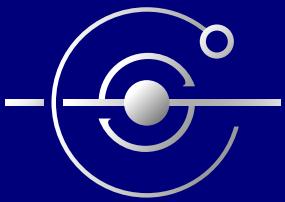


■ 2019 THE FORCE PROTOCOL



THE FORCE PROTOCOL

原 | 力 | 协 | 议

Info@theforceprotocol.com
www.theforceprotocol.com



THE FORCE PROTOCOL

原力协议

去中心化数字资产借贷协议

白皮书 V2.10

2019年01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，加密数字资产抵押借贷成为加密数字资产持有人获得流动性的最佳途径。然而，由于不同加密数字资产借贷应用平台基于不同的规则，使得加密数字资产借贷服务间缺乏有效协同，导致用户使用体验较差。基于EOS区块链平台，原力协议团队计划开发不同场景下加密数字资产借贷的协议和标准。通过将加密数字资产借贷请求转换为类似代币交换，实现对订单格式进行标准化处理。进而结合去中心化代币交换思想和智能合约技术，实现去中心化加密数字资产抵押借贷服务。借贷服务和稳定币可以相互协同，为尽可能消除现有稳定币模型存在的问题，充分发挥稳定币的优势，原力协议团队设计出了双代币模型。该模型通过主动引入托管银行和审计单位等第三方机构，极大的降低了政策风险对加密数字稳定币的影响，首次提出稳定币的可持续完全储备金制度。原力协议代码开源并支持第三方通过API调用建立超级节点，以期为全球不同的加密数字资产借贷应用提供协议层服务，扩充借贷应用的流动性、交易深度。

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、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。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。

CONTENTS

1.项目概述	06
1.1 引言	06
1.2 项目目标	07
1.3 主要贡献	09
1.4 项目效益	10
2.法币背书的稳定币	11
2.1 原力协议超级节点	13
2.2 投资人	15
2.3 借款人	15
2.4 仲裁者	16
2.5 其他参与方	16
3.实施细节	17
3.1 公共订单交易流程	17
3.2 点对点 (P2P) 订单交易流程	22
3.3 借贷订单参数设置	23
3.4 预言机: 安全保障机制	24
3.5 保险: 应急管理	25
4.信息规范	26
4.1 公共订单	26
4.2 点对点订单	28
5.智能合约	29
5.1 相关库	29
5.2 原力协议智能合约	30
5.3 系统安全	36
5.4 合约风险控制	36

CONTENTS

6.API接口	37
6.1 超级节点API	37
6.2 点对点个人交易	39
7.币币贷 2.0: 展示平台DAPP	40
7.1 准备工作	40
7.2 DAPP应用展示	41
8.原力协议原生代币EFOR	47
8.1 EFOR代币用途	47
8.2 EFOR代币分配计划	49
9.原力协议项目	50
9.1 项目团队	50
9.2 顾问团队	52
9.3 战略合作	53
9.4 社区治理	55
9.5 项目路线计划	57
10.法律评估	58
10.1 合同关系	58
10.2 抵押	59
10.3 目标客户定位 (KYC)	59
11.注意事项与风险提示	60
11.1 注意事项	60
11.2 风险提示	61
12.更新和开源事宜	62
12.1 更新事宜	62
12.2 开源社区	63

01

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、EOS等为代表的区块链2.0及以上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2018年1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）的市值已达8300亿美元。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。

尽管当前市面上已经有ETHLend、SALT、CRED等多个借贷项目，但加密数字资产借贷市场上的大部分需求仍未得到有效满足。该情况的出现可归因为下述三个因素：首先，当前已经上线的去中心化借贷平台无法完全实现加密数字资产的跨链交易，而中心化的借贷平台则存在系统安全、信任风险等传统借贷市场中存在的问题；其次，全世界借贷市场的需求分散到多个平台中，每个平台中的订单信息无法在不同平台之间自由流动，无法实现订单在全球范围内的动态撮合，而这成为阻碍加密数字资产借贷市场发展的重要障碍；第三，虽然USDT等稳定币可被用于避免因价格波动造成的加密数字资产价值缩水，但Tether公司财务不公开透明的现状及其未得到美国政府认可的现实，使得USDT面临很大的不确定性风险，进而影响了加密数字资产市场的进一步发展。

基于上述现状，原力协议团队正在开发基于EOS区块链的去中心化加密数字资产抵押借贷协议——原力协议。原力协议为搭建去中心化的借贷平台提供支持，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 构建全球范围的借贷网络

原力协议致力于打破传统借贷市场存在的操作环节程序繁冗、交易不公平现象丛生的现状，打破不同借贷平台之间的信息壁垒，使得全球范围内的借贷需求都能得到有效的满足。原力协议团队认为任何借款人都应该有机会从全球任何地方获得需要的贷款，而不过度依赖传统借贷机构。加密数字资产的出现使得每个人都可以真正拥有自己财产的所有权和支配权，而基于原力协议，借贷机构可以方便的建立属于自己的借贷平台（原力协议中称为超级节点），该借贷平台可以与其他借贷平台共同维护借贷订单簿。原力协议团队的目标就是在技术层面使世界上每个人都能方便安全的参与全球的借贷体系，进而满足自己的借贷需求。

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。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）不要求借贷双方提前购买EOS资源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。通过在超级节点使用类中心化的方式实现跨链交易及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等技术，基于区块链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 法币背书的稳定币

为改进USDT、GUSD、PAX等稳定币的不足，原力协议将引入新的稳定币生成机制，并由超级节点负责具体的发行事宜。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一款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。智能合约给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。

