状态，及时汇报链上安全状况信息。

3.6.1. 一键合约数据迁移

在智能合约升级的场景中，常伴随在新旧合约间进行历史数据迁移的需求。我们调研评估了若干种实施合约数据迁移的方案，包括：

- 硬编码迁移：指在新版本的数据合约中保存一个指向旧版本数据合约的合约地址，从而使新版本数据合约保存增量的数据内容；
- 硬拷贝迁移：指利用外部迁移工具，将旧版本数据逐步拷贝到链下，再从链下重新存储到新版本合约；
- 基于链存储机制迁移：指利用合约账户链上数据存储机制，在底层进行数据对象克隆和再构造的过程。

PlatONE根据其底层存储机制，实现了一套基于默克尔树的合约数据迁移协议，支持一键式地将旧合约数据迁移到新部署合约。该协议支持特性如下：

- 迁移效率高、用户迁移成本小；
- 避免导入导出导致的迁移错误；
- 不会对原有的合约逻辑及数据造成入侵。

3.6.2. CNS（合约命名服务）

在目前主流的区块链中，用户是通过地址来访问智能合约的，比如以太坊。智能合约的地址是一段十六进制字符串，用户需要记住这段冗长的字符串才能访问链上的智能合约。当合约需要升级时，重新部署合约又会产生新的地址，所有依赖于该合约的模块都需要做相应更新。显然现有访问合约的方式对用户是不友好的，因此我们在PlatONE中实现了合约命名服务，用户可以通过合约名称及版本号来访问智能合约。

合约命名服务英文全称为Contract Name Service，简称CNS。合约命名服务维护了名称、版本到合约地址的映射关系，提供了对系统中合约的管理功能，包括合约的注册和注销，合约注册信息和地址的查询等功能。

PlatONE使用系统合约实现了合约命名服务，用户部署合约后可以将该合约注册到系统合约中，后续调用可以通过合约名称及版本进行调用，而无需使用合约地址。如果交易是根据合约名称、版本来调用合约，PlatONE底层自动在系统合约中查询名称版本对应的合约地址，然后调用该地址的合约。

3.7. 高度优化的 BFT 共识算法

BFT (Byzantine Fault Tolerant) 类共识算法是一种即使系统中存在恶意节点（也就是拜占庭节点）也能保证分布式系统的safety和liveness的共识算法。

Lamport于1982年提交的经典论文《The Byzantine Generals Problem》中首先对此类共识问题进行了研究，并提出了口头协议和书面协议两种BFT类共识算法，不过这两个算法的消息复杂度都很高($O(n^m)$)，因此不是很实用。之后，在提高算法的实用性方面人们做了很多工作。其中PBFT算法就是第一个较为实用的BFT类算法，其共识正常流程的消息复杂度降低为($O(n^2)$)。不过，PBFT中的view change流程的消息复杂度仍然很高，为($O(n^3)$)，并且流程较为复杂。因此将PBFT算法应用于区块链中仍然存在优化和改进的空间。

Tendermint共识算法在某种程度上类似于PBFT，主体也是三阶段协议，并且也有round change流程（类似于PBFT的view change流程），不过Tendermint结合区块链的技术特性创新性地将round change流程并入了正常的共识流程，从而将round change的消息复杂度降低为 $O(n^2)$ 。

PlatONE的共识算法同样为BFT类共识算法，其继承了PBFT和Tendermint的三阶段协议设计特点，同时也吸收了Tendermint的算法的优点，同样将round change流程并入了正常的共识流程，从而将round change的消息复杂度降低为 $O(n^2)$ 。同时优化了锁定和解锁的机制，并且在很多地方作了优化和改进。

作为高度优化的BFT共识算法，PlatONE的共识支持超过100个共识节点，极大地提高了系统的去中心化程度，同时也极大地提高了系统的共识效率，使系统拥有很高的TPS。

3.8. 形式化验证

PlatONE中引入形式化验证工具及安全技术验证，可以为智能合约提供安全审计功能，发现智能合约中安全相关的漏洞缺陷，如资金数值溢出，数组越界等，同时也可以发现智能合约中功能逻辑的漏洞，避免合约代码实现与设计不一致所导致的风险。

3.9. 企业级部署与运维工具集

目前市面上大部分联盟链的部署和运维工具集都多多少少涉及到工具的安装以及额外的环境依赖，所提供的部署工具集，也常常需要执行大量的部署步骤，花费较多的时间和学习成本。

PlatONE系统提供了丰富的企业级部署工具，极大的降低了复杂度；以及提供了完善的说明文档和相关的PlatONE运维建议，贯穿着部署与运维的各个方面。以灵活、易用和极低的学习成本为出发点，以最大化提高部署与运维友好度为方向，从而实现更快捷的部署、开发和维护。工具集在如下方面进行了大量优化：

- PlatONE部署工具集不需要安装和额外的环境依赖；

- 工具集可以在不同的架构调整中，灵活的切换；
- 提供了多种命令以及参数，满足用户不同的部署需求；
- 提供了极为完善的说明文档，以及运维维护经验建议，降低试错成本和学习成本；
- 支持一键启动单个或多个节点的联盟链；
- 提供多种自定义的部署以及维护方式，降低了部署与维护成本；
- 系统提供了丰富的运维脚本，极大的降低了联盟链运维难度。

4. PlatONE技术架构

4.1. PlatONE中的基本概念

① 节点

PlatONE中的节点主要有以下几类：

- 观察者节点：只负责同步区块，不参与出块，系统中将会一直存在几个稳定的观察者节点。用于稳定同步区块，同时也用于被其他的节点指定为bootnodes进行连接；
- 共识节点：参与出块，以及同步区块。

