****区块链开发（二）部署并运行第一个以太坊智能合约****

   网络上不少部署智能合约的文章，但是都有一个共同的特点，就是采用命令行的方式来部署，先是建立SOLC的编译环境，然后部署Geth或者Eth节点，然后一步一步生成钱包、ABI、合约地址进行部署，对初学者来说晦涩难懂而且容易失败，本文主要介绍如何在图形化界面下一键部署和调用智能合约。

****一、智能合约和DAPP概述****

****1、智能合约基本概念****

  智能合约是一段代码和数据的集合，可以部署以太坊网络上运行。如果做比喻的话智能合约更像是JAVA程序，JAVA程序通过JAVA虚拟机（JVM）将代码解释字节进行执行，以太坊的智能合约通过以太坊虚拟机（EVM）解释成字节码进行执行，如果你学过汇编，会发现编译后的字节码和汇编很类似。同时智能合约有自己的账户，在时间或事件的驱动下能自动执行一些功能，如可以在相互之间传递信息，修改区块链的状态比如账户信息等。以太坊的智能合约最大的特点是图灵完备，通俗来说可以完全模拟一台计算机所能做的所有事情，大家熟知的比特币其实也可以执行一些简单脚本，但是他就不是图灵完备，比如循环指令比特币就无法执行。

****以太坊虚拟机（EVM）****

        以太坊虚拟机（EVM）是以太坊中智能合约的运行环境。它不仅被沙箱封装起来，事实上它被完全隔离运行，也就是说运行在EVM内部的代码不能接触到网络、文件系统或者其它进程，甚至智能合约之间也只有有限的调用。

****2、  DAPP基本概念****

        初学者经常把智能合约和DAPP搞混，以太坊社区把基于智能合约的应用称为去中心化的应用程序(Decentralized App)。DApp的目标是让你的智能合约有一个友好的界面，外加一些额外的有利于用户使用的东西。典型的DApp例子由一个html界面，web3运行库，一段JS代码以及部署在区块链上的一段智能合约组成。与一般CS架构的网站不同，DApp不能在普通的服务器上运行，DApp必须运行在一台能与以太坊节点交互的服务器上，或者任意一个以太坊节点上。DApp通过提交交易到区块链网络与对应的智能合约进行交互，并且从区块链网络而不是中心化数据库比如（MYSQL数据库）读取重要数据。相对于典型的BS架构用户登录系统不同，以太坊用户被表示成一个十六进制的地址而且所有用户数据和其他数据均保存在本地，与目前的web应用架构有很多不同。

****3、  智能合约高级语言****

   用户不可能直接编写以太坊虚拟机（EVM）字节码，所以以太坊提供了几种编写智能合约的高级语言。

Solidity：类似JavaScript，这是以太坊推荐的旗舰语言，也是最流行的智能合约语言。具体用法参加Solidity文档，地址：https://solidity.readthedocs.io/en/latest/

Serpent：类似Python风格，文档地址：https://github.com/ethereum/wiki/wiki/Serpent

LLL：类似Lisp风格，目前已经被终止了。

可以根据不同的习惯选择不同的高级语言，目前最流行的是Solidity。

****二、在以太坊私有链上部署第一个智能合约****

  本文的智能合约采用以太坊官方的示例合约，功能就是在区块链上存储一个数字，并能够读取出来。代码如下：

contract SimpleStorage {

    uint storedData;

    function set(uint x) {

        storedData = x;

    }

    function get() constant returns (uintretVal) {

        return storedData;

    }

}

    即使没有学过Solidity语言也可以大致看出，该合约set函数存储一个数字在X变量中，get函数从X变量中读取这个数字出来，下面对这个合约进行部署：

****1、  启动私有链****

   启动以太坊私有链Geth和Ethereum-Wallet图形界面（本文使用Geth 1.41版本, Ethereum-Wallet 0.8.1版本）。如果不知道如何启动，请参考我上一篇文章[《区块链开发（一）搭建基于以太坊的私有链环境》](http://blog.csdn.net/sportshark/article/details/51855007" \t "http://www.8btc.com/_blank)，启动后界面如下, Ethereum-Wallet会显示红色的PRIVATE-NET标记。

[](http://7fvhfe.com1.z0.glb.clouddn.com/wp-content/uploads/2016/08/14.jpg)

