

微三云麦超深入浅出讲解数字积分系统：第二节

产品名称	微三云麦超深入浅出讲解数字积分系统：第二节
公司名称	深圳市微二云信息技术有限公司
价格	.00/个
规格参数	
公司地址	广东省东莞市东莞生态园瑞和路1号松山湖高新技术创新园B栋2-5层
联系电话	13929266321 13929266321

产品详情

微三云麦超深入浅出讲解数字积分系统：第二节

引言

前面第一节主要介绍了区/块/链的概念，我们知道区/块/链分为两大类，一是以公有链为代表的无权限控制区/块/链，第二是有权限控制的区/块/链，这个又包括了私有链（Private Blockchain，以Overstock为代表）和联盟链（Consortium Blockchain，以R3为代表），相对于公有链来说，这些链一般都是没有电子货币，因为他们不需要像公有链那样要靠电子货币作为挖/矿的奖励来激励参与，所以速度也是比较快的。

上一章都是讲的抽象的概括，下面我们就深入讲一些具体的东西，这样以便于大家有一个形象的概念，方便理解。我们这一章主要讲讲公有链，以方便讲解以及大家去继续研究，尝试，这里选择在公有链领域社区最为活跃以太坊（Ethereum），对于中国用户来说，其于2016年9月19号刚刚在中国上海举办了DevCon 2区/块/链峰会，很多人可能有所印象。第一步，这个东西怎么读啊？其实这是新构造的一个单词，而非一个已有的英语单词，其读作 [i ' i : ' r i m]。接下来我们会一起过一下涉及的一些概念，后面我会介绍几个如何进行太坊开发的技术工具，以及两个比较好用的应用框架。

大家都知道，学习一个新技术最好的方式就是亲自动手试一把，几乎学习所有新编程语言上来都会写个

HelloWorld并运行一把，在这一章最后一部分我会手把手的带领大家创建并运行一个智能合约。

概念

Ethereum:

Ethereum ([官方链接](#))，是个区/块/链公有链解决方案，如果比/特/币的区/块/链称作区/块/链1.0的话，那Ethereum可以称为区/块/链2.0。其主要特色就是支持可编程的智能合约。这个开源的系统相当于计算机中的操作系统一样，其是一个平台，提供了API及接口，以供其上运行不同的程序共享使用。同时因为它本质上是去中心化的区/块/链，因此号称是零宕机，零审查，以及不会有欺诈与人为篡改。就像所有的公有链需要激励机制的“代币”一样，它除了底层的区/块/链外，还有自己的加密电子货币，Ether，即以太币，在国内有些人戏称为“姨太”。目前一个“姨太”大约11美元，实时的价格趋势可以参见[这个交易所链接](#)

智能合约

智能合约 ([解释链接](#))，其实这个概念本身是远远早于区/块/链产生的（早在1994年就出现了）。智能合约，说白了就是自己写的一段代码放到区/块/链上，在这里可以添加自己需要的业务逻辑等，只是这段代码在创建后不像传统应用是部署到服务器上运行，而是放到区/块/链上，并且自动执行（其运行部署都会消耗Gas（气，也就是若干的以太币））。各个参与方不需要像以前需要一个或若干个中心节点/服务器，大家都各自在自己那里完全按照“合约”执行，中间没有人可以去篡改或者停止，此设计会大大提高flexibility（灵活性）以及互相不信任的问题。比如有一个智能合约定义的逻辑是：当A收到钱后，B就会收到货物，这些操作都是按照合约自动执行，中间不再会有违约或者被人为修改的风险。

这些智能合约是以DAPP (Decentralized Application) 的形式存在。智能合约是部署在区/块/链上，由于区/块/链的透明性，这些合约对任何人都是可见的，当然这个有利有弊。如果其有bug或者漏洞，就有可能被人抓住并利用，比如2016年6月的The DAO攻击，就造成相当于5千万美元的以太币丢失，这也直接导致了以太坊后面的一次硬分叉，这块笔者后面会撰文详解。

Web 3.0

大家可能听说过web 1.0，其是指之前传统的网页技术，比如HTML，传统的JavaScript，VBScript，CSS。而web 2.0是使用所谓的DHMTL，HTML5，Ajax，等众多的JavaScript技术，来创作类似于桌面程序效果般的网页应用。Web 2.0这些技术有个问题，就是过于依赖中心化的服务器/第三方机构，比如除了其应该做的提供网页访问服务外，还有验证，用户行为记录分析等。而这里提出的web 3.0是有如下几个特性，首先是去中心化，比如通用的后台端，使用Swarm与bzz来作为内容寻址的存储系统，基于区/块/链的共识形成机制，基于Whisper的异步消息机制等，这样具体的业务逻辑都会分发到每个客户端去执行，而非位于昂贵且易于出问题的少数中心节点。

