CoinFly™
All-in-One for Miners

Platform’s Technical White Paper

Learn how the platform enables Miners to improve their business operations.

This whitepaper provides an introduction to the CoinFly Platform (All-in-One for Cryptocurrency Miners Platform) architecture and explains how Platform uniquely enables miners with zero knowledge and efforts to improve their daily business operations under a holistic service approach. It is intended for people with technological background who want to obtain a thorough understanding of the technical aspects of platform. This paper dives into the technical capabilities of platform and how these capabilities are provided.

White Paper Draft V.2.53
Table of Contents

ARCHITECTURE VISION AND GOALS ................................................................. 4
PLATFORM OVERVIEW .............................................................................. 5
OVERVIEW OF CONFIGURATION ARCHITECTURE ................................ 8
Configuration modeler ............................................................................. 8
Integrated metamode ............................................................................. 9
A. Test Coinfly platform ........................................................................ 9
B. Install COS as RIG's MAIN OS .......................................................... 11
C. Install COS as ASIC's firmware ....................................................... 13
D. Join Coinfly's mining pool ............................................................... 15
Conclusion ............................................................................................. 16
Standard COS facilities ......................................................................... 16
COS-agent facilities ............................................................................... 16
OVERVIEW OF RUNTIME APPLICATION LAYER .................................... 17
Physical view .......................................................................................... 17
Deployment view ................................................................................... 19
Multitenacy ........................................................................................... 20
Microservice driven ............................................................................... 20
USER INTERFACE LAYER ......................................................................... 22
Dashboard .............................................................................................. 22
Account .................................................................................................. 22
Farm monitoring .................................................................................... 24
User profile settings ............................................................................. 26
Intelligent reports ................................................................................ 26
User notifications .................................................................................. 27
OVERVIEW OF AIRM LAYER ..................................................................... 28
OVERVIEW OF API/DATA STORAGE LAYER .......................................... 29
OVERVIEW OF MSA LAYER ...................................................................... 30
TECHNOLOGICAL STACK ........................................................................ 32
APPLICABLE STANDARDS ........................................................................ 34
REFERENCES .......................................................................................... 35
Architecture Vision and Goals

At CoinFly, our mission is to help cryptocurrency miners, experienced ones but also newcomers to industry, to improve their business operations with world-class, services oriented software that they can install and use for their RIGs and ASICs in order to change and innovate the way they are doing mining and making profits of it, with greater speed and flexibility.

This mission is translated into the following set of architecture goals:

<table>
<thead>
<tr>
<th>Platform as a Service on Microsoft Azure Cloud</th>
<th>A Single Point of Administration, Configuration, Monitoring and Reporting of your Mining Process. Results the most comprehensive and simplified way to run your business.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browser-based Access for All Users</td>
<td>Allows any user, from anyplace and at any time to work with Coinfly Platform with just a simple web browser.</td>
</tr>
<tr>
<td>Fast Mining for Technical and Non-Technical Users</td>
<td>Bridges the gap between business and IT by enabling non-technical users to start making profits by simply using Coinfly for RIGS and ASICS in seconds, leveraging the capabilities provided by an easy process wizard.</td>
</tr>
<tr>
<td>Fast Enrollment and Engagement</td>
<td>Allows instant registration and full operational testing for any miner for easy transition and migration to platform.</td>
</tr>
<tr>
<td>Embedded COS for RIGs and COS firmware for ASICS</td>
<td>Enables the best performance of mining hardware by driving GPUs/CPUs and ASICs in more efficient way than their manufacturers.</td>
</tr>
<tr>
<td>Convenient Management for Mining Landscape</td>
<td>Enables the overall management of any size of mining farms, even for manually controlled ones, whereas providing telemetry services for getting and setting rules and commands for any single mining unit.</td>
</tr>
<tr>
<td>Integrated in Blockchain Nodes, Crypto Exchanges, Notification Channels</td>
<td>Enables real time solutions that leverage critical services and products from 3rd parties to permanently feeding the user with strategical information and facts for on-going decision-making mining strategy.</td>
</tr>
<tr>
<td>Integrated in Mining Pool</td>
<td>Enables the participation in CoinFly's mining pool under stratum protocol, by providing mining jobs for mining applications transparently and with high potential revenues, with pay per share model.</td>
</tr>
<tr>
<td>Standards Compliance</td>
<td>Enables easy adoption with all main mining consensus algorithms and protocols, therefore it is reducing time-consuming compliance for miners thus their total cost of ownership.</td>
</tr>
<tr>
<td>Integrated Statistics</td>
<td>Enables complete statistics and oversights for usage, results, alerts, faults and data of all the corresponded platform entities.</td>
</tr>
<tr>
<td>Integrated in International Accounting</td>
<td>Enables under the IFRS standards accounting and compliance for maximum transparency and proof of internal and external transactions.</td>
</tr>
<tr>
<td>Linear Hardware Scalability</td>
<td>Enables use of any kind of mining commodity hardware, thus keeping total cost of ownership low.</td>
</tr>
<tr>
<td>High Availability</td>
<td>Provides high availability for mining-critical systems, indicators and facts.</td>
</tr>
</tbody>
</table>
Platform Overview

Platform delivers the power and flexibility to configure, drive, automate, integrate and manage mining processes across functions, systems, machines and pools. It is giving you the ultimate control to optimize your mining performance and expand its reach, by targeting to profit maxterm and cost minterm. All-in-One CoinFly Platform has been built as a single product. All features are based on one technology, thus simplifying the end-users experience. The diagram below provides an overview of platform's layers and components.

CoinFly is unique in that it has been designed as a single-point platform capable of bridging four different mining needs: Hardware (RIG, ASIC) Initial Configuration, Mining Process Management & Monitoring, Hardware Real-Time Monitoring and Intelligent Decisions & Resources Management during the mining Process ((as it is shown in the Logical View diagram above).

Engagement Layer corresponds to Portal FrontEnd and is the preliminary phase where user has to choose what configuration metamodell is the best choice for his needs. Details about this architecture are described at the Overview of Configuration Architecture chapter.

Runtime Application corresponds to Portal Front-End and it enables through User Interface Layer all the business modules and components for the Mining Process Management & Monitoring and enables the Hardware Real-Time Monitoring. More details about these components are described at the User Interface Layer section of Overview Runtime Application Layer chapter.

