DM.IO 技术白皮书

**草案：2018 年 5 月 3 日 **

摘要：

DM.IO软件通过新的区块链架构，最终实现去中心化应用的纵向和横向性能扩展。这是通过创建一个类似操作系统的架构来实现的，可以在上面构建应用。该软件提供了帐户，身份验证，数据库，异步通信以及跨越多个CPU内核或集群的程序调度。该技术的最终形式是一个区块链架构，在治理区块链的场景下，可以最终扩展，足以支持每秒数万笔交易，消除用户费用，实现去中心化应用的轻松快速地部署和维护。本白皮书描述了区块链的行业应用案例，以推动形成新的区块链结构原理。另外，本白皮书根据这些应用案例，列出了针对区块链的基本需求和高级体系结构。本白皮书中所呈现的呆马链（DM）描述了区块链结构的演进，并作为商对商（B2B）、商对客（B2C）交易的一种协议。当在相同的网络中发生业务竞争时，呆马链允许在符合规则的前提下支持各种需求。本白皮书后面所描述的内容包括智能合约、数字资产、记录存储库、去中心化的一致网络和密码安全。最后，本白皮书还描述了区块链的主要产品、各行业对性能的要求、身份验证、隐私和机密交易和可插拔的共识模型。

- 密钥被盗后的恢复
- 通信延迟优化
- 通证模型和资源使用
  - 客观和主观度量
  - 接收方支付
  - 授权能力
  - 将交易成本与通证价值区分开
  - 状态存储成本
  - 出块奖励
  - 工作提案系统
  - 治理
  - 冻结账户
  - 更改账户代码
- 目标
- 基于呆马链的行业应用案例
  - 金融资产处置
  - 协作
  - 保险防欺诈
  - 大数据安全
  - 主数据管理
  - 共享经济和互联网
  - 合同及发票防伪
  - 公益追溯
- 典型需求
  - 私下交易和保密合约
  - 身份识别和审计
  - 互操作能力
  - 可移植性
- 脚本 & 虚拟机
- 体系结构
  - 身份识别服务
  - 策略管理服务
  - 区块链服务
  - 智能合约服务
- 应用编程接口
- 网络拓扑
- 区块链4.0跨链共识机制
  - 跨链通讯延迟
  - 完成性证明
- 呆马区块链浏览器
- 呆马发行机制
- 团队主要成员介绍
- 技术方面
  - 区块链
  - 组件模型
  - 服务平台
- 结论

背 景

区块链是一种新兴技术模式。这种技术模式能够快速改进银行、供应链以及其他的交易网络，在降低与业务运营相关的成本和风险的同时，能创造新的创新和增长机会。自2009年比特币在交易领域迅速崛起以来，许多商业组织和行业机构投入了大量资源来研究比特币的底层技术，从而传播这种广受欢迎但又颇具争议的加密币。

区块链是一种点对点（P2P）分布式账本技术。由于能够有效、安全地支持资产的发行、交易、管理和服务，区块链首先在金融行业得到了支持。在不要求中心控制点的情况下，区块链技术能够容易地建立成本合适的商业网络，这与记录系统（SoR）的生态中需要每一个成员维护自己的账本系统、审核与其他成员的交易进展形成了鲜明的对比，因为后者即低效、又昂贵，而且经常是非标准化的内部操作流程。

由于共享账本概念得到了商业世界的支持，区块链智能合约也引起了更多行业的关注。智能合约是商业规则的集合，是部署在区块链上，并由一组利益相关方共享和共同验证。在自动化商业流程中，智能合约是非常有用的，而且诚信可靠，允许所有利益相关方共同处理和验证合约规则。

比特币及其他加密货币的设计是完全开放、去中心化和非授权的：任何人在没有确定身份的情况下都能参与，而且只需要贡献一点时间完成计算周期就行。在区块链的比特币模型中，没有中心机构来发放许可，因为网络是非授权的。由于需要无数的工作量计算来证明，比特币的运行是昂贵的。

呆马链是对传统区块链模型的革新。即使希望授权的区块链作为起点，但呆马链通过提供一个模型，在某种程度上是允许创建授权的和非授权的区块链。另外，呆马链通过一个提供针对身份识别、可审计和隐私的安全和健壮模型，使得缩短计算周期、提高规模效率和响应各个行业的应用需求成为可能。

区块链应用程序的要求

为了得到广泛使用，区块链上的应用程序要求区块链平台足够灵活，能够满足如下要求：

支持百万级别的TPS

想要同阿里巴巴，腾讯,google和facebook这些企业竞争，需要能够处理数千万日活跃用户的区块链技术。 在某些情况下，除非用户数足够庞大，否则应用程序可能无法正常运作，因此，能够应对大量用户的平台至关重要。

轻松升级和故障修复

基于区块链构建应用程序的企业，需要区块链平台具备灵活性，可以为其应用添加新特性来增强完善。区块链平台必须对软件和智能合约的升级提供支持。

所有的非小型的软件都可能会有缺陷，即使是用了最严格的形式验证也是如此。当bug不可避免出现时，区块链平台必须足够健壮，能够修复这些bug。

低延迟

及时的反馈是良好用户体验的基础。延迟时间如果超过了几秒钟，会大大影响用户体验，严重降低基于区块链的应用相对于现有的非区块链应用的竞争力。区块链平台应当支持低延迟的交易。

串行性能

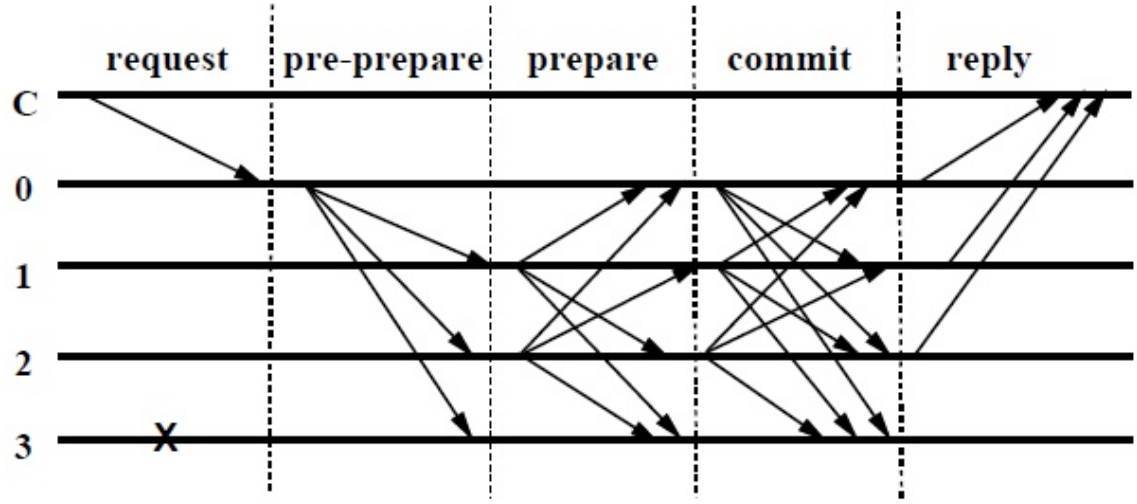
有些应用程序由于必须顺序执行命令，从而无法用并行算法进行实现。诸如交易所之类的应用经常需要足够的串行操作处理性能，以应对大量的交易。因此，区块链需要提供强大的串行性能支持。

并行性能

大型应用程序需要在多个CPU和计算机之间分配工作负载。

共识算法(PBFT)

DM.IO软件采用了目前最优的去中心化共识算法 — 实用拜占庭容错算法Practical Byzantine Fault Tolerance (PBFT).。 这是一种基于消息传递的一致性算法，算法经过三个阶段达成一致性，这些阶段可能因为失败而重复进行。
假设节点总数为3f+1，f为拜赞庭错误节点：
1、当节点发现leader作恶时，通过算法选举其他的replica为leader。
2、leader通过pre-prepare 消息把它选择的 value广播给其他replica节点，其他的replica节点如果接受则发送prepare，如果失败则不发送。
3、一旦2f个节点接受prepare消息，则节点发送commit消息。
4、当2f+1个节点接受commit消息后，代表该value值被确定 。 如下图表示了4个节点，0为leader，同时节点3为fault节点，该节点不响应和发出任何消息。最终节点状态达到commited时，表示该轮共识成功达成。



优点：上述共识算法都脱离不了币的存在，系统的正常运转必须有币的奖励机制，系统的安全性实际上是由系统币的持有者维护保证。当我们区块链系统实际运用到商业应用时，由其承载的资产价值可能远远超出系统发行的币的价值，如果由币的持有者保证系统的安全及稳定性将是不可靠的 。
1）系统运转可以脱离币的存在，pbft算法共识各节点由业务的参与方或者监管方组成，安全性与稳定性由业务相关方保证。
2）共识的时延大约在2~5秒钟，达到商用实时处理的要求。
3）共识效率高，可满足高频交易量的需求。

根据这一算法，在使用DM.IO软件构建的区块链上持有通证的人，可以通过一个持续进行的投票系统来选择区块生产者。任何人都可以选择参加区块生产，只要能够说服通证持有人为其投票，就会有机会参与区块生产。

DM.IO软件可以让区块每0.01秒生成一个。任何时刻，只有一个生产者被授权产生区块。如果在计划的某个时间内没有成功出块，则跳过该块。如果有一个或更多的区块被跳过，则在区块链上会有0.01s或者更久的空白。

使用DM.IO软件，区块的产生是以30个区块(每个出块者六个区块，乘以5个出块者)为一个周期。在每个出块周期开始时，会根据通证持有人所投票数选出5个区块生产者。被选中的区块生产者的顺序会根据5个区块生产者的同意，制定出块顺序的安排。

