



## **White-paper V-1.0**

# TABLE OF CONTENT

## **1. Introduction**

1.1. What is UPGLOBE

1.2. UPGLOBE Use Cases

*1.2.1. Scientific projects*

*1.2.2. Site hosting*

*1.2.3. Game server use-cases*

## **2. UpGlobe technology**

*2.1. IoE, IoT and fog computing*

*2.2. World Computer*

2.3. World Computer SaaS and its API

2.4. Safety and Security

*2.4.1. Safety for miners*

*2.4.2. Dishonest nodes eliminate*

*2.4.3. Safety for buyers*

## **3. References**

# 1. INTRODUCTION

## 1.1. What is UPGLOBE :

UPGLOBE is a decentralised worldwide fog supercomputer for general purpose computing from site hosting to scientific calculations. UPGLOBE is an effective way to solve a worldwide problem - creating a multi-purpose decentralised computational power market.

Unlike widespread centralised cloud services, UPGLOBE project implements a fog computing structure & a decentralised pool of devices, all of which are connected to the internet (IoT / Internet of Everything).

IoT/IoE, as an important part of the available computational power in the world, is one of the key directions of work for the UpGlobe project.

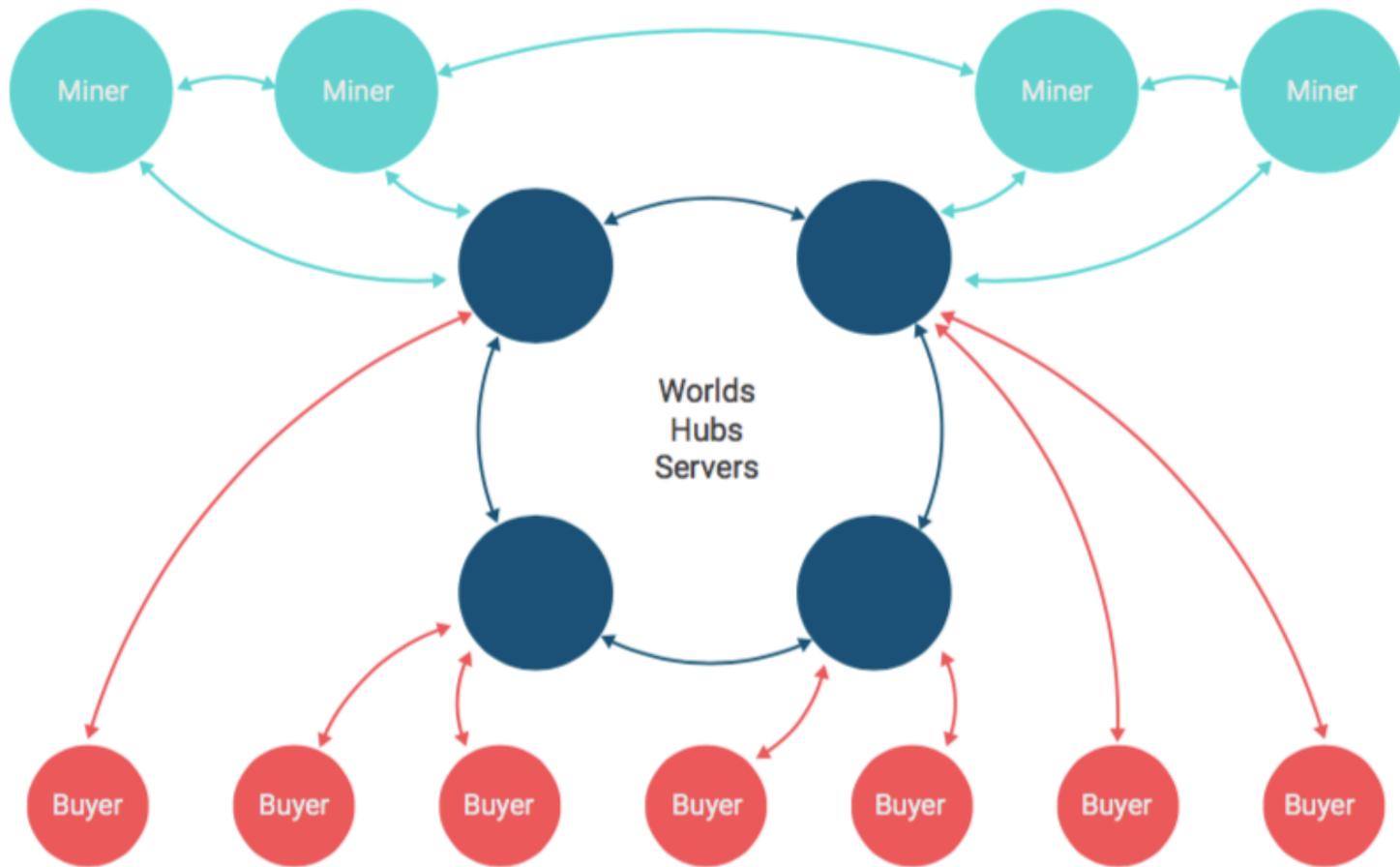
We use cost-efficient fog computing instead of a costly cloud structure, so there is no more need to pay in advance for private and monopolised cloud computing such as with Amazon, Microsoft, Google Cloud, etc. Moreover, since UpGlobe is fully decentralised, there is no single authority that regulates computing resource distribution.