Artificial Intelligence Resources management (AIRM) is an inner layer between User Interface and Composite/Integration μServices layer. Physically it belongs to Portal Back-End and is responsible for analyzing and filtering all the asynchronous telemetry indications being published from μSOA to User Interface Layer. It provides live intelligent suggestions to user, thus assisting him to make decisions for beneficial resources and mining management. Hardware protector component supports adequate temperature control of any registered device with self-control reboot functionality to avoid hardware failure.
Component/Integration μServices (μSOA) layer consists of composite services internal and external and is the fundamental platform layer. Those services are using Internal APIs and Data Storages to provide business logic modules and functions. These virtual services are utilizing the decoupled core/atomic microservices from μServices Grid, the lowest physical layer of CoinFly's architecture.

Most processes that span a mining business are at times structured (process) and at other times more ad hoc (case). Being able to manage processes as they go from structured to case and back again supports the reality of business. Companies benefit from the simplicity of a single system to run their mining businesses with the flexibility to do so as they need. Additionally, CoinFly Cloud Service Brokerage is built on top of Platform to provide automated provisioning and metering of services & applications, smooth and consistent interoperability on the cloud.

The first key objective of CoinFly Platform is to simplify the configuration process for mining hardware service with a high level proprietary operating system (COS) based on Linux. This OS includes a smart linux agent which can drive and monitor any mining ASIC. With Integrated-in telemetry support and fast system-info queue, backed up from an extremely fastest column-based data storage is the utmost instrument to command and monitor in real time all your hardware. At chapter Overview of Configuration Architecture will be more emphasis on user's and system acts upon initial engagement phase.

The second key objective is to simplify the participation in CoinFly's mining pool by setting your miners and switching them to our high profitable mining tasks. The AIRM layer calculates different business patterns, taking under consideration your hardware resources additionally to requests and demands for miners and power, and in return suggesting you in real time the best performance for maximizing your profits and extended your hardware's life cycle.

The third key objective is to bring to users transparency. Miners who use Coinfly, are experiencing efficiency and full transparency, of their workers, shares, profits, tasks completed, current hashrates of their RIGs/ASICs, coins, exchanging rates and payments, with full statistics on their dashboard. This is a different approach to the traditional mining process environments, where platforms enable miners to participate but only under a non-transparent and non-intelligent mining models.

Coinfly takes a radically different approach compared to competitors – The model is that Miner is in Charge of his Business and his Resources. The Portal is not only a single point of management but also a valuable instrument to provide the information he needs to make his overall business more proactive and profitable.

The Coinfly Portal is built on top of AIRM layer and SOA layer environments.Telemetry Hub and Cloud Service Brokerage are the pillars of data transactions while microservices grid is the backbone of data calculations. This approach delivers the following benefits:

- **Robust and secure.** All platform's runtime application features provided with high availability, and security, therefore they directly contribute to mining environment.
- **Testable.** Using a LiveUSB flash drive you are able to test the whole platform in advance and without commitments.
- **Available everywhere.** The full functionality of Coinfly Platform is available through a variety of completely browser-based user interfaces. All interactions with Farms, Rigs, Asics, Pools, Coins, Exchanges, Wallets and any other entity are entirely browser-based. It enables quick and hassle-free involvement of new users or for users who decide to migrate to Coinfly, or for non-technical users. The miners' business new participant needs only a web browser – there is no need to install anything locally – and a USB flash as a boot loader with our COS Agent for his assets in case that he is willing to use the advantages of it. Coinfly Platform provides support for the industry’s most popular browsers, Chrome™, Opera®, Microsoft® Edge, Firefox®, Microsoft® Internet Explorer and Safari®.
- **Zero-knowledge configuration deployment.** Participant has to choose between the manual deploy of configuration for his assets, or automatic deploy. The platform has provisions to smart configuration wizard and drives the user to easy deploy the configuration to his Rigs/Asics in order to facilitate this approach.
• **Cloud computing.** In the same way the Coinfly Cloud Service Brokerage solution is built as an application on top of whole Platform. Current provided services and future ones can leverage the Internet and multitenancy features of Coinfly Platform.

• **Telemetry Service Hub.** This solution offers two-way communication and enables us to ingest high volumes of telemetry from mining devices into the cloud for storage or processing. Provides real-time reliable indications, health checks and workload device metrics of any single unit registered to platform. Hence, provides integrated monitoring of hardware's physical states, malfunctioning boards and chipsets, over/under heating fans to be replaced and keeping a registry of issues and spare parts replacements. Information send to user's interface with the view of insights, notifications, alerts, warnings and statistics.

At the same time, it is being used to send a cloud-to-device commands in the form of messages, in order to drive the devices which are plugged into smart outlets. Therefore, it brings to user a power consumption control panel, which is already enhanced from AIRM's feeds and can take actions being triggered from mining strategies based on real facts such as the profitability of crypto mining, the temperature conditions, the chip's frequency (overclocking) and cost of energy consumptions among others.

Coinfly Platform also provides the following basic features:

• **High availability.** Mission critical alerts must always be available and to ensure there is no single point of failure.

• **Scalability.** Miners can handle from a typical RIG to a group of FARMS with thousands of mining devices every day. The platform is built for this task. It scales vertically (scale up), as well as horizontally (scale out). Vertical scalability is accomplished with addition of unlimited quantity of GPU/ASIC to any RIG, while horizontal scalability is accomplished with addition of unlimited quantity of RIGS to FARMS.

• **Multitenancy.** Cloud computing and mining management scenarios demand multiple accounts /miners /farms /organizations, called tenants, who they must share the same infrastructure. Multitenancy is a basic feature of Coinfly Platform that can ensure the privacy and access control of any mining account.

• **Security.** With cybercrime and mining malware being very common, it is crucial to harness the system appropriately. Coinfly Platform has an advanced set of security measures complying fully with security standards, including access control lists, auditing, encryption, authorization, validation, hashing and digital signature algorithms.

• **Microservice orientation.** Microservice-oriented (µService) architecture consists of two services types (functional and infrastructural) and is the predominant design principle for modern enterprise systems suited for smaller and well partitioned, web-based systems in which is mandatory a greater control for those service types mentioned above. µService orientation belongs to the very core of Coinfly's platform. All interactions are done through core/atomic µServices.
Overview of configuration Architecture

This chapter corresponds to H/W Initial Configuration Component which uses the Engagement layer at Portal's Front-end (see COINFLY MINING PLATFORM LOGICAL VIEW diagram). It drills a little deeper into the RIG/ASIC configuration and registration architecture of Coinfly Platform. It introduces the configuration model driven approach and then describes the highlights of the integrated metamodels by providing a user journey for each one of these. A closing section provides an overview of the standard facilities of the COS (Coinfly's Operating System).