02 法币背书的稳定币 /

近年来，随着加密数字资产技术的发展，使用BTC、ETH和EOS等主流加密数字资产购买商品或服务，已逐渐被越来越多公众接受。由于这些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、ETH和EOS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、ETF等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。

只要将稳定币和法币建立联系并保持1:1的兑换率，其发行就必然要接受各国政府的监管。完全的去中心化思路发行的稳定币，很难在世界各国得到政府的认可。在寻求币价稳定机制方面，算法支持的稳定币也存在缺陷，使得这类稳定币很难得到广泛应用。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。

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、借款人三个主要参与方，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共同维护的联盟链共享借贷订单簿，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。

原力协议体系内的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，超级节点在区块链公网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，智能合约执行所需CPU、内存、带宽等资源，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将投票选出个人或团队，并有一定时间的任期，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EFOR。当超级节点未能在规定时间内补充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 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。

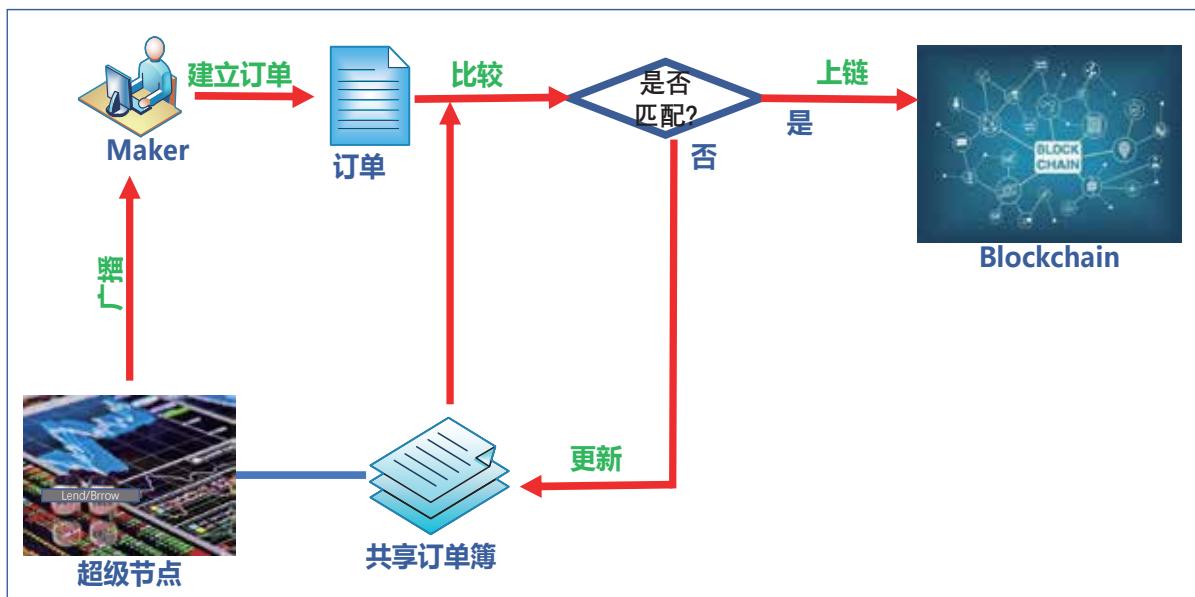
3.1 公开订单交易流程

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、EOS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。

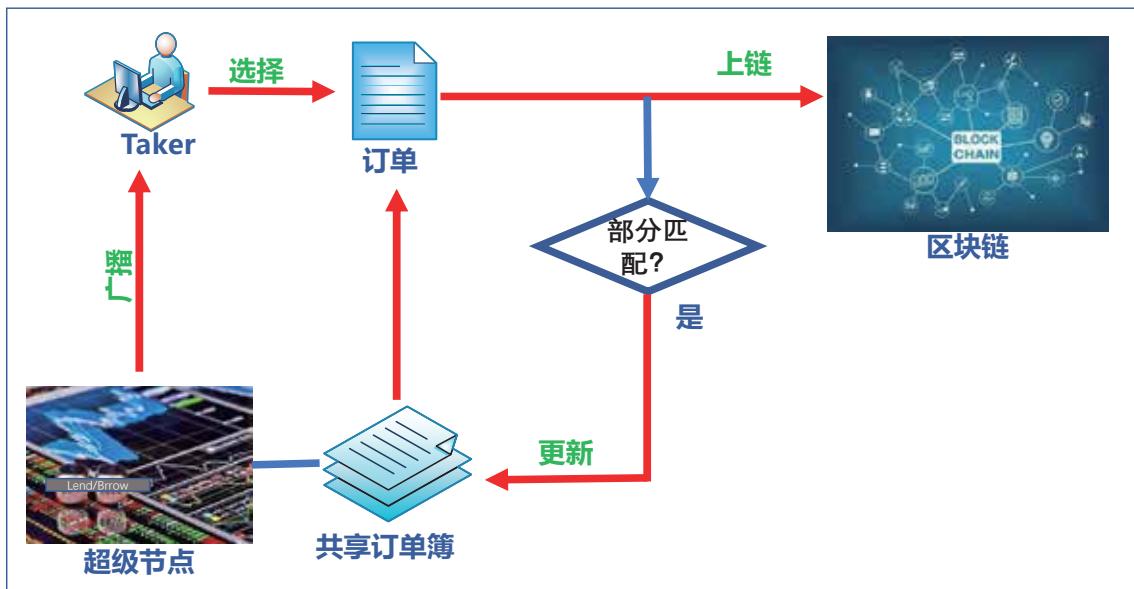
前置工作：超级节点发布借贷规则，规则中包括抵押率等参数。同时公布收费标准、收费地址等内容。