如果出块者错过了一个块，并且在最近24小时内没有产生任何块，则这个出块者将被剔除在考虑范围之外，直到他们通知区块链可以重新开始产生区块。这确保了网络的顺利运行，把被证明为不可靠的区块生产者排除在出块排程之外，通过这一方式使得错过区块的数量最小化。

在正常情况下，PBFT块链不会经历任何分叉，因为区块生产者并非竞争关系，他们合作产生区块。如果有区块分叉，共识将自动切换到最长链。这一方式之所以有效，是因为区块链分叉上增加区块的速度，与具有相同共识的区块生产者的比例直接相关。换句话说，具有更多生产者的区块链长度将比具有较少生产者的区块链增长速度更快，因为，有更多生产者的区块链分叉上，丢块更少。

此外，没有块生产者可以同时在两个区块链分叉上生产块。如果一个块生产者发现这么做了，就可能被投票出局。这类双重生产的密码学证据，也可能会被用来自动移除作恶者。

与传统的POS/POW算法相比较，DM采用拜占庭容错算法(Practical Byzantine Fault Tolerance) ，所有的出块者都要对所有区块签名，以此来确保在同一时间戳或者同一区块高度上，没有区块生产者能够同时在两个区块上签名。一个区块有了5个区块生产者的签名，该区块就被认为是不可逆的。任一拜占庭区块生产者如果想在同一时间戳或者同一区块高度的两个区块上签名，就不得不留下密码学证据。在这一模式下，一秒之内就可以达成不可逆的共识。

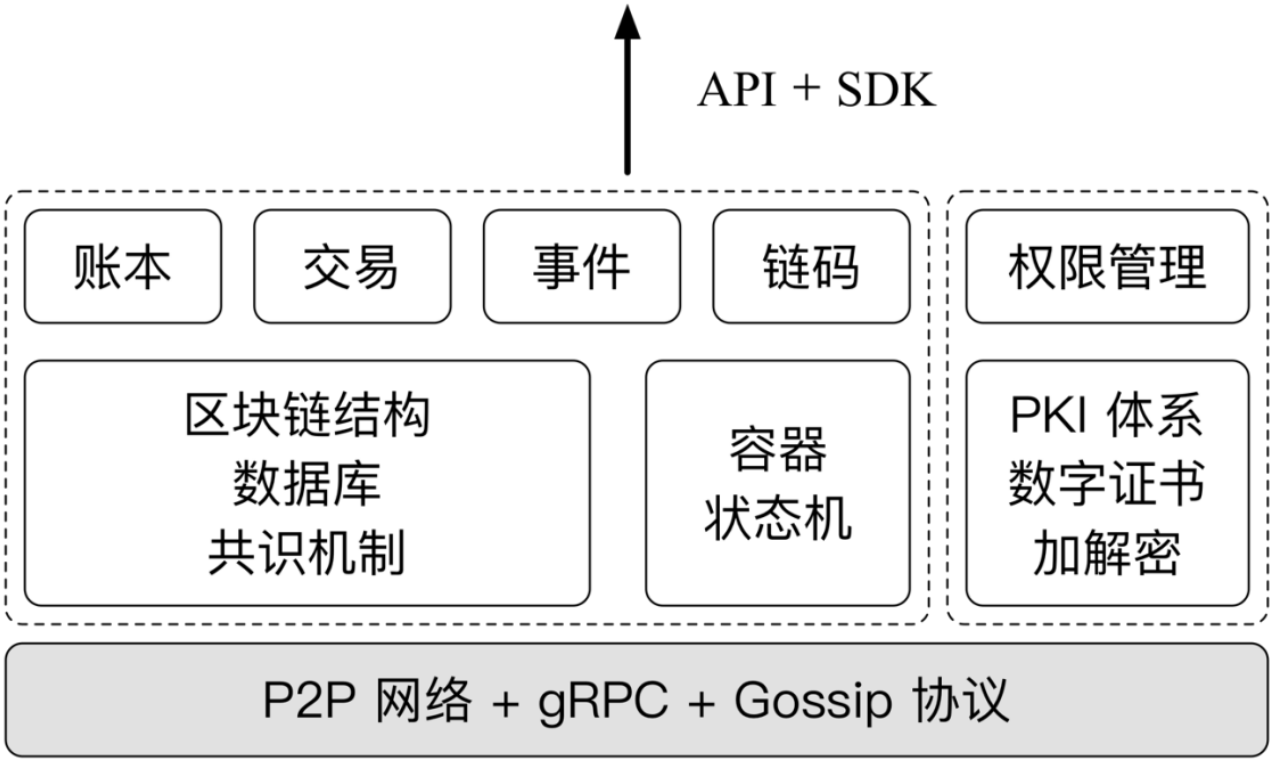
架构

作为一种新兴技术，现有的区块链实现不能满足商业交易中各种复杂的应用需求。可扩展性挑战、机密和隐私交易支持的缺失以及其他限制，使得很多关键业务应用不能投入使用。

为了及时部署弹性平台、支持跨行业的应用需求，需要轻量级、模块化和通过配置插入各种组件（交易验证器、阻止协商一致等）支持可扩展的平台。为了满足今天市场的各种需求，呆马链的设计以行业应用为重点，解决了现有技术的缺点，并拓展了业内先行者原来的工作。

目前，超级账本DM.IO软件架构上核心特性主要包括：

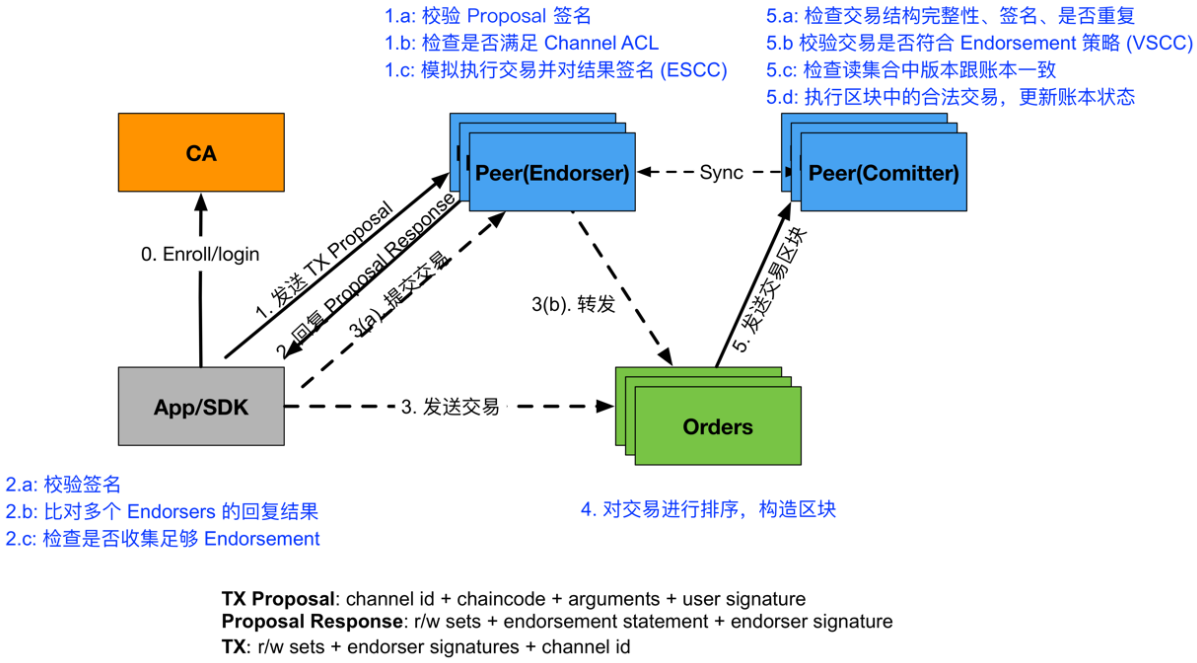
解耦了原子排序环节与其他复杂处理环节，消除了网络处理瓶颈，提高可扩展性； 解耦交易处理节点的逻辑角色为背书节点（Endorser）、确认节点（Committer），可以根据负载进行灵活部署； 加强了身份证书管理服务，作为单独的DM.IO CA项目，提供更多功能； 支持多通道特性，不同通道之间的数据彼此隔离，提高隔离安全性； 支持可拔插的架构，包括共识、权限管理、加解密、账本机制都模块，支持多种类型； 引入系统链码来实现区块链系统的处理，支持可编程和第三方实现。 超级账本DM.IO的整体架构如下图所示。



DM.IO为应用提供了gRPC API，以及封装API的SDK供应用调用。应用可以通过SDK访问DM.IO网络中的多种资源，包括账本、交易、链码、事件、权限管理等。应用开发者只需要跟这些资源打交道即可，无需关心如何实现。其中，账本是最核心的结构，记录应用信息，应用则通过发起交易来向账本中记录数据。交易执行的逻辑通过链码来承载。整个网络运行中发生的事件可以被应用访问，以触发外部流程甚至其他系统。权限管理则负责整个过程中的访问控制。账本和交易进一步地依赖核心的区块链结构、数据库、共识机制等技术；链码则依赖容器、状态机等技术；权限管理利用了已有的PKI体系、数字证书、加解密算法等诸多安全技术。底层由多个节点组成P2P网络，通过gRPC通道进行交互，利用Gossip协议进行同步。

层次化结构提高了架构的可扩展和可插拔性，方便开发者以模块为单位进行开发。

超级账本DM.IO根据交易过程中不同环节的功能，在逻辑上将节点角色解耦为Endorser和Committer，让不同类型节点可以关注处理不同类型的工作负载。典型的交易处理过程如下图所示。