② 账户

相比账户模型，UTXO不支持智能合约，而很多的DAG项目也在积极探索智能合约，但是还没有成熟稳定的解决方案，因此PlatONE选择成熟稳定支持智能合约的账户模型。PlatONE中，每个账户都有一个与之关联的状态（state）和一个20字节的地址（address）。账户分为两类：

- 普通账户：该类型账户由私钥控制，用户可通过钱包客户端或命令行生成。在PlatONE中，普通账户可以创建交易，并使用私钥对交易签名。普通账户除保存账户余额外，智能合约可被授权扩展和访问自定义属性。
- 合约账户：该类型账户没有私钥，由代码控制，合约账户地址在部署合约时产生。与普通账户不同，合约账户不能自行发起新的交易。每当合约账户收到一条消息，合约内部的代码就会被激活，允许它对内部存储进行读取和写入，以及发送其它消息或者创建合约。

4.2. 高度优化的BFT算法

4.2.1. 概述

PlatONE的共识为高度优化的BFT类共识算法，其容错率为 $1/3$ ，在保留即时确认（instant finality）的关键特性的同时，极大地提高了去中心化的程度。共识可以保证上链的区块是确定的，也就是说链不会出现分叉，同时每一个有效的区块都会插入到链上。

PlatONE的共识支持超过100个共识节点。相对于其他一些常见的BFT共识，PlatONE的共识的性能有显著的提升。在10个共识节点的情况下，TPS接近1000。

PlatONE的共识运行的相关参数可以灵活地进行配置，并且PlatONE的共识中的共识节点集合可以灵活地进行更新。近期计划支持共识的插件化，以及共识的可审计性等。

PlatONE共识是在round上进行的。在特定的round上，通过预先设置的策略选取一个出块者节点。出块者节点的选取策略目前支持两种：round robin和sticky proposer。

出块者节点提议区块后，各共识节点进行共识。共识分三阶段，其中后两个阶段为投票阶段，用以保证Safety。PlatONE共识使用round change机制结合锁定和解锁机制来保证共识的Liveness。通过优化解锁机制，解决了业界多个知名项目中存在的共识死锁问题。

PlatONE共识会为每一个链上的区块生成共识证明，也就是对于该区块的各共识节点的有效签名，因而区块可以进行自验证，同时也能支持轻节点。

4.2.2. PlatONE的共识算法详细介绍

共识节点选取机制 - PlatONE中通过系统合约NodeRegister（节点申请）和 NodeManager（节点管理）进行节点的申请和管理。因此可以通过与这两个系统合约进行交互来管理共识节点集合。

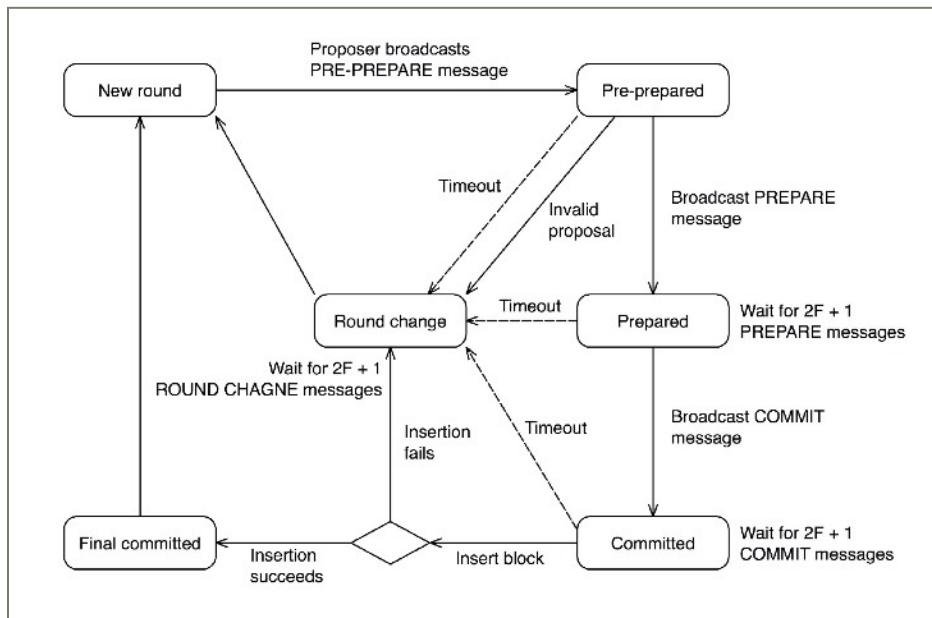
共识流程 - 正常流程

PlatONE的共识为三阶段协议，这三阶段分别为PRE-PREPARE，PREPARE和COMMIT，其中后两阶段：PREPARE和COMMIT都是投票阶段。在PlatONE的共识的正常流程中，这三阶段先后依次执行，其流程简要介绍如下：

- **PRE-PREPARE阶段：**在一个新的round上，对应的区块提议者节点产生区块（包含在PREPARE消息中）并广播给其他共识节点，同时切换到PRE-PREPARED的状态。其他共识节点校验提议区块的合法性，如果合法，则同样切换到PRE-PREPARED的状态，同时进入到PREPARE阶段；
- **PREPARE阶段：**在该阶段，共识节点发送PREPARE消息给其他共识节点，同时等待接收其他共识节点的PREPARE消息。当节点接收到超过2/3的其他共识节点发送的PREPARE消息时，其状态切换至PREPARED，同时进入到COMMIT阶段；