UpGlobe has a hybrid architecture, and therefore supports any kind of computational task without facing Ethereum's out of gas problem.

**From a technical point of view**, UpGlobe is a top layer of underlying P2P technologies & BTSync for data transfer, open source PaaS technology as a decentralised computing platform, and Ethereum Smart Contracts as a consensus system.

There is no central control behind the system and no backdoors or escape hatches. Several existing technologies were combined and modified by our developers to make new technology.

## SCHEME OF THE NETWORK



## 1.2. UpGlobe Use Cases

We have experience with the limitations of BOINC itself – it is scientific software and supports only C++/ FORTRAN/Python, therefore it is not suitable. We started using more advanced solutions like Cocaine and Docker container (which support more languages, including Java, Node.js, Go and etc.) We decided that we will go the other way, and will focus more not just on the distributed calculations eld like BOINC does, but more on fog computing. This way, we can build a more universal platform not only for scientific calculations. The edibility of the UpGlobe platform and its multi-purpose is knit to PoE for non-deterministic task which is a unique technology owned by UpGlobe (Proof of Execution).

### **1.2.1. Scientific projects**

UpGlobe network can be used to run essential scientific calculations requiring massive computing power, for example:

There are several major areas where large computer capacity is needed.

Usually in case of computationally-intensive methods, or (and) tasks with huge input data requirements.

Molecular dynamics and quantum mechanics are examples of such kind of computationally-intensive methods. Molecular dynamics is a computational experiment to study small molecules behaviour and intermolecular interactions. The largest computer power is required to model the interaction between new medicines and biological targets.

### **1.2.2. Site hosting**

The UpGlobe network can be used to host websites without depending on centralised cloud services (AWS / Azure / Google Cloud etc) or hosting providers. We use Cocaine open source PaaS technology to implement virtual machines recognised as servers, with IPFS and other decentralised data storage solutions as an underlying layer.

Website owners can also use our code snippets on their websites to collect payments in UpGlobe or Ether tokens and automatically pay for hosting, according to market value.

It is important to look into TOR operation (The Onion Router). TOR uses pseudo-domains .onion, domain names look like random alignment.

Domain names in the .onion domain are generated based on an open random key server and consist of 16 symbols. These websites are actually not websites at all, but are in fact so-called hidden services. UpGlobe is going to implement such services, of which one application could be hosting websites. Storage and operation will be decentralised.

### **1.2.3. Game server use-cases**

There are lots of MMO games using in-game currencies. Our technology offers a solution for deploying game servers in the UpGlobe network. Furthermore, game currencies can be easily exchanged for UpGlobe tokens and back using our out-of-the-box solution.

In addition, gamers can support their favourite game servers by providing their computing resources in exchange for tokens or in-game currency. In example, Quake dedicated server looks like a fully tuned and settled up, ready-to-play, platform-independent docker container.

# 2. UpGlobe TECHNOLOGY

Nowadays the popular Internet of Things concept(IoT) gives way to the new emerging concept called Internet of Everything (IoE).

Internet of Everything is the combination of all computing resources of humanity. It has core differences with currently widespread centralised cloud computing technology.

In order to develop a system implementing this disruptive idea, the UpGlobe team used the most efficient and proven P2P, distributed computing and blockchain technologies.

UpGlobe is not a monolith product, it's a top layer built on underlying protocols and technologies: Ethereum, BTSync, Docker, Cocaine, etc.

(By the way, Bitcoin creator(s) also combined existing technologies (cryptography, P2P nodes network, git, Proof-of-work concept, etc) to bring a brand new independent decentralised currency/payment system to the world.)

## 2.1. IoE, IoT and fog computing

Before describing the future “World Computer” architecture we need to mention some details regarding IoE, IoT and fog computing concepts.

