

# docs.yom.net — Snapshot

## ( - - )

*Clean documentation content extracted from sitemap.*

## Introduction | YOM Docs

**Source:** <https://docs.yom.net>

Gaming has transformed dramatically over the last decades, driven by technological advancements that redefine player interactions and immersive experiences. Today, three main methods dominate gaming consumption:

- **Mobile Gaming:** Easy development and widespread access, but limited by hardware constraints and app download friction.
- **Console / PC Gaming:** Premium graphics and high performance, but requires costly hardware, significant downloads, and lengthy installations.
- **WebGL Gaming:** Instant access via browsers without installations, yet struggles with optimization and performance across various devices.

Each of these models comes with trade-offs that force developers to balance accessibility, quality, and cost efficiency.

## Pixel Streaming—The Game Changer

Pixel streaming—commonly known as cloud gaming—eliminates hardware barriers completely, enabling AAA games to run instantly and seamlessly on smartphones, tablets, smart TVs, and browsers.

Our vision: pixel streaming will soon become the **standard across all gaming platforms**, due to its unique capabilities:

- **Cross-Device Compatibility:** High-quality gameplay instantly delivered to any device, without installation or compatibility concerns, significantly increasing accessibility and developer reach.
- **Cross-Channel Integration:** Games easily embedded into social media (TikTok, Instagram, Facebook), e-commerce sites (Amazon), and apps, multiplying touchpoints for user engagement.

- **Superior Delivery:** Console-level performance independent of local hardware limitations, ensuring smooth user experiences with seamless transitions across devices.

This revolutionary approach allows websites, apps, and platforms to independently serve sophisticated, interactive content, reshaping digital strategies for content creators, brands, and game studios alike.

Beyond gaming, pixel streaming is poised to accelerate the Open Metaverse, enabling instant global delivery of immersive content without dependency on specialized client hardware—bringing digital twins, virtual real estate tours, personalized AI interactions, and interactive episodic experiences to life instantly.

## The Centralization Problem

Yet, despite its promise, pixel streaming faces a major bottleneck: centralized infrastructure. Current cloud GPU solutions rely heavily on costly, inefficient, and environmentally unsustainable data centers. Massive investments in cooling, maintenance, and continuous hardware upgrades drive soaring operational costs and significant carbon emissions. A handful of technology giants hold near-total control, limiting accessibility and stifling innovation.

Simultaneously, GPU computational power underpins critical advancements—from AI and robotics to cloud gaming and the emerging Metaverse. Demand is exploding, but traditional centralized models simply cannot scale efficiently or sustainably.

Notably, — **% of global GPU capacity currently remains idle in consumer devices**—an untapped resource representing a vast pool of computational power. Leveraging this idle capacity could effectively double available GPU resources worldwide, providing a highly scalable and cost-effective solution.

## Why Decentralization Is the Future

Cloud gaming is the perfect candidate for blockchain-enabled decentralization. Unlike AI training or batch processing (which tolerate centralized infrastructure constraints), cloud gaming requires:

- **Ultra-low latency**
- **High availability**
- **Flexible scalability**

Traditional cloud providers struggle with these critical requirements.

YOM's peer-to-peer cloud gaming network flips the script, utilizing the untapped GPU power of household gaming rigs and consumer-grade PCs. It offers a purely serverless pixel-streaming model—eliminating the need for centralized data centers altogether.

This innovative decentralized approach offers clear advantages:

- **Drastic Cost Reduction:** Eliminating capital expenditures (CAPEX) entirely and significantly reducing operational expenditures (OPEX), bringing costs from typical \$ 100 - 200 /hour down to ~\$ 10 - 20 /hour.
- **Global Reach, Local Performance:** Decentralized architectures naturally expand globally while providing localized connectivity—delivering ultra-low latency (sub-100 ms) and exceptional user experiences.
- **Economic & Environmental Liberation:** Transferring power from large centralized cloud providers back to gamers and hardware owners, while significantly reducing environmental impact by recycling existing GPU resources.

[NextWhat is YOM?](#)

Last updated 10 days ago

Was this helpful?

---

## What is YOM? | YOM Docs

**Source:** <https://docs.yom.net/manifesto/what-is-yom>

YOM is at the forefront of decentralized cloud gaming, harnessing pixel streaming technology via a global network of distributed idle GPU resources. Leveraging this unique approach, YOM drastically reduces cloud gaming infrastructure costs by up to 90%, making premium interactive experiences instantly accessible, sustainable, and economically viable for both creators and consumers.

Here's how YOM works at a glance:

- . Player clicks *Play* → client pings the discovery ledger.
- . HyperOrch scores nearby nodes on latency, load, reputation.
- . Selected node handshakes and streams frames.
- . Fail-over kicks in automatically if the node drops.
- . Node operator earns \$YOM tokens; publisher pays \$ 100 - 200 /session.

*(Protected by three key patents covering secure edge bootstrapping, intelligent orchestration, and decentralized pixel streaming.)*

# Who Benefits?

Here we visualize the YOM ecosystem:

Figure : YOM Ecosystem Flow and Interactions

## Players

- **Instant play**—AAA visuals on even low-end devices
- No downloads, patches, or hardware upgrades

## Node Operators

- Boot from the **NANO** drive → start earning in minutes
- Typical rewards \$ /month per pc at % utilisation
- - month hardware pay-back; withdraw rewards in \$YOM

## Publishers & Studios

- \$ . per session + keep % of game sales
- Embed playable demos on sites, socials, or ads in < min
- UE & Unity SDKs support custom HUDs, widgets, auth, DLC stores

[Previous](#)[Introduction](#)[Next](#)[Infrastructure](#)

Last updated minutes ago

Was this helpful?

---

## Infrastructure | YOM Docs

**Source:** <https://docs.yom.net/manifesto/infrastructure>

YOM's architecture is purpose-built for decentralized, high-performance cloud gaming. It combines a custom pixel streaming solution, secure edge computing, and intelligent decentralized orchestration to deliver interactive streaming with near-local responsiveness and reliability.

The platform spans from a proprietary pixel streaming services and developer SDKs to the underlying network of trusted GPU nodes coordinated by an AI-driven scheduler. Together, these components enable any AAA game to be streamed instantly to any device with sub- ms latency, high reliability (through rapid failover), and end-to-end security – all at a fraction of the cost of traditional cloud gaming.

## **. Peer-to-Peer Pixel Streaming**

YOM has developed its own lightweight pixel-streaming service that runs games on remote consumer-grade GPUs and streams the video feed back to players via WebRTC for minimal latency.

Once a session is approved, the gameplay is delivered over a direct peer-to-peer connection between the node and the player's device.

YOM's streaming protocol establishes an encrypted video/audio stream and input channel without routing through any central server, minimizing latency by eliminating extra hops. Game audio/video and player input data are end-to-end encrypted, so the content remains private and secure even if it traverses relays or public networks.

## **. AI-Powered Orchestration**

Workload allocation across the YOM network is handled by HyperOrch, YOM's decentralized orchestration engine.

HyperOrch continuously gathers telemetry from each node – a multi-dimensional “capability vector” including available GPU/CPU capacity, memory, bandwidth, encoder load, thermal headroom, and more. Similarly, each game title is characterized by a resource footprint profile (GPU demand, memory needs, expected bitrate, etc.).

Using these inputs, HyperOrch's AI-driven scheduler predicts the expected performance (e.g. framerate) for a given game on each candidate node and dynamically matches players to the optimal node in real time. Unlike naive approaches that just pick the nearest server, HyperOrch filters and ranks nodes by multiple criteria (player-to-node latency, hardware suitability for the game, current load, etc.) to find a host that can deliver smooth + FPS and < ms ping for that session.

The orchestrator even learns from each session: it updates its model with actual performance outcomes, continuously improving placement decisions over time. This intelligent scheduling maximizes network-wide performance and efficiency – avoiding bad host-game pairings that might lag, fully utilizing available GPU headroom, and reducing costs by optimally balancing the load on community nodes.

In practical terms, HyperOrch's real-time decision-making ensures each player is connected to a node that will give them a console-quality experience, and it can proactively migrate or redistribute sessions if a better match becomes available.

## . Developer SDKs

The YOM Player Integration Library lets you embed live YOM game streams in any web property via **two approaches**:

Method

Use Case

Key Benefit

### Basic Embed

One-off landing pages, quick demos

Copy-and-paste `<iframe>`

### Custom UI Embed

Full-fledged portals, dynamic controls

Script-based API + UI state manager

Both methods handle secure messaging between your page and the YOM Player, manage UI transitions, and expose an event/command API.

YOM provides optional Unreal Engine and Unity SDKs that integrate the streaming service seamlessly into their games. These SDKs let developers incorporate responsive HUD overlays, site widgets that can communicate directly with the stream, and custom authentication or monetization flows to tailor the player experience.

## . Node OS

The nodes boot a minimal, read-only Linux image. On first launch the operating system:

- **Auto-detects hardware & installs signed GPU drivers** so the discrete card is exposed at full performance (no user intervention required).
- Runs a **registration module** that authenticates the user, measures uplink bandwidth, RTT to regional probes, idle GPU capacity, and temperature; it then publishes this capability vector to the discovery ledger for HyperOrch scheduling.
- Provides an isolated execution sandbox that mounts each game image read-only, passes the GPU through exclusively.

During gameplay the OS reports live telemetry (FPS, -th percentile latency, thermals); if thresholds are exceeded it can dynamically down-clock the game

container or request HyperOrch migration. On shutdown—or if the boot drive is removed—the OS triggers a **rollback routine** that asserts a PCIe reset, DMA-scrubs RAM and VRAM, clears any drive keys, and reboots, ensuring no player data or studio IP persists. Updates follow an atomic A/B scheme: a new signed image is written to the inactive slot and only activated after signature verification and checksum pass, with automatic rollback on failure.

## .Decentralized Discovery & Attestation

Nodes publish signed availability records (location hash, GPU class, live load) to a *distributed hash table / on-chain registry*. Clients query the same ledger, pick candidates within a latency radius, then request the node's latest attestation. If the proof checks out the client proceeds; if not, the node is black-listed until it re-establishes trust. This **serverless matchmaking** eliminates single points of failure and allows the network to scale organically while remaining secure.

## .Node & Stream Management Portal

Operators and studios manage everything through a secure web dashboard. **Node owners** view uptime, earned rewards, temperature metrics, and update status; they can schedule maintenance windows or toggle delegated hosting. **Publishers** upload and manage builds, set session budgets, and track live QoE metrics (latency heat-maps, FPS histograms).

[Previous](#)[What is YOM?](#)[Next](#)[Node Reputation](#)

Last updated      days ago

Was this helpful?

---

## Node Reputation | YOM Docs

**Source:** <https://docs.yom.net/manifesto/node-reputation>

XP serves as a soul-bound metric in the YOM ecosystem, meaning it is permanently tied to each user's unique identity and cannot be traded or transferred. XP rankings appear on both short-term (weekly or monthly) and all-time leaderboards. This dual display encourages fresh participants to climb the ranks while acknowledging veteran members for their longstanding dedication. Here are some of the benefit of high XP:

- . **Enhanced Node Rewards** The more XP a node operator has, the higher their position on the operator leaderboard. This can translate into increased session allocations or bonus multipliers for node earnings.

- . **Airdrops & Exclusive Perks** High-ranking XP holders periodically receive **\$YOM drops** or discounted access to partner programs. They may also enjoy early access to premium features, events, or beta releases within the YOM ecosystem.
- . **Governance Influence**
  - **Voting Power:** While holding \$YOM typically grants voting rights, XP acts as a **multiplier**—the most engaged and knowledgeable community members gain proportionally greater voting weight.
  - **Proposal Privileges:** Certain high-XP brackets may unlock the ability to propose major changes or direct resource allocation in the DAO, guiding crucial development decisions.
- . **Social Recognition** Exclusive badges and titles—displayed within the YOM dApp, community forums, and in-game environments—help top XP earners stand out. These digital “status symbols” showcase trustworthiness and expertise to other users.

## How to earn XP?

The YOM dApp and [Discord](#) facilitate easy access for members to participate in activities where they can earn XP, enhancing their engagement and influence within the YOM community. Core earning avenues are:

- . **Holding \$YOM** Simply by holding \$YOM, your wallets accrues XP.
- . **Node Operation & Reliability** Node owners who run stable, high-uptime machines accumulate XP in proportion to the reliability and performance of their nodes. This ensures that those contributing critical infrastructure earn corresponding reputation points.
- . **Community Engagement & Content Creation** Active Discord or forum participation, creation of high-quality guides or tutorials, and social media advocacy all generate XP. This mechanism rewards both the technical and cultural contributors who help grow and educate the community.
- . **DAO Participation & Governance**

Authors of successful proposals—those that pass and benefit the platform—gain additional XP whilst regular, informed voting on proposals further builds a user’s reputation in the community.

- . **Mod/Peer Voting & Seasonal Challenges:** Periodic events hosted by the YOM Foundation or community-driven committees can offer bonus XP for tasks like stress-testing new features or helping onboard new members. Certain XP awards



are distributed through peer recognition, where outstanding contributions or helpful community moderation receive upvotes or endorsements from other members.

[Previous](#)[Infrastructure](#)[Next](#)[Tokenomics](#)

Last updated      days ago

Was this helpful?

---

## Tokenomics | YOM Docs

**Source:** <https://docs.yom.net/manifesto/tokenomics>

The \$YOM token is the economic linchpin of the network, with its deflationary model and reward dynamics carefully designed to balance short-term incentives (node rewards, studio growth) with long-term scarcity ( % burn on each transaction).

## Medium of Exchange

\$YOM serves as the primary currency for accessing and operating cloud gaming sessions. Studios, brands, and broadcasters purchase \$YOM to power these streams, creating a continuous demand tied directly to network usage.

Node operators who contribute GPU resources are paid in \$YOM, ensuring that high-performance nodes with greater uptime and reliability capture higher earnings. This token-based reward model fuels a self-sustaining cycle: more users and games → more sessions → more demand for \$YOM.

### It's Network Fuel

The token has a strict deflationary structure: no new tokens can ever be minted. Instead, Whenever someone plays a game on the network, a percentage ( %) of the network rewards are burned. Since every purchase of \$YOM directly eliminates a portion of tokens from the market, the float becomes scarcer over time. This design boost long-term token value whilst aligning token holders with network usage.

### Liquidity Provision

While node operation is a key earning avenue, community members can also use \$YOM to provide liquidity on decentralized exchanges. These activities not only secure the network and facilitate smooth token trading, but also offer participants a passive way to accrue additional rewards.

# Flow of Value

Publishers utilize \$YOM tokens to pay node operators for running the sessions. The payment usually starts as a recurring FIAT payment, which gets automatically swapped into \$YOM. Below is the distribution key:

Stakeholder

Distribution %

Description

## Nodes

The majority of the rewards go directly to node operators to incentivize their participation and ensure high uptime and performance.

## Network

Funds allocated to infrastructure improvements, software updates and scaling operations.

## Burned

This mechanism keeps the token supply deflationary, aligning all stakeholders.

