



FOGLINK

雾链系统白皮书 ^{v3.1}

价值传输协议及雾链操作系统

FOG LINK WHITE PAPER

Value transfer protocol and Fog Link OS open platform

2018 01

support@foglink.io

目 录

摘要 Abstract	4
1. 雾计算与区块链的结合	5
1.1. 雾计算概念和特点	5
1.2. 雾计算与区块链技术的结合	7
2. 雾链的设计原则	9
2.1. 雾链的模块化设计	9
2.2. IBM ESB 总线与 FNK 结合的架构方案	10
2.3. IPFS 的网络特征	11
2.4. 雾链的安全性和性能	11
2.5. 雾链的系统化策略	11
3. 雾链系统 (Fog Link OS) 解决方案	12
3.1. 雾链系统架构	12
3.2. 雾链系统 (Fnk OS) 应用场景	13
3.3. 雾链代码示例	14
3.4. 雾链系统设备	16
3.5. 雾链开源计划	17
3.6. 雾链社区	17
3.7. 联盟组织	17
3.8. Road Map	18
4. 团队简介	19
4.1. 项目团队成员	19
4.2. 顾问团队	20
4.3. 早期投资人	21
5. 关于本白皮书	21

摘要 Abstract

Fog Link(简称:FNK 雾链) 将会建设 FNK 超级节点、星际档系统 IPFS(Inter Planetary File System)以及雾计算节点三种类型网络节点。FNK 利用计算机以及计算机的周边外围设备进行网络传输和存储空间等资源的回收利用,根据雾计算设备所提供的能力贡献,用于进行价值传输协议 (Value transfer protocol) 的实现,以及能力的存储和转移工作。

雾链系统 (Fog Link OS), 是 FNK 在去中心化应用开发平台 (DAPP Platform) 的核心能力, Fog Link OS 使应用边界和技术边界得到巨大的释放,让更多的用户可以感受到区块链技术给工作和生活带来的便利和改变。

雾链 (FNK) 会为应用程序开发者建立一个健康良好的开发环境,开发者利用 FNK 可视化的操作定义自己的数字资产,通过自定义的数字资产的带入,从而进行应用的移植和开发。在雾链 (FNK) 资产中, Fog Link OS 将会在传统终端作为切入点。

在前期 Fog Link OS 会支持雾计算盒子 (Fog Computing miner)、区块链手机、私人云盘等设备,随着业务的深入和雾链系统 (Fog Link OS) 的逐渐成熟,将会适配更多的硬件终端设备,同时雾链 (FNK) 将对自身的操作系统进行开源,更多的开发者可以使用雾链系统 (Fog Link OS) 进行自身的设备开发和移植,为区块链行业创造更大的贡献。

我们的设想是在初期通过雾链系统(Fog Link OS)为星际档系统 IPFS(Inter Planetary File System)和若干个雾计算节点进行价值传输的合理优化,本着优质资源优先流通的原则创建与 IPFS(Inter Planetary File System)的超级链接。符合 IPFS 节点的优质资源会通过 FNK 直接转化为 IPFS,在这个过程中 FNK 的中转功能将会以 Fog Mine pool 的概念进行呈现,所有符合 IPFS 价值的终端将会直接获得 IPFS 或是等值的 FNK,使雾计算的能力得到最大价值的利用。此外不符合为 IPFS 提供价值的外围设备,将会成为一个真正的雾计算价值输出终端,为雾链系统 (Fog Link OS) 的稳定运行提供算力。FNK 内部设备以及外围设备产生的算力,可以本着价值最大化的原则进行输出管道的转化,确保利用率和收益为最大化。

Fog Link OS 的原型设计中我们全部采用区块链技术和元素,让区块链合约 (Block chain Contract) 产生的过程具备绝对的流通和存储价值,同时 Fog Link OS 通过适配和移植,可以应用在主流的设备上。例如:在 Fog Link OS 应用在手机的设计上,用户联网启动设备时系统会自动分配钱包地址,用于存储自己的算力所得和资产的保存;此外联系人

属性里也具备钱包地址，用户可以利用通讯录功能进行资产流通。

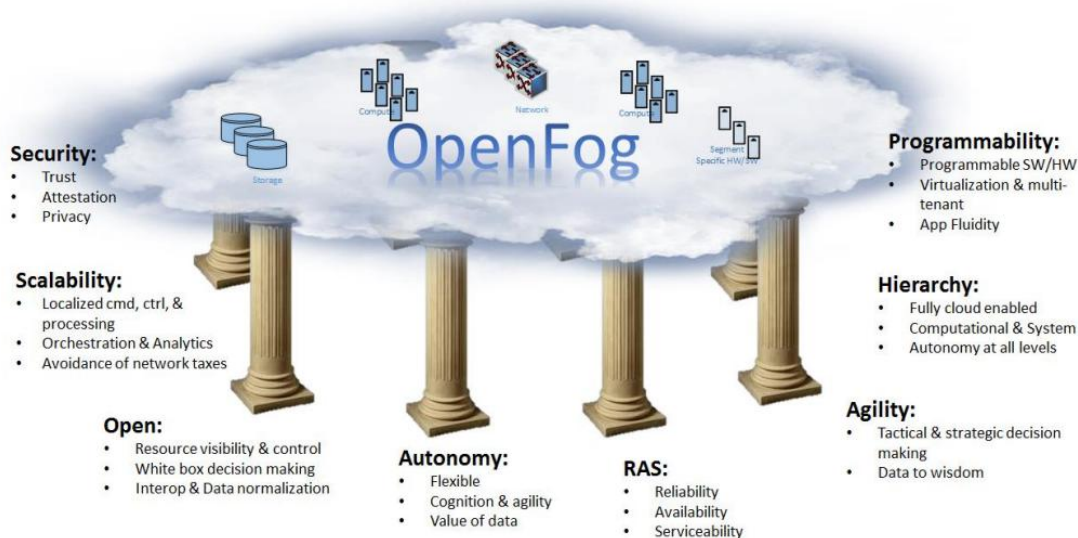
在安全方面，Fog Link OS 采用了特定的加密算法。在移动终端策略上，文字信息和图片信息全部采用加密的设计，利用 FNK 解决网络存储的问题；SIM 卡的关停和唤醒采用 AI 智能学习控制，最终在移动设备上可以实现零骚扰、物理位置转换、传输信息加密等实用的安全功能。最大限度的保护用户的隐私和财产的安全。

