GRIDNET Operating System

~ For The Decentralized Community ~

ℹ️ **LIVE sessions** *All of the mechanics have been all implemented LIVE. You may find videos on YouTube matching the corresponding Tweets at the same time the tweets were created. There are over 10TB of LIVE sessions available on YouTube, growing each and every day since early 2017.*

Table Of Contents

---

1. Introduction

*Welcome dear **cyber-space** sailor, we hope you do find here something for you. We've been **w**orking hard...**driven by passion**.. and remember, should you have questions, feel free to ask!* 🙏

Note:** This document is a work-in progress and expected to be updated pretty often. Its aim is to provide the respected reader with some introductory details regarding* **GRIDNET-OS, while the website (GRIDNET.ORG) awaits a major revamp.*

***In one sentence** - a new 100% decentralized, multipurpose Windows/Linux-like environment available from anywhere without installing a single thing, boarding a super-fast cryptocurrency. Grabbed your attention?*

***Where to find news?** The best places include Twitter (twitter.com/GRIDNETproject) , along with Reddit (https://www.reddit.com/r/gridnetproject/). Reddit is where you'll find longer written reports, along with video-reports. There's also our YouTube channel (YouTube.com/GRIDNETproject) where live programming sessions have been taking place almost 24/7 for the past few years. On Twitter we give updates oftentimes on an hourly basis.*

One of the Video Reports *available here*

Preamble: We are after the truth/freedom and decentralization which shall allow the former two to emanate themselves. To achieve our goals, we aim to deploy the first, global, fully incentive-compatible and openly accessible by anyone Operating System - **GRIDNET OS. The system** will enable for a Linux/Window-like uninterrupted experience despite its decentralized nature. It is powered by a decentralized Virtual Machine **GRIDNET VM**. We have designed an innovative programming language **#GridScript,** with some unique qualities suitable especially for decentralized communities. The language is agile enough to be used both as a programming language and as a user interface through the Decentralized **GRIDNET Terminal** alike.

- **GRIDNET-OS - what's it going to be *good* for?**

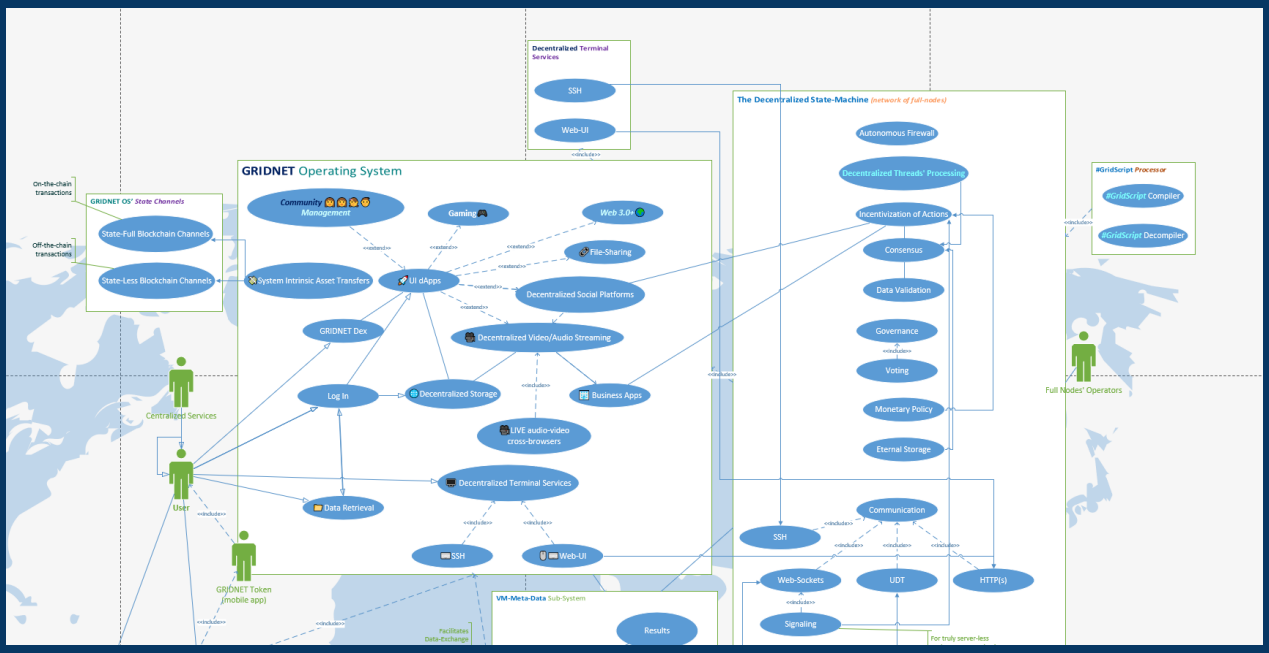


Figure 1: Small fragment of an UML-use case diagram showcasing elements of **GRIDNET**-OS. Entire simplified version is available at the GRIDNET Magazine, with pane/zoom.

GRIDNET-OS is built on top of decentralization.

One thing is certain. It will encourage you to rethink what you perceive as an 'Operating System' and make you realize a plethora of entirely new possibilities now to be available.

When you login into the system you will find it difficult to believe what you see is true

~~ it will be.

~~Logon into the System without installing a single thing. ~~ Directly from within a web-browser, from any device.

~~Be greeted by a beautiful, high-tech, intuitive Windows/mac-OS-like user-interface.

~~ let yourself Discover Decentralized Graphical Applications (GUI dApps) - a concept to be found nowhere else (..)

Come to find that you can drag-and-drop files from your local computer into the File-Manager (running within GRIDNET-OS.. which runs within a web-browser…).. files to be hosted by a decentralized network of computers.

~~ (..) make cryptocurrency yourself by hosting files for others(..)

~~ Stream ULTRA-HD video to your computer, rewarding others with #Crypto (..)

~~ Discover GUI dApps within the system **allowing for real-time end-to-end** encrypted communication and video-streaming.

~~ all without installing a single thing on your computer

~~(..) host decentralized web-pages, ones which cannot be brought down by anyone. Edit them directly through GRIDNET-OS' UI dApps (..)

~~ anyone will be able to deploy new applications through a mixture of JavaScript, WebRTC, WebGL and our innovative #GridScript language..developed through Passion.

With **GRIDNET-OS**? Sky won't be.. the limit..

2. Constitution of The GRIDNET Community

Below we lay-forward the constitution of the **GRIDNET**-community. It consists of just a few, yet of fundamental importance points.

Constitution of the GRIDNET Community

- To constantly promote innovation and progress over backwardness and traditionalism
- Never to introduce trusted entities by design (this includes the **GRIDNET** Development Team), let alone the updates and security updates during infancy of the Project.
- To provide a Sybil-proof incentive for all the parties involved (read on).
- To keep on introducing innovation at a rapid pace. (something Bitcoin failed to do while it still had a chance)
- To never change the initial 'monetary' supply rules - these will be set in stone upon the initial network deployment.

---

3. The Architecture

Below there are depicted some of the key architectural elements comprising the system. From the first glance one might notice employment of technologies such as JavaScript and HTML5, now.. that is curious, isn't it? The mixture allows for some unprecedented possibilities. These will become clearer once we take a look at the core architecture of **GRIDNET**-OS' Decentralized Applications.

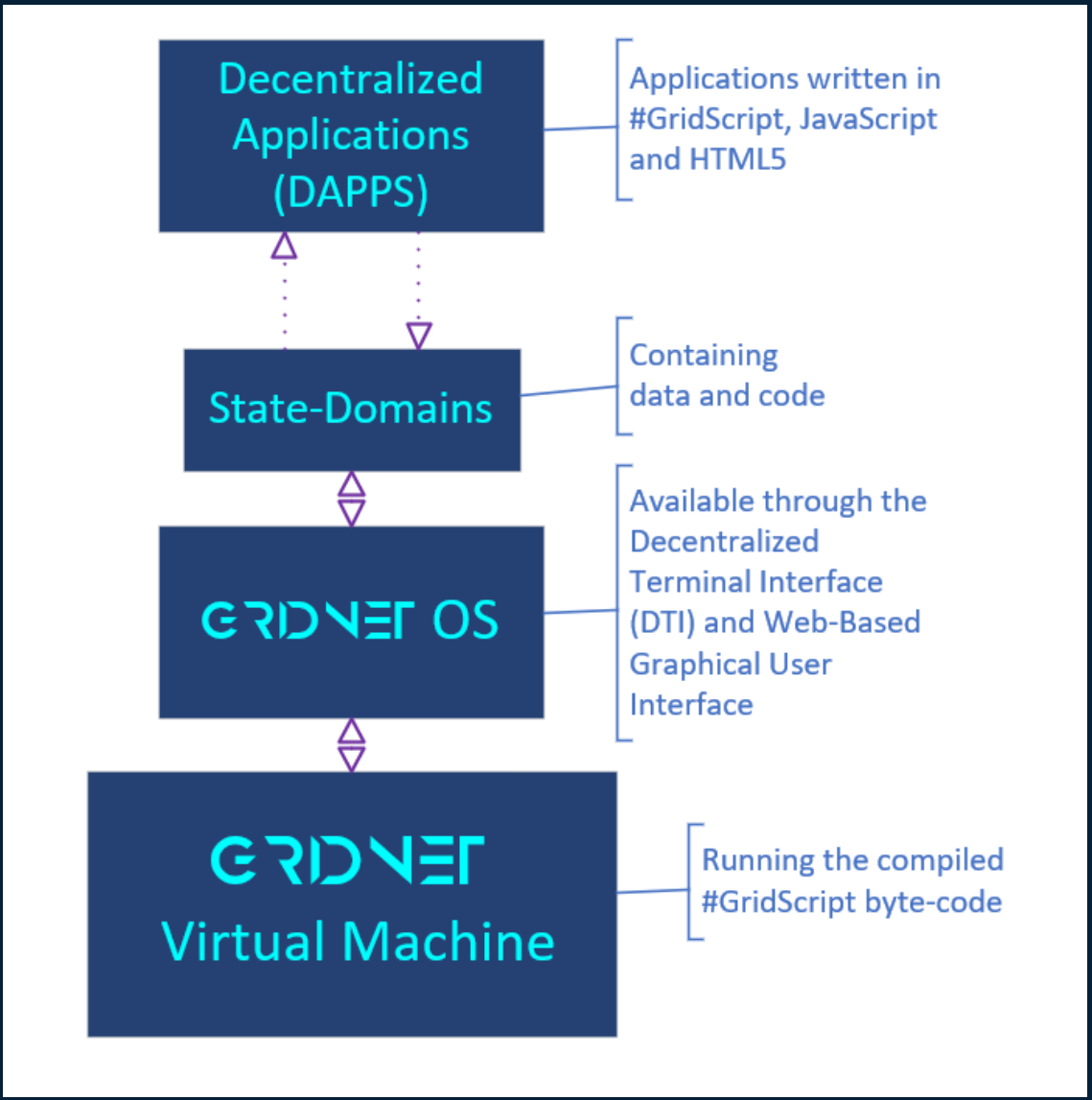


Figure 1: The *overall GRIDNET architecture*

Let us sail forward, shall we..

4. User's identity within the **GRIDNET** Operating System

First off, the system is about anonymity and fairness (in terms of transaction fees, the spread of rewards for data retransmissions etc.). The Wizards 🧙 they have gone to some great lengths to allow for both. The gist from the user's perspective is - you ***may*** want to create and register an Identity Token within the **GRIDNET-OS** decentralized state-machine ***but*** it is not a requirement. Creating an identity token does not compromise your anonymity in any way. However, it make some crazy cool possibilities and features available to you:

- makes it possible to be rewarded for participating in data-exchange (acting as a router)
- when creating an identity token you can choose a 'Friendly Identifier' like .. 'Wizard201', then anyone else would be able to interact with your identity simply by taking that easy to remember identifier. Yes, making a cryptocurrency transfer to you from a **GRIDNET-OS** console would be as simple as

```
$_ send Wizard201 8898
```

Code Snippet 1: The code it takes to send Wizard201 8898 GBUs.

When in command line, once you are ready to commit any kind of operations to the **GRIDNET-OS** decentralized state-machine, use the `commit (ct)` command.

```
$_ ct
```

Code Snippet 2: Use `commit` to request changes to take effect.

Once input, the Virtual machine would ask you a couple of questions like the maximum processing fees you are willing to cover, while automatically calculating and proposing an optimal value.

Note: at this stage, when accessing the VM from a remote terminal a QR Intent would be displayed for it to be scanned by the GRDINEToken mobile app.

- keychains (storage and export of those)
- Identity Tokens
- where data about you is *(not)* stored

5. The *Communication* Subsystem

As you shall come to see, one of the most fascinating aspects of **GRIDNET-OS** is its multi-layered incentivized communication subsystem. Data-exchange is incentivized at all steps through State-Less Blockchain Channels. The communication subsystem employs a variety of protocols including TCP, UDP, UDT, WebSockets and WebRTC. The communication facilities are to be easily accessible for JavaScript and C++ developers through an easily accessible API.

First off, let as begin with a gentle anti-centralization propaganda 🤗

Story: *You install the supposedly state-of-the-art communication app (Signal) which was supposed to provide state-of-the-art anonymity and encryption when all of the sudden the first thing you're shown is a form asking for phone number verification. So they say it's hashed.. used only for peer discovery..sure, alright!*

Question: Since when does cryptography and/or *math* require… phone numbers to work?

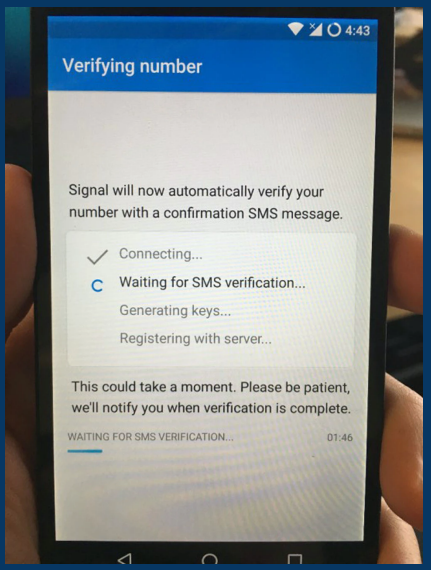
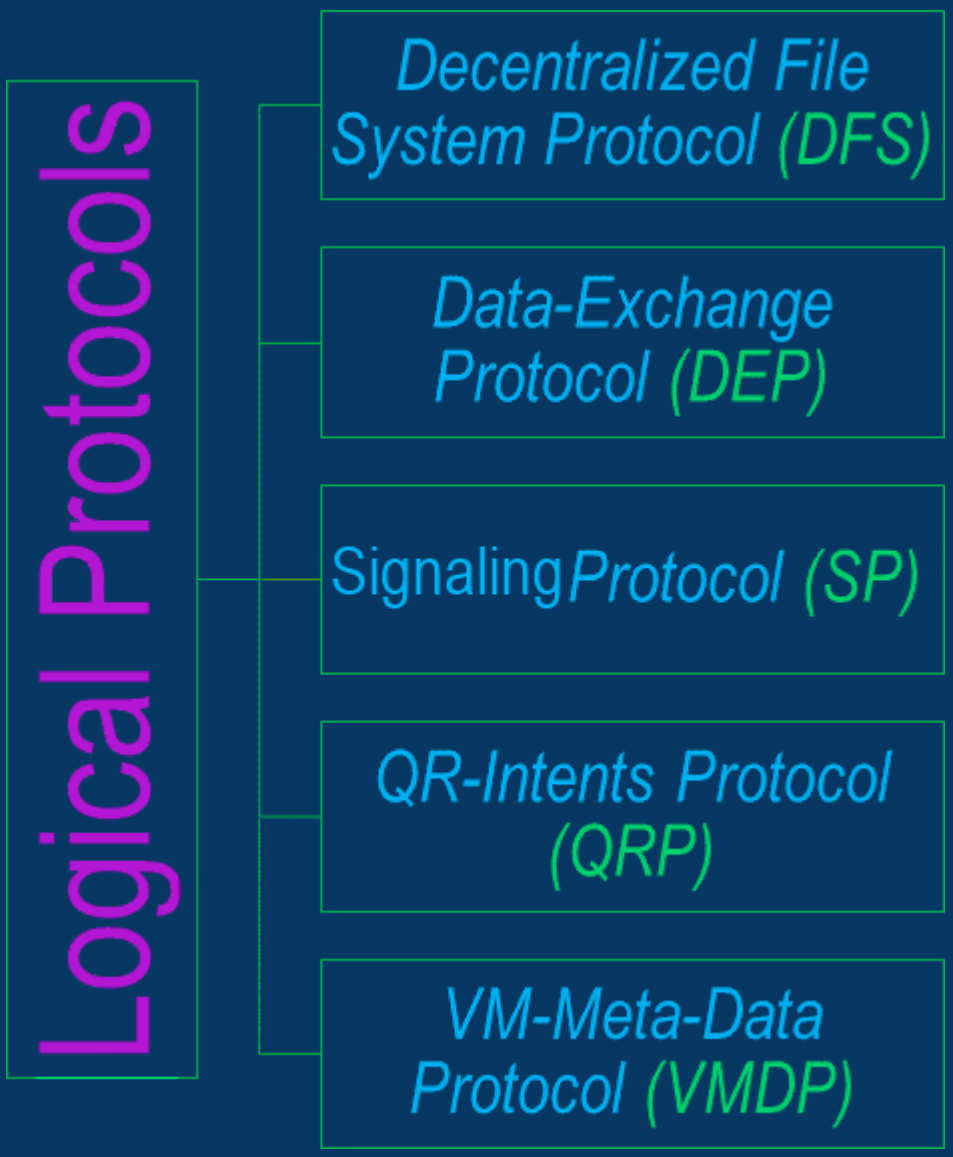


Figure 2: *'Anonymity' at its.. 'best'. Really?*

The Big Question: why let the 'SMS-service company' know that you are an app user in the first place? Do you want anyone to think that you've got something to hide? Why make them want to spy on you? After all it's much more efficient to spy on those who have something to hide than on those not trying to hide anything. Why attach a label to yourself?

Solution: **GRIDNET OS** does not only provide you with end-to-end encryption through open-source-software. It provides an open **API** for data-exchange throughout mobile devices, web-software (running directly within web-browsers) and full-nodes maintaining the network. It not only end-to-end encrypts your text audio and/or video calls but provides an infrastructure for onion-routed, incentivized traffic. In practice, in many cases, you won't need to install anything and use cool-looking software directly from within a web-browser.



Let us now take a short glance at each of ***These*** 🤗

- **The Decentralized File System**

There's a separate section within this document related to this subsystem specifically. There are two subtypes of storage - the Eternal Storage and the Crowdfunded storage. The aim of each is to store data based on the user's requirements and amount of assets the user is willing to spend over time.

ℹ️ **LIVE sessions** *These mechanics have been all implemented LIVE. You may find videos on YouTube matching the corresponding Tweets at the same time the tweets were created.*

- **The Data Exchange *Protocol***

GRIDNET OS is a decentralized system maintained by a network of agents / peers / computers spread throughout the World. A protocol (a set of rules) specifying certain conventions on how these entities talk to each other needs to exist. **GRIDNET OS** is a heterogeneous environment, with multiple kinds of entities communicating in real-time across programming language or platform boundaries.

This has been achieved through porting the same data structures across multiple platforms and employing a very efficient binary encoding (BER). Thanks to it data is not only transmitted very efficiently, especially in terms of binary data, but we achieve a platform agnostic representation of information (Big-Endian / Little-Endian) and extensibility as well.

Despite data-representation, the very same communication logic has been re-implemented across platforms as well (this includes the in-browser **GRIDNET OS's** Web-UI), making it possible for establishment of Onion-Router resilient communication across web-browsers, also web-browsers, full-nodes and mobile devices all speaking simultaneously to each other.

ℹ️ **LIVE sessions** *These mechanics have been all implemented LIVE. You may find videos on YouTube matching the corresponding Tweets at the same time the tweets were created.*

- **The Signaling *Protocol***

Part of the communication subsystem relies on WebRTC, which needs signaling. You may read more details about the process in external sources, which is a very technical matter. What is important is that **GRIDNET OS** employs a signaling protocol of its own, with an interface available from each full-node maintaining the network.

ℹ️ **LIVE sessions** *These mechanics have been all implemented LIVE. You may find videos on YouTube matching the corresponding Tweets at the same time the tweets were created.*

- **The QR-Intents *Protocol***

From our viewpoint, the most important aspect of any system, from the user's perspective is ***usability***. The system wouldn't be used at all, let alone by masses if it wasn't to provide a certain level of satisfaction. Were security and/or privacy concerns more important to users, we wouldn't see platforms such as Facebook flourish in the way they do. Luckily security and anonymity is one of the fundamental requirements from the Wizards'🧙perspective. The **QR-Intents Protocol** enables for computer-vision aided communication across the heterogeneous **GRIDNET OS** environment.

To get a better feeling of how the **QR Intent** protocol cooperates with the rest of the system, let us imagine a sample scenario. Say a ***Multi-Dimensional Token Pool*** (more on these in other sections of this document. These are useful for off-the-chain transactions) is to be generated and registered within the decentralized state-machine.

The Token Pools' underlying secret - the master secret hash key, based on which all of the other dimensions/banks are to be generated needs to remain secret and known only to the user. While, the 'unarmed' version of it successfully makes it into the decentralized VM and is registered with the user's Satet-Domain (account).

Now, there are a couple of ways this could go:

- If the user is a full-node's ***Operator***, he or she can generate a Token Pool directly with an appropriate command providing its properties through commands' parameters. The resulting token-pool would be packed and broadcast throughout the network.