3.1.1 订单提交和自动撮合



1. Maker授权智能合约转移特定数量借货币种或抵押币种的权利。
2. Maker创建订单，明确借货币种及数量、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。
3. 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。
4. 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。

3.1.2 用户自撮合



1. Taker登录任意的超级节点接收最新的共享订单簿;
2. 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。
3. 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。
4. 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

1. 创建智能合约的实例, 参数由上述orderNew和orderMatch确定;
2. 智能合约锁定借款人的抵押代币;
3. 智能合约将出借人地址中的借贷款种转移到借款人地址;
4. 合约正式生效。

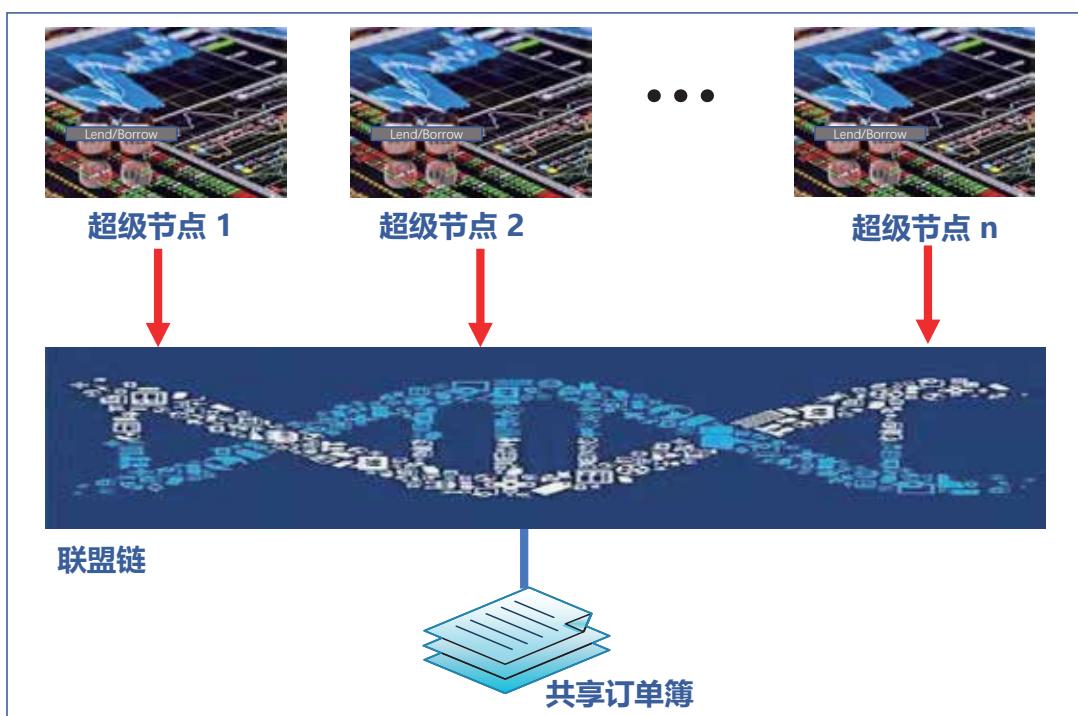
3.1.4 到期结算

当借贷合约到期，将启动还款程序，过程如下：

1. 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；
2. 借款人将应还本金和利息发送给智能合约；
3. 智能合约将得到的代币和利息发送给出借人；
4. 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通过联盟链实现，即全球基于原力协议的所有超级节点共同维护共享订单簿。第1版原力协议将选择基于Aura技术的Parity创建联盟链。受到超级节点的制约。



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。

1. 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；
2. 出借人B发送指定数量的Token B到智能合约；
3. 智能合约将接收到的Token B发送到借款人A；
4. 智能合约开始为合约的有效性提供证明。

3.2.1 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：**▼

1. 出借人B通知借款人A还款；
2. 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；
3. 智能合约将上步中得到的代币发送给出借人B；
4. 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。订单簿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，具体如下页所示：

A_n =向前第n个周期的振幅 (%) ;

$$A_{predict}=A_1+\text{Max}(0, A_2-A_1)/2^{(2-1)}+\text{Max}(0, A_3-\text{Max}(A_1, A_2))/2^{(3-1)}+\cdots$$

$$+\text{Max}(0, A_n-\text{Max}(A_1, A_2, \dots, A_{n-1}))/2^{(n-1)}; \text{ 其中 } n \text{ 通常取 } 3.$$

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：

1. 创建智能合约的实例，参数由上述orderNew和orderMatch确定。
2. 智能合约锁定借款人的抵押代币。
3. 智能合约将出借人地址中的借货币种转移到借款人地址。

3.5 保险：应急管理

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。

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。

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EOS之上（后期将扩展到ETH、NEO及未来的主要公链），原力协议安全性和应用水平将很大程度上取决于EOS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。借贷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。

04

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。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

04

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。

名称	数据类型	描述
addrVersion	address	智能合约地址, 升级后更新
nFeeLender	unit256	出借人手续费
nFeeBorrower	unit256	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

名称	数据类型	描述
addrTokenA	unit256	质押币Token A地址
addrTokenB	unit256	质押币Token B地址
nTokenA	unit256	质押币Token A数量
nTokenB	unit256	质押币Token B数量
nExpiration	unit256	订单有效期
nLendingCycle	unit256	借贷周期
ufMotageRate	ufixedOx256	抵押率
ufInterestRate	ufixedOx256	年化利率
vLender	unit8	出借人对上述参数的ECDSA签名
rLender	bytes32	
sLender	bytes32	
vBorrower	unit8	出借人对上述参数的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	ufixedOx256	抵押率
uflInterestRate	ufixedOx256	利率
vLender	uint8	出借人ECDSA签名
rLender	bytes32	
sLender	bytes32	
vBorrower	uint8	借款人ECDSA签名
rBorrower	bytes32	
sBorrower	bytes32	