- **COMMIT阶段：**在该阶段，共识节点发送COMMIT消息给其他共识节点，同时等待接收其他共识节点的COMMIT消息。当节点接收到超过 $2/3$ 的其他共识节点发送的COMMIT消息时，其状态切换至COMMITTED。此时，共识中的区块就完成最终确认，节点就可以将区块插入到本地区块链中。

如下为PlatONE的共识流程及状态迁移图：



共识流程 - Round change机制

PlatONE的共识算法使用round robin算法根据具体的round来选取区块提议者。同时，在一个具体的round上，如果满足一定条件，例如共识没有在预期的时间内完成，则会触发round change流程。Round change完成后，共识节点将切换到新的round（一般情况下是上一个round+1）上，这也意味着区块提议者节点切换至新的共识节点。

具体来说，round change的触发条件有以下几种，其中任何一个条件满足将触发该流程：

- Round change定时器超时；
- 共识节点接收到无效的PREPARE消息；
- 区块上链失败。

Round change流程如下：

- 当一个共识节点判断上述round change触发条件之一产生时，将向其他共识节点发送ROUND CHANGE消息，其中附带提议的将切换到的新的round的编号，同时等待接收来自其他共识节点的ROUND CHANGE消息。上述新的round的编号的值按如下方式确定：

- 如果节点已经接收到其他共识节点发送的ROUND CHANGE消息，则节点会从数量达到 $F+1$ (F 为最大可能的拜占庭节点的数量) 或以上的round编号中选取最大的那个数值作为新的round的编号的值；
- 否则，节点将「当前的round编号+1」这个值作为新的round的编号的值。
- 任何时候，节点只要接收到同一round编号的 $F+1$ 个ROUND CHANGE消息，并且该round编号高于节点当前的round编号，则节点将发送对应于该更高round编号的ROUND CHANGE消息；
- 节点只要接收到 $n-F$ (n 为共识节点总数， F 为最大可能的拜占庭节点数量) 个对应着同一个round编号的ROUND CHANGE消息，则round change成功，节点切换到新的round上，并确定出新的区块提议者节点；
- 节点终止round change流程的另一个触发条件是节点通过节点间同步接收到被验证通过的区块。

共识流程 - 区块锁定机制

PlatONE的共识算法中，当共识节点针对某个提议区块接收到超过2/3的PREPARE消息时，节点将锁定在该区块上（前提是此前并未锁定在其他区块上）。PlatONE的共识算法通过区块锁定机制来提高共识的效率。

以下对区块锁定机制做一个简要介绍：

- 节点锁定在区块B、round R的含义是指，当前节点只能对区块B、round R的区块投COMMIT票。当一个节点收到了超过2/3的共识节点对区块B的PREPARE投票后，进入PREPARED状态。此时，节点被锁定，等待接收其他节点的COMMIT投票，并且锁定的round即为当前的round；
- 除了共识起始阶段，当同步到更高高度的区块时，或当前共识完成的区块成功上链时，锁定状态重置为非锁定状态，并开始新一轮对更高区块的共识。如未能在锁定期间收到指定round和区块的超过2/3的COMMIT投票，则触发ROUND CHANGE。为避免出现死锁场景，PlatONE的共识算法在代码层面也优化了相关的解锁实现。

共识流程 - Consensus proof 存储机制

PlatONE区块链中，区块中存储了对应的共识证明 (consensus proof)，也就是区块上链之前共识节点所接收到的超过2/3的节点的

Committed Seal签名（从COMMIT消息中获取）。因此consensus proof就可以作为上链区块的合法性的证明。

PlatONE的共识算法计划对consensus proof存储机制进行优化，以实现共识算法的可审计性。这里的可审计性是指，通过链上存储的相关数据来分析和审计共识节点的行为和表现。

4.3. 权限模型

根据系统中的不同实体对象，PlatONE将权限管理进行了模块化的拆分。针对系统中用户账户、节点和智能合约这三类实体的不同行为，分别设计了用户角色管理模块、节点管理模块和合约防火墙模块来进行权限的控制和管理。

角色管理：

PlatONE根据不同的权限，设定了不同的用户角色，并通过系统合约的方式对用户的角色进行管理。根据不同的角色，用户在系统中被赋予不同的权限。**目前设定了如下角色：**

| 用户角色（权限） | 作用 |
|------------------|--------------------------|
| chainCreator | 链创建者，在链创建时生成，是系统中权限最高的账户 |
| chainAdmin | 链管理员，由链创建者设置，可以设置多个链管理员 |
| nodeAdmin | 节点管理员，用于管理系统中的节点信息 |
| contractAdmin | 合约管理员，可以管理系统中的合约相关的权限控制 |
| contractDeployer | 链部署者，该角色表示用户可以在链上部署合约 |

每个角色的权限范围如下：

| | 链创建者 (超管) | 链管理员 (普管) | 节点管理员 | 合约管理员 | 合约部署者 |
|---------------|--------------|--------------|-------|-------|-------|
| 指定或取消链管理员 | √ | | | | |
| 指定或取消节点管理员 | √ | √ | | | |
| 指定或取消合约管理员 | √ | √ | | | |
| 指定或取消合约部署者 | √ | √ | | √ | |
| 新加节点申请 | √ | √ | √ | | |
| 管理所有的节点 | √ | √ | | | |
| 为自己部署的合约设置防火墙 | √ | √ | | | √ |
| 审核已部署的合约 | √ | √ | | √ | |
| 管理自己加入的节点 | √ | √ | √ | | |
| 部署合约 | √ | √ | | | √ |

