GopherChina2018

如何用GO开发一个区块链项目： ABitchain

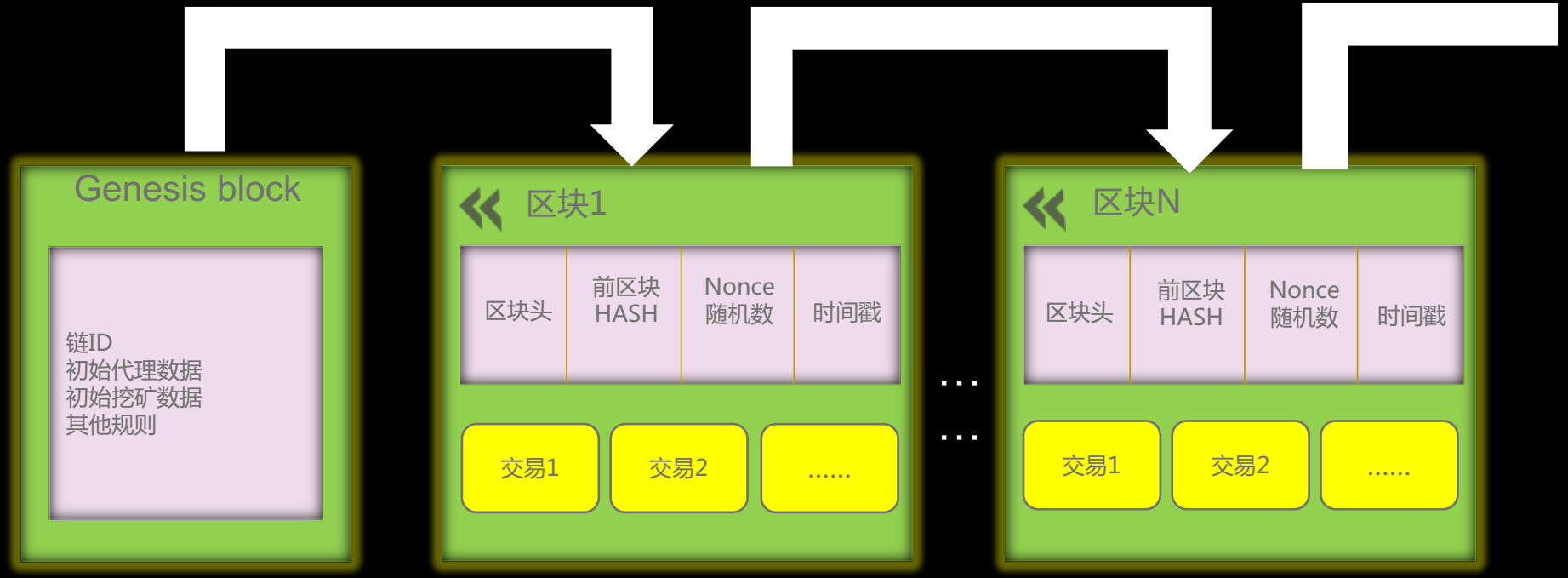
ABitchain

目录

区块链特征

1. 去中心化
2. 可信任的机器，防篡改分布式数据库
3. 通过密码学构建账户体系
4. 共识，P2P通信是交易基础

区块链数据形态


Genesis block
链ID
初始代理数据
初始挖矿数据
其他规则
区块1
区块头
前区块HASH
Nonce随机数
时间戳
交易1
交易2
.....
区块N
区块头
前区块HASH
Nonce随机数
时间戳
交易1
交易2
.....

区块链发开模块

A 共识模块

B 账号地址生成算法

C P2P是怎么实现通信的

D 智能合约

E 智能Pending区规则简单介绍

F 区块数据底层存储

常见的共识算法

1. POW（Proof of Work）
2. POS（Proof of Stake）
3. DPOS（Delegated Proof-Of-Stake）
4. PBFT（Practical Byzantine Fault Tolerance）

