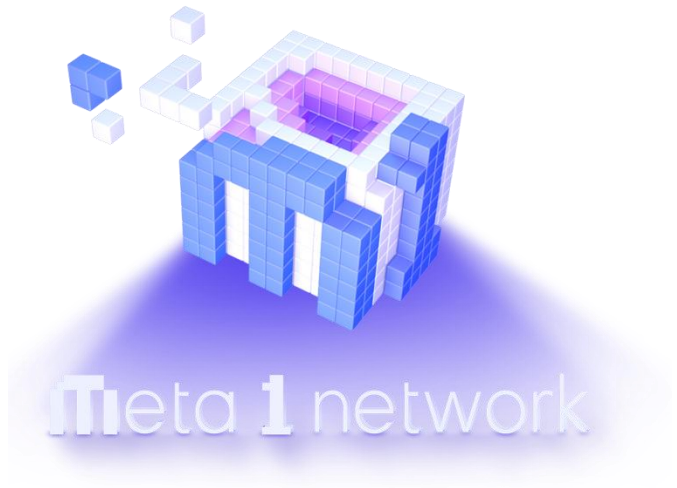


M1 network Project White Paper

Beta 0.2

author: loew

Contents

■ Abstract

M1 network is a decentralized cloud rendering platform for GameFi, virtual live streaming and panoramic social scenes. It naturally supports the data design of the blockchain and will be the infrastructure for the high-definition and full-realistic visualization of metaverse in Web3.

M1 network is a project that helps every chain and every user quickly realize and experience the high-definition and full-realistic visualization metaverse service. It is committed to providing each chain with a high-definition and full-realistic metaverse ecology, and helping them quickly form their own cosmic civilization. We also provide every user pass-through service in metaverse, which can realize the rendering and use with high-definition and full-realistic visualization for each metaverse service through the edge cloud rendering container. Players can switch services among different metaverses in M1 network by using their own coins.

The development of the metaverse until today is inseparable from the decentralized thought and its core economic system. In the future, every chain may derive its own metaverse civilization whose development is inseparable from the formation of the metaverse service ecology with high-definition and full-realistic visualization. Through M1 network, you won't worry about the problem of high-definition and full-realistic visualization, because M1 network Project will provide a universal high-definition and full-realistic visualization infrastructure service for each chain, accelerating the ecological establishment and the arrival of ecology service with high-definition and full-realistic visualization.

M1 network takes you into the metaverse era of high-definition and full-realistic visualization.

■ Challenges

Since the development of the blockchain-based metaverse, there are mainly rendered games based on WebGL, but there are very few open metaverse services. In addition, the vast majority of game developers currently develop games based on engines such as Unity or UE. Compared with the traditional developement of game service, there are a series of chanllenges resulting in the slower development of metaverse at present:

1. Insufficient services for the metaverse cannot quickly form an ecosystem.
The ecological formation of an edge issuance to this chain is a very slow process. The reason is the construction of applications is based on this chain. However, the essence of application construction is the migration of developers, which seriously hinders the ecological formation of the chain. And the lack of application ecology will cause more common users to enter the corresponding metauniverse ecology slowly, thus forming Rashomon.

2. Traditional game developers cannot quickly enter the blockchain space.
There are currently millions of GameFi users and developers, but there are only thousands of GameFi. The essential reason is that traditional game developers cannot quickly convert to GameFi developers.

Because there are still many problems in the development of GameFi, such as design, operation, rendering, etc. These are the basic conditions of the current blockchain, which prevents more developers from developing GameFi. And most of the engines are based on WebGL.

3. The economic model is bigger than the game experience.
At present, GameFi pay more attention to the construction of game's economic models. The rendering and interaction of games are still relatively primitive. Most of them are based on PC browser games, and they are rendered through WebGL. When users play games, there are always too many computer calculations or too long loading time. And the game device also has a greater impact on the compatibility of some games. For example, most web games use WebGL after FLASH can't be used, but browsers have memory limitations for JS. In Chrome X86, the limit for JS is 1.4GB. Players would have poor computer experience when running large games, leading to a high rate of user churn.

