

Whitepaper - WCS - published by: wcs:root : Sat Jan 18 17:06:02 CET 2020 #  
World Compensation System (WCS) **Keywords:** #2020, #Blockchain, #Inter-  
netOfValue, #RSK, #DeFi, #DeFiApp, #DFApp, #wcsDFApp

**Classification:** DeFi : EcoSystem : @rif\_os : **Version:** 0.0.1 (2020-01-15)  
**Status:** *in work*

## Purpose

Simple. Eliminate money invisibility.

## User-story

I dare you to throw out all your money, all your papers and coins  
and individual national currencies, and start over.

Develop an international monetary system that is wide open, totally  
visible, immediately traceable, completely accountable. Establish a  
Worldwide Compensation System by which people would be given  
Credits for services rendered and products produced, and Debits for  
services used and products consumed.

Under the new Worldwide Compensation System, WCS, the transfer  
of Debits and Credits would be immediate and totally visible. That  
is, anybody and everybody could inspect the account of any other  
person or organization at any time. Nothing would be kept secret,  
nothing would be ‘private’

Everything would be on the system of Credits and Debits. Returns  
on investments, inheritances, winnings of wagers, salaries and wages,  
tips and gratuities, everything. The WCS would deduct 10 percent  
of all earnings each year from the income of those *voluntarily re-  
questing* such a deduction. Everyone in the society would be able  
to observe who was choosing to offer the 10 percent for the general  
good of all, and who was not. And everyone’s records would be open  
to everyone else. And nothing could be purchased without Cred-  
its. There would be no other negotiable currency. ([source:#CWG,  
@realNealWealsh](http://ISBN))

**Short-name:** WCS, DeFiApp, wcsDFApp

**Disclaimer:** All quoted phrases are verbatim copies found at Conversations  
with God. Book Two (see Bibliography)

**Bibliography:** - Conversations with God (c) 1997 Neale Donald Walsch. ISBN  
978 0 340 76544 9

## Use-cases

- Typical Use-cases and User-workflow

## Community

- [[@WorldCompensationSystem \(Twitter\)](https://twitter.com/WorldCompensationSystem)](https://twitter.com/WorldCompensationSystem)

## License

Code is under the The Unlicensed. Documentation is under the Creative Commons Attribution license.

## Contributing

Please read our Contribution Guide and [Code of Conduct]

## Donations

BTC-Address: | 0x123..0000 |

# Whitepaper

## Table-of-contents

../tools -  
../networking - # Networking  
../dapps - ### Distributed Apps (status:*in-work*)  
../whitepaper - Whitepaper - WCS - published by: wcs:root : Sat Jan 18 17:06:02 CET 2020  
../operations - # Operations  
.. - # World Compensation System (WCS)  
../lang - ## Languages  
../lang/c - ##### Language C (C99)  
../arch/dfs - # Distributed File-System(s)  
../arch/dfs/dFSwcs - # Distributed File-System WCS

- ../arch - ## Architecture
- ../arch/fs - # File System
- ../users - # Users
- ../commands - ### Commands (status:*in-work*)
- ../apps - ### Applications
- ../services/wcsServer - # wcsServer - World Compensation System server
- ../services - ### Services

## Operating (Eco-)System Concept

1. wcsO(E)S - WCS Operating Eco-System
  1. Platform
  2. Use-cases
    1. Notation
    2. Welcome home
    3. Help
      1. View description manual
    4. User management
    5. File-System (Minimum commands)
      1. Present working directory (pwd)
      2. Listing files (ls)
    6. Applications, Services, Commands and Tools
      1. Running local tool system-service
      2. Running local command system-service
      3. Running local user-application
      4. Running remote user-application
      5. Commands
      6. Tools
    7. Local Services
      1. Financial
      2. Asset Management
    8. Value Operations
  2. **1. Transactions**
    1. Distributed Services
    2. Group citizenship
    3. Work – Get Idle Task (according to current citizenship)
    4. Donate
    5. Contribute to Nation's taxes
    6. Value Creation (out-of-thin-air)
    7. Credits

