亿书
Ebookchain.org

区块链是如何实现版权保护的

朱志文@imfly

2016.11.26

我是谁？

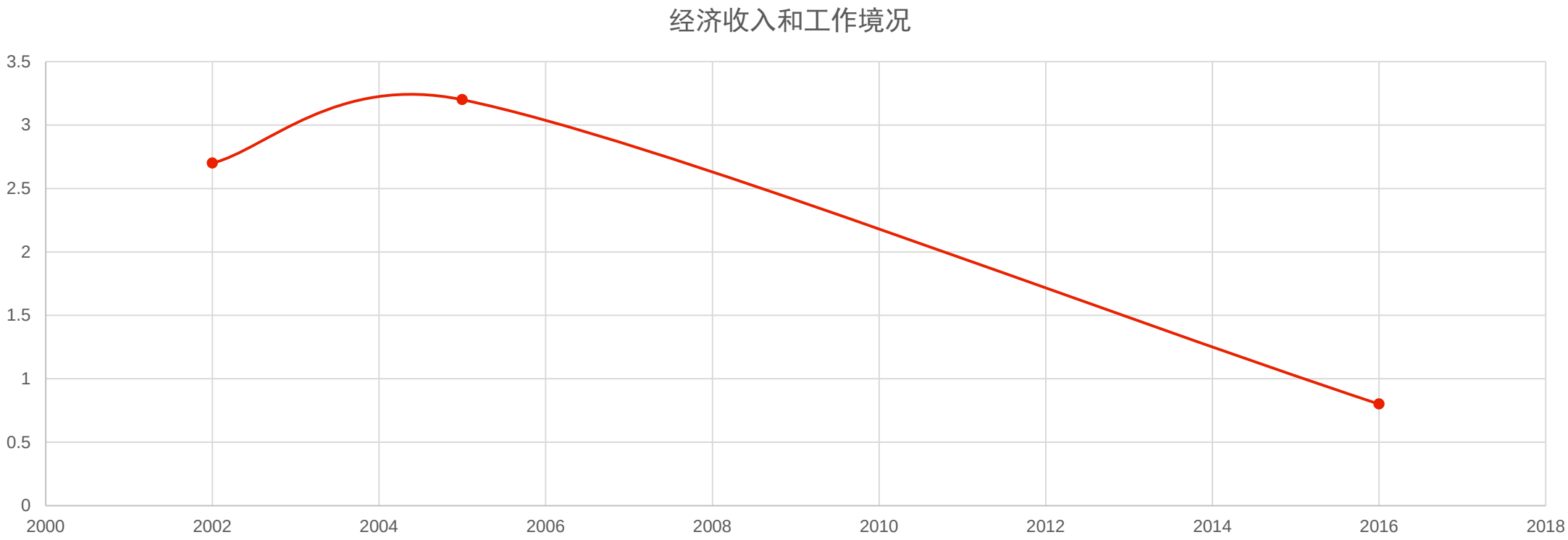
朱志文

梦　想：坚持做一个能支配大脑的程序猿
爱　好：从工作中结交朋友，与喜欢的人一起成长
恶　习：拒绝一切重复的事情
口头禅：退一步也比昨天强
喜欢的签名：做喜欢、擅长、有价值的事情是成功的开端
讨厌的事情：说言不由衷的话，做违背初衷的事

我在做什么？

- 专心研发亿书，让盗版无所遁形；
- 学习传播区块链技术，拉更多喜爱技术的小伙伴入“坑”，为国内技术进步贡献微薄力量；
- 希望5年后，亿书产品和亿书社区，会成为国内开源产品的重要代表之一，把区块链技术的研发和应用降到几乎“0”成本；

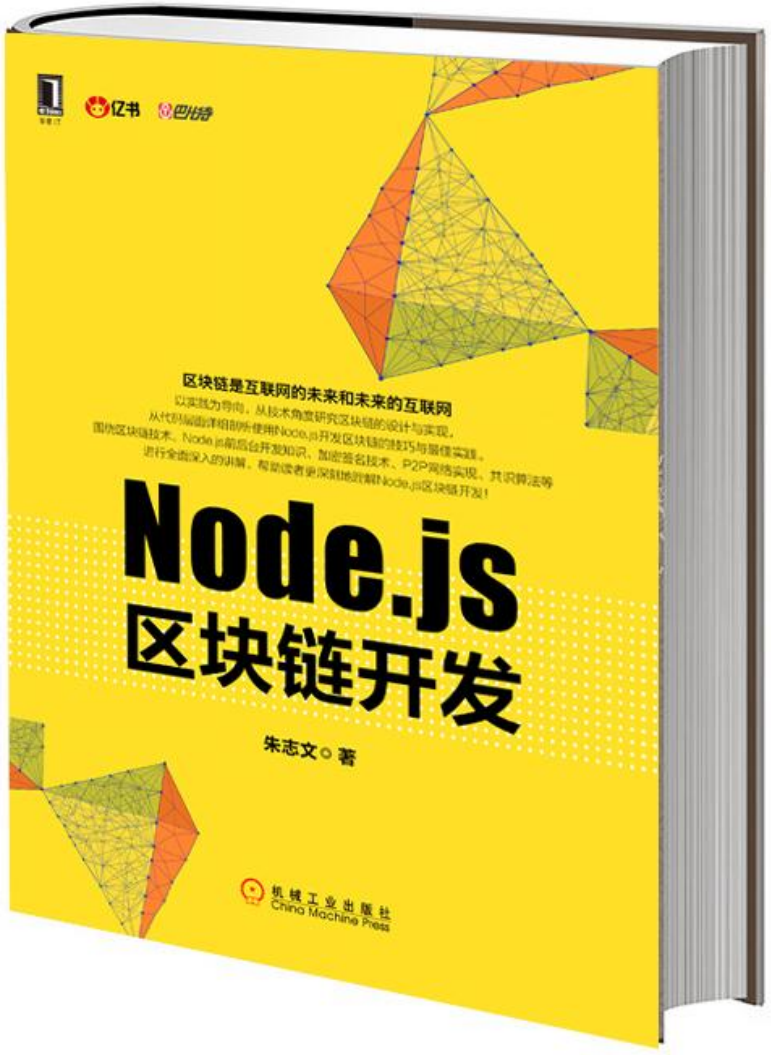
记者朋友的故事


经济收入和工作境况
3.5
3
2.5
2
1.5
1
0.5
0
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018