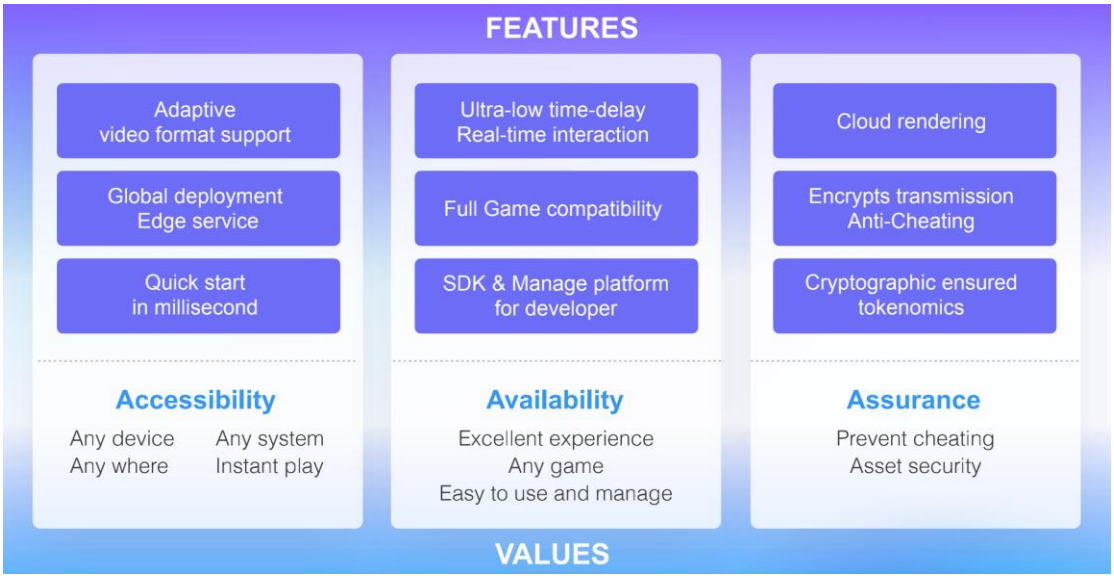
4. Insufficient infrastructure and poor gaming experience.
The development and operation of the game is a complex project, especially for the current metaverse expansion game, which is based on strictly controlled game balance and game experience. If it is just based on Chrome or other game blockchains, it will not develop an immersive metaverse game. Moreover, the current basic equipment of GameFi cannot support the virtualization requirements and rendering requirements of future meta-universe games. When the virtualization and rendering requirements cannot be met, it is difficult to make the game have a better experience. Let's take the ray tracing effect as an example. When there is ray tracing effect, we can get a higher sense of proximity from the game. Because everything that our human eyes can see is reflective or luminous, even the dark universe is gleaming. So we especially need ray tracing for the reality of an object product. If we need to use ray tracing technology, we need to use DirectX 11 or a higher version, or use the Nvidia RTX SDK package, etc. But only using WebGL cannot get a better game experience. Therefore, the blockchain needs a lot of rendering resources in the future.

■ Solution

We solve the problem of rapid construction of the chain ecology by designing a container. Through the container, traditional game developers can quickly develop and publish metaverse services. Whether you are a PC game developer or a mobile game developer, you can quickly enter the metaverse game development, and combine your own economic model with services to provide players with the best experience. We also provide rapid construction of the initial metaverse ecology for a chain. Through our metaverse container, all developers can deploy their services in different chains, providing a basic ecology for each chain. We deploy a batch of containers and each container is a metaverse service operating environment. Through this environment, we provide the necessary computing resources, storage resources and rendering resources for metaverse services. Metaverse developers develop their own metaverse services just like how they develop traditional games, and then they can deploy the services in our containers. We provide metaverse services running in browsers, mobile terminals, TV

sets, and node glasses for metaverse players, and they don’t need to install games. We manage these containers through a decentralized P2P network. When metaverse users need to start the service, we will select the nearest container node to provide services to them. And the users will experience metaverse service through interactive video.

■ Features



✧ High Efficiency

M1 network Project is committed to helping any blockchain establish its own metaverse ecology, allowing metaverse players to quickly enter your metaverse civilization. M1 network Project provides metaverse players with an interactive video channel that can be connected to metaverse through edge cloud rendering, and provides a basic metaverse ecology for blockchain projects, realizing the rapid development of the blockchain ecology.

✧ No Installation

M1 network Project provides containers for metaverse developers to deploy services. The interaction between the player and the service is realized through an interactive video stream. Therefore, metaverse players can enjoy services without installing any client. M1 network Project is installation free for any player in any metaverse.

✧ Low Delay

M1 network Project manages metaverse containers through a huge network. When a metaverse player starts to play in the metaverse, we will choose the nearest service node to provide the player with the metaverse rendering service. The metaverse player's operation and interaction with the metaverse can be controlled within 30ms. We provide low-latency metaverse services.

✧ Superior Quality

M1 network Project mainly realizes metaverse experience and interaction through interactive video streams. M1 network Project can provide users with 720P, 1080P and 4K resolutions, and can provide different experience versions of 30FPS, 60FPS, 90FPS and 120FPS.

✧ Decentralization

M1 network Project adopts decentralized management. The container of M1 network Project adopts P2P network to manage the edge scheduling of users and metaverse. Users contribute their own computing power to serve many other metaverse users. Combined with P2P scheduling capabilities, it achieves decentralized management of containers. M1 network Project Token records transactions between containers and users based on blockchain technology.

✧ High Fusion

M1 network Project is committed to providing users and metaverse developers with a service to develop and connect to the metaverse. It provides basic visualization and interactive services for every metaverse. Through this service, metaverse developers can easily make their own metaverse ecology develop rapidly and make users try all metaverse services without a specific device.

