THE FORCE PROTOCOL
原 | 力 | 协 | 议

原 力 协 议

The Force Protocol

May the force be with you !

https://www.theforceprotocol.com

原力协议

The Force Protocol

分布式加密数字金融服务协议

白皮书 V2.1

2019 年 5 月

本文仅供参考之用，不构成在任何司法管辖区出售证券或招揽购买证券的要约。

摘要

截止2018年1月，全球加密数字资产价值总额最高已达8000亿美金。除加密数字资产交换外，在以太坊体系上的一系列加密数字金融服务已成为加密数字资产持有人获得的金融服务的新选择。然而，由于不同加密数字资产金融服务应用平台基于不同的规则，使得加密数字资产金融服务应用间缺乏互动，协同开发难度大，用户使用体验较差。

原力协议是分布式加密数字金融服务协议，基于主流公链系统和底层跨链协议，通过对分布式金融业务流程的抽象和封装，以SDK及API的形式，用一站式解决方案赋能去中心化金融应用开发。为跨平台资产流转、交易深度共享、跨链加密资产抵押的稳定币发行、通证债券发行、链上支付、交易清结算等金融需求提供解决方案。原力协议在分布式金融生态里搭建去中心化金融服务的通用开发平台，通过将分布式金融当中与具体场景相关的需求框架化、系统化，为分布式网络内的金融服务提供通信标准和开发框架，让加密资产可以在统一的框架内实现安全高效的价值互联互通。

阅读提示：

本白皮书重点描述原力协议的内容计划，旨在开发新的基于区块链技术的底层协议，并生成 FOR（EFOR）代币在原力协议平台使用。本文档中的任何内容均不应被视为对原力协议的业务、平台或代币及如何发展的保证或承诺，也不应被视为平台或代币的效用或价值的保证或承诺。本白皮书中描述的原力协议计划可根据实际项目发展需要和外在环境变化进行适当调整。该项目将不可避免受到客观因素影响，包括：市场波动、政策变化和加密数字资产行业内的因素等。任何对未来的描述均将基于原力协议对本文档中所述问题的分析之上。

本白皮书中代币相关的内容，不构成 FOR（EFOR）代币或任何其他代币购买机制的要约或售卖协议。本白皮书仅作为社区爱好者了解原力协议内容、FOR（EFOR）代币和原力生态等相关问题的文件。投资者做出投资决策时，应结合其他确定的报价文件阅读此白皮书。FOR（EFOR）代币的互换或代币的捐赠将受许多潜在风险的影响，代币互换文件中将描述部分风险。原力代币贡献者应该意识到，此类贡献有失去全部或部分价值的风险。

目录

1 项目概述

1.1 引言

以加密数字资产为代表的区块链技术为世界定义了一种新型的生产关系。在该新型关系中，数字产品在生产过程中的各类生产效益都是共享的，而不再因制订规则标准、控制过程等原因被任何中心化组织切分大部分收益。以以太坊、小蚁等为代表的区块链 2.0 技术的发展使加密代币的发布和应用更加便利，这进一步加快了实物资产上链和加密数字经济的发展。在过去的几年中，全球加密数字资产的市值持续增加。截止 2019 年 4 月，在全球范围内由区块链技术产生的加密数字资产（含加密数字资产和代币）的市值约为 2125 亿美元。这意味着一个使人类和世界紧密连接的全新经济时代即将来临，而加密数字资产借贷将成为数字经济中最基本的金融需求。

目前，借鉴传统金融，加密数字金融领域出现了一些新的尝试，诸如交易所、借贷、债券、金融衍生品、量化投资等。但普遍存在市场较小，体验较差，开发难度大等问题。面对复杂的区块链技术，很多市场参与者望而却步，加密数字金融广阔的发展前景与区块链技术的不成熟之间存在着巨大的鸿沟。

基于上述现状，原力协议团队选择在分布式金融生态里搭建去中心化金融服务的通用开发平台，通过将分布式金融当中与具体场景相关的需求框架化、系统化，为分布式网络内的金融服务提供通信标准和开发框架，让加密资产可以在统一的框架内实现安全高效的价值互联互通。

原力协议为搭建去中心化的金融服务应用提供支持，以借贷场景为例，基于该协议的平台（下称超级节点）可实现借贷订单全球共享，大大增强交易深度。协议还支持超级节点在政府批准后搭建在监管范围内的稳定币体系。针对加密数字资产领域的洗钱现象和加密数字资产价格波动，原力协议还设计了反洗钱策略和用户在急速爆仓情境下降低资产损失的方法。

1.2 项目目标

1.2.1 构建全球范围的加密金融服务网络

原力协议是一个开源的分布式加密数字金融开放平台，向加密数字金融服务

应用开发者提供基于跨链技术的解决方案。具体而言，原力协议将基于当前主流公链及未来会上线的原力协议公链，通过对加密数字金融业务通用模块的抽象和封装，以 SDK 及 API 的形式对外提供服务。原力协议向应用层封装网络通信，协议编解码，异常处理等细节，开放出友好的面向对象的功能接口。应用服务面向接口编程，专注于实现业务逻辑，而不需要承担区块链底层技术实现和维护的开销。

1.2.2 简化用户使用体验

原力协议团队致力于为上述加密数字资产借贷平台提供友好的接口。通过应用此类接口，用户可以很方便的建设借贷平台。在原力协议的借贷场景中，原力协议团队基于对现有各借贷平台的流程分析，将整个流程从 10 多步细小步骤，简化合并成 2-3 步。在成交前，基于原力协议的应用平台（超级节点）不要求借贷双方发送燃料（gas）保障借贷活动的开展，而是利用“授权/锁定”功能从用户钱包接触加密数字资产，减少不必要的步骤，进而降低了燃料消耗。

1.2.3 实现去中心化借贷和监管下的双代币模型

原力协议支持机构用户基于共享订单簿开发全球去中心化的借贷平台。通过引入功能币与稳定币互相结合的双代币模型，原力协议排除了不同加密代币，及不同加密代币与法币之间的障碍。原力协议基于 COSMOS 的 IBC 协议实现跨链交易，通过超级节点的加密货币兑换服务实现加密资产与法币的交易。

1.2.4 实现借贷需求发布和借贷风险分担

原力协议支持用户基于去中心化的借贷平台发布借贷需求。通过发布的借贷需求，用户可方便获取借贷市场的资金流动状态和其他用户为获取资金流动性愿意承担的代价。在本协议未来的更新版本中，出借人可以选择抵押资产的类型以及设置借款人信用要求。此外，协议也将考虑实现群借贷功能，帮助出借人实现风险分担。

1.2.5 剧烈变化市场中保障用户资产安全

原力协议将通过价格反馈机制实现抵押管理。当抵押资产出现剧烈价格波动导致抵押资产触及预警线时，借贷平台（超级节点）将联系借款人补交抵押代币。

如果借款人拒绝补交抵押代币，抵押资产价值到达平仓线后，智能合约自动启动平仓程序，计算应还本金和利息，将等值的抵押资产转移给出借人，并将剩余部分归还借款人。针对可能出现的抵押代币价格波动剧烈导致智能合约来不及清仓的情形，原力协议团队引入了借贷保险产品，购买保险的出借人可从保险池资金获得补偿。

1.2.6 应用人工智能和大数据技术实现精准预测

借贷行为发生后，借贷年化利率、抵押币种、抵押率、交易地址、借贷双方信息、手续费缴纳等数据将会随之产生，信用记录也将产生并永久保存。基于上述数据，引入人工智能和大数据技术，基于以太坊、EOS 分布式账本可实现预测功能，为原力协议未来开发信用借贷提供支撑。

1.3 主要贡献

1.3.1 机构和个人用户 API 接口

原力协议为机构用户和个人用户提供 API 接口，使得建立超级节点或个人点对点借贷变得更加容易。超级节点用户最少情况下只需要建立相应的前端应用和数据库并调用接口即可。个人定向交易用户通过调用原力协议提供的接口，可以借用智能合约为交易双方的资产提供安全保障。

1.3.2 稳定币

为改进 USDT、GUSD、PAX 等稳定币的不足，原力协议将引入两种新的稳定币生成机制，并由超级节点负责具体的发行事宜。

第一种机制，稳定币供应商根据自身的风险控制能力、资产偏好、选择借贷抵押资产，根据原力协议提供的开发框架，开发抵押、风控、平仓等合约，为持有不同公链资产的用户提供去中心化质押资产借稳定币的服务。经过市场竞争，在资产组合、风控、用户体验、治理方面最优秀的 DAPP 团队将会胜出，成为原力协议体系内的主要稳定币供应源之一，维护其发行稳定币的币值稳定。

第二种机制里，作为理财服务的提供者，超级节点和法币投资者签订合约，投资者将资金存入超级节点在第三方托管银行的账户。以此账户资金为储备金，超级节点将发行相同数量的稳定币，整个过程受到第三方托管银行和审计机构的监督。

通过上述操作，基于原力协议的稳定币对应法币都有充足的 1:1 储备。这些稳定币将为加密数字资产市场带来巨大的增量资金，进一步促进加密经济的发展。

1.3.3 共享订单簿

全球所有基于原力协议的超级节点（应用平台）共享同一个订单簿，这将在加密数字资产借贷市场充分释放集市效应，大幅提高每笔订单达成交易的速度和质量。通过共享订单簿和原力协议提供的其他功能，各超级节点可将更多精力集中于提高服务质量，开拓市场以吸引用户，进而增加手续费收入，而不必担心借贷平台构建、平台安全等技术层面的复杂问题。因此，原力协议上线后，将促进各借贷平台提供更好的服务。

1.3.4 价格急速大幅下跌时的保险产品

在加密数字资产价格大幅波动条件下，为避免智能合约来不及强制平仓价格已经跌破平仓线而给出借人带来损失，原力协议团队设计了保险产品。当发生上述情况，购买保险服务的出借人将从保险池中获得赔偿。

1.3.5 基于原力协议的实例

基于原力协议，原力协议团队也将开发 DAPP 作为原力协议生态体系的超级节点之一。通过该超级节点可达到如下目标：首先，该超级节点可作为原力协议上的首个超级节点 DAPP，用户可在其上自由借贷；其次，原力协议团队及相关人员可在超级节点上测试原力协议性能，为后续协议层面的升级完善提供支撑；最后，用户可以从该超级节点获得对原力协议使用场景的直观印象。

1.4 项目效益

1.4.1 去中心化

通过智能合约的自动执行，去中心化的程序设计可以不借助中心机构和服务设施实现借贷服务，从规划层面为借贷提供了安全保障。去中心化极大的改变了传统借贷体系的架构，因为智能合约一旦部署，借贷流程不再依赖对任何一方的信任。智能合约为借贷市场提供了去中心化的、不依赖于交易双方信用的借贷环境，这些在传统法币借贷市场是不可实现的。

1.4.2 信任

交易双方借贷流程完全基于代码。用户在原力协议超级节点建立订单并提交后，相关的智能合约一旦部署，交易双方、借贷平台以及底层的原力协议均无权操纵、阻碍或停止借贷流程的进行。在去中心化借贷平台上开展借贷的用户，对方是否值得信任已经不再是风险因素。通过智能合约，用户能够避免任何与第三方平台相关的风险，用户不需要考虑第三方是否欺诈、遭受黑客攻击或申请破产清算。

1.4.3 透明

区块链技术提供了超级账本功能，部署在链上的每笔交易都将被永久存放并可通过区块浏览器公开查询。区块链账本的该项功能，使得不同金融机构间的交易不再需要参与者之间的信任。在金融市场时间至关重要，由于政策限制、交易不透明、缺乏信任等原因，传统的金融借贷体系无法提供全球范围内资金的高速流转服务，而借助区块链技术可满足此类需求。资产出借者和借款者可随时、随地在线查询对方资产的到账情况，而无需信任和复杂的流程。

2 原力协议稳定币

近年来，随着加密数字资产技术的发展，使用 BTC 等主流加密数字资产购买商品或服务已经逐渐被大众接受。由于 BTC 等主流加密数字资产存在价格波动，在支付时需要经过换算，这给使用者带来了一定的不便。此外，BTC 等主流加密数字资产的持有者更多将其视为具备升值空间的资产，而非随时可以动用的资金，这种思想抑制了人们在日常生活中使用 BTC 等主流加密数字资产。

以 USDT 为代表的币值稳定的加密数字资产/代币，在 2017—2018 年被加密数字资产的投资者/交易所广泛接受。该现象说明广大加密数字资产投资者对于稳定代币存在需求。然而，由于 Tether 公司在财务透明度方面受到质疑，加之其并未得到美国政府的背书，USDT 的地位一直不够明朗。这导致，一方面人们出于对稳定币的需求不得不广泛使用 USDT；另一方面，如果有原理、背景优于 USDT 的稳定币出现，则一定会被大量用户用来替换 USDT。此后，诸如 DAI、TUSD、USDC 等稳定币陆续推出，都体现了业界对改良稳定币内在机制的愿望。

经过多年的发展，以比特币为代表的投资产品能否被美国、日本、欧盟等国家/地区的政府批准，决定了传统金融体系内的天量资金是否有合法合规的渠道进入加密数字资产领域。各大国的监管风向密切影响着投资人、交易所、项目方等各种利益团体的动向。2018 年 9 月，GUSD、PAX 两种受到美国金融体系监管的稳定币面世，被业界普遍解读为加密数字资产领域的一次重大突破，进而在加密数字资产世界掀起了一股稳定币热潮，也为稳定币业务进一步发展指明了方向。

然而，从 GUSD、PAX 等稳定币的发行机制看，随着需求量的不断增加，其发行机构必须不断地投入法币为币价稳定做背书，而发行机构能够投入法币数量是有极限的，一旦需求量超过其发行能力，类似于 USDT 的脱锚增发现象将极有可能发生。此外，为维持自身的业务运作，发行机构将有极强的动机挪用本该被锁定的法币。因此，随着需求端的不断增加，在现有的发行机制不变的情况下，类似于 GUSD、PAX 等的稳定币同样面临体系崩溃的风险。

什么样的稳定币机制更合理呢？

参考法币的发行机制，中央银行通过公开市场操作、再贴现等主要工具控制货币供给。以美国为例，美联储通过购买国债，将美元输送给政府，政府再以各种直接投资和政府采购将美元输送给经济体，带动经济增长。商业银行通过再贴现等途径获得来自美联储的法币，企业和个人又通过企业债、贷款等形式获得资金，促进法币在经济体系内流通。以美国政府的信用做背书进行美元的发行，同时以美国的国家实力和信用确保美元的全球流通，构成了美元发行和流通的主要机制。债权及信用在现代社会货币创造方面具有决定性的作用。以此类推，稳定币的发行机制同样可以建立于对法币的债权及信用之上，这也是 USDT、PAX、USDC 等主流稳定币所采用的模式。

另一种被采用的稳定币模式，是基于超额加密资产质押，以中心化、半中心化或去中心化形式发行的稳定币。这种类型的稳定币，本质上是一种债权凭证（IOU），以发行方所提供的担保物作为资产保障，发行价格稳定的等价物。发行方为了维持币价的稳定，大多会采取一定的机制平衡价格，例如 BitShares，就采取了由发行方进行 2 倍以上超额抵押 BTS 的机制，但是 BitShares 不具备自动调节机制，需要发行方根据市场波动，手动调节担保物的质押率；而 MakerDAO 则利用以太坊的智能合约体系，引入了 MKR 代币作为币价平衡的工具，由用户和智能合约共同来维护 Dai 的币价平衡。MakerDAO 的质押合约量一度占到以太坊质押量的 60%以上，足以说明超额加密资产质押这种模式具有用户需求和应用场景。