As node operators receive \$YOM, some choose to hold to accumulate more XP, become active in the DAO, or provide liquidity on decentralized exchanges, further fueling the network's growth and liquidity. Others may convert tokens to cover operational costs.

[Previous](#)[Node Reputation](#)[Next](#)[DAO](#)

Last updated          minutes ago

Was this helpful?

---

## DAO | YOM Docs

**Source:** <https://docs.yom.net/manifesto/dao>

A key pillar of our mission towards decentralization is the Decentralized Autonomous Organization (DAO), which coordinates decision-making processes on-chain and in democratic fashion.

The DAO is tasked with overseeing the network's foundational governance and long-term vision. Representing the DAO is the YOM Foundation is a non-profit entity.

## Decision-Making

The primary layer of governance influence is embedded within XP (Experience Points). The higher a participant's XP, the more weight their votes carry, ensuring that engaged contributors who have demonstrated consistent, positive impact in the ecosystem shape its evolution.

All major decisions, including votes, funding distributions, and policy changes, are recorded on-chain. This transparent ledger reinforces accountability and ensures that any community member can audit the decision-making process at any time. The DAO manages:

- **Protocol:** Anything that impacts the network at large, such as security standards, advertising guidelines, and core protocol updates.
- **Tokenomics:** Any change to burn rates, reward structures, or additional deflationary mechanisms must be vetted and approved by the DAO.
- **Infrastructure:** Key updates to the underlying protocol, such as improved node benchmarking or new streaming standards, are decided collectively.
- **Budgeting:** A dedicated treasury, supports grants, development incentives, ecosystem programs (BUILD/EARN), and other community-led initiatives.
- **Moderation:** DAO-elected moderators ensure content alignment with broader ecosystem goals and uphold community standards.

Over time, more aspects of YOM's operations will migrate on-chain, from resource allocation in ecosystem programs to deeper integration of node management. This shift aims to minimize central points of failure and consolidate community ownership. Initiatives will increasingly rely on DAO proposals and votes to select and fund recipients. This progressive handover of program administration—from Foundation-led to entirely community-driven—further empowers the network's most active members.

As the platform matures, a series of transparent milestones will detail how specific powers, assets, and policy controls are transferred from the Foundation to the DAO. The ultimate goal is to establish a completely trustless, self-governing environment in which all critical decisions emerge from community consensus and aligned economic incentives.

YOM's governance model harnesses the collective wisdom of its community, elevates committed contributors through XP-based weighting, and grants creators and users the autonomy to shape their own experiences. With a measured transition toward full decentralization, YOM ensures that no single entity holds disproportionate power, sustaining a balanced, community-led ecosystem for the long term.

## DAO Limitations

Within the framework of DAO-approved standards, studios and content creators maintain full autonomy. Only content that contravenes essential DAO policies, such as those related to user safety, or malicious behavior faces moderation or revocation of streaming privileges.

[Previous](#)[Tokenomics](#)[Next](#)[Overview](#)

Last updated      days ago

Was this helpful?

---

## Overview | YOM Docs

**Source:** <https://docs.yom.net/build/overview>

This guide introduces best practices for integrating the YOM SDK with your game, enabling seamless connection between your stream and web technologies.

Developing for YOM differs from traditional projects. Each stream exists on a webpage, instantly accessible from any device. This approach necessitates avoiding loading screens and complex UI in favor of instant, holistic, integral game design. This new methodology requires considering:

- Designing with "responsiveness" and "instant gameplay" in mind
- Debugging with web servers rather than just in UE      viewports
- Bi-directional game-to-web communication
- Web Widgets and Analytics

## Why Build on YOM?

### Rapid Game Deployment

- Both multiplayer and singleplayer.
- Focus on core AAA+ experiences, not menus

## Cross-Platform Reach

- Target large audiences via device-independent compatibility
- Easy embedding in websites and platforms
- Affordable & scalable DePIN hosting

## Integrate Everything

- Seamlessly integrate external data sources & events
- Native tag manager integration for analytics

## Customization / White-label

- White-label user journeys from login to stream
- Integrate custom web widget support for enhanced interactivity
- Manage admin accounts to trigger in-game events

## Integration

The integration SDK includes a simple Node.js web server referencing a sample HTML + JS implementation. When launching a stream via the YOM Toolbar in UE , you can optionally start your preferred web server, allowing local debugging of the entire user journey.

We provide a JS library that enables integration and full customization of the stream experience, from authentication to deployment and execution. We provide functionalities like callbacks and multi-stream capabilities for effective control over stream and authentication processes.

Figure : illustrates our reference HTML implementation demonstrating authentication (included with SDK)

## Stream Embed

Post-authentication, the embedded stream loads as an iFrame, spawning the player and [hiding certain components](#). The stream automatically scales across aspect ratios while maintaining resolution quality. Test this in the reference implementation by resizing horizontally (desktop) or vertically (mobile).

Figure : showcases a sample web widget and the stream with an expanded modal.

To accommodate limited user attention, especially for web-based streams, we utilize a simple, customizable modal that loads post-launching/spawning and its visibility can

be managed via the SDK. This allows focus on game development while minimizing overhead and time spent on UI.

The modal includes a "God-mode" option, accessible via a password in the login field. This feature enables real-time in-game event execution without building web components, facilitating testing and script execution. God-mode benefits both developers and moderators, enabling efficient game management and testing.

Figure : demonstrates the "God-mode" interface, displaying custom functions registered via the SDK.

[Previous](#)[DAO](#)[Next](#)[Getting Started](#)

Last updated      days ago

Was this helpful?

---

## Getting Started | YOM Docs

**Source:** <https://docs.yom.net/build/getting-started>

The GETTING STARTED section provides a comprehensive guide on how to properly set up and deliver your project to ensure compatibility with the YOM build pipeline.

To create your first experience, you need to broadly follow these steps:

- . Create a new or open an existing project.
- . Import the [UE](#) [SDK](#) in your project.
- . Test and debug your experience locally until it runs perfectly.
- . Build and then publish your experience via the [app](#).
- . Integrate your experience on your website with a single line of code.

## A Checklist Before Starting (!):

- If you are opting to use YOM within an existing Unreal Engine project, please check the **Linux support for existing plugins**. If your plugin does not officially support compile to Linux, in most cases it still compiles to Linux and if not on the first try, the fix is usually very simple.
- For multiplayer support you need to download and use the [source version](#) of Unreal Engine.
- You need to install [visual studio](#) to build the headers for the SDK and you need the correct [clang](#) for Linux in order to build the game locally. Install the cross-compile toolchain that you can find on the bottom of the page.

- Ensure you use an NVIDIA RTX or higher for your development machine. This ensures you optimize your game for our network and get the best graphics and experience out of your game.

[Previous](#)[Overview](#)[Next](#)[Compile your Game](#)

Last updated      year ago

Was this helpful?

---

## Compile your Game | YOM Docs

**Source:** <https://docs.yom.net/build/getting-started/compile-your-game>

### Step : Prepare your project

- Ensure your project is in C++ and uses the correct Unreal Engine source version.
- Verify you have the right [Clang](#) version for cross-platform compiling.

### Step : Package the project

- Navigate to the top bar: Platforms → Linux.
- Set the Build Target to Game or Server
- Click "Package Project" at the top.

When selecting a folder to build your project we recommend creating a new folder in your project folder called "Build". Choose this "Build" folder as the destination for both client and server builds.

### Step : Generate builds

- Wait for the build process to complete. Speed may vary depending on the size of your project.
- Verify that two folders, "Linux" and "Linux Server", are created inside the "Build" folder.

### Step : Package for YOM

- In Unreal Engine, go to the top bar: Tools → YOM → Package Project.
- In the opened menu, select the paths to your client and server build folders.

- If you used the recommended "Build" folder, paths should be automatically selected.
- Click "Package Project" to create separate zip files for the client and server.
- After the process completes, find a new folder called "BuildData" in your project directory.

## Result

Inside "BuildData", you'll find two zip files, one for the client and one for the server, that are ready to upload to [app.yom.ooo](https://app.yom.ooo).

[Previous](#)[Getting Started](#)[Next](#)[Deploy your Stream](#)

Last updated      months ago

Was this helpful?

---

## Deploy your Stream | YOM Docs

**Source:** <https://docs.yom.net/build/getting-started/deploy-your-stream>

Fantastic work so far! Now it's time to publish your space on [app.yom.ooo](https://app.yom.ooo).

## Getting started

- Create an account by creating a phantom wallet. *Note: we use this as the login method because you will receive a token kickback for every game that runs on our network.*

## Step      : Create Stream

- Log in with your wallet in [app.yom.ooo](https://app.yom.ooo) and create a new stream:
- **Name your stream:** enter a unique name (e.g., firststream)
- **Select Build Target:** choose your target environment (e.g., Development)
- **Upload Packaged Project:** drag and drop or click to upload your server and client zip files.
- **Choose Deployment Network:** select your preferred network (e.g., AWS)
- **Select Deployment Region:** pick an AWS region for your stream deployment



## Step : Deployment

- . Check your configuration and click on pay + deploy.
- . Wait until both server and client files are uploaded to the designated zone. Do not refresh or leave the app during the upload process.
- . The stream will begin deploying automatically after upload. You may safely close the app once deployment starts.

## Step : Verifiy stream URL

- Use your uniquely provided stream URL to integrate your stream into your desired platform.
- Refer to the [next section](#) for detailed instructions on stream integration on any destination webpage.

[PreviousCompile your GameNextIntegrate your Stream](#)

Last updated      year ago

Was this helpful?

---

## Integrate your Stream | YOM Docs

**Source:** <https://docs.yom.net/build/getting-started/integrate-your-stream>

Great that you have come this far.

Congratulations on deploying your stream. Now, it's time to share it with your fans.

Follow the steps below to integrate your metaverse into your website.

### Quick Start

- Add the YOM Core script to your HTML: Insert this line inside the `<head>` tag of your HTML file:

Copy

```
<script src="https://temp-yom-branding.s3.eu-west-1.amazonaws.com/yom-core.js"></script>
```

- Create a target div for your stream: Place this div within the desired location in your webpage and replace `data-streamurl` with your personalized stream url:

Copy

```
<div id="yomTarget-1" data-streamurl="${STREAM_URL}"></div>
```

- Add User Input and Launch Button somewhere on your page:

Copy

```
<input type="text" id="yomInput-1" placeholder="Your Name">
<button id="yomLaunch-1">Start Experience</button>
```

That's it! You now have a basic YOM integration. Launch your page to see it in action.

## Troubleshooting

If you encounter issues:

- . Check your browser console for error messages.
- . Ensure all required scripts are loaded correctly.
- . Verify that stream URLs are correct and accessible.

For further assistance, contact our team at [\[email protected\]](#).

[PreviousDeploy your StreamNextCustom Authentication](#)

Last updated      year ago

Was this helpful?

---

## Custom Authentication | YOM Docs

**Source:** <https://docs.yom.net/build/getting-started/integrate-your-stream/custom-authentication>

You can define your own custom functions to validate user input on button click. To achieve this add `data-callback="{someFunction}"` to the launch button. Then define your custom function in a JS file that accepts two parameters: `onSuccess`, `onFailure`.

## Example

We have as an example the following HTML elements:

Copy

```
<input type="text" id="yomInput" placeholder="your name" autofocus> <input type="password" id="password" placeholder="Enter password"> <button type="submit" data-callback="validatePassword" id="yomLaunch-1">start the experience</button>
```

In the following example function we perform simple front-end validation to match the data- callback="validatePassword" attribute defined in the yomLaunch-1 button:

Copy

```
`` function validatePassword(onSuccess, onFailure) { const passwordInput = document.getElementById('password'); const enteredPassword = passwordInput.value; const hardcodedPassword = 'secure'; // This is the hardcoded password

  if (enteredPassword === hardcodedPassword) {
    onSuccess(); // Call the success callback if the password is correct
  } else {
    onFailure(); // Call the failure callback if the password is incorrect
  }
}
```

[PreviousIntegrate your StreamNextEvent Handling](#)

Last updated      year ago

Was this helpful?

---

## Event Handling | YOM Docs

**Source:** <https://docs.yom.net/build/getting-started/integrate-your-stream/event-handling>

With YOM you can build custom web widgets that communicate directly with the stream via the `postToYOM` function:

Copy

```
postToYOM('AdjustSetting', {settingValue: 'newValue'});
```

This function facilitates sending commands or adjustments to the stream. Do ensure you [register events](#) inside of your game project, to bind the events to certain functions.

## Example

We assume you have buttons on your webpage ( `#spawnBot` , `#removeBot` , `#clearBot` ). On click we can trigger content within the stream using the `postToYOM` function:

Copy

```
`` document.addEventListener('DOMContentLoaded', function() {
document.getElementById('spawnBot').addEventListener('click', function()
{ postToYOM('SpawnBot', {amount: '  '}); });

document.getElementById('removeBot').addEventListener('click',
function() {
    postToYOM('RemoveBot', {amount: '1'});
});

document.getElementById('clearBot').addEventListener('click',
function() {
    postToYOM('ClearBot', {});
});

}); ``
```

## Listen for Events

Using YOM SDK you can push events from the game to the web browser. From there you can connect your own logic to it.

YOM SDK can [automatically push](#) events to the data layer for GTM and GA integration. Configure your GTM to capture these events for detailed interaction tracking.

[PreviousCustom AuthenticationNextMulti-Stream Setup](#)

Last updated      year ago

Was this helpful?

## Multi-Stream Setup | YOM Docs

**Source:** <https://docs.yom.net/build/getting-started/integrate-your-stream/multi-stream-setup>

YOM Core supports multiple streams on a single page, enabling rich, interactive webpages that can act as content aggregators.

- Create multiple stream containers:

Copy

```
<div id="yomTarget-1" data-streamurl="${STREAM_URL_1}"></div>
<div id="yomTarget-2" data-streamurl="${STREAM_URL_2}"></div>
```

- Add corresponding input fields and launch buttons:

Copy

...

Start Stream

Start Stream

...

- Build widgets for independent streams:

Copy

```
const targetId = '2'; // assuming we want to post to stream 2
postToYOM(targetId, 'AdjustSetting', {settingValue:
'newValue'});
```

[PreviousEvent HandlingNextDynamic Layouts](#)

Last updated      year ago

Was this helpful?

---

## Dynamic Layouts | YOM Docs

**Source:** <https://docs.yom.net/build/getting-started/integrate-your-stream/dynamic-layouts>

To dynamically toggle element visibility in response to stream interaction, you can directly utilize the `.toggle-on-yom-{targetId}` :

Copy

```
<div class="additional-info toggle-on-yom-1">      Information  
that should show after a user joins the stream. </div>
```

If you want to apply custom css, you can use the `data-hook` to associate specific UI elements with the stream, allowing for broader layout adjustments when the stream is activated or deactivated:

Copy

```
<div id="yomTarget-1" data-hook="#{somediv}" data-streamurl="$  
{STREAM_URL}"></div>
```

[Previous](#)[Multi-Stream Setup](#)[Next](#)[UE](#)

Last updated      year ago

Was this helpful?