05

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资源消耗。

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，基本的数学库还是需要被列为库文件，既为了里面不时涉及的计算，主要也是为扩展做准备。EOS智能合约采用C++语言编写，eosio/eos-dev代码中包含本项目所需要的所有数学和字符串依赖库，不需要显式的编写。下面给出部分引用的代码片段。

05



```
5 #include <utility>
6 #include <vector>
7 #include <string>
8 #include <eosiolib/eosio.hpp>
9 #include <eosiolib/time.hpp>
10 #include <eosiolib/asset.hpp>
11 #include <eosiolib/contract.hpp>
12 #include <eosiolib/crypto.h>
13 #include <eosiolib/symbol.hpp>
14 #include <vector>
15 #include <string>
16 #include <eosio.token/eosio.token.hpp>
17 #include <eosiolib/eosio.hpp>
18
19 using namespace eosio;
20
21 using eosio::indexed_by;
22 using eosio::const_mem_fun;
23 using eosio::asset;
24 using eosio::permission_level;
25 using eosio::action;
26 using eosio::print;
27 using eosio::name;
```

5.2 原力协议智能合约

TheForceProtocol包括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。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。

借入合约代码片段

```
//@abi action
void loan::borrow(const account_name &borrower, const int32_t &lending_cycle,
                  const asset_type &token_a, const asset_type &token_b,
                  const int32_t &pledge_rate, const int32_t &interest_rate,
                  const int32_t &fee_rate, const int64_t &order_id) {

    auto order_itr = orders.find(order_id);
    eosio_assert(order_itr == orders.end(), "order already exists");

    eosio_assert(token_a.asset_.is_valid(), "invalid token-a");
    eosio_assert(token_a.asset_.amount > 0, "token-a must be positive quantity");

    ...//空间所限省略代码数行

    eosio_assert(lending_cycle >= 7, "lending-cycle must be greater than 7 days");

    require_auth(borrower);

    action(
        permission_level{borrower, N(active)},
        token_a.contract, N(transfer),
        std::make_tuple(borrower, _self, token_a.asset_, std::string(""))
    ).send();

    // Store new order
    auto new_order_itr = orders.emplace(borrower, [&](auto &order) {
        order.id = order_id;
        order.borrower = borrower;

        ...//空间所限省略代码数行

        order.deadline = eosio::time_point_sec(0);
        order.status = ORDER_STATUS_PENDING;
    });
}
```

外借合约代码片段

```
//@abi action
void loan::lend(const int64_t &order_id, const account_name &lender,
    const asset_type &token_b) {
    auto order_itr = orders.find(order_id);

    eosio_assert(order_itr != orders.end(), "order not found");
    eosio_assert(lender != order_itr->borrower, "cannot lend to self");
    eosio_assert(token_b.asset_ >= order_itr->token_b.asset_ + order_itr->token_b
        .asset_ * order_itr->fee_rate / 10000, "insufficient balance");

    ...//空间所限，省略代码

    require_auth(lender);

    asset token_b_fee = order_itr->token_b.asset_ * order_itr->fee_rate * 2 / 10000;
    asset token_b_to_borrower = order_itr->token_b.asset_ - token_b_fee / 2;

    action(
        permission_level{lender, N(active)},
        order_itr->token_b.contract, N(transfer),
        std::make_tuple(lender, platform, token_b_fee, std::string("")))
    .send();

    action(
        permission_level{lender, N(active)},
        order_itr->token_b.contract, N(transfer),
        std::make_tuple(lender, order_itr->borrower, token_b_to_borrower,
            std::string("")))
    .send();

    orders.modify(order_itr, _self, [&](auto &order) {
        auto deadline = now() + order.lending_cycle * 24 * 60 * 60;//in seconds
        //空间所限，省略部分代码片段
        order.status = ORDER_STATUS_ACCEPTED;
    });
}
```

还款合约代码片段

```
//@abi action
void loan::repay(const int64_t &order_id, const account_name &borrower,
                  const asset_type &token_b) {
    auto order_itr = orders.find(order_id);
    eosio_assert(order_itr != orders.end(), "order not found");
    eosio_assert(token_b.asset_.symbol == order_itr->token_b.asset_.symbol,
                 "invalid token-b");
    eosio_assert(order_itr->status == ORDER_STATUS_ACCEPTED, "operation not allowed");

    require_auth(borrower);
    asset total_to_pay = order_itr->token_b.asset_ + order_itr->token_b.asset_ ...
        * order_itr->interest_rate * order_itr->lending_cycle / 3650000;
    print("pay", (order_itr->token_b.asset_ * order_itr->interest_rate ...
        * order_itr->lending_cycle / 3650000));
    print(" pay", total_to_pay);

    eosio_assert(token_b.asset_ >= total_to_pay, "insufficient balance");

    ...//省略代码
    action(
        permission_level{borrower, N(active)},
        order_itr->token_b.contract, N(transfer),
        std::make_tuple(borrower, order_itr->lender, total_to_pay,
                       std::string ""))
    .send();

    action(
        permission_level{_self, N(active)},
        order_itr->token_a.contract, N(transfer),
        std::make_tuple(_self, order_itr->borrower, order_itr->token_a.asset_,
                       std::string ""))
    .send();
}

orders.modify(order_itr, _self, [&](auto &order) {
    order.status = ORDER_STATUS_COMPLETED;
});
```