加密数字资产借贷业务与稳定币业务存在天然联系，为保障借贷业务的健康快速发展，必然会涉及稳定币业务。为克服 USDT、GUSD 和 PAX 等稳定币经济模型的不足，原力协议提出一种新的稳定币产生机制。在原力协议稳定币体系内，有超级节点、投资人、借款人、加密资产质押合约等主要参与方，各方作用如下：

2.1 原力协议超级节点

超级节点是各国家、地区基于原力协议全球订单簿的金融服务提供商，在合规的前提下，开展法币/加密数字资产借贷、支付、交易、清算等业务。原力协议通过超级节点维护共享借贷订单簿，超级节点的加入和退出，需要由社区治理流

程进行判定。超级节点实体上可以是 DAPP 运营团队、加密数字资产钱包、中心化交易所等。所有的原力协议超级节点都必须持有其所在地政府发行的加密数字资产兑换法币业务许可，接受政府的合规监管。在业务层面，超级节点需要对接原力协议全球借贷协议、法币相关第三方托管银行、投资人、借款人、监管机构等等。

原力协议体系内，法币背书型稳定币由超级节点负责发行。作为资金管理服务的提供方，超级节点与投资人签订投资协议，接收投资人的资金存入第三方托管银行。与此同时，超级节点在以太坊（EOS）网络内发行对应种类和数量的稳定币。超级节点接收的法币资金受到托管银行、审计机构的监督，其发行的稳定币数量也在区块链网络内公开透明，从根本上保证了原力协议体系内每一枚稳定币都有对应的法币作为支撑。由于能够从超级节点提供的资金管理服务获取长期稳定的收益，传统金融体系内的法币资金将大量涌入原力协议的借贷体系内，理论上，原力协议体系内的稳定币发行量可以无限接近全球法币发行的数量，这样的机制设计，可为加密数字资产领域带来巨大的增量资金。

投资人的投资协议到期后，若投资人选择赎回理财产品并提取法币，则超级节点将回收相应面值的稳定币并予以销毁，从而维持住币价的稳定。除了接受投资人的法币生成稳定币，超级节点还将提供小额零售兑换业务，支持法币和稳定币的双向兑换。同样，零售用户支付给超级节点的法币也将被存入第三方托管银行，并生成相应数额的稳定币，确保合规与安全，这将是原力协议体系内稳定币发行和销毁的另一种途径。

原力协议体系内的稳定币，将用于借款人的抵押贷款，也可用于加密经济体系内的流通。借款人在获得贷款后，可以向超级节点进行法币兑换，借款人也将在诸如交易所、加密支付等场景里直接使用稳定币，从而促进稳定币生态的成长。

超级节点的盈利模式包括：

- **订单管理费：**在典型的借贷订单里，借款人的成本包括借贷利息、智能合约执行燃料（gas）、订单管理费，其中订单管理费是超级节点的收入。
- **法币兑换手续费：**如果借款人需要把借入的稳定币兑换为法币，需要通过超级节点合规化操作，理论上而言，超级节点的兑换服务将会是原力

协议体系内的唯一法币转换渠道。

- **投资人理财计划管理费：**投资人旨在通过借出手里的法币或加密数字资产获得稳定的回报，超级节点可以给投资人提供 6 个月至长期的资金管理计划产品，这些产品获得的法币会被转换为稳定币，投资到原力协议体系内的借贷订单。并且享受原力协议提供的交易深度、快速成交和风险保障。资金管理计划产品代替投资人管理其资金，享受原力协议提供的交易深度、快速成交和风险保障，在借贷市场内有效运用募集到的资金，赚取稳健的投资回报。在原力协议内，所有的抵押物资产都由区块链确权和背书，比传统 P2P 理财更加安全、高效。

- **爆仓保险收入：**由于加密数字资产市场的高流动、高波动性，借款人的质押物有可能面临短时间内的价值剧烈下降，虽然原力协议设置了质押补仓机制以及紧急平仓机制，在极端情况下，系统仍可能无法及时进行风险控制操作，导致借贷订单的质押物爆仓，若此时质押物出售后的价值不能覆盖投资人的投入，则投资人将会遭受损失。原力协议体系为了应对极端情况的出现，会针对质押爆仓保险这一需求设计平台层面的解决方案，爆仓保险的参数会随着借贷订单数量的增加进行深入优化，更好的为投资人保驾护航，保险产品所产生的收益也将作为超级节点的一个重要收入来源。

2.2 投资人

超级节点所对接的投资人，包括但不限于：各国家、地区内持有大量法币或加密数字资产，并希望资产获得稳定增值的高净值人群、机构、长线资金、家族基金等。投资人乐于接受中长期的资金管理计划，资金风险偏好低，希望尽可能避免亏损，且投资风险可控。此类要求恰好能被区块链支持的借贷业务满足。在原力协议体系内，抵押物由智能合约所保障，具备极强的风险控制能力。

投资人可通过两种渠道投资获益——法币渠道、加密币渠道。超级节点资金端，只允许法币投入法币退出、加密数字资产投入加密数字资产退出。法币投资进入第三方托管银行后，将生成对应金额的稳定币，用于在原力协议体系内贷出

获益。投资人无法直接获得稳定币，对于投资人而言，超级节点提供的产品就是定期的资金管理计划，其旨在获取利息和本金增值。

2.3 借款人

借款人是原力协议体系的信贷来源，他们分布在全球范围，持有加密数字资产并希望在暂时让渡加密数字资产所有权的情况下获取法币或稳定币，可以是自然人或企业法人，甚至是 AI。通过抵押加密数字资产，借款人将获取资金，用于实体经济消费或投资，或用于加密数字资产市场投机。借款人的群体特征可以归纳为：认可加密数字资产生态，愿意长期持有加密数字资产；短期资金周转困难，看好加密数字资产市场上涨且不愿意卖币；寻求合理的资金借贷成本。

2.4 仲裁者

仲裁者是由原力协议社区投票选出的个人或团队，有一定时间的任期，期满后将进行改选。仲裁者的选举、任职机制最初由原力协议团队在适当时机提交到社区，讨论修改后经由社区投票正式执行。当超级节点的 FOR/EFOR 代币质押率不足时，监控智能合约将通知超级节点补充质押 FOR/EFOR。当超级节点未能在规定时间内补充 FOR/EFOR 时，智能合约会反馈相关信息给仲裁者，仲裁者将根据规则，判断超级节点的继续运作能力。如果仲裁者判断超级节点无法正常提供借贷服务，则会将判断结果提交到原力协议社区治理机构，申请从原力协议系统内剔除相应的超级节点。

2.5 多链加密资产质押的智能合约

基于对稳定币业务本质的理解，原力协议首次创新性的提出，基于多种跨链加密资产实现稳定币的抵押发行，为稳定币发行服务商提供开源框架，允许其自定义抵押资产种类、比例、治理机制，用市场竞争的思路选择最优化的跨链抵押资产组合；在这种模式下，稳定币供应商根据自身的风险控制能力、资产偏好、选择借贷抵押资产，根据原力协议提供的开发框架，开发抵押、风控、平仓等合约，为持有不同公链资产的用户提供去中心化质押资产借稳定币的服务；经过市

场竞争，在资产组合、风控、用户体验、治理方面最优秀的 DAPP 团队将会胜出，成为原力协议体系内的主要稳定币供应源之一，维护其发行稳定币的币值稳定。这一机制能够真正实现私人货币发行和流通的经济体系，在人类历史上首次创造出一个全球共享、非国家化的自由货币市场。

金融的核心是流动性，在传统金融体系里，很大比例的流动性来源于资产的质押，大量的信贷和资金支撑着经济的增长。原力协议要做的就是通过信贷和稳定币作为切入点，让已经存在的链上资产和将来要大量上链的资产，都能被用于质押借贷，生成流动性，进而打通支付、跨境清算和结算等功能，而且链上产生的稳定币还能通过法币连接系统，被重新投入到实体经济，这样就打通了链上和链下的流动性，相互促进经济发展。

2.6 其他参与方

参与原力协议体系稳定币发行工作的相关方还包括第三方托管银行、政府监管部门、审计机构等链外实体，它们的存在能够保障原力协议体系内的各方在合法、合规的情况下开展业务，维护系统的稳定。

3 技术细节

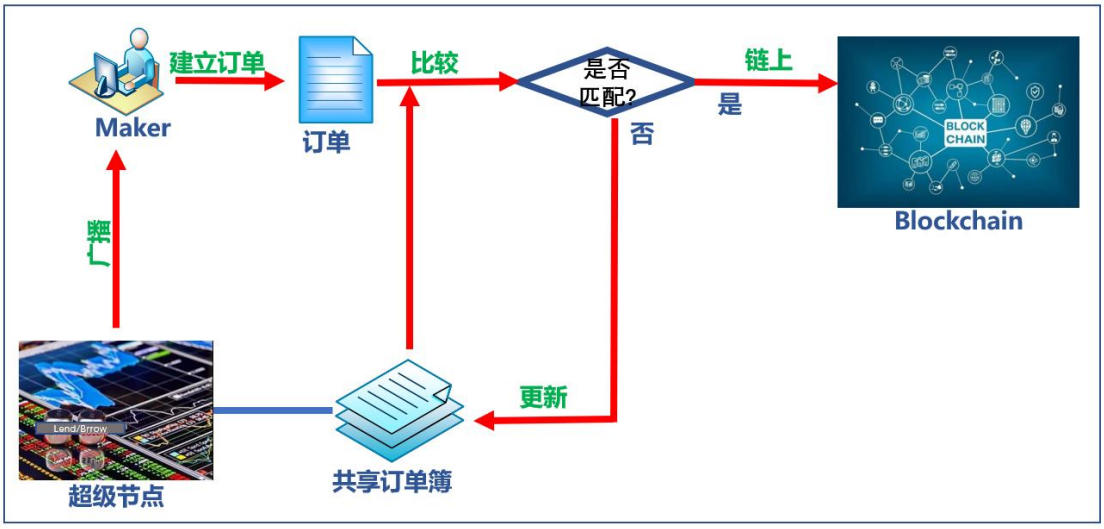
3.1 公开订单交易流程

随着市场流动性的大量涌现，必须存在一个公共的“场所”供借贷用户发布订单请求，这些请求需要快速的汇聚成订单簿，以便于目标用户能够发现此类请求。原力协议支持符合资质的法人或自然人快速建立借贷服务平台，共享借贷订单，并对通过本借贷平台发布请求的交易征收手续费。原力协议将上文中的“场所”、“平台”称为“**超级节点**”，而非传统金融领域的“借贷平台”称谓。超级节点服务于借贷参与方，提供订单信息的发布、自由流动、撮合等功能。超级节点不会偏袒交易的任何一方，因此能够得到交易参与各方的信任。确切地说，交易双方的利益不会受到超级节点的制约。

公共发布的订单并不指定交易另一方的账户地址，其允许订单自动撮合成交或者被用户人工选中并成交。公共发布的订单通常包含*feeLender*、*feeBorrower*、*addrSupernode*和*addrFeeRecipient*，分别表示借贷双方的交易费用、超级节点地址和接收费用的地址。原力协议将订单的发起者称为Maker，而亲自选择订单簿中已有订单成交的用户称为Taker。为方便描述，借贷行为中BTC、ETH等加密数字货币和加密代币统一称为“代币”或“币种”，不再特意区分。根据原力协议规则，借款人和出借人都可以发起订单或者主动选择订单成交。

前置工作：超级节点发布借贷规则，规则中包括抵押率等参数。同时公布收费标准、收费地址等内容。

3.1.1 订单提交和自动撮合



第一步：Maker授权智能合约转移特定数量借贷币种或抵押币种的权利。

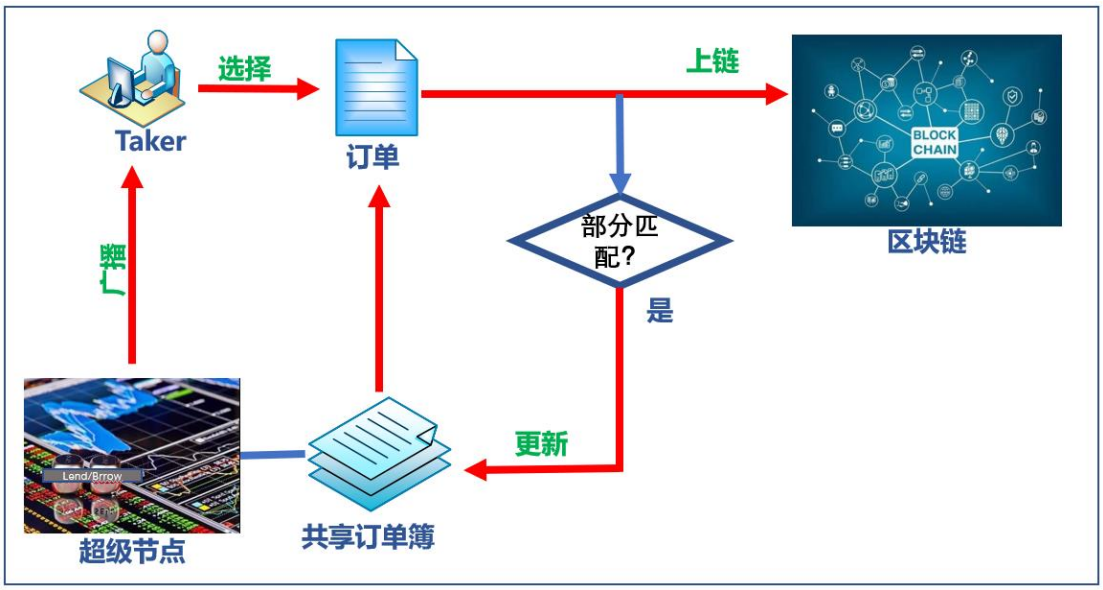
第二步：Maker创建订单，明确借贷币种及数量、抵押币种及数量、借贷周期、年化利率和订单有效期。设置完成后，超级节点自动生成用户需要的手续费及手续费收取地址。Maker用自己的私钥签署交易发送给超级节点。

第三步：超级节点接收到Maker签署的订单，验证订单的有效性（包括用户授权的地址中是否有足够的代币）。如果订单满足超级节点的要求，订单被接受并被标记为orderNew，否则超级节点拒绝接收订单。

第四步：超级节点遍历共享订单簿中的所有订单，验证每个订单对应的地址中有足够的代币并与订单orderNew比对。如果某订单对应的地址没有足够的代币，则判定该订单失效并从订单簿移除。对有效订单：（1）如果订单簿有多个订单匹配orderNew，锁定最早的订单；（2）如果存在部分匹配的情形，从匹配度最高的订单开始从高到低锁定，直到订单再无合适匹配订单；（3）如果无订单能够匹配，则将orderNew更新到订单簿中。将上述锁定的订单或匹配部分命名为orderMatch并从订单簿中移除，部分匹配情形中的未匹配部分也在成交后更新到订单簿中。此处匹配是指两个订单具有相同的借贷周期并且借款利率≥出借利率。

第五步：将OrderNew和orderMatch发送到智能合约以备下一步的上链交易。

3.1.2 用户自撮合



第一步：Taker登录任意的超级节点接收最新的共享订单簿；

第二步：Taker在订单簿中选择订单交易，被选中的订单称为orderMatch。新订单开始自动创建，并明确所需的代币数量、借贷周期、年化利率、订单有效期等。Taker授权智能合约具有转移指定数量数字资产的权利，并签名整个交易，命名为orderNew。

第三步：超级节点收到orderNew和orderMatch后，验证orderMatch和orderNew的有效性。如果两者都满足超级节点的要求，orderMatch被从共享订单簿中移除。否则，orderNew和orderMatch都将被拒绝。

