Transparency and Blockchain Hackathon

QuickStart : Preparation for the Event

We're all looking forward to this Hack Day. We have over 60 registrations for the event including some amazing hackers / coders and a fantastic group of Thinkers / Designers.

We thought some of you might want to get a head start and prepare for Wednesday to be able to jump into the code quicker ;-) So, here are some initial steps you might want to take beforehand.

Readings:

Ethereum White Paper:
https://www.ethereum.org/pdfs/EthereumWhitePaper.pdf
Recent blog posts by the dev team: http://blog.ethereum.org/

What you'll need:

We will be using the latest Alethzero client proof of concept version 7. As a proof of concept this is pre-alpha software and prone to break - as such we have built a virtual machine running a stable version of the client which we would like you to download. The torrent file is included - if you do not wish or are unable to download this file then the instructions for creating the vm are below.

We would like everyone to have a chance to use the client so if everyone could come prepared with the correct software installed, and our VM downloaded - this will save us a lot of time on the day.

You will need :

- Skype
- Vmware Fusion 7 or Workstation 10 (free trial available from http://www.vmware.com/products/fusion/fusion-evaluation)
- Download the torrent file containing the VM here : http://goo.gl/LgZQG4

Creating your own VM:

It is possible to create your own vm, by first downloading the software from the vmware website, creating a VM using Ubuntu linux, and then following the instructions on our building in ubuntu wiki.
The process of constructing **this** vm is detailed below so that you might follow it yourself if you would prefer not to download our (rather large) torrent.

1) Download VMWARE fusion or workstation trial from the VMware website.
2) Download the Ubuntu disc image (ISO) from the ubuntu website

Transparency and Blockchain Hackathon

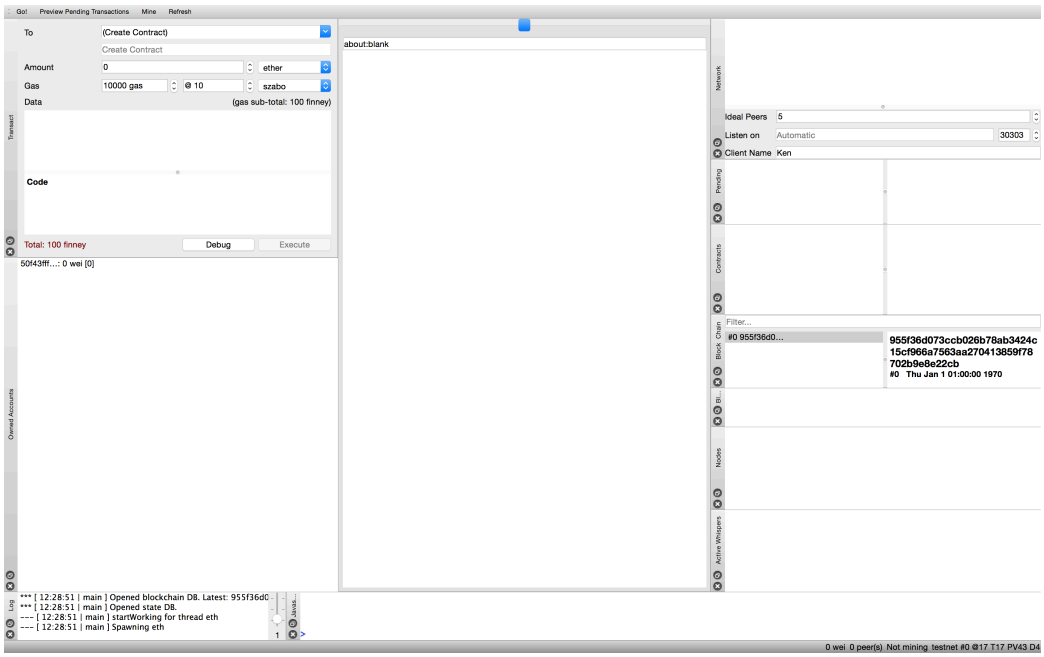
3) Once VMware is downloaded open the file and follow the installation instructions. Run the program and select File> New and create a virtual machine. Choose to create it from a 'disc or image' and select the Ubuntu ISO you downloaded previoulsy. Choose install and setup your vm as you would like it. **make sure you specify the VM has at least 2GB of memory or the build of alethzero will fail.**
4) Once ubuntu is installed you will want to update your package manager. Open the terminal and type:
   - `sudo apt-get update && sudo apt-get upgrade`
5) Now return to your host sytem and find the vmware window go to Virtual Machine > install VMWare tools. This will mount a cd containing files needed by linux to use the hosts sytems hardware and copy paste between your host system and your vm. Go back to your vm and navigate to the drive marked vmware tools - extract the vmware tar ball to the desktop. Then return to the terminal and input:
   - `cd Desktop/vmware-tools-distrib`
   - `sudo ./vmware-install.pl -d`
6) Shutdown and take a snaphot of your vm to revert to later.
7) Start-up your vm and you should now find that you can alter the size of the screen and copy paste between your host and the vm. If you cant google vmware tools and see if you can trouble shoot what went wrong.
8) You now have the base to install the dependencies for Alethzero. Start by entering the below in the terminal:
   - `sudo apt-get install build-essential g++-4.8 git cmake libgmp-dev \`
   - `libboost-all-dev automake unzip libtool libleveldb-dev yasm libminiupnpc-dev \`
   - `libreadline-dev scons libncurses5-dev qtbase5-dev qt5-default qtdeclarative5-dev \`
   - `libqt5webkit5-dev \`
   - `libcurl4-openssl-dev libjsoncpp-dev libargtable2-dev libboost-test-dev cmake`
9) Shutdown and take a snapshot ;)
10) Now add the cryptopp dependencies:
   - `git clone https://github.com/mmoss/cryptopp.git`
   - `cd cryptopp`
   - `sudo scons --shared --prefix=/usr`
   - `cd ..`
11) Shutdown and take a snapshot :p