■ Compared Render Farm

✧ What is Render Farm

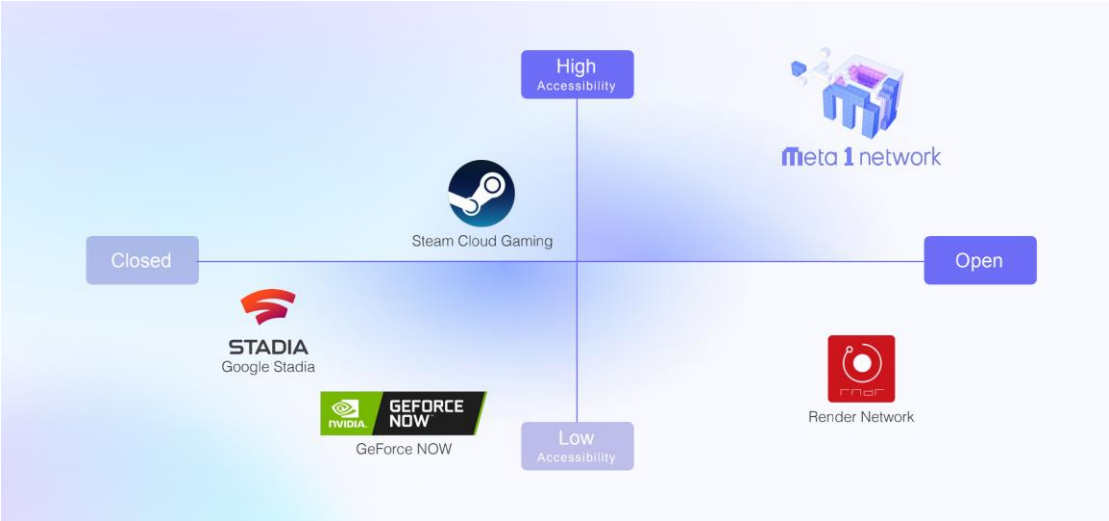
Render farm is actually a popular name. In fact, we should call it "distributed parallel set computing system". It is a parallel computing system built by using ready-made CPU or GPU, Ethernet and operating system. It is mainly used for offline rendering of animation and video. It decomposes tasks and renders pictures frame by frame. In order to achieve the purpose of model or action visualization, it is usually used in CG or traditional animation industries. Due to the need for a lot of computing power, multi-machine parallel computing is usually used, and then the tasks are combined as a whole. The render farm is characterized by good rendering image quality and low timeliness. There is no way to achieve real-time rendering. It is a way to change image quality with time. It has certain limitations and cannot interact in the rendering process.

✧ What is Real-time Rendering

Real-time rendering (lightweight rendering) is that the computer renders the data into a picture for each frame, and then presents it on the screen. The data of each frame is constantly changing, so the picture of each frame is constantly moving. It can manipulate and interact in real time and process 3D images at a very high speed, achieving realistic effects at the same time. Real-time rendering pays more attention to real time and interactivity. Usually the scene needs to be optimized, so as to improve the calculation

speed and shorten the delay. Then M1N is a real-time rendering container. Through real-time rendering, we can make the picture of metaverse service in the container visible and interactive to users. Thus, users can interact with the metaverse service in the container in real time through the interactive video stream.

■ Competitive Product Analysis



✧ RNDR

As a blockchain-based GPU render farm, RNDR is an offline render farm, which mainly provides rendering services for CG content. That project is a distributed offline parallel computing system. Users output video from the platform by submitting CG rendering requirements, which is not an interactive live video. RNDR is aimed at offline rendering market and M1N is aimed at real-time interactive rendering market.

✧ Google

Google currently launches a centralized service for cloud gaming and it is a pure cloud gaming store. It introduces games to provide cloud gaming services for game players. The main difference is that M1N not only provides players with metaverse services rendering, but also provides developers with basic real-time interactive rendering services, completely connecting publishers and players. More importantly, Google's cloud gaming platform is centralized and M1N is a completely decentralized metaverse real-time rendering service. M1N is the most authentic metaverse real-time rendering service platform in Web3.

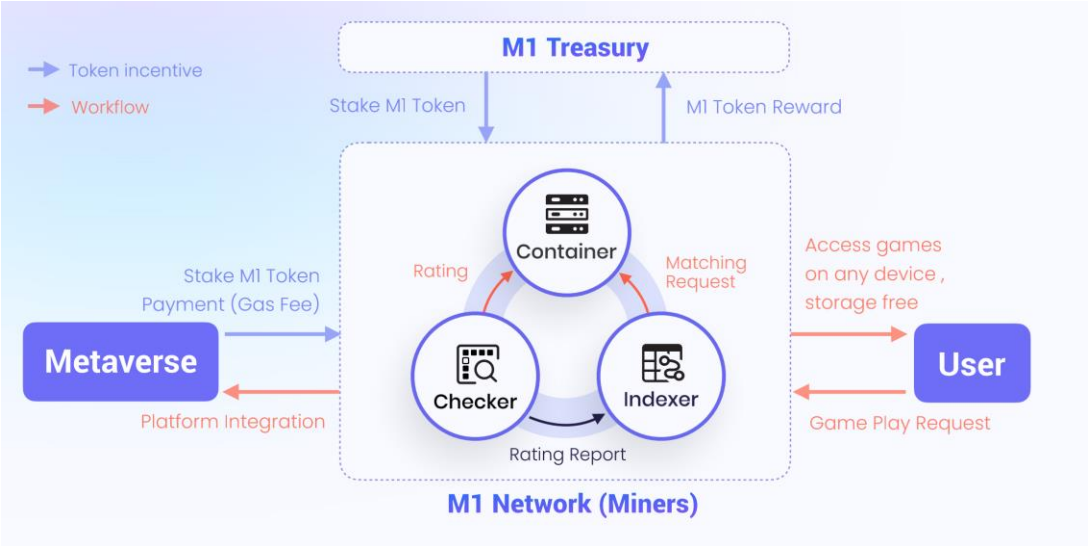
■ M1 network Project Technology

◆ Architecture Design

M1 network Project takes C/C++ as the main body to develop the container and puts the metaverse service on the server to run. It transfers the rendered metaverse audio and video images to the terminal

in the form of interactive video streaming, and the terminal no longer needs to install the metaverse client. Various terminals such as TVs, mobile phones, PCs, and tablets can run these metaverse services. In this way, we don't need to care about how the metaverse adapts to different software and hardware platforms, whether the terminal performance can meet the requirements, and so on.