第四步：超级节点将orderNew和orderMatch发送到智能合约进行交易。

3.1.3 交易上链

第一步：创建智能合约的实例，参数由上述orderNew和orderMatch确定；

第二步：智能合约锁定借款人的抵押代币；

第三步：智能合约将出借人地址中的借贷币种转移到借款人地址；

第四步：合约正式生效。

3.1.4 到期结算

当借贷合约到期，将启动还款程序，过程如下：

第一步：基于原力协议的超级节点通过短信、邮件、应用系统站内消息和电话等方式通知用户订单到期；

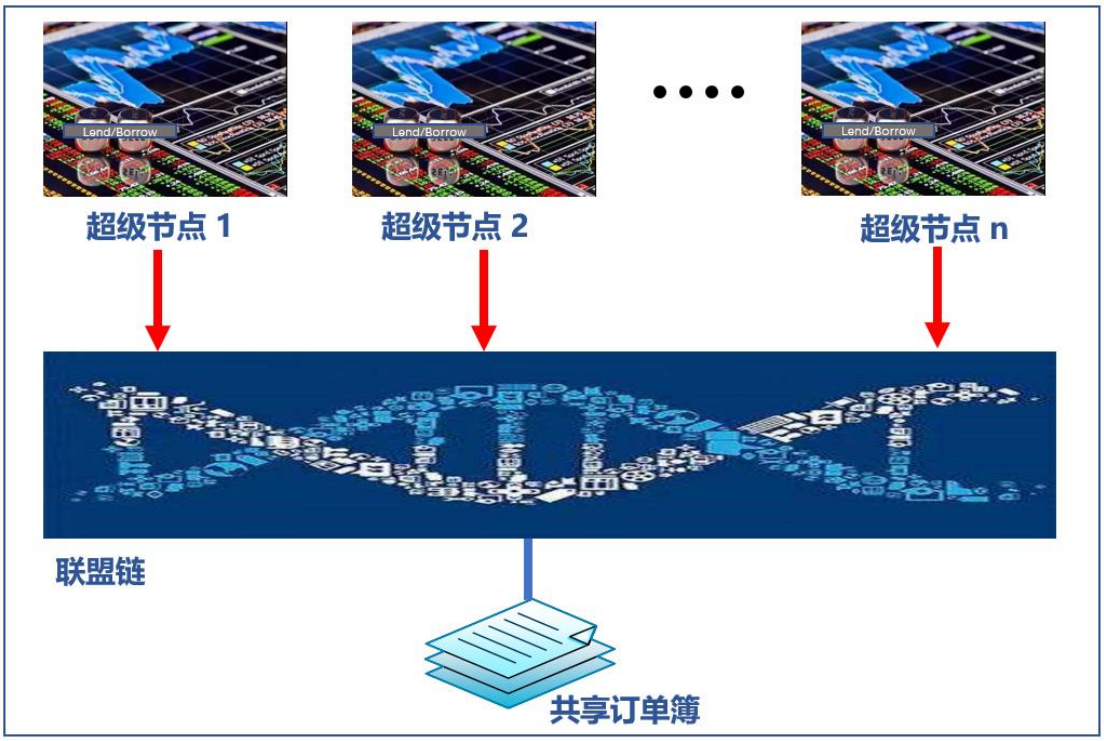
第二步：借款人将应还本金和利息发送给智能合约；

第三步：智能合约将得到的代币和利息发送给出借人；

第四步：智能合约解锁借款人的抵押代币。整个交易结束。

3.1.5 共享订单簿

原力协议共享订单簿目前通过以太坊（EOS）的侧链（联盟链）来实现，未来将通过原力协议具备跨链能力的公链实现。全球基于原力协议的所有超级节点共同维护共享订单簿。



在原力协议联盟链中，每个超级节点都有一个账号名为validator。每当有新的超级节点加入原力协议，都将有对应的validator加入到共享的validator列表。智能合约设定validator加入的条件和策略。在联盟链创始区块中设置时间戳，每3秒钟选择一个法定validator建立、签名和广播一个区块。法定validator索引通过如下公式确定：

index= (UNIX_TIMESTAMP/BLOCK_TIME) % NUMBER_OF_ VALIDATORS;

当订单更新事件发生时，超级节点中的本地validator将检查指令的有效性。如果所涉及的订单已经被锁定，则更新指令将被拒绝。否则，本地validator将广播订单并交由法定validator打包。如果多个指令锁定同一个订单，则接受最早指令的请求。所有指令和结果订单簿状态被法定validator打包进区块。如果51%以上validator接受区块，它将被所有的validator确认。如果在一段时间内没有订单更新，则联盟链暂停生产打包区块。

3.2 点对点（P2P）订单交易流程

3.2.1 借贷流程

场外交易订单允许借贷双方利用任意媒介撮合交易。原力协议在此类交易中的作用是给交易做见证，并防止违约发生。具体流程如下：

前期工作：借款人和出借人通过互联网等媒体撮合交易。其中一方作为 Maker 创建订单并根据原力协议的要求设置所有的参数，然后签署订单并提交到智能合约。

第一步：借款人 A 授权智能合约锁定指定数量的代币 Token A 作为抵押；

第二步：出借人 B 发送指定数量的 Token B 到智能合约；

第三步：智能合约将接收到的 Token B 发送到借款人 A；

第四步：智能合约开始为合约的有效性提供证明。

3.2.2 还款流程

当订单到期，任何交易方可触发还款流程，过程如下：

第一步：出借人 B 通知借款人 A 还款；

第二步：借款人将借入的 Token B 和智能合约中约定的利息发送给智能合约；

第三步：智能合约将第二步中得到的代币发送给出借人 B；

第四步：智能合约解锁借款人被锁定的抵押代币 Token A。

3.3 借贷订单参数设置

3.3.1 利率发现机制

原力协议生态下的借贷利率将由市场供需和借贷双方共同决定，而每一笔借款/贷款订单也会影响市场供需。即借贷利率完全由超级节点根据市场给出参考值，最终由借款人和出借人决定。此处所描述的市场利率均为年化利率，其余周期下的利率也将转化为年化利率。订单薄将按照利率高低进行排名，当借款订单利率大于等于出借订单利率，则撮合成交。

为防止恶意挂单影响利率，系统会对订单做有效性判别，反复挂单、取消订单的用户账户，会在一段时间内被限制使用。此外，系统还将根据市场浮动情况限定利率变化范围，如每 10 分钟内成交订单利率变化不得超过 20%。更加细致的计算规则和使用条件会在具体业务开发过程中进行调查确认。

3.3.2 借贷周期

借贷周期由超级节点和交易者共同确定，即超级节点提供一系列的借贷周期选项供用户选择，最终由订单创建者确定周期。Maker 发起的订单在共享订单簿中更新，Taker 将选择最合适的订单成交。

3.3.3 抵押率和强制平仓线

为控制借贷风险，原力协议最初由团队确定抵押代币类型和抵押率、强制平仓线设定规则。所有这些内容也将存储在智能合约中，只有全部核心成员同意才能对内容和规则进行更改。当原力协议社区成熟后，修改提议将被提交到去中心化的自组织投票确定。

抵押资产风险管理包括抵押代币的选择和抵押率设计。原力协议 1.0 版本将加密数字资产代币流通市值和 24 小时成交额等因素作为判断抵押代币的依据。后期将根据 DAPP 运营数据，不断更新和扩展可抵押代币种类。届时，自治组织

也将发起社区投票，让用户选择其所希望的抵押代币种类。

原力协议抵押率定义为可借资金与抵押物现值的比例。传统借贷机构规定抵押率不得超过 70%，加密数字资产借贷领域并没有明确规定和历史经验可供参考。考虑到加密数字资产历史价格变动幅度较大，现有各抵押平台抵押率设定在50%-80%不等。由于具体币种的历史价格波动幅度和 24 小时主流交易所成交额会有微小差异，我们将根据具体行情设置抵押率，具体如下：

A_n=向前第 n 个周期的振幅（%）；

$A_{predict}=A_1+Max（0，A_2-A_1）/2^{(2-1)}+Max（0，A_3-Max（A_1，A_2））/2^{(3-1)}+...+Max（0，A_n-Max（A_1，A_2，...，A_{n-1}）））/2^{(n-1)}$；其中 n 通常取 3。

抵押率=$Min（70\%，1-A_{predict}）$。

3.4 预言机：安全保障机制

在合约有效期内，超级节点通过网络爬虫实时抓取主流交易所列表中加密数字资产的价格信息，对智能合约涉及到的代币价格进行核算。如果价格波动触及预警线，则调用预言机从上述网站抓取数据作为证据。然后超级节点将预警信息发送给借款人并提醒他们补充抵押代币。预警信息将通过短信平台或者自动语音电话发送，并存入区块链做备份。如果借款人没有及时补充抵押代币，当强制平仓线抵达后将启动强制平仓程序，具体流程如下：

第一步：调用预言机程序，从主流交易所网站上抓取当时价格为平仓提供证明文件并存储在区块链上；

第二步：智能合约计算应还本金和利息，扣除等值抵押资产，并将剩余部分解锁还给借款人；

第三步：智能合约将第二步所扣除抵押资产发送给出借人。

预言机调用智能合约代码片段如下：

```
/* 智能合约构造函数中与预言机相关的代码片段*/
contract VerifyContract is usingOraclize
{
    …
    string public urlDataSource;
    constructor(…, urlDataSource) public
    {
          …
          oraclize_setProof(proofType_TLSNotary | proofStorage_IPFS);
    }
        …
    function onOraclizeVerification()
    {
          …
          Oraclize_Query("URL", urlDataSource);
          …
    }
    …
    function _callback(bytes32 myid, string result, bytes proof){
          if(msg.sender !=oraclize_cbAddress()); //只有授权用户才有权操作
          …
    }
}
```

原力协议团队将持续关注预言机技术在加密数字货币领域的进展，近期已有诸如 DOS Network 等优质预言机项目上线，团队将根据技术的最新发展，更新原力协议的预言机代码。

3.5 保险：应急管理

(1) 应急管理

因加密数字资产市场价格波动较大，极端情况必须被提前考虑。在抵押代币的价格在短时间内迅速跌破平仓线，而智能合约未能及时平仓的情况下，智能合约将不再发送预警信息直接启动平仓程序。

(2) 保险选择

在上述极端情况下，即使智能合约尽快平仓，出借人还会遭受一定程度的损失。有时候，这些损失可能会超出出借人的承受能力，这也是阻碍借贷市场发展的一种常见因素。基于这种情况，原力协议将设计保险产品。当上述极端情况发生时，购买保险的用户可以从保险资金池中获得赔偿。关于保险产品的更多细节将在原力协议的后续版本中详细展示。

(3) 质押代币选择

如果某种类型的代币价格长期剧烈波动，原力协议团队（最终将由 DAO 组织投票）有权将特定类型的加密数字资产从抵押物清单中移除，甚至暂时性的关闭借贷交易。由于建立在以太坊和 EOS 之上（未来将扩展到 RSK、TRON、IOST、NEO、Cybermiles 等主要公链），原力协议安全性和应用水平将很大程度上取决于以太坊和 EOS 等底层公链的发展。从长远角度，原力协议团队将保持对区块链底层技术的关注。加密金融服务市场的发展最终将取决于整个加密数字资产生态的持续和健康发展。

4 信息规范

为使得不同超级节点订单能够共享，原力协议确定了订单标准，这些标准设定了所有的参数和相关签名。通过 Keccak SHA3 算法，所有的参数被映射到 32 字节长度，然后应用 ECDSA 创造签名。成交订单的最终格式将被存储在智能合约变量中。公共订单主要由超级节点以共享订单簿的方式保存和广播，超级节点将收取部分费用。

4.1 公共订单

由于订单的用途不同，超级节点最初建立的订单与区块链智能合约中的订单格式略有不同。下面是存储在超级节点中的订单格式，而智能合约中的订单格式将在智能合约部分展示。

变量名	数据类型	描述
addrVersion	address	智能合约地址，升级后更新
nFeeLender	uint256	出借人手续费
nFeeBorrower	uint256	借款人手续费
addrFeeRecipient	address	手续费收取地址
addrBorrower	address	借款人地址
addrLender	address	出借人地址
addrTokenA	address	质押币 Token A 地址
addrTokenB	address	借贷币 Token B 地址
nTokenA	uint256	质押币 Token A 数量
nTokenB	uint256	借贷币 Token B 数量
nExpiration	uint256	订单有效期
nLendingCycle	uint256	借贷周期
ufMortageRate	ufixed0x256	抵押率

变量名	数据类型	描述
ufInterestRate	ufixed0x256	年化利率
vLender	uint8	出借人对上述参数的 ECDSA 签名
rLender	bytes32	
sLender	bytes32	
vBorrower	uint8	借款人对上述参数的 ECDSA 签名
rBorrower	bytes32	
sBorrower	bytes32	

4.2 点对点订单

变量名	数据类型	描述
addrVersion	address	智能合约地址，并在每次升级后更新
addrBorrower	address	借款人地址.
addrLender	address	出借人地址
addrTokenA	address	抵押代币 A 的地址
addrTokenB	address	出借代币 B 的地址
nNumTokenA	uint256	抵押代币 A 的数量
nNumTokenB	uint256	出借代币 B 的数量
nExpiration	uint256	订单有效期
nLendingCycle	uint256	借贷周期
ufRate	ufixed0x256	抵押率
ufInterestRate	ufixed0x256	利率
vLender	uint8	出借人 ECDSA 签名
rLender	bytes32	
sLender	bytes32	

变量名	数据类型	描述
vBorrower	uint8	借款人 ECDSA 签名
rBorrower	bytes32	
sBorrower	bytes32	

5 智能合约

原力协议开发了一系列的智能合约以覆盖所有的功能。许多经常被调用的基础合约被作为库合约，以减少合约的重新部署对 gas 的消耗。

5.1 相关库

通常来说，原力协议中大部分的数学运算都是在链下执行，而应用预言机中的少量运算也只是为结果提供证明。尽管如此，基本的数学库还是被列为库文件，既为了里面不时涉及的计算，主要也是为扩展做准备。空间所限，此处包括其他合约仅选择部分代码段作为展示之用。

5.1.1 基础数学库 MathLib.sol

```
pragma solidity ^0.4.24;

library MathLib {

    function multiply(uint256 a, uint256 b) pure internal returns (uint256) {
        uint256 c = a * b;
        assert(a == 0 || c / a == b);
        return c;
    }

      … //other common functions

    function add(uint256 a, uint256 b) pure internal returns (uint256) {
        uint256 c = a + b;
        assert(c >= a);
        return c;
    }
}
```

5.1.2 字符串库 StringLib.sol

```
pragma solidity ^0.4.24;

library StringLib {
    struct slice {
        uint _len;
        uint _ptr;
    }

    function memcpy(uint dest, uint src, uint len) private pure {
        for(; len >= 32; len -= 32) {
            assembly {
                mstore(dest, mload(src))
            }
            dest += 32;
            src += 32;
        }
        uint mask = 256 ** (32 - len) - 1;
        assembly {
            let srcpart := and(mload(src), not(mask))
            let destpart := and(mload(dest), mask)
            mstore(dest, or(destpart, srcpart))
        }
    }

    ...             //other functions

    function toString(slice memory self) internal pure returns (string memory) {
        string memory ret = new string(self._len);
        uint retptr;
        assembly { retptr := add(ret, 32) }
        memcpy(retptr, self._ptr, self._len);
        return ret;
    }
}
```