---

## UE | YOM Docs

**Source:** <https://docs.yom.net/build/ue>

### Step : Use a C++ UE Project

You need to use a C++ version of an UE project. This can be achieved by setting up a new project and selecting C++ as the project defaults or by converting your existing blueprint project to C++. There is no disadvantage for picking C++ over Blueprints as you can still use Blueprints throughout your project.

If you are converting from an existing project follow the following steps to convert it to a C++ project:

- Click Tools -> select "new C++ class"

- Select none as the parent class and press next.
- Press create and it will convert your Blueprint project to a C++ project.

## Step : Add the SDK to the project

To incorporate the necessary SDK into your project, follow these steps:

- Close your project if you opened it as part of the previous step.
- Navigate to your root project folder and create a new folder called "Plugins". This folder will serve as the container for the required plugins.
- Obtain the YOM UE SDK for your Unreal Engine version from the [link](#). If you don't have access, ask your YOM contact person. This will trigger a download of a ZIP file. The ZIP file will contain three plugins.
- Extract the zip file and copy and paste the three plugins into the "Plugins" folder you need to create in your project directory. Ensure that the plugins are directly placed inside the "Plugins" folder and not within any additional subfolders.

## Result

By following the steps outlined above, you should now have a folder structure resembling the following:

Copy

```
- YourProject - Plugins - YOM - GltfRuntime -  
VaRest
```

## Step : Building the SDK

Now that everything is set, boot your project and compile the plugin headers. This need to be done once every time you update the SDK or add any other plugin and requires using a C++ project with [Visual Studio](#) installed. After it is set, it will automatically open the project.

Opening the project for the first time can take a bit of time. Just be patient. It will boot.

[PreviousDynamic LayoutsNextConfigure your Stream](#)

Last updated      days ago

Was this helpful?

---

# Configure your Stream | YOM Docs

**Source:** <https://docs.yom.net/build/ue4/configure-your-stream>

This page will explain how to set up your game to work with the features present in the YOM UE4 SDK and assumes a new project. For existing projects, you may already have modified the GameMode and PlayerController classes. To retain compatibility follow the following guide.

## Step 1 : Validate the SDK

YOM SDK should be enabled by default. Validate by checking if a new toolbar appeared in your navigation bar called 'YOM Toolbar'.

### Follow these steps if the toolbar did not appear:

- Click on "Edit" and select "Plugins" from the dropdown menu. This will open the Plugins window.
- In the search bar within the Plugins window, type 'YOM' to search for the YOM Replicator SDK plugin.
- Enable it. If the plugin does not appear, try restarting the editor and repeat this step.
- Do the same for GltfRuntime and VaRest plugins in the same manner and enable them by clicking their respective checkboxes.
- After enabling the plugins, it is recommended to restart the editor. This allows Unreal to load the plugins properly.

## Step 2 : Setting up the SDK

To configure the YOM Replicator SDK for your metaverse project, follow these steps:

- Open the setup screen by navigating to "Tools" in the top menu. Select "YOM" and then choose "Setup Project." This will open the setup window.
- In the setup window, you will find several settings that need to be configured. Here is an explanation of each setting:
  - **Startup World:** This setting sets the startup level. When we make a build this will be the starting point. As example it can be a mainmenu or if you don't have that the scene the game begins
  - **Game World:** This setting sets the game level. This is the level where the game takes place.
  - **Player Asset:** This setting defines the asset that players will control within the world. It can be any character or object that allows player interaction. The default



character provided by the YOM plugin is the Yom\_BP\_ThirdPersonCharacter. Change it from "None" to "Yom\_BP\_ThirdPersonCharacter".

- **Playercontroller asset:** The Playercontroller that is used the game world or you want to use in the game world.
- **GameMode asset:** The GameMode that is used the game world or you want to use in the game world.
- **Webserver path:** The path to the webserver you want to use. We already profite a example webserver.
- **Multiplayer project:** If enabled, this setting allows the code from the SDK to be multiplayer. Enable this if you creating a game for multiplayer
- **Voice Chat:** If enabled, this setting allows players to communicate with each other through voice chat within the experience.

Click "Begin Setup," and if the settings are correctly configured, a message will appear in the bottom right corner, instructing you to restart the editor. Click "Restart" to apply the changes.

In case you want to make any further adjustments or verify the setup, you can run the setup tool again or manually modify the settings.

Validate your configuration

After restarting, navigate to Content Drawer -> Content. You should now see a MetaspacGameMode and a MetaspacPlayer. These assets can be used or extended to add further modifications and behavior to your experience. Then navigate to Settings -> World Settings, then validate the GameMode as shown in the provided screenshot.

*Note that the Default Pawn Class in the MetaspacGameMode is intentionally set to None, as the pawn will be loaded into the game via the MetaspacPlayer.*

[PreviousUE](#) [NextTest your Stream](#)

Last updated      months ago

Was this helpful?

---

## Test your Stream | YOM Docs

**Source:** <https://docs.yom.net/build/ue> /test-your-stream

To test your game locally you can do this in three ways:

- **Stream to localhost:** . This allows you to debug the full web experience, including player authentication. This is the default setting with all toggled turned off and is also the recommended setting to playtest the full experience.
- **Play it in viewport.** If you want to quickly debug locally and you want to bypass the authentication window, you can opt for testing in the viewport. To enable this you need to toggle on 'Run in Editor' and 'Auto Spawn Player'.
- **Play in standalone.** If you want to debug networking features without needing to first pass the authentication window in the web browser, you need to toggle on 'Auto Spawn Player'.

## Toolbar

For these things we made the YOM Toolbar to easily test your game. We added some settings to help you test your game:

- **Launch Game.** Launches the game with the below settings.
- **Run in Editor.** When turned off it runs the game in stand alone and starts the signaling server. When turned on it will run the game in viewport and not start the signaling server.
- **Respawn Player.** When turned off it runs the game in stand alone and starts the signaling server. When turned on it will run the game in viewport and not start the signaling server.
- **Auto Connect Player:** When turned on it will execute the connect logic as you would connect from the pixelstreamer
- **Run Webserver:** When turned on it will start the webserver when we press launch game
- **Run Webclient:** When turned on it will start the webclient when we press launch game
- **Force Authentication.** When true, it forces users to authenticate with their wallet else it will not work.
- **Number of Clients.** You can select the number of clients it needs to boot up. It will also boot up another signalling server. You can connect to it by adding a port.  
is the first one second etc.

The YOM Toolbar it to make it easy to test locally. You can still use the default buttons to play the game and the settings will apply to it.

## Result

Pressing the Launch Stream button, the servers and games will boot and you can debug your game within a webpage. Make sure to add a Player Start to the level if the character does not spawn automatically.

[Previous](#)[Configure your Stream](#)[Next](#)[Responsive Design](#)

Last updated      months ago

Was this helpful?

---

## Responsive Design | YOM Docs

**Source:** <https://docs.yom.net/build/ue4/responsive-design>

In the era of multi-device gaming, it's crucial to ensure your game is accessible and optimized for various platforms, including mobile devices and tablets. While YOM uses pixel streaming to deliver high-quality graphics across devices, understanding and adapting to different screen sizes and input methods can greatly enhance the user experience. Why implement responsive design:

- . Broaden your audience by supporting multiple device types
- . Ensure consistent gameplay experience across platforms
- . Optimize UI and controls for touch-based interfaces
- . Improve accessibility for players with different device preferences

### Step 1 : Detect Mobile Devices

YOM provides a built-in method to identify if a player is connecting from a mobile device:

- . Access the `YomGameInstanceSubsystem` in your Blueprint or C++ code.
- . Check the "GetIsMobile" boolean.

### Step 2 : Adjust for Mobile

Once you've detected a mobile device, you can make appropriate adjustments :

- Increase button sizes for touch input
- Implement different widget blueprints for mobile and desktop
- Utilize Size Boxes and Scale Boxes for flexible layouts
- Use anchors and alignment for position-independent layouts

- Simplify certain menus for smaller screens
- Implement touch-friendly controls

## Step : Test with Chrome's Device Mode

To debug and test your mobile layout:

- . Open your game in Chrome
- . Press F to open Developer Tools
- . Click the "Toggle device toolbar" icon or press Ctrl+Shift+M
- . Select different device presets or set custom dimensions
- . Test your game's responsiveness in real-time

## Result:

By implementing these responsive design techniques, your YOM-powered game will provide a smooth, adaptable experience across a wide range of devices, from high-end desktop computers to smartphones and tablets.

*Note: Always test your responsive design on actual devices when possible, as emulators may not perfectly replicate real-world conditions.*

[Previous](#)[Test your Stream](#)[Next](#)[Registering Events](#)

Last updated      months ago

Was this helpful?

## Registering Events | YOM Docs

**Source:** <https://docs.yom.net/build/ue /registering-events>

The YOM SDK allows you to create and listen for your own custom events, allowing communication between the Unreal client and the pixel streamer /web browser. This guide explains how to set up and use them.

## Step : Register a Stream Event

You get register an stream event by getting the YomGameInstanceSubsystem:

- . Set Event Name:
  - Name of the function to call

- Will be displayed in the "GodMode" modal when enabled
- Used to call the linked event
- . Set Event:
  - Event to be called when the function named FunctionName is triggered
- . Set NeedsGodMode:
  - Determines if the event is added to GodMode
  - Use for admin-only functions (GodMode requires a password)
- . Create the event:
  - Drag from Event and search for "create event"
  - Select "Create Event" node
  - Choose "select a function" and pick from the dropdown or create a new one
- . Set stream event data
  - When you have variables that need to be passed in your function you can already define what names they have so it will show up in the godmode.
  - You need to create a array and add the amount of variables you have to that.
  - Param Name: name of the param variable.
  - Default Value: the value it needs to have or has at the start.

*Note: The function input will be a Va Rest Json object, allowing data to be sent with the event.*

## Step : Handling Event Callback

To process data that is sent with the event:

- . Access the data object passed to the function
- . Use "get field by name and type" to retrieve specific variables
- . Implement your logic using the retrieved data
- . Note that everything we send will be a string and you need to convert that to the type you need.

[PreviousResponsive DesignNextPush Events](#)

Last updated      months ago

Was this helpful?

---

## Push Events | YOM Docs

**Source:** <https://docs.yom.net/build/ue /push-events>

The YOM SDK allows you to push your own custom events to the browser, allowing communication between the Unreal client and the pixel streamer /web browser. This guide explains how to set up, determine their event scope, and use them.

### Step : Use PushYomEvent Function

Once you have the YOMGameInstanceSubsystem, use the PushYomEvent function:

Copy

```
YOMGameInstanceSubsystem->PushYomEvent(PEventScope, PFunction, PVariableNames, PVariableValues);
```

Use the "PushYomEvent" function with these parameters:

- Function: Name of the function to call
- Variable Names: Array of variable names
- Variable Values: Array of variable values
- DataLayer: Boolean to determine whether the event should be [automatically pushed forwards to the DataLayer](#), used by marketing and data services Google Analytics.

### Result:

By following these steps, you can push communication from your Unreal Engine project to the pixelstreamer/web browser.

[PreviousRegistering EventsNextAirdrops](#)

Last updated      months ago

Was this helpful?

---

## Airdrops | YOM Docs

**Source:** <https://docs.yom.net/build/ue /airdrops>

It is possible to incentivize and reward users in experiences by dropping items directly to their wallet as part of the experience mechanics. This should be done on the server side, and the player must have a connected wallet for it to work. Make sure the inventory is set up as well. In the airdrop node, you need to set two values:

- . YomPlayer: Specify the player to whom the item will be airdropped.
- . Item ID: Provide the ID of the item that will be airdropped. There are separate IDs for devnet and mainnet.

The airdrop node will add an item based on the item ID to the player's wallet. Please note that the airdrop function will not work for users who haven't logged in with a wallet.

## Checking if airdrop received

If you want to check if a user has already received an airdrop, you can follow these steps:

- . Retrieve the YomPlayer of the player.
- . Get the inventory of the YomPlayer.
- . Check if there is an item in the inventory with the ID of your master NFT ID. If an item with that ID exists, it means the user has already received the airdrop.

Blueprint to check if airdrop was received

## Checking wallet connection

To check if a user has a wallet connected, follow these steps:

- . Get the player you want to check.
- . Retrieve the YomPlayer from that player.
- . Get the wallet of the YomPlayer.
- . Check if the wallet ID is empty. If it's not empty, it means the user has a wallet connected.

Blueprint to check wallet connection

## Checking if wallet is fetched

Fetching the wallet may take some time, so there is a callback function called "OnInventoryFetched" that you can use. To check if the wallet has been fetched, do the following:

- . Get the YomPlayer from the player.
- . Check if the YomPlayer has a valid inventory.
- . Bind an event to the "OnInventoryFetched" callback to be notified when the inventory is fetched.

Blueprint to check if wallet is fetched

Note: Airdrops last until the maximum supply of the item has been reached.

[PreviousPush EventsNextGated Access](#)

Last updated      year ago

Was this helpful?

---

## Gated Access | YOM Docs

**Source:** <https://docs.yom.net/build/ue4/gated-access>

To create exclusivity and restrict access to experiences based on specific item ownership in YOM, you can follow these steps:

### Step 1 : Adding the Gated Access component:

- By default, the Gated Access component (YomGatedAccess) should already be connected to your Metaspaces Player. If you don't see it as a component, you'll need to add it manually.
- To add the component, click on the "Add Component" button in the blueprint editor and search for "Yom Gated Access." Add it to your Metaspaces Player blueprint.

### Step 2 : Adding the Gated Access actor to the level:

- To implement the Gated Access functionality, search for "Yom Gated Access" in the Place Actors window and drag it into the level where you want to create the restricted zone.



## Step : Setting the Gated Access Item ID:

- Click on the Gated Access actor in the level.
- Look for the "Item ID" property and insert the NFT address or master ID of the NFT item that is required for access. Note that you have separate IDs for devnet and mainnet.

Setting the Item ID for gated access

## Step : Setting up a Collision Layer:

- Set up a collision object channel for the Gated Access to work properly.
- Go to Project Settings and navigate to the Collision section.
- Click on the "New Object Channel" button.
- Name the channel as "GatedAccess" and set the default response to "Overlap."

Example gated access volume, only accessible when possessing a certain Item

Notes:

- You can check if a player has access to the Gated Access zone by calling the blueprint function "Can Player Access."
- Make sure the inventory is enabled for the player for this functionality to work.
- The user's wallet needs to be fetched before they can go through the Gated Access.

[Previous](#)[Airdrops](#)[Next](#)[Characters](#)

Last updated      year ago

Was this helpful?

---

## Characters | YOM Docs

**Source:** <https://docs.yom.net/build/ue4/characters>

This guide explains how to create and customize a player character for your YOM stream. It covers using YOM's pre-built character or creating your own, understanding existing functionality, and how to modify or extend character features.