◆ Application Workflow



M1 network Project is an infrastructure project that connects users with metaverse. It is committed to bringing more and more users a better metaverse service experience through low-cost methods, and accelerating the development of metaverse talents and infrastructure. First of all, every metaverse service developer can publish services to M1 network, and accelerate their own metaverse service rendering and distribution by purchasing a certain container, which is much faster than investing a lot of resources to promote users to download and install APP. In addition, M1 network will reward metaverse service developers to give them greater motivation to make better services. And container providers who provide higher rendering resources for edge metaverse players by contributing their own idle resources will also get a kind of income. When a metaverse player finds a service where he wants to play, he can play under this metaverse service by paying a certain fee, and can gain revendelayue from the service.

◆ Technical Difficulties

Real time: The overall delay of the metaverse service includes service logic operation time, audio and video rendering time, encoding delay, network transmission delay, client-side decoding delay, and the delay of client-side sending control information to the server. The real-time interaction of metaverse must reach a level to be accepted by players, which poses great technical challenges. Of course, it also depends on the performance of the hardware and the network.

Virtualization: Virtualization is very mature on the server side. We have virtual machine technology and various container technologies, but it is not that mature on the desktop. Ordinary virtual desktops do not

support GPU virtualization, and metaverse service rely heavily on GPU rendering. Without GPU virtualization, M1N cannot be achieved. So virtualization is a big technical bottleneck.

Decentralization: M1N's operation and maintenance management is different from traditional servers, because the server hardware that we use is different. M1N adopts fully decentralized container management, and the hardware load is very high, which poses new challenges to container management. So these problems must be solved technically.

◆ Platform Selection

There are many platforms for metaverse to run, but the only platform that is more suitable for metaverse is Windows. Although the Linux platform is open, it has no metaverse service support, and other console game platforms are basically closed technologies. The Android platform is also very suitable for metaverse. The concept of running an Android metaverse service on the server and then transferring it to the Android device looks strange, but actually metaverse service providers like this concept very much. Because many devices don’t allow the installation of third-party applications, and the monitoring is stricter. So we use the cloud to bypass this limitation. It is very helpful for closed hardware products, so the Android platform is also something we have to consider. There are two main routes for the virtualization technology of Windows games. One is the virtual machine solution, but the main problem is that the GPU virtualization technology is immature. It may require some professional graphics card support. The cost is very high and the performance loss is very large. Every game runs a Guest OS, which is a waste of memory. The plan was rejected by us. And there is a lack of available container-level technology on Windows. We can only use API Hook to manually implement virtualization. We call it the Sandbox solution. The Sandbox solution is to hook over all the system APIs used by the game, so that the game thinks that it is running on a normal OS. But it is actually an OS that we have taken over. The advantage of this is that the performance loss is very small, and there is basically no additional loss. However, it is more painful to adapt to each API. Each game needs to be adapted, and the game is usually not open source. Game developers usually do not cooperate with you to modify the code, so some hack techniques are needed to adapt to each game. Therefore, M1N will deploy and run metaverse services that are from Windows, Linux, Android and other terminals.

◆ Audio & Video Technology

The video stream uses H.264/H.265/VP9 encoding and mainly 720P/1080P@30fps, 1080P@60fps. Audio coding uses AAC. Since the standard encapsulation format does not contain control flow and cannot transmit user operation data, we have defined an encapsulation format, simply encapsulating the raw streams of H.264 and AAC and other formats and sending them to the client. M1N has made special optimizations for audio and video.

First, avoid using B frames to reduce the delay. Then, make larger GOP settings to reduce the proportion of I frames. Then ensure that the bit rate consumed by each frame is within a maximum controllable range. Besides, zero delay setting ensures that every time a frame of data is input, the encoder will

immediately output the encoded data of this frame, avoiding the encoder to buffer the frame data. In addition, adopt bitrate control. Using a fixed bit rate algorithm is not suitable. Because there are always still pictures in the game for a period of time. The bit rate is very low at this time. The variable frame encoder will allocate a large number of bits to encode, which will cause the data of this frame to be extremely large and then additional network data transmission is delayed.

Therefore, we adopted an adaptive algorithm to ensure that the overall bit rate is within the maximum range and the bit rate consumed by each frame is within a maximum controllable range meanwhile, ensuring that the data transmission delay of each frame is controllable. For some hardware, turn off the buffer and directly output the picture, and use software decoding to support zero delay output. The performance of Android devices is uneven, and the performance of early low-end chips does not meet real-time decoding, and GPUs need to be used for some acceleration. In terms of network transmission, TCP is used with the current RTC protocol to do some tuning at the network layer to reduce the delay.

◆ Game Developer & Publisher

The release of M1N-based metaverse service is relatively simple. Metaverse developers only need to access the SDK and release the metaverse service according to the M1N operating documents. The basic process is as follows:

- The metaverse service team can independently choose the commonly used 3D engines on the market to develop their own metaverse services, such as Unity or UE.
- After the main part of the metaverse service is developed, it can be connected to the corresponding wallet and payment channel based on the signaling channel provided by M1N.
- Package the metaverse service into an apk on the mobile terminal or an exe on the desktop.
- Publish the metaverse service installation package to the M1N metaverse center through the M1N publishing platform.
- M1N generates a web page or h5 address for preview or formal operation interface.
- The metaverse service team can distribute their own M1N metaverse entry and enter.
- Metaverse players can open the address to log in or pay through the wallet, and experience their own metaverse life.
- All player operations are sent to the metaverse service in the container through signaling, cloud container, and cloud container proxy.
    - The cloud container generates a real-time video stream to realize the real-time interaction between the player and metaverse service container.