- **节点管理**

PlatONE通过节点管理合约对节点进行管理，包括节点是否能够接入网络，节点是否能够参与共识以及节点信息的维护等功能。根据之前用户角色的设定，只有chainCreator、chainAdmin和nodeAdmin这三类用户才可以设置系统合约中的节点数据，当需要添加节点、更新节点状态、删除节点时都需要这三类账户来调用节点管理合约。

- **合约防火墙**

PlatONE中合约的调用权限由合约防火墙控制，只有合约的创建者才可以设置该合约的防火墙。

合约防火墙具备合约接口级别的访问控制，**通过如下两个列表实现：**

- **ACCEPT:** 可以访问相应接口的地址列表，相当于白名单；
- **REJECT:** 拒绝访问相应接口的地址列表，相当于黑名单。

4.4. WASM 创新与优化

Wasm在一开始的设计中其目标平台是浏览器，为了使Wasm更加适应区块链系统，PlatONE做了以下创新和优化：

- 改造原生Wasm中浮点数导致计算非确定性的设计，确保计算的确定性；
- 扩展对address、hash等数据类型的支持；
- 通过import的方式以支持标准库的方法和提供区块链的功能方法；
- 通过import的方式提供区块链接口给智能合约使用，从而完成智能合约和区块链的交互；
- 增加Gas机制，解决了智能合约中的恶意的死循环攻击问题；
- 提供沙箱运行环境，实现资源隔离，使智能合约自身的崩溃不会影响到其他智能合约的运行和底层安全；
- 提供编写智能合约所需的各种语言的类库；
- 提供合约调试功能接口。

4.4.1. WASM/EVM合约互调用

在PlatONE中，实现Wasm与EVM合约互调用需解决的问题：Wasm合约与solidity合约互相调用过程中涉及到的不同输入的编码方式、指定调用合约函数的函数名、参数与参数类型解析、返回值与返回值类型解析以及底层虚拟机解释器的自动切换等问题。

为解决上述问题：PlatONE中引入了一个新的precompiledContract，作为Wasm与EVM合约互调用的桥梁，以此解决合约调用过程中涉及到的输入的不同的编码方式、调用合约的函数名和参数等问题；通过底层虚拟机解释器做到对于不同类型合约的返回值的兼容处理，解决了合约互调用中返回值与返回值类型解析的问题；通过对于合约字节码的处理，分辨出当前虚拟机所运行的合约的类型，从而完成虚拟机解释器的自动切换。

4.5. 合约数据迁移协议

PlatONE实现了一套合约数据迁移协议，支持一键迁移全量旧版本合约数据。

PlatONE底层账户数据存储模型采用了默克尔树。在每个合约账户对应的默克尔树的叶子节点上，存储了用户数据的键值对。在PlatONE中，合约业务层写入的 Key - Value，在底层分别对应为

Hash(ContractAddress+Key), Hash(Value), 可通过映射表Pre-Image实现键值Hash值到其原文的映射。

在迁移时，我们基于旧版本合约的默克尔树根，遍历地为新合约构造存储树。在遍历过程中，重新生成正确的键值，即实现每个存储键Hash(ContractAddress+Key) 中合约地址的替换，从而实现正确的全量数据克隆。

4.6. CNS（合约命名服务）方案

合约命名服务通过系统合约cnsManager维护合约名称、版本到合约地址的映射关系，合约信息如下所示。

```

1. Struct ContractInfo
2. {
3.     std::string name; //注册合约名
4.     std::string version;//合约版本，如1.0.0.0
5.     std::string address;//合约地址0x...
6.     std::string origin;//创建者地址0x...
7. };

```

cnsManager合约提供注册、查询等功能，cnsManager合约的地址固定为"0x00011"，用户可以通过该地址来注册、查询合约信息。假如用户查询合约信息时未指定版本，则默认访问的是合约名称对应的最新版本合约。

PlatONE兼容两种方式调用合约，既可以根据地址调用，也可以根据合约名称、版本调用。两种调用方式通过交易类型区别，PlatONE中的transaction类型如下所示，其中的txType字段表示交易的类型，to字段表示要调用的合约地址、data表示调用的方法参数信息。

```

1. Type transaction struct {
2.     to    *common.Address
3.     nonce uint64
4.     amount *big.Int

```

```

5.   gasLimit uint64
6.   gasPrice *big.Int
7.   data []byte
8.   txType uint64
9. }
```

| txType | data | to | 类型说明 |
|---------------|---|-----------|--------------|
| 0x2 | [txType, "method", arguments] | 合约地址 | 根据地址调用合约 |
| 0x11 | [txType, "name:ver", "method", arguments] | nil | 根据合约名称版本调用合约 |

在交易执行时，系统会根据交易类型执行交易。如果交易是根据合约名称、版本来调用合约，PlatONE底层自动在系统合约中查询名称版本对应的合约地址，然后调用该地址的合约。