1. 雾计算与区块链的结合

1.1. 雾计算概念和特点

雾计算（Fog Computing）这个名字由美国纽约哥伦比亚大学的斯特尔佛教授起的，他当时的目的是利用“雾”来阻挡黑客入侵。后来思科首次正式提出，赋予雾计算的新含义。雾计算是一种面向物联网的分布式计算基础设施，可将计算能力和数据分析应用扩展至网络“边缘”，它使客户能够在本地分析和管理数据，从而通过联接网络获得实时的见解。

在 2012 年由萨尔瓦多等人在一篇关于云数据安全的文章中提出，通过使用假信息做诱饵，“钓”出窃密的“鼯鼠”，进而达到保护用户真实信息的目的与云计算将数据、数据处理和应用程序全部保存在云中不同，雾计算将他们分散在网络边缘的设备中。即在云服务器和物联网（IOT）设备之间，利用网络设备（路由器、手机、开关、机顶盒、代理服务器等）或者专用设备提供计算、存储和网络通信服务，使得数据和计算更靠近终端设备，进而降低云服务器的计算和存储开销，并且提高了应用系统的回应速度和网络带宽。“雾计算”这一名称是因为相对云而言雾更接近地面。雾计算没有强力的计算能力，因为提供算力的都是计算机周边和外围以及零散的计算设备。



图：OpenFog 开源雾计算价值图

雾计算 (Fog Computing) 主要使用的是边缘网络中的设备, 数据传递具有极低延时。雾计算具有辽阔的地理分布, 是具有大量网络节点的大规模传感器网络。雾计算移动性好, 手机和其他移动设备之间可以互相直接通信, 信号不必到云端甚至基站去绕一圈, 支持很高的移动性。

在 FNK 项目中, 雾计算部分采用 OpenFog 开源部分, OpenFog 可以满足运输行业的三个基本需求: 低延时, 维护用户隐私, 和不同层面的获取资源。

雾计算并非是一些性能强大的服务器, 而是由性能较弱、更为分散的各种功能的网络设备组成, 雾计算是半虚拟化的服务计算架构模型, 强调数量, 不管单个计算节点能力的强弱都要发挥作用。与云计算相比, 雾计算所采用的架构更呈分布式, 更接近网络边缘。雾计算将数据存储、数据处理和应用程序集中在网络边缘的设备中。雾计算是新一代分布式计算, 符合区块链的“去中心化”特征。自从思科提出了雾计算, 已经有 ARM、戴尔、英特尔、微软等几大科技公司以及普林斯顿大学加入了这个概念阵营, 并成立了非盈利性组织开放雾联盟, 旨在推广和加快开放雾计算的普及, 促进物联网和区块链行业的发展。

在雾链项目中关于 FNK 与 OpenFog 的部署, 从基础架构角度来看, 雾节点和雾层可以出现在 FaaS 中。通过 FaaS, 每层位置和节点部署, 不需要遵从单一数据中心。但是, 这样并不意味着不需要安全性。因为分布式数据存储和网络拓扑, 用户和雾服务提供者都面临安全性威胁。

安全基于“物”。这些物, 必须在受信硬件基础上。这种“可信根”, 必须通过其上运行的软件来证明。因为邻近最终用户以及边缘位置, 雾节点必须首先被进行访问控制和加密, 提供完整性和隔离, 控制隐私敏感的数据。随着更加复杂的拓扑结构产生, 整个雾节点“链”都要受信, 对于其他雾节点, 云端, 都要提供安全保证。因为, 雾节点也会动态实例化, 所以软硬件资源必须可信赖。非法的组件不可以参与到雾节点中。

安全实现, 可能有许多描述和属性例如: 隐私、匿名、完整性、信任性、证据、硬件可信根 (ROT), 验证和测量。

1.1.1. 可扩展性

对于驱动用户进行雾节点部署, 处理动态技术和业务需求很重要。雾网络中的可扩展性可以理解为必须通过一个有效的系统来进行实现, 其扩展性是通过适配更多设备的系统来进行。

1.1.2. 开放性

开放性是雾计算生态系统, 成功应用于物联网平台的关键因素。如果只为单个供应商所

有，限制供应商多元化，那么对于生态系统的成本、质量和创新都会产生消极影响。

1.1.3. 操作自主性

让雾部署可以具有面对外部服务失败的功能，同时也可以获取跨越整个层次的支持。在边缘网络中的自主，意味“本地设备”产生的智能和“同伴”的数据，可以完全满足业务需求。雾计算是通过靠近设备进行决策，不像云那样集中化决策，支持一些列自主功能，例如：可编程，带来高度自适应的部署，可以让雾节点和层，完成新的任务，容纳动态操作，完全自动化。

自主性，对于雾计算系统，实现快速机敏的业务操作决策来说很重要。FNK 设备系统产生的数据，和人类单独能够理解的数据，知识和决策的模式不同。在 OpenFog 架构中，自主性可以确保 FNK 系统设备产生有用数据，快速传输，并且自动快速决策，以及自动化处理。