在行动中了解需求

不是广告的广告

本书原名《Nodejs开发加密货币》
网络分享永远免费提供，地址：
https://github.com/imfly/bitcoin-on-nodejs



一、我是怎么开始探究版权保护的

二、数字出版领域的主要困境

三、当前版权保护技术的方法和局限

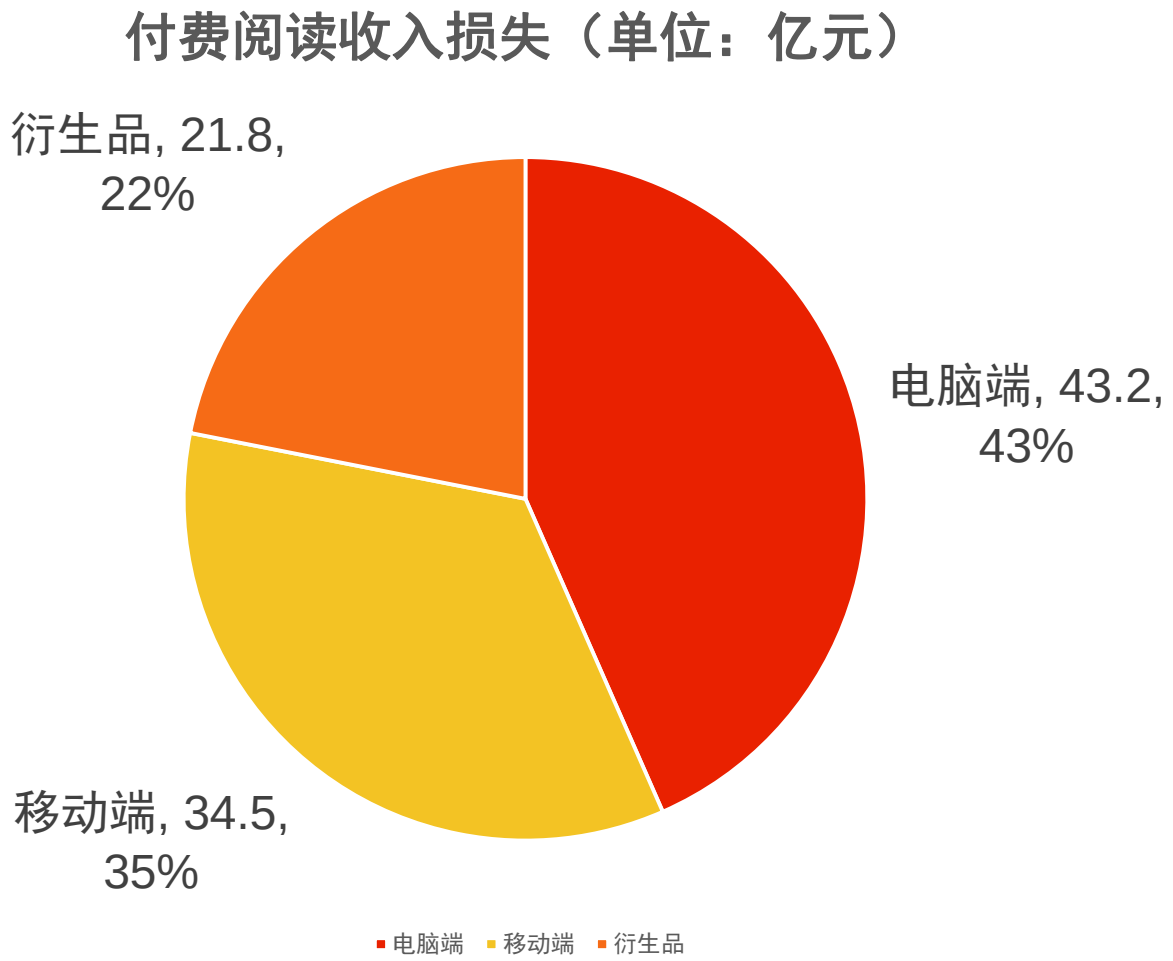
四、区块链在版权保护上的主要特点

五、区块链在版权保护上的基本实现

六、智能合约在版权保护中的初步探索

主要困境

1、碎片化严重
2、侵权盗版严重
3、知识创新者没有主导地位



《2015年中国网络文学版权保护白皮书》：2014年，盗版网络文学直接损失

某位作家发了一条这样的微博：“……苦逼的编辑们，揣着高学历，名牌大学的文凭，吃着盒饭，挤着公交，坐地铁上看稿子，每晚星星齐了回家，给女友吻都送不及，倒在沙发上睡了。编辑苦，出版人苦，作者同样苦极……”

主要问题

- 登记确权流程繁琐
- 调查取证手段匮乏
- 法律法规亟待完善

国家政策层面的做法

- 2014年8月18日，中央全面深化改革领导小组第四次会议审议通过了《关于推动传统媒体和新兴媒体融合发展的指导意见》，习总书记作了重要讲话。
- 2015年，李克强总理在政府工作报告中提出政府的工作重点，首次提出“互联网+”行动计划、“大众创业，万众创新”。财政部等中央部委，纷纷下发文件，拿出专项资金扶持推动传统媒体和新兴媒体融合发展。
- 对应的法律法规也相继出台，各类版权保护的行动开始实施

一、我是怎么开始探究版权保护的
二、数字出版领域的主要困境
三、当前版权保护技术的方法和局限
四、区块链在版权保护上的主要特点
五、区块链在版权保护上的基本实现
六、智能合约在版权保护中的初步探索