- If the user is accessing **GRIDNET OS** through a remote text based **Terminal** he can execute the same command but this time a QR-code would be displayed directly ***within*** the command line, instructing the mobile app to generate the Token Pool as specified by the user provided criterias. The full-node would be now awaiting the token-pool to be delivered from the mobile app. Note that the ***QR-Intent*** contains all information required for

communication and secure delivery of data requested between the full-node and mobile app.
- When connected through a remote terminal equipped with a graphical user interface, a good looking, animated, **QR-Intent** would be generated by the ***WebUI*** autonomously.

Thus, the **QR-Intent** (that's how we call a QR-code formulated by the **GRIDNET OS** and bound with a certain task) can be generated:
- by a full-node itself
- a full-node can order generation directly from the WebUI
- as a result of executing a command-line instruction
- an UI dApp can request certain actions which will result in a QR-Intent being displayed to the user
- the WebUI itself can generate one for the needs of its own (ex. a request to provide login information so that the WebUI along with its UI dApp could be customized for the specific user).

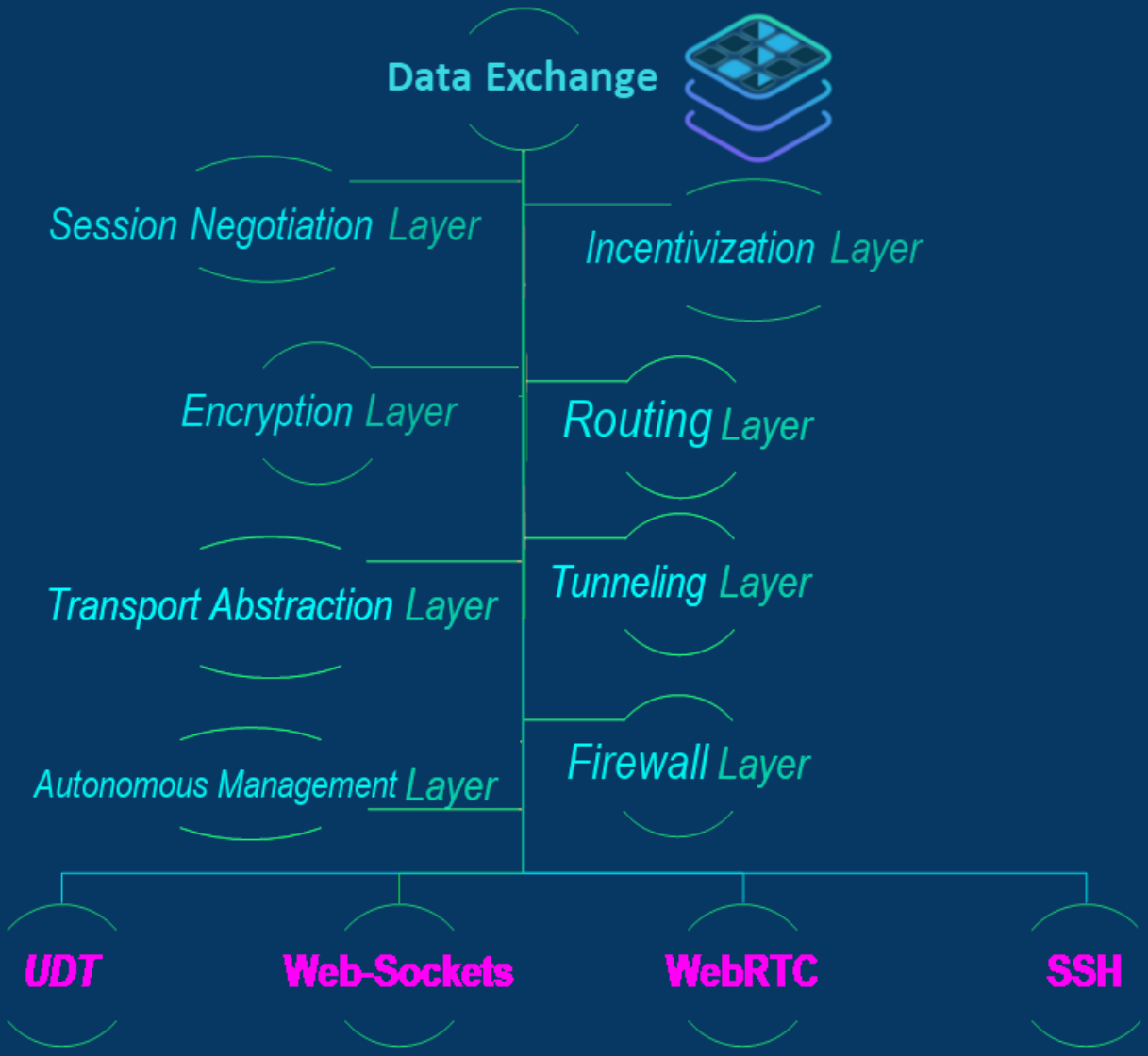
IIn some cases, should the QR-Intent processing be requested by the VM itself, the QR-Intent Protocol takes heavy use of the **VMDP** protocol described below.

ℹ️ **LIVE sessions** *These mechanics have been all implemented LIVE. You may find videos on YouTube matching the corresponding Tweets at the same time the tweets were created.*

- **The VM-Meta-Data *Protocol* (VMDP)**

The decentralized state-machine runs a certain kind of a Turing complete language (#GridScript). Wouldn't you agree that it would be beneficial if other extrinsic entities could understand the results of processing performed by full-nodes i.e. machines running instances of the **GRIDNET OS VM** or maybe even request processing from the decentralized **VM** itself? That's precisely where the **VM-Meta-Data *Protocol* (VMDP)** comes into play.

VMDP describes BER-encoded data-bundles along with rules governing formation and interpretation of these. **VMDP** is a low-level protocol and other protocols may be based upon it (ex. the QR-Intents Protocol).



- ***Autonomous*** **Node Management**

In a typical system one needs to configure IP addresses or perform any additional configuration. *Wizards* 🧙 have introduced lots of automation in these regards. The IP addresses are attempted to be detected automatically (both internal and public addresses if peer appears to be behind NAT). Changes to the underlying network configuration are detected automatically. If for any reason auto-detection fails, the Operator is presented with an inquiry, including a suggested value if available. That's as far configuration goes.

Now, when GRIDNET-OS is Operational, the software employs pristine connection management algorithms. Inactivity is detected (the process which is fully transport-layer agnostic and works across all the underlying protocols)

and when needed, automatic, gentle connection termination procedure is initiated, the other peer is/are notified about a pending disconnection. Should anything go wrong, more aggressive procedures kick-in automatically.

Figure 3: *Military style fall-back mechanisms Twitter URL*

ℹ️ **LIVE sessions** *These mechanics have been all implemented LIVE. You may find videos on YouTube matching the corresponding Tweets at the same time the tweets were created.*

- ***Auto*-Firewall and Logging** *Layer*

The auto-firewall functionality facilitates a possibly separate (if the user already employs a firewall in his system) layer of application and network-level security. Even if an external firewall system is present it is most

likely not aware of the **GRIDNET OS** protocol specifics. By employing state-of-the-art logging and event processing The Wizards🧙 were able to introduce firewall functionality directly into the **GRIDNET OS** Core .

Types of events processed by the internal firewall include (not limited to):

- ❖ the frequency of key-storkes (in case of remote-terminal sessions)
- ❖ actions deemed as invalid (ex. invalid response to a request, no attempt to participate in session negotiation etc.)
- ❖ number of session negotiation attempts
- ❖ number of connections from a single IP address
- ❖ data packet size
- ❖ the depth of Onion-Layers
- ❖ authentication fields required for certain types of Routing-Table entries

Internally almost every action undertaken by the software are attached to events. Events have multiple categories and scope. Events can be associated with network tasks. The Operator can select, during setup, the categories and priorities of events he is interested in. Enabling notifications about lowest-level events is not advised for performance reasons, however if enabled the software reports content of transferred datagrams, the flags set (type of encryption, authentication, endpoint types and much more). Probably the most interesting fact about the logging subsystem is the ability to present the reasoning behind particular ***decisions, which dramatically improves debugging and implementation of*** **GRIDNET OS.**

ℹ️ **LIVE sessions** *These mechanics have been all implemented LIVE. You may find videos on YouTube matching the corresponding Tweets at the same time the tweets were created.*

- **Session Negotiation** *Layer*

Peers participating in communication might have different requirements and/or preferences in regard to encryption and authentication. These differences might be due to hardware differences and/or required security levels. By default all agents employ full end-to-end encryption.

ℹ️ **LIVE sessions** *These mechanics have been all implemented LIVE. You may find videos on YouTube matching the corresponding Tweets at the same time the tweets were created.*

- **Data-Exchange Incentivization** Layer

  GRIDNET OS employs unique data-exchange incentivization mechanics. It is efficient for remunerating both system intrinsic and extrinsic data-exchange. The mechanics are fully transport-layer agnostic.

  Abstract from our pending research paper: ”Decentralized, open-access blockchain systems opened up new, exciting possibilities – all without reliance on trusted third parties. Regardless of the employed consensus protocol, the overall security, decentralization and effectiveness of such systems, largely depend upon properly structured incentives. Indeed, as has been previously spotted by Babaiaff et al. Bitcoin-like systems, oftentimes lack some of these. Specifically, current blockchain-systems fail to incentivize one of their crucial aspects–the underlying data exchange. As we rationalize, proper incentivization of that layer could lead to lower transactions’ confirmation-times, improved finalization guarantees and at the same time to discouragement of malicious behaviors such as block-withholding attacks. Indeed, incentivization of the data-exchange layer allows the system to remain operational when all agents, including routing nodes, are assumed to be rational. In this work, while focusing on the problem of sybil-proof data exchange, we revisit previous approaches, showcasing their shortcomings and lay forward the first information exchange framework; with integrated routing and reward-function mechanics, provably secure in thwarting Sybil-nodes in 1-connected or eclipsed networks. The framework neither requires nor assumes any kind of constraints in regard to the network’s topology (i.e. the network is modeled as a random-connected graph). The proposal, while being storage and transmission efficient is suitable for rewarding not only consensus-related datagrams (both data-blocks and transactions) but consensus-extrinsic information as well, thus facilitating an universal sybil-proof data-exchange apparatus, provably valid under the assumption of existence of a data store whose property of non-malleability emerges as time approaches infinity. Our research was conducted under two scenarios - with a round leader known and unknown in advance of each transactional round. ”

- **Encryption and Authentication** Layer

  To get a feeling of the amount of work involved in development of the encryption and authentication layer one might skim through Tweets made available each day by *The Wizards* 🧙:
  The below URLs perform search on Twitter using the below keywords:
  - AEAD
  - ECIES
  - ONION
  - PROTOCOL

ℹ️ **LIVE sessions** *These mechanics have been all implemented LIVE. You may find videos on YouTube matching the corresponding Tweets at the same time the tweets were created.*

All in all, the underlying communication layer employs Diffie Hellman Secret establishment over the elliptic curve X25519. One way hash function SHA3 is used for key exponentiation. Once the shared secret is established, ChaCha20 is being used as a symmetric cypher (its strength is comparable with AES while much faster when hardware acceleration is not available). Ephemeral key-pairs are being used whenever efficient. When encrypting to static public keys - a pseudo-random IV is always generated anew. No information, including meta-data regarding the communicating parties is ever stored on any device within a non-volatile storage. Poly1305 is used as message authentication code when applicable.The system supports explicit cryptographic signatures at multiple levels (i.e. datagram level and higher level such as the level of 'Transactions' etc.). Signatures are used seldomly to allow for full-forward secrecy whenever viable.What does it mean? If we can achieve state-of-the art encryption and authentication levels without making it possible for third parties to associate the communicating parties with datagrams we do not unnecessarily employ signatures which would prove in the eyes of third parties that communication with certain entities in a given moment in time indeed took place. Besides, ECC signatures occupy space and if can be omitted that's a good thing.

To give you a more technical example, one could design a communication protocol where each datagram conveys a cryptographic signature. Each datagram aram would carry 64 bytes output by Ed25519 when payload fed into it. There would be a couple of problems with this approach:

- storage efficiency (additional 64 bytes alongside each datagram)
- transmission efficiency (bandwidth utilization increases)
- computational efficiency (ECC signatures might not be the most computationally intensive operation yet still, computing these at high throughput without hardware acceleration would be)
- and the probably least imminent thing: if one records the conversation he would be able to prove in the eyes of a third party, knowing the public key that each and every datagram was produced by it.

So, instead for now in short ~ *The Wizards* 🧙they have made sure that, THAT is not the case🙏

Thus, one may infer two facts:

- **GRIDNET OS** employs an ECIES scheme when encrypting to a public key
- when session negotiation is enabled, the use of ECIES is scarce to save on elliptic curve handshakes and the negotiated shared secret is used directly.
- AEAD containers are used both during session-based and session-less communication to provide authentication and integrity checks.

The above mentioned mechanics are used by the mobile app, full-nodes and **GRIDNET OS WebUI** at all times (yes it has been all implemented in JavaScript as well. All the custom crypto operandi, including BER decoding/encoding and ECC crypto have been ported from C++ to JavaScript as well).

- **Transport Abstraction** Layer

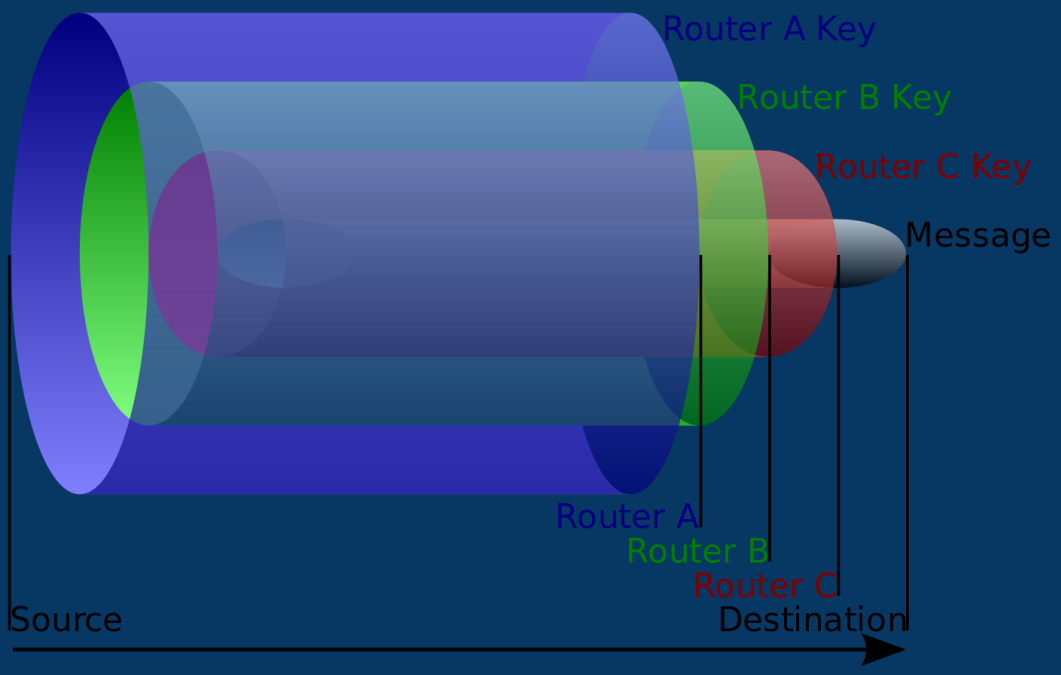


Figure 3: *Datagram encapsulation within an 'Onion-Routed' connection.*

Onion Routed connections are supported by the GRIDNEToken mobile app, full-nodes and even software running directly within a web-browser (implemented in JavaScript as well, yeah we know, it's crazy..).

LIVE sessions *These mechanics have been all implemented LIVE. You may find videos on YouTube matching the corresponding Tweets at the same time the tweets were created.*

Onion Routing is used by default by all peers.The system supports incentivization of each intermediary (each router) on a per-byte basis through State-Less Blockchain Channels.

6. GRIDNET-OS. Decentralized Applications (a technological-perspective)

- ***An*** Overview

In the below infographics you might notice that there are much more elements involved than in a traditional decentralized application.

Figure 2: Decentralized Applications Evolved

The positional relationship between wheels reflects quite accurately their real-life in-system interactions. GRIDNET-VM is the main wheel. It runs #GridScript. Execution of arbitrary decentralized code implemented within the latter can be invoked through JavaScript. Through JavaScript we have a plethora of other Web-Technologies available.

The mixture of all these allows for some unprecedented possibilities.

The middlemen functionality between JavaScript and #GridScript is provided through what we call the **GRIDNET OS** JavaScript Context. It is a JavaScript library implemented to hide away the specifics of decentralized environments from third-party application developers. The context takes care of tasks such as maintaining connection with random full-nodes or providing an easy to use subsystem for creating UI-windows and controls including buttons and text-fields. More information regarding the **GRIDNET OS**' JavaScript context will be provided below.

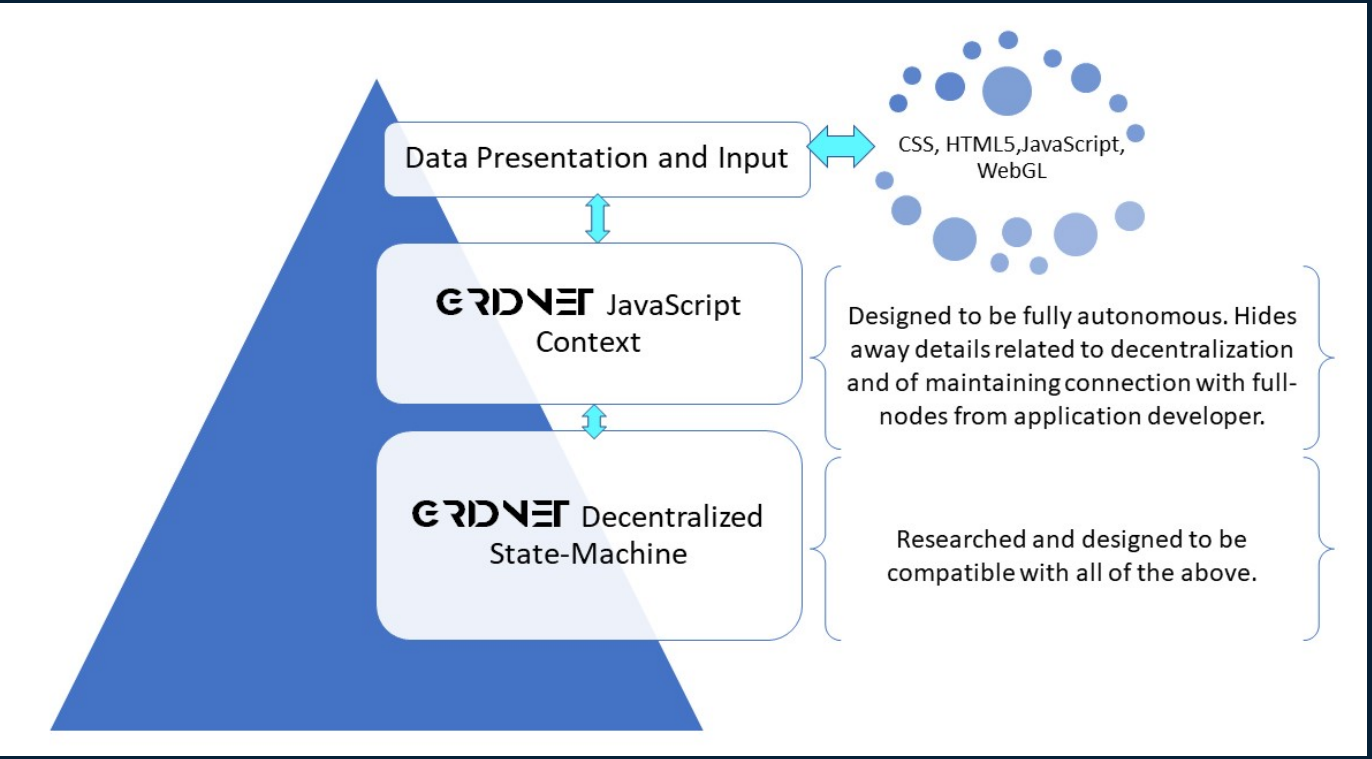


Figure 3: Architecture of GRDINET-OS Decentralized Applications

Reader: "*Hold on hooold on.. waaaait a sec... I like the idea! I like the architecture, still..wouldn't one be able to replace the GRIDNET-VM with Ethereum's VM? Ethereum has a large user-base. Couldn't we just use it as the decentralized state machine for all the fancy stuff and move on?*"

Short answer: No.

Same as we cannot use Bitcoin for what Ethereum does. Even though both are 'decentralized state-machines'. There would be too much to reinvent to keep the system under the same name and backwards compatible, - and as we all know backwards compatibility is one of the top requirements when one wants to maintain true decentralization.

(..) 'under same name'? Or under 'same brand' shall we rather say?

We all know that in this 'business' brand and trust are almost all that matters.

The only reason for why Bitcoin is still around and for why it hasn't been replaced entirely by newer technologies is the people's lack of understanding of the technology itself. The price for its inefficiency is rather huge (in terms of power consumption vs allowed transaction throughput). Yes, even if we require top-security and decentralization, even if we know our stuff, and believe that Proof-of-Work is always the way to go, if we are about true decentralization governed solely by the laws of physics, then stil - there have been more efficient approaches for years,- including the Bitcoin-NG protocol, a variation of which we employ in our Project.