数据，是信息系统的关键，OpenFog 架构也一样。由传感器和系统生成的数据，是混乱且突发的，有时数据量很大。最重要的是，数据没有上下文，上下文基于 FNK 系统内操作决策，上下文只有在数据被整理，聚合和分析的时候才用到。数据的分析可以在云层面进行，但是这样增加了延时，和多层传输带来的不确定性。所以，解决方法是将所有可操作的决策，在数据一旦被转化为有意义的上下文时，就做出。这样系统可以更快，更好的做决定。OpenFog 计算资源，可以按照 FNK 系统的功能需求，进行逻辑分层。雾网络，需要支持分层结构，具有本地，邻居，和区域级别，有效的划分计算任务。

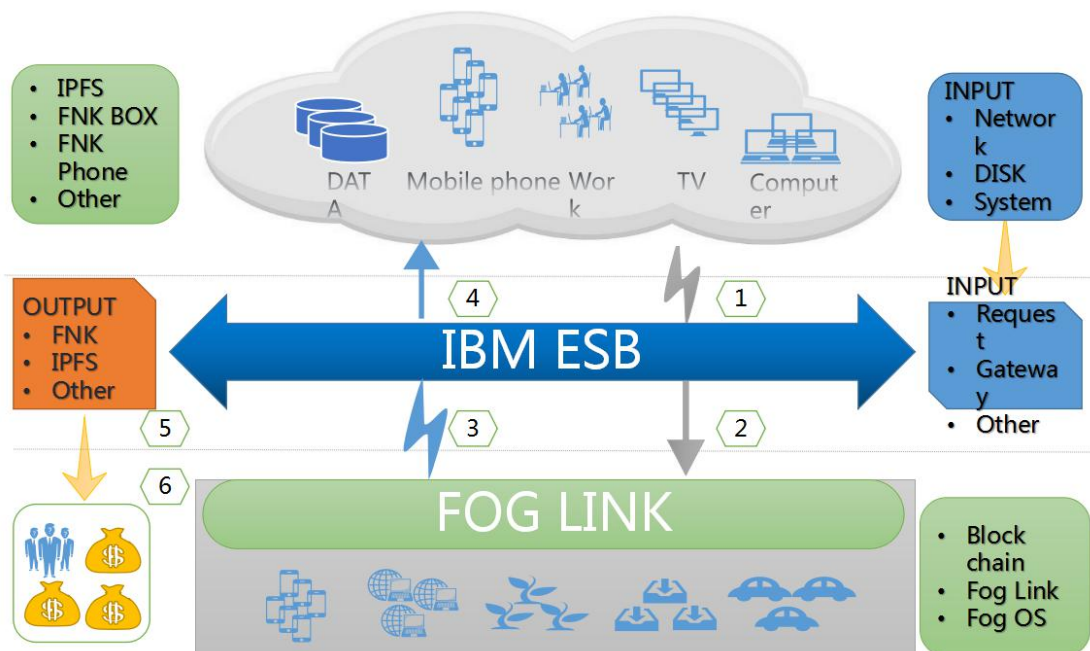
雾计算和云计算完全不同。云计算是以 IT 运营商服务，社会公有云为主的。雾计算以量制胜，强调数量，不管单个计算节点能力的强弱都要发挥作用。雾计算扩大了云计算的网络计算模式，将网络计算从网络中心扩展到了网络边缘，从而更加广泛地应用于各种服务。雾计算有几个明显特征：低延时和位置感知，更为广泛的地理分布，适应移动性的应用，支持更多的边缘节点。这些特征使得移动业务部署更加方便，满足更广泛的节点接入。

1.2. 雾计算与区块链技术的结合

雾计算架构，使用大量边缘设备和计算终端，与传统云服务一起，进行数据存储、计算、网络连接以及管理相关的任务。雾计算架构和传统架构相比，特征如下：

- (1) 在用户和商业集中的位置附近部署，进行低延时存储；
- (2) 设备形态小，重量轻，便于存放和携带；
- (3) 设备展现形式多样化，对运行系统要求低，便于移植；
- (4) 靠近最终用户进行运算，避免延时，降低网络和带宽损耗；

- (5) 低延时通信，而不是所有通信都要经过骨干网路由进行同步；
- (6) 靠近最终节点实现管理元素，包括网络测量，控制和配置；
- (7) 可靠性/可用性/可服务性（RAS）；



图：Open Fog 与区块链的结合

区块链技术可以理解为如果我们把数据库假设成一本帐本，读写数据库就可以看做一种记账的行为，区块链技术的原理就是在一段时间内找出记账最快最好的人，由这个人来记账，然后将帐本的这一页信息发给整个系统里的其他所有人。这也就相当于改变数据库所有的记录，发给全网的其他每个节点，所以区块链技术也称为分布式帐本（Distributed Ledger）。结合雾计算和区块链的特点，用户可以利用 FNK 的雾计算设备来提交特定的磁盘和网络资源，从而获得 FNK 数字资产。

FNK 将区块链合约（Block chain Contract）作为资产管理者进行数字资产的管理。在雾链（FNK）资产中，Fog Link OS 会生产雾计算盒子（Fog Computing miner）、基于区块链的手机、私人云盘等设备。

随着业务的深入和雾链系统（Fog Link OS）的逐渐成熟，将会适配更多的硬件终端设备，同时雾链（FNK）将对自身的操作系统进行开源，更多的开发者和厂商可以使用雾链系统（Fog Link OS）进行设备开发和移植，同时可以发布价值合约（Value Contract）、应用合约（Developer Contract）以及混合合约（Mixed Contract）三种合约，此三种合约可以理解为 POS（全称 Proof of Stake，股权证明）和 POW（全称 Proof of Work，工作