5.1.3 订单库 OrderLib.sol

```
pragma solidity ^0.4.24;

library OrderLib {

    struct LendingOrder {
        address      addrContractVersion;
        address      addrSupernode;
        address      addrBorrowerer;
        address      addrLender;
        address      addrCollateralContract;
        address      addrCollateralBorrowerer;
        ufixed0x256  ufMortageRate;
        ufixed0x256  ufInterestRate;
        uint256      nTokenLend;
        uint256      nTokenMortage;
        uint256      nExpiration;
        uint256      nLendingCycle;
        …
        uint8        vLender;
        bytes32      rLender;
        bytes32      sLender;
        uint8        vBorrower;
        bytes32      rBorrower;
        bytes32      sBorrower
    }

}
```

5.2 原力协议智能合约 TheForceProtocol.sol

TheForceProtocol 是整合交易和风险管理所有功能的主合约。当一个交易被提交给智能合约的时候，都将创建主合约的一个实例并存储在以太坊（EOS）区块链上。智能合约实例将调用合约中近十余个内外部函数、库函数、子合约以实现智能合约的功能。这些函数包括 *LockCollateralFromBorrower()*、*SendTokenToBorrower()*、*RepayTokenToLender()*、*UnlockCollateralToBorrower()*、*WithdrawLend ()*、*WithdrawBorrow ()*等。需要注意的是 *WithdrawLend ()*和

*WithdrawBorrow ()*必须由出借人和借款人发起，并且得到交易对口方的统一签名才可以执行。限定函数调用发起人，可以利用修饰符 *onlyLender()*和 *onlyBorrower()* 实现。空间所限，后面将仅列出部分代码，细节内容将在不久的未来展示在原力协议开源社区并接受更多开发者的审核。

```
pragma solidity ^0.4.24;

…
import “./OrderLib.sol”;
import “./oraclizeAPI.sol”;

Contract TheForceProtocol {
      …
      Address     addrCreator;
      LendOrder  _LendOrder;

          …
      constructor (
              address[3]    arrayBaseAddresses,
              uint256[5]    nContractSpecs,
              …
               Bytes32[4]   strContractSpecs
              ) public
      {
          addrCreator=msg.sender;
          _LendOrder. addrContractVersion= arrayBaseAddresses[0];
          _LendOrder. addrBorrowerer= arrayBaseAddresses[1];
          … //other parameters
          _LendOrder.sBorrower= strContractSpecs[3];

      }

      modifier onlyLender() {
                require(msg.sender == _LendOrder. addrBorrowerer);
                _;
            }
      function isValidSignature(
            address signerAddress,
            bytes32 hash,
            uint8 v,
            bytes32 r,
            bytes32 s
      ) public pure returns (bool) {
            return   signerAddress   ==   ecrecover(StringLib.Encode(hash),   v,
                                          r, s );
      }

      …
}
```

5.3 预言机接口合约 OraclizeAPI.sol

原力协议中预言机主要用来为后面合约采取的平仓、预警等措施和行为提供证明。因此，预言机中的函数使用不频繁，也就不需要专门设定独立的合约调用预言机智能合约。所有的预言机函数都可以通过预言机接口 API 提供，在合约中仅仅需要几行代码就可以调用预言机函数并达到相应的效果。此处给出几行代码展示预言机接口的结构。

```
pragma solidity ^0.4.24;

…
contract OraclizeI {
        address public cbAddress;
        function query(uint _timestamp, string _datasource, string _arg) external
                payable returns (bytes32 _id);
        function query_withGasLimit(uint _timestamp, string _datasource,
                string _arg, uint _gaslimit) external payable returns (bytes32 _id);
        function getPrice(string _datasource) public returns (uint _dsprice);
        function getPrice(string _datasource, uint gaslimit) public
                returns (uint _dsprice);
        function setProofType(byte _proofType) external;
        function setCustomGasPrice(uint _gasPrice) external;
}
        contract OraclizeAddrResolverI {
                function getAddress() public returns (address _addr);
        }
        contract usingOraclize {
                byte constant proofType_TLSNotary = 0x**;
                …//other status parameters
                modifier oraclizeAPI {…}
                modifier coupon(string code){…}
                function oraclize_setNetwork(uint8 networkID) internal
returns(bool){…}
                function oraclize_setNetwork() internal returns(bool){…}
        function _callback(bytes32 myid, string result) public {…}

        …//other functions
}
    …//other APIs
```

5.4 智能合约注册合约 TheForceProtocolContractRegistry.sol

为保障每个交易对应的智能合约实例顺利执行，原力协议设计了注册智能合约。当合约更新或者废弃时，需要注册合约设定更新规则及对应的智能合约白名单。在开始阶段，合约中白名单的确定更多带有中心化的成分。后面随着原力协议社区的成熟，此类事务将由社区通过去中心化的治理方式解决。

```
pragma solidity ^0.4.24;

contract TheForceProtocolContractRegistryInterface {
      function addAddressToWhiteList(address contractAddress) external;
      function isAddressWhiteListed(address contractAddress) external view
                returns (bool);
}
contract MarketContractRegistry {
      mapping(address => bool) public isWhiteListed;
      address[] public addressWhiteList;
      event AddrAddedToWhitelist(address indexed contractAddress);
      event AddrRemovedFromWhitelist(address indexed contractAddress);
      function isAddressWhiteListed(address contractAddress) external view
                returns (bool) { return isWhiteListed[contractAddress]; }
      function getAddressWhiteList() external view returns (address[]) {
                return addressWhiteList;
      }
      function removeContractFromWhiteList(address contractAddress,
                 uint whiteListIndex) external onlyOwner returns (bool) {
            require(isWhiteListed[contractAddress]);
            require(addressWhiteList[whiteListIndex] == contractAddress);
            isWhiteListed[contractAddress] = false;
            …//other code
            emit AddrRemovedFromWhitelist(contractAddress);
      }
      function addAddressToWhiteList(address contractAddress) external {
            require(msg.sender == owner);
            require(!isWhiteListed[contractAddress]);
            isWhiteListed[contractAddress] = true;
            addressWhiteList.push(contractAddress);
            …//other code
            emit AddrAddedToWhitelist(contractAddress);
      }
}
…//other contracts
```

5.5 其他智能合约

除上述描述的合约，还有大量其他通用类型智能合约在原力协议的执行过程中发挥着重大作用。虽然此类合约不为原力协议所独有，即在很多其他平台的智

能合约中都有出现，但其发挥的作用却不容小觑。此处选型典型的几个通用合约的代码作为展示之用。

5.5.1 创建功能接口 CreatorAPI.sol

```
pragma solidity ^0.4.24;

contract CreatorAPI {
	address public creator;
	constructor () public {
		creator = msg.sender;
	}
	event CreatorChanged (address indexed currentCreator, address indexed newCreator);
	function ChangeCreator(address newCreator) onlyCreator public {
		require(newCreator != address(0));
		emit CreatorChanged (creator, newCreator);
		creator = newCreator;
	}
	modifier onlyCreator() {
		require(msg.sender == creator);
		_;
	}
}
```

5.5.2 链接功能合约 Linkable.sol

```
pragma solidity ^0.4.24;

contract Linkable {
	address public linkedAddr;
	constructor (address addrToLink) public {
		require (addrToLink != address(0));
		linkedAddr = addrToLink;
	}
	modifier onlyLinked() {
		require(msg.sender == linkedAddr);
		_;
	}
}
```

5.5.3 部署功能合约 Migrations.sol

```
pragma solidity ^0.4.24;

contract Migrations {
    address public owner;
    uint256 public last _migration;

    modifier onlyOwner() {
        if (msg.sender == owner) _;
    }

    constructor() public {
        owner = msg.sender;
    }

    function setCompleted(uint completed) public restricted {
        last_migration = completed;
    }

    function update (address new_address) public restricted {
        Migrations update = Migrations(new_address);
        update.setCompleted(last_migration);
    }
}
```

5.6 系统安全

粉尘攻击

在系统运行过程中，可能存在恶意用户广播大量尘埃订单的情况。在进行交易撮合的过程中，系统将根据撮合交易的规则抛弃此类订单，减少订单簿压力，使得尘埃订单不会对系统有任何影响。

订单数据篡改

生成订单簿的过程中将涉及网络安全问题，在数据传输的过程中采用加密的方式进行数据传输，避免网络数据被截取，防止数据被篡改。

网络拥堵问题

信息量过大不加以限制，超额的网络流量就会导致设备反应缓慢，造成网络延迟。在提交智能合约的过程中，采用消息队列的方式，计算带宽速度，计算并发量，避免造成网络不必要的拥堵。

5.7 合约风险控制

借贷合约全局风控参数设置如下表：

参数	规则
抵押代币选择	代币历史价格波动方差，24 小时成交额，上市主流交易所数量
抵押率	借款周期内价格波动幅度，当前代币价格，置信水平
借款锚定币种	BTC、ETH（EOS）、USDT（EUSD）
借款周期	7 天、14 天、1 个月、2 个月、3 个月、6 个月、9 个月、12 个月
平仓预警通知	抵押代币总价值下降到应还本息的 120%，发出通知后 24 小时补充抵押资产
平仓	抵押代币总价值下降到应还本息的 105%
还款通知	在借款合约应还时间前 24 小时通知
逾期未还	平台垫付借出方应收本息，卖出抵押代币
逾期未还惩罚	收取应还本息的 5%违约金

6 API 接口

原力协议通过 TheForceProtocol.js 接口文件为超级节点提供服务。具体来说，原力协议提供了一个接口库，当用户创建超级节点或者进行个人点对点交易的时候，仅需调用相应的接口。随着原力协议的成熟，其将提供越来越好用的接口供用户调用。限于空间，此处仅展示几个接口给读者一个直观的印象。后续代码的内容描述及更新，将在代码经过几轮严格的测试和审核后发布到原力协议开源社区。

6.1 超级节点 API

原力协议为机构用户提供了多种类型的 API 以创建超级节点，这些 API 的功能涵盖构建许可链、构建超级节点、更新共享订单簿、订单提交、稳定币生成、跨链交易（与 BTC、EOS 等）等内容。

```
/*****************************************************************/
/* MakeLendOrder() 接口可被出借人在超级节点上创建订单时使用*/
/*****************************************************************/
TheForceProtocol.ETH.MakeLendOrder(
    address       addrVersion,
    address       addrLender,
    address       addrSuperNode,
    bytes32       strTokenLend,
    uint256       nTokenLend,
    uint256       nTokenCollateral,
    uint256       nExpiration,
    uint256       nLendingCycle,
    ufixed0x256   ufRate,
    ufixed0x256   ufInterestRate,
    uint8              vLender,
    bytes32       rLender,
    bytes32       sLender
)
```

```
/*****************************************************************/
/* MakeBorrowOrder() 接口可被借款人在超级节点上创建订单时使用*/
/*****************************************************************/
TheForceProtocol.ETH.MakeBorrowOrder (
    address       addrVersion,
    address       addrBorrower,
    address       addrSuperNode,
    bytes32       strTokenBorrow,
    uint256       nTokenBorrow,
    uint256       nTokenCollateral,
    uint256       nExpiration,
    uint256       nLendingCycle,
    ufixed0x256   ufRate,
    ufixed0x256   ufInterestRate,
    uint8              vBorrower   ,
    bytes32       rBorrower,
    bytes32       sBorrower
)
```

```
/*****************************************************************/
/*MakeSuperNode()接口用来创建超级节点，包括加入的步骤*/
/*****************************************************************/
TheForceProtocol.ETH.MakeSuperNode (
    address                 addrVersion,
    address                 addrSuperNode,
    map(address=>uint)      addrReceipt,
    bytes32[]               strReceipt,
    …                       //还需要有大量的证明材料的数字化证明.
)
```

6.2 点对点个人交易 API

下面是用于个人用户点对点交易的 API。此类 API 可以用平台 DAPP 包装后以界面形式调用，也可以直接利用命令行的形式直接调用。

```
TheForceProtocol.P2P.lendToken(
    address      addrVersion,
    address      addrLender,
    address      addrBorrower,
    address      addrLender,
    bytes32      strTokenLend,
    bytes32      strTokenCollateral,
    uint256      nTokenLend,
    uint256      nNumTokenCollateral,
    uint256      nExpiration,
    uint256      nLendingCycle,
    ufixed0x256  ufRate,
    ufixed0x256  ufInterestRate,
    uint8        vLender,
    bytes32      rLender,
    bytes32      sLender
)
```

```
TheForceProtocol.P2P.BorrowToken(
    address      addrVersion,
    address      addrBorrower,
    address      addrLender,
    bytes32      strTokenLend,
    bytes32      strTokenCollateral,
    uint256      nTokenLend,
    uint256      nNumTokenCollateral,
    uint256      nExpiration,
    uint256      nLendingCycle,
    ufixed0x256  ufRate,
    ufixed0x256  ufInterestRate,
    uint8            vBorrower,
    bytes32      rBorrower,
    bytes32      sBorrower
)
```

7 币币贷 2.0：展示平台 DAPP

本部分将展示基于原力协议构建的 DAPP，从用户使用的角度描述此类 DAPP 能够实现的功能。预计该平台将作为原力协议团队现有项目币币贷借贷平台的升级版，称为币币贷 2.0 版。使用币币贷 2.0 平台，需要用户安装谷歌浏览器 Chrome 和 MetaMask（Scatter）插件。MetaMask（Scatter）插件使用户不安装以太坊（EOS）全节点即可进行相应的借贷操作。