12) Install the JSONRPC library - currently we are using an older version which may not be compatible with future builds. Make sure you check for updates on the 'building on ubuntu' wiki if you experience issues with the javascript api:
    - `git clone https://github.com/cinemast/libjson-rpc-cpp`
    - `cd libjson-rpc-cpp`
    - `git reset --hard eaca2481e2889d5a5b748383fb02b1d395969cd4`
    - `mkdir -p build`
    - `cd build`
    - `cmake .. && make`
    - `sudo make install   #Not required, but makes it easier to use`
    - `sudo ldconfig       #only required for linux`
    - `cd ../../`
13) Shutdown and take a snapshot :p
14) Install Unittests:
    - `git clone https://github.com/ethereum/tests`
    - `cd tests`
    - `git checkout develop`
    - `cd ..`
15) Shutdown and take a snapshot (doh).
16) Now you are going to clone the cpp-ethereum develop git:
    - `git clone https://github.com/ethereum/cpp-ethereum`
    - `cd cpp-ethereum`
    - `git checkout develop`
    - `On this build we have used commit e81197606ae29c4be3ac37ad4e8aaf699fbcd039 if you would like to use this commit also use command:`
    - `git reset --hard e81197606ae29c4be3ac37ad4e8aaf699fbcd039`
17) Inside the cpp-ethereum folder you will create a build folder and run the build process:
    - `mkdir build`
    - `cd build`
    - `cmake .. -DCMAKE_BUILD_TYPE=Release && make`
18) Once completed shutdown and take a snapshot.
19) Voila! you now have a Linux VM with a copy of the Alethzero client running. This is an optimal approach to development during the pre-alpha stage with new versions of the client being released daily - you can try new builds and features without sacrificing a stable development environment.

About the Client: We'll be using AlethZero, the C++ implementation of Ethereum designed for developers. In particular, we're going to install the

Transparency and Blockchain Hackathon

'development' version, which contains all the latest goodies.
If all went well, you should see something like this after you start AlethZero, depending on your platform and screen resolution:



If your resolution varies, and not all the interface is visible, manually close all panes in the display by clicking their 'x' symbols until they are all gone, resize the screen to fit your resolution and re-add them manually by right clicking just below the titlebar (and just to the right of the 'refresh' button).

At the center of the screen is a browser window, a Webkit view to be exact. You can browse the existing web from there, just like any other browser, try google for example.

The rest of the panels contain debug and technical information, but pay careful attention to the transact pane - it is where we will be uploading contract code and sending transactions.

If you would like to try using the browser on the testnet just hit go! You should connect to the test network and see the blockchain download. Hit mine if you would like to get some test ether (this may take some time). Try sending a transaction to another user, or uploading a contract.

For Further Information:

www.github.com/ethereum
reddit.com/r/ethereum
forum.ethereum.org
IRC Channels #Ethereum and #Ethereum-dev