Configuration modeler

Engage Process takes a model driven approach to mining device configuration by providing four integrated metamodels through a smart process wizard. A key principle is “Running a mining business shall not be a rocket-science”. All configuration activities are applicable via the ENGAGE workspace, a browser-based integrated environment for fast and easy configuration, which provides tasks, definitions and actions upon user decision.

Users can use their browser and choose one of the 4 main configuration metamodels:

A. To try and test Coinfly platform for mining process and management by using a LiveUSB flash drive.

B. To install the COS at their RIGs, for making it as the main mining operational system.

C. To install COS as a firmware to their ASIC miners and boost the productivity index of their devices.

D. To join Coinfly’s mining Pool and get benefited from the utmost transparency and usage capabilities.
Integrated metamodel

Each of these models is based on a single integrated metamodel. This metamodel does not only trigger the configuration wizard, but also through its visualization interface, notifies and warns the user for critical tasks and finally for the successful configuration completion and so forth. A key objective is to guard consistency between human and system actions while providing assistance to user’s decisions.

A configuration model that successfully passed and applied to any RIG/ASIC should be deployable instantly and should appear to his Dashboard at FARM domain. This is particularly important for start mining immediately after this phase.

In order to achieve that, each metamodel triggers via its class all the necessary microservices at back end which take care of the runtime's architecture. The unique characteristic that is responsible for the multitenancy environment is the FARM id which is being produced cryptographically wised, during user’s registration process. This unique key element is responsible for coupling RIGs/ASICs to specific user account and maintain the consistency with a single view of insights and management during mining runtime. A simplified view of a part of one metamodel looks like this:

User’s journey upon configuration based on our current metamodles is following for each one separately.

A. TEST COINFLY PLATFORM

This approach gives a user the ability to try the platform for his RIGs, simply by using a LiveUSB flash drive. The task based on this particular metamodel demands human and system action. Human is prerequisite task and system is self-automated task. Human task is limited to preparing the LiveUSB flash drive and the BIOS of RIG's motherboard with a USB boot option, and plug the drive to any USB port in order to boot the system intact. System task has been automated into LiveUSB flash drive, so after booting your system with it, the RIG appears as a registered one to user's web-based Farm environment. There is not any kind of permanent installation required in order to start testing the platform. In case that your user decides to turn back to previous regime, he must remove the flash drive during the next reboot process.
It is an optimized wizard task flow in one screen, which in advance requires from user to Login or to Register an account for security reasons. After user's login, the wizard requests from the user to mark the next 5 checkboxes as they appear in the above screenshot, as a written user's consent for his flash drive preparation.

At Engagement process wizard drives user through 6 easy steps, where user and system tasks are intersected to perform the configuration.
Choosing at 1st step to download the COS Flash Maker Utility - an assistance utility based on BALENA ETCHER open source project - for windows or Linux operating system, wizard starts the downloading process to your local drive.

After finishing the download process, then wizard is moving to 02 STEP at which user runs the flash utility and plugs his flash drive to his local computer at any USB port. By choosing the LiveUSB version of COS and selecting his target USB drive, user can start flashing the USB and make it as a bootable medium.

At 3rd step, wizard requires of user to plug the flashed USB drive to his RIG’s USB port.

At 4th step, wizard requires of user to BOOT the RIG from this USB flash DRIVE. In case that the BIOS/UEFI of RIG’s motherboard has not yet been enabled with the option “Boot from USB”, then user must take action and based on the BIOS guide of his RIG must enable this option in order to move forward.

At 5th step, wizard requires of user to wait until the RIG after the its BOOT process will automatically connect to user’s account, being registered under it and finally will appearing in to user’s Dashboard. This process has been fully automated from Coinfly therefore during the next few minutes the RIG will connected and appear, without any other requirements or further human or system tasks being taken.

And finally, at 6th step the wizard just informing the user that the connection has been completed, the RIG is already connected to his account and appeared to his Dashboard, therefore it is ready for start mining process. In case that the RIG will not be appeared in the next 5-10 minutes, then the wizard provides direct support with “CONNECT SUPPORT” button through web-browser. Technical support will be provided to the user immediately, from very experienced tech supporters in order to escalate any issues which may appear upon procedure.

B. INSTALL COS AS RIG’S MAIN OS

This approach gives a user the ability to install the COS as his RIG’s MAIN operating system. This is the best way to use the full functionality of CoinFly's platform by leveraging all the facilities provided from our COS. It is required for a production environment. The task based on this particular metamodel demands human and system action. Human is prerequisite task and system is self-automated task. Human task is limited to preparing the USB flash drive and the BIOS of RIG’s motherboard with a USB boot option, and plug the drive to any USB port in order boot and install the COS to RIG permanently. System task has been automated into USB flash drive, so after booting your system with it, the installation process is up and running until the completion. Procedure usually takes about 1 hour, and at completion system reboots itself and is ready for mining. The RIG appears as a registered one to user’s web-based Farm environment. His installation is permanent and recovery to your old RIG operating system is not provided automatically. In case that user decides to return back to previous OS after the completion, then he must follow the instructions provided form the hardware vendor of the RIG.
It is an optimized wizard task flow in one screen, which in advance it requires from user to Login or to Register an account for security reasons. After user’s login, the wizard requests from the user to mark the next 5 checkboxes as they appear in the screenshot above, as a written user’s consent for his flash drive preparation and permanent COS installation to his RIG.

At Engagement process wizard drives user through 6 easy steps, where user and system tasks are intersected to perform the configuration.

Choosing at 1st step to download the COS Flash Maker Utility — an assistance utility based on BALENA ETCHER open source project — for windows or Linux operating system, wizard starts the downloading process to your local drive.

After finishing the download process, then wizard is moving to 2nd step at which user runs the flash utility and plugs his flash drive to his local computer at any USB port. By choosing the Full version of COS and selecting his target USB drive, user can start flashing the USB and make it as a bootable medium.

