# RNDR Network User Manual


## Section I: General

# Section II: Using the Network

# Section I

## A) About RNDR

RNDR is a blockchain peer-to-peer GPU compute network that connects artists in need of additional computation power for rendering their scenes, to providers that receive RNDR tokens for their GPU power[1]. It allows complex GPU-based render jobs to be distributed and processed on a peer-to-peer (P2P) network, making the transactional process of rendering and streaming 3D environments, models, and objects simpler for end users. The blockchain functions as a public database for the RNDR network, providing an improved means of tracking digital assets and objects - in which content is time-stamped and stored with a unique ID. With RNDR, users can create ultra-high resolution media, crowd source projects and manage complex 3D digital rights, providing a vibrant marketplace to produce and share digital works, assets, and applications that anyone can access and leverage.

The Internet was built on the promise of distributed computing and the open exchange of ideas and data. As we move from hypertext and hypermedia to a fully immersive metaverse built on interactive 3D content, the challenges and opportunities are exponentially greater[2]. In order to support the emerging media of tomorrow, it is crucial to be able to bridge the gap between the promise of these breakthroughs and the feasibility of widespread access to them. Until now, no system scales rendering speed across many dimensions of work in order to allow content creators to tap into the vast pool of graphics cards from an online network.

### Problem

We are on the cusp of a technological transformation of our very view of reality, affecting everything from computation to physics. Everything is becoming more virtual, from the explosion of mobile content over the past decade to engineers producing new realities, augmented and virtual, that allow us to immerse ourselves in new computer-generated worlds. As entertainment companies adopt these new ways of producing new visual effects, content creators and editors find themselves facing new dimensions of complexity. Larger and more complex jobs spanning thousands of frames across time (for animations) and space (for VR walkthroughs) require external servers and additional resources. Complexity of rendering is also exponentially increasing due to higher frame resolution and frame rate (e.g. UHD 8K@240 fps is 256 x the work of HD 720p30). Further raising complexity are increases in views per frame (e.g stereo rendering doubles workload to support left and right viewpoints). In addition to increasing complexity, the use of computer generated imagery (CGI) is accelerating. Over the past decade, the amount of visual effects (VFX) scenes per Film and TV productions is accelerating has

[1] J. Urbach (2010) Token-based billing model for server-side rendering service: https://patents.google.com/patent/US9197642B1/en

[2] https://medium.com/render-token/the-future-of-rendering-photons-tokens-and-next-steps-towards-a-blockchain-driven-metaverse-555724605705

grown dramatically, while Architecture, Design, Marketing and Engineering firms increasingly rely on advanced CGI and photorealistic rendering to support new forms of visualization.

Already within the public GPU cloud an increasingly large number of market segments are competing for limited High Performance GPU compute resources. Centralized data centers have been unable to expand fast enough or price competitively to meet demand due to high upfront CAPEX and rapid GPU hardware depreciation. Because centralized investment in GPU infrastructure risks underutilization when demand falls below capacity, pricing has remained high - or requires large upfront commitments to offset monetization risk. For independent artists wanting to produce virtual and augmented reality experiences, local GPU render farms are prohibitively expensive and impractical to install. As a result, they have to choose between costly GPU cloud rendering services or building expensive local GPU server infrastructure.

Yet, the current system harbors many inefficiencies. Most developers' GPUs remain idle when they are not rendering their own work, reducing the productivity potential of local GPU infrastructure. Further, excess GPU supply from proof-of-work cryptocurrency mining has led to an arms race dynamic in which increasing compute resources are dedicated to mining fixed (or regressive) block rewards. The rise of ASIC (Application Specific Integrated Circuit) and FPGA (Field Programmable Gate Array) mining, in which hardware is designed exclusively to solve blockchain hash functions, has only exacerbated this trend. The result is that the use of GPU's for proof-of-work blockchain mining consumes tremendous power with fewer productive returns, leading to waste or shutdown in certain cases. With the rise of more computationally efficient blockchain protocols, there is an opportunity to more productively use latent GPU compute resources to create the next generation of 3D and holographic media.

In distribution, not only do artists face exponential rendering increases - with the push to higher frame rates, higher resolution, and more realistic images - but it is also difficult to monetize 3D works across a fragmented app ecosystem. Holographic media requires new layers of 3D metadata, asset interoperability, and granular Digital Rights Management (DRM) when compared to less interactive media formats like 2D rasterized imagery. Even today, studios or independent content creators have difficulty tracking the media they create. As the Internet grows, it becomes almost impossible to effectively track digital assets and to prove an infringement through copyright and trademark laws. As a result, business models have forced users into all or nothing subscription paywalls, costly downloads, or intrusive data mining for ads. Despite innovations in eCommerce, producers and consumers have few options for micro-consumption.

The lack of a 3D media monetization framework is leading to a market failure depressing the growth of immersive virtual and augmented reality production. Developers and content creators are often forced into walled-garden app stores, resulting in platform lock-in and increased intermediary costs, harming both consumers and producers. Further, app store fragmentation limits artists ability to monetize complex 3D works across the widest audience, making it less economically viable to produce more ambitious immersive works.

Ultimately, pressures on GPU cloud rendering as well as technical limitations in distributing and monetizing complex holographic works are constraining the growth of holographic media. This next generation of media, which is more costly to produce and more complex in its distribution, is most efficient through an open smart contract based micro transaction framework for effective monetization.

## RNDR Solution

Most artists' GPUs remain idle when they are not rendering their own work. By utilizing the RNDR Token network ecosystem, developers can choose to monetize their idle GPU power by performing renders for RNDR Tokens. The RNDR network provides a mechanism for idle GPU power from around the world to be efficiently allocated for complex 3D rendering tasks, providing a scalable peer-to-peer GPU cloud computing network[3]. Through this process, RNDR minimizes GPU waste, provides Artists with affordable and fast on-demand parallelized rendering, and enables efficient dual use of compute resources. Developers with idle GPUs can contribute their spare capacity to the network and earn tokens, which they can re-use when rendering demands exceed local infrastructure and additional scale is needed.

The blockchain based rendering network facilitates efficient, reliable, and remunerative rendering of time-stamped tasks on a peer-to-peer basis. As miners successfully complete scenes on the network, they build trust needed to perform higher tier rendering tasks. Nodes work by accurately carrying out the rendering instructions encoded by a user into a 3D scene graph. When a user confirms that the work was successfully executed, the node becomes more trusted. The network refers to this process as **Proof-of-Render** where trustworthiness is built on rendering 3D scenes rather than the proof-of-work mining used by other blockchain protocols.[4]

Blockchain technology has also evolved to now store, validate and time-stamp complex mixes of technical specifications, schedules, accounts, regulations, protocols, standards, and property rights. This technology can handle digital rights management, needed for complex digital assets that can be routinely copied and for which time-stamped proof of authorship is crucial. Recently, the Ethereum blockchain has enabled tokens, which allow for immediate and more complex transactions to be executed using smart contracts[5]. Inside each RNDR transaction backed by an **ORBX package**, there is a hash of all assets in the system used to build the source render graph for the render job. Within the ORBX module, application and plugins can be launched, enabling users to access remote 3D experiences as ORBX.js or HTML 5 live streams.[6] This means flow of RNDR tokens that utilize and render each asset can be spread back to creators of these assets (e.g. 3D models, texture packs, industrial scans, toolchain services) whenever they are transactionally connected back to their source render job (either once, or recurring).[7]

---

[3] J.Urbach Patent US880382 (2017) Allocation of GPU Resources Across Multiple Clients

[4] https://bitcoin.org/bitcoin.pdf

[5] https://github.com/ethereum/wiki/wiki/White-Paper

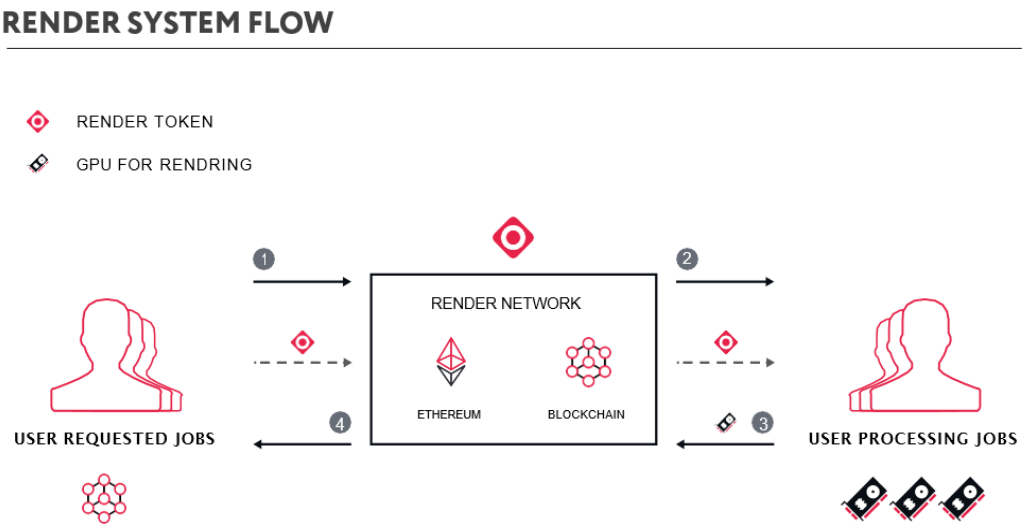
[6] https://medium.com/render-token/rndr-day-1-let-there-be-light-f2665133a5dc

[7] https://medium.com/render-token/rndr-a-photon-driven-economy-e5f5c6896a67

With RNDR, physically correct rendering tasks are completed quickly and efficiently in a blockchain based peer-to-peer network with no error or delay and with securely protected property rights. GPUs that are otherwise wasted on proof of work mining are used to render ultra-high resolution images, generating more productivity from these compute resources. RNDR, powered by the blockchain, uses the act of creation — rendered by the laws of physics and light — to open the door to a new model of holographic computing.

# B) Architecture & Network flow

RNDR connects Artists looking for additional compute power for rendering with GPU Compute Providers (Miners) who have idle GPU capacity to process the renders. Users send RNDR Tokens to the individual performing the render work and the RNDR network receives a small percentage of RNDR Tokens for facilitating the transaction and running the render network. By enabling the use of RNDR tokens for compute services, RNDR facilitates the coordination of GPU Compute Demand and GPU Compute Supply across a peer-to-peer network. Through smart contracts, the network is able to run without any manual intervention and is able to identify and address issues automatically and robustly. RNDR is the primary unit utilized to obtain rendering and streaming services. The token allows users to utilize the wide array of available GPUs in the peer-to-peer network facilitated and kept track of by the blockchain[8].




Users create accounts linked to the Ethereum blockchain through smart contracts and unique wallets. With RNDR Tokens, users are able to use these tokens on the network for various rendering and streaming services. During this process, the network will send a request for a RNDR token smart contract in order to enter a transaction with both parties - the person or server processing the render/streaming and the person who requests the rendering services. The cost of the job is calculated and determined in RNDR tokens. The smart contract will then move RNDR tokens across accounts once the allotted render job has been completed and approved. After the process is completed, users will then be able to use the RNDR Tokens they

[8] https://github.com/rndr-network/Token-Audit

earn on the network for future rendering tasks or use them in order to pay operational costs. Key roles can be broken down into the **Artist** uploading and the **Miner** side rendering process, with job allocation and confirmation provided by the **RNDR Network**, with **Dual Use** users embodying both roles.

# C) Network Roles

RNDR is a multi-sided peer-to-peer network with four essential actors.

- **The RNDR Network:** Facilitates the coordination of peer-to-peer GPU Rendering Supply and Demand, providing the protocol for exchanging microservices using smart contracts. The network operates smart contracts that confirm and payout rendering tasks in conjunction with an artists approval, as well as automatically allocating computing tasks between artists and miners. The use of transactional tokens is provided by the network.

- **GPU Providers (Miners)** Provide Proof-of-Render work processing computing tasks requested by Artists. Miners are able to freely join and leave the network using a desktop based client. Miners receive user IDs attached to their user Wallet, from which their activity on the network can be coordinated. Upon joining the web client, the network will determine critical performance information used for node benchmarking.

- **GPU Requestors (Artists)**: Submit rendering tasks to the network to be processed by miners. Artists use a web interface where they can select job and rendering output preferences. Once a work is processed, Artists confirm and pay for processed frames

- **Dual Use Miner / Artists:** Some users will both request and process rendering tasks. Using RNDR, Dual Use Miner / Artists are able to use their idle GPU power to accumulate RNDR tokens that they can then use to process future work, for which they may need additional GPU compute resources available on the RNDR network.

# D) Network Core Technologies

The RNDR Network is built on **OctaneRender**, the world's first and fastest GPU-accelerated, unbiased, and physically correct renderer. The web-based client that a Miner installs runs a version of Octane on the node in order to process rendering tasks. Artists requesting rendering tasks use OctaneRender to configure their scenes for processing. Octane renders photorealistic images by exploiting graphics cards to run light scattering computations on thousands of parallel GPU cores - processing scenes faster than conventional CPU rendering techniques. Because of its efficiency, OctaneRender was the first commercial render engine to deploy fully unbiased

path-tracing methods to achieve physically correct rendering results[9]. Tracing how light and energy bounce around a scene, Octane necessarily anchors its algorithms in the laws of physics, accounting for everything from the velocity of light in exotic substances to interference patterns in sub-surface scattering in human skin. Octane uniquely combines its blazing speed with supreme accuracy, enabling artists to achieve ultra realistic results in a fraction of the time. Users can test rendering speeds using OctaneBench, a GPU benchmarking tool generated by Octane users running speed tests across a wide variety of GPU and system configurations.