5. 技术路线图

- **第一阶段：2019年9月**

- 优化BFT
- 支持WASM
- 支持权限模型
- 多种合约管理机制
- 丰富的密码学算法
- 完备的运维工具与开发包
- 开源

- **第二阶段：2019年10月**

- Wasm - 支持更多前瞻性语言
- 工具优化
- 网络层优化
- 支持代理重加密

- **第三阶段：2020年1月**

- 新增共识算法并实现插件化
- 提升去中心能力
- Wasm - 支持更多前瞻性语言
- 支持共识审计
- 运维工具升级

- **第四阶段：2020年5月**

- 支持多链架构
- 支持并行计算
- 大数据存储优化
- 增加治理机制
- 增加形式化验证

- **第五阶段：2020年9月**

- 支持跨链平滑升级
- 链存储数据工具
- 支持更多常用的数据库

6. 应用场景

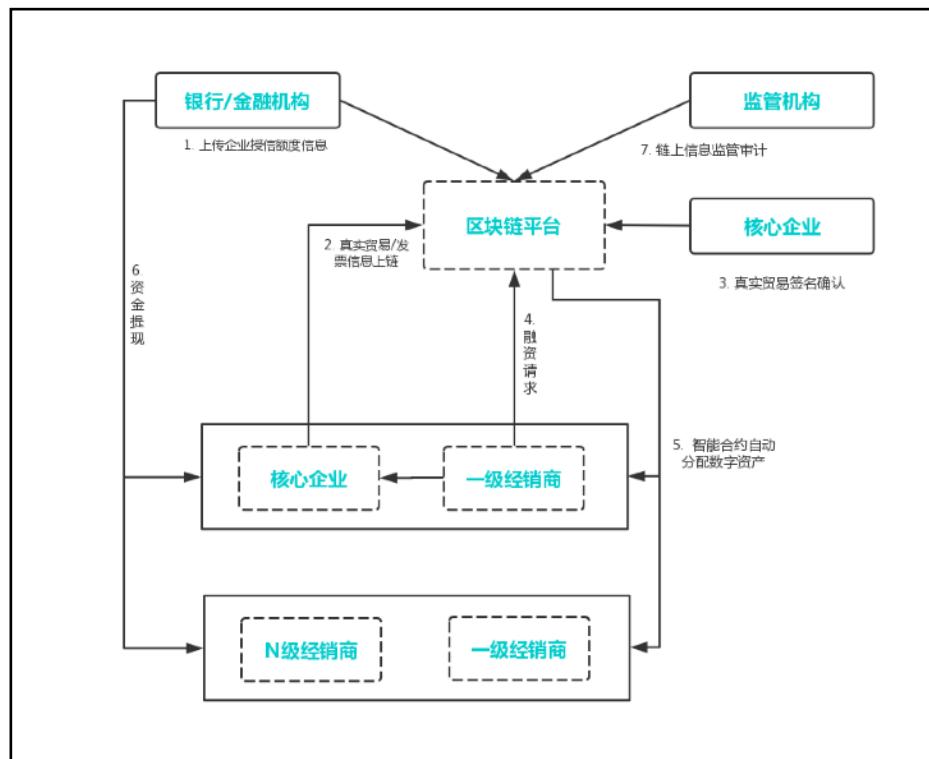
基于PlatONE联盟链基础平台，利用其提供的技术特性，可以实施开展和行业相契合的各种业务场景，包括如下方面。

6.1. 供应链金融

可为供应链上下游企业构建一个信息对称共享、核心企业信用价值可传递、商票可拆分流程、风险可控的新型供应链金融融资模式，并为监管提供数据追溯便利，提升行业整体服务效率。

传统供应链金融的痛点：

- 监管难以穿透** - 供应链层级的繁复，使贸易真实性和交易透明性无法简单通过系统进行确认和审核，造成了监管的不便利性。
- 商票不可拆分、流转** - 传统的商票不可拆分，供应商无法基于商票再次背书转让，核心企业信用无法有效传递给多级供应商体系。
- 金融机构风险敞口较大** - 供应商、经销商之间的约定或合同信息无法得到有效确认或核实，使得金融机构存在较大的授信风险。
- 核心企业信用无法传递** - 在传统供应链金融多级供应商体系下，信息难以有效传递，使得一级供应商以外的其他层级供应商无法享受到核心企业的信用，融资较难。



基于区块链技术和密码学算法，PlatONE为供应链金融提供了资产可数字化确认、处理、流转的平台解决方案，主要功能模块有：

- **资产登记** - 企业债权可通过区块链进行登记存储，形成不可篡改的数据记录，实现各参与机构间的信息实时共享。
- **资产确权** - 通过相关参与方的确认，由智能合约自动将应收账款和核心企业信用转化成数字资产并登记到相应账户，实现资产的确权。
- **资产数字化** - 以链上确权数据信息为基础，通过智能合约自动为企业建立可在区块链联盟间进行交易和流转的数字资产。
- **数字资产管理** - 支持不同属性资产的统一管理和查询，通过预设的智能合约实现链上资产的自动化分配、拆分、流转和注销。
- **监管审计** - 提供监管审计入口，赋予监管机构审计权限，可查看平台上所有资产的交易。
- **多层级隐私保护** - 运用广播加密、同态加密、零知识证明等加密算法保护供应链金融各参与方的数据安全和隐私保护。

适用场景：

- **授信融资** - 金融机构对客户授予信用额度，在这个额度内客户向银行借款可减少繁琐的贷款检查。
- **应收款融资** - 企业以自己的应收账款转让给银行并申请贷款。
- **票据融资** - 将商业票据转让给银行，银行按票面金额扣除贴现利息后将余额支付给收款人。

PlatONE已被用于打造国内首款基于区块链技术，专注于汽车供应链，服务于汽车产业核心企业的金融产品。产品提供供应链应收货款融资的高效解决方案，解决了多方信任的问题，在保障数据安全的条件下解决数据主权问题。另外，PlatONE还被业内高科技公司和银行共同打造成基于区块链供应链金融平台，平台活跃用户已达56家，融资额累计至数千万。