传统的技术手段

1、公众认知确认法
2、推定代理法
3、网络注册号和密码验证法
4、电子备案和著作权登记法

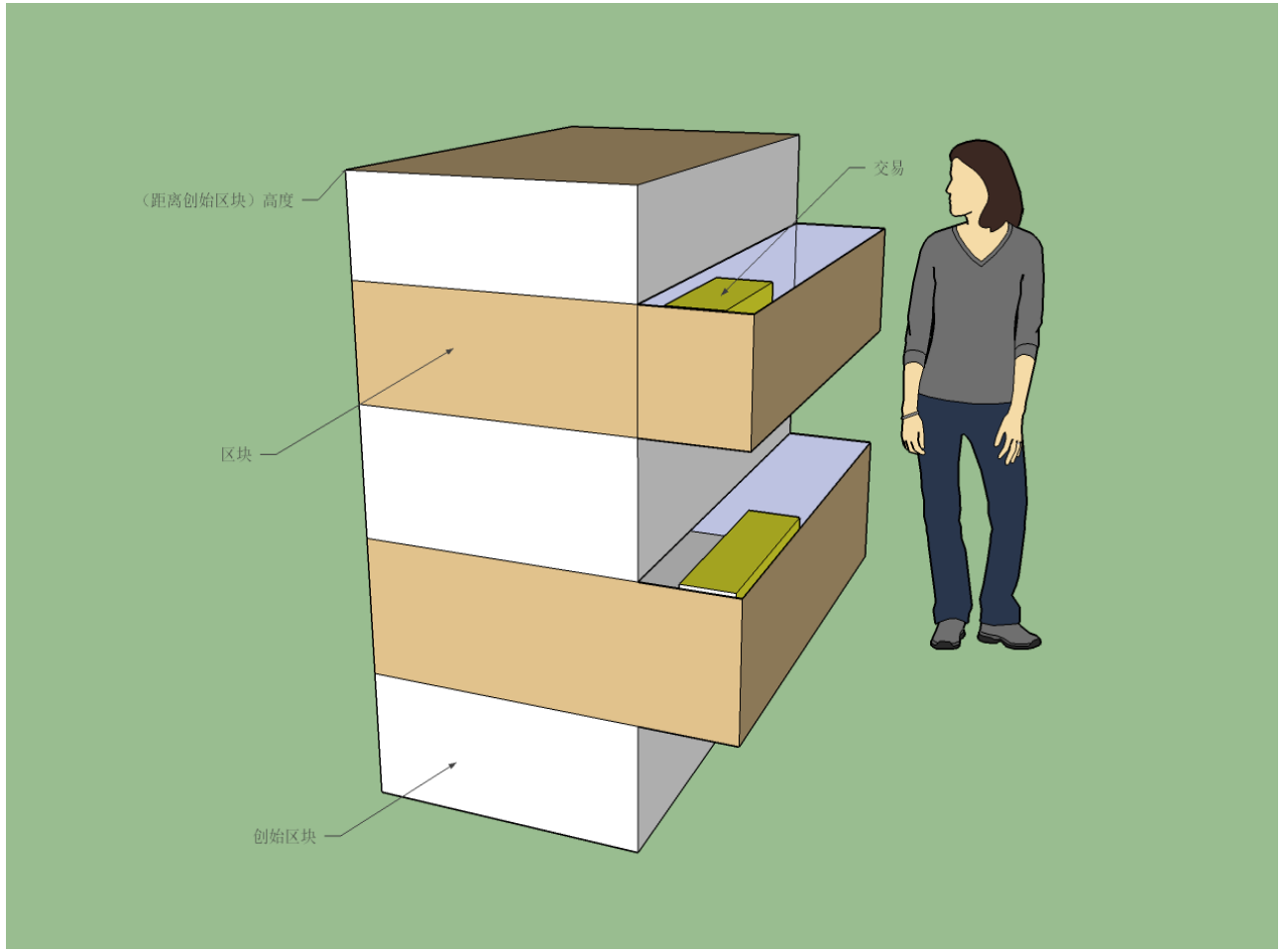
传统技术手段的局限

1、不够精确，容易出错；

2、基于中心化的网站，容易遭受攻击；另外，中心化的网站，一般都可以人为操纵，没有严格可信的可追溯性，给调查取证带来严重问题；

3、最为人熟知的“著作权登记”，因其费用高、费时长，也无法满足网络时代作品“产量多、传播快”的特点

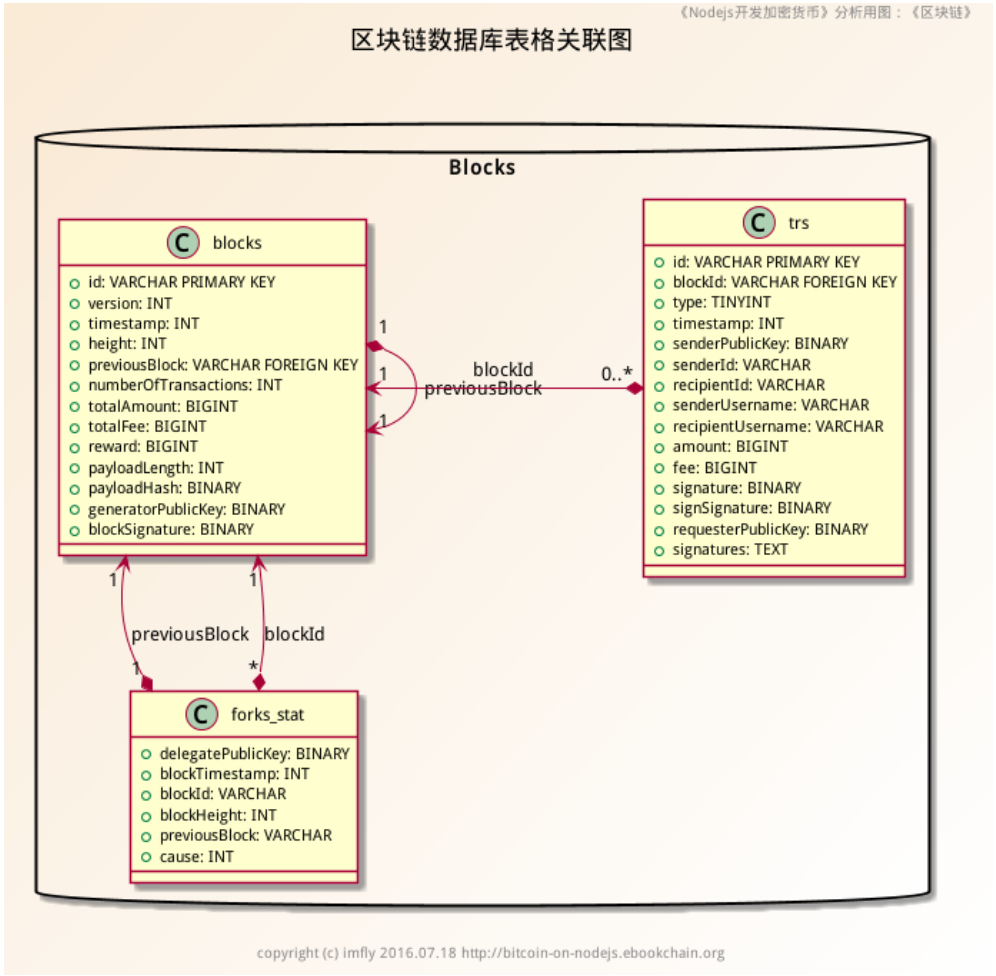
一、我是怎么开始探究版权保护的
二、数字出版领域的主要困境
三、当前版权保护技术的方法和局限
四、区块链在版权保护上的主要特点
五、区块链在版权保护上的基本实现
六、智能合约在版权保护中的初步探索



区块链是什么？

狭义的理解，就是一张公开存储、无法更改的“自引用”数据库表。

亿书的区块链数据结构


《Nodejs开发加密货币》分析用图：《区块链》
区块链数据库表格关联图
Blocks
blocks
id: VARCHAR PRIMARY KEY
version: INT
timestamp: INT
height: INT
previousBlock: VARCHAR FOREIGN KEY
numberOfTransactions: INT
totalAmount: BIGINT
totalFee: BIGINT
reward: BIGINT
payloadLength: INT
payloadHash: BINARY
generatorPublicKey: BINARY
blockSignature: BINARY
trs
id: VARCHAR PRIMARY KEY
blockId: VARCHAR FOREIGN KEY
type: TINYINT
timestamp: INT
senderPublicKey: BINARY
senderId: VARCHAR
recipientId: VARCHAR
senderUsername: VARCHAR
recipientUsername: VARCHAR
amount: BIGINT
fee: BIGINT
signature: BINARY
signSignature: BINARY
requesterPublicKey: BINARY
signatures: TEXT
1
1
1
blockId
previousBlock
0..*
1
1
previousBlock
blockId
1
*
forks_stat
delegatePublicKey: BINARY
blockTimestamp: INT
blockId: VARCHAR
blockHeight: INT
previousBlock: VARCHAR
cause: INT
copyright (c) imfly 2016.07.18 http://bitcoin-on-nodejs.ebookchain.org