## **Step : Create Your Player Character**

- . Create a New Blueprint:
  - Right-click in the Content Browser
  - Select "Blueprint Class"
  - Search for "Yom\_BP\_ThirdPersonCharacter" to use YOM's existing character framework
- . Assign the Player Character:
  - After creating your blueprint, assign it as the player character for your project

## **Step : Understand Yom\_BP\_ThirdPersonCharacter**

Yom\_BP\_ThirdPersonCharacter includes:

- . Movement controls (WASD keys)
- . Interaction system (E key, requires BP\_Interact interface)
- . Camera movement (third-person and first-person)
- . Camera switching (Q key)
- . Crouching (C key)
- . Jumping (Spacebar)
- . Walking/Running toggle (Left Shift)
- . Touch support for mobile devices
- . Customizable settings (walk/run speed, starting view)

## **Step : Modify Functionality**

To customize Yom\_BP\_ThirdPersonCharacter:

- . Override the function you want to change
- . To add functionality:
  - Call the parent function
  - Add your custom logic
- . To remove functionality:
  - Override the function
  - Leave it empty
  - Do not call the parent function

*Note: Overriding **preserves** changes during SDK updates*

## Step : Manage Controls

Default controls use Unreal's Enhanced Input System and are mapped to a controls customization modal that end-users can alter from within the stream:

. To modify:

- Create your own "InputMappingContext" in the content folder
- Add required defaults
- Create new key mappings for your project

*Caution: Direct changes to default controls may be overwritten by SDK updates*

## Result:

Following these steps allows you to create, customize, and manage your player character while leveraging YOM's existing functionality and maintaining compatibility with future updates.

[PreviousGated AccessNextControl Hints](#)

Last updated      year ago

Was this helpful?

---

## Control Hints | YOM Docs

**Source:** <https://docs.yom.net/build/ue /control-hints>

## Displaying the Control Hints Panel to the player's screen

The YOM SDK allows creators to display a UI widget that would allow the player to understand the basic game controls. This guide will explain how to display and customize your own Control Hints with our ecosystem.

## Step : Adding Control Hints component

To use the features of the Control Hints component, it needs to be connected to your player. This should be done by default. However, if you do not see YomControlHints

under your player as a component, follow this step. Otherwise, you can skip to step

YomControlHints is a component therefore we need to add it to the player blueprint. To add the component you will have to click add on component menu and search for Yom Control Hints .

Pick YomControlHints Component

After this, you need to assign a widget to the YomControlHints component. By default an existing widget should be already available, otherwise you would have to create a one from scratch (step ).

Pick YomControlHints Widget

## Example of ControlHints

YomControlHints Example

[PreviousCharactersNextNameplates](#)

Last updated      year ago

Was this helpful?

---

## Nameplates | YOM Docs

**Source:** <https://docs.yom.net/build/ue4/nameplates>

## Adding a nameplate to your character

The YOM SDK allows creators to design their own nameplates. This guide will explain how to create and use your own nameplates with our ecosystem. If you just want to use our default nameplates, but it has not been enabled yet on your character you can move to Step      and use the YomDefaultNameplateWidget as your widget.

### Step      : Creating a custom nameplate

To create a custom nameplate you will have to create a new blueprint asset. Right-click in a folder in the Content Browser and under Create Basic Asset click on Blueprint Class .

Blueprint Class

This will open up the blueprint dialog box in which you can search for `YomNameplateWidget`. Once you find the `YomNameplateWidget` select it.

Pick `YomNameplate`

You can now customize your nameplate as you like, but one of the widgets must be a `TextBlock` widget with the name `WalletID`. In this `TextBlock` the `walletID` of the players will be printed when they are loaded into the game. Also give this `walletID` block a nice default name like `Unknown` for when the nameplate could not be loaded.

## **Step : Adding the nameplate to your character**

Open up your character (this is not your `MetaspacePlayer`, but the character that the players will use to move around with) and under the `Components` tab add a `Widget` component.

Pick `Widget` Component

Give this widget a nice name like `Nameplate` and select it. Under the `Details -> User Interface -> Widget Class` you can set your widget.

Select `Widget Class`

Tip: If you want the widget to always face the player, change `Space` from `World` to `Screen`.

In the `Viewport` tab, set the nameplate to the desired position for your character.

## **Step : Add the YomNameplateManager to your MetaspacePlayer**

The nameplates will only work when you tell your `MetaspacePlayer` to load the `YomNameplateManager` in your game on startup. Go to your `MetaspacePlayer` and under `Details -> Yom Player -> Managers` add an element and insert `YomNameplateManager`.

Add `YomNameplate Manager`

You should now be able to see a nameplate when you spawn into the game.

[PreviousControl HintsNextCounters](#)

Last updated      year ago

Was this helpful?

---

# Counters | YOM Docs

**Source:** <https://docs.yom.net/build/ue4/counters>

## Adding a counter to your metaverse

The YOM SDK allows creators to design their own counters. This guide will explain how to create and use your own counters with our ecosystem.

### **Step 1 : Adding counter actor to the world**

To use the features of the counter, it needs to be added to the world first.

YOM Counter is an actor and needs to be manually added to the world to start working. To add the actor, you will have to slide the actor from `Content Browser` under `Plugins -> YOM Content -> Counter` into your world.

After this, you can call counter functions through blueprints without errors.

### **Step 2 : Adding counter component to the player**

To use the auto replicating feature of the counter (updating a counter value is shared with other clients), you will need to add the counter component to the player blueprint. You can do this by going to your `MetaspacePlayer` and using the add button on the top left corner of the screen.

Pick `YomCounter` Component

After you do this, you can define what to do when a counter value is updated. This is done by using a callback event. This callback event gives you the name of the counter which is updated and its new value.

`Yom Counter Updated`

When a counter value is updated with `GetCounterPlayer` blueprint function, this callback function will be called on all of the clients with a counter component. Returned value and the name can be used however you like.

### **Step 3 : Using blueprint functions to use the component**

From the blueprint actions box (it is automatically open or you should right click to the screen), search for `Yom Counter` in order to see the available functions.

Example use case for `Get Counter` function

## Get Counter

In this example, we are trying to get `testCounter` in the current metaspace. Counter value is returned by a callback therefore we drag and drop P Callback node and create a custom event. P value returned by the event is the value of the counter which can be used however the user wants. In this example case it is printed to development screen.

## Create Callback Event

Example use case for `Add to Counter` function

## Add to Counter

In this example, counter is set to increment the `testCounter` in the current metaspace by value `1`. Therefore, this will increment that counter value by `1`. P Callback function is the function which is returned after API Request is done. It can be used to execute more commands when your increment/decrement request is done.

If callback function is not required, a dummy callback function can be created as it can be seen from the image.

Finally, we can use `Get Counter Player` blueprint function in order to replicate the get request of the counter value. When it is called from a client, it makes a server call which is replicated to other clients as well. This way, all of the connected clients are getting the new value of the counter as well as the name of the counter.

Example use case for `Get Counter Player` function

## Get Counter Player

Note: we recommend not updating the counter too much, but 'batch' the counts, as everytime the counter is updated it needs to be updated on the blockchain, which costs money.

[PreviousNameplatesNextInventory](#)

Last updated      year ago

Was this helpful?

---

## Inventory | YOM Docs

**Source:** <https://docs.yom.net/build/ue/inventory>

# Setting up an inventory

The YOM plugin allows creators to completely design their own inventory system. This guide will explain how to create your own inventory system and how to use your own inventory system in your metaspace.

## Step : Creating your own buttons

The inventory system consists of buttons that can be pressed to spawn items. A single button needs to be designed for the system to work. To create a button you will have to create a new blueprint asset. Right-click in a folder in the Content Browser and under Create Basic Asset click on Blueprint Class .

### Blueprint Class

This will open up the blueprint dialog box in which you can search for YomButton . Once you find the YomButton select it.

### Pick YomButton

Name the button something like MyButton and open up the blueprint by double-clicking the new asset. In the blueprint editor, you will now have an empty screen, in which you will have to design your button. But before you start the button will need some widgets:

- Canvas ( Overlay )
  - The canvas in which the button can be built
- Button ( Button )
  - The button that can be clicked
- RarityWidget (Any Widget)
  - The rarity widget, can be seen as the border around the button
- LoadingWidget (Any Widget)
  - The widget that is shown when the item is being loaded

All of these widgets also need to have the name of the items described above. In the default YomButton the hierarchy looks like this:

### YomButton Hierarchy



## Tips

- Set the visibility of the RarityWidget to Not Hit-Testable (Self & All Children) by clicking on the RarityWidget in the hierarchy and going to Details -> Behavior -> Visibility . This prevents the Rarity widget from 'overlapping' the button which would prevent a user from pressing the button.
- Design your button with a fixed size this will prevent the buttons from 'stretching' when loaded

## Step : Creating your own Inventory Widget

Create another blueprint like in step , but this should have YomPlayerInventoryWidget selected.

Pick YomPlayerInventory

Give this a name as well like MyPlayerInventoryWidget and open up the blueprint by double clicking the new asset. This will again give you an empty screen. On this screen, you can design your layout of the inventory. It will need one widget:

- BaseButtonWidget (A widget with slots for other widgets like the Scroll Box )
  - In this widget, your buttons will be created

The hierarchy of the default InventoryWidget looks like this:

InventoryWidget Hierarchy

Furthermore, your MyPlayerInventoryWidget needs to have some settings applied to it before it works. Click on MyPlayerInventoryWidget in the hierarchy and go to Details -> Yom Inventory . Here you will see the following settings:

- Yom Button Widget
  - The widget for your button, set this to the button you created in step .
- Base Button Widget
  - This one should be set to the BaseButtonWidget in the hierarchy, but if it is not set, you can set it here

- Start Button Amount
  - This value tells the inventory system how many buttons it should spawn when the player is connected. If the player has more items than this amount more buttons will be loaded.

### Tips

- Keep in mind that players could have more items in their inventory than you expect. Your inventory should not deform when a lot of items are being loaded.

## Step : Creating a component

Create another blueprint like in step , but this should have `YomInventoryComponent` selected.

Pick `YomInventoryComponent`

Give this a name as well like `MyInventory` and open up the blueprint by double clicking the new asset. In the asset, you will have to set one setting. Go to `Details` -> `Yom Inventory Component` -> `Inventory Widget` and set it to the widget you created in Step

## Step : Setting your inventory in your player

Open your `MetaspacePlayer` and under components remove `YomDefaultInventory` if present. Then add the component that you created in step . When you start the game now you can see your own inventory in action when pressing 'I' while playing.

[PreviousCountersNextPortals](#)

Last updated      year ago

Was this helpful?

---

## Portals | YOM Docs

**Source:** <https://docs.yom.net/build/ue /portals>

# Adding a portal to your Stream

The YOM SDK allows creators to design their own portals. This guide will explain how to create and use your own portals with our ecosystem. If you just want to use our default portals, you can use the portals in the `Content Browser` under `Plugins -> YOM Content -> Portals` and skip to step . The portals that you can use are the `YomDefaultPortal` and the `YomFXPortal`, where the `YomFXPortal` is more visually appealing. Note that you must have the option `Allow teleporting` set to true, you can verify this by going into your `MetaspacePlayer` and checking if it has a `YomPortallingComponent`.

## Step : Creating a Custom portal

To create a custom portal you will have to create a new blueprint asset. Right-click in a folder in the `Content Browser` and under `Create Basic Asset` click on `Blueprint Class`.

Blueprint Class

This will open up the blueprint dialog box in which you can search for `YomPortal`. Once you find the `YomPortal` class select it.

Pick YomPortal

You can now start to customize the portal in every way you like. If you want the plugin to automatically set the material you must set `Details -> Auto Set Portal Material` to true and the portal must have a `StaticMeshComponent` called `PortalDisplay`, this will automatically edit the material of the `StaticMeshComponent` at runtime when you connect to a different stream. If you want to set the material yourself you can use the `Get Portal Material` node in the `Event On Portal Open` and `Event On Portal Close` events. Furthermore the portal will need a `Collision` component for triggering the portal behaviour and you can set a default material under `Details -> Materials` for when the portal is not connected.

## Step : Adding the portal to your level

You can now add the newly created portal to your level by taking it from the `Content Browser` and sliding it into your level.

## **Step : Adding a button to your level**

In this step we will describe how to create your own button, if you want to use a default button created by YOM please use the `YomDefaultPortalButton` , which works by pressing `E` in front of the button and go to step .

To create your own button we will have to create a blueprint. Right-click in a folder in the `Content Browser` and under `Create Basic Asset` click on `Blueprint Class` .

Blueprint Class

This will open up the blueprint dialog box, in this box click on `Actor` .

Pick Actor

Open up the blueprint and add a collision (we use `Box Collision` ).

Pick Box Collision

Click on the new collision component and under `Details -> Events` click on the `+` sign on the `On Component Begin Overlap` .

Pick `OnComponentBeginOverlap`

This will now open the event graph. Under `Variables` press the `+` sign and add the following two variables:

Setup `PortalButtonVariables`

Note that you will have to set the types of the variables yourself, and that the 'eye' needs to be open.

After this setup your graph like this:

`OpenPortalGraph`

You have now created a button that can open a portal when a player runs into the box.

## **Step : Adding the button to your level**

You can now add the newly created button to your level by taking it from the `Content Browser` and sliding it into your level.

## Step : Opening the portal

In the `Outliner` find the button that you just put into your level. In the `Details` panel of this actor you can now find the `Portal` and `Url` variables that you created. In the portal set the portal that you added to the level in step . The url of the portal should be a valid url to another stream.

## Step : Checking the images that are generated

If you start the game now, trigger the button and walk over to your portal you will see the image of your stream in the portal. Furthermore, a directory called

`PortalImages` is added to your project root in which the images are stored. These are needed when submitting your project to YOM as these will be used to display your stream in other streams. When your level is hosted the image of another stream will be set in here.

[PreviousInventoryNextUnity](#)

Last updated      year ago

Was this helpful?

---

## Unity | YOM Docs

**Source:** <https://docs.yom.net/build/unity>

To be created.

[PreviousPortalsNextOverview](#)

Last updated      days ago

Was this helpful?

---

## Overview | YOM Docs

**Source:** <https://docs.yom.net/earn/overview>

In the YOM ecosystem, **nodes** are the backbone of a distributed cloud gaming platform. Unlike traditional crypto “nodes” that often sit idle or rely on speculative token emissions, YOM nodes perform **real work** – streaming live D game sessions to users – and **get paid based on actual usage**.

This chapter provides a comprehensive look at how YOM nodes function, how rewards are calculated, and how node operators can maximize their earnings. We'll also explore practical examples and best practices, presented in a narrative yet professional tone to guide even first-time participants through the technical details.

## What is a Node and How Does It Work?

A YOM node is essentially a slice of a gaming computer or server dedicated to running a cloud gaming session for a player. When a player launches a game through YOM's platform, the game is actually rendered on a node operator's hardware and streamed to the player's device in real-time. Each node in the network streams a AAA game to a single concurrent player.

In simpler terms, running a YOM node is akin to hosting a small game server that *one player connects to for a session*, except the player only sees the video feed. Because YOM is decentralized, these nodes aren't in a corporate data center – they're run by everyday operators on their own hardware. Every time content is streamed or rendered by your node, you get paid for that session.

Anyone can run a YOM node, provided they obtain a node license and have access to suitable hardware or cloud infrastructure. A node license is essentially an access token (often an NFT or code) that registers you as an official node operator in the network. Once you have a license, you can choose how to deploy your node: either self-hosted on your own PC/rig or delegated to a verified Node-as-a-Service provider who runs the node on your behalf.