6.2. 防伪溯源

利用区块链链上信息不可篡改和可追溯的特性，与现有业务场景相结合，提供数据存储、溯源和验证的一站式服务，实现可信的商品防伪溯源。基于区块链、物联网和智能防伪等技术记录产品生产各环节的信息，提供产品的区块链溯源服务和企业营销服务。

PlatONE已在防伪溯源领域进行了相关方案设计和POC，可实现以下业务流程：

① 数据存储与登记：

- 物联网设备扫描信息后直接上链，确保一手数据来源的真实性。
- 产品全流程信息区块链存档，不可篡改。

② 数据查询与验证：

- 各环节用户或消费者可自主选择区块链节点进行信息查询。
- 通过智能合约维护数据处理逻辑，为用户提供原始数据的验证渠道。

③ 数据统计与分析：

- 支持消费者查询数据的统计，支持精准营销。
- 基于扫码数据进行销量分析，指引销售策略。

基于PlatONE联盟链实现的防伪溯源方案具备以下优点：

- 物联网增强数据真实性 - 将区块链SDK嵌入产线扫码设备，实现扫码数据直接上链记录，减少人工干预，提高数据真实性。
- 数据隐私保护 - 加密算法保护平台参与机构间的溯源核心数据安全，减少开放服务的交易摩擦，保障平台信息安全。
- 平台化运营 - 建立行业溯源平台生态，进行生产流通数据的精细化运营挖掘，协助企业生产决策。

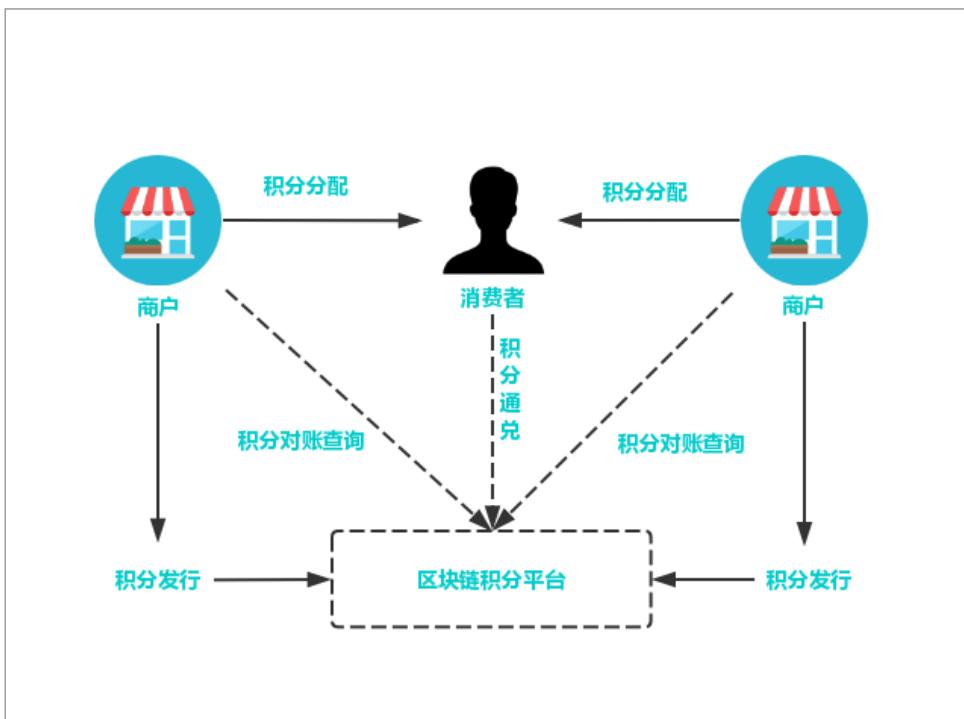
6.3. 积分管理

基于PlatONE可实施一站式积分管理平台，引入积分发行方、商户、消费者。将积分的发行、流通、消费等行为审计上链。

现有积分体系的痛点：

- 积分使用不便 - 单一企业的积分存在使用限制和兑换局限，难以提升消费者积分使用积极性，无法形成积分奖励策略的良性循环。
- 积分难以流通 - 缺少便捷的异业通兑渠道，跨行业或机构的积分结算较复杂，难以形成广泛的积分转让、赠送、跨平台使用渠道。
- 系统建设成本高 - 自建积分系统成本高，涉及合作伙伴间的积分兑换，则需要系统对接并设置兑换比例，增加了管理和维护成本。

方案详情：



基于PlatONE打造的积分通兑互换平台，可支持不同企业以合约的方式快速进行积分的发行、兑换比例设置和交易结算。

- 积分发行 - 积分发行商可通过平台自主维护积分发行数量，设定积分奖励比率。
- 积分兑换 - 通过智能合约维护积分兑换比例，实现消费者消费时积分的自动增减。
- 积分通兑 - 通过上层合约，可实现跨机构间的积分使用，并按照各企业设置的积分兑换比例，快速完成积分转换和账务核对。
- 积分查询 - 积分参与方或消费者可在链上实时查询积分奖励和交易记录。
- 积分对账 - 根据积分链上流转记录，支持积分发行商与商家、积分发行商之间的自动对账。

适用场景：

- 积分商城 - 用户可凭积分兑换物品或服务。
- 员工福利 - 企业可用积分激励或奖励员工，员工可使用积分兑换福利。
- 会员积分 - 商户给消费者发放的会员卡（或账户）进行积分，积分一般限于当地或者发卡商户使用。
- 异业积分通兑互换 - 跨行业积分联盟之间积分互兑，例如：航空积分兑换酒店住宿。