需注意，下述任何有关币币贷 2.0 平台的界面、操作流程、步骤都仅用于展示，不代表最终产品的形态，原力协议团队将依据实际情况和最新的项目研发进展进行后续开发。

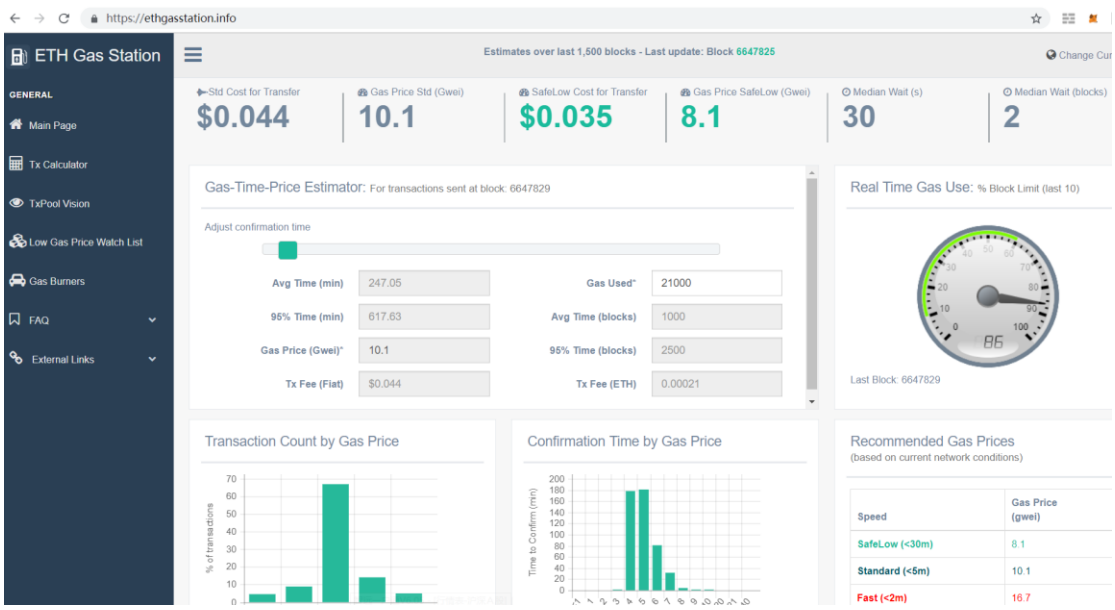
7.1 准备工作

本部分内容请参考附录部分。

7.2 DAPP 应用展示

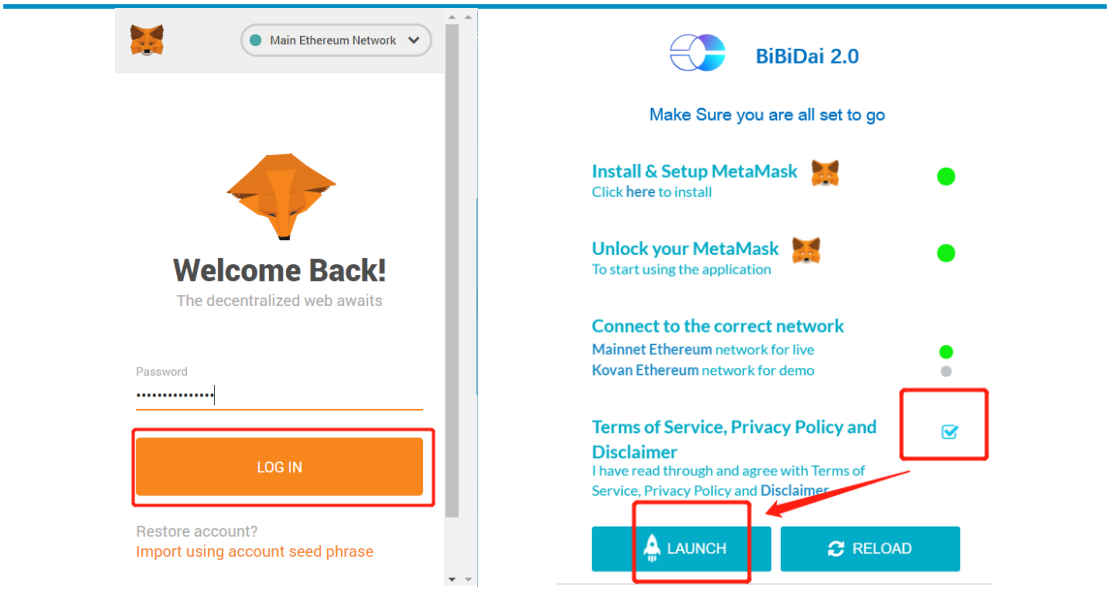
本部分主要用于展示基于原力协议构建的 DAPP 使用方式。由于每个超级节点提供的服务存在差异性，其最终使用的 DAPP 也与此处展示的样式、内容存在些许差异。

在借贷之前，用户可以在下述网站查询当前环境下以太坊交易的 gas 价格：https://ethgasstation.info.



下面是实例 DAPP 的界面，用户可以选择借款、投资和购买理财产品选项，开启借款、出借和购买理财产品流程。

点击上述菜单中的任何一项，将跳出如下的登录界面，要求用户输入账户密码。



此时将出现另一个弹窗。选择上图的复选框表示同意服务条款和隐私保护和相关声明，然后选择登录。

7.2.1 创建借款订单

点击“创建借款订单”菜单，将将显示下面的“创建借款订单”页面。

选择借入代币类型、抵押代币类型、借贷周期、年化利率、还贷形式。然后输入借入代币的数量，超级节点将自动计算抵押代币的数量和交易手续费。点击锁定抵押代币按钮，前往钱包授权智能合约能够锁定相应数量的代币。选择同意

合约条款和已经锁定代币选项，点击下一步完成借贷订单。

7.2.2 创建出借订单

点击创建出借订单菜单，页面将展示创建出借订单页面。

选择出借代币类型、贷款周期、年化利率和还款形式，然后输入出借代币的数量，此时系统将自动显示需要缴纳的手续费。点击锁定出借代币超链接，进入钱包授权智能合约能够转移指定数量的代币。最后选中同意借贷协议和确认已经锁定出借代币两个复选框，点击下一步创建出借订单。

7.2.3 浏览所有的借款订单

点击浏览所有借款订单菜单，将出现浏览所有借款订单页面。

用户可查询所有订单，并可在上方筛选器中选择查询条件。找到合适的订单请求后，用户可以点击“出借”超链接，此时将跳出填写订单资料页面。

在该页面填写出借代币的数量，平台将自动计算所需的手续费。点击锁定出借代币前往钱包，授权智能合约转移指定数量代币的权利。选中代币已锁定和同意借贷协议复选框，点击提交完成交易。

7.2.4 浏览所有出借订单

点击浏览所有的出借订单菜单，将显示浏览所有出借订单页面。

用户可查询所有订单，并可在上方筛选器中选择查询条件。找到合适的订单请求后，用户可以点击“借入”超链接，此时将跳出填写订单资料页面。

填写借入代币的数量，平台将自动计算手续费和需要抵押代币的数量。点击锁定出借代币前往钱包，授权智能合约锁定指定数量抵押代币的权利。选中抵押代币已锁定和同意借贷协议复选框，点击提交完成交易。

7.2.5 购买理财产品

点击购买理财产品菜单，将出现购买理财产品页面。

用户可以浏览所有现有的理财产品，发现合适的产品可以点击理财产品最后一列的购买按钮。点击购买按钮后，页面将跳转到银行付费页面，并将在购买完成后弹出购买成功信息框。

理财产品到期后，系统将把资金连同利息发送到用户的原始账户，并通过用户预留的联系方式通知用户。

8 原力协议原生代币

原力协议将发行基于 Etherum 的 ERC-20 代币，符号为 FOR；以及基于 EOS 的代币，符号为 EFOR。未来，在原力协议的公链上线后，FOR 和 EFOR 代币将与原力协议主链上的功能代币进行互换（swap）。在原力协议生态系统内，FOR（EFOR）代币将发挥重要作用，由于两种代币在各自的公链生态中起到了相同的功能，故下文将会把两种代币放到一起做阐述。

8.1 代币用途

FOR（EFOR）代币不仅可以有效促进生态系统的运行，而且可以作为去中心化组织自治的载体。在原力协议生态系统中，FOR（EFOR）代币将发挥以下作用:

8.1.1 交易手续费抵扣

在原力协议体系内，当借贷订单匹配并进行撮合时，智能合约将扣除借贷双方少量的挂单代币，分别发送给提交双方订单的超级节点，作为超级节点的服务费收入。常规情况下，手续费为 0.5%，双向收取。当用户持有 FOR（EFOR）代币时，智能合约将根据用户的持币量，计算出手续费的优惠额度，然后扣除用户的 FOR（EFOR）支付手续费。为了防止超级节点在收取 FOR（EFOR）手续费后向市场集中抛售造成币值下降，原力协议系统对每一笔以手续费形式获取的 FOR（EFOR）都会设置冻结期，待冻结期结束后，超级节点才会获得手续费 FOR（EFOR）的收入，以此避免超级节点集中抛售 FOR（EFOR），稳定原力协议生态。

8.1.2 超级节点质押锁仓

在原力协议体系内，每个超级节点在上线时都需要质押一定数量的 FOR（EFOR）代币，这部分代币将由专门的智能合约进行托管。智能合约还将定期

扫描超级节点 FOR（EFOR）代币的质押水平，若质押量低于系统要求的最小值，超级节点将接到补充质押资产通知。如果该节点未在规定时间内补充质押 FOR（EFOR），则系统将根据预设条件将信息提交给仲裁者，判断节点是否能正常履行功能，若判断结果为负，则仲裁者会向社区治理体系提交删除该超级节点的提案。

为了防止超级节点提交恶意订单甚至破坏原力协议网络的稳定性，若超级节点向网络提交了大量的无效订单或者恶意订单，则系统将根据预设条件判定超级节点作恶，相应的信息将被提交给仲裁者，判断节点是否确实执行了危害整个原力协议网络的行为，若判断结果为真实，则仲裁者会向社区治理体系提交扣除该超级节点质押 token 的提案，并对受影响的投资者和用户进行赔偿。

8.1.3 社区治理

FOR（EFOR）是原力协议社区成员参与社区投票的唯一工具。首先，当有任何重要事项需要提交社区治理委员会讨论时，提议者必须持有 FOR（EFOR），在递交提案时需要向专门的智能合约抵押一定数量的 FOR（EFOR）以后才能将提案提交到社区讨论版面。社区持币人可以在一定时间内就提案的内容提交修改建议，所有的改动都会形成迭代版本并被区块链记录。在规定时限结束后，FOR（EFOR）的持币人将对提案内容进行投票，所有参与了锁仓的 FOR（EFOR）都将不被计算在票仓内，不同的提案需要满足具体的票数要求才能获得通过。所有用于投票的 FOR（EFOR）都将被智能合约锁定一定时期，在一段时间内暂时退出流通体系。

8.1.4 超级节点挖矿

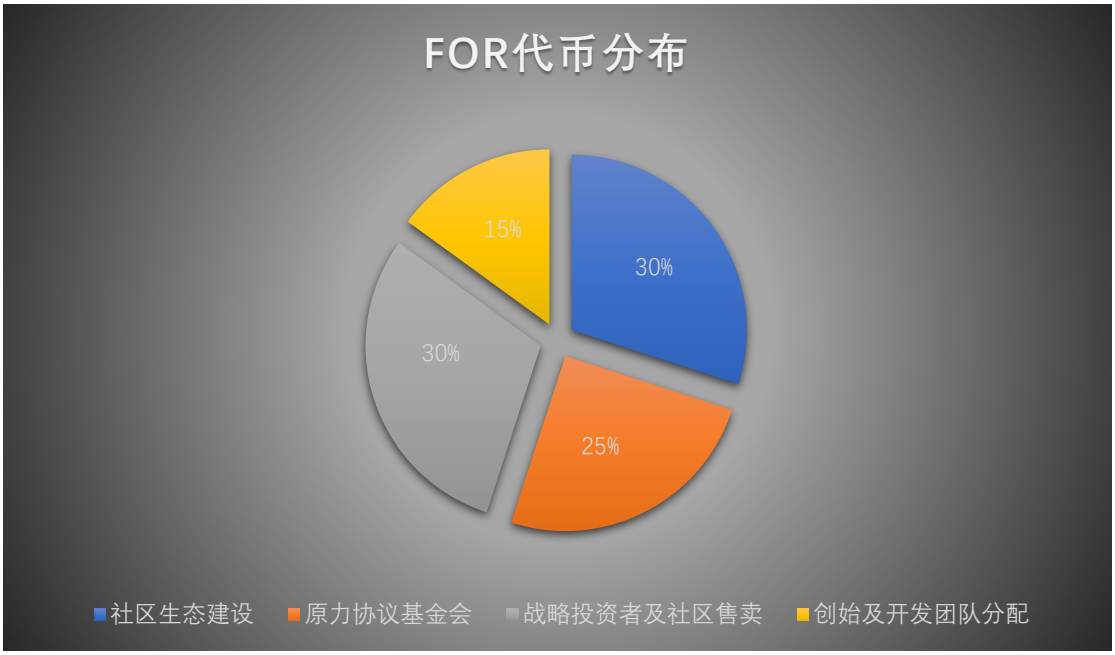
对于 EFOR 代币，增加了超级节点挖矿功能，超级节点质押并锁仓相应份额的 EFOR，接入 The Force Protocol 金融网络并贡献计算能力，从而获取相应的 EFOR 回报。举例，超级节点在更新/修订借贷订单簿区块链的过程中，可以获取 EFOR，其他的挖矿场景将随着 The Force Protocol 网络的不断完善而进行更新。

8.1.5 其他功能（待定）

原力协议在发展过程中，FOR（EFOR）代币将会找到更多的应用场景，届时，社区成员和原力协议基金会等各个参与方都可以通过向社区提交相应的议案，为 FOR（EFOR）代币赋予更多的实用功能和使用场景。

8.2 FOR 代币分配计划

FOR 代币总量 10 亿，永不增发。在原力协议发起团队主导下，将会有 85%的 Token用于社区建设和社区捐赠计划，其中社区生态建设占 30%，原力协议基金会占 25%，战略投资者及社区捐赠占 30%。剩余 15%的 Token 将由原力协议创始和开发团队预留，作为其在项目初期做出贡献的奖励，以及为后续新团队成员的预留。分配给团队的代币自首版应用上线开始锁仓 3 年，应用上线后 12 个月释放 30%，24 个月后释放 30%，36 个月后释放 40%。FOR 代币分配比例如下图所示。



8.2.1 社区生态建设

社区生态建设包括但不不局限于：原力社区区块链应用（DAPP）生态孵化

和激励、开发者社区建设、商业合作和产业合作、市场营销推广、学术研究、教育投资、法律法规等。

8.2.2 原力协议基金会

我们已经在新加坡注册非营利性原力协议基金会，该基金会主要任务负责原力生态的搭建和运营、开发战略方向的制定、FOR 代币发行及管理等，公开透明地管理由代币捐赠而获得的资金。

8.2.3 战略投资者及社区捐赠

根据项目发起及运营需求，我们将会预留 30%的代币回馈战略投资者及社区成员的资助。

轮次	代币价格 / USDT	出售比例	锁仓
基石轮	0.018	5%	永不释放
私募轮	0.030	17.5%	上所前发20%，上所后每3个月解锁40%，解锁两次

基石轮投资由团队创始成员们自筹资金完成，出于对项目的长期看好和自我激励，团队决定在基石轮所投入的资金对应的 FOR 代币永不解锁。

8.3 EFOR 代币分配计划

EFOR 代币总量 10 亿，永不增发。在原力协议发起团队主导下，10%用于兑换发行和空投以及市场宣传、社区治理；20%用于 EFOR 生态发展；70%通过 DPOS 机制，由维护 EFOR 生态的超级节点进行挖矿（staking 模式，挖矿开始时间待通知）。

8.3.1 兑换发行和空投

由于 EFOR 不面向投资者融资，在项目发展初期，用户只能通过 FOR 代币置换、空投两种方法获得 EFOR 代币。EFOR 总量的 10%将被预留，用于 FOR

代币兑换 EFOR、面向符合条件的 EOS 持币人空投和市场宣传等方面。在 10%的 EFOR 中，80%（占总量 8%）将用于兑换发行，兑换发行停止时，没有兑换完成的代币全部销毁。20%（占总量 2%）将用于项目启动、市场宣传、生态合作、社区治理权益赋能等。

8.3.2 生态发展

占总量 20%的 EFOR 代币将用于原力协议在 EOS 公链的生态发展，这部分的 EFOR 代币将用于交易手续费抵扣、超级节点质押锁仓、社区治理和市场流动性等部分，这部分代币的释放将由基金会根据具体情况决定。

8.3.3 超级节点挖矿

基于 EOS 公链的底层技术特点，占总量 70%的 EFOR 代币将被用于超级节点挖矿，超级节点质押锁仓相应份额的 EFOR 接入原力协议金融网络并贡献计算能力，从而获取相应的 EFOR 回报。举例，超级节点在更新/修订借贷订单簿区块链的过程中，可以获取 EFOR，其他的挖矿场景将随着原力协议网络的不断完善而进行更新。

9 原力协议项目

9.1 项目团队

Allen An——联合创始人 CEO

原 EnactusChina-Tianjin&ShanDong&Hebei Province Director，乾盛汇资本董事总经理，Zonff Partners 管理合伙人，聚焦并专注于互联网金融行业的研究及互金项目投资，并先后在消费分期、车贷、消费金融等细分领域投资出诸多优质项目。早期 Bitcoin Talk 社区成员，Xdag 中国社区早期成员，AlphaCoinFund 合伙人，以“代码即法律，隐私即自由，计算即权利”的主张，先后在区块链领域投资了诸如 RSK、Celer、Top、IRIS、Certik、Algorand 等六十个项目，提早布局了矿场、搭建了矿池，投资了钱包、交易所、媒体等行业生态项目。在原力协议、原力矿场及原力资本负责项目落地及生态合作和建设投资。