在整个交易过程中，各个组件的功能主要为：

- 客户端（App）：客户端应用使用SDK来跟DM.IO网络打交道。首先，客户端从CA获取合法的身份证书来加入到网络内的应用通道。发起正式交易前，需要先构造交易提案（Proposal）提交给Endorser进行背书（通过EndorserClient提供的ProcessProposal(ctx context.Context, signedProp *pb.SignedProposal)* (pb.ProposalResponse,error)接口）；客户端收集到足够（背书策略决定）的背书支持后可以利用背书构造一个合法的交易请求，发给Orderer进行排序（通过BroadcastClient提供的Send(env *cb.Envelope)error接口）处理。客户端还可以通过事件机制来监听网络中消息，来获知交易是否被成功接收。命令行客户端的主要实现代码在peer/chaincode目录下。

- Endorser节点：主要提供ProcessProposal(ctx context.Context,signedProp *pb.SignedProposal)* (pb.ProposalResponse,error)方法（代码在core/endorser/endorser.go文件）供客户端调用，完成对交易提案的背书（目前主要是签名）处理。收到来自客户端的交易提案后，首先进行合法性和ACL权限检查，检查通过则模拟运行交易，对交易导致的状态变化（以读写集形式记录，包括所读状态的键和版本，所写状态的键值）进行背书并返回结果给客户端。注意网络中可以只有部分节点担任Endorser角色。主要代码在core/endorser目录下;

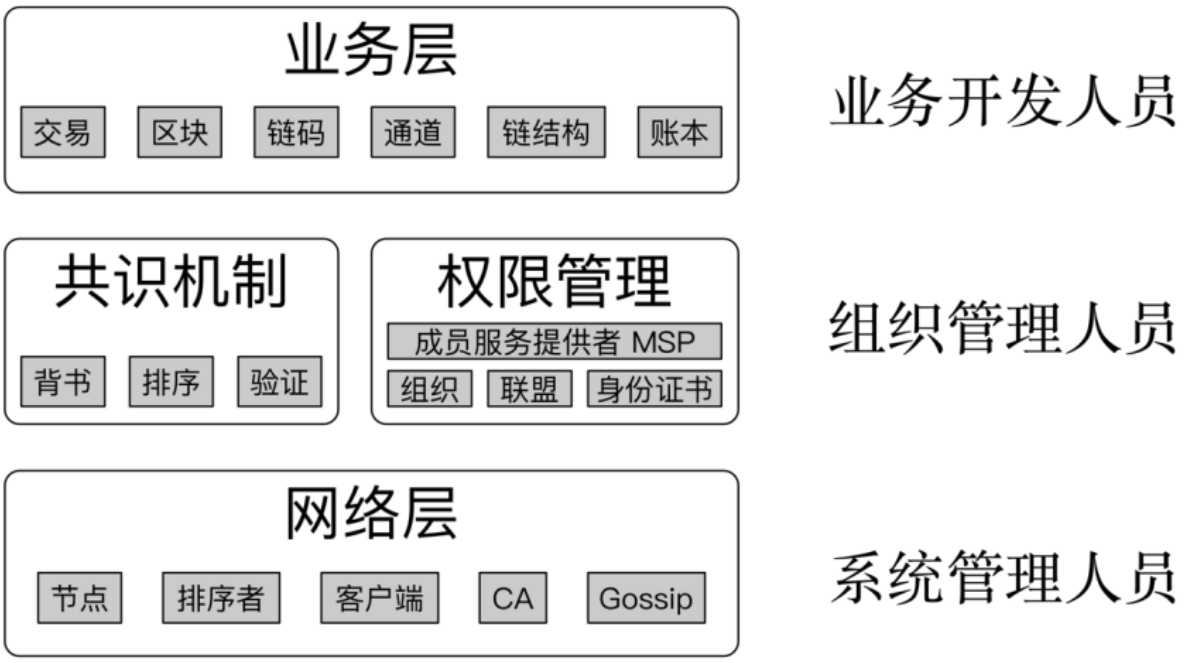
- Committer节点：负责维护区块链和账本结构（包括状态DB、历史DB、索引DB等）。该节点会定期地从Orderer获取排序后的批量交易区块结构，对这些交易进行落盘前的最终检查（包括交易消息结构、签名完整性、是否重复、读写集合版本是否匹配等）。检查通过后执行合法的交易，将结果写入账本，同时构造新的区块，更新区块中BlockMetadata[2]（TRANSACTIONS_FILTER）记录交易是否合法等信息。同一个物理节点可以仅作为Committer角色运行，也可以同时担任Endorser和Committer这两种角色。主要实现代码在core/committer目录下;

- Orderer：仅负责排序。为网络中所有合法交易进行全局排序，并将一批排序后的交易组合生成区块结构。Orderer一般不需要跟账本和交易内容直接打交道。主要实现代码在orderer目录下。对外主要提供Broadcast(srv ab.AtomicBroadcast*BroadcastServer)error和Deliver(srv ab.AtomicBroadcast*DeliverServer)error两个RPC方法（代码在orderer/server.go文件）；

- CA：负责网络中所有证书的管理（分发、撤销等），实现标准的PKI架构。主要代码在单独的DM.IO项目中。CA在签发证书后，自身不参与到网络中的交易过程。
  由于跨链通信，我们可以根据需要在不同链之间分配工作量。代币可以可靠并且安全的在不同链之间转移。由于相同（或不同）区块生产者并行运行1000条链，我们可以看到每秒数万的交易。

核心概念与组件

超级账本DM.IO采用了模块化功能设计，整体的功能模块结构如下图所示。



超级账本DM.IO面向不同的开发人员提供了不同层面的功能，自下而上可以分为三层：

网络层：面向系统管理人员。实现P2P网络，提供底层构建区块链网络的基本能力，包括代表不同角色的节点和服务; 共识机制和权限管理：面向联盟和组织的管理人员。基于网络层的连通，实现共识机制和权限管理，提供分布式账本的基础; 业务层：面向业务应用开发人员。基于分布式账本，支持链码、交易等跟业务相关的功能模块，提供更高一层的应用开发支持。 下面介绍网络层相关组件的功能和作用。

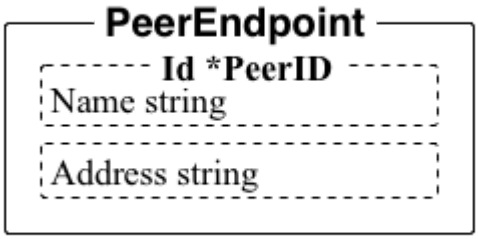
网络层相关组件

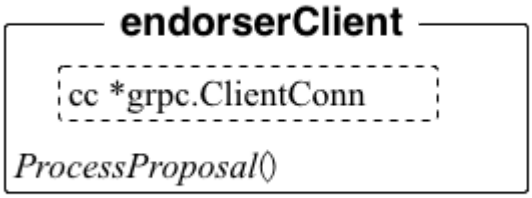
网络层通过软、硬件设备，实现了对分布式账本结构的连通支持，包括节点、排序者、客户端等参与角色，还包括成员身份管理、Gossip协议等支持组件。

节点（Peer）的概念最早来自P2P分布式网络，意味着在网络中担任一定职能的服务或软件。节点功能可能是对等一致的，也可能是分工合作的。在超级账本DM.IO网络中，Peer意味着在网络中负责接受交易请求、维护一致账本的各个DM.IO-peer实例。这些实例可能运行在裸机、虚拟机甚至容器中。节点之间彼此通过gRPC消息进行通信。按照功能角色划分，Peer可以包括三种类型：

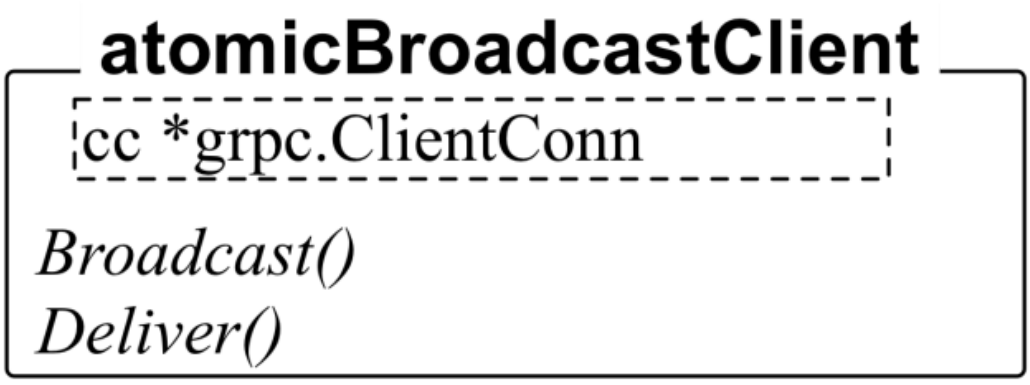
Endorser（背书节点）：负责对来自客户端的交易提案进行检查和背书； Committer（确认节点）：负责检查交易请求，执行交易并维护区块链和账本结构； Submitter（提交节点）：负责接收交易，转发给排序者，目前未单独出现。 这些角色是功能上的划分，彼此并不相互排斥。一般情况下，网络中所有节点都具备Committer功能；部分节点具有Endorser功能；Submitter功能则往往集成在客户端（SDK）进行实现。

Peer节点相关的主要数据结构包括PeerEndpoint和endorserClient。前者代表一个Peer节点在网络中的接入端点；后者实现EndorserClient接口，代表连接到Peer节点的客户端句柄，提供对Endorser角色实现的ProcessProposal(ctx context.Context,signedProp *pb.SignedProposal)(*pb.ProposalResponse, error)方法的访问。如下图所示。





排序者（Orderer），或称为排序节点，负责对所收到的交易在网络中进行全局排序。Orderer主要提供了Broadcast(srv ab.AtomicBroadcast*BroadcastServer) error和Deliver(srv ab.AtomicBroadcast*DeliverServer) error两个接口。前者代表客户端将数据（交易）发给Orderer，后者代表从Orderer获取到排序后构造的区块结构。客户端可以使用atomicBroadcastClient结构访问这两个接口。atomicBroadcastClient结构如下图所示，维持了一个gRPC的双向通道。