Now, in order to allow for all the elements, depicted in the above infographics, to interact with each other, we had to design a programming language and thus its underlying State-Machine to allow for ad-hoc spawning of a SandBox VM; within which, user actions would be reflected live and which could be then 'committed' to the decentralized state-machine once user is satisfied with the results and ready to do so .

Let us imagine a sample scenario. The user logs-in through.. SSH. We want him/her to be able to create files/directories using Linux/Dos/Windows-like syntax he/she might be already well familiar with. Or we simply want him to be able to do that easily.

In Ethereum there's no concept of files and directories to start with. There is no concept of a terminal interface available remotely to the user. Ethereum does not allow for ad-hoc formulation of end-user commands which could be executed and then committed to the network.

It simply would not work. We could go a long way describing why Ethereum would not be suitable. Still, there are so many exciting things to describe related to the Project of our own...

To make everything possible and easily accessible by end users we have gone to some great lengths, including custom elliptic-curve-based cryptography.

Focusing on some further, concrete rationalization, - the system had to allow for autonomous formulation of instructions which would reflect user-interactions performed either through the Remote Shell or through the Graphical User Interface. Finally, there were a plethora of additional problems such as allowing users to authenticate all the actions performed within the Sandbox environment and for all of these to be verifiably presented to the rest of the network.

Further, the interactions between the decentralized VM and JavaScript required a communicational facility along with an API to be available for third party developers willing to create UI dApps. Such a middlemen facility would need to be easy to use for both the end-user and the programmer writing applications. Plus lots and lots of other problems.

GRIDNET OS provides effective, high-tech solutions to each and everyone one these. Some of the intermittent effects you can already judge by yourself.

GRIDNET OS - Decentralized Applications Evolved

○ *Multi-Tasking* and Application *Boundaries*

In modern, yet 'traditional' already, 'operating systems', one of the crucial aspects is allowing for 'multi-tasking'. In short, it is about allowing a single user to run multiple applications at the same time. We all are well familiar with the ability of listening to music while surfing the web on Windows or Linux.

Reader: "ok..ok.. hold on..wait a sec..you said **GRIDNET**-**OS** runs on the decentralized consensus.. meaning it takes use of the consensus when required.. other than that.. each full-node is able to deploy a remote 'shell', one with which the user can interact with in the same way as with a local Linux or Windows command line... Be it through SSH or through the in-browser UI-dApp 'Terminal' dApp.. now…. where does the multi-tasking come into play?"

As you might know a typical operating system ensures boundaries across applications. **GRIDNET**-**OS** does the same. That takes place at multiple levels, both at the full-node software ***and*** within the web-browser.

At full-node, each decentralized application, written in **#GridScript** has its own address space. Applications can reserve memory but they cannot access memory (read/write/execute rights disabled) outside of their address-space boundaries. Applications can access memory assigned to them initially by the **OS**, and the memory which they allocate. Recall that each memory reservation cosumes Gas/Erg.

Reader: 'ok, interesting, I get it.. The full-node is your realm. It was implemented by you. What about the web-browsers? You said that **GRIDNET**-OS decentralized applications employ JavaScript; That those fancy decentralized UI applications of yours are written in it. Partially at least. That 3rd party developers are to implement decentralized applications in your eco-system as a mixture of JavaScript and that '#GridScript' of yours, that CSS together with HTML5 could be used to style the applications. How about boundaries over there? I think web-browsers are not ready for ***That*** kind of thing.. How do you go about that? How do you ensure a CSS style of a single application does not accidentally affect the other? After all these are all CSS renders within a web-page, right?"

Ever heard of ***Shadow***-**DOM**?🧐

Each **GRIDNET**-**OS** UI application is stored as a bundle of JavaScript and #GridScript code, but also encapsulating all of its looks, resources (icons, images

etc) and other assets. Such application bundles will be available for download directly from the Decentralized State-Machine through the *Decentralized Applications Manager (DAM)*. Users will be free to store/cache these within their local-drives as well.

Once the application is installed and launched, its UI is rendered through Shadow-DOM - a technology supported by most modern web-browsers.

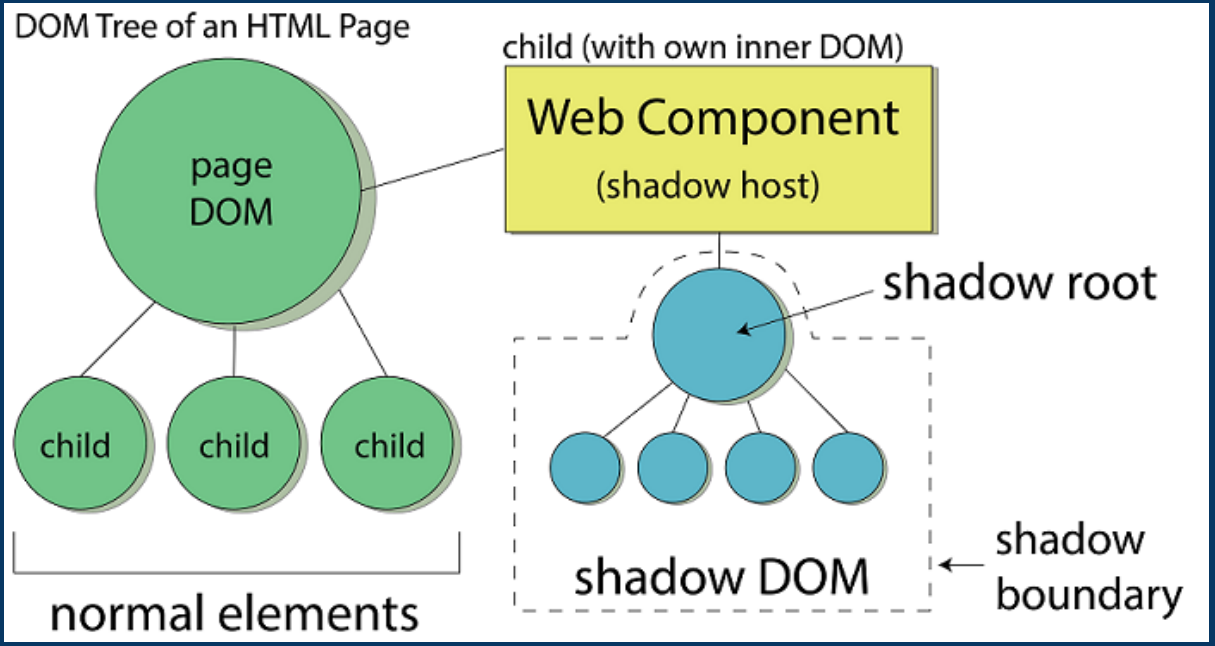


Figure 4: Architecture of **GRDINET-OS** Decentralized Applications

As a result, the UI of each application can be said to execute within a realm of its own. With its UI-components (CSS styles etc.) unable to accidentally affect the rest of the system. Feel free to read more about the Shadow-DOM technology over here.

Reader: "Interesting.. ok .. ok.. how about networking..in traditional…"

Got your point! Let us take it from there..

A traditional operating system is involved in much more than assuring memory boundaries across applications and management of 'windows'. Actually,it is also the case with **GRIDNET-OS**.

Let us give you an example - a decentralized **GRIDNET-OS** UI application does look-up a file from the Two-Tier data-storage subsystem. What does the interaction between the UI App and the OS look like? It is based on events.**That is right.** Application developers may expect a fully event-driven development.

The **GRIDNET-OS** (precisely the **GRIDNET-OS**' JavaScript context) will be taking care of keeping track of which app performed which requests and once the response arrives from the network - it is going to be *routed* to the particular JavaScript-based application / Window, the one which made the request. The application developer would just need to subscribe for the desired types of events and notifications. In most cases it would be enough to provide an appropriate callback function to the DataStorage API.

Reader: "Actually I was paying attention to what you've said. I've looked up the Shadow-DOM technology. You've seemed to imply that Shadow-DOM does protect against an ***accidental*** modification of elements. How about an intentional modification? How about a case in which a malicious dApp executes some evil JavaScript ? Nothing prevents that right? Shadow-DOM might protect against accidental modification i.e. give a sandbox of its own to each UI dApp but it does not prevent its JavaScript to iterate over the entire DOM if it wills. How about that? "

And you are right yet again! Let us explain.

We do take into account such a possibility. We are prepared for 'malicious' applications. In the end it is the same as with any other operating system. You need to trust what you run, at least to some extent.But Lo and Behold! In **GRIDNET-OS** anything that happens within the web-browser will not, under any circumstances affect the decentralized state-machine UNLESS it is authenticated by the user through the mobile **GRIDNET'oken** mobile app. Thus, anything the evil application does, anything at all, concerning the decentralized consensus, would not take effect on the network, UNLESS it gets confirmed by the user. The user always has a chance to verify pending actions before commiting. (list of pending actions visible when mousing over the ***Magic Button*** at any moment).

- The P2P ***Swarms*** API

Note: development of the SWARMS API like everything else, has been recorded LIVE on YouTube.

Preface

O.K. Listen up.. here's the thing. We need to communicate. All of us do. We've got mobiles, we've got Facebook's Messenger available etc. etc. ***Right***?

Reader: *'Right it is! I like what I use! Hold on! ~Waaait a sec, what's wrong with these?'*

Well, for starters, the communication might ***not*** be encrypted. Even if it is then, then maybe just maybe maybe the service provider has access (he does) to the encryption keys. Anyway, how do you know? *How do we know? End-to-end* encrypted they say? Maybe so. Maybe not.

Reader: *'What are you saying, the platform of my choice is an Open-Source project!'*

It is GOOD for you that you've headed Towards Open Source, verifiable solutions already. Just make sure whether it's not the case that creators of the app put themselves in a trusted, centralized position ensuring that all the data and/or meta-data flows through their servers and then for instance allowing for associating communication with your identity through your IP address and or the mobile phone number they do collect. Rings a bell?

Indeed, maybe The Service Provider (Lo and Behold!) stores meta-data related to communication? Kindly do note that staying anonymous and ensuring privacy among your colleagues is one thing ~ ensuring privacy against The Provider is a **totally** different thing.

The 'typical' IMs (including Facebook's Messenger, Skype and applications including supposedly privacy concerned users all make data flow through their servers).

Telegarm's FAQ:

Q: Can I run Telegram using my own server?

Our architecture does not support federation yet. Telegram is a unified cloud service, so creating forks where two users might end up on two different Telegram clouds is unacceptable. To enable you to run your own Telegram server while retaining both speed and security is a task in itself. At the moment, we are undecided on whether or not Telegram should go in this direction.

Oh well. Indeed, communication providers (i.e. including IMs) in most countries (yours ***included***) are **forced to** maintain records of the so called

‘meta-data’:

https://en.wikipedia.org/wiki/Data_retention (*yeah it’s true sorry* 😢)

The result is,- even if the conversation itself is not recorded (which rarely is the case), The Provider needs to be able to say that the communication with certain parties at least took place. Ever wondered why supposedly pro-privacy projects, including those supposed to ensure end-to-end encryption collect and verify phone numbers? Now, ***Fast Forward****....* ***The real question?*** *Why pay the price of privacy when communicating peers can manage just fine by* ***themselves?*** *Why do we even need big companies delivering services if we can manage perfectly fine on our own? Why let* ***Them*** *collect what we do and where we go?!*

Why don’t we just Get Rid of The Provider?!

They might want you to like 'em’ but the facts are there.

~Still, They need to obey. *That’s how they Play.*~ 😢

Reader: *’Get rid of the provider? But someone needs to provide, right? I want someone to provide for me! What now...’*

Calm down calm down, we shall just provide among ourselves and for ourselves. Of course, in the case of decentralized solutions, there are some inherent tradeoffs and troubles like problems with synchronization among a large number of simultaneously connected P2P peers when delivering the same data-stream to hundreds of people etc. etc. But here it’s an open area of research, we shall be pushing on ***Forward****. Decentralized solutions can already do A LOT and probably already do meet all you need.*

Reader: *’I will ask you straight. For chatting with my peers for doing a conference among ‘em would a decentralized solution meet my expectations?’*

YES. The **GRIDNET-OS** *embedded eMeeting UI dApp allows for decentralized communication including* ***LIVE*** *audio and video exchange. It has actually been battle-proven for remote classes during the Covid-19 Pandemic at the Poznan University of Technology.*

The Actual Thing

The GRIDNET-OS' ***Swarms*** API is a JavaScript API built upon WebRTC.

Reader: *'web .. web.. web-what? I'm not a developer!'*

Got you covered! WebRTC is a technology made available in recent versions of modern Web-Browsers allowing for direct communication between these. Direct? Well ~ **almost**.

While the actual data-exchange *might not* (more on that in a sec) be coming through any particular centralized 'server', the technology does indeed require aid from an external entity, or from an external data-exchange channel during the connection-formation process. Which is Good, as long as, - you can trust the latter. Can you?

The process is called ***'signaling'.*** **In most cases, the process succeeds and connection is established.** Still, in some cases (NATs), constant help from one of the external servers (TURN) is needed as it needs to constantly be passing data on through the communicating peers (you can read more about it here).

Now, as you might have noticed, there are a couple of problems:

- In most cases one **needs** a signaling server/service. Unless one wants Google to provide signaling and collect meta-data.
- Then, in *some* cases one does need a TURN server (the one passing data through and ***nobody*** is going to deliver one for free! You can read more about it here)
- While WebRTC was supposed to be ***easy*** (thanks to the overall black-box attitude towards data passed on for secure connection's establishment and maintenance,in theory requiring developers to just pass some data through to the other peer, without understanding what it is). ***The practice shows*** that the WebRTC API is very cumbersome, it changes ***very*** often and even (Lo and Behold!) Master Google himself encourages the use of high-level compatibility libraries. ***Oh well.***

- Developers still need to research and implement a suitable application data-exchange and signaling protocol of their own (nothing wrong here! Still, somebody could give a helpful hand, right? It wouldn't be bad would it? Nah, it would not 😀)

Yet again, The Team comes in rescue 

The Swarms API provides an easy to use facility for establishing *and* maintaining peer-to-peer data-exchange including multiple parties. The entire communication process, including signaling, is 100% decentralized, end-to-end encrypted and incentivized through cryptocurrency. The mechanics are suitable for covering **LIVE** audio and video streaming but also are perfectly fine for raw data. ***Participants only need a web-browser!*** Indeed, from the developer's perspective, all functionality is delivered through an easy to use JavaScript API, directly from the 100% Open Source **GRIDNET**-OS' JavaScript context.

In Short, here's how it goes:

- ***full-nodes*** act as turn-servers when needed and are paid through state-less blockchain channels, ~ when needed.
- ***100% decentralized*** there are no servers at all. GRIDNET-**OS**' full-nodes take on the burden of being signaling and or / TURN servers and are selected ***at random autonomously when needed. They do not record any kind of data. 100% open source you can check for yourself.***
- ***easy-to-use -*** the mechanics are fully autonomous requiring JavaScript developers to provide an ID of the Swarm they are willing to join or create and to subscribe for events and be notified of incoming data and or video/audio streams.
- ***100% end-to-end encrypted*** (true for audio / video/ data ~ the entire triplett **covered**).

Everything else happens in the **background** 💋

The mechanics are already being tested during remote classes at the Poznan University of Technology and for communication among GRIDNET-OS' Team Members (note the eMeeting UI dApp described later)

7. The Decentralized Consensus *Mechanics*

Figure 5: Decentralized network maintains the GRIDNET-VM

As we have already stated, we are after true decentralization governed by the laws of physics.

Thus, full-nodes maintaining the system operate an improved variance of the Proof-of-Work-based Bitcoin-NG consensus protocol. This allows for transaction throughput limited solely by network propagation times and computational power of agents maintaining the decentralized state-machine. While the operation of the **GRIDNET**-**OS**'s **VM**, in terms of processing of transactions and data-storage might be similar to the Etheruem's virtual machine, the latter was implemented to advance upon Bitcoin. Back in Bitcoin times people were happy since they could make an anonymous value transfer from point A to point B. Ethereum came along and changed all that. Now people could implement any kind of 'Smart-Contracts', meaning small programs incorporating any kind of Turing-complete programs or sets of instructions, running at full redundancy on each of the agents (computers) maintaining the Ethreum's decentralized State-Machine.

Was that so different from Bitcoin? Actually no,- it was not.

Bitcoin itself employed a programming language of its own to carry out even the most simplistic ‘transactions’. Thing is lots of work was needed and a new programming language to make the concept of truly decentralized computational machines a usable reality.

Ethereum-guys did just that and they did a wonderful job.

Thing is, times change. Technology evolves. Peoples’ awareness does change. We have implemented **GRIDNET**-**OS**, from the ground-up, while analyzing architectural and technical rationale behind each and every ‘moving part’ of many of the previous major ‘blockchain’ projects to learn upon them and to find new ways to advance upon..and it became our obsession.

The full-node software is equipped with autonomous intelligence allowing for detection and mitigation of Bitcoin-NG’s consensus related problems. Data structures which we call ‘Key blocks’ are the ones containing the actual proof-of-work. These are released every few minutes. Now, once the round leader - the one elected through the Proof-of-Work lottery, broadcasts a key-block, it can go ahead and proceed with broadcasting additional data-blocks, containing transactions. That is one of the key differences between the traditional Bitcoin consensus. Why waste so much power to elect a leader just to allow him to provide a single set of transactions? That is simply irrationale.

Now, the number of data-blocks following a key-block is virtually unlimited. These can be broadcast freely until another leader gets elected through PoW. The new leader gets a share from transactions fees received by the previous leader. The ‘transactions’ are collected from users and contain authenticated data-structures, instructing for a change of one or more of the decentralized state-machine’s variables. Of course this involves a cryptographic signature. The way a transaction formulation is carried out and the way of providing the just mentioned signature are very innovative and unique here to meet the requirements laid upon by the final decentralized operating system-experience and possibilities.

There are some technical and security related difficulties to the Bitcoin-NG-like consensus mechanism. We have addressed all of which. If for instance the round leader decides to broadcast two data-blocks, at the same blockchain-height and if any other node notices the fact, the malicious node would lose all reward, while the one who reported the case would be rewarded instead. We have actually improved upon the Bitcoin-NG consensus protocol by a lot, lowering susceptibility to certain kinds of attacks.

The Core has been tested continuously and improved upon for the last couple of years, with statistics made available live. As for the consensus mechanics we are confident we have prepared the system for every possible scenario.

8. Accessibility and New Authentication Methods

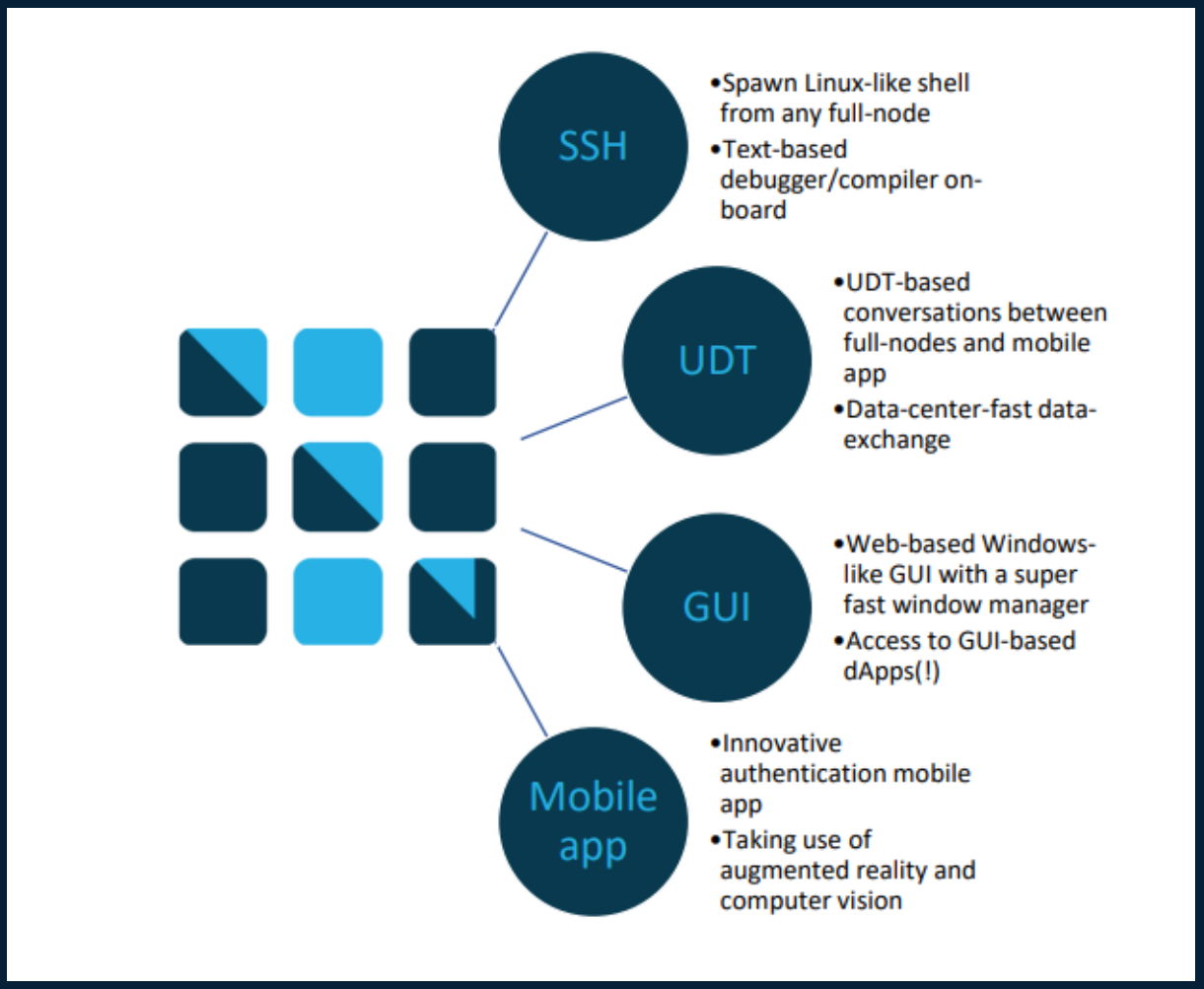


Figure 6: *Access methods to the GRIDNET-OS*

Somewhere along the way we came to see that although Ethereum might have extended possibilities made available by decentralized state machines. Nonone knows how to use it. People recall it as that 'smart'/programmers thing. We took the idea Etherum strived to achieve and reinvented everything anew. Big Time.