****2、创建两个钱包****

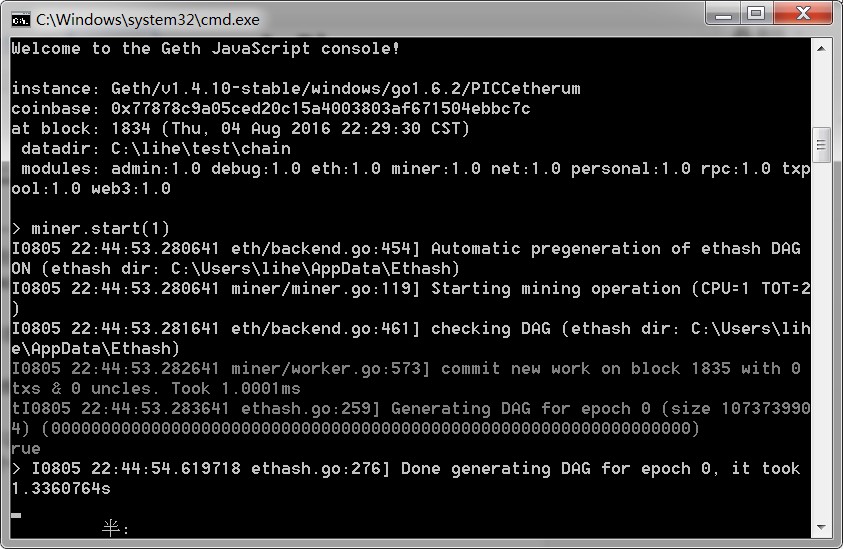
  点击”ADD ACCOUNT” 按钮，添加一个钱包，程序会弹出一个对话框，提示输入两遍密码，输入完密码后，账号即创建成功。创建其他的账号，一样的操作，从上面的截图可以看到，有三个账号，一个是MAIN ACCOUNT，一个是ACCOUNT2，一个是ACCOUNT3

****3、挖矿获取一些以太币****

  账号创建后，还没有以太币，需要在私有链上挖矿，切换到Geth界面，输入

miner.start(1)

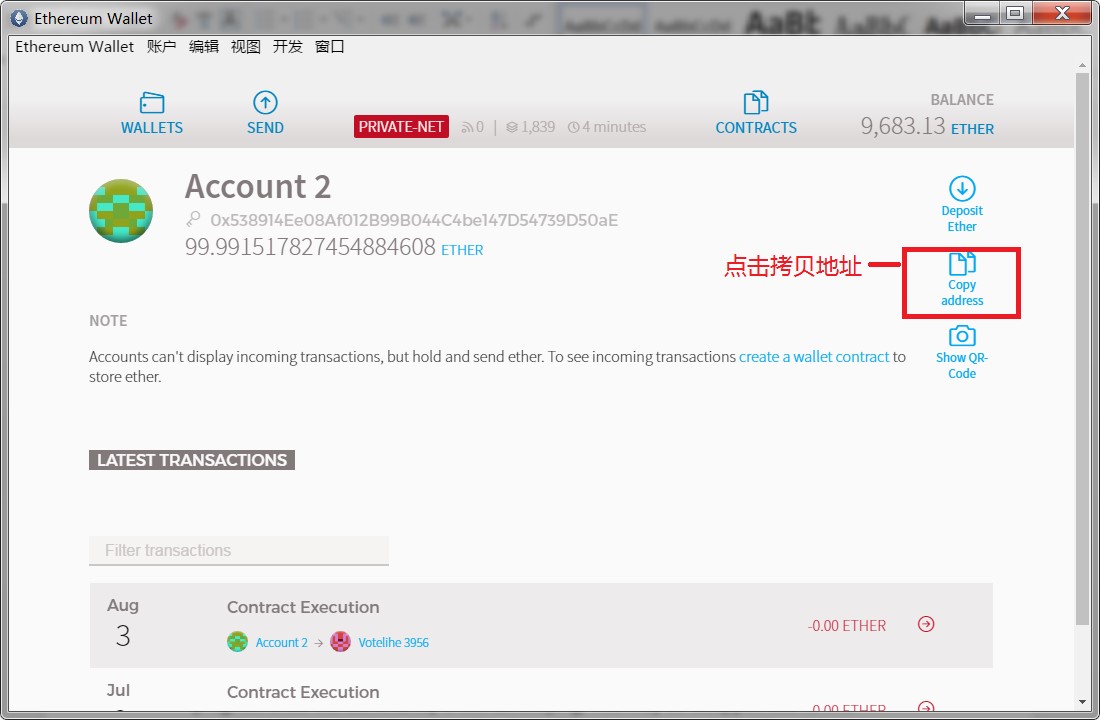
  miner命令括号中的1表示用一个线程进行挖矿，如果不配置，就会让CPU全速运行，影响计算机的使用。

[](http://7fvhfe.com1.z0.glb.clouddn.com/wp-content/uploads/2016/08/21.jpg)