证明) 以及 POS+POW 的混合合约。

2. 雾链的设计原则

2.1. 雾链的模块化设计

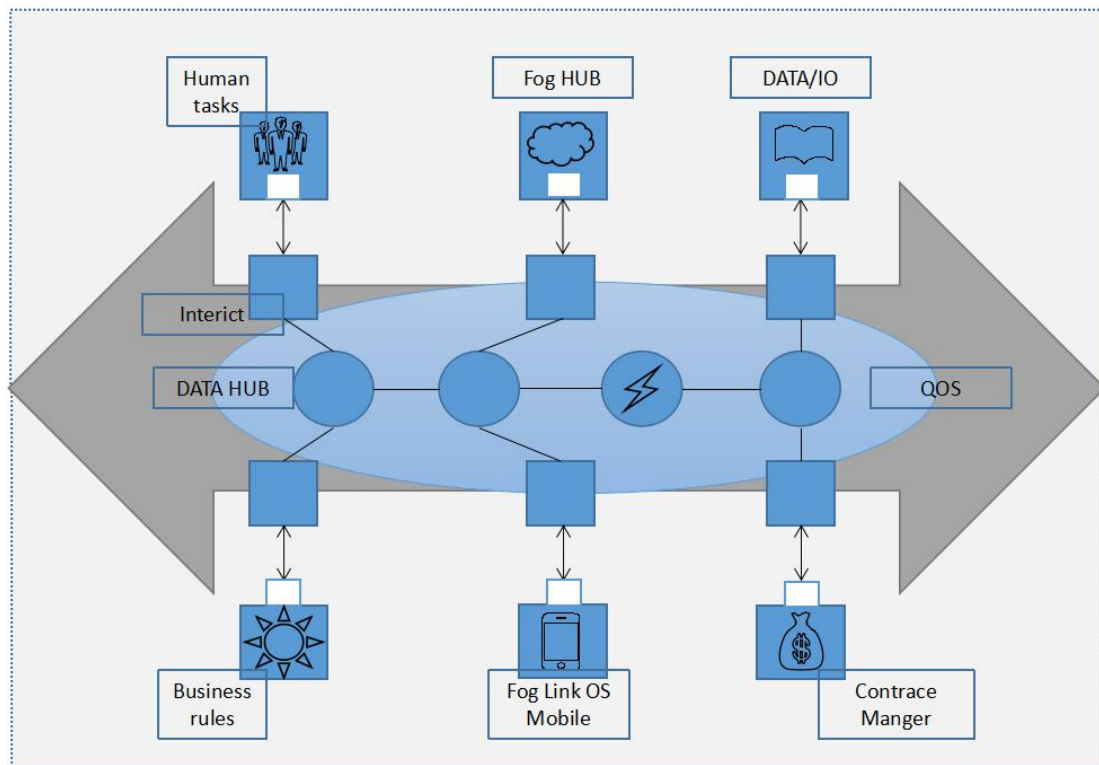


图：FNK 系统模块耦合图

雾链的初始化设计是采用模块进行搭建，其中包括合约管理模块 A、合约管理模块 B、交易模块、中间件模块、输入输出模块等。

在推动应用实施的过程中，开发者可以使用自己所需要的模块进行软件开发，在雾计算中最大的特色就是本地编辑，雾端实现。

2.2. IBM ESB 总线与 FNK 结合的架构方案

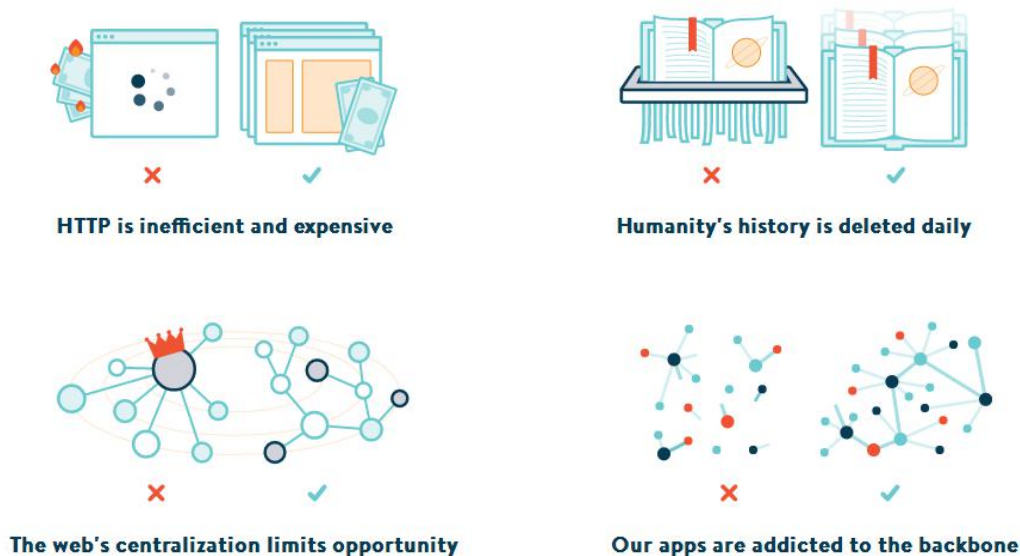


图：ESB 与 FNK 的结合设计图

在区块链领域中，雾链是首个采用 IBM ESB 技术来实现去中心化分布式的网络节点公链。IBM 有三款 ESB 产品：WebSphere ESB (WESB)，WebSphere Message Broker(。在雾链中使用的 ESB 总线为 WMB，强大的处理速度和无缝的网络资源切换，可以让 FNK 系统不受到雾计算的稳定性带来的干扰，进而可以使设备更加安全稳定的运算。

在中心化的 ESB 总线中，实现去中心化的管理，完全取决于数据的存储和传输的机制，在雾链中虽然使用了 ESB，但我们在实验室阶段就开始了 ESB 与去中心化机制的流程测试，利用雾计算节点的规范化，已经试验论证通过，完全可以保障矿工和雾计算提供者的利益。当大量的设备加入到雾链之后，ESB 的安全和高效是显而易见的。尽管如此，雾链的系统架构方案中还借鉴了星际档系统 IPFS(Inter Planetary File System)，用于兴趣网络的持久性链接，在人口较为集中的城市里，设备的运行效率和算力的回报会更加的凸显。