## How Many Nodes per Machine?

Notably, one physical machine can run **multiple node instances** (multiple licenses) in parallel if it has enough resources. YOM's software modularizes nodes as containers or processes, each serving a different gamer. For example, a powerful GPU like an NVIDIA RTX can host about **concurrent game sessions** by running node licenses simultaneously on the same rig. In fact, each PC can run up to *nodes at once* with the plug-and-play YOM setup

This means an operator with a high-end system can multiply their earnings by supporting several players at the same time using a single machine (essentially slicing their GPU/CPU to create multiple nodes). YOM's recent node model update was designed to leverage this: by decoupling licenses from hardware, a single expensive GPU can replace what previously required multiple separate rigs. As a result, the network became more scalable and accessible – operators with modest setups can start with one or two licenses, while those with beefier hardware can **scale up** by adding more licenses to fully utilize their machine.

# How Node Rewards Work

**YOM's reward model is session-based and usage-driven**, aligning incentives with real player activity. This is a key differentiator from many crypto networks that simply act as token farms. Instead, YOM nodes earn in proportion to the gaming sessions they host and continuously attract buying demand.

Every game session your node hosts earns you a fixed reward. Each session typically lasts around                minutes and includes a brief advertisement, generating consistent revenue. More sessions translate directly into higher earnings. Let's break down how the rewards are calculated:

## Regional Modifiers

Advertising rates vary globally. YOM accounts for regional differences in ad revenue (CPM) and operating costs to ensure fair compensation:

- **High-Value Regions:** Operators in markets with strong advertising demand (e.g., North America, Western Europe) receive proportionally higher per-session payouts.
- **Cost-of-Living Factors:** For regions with lower ad rates but also reduced electricity or bandwidth costs, the network adjusts rewards to remain competitive and profitable for local operators.

By tuning rewards daily to reflect changing CPMs or electricity expenses, YOM ensures a stable earning environment for operators worldwide.

## Workload Modifiers

Games that are poorly optimized or highly graphically intensive consume more processing power. Additionally, streaming at    **K**,                **FPS**, or handling popular AAA titles can dramatically increase GPU usage. The simplified reward formula is:

Copy

$$\text{Total YOM/hour} = \text{Base Reward} \times \text{Streaming Difficulty Multiplier} \times \text{Regional Modifier}$$

Where:

- **Base Reward** corresponds to the session-based ad revenue (aiming for ~\$    /hour).
- **Difficulty Multiplier** reflects game intensity or stream quality settings.
- **Regional Modifier** adjusts for advertising CPM and local operating costs.

In simpler terms, **your node earns more for harder work**. This dynamic reward component incentivizes operators to accept and successfully run even the most demanding game sessions, since those yield higher returns. It also motivates game developers to optimize their content (because highly unoptimized content costs more to stream), balancing the ecosystem's efficiency.

## Earnings Expectations: Practical Examples

One of the most common questions potential node operators ask is “**How much can I earn?**”. YOM provides a realistic forecast by modeling node earnings as a **Gaussian (bell curve) distribution** centered around an expected average.

The assumption is that on **average, a node will be utilized about**                      **% of the time** it's available. In other words, roughly half the hours your node is up, it will be actively streaming a session (the other half it might be idle, waiting for demand). Based on network simulations under current conditions, YOM estimates the following monthly earnings per node (for one license running full-time):

- **Average Monthly Earnings:** \$                      per node (assuming                      % workload allocation).
- **Lower Bound (-                      SD):** \$                      per node, reflecting minimal workload assignments.
- **Upper Bound (+                      SD):** \$                      per node, representing optimal demand and performance.

These figures give a **typical range** for a single node license's monthly income. The distribution is roughly bell-shaped, meaning most months you can expect to be somewhere around the middle of this range, with occasional dips toward the low end or spikes toward the high end.

It's worth noting that **these numbers scale linearly with the number of node licenses you run on your hardware** (assuming your machine can handle it). For example, if you operate a rig with                      **licenses concurrently (e.g. a high-end GPU running five game streams)**, your projected earnings would be about five times the single-node values. Concretely, *five* nodes could yield around \$                      **per month on average**, with a lower-bound near \$                      and an upper-bound around \$                      .

YOM also offers a **minimum earnings guarantee** of \$                      /month per node if your node remains consistently available (                      hours daily) but receives no sessions, ensuring a baseline reward.



# Getting More Work: Maximizing Workload Allocation

Understanding **how the YOM network distributes workload** is crucial, because it directly affects your earnings. While node rewards are fixed per region, the key to increasing total earnings lies in securing more workload by optimizing several factors that influence allocation. Workload distribution is determined by a combination of **demand-driven factors** and **operator performance metrics**.

- **Reputation:** High uptime, reliability, and active community engagement (accumulating XP through staking and active participation) significantly boost your node's priority.
- **Connectivity:** Nodes offering low ping and stable connections are preferred for session assignments. Improve this by hosting your node close to target players and using a reliable wired internet connection.
- **Time-of-Day:** Gaming demand peaks evenings and weekends. Ensuring your node is available during these periods increases your likelihood of receiving session assignments.

Apart from the above factors, YOM's orchestrator also performs some balancing to prevent any single node from hogging all jobs or remaining completely idle for too long. If two nodes are equally qualified to take a session (similar latency and reputation), the network might rotate assignments to spread out earnings. Likewise, during periods of **low overall demand**, not every node can be busy %+ of the time – some will sit idle. In such cases, the system may cycle through available nodes so that each gets at least a few sessions instead of one node taking them all. This balancing act is also constrained by *regional demand*: a surplus of nodes in one region won't help another region's users. Thus, part of being a node operator is understanding the supply-demand in your chosen region and possibly adapting (which we'll address in the next section on increasing your workload).

## Practical Steps to Increase Earnings

- **Optimize for Uptime:** Aim for %+ uptime. Utilize reliable hardware, power backups, and monitoring tools to maintain consistent availability.
- **Strategic Deployment:** Deploy nodes in high-value or underserved regions to capture higher session volumes and per-session rewards.
- **Scale Smartly:** Add additional node licenses as your hardware comfortably allows, maximizing GPU usage and total earnings.
- **Engage Actively:** Participate in community events and stake earned \$YOM tokens to boost your node's reputation, placing you at the front of the line for session allocation.

# Deployment Options: Self-Hosting vs. Node-as-a-Service

Choose based on your technical comfort and desired level of involvement. Many operators use a hybrid approach for convenience and to max out profitability.

Feature

Delegation

Self-Hosting

Technical Knowledge

None

Required (Basic to Advanced)

Setup Complexity

Very Low

Moderate to High

Earning Potential

Fixed & Consistent

Higher Potential with Optimization

Hardware Requirements

None

Gaming PC or YOM Device Required

Maintenance

Managed by NaaS Provider

Self-Maintained

## Conclusion & Best Practices Summary

Running a YOM node is a practical opportunity to earn steady passive income from real-world gaming activity. Remember these best practices:

- **Maintain high uptime and reliability.**
- **Optimize latency and connectivity.**
- **Target high-demand periods and regions.**
- **Actively manage your node reputation and XP.**

By implementing these strategies – from technical tuning to strategic planning – you can **maximize your node's workload**. The result is not only higher earnings but also a more resilient and valuable network. YOM's model explicitly **rewards performance and active contribution**, so operators who put in the effort to optimize will see that reflected in their revenue. The network is still young, which means early and proactive participants can carve out a strong position as “go-to” nodes as demand scales up.

[Previous](#)[UnityNext](#)[Getting Started](#)

Last updated      days ago

Was this helpful?

---

## Getting Started | YOM Docs

**Source:** <https://docs.yom.net/earn/getting-started>

As a node operator, your gaming PC becomes part of YOM's distributed infrastructure. Your GPU power is used to process and stream games, enabling seamless gaming experiences. In return, you earn \$YOM tokens based on the workload handled by your nodes.

This manual provides step-by-step guidance for setting up, operating, and optimizing a YOM node, ensuring you maximize your earnings and contribute effectively to the decentralized gaming ecosystem.

[Previous](#)[OverviewNext](#)[Hardware Requirements](#)

Last updated      days ago

Was this helpful?

---

# Hardware Requirements | YOM Docs

**Source:** <https://docs.yom.net/earn/getting-started/hardware-requirements>

Optimizing your rig for YOM's self-hosted nodes is crucial to maximizing profitability and ensuring efficient operation. Whether you're using a **plug-and-play device** or the **code-based approach**, understanding the balance between hardware investment and potential earnings will help you make informed decisions. This guide focuses on key aspects such as **hardware specifications, cost considerations, and scaling strategies** for optimal node performance.

## Understanding Rig Costs and Profitability

When setting up your rig, it's important to consider both **initial investment** and **long-term operational efficiency**. The total cost of your setup depends on:

- **Rig Cost (R):** The cost of acquiring or upgrading your gaming hardware.
- **Device Cost:** The price of YOM's plug-and-play device, which includes pre-installed Node OS and two free node licenses.
- **Additional Node Licenses:** The cost of expanding your rig's capacity by purchasing extra node licenses.

## Cost Optimization Considerations:

### Key Factors

#### Impact on Cost/Profit

Rig Type (Pre-owned vs. New)

Lowering upfront costs by utilizing existing hardware.

Capacity Expansion

Adding more nodes increases earning potential.

Power Efficiency

Energy-efficient rigs lower operational costs.

Technical Setup

Plug-and-play is easier but may have higher upfront costs.

# Cost Overview (Based on Setup Type)

The table below provides estimated costs for your potential investment. As you can see the device covers            base licenses. To achieve max yield, we recommend upgrading your rig to max out its node capacity:

**Rig Cost (R)**

Node **Capacity**

**Device Cost**

**Additional Node Cost**

**Total Cost (Plug-and-Play)**

\$

\$

\$

\$

\$

\$

\$

\$

\$ ,

\$

\$

\$

\$ ,

\$

\$

\$

*Note that these are approximations, assuming you are investing into a new rig. Exclude the rig cost in your personal calculations if you already own a gaming machine.*

## Recommended System Specifications (For Higher Yields):

- **GPU:** NVIDIA RTX 4090 / RTX 5090 or AMD RX 7900 XTX for maximum session handling.
- **CPU:** Intel i9-14900K / i9-13900K or AMD Ryzen 9 7950X / 9950X for seamless processing.
- **RAM:** 64GB+ for smooth operations with multiple nodes.
- **Storage:** 4TB NVMe SSD for quicker load times and caching of game data.
- **Internet:** 10Gbps fiber connection with a wired Ethernet setup for low latency.

## Maximizing Performance with the Right Setup

To optimize your rig for the best efficiency and earnings, consider the following:

### Cooling Solutions:

- Invest in high-performance cooling systems (e.g., liquid cooling, additional case fans) to prevent overheating and maintain stable operation.
- Ensure proper airflow in your setup to extend hardware lifespan.

### Power Management:

- Use energy-efficient power supplies (80+ Gold or Platinum rated) to reduce electricity costs.
- Enable power-saving settings without compromising performance.

### Scaling Up:

- Start with a smaller capacity and gradually expand by adding node licenses to your existing rig.
- Consider multiple rigs if your space and budget allow for better distribution of workloads.

# Flexibility and Expansion Opportunities

YOM provides flexible options for expanding your node operations based on your financial and technical capabilities.

## Key Flexibility Features:

- **Delegation Option:** If you later decide to reduce operational effort, you can delegate nodes to NaaS providers without selling your licenses.
- **Incremental Growth:** Start with a lower-capacity rig and expand based on your earnings and budget.
- **Regional Optimization:** Deploy nodes in high-demand regions to maximize profitability through regional modifiers.

This might be your machine.

[Previous](#)[Getting Started](#)[Next](#)[Installation Guide](#)

Last updated      months ago

Was this helpful?

---

## Installation Guide | YOM Docs

**Source:** <https://docs.yom.net/earn/getting-started/installation-guide>

Certain gaming rig configurations will automatically recognize the external device as the primary boot drive, and will be entirely Plug and Play. However, as some systems will require a change in the boot order, this installation guide provides instructions for both. If you are unsure, following the instructions for setting up Plug and Play is advised.

### Setting up Plug and Play

- . Bitlocker disabled: Open Control Panel, click System and Security, navigate to Bitlocker Drive Encryption and disable it & store the recovery key in a safe location
- . Booting in the bios: Restart the computer and hit the designated key to enter the boot menu if you're unsure what key this is the most common keys are F<sub>2</sub>, F<sub>10</sub>, F<sub>11</sub> & DEL.
- . Advanced settings: Look for the advanced settings menu and find the corresponding key to navigate to the menu, F<sub>2</sub>, F<sub>10</sub>, F<sub>11</sub> are most commonly used

- . Boot order: navigate to the boot tab in the advanced menu and look for the boot order settings. Select the node as the first option. (tester note Ubuntu Exascend)
- . Save the settings: navigate to the Save setting tab and select Save changes and restart or something similar to this

## Plug and Play Installation

- . Plug in the Device: Insert the pre-configured SSD or USB device into your gaming PC.
- . Boot Your Computer: The system will automatically launch the YOM Node OS.
- . Register Your Node: A QR code will appear on the screen. Scan it using the wallet that contains your \$YRX tokens.
- . Automatic Setup: The YOM Node OS downloads necessary updates and finalizes the setup. The display will show what stages of the installation you're in.
- . Monitor & Manage: A QR code will appear on the screen. Scan it to track node performance and earnings.

If any issues are encountered please navigate to the Troubleshooting & Common issues section of this manual or find out how to contact us in the Community & Support section.

[Previous](#)[Hardware Requirements](#)[Next](#)[Monitoring Your Node](#)

Last updated      months ago

Was this helpful?

---

## Monitoring Your Node | YOM Docs

**Source:** <https://docs.yom.net/earn/getting-started/monitoring-your-node>

The YOM Dashboard at [app.yom.net](https://app.yom.net) provides real-time insights into your node's performance and earnings.

### Key Features

- Live GPU utilization & memory usage tracking
- Workload distribution monitoring
- Performance analytics & earning history

Regular monitoring helps optimize uptime and maximize earnings.

[Previous](#)[Installation Guide](#)[Next](#)[NANO](#)



Last updated      months ago

Was this helpful?

---

## NANO | YOM Docs

**Source:** <https://docs.yom.net/earn/nano>

Our **Plug-and-Play disk** offers a straightforward way to self-host a YOM node, combining the benefits of full control with an easy setup process. With this option, you can quickly deploy your node without dealing with complex software installations. However, to ensure optimal performance and maximize your earnings, your system must meet the necessary hardware requirements.

### Why NANO?

The Plug-and-Play device is ideal for users who want to self-host their nodes with minimal technical effort. It provides:

- **Quick and Easy Setup:** Simply plug the device into your gaming rig and start earning.
- **Pre-Configured System:** Comes with YOM's Node OS pre-installed and optimized for cloud gaming.
- **No Upfront License Purchase Needed:** Includes      **free node licenses**, allowing you to start immediately.
- **Scalability:** Easily expand your setup by adding more node licenses via the YOM dashboard.

