

```

1 pragma solidity ^0.4.21;
2
3 // import "github.com/oraclize/ethereum-api/oraclizeAPI.sol";
4 import "oraclize-api-master/oraclizeAPI_0.4.sol";
5
6 contract SmartExchange2 is usingOraclize {
7
8     string blockchainInfoApiAddress = "https://testnet.blockchain.info/q/
getreceivedbyaddress/";
9
10    string target_btc_address;
11    address eth_dest_address;
12    string target_btc_amount;
13    uint balance;
14
15    event logMe (string message);
16    event logAddress (address dest_address);
17
18    // Constructor
19    function SmartExchange2(string btc_address, address eth_address, string satoshi_amount)
payable{
20        target_btc_address = btc_address;
21        eth_dest_address = eth_address;
22
23        // Due to complicated type conversion, let's keep it as string.
24        target_btc_amount = satoshi_amount;
25
26        balance = msg.value;
27
28        emit logAddress(eth_address);
29        emit logMe(strConcat("Will transfer contract balance to address from previous event,
if ",
30            satoshi_amount,
31            " satoshi will be tranferred to ",
32            btc_address));
33    }
34
35    // Oraclize callback
36    function __callback(bytes32 myid, string result) {
37        if (msg.sender != oraclize_cbAddress()) throw;
38
39        if (keccak256(target_btc_amount) == keccak256(result)){
40            eth_dest_address.transfer(balance);
41        }
42        else {
43            emit logMe("Unknown Oraclize response received.");
44            emit logMe(result);
45        }
46    }
47
48    function update() {
49        oraclize_query(60, "URL", strConcat(blockchainInfoApiAddress, target_btc_address));
50        emit logMe("Query sent to Oraclize.");
51    }
52 }
53 }

```