SciChain: Science on the Blockchain

Authors (in alphabetic order): Bruno Nazário Coelho, Fabio Cardoso, Igor Machado Coelho and Vitor Nazário Coelho
Discord users: igormcoelho, vncoelho, fabiocardoso, brunonazario
Main developers: fabiocardoso, vncoelho, igormcoelho
Official website: https://scichain.org
GitHub code: https://github.com/igormcoelho/scichain
Project Wallet: AdfJSyxgpTZTGSWCnApWPA49hQG7ANVdht
Contract hash: 0xfc47d1b7b2fe38096433af7c5e543033d87a86d1
MD5: 8cb15696eabd0eb391a4a6799fb02bcb

Description:

Science is the core of the modern society, bringing us a deeper understanding of the world and providing us with extraordinary tools to cure diseases, allow fast transportation and communication. We cannot put Science into words, but we can try to make the processes behind it even better, faster and more transparent.

Our team includes experienced researchers on Computer Science, Engineering and Material Physics fields, and we want to give Science a better tool for dealing with one of its most important feature: *the peer-review system*.

Perhaps not so many people know that, for a scientific study to be properly "accepted" as "true", it is commonly required for it to pass the *peer-review system*, where unknown colleagues of the same field judge the article methodology, background and results. If it passes peer-review and is published in a qualified journal, the study is considered valid and is referenced in future studies. And the chain goes on...

In order to achieve a truly transparent process in peer-review, we propose **SciChain**. As **SciChain** is build on top of *NEO Blockchain* (a blockchain can be seen as a public database where the defined rules are always respected), there will be no way of "cheating" in peer-review, for example, indicating known friends as reviewers, breaking the *double-blind* principle (what means that both the author and reviewer must not know who is each other), and many other bad actions that hurt our precious Science :'(

One the many flaws in scientific studies is also related to low reproducibility of scientific studies, since few journals are available and they usually "accept" only "new" results. Most of these journals require authors to pay for the download of each publication, what strongly undermines the capacity of the researchers (mainly in underdeveloped countries) to be updated with newer discoveries.

Usage:

The SciChain described in this document correspond to alpha 0.1 version, an MVP to demonstrate the most basic functions, although many others are already under development in the GitHub contract.

Operation Register Journal/Editor

In order to create a new journal, the editor in charge can pass its address (20 bytes) to RegisterEditor function.

neo> testinvoke fc47d1b7b2fe38096433af7c5e543033d87a86d1 "526567697374657245646974f722829" ["AUk19KeZqgry WfxfvfAkCcEFLNGhzoPivi"]
Used 1.472 Gas

This function returns an EditorKey (32 bytes). This EditorKey plays an important role in the next operations.

If an Editor tries to register multiple times, system will return the appropriate message: “Editor is already registered”.

Operation Register Reviewer/Referee

Each editor can register a set of reviewers using RegisterReviewer function, passing both its address (20 bytes) and the reviewer address (20 bytes).

neo> testinvoke 0xfc47d1b7b2fe38096433af7c5e543033d87a86d1 "526567697374657252657669657765722829" ["AUk19K eZqgryWfxfvfAkCcEFLNGhzoPivi","APLJBPhtRg2XLhtpxEHd6aRNL7YSLGH2ZL"]
Used 1.818 Gas

This function returns true (1) if reviewer is succefully registered. If the reviewer is included many times, system will return the appropriate message: “Reviewer already registered”.

This function has already many internals that deal with Endorsement invocations (interacting with several other functions), and in future works (see roadmap below) we will allow for self-managed journals to filter reviewers automatically (integrated with a Dynamic Invoke smart contract called Smart Rules, in the same github project).

Operation Request Article

When an author intends to submit a paper to an specific journal (scientific community) , it can call RequestArticle function. The first parameter addresses the self address, then the

data content (title + abstract + keywords), then the editor address. Note that the editor itself is allowed to participate.

"526571756573744172746963 6c652829"
["AK2nJJpJr6o664CWJKi1QRXjqeic2zRp8y","HelloWorldPaper","AK2nJJpJr6o664CWJKi1QRXjqeic2zRp8y"]

This function returns an special ProcessKey (32 bytes) that can track the author communication with the editor/journal, and store process-related information. Initially, the payload is used for title, abstract and keywords submission, and this information is used by the editor to decide upon reviewer assignment. This part is intended to connect with Endorsement functions in order to allow for autonomous reviewer selection in SciChain self-managed journals.

Operation Process Status

When invoking GetProcessStatus function, the author can track the submission. The return value is a byte definition for the status (taken from ProcessData field):

"47657450726f6365737353746174747573 2829"
[b'3dc75e810131dcd386db94f456b8ba9f2305e0b771e4676539eb5b98ada6a14a']

Status options are:

Not found -> 0
Process rejected -> 1
Waiting editor acceptance -> 2
Waiting encrypted article -> 3
Waiting Reviewers send grades and comments -> 4
Waiting Editor approval -> 5
Waiting decrypted article -> 6
waiting reviewers approval -> 7
Waiting for publication -> 8
Published -> 9

The ProcessData contains:

status (1 byte) :
author key (32 bytes)
editor key (32 bytes)
number of reviewers (1 byte)
reviewers keys (32 bytes / reviewer)

Operation Send Data to Process

Finally, author can send information in plain-format or specially encoded for each reviewer that is enclosed by ProcessData. This part can be handled by the graphical interface, and can also integrate with different and specific Journal rules. This point impacts directly if journal chooses **Single-Blind**, **Double-Blind**, and different rules for **Open-Science**, including **Open-Review** and rebuttal options. This function can be used to change different process Status codes and change the payload in specific situations.