这是刚刚提到的架构图

Solidity

上面说到的这些智能合约一般来说是使用一种特殊的编程语言来创建的，即Solidity，这个语言是以太坊提出并创造的，面向对象的DSL特定领域编程语言（Domain Specific Language），它是以太坊支持的4种语言（另外三个是Serpent，LLL和Mutan），只不过其是最流行的一个语言。从技术上来讲，solidify源代码会编译成字节码，然后运行于EVM（Ethereum Virtual Machine）上面。如果你看到源代码后就会觉得其实Solidity是与JavaScript十分类似的语言，如下是一段代码：

Gavin Wood（Solidity之父）说Solidity就是根据ECMAScript（是JavaScript，ActionScript等的标准祖先）所创建的，这样对于大多数开发人员来说学习曲线会很平滑。

开发工具

由于发展时间不是很长，目前市面上可用的开发环境IDE不太多。下面介绍下稍微比较成熟可靠的开发工具。

Microsoft Visual Studio Ethereum 插件

没错，就是那个市面上已经非常常见的visual studio，也就是dotnet的开发工具，不是一个全新的开发工具。此开发集成环境只需要安装solidify插件即可。这个也从侧面可以看到微软对于以太坊以及区/块/链的野心。

安装此插件后在微软的Visual Studio后就可以在新建项目时的模板里看到这个Solidity选项：

当选择此模板后，visual studio他会自动构造出一个应用的基本文件结构。这样你可以省去一些每次开发一个智能合约都要重复的工作。你就可以集中时间精力到真正业务代码上。

如下就是这个IDE自动生成的代码

Ethereum Studio

除了背靠微软这个大旗的visual studio集成开发环境外，还有一个方便大家使用的免费的IDE。这个是基于Cloud9平台的一个在线IDE，其完全运行于浏览器中，不用安装，可以用于任何的操作系统。如下就是这个在线集成开发环境的样子。这个还是比较推荐的开发环境：

智能合约应用开发框架

目前比较常用的智能合约构架有如下几个，都是开源并且免费的。这里我们来手把手的创建并运行一个智能合约，来体会一下。

Embark

首先推荐的是这个叫做Embark的框架，他是一个让你可以轻松开发部署Dapps的平台，它支持的功能包括，在JS代码中部署智能合约，智能合约的热部署，可以集成grunt等构造工具。支持TDD（即测试驱动的开发）比如支持mocha等测试框架，可以方便的使用IPFS等去中心化的系统，支持增量，智能的部署修改过的智能合约等。这个工具是使用nodejs写的，因此你需要先安装nodejs的环境。这个平台会在你本地启动一个区/块/链服务模拟器，这样你就可以完全在本地开

发测试，大大

提高了工作效率。如下是启动后的截图。Embark的安装及源代码位于[Github这里](#)

首先你需要来安装Embark以及区/块/链模拟器。

```
# 安装Embark npm -g install embark - framework #  
安装区/块/链模拟器 npm -g install ethersim # 启动RPC模拟器 embark  
simulator
```

启动的模拟器是下面这个样子

然后我们去创建一个新的智能合约：

```
# 创建一个叫做demo的智能合约基础结构 embark demo #  
进入这个目录，下面含有配置文件 embark.yaml cd embark__demo # 启动应用  
embark run
```

启动后，首先在后台看，Embark帮忙使用coffeescript等构造并部署了合约。

你可以使用浏览器试验一下，比如打开<http://localhost:8000>，然后你可以试着输入个数值，去试试看看它是不是已经能够响应处理你的输入了：

是不是很神奇，短短两三分钟，已经从零开始构造出一个可以运行的以太坊DAPP，并运行于区/块/链之上。接下来我们介绍另外一个框架选择方案。

Truffle

Truffle，是跟前面提到的EthereumStudio同一个公司(Consensys)开发的一个框架，这个跟前面的embark类似，也是可以提供智能合约的开发测试平台，他的一个特色就是它可以集成nodejs里面强大的测试功能，比如Mocha, Chai等等。像Embark一样，你需要另外安装运行其他软件，来启动以太坊客户端模拟器，最常用的是EthereumJS TestRPC [Githublink](#)，它会在内存中启动一个Ethereum的客户端，这样可以快速测试你开发的应用。

因为这个也是使用nodejs创建的应用，因此使用如下命令来安装此程序，安装好了启动此应用

```
#安装以太坊模拟器 npm install -g ethereumjs-testrpc #
启动模拟器 testrpc
```