区块链的特点

- **分布存储**：区块链处于P2P网络之中，无论什么公链、私链，还是联盟链，都要采取分布式存储，使用一种机制保证区块链的同步和统一;
- **公开透明**：每个节点都有一个区块链副本，区块链本身没有加密，数据可以任意检索和查询，甚至可以修改（改了也没用）;
- **无法篡改**：这是加密技术的巧妙应用，每一区块都会记录前一区块的信息，并实现验证，确保无法篡改。这里的无法篡改不是不能改，而是局部修改的数据，无法通过验证，要想通过验证，必须修改整个区块链，这在理论上可行，操作上不可行;
- **方便追溯**：区块链是公开的，从任一区块都可以向前追溯，直到第一个区块，并通过区块查到与之关联的全部交易

区块链如何能解开IP确权的难题？

1. **记录所有环节**。在所有涉及版权的使用和交易环节，区块链都可以记录下使用和交易痕迹，并且可以看到并追溯它们的全过程，直至最源头的版权痕迹。
2. **去中心化**。它打破了现在的从单点进入数据中心去进行版权确权的模式。区块链技术可以实现多节点进入，而且所有节点都能看到完整的版权使用和交易过程。
3. **绝对安全可信**。区块链所记录的版权追溯全过程，是不可逆且不可篡改的。版权资产有了全过程追溯的确权保障。

一、我是怎么开始探究版权保护的
二、数字出版领域的主要困境
三、当前版权保护技术的方法和局限
四、区块链在版权保护上的主要特点
五、区块链在版权保护上的基本实现
六、智能合约在版权保护中的初步探索