The major new concepts that required reinventing included:

- New Cryptography - our language requires operations to be signed and encrypted with the same-public key. Something like that was not available thus we've researched it and implemented it as a contribution to the Botan crypto library.

- New programming language - we required a language which could be used easily both as a programming language and a terminal-command-line tool. The language and the VM which runs it would need to allow for ad-hac operations while the commit and authentication are provided at the very end.
- New decentralized Virtual Machine - we required the State-machine to be accessible by a variety of means including the ability to spawn GUI/command line terminals to users.
- New authentication methods - alright someone logs into the system, through a fancy Graphical-User-Interface, performs some operations, but what then? How do we perform authentication? How do we prove in the eyes of the full node that the guy creating new directories is the owner of a specific State-Domain? (for info about state-domains read some of our research papers). We couldn't simply provide the full-node with a private-key right? We would need to trust it in the first place. Read on for what we've invented ^^
- Decentralized File System - users need to store their files somewhere right?
- **Many many more**

The system will be accessible by a variety of means. The most user friendly way would be to access **GRIDNET-OS** through a web-browser. Once a user visits the website, one hosted by us or anyone else; a connection to any randomly chosen full-node will be made and the user will be presented with a rendered graphical user interface directly within the web-browser. Now, the GUI will have a typical Windows-like user interface. Will be the first to allow graphical decentralized applications.

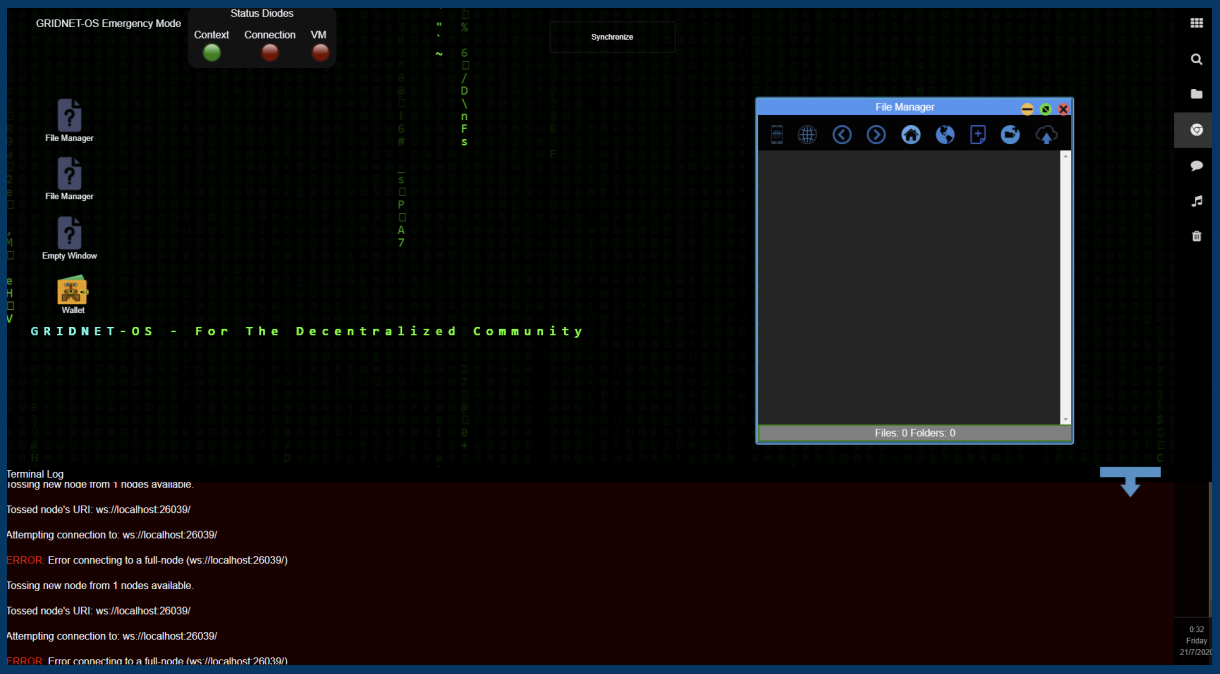


Figure 7: *The Web-based Window-Manager*

Take a look at some of our tweets-showcasing the Window-manager (work-in progress). You might also want to take a pick at high-quality YouTube video here and here

And yes.. in **GRIDNET-OS** you'll be able to launch decentralized applications (dApps) both graphical and Terminal-based with a .. double click. That's right.

For a sample graphical dApp you might want to take a look at Walle(t), the wallet application which runs directly within **GRIDNET-OS**'s WIndow Manager.. which runs… within a Web-Browser.. that is right. Have I said that we were reinventing everything from the very groundup?

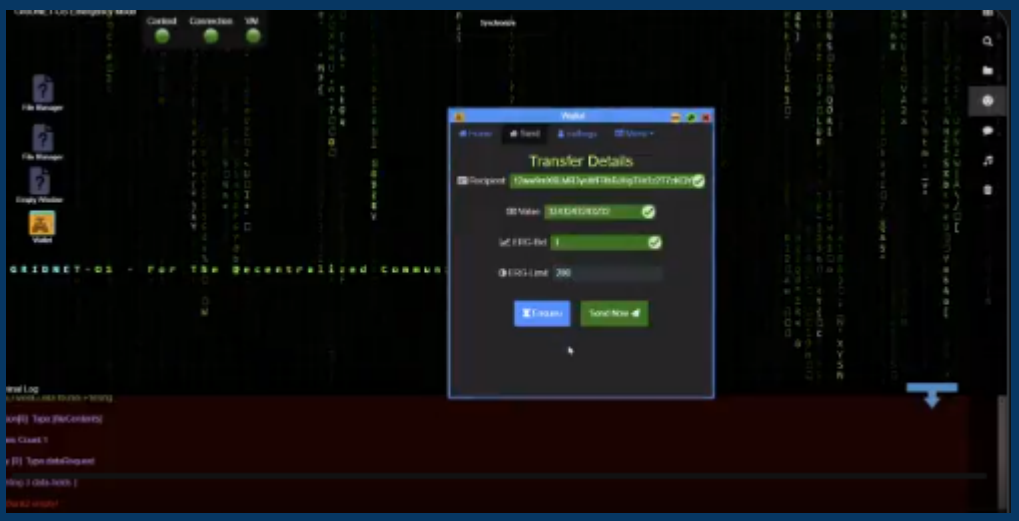


Figure 8: Wallet UI-dApp within the Web-Based Window Manager

Twitter Video: https://twitter.com/i/status/1296914861398667265

The system will be able to handle and kind of Turing complete applications both console-based and those providing user access through a Web-Browser.

In order to allow for these kinds of interactions we have gone to great lengths. From

implementing our own Elliptic-Curve cryptography to inventing and implementing innovative authentication methods based on computer vision. Video-Report regarding this functionality available here.

9. #JavaScript Context

The Graphical User interface will be implemented entirely in the latest version of JavaScript (ECMA6) and taking use of a library which we have nicknamed as a **GRIDNET-OS** JavaScript context. The context will allow third party-developers to implement web-application taking use of the decentralized state machine, one facilitating our environment, effectively creating graphical decentralized applications. Such dApps will be taking use of the full-power of the user's computer - be it CPU or GPU and utilize the decentralized **GRIDNET-OS'VM** only when needed. The applications will be stored within the decentralized storage, so will remain always accessible to the user from anywhere.

Accessibility

All that will ever be needed to access the system will be a web-browser making **GRIDNET-OS** accessible from any device capable of running it. Thanks to our unprecedented authentication methods (read on) users will be able to access the system from any kind of untrusted device securely, without the need to install anything. The only device that would be assumed as trusted (and only from the particular user's perspective) would be his/hers mobile device.

Consensus Mechanics

Most of us are well familiar with limitations imposed by centralized institutions and / or systems. Here, the overall system is secured and guarded solely by decentralized algorithms and decisions made by the majority - with everything laid upon and secured by the laws of physics alone. There will not be any kind of master nodes, just pure physics and equality. We have designed the system to be extendable and capable of handling thousands, or millions of simultaneous users, factories, power systems and regular users all alike and at the same time, simultaneously.

GRIDNET employs a unique Proof-of-Work memory-bound algorithm to choose the leader of each 'round', but then, GPU-processing power is not wasted during processing and validation of data-bundles usually deemed as 'transactions' as these updates are done through 'data-blocks' instead of 'key-blocks', where only the latter contain Proof-of-Work.

The underlying consensus algorithm is secured by a memory-bound Proof-of-Work, yet, the system-mechanics allow for throughput of 'transactions' - i.e. updates to the decentralized data-store to be capped solely by the CPU performance of full-nodes (voluntary computers maintaining the network) and network propagation times - thanks to an unprecedented and innovative relationship between blockchain-blocks (similar to Bitcoin-NG yet different); - thus far exceeding the processing capabilities of Bitcoin-like systems by orders of magnitude. We have implemented everything meticulously. We have accounted for all the related details and nuances such as detection of data-blocks signed by the same leader (owned of the current key-block as per the Bitcoin-NG algorithm) and implementation of proper penalty measures.

In **GRIDNET-OS** everything is and shall be built upon properly validated protocols (TLA+) and science. In here, we have never practiced the 'easy run for the money' methodology- and we never will.

10. Data Storage

~The Gist: GRIDNET-OS employs a Two Tier storage subsystem. The first Tier consists of a typical storage, with the property of non-malleable as time approaches infinity. Here, data is stored directly within the decentralized VM. The second Tier is facilitated through an incentive compatible version of the Torrent protocol which is also tightly integrated into the OS. Despite the chosen Tier, users always have the experience and feeling of accessing a file with a double click from within the File Manager UI dApp. It is the same when uploading. User simply drags a file from the

local system onto the File Manager UI dApp (within the web-browser) and answers one or two questions. When opening a file, everything happens transparently, behind the scenes with everything, including video streaming software available directly from within the web-browser. **GRIDNET-OS** thus facilitates the functionality of an incentive-compatible content-delivery network with peers hosting files/video streams being rewarded with a system-intrinsic cryptocurrency.

~The Story Goes On. In traditional decentralized state-machines (Bitcoin, Ethereum etc.) one needs to deal with full-replication of data-storage. Now, even if Ethereum aspired to become a world-wide decentralized computer, two major yet indecent properties remained hidden, for the most of it, from the community's awareness. At least these were not marketed (understandably):

- **full-redundancy of processing**

That property leaves us in a situation in which even though we might have a vast network of full-nodes, the processing capability of the entire thing is on par with your laptop computer.

Doesn't sound good enough for a world-wide computer, does it? Well even still, one always needs to use the proper tool for the job. Ethereum gives properties (non-malleability, non-repudiation, decentralization etc.) But was the community aware of the imminent short-commings? When marketed as a decentralized computer? Were they even given the possibility to use it as something that resembles a computer?

Indeed, that has been a sad thing throughout the crypto-scene, with lots of projects trying to ride upon the lack of people's knowledge and awareness. There were projects promising slogans such as utilizing spare computer power, despite the fact that lots of underlying scientific problems remained unsolved, which was nowhere to be mentioned.

Now, how is that? Well, allowing for true decentralized computing, one verifiable through a decentralized state-machine would require a solution to the problem verifiable computing which has been under heavy research for decades and with current solutions far from practical (since solutions would require a solution to the problem of 'fully homomorphic encryption' solutions to which on the other hand are currently largely impractical in short). Still, some decided to promise much and as far as we know, stopped short at making it possible to order rendering of Blender scenes through Blender-rendering software, with no actual decentralized computing taking place.

Hold on hold on.. alright then.. Are we going to make things different when it comes to the first roadblock then? Short answer: Not anytime soon. We have been researching the problem for years, and got some intermediate practical results which are employed by **GRIDNET-OS** *right now. Also,* **GRIDNET-OS** *does employ processing on both the end-user's computer (for things like rendering of graphics) and takes use of the on-the-chain processing only when needed and when it makes sense.*

Somewhere in 2018 we've implemented into the Core of **GRIDNET-OS** the possibility of broadcasting OpenCL computational packages. Not going into lots of technicalities these computational-bundles are of various kinds, allowing for GPU processing, CPU processing or both(hybrid-mode). In short, one can imagine a computational package containing an OpenCL kernel broadcast within the network, with certain finalization criterias embedded and a bid in cryptocurrency. Once broadcast, nodes decide whether they want to do the processing or not with results delivered to the client.

The aspect of decentralized computing is built in and will be further researched and extended upon. With that said, researching further the functionality and bringing it to end users as a nicely looking UI dApp is not on the list of our current priorities but it's there on our minds. The current implementation already is on a higher level of abstraction than issuance of Blender rendering tasks and happens on a much granular and universal level of OpenCL kernels embedded within what we call 'computational bundles' that can be broadcast within the Network.

An interesting thing? 'Mining' done by the **GRIDNET-OS** full-nodes employs that very subsystem (including issuance of computation bundles, reporting progress and meeting finalization requirements once done). We've spent long months on it back in 2018 but it's better to say to not expect any pracitial GUI dApps employing the subsystem anytime soon than to promise too much. At least that's what we believe. Implementing things such as a fully decentralized DEX as one of GUI dApps within the **GRIDNET-OS** sounds like a much more sought for opportunity right now, well wait, did we just say that?

No offence for other projects around but sometimes the lack of putting things out how they are is sickening to say the last. We all want decentralized computing platforms; things are just not that straightforward.

- **full-redundancy of storage**

Lo and behold! Now, that is one of the fields where we are to introduce lots of improvements and innovation compared to traditional projects. More! We are going to do it out-of-the box in a matter of weeks (i.e. even before the ICO starts). In **GRIDNET-OS**, one will be able to take use of a Two-Tier storage subsystem. Wait..wait.. wait a sec.... Two-tier... like there are various levels of storage? *Let us explain.*

With **GRIDNET**-OS users will have a choice to either use the **GRIDNET-OS**' Eternal Storage or the **GRIDNET-OS**' Crowd-Funded Storage.

- **Eternal Storage** - in short, that is the kind of storage one would expect from a traditional blockchain. User gives a bid in ERG (**GRIDNET-OS**' Gas-counterpart) and if the network agrees, data is buried within the decentralized state machine. Forever. Hold on, that might not be fully true in our case since in **GRIDNET-OS**, in contrast with Ethereum, we have implemented effective data-pruning mechanism ~so if user one day decides to remove the file or replace it with another one, the file would be entirely removed from the chain after a sufficient number of iterations assumed to be irreversible).

Benefits of Eternal Storage? Obvious. Full-redundancy of storage, safety, non repudiation etc. In short - a perfect place to store high-value documents, possibly encrypted with symmetric-encryption. Of course it might not be cheap to do so as that kind of storage *is not supposed* to be cheap. As for the price, it's the same as with Ethereum and to be decided by the decentralized consensus, at full-nodes' discretion.

- **Crowd-Funded Storage** - did we say incentivized data-exchange to be one of our major research fields? Ok, here it goes. Let's face it - most of ust *do not* need full redundancy of storage. Besides, it is not good for the Network. Moreover, we want to store large files like HD or 4K video clips so not to waste storage on our precious computers. For most of us it would be good enough to see the file on

our account once we log-on into **GRIDNET-OS**, double click it and let it play, isn't it true? Well, let us deliver, actually,- *let The Community deliver* and let us make it possible, shall we.

~The Story goes on. One might be well familiar with the Torrent protocol. Indeed, it is one of most proliferated, battle-proven protocols for P2P file exchange, one which has been around for a decade or so and one which continues to be one of the best decentralized solutions.

~Now, what we did is - we took the Torrent protocol, its WebTorrent version and made these incentive-compatible so that each Torrent-node hosting files is rewarded with **GRIDNET**-Coins from the ones downloading data with the entire functionality-set available straight from a Web-Browser without the need of installing a single thing.

Indeed, providing an incentive for Torrent-nodes to host files is extremely beneficial all in itself. Thanks to such an approach nodes, this time, would not only have an incentive to share pirated copies of latest blockbuster movies (since they might have wanted to download these themselves and let

them be on their HDD seeded for a while by an accident), but to store and share users' encrypted data-files as well. Files whose incentive of possession possibly lies only for one person (the respected owner) and thus of which the traditional Torrent network would have no incentive for hosting. Note that here, the incentive for nodes to host user files would remain for as long as the user keeps providing nodes with an incentive for doing so (the innermost technicalities will be made available as one of our upcoming separate research papers). Here, let us focus on the most important, architectural parts, shall we.

The modifications to the Torrent-protocol involved making it compatible with the concepts introduced by us in some of our previous research papers. Mainly the Identity Tokens, Token Pools and Transmission Tokens. These are used to facilitate what we have established as State-Less Blockchain Channels which can be used to reward parties securely, off-the-chain. (i.e. we do not want 1000 transactions on the chain for a delivery of 1000 parts of a single 1Gb file).

~~Let us imagine a sample scenario. Let us say Bob wants to store a 1GB data-file within the fully decentralized **GRIDNET-OS**. Let's make an additional assumption that the file is encrypted so nobody else would have an incentive to be in its possession. Also, note that the file is also way too large for traditional blockchain systems to cope with.

~~Now, what Bob does is - he sacrifices a certain amount of cryptocurrency (don't worry it will be worth it!) say 1 GNC to be used for data-transmission and data-storage purposes. Then, he creates what we call a Token-Pool internally represented by a hash-chain - feel free to check out our research papers for more. The token pool references the just made Sacrificial Transaction and specifies the value of a single Transmission Token (or simply Token) from within that very Pool.

~Think of the Token Pool as a store of frozen assets, with the total value of the value-store described by the value of a Sacrificial Transaction and with each Token, representing the smallest spendable value from within that value store. The Token Pool gets registered within the **GRIDNET-OS**' Virtual machine. Creating and registering a Token Pool with the VM is a one-time process since the Pool might contain millions of tokens and be used for storage or deliveries of many files.

Now, let us get back to Bob, shall we. As soon as Bob drags a file from his Windows/or Linux onto the **GRIDNET-OS**' file manager (within the web-browser) and once he select the Crowd-Funded storage he would be asked to select from one of his registered Token Pools and to specify the total amount of assets (in GNC) which would be used to reward others for storage and hosting of the very file is is to upload, for a specified timeframe that is. The request would be broadcast within the Network and the willing-ones would download the file from Bob making it available for later use ex. from any location.

From now on, the **GRIDNET-OS** would be taking care of verifying portions of the file to be available from the ones who promised to be hosting it, while steadily delivering to Tokens from Bob's Token Pool in return. The ones hosting files would be able to redeem tokens for tradable cryptocurrency.

~~The two storage subsystem are connected. Meaning, when uploading a file to **GRIDNET-OS** (yes, a simple drag-and-drop to web-browser would suffice) one would be given a choice of the storage method. If, user chooses Crowd-Funded Storage, the file's footprint would be left within the Eternal Storage anyway with the file uploaded to peers who agree to store the data for a satisfactory amount of time. In any case, the user always has the feeling of the file to be residing within his or her folder despite the underlying storage apparatus. The upload mechanism is also very similar, the user just does a simple drag-and-drop from his or her local system and answers a couple of quick questions. Note that rewarding nodes for hosting user files happens off the chain. It is in the intention of hosting nodes to close the State-Channel the later the better since doing so involves a fee.

~~The GRIDNET-OS' file storage system works also the other way. It can act as a content-delivery network. Let us imagine Allice who wants to watch a copyright-free (needless to say) movie titled "Beautiful Roses", shall we. Alice would be able to do the search directly within the **GRIDNET-OS** File Manager UI dApp (all within the web browser, available even from her TV). Once found she clicks **~and it plays.** All she ever wanted. The system would be using one of her predefined Token Pools to reward the underlying nodes that would be delivering parts of the file, resulting in an uninterrupted high-definition Video-Stream.

a. Technicalities Related To Data Storage

For **GRIDNET-OS,** data-storage is just a single of many aspects of the entire ecosystem. As you might know, currently there are a couple of projects aspiring to provide decentralized File-Storage mechanics for an 'average user'.

- Filecoin
- Interplanetary File System (IPFS)
- BitTorrent File-System (forked IPFS and integrated with Tron)

Sources:
https://www.reddit.com/r/Tronix/comments/hi01xn/bittorrent_file_system_start_btt_mining_anywhere/
https://cryptopotato.com/whats-the-difference-between-btfs-and-ipfs/

Indeed, **it is** sort of funny to see sponsored articles create confusion of rivalry, when in fact it is Tron trying to make money by integrating the earlier into its blockchain and marketing it under Torrent Inc which it acquired.