The RNDR Network uses Octane in conjunction with the open standard ORBX media and streaming framework in order to support fully distributed rendering[10]. ORBX is a container format that captures scene data (assets) and an XML (extensible markup language) render graph which describes the semantics of a scene. Just like a web page, it can be cached to archive (.orbx file) or streamed from a URL or URI over raw UDP/tcp or web wss or https[11]. The ORBX format supports over 20 of the industry leading DCC (Digital Content Creation) tools, and contains industry standard sub formats like Alembic, OpenVDB, EXR, Open Shader Language (OSL), and gLTF. The ORBX interchange format is critical for splitting rendering work from host 3D applications. A one-click ORBX export from a 3D content creation tool fully decouples all assets and code needed to perform a remote GPU render job on multiple mining nodes. By fully abstracting scene data from third party software tools with the ORBX scene format, RNDR is able to parallelize work across a blockchain peer-to-peer network. ORBX interchange works regardless of host application, providing efficient, open, software-agnostic distributed rendering.

The RNDR network also leverages the ORBX scene graph to enable deep chain of authorship and validation, allowing users to build microservices on top of a peer-to-peer rendering platform. Every time a user uploads a scene and a miner processes a job on the RNDR network, all assets and settings in the ORBX render graph are hashed. With assets in a scene attributed in ORBX XML render graph, the network has a semantic history of every object and setting within a scene. As a result, with the ORBX scene graph, the network is able to track transfer, recomposition and re-usage rights through tokenized rendering and streaming processes. The **Hashing** of the ORBX render graph and scene data provides immutable and granular history of all assets and work processed on the RNDR network. Thus, as assets move through the network, ORBX provides full traceability needed for attribution and authorship.

Finally, ORBX is used as a file standard for holographic media payback and distribution[12]. The format enables interoperability across 3D tools and well as the distribution of fully volumetric six-degrees-of-freedom (6-DOF) scene data for holographic experiences like playback for the Manifold VR camera. ORBX.js streaming technology delivers high performance 3D games and desktop applications to the open Web – using only HTML5 and JavaScript[13]. Because ORBX.js is browser based, it bypasses apps or downloads, enabling frictionless publishing of

[9] https://en.wikipedia.org/wiki/Path_tracing

[10] https://home.otoy.com/wp-content/uploads/2017/08/m41018-An-introduction-to-ORBXJU-6-ATH.pdf

[11] https://news.ycombinator.com/item?id=14950148

[12] https://www.reddit.com/r/RenderToken/comments/ba2549/immersive_digital_experiences_alliance_idea_to/

[13] https://blog.mozilla.org/blog/2013/11/05/mozilla-otoy-and-autodesk-work-to-deliver-high-performance-games-and-applications-on-the-web/

immersive media experiences to multiple endpoints like VR, AR, and mobile. With ORBX, users can share ultra-realistic low latency graphics, as well as leverage the format's scene graph to develop hybrid crowd sourced rendered experiences that combine ultra high resolution or light-field precomputing with real time raytracing[14]. The ORBX framework is open and accessible in OctaneRender Modules, allowing third party developers to build additional ORBX-based applications or services through an API[15].

Both ORBX and Octane are extensible, supporting most mainstream 3D content creation tools, The Octane SDK is also available to third parties that want to expand the ecosystem to new tools. For example, the RNDR API can be used for third party applications that leverage the network to re-render multiple permutations of scenes with quick and lightweight material and texture swaps. Capabilities like Delta sync enable users to leverage RNDR to precompute scene changes without re-uploading a new ORBX file - opening up new B2B applications on the network. Octane is available on Windows, Linux, and iOS with a new Vulkan multi-back compute framework designed for Apple Metal, AMD and Intel Integrated graphics.

# E) RNDR Compute Pricing

One RNDR Token is mapped to a unit of work called OctaneBench http://octanebench.com, which is measured in seconds and in concurrent OctaneBench power. Users select from preferences for speed, cost, system hardware, node reputation, and security - and the amount of OctaneBench compute work varies based on a user's preferences as well as aggregate GPU supply and demand on the network. These preferences are organized into three tiers: **Tier 1**, **Tier 2**, and **Tier 3**. The pricing for each tier is stable in order to allow artists to be able to accurately manage rendering costs and for miners to operate sustainably, above marginal cost. The tier pricing algorithm periodically updates the amount of compute work a user receives in each tier based on changes in GPU performance, the current price of GPU cloud rendering from other providers, electricity costs, and network supply and demand patterns. For example, adjustments to OctaneBench are made with new OctaneRender releases in order to reflect increased rendering performance as well as the efficiencies of new GPU hardware generations.

RNDR Multi-Tier Pricing (MTP) provides a framework to enable GPU supply and demand to self-organize in the most efficient way possible, ensuring that RNDR is working at peak capacity and there is little wasted GPU power[16]. Tier 3 provides users primarily concerned about price - and who currently do not use GPU cloud rendering networks due to high costs - with a highly economical rendering solution. Tier 3 has no queue priority so reductions in cost come at the expense of speed. Tier 2, which provides queue priority and higher hardware and reputation requirements, enables users to access high quality parallel GPU nodes with minimal delay, offering the potential for significantly accelerated rendering speeds when compared to Tier 3.

[14] J. Urbach (2015) USPTO (US8957946B2), Re-utilization of render assets for video compression

[15] https://render.otoy.com/forum/viewtopic.php?f=118&t=55184

[16] https://medium.com/render-token/rndr-tokenomics-update-multi-tier-pricing-mtp-338d5dea1d29

For artists on a deadline, Tier 2 reduces operational delays like rendering bottlenecks by providing 'first-in-line' access to decentralized GPU compute supply on the network. Finally, Tier 1 provides levels of security and pricing commensurate with current GPU cloud rendering costs.

MTP works in conjunction with the network's **Reputation, Allocation** and **Governance** systems to achieve an equilibrium. Multiple price tiers enable the network to meet the needs of artists with different preferences (i.e. speed, cost, or security) as well as creating a mechanism for artists to prioritize these preferences. When a user identifies their preferences by selecting a tier, it helps the network's **allocation** process operate more efficiently, enabling better matching of variegated GPU compute supply and demand. For Miners, higher tier work can only be processed with sufficient levels of trust and positive reputation history. MTP therefore incentivizes nodes to maintain a good **Reputation** and current, up-to-date hardware.

MTP is designed to create a virtuous cycle that maximizes overall network utility. Tier 3 is priced close to operating cost, enabling the network to capture marginal demand. An efficient market occurs when pricing is at marginal cost (P = MC), providing Pareto Optimality to producers and consumers. Thus, Tier 3 generates marginal demand from users who would otherwise forgo GPU cloud rendering due to price distortions. Capturing marginal demand is critical for the network because it increases overall network utilization, enabling miners to generate positive reputation history needed to process higher Tier 2 work. As more marginal demand is added to the network, increased congestion incentivizes users with preferences for speed or premium hardware to upgrade to Tier 2 with queue priority and demonstrated node reliability.

Because miners earn more tokens per compute cycle when processing Tier 2 work, the network creates an incentive for miners to successfully process work at Tier 3 and build a positive reputation history. The process of organizing nodes into tiers based on rendering activity provides the foundations for the network's **Proof-of-Render Governance** system. Thus, by capturing marginal demand with Tier 3 (in which price is close to marginal cost) network utilization is increased, and the network creates node history needed for the creation of additional services like priority and trusted rendering.

Multi-tier pricing operates using a multiplier attached to the OctaneBench algorithm. Tier 1 work has an OctaneBench multiplier of 1, thus 1 RNDR = 25 x OB4 * HR x a multiplier of 1. Tier 1 is the basis for RNDR compute pricing. Tier 2 has an OctaneBench Multiplier between 2x and 4x, providing users 2 to 4 times the amount of cumulative OctaneBench compute work per token. Similarly, Tier 3 has an OctaneBench multiplier between 8 and 16. The OctaneBench multiplier for different RNDR tiers may be fixed, or algorithmically adjusted based on real-time supply and demand conditions on the network - for example, a multiplier starting at the high end of the OB range when demand is low, and moving towards the lower bound when congestion increases. Below is a description of how much OctaneBench compute a user receives in each tier.

| Pricing | OctaneBench Work Per RNDR | Priority | VRAM \| RAM Minimum | Concurrent OctaneBench |
|---|---|---|---|---|
| **Tier 1** (Secure) | 1x RNDR = 25 x OB4 * Hr (with $1 of RNDR = 100 x OB4H) | High: Separate Secure Node Pool | High | High |
| **Tier 2** (High Priority) | 2-4X cumulative OctaneBench(OB) vs.Tier 1 | 1 | High | High |
| **Tier 3** (Low Priority) | 8-16X cumulative OctaneBench(OB) vs. Tier 1 | 2 | Lower | Lower |

[Figure 1 — MTP Table with General Parameters for Tiers]

## Tier 1: Highest Security and Priority Tier

- Tier 1 Render Services provide the equivalent of 256 OctaneBench x 256 seconds adjusted for OctaneBench4 and normalized to one US dollar. **$1 worth of RNDR = 100xOB4/Hr**. This pricing reflects the current costs of GPU cloud rendering on centralized services like Amazon Web Services (AWS)

- As the network develops, and decentralized nodes are added to Tier 1, there will be additional audits and/or security procedures that Tier 1 nodes need to follow.

## Tier 2: Highest Priority, Parallelized Peer-to-Peer Rendering

- Tier 2 Render services provide 2x to 4X the total OctaneBench work as a Tier 1 RNDR Token on the network providing 200- 400% more compute power than Tier 1.

- Tier 2 rendering work has Queue Priority over Tier 3, providing accelerated first in line parallelized rendering services on the network when compared to Tier 3 rendering

## Tier 3: Low Priority

- Tier 3 Services provide 8-16x the OctaneBench work as a Tier 1 RNDR Token. As a result, 1 RNDR Token will provide 800 - 1600% more compute power than Tier 1.

- Tier 3 services have the lowest queue priority, so they **are not recommended for time sensitive rendering jobs**.

Miner nodes are tiered based on factors like Total System OB, Node History (Successful % of jobs processed), Minimums for VRAM and RAM which construct a Composite node Score. Users selecting a specific tier of work are matched to nodes from the selected tier on RNDR.

| Node Tier | VRAM | RAM | Total System OB | Node History | Overall Node Score |
|---|---|---|---|---|---|
| **Tier 1** | High | High | High | Verified Node | NA |
| **Tier 2** | High | High | High | High Success Rate, Positive Reputation Score | Above Tier 2 Node Benchmark |
| **Tier 3** | Lower | Lower | Lower | Less Reputation History | Minimum Network Benchmark |

**Figure 2 — MTP Table with Tiers and Select Rolling Factors**

With Multi-Tier Pricing (MTP), the network is able to efficiently sort decentralized supply and demand, reducing deadweight loss when supply and demand don't intersect at equilibrium[17]. The network pools GPU compute resources, enabling users with different preferences to meet on a peer-to-peer network. Multi-Tier Pricing organizes these preferences and resources according to price elasticity. As a result, both price sensitive and inelastic users derive utility from using the network, while miners are rewarded commensurately for their resources and trustworthiness. Multi-Tier Pricing helps eliminate some trade-offs between speed, quality and resolution that artists have previously face when using current GPU cloud rendering platforms. RNDR enables artists to scale to the cloud with reduced cost or time pressures, providing more users with access to GPU cloud rendering tools needed to create boundary pushing cinematic work. Meanwhile, miners have a platform to generate productivity from idle compute power.

# F) RNDR Tokens and Currency Gateways

Artists are able to use RNDR Tokens or a fiat currency gateway (Stripe or PayPal) for compute services on the Render Network.

[17] https://scholar.princeton.edu/sites/default/files/reinhardt/files/597-2016_efficiency_in_economics-conceptual_issues.pdf

# G) Governance and Incentives

The RNDR Network functions using both automated and user generated verification processes to confirm work. Artists are encouraged to promptly approve jobs that are correctly processed through the system. Final renders are withheld until full resolution watermarked previews are confirmed. As a result, artists are incentivized to expediently confirm correctly processed frames. If a frame is not successfully processed, an artist has the ability to reject the frame, and the frames will automatically be assigned to another node on the network for re-rendering.

**Reputation scores** are essential for both artists and miners because they are central to the resource allocation process on the network. For example, artists with higher reputation scores will have access to more concurrent nodes while rendering. Similarly, miners with better reputation scores will get jobs assigned faster than mining nodes with lower reputation scores. For example, if the network is not at 100% utilization, nodes with the lowest reputation scores will be last in the queue, incentivizing miners to build a positive history. Reputations scores also serve as proxies for detecting malicious behavior. For example, if an artist regularly rejects work from a number of nodes with high success rates, these excessive failures will lead to a job being terminated. In such a case, the network will quarantine the job and an artists reputation score may be negatively impacted. Similarly, if a node repeatedly fails to adequately process frames that are subsequently successfully rendered by another mining node, its reputation score will be negatively impacted, and the node will receive a lower ranking.