Nowadays, the concept of Internet of Things (IoT) is commonly known. According to the IoT concept, Thing is any natural or artificial object able to have an IP address and transfer data over the network.

**Internet of Everything (IoE) represents further development of IoT concept:** “Cisco denotes the Internet of Everything (IoE) as the networked connection of people, process, data, and things. The benefit of IoE is derived from the compound impact of connecting people, process, data, and things, and the value this increased connectedness creates as “everything” comes online.

IoE is creating unprecedented opportunities for organisations, individuals, communities, and countries to realise dramatically greater value from networked connections among people, process, data, things.”

This definition emphasises a very important aspect of IoE, which distinguishes IoE from IoT: namely, the so-called “network effect”, formulated by James Macaulay from the Cisco IBSG consulting department. The term “network effect” refers to a decentralisation of organisations included in IoE.

## 2.2. World Computer

The so-called “computing fog” is the layer of computational resources able to process some kind of task.

However, aside from computing fog, the system also involves its users setting computational tasks, and some middleware distributing these tasks among the fog resources, which then returns the result of the calculations.

This system is called “World Computer”.

The first mention of the World Computer term was in Vitalik Buterin’s project Ethereum. It is implemented using blockchain technology’s ability to include executable code into transaction blocks, so every miner’s machine automatically executes this code.

Thereby, Ethereum in fact is the World Computer working like a Turing Machine, with blockchain used as a state register tape.

This also implies that due to the fact that every program must be run on every machine in the Ethereum network, it is very costly and only a limited range of tasks can be run using this platform.

## 2.3. World Computer SaaS and its API

Example of the simple application that can be run:

```
#!/usr/bin/env python

from cocaine.services import Service
from cocaine.worker import Worker

storage = Service("storage")

def process(value):
    return len(value)

def handle(request, response):
    key = yield request.read()
    value = yield storage.read("collection", key)

    response.write(process(value))
    response.close()

Worker().run({
    'calculate_length': handle
})
```

## . 2.4. Safety and Security

In this chapter, we look at security aspects for miners and buyers

### 2.4.1. Safety for miners

**Safety from hostile workloads. Docker isolation.**

One of the docker's software packages is a daemon - which consists of a container server, launched via the "docker -d" command), client tools which permit the user to control the modus and containers directly via the command line interface and an API which permits the user to control the containers via a REST-style program.

### 2.4.2. Dishonest nodes eliminate

The contract also has the Suspected and Punished conditions. In the Registered state – the state when the contract can be registered in the whitelist – the DAO and only DAO can invoke the suspect function, thus setting the contract's stats to suspected – suspected of being malicious. This function blocks all funds on the contract's wallet for 120 days. In the suspected state the following functions can be invoked by the DAO exclusively:

This mean that the only way to hub to get all his money including lockedFunds - is to invoke this function and pay DAO 0.5% of lockedFunds. In any other case 30% from all hub's operation's will be locked on the contract balance.

### 2.4.3. Safety for buyers

**How is ensured to honest buyers their task is being run for given time?**

**The rst method is** by query metrics. Hub should understand how much resourses are consumed by one instance of task, depending on the number (for example) of the input connections. Therefore, Hub can es- timate the one connection value and then estimate work by its number. Of course, it gives just an approximate result and works only with stateless tasks, where users don't transfer the complete data volume into application (or we know in advance the approximate data volume). I.e. relatively talking, it will work for the casual websites and applications, but, for example, for such kind of service as photo and video editing

# 3. REFERENCES

[1] [https://en.wikipedia.org/wiki/Fog\\_computing](https://en.wikipedia.org/wiki/Fog_computing)

[2] [https://en.wikipedia.org/wiki/Internet\\_of\\_things](https://en.wikipedia.org/wiki/Internet_of_things)

[3] [http://internetofeverything.cisco.com/sites/default/files/docs/en/ ioe\\_value\\_at\\_stake\\_public\\_sector%20\\_analysis\\_faq\\_121913\\_nal.pdf](http://internetofeverything.cisco.com/sites/default/files/docs/en/ ioe_value_at_stake_public_sector%20_analysis_faq_121913_nal.pdf)

[4] <https://en.wikipedia.org/wiki/Crypto-anarchism>

[5] <http://internetofthingsagenda.techtarget.com/definition/fog-computing-fogging>

[6] [https://en.wikipedia.org/wiki/Turing\\_machine](https://en.wikipedia.org/wiki/Turing_machine)

[7] <https://users.soe.ucsc.edu/~abadi/Papers/iss02.pdf>

[8] <https://www.cs.bham.ac.uk/~mdr/research/papers/pdf/11-applied-pi.extended.pdf>