Bittorrent protocol was invented and implemented by Bram Cohen as an open-source technology with an obvious possibility of being extended by anyone. Since then, it has been used by millions worldwide for which *all of us are thankful*. In 2018, Bram Cohen left BitTorrent, Inc. after it was acquired by 'Tron' startup, which exhibits aggressive, non-professional behaviour to say the least (judge for yourself, there are many examples available).

(..) After forking others' work...**(..)**

Still, **let us rejoice**! The **Bittorrent** protocol is **an extendible one** , with very well documented mechanics on how it is supposed to be done; ex. https://libtorrent.org/extension_protocol.html

Currently, large portions of the protocol used by most widely proliferated clients employ the 'extension protocol' to deliver functionalities, such as, but not limited to swam-participation and ut_metadata (BEP 9 - for deliveries of torrent-meta-data so that files can be fetched from P2P swarm, omitting reliance on trackers, based solely on 'infohash' without need for the actual .torrent file).

GRIDNET-OS employs the very same extension mechanics to enable for incentivized data-exchange and storage and so remains 100% compatible with Torrent protocol. In the same way, as µTorrent software remains compatible with any other Torrent client. The mechanics will be available to all **GRIDNET-OS** users out-of-the box, through the **GRIDNET-OS Web-UI** (i.e. through a graphical file-manager UI dApp).

In contrast with other projects **GRIDNET-OS provides:**

- **Linux / Windows-server access-rights / privilege-provisioning capabilities.** We've been literally looking through Linux's setfacl and getacl

man-pages to make GRIDNET-OS commands compatible! **Let us rejoice!**

Video URL: https://www.youtube.com/watch?v=2uv1GxC912w

- **In GRIDNET-OS user will be able to modify files whenever he/she wishes. No Read-Only folders!** Of course, owner will be able to set it as read-only but its his/her choice! **Let us rejoice.**
- Revoke / change access rights on demand
- **No need** to install **anything**, everything available through a beatiful high-tech User Interface (through a web-browser thanks to WebRTC)
- **GRIDNET-OS** rewards on-per-byte basis **during data exchange** if no payment received by the peer is cut-off. (with got research papers in the field since 2017 with major ones pending),
- Thus, in contrast with others, **GRIDNET-OS does** give support for a new decentralized YouTube's counterpart (to be available as a graphical UI dApp) where peers are rewarded **during the very data/video** transmission.

- Imagine getting paid for the actual bytes, with popular videos getting rewarded most (peers who deliver getting rewarded). In contrast with DLive, where rewards are issued by trusted servers after the user feels generous enough to 'like' a video.
- **GRIDNET-OS** **does** NOT involve any points of trust, from the user's perspective, let alone for his trusted **GRIDNEToken** application containing the user's private key which **NEVER leaves the device**. We have gone to some great lengths to achieve this. In **GRIDNET-OS** even the browser used by a user is assumed **not to be trusted**.
- **We're after perfection.** In contrast, BTFS employs an architecture where one needs to trust servers that verify 'shards' (parts of file) and issue rewards based on that. The actual data-transmission is not being rewarded at all.
- **GRIDNET-OS** allows for a **fully-open market**, with nodes, clients peers, those watching videos able to specify and negotiate ad-hoc price for data deliveries. Rewarded values might change during the very data-transmission.

In **GRIDNET-OS** incentivization of data-exchange is facilitated and carried out through the coined by us Blockchain State-Less Channels (as described within our Journal-publications).

We **would** gladly publish more details here but as you see, there are entities that would gladly 'fork as away' the minute we did. Even still, **we're** after openness, thus live-streams include all of the details. These will be at least making 'em' watch these through. Expect the very details to make part our next journal-publications.

b. The Incentivization Mechanics

Now, let us give here a short-overview of the incentivization mechanics in regard to the Torrent-storage subsystem. The mechanics employ *State-Less Blockchain channels*, with a modified version of *Token Pools*. We have codenamed the modified data-structures '*Multi-Dimensional Token Pools*'.That is to support multiple concurrent peers during downloads (even hundreds of simultaneous peers being rewarded at a time, with millions of simultaneous transmissions worldwide). All of the mechanics described herein will be hidden and transparent to the user.

As far as user-involvement goes, each and every user will be able to create/deploy a *Token-Pool* by sacrificing a certain amount of cryptocurrency. This will be possible through the Decentralized Command-Line (takes a single command) but through the fancy-looking UI as well (a couple of *clicks*).

Let us say you sacrifice 1 **GRIDNET Coin** (1000000 **GBUs**) to create a single *Multi-Dimensional Token-Pool*. Recall that the *Token-Pool* represents an off-the-chain value store. When you do so, you specify the number of *Dimensions* (or

‘Banks’) to be available from within that Token-Pool and the value of a single Token/hash from within the Pool (values will be proposed automatically). Still, for the purposes of sound explanation let us deeper. If you specify 100 dimensions and the value of a single *Token* as 1 **GBU**, then the amount of tokens available within a single Bank will be equal to 1000000/100, thus with each Token worth precisely 1 **GBU**. Yes, each dimension is finite, with boundaries constantly verified. During data download the software which runs within the web-browser will be delivering small portions of the accumulated value to the ones who deliver file/video fragments (possible once it is being played).

As far as the user is concerned, if he is a client, he will need to register a Token-Pool and be automatically informed once it gets depleted. In case of a data-delivery node he/she will be constantly informed of the amounts of just received rewards/cryptocurrency. Once a threshold of received rewards is reached, the software will automatically cash-out received tokens for the on-the-chain Live currency. In case of invalid tokens/ malicious peers etc. these will be cut-off automatically.

During data-transmission/reception, the Token-Pool’s dimensions/banks will be chosen and switched between autonomously by the software (the **GRIDNET-OS** JavaScript context). During data-transmission, the data-delivery node, with each data-bundle (ex. 100KB) receives a *Token* (hash), dispatched by client, one generated from within a certain *Token-Pool’s* dimension/bank. Here, each hash represents part of the total value accumulated within the particular Token-Pool. Do note that rewarding data-delivery does happen off-the-chain during the entire data-transmission between points A and B. It is in awardee’s best intention to postpone cashing out of the accumulated rewards (which this time does involve the decentralized state-machine), the later the better, since doing so involves a fee.

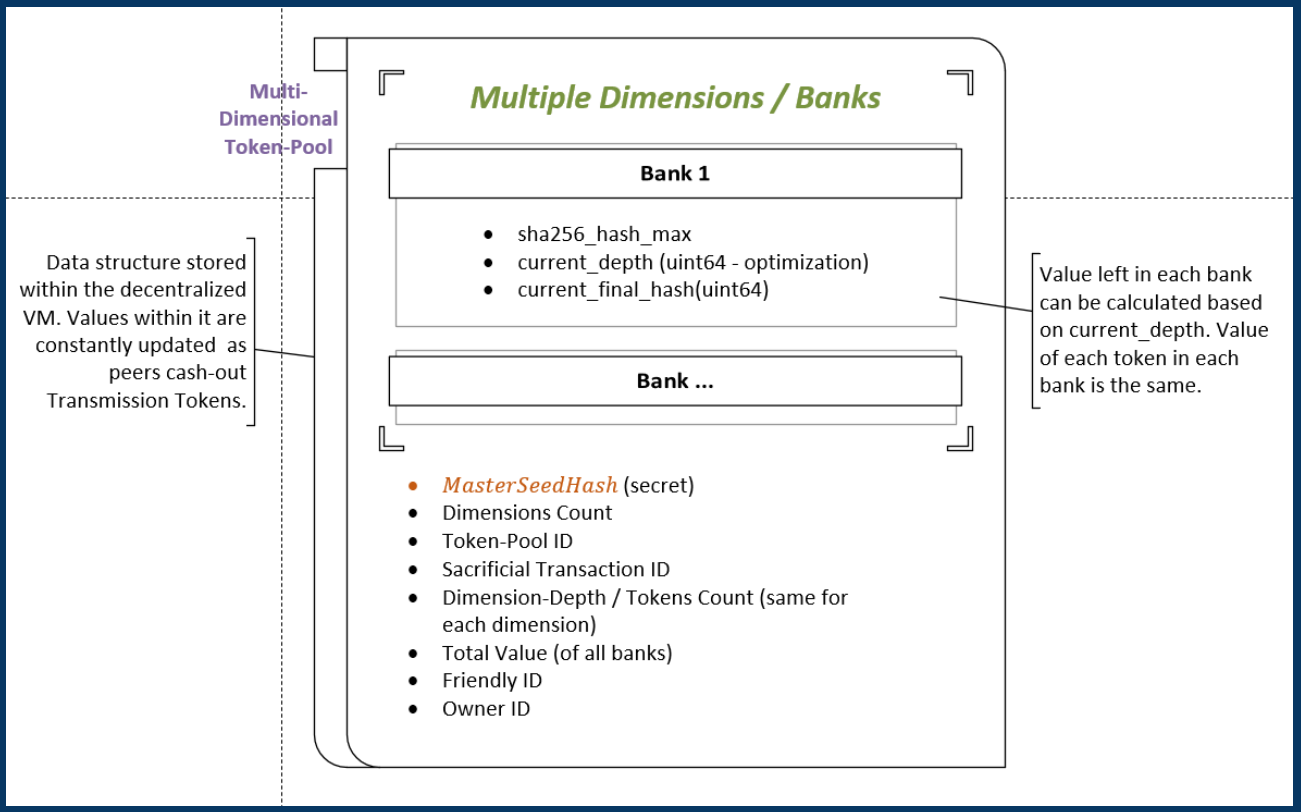


Figure 10: A Multi-Dimensional Token-Pool

Notice how efficient the mechanics are. Formulation of a *Transit Pool* (the data-structure which is to end-up within the VM at the end of data-transmission, which can actually cover for multiple data-transmission / files between A and B) involves mostly just a single hash, one revealed from the chosen Token Pool's dimension/bank. Here, it is important to note that only a single awardee can be rewarded from a single Bank/Dimension at a time, before he/she decides to cash-out his/her accumulated reward. The most recent hash suffices to validate the entire sub-chain of a bigger hash-chain, represented by a Token Pool's ***dimension***. Moreover, storage of the *Multi-Dimensional-Token Pool* on-the-chain is even more efficient. The secret ***MasterSeedHash*** is all it takes to recover information in regards to every ***Dimension/Bank*** with its value known only to the owner (stored off-the-chain). In that respect (storage and information recovery) the data-structure is similar to HD-Wallet keys, known from the Bitcoin-realm.

During data-transmission, after receiving a ***Transmission Token*** it is possible ***, at any moment ,*** for the rewardee to verify the current state of the on-the-chain-preserved ***Token-Pool*** and thus to check against a possible double-spend attempt by querying full-node for the Token-Pool to which the given ***Transmission Token*** corresponds to. If, the hash-value revealed turns out to overlap part of the hash-chain which is already known, that indicates a double-spend attempt and the data-delivery node can refuse further data-transmission.

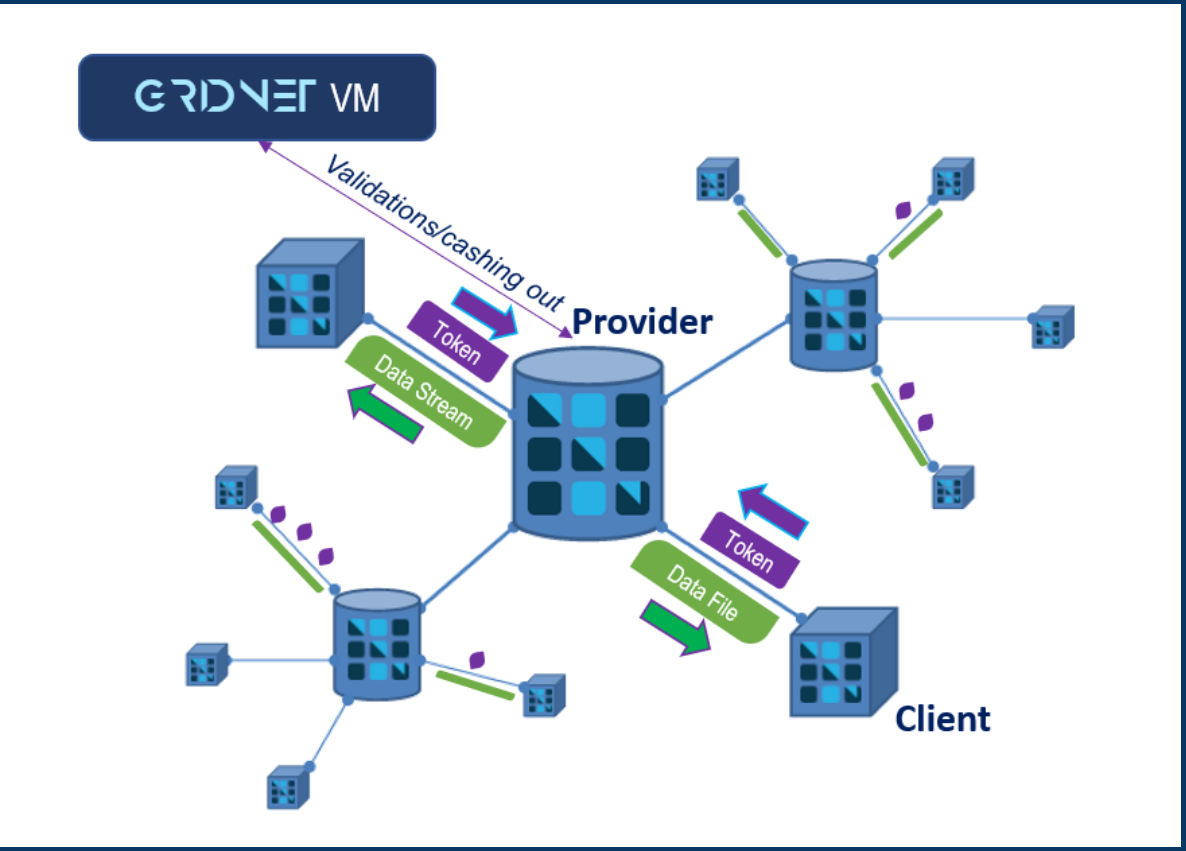


Figure 11: Overview of the incentivized data-transmission mechanics.

Once the Provider's software decides to cash-out tokens (as per user's preferences) for live ***on-the-chain*** currency, it will autonomously formulate a **#GridScript** transaction with the embedded Transit Pool (encapsulating the most recently revealed hash from a particular dimension, from a particular token-pool).

Now, the full-node which receives such a transaction would extract and verify the ***Transit Pool*** against the current state of the ***Token Pool*** (which should by now be available at the chain at all times). If validated, the Provider is rewarded with the amount of Live-cryptocurrency which corresponds to the total value represented by the received Transmission-Tokens (after processing fees have been deducted).

Kindly do note that here, everything is presented as a high-level overview and additional case-specific technicalities are involved. For instance, we employ additional variations of the Transmission-Tokens, including their Authenticated-subtype, one which renders the mechanics resilient against a malicious full-node. For instance, one might observe that under the so-far described assumptions - of each hash uncovering a certain amount of value -then, without additional authentication this would render the mechanics susceptible to a malicious

full-node which could claim the Token/hash as his, instead of node's that performed data-deliveries. Things get even more complicated when we employ additional mechanics to cope with Sybil-nodes in case of routed data-transmission (***Tunneling / IoT***) . *That very subject alone* is described in our upcoming 30+ page long research paper.

11. The Quality of Implementation

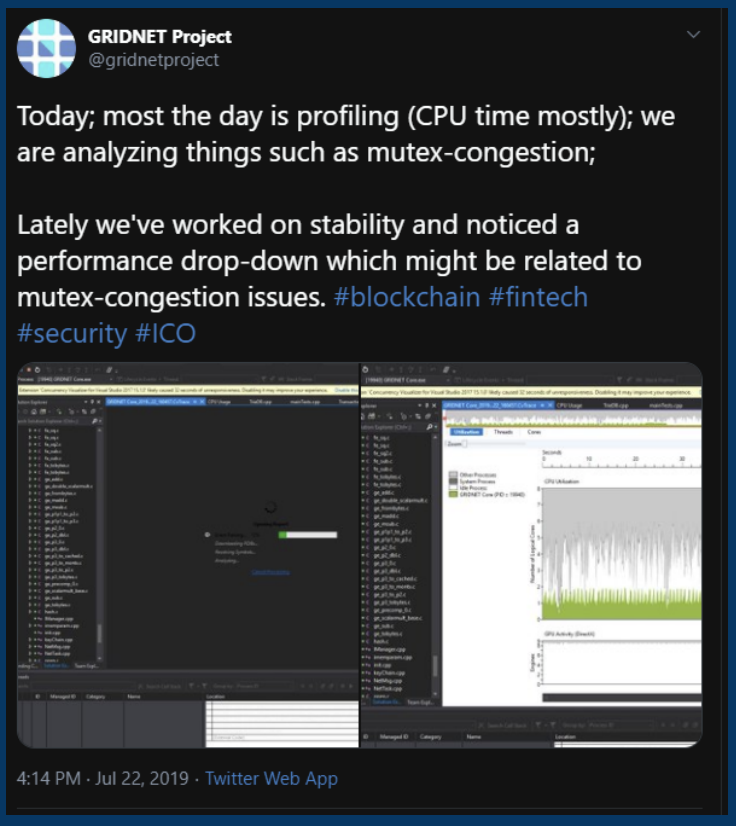
The project has been under heavy implementation since 2017 on almost 24/7 basis. We have been recording the majority of the recent implementation process LIVE on YouTube, so basically the entire 2019 has been recorded. No kidding. It is all there available to you for your consideration. We record everything including implementations of our custom elliptic-curve cryptography. We implement everything by ourselves. When needed, for security and user convenience we often go to great lengths, such as implementing the crypto-algorithms in JavaScript for use within the browser itself. We rely on our custom computer-vision algorithms to ensure that users never need to copy data-over through possibly insecure channels.

We expect **GRIDNET**-**OS** to run AT LEAST at NASA-requirement-levels and quality of stability we expect NOTHING less.

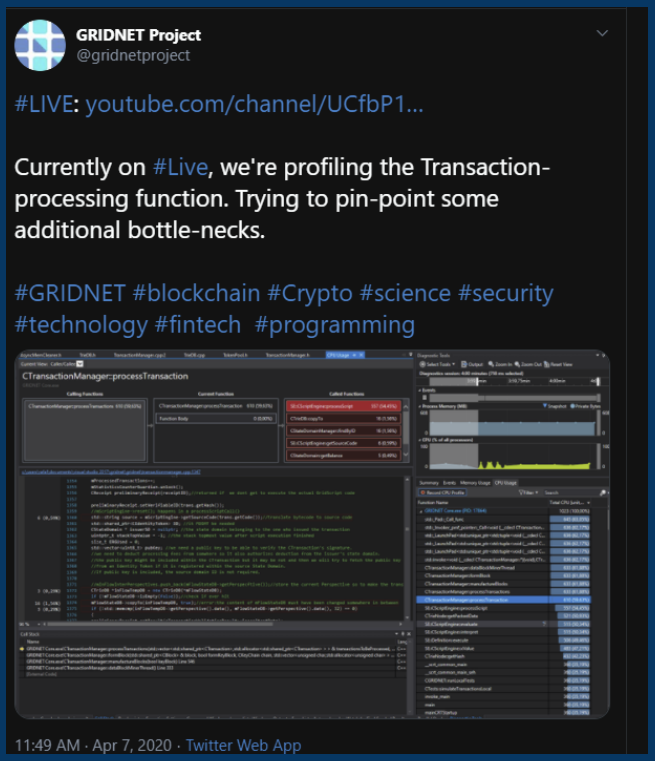
Taking a look at some of the past Tweets back from the early/midst of 2019. Back then we had basically the entire decentralized VM-up and running with its essential components.

- Here, we were thoroughly analyzing each and every function facilitating the fully decentralized **GRIDNET**-**OS**' VM. Trying to save on each and every nanosecond of processing time:

- Here, back in the midst of 2019,we were doing advanced mutex-congestion analysis:



- Long-time followers.. recall how, well over a year ago, we were doing Live-analysis of the **GRIDNET-OS Core** itself. Back then, we employed some of the most advanced methodologies known to Computer Science.



It is all there recorded on YouTube. The entire Window-Manager, the JavaScript Context and the showcased mobile authorization app - they have been all implemented LIVE. No kidding.

Scientific Papers

We've got scientific publications at most renowned scientific conferences and journals already; - with more pending. We were the first to come up with a communication protocol rewarding with the concept of a cryptocurrency on a per-byte basis (https://ieeexplore.ieee.org/abstract/document/7899560). We have introduced the concepts of State-Full and State-Less Blockchain channels and proposed the first fully decentralized blockchain based-architecture for Smart-Grid environments (https://ieeexplore.ieee.org/document/8340700) as well. If, you are new to the recently introduced by us groundbreaking concept of State-Domains you might want to check-out one of our latest research papers entitled "**The Open Blockchain-Aided Multi-Agent Symbiotic Cyber–Physical Systems" - OBAMA-CPS** in short, published in one of the most renowned, peer-reviewed, scientific journals on the planet Earth:

https://www.sciencedirect.com/science/article/abs/pii/S0167739X18307520

.. already gaining quite a few citations..

While there **are** other projects around, the prominent underlying idea behind the **GRIDNET Project** is to provide an ultimate Decentralized Operating System experience, with no hard-coded constraints at all. Everything shall be decided through decentralized governance and the forces of supply and demand.

We want to allow for an uninterrupted deployment of innovation leading to freedom, while providing game-theoretic remuneration for all the parties involved within the system.