2.3. IPFS 的网络特征



图：IPFS 基于传统网络和存储协议做出的改变

IPFS 颠覆了 HTTP 协议，通过去中心化的网络将价值传输合规到自身的协议中，在 IPFS 的价值传输过程中，FNK 的盒子通过 IBM ESB 稳定了算力系统，继而为 IPFS 输出更多合规的算力值。

2.4. 雾链的安全性和性能

WMB 本身提供了两个层次上的安装，一个是部署时安全性，管理部署 bar 档到 Broker 以及运行 WMB 管理命令的权限控制；另一个是运行时安全，涉及的权限控制包括发送消息到相应的消息流，以及消息流可以访问哪些 MQ 资源和非 MQ 资源，如数据库系统。

WMB 底层是使用 C++ 开发的，在性能上相对于 WESB 有很大的提高，同样在 FNK OS 的底层上，也采用了 C++ 进行开发，可以与 WMB 进行很好的协同，可以达到处理的消息数量级为几千到几万之间。

2.5. 雾链的系统化策略

雾链的系统的优化得益于 IBM 的 ESB 中间件解决方案，从 IBM 的立场来说，ESB 不仅仅是一个概念，而是一种中间件模式；它不是某个产品，而是一种全新的集成应用，协调资源和操纵信息的框架。

在利用雾计算在为区块链提供算力的这个过程中，设备类型的多样化和 AI 的自主判断

能力是无法应对外围设备所做出的改变的。

但雾链是一个具有 ESB 特征的产品, 对于一个具有 ESB 类的产品, 管理也是一个重要的方面。在 FNK 进行雾计算过程中, 需要用若干个较低的能耗设备, 解决合约的管理和稳定的流通以及各类的应用场景。在这个过程中, IP 地址和磁盘都会存在不可预知的不稳定性, 例如, 当一个服务从一个地址切换到另一个地址, 在结构等不发生改变的时候, ESB 类产品会提供一个方便的途径适应这种改变。而雾链就是采用这种技术, 在雾计算中适应不同的环境和条件的转化, 从中做出最优的选择。

WMB 是 ESB 的一个高级版本, 提供了很多功能强大的内置节点支持消息的路由, 如 Filter 节点、Label 节点等, 在新版本的 WMB 中又引入了 Router 节点, 该节点几乎和 WESB 中的 Router 节点一样。若需要实现动态路由, 可以使用 WSRR 作为服务的存储, WMB 和 WSRR 有很好的集成, 通过 RegistryLookup 和 EndpointLookup 我们可以在消息流中实现动态路由。

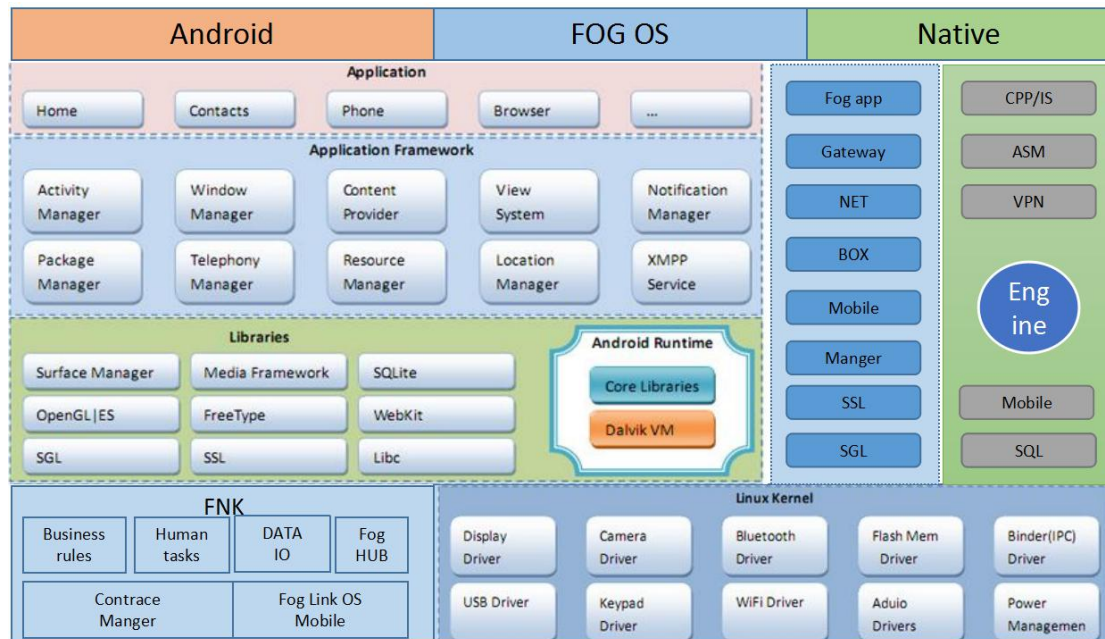
3. 雾链系统 (Fog Link OS) 解决方案

3.1. 雾链系统架构

Fog Link OS 的系统架构设计原则就是本着实用可移植原则, FNK 超级节点、星际档系统 IPFS(Inter Planetary File System)以及雾计算节点三种类型网络节点将会在 Fog Link OS 中进行最佳的配置, 在 Fog Link OS 系统中, 采用 C++ 语言重新对 ARM 和 Native 进行编写匹配, 优化了 Android 系统运行中卡顿的用户体验, 在引擎中全部使用 C++ 语言, 执行效率会高于 JAVA 很多。

在 Fog Link OS 系统中, 针对不同的终端设备需要进行独立的适配, 尽管适配的工作量加大, 但是支持的设备终端类型和可编程性也是区块链手机操作系统必须具备的。