"53656e6444617461546f50726f636573732829"
["AK2nJJpJr6o664CWJKi1QRXjqeic2zRp8y",b'3dc75e810131dcd386db94f456b8ba9f2305e0b771e4676539eb5b98ada6a14a',b'0302']

Future works and roadmap:

There are many future works for SciChain, and we will continue to develop them, as the idea emerged in casual talks in early February 2017. We believe this project can be the start of a revolution in the way we do science, and perhaps this can bring birth to many other similar projects that will fulfill the purpose of bringing a more transparent science.

Our development roadmap includes:
Q2 2018
- Integrating basic smart contract operations with scichain.org website
- Improving hability endorsement calculations in smart contract
- Integrating SmartRules smart contract with Dynamic Invoke, to allow remote/personalized judgements

Q3 2018
- Integrating cryptographic end-to-end compilations in website
- Integrating advanced smart contract operations with scichain.org website

---

Appendix (import configuration):

import contract fc47d1b7b2fe38096433af7c5e543033d87a86d1.avm "0710" "05" true false
contract properties: 3
Please fill out the following contract details:
[Contract Name] > SciChain
[Contract Version] > 0.1
[Contract Author] > Bruno Nazario Coelho, Fabio Cardoso, Igor Machado Coelho and

Vitor Nazario Coelho
[Contract Email] > igor.machado@gmail.com
[Contract Description] > Science on the Blockchain
Creating smart contract....
Name: SciChain
Version: 0.1
Author: Bruno Nazario Coelho, Fabio Cardoso, Igor Machado Coelho and Vitor Nazario Coelho
Email: igor.machado@gmail.com
Description: Science on the Blockchain
Needs Storage: True
Needs Dynamic Invoke: True
{
"script":
"52c56b6c766b00527ac46c766b51527ac46c766b00c31247657450726f63657373537461747573732829876422006c766b51c3c0519c63080000616c75666c766b51c300c361651702616c75666c766b00c3105265717565737441727469636c652829876432006c766b51c3...

[I 180225 23:42:35 Transaction:611] Verifying transaction:
b'fb90ab8e9639534b74c6654135c38d712db7d867e8b9eaab705ea9b234c6a1a5'
Relayed Tx:
fb90ab8e9639534b74c6654135c38d712db7d867e8b9eaab705ea9b234c6a1a5

Appendix (ABI):


```
{
    "hash":"0xfc47d1b7b2fe38096433af7c5e543033d87a86d1",
    "entrypoint":"Main",
    "functions":
    [
        {
            "name":"Main",
            "parameters":
            [
                {
                    "name":"operation",
                    "type":"String"
                },
                {
                    "name":"args",
                    "type":"Array"
                }
            ],
            "returntype":"Any"
        },
        {
            "name":"GetProcessStatus",
```

```
        "parameters":
        [
            {
                "name":"processkey",
                "type":"ByteArray"
            }
        ],
        "returntype":"Integer"
    },
    {
        "name":"RequestArticle",
        "parameters":
        [
            {
                "name":"address",
                "type":"ByteArray"
            },
            {
                "name":"data",
                "type":"ByteArray"
            },
            {
                "name":"editorAddress",
                "type":"ByteArray"
            }
        ],
        "returntype":"ByteArray"
    },
    {
        "name":"SendDataToProcess",
        "parameters":
        [
            {
                "name":"address",
                "type":"ByteArray"
            },
            {
                "name":"processkey",
                "type":"ByteArray"
            },
            {
                "name":"data",
                "type":"ByteArray"
            }
        ],
        "returntype":"Boolean"
```

```
},
{
    "name":"ReceiveFromProcess",
    "parameters":
    [
        {
            "name":"address",
            "type":"ByteArray"
        },
        {
            "name":"processkey",
            "type":"ByteArray"
        }
    ],
    "returntype":"ByteArray"
},
{
    "name":"Publish",
    "parameters":
    [
        {
            "name":"address",
            "type":"ByteArray"
        },
        {
            "name":"processkey",
            "type":"ByteArray"
        }
    ],
    "returntype":"ByteArray"
},
{
    "name":"RegisterEditor",
    "parameters":
    [
        {
            "name":"address",
            "type":"ByteArray"
        }
    ],
    "returntype":"ByteArray"
},
{
    "name":"RegisterReviewer",
    "parameters":
    [
```

```
                {
                    "name":"address",
                    "type":"ByteArray"
                },
                {
                    "name":"ReviewerAddress",
                    "type":"ByteArray"
                }
            ],
            "returntype":"Boolean"
        },
        {
            "name":"Endorse",
            "parameters":
            [
                {
                    "name":"address",
                    "type":"ByteArray"
                },
                {
                    "name":"toaddress",
                    "type":"ByteArray"
                },
                {
                    "name":"skill",
                    "type":"ByteArray"
                }
            ],
            "returntype":"Boolean"
        },
        {
            "name":"GetEndorseData",
            "parameters":
            [
                {
                    "name":"address",
                    "type":"ByteArray"
                }
            ],
            "returntype":"ByteArray"
        }
    ],
    "events":
    [
    ]
}
```