At 3rd step, wizard requires form user to unplug the USB flash drive from his current PC and plug the flashed USB drive to his RIG’s USB port. There is a WARNING notice that otherwise a user can face unpredictable situation with an unwanted installation of COS to his local PC instead of his RIG, with a permanent LOSS of his OS and data.

At 4th step, wizard requires from user to BOOT the RIG from this USB flash DRIVE. In case that the BIOS/UEFI of RIG’s motherboard has not yet been enabled with the option “Boot from USB”,
then user must take action and based on the BIOS guide of his RIG must enable this option in order to move forward.

At 5th step, wizard requires form user to wait until the RIG after its BOOT process will automatically connect to user's account, being registered under it and finally will appear in the user's Dashboard. This process has been fully automated from Coinfly therefore during the next few minutes the RIG will be connected and appear, without any other requirements or further human or system tasks being taken.

And finally, at 6th step the wizard just informing the user that the connection has been completed, the RIG is already connected to his account and appeared to his Dashboard, therefore it is ready for start mining process. In case that the RIG will not be connected in the next 5-10 minutes, then the wizard provides direct support with “CONNECT SUPPORT” button through web-browser. Technical support will be provided to the user immediately, from experienced tech supporters in order to escalate any issues which may appear upon procedure.

C. INSTALL COS AS ASIC’S FIRMWARE

This approach gives the user the ability to upgrade his ASIC's firmware with COS. This is the best way to use the full functionality of COINFLY's platform by leveraging all the facilities provided from our COS firmware. It is required for a production environment. The task based on this particular metamodel demands human and system action. Human is prerequisite task and system is self-automated task. Human task is limited to download the proper firmware file to your local drive and flash it to your ASIC by choosing the SYSTEM/UPGRADE FRIMWARE option via ASIC's Control Panel. System task has been automated into firmware file, so after flashing your ASIC and rebooting it, new firmware will continue the rest of the system process until the completion. Procedure usually takes few minutes and at completion ASIC is ready for mining. The ASIC appears as a registered one to user's web-based Farm environment. It is a permanent installation and recovery to your previous firmware is not provided automatically. In case that user decides after the completion to return back to previous firmware, then he must follow the instructions provided from the hardware vendor.

It is an optimized wizard task flow in one screen, which in advance it requires from user to Login or to Register an account for security reasons. After user's login, the wizard requests from the user to mark the next 5 checkboxes as they appear in the screenshot above, as a written user's consent for downloading and flashing with the proper COS firmware file his ASIC.

At Engagement process wizard drives user through 5 easy steps, where user and system tasks are intersected to perform the configuration.
Choosing at 1st step to download the appropriate firmware file based on chosen ASIC model, wizard starts the downloading process to your local drive.

After finishing the download process, then wizard is moving to 2nd step at which user must access his ASIC's control panel usually by using his IP address via his web browser address bar.

At 3rd step wizard requires from user to choose through ASIC's control panel menu the SYSTEM → Upgrade Firmware option and start the flash firmware process, by browsing the new downloaded firmware file to his local disk path.

At 4th step wizard informs user that after the successful flashing of COS firmware, the ASIC will reboot itself, and the process will continue automatically. Wizard also warns the user that in case of electricity outage or any other inappropriate action which can cause flashing error or incompetence of the current process, user must recover ASIC's firmware in his own pace.

At 5th step, wizard requires of user to wait until the ASIC will finalize its BOOT process and automatically connect to user’s account, being registered under it and finally will appearing in to user’s Dashboard. This process has been fully automated from Coinfly therefore during the next few minutes the ASIC will be connected and appear, without any other requirements or further human or system tasks being taken.

And finally, at 6th step the wizard just informing the user that the connection has been completed, the ASIC is already connected to his account and appeared to his Dashboard, therefore it is ready to start mining process. In case that the ASIC will not be appeared in the next 5-10 minutes, then the wizard provides direct support with “CONNECT SUPPORT” button through web-browser. Technical support will be provided to the user immediately, from experienced tech supporters in order to escalate any issues which may be appear upon procedure.
D. JOIN COINFLY’S MINING POOL

This approach gives the user the ability to upgrade his ASIC’s firmware with COS. This is the best way to use the full functionality of COINFLY’s platform by leveraging all the facilities provided from our COS firmware. It is required for a production environment. The task based on this particular metamodel demands human and system action. Human is prerequisite task and system is self-automated task. Human task is limited to download the proper firmware file to your local drive and flash it to your ASIC by choosing the SYSTEM/UPGRADE FRIMAWARE option via ASIC's Control Panel. System task has been automated into firmware file, so after flashing your ASIC and rebooting it, new firmware will continue the rest of the system process until the completion. Procedure usually takes few minutes and at completion ASIC is ready for mining. The ASIC appears as a registered one to user’s web-based Farm environment. It is a permanent installation and recovery to your previous firmware is not provided automatically. In case that user decides to return back to previous firmware after the completion, then he must follow the instructions provided from the hardware vendor.

Second section is the input of user's crypto wallet public address to which Pool must deposit his rewards.
And at third section user can find the proper connection settings for his RIGs/ASICs based on stratum & tcp protocol and pool address with the concrete algorithm for specific coin.

**Conclusion**

Configuration Architecture provides a set of metamodels for user by offering a strong foundation for fast mining setup, “headache-free” mining equipment configuration and finally an efficient real-time management & driving mechanism for each mining device. Using Coinfly metamodels A, B & C, user actually enables the standard COS facilities by applying them to his farm/devices.

**Standard COS facilities**

The COS is a server-client software which provides a set of standard facilities for all the configuration metamodels. It is an ubuntu based operating system dedicated to Rigs/Asics hardware. Works on PC RIGs (GPU or CPU). It contains enhanced GPU drivers, miners, scripts and COS-Agent. The last is a client-side application which is running in mining equipment and provides a list of facilities to the platform being connected to COS Service, a Server for all COS-Agents.

**COS-AGENT FACILITIES**

- Obtain system information including hardware and software telemetry, from the mining devices and forwarding them to COS-Service (Server side) in real time.
- Receives hardware and mining commands from COS-Service in real time for configuration adjustments after the initial configuration procedure.
- Controls miner application based on users decisions based on Process intelligence patterns. (Launch, Stop, Restart)
- Changes GPU drivers settings based on user’s decisions and Process intelligence patterns. (overclocking, fan speed)
- Manage local stored configurations for each device.
- Takes action for miner or hardware, according to system state and telemetry indications.
Overview of Runtime Application layer

This chapter drills a little deeper into runtime application of Coinfly Platform. It introduces the microservices driven approach and provides their logic view as well as their anatomy.