在 Fog Link OS 系统中用 C++ 代码对设备端的 CPU 运算效率进行了极致的优化, 使输入和输出的流程符合物理价值传输体系 (Physical value transmission system), IBM ESB 总线在中间件的角色上也起到了强大的协调作用, 使系统变得更加流畅高效。



图：FNK OS 系统架构图

上图为雾链的系统架构图，该图中的模块化设计通过与 LINUX 的共有底层实现了与 Android 的高度兼容，在为手机提高运算效率的同时，让更多的设备和芯片参与其中。

3.2. 雾链系统 (Fnk OS) 应用场景

Fog Link OS 的系统支持 ARM 和 X86 架构，若干个分布式计算的不规则节点（网络 and IO 的能力不同）形成庞大的雾计算网络，通过该网络进行数据传输和分布式的存储。在应用场景中，基于 Fnk OS 的操作系统，描述如下：

当下市场上并没有真正意义的区块链手机，通过调研，用户也认为区块链手机应该具备传统手机不具备的功能，而这些功能需求都是高难度并且非常实用的。根据我们设计的区块链手机方案，在前期可以满足 80% 的用户需求，此类需求大概为：

3.2.1. 雾计算区块链手机

提高 Android 手机的启动速度和运行效率，在此部分满足上，Fnk OS 针对系统的启动部分用 C++ 进行了重构，可以提高启动速度。而在效率上，我们使用了国防级的沙箱容器，通过沙箱来管理 App 的运行效率和安全。

3.2.2. Fnk BOX

私人专属云盘（Fnk Box），此部分功能主要还是为发烧级用户提供代表着雾算力的工作量和工作效率证明，通过用户提供的算力，按照带宽、存储、效率、难度等方面为用户提供 POW 证明。在 Fnk BOX 的原型设计中，支持视频播放和云盘的功能，而云盘的功能利用骨干网络和雾计算节点为用户的私有数据存储进行加密和传输。

在 Fnk Box 进行数据传输的过程中，调动了其他用户的设备相应，从而等同于其他用户在进行协同式的分布式计算，从而为他带来了工作量证明（POW）。

3.2.3. 区块链手机沙盒

这个沙盒功能，也是区块链手机系统里的核心部分，也是最不可缺少的功能。

沙盒可以利用各种策略为用户提供覆盖“端、管、雾”全方位、一体化的移动安全评估、安全检测、安全加固、安全沙箱、安管平台等产品和服务，可以有效保护开发者的自身权益、有力保障企业移动化办公的可靠性，最终保护个人用户的数据安全，产品完全可以应用于互联网金融、电子商务、电子政务、智慧城市、智慧医疗等。

通过沙箱数据的备份和恢复机制，可以满足用户在雾计算终端设备上的无缝迁移、数据备份、数据恢复，即使在不同的终端设备上，也可以实现系统环境的统一，避免因设备切换时产生数据不对称。

实现完全独立运行的沙箱桌面，支持基本的沙箱设置、应用管理，终端沙箱应用同时支持 APP SDK 和 Wrapping 两种移动应用沙箱化方式；沙箱桌面完全支持应用分发、内置沙箱应用、沙盒管理全部功能；实现可对沙箱桌面进行控制的管理平台，可以对沙箱桌面和沙箱应用进行全面的平台总体、用户管理、数据管理、应用管理、设备管理、策略管理、密钥管理、备份恢复；

3.3. 雾链代码示例

在这个代码片段里，以雾链的启动代码与 Android 进行比对，在启动流程上 Android 和 FNK OS 的启动都分为进程初始化和系统 Framework 启动两步骤。

下面截取的代码片段中，展示了二者系统启动的不同，以及 FNK OS 如何对 Android 进行了替代。

Android 系统启动过程

①Linux init 进程初始化

```

1 service uventd /sbin/ueventd
2 service logd /system/bin/logd
3 service logd-reinit /system/bin/logd --reinit
4 service healthd /sbin/healthd
5 service console /system/bin/sh
6 service adbd /sbin/adbd --root_seclabel=u:r:su:s0
7 service servicemanager /system/bin/servicemanager
8 service zygote /system/bin/app_process -Xzygote /
system/bin --zygote --start-system-server
9 service vold /system/bin/vold
10 service netd /system/bin/netd
11 service ril-daemon /system/bin/rild
12 service surfaceflinger /system/bin/surfaceflinger
13 service drm /system/bin/drmservice
14 service media /system/bin/mediaserver
15 service defaultcrypto /system/bin/vdc --wait crypt
tfs mountdefaultencrypted
16 service encrypt /system/bin/vdc --wait cryptfs en
ablecrypto inplace default
17 service gatekeeper /system/bin/gatekeeper /data
/misc/gatekeeper
18 service installld /system/bin/installld
19 service flash_recovery /system/bin/install-recover
y.sh
20 service racoon /system/bin/racoon
21 service mtpd /system/bin/mtpd
22 service keystore /system/bin/keystore /data/misc/
keystore
23 service dumpstate /system/bin/dumpstate -s
24 service mdnssd /system/bin/mdnssd
25 service afm_server /system/bin/afm_server
26 service uncrpt /system/bin/uncrpt

```

FogLink OS 启动过程

①Linux init 进程初始化

```

1 ...
2 service foglink /system/bin/foglink

```

②Android framework 启动

```

1 int main() {
2     AppRuntime runtime;
3     if (zygote) {
4         runtime.start("com.android.internal.o
s.ZygoteInit",
5             startSystemServer ? "start-system-s
erver" : "");
6     }
7 }
8 // 在这里加载 Android dalvik/art 虚拟机, 并加载运行 java
代码, java 接管整个UI
9 void AppRuntime::start(const char* className, const
char* options) {
10
11     // start the virtual machine Java
12     JNIEnv* env;
13     if (startVm(&mJavaVM, &env) != 0) {
14         return;
15     }
16
17     if (startReg(env) < 0) {
18         return;
19     }
20
21     jmethodID startMeth = env->GetStaticMethodID(
startClass,
22         "main", "(Ljava/lang/String;)V");
23
24     jclass startClass = env->FindClass(className
);
25
26     env->CallStaticVoidMethod(startClass, startM
eth, strArray);
27 }