In addition to review, the network uses progressive rendering, in which artists can watch the render progress in real time as more samples are processed and the image goes from noisy to noise free. By watching a frame progressively render, artists will have additional tools to detect processing abnormalities early on in the processing of a job. The Network also automatically checks jobs by comparing a node's potential and a job's parameters through the client .exe application. If there are large disparities between a node's OctaneBench potential and a node's output, the RNDR network can detect these anomalies and reroute work to properly functioning nodes. Octane's AI denoising algorithms can also be run on a render to detect "cheats" when samples have not been fully processed and noise filtering has been applied. Finally, the network .exe continually interrogates a node, allowing it to detect changes like underclocking. In the case of underclocking or poor performance relative to a node benchmark, the node's OctaneBench score is recalculated and adjusted to reflect the underclocked value.

Finally, RNDR deploys incentives designed to reduce the risks of reputation gaming, in which a user with a poor reputation or a malicious user just creates multiple accounts to avoid the consequences of bad behavior[18]. Artist accounts are tied to their unique Octane Account, which includes software license entitlements. Without these software entitlements, users cannot create renders for use on the network. Additionally, artists with better reputations get access to more nodes, creating an incentive to build history. For miners, the network's reputation system is

[18] http://faculty.haas.berkeley.edu/stadelis/Annual_Review_Tadelis.pdf

used for the creation of higher tier node pools, which provide greater returns per compute cycle, with sustained positive reputation history. Thus, it is disadvantageous to start over with a new account. To make this system work, the network corrects for system error and includes recency bias functions in its scoring algorithm that allow users to self-correct over time.

Thus, disputes are resolved through a combination of user generated review and through automated machine learning processes. Finally, **RNDR Support** provides users an outlet to handle disputes and correct for idiosyncrasies or systemic network errors.

# H) Security

When a file is uploaded to the network, the scene is exploded into individual assets and each individual asset is encrypted and hashed, providing end-to-end traceability that enables the network to detect malicious behavior and protect chain of custody. Further, more complex permissions can be created through **API Tokens** - providing security management tools in more complex studio workflows consistent with emerging cloud production security principles.[19] These assets remain encrypted end-to-end, both in transit and at rest, for additional security. The outputs themselves are also encrypted, so network traffic cannot be sniffed to peek at while the files being sent. Finally, RNDR is designed so that unencrypted assets are never stored on disc or in memory, making them irretrievable from a malicious actor. Mining nodes also do not store rendered outputs, reducing the potential for leaking of sensitive IP.

Using encryption, watermarking, scene explosion, and hashing, RNDR is built as a **Zero Trust** security framework, that does not rely on or assume hardware security. With Zero Trust, RNDR is able to scale with reduced network vulnerability. Incentives like the network's **Reputation Score** that are built into the RNDR governance framework further deter malicious behavior.

# I) Watermarking

On RNDR, watermarking is used in conjunction with smart contracts to coordinate the flow of services on the network. Assets are viewable in watermarked form until a job is confirmed and a user pays for the completed work. Once the work has been paid for, a RNDR smart contract is triggered to release the unwatermarked output to the artist and release tokens from the smart contract to the miner. When the conditions for a completed job are met, the smart contracts release the asset from its watermarked state. These conditions can include the confirmation of a successfully rendered frame, or the transfer of entitlements from preview to download or stream. As a result, watermarking is essential for the provision of services on the network.

[19] https://movielabs.com/production-technology/

More advanced watermarking capabilities are based on OctaneRender AI-accelerated path-traced rendering technology and the ORBX semantic scene graph.

# J) Reputation Score

The network scores users based on their success rate and past performance that is used to coordinate activity on the network. When users submit or process jobs to the network, they will either incur a positive or negative reputational result depending on whether they met the conditions of the smart contract. If artists reject work that is properly rendered, their reputation score is reduced, while if miners do not successfully process work, they will see a downgrade of their reputation. These combinations of fraud prevention make up something that we refer to as "**proof-of-render**" - where human job confirmation and algorithmic rendering checks are used to confirm work. Reputation scores help advance **proof-of-render**.

Reputation scores incentivize good behavior on the network, enabling Artists and Miners to generate more productivity from the network. For **Artists**, reputation scores are used to determine the amount of concurrent mining nodes a user can access at any given time. Thus, artists with higher reputation scores are able to process work faster, incentivizing them to build a positive success rate that often come from thoroughly checking scenes prior to uploading them to the network. Artist side reputation scores also ensure that requestors without positive histories do not clog up the network with work that needs to be re-rendered, creating disruptions in job allocation. Similarly, **Miners** will only be able to process higher tier work - which provides increased token rewards per compute cycle - by meeting the reputation score requirements. Additionally, miners with higher reputation scores are assigned work faster than other users with lower reputation scores, incentivizing them to maintain high success rates. Through this process, reputation scores are both used as a coordination and an incentive mechanism.

In allocation, although the possibility of a backup on the render queue is low, the user ranking system will serve as a needed tie-breaker in specific situations - further incentivizing consistent positive activity. For example, if there are 20 GPU's currently available, and two users have a render job that needs to be completed that will require 20 GPU's. User A is a new user who just joined the network and does not have any history of requesting work; User B is an established user that requests render jobs on the network every day. All other factors being equal in this case, User B would take priority for the render job because they have rank over User A.

# K) Job Assignment

In order to effectively match GPU requests to providers, the RNDR network incorporates node reputation history and node power in its automated assignment process. Using Multi-Tier Pricing, artists select from a menu of preferences for cost, speed, and security, enabling the

network to optimally sort jobs based on the nature of demand at any given moment. The assignment processes incentives GPU providers to maintain high node success rate and allows artists to optimize their preferences. Job assignment and Multi-tier pricing are based on a tier system. Nodes are grouped into Tiers which are based on attributes like **priority** and **performance benchmarks** and **reputation score**. This enables jobs to be matched with the appropriate region or tier. For example, a user selecting Tier 2 work is matched with any nodes meeting the conditions of Tier 2 nodes prior to any work from Tier 3 being assigned to those nodes. Only after all Tier 2 work is assigned to Tier 3 nodes, will Tier 3 work be assigned.

Job Assignment (within each tier) aims to create a balance across the amount of OB assigned to each job at any given time. To do this, rather than assigning frames in the order they were created or just randomly selecting an available frame, the assignment process looks at the amount of nodes actively working on each job. By doing this, the system is able to determine which job is most in need of power and prevents situations where two jobs with drastically different render times get assigned at the same pace. Additionally, as the network of nodes settles across all available jobs, an equilibrium is met where jobs are getting equal attention and in turn creates a stickiness of the node-to-job pairing - meaning that a node will continue to be paired with a job that it is being processed successfully. As all active jobs have equal power attached to them, a node completing a frame will then cause the job with the frame that was just completed to have the lowest amount of power attached to it - meaning the node will get assigned a frame from the job it just completed. This is advantageous because the node will not need to re-download assets and can spend a greater portion of its time rendering.

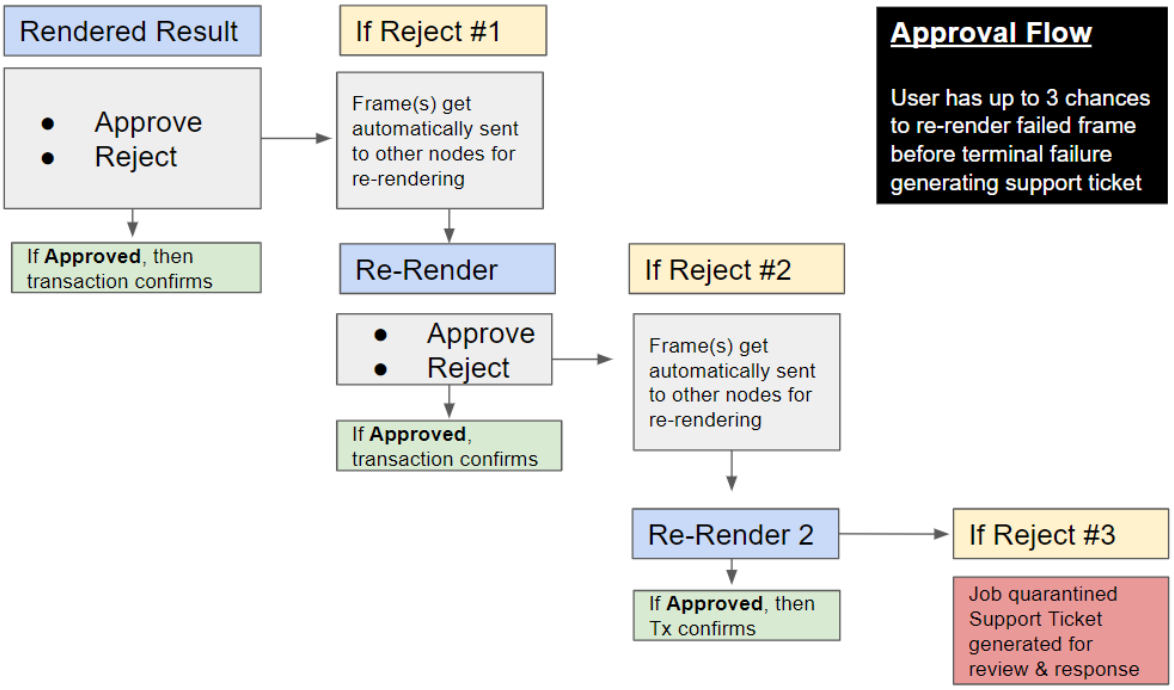
The network uses intelligent automated assignment to keep network supply and demand in equilibrium, efficiently matching jobs to nodes[20].

# L) Job Approval and Dispute Resolution

When frames are not properly processed, an artist has the opportunity re-render the frame in order to receive fully completed work. When an artist elects to re-render, the failed frames are automatically sent to other nodes for reprocessing. Because the job is automatically sent for re-rendering, the artist will not have the ability to modify the scene in any way. As a result, there is no benefit to rejecting an undesired render that results from an artist improperly configuring their scene (for example, setting the wrong colors, cameras, lighting, resolution or samples). After a number of multiple failed re-renders, the job is quarantined, and a support ticket is generated for review. If other nodes successfully render previously failed frames, the **reputation score** of the node that initially failed the render is temporarily reduced. If multiple attempts at re-rendering did not resolve the failed frame, and support review indicates that the work was initially correctly processed, and the error lies with the artist, then the artists' reputation score is temporarily

[20] J.Urbach US880382 (2017) Allocation of GPU Resources Across Multiple Clients

reduced. The dispute resolution process can be seen below and additional information about this process is available in the RNDR Helpdesk.




# M) Proof-of RNDR

There are many challenges that must be solved in order for a blockchain-based render economy, built on proof-of-render, not just proof-of-work, to function successfully. RNDR uses the successful processing of rendering work to build a peer-to-peer network of sufficiently trusted nodes. Trust is generated by the successful processing of frames, validated by a number of inputs like: artist approval, the performance of a node relative to node benchmarking, and automated render checks - for example, the use of the Octane AI denoiser to automatically spot noise filtering. These processes can be applied to a render job data and when conditions are not met, flag it as a bad result[21]. As a result, trust on the RNDR network is generated by successfully rendering images in a similar way that proof-of-work protocols use the completion of cryptographic puzzles to achieve trust. Further, because works are exposed to users for confirmation, and can be checked with scene data, verification of disputed events is possible.[22]

[21] https://www.reddit.com/r/RenderToken/comments/748zwp/proofofrender/

[22] https://medium.com/render-token/rndr-a-photon-driven-economy-e5f5c6896a67

# N) Hashing

As a work is created on RNDR, its corresponding ORBX scene graph is hashed. This hash represents a unique identifier for what is inside the ORBX that cannot be reverse-engineered or duplicated by a different scene. With all assets and work hashed on the network, completion of render jobs can be codified and tracked - for example the releasing of a confirmed and paid render from its watermarked state to an artist. With the hashing of all assets on the network, RNDR provides traceability and attribution needed for deep chain of authorship. Through a RNDR API, complex applications can be developed that leverage hashed assets with unique identifiers.

### RNDR API

The RNDR network provides an open API, enabling users to build custom applications on top of RNDR. The API supports customizable asset rendering pipelines, for example swapping specific files like .OCS textures on RNDR, rather than re-uploading complete ORBX files. This enables users to build new micro-services and applications on top of RNDR. Further the RNDR API provides developers with access to a templated UI that they can customize to their workflow needs. Using the RNDR API, administrators can create AUTH permissions for multiple users in a shared studio, with credential management tools for uploading scenes and submitting jobs. As a result, a studio can customize their account administration.

# O) Transaction Fees

The RNDR Network operates through the collection of transaction fees associated with the processing of jobs. Because the RNDR network is a multi-sided economy of rendering work and a marketplace for exchanging assets, the incentives are structured to reduce excessive intermediation. The nominal transaction fee is anticipated to be a function of transaction volume and capped at a certain percentage between .5% and 5% depending on the nature of the transaction. Transaction fees are intended to be used to capture storage and bandwidth costs associated with operating the network.