### Step      : Ensure Your Hardware Meets the Requirements

To run the Plug-and-Play device efficiently, your gaming rig must meet the following **minimum hardware requirements**:

#### Recommended System Requirements for Optimal Performance:

- **GPU:** Any NVIDIA RTX config (      RTX and higher) with high VRAM to maximize node capacity.
- **CPU:** Intel i    /i      or AMD Ryzen      /      for better multitasking and stability.
- **RAM:**      GB would be sufficient on low config setups, but      GB+ is recommended.
- **Storage:** No disk needed as you will boots from the provided device, containing      TB of SSD capacity.
- **Internet:** Wired Ethernet connection (min      mbps upload).

- **Cooling System:** Proper airflow and cooling solutions to prevent overheating.

## **Step : Order Your Plug-and-Play Device**

Once you've confirmed your system meets the requirements, you can proceed with ordering the device:

### **. Visit the YOM Marketplace:**

- Go to [app.yom.net](https://app.yom.net) and navigate to the hardware section.
- Order the **Plug-and-Play device** for \$ , which includes free node licenses.

### **. Provide Your Shipping Details:**

- Enter your delivery address and complete the payment process.
- Track your order status in your YOM account dashboard.

### **. Wait for Delivery:**

- Devices typically arrive within - **business days**, depending on your location.
- Ensure your gaming rig is ready for installation.

## **Step : Setting Up the Plug-and-Play Device**

Once your device arrives, follow these simple steps to get it up and running:

### **. Connect the Device to Your PC:**

- Plug the SSD into your PC, or via an external USB enclosure if supported.
- Ensure the connection is secure and properly detected by your system BIOS.

### **. Boot from the Plug-and-Play Device:**

- Restart your PC and access the BIOS by pressing F2 , F12 , or DEL during startup.
- Set the YOM SSD as the primary boot device.
- Save and exit the BIOS settings to continue.

### **. Follow On-Screen Setup Instructions:**

- Once the system boots into the YOM Node OS, follow the guided setup to:
  - Connect your Web wallet (MetaMask or compatible wallet).

- Activate your included node licenses.
- Configure your internet connection and other performance settings.

## Step : Expanding Your Node Capacity

After your initial setup, you can increase your earnings potential by adding more nodes to your existing setup.

### . Purchase Additional Node Licenses:

- Buy extra licenses from the YOM marketplace.
- Store them securely in your Web wallet.

### . Assign More Nodes to Your Device:

- Log into the YOM dashboard and navigate to the **Node Management** section.
- Select your device and assign additional nodes to increase workload capacity.

### . Optimize for Efficiency:

- Monitor system resources and ensure your hardware can handle additional workloads without performance degradation.

## Step : Monitoring and Managing Your Node

Once your node is operational, you can track its performance using the YOM dashboard.

### Dashboard Features:

- **Earnings Overview:** Monitor your daily and monthly earnings.
- **Uptime Tracking:** Ensure your node remains online for maximum rewards.
- **Workload Distribution:** View how your node is contributing to the network.

### Tips for Maintaining Performance:

- Keep your system powered on / to ensure maximum uptime.
- Use a stable wired internet connection to avoid packet loss.
- Regularly update your system firmware and software patches.

[Previous](#)[Monitoring Your Node](#)[Next](#)[Delegation](#)

Last updated      days ago

Was this helpful?

---

## Delegation | YOM Docs

**Source:** <https://docs.yom.net/earn/delegation>

Delegating your node to a **Node-as-a-Service (NaaS)** provider is the easiest and most convenient way to participate in the YOM network and earn passive income. With delegation, you don't need to worry about technical complexities such as setup, monitoring, and maintenance. Instead, a trusted NaaS provider takes care of everything while you enjoy consistent earnings from your node.

### Why Choose Delegation?

Delegating your node provides several advantages, making it the perfect choice for users who prefer a hands-off approach. Key Benefits of Delegation:

- . **Ease of Use:** No technical knowledge required—just delegate and start earning.
- . **Hassle-Free Scaling:** Easily expand by purchasing and delegating additional node licenses without worrying about hardware capacity.
- . **Fully Managed Service:** A Node-as-a-Service (NaaS) provider takes care of setup, monitoring, and maintenance.
- . **Predictable Earnings:** Enjoy stable, passive income without the need for active involvement.
- . **Lower Risk Exposure:** No hardware investments or operational risks associated with self-hosting.

### Step : Acquire a Node

Before you can delegate your node, you need to acquire one. Owning a node grants you access to the YOM network and allows you to delegate your node to a NaaS provider. How to Purchase a Node License:

- . **Visit the [app.yom.net/nodes/sales](https://app.yom.net/nodes/sales):**
  - Purchase a node license (NFT) via YOM's trusted partners.
  - Ensure that the seller is a verified YOM partner to avoid fraudulent transactions.
- . **Store Your License Securely:**
  - Once purchased, transfer the Node NFT to your Web wallet (e.g., MetaMask).

- Ensure you back up your wallet's recovery phrase securely.

## **Step : Delegating Your Node**

After obtaining your Node License, follow these steps to delegate your node to a NaaS provider and start earning:

### **. Access the YOM Webapp:**

- Navigate to [app.yom.net](https://app.yom.net) and connect your Web wallet that holds the Node NFT.

### **. Navigate to the Node Management Section:**

- In the dashboard, find the "**Node Management**" tab.
- Select the node license you wish to delegate.

### **. Choose a NaaS Provider:**

- Browse through the list of available Node-as-a-Service providers.
- Evaluate providers based on their uptime guarantees, historical performance, and service fees.

### **. Confirm Delegation:**

- Click the "**Delegate**" button and confirm the transaction in your Web wallet.
- Once confirmed, your node will be handed over to the NaaS provider for management.

## **Step : Monitoring Your Delegated Node**

Once your node is successfully delegated, you can monitor its performance through the YOM dashboard. The dashboard provides valuable insights, including:

- **Earnings Overview:** Track your daily, weekly, and monthly rewards based on workload allocations.
- **Performance Metrics:** See how well your node is performing in terms of uptime and resource allocation.
- **Leaderboard Ranking:** Monitor your rank and reputation in the network based on \$YRX accumulation and performance.

## Step : Earnings and Payouts

With delegation, you can expect monthly earnings within an estimated range of:

- \$ to \$ per month per node, depending on network activity and workload saturation.

### Important Considerations:

- **Earnings vary based on:**
  - The overall demand for cloud gaming sessions.
  - Regional network saturation and available workload.
  - Your accumulated XP score, which influences workload priority.
- **Payouts are processed automatically** at the end of each earning period and distributed directly to your Web wallet.

## Step : Expanding Your Delegation Portfolio

Once you are comfortable with delegation, you can increase your earning potential by acquiring and delegating additional nodes. This allows you to benefit from multiple revenue streams with minimal effort.

### To scale up your earnings:

- . Purchase additional node licenses from the YOM marketplace.
- . Follow the same delegation process via the dashboard.
- . Monitor multiple nodes and optimize your delegation strategy.

## Frequently Asked Questions (FAQs)

. **Can I switch to self-hosting later if I change my mind?** Yes, you can revoke delegation at any time and transition to self-hosting if you prefer a more hands-on approach.

. **What happens if my NaaS provider underperforms?** If your provider fails to meet uptime and performance standards, you can re-delegate your node to another provider via the YOM dashboard.

. **Are there any fees associated with delegation?** NaaS providers may charge a small service fee, which is deducted from your monthly earnings. Be sure to check provider details before delegation.

. **How secure is the delegation process?** The delegation process is secured through blockchain smart contracts, ensuring transparency and immutability. Your Node NFT remains safely stored in your wallet.

[PreviousNANONextToken Disclaimer](#)

Last updated      months ago

Was this helpful?

---

## Token Disclaimer | YOM Docs

**Source:** <https://docs.yom.net/compliance/token-disclaimer>

PLEASE READ THE ENTIRETY OF THIS "NOTICE AND DISCLAIMER" SECTION CAREFULLY. NOTHING HEREIN CONSTITUTES LEGAL, FINANCIAL, BUSINESS OR TAX ADVICE AND YOU SHOULD CONSULT YOUR OWN LEGAL, FINANCIAL, TAX OR OTHER PROFESSIONAL ADVISOR(S) BEFORE ENGAGING IN ANY ACTIVITY IN CONNECTION HEREWITH. NEITHER YOM B.V. (THE **COMPANY**), ANY OF THE PROJECT TEAM MEMBERS (THE **YOM TEAM**) WHO HAVE WORKED ON YOM (AS DEFINED HEREIN) OR PROJECT TO DEVELOP YOM IN ANY WAY WHATSOEVER, ANY DISTRIBUTOR/VENDOR OF \$YOM TOKENS (THE **DISTRIBUTOR**), NOR ANY SERVICE PROVIDER SHALL BE LIABLE FOR ANY KIND OF DIRECT OR INDIRECT DAMAGE OR LOSS WHATSOEVER WHICH YOU MAY SUFFER IN CONNECTION WITH ACCESSING THE PAPER, DECK OR MATERIAL RELATING TO \$YOM (THE **TOKEN DOCUMENTATION**) AVAILABLE ON THE WEBSITE AT WWW.YOM.NET (THE **WEBSITE**, INCLUDING ANY SUB-DOMAINS THEREON) OR ANY OTHER WEBSITES OR MATERIALS PUBLISHED BY THE COMPANY.

### . PROJECT PURPOSE

You agree that you are acquiring \$YOM to participate in YOM and to obtain services on the ecosystem thereon. The Company, the Distributor and their respective affiliates would develop and contribute to the underlying source code for YOM. The Company is acting solely as an arms' length third party in relation to the \$YOM distribution, and not in the capacity as a financial advisor or fiduciary of any person with regard to the distribution of \$YOM.

### . NATURE OF THE DOCUMENTATION:

The Token Documentation is a conceptual paper that articulates some of the main design principles and ideas for the creation of a digital token to be known as \$YOM.

The Token Documentation and the Website are intended for general informational purposes only and do not constitute a prospectus, an offer document, an offer of securities, a solicitation for investment, any offer to sell any product, item, or asset (whether digital or otherwise), or any offer to engage in business with any external individual or entity provided in said documentation. The information herein may not be exhaustive and does not imply any element of, or solicit in any way, a contractual relationship. There is no assurance as to the accuracy or completeness of such information and no representation, warranty or undertaking is or purported to be provided as to the accuracy or completeness of such information. Where the Token Documentation or the Website includes information that has been obtained from third party sources, the Company, the Distributor, their respective affiliates and/or the YOM team have not independently verified the accuracy or completeness of such information. Further, you acknowledge that circumstances may change and that the Token Documentation or the Website may become outdated as a result; and neither the Company nor the Distributor is under any obligation to update or correct this document in connection therewith.

## **. TOKEN DOCUMENTATION**

Nothing in the Token Documentation or the Website constitutes any offer by the Company, the Distributor, or the YOM team to sell any \$YOM (as defined herein) nor shall it or any part of it nor the fact of its presentation form the basis of, or be relied upon in connection with, any contract or investment decision. Nothing contained in the Token Documentation or the Website is or may be relied upon as a promise, representation or undertaking as to the future performance of YOM. The agreement between the Distributor (or any third party) and you, in relation to any distribution or transfer of \$YOM, is to be governed only by the separate terms and conditions of such agreement.

The information set out in the Token Documentation and the Website is for community discussion only and is not legally binding. No person is bound to enter into any contract or binding legal commitment in relation to the acquisition of \$YOM, and no digital asset or other form of payment is to be accepted on the basis of the Token Documentation or the Website. The agreement for distribution of \$YOM and/or continued holding of \$YOM shall be governed by a separate set of Terms and Conditions or Token Distribution Agreement (as the case may be) setting out the terms of such distribution and/or continued holding of \$YOM (the Terms and Conditions), which shall be separately provided to you or made available on the Website. The Terms and Conditions must be read together with the Token Documentation. In the event of any inconsistencies between the Terms and Conditions and the Token Documentation or the Website, the Terms and Conditions shall prevail.



## **. DEEMED REPRESENTATIONS AND WARRANTIES**

By accessing the Token Documentation or the Website (or any part thereof), you shall be deemed to represent and warrant to the Company, the Distributor, their respective affiliates, and the YOM team as follows:

- (a) in any decision to acquire any \$YOM, you have not relied on and shall not rely on any statement set out in the Token Documentation or the Website;
- (b) you will and shall at your own expense ensure compliance with all laws, regulatory requirements and restrictions applicable to you (as the case may be);
- (c) you acknowledge, understand and agree that \$YOM may have no value, there is no guarantee or representation of value or liquidity for \$YOM, and \$YOM is not an investment product nor is it intended for any speculative investment whatsoever;
- (d) none of the Company, the Distributor, their respective affiliates, and/or the YOM team members shall be responsible for or liable for the value of \$YOM, the transferability and/or liquidity of \$YOM and/or the availability of any market for \$YOM through third parties or otherwise; and
- (e) you acknowledge, understand and agree that you are not eligible to participate in the distribution of \$YOM if you are a citizen, national, resident (tax or otherwise), domiciliary and/or green card holder of a geographic area or country (i) where it is likely that the distribution of \$YOM would be construed as the sale of a security (howsoever named), financial service or investment product and/or (ii) where participation in token distributions is prohibited by applicable law, decree, regulation, treaty, or administrative act (including without limitation the United States of America, Canada, and the People's Republic of China); and to this effect you agree to provide all such identity verification document when requested in order for the relevant checks to be carried out.

The Company, the Distributor and the YOM team do not and do not purport to make, and hereby disclaims, all representations, warranties or undertaking to any entity or person (including without limitation warranties as to the accuracy, completeness, timeliness, or reliability of the contents of the Token Documentation or the Website, or any other materials published by the Company or the Distributor). To the maximum extent permitted by law, the Company, the Distributor, their respective affiliates and service providers shall not be liable for any indirect, special, incidental, consequential or other losses of any kind, in tort, contract or otherwise (including, without limitation, any liability arising from default or negligence on the part of any of them, or any loss of revenue, income or profits, and loss of use or data) arising from the use of the Token Documentation or the Website, or any other materials published, or its contents (including without limitation any errors or omissions) or otherwise arising in connection

with the same. Prospective acquirors of \$YOM should carefully consider and evaluate all risks and uncertainties (including financial and legal risks and uncertainties) associated with the distribution of \$YOM, the Company, the Distributor and the YOM team.

## **. \$YOM TOKEN**

\$YOM are designed to be utilised, and that is the goal of the \$YOM distribution. In particular, it is highlighted that \$YOM:

(a) does not have any tangible or physical manifestation, and does not have any intrinsic value (nor does any person make any representation or give any commitment as to its value);

(b) is non-refundable and cannot be exchanged for cash (or its equivalent value in any other digital asset) or any payment obligation by the Company, the Distributor or any of their respective affiliates;

(c) does not represent or confer on the token holder any right of any form with respect to the Company, the Distributor (or any of their respective affiliates), or its revenues or assets, including without limitation any right to receive future dividends, revenue, shares, ownership right or stake, share or security, any voting, distribution, redemption, liquidation, proprietary (including all forms of intellectual property or licence rights), right to receive accounts, financial statements or other financial data, the right to requisition or participate in shareholder meetings, the right to nominate a director, or other financial or legal rights or equivalent rights, or intellectual property rights or any other form of participation in or relating to YOM, the Company, the Distributor and/or their service providers;

(d) is not intended to represent any rights under a contract for differences or under any other contract the purpose or pretended purpose of which is to secure a profit or avoid a loss;

(e) is not intended to be a representation of money (including electronic money), security, commodity, bond, debt instrument, unit in a collective investment scheme or any other kind of financial instrument or investment;

(f) is not a loan to the Company, the Distributor or any of their respective affiliates, is not intended to represent a debt owed by the Company, the Distributor or any of their respective affiliates, and there is no expectation of profit; and

(g) does not provide the token holder with any ownership or other interest in the Company, the Distributor or any of their respective affiliates.