```

②Fog Link OS 启动过程

```

1 int main() {
2     load_config_file();
3     FogFramework fogos;
4     fogos.start();
5 }
6 //为提高CPU/内存使用效率, FFK OS 完全由C++处理所有UI, 不加
载 dalvik/art虚拟
7 //机, 这是 FFK OS 与Android 最大的不同
8 void FogFramework::start() {
9     start_fog_service_manager();
10    start_fog_framework();
11    start_fog_ipc();
12    binding_scripts();
13    dispatch_system_events();
14 }
15 // 在FKK OS 中运行 Android 应用时由于系统化的 Android 框
架执行
16 void FogAndroidSubSystem::startApplication(int argc
, char* argv[]) {
17
18 }

```


3.4. 雾链系统设备

3.4.1. 雾链盒子

FNK 盒子可以达到星际档 (IPFS) 传输系统的协议要求, 配备千兆网卡和 USB3.1 界面, 设备自身支持 1-5 盘位, 满载功耗不超过 45W。

为了满足大磁盘阵列的要求, 我们按照磁盘阵列的外观设计了盒子的 DEMO, 内置 FNK OS, 硬件同时支持多种类型的磁盘。

在实用性的角度上考虑, FNK 的磁盘阵列设计方式为用户节约了大量的后续维护成本, 在安装使用时, 无需单独购买移动硬盘, 开机后进入系统预先设置好的 RAID 容量逻辑。

雾链的盒子产品是一台具备星际档系统传输的设备, 在为盒子的持有者提供云盘和媒体中心功能外, 还可以为 IPFS 提供一部分能力输出以及 Fog Link OS 系统运行所需的网络资源。FNK 盒子目前是系统和硬件部分同时进行, 下图是盒子的工业设计和效果图的部分截取。

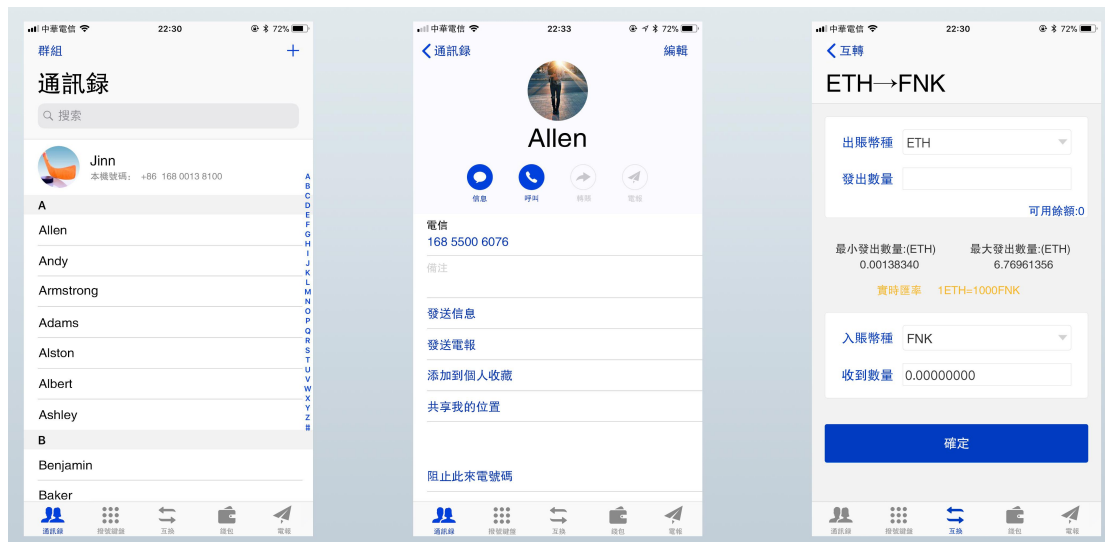


图：FNK 雾计算盒子 DEMO 图

3.4.2. 区块链手机

Fog Link OS 在适配和调优后, 可以直接刷到手机系统里, 在手机项目的布局中, 2014 年我们的系统已经可以在 HTC、Motorola 等手机中运行, 并实现了与 Google Android 的高度兼容。

我们会继续完善 UI 和用户体验部分, 力争发布全球首台区块链手机, 为去中心化的产业布局迈出实质性的一步。



图：Fog Link OS UI 设计

上图是 Fog Link OS 在手机上应用的截图，这是全球首款手机操作系统的深度移植，所以在 UI 设计上一切都是设想，最终以实际研发成果为准。

3.5. 雾链开源计划

在去中心化的区块链产业里，是一个开放融洽的，我们希望在 FNK 上线后直接进行源码的开放，同时也包括 FNK 中的 OS 部分。

开发者可以很方便的参与到 FNK 的建设完善中，我们针对软件开发者、游戏开发者、设备制造者、雾计算贡献者等群体，提供大量的 Token 支持，所有参与者都可以根据贡献来获得 FNK 的糖果。