Orderer可以支持多通道。不同通道之间彼此隔离，通道内交易相关信息将仅发往加入到通道内的Peer（同样基于gRPC消息），从而提高隐私性和安全性。在目前的设计中，所有的交易信息都会从Orderer经过，因此，Orderer节点在网络中必须处于可靠、可信的地位。

从功能上看，Orderer的目的是对网络中的交易分配全局唯一的序号，实际上并不需要交易相关的具体数据内容。因此为了进一步提高隐私性，发往Orderer的可以不是完整的交易数据，而是部分信息，比如交易加密处理后的结果，或者仅仅是交易的Hash值、Id信息等。这些改进设计会降低对Orderer节点可靠性和安全性的需求。

客户端是用户和应用跟区块链网络打交道的桥梁。客户端主要包括两大职能：

操作DM.IO网络：包括更新网络配置、启停节点等； 操作运行在网络中的链码：包括安装、实例化、发起交易调用链码等。 这些操作需要跟Peer节点和Orderer节点打交道。特别是链码实例化、交易等涉及到共识的操作，需要跟Orderer交互，因此，客户端往往也需要具备Submitter的能力。网络中的Peer和Orderer等节点则对应提供了gRPC远程服务访问接口，供客户端进行调用。目前，除了基于命令行的客户端之外，超级账本DM.IO已经拥有了多种语言的SDK。这些SDK封装了对底层gRPC接口的调用，可以提供更完善的客户端和开发支持，包括Node.Js、Python、Java、Go等多种实现。

CA节点（DM.IO-CA）负责对DM.IO网络中的成员身份进行管理。DM.IO网络目前采用数字证书机制来实现对身份的鉴别和权限控制，CA节点则实现了PKI服务，主要负责对身份证书进行管理，包括生成、撤销等。需要注意的是，CA节点可以提前签发身份证书，发送给对应的成员实体，这些实体在部署证书后即可访问网络中的各项资源。后续访问过程中，实体无须再次向CA节点进行请求。因此，CA节点的处理过程跟网络中交易的处理过程是完全解耦开的，不会造成性能瓶颈。

Gossip 数据传输协议

DM.IO网络中的节点之间通过Gossip协议来进行状态同步和数据分发。Gossip协议是P2P领域的常见协议，用于进行网络内多个节点之间的数据分发或信息交换。由于其设计简单，容易实现，同时容错性比较高，而被广泛应用到了许多分布式系统，例如Cassandra采用它来实现集群失败检测和负载均衡。Gossip协议的基本思想十分简单，数据发送方从网络中随机选取若干节点，将数据发送过去；接收方重复这一过程（往往只选择发送方之外节点进行传播）。这一过程持续下去，网络中所有节点最终（时间复杂度为节点总个数的对数）都会达到一致。数据传输的方向可以是发送方发送或获取方拉取。

Gossip协议

Peer利用gossip以可扩展的方式广播分类帐和channel数据。Gossip消息是连续的，channel上的每个peer都在不断接收来自多个peer的当前和一致的分类帐数据。每个gossip消息被签名，从而拜占庭人员发送伪造的消息会容易地识别出来，并且将消息分发到不想要的目标以被阻止。受延迟，网络分区或导致丢失的块的其他因素影响，peer最终将通过联系拥有这些丢失块的peer将同步到当前分类帐状态。

基于Gossip-based数据传播协议在Hyperledger Fabric网络上执行三个主要功能：
管理peer发现和通道成员资格，通过不断识别可用成员peer，并最终检测已脱机的peer。在channel上的所有peer上传播分类帐数据。任何与channel其他部分不同步的数据都可以通过复制正确的数据来识别缺失的块和同步自身。 通过允许对账本数据的点对点状态传输更新，使新连接的peer达到速度要求。 Gossip-based的广播通过peer接收来自该channel上的其他peer的消息，然后将这些消息转发到channel上的多个随机选择的peer，其中该数量是可配置的常数。Peer也可以执行pull机制，而不是等待发送消息。该带有channel成员资格循环重复，分类帐和状态信息不断保持实时性和同步。为了传播新区块，该channel的leader peer从ordering服务中提取数据，并向peer中发起gossip传输。

Gossip消息

在线peer通过不断地广播“alive”消息来指示它们的可用性，每个消息包含公钥基础设施（PKI） ID和消息中发送者的签名。Peer通过收集这些alive的消息来维持渠道成员资格; 如果没有peer从特定对等体接收到活动消息，则该“死”对等体最终从信道成员资格中清除。由于“活着”消息是加密签名的，所以恶意peer不能伪造其他peer，因为它们缺少根证书颁发机构（CA）授权的签名密钥。

除了接收到的消息的自动转发之外，状态协调进程通过channel上的peer同步全局状态。每个peer不断地从channel上的其他peer中提取块，以便在识别出差异时修复自己的状态。由于不需要固定连接来维护基于gossip的数据传播，因此该流程可以可靠地为共享分类帐提供数据一致性和完整性，包括对崩溃节点的容错。

由于channel被隔离，一个channel上的peer不能在其他channel上发送消息或共享信息。虽然任何peer可以属于多个channel，但是分区消息通过基于peer的channel订阅应用消息路由策略来防止块被传播到不在channel中的peer。 在DM.IO网络中，节点会定期地利用Gossip协议发送它看到的账本的最新的数据，并对发送消息进行签名认证。通过使用该协议，主要实现如下功能：

- 通道内成员的探测：新加入通道的节点可以获知其他节点的信息，并发送Alive信息宣布在线；离线节点经过一段时间后可以被其他节点感知。

- 节点之间同步数据：多个节点之间彼此同步数据，保持一致性。另外，Leader节点从Orderer拉取区块数据后，也可以通过Gossip传播给通道内其他、节点。

交易确认

使用PBFT算法的区块链，一般出块者都是100%参与的。一笔交易在广播后平均0.25秒，就可以认为具有99.9%的确定性了。

DM.IO 采用异步拜占庭容错(aBFT, asynchronous Byzantine Fault Tolerance), 来实现更快的不可逆性。aBFT算法使得在1秒内就可以对不可逆性得到100%的确认。

作为权益证明的事务(TaBFT)

DM.IO软件要求每个交易都要将最近区块的区块头哈希的一部分包括在其中。 这一哈希有两个目的：

1.防止在区块链分叉上的交易重放，这些交易并不包含参考区块;

2.通知区块链网络，某一用户及其资产(stake)处于特定的分叉上

随着时间的推移所有用户都能直接对区块链进行确认，这使得伪造链难以作伪，因为伪造的链无法从合法的链上转移交易。

账户

DM.IO软件允许使用唯一的可读的名称来实现对帐户的引用，名称最长为12个字符。该名称由帐户的创建者选择。账户创建者必须留出RAM空间用于存储新的账户，直至新建的账户抵押了通证以获得自己的RAM空间。

在去中心化的情境下，应用程序开发人员将为创建帐户支付名义成本来注册新的用户。通常企业已经以广告和免费服务等形式，为所获取的每个用户花费了大量资金。相比之下，创建新的区块链帐户所需的资金成本是微不足道的。并且幸运的是，没有必要为已经由另一个应用程序注册的用户创建帐户。

动作(Action)及处理器

每个帐户可以将结构化的Action发送到其他帐户，并且可以定义Action被接受后的处理脚本。DM.IO软件为每个帐户提供其自己独有的数据库，只能由自己的action处理程序访问。Action处理脚本还可以向其他帐户发送Action。Action和自动的action处理程序的组合正是DM.IO定义智能合约的方式。

为了支持并行执行，每个账户都可以在其数据库中定义任意数量的范围(scope)。区块生产者会对事务进行排程，不会在访问scope的内存时出现冲突，因此事务可以进行并行执行。

权限映射

DM.IO软件允许每个帐户定义在合约/动作或任何其他帐户的合约，以及账户自己的命名权限级别之间进行映射。例如，账户持有人可以将账户持有人的社交媒体应用程序映射到帐户持有者的“朋友”权限组。通过此映射，帐户的任何朋友都可以和帐户持有者一样，在帐户的社交媒体上发布内容。即使他们会作为帐户持有者来发帖，他们仍然使用自己的密钥来为Action签名。这意味着总是可以辨别出来哪些朋友以何种方式使用了其帐户。

密钥被盗后的恢复

DM.IO软件为用户提供了一种在密钥被盗时恢复其帐户控制权的方法。 帐户所有者可以使用在过去30天内活动的所有者(owner)权限的密钥，以及账户所有者所指派的帐户恢复合作伙伴的许可，来重置其帐户上的所有者密钥。不经过帐户所有者的配合，帐户恢复合作伙伴无法重置其帐户的控制权。

对于攻击帐户的黑客而言，由于其已经“控制”该帐户，因此尝试执行恢复过程没有任何收获。此外，如果他们的确进行恢复的过程，那么恢复合作伙伴可能需要身份认证和多因素认证（如电话和电子邮件）。这或者会暴露黑客的身份，或者黑客在恢复过程中毫无所得。

这个过程与简单的多重签名机制有极大的不同。通过多重签名的交易，有一个对象会执行并参与每一笔交易。然而，通过恢复过程，恢复过程的合作伙伴仅参与了恢复的过程，并没有权力参与日常的交易。这极大降低了相关参与者的成本和法律责任。

通信延迟优化

延迟时间是一个帐户将动作(Action)发送到另一个帐户并收到响应所需的时间。DM.IO软件的目标是使两个帐户能够在单个区块内来回交换Action，而不必在每个Action之间等待0.5秒。为了实现这一点，DM.IO软件将每个区块分为周期（cycle）。每个周期分为多个碎片（shard），每个碎片(shard)包含一组事务列表。每个事务包含一组要传递的动作(Action)。该结构可以被可视化为树，其中各层依据其特性被顺序处理或者并行处理。

在一个周期中生成的事务可以在任何后续的周期或区块中传送。区块生产者不断地向一个区块中添加cycle，直至达到了最大的时钟时间，或者没有新生成的需要传送的事务为止。

可以使用区块的静态分析来验证在给定周期(cycle)内是否存在两个碎片(shards)包含了修改同一个帐户的事务。只要这种静态分析机制一直起作用，就可以通过并行运行所有线程来处理区块。

通证模型和资源使用

请注意: 在本篇白皮书中，所指的加密通证是使用DM.IO软件所构建区块链中的加密通证。

所有的区块链都是资源受限的，需要系统防止其滥用。在使用 DM.IO 软件的区块链中，应用程序会消耗三大类资源:

1. 带宽和日志存储 (磁盘);
2. 计算和计算积压 (CPU);
3. 状态存储 (RAM).