强制还款合约代码片段

```
//@abi action
void loan::forcerepay(const int64_t &order_id, const asset_type &token_a) {
    auto order_itr = orders.find(order_id);
    eosio_assert(order_itr != orders.end(), "order not found");
    eosio_assert(order_itr->status == ORDER_STATUS_ACCEPTED, "invalid operation");
    auto time_now = time_point_sec(now()/*current time*/);

    eosio_assert(order_itr->deadline <= time_now,
                 "operation not allowed before deadline");

    eosio_assert(token_a.asset_.is_valid(), "invalid token-a");
    eosio_assert(token_a.contract == order_itr->token_a.contract, "invalid token");
    eosio_assert(token_a.asset_.amount > 0 && token_a.asset_ <= order_itr->token_a
                 .asset_, "exceed the range of token-a");
    ...//省略代码

    require_auth(_self);

    action(
        permission_level{_self, N(active)},
        order_itr->token_a.contract, N(transfer),
        std::make_tuple(_self, order_itr->lender, token_a.asset_, std::string("")))
    .send();

    action(
        permission_level{_self, N(active)},
        order_itr->token_a.contract, N(transfer),
        std::make_tuple(_self, order_itr->borrower, order_itr->token_a
                       .asset_ - token_a.asset_, std::string("")))
    .send();

    orders.modify(order_itr, 0, [&](auto &order) {
        order.status = ORDER_STATUS_FORCEREPAYED;
    });
}
```

平仓代码片段

```
//@abi action
void loan::closeposition(const int64_t &order_id, const account_name &by) {
    auto order_itr = orders.find(order_id);
    eosio_assert(order_itr != orders.end(), "order not found");
    eosio_assert(order_itr->status == ORDER_STATUS_ACCEPTED, "invalid operation");

    auto time_now = time_point_sec(now()/*current time*/);

    if (order_itr->deadline > time_now) {
        eosio_assert(by == _self, "only owner can operate before deadline");
    } else {
        eosio_assert(by == order_itr->lender || by == _self,
                    "only lender or owner of this contract can operate");
    }
    ...//省略代码

    require_auth(by);

    action(
        permission_level{_self, N(active)},
        order_itr->token_a.contract, N(transfer),
        std::make_tuple(_self, order_itr->lender, order_itr->token_a.asset_,
                       std::string ""))
    .send();

    orders.modify(order_itr, 0, [&](auto &order) {
        order.status = ORDER_STATUS_LIQUIDATED;
    });
}
```

取消订单代码片段

```
//@abi action
void loan::cancelorder(const int64_t &order_id, const account_name &by) {
    auto order_itr = orders.find(order_id);
    eosio_assert(order_itr != orders.end(), "order not found");
    eosio_assert(by == order_itr->borrower || by == _self, "only borrower or contract can operate");
    eosio_assert(order_itr->deadline == eosio::time_point_sec(0) && order_itr->lender == N(nullptr) &&
                order_itr->status == ORDER_STATUS_PENDING, "order is already accepted");
    require_auth(by);

    action(
        permission_level{_self, N(active)},
        order_itr->token_a.contract, N(transfer),
        std::make_tuple(_self, order_itr->borrower, order_itr->token_a.asset_, std::string ""))
    .send();
    orders.erase(order_itr);
}
```

5.3 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。

5.4 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	借款周期内价格波动幅度，当前代币价格，置信水平
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
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

06 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

```
/****************************************************************************
 * MakeLendOrder() 接口可被出借人在超级节点上创建订单时使用*
 ****************************************************************************/
TheForceProtocol.EOS.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.EOS.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，包括加入联盟链的步骤*/
TheForceProtocol.EOS.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  
)
```

07

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，用户可安装谷歌浏览器Chrome和Scatter插件，使用户不需要安装EOS全节点即可进行相应的借贷操作。