亿书基本架构

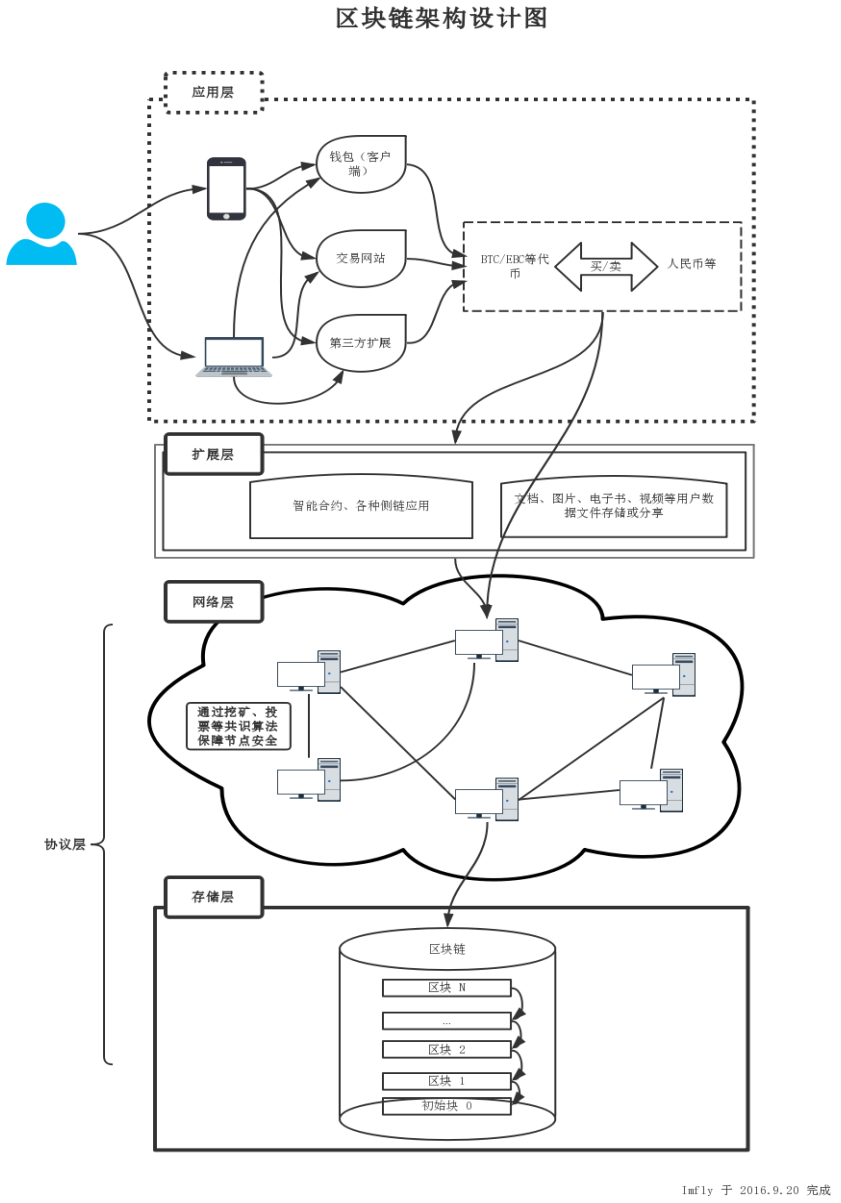
从下往上：

1、存储层

2、网络层

3、扩展层

4、应用层

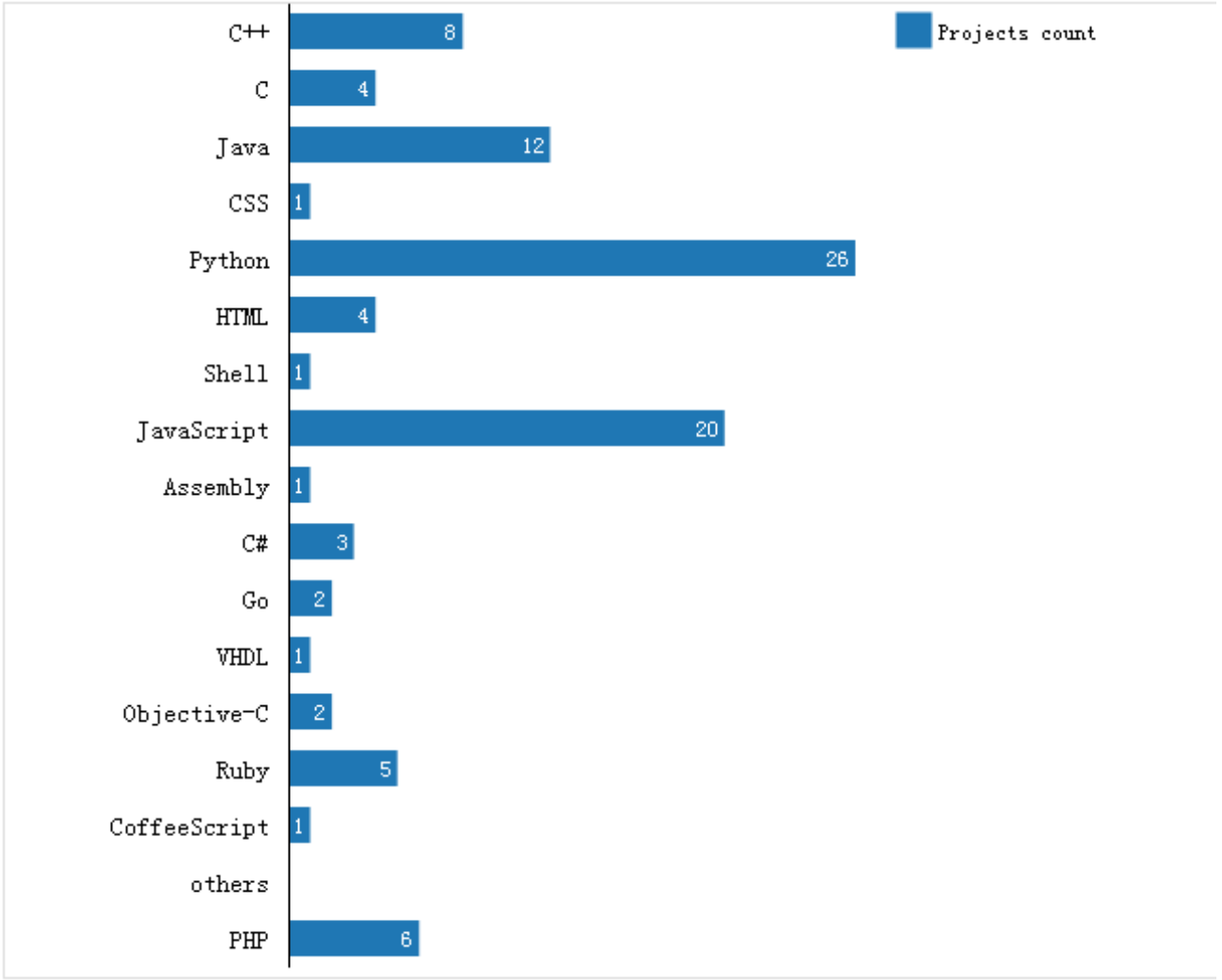


为什么选择 Node.js?