共识机制 ：DPOS + BFT

定时任务管理

洗牌算法

代理池维护
和投票机制

定时任务管理

01

NTP（Network Time Protocol，网络时间协议）

02

基于NTP时间定时任务：

- 定时洗牌
- 代理出块
- 节点数据同步

洗牌算法介绍

通过洗牌方式，得到一个固定的代理出块顺序

01
功能

02
设计目的

最大程度保证对等节点上，洗牌结果一致

代理池和投票机制

1、用户通过注册的方式成为代理候选人

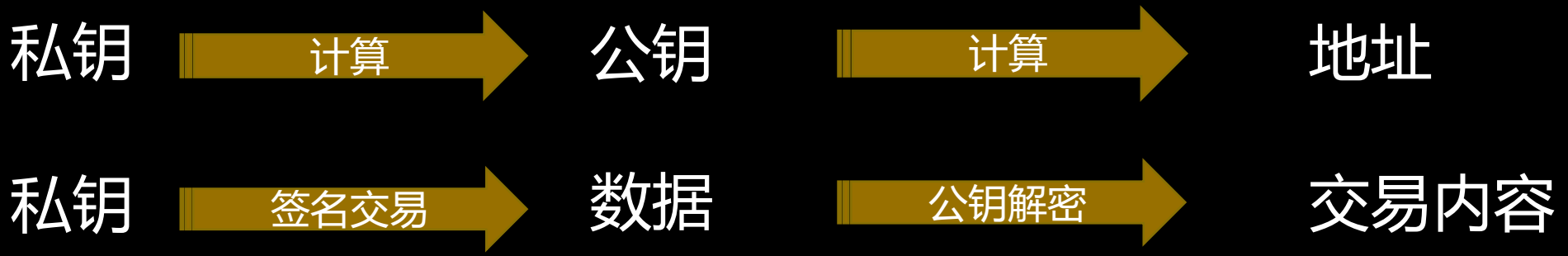
2、用户可以通过投票的方式，支持候选人

3、实时选出获得投票最多的代理候选人

地址设计目的：

1、证明数字资产所有权问题

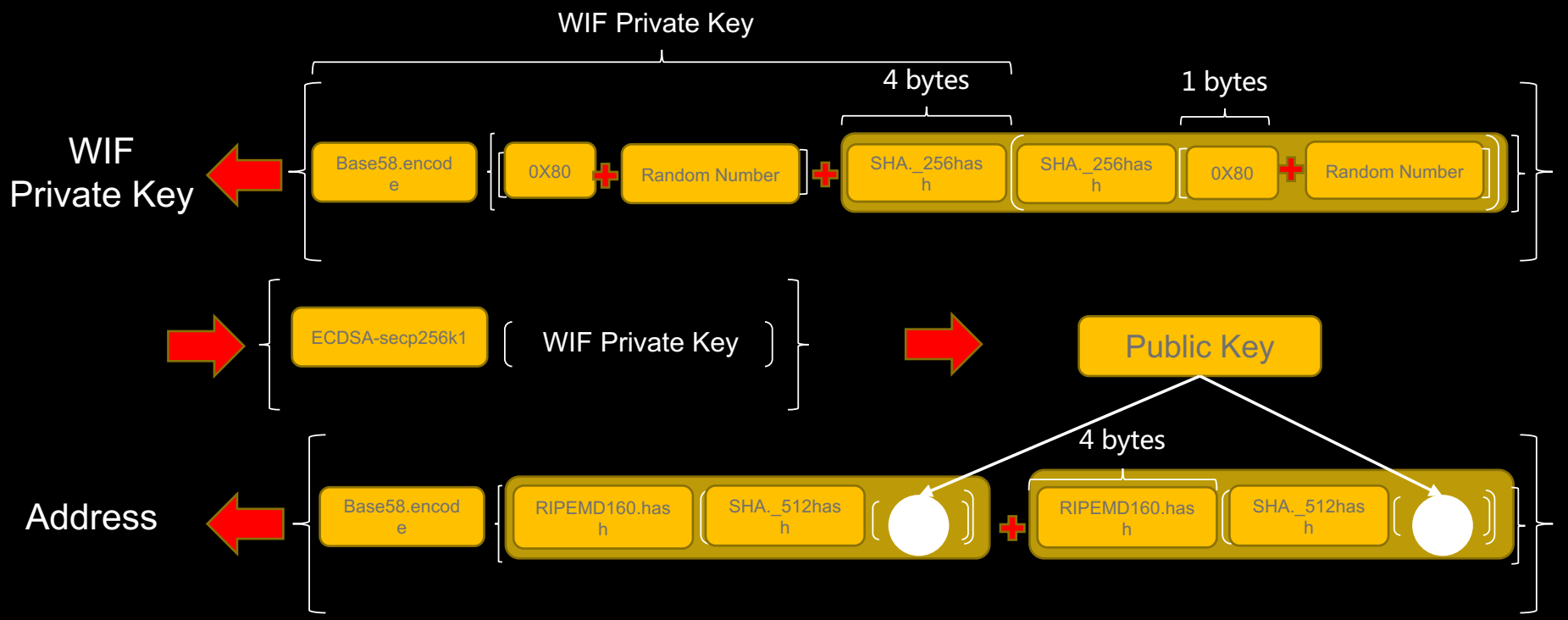
2、验证交易合法性



ABitchain

B. 帐号地址生成算法

下图是以Achain地址生成为例详细说明生成过程：



ABitchain

C. P2P实现通信

Kad路由表：

基于Kademlia（简称Kad）一种分布式哈希表技术，构建了P2P网络拓扑结构。

a 启动时候生成节点编号。

b 例如：距离0000000011（3）映射到K2桶中。

C 路由表中会预置公共节点

D 循环更新到本地路由表

I	距离	邻居
0	$[2^0,2^1)$	(IP address,UDP port,Node ID)$_{0\text{-}1}$ ..... (IP address,UDP port,Node ID)$_{0\text{-}k}$
1	$[2^1,2^2)$	(IP address,UDP port,Node ID)$_{1\text{-}1}$ ..... (IP address,UDP port,Node ID)$_{1\text{-}k}$
2	$[2^2,2^3)$	(IP address,UDP port,Node ID)$_{2\text{-}1}$ ..... (IP address,UDP port,Node ID)$_{2\text{-}k}$
.....	.....	.....
i	$[2^i,2^{i+1})$	(IP address,UDP port,Node ID)$_{i\text{-}1}$ ..... (IP address,UDP port,Node ID)$_{i\text{-}k}$