- **Hold on, wait a sec, aren't other blockchain-systems incentive-compatible already?**

Answer: Current blockchain systems run mostly on altruism (https://hackingdistributed.com/2015/12/22/bitcoin-runs-on-altruism/). We've designed **GRIDNET** to incentivize all the relevant aspects and parties involved (including data exchange, data storage, block propagation, transaction propagation etc. - scientific publication pending), which should lead to further decentralization and elimination of trusted points of failure. If, parties not having a chance to become a leader would still be incentivized (in a Sybil-proof manner) to route data for others, this would lead to lower confirmation times and overall improvements to the network's stability.

- **What are Sybil nodes and why they are bad?**

Answer: Sybil nodes are artificial entities that distort the real-perspective on things, introduced by dishonest parties in order to gain profit. **GRIDNET-OS** is designed in such a way that it discourages Sybil-nodes entirely (still, you'll need to wait for our upcoming scientific publication to get more details on this one). The problem of Bitcoin-like system not in incentivizing data-exchange was first spotted by Babaiaf et. al. from Microsoft Research Download

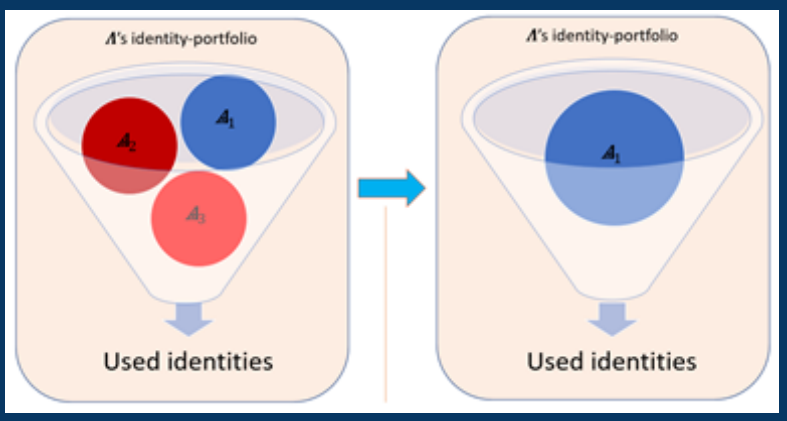
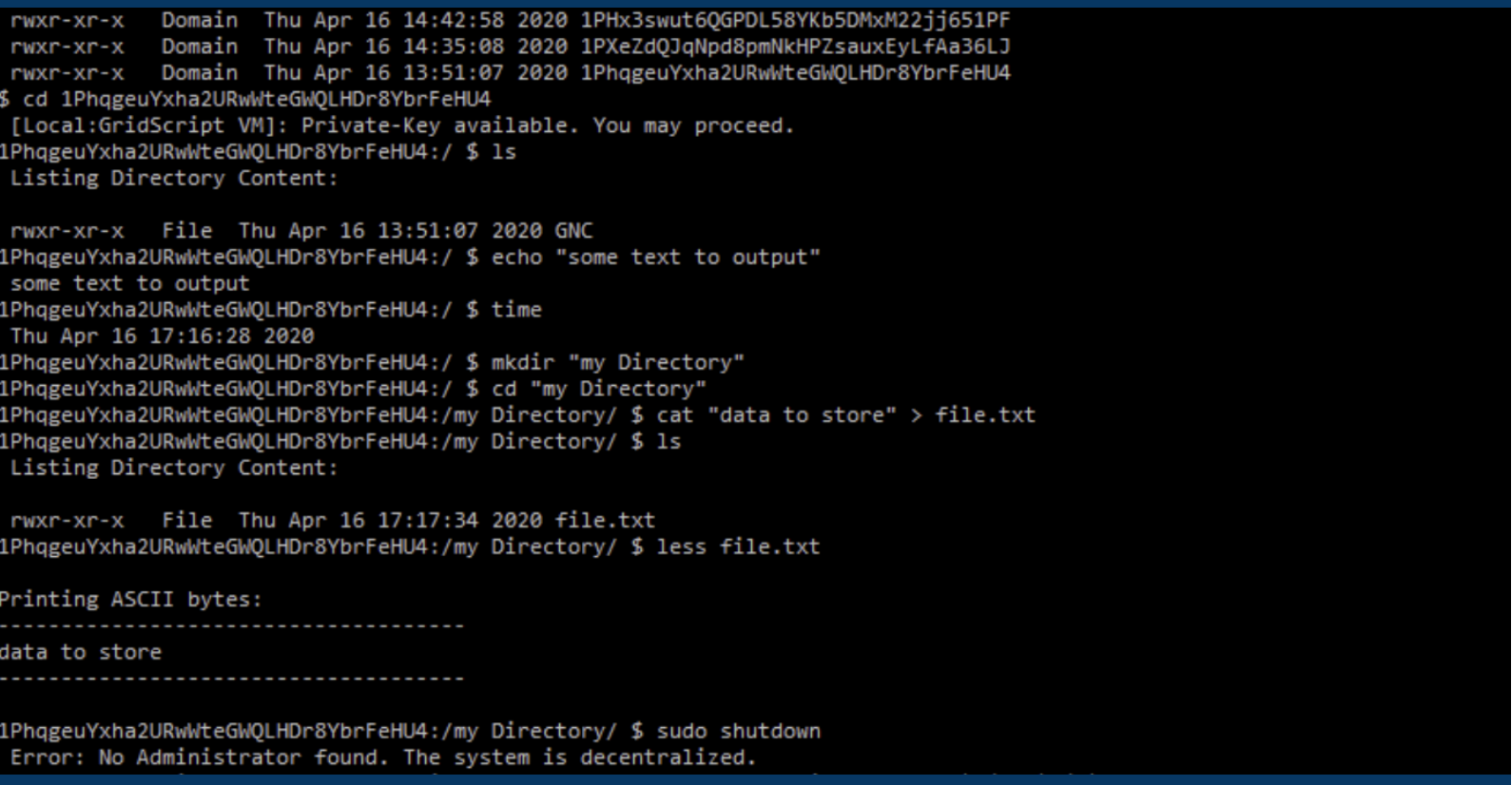


Figure 10. *Sybil nodes / entities and why they are bad.*

- **What is unique about #GridScript , I know Ethereum has solidity, Bitcoin also has some sort of scripting language.**
  Answer: In short take a quick glance at a video from our Live YouTube-stream: Link

#GridScript is a quite an artistic variance of Forth, would you have guessed? Got ya. The language allows developers to perform low level memory allocations / manipulation (for nerds: yes, certain functions indeed provide just a gentle wrapping around 'malloc'), while on the other hand providing a familiar high-level Linux-like experience for users unexperienced with the concept of 'Stack' and those who want to get things done fast. The decentralized Command Line built into GRIDNET-Core provides a debugging environment, with automatically generated comments for each Stack-entry, for the convenience of each respected user and/or developer. (much



more information and documentation to come).

- Wait a sec, I dunno what a 'Stack' is and a few other things you were talking about..

Answer: That is precisely why #GridScript might be suitable for both you and professional developers alike. The language allows for high-level in-line parametrized command invocations, many computer users are already well familiar with. Do you know what `'mkdir myDiroctory'` does, rings a bell? Then fine you should be - it is a #GridScript instruction as well. In a very similar way you'll be able

to execute decentralized Smart-Contracts created by other people. Just by ‘CD’ing into their State-Domain and executing ‘exec smartContractName’ or ‘run smartContractName’ or ‘call smartContractName’ providing inline parameters when needed.

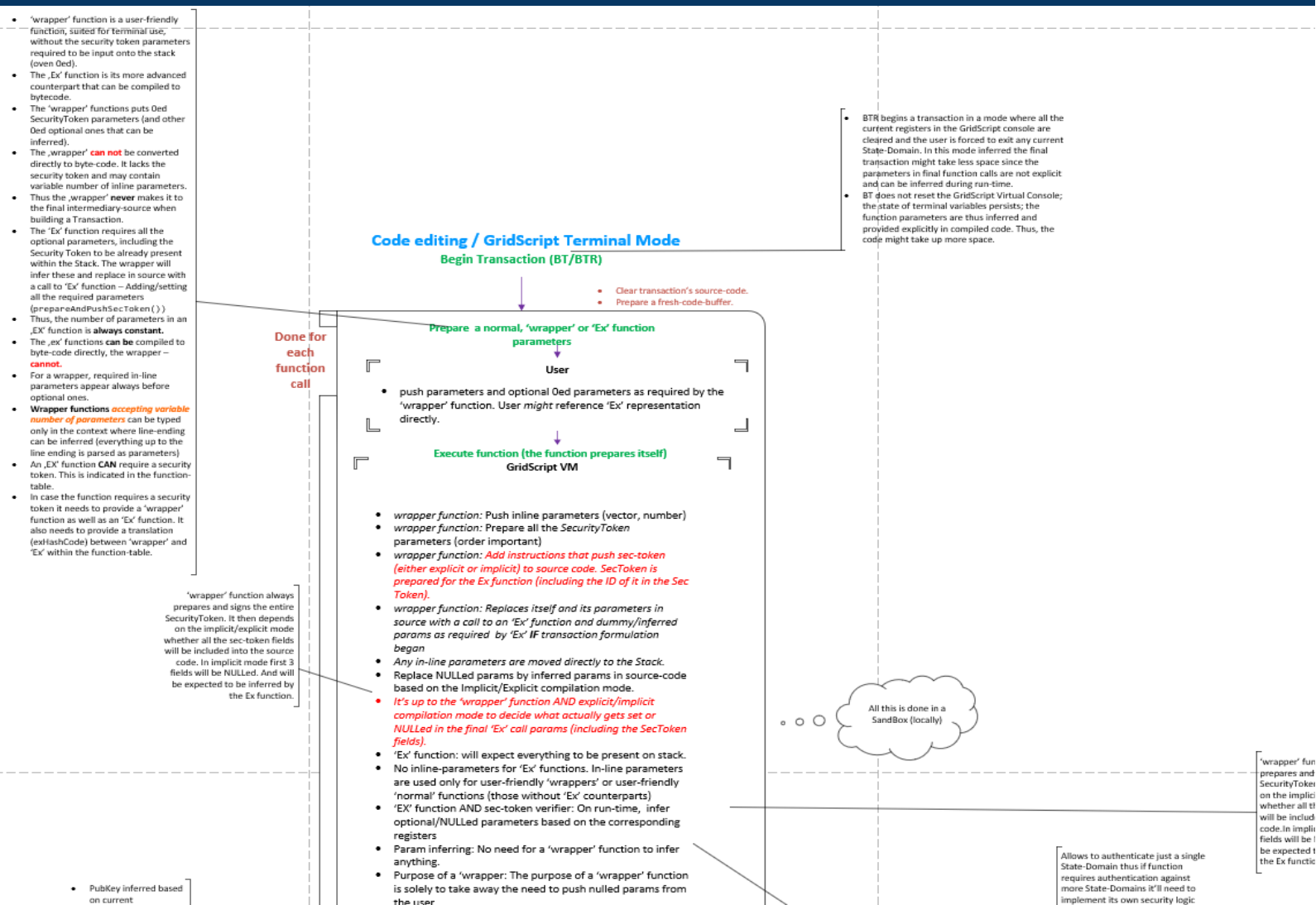
- **Why an entirely new Blockchain platform?**

Answer: *Quick question; have you ever seen or heard of anything we’ve just described?* Let us now give you a brief explanation / rationalisation from both the technical and economical perspectives alike.

- **Technical Rationalisation**

Bitcoin forked due to much shallow reasons than the ones proposed in the **GRIDNET Project.** We have got many exciting things anew. We were never afraid to redesign everything from the ground-up. **GRIDNET** introduces so many new concepts and requirements that there simply was no other way to approach our ambitious conceives.

The ideas of rewarding remuneration for data-exchange and the concept of providing an uninterrupted decentralized Operating-System experience required reconsideration of the most fundamental building-blocks of decentralized open-ledger systems. During the design stage, we have meticulously analyzed the available solutions, including projects such as Bitcoin, Ethereum and Bitcoin-NG.



While the **GRIDNET Project** borrows many aspects from the prior, we stress that there is *not a single* line of code common with other projects; in fact - authors of the **GRIDNET Project** have never had time to look through these implementations and focused on analyzing the available scientific publications instead.

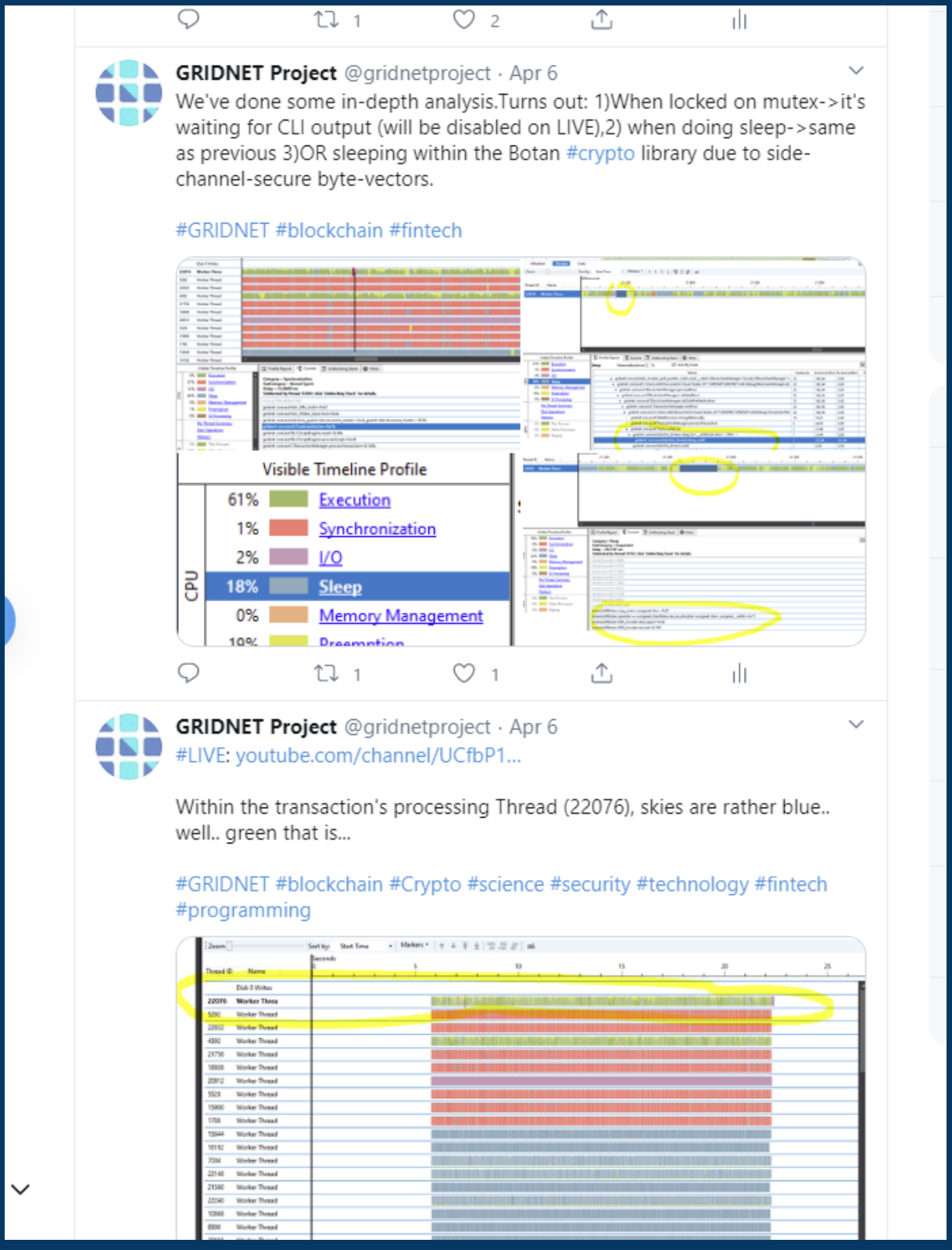
There simply was no time to Reverse-Engineer often-times messy, open-source code - a process which would most likely end-up as an exercise in futility.

It was much more rewarding to start anew even though it took a lot of time, and required a lot of investment - mostly in time - now counting in years. The developers of the **GRIDNET Project** never were after 'easy-money', with each core algorithm meticulously verified using methodologies such as TLA+ before proceeding to the implementation stage.

Now and then we went to great lengths to maintain and fulfill our self-imposed requirements, for instance - as far as modifying crypto-libraries themselves in order to make **#GridScript** as easy and attractive to use for end-user as possible (https://github.com/randombit/botan/pull/1239) - the implementation within the well established Botan Crypto-Library allows for *both* encryption and signatures using the "NSA-proof" X25519 Elliptic Curve - using the same private key for both. That was never seen before as far as we know.

- **Financial Rationalisation**

For the past few years, the **GRIDNET Project** has been developed with thanks to private financing made on behalf and through the generosity of Rafal Skowronski - the head of the **GRIDNET Project**, with additional support and close cooperation from the Poznan University of Technology, also in cooperation with the GRIDNET Technologies LLC Poland (established mostly from a sparse amount of financial assets collected during the pre-initial coin-offering that took place in late 2017 to amount for the total of less than 10 000 US dollars). These assets were officially brought to the company's capital, with all the spending officially accounted for, recorded and made available for control to suitable governmental agencies with all due taxes paid.



Transparency of development.

Since then, **GRIDNET Project** has been in a continuous, uninterrupted development with daily/hourly updates made available on Twitter (https://twitter.com/gridnetproject) and live-programming/debugging/testing sessions

recorded Live on our YouTube channel with updated made available on hourly/daily basis :

https://www.youtube.com/channel/UCfbP15L0fMwQoAD69ZaT7OQ/

Despite the minimalistic funding, we were never after easy-money and stubbornly continued to push The Project further, oftentimes working on a 24/7 basis.

The developers have always been after openness. Thus, on social networks you may find the good and the bad. The 'hype' stuff we came up with, that was implemented; and the roadblocks, troubles we have stumbled upon, that oftentimes took days or even weeks to get over with. The code quality can be checked and analyzed to the granularity of how fast we type code - it is all there, recorded on YouTube.

Conversely, the team behind Bitcoin is estimated to include hundreds of people. Whereas, the **GRIDNET** project has continued to be uniterpatedly developed with the help of people whose numerosity could be counted on the fingers of one hand (for nerds: yes, `base_10`).

Still, with all that said, we've already researched and implemented functionality that far exceeds and surpasses the one available in Bitcoin and can firmly compete with possibilities made available by projects such as Ethereum - financed by fat millions of dollars in cooperation with centralized companies such as Microsoft.

To the core of financial matters: GRIDNET Project will set off with a Genesis Block in which State-Domains of each respected initial investor and the **GRIDNET**-**OS** Team itself - will be created. These virtual assets are supposed to secure the development of the platform during the foreseeable future. These virtual assets will be used to cover for the expenses of the development-team, artistic-team, marketing-team etc.

The crucial point is not to face the problem of the lack of incentive for the internal teams to lose their will to cooperate as often seen in other projects. One of the main underlying ideas of the **GRIDNET**-**OS** is to incentivize all the parties involved - consequently this includes the **GRIDNET-Team**. We'll always strive to maintain the highest standards of code-quality, documentation and to incentivize rapid deployment of innovation. People involved in #GRIDNET oftentimes treat it as their full-time (oftentimes literally) job.

Sample applications available within the UI-mode of GRIDNET-OS
(work-in progress)

GRIDNET-OS has been designed and implemented from the bottom to target users' needs. We target both computer professionals, hackers, programmers and typical end-users. When we target, we deliver. While implementing the eco-system since recently, we've been striving to provide users with some sample easy to use, demonstrational applications. These applications employ power of both the user's computer as well as of the GRIDNET's VM.

If, the decentralized state-machine, running the #GridScript language upon which the GRIDNET-OS executes is the brain of the system, then the Decentralized Terminal Interface (DTI) facilitates the main testicles of living organism which the GRIDNET-OS eco-system certainly is.

There are a couple of ways Decentralized Terminal Interface (DTI) can be accessed:

- **Directly at full-node software**

  When running the full-node software, the Operator will have the possibility of switching between the Events View and DTI view all within the Linux/Windows command-line software. Yes, we've gone to some great lengths to enable 'views' within CLI, especially on Windows.
- **Through SSH**

  Like any other popular operating system, **GRIDNET-OS** can spawn a remote shell and thus be accessible remotely and securely through an encrypted communication channel over SSH. For more information regarding Secure Shells, the respected reader is advised to look over here
- **In Graphical User Interface through the Terminal dApp**

  GRIDNET-OS facilitates probably the most beautiful graphical-terminal interface on the planet Earth. Wouldn't you say?