Physical view

A physical view of the Process Platform architecture has a strong resemblance with the COINFLY MINING PLATFORM LOGICAL VIEW illustration provided before, although there are a few differences per layer, that do justice to the technical aspects.

Each layer is described as follows:

1 User Interface Layer

The User Interface Layer contains all the User Interface components, such as dashboard, account, farm, reports, settings, notifications and engage which was described at overall of configure architecture section. These user interfaces are built on top of the AIRM layer as defined in the next layer.

2 Artificial Intelligence Recourses Management Layer

This layer hosts commands and sensors as data translators from the interface layer and uses a cognitive agent with perception of causes and reasons to send action behaviour patterns back to it.
3 Internal APIs/Data storages Layer

This layer hosts all the reactive end point APIs and Data storages, relevant to the business domain; notable examples being account, security, statistics, transactions, web services form third external vendors. These services are all built on top of the μServices layer.

4 Services Grid Layer

The Services Layer provides the fundamental μMSA grid functionality used by the other layers.
Deployment view

All runtime components are linked through the μMSA Grid. The μMSA grid provides three main facilities.

1 Routing of service’s messages

The services deliver their messages to the Cloud Service Brokerage (CSB) for distribution and guaranteed delivery. Given the required quality of service and whether to use a reliable transport, it chooses a channel and delivers the message to the recipient. The messages can be transferred over a variety of protocols, ranging from plain TCP/IP sockets to message queues.

2 Load balancing

As the load increases, not everything can be handled on a single system, so multiple systems might run the same service, thus sharing the load. The CSB has pluggable load balancing algorithms to decide which service instance to address.

3 Failover

If one of the service instances fails, load should immediately be moved to other instances and business should go on as usual. This is controlled by the failover features of the CSB.

CLOUD SERVICE BROKERAGE

All participants on the bus are equal, so there is no central bus coordinator or single point of failure. The service icons in the diagram represent a random set of the service groups in to Platform. A service group denotes a conceptual service. The actual implementation is done through a service container. To provide load balancing and fast failover one service group might be implemented through multiple service containers, most of the time running on different systems.
The previous illustration gave an abstract representation of the relationship between services. The following illustration provides a concrete view of the interaction between services in an example deployment configuration of the COS Service with telemetry hub and statistics api.

**Multitenancy**

The Process Platform SOA Grid represents the physical or deployment aspect of the Runtime architecture. The Farm Id notion represents a logical concept in the Runtime Platform architecture. All functionality is invoked in the context of a user and his Farm Id.

The Farm Id is not really located on a particular node. All service containers, wherever located, can execute the functionality on behalf of a user in the context of that Farm Id. So, if a user starts a user interface it will be in the context of a Farm Id of which the user is the owner.

And if the user interface invokes a call on a service container, it will execute in the context of that Farm Id and that particular user. Before it is executed, the access credentials of user are validated against the Access Control List of the service. If the user does not have the required authorization, the logic will not be executed.

A single user is bonded only to a single Farm Id. He can’t have access rights or being bonded with another Farm Id. Service containers exist in the context of a Farm Id. The functionality in a service container is exposed via REST interface.

When a REST call is initiated, it is always done in the context of a Farm Id. For efficiency and reuse, a called service acts as a fall-back mechanism for Farm Ids. If the service is not implemented for the current Farm Id, the call is delegated to the service container and executed there, but still in the context of the invoking user and Farm Id.

**Microservice driven**

For the best user experience and interface these services are interacting each other in the form of components. Components are simply a grouping of related functionality, encapsulated behind this well-defined interface. The diagram below captures a complete overview of the platform’s architecture, the service containers, their components and their interactions, the data flows and its direct path in between user, devices and third party web-services.
CoinFly components
User interface layer

Users interacting with the Coinfly platform through a single point component inside Portal Frontend service container. Frontend is the user interface from where user can control all the aspects of mining process primarily at configuration stage, and secondary during the runtime mining process. It relies on the PORTAL Back End container, and interacts with its microservices and their API points by fact and on demand. All the business logic is appearing into user’s interface as individual workspaces on horizontal main menu.

Dashboard

It is an overview of statistics and insights, concerns the whole FARM. It represents graphically all the RIGS, all the GPUs, the current Hash Rate and the energy consumption. Additionally, it provides statistics for the mining workers, the daily revenue of user’s Farm, plus with a chart graph visualization the Coinfly pool and the telemetry of the devices sorted for the last week or current 24 hours of process.

Account

A section which reflects with high detail, all the financial figures for user’s account. It is divided in to five logical financial categories of transactions. It uses Portal backend components and microservices such as: Account.Api, Coins.API, Transactions.API, Transactions service, Statistics. API, Statistcs, SmartConfiguration service and 3rd party web services from exchange and blockchain nodes.
- Accounts (Details about user’s crypto coins and their current balances)

- Withdrawals (Details about user’s withdrawals requests and a scheduler for automation).

For security reasons all withdrawals must be confirmed through email exchange. Coinfly’s security team will get the email confirmation and will proceed with the withdraw to your desired crypto wallet.

All crypto wallets are living outside of our platform, therefore we are not compromising their security, as far as we are not keeping their balances in to our storage.
Transactions (All the transactions for coins, shares, profits, transfers, rewards and withdrawals as a full ledger).