Notwithstanding the \$YOM distribution, users have no economic or legal right over or beneficial interest in the assets of the Company, the Distributor, or any of their affiliates after the token distribution.

To the extent a secondary market or exchange for trading \$YOM does develop, it would be run and operated wholly independently of the Company, the Distributor, the distribution of \$YOM and YOM. Neither the Company nor the Distributor will create such secondary markets nor will either entity act as an exchange for \$YOM.

## **. INFORMATIONAL PURPOSES ONLY**

The information set out herein is only conceptual, and describes the future development goals for YOM to be developed. In particular, the project roadmap in the Token Documentation is being shared in order to outline some of the plans of the YOM team, and is provided solely for **INFORMATIONAL PURPOSES** and does not constitute any binding commitment. Please do not rely on this information in deciding whether to participate in the token distribution because ultimately, the development, release, and timing of any products, features or functionality remains at the sole discretion of the Company, the Distributor or their respective affiliates, and is subject to change. Further, the Token Documentation or the Website may be amended or replaced from time to time. There are no obligations to update the Token Documentation or the Website, or to provide recipients with access to any information beyond what is provided herein.

## **. REGULATORY APPROVAL**

No regulatory authority has examined or approved, whether formally or informally, any of the information set out in the Token Documentation or the Website. No such action or assurance has been or will be taken under the laws, regulatory requirements or rules of any jurisdiction. The publication, distribution or dissemination of the Token Documentation or the Website does not imply that the applicable laws, regulatory requirements or rules have been complied with.

## **. CAUTIONARY NOTE ON FUTURE-LOOKING STATEMENTS**

All statements contained herein, statements made in press releases or in any place accessible by the public and oral statements that may be made by the Company, the Distributor and/or the YOM team, may constitute forward-looking statements (including statements regarding the intent, belief or current expectations with respect to market conditions, business strategy and plans, financial condition, specific provisions and risk management practices). You are cautioned not to place undue reliance on these forward-looking statements given that these statements involve known and unknown risks, uncertainties and other factors that may cause the actual future results to be

materially different from that described by such forward-looking statements, and no independent third party has reviewed the reasonableness of any such statements or assumptions. These forward-looking statements are applicable only as of the date indicated in the Token Documentation, and the Company, the Distributor as well as the YOM team expressly disclaim any responsibility (whether express or implied) to release any revisions to these forward-looking statements to reflect events after such date.

## **. REFERENCES TO COMPANIES AND PLATFORMS**

The use of any company and/or platform names or trademarks herein (save for those which relate to the Company, the Distributor or their respective affiliates) does not imply any affiliation with, or endorsement by, any third party. References in the Token Documentation or the Website to specific companies and platforms are for illustrative purposes only.

## **. ENGLISH LANGUAGE**

The Token Documentation and the Website may be translated into a language other than English for reference purpose only and in the event of conflict or ambiguity between the English language version and translated versions of the Token Documentation or the Website, the English language versions shall prevail. You acknowledge that you have read and understood the English language version of the Token Documentation and the Website.

## **. NO DISTRIBUTION**

No part of the Token Documentation or the Website is to be copied, reproduced, distributed or disseminated in any way without the prior written consent of the Company or the Distributor. By attending any presentation on this Token Documentation or by accepting any hard or soft copy of the Token Documentation, you agree to be bound by the foregoing limitations.

[Previous](#)[Delegation](#)[Next](#)[Terms and Conditions](#)

Last updated      months ago

Was this helpful?

---

## **Terms and Conditions | YOM Docs**

**Source:** <https://docs.yom.net/compliance/terms-and-conditions>

PLEASE READ THIS AGREEMENT CAREFULLY BEFORE ACCESSING OR USING THE **WEBSITE** (THE WEBSITE, INCLUDING ANY SUB-DOMAINS THEREON), INCLUDING, WITHOUT LIMITATION, THE CREATION, MINTING, LISTING, PURCHASE, SALE, EXCHANGE, OR MODIFICATION OF CERTAIN DIGITAL ASSETS (COLLECTIVELY, THE **SERVICE**). BY ACCESSING OR USING ANY PART OF THE WEBSITE, YOU AGREE TO BECOME BOUND BY THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT, THEN YOU MAY NOT ACCESS THE WEBSITE, USE ANY SERVICES, OR BUY CRYPTO ASSETS. THE WEBSITE IS AVAILABLE ONLY TO INDIVIDUALS WHO ARE AT LEAST        YEARS OLD.

THE WEBSITE IS OFFERED SUBJECT TO YOUR ACCEPTANCE WITHOUT MODIFICATION OF ALL THE TERMS AND CONDITIONS CONTAINED HEREIN AND ALL OTHER OPERATING RULES, POLICIES AND PROCEDURES THAT MAY BE PUBLISHED FROM TIME TO TIME ON THE WEBSITE (TAKEN TOGETHER, THE **AGREEMENT**). YOM IS A PLATFORM. YOM IS NOT A BROKER, FINANCIAL INSTITUTION, OR CREDITOR. THE SERVICES OFFERED ON THE PLATFORM ARE OF ADMINISTRATIVE NATURE ONLY.

## **. PRIVACY POLICY**

YOM Privacy Policy explains the way we handle and protect your personal data in relation to your use and browsing of the Website. By agreeing to the present terms and conditions and to be able to use the Service, you also agree to our Privacy Policy.

## **. ACCOUNT REGISTRATION**

In order to list and purchase crypto assets, you need to register for an account on the Service ("Account"). By creating an Account, you agree to:

- a) provide accurate, current and complete Account information about yourself;
- b) maintain and promptly update from time to time, as necessary, your Account information;
- c) maintain the security of your password and accept all risks of unauthorized access to your Account and the information you provide to us; and
- d) immediately notify us if you discover or otherwise suspect any security breaches related to the Service, or your Account.

YOM will block multiple accounts of the same user.

Also, you agree that you will not:

- a) create another account if YOM disabled one you had, unless you have YOM's written permission first;
- b) buy, sell, rent or lease access to your Account or username, unless you have YOM's written permission first;
- c) share your Account password with anyone; or
- d) log in or try to login to access the Service through unauthorized third party applications or clients.

YOM may require you to provide additional information and documents at the request of any competent authority or in case of application of any applicable law or regulation, including laws related to anti-laundering (legalization) of incomes obtained by criminal means, or for counteracting financing of terrorism. YOM may also require you to provide additional information and documents in cases where it has reasons to believe that:

- a) your Account is being used for money laundering or for any other illegal activity;
- b) you have concealed or reported false identification information and other details; or
- c) transactions made via your Account were made in breach of these Terms.

In such cases, YOM , in its sole discretion, may pause or cancel your purchase transactions until such additional information and documents are reviewed by YOM and accepted as satisfying the requirements of applicable law. If you do not provide complete and accurate information and documents in response to such a request, YOM may refuse to provide the Service.

By registering an Account on YOM, you give us permission to use your name and picture for marketing and promotional purposes. Users registered as creators also understand and agree that YOM may display, reproduce, and distribute their works represented in digital assets minted, listed and tradable on YOM, for the purpose of operating, promoting, sharing, developing, marketing, and advertising the Website, or any other purpose related to YOM.

## **. MODIFICATION TO TERMS OF SERVICE**

Within the limits of applicable law, YOM reserves the right to review and change this Agreement at any time. You are responsible for regularly reviewing these terms and conditions. Continued use and browsing of the Website after such changes shall constitute your consent to such changes.

## **. COMMUNICATION PREFERENCES**

By creating an Account, you consent to receive electronic communications from YOM (e.g. via email or by posting notices to the Service).

These communications may include notices about your Account (e.g. password changes and other transactional information) and are part of your relationship with YOM.

You agree that any notices, agreements, disclosures or other communications that YOM sends to you electronically shall satisfy any legal communication requirements, including, but not limited to, that such communications be in writing.

YOM may also send you promotional communications via email, including, but not limited to, newsletters, special offers, surveys and other news and information we think will be of interest to you. You may opt out of receiving these promotional emails at any time by following the unsubscribe instructions provided therein.

You acknowledge that the ownership of digital assets (NFTs) made available or purchased on the Website may give you the right to view, store, exchange, sell and display the NFT publicly but does not allow or imply commercial use or ownership of intellectual property on the brand, design, music, video, art or other media displayed in your digital asset NFTs, unless specifically stated otherwise.

## **. DISCLAIMERS**

Except as expressly provided to the contrary in writing by YOM, the Service and content contained therein, and crypto assets listed therein are provided on an “as is” and “as available” basis without warranties or conditions of any kind, either express or implied.

YOM makes no warranty that the Service:

- a) will meet your requirements;
- b) will be available on an uninterrupted and timely basis.

YOM disclaims all other warranties or conditions, express or implied, including, without limitation, implied warranties or conditions of merchantability, fitness for particular purpose, title and non-infringement to the Service, as well to the content published therein.

While YOM attempts to make your access to and use of the service and content safe, you accept the inherent security risks of providing information and dealing online over

the internet and will not hold YOM responsible for any breach of security unless it is due to our gross negligence.

YOM shall not be responsible or liable to you for any loss and take no responsibility for, and shall not be liable to you for any use of crypto assets, including but not limited to any losses, damages or claims arising from:

- a) user error, such as forgotten passwords, incorrectly contruced transactions, or mistyped addresses;
- b) server failure or data loss;
- c) corrupted wallet files;
- d) unauthorized access to applications;
- e) any unauthorized third party activities; including, without limitation, the use of viruses, phishing, brute forcing, or other means of attack against the service or crypto assets.

Crypto assets are intangible digital assets. They exist only by virtue of the ownership record maintained in a blockchain. Any transfer of title that might occur in any unique digital asset occurs on the decentralized ledger within the blockchain. YOM does not guarantee that YOM can effect the transfer of title or right in any crypto assets.

However, YOM requests its users who register as creators to warrant that the crypto assets they mint as NFTs and list through YOM are their own individual creations, which have not previously been published and/or exploited in any manner, in order for them to comply with these Terms, under liability to YOM and other users.

YOM is not responsible for sustained casualties due to vulnerability or any kind of failure, abnormal behavior of software (e.g., wallet, smart contract), blockchains or any other features of crypto assets. YOM is not responsible for casualties due to late report by developers or representatives (or no report at all) of any issues with the blockchain supporting crypto assets including forks, technical node issues or any other issues having fund losses as a result.

Nothing in these Terms shall exclude or limit liability of either party for fraud, death or bodily injury caused by negligence, violation of laws, or any other activity that cannot be limited or excluded by legitimate means.

Some jurisdictions do not allow the exclusion of implied warranties in contracts with consumers, so the above exclusion may not apply to you.



## **. ASSUMPTION OF RISKS**

You accept and acknowledge:

- a) The prices of blockchain assets are extremely volatile. Fluctuations in the price of other digital assets could materially and adversely affect the crypto assets, which may also be subject to significant price volatility. We cannot guarantee that any purchasers of crypto assets will not lose money.
- b) You are solely responsible for determining what, if any, taxes apply to your crypto assets transactions. YOM is not responsible for determining the taxes that apply to crypto assets transactions.
- c) There are risks associated with using an Internet based currency, including but not limited to, the risk of hardware, software and Internet connections, the risk of malicious software introduction, and the risk that third parties may obtain unauthorized access to information stored within your wallet. You accept and acknowledge that YOM shall not be responsible for any communication failures, disruptions, errors, distortions or delays you may experience when using the crypto assets, however caused.
- d) A lack of use or public interest in the creation and development of distributed ecosystems could negatively impact the development of those ecosystems and related applications, and could therefore also negatively impact the potential utility or value of crypto assets.
- e) The regulatory regime governing blockchain technologies, cryptocurrencies, and tokens is uncertain, and new regulations or policies may materially adversely affect the development of the Service and the utility of crypto assets.
- f) There are risks associated with purchasing user generated content, including but not limited to, the risk of purchasing counterfeit assets, mislabeled assets, assets that are vulnerable to metadata decay, assets on smart contracts with bugs, and assets that may become untransferable. YOM reserves the right to hide collections, contracts, and assets affected by any of these issues or by other issues. Assets you purchase may become inaccessible on YOM. Under no circumstances shall the inability to view your assets on YOM serve as grounds for a claim against YOM.

## **. LIMITATION OF LIABILITY**

To the fullest extent permitted by law, in no event shall YOM be liable to you or any third party for any lost profit or any indirect, consequential, exemplary, incidental, special or punitive damages arising from these terms, the Service, or for any damages related to loss of revenue, loss of profit, loss of business or anticipated savings, loss of

use, loss of goodwill, or loss of data, and whether caused by tort (including negligence), breach of contract, or otherwise.

The access to and use of the Services are at your own discretion and risk, and you shall be solely responsible for any damage to your computer system or mobile device or loss of data resulting therefrom.

Notwithstanding anything to the contrary contained herein, in no event shall the maximum aggregate liability of YOM arising out of or in any way related to these terms, the access to and use of the service, content, crypto assets, or any products or services purchased on the service exceed the greater of:

- a) the amount received by YOM from the sale of crypto assets that are the subject of the claim, and
- b) the operational costs from the sale of crypto assets that are the subject of the claim.

The foregoing limitations of liability shall not apply to liability of YOM for:

- a) death or personal injury caused by a member of YOM's negligence; or for
- b) any injury caused by a member of YOM's fraud or fraudulent misrepresentation.

Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you. Some jurisdictions also limit disclaimers or limitations of liability for personal injury from consumer products, so this limitation may not apply to personal injury claims.

## **. TERMINATION**

Notwithstanding anything contained in these Terms, we reserve the right, without notice and in our sole discretion, to terminate your right to access or use the Service at any time and for any or no reason, and you acknowledge and agree that we shall have no liability or obligation to you in such event and that you will not be entitled to a refund of any amounts that you have already paid to us, to the fullest extent permitted by applicable law.

## **. SEVERABILITY**

If any term, clause or provision of these Terms is held invalid or unenforceable, then that term, clause or provision shall be severable from these Terms and will not affect the validity or enforceability of any remaining part of that term, clause or provision, or any other term, clause or provision of these Terms.

## . APPLICABLE LAW

This Agreement shall be governed in all respects by the substantive laws of the Netherlands. Any controversy, claim, or dispute arising out of or relating to the Agreement shall be subject to the jurisdiction of the competent courts of the Netherlands, the jurisdiction of the Dutch Court being expressly reserved.