# P) Network attack vectors & mitigation

While the networks **Reputation** system, which is used for job allocation, incentivizes artists and miners to act in good faith, there are also additional technical processes to reduce malicious behavior on the network. To prevent spamming attacks, like submitting thousands of jobs at only a few samples, the network will not let a user submit a new job without first paying for prior work or having enough tokens in their account to pay for the work. As a result, such an attack

would be extremely wasteful. When malicious or bot-like activity - like rapid job generation - is detected, a user or node will be quarantined. On the miner side, the RNDR client uses a heartbeat to ensure that a node is online and to prevent crashes. If a return signal is not detected the client is restarted. This process helps ensure reliable uptime.

RNDR also has log files from the RNDR client that generates forensic data to be used for security. A system tray application is the RNDR software itself. If someone manages to steal and reverse-engineer the systray app, at its core it is a wrapper that can connect to an external system. The Octane software is well protected, because the miner client doesn't hold Octane itself, but allows the node to run stripped down parts of Octane that are needed 'on the fly'.

Finally, RNDR client software goes through periodic automatic pull-downs of new versions in order to minimize exposure to malicious behavior from older clients. Finally, all assets are fully encrypted end-to-end and at rest, reducing hardware vulnerability.

# Glossary

- **Artists:** Users on the RNDR network who are purchasing GPU compute power in order to process and render scenes/pieces.

- **Miners:** Users on the RNDR network who have registered their GPUs for use in processing and rendering scenes/pieces.

- **RNDR Token:** The Primary token unit used to obtain rendering, processing and streaming services.

- **OctaneBench (OB):** GPU benchmarking tool generated by Octane users running speed tests across a wide variety of GPU and system configurations.
    - **OB Unit:** A unit of work measured in seconds and concurrent OctaneBench power.

- **ORBX:** A container format that captures scene data (assets) and an XML (extensible markup language) render graph, which describes the semantics of a scene.

- **Proof-of-Render:** Confirmation of rendering work by users and systemic algorithmic rendering checks.

- **Reputation Score:** Score assigned to nodes based on reliability, success rate, past performance and rendering ability. Used in order to determine resource and job allocation for rendering jobs, as well as Tier assignment.

- **Ordinality:** Process by which user activity and work is archived within the RNDR network.

- **Hashing:** The registration of scenes and assets on an immutable database to track entitlements and recomposition rights.

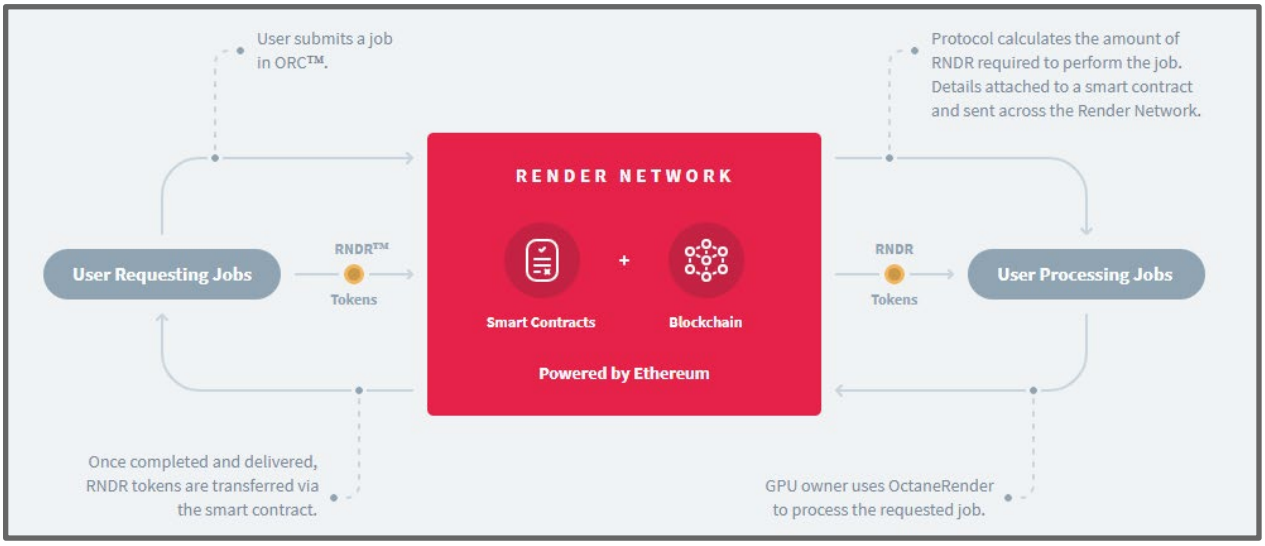
# Section II: Using The Network

## Recommended RNDR Hardware Requirements

- Windows OS for miner's client (Linux is coming soon).
- Latest editions of OctaneRender are installed & periodically updated in the RNDR client
- **GPU**: An NVIDIA GTX 1070 is recommended minimum. Currently, we only support NVIDIA GPUs but AMD and Apple Integrated graphics are on the roadmap.
- **VRAM**: 8 GB+
- **System RAM**: 32 GB+
- **Network Connection**: Strong Internet connectivity is essential
- **IP**: Static IP is preferred

**Network elements**

- Central servers (Web Server, Explosion node, Job allocation, Storage server)
- Artist (Actor) (requestor, user, client)
- "RNDR client nodes" (workers, engines, providers, miners)
- Ethereum Blockchain (Locking smart contract, Token contract, Wallet)
- Payment providers (Paypal or Stripe)




**Artist Flow**

ORBX upload (webserver) -> ORBX explosion (OTOY node) -> Job creation -> RNDR locking -> Job allocated / completed -> Frame-by-frame approval -> Miner payout -> Job download

**Miner Flow**

Work assigned -> Asset download (webserver) -> Frame rendering -> Output upload (webserver) -> RNDR received (upon approval)

# The RNDR Flow In Depth

1. **Artist prepares a scene**
2. **Artist uploads new scene to be rendered**
3. **Artist creates new job with render parameters**
4. **Artist performs user-side job estimation**
5. **MetaMask Instructions for artist on-boarding**
6. **Artist loads estimated amount of RNDR Tokens for processing the job**
7. **RNDR Server distributes Submitted Frames to Miners**
8. **Miners receive & process Frames**
9. **Artist approves rendered results**
10. **RNDR server releases tokens to Miners from locking smart contract**
11. **Artists downloads rendered results**

# Using RNDR

Updated network FAQ materials are also available at the RNDR Helpdesk.

## 1. Artist Prepares a Scene

### Using ORBX

The RNDR Network functions using the ORBX Scene File Format. Any works that a user wants to render on the network need to be exported as ORBX files using OctaneRender™ Standalone or an OctaneRender Plugin Integration. Once a scene is exported as an ORBX file, it can then be uploaded to RNDR for peer-to-peer rendering. Please See OctaneRender Documentation for packaging ORBX Scenes to be uploaded to RNDR as well as FAQ material here.

When preparing a scene it is highly recommended that prior to rendering on RNDR an artist:

- 1) Checks the scene in OctaneRender standalone in order to ensure that all materials and settings convert properly and no additional optimization is required.
- 2) Run a smaller test prior to submitting a large scale job. For example submitting a smaller test of a frame range prior to uploading a large animation sequence.

## 2. Artist uploads new scene to be rendered

### Note on Storage

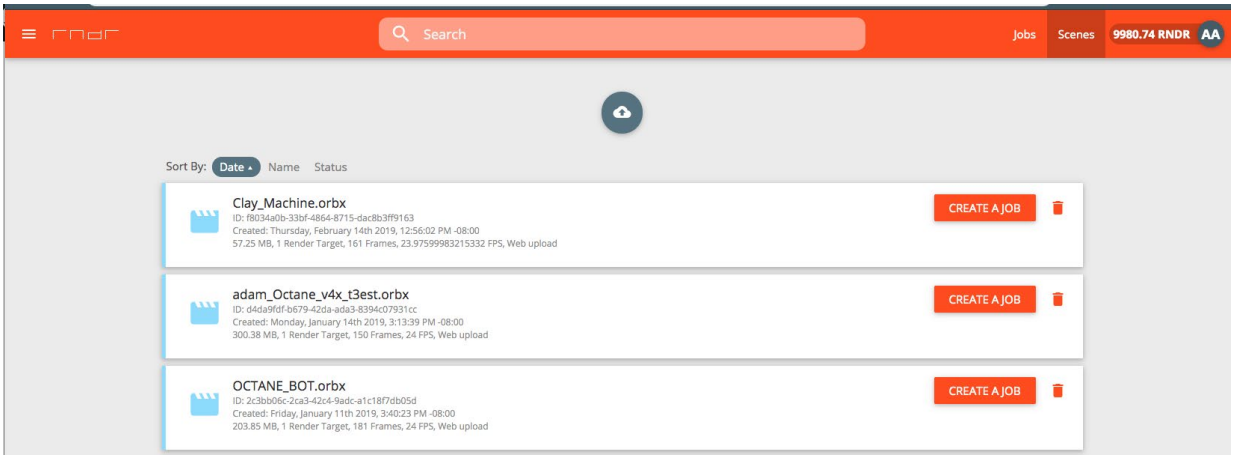
Files that are uploaded to the RNDR Network will only be held for two weeks, and cannot be retrieved once on the network. As a result, the RNDR Network ***should not*** be used for storage. For scenes or files that you want to keep, please ensure that you have a local copy available.

### Security Note

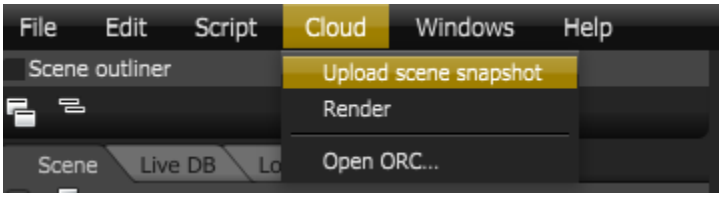
When you are uploading Scenes to RNDR, each file is encrypted during the uploading process and cannot be retrieved once on RNDR. As a result, RNDR does not fully function as cloud storage and we encourage storing a backup of your work internally.

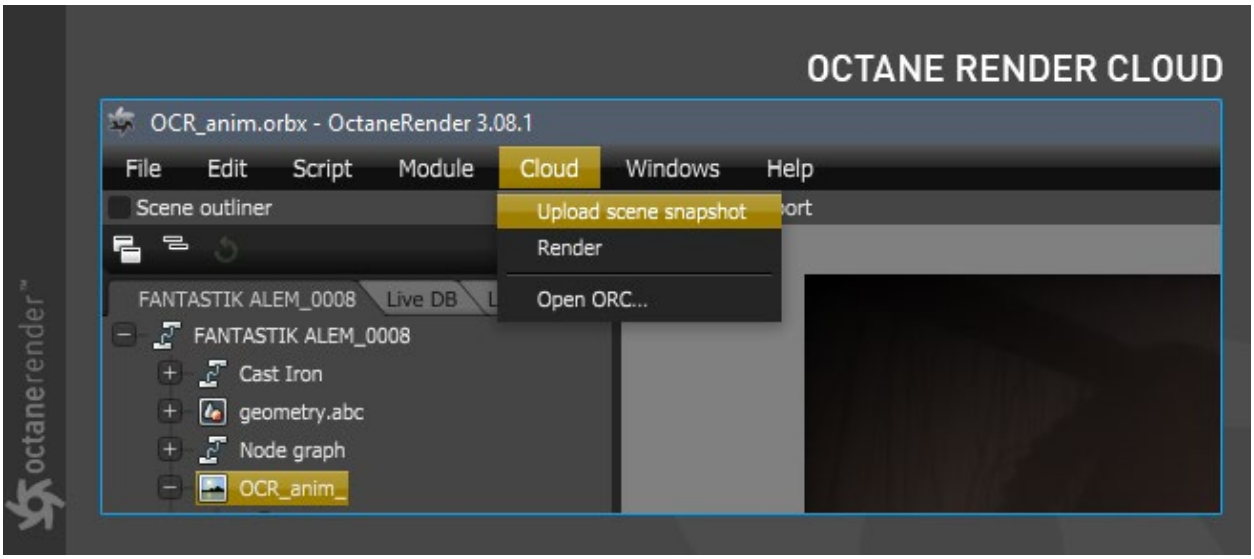
### Uploading a Scene

There are two ways to upload a scene to RNDR. The first is to use the RNDR web interface. Using Octane Standalone or your preferred plugin integration, export your scene as an ORBX file. Then navigate to the Scenes page by clicking on the Scenes tab in the top right then click on the orange cloud icon and select an OctaneRender Package File (files with .orbx extensions).




In the future, RNDR will become compatible with uploading scenes to the network directly from OctaneStandalone or your preferred plugin. Using this method, you will need to use current versions of OctaneRender. Open your project then click on the cloud tab (in the top left of the program) and select "Upload scene snapshot".