于宏学——联合创始人 CTO & 币币贷 CTO

北京航空航天大学计算机专业硕士，曾任职搜狗科技发展有限公司，搜狗大数据平台核心开发成员，牵头负责 ETL、核心指标计算、任务监控、任务调度、任务优化，并参与反作弊及推荐算法研究。作为区块链技术早期关注者，对比特币、以太坊、EOS 源码有深入研究，并为多个开源项目贡献代码、提交安全漏洞补丁。在原力协议牵头产品开发、区块链技术实现。

雷宇——联合创始人 架构师 & 原力稳定币业务负责人

清华大学硕士，2011 年起接触比特币挖矿，2016 年起系统研究区块链技术与加密经济，对区块链产业潜力和未来方向有独到见解。熟悉各种数据结构，精通密码学，安全协议和加密算法，对金融系统运作有深入的研究和理解。曾先后投资了 EOS、Filecoin、cybermiles 等优质项目，观察并参与多个加密数字资产项目的建设与运作，曾是 XDAG 中国社区开发及运营的主要成员。在原力协议牵头系统架构、稳定币业务和战略研究。

许超——联合创始人 COO & 币币贷业务负责人

哈尔滨工程大学硕士，曾任中兴通讯云计算架构师，国内 SaaS 云服务创新项目负责人，创建 SaaS 产品部署架构和业务运营体系，从 0 做到 1 亿营收规模。对网络结构，协议开发具有深入研究。参与中兴通讯基于区块链技术的电子证照方案规划和设计，参与中国联通研究院区块链技术路线规划和设计。区块链技术爱好者和早期社区项目参与者，曾参与 NEO、EOS 等项目众筹。在原力协议负责运营、商业模式和对外合作。

张琳波——联合创始人 首席科学顾问

中国科学院人工智能方向博士，高级工程师，工信部高级网络规划设计师，研究方向包括人工智能和数据挖掘。曾任职交通运输部科学研究院，期间独立完成中央级项目 1 项，主持省部级及以上信息化项目 10 余项，参与省部级及以上信息化项目 30 余项。主导交通运输部网约车（即现在的打车软件）、客运联网售票和自动驾驶技术在国内应用政策研究。常用 C、C++、C#、JAVA、Go、Python 等编程语言，对分布式账本技术、加密数字资产前沿技术、跨链信息交互具有深入研究。在原力协议牵头密码学、人工智能和区块链技术底层研究。

王桂杰——合伙人 系统安全负责人

Thinkbit Pro 交易所创始人，前陌陌高级工程师，北京航空航天大学硕士辍学，早期参与比特币/以太坊挖矿，EOS 社群持仓大户，区块链技术信仰者，对区块链技术有深入研究，坚信区块链技术会普惠所有人。

王鑫伟——市场总监 原力矿业负责人

毕业于北京大学，原力矿场业务负责人，早期矿工，2017 年起先后参与诸如 RSK、Celer、Box 等十多个项目的投资和社区建设，BetaCoin Fund 合伙人，成功操盘了国内外矿场、区块链媒体等数个项目的商务运作。

刘刚——产品总监

武汉大学软件工程本科，曾在多家 P2P 公司和消费信贷公司担任产品总监

职务，从 0 到 100 参与公司创立和运营，具有丰富的网络借贷从业经验。他所负责的产品交易额超千亿，用户达数千万。擅长产品设计，用户增长和金融风险管理。

郑亚军——区块链开发工程师

哈尔滨工业大学计算机专业本科，曾任百度资深研发工程师，深圳正前方金服风控总监，数据中心总监，设计公司整体业务风控体系和架构，带领团队完成大数据分析、信用评级和贷款管理等多方面风控模型开发。在互联网和金融领域有丰富工作经验。区块链技术极客，参与多个 GitHub 开源项目代码提交。

王捷——融资与财务

浙江大学数学系博士，曾任银河证券保荐代表人，参与多家公司上市及兼并重组，在资本市场具有丰富经验。

9.2 顾问团队

Frozen Xie

公链和 Dapp 开发者，开源项目 XDAG 核心开发者&维护者，多个区块链和开源项目贡献者；共识之道联合创始人，TeamTaoist 工作室创始人；IBM DB2 专家，网络通信专家，区块链技术专家，资深 iOS/Android/Html5 工程师，游戏制作人；前 Lucent 贝尔实验室技术工程师、项目经理。

田鸿飞

麻省理工学院获硕士，现任松禾远望资本合伙人，在此之前就职于 SIG 海纳亚洲创投基金，担任合伙人一职。田鸿飞先生在电子商务和网络安全领域有超过 15 年的工作经验，工作范围遍及硅谷、德国、中国的高科技产业和投资银行。

Andrew Yi

曾工作于世界 500 强甲骨文、谷歌，曾任中国最大的证券公司之一申万宏源研究所执行院长，中国互联网金融、大数据、云计算产业互联网的第一推动和提倡者，中国证券行业最佳分析师排第一名，腾讯、新浪评选的中国金融科技年度人物，投资了近百家相关领域创业公司。

Simon Liu

清华大学 ，计算机硕士，百度搜索技术开发工程师，百度早期员工，Google 中文搜索核心工程师，微软 Bing 搜索首席工程师。

9.3 战略合作

TokenInsight

TokenInsight 作为全球通证数据与评级机构，专注于通证风险评级，帮助投资者规避风险、提高收益。原力协议与 Tokeninsight 在加密数字资产抵押品选择、抵押率设置以及风控水平等方面紧密合作，共同提高抵押借贷风控水平。

BabelBank

贝宝(BabelBank)是值得信赖的区块链银行，致力于打造一个开放共享的区块链资产金融服务体系，业务内容包括数字货币存贷、区块链资产融资、数字货币融币等，未来会推出更多业务及相关金融衍生品。作为数字货币质押借贷领域的先驱，贝宝将在原力协议去中心化借贷网络上线后，作为超级节点共同维护全球借贷订单池，优化资源配置，共同制定去中心化数字货币借贷行业规范，为用户提供快速便捷的借贷服务。

InVault

InVault 作为亚太首家持牌的虚拟资产托管平台，拥有香港地区的信托及公司服务者牌照，已为数字借贷平台和量化交易基金两种垂直性业务推出了虚拟资产托管解决方案。InVault 除去中心化企业钱包，还提供协同托管和专户托管等适用于各类业务场景的虚拟资产托管解决方案，包括一级市场基金托管、二级市场

基金托管、交易所钱包托管、FOF 基金托管、质押物托管、OTC 资金监管等。原力协议超级节点可联合 InVault 企业级虚拟资产托管技术和方案，面向用户推出抵押贷、多币种组合抵押贷和大额多签账户抵押贷款等服务。其多层级风控体系保障用户抵押资产在价格大幅波动情况下可将损失减到最小。

币涨 BIRISE

币涨(Birise)是一款带有行情量化分析功能的数字资产钱包，提供综合行情，量化分析，新闻公告等等衍生服务，为用户解决数字资产投资痛点，实现数字资产增值，使用户拥有更安全，便捷，增值的数字钱包。原力协议将与币涨在数字资产借贷方面进行合作。

IOST

IOST 致力于构建一个超高性能的区块链基础设施，以满足去中心化经济对安全性与可扩展性的需求。IOST 在世界级投资者的支持与连续成功创业的创始团队带领下，立志成为未来在线服务的底层架构。

KCASH

Kcash 作为一个安全、便捷、高效的数字资产管理金融平台，要让每一位用户可以随时随地便捷地使用数字资产。对于数字资产金融领域的开发，Kcash 一直走在前列，推出了币生币、活币宝等金融产品。作为国内首款多链数字资产钱包，Kcash 自 2017 年上线以来，注册用户已经突破 200 万，支持数字资产代币种类达到 1 万+以上，管理数字资产超过 150 亿人民币。

MyToken

MyToken 是业内最具影响力的数字资产行情 App，立志于为全球用户提供体验感最佳的一站式数字资产投资服务，呈现具有充分价值的内容资讯和社交体验，并集成更多深度投资工具。

9.4 社区治理

原力协议团队认可去中心化社区自治管理模式，每个原力协议社区的参与者都有权参与项目管理和生态规则制定。原力协议团队在新加坡成立了非盈利基金会（The Force Protocol Foundation，下称原力基金会）做为项目管理主体，负责公正，公开，透明，不以盈利为目的地运行原力协议项目，维持原力协议和社区的正常发展和运营，管理所有募集加密数字资产的安全性，并且对原力协议的开发和运营团队提供支持。该基金会如有利润所得，将被继续保留作为其他活动的经费，而不在成员中分配利润。原力基金会由 Accounting And Corporate Regulatory Authority（新加坡会计与企业管理区，ACRA）批准建立，受新加坡公司法监管，该基金会独立管理运营并独立于政府之外。

为帮助原力基金会在公正，公开，透明的前提下合理利用基金会的资金，资源，不断推进原力协议的快速发展，扩展原力协议的应用场景，吸收更多机构、公司、项目、组织和开发者进入原力生态，基金会设立决策委员会，在决策委员会下设置代码安全审计委员会，原力生态共建委员会，财务及人力资源委员会，市场与公共关系委员会。

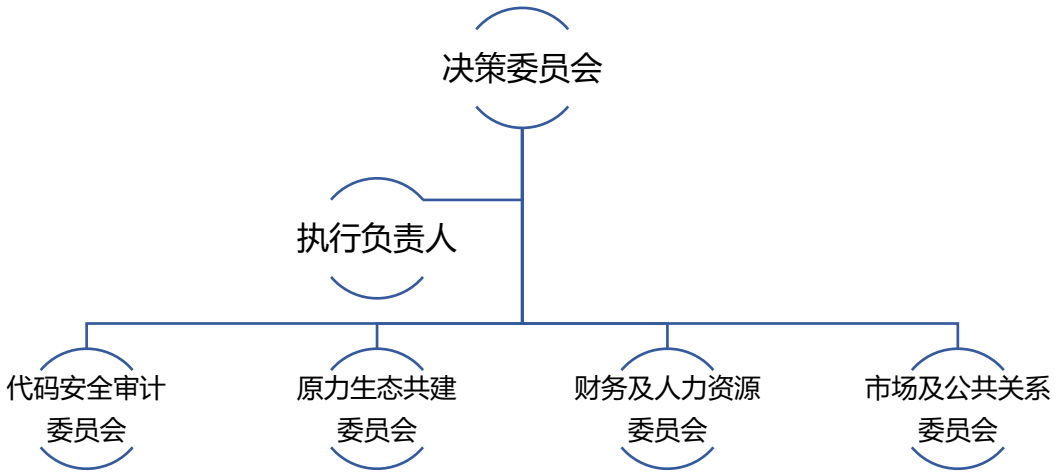


图 9.1 原力协议基金会组织结构图

决策委员会：

决策委员会是原力基金会的最高决策机构，承担最终决策职能，负责对基金会战略规划、年度计划、预算等重大事项进行审议，并代表基金会对原力生态的重大议题作出表决。

执行负责人：

执行负责人由原力决策委员会选举产生，负责基金会的日常运营管理，各下属委员会的沟通协调，主持决策委员会会议，并定期向决策委员会汇报工作情况。

代码安全审计委员会：

负责原力协议所有代码安全审计，技术研发方向的制定和决策，数据接口开放，技术专利的开发。此外，委员会也将在社区中与社区成员和生态参与者保持沟通交流，并不定期举办技术交流会。

原力生态共建委员会：

负责原力生态发展和合作伙伴建设，委员会将使用募集资金开展生态建设和商务合作，激励更多地开发者基于原力协议构建应用，更多地把潜在合作者纳入到原力的生态体系中来。

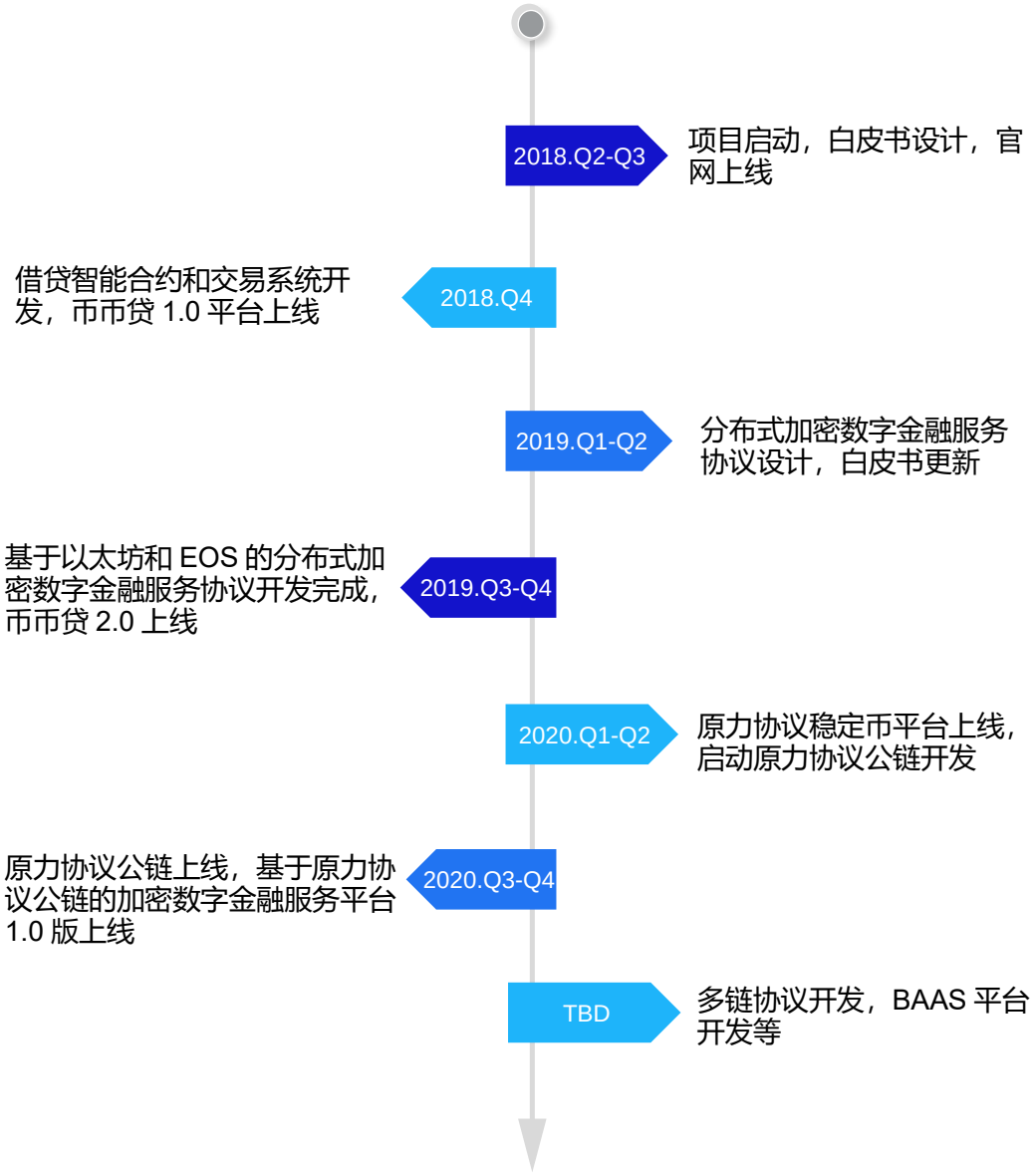
财务与人力资源委员会：

负责基金会资金的运用和审核，人员聘请及薪酬管理，日常运营费用管理。

市场与公共关系委员会：

负责原力协议及生态项目的市场宣传和推广，举办社区沟通交流会，参与区块链领域展会和学术研讨会。公共关系维护，保持与行业协会及政府监管组织的良好沟通。

9.5 项目路线计划


2018.Q2-Q3
项目启动，白皮书设计，官网上线
借贷智能合约和交易系统开发，币币贷 1.0 平台上线
2018.Q4
2019.Q1-Q2
分布式加密数字金融服务协议设计，白皮书更新
基于以太坊和 EOS 的分布式加密数字金融服务协议开发完成，币币贷 2.0 上线
2019.Q3-Q4
2020.Q1-Q2
原力协议稳定币平台上线，启动原力协议公链开发
原力协议公链上线，基于原力协议公链的加密数字金融服务平台 1.0 版上线
2020.Q3-Q4
TBD
多链协议开发，BAAS 平台开发等