<table>
<thead>
<tr>
<th>Date</th>
<th>Wallet</th>
<th>Coin</th>
<th>Amount</th>
<th>Fee</th>
<th>Type</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.09.2019</td>
<td>Main</td>
<td>ETH</td>
<td>0.210319322668645</td>
<td>0</td>
<td>MiningReward</td>
<td>Successful</td>
</tr>
<tr>
<td>26.09.2019</td>
<td>Main</td>
<td>ETH</td>
<td>0.12327316050817</td>
<td>0</td>
<td>MiningReward</td>
<td>Successful</td>
</tr>
<tr>
<td>27.09.2019</td>
<td>Main</td>
<td>ETH</td>
<td>0.077312984880257</td>
<td>0</td>
<td>MiningReward</td>
<td>Successful</td>
</tr>
<tr>
<td>27.09.2019</td>
<td>Main</td>
<td>ETH</td>
<td>0.116031499682333</td>
<td>0</td>
<td>MiningReward</td>
<td>Successful</td>
</tr>
<tr>
<td>28.09.2019</td>
<td>Main</td>
<td>ETH</td>
<td>0.107679108400426</td>
<td>0</td>
<td>MiningReward</td>
<td>Successful</td>
</tr>
<tr>
<td>28.09.2019</td>
<td>Main</td>
<td>ETH</td>
<td>0.0801946825090052</td>
<td>0</td>
<td>MiningReward</td>
<td>Successful</td>
</tr>
<tr>
<td>29.09.2019</td>
<td>Main</td>
<td>ETH</td>
<td>0.108319906232842</td>
<td>0</td>
<td>MiningReward</td>
<td>Successful</td>
</tr>
<tr>
<td>29.09.2019</td>
<td>Main</td>
<td>ETH</td>
<td>0.0810239891357797</td>
<td>0</td>
<td>MiningReward</td>
<td>Successful</td>
</tr>
</tbody>
</table>

Wallets (All the wallets user wants to add and use for any mining worker, except that these wallets can also be used as withdraw accounts). In this section CoinFly user, provides only his public key of crypto wallets such as Bitcoin, Ethereum and others. CoinFly does not store or operate internal crypto wallets, for high secure of their crypto balances. Wallets living at their native blockchain protocols.

Farm monitoring

The main component where the user has a single view of his mining landscape. It provides functionality for controlling and managing RIGs/ASICs. Supports bi-directional communication with each single device, with any configured and registered RIG/ASIC. It is fetching and representing the current state and telemetry indications of any device.
It also supports actions massively for a group of chosen devices which is very important for the user especially when it is required to handle large quantities of devices at once. It utilizes mainly the FARM.API and STATISTICS.API from the PORTAL Backend, and relies on Telemetry hub for fetching indications, and on cloud brokerage for interacting with the MSA grid for sending commands and settings to mining devices or to miners via queue middleware. Main tasks that user can execute via FARM workspace and interact with his devices are:

- Rename TAG of any device/s
- Change settings of any device/s
- Start mining process for any RIG/s
- Stop mining process for any RIG/s
- Restart mining process for any RIG/s
- Manage/Change mining profile
- Manage/Change hardware profile
- Reboot any system/systems
- Halt any system/s
- Open SSH terminal with any system
- Delete any device, RIG, ASIC.
User profile settings

The Security and Profile settings component provides to user a workspace form where he is able to manage his Coinfly account. It utilizes the mainly the SECURITY.API from Portal Backend, and feeding the SmartConfiguration service with information about coins and wallets as well as the notification preferences. Main tasks that user can execute via Settings workspace are:

- User Authentication (2FA settings)
- Password Management (Change password)
- Farm ID (Get his unique ID for using it for manual configuration)
- Sets the notifications preferences (Telegram, Email and In-Portal Notifications)

Intelligent reports

The component which provides analytics of Coinfly's mining Pool and Telemetry Indications. It utilizes mainly the STATISTICS.API, the Telemetry.HUB, the COIN.API and relies on Transactions service for fetching data for the Mining Pool activity.

Main categories of analytics that user can preview via Reports workspace are:

- General Analytics from Coinfly's mining pool. (From date to date complete historical archive for estimated daily and weekly revenue, the balance of each coin, and the aggregated coin reward amount from the pool)
- Analytics for Workers. (A detail hasrate for each worker per RIG, the total shares and the last share. The diapason can be instant, last 24 hours, Period (from – to), per each coin individually.)
Rewards (Analytics sorted per day for the amount that user has been rewarded from Pool, and the fees that user paid to Coinfly).

BI tools can use this API for extracting, transforming and loading data to their OLAP data lakes. This approach enables enterprises to create detail out-of-the-box dashboards and real-time indicators reflecting any kind of data dimension.

So far, Coinfly is the only All-in-One platform for mining business, which provides a full historical report of awards of mining coins, available in IFRS (International Financial Report Standard) for full compliance with usual accountant's requirements. This approach gives the maximum transparency for mining business indifferently of legal entity's location and governmental tax framework.

User notifications

The component which provides in-Portal notifications based on user preferences and settings. It utilizes mainly the Notifications.Hub from Portal Backend, and relies on notifications queue which delivers any kind of notification via the notifications.robot to each independent notification.channel. Currently we are supporting notifications to email and to telegram accounts additionally to built-in Portal notification workspace.
Overview of airm layer

This is a lower level layer, which interacts between the User Interface Layer and the MSA grid of services using APIs/Data Layer feeds.

Based on AI built-in algorithm its main key objective is to perform analysis of all the indications internal and external from/to platform in order to recommend and suggest to user modifications and micromanagement where this is considered appropriate especially for device management, farm and mining management, and finally for the overall profitability.

Collecting and analyzing all possible indications which telemetry.API provides along with the feeds from external 3rd party web-services. At output, generates intelligent decisions, forming those as human readable notifications and notifying user about.

Assists user to make strategic business decisions for these usual and common wonderings:

- What is the most profitable coin for mining?
- When mining becomes not profitable/more profitable?
- When hardware must be switch to maintenance or to safe mode?
- When the exchange coin rates are more advantageous?

AIRM’s logical component of most profitable coin, taking under consideration in real-time factors such as Network’s current complexity and difficulty, current type of mining equipment registered to the farm along with the current exchange rate of the mining coin. Based on these calculations the component always suggesting the user the best choice for him in order to maximize profits from his resources.

The Cost Energy Saver component always suggests the user the best time to stop or start mining based on overall energy consumption in order to avoid huge electricity costs and increase the profits/direct expenses rate to maximum.

Notice that platform also supports manual hardware configuration, for users who decide to use the mining pool service only and preferred to drive their hardware outside of CoinFly’s operating system and its agent.

If user will choose the manual RIG/ASIC configuration option, without any installation of COS/ COS Agent to his RIG/ASICS, then this Layer becomes unavailable and user can’t benefit from its intelligent suggestions and decisions. Therefore, prerequisite for enabling and utilize the AIRM layer is the completion of initial H/W configuration as it described at corresponded chapter.