This process varies for some OctaneRender plugins that are more deeply integrated into the Host 3D Content Creation tool. Below is a description of how this process works for the OctaneRender for Cinema4D plugin

# Uploading a Scene with Cinema4D

Copyright Ahmet:
http://www.aoktar.com/octane/OCTANE%20HELP%20MANUAL.html?PreparingSceneUpload.html

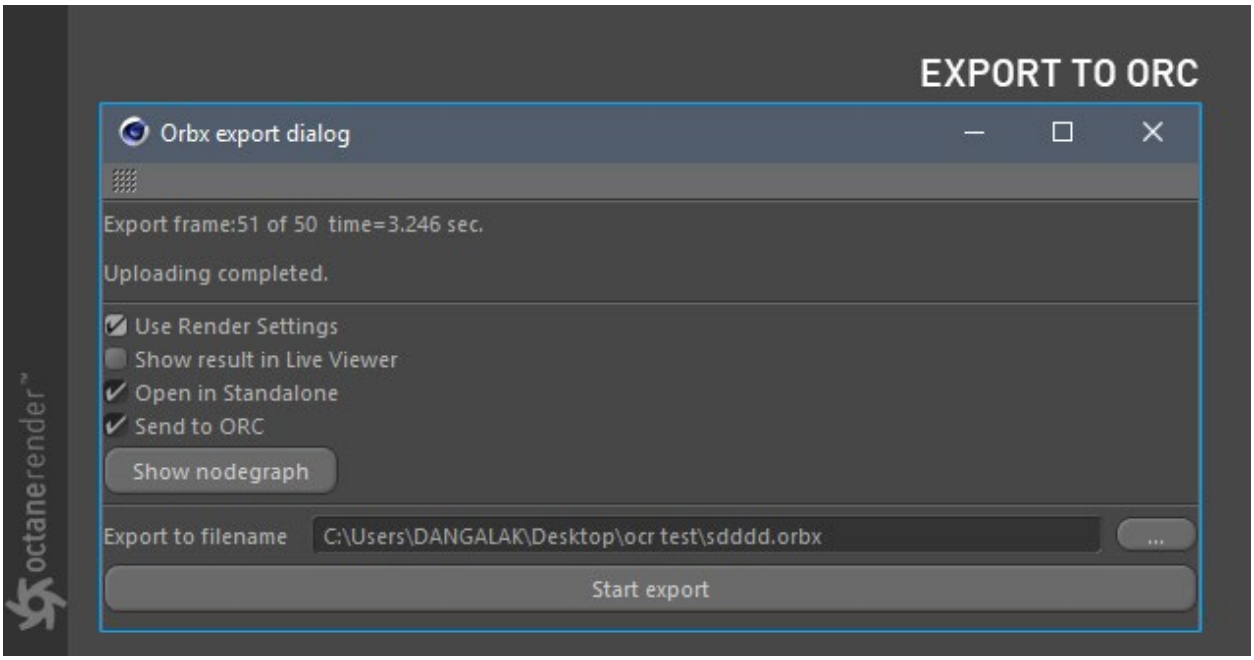
When uploading scenes created using OctaneRender plugins, there are a few things you need to do before you send the scene to the RNDR. Let us explain them step by step at the example of Cinema 4D.

**PREPARING SCENE**

First, prepare your scene. Set the frames range. Make all settings from Cinema 4D's render setup. Do not worry about the output format and the save path because you will set them in the cloud.

**EXPORTING SCENE FOR CLOUD**

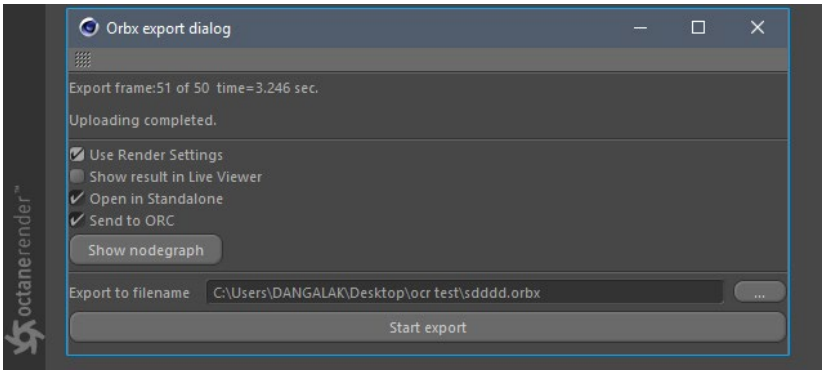
If you finished with the scene setup, go to the "Cloud" menu in Live Viewer and select "Send Scene". As soon as you choose this, a save window will open. Save the scene. The scene will be saved in ORBX format. In this format, problems such as the relative file path is not a problem because the whole scene is saved into a single container file. You can think like FBX. If there are complicated particle/dynamics/hair or cloth systems on your scene, be sure to bake or cache them. The file size may be too large according to your scene structure, it is normal. (If you are wondering what the ORBX format is, you can look at this link). Once you've saved the scene, you will see the " ORBX export dialog" window as you see in the picture below, and the frames will be processed one by one.




The most important of these options is "Use render Settings". So you have used the settings you made in Cinema 4D render settings. If you do not check this, the setup in Live Viewer will be exported accordingly.

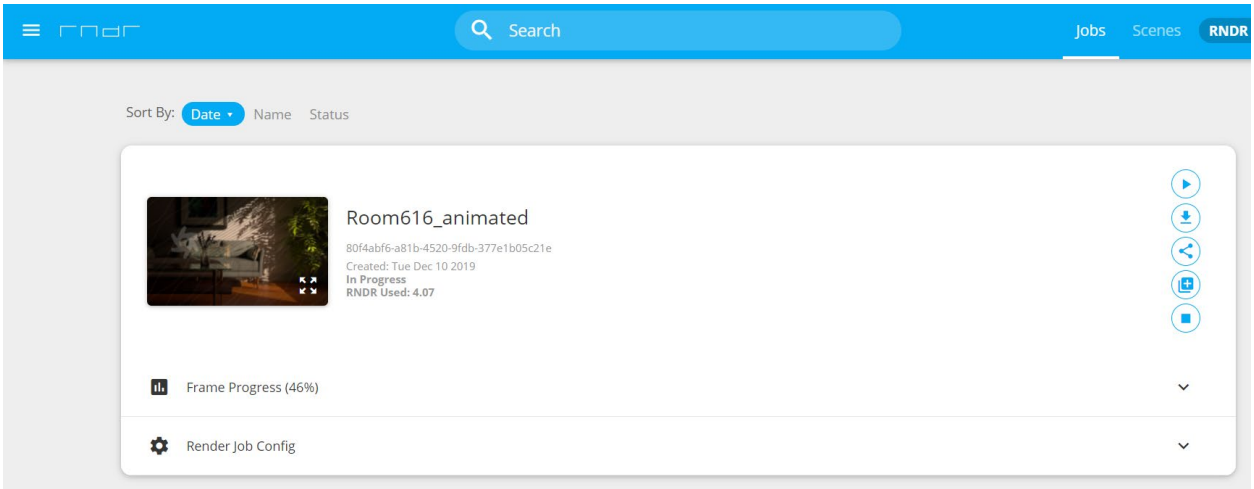
The "Open Standalone" option is for you to view your scene in the standalone software as soon as the export progress finished. This way you can check whether your scene is correct. If you double-click on "Render target" from the "node graph", your scene will start to render. We recommend going through the various frames to check for any errors. You can leave Standalone when you are done.

"Export to filename" is for determining where your orbx file will be saved. Then, you can click "Start Export" and start exporting your scene. Depending on your scene, this progress can be long or short.

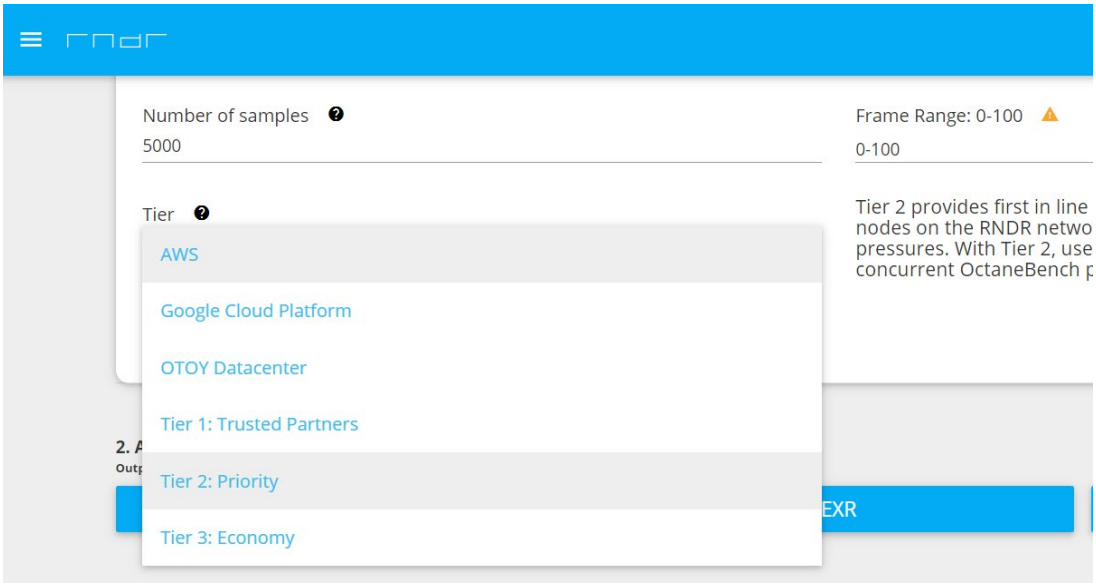


# 3. Artist creates new job with render parameters

Once you have completed this process you will be able to find your Scene on the Scenes page. There you can create Jobs, delete Scenes, or sort Scenes by either date, name, or status. Of course this feature really shines when you upload Scenes in large numbers or with similar names.

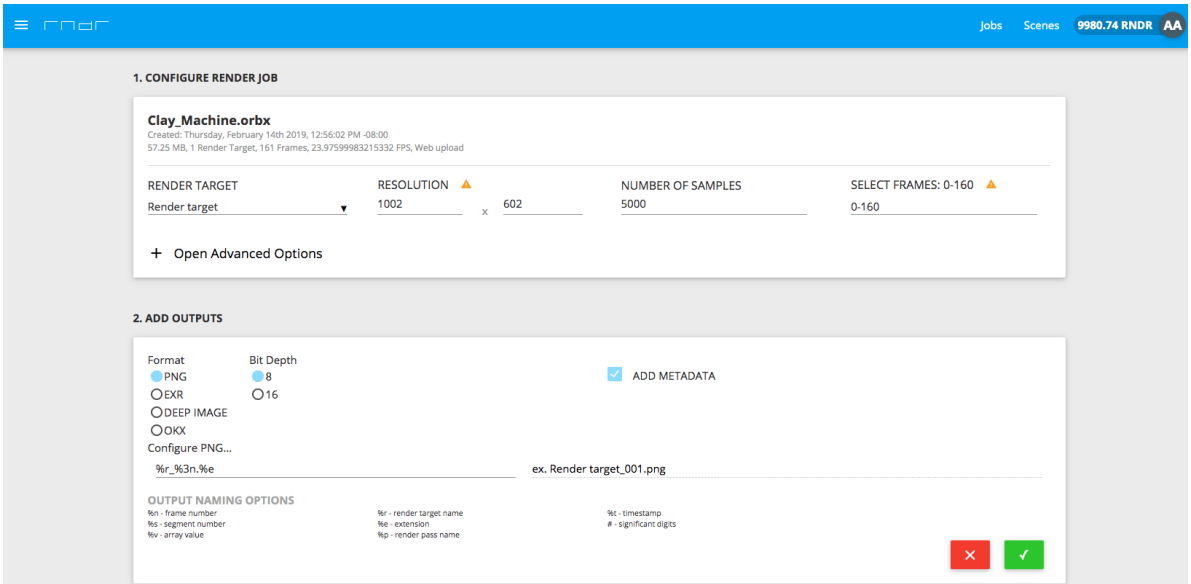



Within the Job parameters, the artist can select what Tier they want to use for their render, between **Tier 1: Trusted Partners**, **Tier 2: Priority**, and **Tier 3: Economy**. The pricing and description of the services in each Tier is described in detail in **Section 1: RNDR Pricing**.




When you have a scene to render, you can create a Job. To begin this process, click the "Create a Job" button your scene. Then you will be redirected to the **'Create a Job**' page.

Here you will have several options.




# Main parameters

- **# of Samples:** Just like in OctaneRender, this sets the maximum number of samples per pixel before the rendering process stops. The higher the number of samples per pixel, the cleaner the render. For quick animations and scenes with predominantly direct lighting, a low amount of samples (500-1000) may suffice. In scenes with lots of indirect lighting and mesh lights, a few thousand samples may be required for a clear result

- **Resolution:** This must be between 1 and up to 65000 for both width and height.

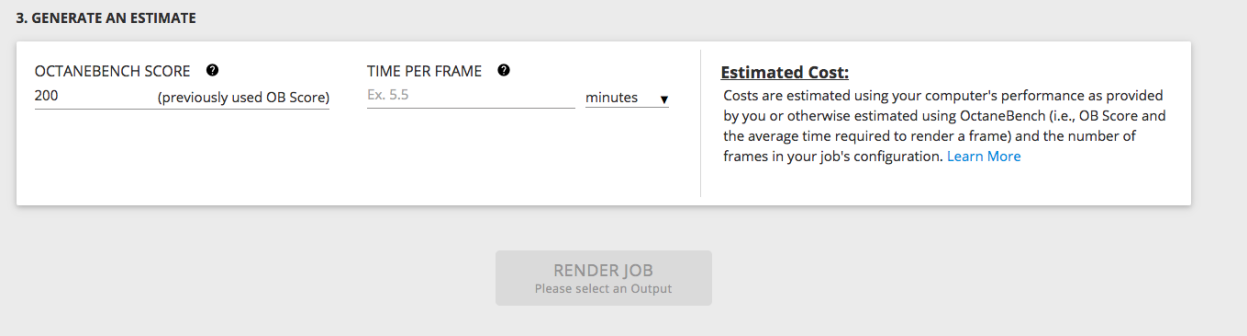
- **Select Frames:** Range of frames you want to render (this can be a single frame, a range, or a mix of ranges)

- **Output:** You have several options for the Format and the Output Naming Format. To change the format, simply click on the format you want. To change the Output Naming Format just enter one or more of the Output Naming Options in the field on the left.