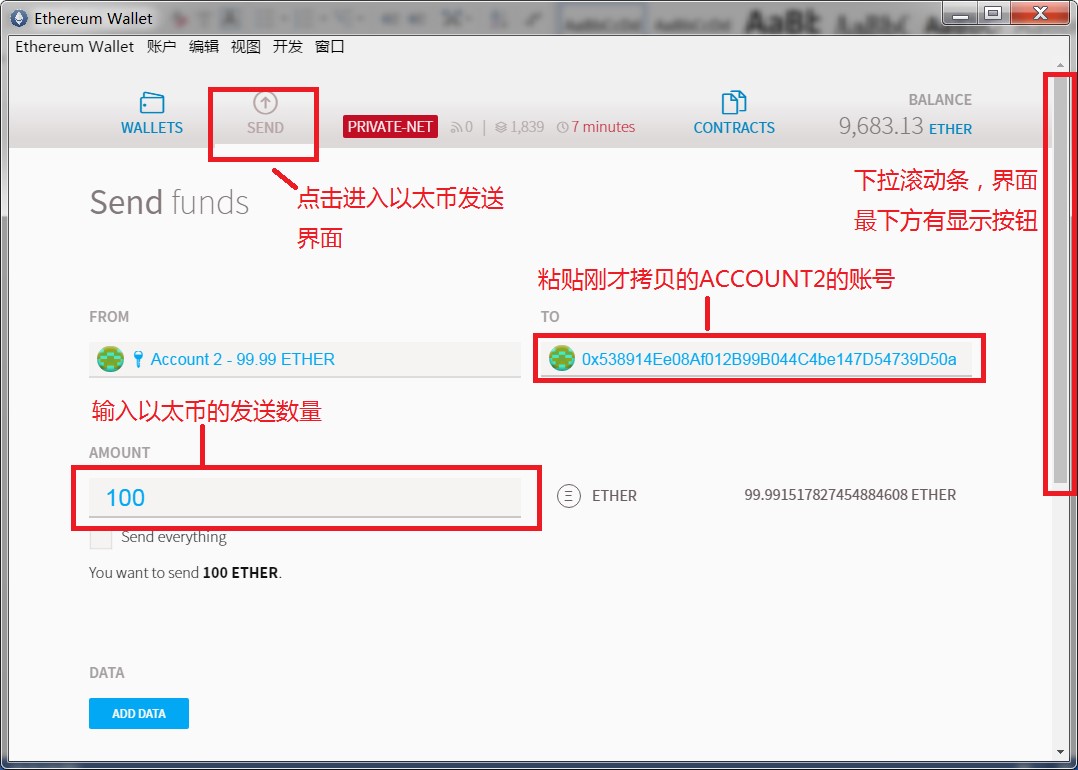
  运行一会后，主账号就会获取很多以太币，这个时候屏幕会快速刷屏，不用管，输入命令miner.stop()停止挖矿。

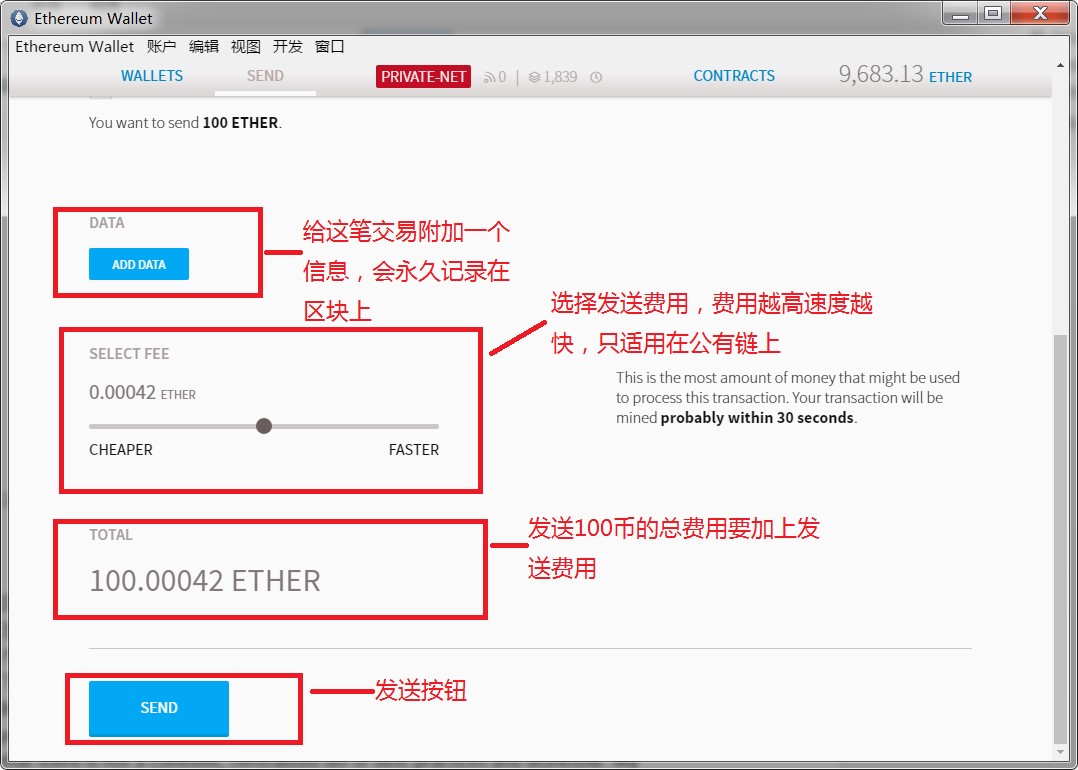
****4、 将一个钱包的以太币转到另一个钱包中****

  先点击ACCOUNT2账号，进入账号详情的界面，点击右侧的“COPY address”，把ACCOUNT2的地址拷贝下来，系统会提示你现在你处在一个测试网络，不要转入真正的以太币到这个账号。