We strongly recommend that all users, experienced or not, to configure their devices with our proprietary operating system and its agent in order to get the maximum functionality that Coinfly provides.
Overview of apis/data storage layer

**STATISTICS.API**
Provides statistics data for charts, datasheets and summaries on demand

**STATISTICS.HUB**
Same as Statistics.API but stands for delivering statistics data on update

**STATISTICS.DATA**
The main storage with all statistical dimensions and facts about shares provided from Mining Pool.

**TELEMETRY.HUB**
Push service to deliver Telemetry data updates to the portal

**FARM.API**
Set of services to provide rigs data and accept commands and configurations from user

**FARM.DATA**
The main storage of Farm’s data, including commands being registered and implemented to mining devices via Smart Configuration Service.

**ACCOUNT.API**
- Provides account balances data
- Operates user’s wallets
- Operate withdrawals to user’s wallets

**ACCOUNT.DATA**
The main storage of account’s data, including blockchain registered transactions, shares which corresponded to these transactions from Mining pool and data from exchanges about coins and rates.

**TRANSACTIONS.API**
Provides transactions information (mining rewards, withdrawal, other transfers)

**SHARES.DATA**
The main storage of Mining Pool with the shares per each farm Id.

**COINS.API**
Provides coins information including exchange rates and blockchain networks statistics.

**NOTIFICATIONS.HUB**
Stands for notifications delivery to the portal on update.

**THIRD PARTY WEB-SERVICES INTEGRATIONS**

**BLOCKCHAIN NODES**
Each coin supported at mining pool requires a blockchain node instance.
- Obtain network data (transactions / stats)
- Send mined blocks to the network
- Operate with transactions (send withdrawals / receive deposits)

**EXCHANGES**
Integration points with crypto exchanges. Collect coins rates.

The problem that user is usually facing after crediting his mining coins is divided into three separate problems: choosing the optimal storage location for those cryptocurrencies that he does not intend to sell yet, choosing a method of delivery of cryptocurrencies to the point of sale, and finding the best rate.

Coinfly is solving all those three problems by using third party trusted crypto exchange service. While pool is crediting the mining shares to each user’s wallet, the platform indicates if there is any other desirable cryptocurrency for this coin, and if so then immediately and “on the fly” the exchange will take place and will convert at the best market’s rate the mining coins to the desirable ones and transfer them to user’s crypto wallet.

Coinfly’s pool supports several coins for mining, but has the advantage through this service to provide to user unlimited available crypto currencies. User doesn’t need to open any kind of exchange account, to pass any kind of KYC procedure or to transfer coins for trade and exchange along with loosing valuable time to follow and recording the current exchange rates and trying to guess the best one.
Overview of MSA Layer

In CoinFly’s MSA layer we are using decentralized logic and governance. Here are the key aspects of implementing this layer.

- Each microservice has private database to persist the data that requires to implement the business functionality offered from it.
- A given microservice can only access the dedicated private database, but not the databases of other microservices.
- At business scenario when there is a need for several databases update, from a single transaction, then service.Apis undertake this responsibility, instead of direct access to database from different microservices.

In our microservices architecture each service is a self-contained entity therefore we have to deal with many services to fulfill a business function. Their location is changing dynamically because of rapid and agile development/deployment nature of them. Therefore in order to find their locations, we are using a service registry.

Service Registry holds the microservices instances and their locations. Microservice instances are registered with the service registry on startup and deregistered on shutdown. The consumers can find the available microservices and their locations through service registry. To find the available microservices and their location, we are using a client-side discovery service mechanism.

SECURITY SERVICE
Service for securing User’s authorization and access credentials to platform.

SMART CONFIGURATION SERVICE
An AI service generating mining and hardware configurations and commands for mining rigs according to market situation, blockchain network stats and hardware specifics.

TRANSACTIONS SERVICE
Operate internal transactions:
- deposit rewards according to mined shares
- internal transfers
- conversions and exchanges
Operate external transactions:
- withdrawals (from system to blockchain wallet)
- deposits (from blockchain to system)
QUEUE MIDDLEWARE
The communication between the consumer/producer microservices is facilitated through this message broker, which is based on asynchronous messaging standards. With this middleware we are providing scalability and also, we are improving reliability and flexibility for current and future services.

COINS DATA SERVICE
Provides coins rates collected from external web services from FIAT-to-CRYPTO exchanges.

BLOCKCHAIN DATA SERVICE
Provides blockchain network stats to system, in order to justify the mining blocks, the fees and rewards of them, for each coin.

STATISTICS SERVICE
Aggregates mining statistics data, being fed from shares and their calculations of Coinfly’s mining pool.

NOTIFICATION SERVICE
Generates user notifications according to system events and stats.
Notifications robot: Delivers user notifications to external channels.
Notifications channels: Integration points to external notifications channels such as messengers, email, etc.

MINING POOL SERVICE
Works as server for miners. It operates with blockchain nodes, obtaining data to provide jobs for miners. Mining pool accepts Hashes from miners with less difficulty than required by network (Shares). This allows mining pool to record all miners work to spread rewards.
Mining pool sends blocks signed with hash of the appropriate network difficulty to the blockchain node.

COS SERVICE
Works as a “server side” for COS-Agents. It enables direct client-to-microservice communication, where “client” is the COS Agent, installed on mining devices. This communication is bi-directional as it provides telemetry indications from any device back to service. At the other direction, provides a forwarding action for any command being sent via user interface to the service with end deliver to the device where the agent runs.
## Technological Stack