◆ Command Channel

As a container-level design, M1 network Project can use data channel capabilities to establish communication between the client and cloud applications. The cloud application creates a UDP server and monitors a UDP port (localhost 127.0.0.1, the recommended range is 10000-20000), and starts to wait to receive UDP packets. The metaverse rendering client calls the metaverse cloud rendering SDK interface to create a transparent transmission channel, and the target port parameter in the SDK interface

is the port monitored by the cloud. The cloud rendering client first sends a custom data packet, and the cloud application UDP will receive the request and parse out the local proxy port. The obtained local proxy port sends a custom data packet, and the data packet will be returned to the client application through the created data channel.

- UDP server example：

Sample code:

```
int main() {
	int udp_socket_fd = socket(AF_INET, SOCK_DGRAM, 0);
	if (udp_socket_fd == -1) {
		printf("socket failed!\n");
		return -1;
	}
	struct sockaddr_in bind_addr = { 0 };
	bind_addr.sin_family = AF_INET;
bind_addr.sin_port = htons(xxxx);
bind_addr.sin_addr.s_addr = inet_addr("0.0.0.0");
	int ret = bind(udp_socket_fd, (struct sockaddr *)&bind_addr, sizeof(bind_addr));
	if (ret < 0) {
		perror("bind fail:");
		close(udp_socket_fd);
		return -1;
	}
struct sockaddr_in    upstream_addr = { 0 };
int len = sizeof(upstream_addr);
	char buf[1024] = { 0 };
	while (true) {
	ret = recvfrom(udp_socket_fd, buf, sizeof(buf), 0, (struct sockaddr *)
& upstream_addr, &len);
		if (ret == -1) {
			break;
		}
		const char* response = "response";
		sendto(udp_socket_fd, response, strlen(response), 0, (struct sockaddr *) &
upstream_addr, sizeof(upstream_addr));
	}
	return 0;
}
```

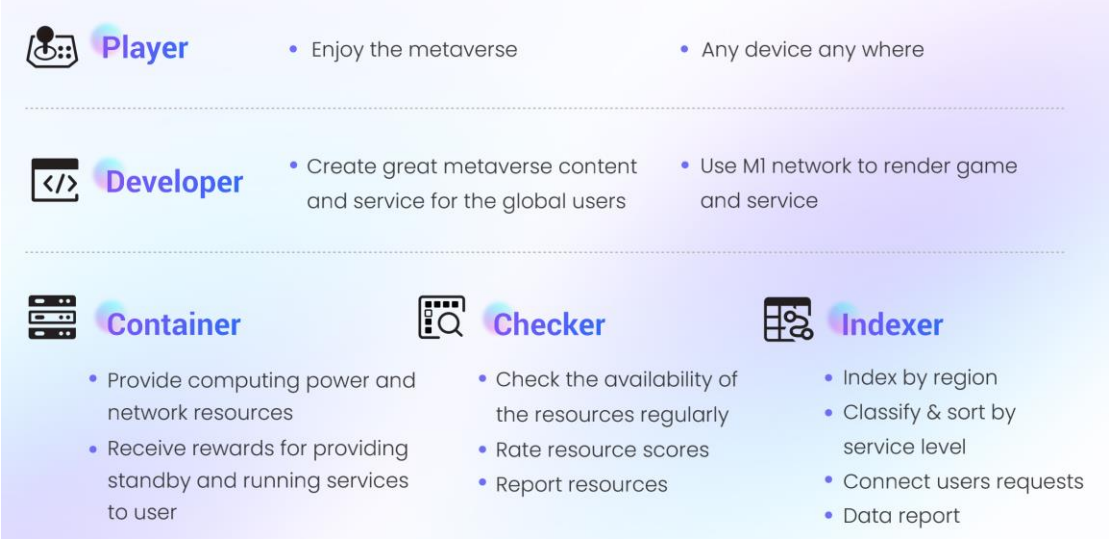
- Front-end example:

Smaple Code

```
(async _ => {
    const onMessage = msg => {
        console.log("recvice data:", msg);
    };
    const result = await new Promise((resolve, reject) => {
        const timer = setInterval(async _ => {
            const ret = await M1NSDK.createCustomDataChannel({
                destPort: xxxx, onMessage
            });
            if (ret.code == 0) {
                resolve(ret);
                clearInterval(timer);
            }
        }, 2000);
    });
    if (result.code == 0) {
        result.sendMessage('test');
    }
    result.sendMessage(`${custom_data}`);
})();
```

■ M1 Token Design

◆ Participant Role



M1 network Project is an open and decentralized metaverse infrastructure platform. In the future, the main participants are:

Container Contributor: M1 network Container Contributer contributes its idle CPU/CPU resources through M1 network Container, and obtains the corresponding service fee by serving the contributed

resources to the edge metaverse players.

Indexer: M1 network Indexer is an indexer that mainly provides container information for M1 network Container, and realizes the communication of containers between Game Player and Game Publisher.

Checker: M1 network Container provides containers to game developers and gamers. The validity of the container is verified by M1 network Checker to ensure reasonable compliance of the container.