[](http://7fvhfe.com1.z0.glb.clouddn.com/wp-content/uploads/2016/08/3.jpg)

   点击钱包中“SEND”按钮，从MAINACCOUNT给ACCOUNT2转入一定以太币，同时可以看到执行这笔交易的费用是0.00042个以太币。

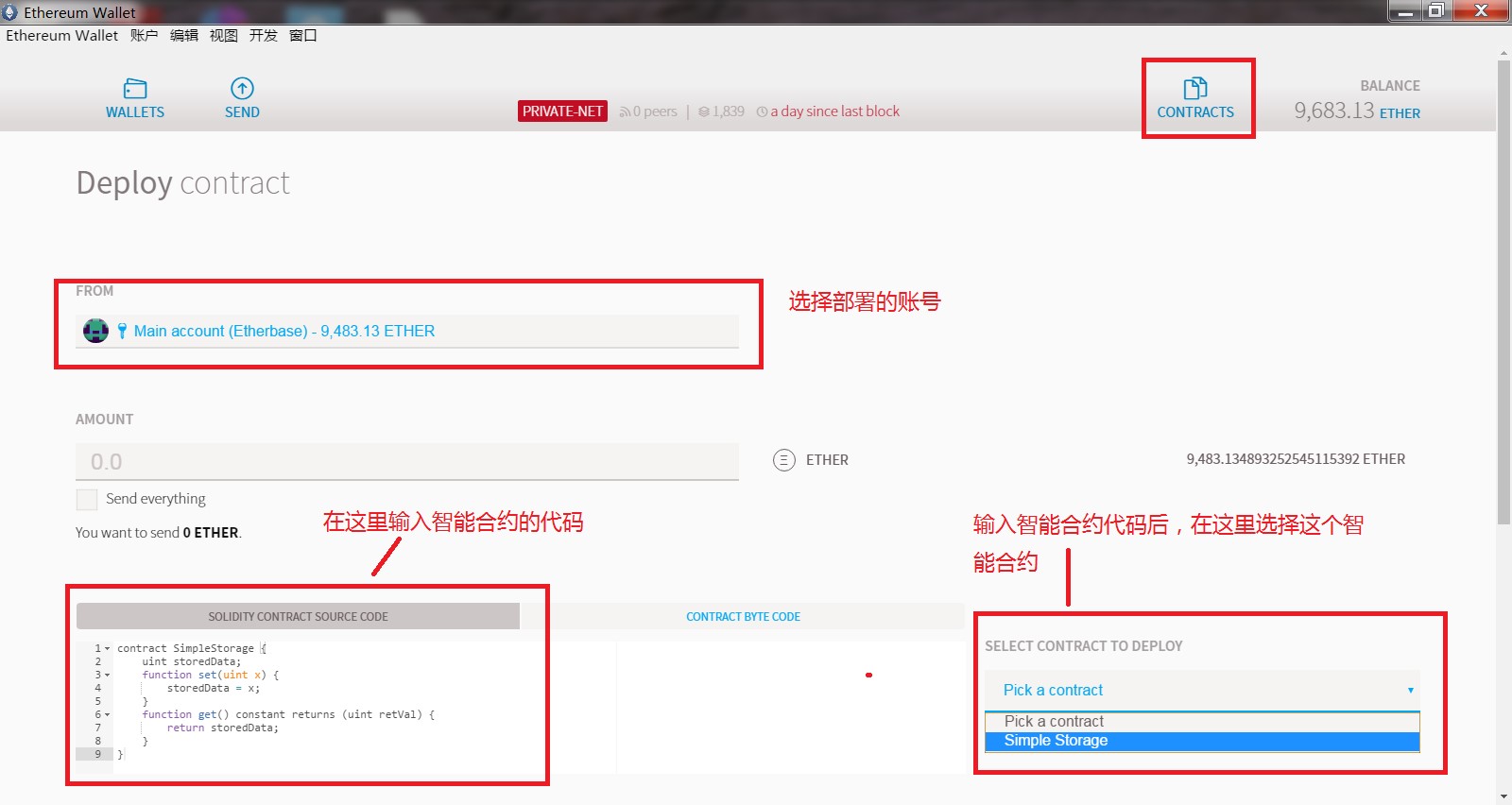
[](http://7fvhfe.com1.z0.glb.clouddn.com/wp-content/uploads/2016/08/4.jpg)

[](http://7fvhfe.com1.z0.glb.clouddn.com/wp-content/uploads/2016/08/5.jpg)