启动后是这样子的

模拟器启动好了，接下来执行下面的命令来初始化 truffle 应用。

```
mkdir firstApp cd firstApp truffle init
```

上面最后一个命令就会自动帮你构造好的程序框架，包括一些最基本的 JavaScript 文件，几个智能合约源代码，主应用程序的 HTML 代码及配套的 CSS 等文件。如下是这个基本框架：

接下来你可以添加自己的代码到 contracts 目录下的智能合约文件，也可以什么都不动，因为 truffle 已经自动生成了最基本的框架。

```
#这个命令会把智能合约源代码编译成字节码 truffle compile
```

编译好的代码需要部署到区/块/链上才可以执行，在 truffle 中这个工作是由 migrates 目录下定义的 migrate 作业执行的，我们去修改文件 2__deploy__contracts.js 为如下：

```
module.exports = function(deployer) { //
  deployer.deploy(ConvertLib); //
  deployer.autolink(); //
  deployer.deploy(MetaCoin);
  deployer.deploy>HelloEthereum); };
```

然后执行如下命令去执行代码部署，它除了把你的智能合约发布到区/块/链之外，还会做一些相关工作，比如 link 用到的 library 等。deployer 可以使用 promise 的方式（e.g. .then(function(xx)）来执行其他额外的工作等，比如创建一个其他的合约并调用，等。这个便于你来灵活的扩展应用。

```
truffle migrate truffle build
```

结语

好了，至此我们已经了解了什么是以太坊已经其上运行的智能合约，DAPP 等概念。后面又介绍了开

发智能合约的工具已经可复用的框架，最后又手把手亲自做了一个智能合约。这样大家应该对区/块/链以及以太坊等公有链有了一个形象具体的感觉了吧。如果这里有什么问题或者建议，欢迎通过下面的联系方式与我沟通。

附：微三云区/块/链积分系统——基于腾讯区/块/链底层开发的数字积分共享系统

微三云区/块/链积分系统开放资源融入、积分采集、权益卡、资讯、自定义5大板块发放企业权益，灵活适配各种业务场景需求，帮助解决企业运营过程中遇到的资源融入难、留存转化难、产品销售难、资源整合难等痛点。

还能开放接口对接企业现有或未来拓展的应用系统，如对接微三云的单品商城系统，打造一款爆品快速切入市场，数字积分以产品利润托底，消费送数字积分，推广送数字积分，按照梯度送数字积分等等，结合单品商城系统的差价、团队月度计酬、淘汰机制等8种时下最热门的代理制度，帮助企业打造扎实紧密的线上线下团队，以数字积分调动积极性，拉动商城消费，进行全国布局，另外微三云区/块/链积分系统还可对接综合商城、内容付费等领域，已扶持多家企业上链，区/块/链商业落地快人一步，就用微三云区/块/链积分系统，更多详情欢迎咨询微三云麦超总监了解，关注z i h o n g 3 3 3公众号了解更多区/块/链、互联网营销技巧。

真正的区/块/链应具备合规、创新链接和资源配置三大能力。

0 1 . 具备合规

具备研究、发现和适应合规的能力是项目长期发展的必要前提。

0 2 . 创新链接

创新链接能力指除了在联盟链内互通互联，还应该能进行跨界合作，这将是判断区/块/链应用好坏的重要标准。例如微三云区/块/链积分系统，可链接电商、内容付费、知识付费等领域进行跨界合作，目前已成功对接单品商城、综合商城、微云视系统等，创新卖货模式，消费（卖货、卖课程等）送积分，积分账户互通互联，积分可持有、交易、变现，也可兑换课程/课时或商城购物，平台互补消费，提升转化率。

0 3 . 资源配置

而资源配置能力，则是因为区/块/链行业乱象期已过，缺乏合理商业逻辑，缺乏实际资产支撑的项目不再有发展空间，只有具备配置真实世界资源能力的项目，才能在竞争中存续。例如客户购买微三云区/块/链积分系统，将发布一份注明以实际资产托底的白皮书，将企业“生产剩余价值分配”升级为“全民资本红利分配”的商业模式。

另外还需要注意的是，“区/块/链应用”运用哪个底层技术，因为研发一个底层技术需要消耗大量的资金成本和人力投入，如果没有庞大的技术团队及资金是无法完成的。像深圳市税务局推出的区/块/链电子发票，其区/块/链底层也非其自主研发，是由腾讯公司为其提供底层技术支撑做落地，而微三云区/块/链积分系统也是与区/块/链电子发票使用同款底层技术——腾讯区/块/链作为底层技术支撑，大品牌有保障。