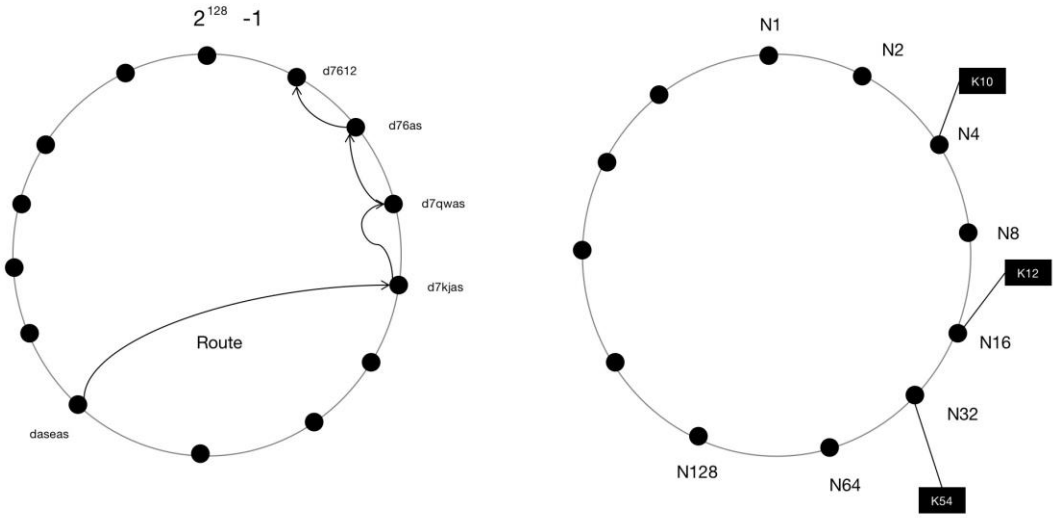
Game Publisher: M1 network Game Publisher is the main metaverse service contributor in M1N. It develops its own metaverse service through various existing game engines and deploys its service on M1N Network. It provides metaverse players with its service and obtains corresponding income through the metaverse. It can also purchase M1 network Container to provide a better metaverse experience for some metaverse players.

Game Player: M1 network Game Player is one of the core participants of M1 network. They are the main user group of M1N in the future. After metaverse developers release their services through M1 Network Container, Game Player can experience metaverse through M1N and obtain their own revenue from the metaverse service. In addition, M1 network Game Player will also be the main consumer of M1 network in the future, and acquire more rendering resources by purchasing container.

Let’s take a simple process as an example. First, the container contributor deploys its own idle resources to the M1N Container, which means that the Container is waiting for the access list. At this time, Checker initiates a calculation instance, which is run by the Container and the combination is returned to Checker. After the Checker has passed the verification, the Container enters the standby state, indicating that the Container can provide services to the outside world. The information will be sent to the Indexer data through the Checker. Indexer will index a DHT. After the Indexer is updated, when a metaverse service publisher or developer needs a container, it will obtain the DHT routing table from Indexer, and the corresponding Container can be found through the routing table, and the corresponding game can be deployed in the Container. When a game player enters, he can directly pull up the game to play the game.

◆ Dispatch Algorithm

The Container of M1 network adopts the peer-to-peer method, based on the P2P fully distributed structured topology (Decentralized Structured Topology DHT). The DHT (distributed hash table) is actually a huge hash table maintained by a large number of nodes in a wide area. The hash table is divided into discrete blocks. Eeach node is assigned to a hash block of its own, and becomes the manager of this hash block. The nodes of DHT are dynamic nodes, and the number is huge. Therefore, decentralization and atomic self-organization have become two important goals of design. Through the cryptographic hash function, the name or keyword of an object is mapped to a 128-bit or 160-bit hash value. All nodes in a system using DHT are mapped to a space, for example, a hash function mapping a bit name to a hash value.



The DHT structure can adapt to the dynamic joining/exiting of nodes, and has good scalability, robustness, uniformity of node ID distribution, and self-organization capabilities. Due to the deterministic topology of the overlay network, DHT can provide precise discovery. As long as the destination node exists in the network, DHT can always find it and the accuracy of discovery is guaranteed. The most classic cases are Tapestry, Chord, CAN, and Pastry.

The biggest problem with the DHT structure is that the maintenance mechanism of DHT is more complicated, especially the network fluctuation (Churn) caused by frequent joining/exiting of nodes, which will greatly increase the maintenance cost of DHT. Another problem is that DHT only supports exact keyword matching queries, and cannot support complex queries such as content/semantics. Therefore, the management of M1 network peer-to-peer is based on the accounting method of the blockchain itself, combined with the characteristics of DHT to develop DHT on blockchain. We use the record information of blockchain as a routing table of DHT. When there are enough routing tables, it can not only provide the efficiency of single-point query, but also can use blockchain's current analysis tools as the basis for content and semantic complex queries. And combined with the location information, the blockchain Node and the Peer Route Table’s location information are bound, so as to realize the fastest, the closest location and Container query, and to realize Container positioning and scheduling.

◆ Rendering Algorithm

M1 network not only solves the management of the rendering node based on blockchain, but also solves the process of rendering interactive video with a single container. When the physical link between the container and the game player is close to the logical network link, it does not matter whether the game is essentially installed locally. Whether it is outputting 720P or 1080P or 4K resolution, we can render the user's image quality in the form of interactive video, and greatly balance the relationship between user resources and user experience. The maximum real-time interaction capability is achieved by constructing

the video rendering downlink channel and the user interaction uplink channel. And M1 network Project supports the rendering of 360-degree VR content, supporting H264/265 and VP8/9 video coding and decoding algorithms for ordinary video rendering, supporting AAC and other formats for audio coding and decoding, and supporting ERP and other video coding and decoding for VR content.