瞬时使用和长期使用的这两类组件都会消耗带宽和计算。区块链系统将维护所有Action的日志，这些日志将会被所有的完整节点下载和存储。通过日志，可以重构所有应用程序的状态。

计算债务(computational debt)是指通过对 Action 的日志重新生成状态的计算消耗。如果计算债务的增长太大，就有必要对区块链的当前状态进行快照，并抛弃区块链的历史状态。如果计算债务增长过快，区块链将会耗用6个月的时间来重放1年的交易。因此，对计算债务进行谨慎管理是至关重要的。

区块链存储的状态指那些可从应用程序逻辑访问的信息。它包括诸如订单和帐户余额等信息。如果应用程序不读取该状态，则不应该存储它。例如，博客文章的内容和注释不被应用程序逻辑读取，因此它们不应该存储在

区块链的状态中。与此同时，博文或评论是否存在这一状态信息、投票数和其他属性将作为区块链状态的一部分被存储起来。

区块生成者可以公布它们可用的带宽、计算资源和状态的容量。DM.IO系统根据账户在期限为三天的抵押合约中所抵押的通证数量，允许每个帐户可以消耗一定比例的可用容量。例如，假设一个基于DM.IO系统的区块链应用启动，如果一个帐户持有该区块链提供的总通证的1%，那么这个帐户就有可能利用该区块链1%的状态存储容量。

使用DM.IO系统的区块链上，带宽和计算能力的分配是基于部分储备机制的，因为它们是短暂的(未使用的容量不能存储下来为将来使用)。DM.IO系统将使用类似于Steem的算法来限制带宽使用速率。

客观和主观度量

正如前面所讨论的，可度量计算的使用对性能和优化有很大的影响；因此，所有资源的使用限制最终都是主观的，并且根据各自的算法和估计来执行。通常由区块生产者创建自定义的插件来实现。

除此之外，有些无足轻重的东西，可以进行客观度量。所传递动作(Actions)的数量和存储在内部数据库中的数据大小，这些都很容易进行客观测量。DM.IO软件允许区块生产者在这些客观的度量上应用相同的算法，但是也可以在主观度量上选择更严格的主观算法。

接收方支付

历来是由企业为办公空间、计算电力以及运营业务所需的其他费用买单。客户从企业购买特定产品，而这些产品的销售收入将用于支付企业的运营成本。同样，没有任何网站要求访问者为维护服务器而支付小额费用。因此，去中心化应用程序不应该强迫它的客户为使用区块链而向区块链支付直接费用。

使用DM.IO软件的区块链不要求用户直接向区块链支付使用费用，因此不限制或阻止企业制定其产品的货币化策略。

虽然接收方可以付费，DM.IO 软件允许发送方为带宽，计算和存储付费。这样的话，应用开发者可以选择最适合他们应用程序的方式。在很多情况下，对于不想要自己实现定量配给系统(rationing system)的开发者而言，发送方付费这一方式显著降低了复杂性。应用开发者可以将带宽和计算能力代理给他们的用户，然后采用“发送方付费”的模式来使用应用。从DM.IO发一个token需要支付300只呆马币，每调用一次合约支付1只呆马币来奖励区块生产者，从终端用户的角度来看仍然是免费的，但是从区块链的视角看，这是属于发送方付费的模式。

授权能力

在一条使用DM.IO系统开发的区块链上，通证的持有人可能不需要立即消耗可用带宽的全部或部分资源，他们可以选择将未消耗的带宽委托或租赁给他人；在这一区块链上运行DM.IO 软件的区块生产者将识别这一授权并分配相应的带宽。

将交易成本与通证价值区分开

DM.IO软件的主要优点之一是，应用程序可用的带宽数完全独立于通证的价格之外。如果应用程序所有者持有相应数量的通证，那么应用程序可以在固定的状态下，使用固定的带宽资源持续运行。开发人员和用户不会受到通证的市场价格波动的影响，因此不会依赖于喂价。换句话说，使用DM.IO程序运行的区块链，可以让区块生产者能够自然地增加每单位通证可用的带宽、计算资源和存储资源，这与通证的价值无关。

使用DM.IO软件的区块链，区块生产者每次产生区块，都会得到一定的通证奖励。通证的值将影响一个区块生成者能够有钱购买的带宽、存储和计算量；这个模型自然会利用通证价值的上涨来提高网络性能。

状态存储成本

带宽和计算可以可以代理给他人，但是应用程序状态的存储需要开发者持有通证，直至状态删除为止。如果程序的状态永不删除，那么实际上这部分通证就退出了流通。

出块奖励

在使用DM.IO软件构建的区块链上，每生成一个区块，出块者都会得到一些新的通证作为奖励。在这一状况下，新创建的通证数量是由所有区块生产者公布的期望报酬的中位数而决定的。可以配置DM.IO软件，限制区块生成者所得奖励上限，使得通证供应的年总增长率不超过5%。

工作提案系统

在基于DM.IO 软件的区块链上，通证持有人除了选举区块生产者，还可以选出一些旨在造福社区的工作提案。获胜的提案能够得到通证奖励，所配置的每年通证的膨胀率减去已支付给区块生成者的部分，就是这部分奖励的最大值。这些提案将按照所得到的选票比例来获得通证的分配，上限是他们进行工作所要求的通证数量。所选出的提案可以由通证持有人新选出的提案所替代。

治理

治理是社区之中成员的如下过程：

1. 有些事实无法通过软件代码来收集，而人们通过搜集这些事实，就主观问题达成共识;

2. 执行他们达成的决策;

基于DM.IO 软件的区块链实现的治理过程，有效地引导了区块生产者的现有作用。以前的区块链缺乏定义好的治理过程，依赖于临时、非正式和经常存在争议的治理过程，从而导致不可预知的结果。

基于DM.IO 软件的区块链认为，权力来自于通证持有人，他们将权力代理给区块生产者。区块生产者被赋予了有限和经审查的授权，可以冻结账户，更新有缺陷的应用，并提出对基础协议进行硬分叉的变更。

DM.IO 软件内嵌了区块生产者的选举机制。在对区块链进行任何更改之前，这些区块生产者必须批准它。如果区块生产者拒绝按通证持有人的期望做出变更，那么可以投票将其替换。如果未经通证持有者的允许区块生产者就擅作更改，那么所有其他非出块的全节点验证者(交易所等)将拒绝该更改。

冻结账户

有时，智能合约会发生异常或不可预知的状况，无法按预期执行；有时，应用程序或帐户可能会利用漏洞，使其消耗不合理的巨量资源。当此类问题不可避免地发生时，区块生产者应当有权力纠正。

所有区块链上的区块生产者都有权选择将哪些事务包含在区块之中，这给予了他们冻结账户的能力。使用DM.IO 软件的区块链将这一权力正式化了，对一个账户冻结的决策会提交给5个节点进行投票，如果得到5个节点的通过，则会冻结账户。如果出块者滥用权力，他们可以被投票换掉，而冻结的账号也会解冻。

更改账户代码

当其他一切都失败了，而“无法停止的应用程序”以一种不可预知的方式运行时，使用 DM.IO 软件的区块链，允许区块生产者在不需要硬分叉整个区块链的情况下就能替换掉帐户的代码。与冻结帐户的过程类似，替换账户的代码需要得到被选中出块节点中5个节点的投票同意。

目标

通过呆马链，我们规划了针对区块链技术的未来愿景。我们相信区块链技术将会对现实生活的很多方面产生根本性影响，包括从商业到数据存储和其他事情。基于这一点，我们认为针对区块链/分布式账本技术制定健壮的和开放的标准是必要的，因为，这能推动这样的技术在主流商业化领域得到应用。我们相信未来世界将建立许多内部互联的分布式数据库和区块链，每一个分布式数据库和区块链都将满足特定用户的需求，但也要求与其他的账本进行通信。

因此，我们认为任何针对区块链技术的开放标准都必须尽量模块化。未来让开发人员能够按照自己的意愿来回替换不同版本的各种区块链组件，必须建立这样的标准。例如，一些区块链应用案例在要求快速一致算法的同时要求更多的可信，而某些应用案例可能不要求速度但要求更加可信。密码算法、智能合约和数据库存储是需要实现“即插即用”的其他特征。

我们对呆马链的长期愿景就是它包含丰富的、易用的API和数量庞大的核心模块，这样就能容易实现开发和互操作。虽然我们希望核心的呆马链模块能够满足尽量多的应用案例，但我们也知道呆马链的核心内容是不可能覆盖每个行业的应用案例。然而，我们的API应该足够灵活，使得不使用呆马链核心组件构建的应用案例能很容易地与核心的呆马链组件和区块链实现互动。

我们不可能考虑到呆马链和通用区块链技术将来所有的使用方式，因此，为了能够容纳将来未知的开发，呆马链的设计是尽量模块化和可扩展。除此之外，呆马链的模块化应该能让更多的人围绕呆马链工作。我们希望这种模块化的方式允许发明或开发新的区块链技术的人们发现：使用或与呆马链合作是很容易的。