# 4. Artist performs user-side job estimation

Artists are able to generate a cost estimate for a render by providing their local hardware's OctaneBench (OB) score and the time it takes to render one frame before sending the render job out to the network. This is to avoid final job cost exceeding the amount of tokens that were

temporarily locked in a smart contract. It is important to note that this is just an estimate, as there are often significant frame to frame disparities in rendering work that cannot be projected prior to completing rendering work.



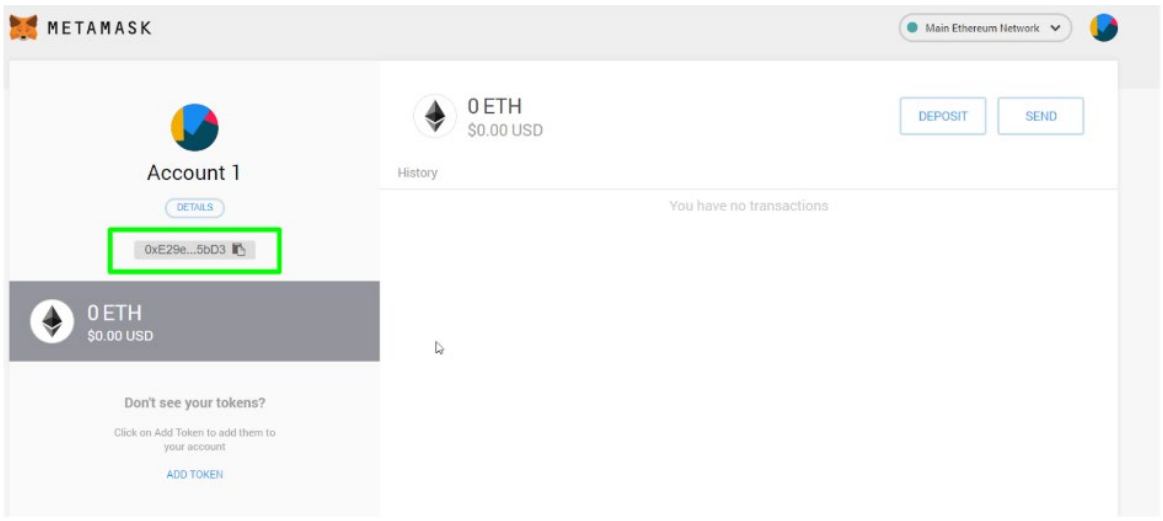

RNDR job estimator

Download the latest version of OctaneBench: http://octanebench.com ...

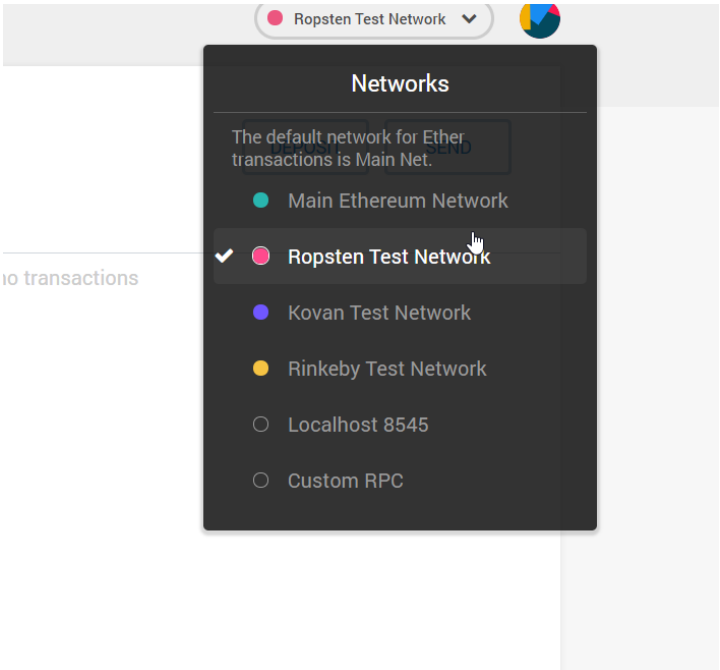
# 5. MetaMask Instructions for artist on-boarding

To use RNDR, the network uses Metamask (Ethereum's payment gateway):

1. To start please visit: https://metamask.io/ and add the chrome extension
2. After the extension is installed, you should **"create a wallet"** and click **"I agree"**
3. The following screen should ask you to create a password. Next Metamask will provide you with a secret back-up (please keep that in a safe place, don't share)
4. Upon confirming your secret back-up phrase, your account will be created
5. You should now be on the page below: Now send us the code that starts with 0x below "Details". Send that address




6. To be on staging, go to the top right corner to switch to the Ropsten Test Network




7. Access the URL to the artist-side web application for RNDR:

8. Please sign up and authorize your account by confirming your email

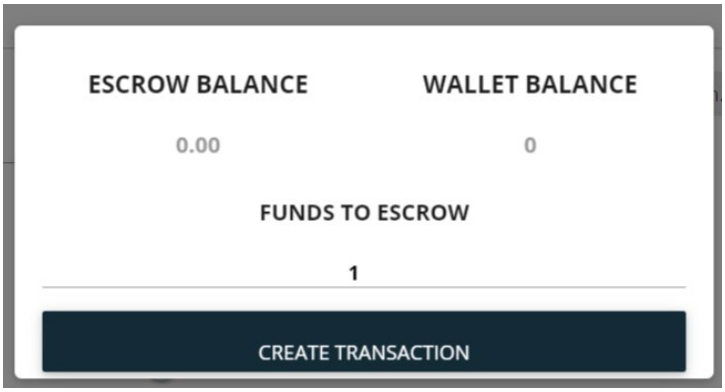
9. Once you are signed in, click the “Scenes” tab on the top right of the screen and upload your .ORBX scene file

10. Once uploaded, click **“Create Job”**

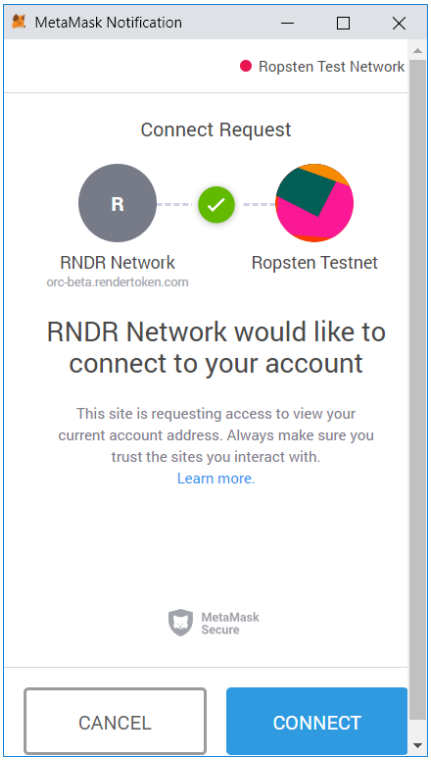
11. Now follow the instructions in this video for submitting a job and setting the parameters:

# Uploading a Scene with Cinema4D

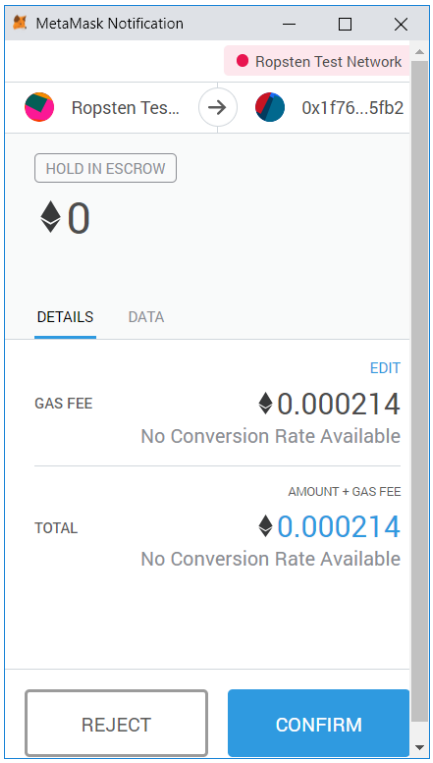
Artists add RNDR Tokens or an amount of currency to their Render Account. An Artist can add an estimated amount of tokens to cover a single rendering job, or add more tokens to their account in order to process current and future rendering jobs.




**Step 1**: When you click the top right user button, send RNDR or fiat from your wallet




**Step 2**: Connect RNDR to your Metamask wallet




**Step 3**: Confirm the amount from above

### USING RNDR AUTH Permissions for Credential Management

Within the RNDR Account Portal, AUTH Permissions allow an administrator to grant & restrict permissions to individual users within their studio. For example, an administrator could create an access token with only the ability to upload scenes, while another access token only has permission to run jobs based on the scenes already uploaded. This provides granular permissions to users within an organization and better enables corporate governance. This allows a studio to manage usage of RNDR Tokens, while also enabling other users to access a studio's network account for collaborative workflows. AUTH permissions allow an administer to both manage and attribute tasks in an account.

### RNDR User-based locking system

Tokens are temporarily locked in a smart contract prior to beginning rendering work to guarantee that tokens in a user's wallet will not move while work is being performed. The tokens are locked in a smart contract with a mapping to the user within the internal system. The locking smart contract is agnostic meaning many different ETH addresses could fund a single user. The tokens available for a specific user can be increased at any time but cannot be withdrawn once locked. An OTOY-International address is the only address able to release tokens.

## 7. RNDR Servers Distribute Frames

Job Allocation is currently organized so that the job that has gone longest without work is matched to the node that has gone the longest without work assigned to it. However, the system also has implemented matching processes that ensure a Node will have sufficient memory (VRAM and System Memory) for processing the job prior to assignment. The RNDR Network collects data about job history and system hardware to optimize job allocation and job matching.

## 8. Miners Receive and Process Frames

Miners receive their job parts, process them and send results (rendered frames) back to RNDR Network.

**Process Flow for Miners**: Download an executable program from OTOY's website → input your wallet ID and run application → connect to RNDR → look for jobs → download scene to system memory → render & periodically send status back to central server → report job complete → upload to storage server → check node availability → repeat cycle

Here is a short description of the steps involved for the nodes to download and run the RNDR token application.

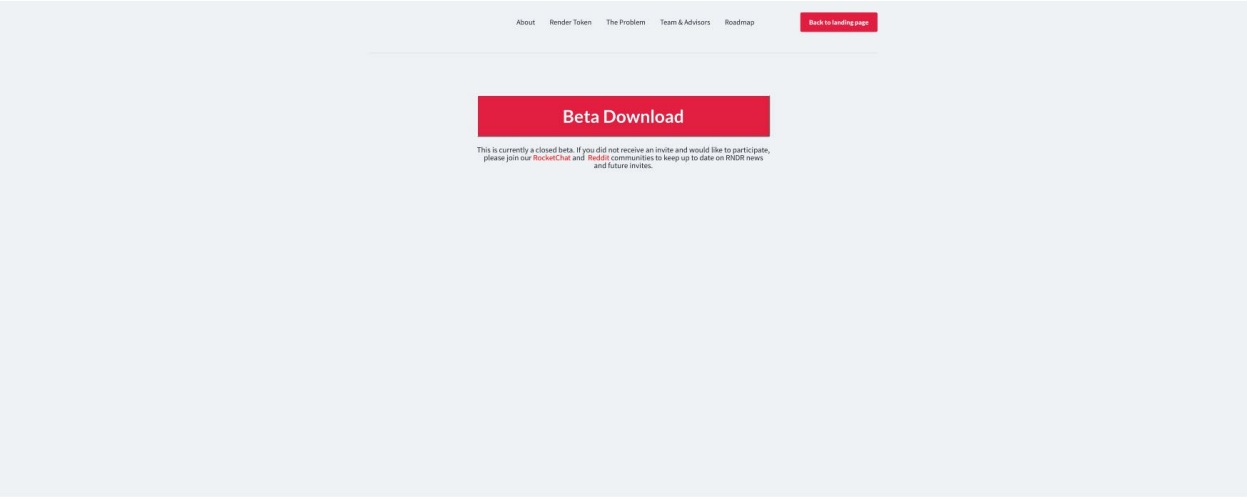
1. Access the URL provided by the RNDR network in onboarding process.

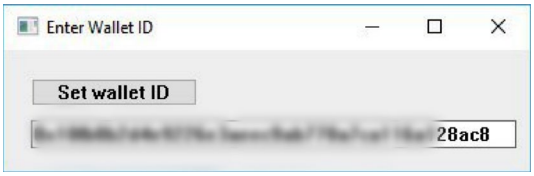
2. Once prompted, fill in the credentials sent by the RNDR team.