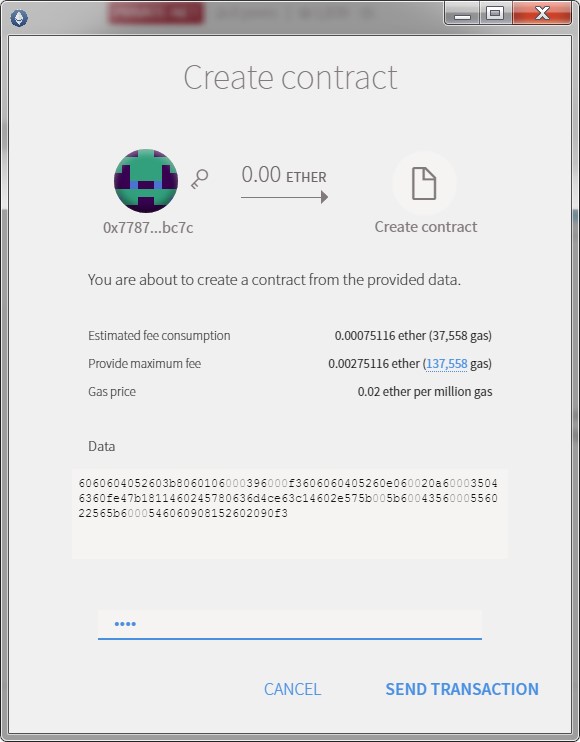
    点击发送后会提示如输入密码，但这时还没有发送成功，根据区块链的交易规则，需要矿工的确认，并且每笔交易需要确认12个块，一个块是16秒的生成时间。切换到Geth程序，输入挖矿命令，直到ACCOUNT2上显示100个以太币，然后停止挖矿。

****5、  部署智能合约****

    点“CONTRACTS”，进入智能合约管理界面，点击“DEPLOY NEW CONTRACT”,开始部署智能合约，选择部署智能合约的账号，并输入智能合约的代码后，如下图

[](http://7fvhfe.com1.z0.glb.clouddn.com/wp-content/uploads/2016/08/6.jpg)

   输入完毕后点击“DEPLOY”，系统会提示输入账号密码，因为部署智能合约是需要费用的。

[](http://7fvhfe.com1.z0.glb.clouddn.com/wp-content/uploads/2016/08/7.jpg)

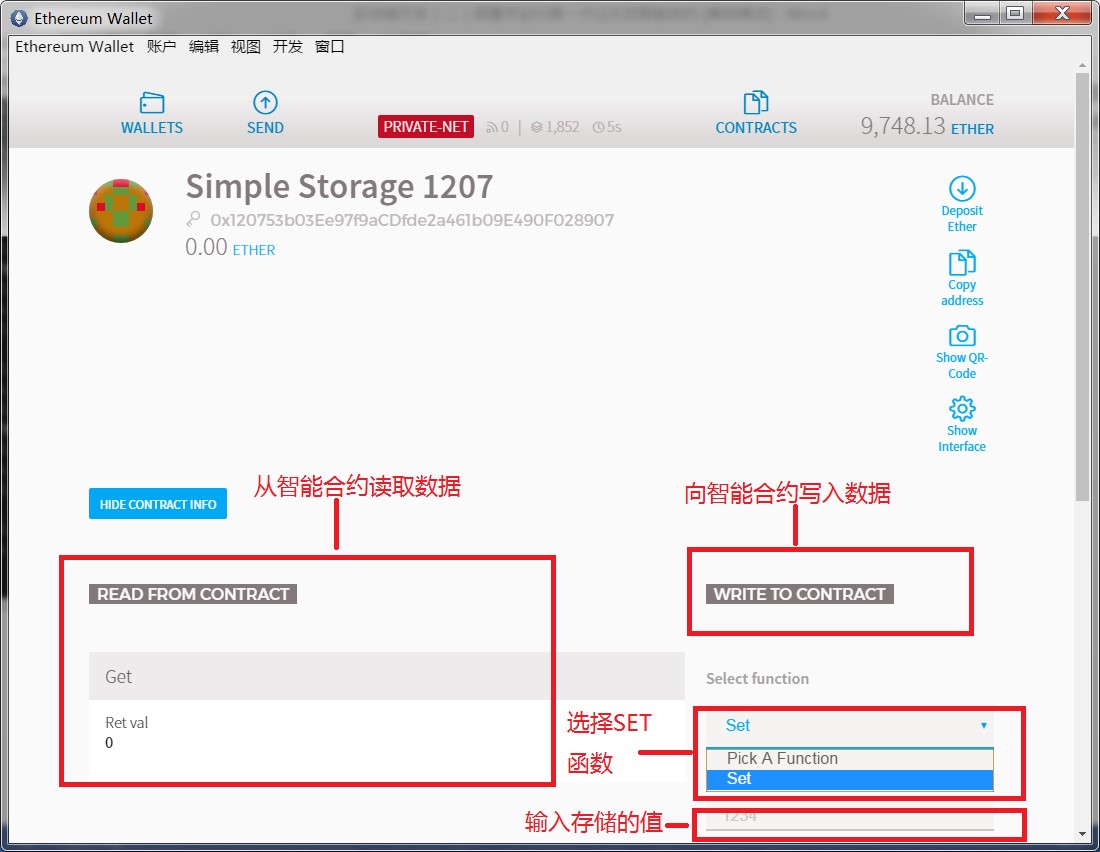
    这个时候是看不到部署的智能合约的，切换到Geth界面，进行挖矿，在12个块后，智能合约就能确认并显示出来。