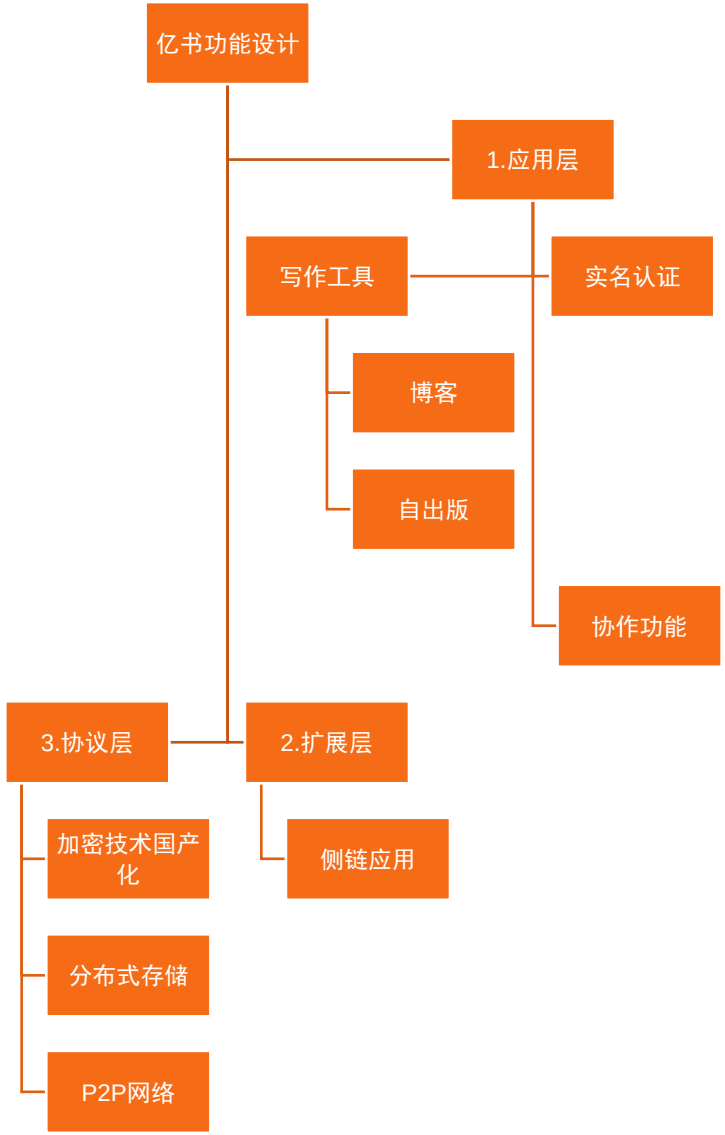
1、个人喜好

2、比较流行

3、网络编程简单



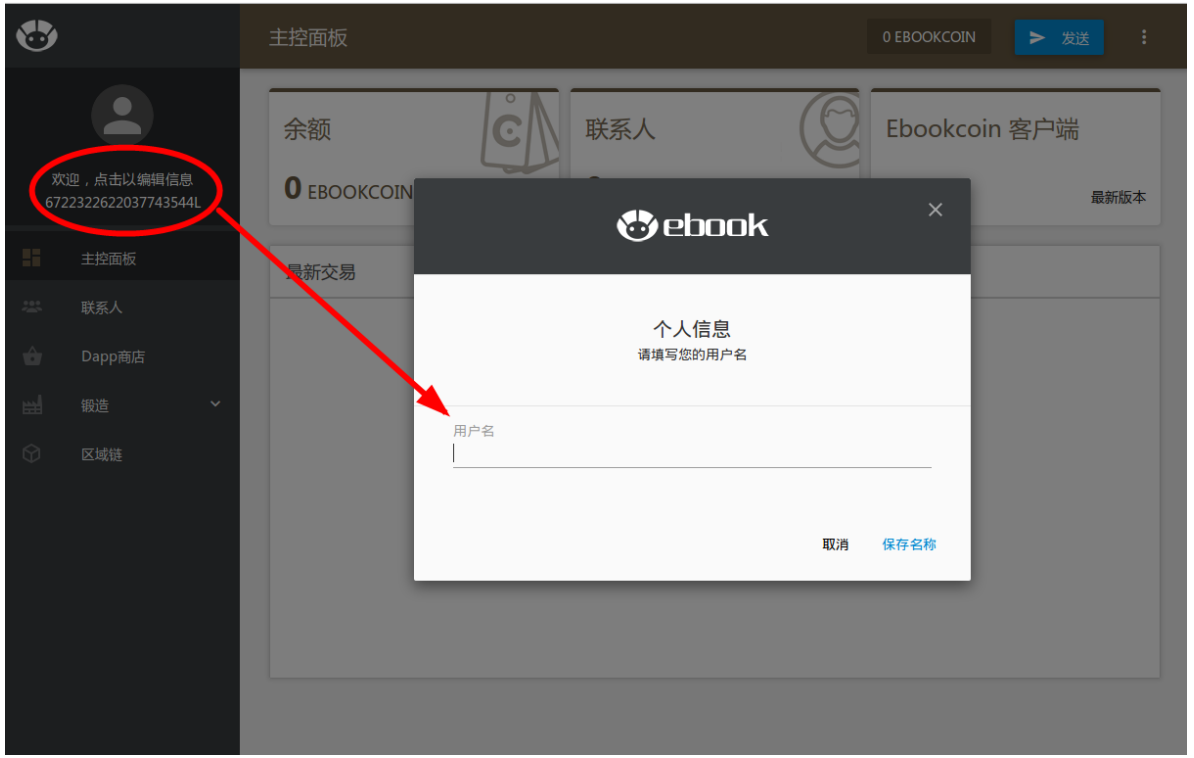
《2016年github官方报告》

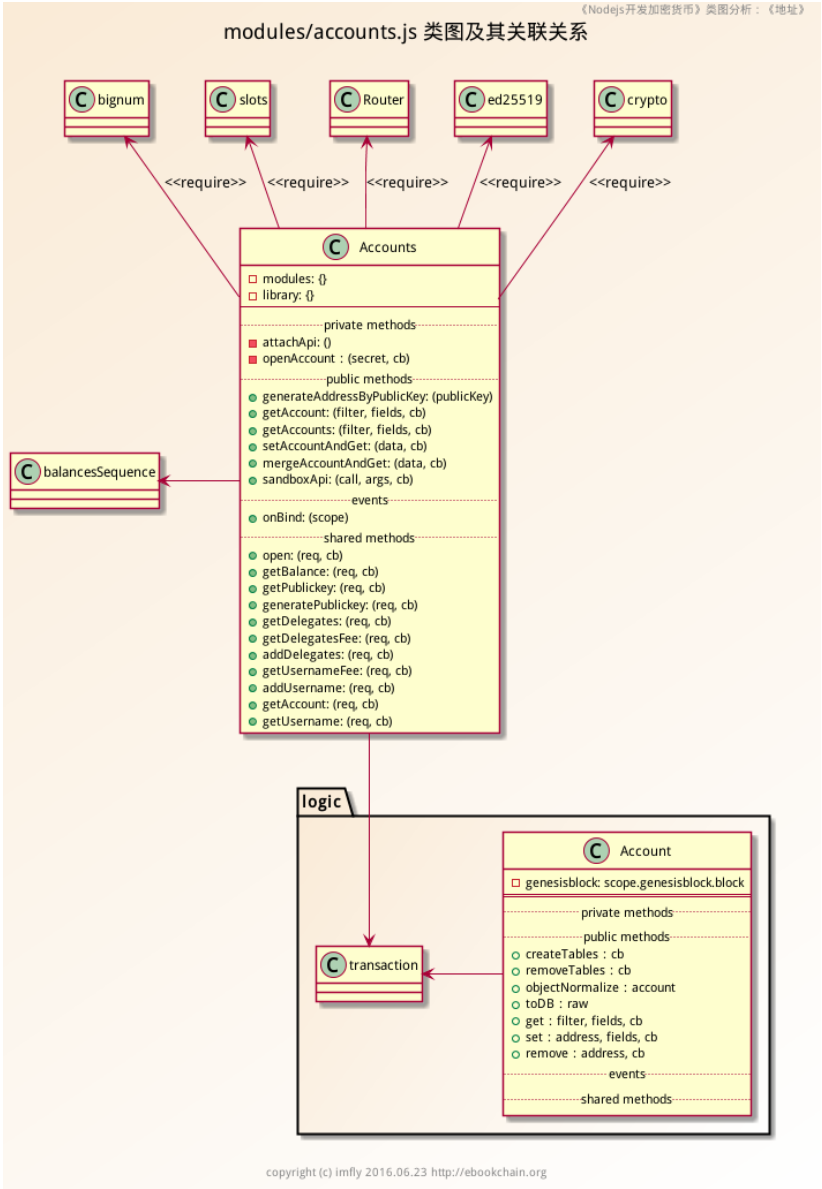

亿书功能设计
1.应用层
写作工具
实名认证
博客
自出版
协作功能
3.协议层
2.扩展层
加密技术国产化
侧链应用
分布式存储
P2P网络

实名认证

亿书允许用户注册一个用户名，它相当于是用户帐户的一个别名，其它用户可以直接向该用户的用户名付款（类似于人们常用的支付宝帐号）

亿书鼓励用户提供真实姓名等信息，进行实名认证，这非常有利于版权认证和保护。对于不提供真实信息的存储、交易和验证，将会收取相对较高的交易费用。




modules/accounts.js 类图及其关联关系
bignum
slots
Router
ed25519
crypto
<<require>>
<<require>>
<<require>>
<<require>>
<<require>>
Accounts
modules: {}
library: {}
private methods
attachApi: ()
openAccount : (secret, cb)
public methods
generateAddressByPublicKey: (publicKey)
getAccount: (filter, fields, cb)
getAccounts: (filter, fields, cb)
setAccountAndGet: (data, cb)
mergeAccountAndGet: (data, cb)
sandboxApi: (call, args, cb)
events
onBind: (scope)
shared methods
open: (req, cb)
getBalance: (req, cb)
getPublickey: (req, cb)
generatePublickey: (req, cb)
getDelegates: (req, cb)
getDelegatesFee: (req, cb)
addDelegates: (req, cb)
getUsernameFee: (req, cb)
addUsername: (req, cb)
getAccount: (req, cb)
getUsername: (req, cb)
balancesSequence
logic
Account
genesisblock: scope.genesisblock.block
private methods
public methods
createTables : cb
removeTables : cb
objectNormalize : account
toDB : raw
get : filter, fields, cb
set : address, fields, cb
remove : address, cb
events
shared methods
transaction
copyright (c) imfly 2016.06.23 http://ebookchain.org