需注意，下述有关币币贷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。

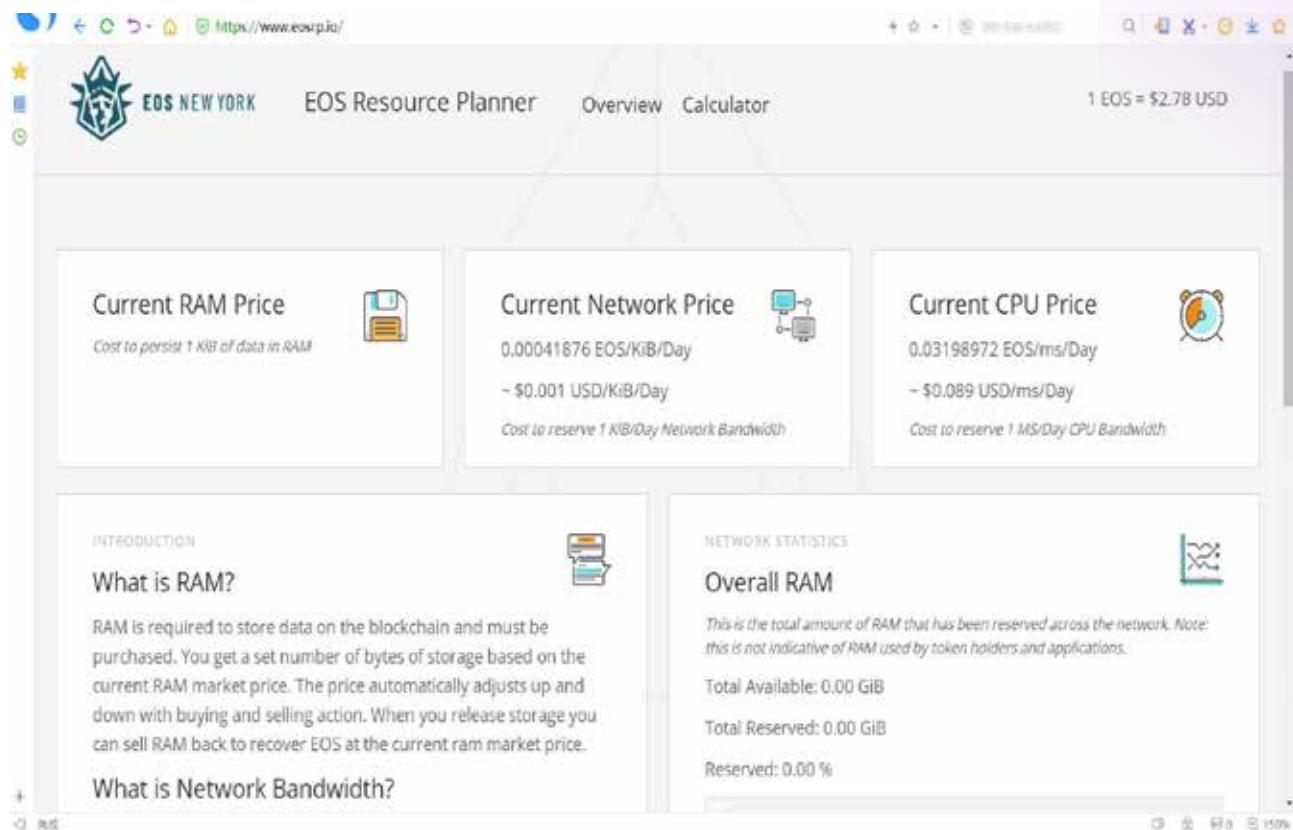
7.1 准备工作

使用本DAPP需要使用Google Chrome浏览器安装EOS钱包Scatter插件，具体步骤参照官方说明文档执行。

07

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。EOS用户可以在下述网站查询EOS抵押和赎回价格：<https://www.eosrp.io/>。然后可从钱包抵押的EOS通证获取分配CPU、内存和带宽。然后可进入DAPP首页。



7.2.1 首页欢迎界面

进入币币贷DAPP, 首先将看到平台介绍相关内容, 如下页所示:



点击“立即使用”按钮，将跳转到登录/注册界面，如下图所示。



输入手机号并识别页面中的文字验证码，点击“发送验证码”。输入手机短信接收的验证码。点击“登录或注册”按钮。页面将跳转到币币贷DAPP首页，如下页所示。



如右图所示，币币贷平台首页上半部分将呈现平台近期的热门活动、通知和公告，并在下方精选部分热门借币项目为用户做出推荐。

出借用户可浏览“热门推荐”内容，选择合适项目直接点击后面的“立即出借”，即可跳转到立即出借界面，如下页图中所示。



用户也可以在首页直接点击下方的“出借”标签，进入目前有借币需求的订单列表，如右图所示。用户可以点击“立即出借”，进入出借界面，可以达到跟首页点击相同的效果，如上图所示。

在出借界面，用户需要阅读《币币贷借币规则》、勾选“我已经阅读并同意《币币贷借币规则》”，点击“立即出借”按钮即可完成出借过程。





用户填写借币数量、借币利率、质押数量，系统自动反馈应还本息和实际到账情况。

08

原力协议原生代币EFOR /

原力协议将发行基于EOS的EFOR代币。在原力协议生态系统内，EFOR代币将发挥重要作用。

8.1 EFOR代币用途

08

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，EFOR代币将发挥以下

8.1.1交易手续费扣

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EFOR支付手续费。为了防止超级节点在收取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EFOR的收入，以此避免超级节点集中抛售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。

8.1.3 借贷挖矿

为了促进用户进行抵押借贷，我们保留对EFOR代币进行交易挖矿的策略，将在平台上线运转后的适当时间，由开发小组向社区治理机制提交关于交易挖矿的详细方案。方案内容将包括交易挖矿总量、规则、时间等一系列关键的因素。经由社区讨论、决策后进行投票，方案通过后将按计划执行。

8.1.4 抵押物

我们会在适当的时机推动EFOR上线全球的主流交易所，预计EFOR将成为原力协议生态内主要的抵押物，受到借款人和投资者的青睐。EFOR在作为借贷抵押物的时候将享受优惠的质押率和手续费抵扣，该部分内容可详细参考“交易手续费抵扣”和“提高借款质押率”章节。

8.1.5 提高借款质押率

如果用户使用EFOR作为借款抵押物，则可以在基础质押率之上，享受到一定的上浮比例。用户可以获得更多的借款，这一设定可以促进EFOR的使用，增加用户对于原力协议服务的黏性。质押比率的具体数值在系统上线初期将有一个预设值，随着借贷订单的不断累积，我们可能会将EFOR的质押比率修改到更加合理的数值，并提交社区投票通过。

8.1.6 社区治理

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EFOR，在递交提案时需要向专门的智能合约抵押一定数量的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，EFOR的持币人将对提案内容进行投票，所有参与了锁仓的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。

08

8.1.7 资产上链抵押

原力协议作为去中心化的抵押借贷平台，欢迎所有符合社区用户利益的加密资产上线进行借贷交易。然而，为了防止部分“空气币”、“骗子币”上线平台，占有社区的公共计算资源。我们也将设置一套机制，所有上线原力协议的抵押物，除BTC、ETH、XRP、BCH、EOS、XLM、LTC、ADA、XMR、TRX、DASH、BNB、NEO、ONT、ETC、XEM、ZEC、USDT、USDC、TUSD、GUSD、PAX等主流币种外，均需要由加密资产的管理方持有并抵押一部分的EFOR代币，通过社区投票之后才能上线作为抵押物。我们必须确保所有上线原力协议平台的加密资产都是符合社区利益的，一切可能损害社区利益的加密资产都将被禁止上线作为抵押物。