8. Value Transfer - Request for service
9. Value Transfer - Investment
  1. Communities
10. Running DApplication in debug-mode
11. User-management
12. User-Application development
  1. DApplication development help
13. Returning home
14. Lend User-service
15. Claim lend User-service
16. Borrow service ## (Eco-)System Application notes

```

homeland$
homeland$help
homeland$man ver
homeland$ver
homeland$home
homeland$user create user1
homeland$pwd
homeland$ls
homeland$stat
homeland$cmd1 --verbose
homeland$app1 --verbose
homeland$dapp1 --verbose
homeland$commands
homeland$tools
homeland$app1
homeland$wallet
homeland$credits
homeland$debts
homeland$assets
homeland$send 2 user1
homeland$credit tetris
homeland$neighborhood
homeland$discover
homeland$connect
homeland$ls
homeland$ls nations
homeland$citizen
homeland$citizen federation1
homeland$idle federation1
homeland$donate -idle neighborhood
homeland$tax 8
homeland$offer -idle federation1
homeland$offer 8
homeland$offer 6

```

```

homeland$credits
homeland$tetris
homeland$credits
homeland$invest kernel.org 2
homeland$assets --all
homeland$ls communities
homeland$greetings community1 me --verbose
homeland$offers
homeland$accept 1
homeland$credits
homeland$value dapp1
homeland$dapp1 --verbose --debug
homeland$login
homeland$
homeland$ls
homeland$rate dapp1 ***
homeland$SMS user1
homeland$share dapp1 user1 2
homeland$lend dapp1
homeland$claim dapp1

```

## Architecture

### Concept

Create a World Compensation Ecosystem based on Decentralised Financial Applications.

Implementation: Operation System (including fs, dfs, time-shared applications)

wcsOS – linux based distribution

Layers: 1. Distributed peer-2-peer (P2P) Network (Blockchain based) 1. Distributed File system (dfsWcs) 1. Nodes are Servers 1. Servers 1. run System- and Users-services 1. route User- and System- interactions (transactions) 1. Users are Clients 1. Clients decide to participate or not (mounting/unmounting) as service suppliers in the network 1. Clients interact with other Clients 1. Clients request services from Servers (service suppliers) 1. Via Remote Procedure Call (RPC) returning values in JSON format 1. Clients transfer value-assets to single or multiple-users or services 1. Light-Clients connect and use the network only for short-time (SMS, PPP) 1. Value-assets are represented via Addresses in the Distributed File system 1. Clients and Servers interact via read/write file operations with eachother 1. Servers providing User-services are debted certain agreed amount per-use 1. Servers providing System-services are debted an agreed amount per-use, daily, monthly or yearly on donation basis

## Network

**Topology:** Flower or Tree-of-Life (sacred geometry star 1:N, N:=6) \*  
[https://en.wikipedia.org/wiki/Overlapping\\_circles\\_grid#Modern\\_usage](https://en.wikipedia.org/wiki/Overlapping_circles_grid#Modern_usage)

### Nodes:

Full-nodes: store the complete history of command-blocks (analog to batch-files (a.k.a transactions))  
Light-nodes: store, validate and reconstruct environment from all nodes in local network (or

## Local File-system

### RK1.0

/ - WCS root Ecosystem  
/commands  
/dbin/ - Decentralised System services  
/users/ - connected user addresses {publickey:addresshash:alias:inbox} (analog to /mnt)

### RK1.1

/apps  
/lang - implementation language specific files  
/tools - Utility tools

### RK1.2

/dapp/ - Decentralised User or Third-Party Applications (executable -- analog to /usr/bin)

## Remote (distributed) File-system

/arch/dfs

### RK2

/dapp/DeFi/ - Decentralised Financial Apps

## Realisation

TCP/IP Server :port UNIX's "Everything is a File" -> (name:Address) - Network  
(distributed) File-System