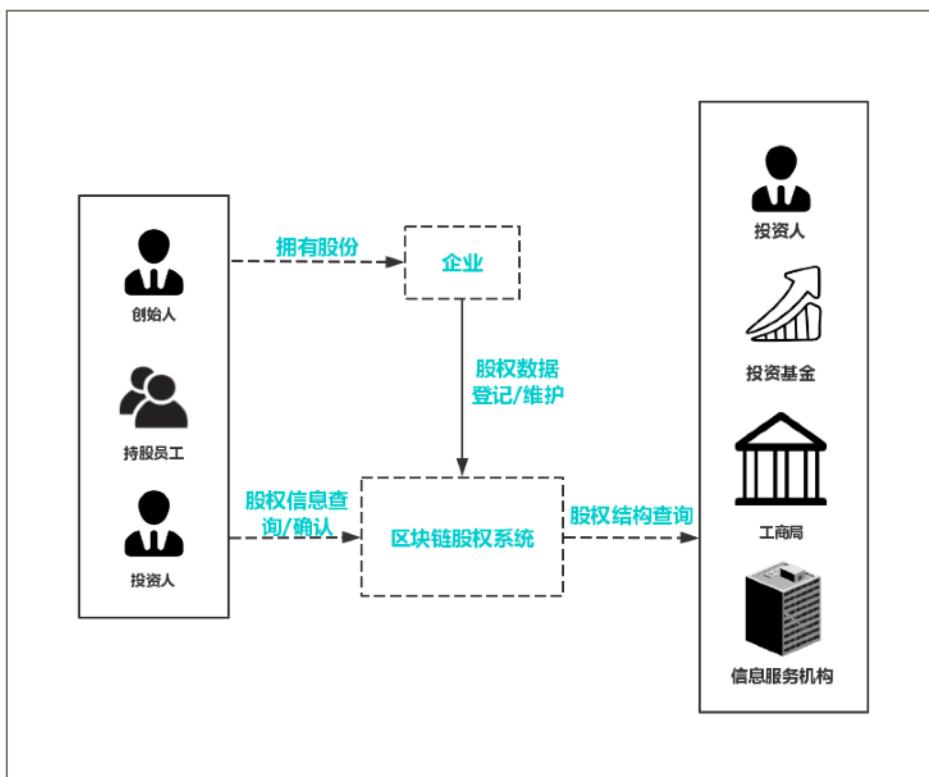
6.4. 股权登记

基于PlatONE可实现企业股权信息的及时登记确认与维护，通过分布式架构降低机构对接门槛，实现股权数据的实时安全共享，保证重要信息披露的准确性和透明性，为企业提供更为便利和高效的金融服务支持。

传统股权登记流程的痛点：

- 信息同步滞后 - 依托工商局的股权登记，具有滞后性，股权的变更信息不能实时同步，不利于潜在投资者查看公司真实股权信息。
- 股权所属权维护关系薄弱 - 企业频繁的股权变更，会给股东名册的维护增加困难，不利于历史交易的维护和跟踪。
- 传统方式不便利 - 纸质材料传递和人工办理的不便利。

方案详情：



基于PlatONE构建的股权登记、查询平台，可为企业、股东提供可靠的股权登记、变更及交易模式，并支持与工商、信息服务机构的对接，提升场外股权登记、维护效率和可信度。

具体功能模块有：

- 股权信息登记 - 在链上进行股权信息的登记和存储，区块链的节点共识特性可保证参与方间数据的一致性。区块链完整记录了股权所有者信息和变更记录，确保数据的不可篡改。
- 股东名册维护 - 企业可将股东名册登记上链，股东可在第一时间获得股东名册的更新信息并通过签名进行确认，以此来确保股东名册的一致性和有效性。

- 股权信息同步 - 与工商或者服务机构间的信息同步和确认，基于区块链的股权登记平台，可形成企业可追溯、可审计、可靠的企业信息及股权登记变更档案。

适用场景

- 创业公司股权管理 - 创业公司员工持股、股权激励、预分配等。
- 场外市场股权登记 - 场外股权托管、挂牌、交易等。

6.5. 物流

在现有的整车物流运输业务中，使用传统的纸质运单作为物流过程中的作业交接凭证和结算凭证，并通过经销店确认后流转回主机厂进行核对结算。该方式具有运单流转周期长、效率低、结算核对繁重、成本高等缺点。

2018年11月30日，万向区块链与物流、银行等合作伙伴于上海联合宣布，基于区块链技术的“运链盟-汽车供应链物流服务平台”正式上线。其中最新版的平台中所运用的区块链技术正是PlatONE，而该平台是国内首个区块链技术在汽车整车物流行业的落地案例，目前越来越多的用户加入运链盟平台。

最新一代的运链盟是一个基于PlatONE区块链技术，以汽车整车物流作为实际业务场景，集物流、结算与供应链金融三大功能模块的综合服务平台，旨在利用区块链技术，通过解决价值传递过程中博弈多方互信等痛点，为实体经济注入新的力量源泉。

- 首先，运链盟实现了物流运输过程中的订单、运单电子化，以及上下游企业在线对账模式，能够有效降低传统纸质单据的成本。
- 第二，业务流程链上管理，上下游企业可实现数据共享，提高整体运作效率。
- 第三，金融机构依托汽车主机厂商的信任传递，以及在线应收账款记录和发票，为承运商提供融资服务，中小承运商也能获得更多融资机会。
- 第四，区块链可保障记录数据真实可靠，为所有业务方提供全流程可追溯、穿透式资产确权和验证渠道，减少造假可能性，推动行业健康稳定发展。