ABitchain

D. 智能合约

1 什么是智能合约

2 智能合约案例

3 智能合约所能达到效果

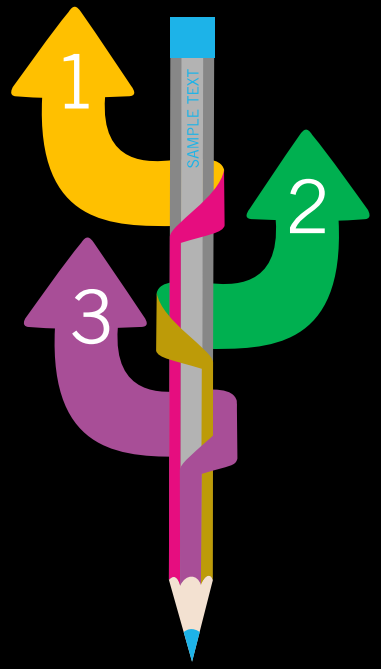
4 智能合约执行的所需的费用

项目	子项	花费 （gas）
固定花费	创建合约	53000
	其他交易	21000
	交易数据非零字节，每字节	68
	交易数据零字节，每字节	4
预编译合约	EcrecoverGas	3000
	sha256hash基本	60
	Sha256hash每字节	12
	Ripemd160基本	600
	Ripemd160每字节	120
	dataCopy基本	15
	dataCopy每字节	3

ABitchain

E. 智能Pending

什么是Pending区



Pending区作用

Pending区规则设计建议

区块链存储数据结构需要满足：

1、快速验证交易

2、有效防篡改

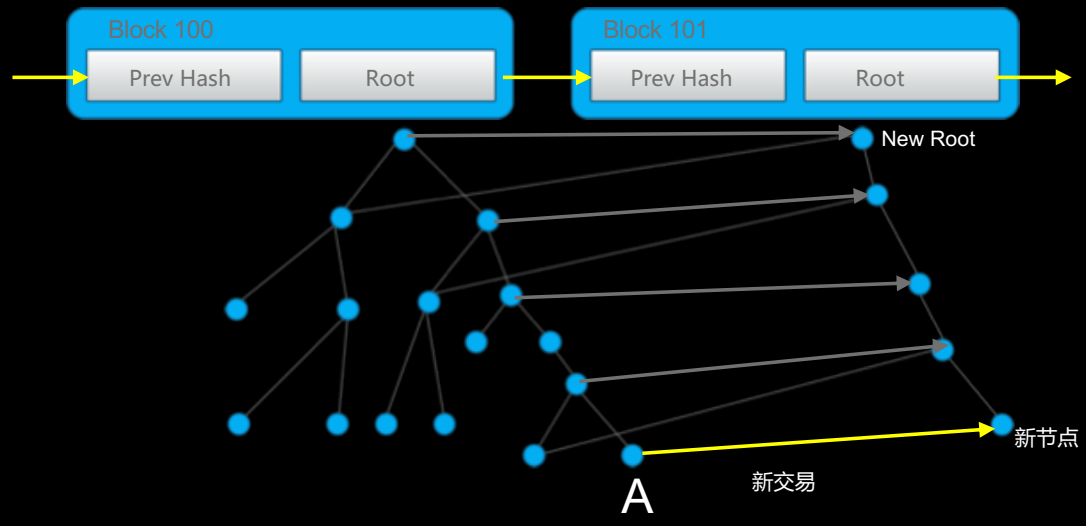
3、快速检索数据

4、分叉后能快速回滚

F. 底层存储

比特币：Merkle
以太坊：Merkle Patricia Tries (MPT)

一笔交易如何存入区块链中



区块链项目使用的编程语言需要满足

1、执行效率高

2、高并发

3、跨平台

4、高效的网络处理能力

所以早期的区块链项目是以 c++ 为主

当区块链遇到Go

1、编译速度快，部署简单，跨平台

2、高性能，语言底层支持高并发

3、和C的良好的交互性

4、良好的语言设计，更重要的是 自带完善的工具链，大大提高了团队协作的一致性

5、自动垃圾回收，省去了不少麻烦

6、特殊的channel机制，解决系统内部频繁的通信

Go的最佳应用体现——ABitchain主链即将上线！

1、为应用而生的区块链，区块链3.0的定义者

2、打造最快的区块链，再多的加密猫加密狗都不再堵

3、现在就可以用的区块链

4、“三帮一扶”加快区块链应用落地

ABitchain

加入我们，请联系：

- 张博
- 邮箱: bo.zhang@abitchain.io
- Telegram: +86 186 1117 8194
- 手机: +86 186 1117 8194