我们相信，针对任何区块链结构根本需求的一个方面就是网络中任何一方行为的识别和模式必须不能被未授权方通过检查账本就能查明。我们也期望某个需求允许区块链用户确认业务逻辑和/或交易机密的其他参数、使他们对任何人都是不可访问的，而不是合同的利益相关方或资产被转移。

呆马链应该为核心协议之上轻松实现的各类丰富应用提供支持。这必将要求支持各种交易语义、密码算法、协商机制和数据库存储协议。例如，加密的呆马链应该包括所有的加密、签名和更高级的功能密码，从简单的、快速的对称加密到复杂的功能加密和基于属性的签名。这些基本的技术原理应通过配置来支持重要的商业交易，例如不同程度的授权交易的不可改变性和可审计性。

总的来说，呆马链是一个易用、十分有用和健壮的平台，任何对构建区块链软件的机构和个人都可以把它用来做区块链存储。尽管由于实际考虑不足，呆马链针对每个潜在用户和应用案例可能缺乏这种理想的功能，但我们的目标就是使呆马链尽可能接近这种理想的状态。

基于呆马链的行业应用案例

我们已经编制了一套本质上支持下述抽象应用案例的区块链初始需求。这些应用案例并不代表呆马链的所有应用案例，而是一组展示呆马链某些能力和特征的典型样本。

> （注：这些应用案例为体系结构设计和测试驱动的开发提供指南。虽然还是一项推进中的工作，这些应用案例应该是所有贡献者一致认同的，无论是内容还是堆栈中排名的优先次序。如果您觉得这些内容有欠缺，可以提出改变建议。理想的情况是不超过四个抽象应用案例中有三个是首选。）

支持可信设备

呆马区块链的底层区块链存储技术，将矿机与传统的智能设备无人售货机、无人值守、洗衣机等智能设备，严谨的整合到一起。无人售货机的每一次打开以及每一次的取货开门数据、重量数据甚至与供货商的结算数据都以区块的形式在呆马公链中进行存储。
与传统模式相比，智慧物联网公司需要向类似阿里云、腾讯云、微软云这种服务器存储公司缴纳高额的数据存储费用。使用链式数据库以后不仅没有了这些存储费用，并且可以大大加强它的安全性，以及去解决供货商与无人售货机智能设备厂商、投放商之间的信任问题。每一次的开门数据、拿取数据、清算数据都会被完整的记录在呆马区块链的各个节点上，不容篡改。

金融资产处置

诸如证券这样的金融资产必须能在区块链网络上实现去中心化，这样所有同种资产的利益相关方就能直接访问这一资产，进而发起交易，获取相关信息而不需要通过层层中间环节来进行了。

交易可以在利益相关者之间商定的时间期限内解决, 交易可以实现实时结算，利益相关者都可以实时掌握资产情况。对于任何种类的资产，利益相关方应该有权增加商务规则，这样也能通过自动化逻辑的应用来降低成本。

创建资产的人必须像用例保证的那样，实现资产和相关交易规则保密或者公开。例如，资产创建者应该能够创建资产，而这一资产的交易记录以及交易模式对于利益相关者之外的群体是不可见的，甚至创建人本身也不能访问。

协作

公司A发起一个协作的事件请求，无论这个过程中涉及多少中间环节（如代理接收/支付，CSD，ICSD，本地/全球保管银行，资产管理公司等）公司A需要将邀请的完整细节信息实时发送给利益相关方。一旦利益相关者作出交易决策，这个决策也需要被实时处理完成（包括作为协作事件一部分的新增份额）。

如果需要，投资者的响应会被保密，这样他们就可以基于价值作出决策，而不用担心自己的操作行为带来的负面影响。

供应链

区块链的框架必须满足供应链中每一位参与者的如下需求：录入并追踪原材料的来源；记录部件生产的遥测数据；追踪航运商品的出处；保证包括成品生产、储存、销售及后续事宜在内的所有数据都不被篡改。

除了之前描述的商务合约和资产存管模式的特征，供应链这一用例更多强调的是其深度可搜索性，保证能够在过去的层层交易中追溯所需记录。其核心是为每一个基于其它部件构成的商品创建出处（可追溯的源）。

保险防欺诈

利用区块链共识机制、防篡改机制和可追溯机制，在保险代偿、追偿时提供有效证据支撑，。 以车险理赔为例，通常包含车主、4S 店或维修厂、保险公司、交管部门等多个主体，时常发 生骗保等理赔欺诈问题。

依托区块链技术和车联网技术，在车辆上安装相应传感记录设备，保证信息的真实、准确和不可篡改，在出险时，实时或准实时地将车辆事故数据提交给应用区块链技术的“事故认证平台”系统，交警裁决数据、传感记录器数据、维修厂数据等都实时同步，从根本上解决车险 理赔欺诈问题，同时提高保险理赔案件的效率和准确性。

大数据安全

区块链可以解决大数据的安全性问题，保证数据的隐私性。区块链的可追溯特性使得数据 从采集、交易、流通，以及计算分析的每一步记录都可以留存在区块链上，使得数据的质量获 得前所未有的强信任背书，也保证了数据分析结果的正确性和数据挖掘的效果，能够进一步规 范数据的使用，精细化授权范围，追溯数据使用情况，全面保障数据使用的安全合规。

脱敏后的数据交易流通，则有利于突破信息孤岛，建立数据横向流通机制，逐步推动形成 基于全球化的数据交易、数据资产保护等全新的应用场景。

主数据管理

主数据通常并不是交易信息数据，而是行业信息的关键和基础组成部分，如：customer（客户）、employee（员工）、supplier（供应商）、product（产品）、location（地址）和contract（合同）等。授权认证机构发起变更并对变更进行校验，维护核心数据的唯一性和真实性可以解决许多数据质量和一致性问题。

共享经济和物联网

共享经济将在许多传统行业领域产生可带来营收的新型产业，如：智慧城市、互联家园、自动化、运输、医疗、分销、建筑、教育、健身等领域。

交易中的个体、组织以及监管机构并不总是相互信任。善加利用基于分布式账本的区块链技术有助于解决交易各方相互间的信任问题。区块链技术同时也有助于交易的实时处理和资产状态的实时访问。灵活的部署模型，

可插拔的共识机制，私下交易以及保密合约对于呆马链的部署都很重要。

合同及发票防伪

电子合同和电子发票的日益普及，为我们日常生活和商业活动带来很多便利的同时，也面 临合同造假、发票造假及重复报销等许多新的问题，需要监管部门和企业共同探索有效的解决 方案。在开具电子合同、电子发票的同时，通过联盟链完成向监管部门的备案，在发生造假、 重复报销等情况时，通过核对已备案电子合同、电子发票的区块链 ID“身份证“，可以快速判 定造假事实，确定造假主体，实现实时监管。

公益追溯

应用区块链技术支撑公益项目的阳光、透明和可追溯。爱心物资经由高效的物流体系直接 配送到公益项目地，并由公益机构执行人员发放至受助人手中。捐赠人可通过客户端实时查询 所捐赠物资的物流状态，直观地看到物资发放到受助人手中的全过程。

从选购爱心物资开始的全部过程信息、参与主体信息均使用区块链技术防止篡改，确保公 益透明性、可追溯，极大增加公益平台的权威性和可信度。

典型需求

我们接下来描述呆马链的典型需求。这里描述的典型需求满足了多种用例和商务情境，我们希望产记账本将来能够发展出更多的特性。

首先，呆马链最关键的需求是架构。正像我们反复强调的，不同的应用会对机密算法、一致性算法和数据库存储方式有着不同的需求。然而，我们基于架构细化一些更具体的需求，可以广泛应用于更多领域。

私下交易和保密合约

呆马链最终应该支持多种加密工具和方法，确保满足相应的加密和隐私管理需求。这些工具用于确保诸如身份识别、交易属性、智能合约状态等信息的真实性，同时不会侵犯信息的私密性。

与那些金融领域的用例不同，某些用例（如物联网）需要性能优化的基本保密功能，其加密和共识算法需要兼顾基本的加密功能和复杂的定制需求。

身份识别和审计

不考虑私下交易和保密交易，呆马链使用基于PKI（公钥基础设施）的加密算法实现了交易中的身份识别和审计功能。

呆马链对用户和交易相关者除了单纯提供基于PKI的身份识别功能还应该支持对这些访问和识别操作的归档功能，包括交易相关者之间的加密请求，以便实现对涉及所有权变更的相关用例进行基于文档的审计追踪。

除了主动进行身份识别，呆马链也允许用户在特定情况下隐藏身份识别操作，仅当需要的时候才提供证明。当然，这已经超出了传统的身份识别概念。此外，PKI非常灵活，允许用户根据特定的需求选择不同强度的加密措施。

互操作能力

在松耦合的网络中，独立的网络相互间不需要了解彼此的运行细节。然而，这样的独立网络也需要具备一定的共性，以实现彼此间正确的信息交换。尤其是对着区块链技术的普及，应该考虑各种不同的区块链系统相互间的信息交换操作。各类区块链网络实现上的差异以及其演进和不断变化的特性会导致实现的高度专业化。制定专业的通信分类标准，创建在多种网络间通信的通用语言将是一项漫长而艰巨的工作。

区块链技术在设计和实现上存在差异，当不同服务彼此间交互操作，互操作就产生了。

呆马链定义可在两个或多个系统（组件）间进行信息交换，并使用交换的信息。为实现跨行业和跨用例的广泛应用，呆马链支持两个或多个区块链间进行信息交换的协议功能。

可移植性

呆马链项目通过从其核心组件接口中提取的增值系统实现移植操作。例如，智能合约就可以不做任何变更地移植部署。可移植性的增值系统，诸如：API（应用程序编程接口）库，GUIs（图形用户接口）开发应用，扩展库等，保证了呆马链的增值系统可以跨版本使用、实现和部署，同时也保证了呆马链项目的功能在异构环境下的大型区块链网络中得以实现。