6.6. 慈善行业

在《慈善法》正式施行3周年之际，慈善信托这个曾被业界视为拥有“千亿蓝海”市场的信托业务，迎来了备案数量的爆发式增长。

统计数据显示，近一年来，已经备案的慈善信托数量达108单，同比大增83%。在备案数量大幅增长的情况下，慈善信托的备案规模却出现同比下降的情况。

为慈善组织设立单独信托账户进行资产管理，用户使用区块链技术实现即入账即配置，在一个账户配置金融产品、管理财产权；协助慈善组织优化现金流管理，提升投资管理的水平。利用信托公司强大的中后台实力，为慈善组织提供预算管理、善款发放、期间管理等资助项目托管服务；以及慈善资产相关的财务、法律、合规的外包服务。使慈善组织可以更好地专注慈善目标，节约时间与成本。

除了以上场景，PlatONE还助力在大宗商品监管与贸易融资，汽车市场、农产品溯源、信托与慈善服务等领域完成了POC。

7. 术语表

专用集成电路：专用集成电路是针对整机或系统的需要，专门为之一设计制造的集成电路。

拜占庭容错：拜占庭将军问题首次由Leslie Lamport, Robert Shostak和Marshall Pease在1982年提出。具备拜占庭容错能力的分布式网络能减轻恶意节点对网络的影响并在诚实节点间达成正确共识。目前有几类BFT协议可以提高系统拜占庭容错能力，Miguel Castro和Barbara Liskov提出的实用拜占庭容错(PBFT) 是这些协议之一。

电路：一种通用的计算表现形式，由不同类型的门（gate）组成。由逻辑门构成则成为布尔电路（Boolean circuit），由算术门构成则叫算术电路（Arithmetic circuit）。

安全多方计算：无可信第三方场景下多个参与方协同计算，获取计算结果，并不泄露各自输入信息。

(全) 同态加密：在密文上进行计算，既能保证隐私又能提供可操作性。全同态是指支持所有操作的计算。

可验证计算：可有效验证结果数据是否按照原始数据依照指定逻辑计算而来。

零知识证明：证明者让验证者确信某个事实的正确性，并不泄露其他任何信息（零知识）。

形式化验证 (Formal Verification) :指从数学上完备地证明或验证电路的实现方案是否确实实现了电路设计所描述的功能。形式化验证方法分为等价性验证、模型检验和定理证明等。

LLVM IR: LLVM是一个构架编译器（compiler）的框架系统。IR（Intermediate Representation）是其编译器前端输出的一种硬件无关的类汇编中间语言。

默克尔树：是一种二叉树，包含含有基础信息一组叶子节点，一组中间节点，和一个树根。节点都是它的2个子节点的哈希。

TPS: 即Transactions Per Second, 每秒处理的事务数量，用以衡量系统扩展性。

安全性 (Safety) : 在分布式系统的算法和设计中，指不好的事永远不会发生。

活性 (Liveness) : 在分布式系统的算法和设计中，指好的事情最终一定会发生。

8. 参考文献

Castro, Miguel, and Barbara Liskov. "Practical Byzantine fault tolerance." OSDI. Vol. 99. 1999.

S. Goldwasser, S. Micali, C. Rackoff , “The knowledge complexity of interactive proof systems” , SIAM Journal on Computing, 1989.

B. Manuel, F. Paul, M. Silvio “Non-Interactive Zero-Knowledge and Its Applications” . STOC, 1988.

R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. In Foundations of Secure Computation, 1978.

C. Gentry. Fully Homomorphic Encryption Using Ideal Lattices. STOC, 2009.

A. C. Yao, Protocols for Secure Computations (Extended Abstract). FOCS, 1982.

A. C. Yao, How to Generate and Exchange Secrets (Extended Abstract). FOCS, 1986.

O. Goldreich, S. Micali, A. Wigderson:How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. STOC, 1987.

S. Micali, “Computationally Sound Proofs” . SIAM Journal on Computing, 2000.

B. László, F. Lance, L. A. Leonid, S. Mario, “Checking Computations in Polylogarithmic Time” . STOC, 1991.

S. Goldwasser, Y. T. Kalai, G. N. Rothblum. “Delegating Computation: Interactive Proofs for Muggles” . STOC, 2008.

G. Rosario, G. Craig, P. Bryan. “Non-Interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers” . CRYPTO, 2010

Parno, Bryan, et al. "Pinocchio: Nearly practical verifiable computation." 2013 IEEE Symposium on Security and Privacy. IEEE, 2013.

R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. In Foundations of Secure Computation, 1978.

Armknecht, Frederik; Boyd, Colin; Gjøsteen, Kristian; Jäschke, Angela; Reuter, Christian; Strand, Martin (2015). A Guide to Fully Homomorphic Encryption.

Craig Gentry. Fully Homomorphic Encryption Using Ideal Lattices. In the 41st ACM Symposium on Theory of Computing (STOC), 2009.

C. Gentry, S. Halevi, and N. P. Smart. Better Bootstrapping in Fully Homomorphic Encryption. In PKC 2012 (Springer)

C. Gentry, A. Sahai, and B. Waters. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. In CRYPTO 2013(Springer)

《中国区块链技术和应用发展研究报告（2018）》，中国区块链技术和产业发展论坛，2018.12.18

《中国区块链技术和应用发展白皮书（2016）》，中国区块链技术和产业发展论坛，2016.10.18

《区块链 隐私计算服务指南》，中国区块链技术和产业发展论坛，2019.07.19

《区块链 隐私保护规范》，中国区块链技术和产业发展论坛，2018.12.18

《区块链 参考架构》，中国区块链技术和产业发展论坛，2017.05.16