C++ and Smart Contract Development

By Bucky Kittinger
Breakout Talking Points

1) Getting Started with EOSIO and running a minimal chain.
2) An explanation of the tools used.
3) Creating a hello world smart contract.
4) What are multi_index tables? And why you need them.
5) Creating a contract that uses tables.
6) Creating a contract that uses inline actions.
7) What is eosio.token? And how to use it.
Getting Started

- Git clone the vt-blockchain-bootcamp-starter repo.

- Run the script `first_time_setup.sh`
  - This will install the binaries for eosio and eosio.cdt.

- Run the script `start_blockchain.sh`
Getting Started...

Very Brief Overview of EOSIO Tools:

- nodeos
  - The software that is the “block chain”.
- cleos
  - The client side software to communicate with nodeos.
Because of the vastness of configuration options for nodeos, it is best to use the `start_blockchain.sh` script for your work.
Getting Started…

Common Cleos Commands

- create key
  - Used to generate public/private key pairs.
- create account
  - Needed to create a new account on the block chain.
- push action
  - Used to send actions to your smart contracts.
- set code | abi
  - Sets the code or ABI to an account.
- **get info**
  - Gets the current head block from the chain and displays the information from it.

- **wallet create**
  - Used to create wallet to store your private keys for signing transactions.

- **wallet import**
  - Used to import your private keys to a specified wallet.
Eosio.CDT

(Contract Development Toolkit)

Tools of Interest:

- eosio-cpp
- eosio-cc
- eosio-ld
- eosio-init
Hello, World!

```
#include <eosiolib/eosio.hpp>
using namespace eosio;

CONTRACT hello : public contract{
    using contract::contract;
    ACTION hi(name user) {
        print_f("Hello : %\n", user);
    }
};

EOSIO_DISPATCH( hello, (hi) )
```
Compiling hello.cpp

eosio-cpp --abigen -o hello.wasm hello.cpp

This will produce the WebAssembly binary hello.wasm and ABI file hello.abi
This is the glue that binds EOSIO tooling to your smart contract, e.g. it allows tools like cleos to push actions to your contract.
**multi_index**

Persistent data in smart contracts.

Eosio uses tables to allow the user to have persistent data between invocations of their smart contract actions.

`multi_index` is a way in which you can index the same contained data with more than one index. (A `std::vector` can only be indexed by one index, a `size_t` value).
Inline Actions

Your way to communicate with other smart contracts on the block chain.