**TRAINING (Tentative 10 Day Plan)**

Day 1: JavaScript (Server side)

Day 2-5: Solidity and Eris

Day 6-10: BigchainDB and continued Solidity/Eris

**Day 1**: Training on node JavaScript.

Goal: Get up to speed with how to use server-side JavaScript and create API calls.

Task: Create a JavaScript file that contains one variable (call it state). Expose the file on a port to allow API calls using either the express or http library. This file should allow two API calls. Setting and getting of the state variable.

**Day 2**: Eris-Contracts JavaScript library.

Goal: Develop an understanding of the Eris-contracts library, and its ability to communicate with smart contracts.

Task: A smart contract will be provided that sets and gets a variable. The JavaScript written on *Day 1* is to be modified. The API calls should now communicate with the provided smart contract on the block chain.

**Days 3 and 4**: The block-chain in Eris.

Goal: Develop an understanding of interacting with chains and our chain configuration model. Get feet wet with Solidity.

Task: Create a smart contract that is permissioned to one user. It should represent a bank account. It should allow one to make deposits and withdrawals with a few constraints. Deploy the contract onto a chain.

**Day 5**: Talking to the block-chain with the EPM.

Goal: Gain an understanding of the EPM. Continue interacting with contracts through javascript as well.

Task: Create a javascript file that interacts with the bank account contract from *Days 3 and 4*. It should allow API calls. Also, write an EPM job for deposits and withdrawals. The EPM should specify the account that interacts with the contract.

**Day 6:** Crash Course on Bigchain

Goal: Gain an understanding of the bigchain driver and the encryption folder system.

Task: Using cURL, make a transaction with the driver to add values and get values.

**Day 7 and 8:** Oraclizers

Goal: Understand the bigchain Oraclizer.

Task: Write out the flow of the bigchain Oraclizer and how it works. If there is time, do the same for the verification Oraclizer.

**Days 9 and 10**: Interaction with the Oraclizers

Goal: Understand how to call the oraclizers.

Task: From a javascript, call both the bigchain Oraclizer and verification oraclizers. This should include calling the bigchain Oraclizer to put in data, wait for the callback response and store it as a variable. The same should be done using the callback response to get the inputted data. For the verification Oraclizer, a verification query should be done.

Day 2 provided contract on next page:

contract test

{

string val;

function test()

{

val = "";

}

function getVal() returns (string value)

{

value = val;

}

function setVal(string newVal)

{

val = newVal;

}

}