3.6. 雾链社区

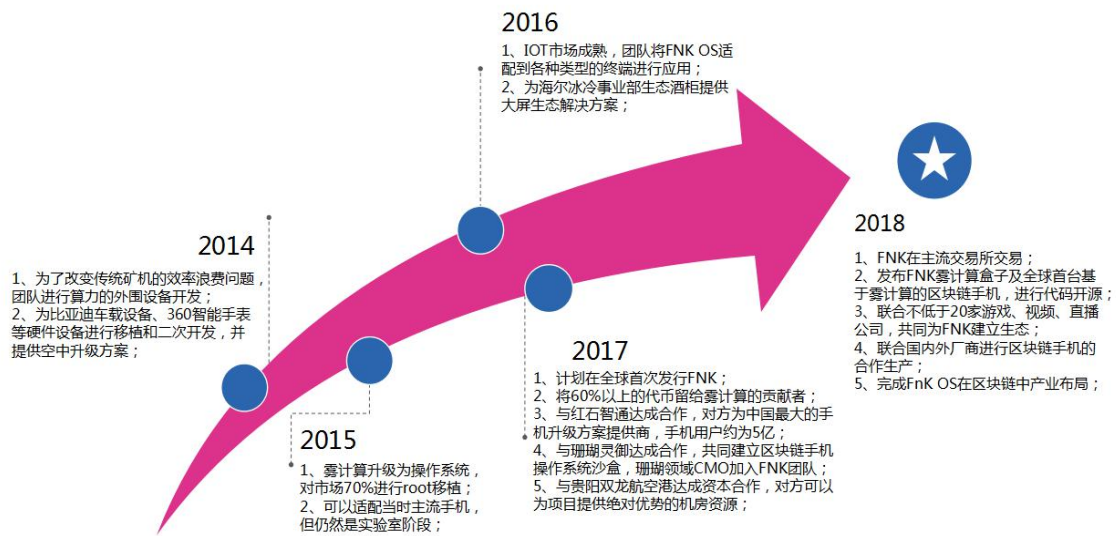
我们会为爱好者建立雾链社区，同时雾链社区也会扶持有潜力的开发者团队。目前市场上闲置的终端设备数量非常多，操作系统也不统一，我们很想通过闲置设备的再利用，按照规范进行能力的划分，使闲置的设备可以继续体现其价值。

3.7. 联盟组织

在雾计算的生态系统中，是由众多角色共同创建的生态环境，在此前我们与 IPFS 官方取得联系，并为 IPFS 技术协议进行推广及合作。

IBM 和 Open Fog 在技术上为我们提供了非常大的帮助，后续我们希望并期待更多的企业和个人能加入到 FNK 联盟中来。

3.8. Road Map



团队于 2014 年正式成立至今，主要致力于区块链硬设备和基于雾计算领域中的 IOT 开发，通过团队的 FNK OS 产品准备于 2018 年发布全球首款基于雾计算的区块链手机产品，并对产品代码进行开源。

4. 团队简介

4.1. 项目团队成员



Bruce Song

- 中科院网络安全博士；
- IBM 高级系统架构师；
- 国家重点实验室高级架构师、大数据专家；
- 在分布式计算、并行处理、图像识别、语言识别、自然语言理解、深度学习、数据挖掘等技术领域有深入的理论研究和丰富的实践经验，在电信、电商、金融、军工、航天等业务领域有丰富的大数据开发经验；



Job

- 前索尼爱立信移动通讯公司 (Sony Ericsson) 高级硬件工程师；
- 移动新媒体瑞媒创始人；
- 电信运营商诚信合伙人；
- 专注手机产品和技术领域，在移动游戏，移动广告，移动支付等领域多年运营推广经验；



Fed Wu

- Fog Link OS 操作系统创始人；
- Fota 业务高级架构师
- 千橡互动移动事业部高级工程师；
- 美国媒体摩比软件工程师；
- 专注移动广告、新媒体运营、第三方支付、操作系统、移动端设备升级等领域；



Jinn

- 清华大学 EMBA；
- 宇龙酷派副总裁；
- 摩托罗拉 (Motorola Inc) 副总裁；
- 赛龙通信公司副总经理；
- NDAYS 手机系统创始人；
- 在工业设计、电路设计、手机设计、智能硬件设计、语言识别、AI 智能等领域具有丰富的经验，MOTO V3 系列手机主导者；

**Brook Yang**

- 印度国家信息学院 北京航空航天大学
- 酷派手机互联网事业部副总裁；
- 千橡互动高级软件工程师；
- 中国移动基地项目经理；
- 在区块链技术应用、共识算法、总体技术框架等技术领域有深入的理论研究和丰富的实践经验，在电信、电商、金融、区块链等业务领域有丰富的大数据开发经验；

4.2. 顾问团队

**郭宏才**

数字货币领域知名人士，比特币以及以太坊初创公司的著名天使投资人，对于虚拟货币及去区块链行业有多年研究和探索。

**陶 鹏**

- Johns Hopkins University 统计模式识别博士；
- 数理统计科学和计算机科学双硕士学位；
- Silicon Valley 多家 Pioneer 公司核心算法工程；(Acculmage /Merge Health care 及 Symphony 等)
- 专注算法学、图像识别、大数据分析、金融、互联网媒体、勘探等多个领域；

4.3. 早期投资人

Julia 安芙兰资本 创始人/董事长

上海交通大学应用化学学士，中国人民大学工商管理 MBA 硕士，长江商学院 EMBA。1995-2004 年间先后创办多家实业公司，拥有丰富的实体经营管理经验。2004-2015 年间成功设立管理咨询公司和多支私募股权投资基金，对中国资本市场具有较为深刻的认知和丰富的实践操作经验，投资的企业已有多家在国内及香港成功上市，是资本市场颇具影响的专业投资人。

5. 关于本白皮书

本白皮书只做交流之用，其中包含的信息或分析内容不能构成购买提议或劝导。本白皮书也不应被理解成为提供的技术指导和买卖行为，也不是任何形式上的合约或者承诺。

需要对技术进行商讨，请通过下面的 Email 和我们取得联系。

support@foglink.io

电报群：

<https://t.me/fnkfans>

<https://t.me/joinchat/IDjQORDd4nFtVav4C3x1>

参考文献：

<https://foglink.io/>

<https://ipfs.io/>

<https://www.ibm.com/developerworks/cn>

<http://www.openfogconsortium.cn/>

《OpenFog Architecture Overview》

感谢：