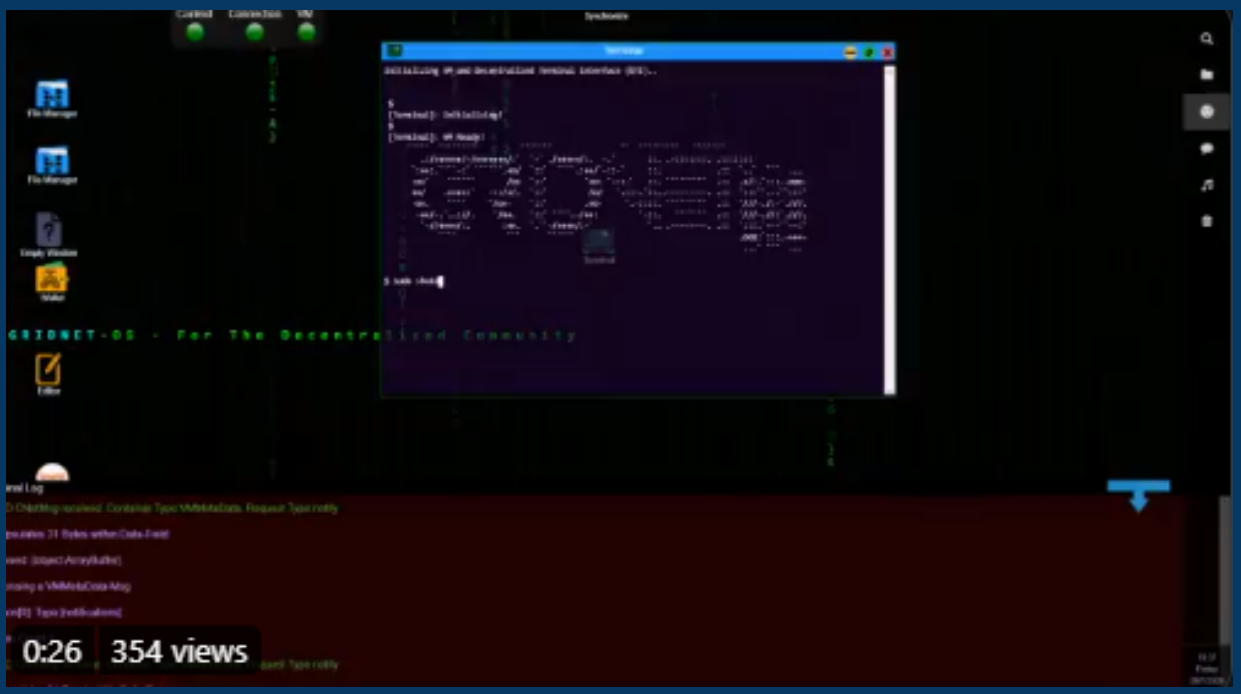


Figure 11: Don't be shy, you can watch it over here

As with everything else related to GRIDNET-OS, also here we've employed top-notch technologies, to enable for the vision of a true terminal-interface, within the decentralized, remote graphical user interface (sounds like science fiction) to emerge. Here, the remote shell is spawned and processed entirely at the discretion of a full-node facilitating communication with User, at any given moment. That means that each keystroke is being sent and processed by full-node before displayed within the Terminal. Each key-store is displayed EXACTLY if allowed and as deemed by the full-node software with which the Graphical end-point communicates with.

For this have employed Web-Socket based communication, with binary encoding, performed by a custom implementation of a BER/DER encoder. The graphical end-point is facilitated through a sound orchestration of xTerm.js (battle-proven library used by companies such as Microsoft within its Visual Studio-product line, also in Eclipse, Atom and others) and the **GRIDNET-OS**' JavaScript Context.

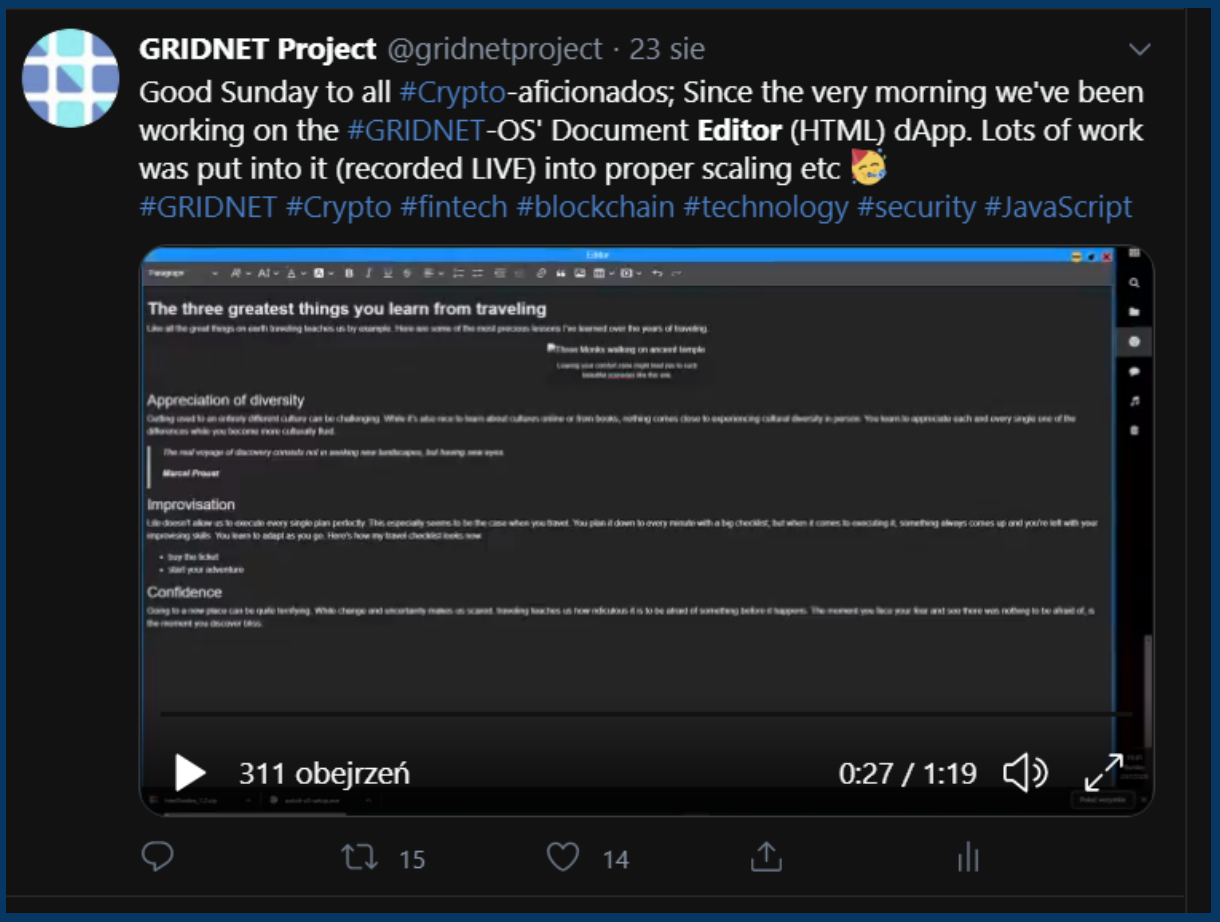
Also here, the Terminal supports multiple views, the terminal is fluently scaled in response to user actions (number of columns and rows constantly and fluently adjusts as the user resizes the window). For all the nerds our there - YES it does support colors and YES - there's BELL :-)

Independently from how the Terminal is accessed be it SSH or GUI, the Security Subsystem at GRIDNET's full-node-software constantly monitors network communication for mis-use.Shall the User input characters artificially too often, the

respected user will be warned and shall the behaviour continue, the connection would be dropped with the user's IP address temporarily banned. Security of the Network has been our top-most priority from Day-1.

The Instance of a virtual machine required for processing of user's input would also be spawned and deactivated autonomously, on demand.

- **Editor**



A full-blown text/html editor available straight from the Desktop, offering intuitive Desktop-like experience with the ability to save work directly to the decentralized storage watch it here

LIVE sessions *These mechanics have been all implemented LIVE. You may find videos on YouTube matching the corresponding Tweets at the same time the tweets were created.*

- **Walle(t)**

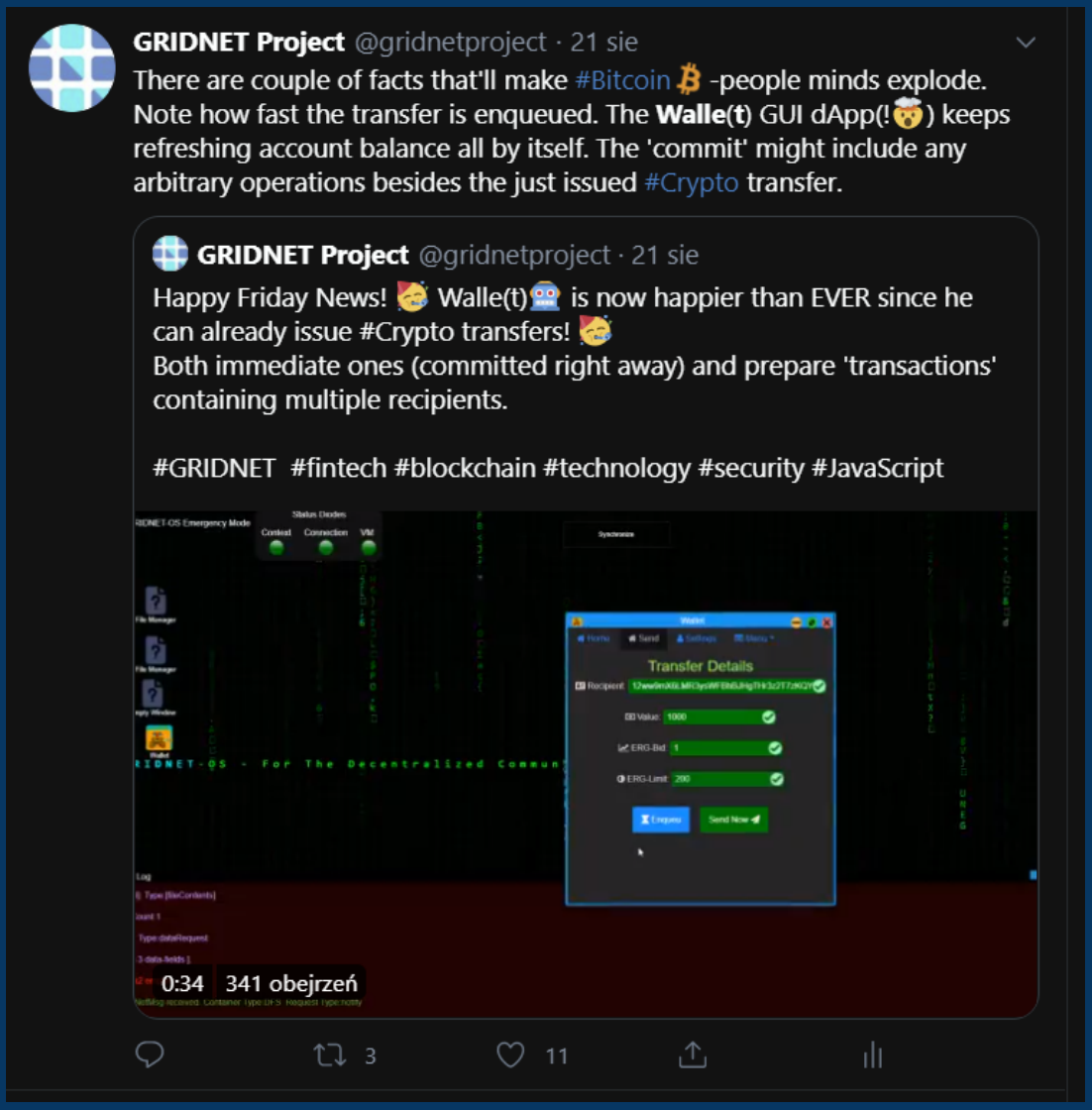
Watch: here

Walle(t) is an example of a GUI-dApp which awaits and reports changes occurring within the GRIDNET's decentralized State-Machine. Similarly to all the other aspects of the system it was designed to be as autonomous as possible. Once wallet's address is input by the user, the account status will keep updating all by itself. In case of disconnection from a particular full-node another one will be chosen randomly, transparently to the user (also transparently to Walle(t)).

It is worth to note some of the details such as animations occurring during fields' verification, making it hard to believe that things like this were made possible right within the browser.

Wallet(t) takes use of the GRIDNET-OS' underlying dynamic, autonomous transaction formulation. Meaning the #GridScript code describing the transaction will be generated under-the-hood, transparently to the user. User has a choice of either issuing an unitary transaction to a specific recipient and sending it right away or to continue with additional transactions. If user chooses to issue an instantaneous transaction, after a single click, the #GridScript code describing the transaction would be formulated by the **GRIDNET**-**OS** Virtual Machine and the full-node handling the particular connection would kindly ask user to scan an QR-Intent shown nicely to to the user on the screen with hers or his mobile GRIDNEToken app. The token app would then deliver the required signature to the particular full-node and broadcast the transaction to the rest of the Network.

If, however, user chooses to continue with other tasks, such as issuing additional value transfers, and or creating directories, signing files or whatever the user's intention - the **GRIDNET**-**OS' VM** would continue overwatching user's actions while formulating the required #GridScript code, transparently to the user, in the background. All the actions would be then visible to the user and awaiting confirmation through the Magic Button.



Watch: here

LIVE sessions *These mechanics have been all implemented LIVE. You may find videos on YouTube matching the corresponding Tweets at the same time the tweets were created.*

- **Messenger**

Like we said, we're about freedom and decentralization. And we mean it. If we mean it, we deliver. Messenger (sounds familiar to anyone?) is a GUI application allowing for a fully end-to-end encrypted communication. Interestingly, the communication process does not go through any third-party servers (including full nodes maintaining the **GRIDNET-OS** eco-system!) giving not a single chance for anyone to store and maintain meta-data regarding your private conversations. The app will be available once the system goes LIVE, so you'll be free to analyze the communication traffic with tools like Wireshark to make sure that that indeed is the case. The app has been implemented using technologies such as WebRTC.

ℹ️ **LIVE sessions** *These mechanics have been all implemented LIVE. You may find videos on YouTube matching the corresponding Tweets at the same time the tweets were created.*

-  File-manager

  What good would an operating-system be without a nicely looking file-manager?

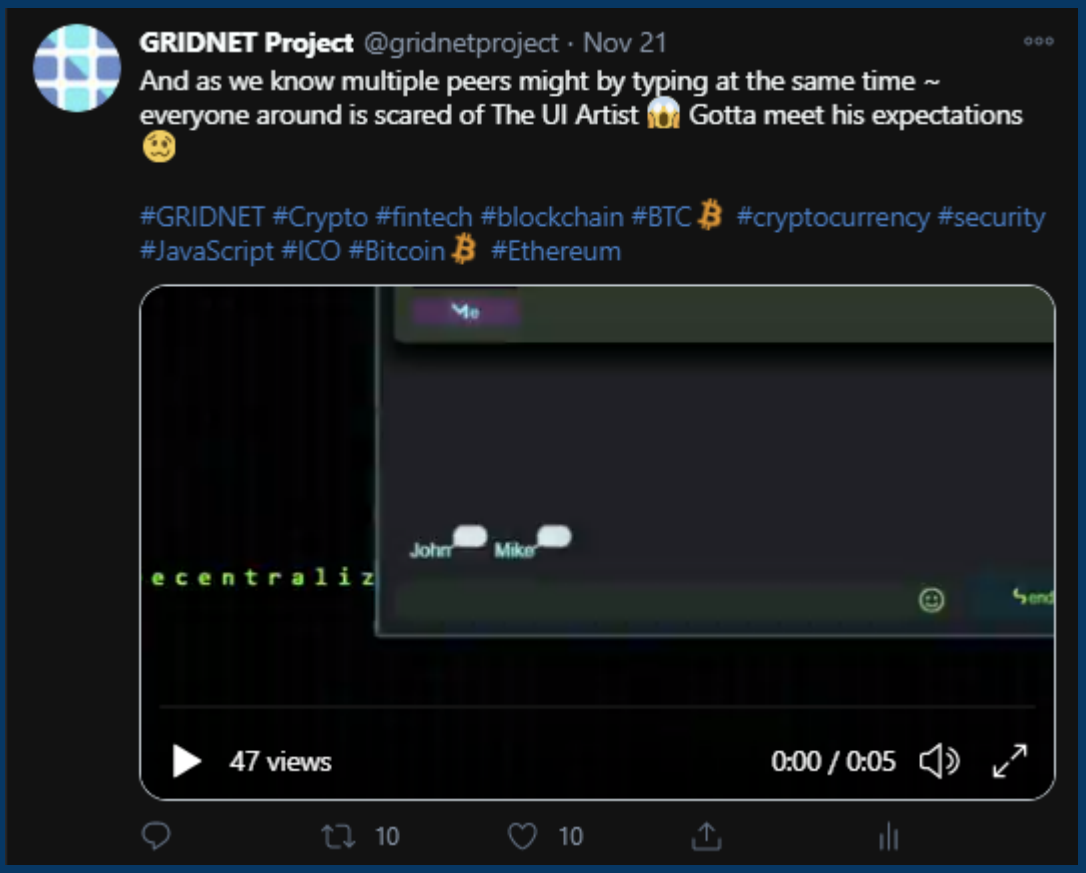
  For some additional brains-out experience you might want to watch how easy it is going to be to transfer files from local Windows/Linux/mac machine to **GRIDNET-OS**: here

 **LIVE sessions** *These mechanics have been all implemented LIVE. You may find videos on YouTube matching the corresponding Tweets at the same time the tweets were created.*

-  eMeeting - one of the most technologically advanced UI decentralized applications to be available on **GRIDNET-OS** as soon as it becomes publicly available.The application takes use of the **GRIDNET-OS**' P2P *Swarms* API. Nobody is hosting the conference we are dealing with full decentralization. The conference is established by providing a conference identifier and only those knowing the secret ID are allowed to join. The development of the application, including design stage and implementation in large portions has been recorded LIVE on YouTube.

The signaling node is chosen at random from the network of full-nodes. Full end-to-end encryption is provided among participants at all times. This holds true for raw data and/or audio/video streams. Participants can share the entire (selected) screen and/or windows of a particular application (such as PowerPoint slides). This allows for carrying out business meetings and is being battle-proven at the Poznan University of Technology for remote classes during the Covid-19 pandemic.

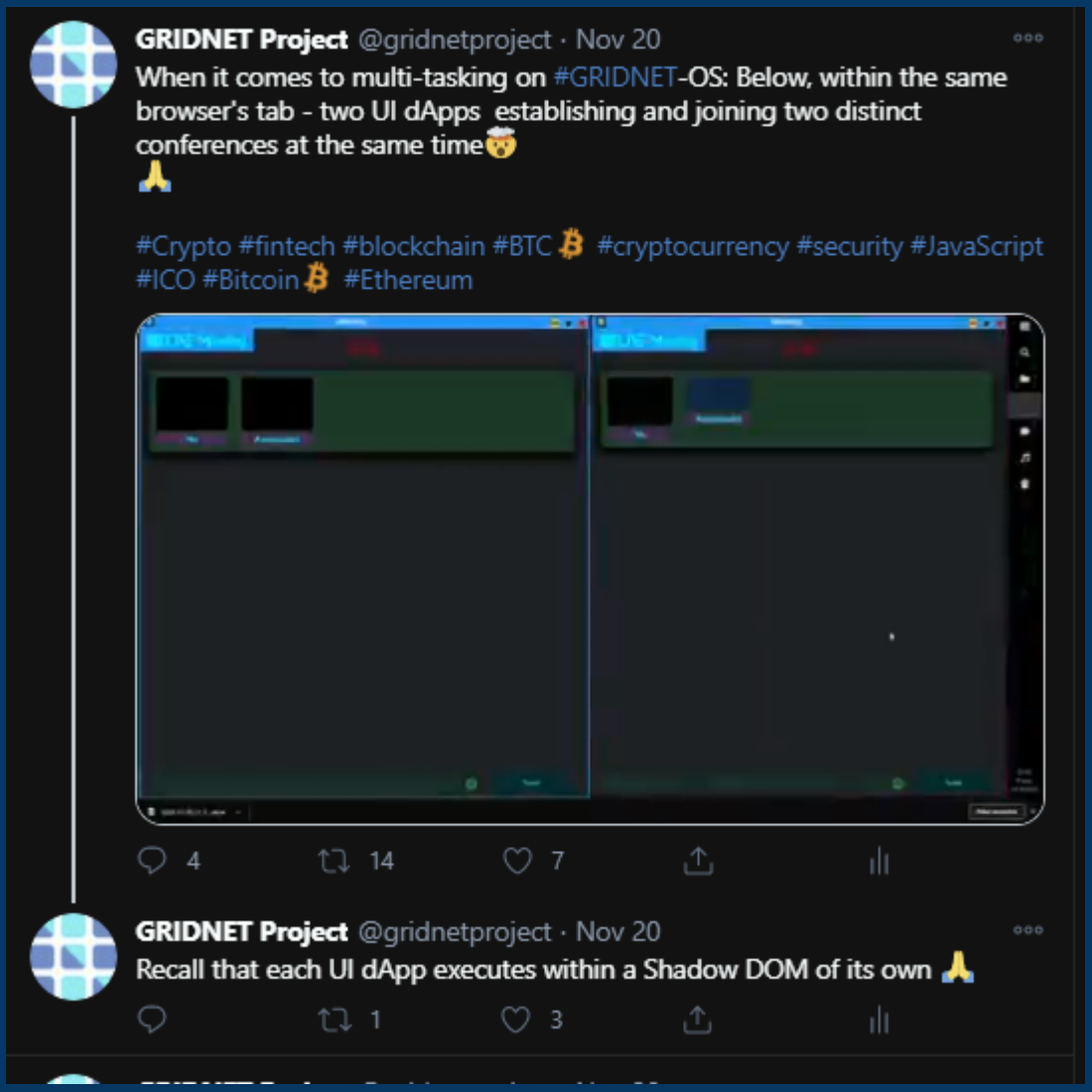
- Make sure to see the amazingly animated retractable emoticon's pane: here
- Take a look at how errors and/or notifications are presented here
- See how easy and intuitive it is to switch selected peer's video stream to full-screen and mouse out to minimize here
- Full Messenger's like experience with even better animations all around. Below, notifications when someone is typing:



Watch: here

ℹ️ **LIVE sessions** *These mechanics have been all implemented LIVE. You may find videos on YouTube matching the corresponding Tweets at the same time the tweets were created.*

- Joining two distinct conferences within the same web-browser’s Tab:



Watch: here

- The overall UI is a real work of Art indeed. Lots of work was put into ensuring that all the elements scare properly among a variety of screen sizes and with participants ranging from a couple to dozens:

Watch: here

The eMeeting UI dApp has is now being constantly used among Team Members during and for the purposes of development of the **GRIDNET-OS** itself.