脚本 & 虚拟机

DM.IO 软件首先是一个平台，协调已认证信息(称为Action，动作)在账户间的传递。脚本语言和虚拟机的实现细节将独立于 DM.IO 技术。任何开发语言或虚拟机，只要具有确定性，经过了恰当的沙盒化并具有足够的性能，，都可以与 DM.IO 软件的 API集成。

体系结构

当前的架构，包括四个大类：身份识别服务，策略服务，区块链和智能合约。这些分类都是逻辑结构，而不是将组件划分成独立的进程、地址空间或（虚拟）机的物理描述。

身份识别服务负责管理诸如资产、智能合约这样的实体、参与者和分类帐对象的身份识别。（参与者通过注册获取身份，之后通过授权机构发放的密钥进行交易。）

策略服务负责管理访问控制、隐私、联盟规则、共识规则等。

区块链服务负责通过点对点通信协议管理分布式账本。经过优化的数据结构可以有效维护在众参与者间复制的整体状态信息。不同的共识算法或将嵌入到每一个配置中，以保证高度一致性（通过BTF算法处理错误，通过崩溃容忍机制处理延迟和中断，或借助工作量证明方案应对审查。）

智能合约服务负责提供安全又轻便的方式供智能合约在验证节点上运行。

身份识别服务

身份识别是呆马链协议中的普遍需求。身份是被服务负责管理身份识别需求，包括：参与者、验证者和处理者；账本中包含的诸如资产和智能合约等对象：系统中诸如网络服务和执行环境等组件。

身份识别服务包括在账本中的各种角色对象的标识形式。验证器可以在网络配置时定义交易所需的权限。

网络配置时可以定义允许快速便捷的自由访问，或者定义更多的限制条件。

策略管理服务

策略管理服务负责系统策略的配置和管理，包括：访问控制、授权权限、联盟策略（以代码形式实现的针对成员、非成员的规章制度）、身份注册和验证策略、隐私、保密和问责策略、共识策略。

区块链服务

区块链服务包括三个关键组件：点对点（P2P）协议、分布式账本和共识管理器。

点对点（P2P）协议支持的功能：双向数据流、流量空盒子和连接的请求复用。最重要的是，它可以在现有的互联网基础设施中使用，包括：防火墙、代理服务器及安全设备。这个组件定义了对等节点所使用的信息，可以是点对点，也可以是组播。

分布式账本通过对交易的处理和校验、更新和维护账本中对象的状态来管理区块链以及整体状态信息。

分布式账本使用数据存储维护数据集，同时建立内部的数据结构区别不同的状态，以此满足上述三个属性。大文件使用链外存储，不记录在账本中。它们的哈希函数值作为交易的一部分被存储在数据链中，以此维护文件的完整性。共识管理器是共识算法和其它呆马链组件间接口的抽象定义。共识管理器接收交易请求，借助相关算法判断如何组织以及何时执行交易。交易的成功执行将导致呆马链状态发生改变。

通过模块化的可插拔共识功能，呆马链支持为评估和记录特定系统风险而设计的各种共识模块。

呆马链提供pub/sub（消息的发布/订阅）模式的事件管理框架，这样外部应用就可以监控和收到呆马链的事件告警。

智能合约服务

智能合约是一组运行在验证节点上的去中心化交易程序。 智能合约服务包括安全运行环境、智能合约注册以及声明周期管理。

应用编程接口

api、sdk、cli 呆马链的特色之一是提供了一套易用、可灵活扩展的API接口。 呆马链的每个模块都清晰完整地定义了相应的API接口，因此这些模块可以实现“即插即用”。例如共识算法的API支持用户无需修改算法代码就可以在各类用例中使用这一算法。完善的API接口为呆马链支持各类用例提供了有力保障。

此外，非模块对模块通信的外部API接口设计更便于普通开发人员在呆马链顶层编写代码。

一整套完全独立的API接口、智能合约模块以及共识协议模块是保障参与者能够在整个生态系统中提供贡献的基础。这一特性保障了整个生态系统的快速成长。

网络拓补

理论上讲呆马链的的网络拓扑应该是完全不同的：特别是参与者可以通过云服务操控各种类型的对等节点，包括验证节点，或者参与者本身就是验证节点。呆马链运行在不可知的底层网络结构中，我们无法得知究竟谁在使用这些节点设备。

假如云节点是主服务节点，那么就必须考虑使用更加有效的加密解决方案避免云服务器中的信息被恶意泄露。

有些部署的呆马链可能会面临较大的系统变化，导致节点间通信延迟。网络失效，节点失效，因此网络的冗余性和可恢复性在部署之初就应加以考虑。

区块链4.0跨链共识机制

呆马浏览器支持跨链共识机制，支持跨链数据存储；呆马获得司法鉴定所认证，目前在中国大陆各司法鉴证所，各法院往链上存的数据均可作为庭上证据使用；其存储节点为国家超级计算机济南中心；

> “国家超级计算济南中心” 该中心建有中国首台全部采用国产CPU和系统软件构建的千万亿次计算机系统，标志着中国成为继美国、日本之后能够采用自主CPU构建千万亿次计算机的国家。该中心系三大国家千万亿次超级计算中心之一，另两个中心为国家超级计算天津中心、国家超级计算深圳中心。国家超级计算济南中心由山东省科学院建设、运营和维护，2011年3月开始建设，近期建成并投入运行。 济南中心装配的神威蓝光计算机系统，由国家并行计算机工程技术研究中心研制，系统采用万万亿次架构，全机装配8704片由国家高性能集成电路 神威蓝光千万亿次系统 神威蓝光千万亿次系统 （ 上海 ）设计中心自主研发的“申威1600”处理器，峰位性能达到1.0706千万亿次浮点运算/秒，持续性能为0．796千万亿次浮点运算/秒，运行（ LINPACK ）效率达到74．4%，性能功耗比超过741百万次浮点运算/秒·瓦，组装密度和性能功耗比居世界先进水平，系统综合水平处于当今世界先进行列。济南中心全部采用国产CPU和系统软件，实现了国家大型关键信息基础设施核心技术的自主可控。 申威1600 申威1600 中国 [2] 《国家中长期科学和技术发展规划纲要》将千万亿次高效能计算机研制列入优先主题，科技部明确提出要掌握千万亿次高效能计算机研制的关键技术，并将“高效能计算机及网格服务环境”列为“十一五”863重大项目。高性能计算机的研制能力和应用水平是一个国家科技发展水平和综合国力的重要标志之一，也是世界发达国家竞相争夺的科技战略制高点。 DM.IO 软件旨在促进区块链间的跨链交互，这通过简化Action存在证明(proof of Action existence)和Action顺序证明(proof of Action sequence)的生成过程来实现。这些证明与围绕Action 传递而设计的应用架构结合起来，将跨链通讯以及验证证明的细节对应用的发者隐藏，展现给开发者的是高层次的抽象。

如果客户端不需要处理所有的事务(transaction), 那么，与其他区块链交互将变得非常容易。毕竟，一个交易所只会关注交易所的出账和入账信息，而不会关心其它。更理想的情况是，对于交易所自身所维持的链来说，如果可以将轻量级的默克尔存款证明应用其中，那么就不必完全依赖自己的区块生产者。至少，某个区块链的生产者在同步另一条区块链时，会希望尽可能减小开销。

区块链4.0跨链共识机制的目标是能产生相对轻量级的交易存在证明，其他人只需要追踪一个相对轻量级的数据集，就可以对此进行验证。既然如此，目标就是证明一笔特定的事务被一个特定的区块所包含，并且 某一条特定的区块链的验证历史中，已经包含了该区块了。

比特币的轻量级验证方式是，假设所有节点都可以读取区块头数据的完整记录，区块头数据每年增长4MB。 假设每秒产生10笔交易，一个有效的证明需要512 bytes，这对于一个出块时间为10分钟的区块链来说是可行的。但对于一个出块时间为 0.5 秒的区块链来说，这就远远谈不上“轻量”了。

DM.IO 软件的轻量级证明中，在某笔交易被包含到区块链之后，只需要对任意一个不可逆的区块头进行验证即可。使用下图中的哈希链表架构，可能只需要不到1024个字节大小的证明，就可以验证任意一笔交易的存在。

给定区块链上任意一个区块的区块id，以及一个不可逆的可信区块的区块头。可以证明某个区块是包含在区块链上的。这一证明的算法复杂度是（ log2(N) ），其中 N 是区块链上的区块数量。给定 SHA256 加密算法的类型，只用 864 个字节，你就能够证明在一条包含了 1亿个区块的链上，一个任意的区块是否存在。

生成区块的时候如果使用合适的哈希链表来产生这些证明，只会带来很小的增量开销，这意味着没有理由不以这种方式去生成区块。

对其他链上的证明进行验证时，在时间、空间和带宽方面都有很大的优化空间。跟踪所有区块头数据(420 MB/年)可以让证明的尺寸最小。只跟踪最近的区块头，可以在长期存储文件的最小化和证明尺寸的最小化之间，得到均衡。或者，一个区块链可以采用惰性评估的方式，只记录过去证明的中间哈希值。新的证明只需要包含指向已知的sparse tree（稀疏树）结构的链接。实际使用的方式，需要根据在merkle 证明中所引用的交易位于外链上所占的比例来决定。

在一条区块链上，可以将另外一条区块链的所有区块历史都包含其中，不需要再进行跨链的证明。在跨链关联密度达到一定程度之后，这会是更高效的做法。出于性能考虑，理想情况是尽可能降低将跨链证明的频率。