3. You'll be directed to a screen with the beta link. Click the Download button and wait for the .exe file to install.

4. A screen pops up for users to enter the ETH wallet ID they wish to receive the tokens in. The same wallet IDs are used to identify miners. RNDR does not own or hold any information in regards to these wallets, they are accessed by Metamask.

5. Click Close and that's it! The RNDR Token application will start running.



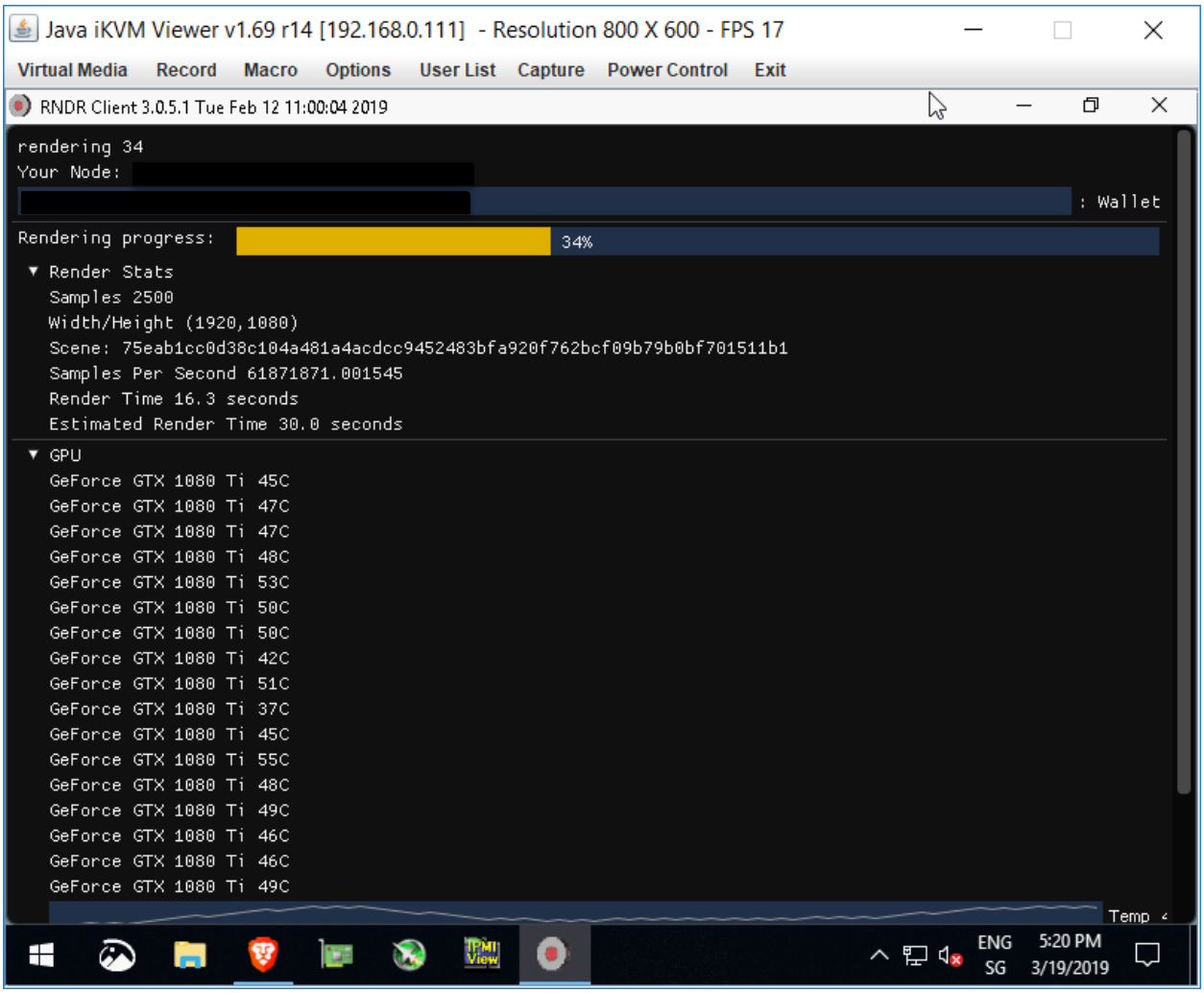

After completing these steps, nodes are now in the network and can pick up frames from submitted jobs.

Those who contribute GPU processing power can see the progress of each frame, the frame number from the scene, and the status and utilization of the GPUs and VRAM (free VRAM vs. used VRAM).

While the application is running, it shows different messages for the node:

- **Downloads done:** Downloading to memory. In this particular workflow, each frame is downloaded one step at a time.
- **Sha256:** Verification hash function.
- **Node waiting on message/Node message received:** A node is waiting on a job to be posted/ A job was sent to the node.

If the node‘s administrator decides to close the application during rendering, it will render the last frame and send it back to the artists before it closes the application. Be aware that this is only if they exit from the system tray; you can close the window and rendering will not be interrupted.

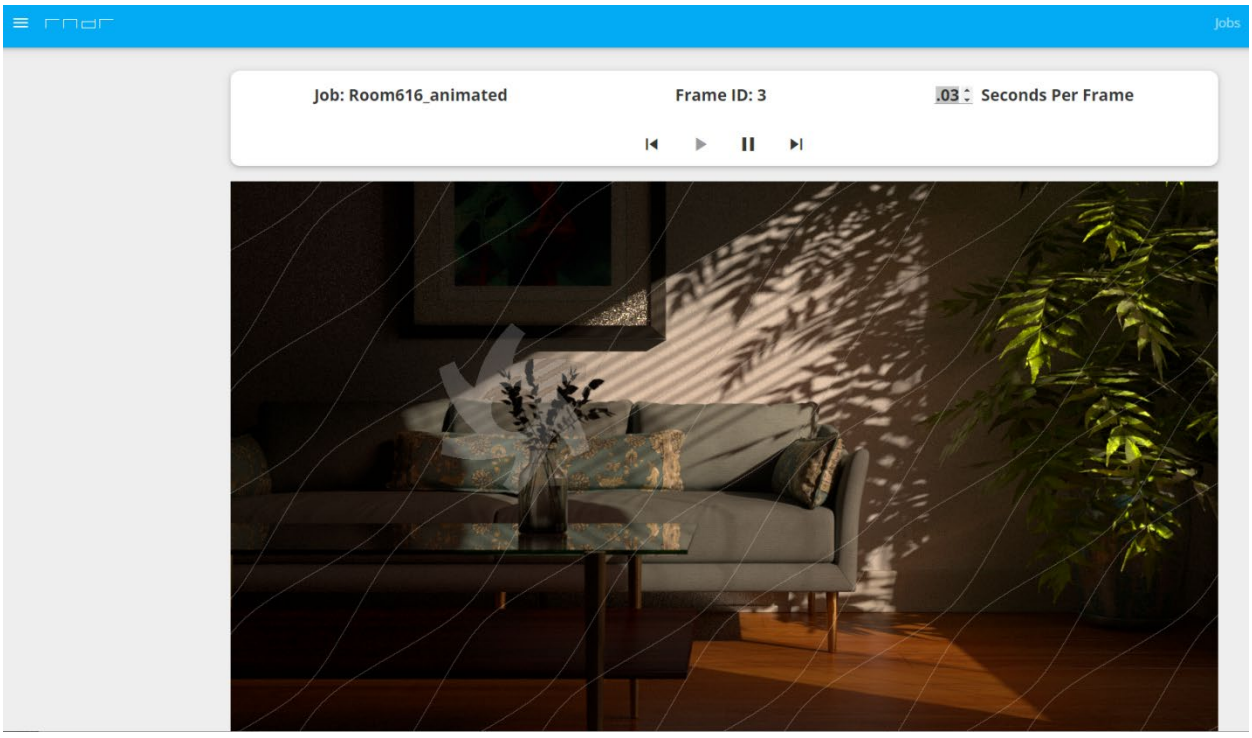


The node client will reuse assets when it can once those scene assets are held in local node memory. This minimizes setup time and improves the rendering to general work time ratio. The node wants to hold onto as much as possible for later work, and there's some intelligence about when it downloads assets vs clearing out what is unlikely to be reused.

# 9. Artist reviews, approves and downloads results

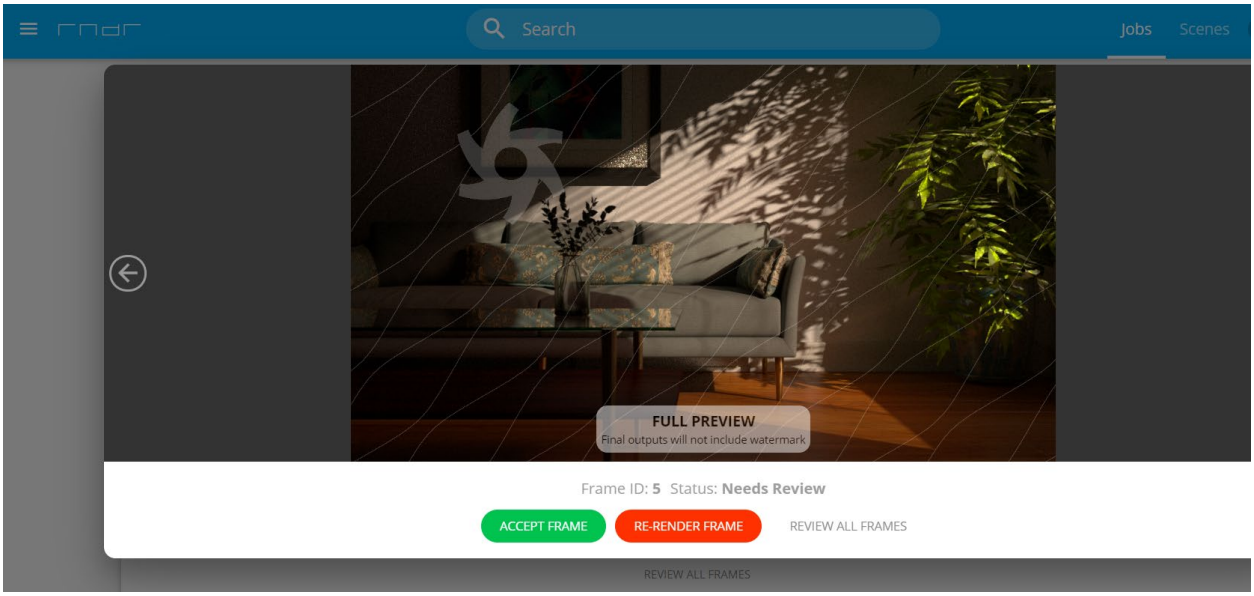
RNDR is a Two-Sided network of Miners and Artists. In order for the network to function, payments need to be made expediently to miners, and artists need mechanisms to ensure they are receiving the requested Rendering work. RNDR has implemented a frame approval system to accelerate job confirmation, reducing payout latency, and providing a dispute mechanism for artists who get failed renders.

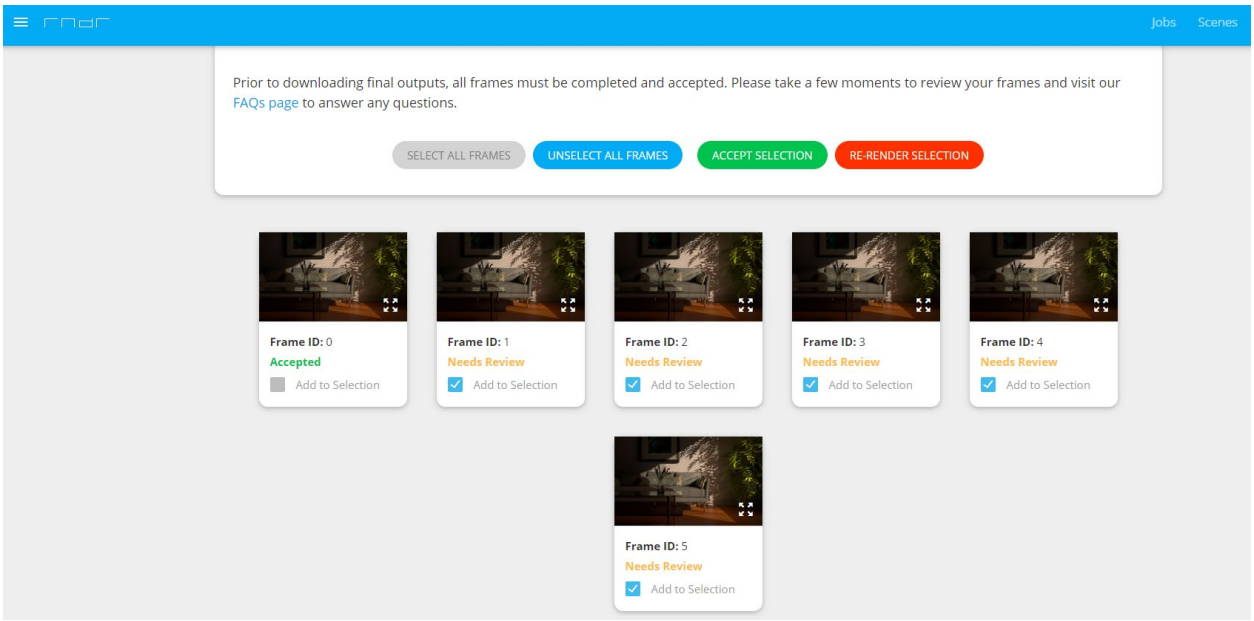
Artists are able to review frames as they are rendered with a full watermarked preview. This allows an artist to ensure their renders are properly executed. Additionally, artists have the ability to watch an animated preview, and can control how long each frame is displayed.