Watch: here

8. GRIDNET-OS *vs* other 'security concerned' Operating Systems

Let us wonder, how will this OS stand up to the likes of OS such as: Qubes, Tails, and Whonix? What does it mean that the system is "decentralized"?"

Our ultimate goal is providing the ultimate functionality set for decentralized communities, including what you've mentioned, through employment of decentralized consensus, through employment of a decentralized state-machine.

Gnutella.. it's been long since I've heard about it. You might notice in the first lines of wiki's article " June 2005(..)" Gnutella pre-dates even Bitcoin and has very little to do with the concept of decentralized state-machines. Gnutella is or either mostly was about data-exchange only.

Decentralized computers have been never around as the nomenclature of a computer relates to a single physical entity. What has been around are decentralized state-machines which are a more abstract, higher level concept if I may. Bitcoin can be said to be one of these.

Let us take a look at the projects you've mentioned:

- **Qubes OS** - In short the major selling point of Qubes OS is that it isolates one app from another.It's major aim is to isolate applications running on a single computer for security reasons; as it's wiki states:"Qubes OS is a security-focused desktop operating system that aims to provide security through isolation."

All that implies that it 'runs' on a single computer. Meaning, nothing is decentralized. Actually **GRIDNET-OS** is the opposite - it provides security through decentralization rather than isolation.

Of course, the understanding of 'security' might be different here. The rationality of Quebes OS is not obvious since for it be beneficial one needs to question the security of an operating system like Windows/Linux itself. Still the security of Qubes relies on security of the underlying hypervisor Xen - upon which the very Qubes executes. One might think that the hypervisor might be more difficult to break, still, we've seen that happen already with recent Intel-based flaws related to side-channel attacks.

- **Tails** - it does not aim to provide low-level isolation between processes. Still, it is similar to the above in the sense that it runs on a single computer. Nothing is decentralized here either. It 's 'just' Linux running on your computer with some anonymity-related software routing everything you do through the Tor network.

- **Whonix** - (formerly TorBOX) it is about your anonymity on the Internet. It has been integrated into Qubes OS recently.

Now, none of the above projects will enable you decentralized voting, decentralized applications whose computational results are guaranties to be legitimate, none of them will allow you to prove in the eyes of the others that you indeed created a file with a certain content at a certain point of time, none of the three will provide a secure fast value-exchange asset,They won't allow you to create application that react to facts reported by others, the list goes on, further they won't provide the ability to execute applications which assign assets to others based on some verifiable criteria and/or facts. They won't allow you to access the ecosystem from anywhere from any 'access-point' remotely and securely. The list goes on.

You might be better off comparing **GRIDNET-OS** with platforms such as Ethereum. Much more similarities there. Ethereum provides a decentralized state-machine similar to the one **GRIDNET-OS** does. Now, Ethreum stops short at that. We've designed **GRIDNET-OS** from the ground up to enable for a decentralized Operating System experience, that includes the possibility of running decentralized graphical applications, shells over SSH etc.

ICO

Everyone wants to be rewarded for their hard work right?😆 But making a good investments sounds even better🤩 Above all, value assets collected during the 'ICO' will be used to secure further development of **GRIDNET-OS** for the foreseeable future.🙏 Details related to the it are yet to be announced, you may proceed to the FAQ section below for some additional information available on that matter.

Update: we have been receiving A LOT of questions in regard to becoming an early investor. Most probably the uni-swap algorithm will reward early investors should the ICO last for a prolonged period of time. For any additional details please do follow us on Twitter.

FAQ

- **When does it go LIVE?**
  The Test-Net is already up and running. So are the mobile apps for both iOS and Android. All transactions taking place on test-net will have a permanent effect on LIVE-Net. Thus the underlying crypto asset may be considered as fully operational right now.

- **Why would you decentralize an Operating System?**
  Why not ~~Wizards🧙 are decentralization fanatics. Seriously though - decentralized services. That includes decentralized graphical user applications. That includes those employing decentralized storage and so on and so on. It allows other developers to deploy user-based applications, tapped into some of the cutting edge technology at a minimum amount of work.

- **Who are the Wizards?**
  Wizards🧙.. They come from the lands of bits and bytes.. the 0s and 1s.. armed with the guns of computer science and cryptography no mercy for centralization have they got. There are some who write new Elliptic Curve-based cryptographic primitives, extend upon most popular libraries, some that know they way round latest UI/UX technologies and some who remember ANSI sequence codes.
  (..) (..) then there are those who specialize on mobile technologies, best of these .. Android.. iOS.. computer vision.. artificial intelligence.. you name itMagiczna różdżka They all keep working together bringing all of us towards the Glory of Decentralization. 7 days a week.
  (..) let us not forget those who clean up after the others knowing their way around temporal logic of actions, constantly checking out the algorithms with TLA+.

- **How is the above achieved?**
  Through a mixture of technologies only including the Blockchain technology to create an immersive decentralized operating system experience and never before seen possibilities.

- **Why is the project not on GitHub as of yet?**

  The GRINDET-OS' source code is not on GitHub yet; From the investors' perspective it's even better. Why? You may take an MIT

programmer, have him look on Twitter to see what's been worked on since early 2017 and have him watch our devs working on the very thing by correlating Tweets with live-streams recorded on YouTube by date.

Investors want to make money and not to see the product copycated the moment it's uploaded to GitHub. Each and every day *it's not* over there - we're leaving competition behind and making it harder for them to tinker with in the future once it's there. It's like a zero-knowledge-proof... One gets to see the quality, the entire development process ..like sitting at the back of our heads ...one may use everything to give it a test-ride.. Yet still .. one doesn't get to copycat. Now ask yourself ... How many 'crypto projects' on GitHub were not Bitcoin/Litecoin/Ethereum (and other major ERC20 token) clones ? 0.0001%? Why so? Simple. Otherwise it would be too costly and too dangerous.There are now over 20TB of live recordings stored on YouTube. Thanks Google! You may also use the products (the test-net, the mobile apps for both iOS and Android and soon the WebUI). Such an attitude also assures no freerides trying to make money on it during fundraising.

- **What's the main difference between GRIDNET-OS' dark wizards🧙 and the Bitcoin's development team?**

  We do not aim to exploit people's tendency to believe in abstract concepts like bitcoin's developers do so to get traction - inevitably at the cost of innovation.

  The Team is to provide PURE UNPRECEDENTED POSSIBILITIES AND NEVER STOPPING INNOVATION INSTEAD.

  The Wizards🧙 actually *LOVE* what they do. They *BELIEVE* in what they are doing and they *DO* know their stuff. They are self-powered like a fusion reactor. The mixture makes them unstoppable. They could be making millions in so much easier ways. And yet, here they are ~ *Leading all of us towards the Glory of Decentralization* 🙏

- **In short, what will the ICO look like?**
  Automatic uni-swap between Eth/BTC and GNC. The bought asset will be available immediately for trade. There will be an UI dApp available on GRIDNET-OS which will provide an Eth/BTC address for each investor. Read further for more.

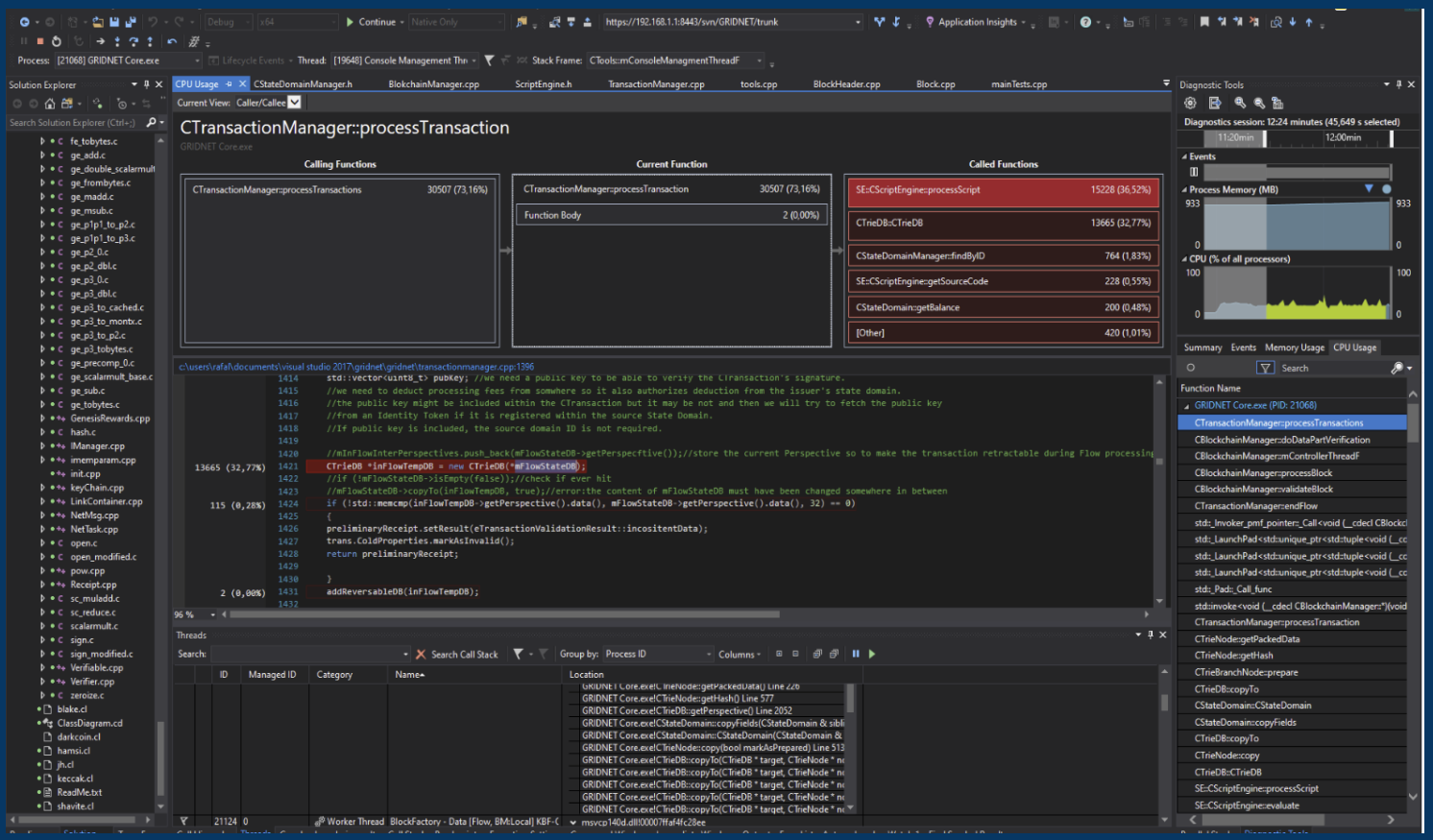
- **Doesn't everyone shill Bitcoin?**

The real question is - should people be shilling Bitcoin just because it has managed to gain traction? Wouldn't it be more justifiable **IF** the Bitcoin's development team **did their due** technological upgrades to secure its widespread usability? (the transactions per second parameter at least).The main trouble is - **a modified bitcoin is bitcoin no more.** And it was supposed to be something special, right? That's their major selling point.

- **So, I have heard that there is going to be a decentralized exchange on top of GRIDNET-OS, is that true?**
  Indeed. Currently we are in no position to give any details, taking into account the OS has not been publicly available as of yet, but it definitely has been on our mind for quite some time now. The exchange is to be implemented as a decentralized UI dApp available on our system.

- **Who are we looking for ?**

Answer: Right now? Hackers and artists.. Fundamentally, most of all and above all - we have been driven by passion. We perceive computer science, programming and development as a form of Art. In fact, some team members actually do paint/draw and they perceive the keyboard as just another tool in their romantic arsenal.



analyzing.. optimizing.. testing..

So, no matter what your field of expertise, if you feel passionate about what you do, and are really really good at what you do; if you do find some common ground with what we strive to achieve then get in touch!

We'll be using resources gathered through the main-round of investment to appropriately reward anyone involved. We have never had and we will never forget about anyone involved. The investments collected from the respected investors during the pre-initial coin offering might had been sparse, there might had been no publications back then, still The Project has faced many transformations and we've always continued to add more, to require more from ourselves and to make **GRIDNET-OS** as exciting and as much at the cutting edge of science as it ever could possibly be. We were and we are constantly taking into account the changing environment and updating our perspective on things.

Why hackers? We've been constantly testing, validating and testing all over again all of the underlying components and still - the potential attack surface for an entirely new blockchain implementation, one with a Turing complete language capabilities is considerably huge. Some hands to help validating all of the potential loopholes are more than welcome at all times.

Why artists? The current GRIDNET.ORG website has not been updated in quite a while; if, it is possible to bring the ideas of freedom, decentralization and openness through to people - it surely could be done through some intriguing, inspiring use of Art.

- What should the artworks look like?

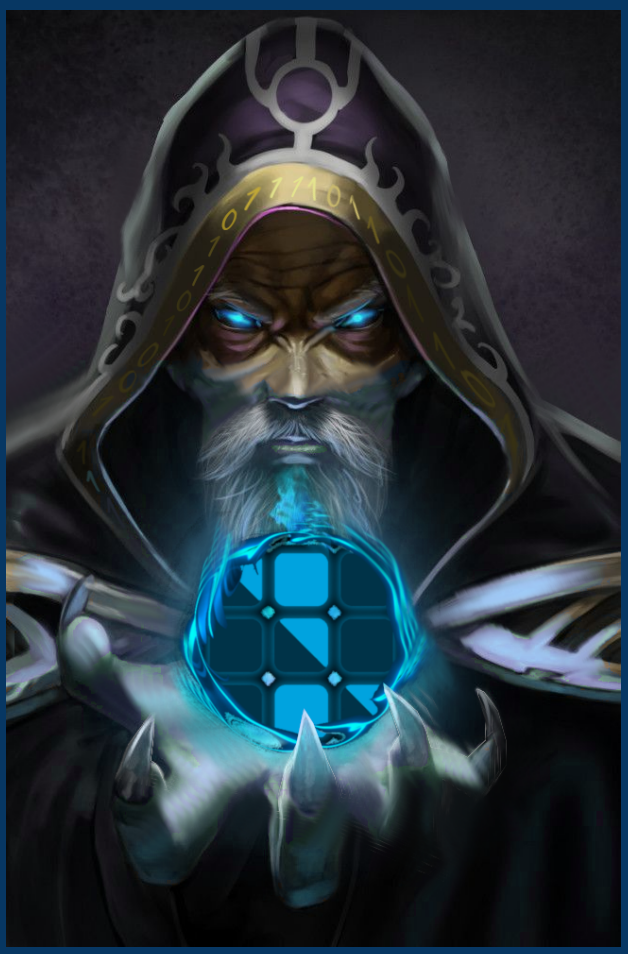
Answer: It would be great if we could work together on the layout of the new-web page, any kind of inspiring art-stuff that could be made part of it or available through the website, twitter tweets etc. Be creative. It's not that the art has to include the logo of The Project - we are after freedom and open-mindness. Gently cyber-punk, intriguing stuff might be always in line with our spirits, still recall it needs to be clean and professional.

- Can I do a bare-bones installation of **GRIDNET**-**OS** on device?
  No. **GRIDNET**-**OS** runs on top of a decentralized State-Machine maintained by a network of computers worldwide. That is why it cannot be installed on a single computer. Still, it can be accessed from almost any kind of device.
- Why do we need this if there's Bitcoin?
  GRIDNET-**OS** covers Bitcoins functionality at an immensely improved performance plus adds zilions of additional functionalities, at same security levels.

- **Why do we need this if there's Ethereum?**

Folks at Ethereum extended the Bitcoin's concept by a lot. Yet nonone knows how to use it. Initially people believed it would provide a multipurpose decentralized computer, what they got is 'a smart thing' perceivably light-years away and used mostly by other projects to deploy new 'tokens'. We are creating true Desktop-like decentralized experience with all use-case scenarios offered by Ethereum covered as well.

- **Why is the majority of *The Team* anonymous? Who are *The Wizards*🧙?**

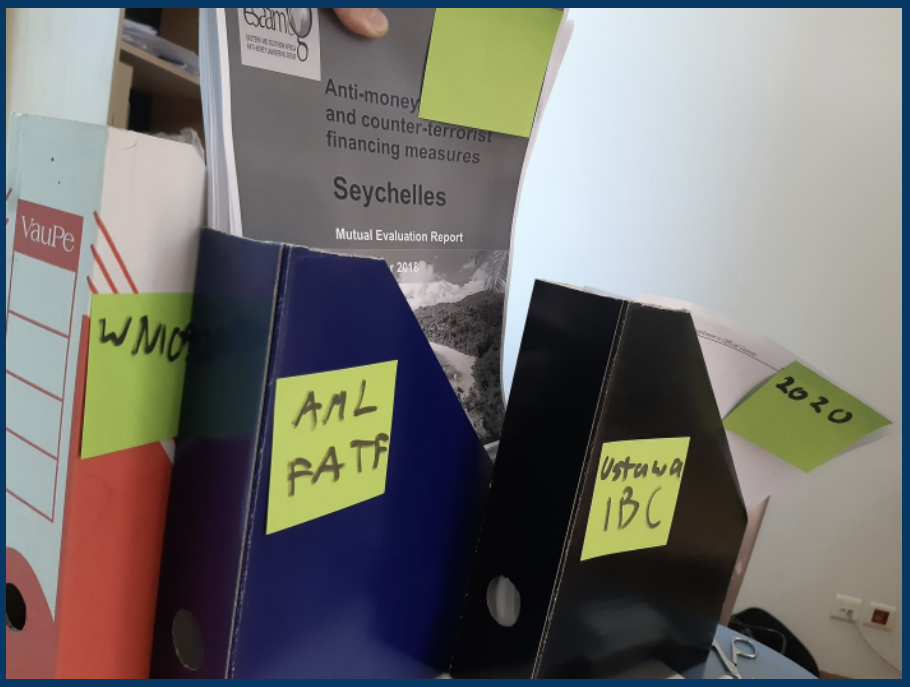


“(..) Thee shall know that Wizards 🧙‍♂️🧙‍♀️(..) No mercy for centralization have they got (..) The guns of cryptography they do wield (..) They're on their way🙏”

The **GRIDNET-OS** Project is a legitimate research project carried out since early 2017. With that said, the majority of The Team is to remain anonymous till the end of time. We live in times where it is better to remain than not to. Keep note that we have designed the system to allow others to remain as such as well. We have certainly earned our reputation and credibility by reporting progress over the years on an hourly basis along with hourly, daily live-programming sessions taking place almost 24/7. If that does not convince you, worry not - we first deliver. With all that said, keep in mind that we all never got to know who Satoshi Nakamoto was in the first place. We are always at your disposal ready to answer and address most sophisticated technical questions publicly almost 24/7. We do not do AMA, as we are always there for you. With developers replying in matters of hours/minutes. Go ahead, check for yourself. In all rationality, if we have managed to pull everything off with the initial outside funding of less than 10 000 USD, back in 2017, just imagine where we are headed.

So when will the main round of funding start? Where will it be held?

Answer: The main round of funding will start shortly after we have a properly validated and working prototype (commonly known as Test-Net) made available to the public - which is - ‘pretty soon’. Follow us on Twitter to stay up-to-date. We have analyzed the political and legal condition regarding Initial Coin Offerings and overall crowdfunding all over the world; and we have chosen Seychelles as a home country of a Limited Liability company which is to facilitate further operations.



We've been working to make sure everything remains in good standing with the current AML/FATF guidelines - for Anti-Money Laundering regulations.We'll be making this as open as possible (you know our spirits;]) - Toward Freedom and Decentralization! Thus, if we are not forced in any way not to let through investors from certain countries as per the Seychelles regulations, we won't be doing anything too limiting; we'll be as liberal as possible. The asset bought through ICO will be tradable right away (so we'll be dealing the asset in real-time). Note that the ‘asset’ in all regards should be treated as a utility backing up the **GRIDNET**-**OS**’s Decentralized Virtual Machine, rather than a security. Right. You'll be able to sell it right away once bought,offsetting the risks as you see fit, which is pretty innovative all in itself. As for the tokenomic the details are yet to be announced. The nature of The Project has changed by a lot since 2017 when we did the pre-ICO. Of course, we’ve discussed these changes with each and every respected pre-ICO investor.

We strive to introduce new exciting things to the project every day. Oftentimes over the weekend, we simply cannot go away from the keyboards. We are more than excited to see the project grow the way it kept growing for the past few years. There has never been a roadmap, we go with the flow. The results you may assess by

yourself. We've got one ultimate goal - to design the most functional, easily approachable, multipurpose decentralized environment on the Planet Earth (and beyond).

With Kind Regards

Rafal Skowronski

Artist, hacker, programmer and
Head of the **GRIDNET**-**OS** Development Team
simply - a human being.

Hans Zimmer - Time
"Do not go gentle into that good night" by Dylan Thomas
Final Masquerade - Linkin Park