****三、运行智能合约****

****1、 在本节点上运行智能合约****

  点击“CONTRACTS”进入智能合约界面，可以看到刚才部署的智能合约“SimpleStorage”，点击进入该智能合约，进入详情界面，其中有智能合约写入区域和读取区域，首先启动Geth的挖矿，然后在写入区域选择相应的智能合约函数SET，在下面的数值输入框输入想设置的数值，运行一会后就可以在读取区域看到智能合约函数GET中Retval的返回值有变化。

   其他智能合约的运行也是一样，无非就是函数多点，输入多点。

[](http://7fvhfe.com1.z0.glb.clouddn.com/wp-content/uploads/2016/08/8.jpg)

****2、 在其他节点上运行智能合约****

  此时的智能合约只能自己能看到，别人是无法看到和运行的，如果其他人要运行你部署好的智能合约需要提供一些信息，就是其他教程中所说的智能合约的ABI和地址。

进入刚部署的“SimpleStorage”智能合约界面，右侧有四个按钮

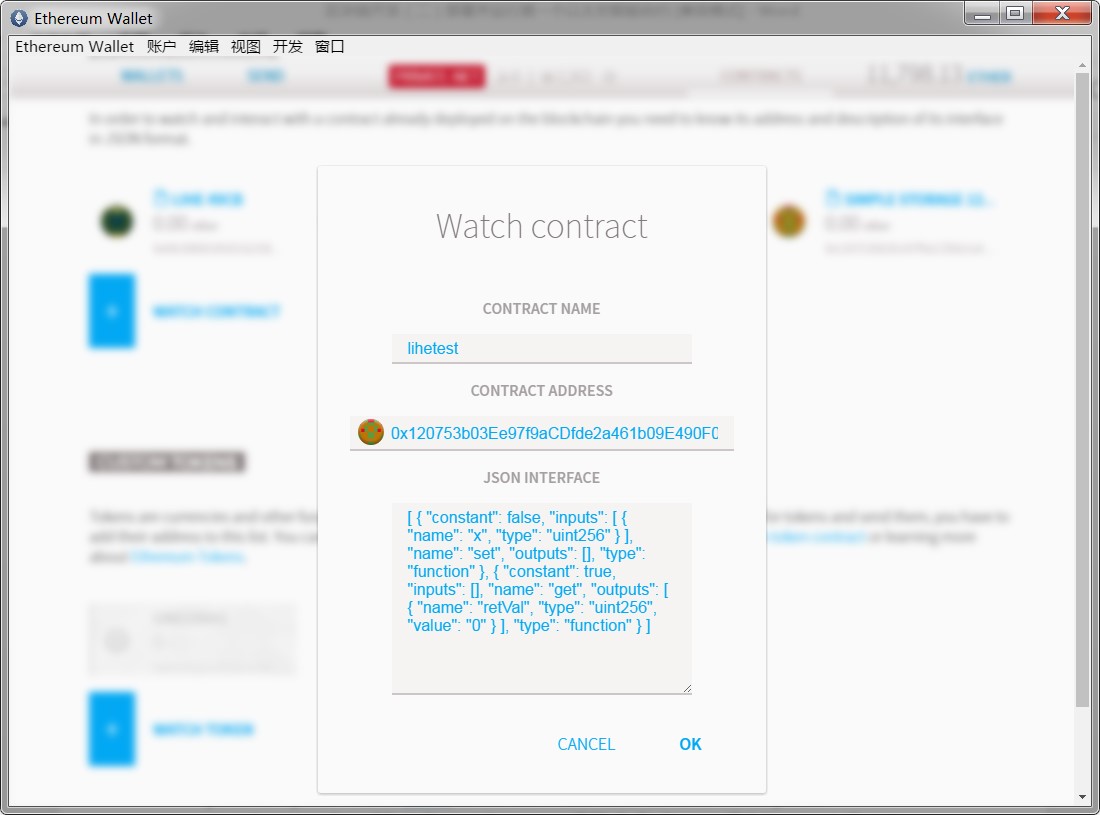
A．“Deposit Eher”：向该智能合约发送以太币

B.“Copy address”：拷贝该智能合约的地址

C．“Show QR Code”：显示一个二维码，如果用手机扫描的话，显示该智能合约的地址

D．“Show Interface”：显示该智能合约的JSON接口，也就是ABI

        首先我们点“Copy address”，拷贝该智能合约的地址，然后点“Show Interface”将智能合约的JSON接口全部拷贝出来，在另一个需要运行智能合约的节点打开Ethereum-Wallet，打开“CONTRACTS”界面点击“WATCH CONTRACTS”添加一个智能合约。