## Support Tools

neo4.js - Graph Database

## Common Use-cases

- transfer value-assets

## Network Startup

```
$wcsStart &  
World Compensation System server (wcsc) running  
Listening on port:280182
```

```
$wcsStatus  
Status: OK
```

## Command, Services and Tools

```
mount - loads/unloads foreign WCS Networks  
ls - show current Users in the Ecosystem  
mv - allocate users in different sub-network  
whoami -  
ps - process  
ping - ping  
telnet - establish connection and echo server  
mailto - use mail-alies to transfer transfer value-assets in the network
```

## Application Notes

```
$whoami  
{0x123456:mstcroix:mstcroix@protonmail.com}
```

```
$pwd  
/
```

```
$ls  
/apps  
/commands  
/dapps  
/services  
/users
```

```
$ls /users  
/users/{0x123456:mstcroix:mstcroix@protonmail.com}  
/users/{0x423456:mstcroix:none}  
/users/{0x223456:anonymous:none}
```

```
/users/{0x723456:mstcroix:mstcroix@protonmail.com}
```

```
$ps
```

```
0 applications running
```

```
$/apps/App1 &
```

```
$ps
```

```
/apps/App1 running. 5 users connected
```

```
50fc328aad939c00fb848432a94943c9 ../arch/README.md 856227b6ba8be28ee97e25f18b276666  
README.md d3777eb628218cf79d50e576d5c95bbd customer.md ca8f6611e7334b5878a412f6908fab36  
platform.md bb9f952448806a9465104b3c5895d5356 wcsOES.md 70a1947487f1741ee64cb109b8cddc82  
whitepaper.md ## 2020 (CC) Creative Common License bb9f952448806a9465104b3c5895d5356  
wcsOES.md
```

```
bc9b4920af19ea249c5e51730a986a9a ../tools/README.md  
f92f6755c1f6c83858630cb4d4c419aa ../networking/README.md  
3582056e21f163e556a92a29f26da4bc ../GLOSSARY.md  
1d4ba0b4f97b65cb239ac157fa453df6 ../dapps/README.md  
ca8f6611e7334b5878a412f6908fab36 ../whitepaper/platform.md  
bb9f952448806a9465104b3c5895d5356 ../whitepaper/wcsOES.md  
d3777eb628218cf79d50e576d5c95bbd ../whitepaper/customer.md  
56617f7fb9a1beca7fc7fb4135a2d9a6 ../whitepaper/README.md  
70a1947487f1741ee64cb109b8cddc82 ../whitepaper/whitepaper.md  
b85f6f905757b8a0d3f75430e13c47ad ../operations/README.md  
4e57eedfde6cb02c52aec8be79b015c9 ../README.md  
94063115eb82858ccfd15ef5a3b21814 ../project/integration.md  
68f05ceb68281268217108fb55876082 ../project/deployment.md  
5cd4aa50a1a9f8d1b46b0b63c9d82e27 ../project/CONTRIBUTING.md  
9c060f1741bc37163838ead55b73c8ab ../project/workproducts.md  
f16bab90fe5bf837c86b04e89f7dbb86 ../lang/README.md  
2eaaf2bbe0e2dae25cbc17345d4ba75a ../lang/c/README.md  
5fe7603d97b3315406ce7c051f273a3e ../arch/dfs/README.md  
b6a84991d4f8957e69ccfd6d3e935e02 ../arch/dfs/dFSwcs/README.md  
50fc328aad939c00fb848432a94943c9 ../arch/README.md  
4ae120d33503361b35768677302e8c75 ../arch/fs/README.md  
1c309bf14fbd49d5afcfac8da0635b5b ../users/README.md  
41990a8f6b22e6e1b72009ca47b6ffc7 ../commands/README.md  
15fd8a06e94ecf5f079451536356171c ../apps/README.md  
a3c6c1e9fbc0dd9e6723f73f7402b08a ../services/wcsServer/README.md  
06ae6a9d35733170f372c50e1e6ed749 ../services/README.md
```