8.2 EFOR代币分配计划

EFOR代币总量10亿个，且永不增发。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20%，战略投资者及社区捐赠占35%。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上交易所开始锁仓3年，上交易所后12个月释放30%，24个月后释放30%，36个月后释放40%。代币分配比例如下图所示。

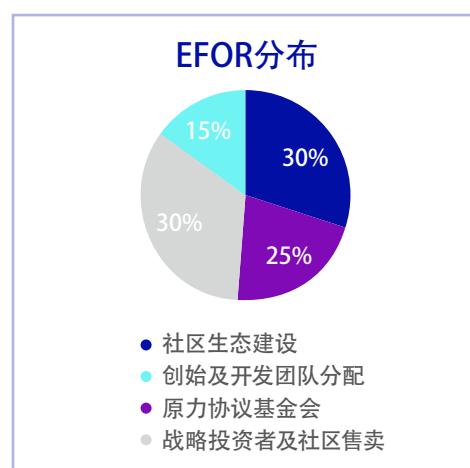
8.2.1 社区生态建设

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、EFOR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。

EFOR代币分配比例



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35%的代币回馈战略投资者及社区成员的资助。

09

原力协议项目 /

9.1 项目团队

/ Allen ——联合创始人 CEO

原EnactusChina天津山东河北地区大区经理, 乾盛汇资本董事总经理, Zonff Capital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, 并先后在分期、现金贷、车贷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, AlphaCoin Fund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、Box、Dcc等十多个项目, 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。在原力协议牵头财务、融资及对外合作。

/ 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 COO & 原力稳定币总裁

清华大学硕士, 2012年起接触比特币挖矿, 2017年初开始系统研究区块链技术与加密数字资产ICO, 对区块链产业潜力和未来方向有独到见解。他曾先后投资了EOS、Filecoin、Cybermiles等优质项目, 观察并参与多个加密数字资产项目的社区建设与运营, 拥有丰富的社区组织运作经验。他还是XDAG中国社区开发及运营的主要成员, 币圈早期自媒体“BW投资”创始人。在原力协议牵头运营、媒体宣传和战略研究。

/ 张琳波——联合创始人 首席科学家

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。

/ 许超——联合创始人 商务总监

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。在原力协议负责产品、商业模式和对外合作。

/ 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。

/ 戴邵鹏——区块链开发工程师

中科院计算技术研究所博士生，主要研究数据挖掘、机器学习及Benchmark等领域，负责源码编写。

/ 王捷

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

/ 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。

/ 李忠楠

和创金服产品合伙人，先后带领团队推出优分期、简单借款及和创汇金等产品。优分期作为首批大学生分期购物商城，16年综合行业排行第三名，日最大交易额过亿。前美团产品设计专家，负责美团所有移动平台产品设计工作，制定移动产品设计体系与标准。早年，在腾讯负责QQ产品和阿里负责天猫移动端产品的工作经历，使得李忠楠先生对产品设计和研发管理具有丰富的经验。

/ 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。

/ 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。

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。其多层次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。

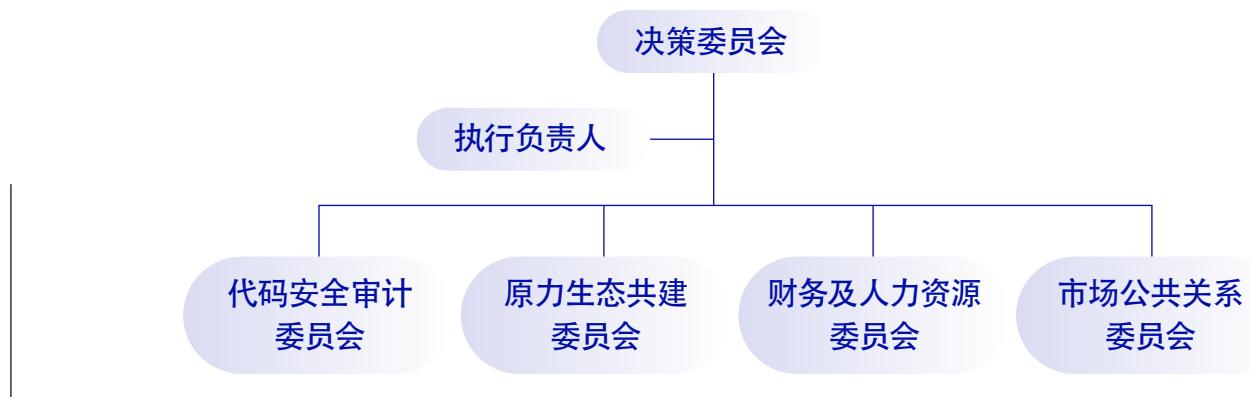
和法链联合实验室

原力协议与人民大学法链实验室联合成立和法连labs,专注于区块链法律合规,监管政策研究,利用区块链技术实现对区块链行业的监管和治理。作为中国互联网金融监管中心的学术指导单位,积极参与金融科技,区块链金融等领域法律监管研究。