[](http://7fvhfe.com1.z0.glb.clouddn.com/wp-content/uploads/2016/08/9.jpg)

  如上图所示，CONTRACT NAME随便填，CONTRACT ADDRESS填写智能合约地址，JSON INTERFACE填写刚才在”Show Interface “中拷贝的内容。点OK后，就可以看到这个智能合约并运行了。

****四、智能合约的部署原理****

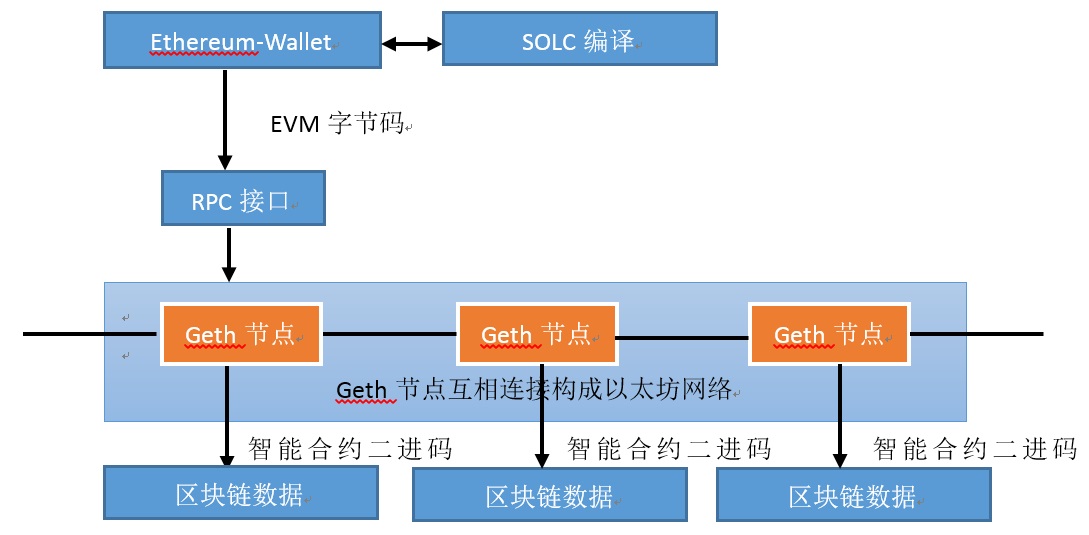
****1、智能合约的部署架构****

  本文介绍的智能合约的部署虽然是在图形化界面编译和执行，但其实最主要的是依赖于后台运行Geth的节点，此时Geth提供了一个RPC的接口向图形化界面的钱包提供区块链的信息。

RPC接口

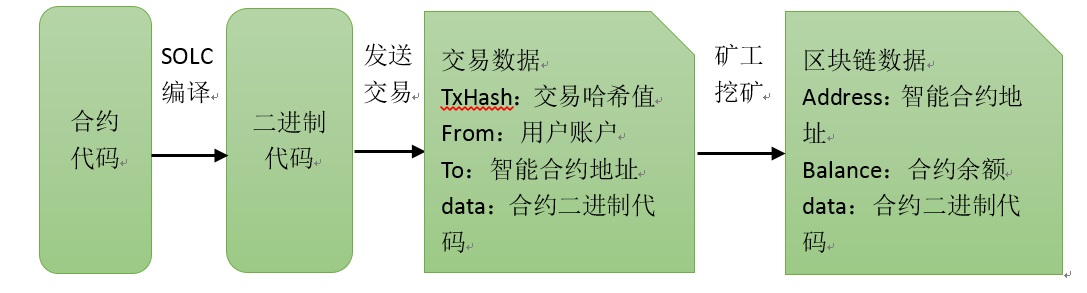
  Geth在8545端口提供了JSON RPC API ，数据传输采用JSON格式，可以执行Web3库的各种命令，比如向前端，比如mist等图形化客户端提供区块链的信息，默认访问地址为[http://localhost:8545](http://localhost:8545/" \t "http://www.8btc.com/_blank)。

  我们部署一个智能合约时，首先Ethereum-Wallet调用SOLC智能合约编译器将代码编译成成EVM字节码，然后将EVM字节码通过Geth的RPC接口发送到以太坊网络，经过全网验证后，同时写入到每个Geth管理的区块链中，架构如下

[](http://7fvhfe.com1.z0.glb.clouddn.com/wp-content/uploads/2016/08/10.jpg)

****2、 部署的数据流****

   首先代码先经过SOLC编译变为了二进制码，然后通过一笔交易来创建智能合约，该笔交易包含了创建者账号、智能合约内容、智能合约的地址这几个关键信息，其中智能合约地址的生成是由创建者的账号和发送的交易数作为随机数输入，通过Kecca-256加密算法重新创建一个地址作为账号。

[](http://7fvhfe.com1.z0.glb.clouddn.com/wp-content/uploads/2016/08/112.jpg)