跨链通讯延迟

与外部区块链通讯时，区块生产者必须等到一笔事务(transaction)经过外部区块链的确认，达到了100%的不可篡改的确定性之后，才能够接受该笔事务，认可为有效的输入。在一条基于 DM.IO 软件的区块链上，借助于PBFT 0.5秒的出块速度，以及拜占庭容错的不可篡改性，这一过程大约耗时为0.5秒。如果某个链上的区块生产者不等到交易不可篡改，就像一个交易所接受了一笔存款而后这笔操作又撤销了的情况一样，这会影响这条链共识的有效性。DM.IO 软件使用了 PBFT 和 aBFT(异步拜占庭容错)算法，提供快速的不可篡改性。

完成性证明

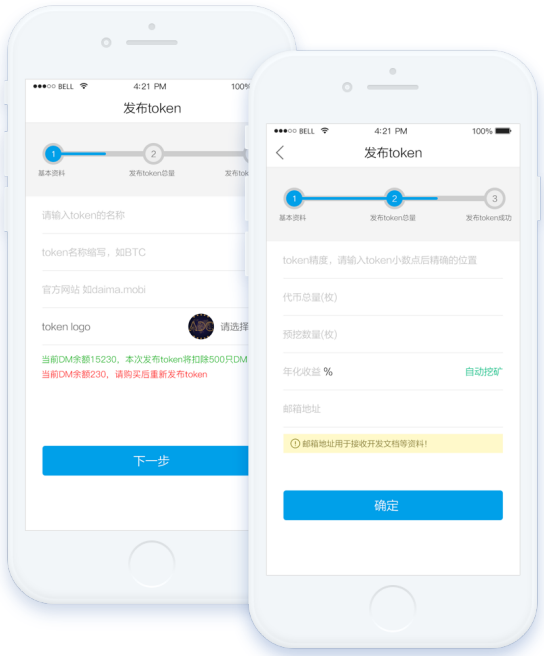
使用外部区块链的merkle 证明，知道所有处理过的事务都是有效的，和知道没有事务被跳过或忽略，这两者之间有明显差别。虽然无法证明最近的所有事务都是已知的，但是，要证明在事务的历史中不存在遗漏，还是可能的。DM.IO 软件为传递给每个账户的每个Action都指定了一个序列号，使得这一点成为可能。用户可以用这些序列号来证明，与某个账户相关的所有的 Action 都得到了处理，并且是按顺序处理的。

呆马区块链浏览器

呆马区块链浏览器旨在为普通用户提供一款安全放心、简单好用、功能强大的数字资产钱包应用。去中心化的呆马区块链浏览器，在服务器上不会存储用户的密钥、助记词等敏感信息，密钥/助记词都是只存在于用户自己的手机、电脑上，并不会同步上传到呆马区块链浏览器的服务器，而且在开发层面也默认关闭了iCloud等云同步，最大可能的保证用户的敏感信息；另外，用户的核心操作均可以自己完成，如钱包生成、导入、密钥存储、助记词备份、转账等，并不需要和呆马区块链浏览器服务器做交互；当然转账操作最终是要连接区块链节点的；所以，作为去中心化的数字钱包，呆马区块链浏览器天然基础上是更安全的；
只要学会妥善保管和安全备份你的私钥，去中心化钱包比中心化托管钱包是要安全的多的。可以放心使用。

支持侧链token发布

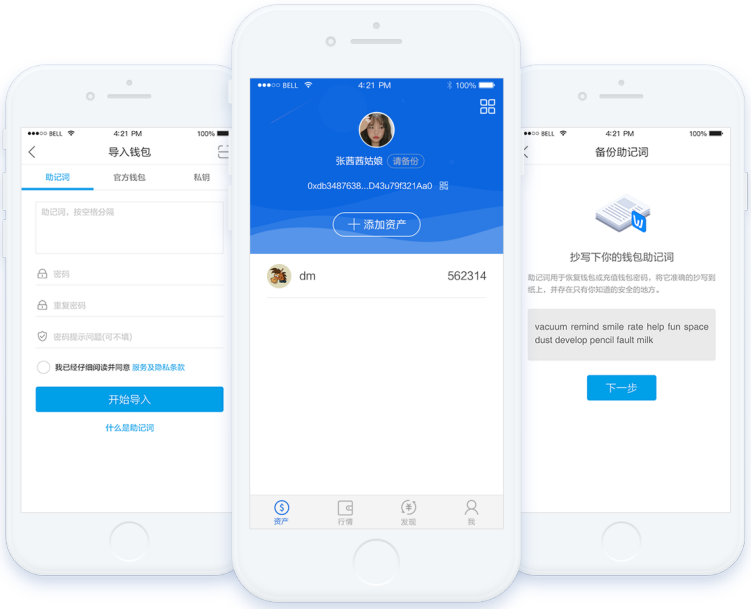
- 全球首款智能自主加密侧链token发布系统
- 只需简单3步完成发布侧链token
- 随时随地发布你的区块链侧链



支持4.0跨链共识机制 资产管理

- 私钥本地安全保存，资产一目了然
- 支持多种钱包类型，轻松导入导出

- 助记词备份防丢失，多重签名防盗



一键添加

- 一键添加数字资产，实时跟踪转入转出动向，关注资产余额变化

呆马发行机制

呆马发行10亿只，每次流通自动湮灭10%，最终恒量1亿只；呆马的分配，目前只用于呆马区块链发展基金无偿捐赠给中国各创业园与高科技孵化中心，用于中国传统企业上链给区块生产者支付使用，发布侧链需要支付300只呆马，每调用一次合约支付1只呆马币来奖励区块生产者，不用于市场流通。

团队主要成员介绍

林学民

国际电气电子工程师学会会士
新南威尔士大学计算机科学及工程学院教授、数据库研究实验室主任
新南威尔士大学首席教授
早年就读于复旦大学数学系，师从著名数学家苏步青教授
林教授于1992年获得澳大利亚昆士兰大学计算机学科博士学位，并曾先后在昆士兰大学和西澳大学从事科研和教学工作。他于2010年入选中国第四批“千人计划”教授。 时空数据和流数据的查询、图和文本的匹配查询、不确定数据的概化查询及图数据可视化方向、算法研究中取得了一系列的突破性进展。 先后主导十几个澳大利亚国家科研基金和中国国家自然科学基金项目 发表国际顶级期刊和会议论文(CCF A类)140余篇

帕尔木·那拉希姆汗（PALEM NARASIMHULU）

毕业于印度韦斯科技大学，硕士学位，计算机科学与工程专业
先后主持开发美国运通网上银行系统、荷兰银行网银账户管理系统、葛兰素史克医疗系统，并主持其高并发大数据架构研发工作。由于其突出贡献，先后获得易安信工程认证及圣雄甘地奖。

潘迪卡拉·苏巴拉郁（PANDIKALLA SUBBARAYUDU）

毕业于著名的尼赫鲁大学
为帕尔木·那拉希姆汗先生重点推荐的优秀工程师，与帕尔木·那拉希姆汗先生密切合作多年。潘迪卡拉·苏巴拉郁先生一直致力于银行线上交易和企业市场管理系统研发，对跨境电商应用所涉及到的银行间合作工作流、高并发销售自动化架构等方面拥有丰富经验和实践价值。

岳涛

数学学士和计算机硕士
阿里云认证工程师，熟悉前沿技术和golang
负责过某外企、国内知名互联网融公司框架架构，区块链深度研究者，参与过基于区块链应用的开发。

杨启东

北京航空工航天大学 硕士
前猎豹移动（原金山网络）、北京聚信量化科技技术专家 Python高级数据分析研发工程师。曾在猎豹移动一起创业，与solo launcher产品的CEO一起打拼，所在团队构建了2亿用户群的用户产品猎豹清理大师。有丰富的数据采集开发、数据建模经验，实现了组件化的数据采集服务。

郭雪雪

山东师范大学 硕士
原滴滴打车高级产品经理，曾先后在八金社（北京）、浪潮等公司负责产品设计工作。深耕互联网金融行业，在C端和B端产品上均有丰富的产品经验。

技术方面

第三方网络（如交易所、电商系统、支付系统）对接API

呆马链各交易所通用接口文档 API接口

区块链协议

呆马区块链协议作为最底层的架构设计，定义了区块链的数据格式标准，包括账本状 态、历史证明、账本操作集、合约指令集 4 个方面的数据标准。 **组件模型**

“组件模型”是区块链逻辑组件的框架模型，是对呆马区块链协议的实现框架。包括了 共识网络、账本、持久化引擎、合约引擎四个组件。 **服务平台**

“服务平台”是对上层的区块链协议和组件模型的具体实现，由网关、服务、节点网络、SDK 和一套工具集组成。

> 出块和交易 呆马链的出块速度大约是2秒钟产生1个、速度是以太坊的8倍、比特币的300倍. 那么为什么会是这样的呢? 因为算法不一样、主要区别在于呆马链的认证节点数量是可控的、且部署在内部网络中、可以进行高速处理。
>
> 呆马链和保证速度的同时、交易的安全也是非常高，它通过使用高强度加密证书和访问策略来保证节点的安全，每一笔交易都是由多台去中心化的验证节点共同检查后、才会生成块保存到所有节点的账本中。
>
> 呆马链对开放性特别重视、第三方应用平台如果想对接到呆马链上，那是非常容易的、通过与呆马链的开发SDK对接、可以快速实现区块链的应用

结 论

呆马链的任务是将区块链技术引入主流产业市场。回顾了可行的区块链解决方案，也了解了业界领先者及技术推广者给出的相关用例后，我们相信区块链将会成为至关重要的技术模型，推动众多工业和企业进行革新。

我们注意到，业内急需一套为企业打造的区块链架构，做到既高效又可扩展，并且能够支持企业级的加密和隐私保护。此外，我们还发现针对众多的区块链用例目录需求需要不同的底层实现。为挖掘区块链技术的潜力和创建适应不同用例的标准，我们设计 呆马链 框架时兼顾了灵活性和可扩展性。