<table>
<thead>
<tr>
<th>Technology</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kubernetes</td>
<td>Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.</td>
</tr>
<tr>
<td>.Net Core 3.0</td>
<td>.NET Core is a free and open-source, managed computer software framework for Windows, Linux, and macOS operating systems.</td>
</tr>
<tr>
<td>C++ (std17)</td>
<td>C++17 is a revision of the ISO/IEC 14882 standard for the C++ programming language.</td>
</tr>
<tr>
<td>ZeroMQ</td>
<td>ZeroMQ (also spelled ØMQ, 0MQ or ZMQ) is a high-performance asynchronous messaging library, aimed at use in distributed or concurrent applications.</td>
</tr>
<tr>
<td>Thrift</td>
<td>Thrift is an interface definition language and binary communication protocol used for defining and creating services for numerous languages. It forms a remote procedure call (RPC) framework for «scalable cross-language services development».</td>
</tr>
<tr>
<td>Kafka</td>
<td>Kafka is an open-source stream-processing software platform. Provides a unified, high-throughput, low-latency platform for handling real-time data feeds. Can connect to external systems (for data import/export) via Kafka Connect and provides Kafka Streams.</td>
</tr>
<tr>
<td>Clickhouse</td>
<td>ClickHouse is an open-source column-oriented DBMS (columnar database management system) for online analytical processing (OLAP).</td>
</tr>
<tr>
<td>MongoDB</td>
<td>MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schema.</td>
</tr>
<tr>
<td>Redis</td>
<td>Redis (Remote Dictionary Server) is an in-memory data structure project implementing a distributed, in-memory key-value database with optional durability. Redis supports different kinds of abstract data structures, such as strings, lists, maps, sets, sorted sets, HyperLogLogs, bitmaps, streams, and spatial indexes.</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>PostgreSQL, also known as Postgres, is a free and open-source relational database management system (RDBMS) emphasizing extensibility and technical standards compliance. It is designed to handle a range of workloads, from single machines to data warehouses or Web services with many concurrent users. PostgreSQL features transactions with Atomicity, Consistency, Isolation, Durability (ACID) properties, automatically updatable views, materialized views, triggers, foreign keys, and stored procedures.</td>
</tr>
<tr>
<td>Angular 8</td>
<td>Angular 8 is a client-side TypeScript based framework which is used to create dynamic web applications. It is very similar to its previous versions except having some extensive features.</td>
</tr>
<tr>
<td><strong>Bootstrap</strong></td>
<td>Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation and other interface components.</td>
</tr>
<tr>
<td><strong>HTML5+SCSS</strong></td>
<td>HTML5 is a software solution stack that defines the properties and behaviours of web page content by implementing a markup based pattern to it. Sass is a pre-processor scripting language that is interpreted or compiled into Cascading Style Sheets (CSS). Sass Script is the scripting language itself.</td>
</tr>
<tr>
<td><strong>Electron</strong></td>
<td>Electron is an open-source framework which allows the development of GUI applications using web technologies. It combines the Chromium rendering engine and the Node.js runtime.</td>
</tr>
<tr>
<td><strong>Azure DevOps</strong></td>
<td>Azure DevOps is a Software as a service (SaaS) platform from Microsoft that provides an end-to-end DevOps toolchain for developing and deploying software. It also integrates with most leading tools on the market and is a great option for orchestrating a DevOps toolchain.</td>
</tr>
</tbody>
</table>
## Applicable Standards

### WEB SERVICES AND INTERNET STANDARDS

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSS3</td>
<td>Style sheet language to describe the look and feel of an HTML document</td>
<td>2.1</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol is the method used to transfer information over the Internet</td>
<td>1.1</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Combination of a normal HTTP interaction over an encrypted secure socket layer (SSL) or transport layer security (TLS)</td>
<td>1.1</td>
</tr>
<tr>
<td>HTML5</td>
<td>A standard for definition of the properties and behaviours of web page content by implementing a markup based pattern to it.</td>
<td>5.2</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation (JSON) is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute-value pairs and array data types (or any other serializable value).</td>
<td>5.0</td>
</tr>
<tr>
<td>JavaScript (ECMAScript 5, ECMAScript 2015)</td>
<td>ECMAScript (or ES) is a scripting-language specification standardized by Ecma International in ECMA-262 and ISO/IEC 16262. JavaScript has remained the best-known implementation of ECMAScript since the standard was first published.</td>
<td>2015</td>
</tr>
<tr>
<td>WebSocket</td>
<td>WebSocket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection.</td>
<td></td>
</tr>
</tbody>
</table>

### SECURITY STANDARDS

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL / TLS</td>
<td>Commonly-used protocol for managing the security of a message transmission on a network</td>
<td>3.0/1.2</td>
</tr>
<tr>
<td>ASVS</td>
<td>Security standard OWASP Application Security Verification Standard (ASVS) Project provides a basis for testing web application technical security controls and also provides developers with a list of requirements for secure development.</td>
<td>V4.0.1</td>
</tr>
<tr>
<td>ISO/IEC 25001</td>
<td>Specification provides requirements and recommendations for an organization responsible for implementing and managing the systems and software product quality requirements specification and evaluation activities through the provision of technology, tools, experiences, and management skills.</td>
<td>25001</td>
</tr>
<tr>
<td>PCI DSS</td>
<td>The Payment Card Industry Data Security Standard (PCI DSS) is an information security standard for organizations that handle branded credit cards from the major card schemes.</td>
<td>3.2.1</td>
</tr>
<tr>
<td>2FA</td>
<td>Security standard Two-factor authentication (2FA) is a way to add additional security to your account. The first «factor» is your usual password that is standard for any account. The second «factor» is a verification code retrieved from an app on a mobile device or computer.</td>
<td>1.1 *</td>
</tr>
</tbody>
</table>

*Note: The numbers in parentheses refer to the version of the standard.*
References

01 https://en.wikipedia.org/wiki/Cryptocurrency#Mining
02 https://en.bitcoin.it/wiki/Mining_rig
03 https://en.bitcoin.it/wiki/ASIC
04 https://en.bitcoinwiki.org/wiki/Mining_farm
05 https://en.wikipedia.org/wiki/Mining_pool
06 https://en.wikipedia.org/wiki/Blockchain
07 https://en.wikipedia.org/wiki/Platform_as_a_service
08 https://en.wikipedia.org/wiki/Microsoft_Azure
09 https://en.wikipedia.org/wiki/Microservices
10 https://en.wikipedia.org/wiki/Application_programming_interface
11 https://en.wikipedia.org/wiki/Message_queue
12 https://en.wikipedia.org/wiki/Telemetry
13 https://en.wikipedia.org/wiki/Cryptocurrency
14 https://en.wikipedia.org/wiki/Telegram_(software)
15 https://en.wikipedia.org/wiki/Gmail
17 https://en.wikipedia.org/wiki/Firmware
18 https://en.bitcoinwiki.org/wiki/Hashrate
19 https://en.bitcoinwiki.org/wiki/Difficulty_in_Mining
20 https://clck.ru/MLBTm
22 https://en.bitcoin.it/wiki/Stratum_mining_protocol
23 https://en.bitcoin.it/wiki/Difficulty
25 https://en.wikipedia.org/wiki/Live_USB