  部署过程中，需要通过交易来部署，同时数据要存储到区块链上，这些需要使用到GAS。

****交易（Transactions）****

  一笔交易是一条消息，从一个账户发送到另一个账户。交易可以包含二进制数据（payload）和以太币。

  如果目标账户包含代码，该代码和输入数据会被执行。

  如果目标账户是零账户（账户地址是0），交易将创建一个新合约。正如上文所讲，这个智能合约地址不是零地址，而是由合约创建者的地址和该地址发出过的交易数量计算得到。创建合约交易的payload被当作EVM字节码执行。执行的输出做为合约代码被永久存储。这意味着，为了创建一个合约，你不需要向合约发送真正的合约代码，而是发送能够返回可执行代码的代码。

****Gas****

  以太坊上的每笔交易都会被收取一定数量的gas，gas的目的是限制执行交易所需的工作量，同时为执行支付费用。当EVM执行智能合约时，gas将按照特定规则被逐渐消耗，其实GAS就是以太币的比较小的单位，如果以太币比作100元，那么GAS可以看作是1分钱。如果只有以太币，会有问题，以太币是需要大家买卖的，市场会有价格波动。可能会出现比特币这样的状况，一天跌50%涨50%。这个对计算的成本是不能接受的，例如今天做一个加法需要十块钱，明天做一个加法需要一百块钱。所以这里引入gas来解耦。把市场的波动和计算的开销来解耦，也就是说以太币和gas之间是有汇率的，以太币涨没关系，gas价格下降就可以了。它要保证我做同样的计算，消耗的法币是一致的。

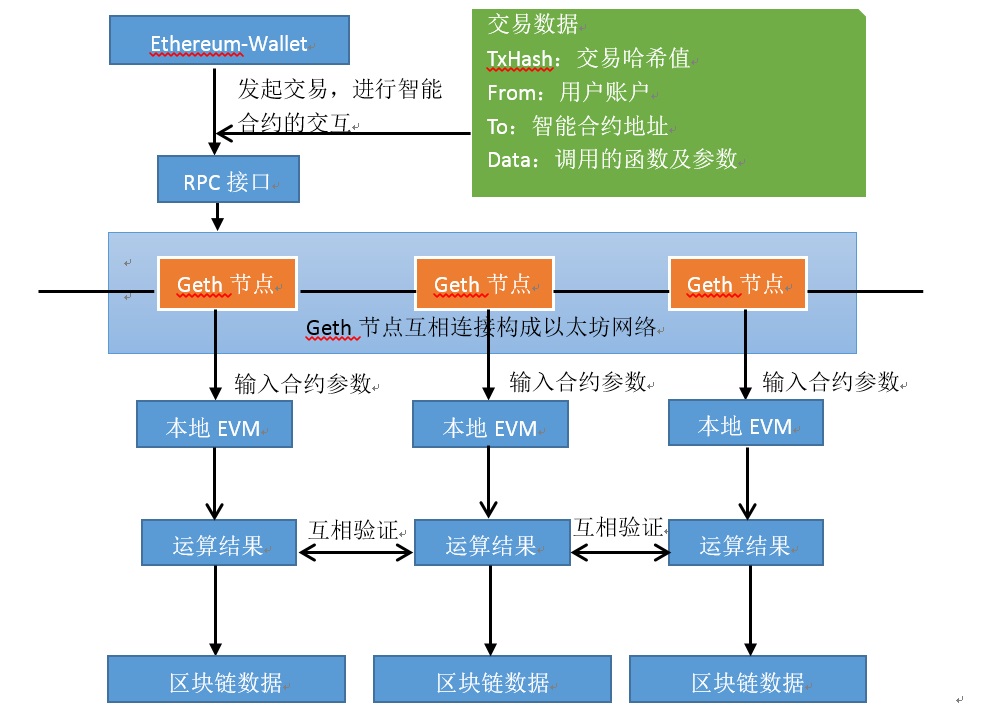
  gas price（gas价格，以太币计）是由交易创建者设置的，发送账户需要预付的交易费用 = gas price \* gas amount。 如果执行结束还有gas剩余，这些gas将被返还给发送账户。

  前文中曾经提到部署智能合约使用了0.00042个以太币，换算成gas就是21000个gas。

  无论执行到什么位置，一旦gas被耗尽（比如降为负值），将会触发一个out-of-gas异常。当前调用帧所做的所有状态修改都将被回滚。

****五、智能合约的运行原理****

  智能合约是部署在区块链的代码，区块链本身不能执行代码，代码的执行是在本地的EVM中，实际上，部署在区块链上代码是能够在本地产生原智能合约代码的代码，可以理解区块链为一个数据库，而客户端从数据库中读取了存储的运行代码，并在本地运行后，将结果写入到了区块链这个数据库中。

[](http://7fvhfe.com1.z0.glb.clouddn.com/wp-content/uploads/2016/08/12.jpg)

    本质上，以太坊的钱包也是智能合约的一个应用，以太坊搭建的是一个可供编写各种应用的平台。下一篇，将详细讲述智能合约的开发编写和调试方法