As Frames are rendered and after a job has been completed you can individually approve or reject each frame. You can also approve frames in batch sequences. For example, at frame 1000, you can approve all previous frames from 0-1000 in one click. Once a frame is approved,

the tokens will be debited and used to pay for the approved frames. Once you have successfully paid for a frame, you can download the frames in your Artist Portal.




You can approve frames individually frame-by-frame or select all frames, or a larger sequence of frames (see graphic). This can happen during or after the job is complete.

***Please note that if no approval or rejection of frames or frame sequences is given within 72 hours of Job completion, all frames will be automatically approved.***

In some cases, you will need to add RNDR Tokens or currency in order to download the approved frames. See Here for more information on adding Tokens or currency to your account.

## Frame Rejection

Select the Individual Frame or Sequence of Frames and click Reject. Once a Frame is Rejected, it will automatically be sent to another node on the network to be re-rendered. You ***will not*** be charged additional RNDR for this work. However, you will also not be allowed to modify the frame or sequence in any way. Jobs should only be rejected if there was an error processing the frame like black frames, partially rendered frames, or broken frames.

After three failed re-renders, a job will be terminated and a support ticket will be automatically created. A support team member will review the failure and follow up. For support inquiries please contact support@rendertoken.com

## Cancellation vs Frame Rejection

If, for example, you want to modify the Sample rate or change elements in a scene like the camera, lighting, color etc., please cancel the Job, make the changes and Submit a new Job to be processed. This means it is very important to check a Job prior to rendering.

All correctly processed rendering work will be paid for as the network needs to compensate miners for their work in order to be sustainable. See Job Cancellation FAQ's for more details on cancelling scenes you no longer want rendered.

## Frames Automatically Approved After 72 Hours

If no Approval or Rejection of frames is given within 72 Hours after Job completion, all frames will automatically be approved. RNDR is a two-sided network of Miners and Artists, and Miners who have hard infrastructure costs need to receive Tokens in a timely manner so that they are able to maintain operations. For this reason, release of tokens needs to be expedient. With rapid confirmation, Miners are able to more easily use the network, building critical GPU supply that is intended to lower costs and increase availability for Artists.

## Adding Tokens or Currency to Download Frames

It is not possible to estimate Job costs with 100% accuracy prior to actually doing the rendering work. Often times there is significant variance in the rendering work required between frames as things like the camera positioning, lighting, and a host of other factors change from one frame to another. For this reason, the Tokens in the locking smart contract are only a base estimation, and the actual rendering work will occasionally exceed the initial estimation.

When a job exceeds Tokens in the locking smart contract, it will automatically be paused, and a notification will pop up asking you to add Tokens or currency to your account. However, all the frames that are currently in progress will be rendered. Once approved, completed frames can

be downloaded. In order to continue a job, you will need to add Tokens or currency to your account and resume the render.

## Job Cancellation

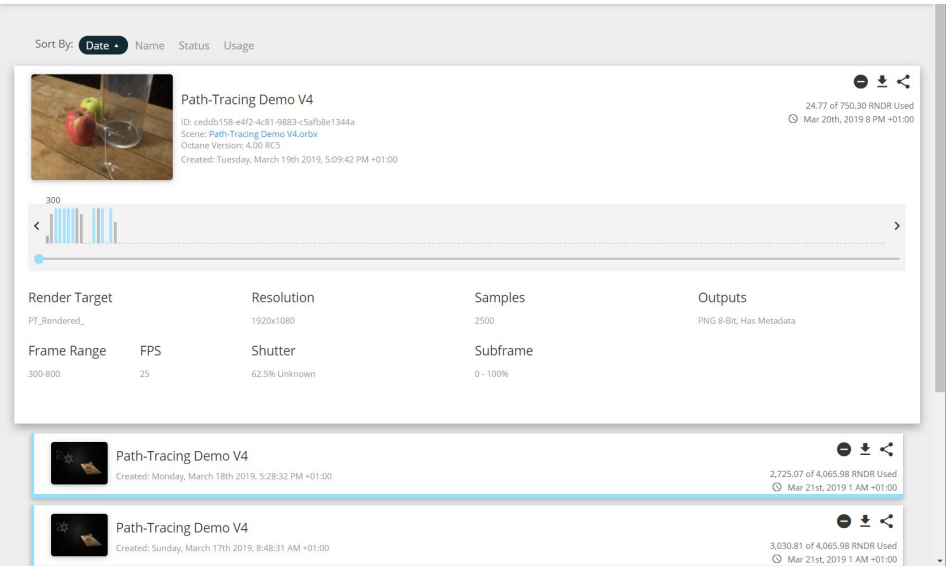
Jobs can be cancelled at any point. However, in progress frames will be completed and miners will be compensated for this rendering work, with RNDR released from an Artist's wallet for the completed frames. These frames will then be available for download.

## Review Process for Approval

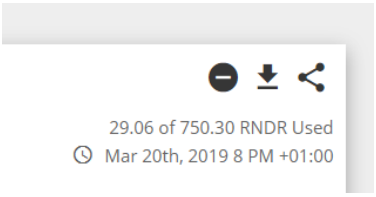
By default, you will be shown your Rendered Jobs. Here you will see a small preview of your Job as well as be able to inspect, delete it, download it, or share your Job.

Each column represents a frame in the scene and will be displayed in the preview box. Since this an animated Job we have many columns to choose from but in the case of a still, there will only be a single frame which will be represented by a single column.

Sharing and downloading animated Jobs can be done after a job is paid and downloaded.

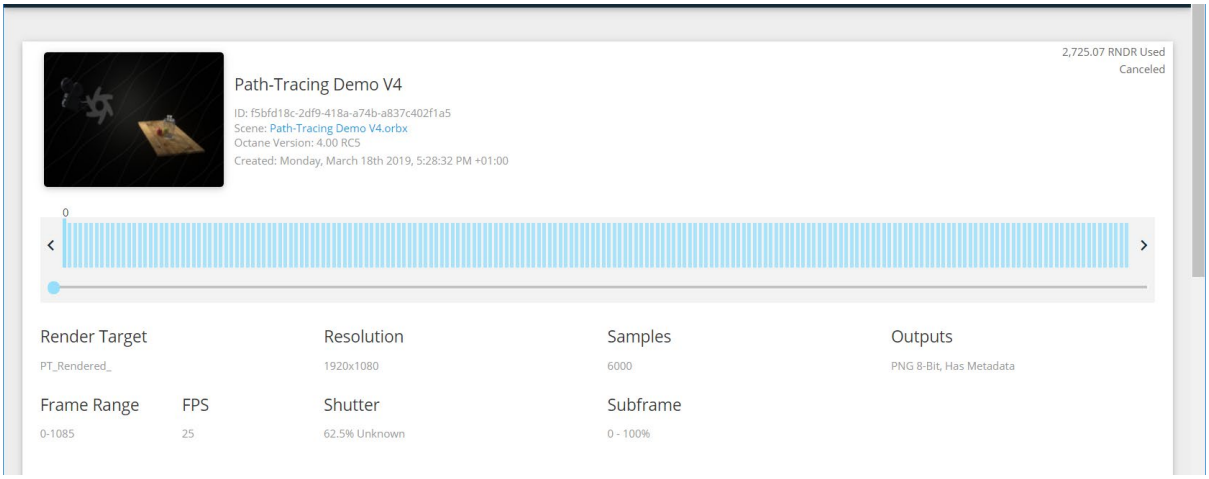


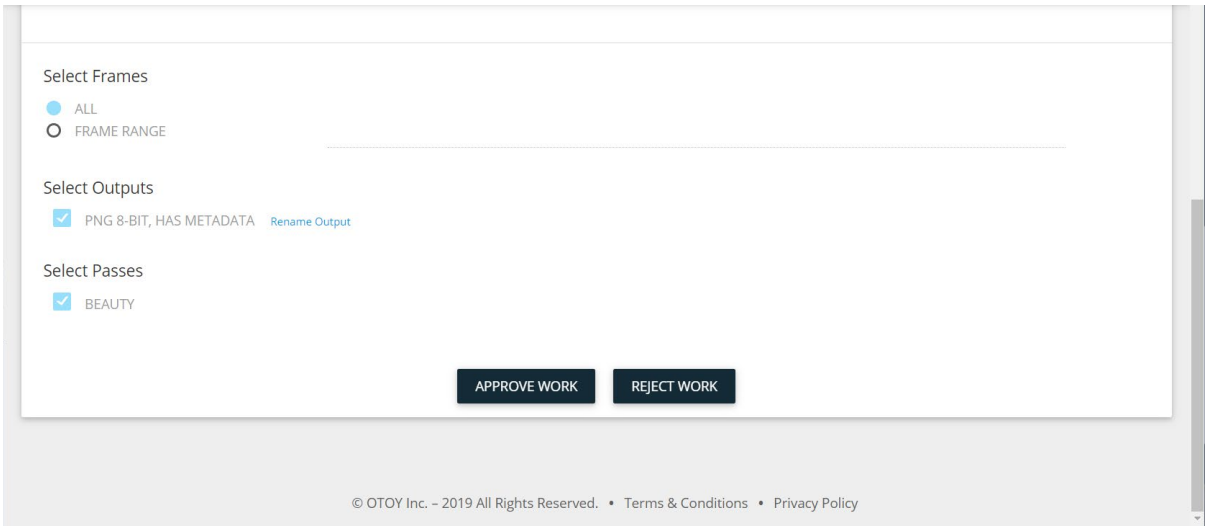

The last and most important feature of the Jobs page is being able to delete, download, and share your Job. Which are represented by the 3 icons on the right side of your Job.



Artists can delete, download or share a Job. Downloading and Sharing a Job operate similarly. When either function is selected, it will redirect you to another page where you will be able to select the range of frames and output format(s) that you want to download/share.

Once you have done that click on the button below and you will be redirected to a page where you need to approve the work first. If approved, the release of Tokens from the locking smart contract to the nodes is triggered and you are granted access to download the final, watermark free images.




In the case of sharing your Job you will be redirected to this same page but will also have a link that you can give out at the top of the page.

The artist receives preview thumbnails/JPEG images watermarked with an Octane logo and is asked to approve or decline the renders frame-by-frame. Watermark-less downloads will be restricted until after artist approval has been received.

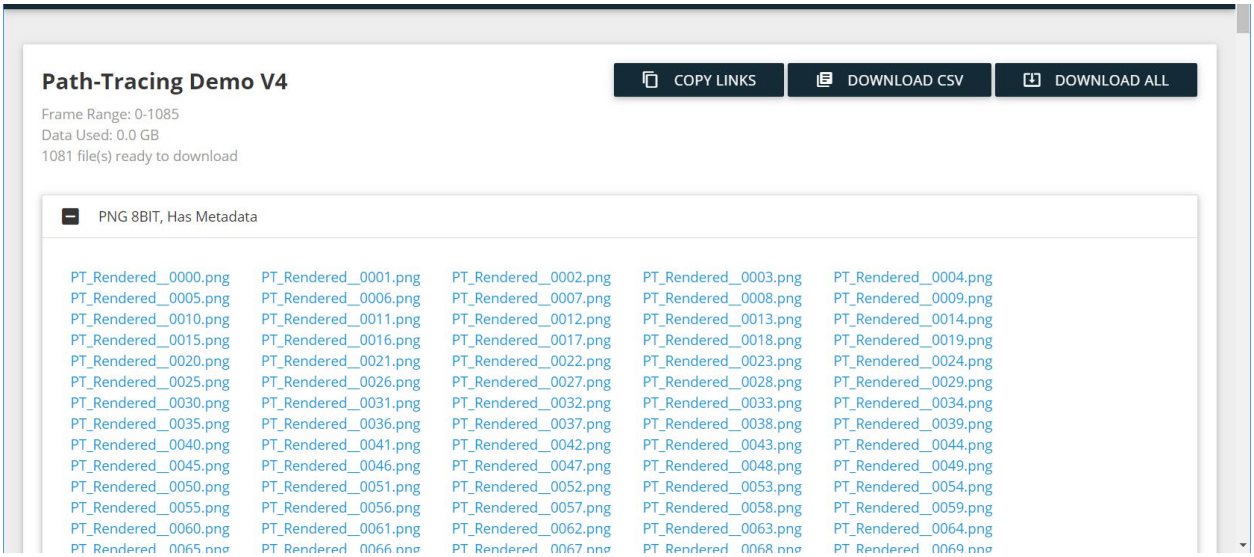
If approved, the release of tokens from the locking smart contract to the nodes is triggered and the artist is granted access to download the final, watermark free images. If declined, the artist never gets to obtain those downloads and the workers whose frames were declined don't get paid.

# 10. RNDR Smart Contract Releases Tokens to Miners

After artist approval of after the 72-hour period after a frame is completed, Tokens will be released from the smart contract and paid to miners.

# 11. Artists downloads rendered results

After Tokens are released from the locking smart contract, an artist can download the rendered frames.



(screenshot of the download page where an artist can download their results)

Artists will have a CSV report of their Job, as well as the ability to batch or individually download the files to their desired Folder on their computer.