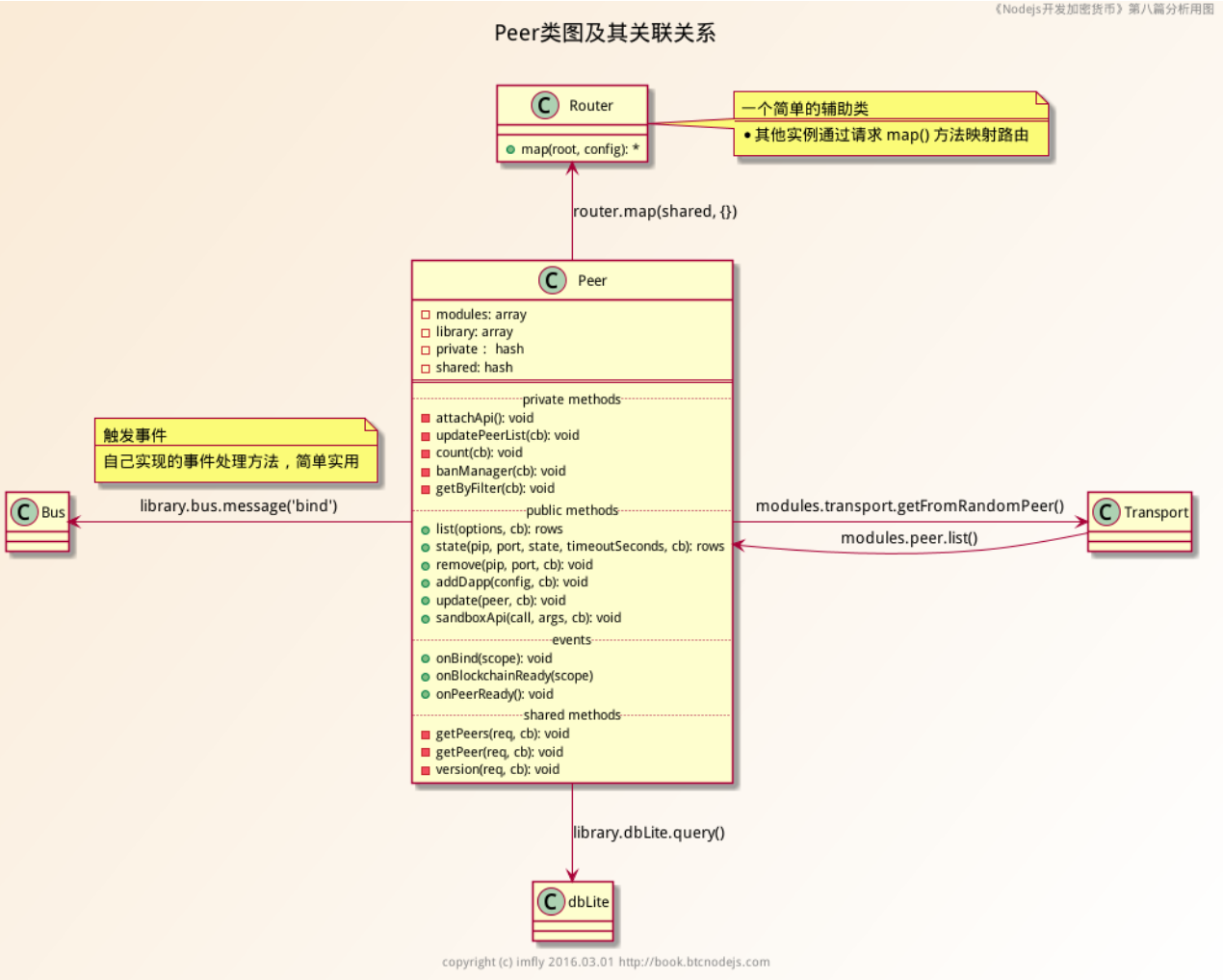
写作工具

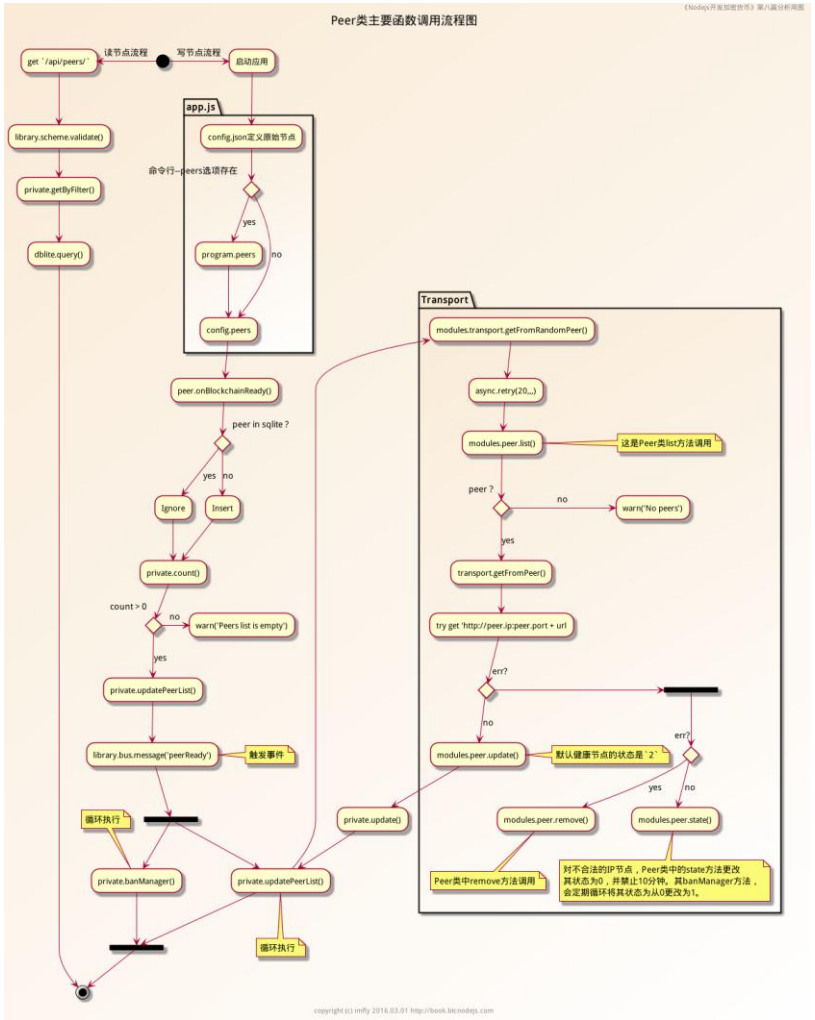
极简的写作工具，一切都是为了内容创作而生。。。

配合去中心化的网络，可以实现即时检索、实时分享

简化的是表象
复杂的是内核

P2P网络


《Nodejs开发加密货币》第八篇分析用图
Peer类图及其关联关系
Router
map(root, config): *
一个简单的辅助类
• 其他实例通过请求 map() 方法映射路由
router.map(shared, {})
Peer
modules: array
library: array
private : hash
shared: hash
private methods
attachApi(): void
updatePeerList(cb): void
count(cb): void
banManager(cb): void
getByFilter(cb): void
public methods
list(options, cb): rows
state(pip, port, state, timeoutSeconds, cb): rows
remove(pip, port, cb): void
addDapp(config, cb): void
update(peer, cb): void
sandboxApi(call, args, cb): void
events
onBind(scope): void
onBlockchainReady(scope)
onPeerReady(): void
shared methods
getPeers(req, cb): void
getPeer(req, cb): void
version(req, cb): void
触发事件
自己实现的事件处理方法，简单实用
Bus
library.bus.message('bind')
modules.transport.getFromRandomPeer()
modules.peer.list()
Transport
library.dbLite.query()
dbLite
copyright (c) imfly 2016.03.01 http://book.btcnodejs.com