M1 network Project uses three-level rendering technology. The first level is engine-level rendering technology. The second level is container-level rendering technology, and the third level is video-level rendering technology. They correspond to the running state, output state and experience state of the game respectively. Engine-level rendering technology mainly provides rendering image quality at the data-driven stage. Video-level rendering technology provides image quality effects from the video codec level. Experiential rendering technology provides experience from the transmission and terminal decoding levels.

Engine-level rendering technology:
M1 network Project mainly solves the rendering interaction problems of Unity, UE and other commonly used game engines. Compared with popular GameFi, M1 network provides more game engine choices. M1 network Project implements OpenGl, Direct3D, GDI and other low-level rendering engines at the bottom, and does a lot of optimization work in light reflection. We assume that three points are created in an empty three-dimensional space. A plane has at least three points. The object is composed of different planes. In graphics, triangles are generally used as examples. Assuming that in a three-dimensional space, if you want to see a triangular surface in a two-dimensional display, you need to render the triangle from the three-dimensional space to the two-dimensional space. In the real world, triangles scatter light in all directions. For a browser-level WebGL, it is impossible to have the resources to calculate all rays of light in different directions. The computer-rendered image only calculates the light scattered to the direction of the human eye, which is also called the camera direction. So by adding a camera in the 3D space and placing a screen network at the front perspective point, each frame is a pixel of the rendered image, drawing a ray that intersects the camera perspective point. If these rays intersect on the screen, the screen can observe this triangle, then mark this triangle border. We only render the pixels in the overlapping part between the border and the pixels without processing the other parts, and we get a triangular image. This method of projecting to the pixel grid is one of the simple rendering methods. From it we can understand that points and planes are one of the cores that affect the resources required for rendering. Therefore, M1 network Project uses AI algorithms to greatly optimize the point-surface relationship in the rendering process to reduce the surface without affecting the rendering effect. Aiming at the key points, M1 network Project makes the light splitting more suitable for the relationship between the human eye and the screen, so as to provide visual artifacts, thereby providing the rendering effect of the final output video. M1 network Project takes into account that most of the current game resources are based on the point and surface infrastructure as the game asset, because the M1 network Augmented Reality technology was developed on the basis of reducing the surface and increasing the points to achieve the ultimate engine-level effect improvement.

Video-level rendering technology:

M1 network Project mainly solves the terminal and optimal real-time encoding technology. At present, the commonly used video coding includes H265/H264/VP8/VP9 and so on. Since the characteristics of different terminals are different, we use Android as an example hardware decoding: We can call the special module code of GPU to solve it and reduce CPU operation, and then have lower requirements on CPU and other hardware. Software decoding requires CPU operations, which increases the burden on the CPU and increases power consumption. Hardware decoding is to hand over part of the video data originally processed by the CPU to the GPU, and the parallel computing power of the GPU is much higher than that of the CPU. This can greatly reduce the load on the CPU. After the CPU occupancy rate is reduced, some other programs can be run at the same time. For Android devices, the most commonly used SOCs are Qualcomm, HiSilicon and MediaTek. Most of these SOCs integrate a lot of functions, such as CPU, GUP, DSP, ISP, video decoding, audio decoding, etc. So when we say SOC, we are not directly talking about the CPU.

Advantage of Hardware Decoding: more power-saving, suitable for long time mobile video plays and live streaming. When the battery of the mobile phone is limited, it will be better to use hardware decoding. To reduce CPU usage, CUP can be given to other threads to use, which is conducive to the fluency of the phone.

Advantages of Soft Decoding: It has better adaptability. Software decoding mainly occupies the operation of CUP. Software decoding does not care about the hardware decoding support of the device. It can be used with CPU, but it occupies more CUP. It is very performance-consuming and power-consuming. It is better to use software decoding when the device has sufficient power or when the hardware decoding support of the device is insufficient. We adopt adaptive codec solutions. According to the characteristic parameters of the terminal, it is paired with the following container, and is automatically scheduled to the most optimized encoding container to achieve the best terminal experience to ensure the rationality and experience of video encoding.

Transmission-level rendering technology:

M1 network Project mainly solves the problem of anti-weak network. Delay is a concern for everyone. Regarding latency, our research found that for different types of games there are different latency requirements. The requirement for FPS game delay is stricter. If it exceeds 100ms, players may not accept it. In addition, the requirement for RPG game delay is within 500ms, and the requirement for real-time strategy game is within 1000ms, that is, one second, which is still acceptable. Therefore, in terms of transmission, in addition to the original RTSP method, we also added WebRTC. WebRTC provides low-latency, peer-to-peer communication, which is more suitable for the application scenario of Container. Our test found that in a weak network environment, WebRTC will have a better experience. In an environment with about 3% packet loss, WebRTC will have better image quality and lower latency. WebRTC is open-standard and supported by mainstream browsers. It is alsocross-platform and can be used on all platforms without requiring plug-ins. Just open the browser and use it. Because WebRTC is open-standard, other native applications can be accessed as long as they implement the WebRTC standard. This is also the benefit of using the WebRTC solution.

Therefore, M1 network rendering algorithm adopts a three-layer construction from the engine level to

the video level and then to the transmission level, combined with the edge scheduling algorithm to jointly control the delay of the M1 network below 50ms, bringing a better experience for game players.