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。



➤ **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:**

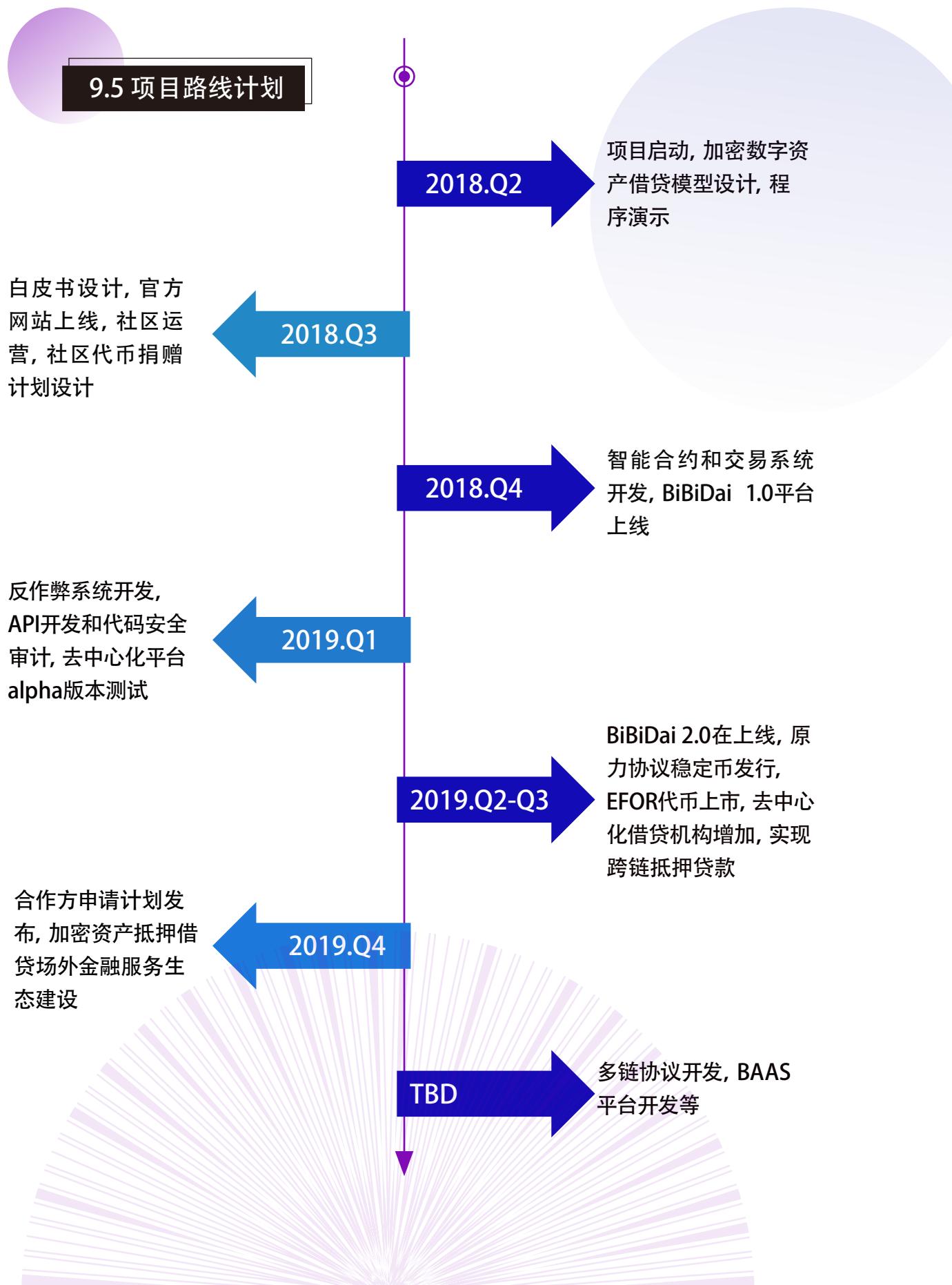
负责原力生态发展和合作伙伴建设, 委员会将使用募集资金开展生态建设和商务合作, 激励更多地开发者基于原力协议构建应用, 更多地把潜在合作者纳入到原力的生态体系中来。

➤ **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。



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，仅作为区块链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EOS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EOS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EOS的借贷存在一定程度的不确定性，因为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 /

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EFOR代币而获得任何直接或者间接收益或者分红。

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

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

系统更新和开源社区/

12.1 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再改变。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。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EFOR代币持有者的财产安全。

12

12.2 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:

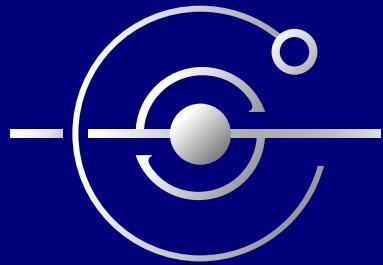
<https://github.com/TheForceProtocol/TheForceProtocol.git>

在共享社区中, 所有的智能合约都将存储在“\contracts”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, 所有API文件都用JavaScript语言编写。此外, 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, 尚无代码上传。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。

THANKS!



THE FORCE PROTOCOL

原 力 协 议

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。

ThinkBit

ThinkBit是下一代数字资产交易平台，让你方便地交易、储蓄数字资产，合规、专业、贴心。全球首家采用100%冷钱包技术的交易所，极致安全。原力协议与ThinkBit在区块链技术、数字资产管理和平稳币方面展开合作。

eArk——绿色区块链

绿色链融合卫星通信网、智能物联网、信息互联网，构建全球绿色价值网络，实现环境数据资产的高效共识、可信流通、全球交易，提供人类环境保护的终极解决方案。原力协议与绿色链结合，为环境加密数字资产的提供高效流动性。

和创金服

和创金服是一家专注于个人和小微金融创新的科技企业。和创金服致力于数据工程、人工智能、云计算等技术在普惠金融领域的前瞻性研究与深度化应用，秉承“让金融没有边界”的愿景，持续努力实现“为人人提供平等而有效的金融服务”的使命，为广大个人用户及小微企业提供平等而便捷的互联网金融信息服务，发挥科技创新对经济社会发展的引领作用。



选择相应订单，可选择“立即还款”和“补仓”如右图所示。

点击左图中“我要借币”确认借款。然后可以在“我的订单”中查询“借币”和“出借”情况，如下图所示。



补仓 立即还款