Peer类主要函数调用流程图
get '/api/peers/'
读节点流程
写节点流程
启动应用
app.js
library.scheme.validate()
config.json定义原始节点
命令行--peers选项存在
private.getByFilter()
yes
dblite.query()
program.peers
no
config.peers
peer.onBlockchainReady()
peer in sqlite ?
yes
no
Ignore
Insert
private.count()
count > 0
no
warn('Peers list is empty')
yes
private.updatePeerList()
library.bus.message('peerReady')
触发事件
循环执行
private.banManager()
private.updatePeerList()
循环执行
Transport
modules.transport.getFromRandomPeer()
async.retry(20,...)
modules.peer.list()
这是Peer类list方法调用
peer ?
no
warn('No peers')
yes
transport.getFromPeer()
try get 'http://peer.ip:peer.port + url
err?
no
err?
modules.peer.update()
默认健康节点的状态是 `2`
yes
no
private.update()
modules.peer.remove()
modules.peer.state()
Peer类中remove方法调用
对不合法的IP节点，Peer类中的state方法更改其状态为0，并禁止10分钟。其banManager方法，会定期循环将其状态为从0更改为1。

加密与签名

对data进行加密（SHA256 杂凑密码算法）

```
var hash =
crypto.createHash('sha256').update(data).digest()
```

使用ED25519对上述密文进行公钥签名

```
var keypair = ed.MakeKeypair(hash);
```

“可信时间戳”——由权威机构签发的，能证明数据电文在一个时间点是已经存在的、完整的、可验证的，具备法律效力的电子凭证

签名验证

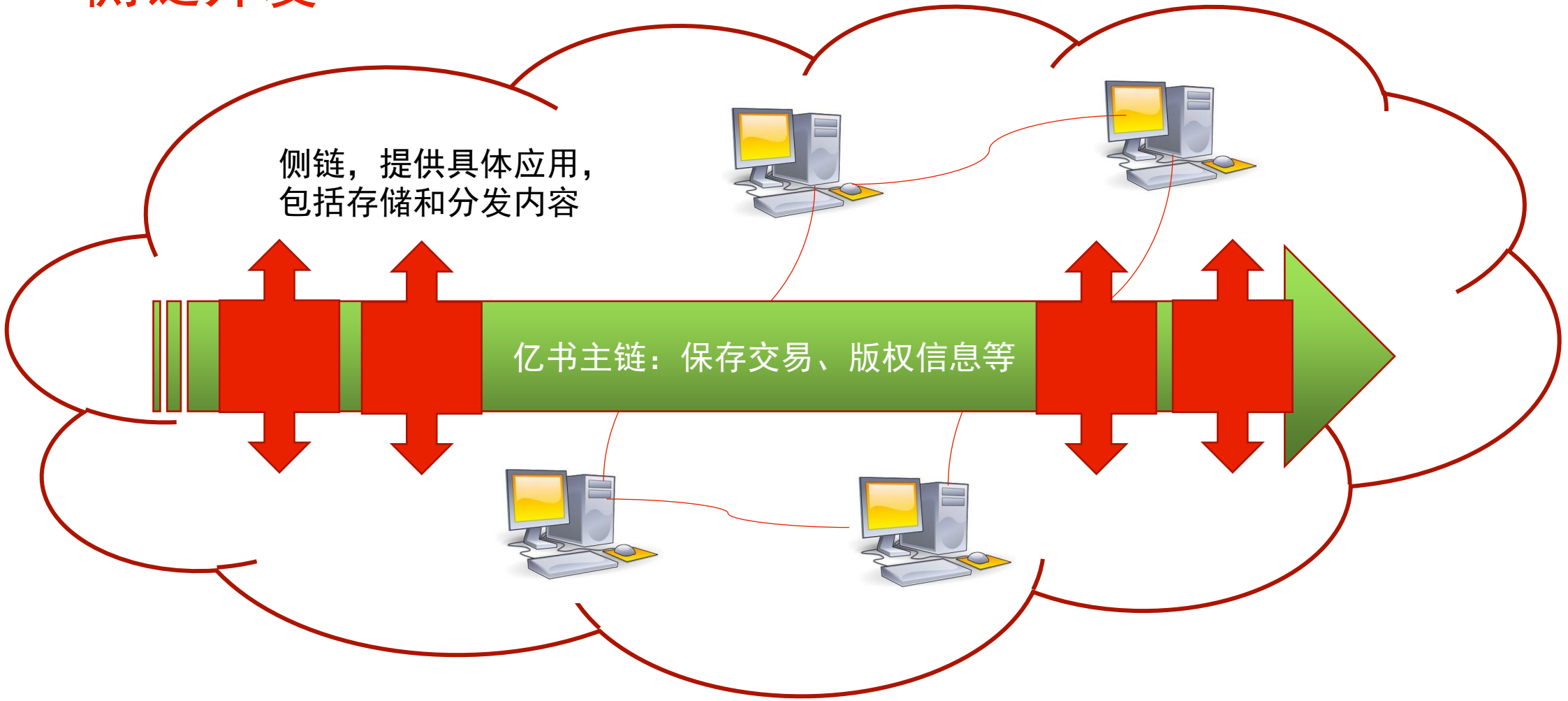
ED25519验证速度非常高，操作也非常简单

```
var res = ed.Verify(hash, signatureBuffer || ' ', publicKeyBuffer || ' ');
```

再次使用杂凑密码算法SHA256算法计算作品的数字指纹，如果得到的数字指纹和版权持有者公布的数字指纹一样，便可确认版权持有者的版权。到此，便解决了传统的注册、确权和验证问题。

实际上，完整不变的盗版是有的，而更多的是局部内容的盗版，我们还要继续想办法…

侧链开发



一、我是怎么开始探究版权保护的
二、数字出版领域的主要困境
三、当前版权保护技术的方法和局限
四、区块链在版权保护上的主要特点
五、区块链在版权保护上的基本实现
六、智能合约在版权保护中的初步探索

区块链，可编程的“利益”转移手段

利益依附欲望而生，而人的基因确定了欲望的存在，组成社会的基本元素是人，就不可避免地出现了：阶级、政治、 战争……利益冲突决定着一切。

- 利益，魔鬼与天使的共同目标
- 利益，主宰着人类行为

马克思说过：“人们奋斗所争取的一切，都同他们的利益有关”

智能合约

就是“合约智能化”，主要特点是：

- 合同条款不可篡改，有效性得到保障；
- 合同制定和执行的全过程透明公开，便于监督；
- 合同执行过程可编程，能够自动执行，不受干预。

社区管理共识机制

比特币的创新不仅仅体现在技术上的创新，更体现在奖励规则对社区管理的创新，所以才有了人们提出的DAC（分布式自治机构）的概念。比特币等区块链产品本身就是这样的DAC。

谢谢大家！

区块链，是互联网的未来和未来的互联网，你不能错过！