◆ Token Design

M1 network Project Token adopts exponential decay method. We can define the exponential decay formula as

$$-\frac{\mathrm{d}N}{N} = \frac{\mathrm{d}t}{\tau}$$

This [formula] is a number with a time dimension, and its meaning will be seen later. In fact, you can look at this formula from another angle. For this, we can change it a little bit:

$$-\frac{\mathrm{d}N}{\mathrm{d}t} = \frac{N}{\tau}$$

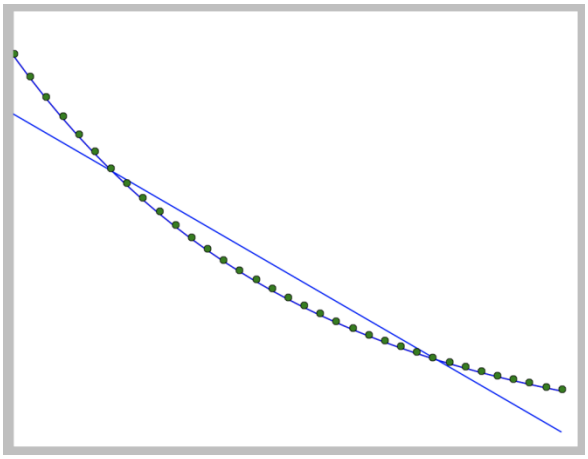
In this way, we can find that the left side of the equal sign is the rate of decrease in the number of tokens, and the negative sign means decrease. The right side of the equal sign is [formula] times the current Token number. In other words, the rate of token decay is proportional to the number of tokens. We can get the function between the number of atoms [formula] and time [formula].

$$N(t) = N_0 e^{-\frac{t}{\tau}}$$

The number of Tokens decreases with time in a negative exponential manner of [formula], and we obtain [formula] from the formula through definition. In this way, the meaning of [formula] is very obvious, which is that the initial number of tokens is 100000000. We use $T_{1/2}$ to indicate the half-life is 4 years, then we can deduce:

$$\frac{N_0}{2} = N(T_{1/2}) = N_0 e^{-\frac{T_{1/2}}{\tau}}$$

The specific curve diagram is as follows:

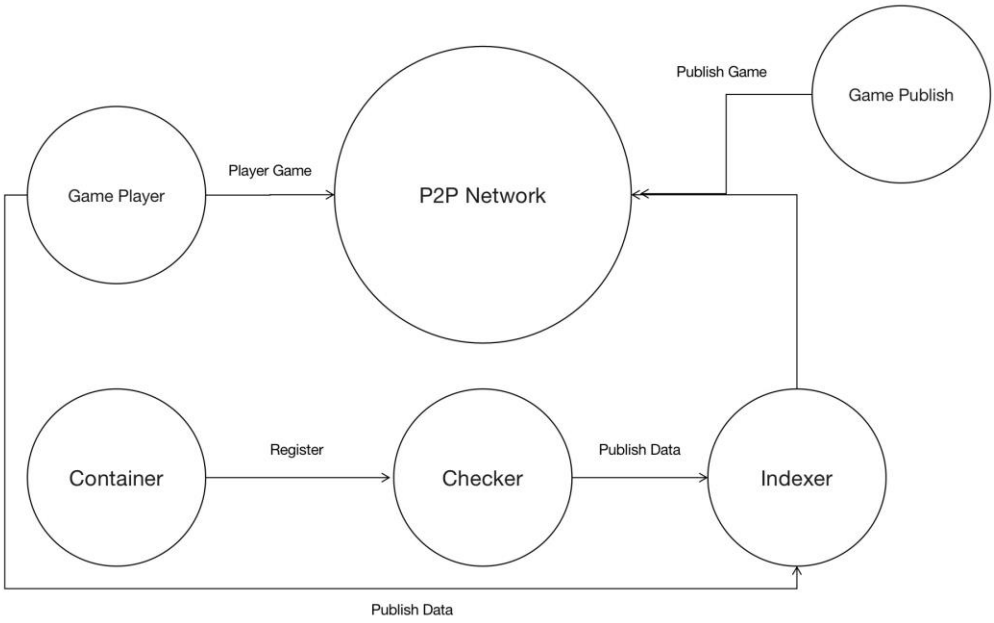


（Release Curve）

Through this release curve, we can add more participation to the M1 network Project in the early stage, and ensure Token's value and sustainability by reducing one-half of the release with a half-life of 4-year.

◆ Consensus Algorithm

M1 network Project adopts improved PoS, which stands for proof of stake. It is a consensus algorithm that is more environmentally friendly and greener than PoW(proof of work). PoS mechanism changes the competing computing power in the PoW mechanism to system equity. The greater the equity, the greater the probability of becoming the next bookkeeper. Through PoS, there is no need for mining and consuming a lot of electricity and energy. And it is more decentralized compared to PoW type cryptocurrencies such as Bitcoin. Compared with the 51% computing power attack of the PoW algorithm, PoS requires the purchase of 51% of the currency, which is more costly and there is no need to attack. To avoid inflation, the cryptocurrency of the PoS mechanism adds new currencies at a certain annual interest rate, which can effectively avoid the occurrence of austerity and maintain basic stability.



M1 network Project will be based on a hybrid model of variable computing power + Indexer + Checker. Take our basic workflow as an example. After the user deploys the M1 network Container, it is verified through Checker and the routing information is stored through Indexer. The entire process manager, Container, starts to join the network, and then Checker will check the standby container every 15 minutes. If the Checker is not checking after the Container enters the service state, Game Player will synchronize the Checker information to the Indexer. Game Publish can selectively purchase Containers based on standby Container on P2P Network.

■ Find Out More

- **Official website**

www.m1nk.io