10 法律评估

原力协议借助区块链技术实现了“去中介”的借贷形式，避免了传统借贷机构需要在借贷前完成尽职调查，并需要借用第三方机构维护的本地评级或评分系统的过程。由于各国的立法不同，原力协议全球化借贷平台面临的主要问题是各个超级节点需要遵循所在国家和地区的立法，同时保护交易双方的合法利益。

对于世界各国的政府来说，加密数字资产的概念还比较新颖，各国政府在定义加密数字资产的状态和出台法律规范方面还不擅长。此外，税法等其他领域法律给出的定义或规则并无法直接沿用到区块链的加密数字资产领域。因此，目前管理加密数字资产相关的交易最好采用合同协议的形式。

10.1 合同关系

目前加密数字资产尚未纳入大部分国家的监管规范，合约可以被视为不同团体间的合同合约。也即，借款人和出借人之间的合同问题可以依据合同法执行。然而问题在于，合同法随着管辖区域的不同存在很大差异，当全球性借贷问题发生时很难确定具体按照哪一方所遵守的合同法执行。但上述问题不会成为原力协议面临的重要问题，因为所有合同法都遵循自由交易的基本原则，借贷可以在任何人之间自由进行，而由双方提前认可的智能合约处理纠纷。

尽管基于智能合约技术和去信任的环境，出借人和借款人还会接受当地政府的监督。在传统模式下，合约规定的借贷合同可能是有效的，但是即使政府对违约也无能为力。基于原力协议的借贷平台，可应用智能合约明确借贷周期、质押代币种类、交易双方及其需要尊重规则和争端解决方案，使得借贷用户在参与借贷过程中不需要关注太多合法性问题。

10.2 抵押

基于区块链技术的借贷平台采用新的抵押品类型。传统借贷市场的大部分情形中，抵押通常是由第三方保管并从法律上保障借贷双方的利益。抵押物所有权的变化只有在异常导致的违约发生时才会执行。而且，大部分传统借贷市场中抵

押品的规则都是针对不动产抵押，尚不存在针对 ERC20 代币抵押的相关法律研究。

传统抵押涉及抵押品所有权的确认和转移，抵押物需要经过一系列的手续才能真正生效，因此在默认情况下即可发生所有权的转移。这些规则适用于大多数司法或行政区域，而且抵押品必须要转移给借出方。加密数字资产市场的情况略有不同，抵押物可能由程序控制的第三方持有，也即抵押品由借出方持有，而是保存在由智能合约控制的区块链上。在传统借贷中，借出方占有抵押物并通知第三方已经质押；而在基于区块链的借贷中，交易的细节信息可以很容易地从区块链查询得到。因此，原力协议平台的借贷可以由超级节点把抵押物质押在智能合约，直到借款归还。抵押信息对所有参与方透明，用户可以在任何时间、任何地点联网查询抵押情况。

10.3 目标客户定位（KYC）

原力协议中仅充当交易平台角色的超级节点不出借或者持有加密数字资产，仅作为以太坊网络上的去中心化应用平台。即使超级节点开始行使理财产品代理投资方角色，其身份也是作为贷款参与方，不会在平台上享受特权。超级节点和原力协议不存储影响合约执行的任何数据，所有相关数据都运行在去中心化的以太坊智能合约上。超级节点本身不控制借贷双方的任何资产。也即当借贷双方达成交易并在智能合约中部署后，数据将向整个以太坊网络广播，而不是仅仅存放在当地的服务器上。超级节点可被视为制作、撮合、广播订单的工具集。

KYC 规则主要适用于借贷双方之间的现金借贷，虽然阈值因地域差别而不同，但整体原则上相近。然而，基于以太坊的借贷存在一定程度的不确定性，因为 ERC20 形式的代币不是政府发布的法定货币，是否受制于 KYC 还是悬而未决的问题。原力协议支持 KYC 政策，因为区块链及相关服务要想持续发展，必将需要权威机构的监管，即使是去中心化的环境也不能避免。原力协议及建于其上的超级节点，将倡导遵循 KYC 规定。KYC 政策的目的是在必要的时候能够及时提供 KYC 信息，现有的 KYC 主要通过在借贷双方的信息交互实现。

未来，原力协议团队将致力于研究链上的 KYC 解决方案。目前最简单的方

法就是要求借款人在资料中插入与 KYC 相符的超链接。这些资料包括身份证明、地址证明、资金来源。然后这些 KYC 数据使用去中心化方式存储。原力协议团队将保持对链上 KYC 解决方案行业进展的关注，在适当的时机引入合适的链上 KYC 机制。

11 注意事项与风险提示

11.1 注意事项

本白皮书仅作为一份概念性文件，用于描述原力协议技术方向，发展规划和FOR（EFOR）代币，并不构成招股说明书，要约文件，证券要约，投资招标或出售任何产品，资产的要约。基金会和原力协议团队无法保证白皮书信息的绝对准确性和完整性，投资者应该在参与本白皮书中所述任何活动之前咨询自己的法律、财务、税务或其他专业顾问。

所有原力协议项目的支持者，应当仔细阅读白皮书和官方网站的相关说明，全面理解区块链技术，明确了解原力协议项目的风险，投资者一旦参与投资即表示了解并接受该项目风险。投资者也应该明白获取原力协议代币本质上为捐赠行为，愿意承担风险与原力协议社区共同成长，并不会因为获取 FOR（EFOR）代币而获得任何直接或者间接收益或者分红。

FOR（EFOR）仅作为原力协议生态的使用通证，并不代表分红、增值、股权、证券及其衍生品的收益许诺，项目方不提供任何回售渠道，持有人获取后有权自主决定使用。本白皮书有多种语言版本，如存在任何分歧，以英文版为准，参与者承认已亲自阅读并理解本白皮书的英文版。

11.2 风险提示

1. 目前世界上主要国家对于区块链项目即使用加密数字资产融资的态度和政策尚不明确，存在由于政策原因造成投资者损失的可能性；

2. 包括 FOR（EFOR）在内的加密数字资产交易具有极高的不确定性，并且缺乏合理的监管，所有的加密数字资产都存在暴涨暴跌，受到庄家操控的风险；

3. 当前区块链技术领域项目众多，竞争激烈，存在非常强的市场竞争。原力协议团队拥有丰富的经验和产业资源，也将全力确保原力协议项目的继续发展壮大，但我们无法确保项目的必然成功，也不会就项目的发展状况做出任何承诺；

4. 原力协议团队将不遗余力实现白皮书中提出的目标，并积极探索项目更

长远的发展空间，然而由于外部环境和内部资源的不确定性，我们将保留对白皮书描述内容进行调整的权力。白皮书内容的所有变更我们并无主动告知义务，请参与者通过相关渠道及时了解更新；

5. 原力协议的架构将基于区块链技术和密码学算法构建，目前区块链技术仍然是一项非常早期的技术，密码学也一直处于高速的发展过程中，原力协议团队不能完全确保所有技术的顺利落地，同时所有的技术类项目都具有被黑客攻击或代码漏洞造成用户损失的可能；

6. 除上述风险外，由于加密数字资产投资仍然是一个崭新的领域，可能还有各种我们尚未提及或尚未预料到的风险。

12 更新事宜

原力协议智能合约一旦部署，其内部的逻辑结构将不能改变。因此，当原力协议中的智能合约更新的时候，需要重新部署。负责注册的智能合约将把更新后的智能合约列入白名单，并注明版本号。所有未成交订单都将中断并赋予最新的智能合约转移/锁定指定数量代币的权利，而已成交的交易将继续按照原智能合约执行，届时区块链平台上有段时期将存在多版本同时并存的现象。为防止更新影响上层应用，原力协议将采用抽象合约的策略来处理更新过程中合约的调用问题。

在借贷环境下，每笔交易都将创建一个合约实例，两者一一对应。更新前达成的交易按照旧版本合约继续执行的机制存在风险，例如在最坏的情况下抵押物可能被黑客截取。FOR（EFOR）代币将被用来驱动更新机制的研究，在保障升级顺利进行的同时保护用户和 FOR（EFOR）代币持有者的财产安全。

13 开源社区

为保障系统安全，原力协议项目开发完成后，将进行数十轮的代码测试及安全审计。在确保代码不存在明显问题和漏洞后，所有的关键代码都将上传到 Github 等开源社区进行共享，具体地址如下：

https://github.com/theforceprotocolgroup

在共享社区中，所有的智能合约都将存储在“\smartcontract”目录，所有的库文件都存储在该目录的“\lib”子目录下。API 接口文件将存储在“\APIs”目录，用于部署的接口文件也将在开源社区发布。

目前，源代码还在开发过程中，并陆续上传到相应代码库。代码上传后，如果有任何问题需要原力协议团队支持或者回应，社区成员可以在开源社区新建进程，和团队进行交流探讨。

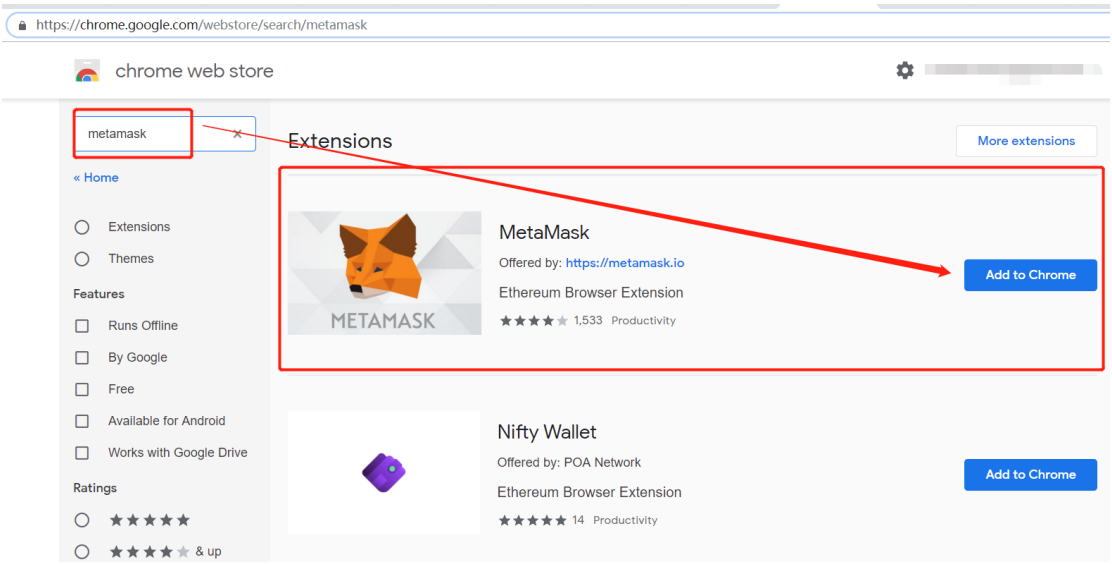
此外，若 Github 网站发生任何可能影响到原力协议代码正常开源或社区正常交流的问题，原力协议团队保留更改代码公布渠道的权限，并将通过相关渠道向社区进行消息通知。

14 附录：应用币币贷 2.0 准备工作

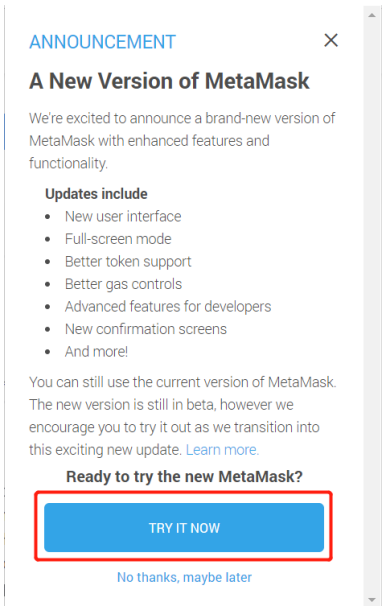
该部分主要是使用基于原力协议团队开发的 DAPP 的指南，以 Ethereum 网络的 DAPP 为例，EOS 网络的 DAPP 与之类似，不再重复进行说明。

从下述站点下载 Google Chrome 浏览器：https://www.google.com/chrome/，点击下载按钮即可下载。

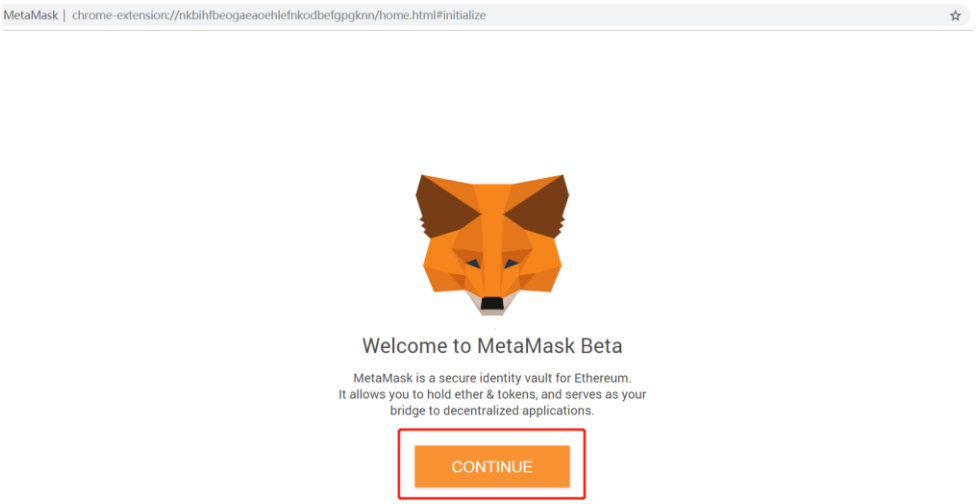
然后访问 https://metamask.io/下载 Metamask，如果在中国境内打不开，可以登录：https://chrome.google.com/webstore/category/extensions，页面如下图所示，然后在搜索栏搜索“**metamask**”。