[PreviousToken DisclaimerNextPrivacy Policy](#)

Last updated      year ago

Was this helpful?

---

## Privacy Policy | YOM Docs

**Source:** <https://docs.yom.net/compliance/privacy-policy>

THIS PRIVACY POLICY WILL EXPLAIN HOW YOM USES THE PERSONAL INFORMATION WE COLLECT WHEN YOU USE OUR WEBSITE. AS USED HEREIN, **PERSONAL DATA** REFERS TO DATA THAT SPECIFICALLY OR INDIRECTLY IDENTIFIES OR IS REASONABLY CAPABLE OF IDENTIFYING AN INDIVIDUAL, AS WELL AS DATA THAT CAN BE LINKED TO A KNOWN OR REASONABLY IDENTIFIABLE INDIVIDUAL.

### WHAT DATA DO WE COLLECT?

Information we collect includes both information you knowingly and actively provide us when using or participating in any of our services and promotions, and any information automatically sent by your devices in the course of accessing our products and services, as well as information we obtain through third party sources.

## . HOW DO WE COLLECT YOUR PERSONAL DATA?

### Personal Data we could collect from you:

Directly from you, we can collect the following categories of Personal Data:

- Personal information, such as a person's first and last name, email address, username, password, phone number, and mailing address and any other information you directly provide to us on or through the services;
- Transaction information, such as trading, order activity, deposits, withdrawals, and account balances and other commercial activity;

- Communications, including details you give in correspondence, such as account registration, support requests and customer service;
- Sensory information, such as images that you upload to your User Account.

### **Personal Data we collect automatically:**

Through your use of our services, we can automatically collect the following categories of Personal Data:

- Internet identifiers like IP addresses and domain names;
- Usage data, such as system operation, internal and external information related to sites that you visit;
- Device information, such as hardware, operating system, and browser;
- Data on geolocation;
- Cookies can be used to collect Personal Data automatically.

### **Personal Data we could collect from third parties:**

Third-party sources can be used to obtain and/or validate the following types of Personal Data about you:

- Personal information, such as a person's name, email address, phone number, and mailing address;
- Transaction information, such as public blockchain data;
- Analytics information, provided by third party analytics services that we engage to help analyze how users use our services;
- Additional information, at our discretion, to comply with legal obligations.

## **. HOW WILL WE USE YOUR DATA?**

We collect Personal Data about you in order to give you the best possible experience, protect you from the risks of inappropriate use and fraud, and to help us retain and develop our Services. We can use your Personal Data in the following ways, in order to:

- Provide and administer our services to you. In accordance with our Terms and Conditions, we use your Personal Data to provide you with our Services and comply with all relevant laws and regulations. Your Personal Data is processed in accordance with relevant laws and regulations.
- Operate, maintain, improve, personalize, and analyze our services.
- Facilitate the creation of your Account on YOM and its security (including possible multi-step verification).
- Notify you about important changes to our services, such as security or support and maintenance advisories.

- Inform you about other news about our services or any features and content thereon, including newsletters, surveys, offers, promotions, contest, and events via optional newsletters and emails, as described in the section below.
- Detect, prevent or investigate security breaches, fraud and any other unauthorized or illegal activity. We use your Personal Data to identify and prevent account fraud, as well as other prohibited activities.
- Ensure the protection and security of our facilities. To keep your account and YOM safe, we use your Personal Data, which includes information about your computer and your behavior on YOM.
- Assist you with customer service, care and support. When you contact our support team with questions or concerns about your account, we process your Personal Data.
- Promote our goods. We may contact you to provide you with information about our products and services. Only with your permission, which you may revoke at any time, will we do so.
- Other commercial uses. If the intent is revealed to you before we collect the information or if you agree, we can use your Personal Data for additional purposes.

## **. MARKETING**

YOM would like to periodically send you promotional information about products and services of ours that we think you might like.

If you have agreed to receive marketing communications, you may always opt out at a later date.

You have the right at any time to request YOM to stop contacting you for marketing purposes.

If you no longer wish to be contacted for marketing purposes, please contact [\[email protected\]](#).

## **. WHAT ARE YOUR DATA PROTECTION RIGHTS?**

YOM would like to make sure you are fully aware of all your data protection rights. Every user is entitled to the following:

- The right to access – You have the right to request YOM for copies of your personal data.
- The right to rectification – You have the right to request that YOM correct any information you believe is inaccurate. You also have the right to request YOM to complete the information you believe is incomplete.

- The right to erasure – You have the right to request that YOM erase your personal data, under certain conditions.
- The right to restrict processing – You have the right to request that YOM restrict the processing of your personal data, under certain conditions.
- The right to object to processing – You have the right to object to YOM's processing of your personal data, under certain conditions.
- The right to data portability – You have the right to request that YOM transfer the data that we have collected to another organization, or directly to you, under certain conditions.

If you make a request, we have one month to respond to you. If you would like to exercise any of these rights, please contact us at our email: [\[email protected\]](#)

## **. COOKIES**

Cookies are text files placed on your computer to collect standard Internet log information and visitor behavior information. When you visit our Website, we may collect information from you automatically through cookies or similar technology.

For further information, visit [aboutcookies.org](http://aboutcookies.org).

## **. HOW DO WE USE COOKIES?**

YOM uses cookies in a range of ways to improve your experience on our website, including:

- Functionality. We use Cookies to ensure that the Website functions properly, including page loads, keeping you signed in and saving preferences.
- Performance and Analytics. We use Cookies to gather usage and performance data. This enables us to understand how you use our website, and to monitor and improve YOM's performance, our services, and your experience, as well as to detect, prevent and address technical issues.

## **. HOW TO MANAGE COOKIES**

You can set your browser not to accept cookies, and the above website tells you how to remove cookies from your browser. However, in a few cases, some of our website features may not function as a result.

## **. CHANGES TO THIS PRIVACY POLICY**

At our discretion, we may change our privacy policy to reflect updates to our business processes, current acceptable practices, or legislative or regulatory changes. If we

decide to change this privacy policy, we will post the changes here at the same link by which you are accessing this Privacy Policy.

If required by law, we will get your permission or give you the opportunity to opt in to or opt-out of, as applicable, any new uses of your personal information.

. **HOW TO CONTACT US**

If you have any questions about YOM's privacy policy, the data we hold on you (if any), or you would like to exercise one of your data protection rights, please do not hesitate to contact us.

Email us at: [\[email protected\]](#)

[PreviousTerms and ConditionsNextToken Allocation](#)

Last updated        months ago

Was this helpful?

---

# Token Allocation | YOM Docs

**Source:** <https://docs.yom.net/about/token-allocation>

Attribute

Value

Total tokens

, ,

Chain

Solana (liquidity) + peaq (product)

TGE Price

.

Initial market cap

\$ , ,

TGE FDV

\$ , ,

## Token Allocation:

- **Ecosystem** ( . %, **months linear lockup**). This pool is reserved for node operators to support their idle node time via a flat payout rate. This pool gets replenished with % of continuous node rewards. All transactions from this pool are programmatic and on-chain and serve as a buffer for getting to global coverage.
- **Treasury** ( %, **months linear lockup**). The treasury involves all other non-programmatic transactions that have the mission to stimulate adoption towards decentralized cloud compute. This involves payments to get onto exchanges, provide grants to onboard game studios and pay for expenses that bolster YOM's non-profit mission.
- **OPEX** ( , %, **months linear lockup**). Involves a total grant pool allocated to the team, advisors and OPEX expenses.
- **Community / Burned** ( %): This pool is the transfer for our OG holder community who decided to never sell ( months vesting) and bridge. The tokens that didn't vest/bridge, got burned
- **Public Round** ( %). We aim to reserve a small fraction of our supply to cover the marketing costs we spent around the TGE and node rollout.
- **Liquidity pool** ( %). The token remainder is reserved for and added to liquidity pools to facilitate trading on DEXs and exchanges.

[Previous](#)[Privacy Policy](#)[Next](#)[Social Media](#)

Last updated months ago

Was this helpful?

---

## Social Media | YOM Docs

**Source:** <https://docs.yom.net/about/social-media>

YOM is active on different types of social media platforms.

You will find different types of content and communities throughout our social channels. We invite you to join us in the conversation about Your Open Metaverse, the Metaverse itself, and related topics.

◆ [Twitter](#)

◆ [Telegram Community](#)



- ◆ [Telegram Announcements](#)
- ◆ [Discord](#)
- ◆ [YouTube](#)
- ◆ [Medium](#)
- ◆ [Tiktok](#)
- ◆ [LinkedIn](#)
- ◆ [LinkedIn Group](#)
- ◆ [Instagram](#)

[Previous](#)[Token Allocation](#)[Next](#)[FAQ](#)

Last updated      days ago

Was this helpful?

---

## FAQ | YOM Docs

**Source:** <https://docs.yom.net/about/faq>

### . General Questions

#### What is YOM?

YOM is a **decentralized cloud gaming infrastructure (DePIN)** that allows users to stream immersive gaming experiences to any device without the need for expensive hardware. YOM connects idle GPU resources from self-hosted and delegated nodes to studios, brands, and broadcasters, offering scalable, cost-efficient cloud gaming solutions.

#### How does YOM work?

YOM utilizes a decentralized network of gaming machines (nodes) running our custom **Node OS**, which stream gaming sessions to end users. Studios and brands pay for these services, and node operators earn rewards in \$YOM tokens for providing computing resources.

## Who can benefit from YOM?

- **Node Operators:** Earn passive income by contributing GPU resources to the network.
  - **Community Members:** Participate in governance, earn XP rewards, and engage with the ecosystem.
  - **Studios & Brands:** Leverage YOM to provide immersive, high-quality experiences to their audiences at a fraction of traditional cloud gaming costs.
- 

## . Node Operators

### How can I start running a YOM node?

You can start by choosing one of two options:

#### . Delegation (Hands-off approach):

- Purchase a node license for \$\_\_\_\_\_ and delegate it to a Node-as-a-Service (NaaS) provider.

#### . Self-Hosting (Hands-on approach):

- Purchase a **plug-and-play device** or set up the node via a **manual code installation** on your gaming rig.

### What are the hardware requirements for self-hosting?

- **GPU:** Any NVIDIA RTX config (\_\_\_\_\_ RTX and higher) with high VRAM to maximize node capacity.
- **CPU:** Intel i\_\_\_\_\_/i\_\_\_\_\_ or AMD Ryzen\_\_\_\_\_/\_\_\_\_\_ for better multitasking and stability.
- **RAM:**\_\_\_\_\_ GB would be sufficient on low config setups, but\_\_\_\_\_ GB+ is recommended.
- **Storage:** No storage needed for plug n play solution. For custom Linux boot deployment:\_\_\_\_\_ TB+ available space needed.
- **Internet:** Wired Ethernet connection (min\_\_\_\_\_ mbps upload).

### How much can I earn by running a node?

Earnings depend on workload demand and regional factors. On average:

- **Lower Bound (-SD):** \$\_\_\_\_\_ per node/month
- **Average:** \$\_\_\_\_\_ per node/month
- **Upper Bound (+SD):** \$\_\_\_\_\_ per node/month

Earnings assume      % workload utilization across different demand periods.

### What is the difference between delegation and self-hosting?

- **Delegation:** Fully managed by a NaaS provider, with minimal effort required. Lower technical involvement but lower optimization potential.
- **Self-Hosting:** Provides full control over operations and the potential for higher earnings with greater responsibility for setup and maintenance.

### What costs should I consider when self-hosting?

Costs include:

- **Hardware acquisition:** New rigs vs. using existing setups.
  - **Electricity costs:** Managing power efficiency for profitability.
  - **Scaling costs:** Adding more node licenses to increase earnings.
- 

## . Studios & Brands

### Why should I use YOM for cloud gaming?

YOM offers:

- **Cost Efficiency:** Reduce cloud gaming costs from \$ /hour to \$ . /hour.
- **Scalability:** Access a global network of nodes without infrastructure investment.
- **Ease of Integration:** Embed streams directly into websites, apps, and platforms with a single line of code.
- **Customization:** Implement branding, analytics, and monetization strategies easily.

### How is game performance ensured across different devices?

YOM uses an **AI-powered Resource Orchestrator** that dynamically selects the best nodes based on:

- **Latency:** Ensuring the lowest possible ping.
- **Node Reputation:** Prioritizing high-performing nodes with good uptime.
- **Game Availability:** Allocating nodes with pre-loaded game assets for faster deployment.

## What pricing model does YOM use?

YOM follows a **session-based pricing model**, where studios pay:

- \$ . **per gaming session** (based on an average -minute session).
- \$ /month **base fee** for access to analytics and dashboard services.

## How does advertising impact earnings and costs?

The YOM ecosystem integrates advertising seamlessly into gaming sessions. Studios benefit by:

- Leveraging **pre-session ads** that align with their goals.
  - Receiving a revenue share from ad impressions during gameplay.
  - Reducing costs via ad-supported models while maintaining player engagement.
- 

## . Tokenomics & Rewards

### What is \$YOM and how does it work?

\$YOM is the native utility token of the YOM ecosystem, used for:

- Paying for cloud gaming services.
- Rewarding node operators for their contribution.
- Staking to earn additional rewards.
- Ensuring network security and sustainability through a **deflationary model** ( % burn on usage).

### What is XP and its role in the ecosystem?

XP is a **soul-bound value** that represents user contributions and engagement in the ecosystem. It is used to:

- Boost leaderboard rankings for better workload allocation.
- Participate in governance decisions via the YOM DAO.
- Gain eligibility for additional \$YOM drops and rewards.

### Is \$YOM inflationary or deflationary?

YOM follows a **deflationary model**, meaning no new tokens are minted, and a portion of each transaction is burned, reducing supply over time. This ensures value retention for stakeholders.

## How can I earn more rewards?

Increase earnings by:

- Maintaining high uptime and reliability for better workload allocation.
  - Participating in the **Streak Program**, which provides bonus rewards for consistent online presence.
  - Accumulating XP to boost node rankings and priority.
- 

## . Governance & Community

### How does governance work in the YOM ecosystem?

YOM operates under a **DAO model**, allowing the community to influence key decisions, such as:

- Platform policies and content guidelines.
- Allocation of treasury funds to support network growth.
- Voting on feature updates and ecosystem improvements.

### What are the roles within the DAO?

- **Moderators:** Manage community engagement and uphold policies.
- **Commissioners:** Oversee quality assurance and approve projects.
- **Cardinals:** Propose policies and contribute to strategic planning.

### How can I participate in the YOM community?

You can engage with YOM through:

- Joining discussions in **Discord and Telegram**.
  - Contributing to development and governance through XP-based voting.
  - Running community events to onboard new users and expand the ecosystem.
- 

## . Technical & Troubleshooting

### What should I do if my node is not appearing in the dashboard?

- Ensure your wallet is properly connected.
- Verify your internet connection and reboot the system.
- Contact YOM support if the issue persists.

## How do I optimize my rig for better performance?

- Ensure proper cooling and power efficiency.
- Use high-speed wired internet connections.
- Regularly monitor performance metrics through the dashboard.

[Previous](#)[Social Media](#)

Last updated      days ago

Was this helpful?

---