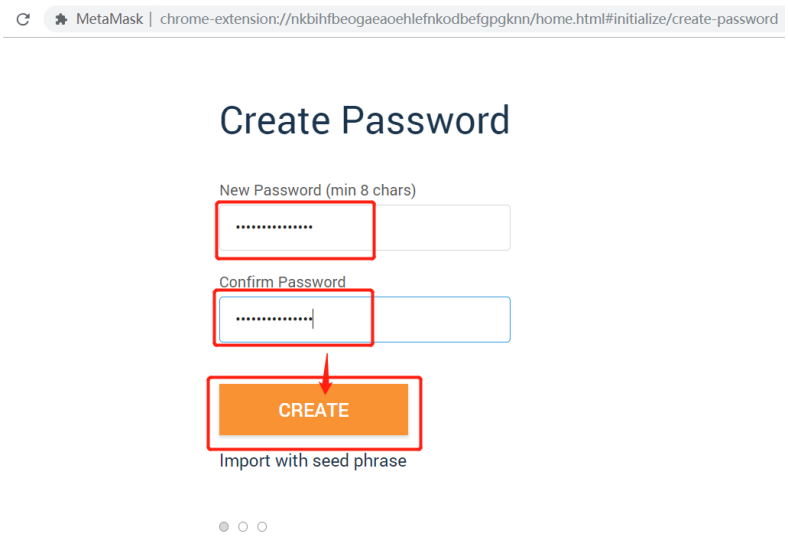
点击“添加到 chrome”按钮，等待下载过程结束。



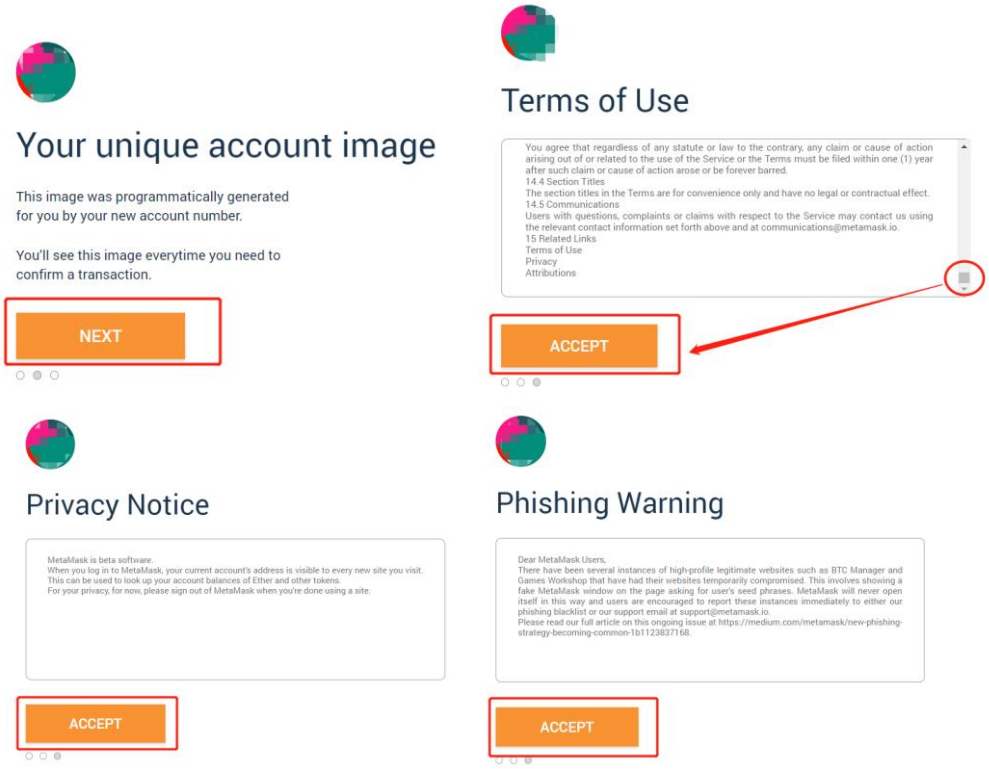
在选择新版本页面，选择“试用新版本”按钮或者选择“以后再说”的超链接。



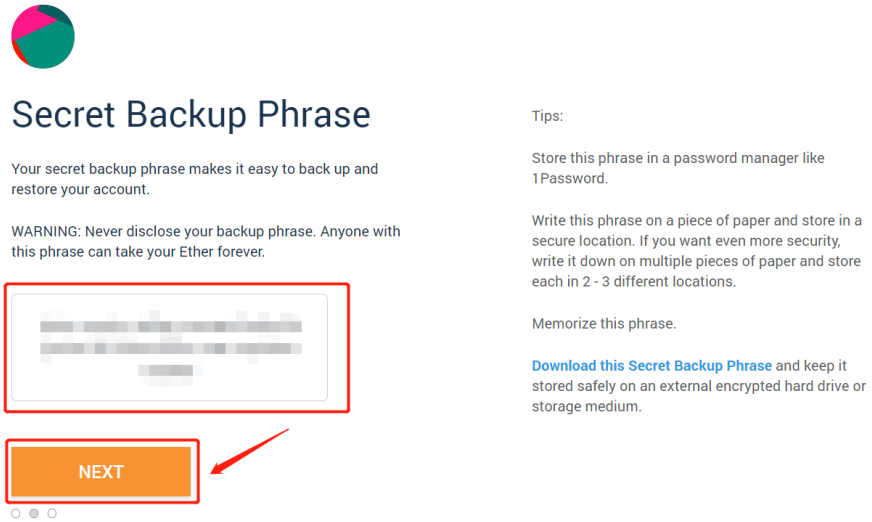
点击“继续”创建新账户。



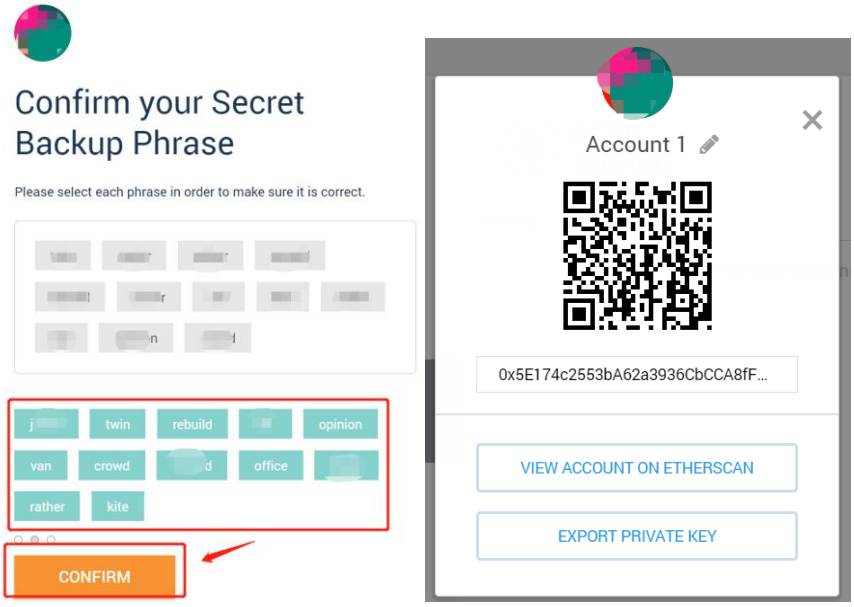
创建用户密码并输入两次，点击“创建”按钮。



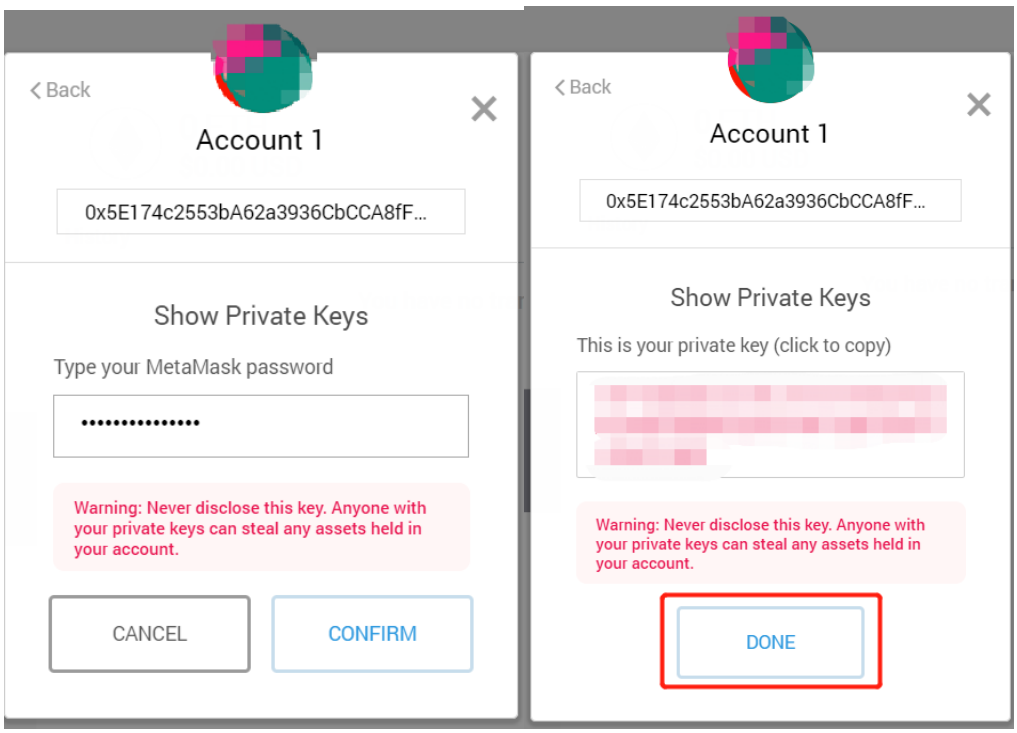
连续点击“下一步”或“接受”，直到碰到备份密码助记词页面。

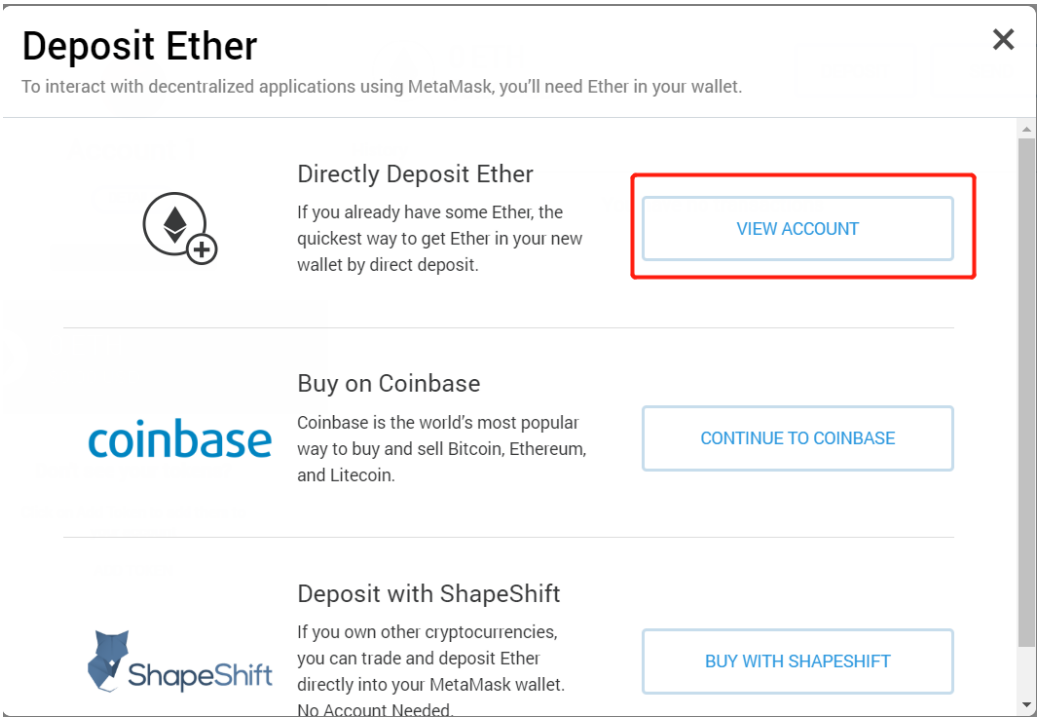


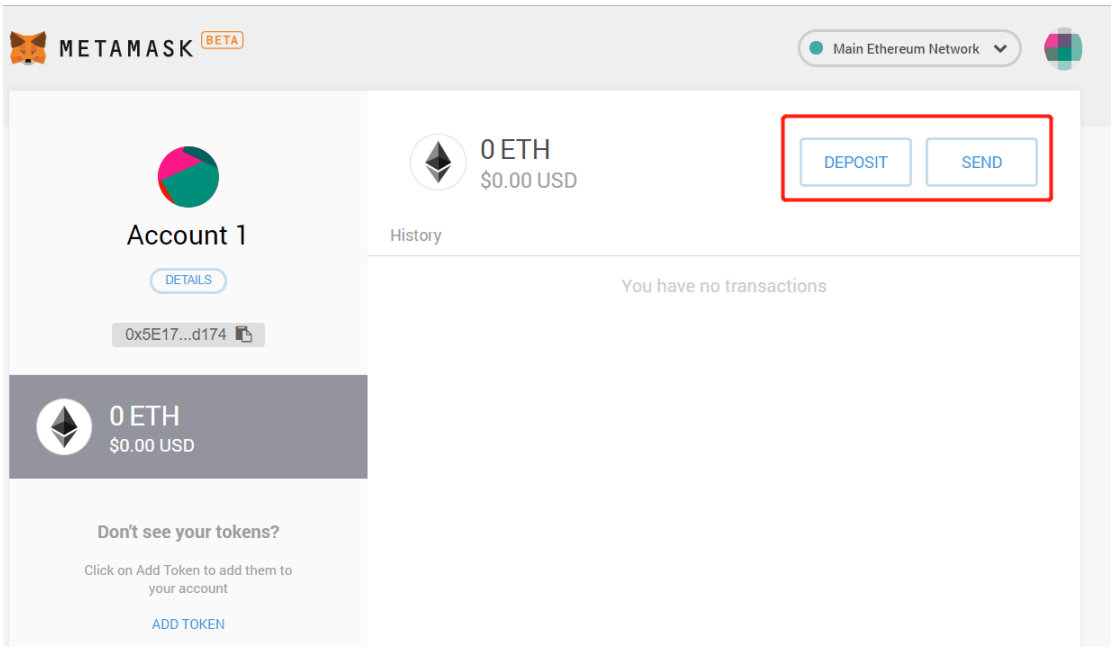
将页面中的助记词存放在安全的位置，不要泄露给任何人。然后点击“下一步”。



在这个页面中，你需要按照前页的顺序依次点击助记词。最后点击“确认”，你的账户就被创建了。你可以用它登录原力协议的 DAPP 或者其他钱包操作。


Back
Account 1
0x5E174c2553bA62a3936CbCCA8fF...
Show Private Keys
Type your MetaMask password
Warning: Never disclose this key. Anyone with your private keys can steal any assets held in your account.
CANCEL
CONFIRM
Back
Account 1
0x5E174c2553bA62a3936CbCCA8fF...
Show Private Keys
This is your private key (click to copy)
Warning: Never disclose this key. Anyone with your private keys can steal any assets held in your account.
DONE


Deposit Ether
To interact with decentralized applications using MetaMask, you'll need Ether in your wallet.
Directly Deposit Ether
If you already have some Ether, the quickest way to get Ether in your new wallet by direct deposit.
VIEW ACCOUNT
Buy on Coinbase
coinbase
Coinbase is the world's most popular way to buy and sell Bitcoin, Ethereum, and Litecoin.
CONTINUE TO COINBASE
Deposit with ShapeShift
ShapeShift
If you own other cryptocurrencies, you can trade and deposit Ether directly into your MetaMask wallet. No Account Needed.
BUY WITH SHAPESHIFT


METAMASK
BETA
Main Ethereum Network
Account 1
DETAILS
0x5E17...d174
0 ETH
$0.00 USD
Don't see your tokens?
Click on Add Token to add them to your account
ADD TOKEN
0 ETH
$0.00 USD
DEPOSIT
SEND
History
You have no transactions