



## Complete Feature List

Roar engine is the complete, cross-platform, one-stop social game mechanics system designed to give you the power of a modern, custom built social game platform.

Create, grow and monetise your project, and focus on creating games (not building backends).

---

### Incredible game mechanics

Custom built to support a huge range of game mechanics, imagine what you can build with Roar engine.

#### Custom attributes

Create and apply custom attributes to your players, from traditional RPG stats like [strength](#) or [intelligence](#), to alternative, contemporary stats like [fame](#), [mojo](#), [cute](#) or [cool](#).

#### Recharging resources

Enable complex gameplay using resources that increment (or decrement) over custom time periods. Resources like [energy](#) can be 'spent' doing tasks, only to slowly regenerate. Or [mojo](#), that gradually decreases until a player completes an [action](#) (like "play a gig", or "score a goal").

#### XP & levels

Configure as many levels and their XP brackets as needed, as a mechanic to progress players through your game. Reward XP from successful [tasks](#). (XP is an optional feature supported by Roar engine)

## Items (game objects)

Populate your game world with **items**, the highly configurable objects at the heart of Roar engine. From aesthetics (items used for display) and functional items (items used to affect the game world) to tokens (objects that convey information) and boosters (hidden items that affect **stats**), you can create your whole game world with Roar engine.

## Tags for sorting

Organise your Roar engine content using tags, and use those tags to determine how to display or use that content. Create tags like, 'admin', 'noshow', 'hero' or 'magic', and setup custom client handlers to support a vast array of uses.

## Custom item attributes

Enable completely custom functionality in your **items** by creating and defining item attributes. eg. By creating **x\_pos** and **y\_pos** item attributes, you could set and restore the positions of objects in your player's room/town/aquarium/farm/stage. Create any attributes you want, sky is the limit.

## Bonuses

Embed exciting and amazing bonuses into the **items** you create, based on the **attributes**, **resources** and **currencies** you have setup. Create the **Pick of Destiny** to +5 your **music skill**, or the **Sponsored Suit** to +4 your **max charm** but -2 your **credibility**. This powerful feature is at the heart of compelling game mechanics.

## Equip & activate elements

Define custom item types to specify how many can be equipped to a player, like a 'hat'-type with an equip limit of 1, or 'weapons' limit 3, or 'aura' limit 5. Equipped items activate their item bonuses, making for some spectacular game mechanics.

## Appointment mechanics

Bring your players back again and again using the wide range of time-based reward mechanics, including continuous rewards, as well as featuring a 'collect' reward window (come back to get the reward), and even a 'reward limit' window (come back in time, or lose the reward - think "wilting **crops**"). Even limit reward regeneration repeats (eg. a **cake** would only reward once, a **boost pack** might reward 5 times, but a **casino property** might continue to reward indefinitely).

## Tasks, Missions, Quests, Actions

The ultimate game mechanics system, built as a powerful RPG action generator. Create everything from simple, one-step tasks through to intricate, complex chains of actions for your game based on highly configurable requirements, costs and rewards.

## Support complex requirements

Define complex (or simple) combinations of requirements to determine whether a task will succeed, including setting XP minimums, requiring specific numbers of an item

(which can optionally be consumed) or specifying certain levels of any attributes, resources or currency you have created in your game setup.

### Variable 'casino' rewards

Setup variable rewards for tasks, based on weighting probabilities. Maybe success means nothing, but maybe it means a 1% chance to get the [Ring of Awesome](#). Enable random 'drops' from your [tasks](#) as a compelling game mechanic.

### Crafting system

Using the powerful Roar engine [tasks](#) system, construct elaborate crafting systems: eg. convert [stones](#) and [pixie dust](#) into a [magic gem](#) (a prerequisite for your Epic Mission series of [quests](#) of course). Tag your own crafting classes: blacksmith, carpenter, alchemist, enchanter, anything!

### Achievements & Badges

The classic game mechanic to hook players – add smart badges and achievements based on server trackable progress against [tasks](#) (not just a dumb spreadsheet), and save that data for each player (and even use the API to display player info on your website!)

### Leaderboards

Nothing like a little competition among friends. Setup leaderboards to track at the server level any number of attributes and player statistics you like, including XP, gold, custom attributes, and anything else you create in your game!

### Friends

Games are more fun with friends, so develop a community within your game using Roar engine friend functions, enabling players to build up their friends list (and don't forget you can send [items](#) between friends).

## Social and viral channels

Leverage the power and reach of social networks and grow your game using viral channels.

### Facebook integration

Leverage the reach of Facebook, with support for Facebook login and authentication built right into Roar engine.

### Post status to feed

Give your players the ability to boast about their latest [achievements](#) and completed [missions](#) in their Facebook feed, while building exposure for you game amongst their friends.

### Invite friends

Enable your highly engaged players to send invitations to their friends to join in the fun across Facebook and tap into entire new pools of users.

### Gift items

Build the spirit of co-operation and engagement right into your game by enabling players to send [items](#) between each other using Roar engine gifting functionality.

## Virtual Currency, Payment & Shops

Customise, run and manage your virtual economy, fully integrated with real money transaction support.

### Real money currency support

Earn money from virtual currency sales using real money payment gateways. Roar engine supports premium currency purchases through supported payment gateways, as well as an admin API hook for implementing your own custom payment providers.

### Configurable currencies

Create and use multiple, custom currencies in your game economy. Along with a default real money currency, create your own, such as [coins](#) and use these as rewards and to buy and sell items. Get creative and use imaginative currencies like [hero points](#) or [kisses](#), which can be awarded for special actions or achievements and traded for amazing items in the shop.

### Shops & Markets

Setup a market for your virtual goods right from within your game. Create items, set their price based on currencies you create, and make them available in your game shop.

### Sale price in any currency

Items bought from your shop can be sold to players for any currency you create, including premium, real money currency.

### Custom payment gateway

Integrate your own preferred payment processing solution to talk directly to Roar engine, and update players' [premium currency](#) balances. (Of course you can always use our in-built supported payment gateways)

# Server & Backend Management

## Content Management, Servers and Analytics

### RESTful API

Go cross platform with our easy to use, easy to integrate RESTful API. It's as simple as calling a URLs to access your game server eg. <http://roar.io/lpm/game/ping> (returns 'hello' in XML – don't forget to 'view source' if you try it in a browser).

### Custom player data save

Use Roar engine as a network save drive for player data. Simply create [custom attributes](#) for variables you want to track, and save the data to the cloud.

### API Console

Preview all your game server data through the live API Console (demo: <http://roarengine.com/codeconsole/>). Gives you direct, immediate access to the same game data your game client, website or application will tap into.

### Content management dashboard

The one-stop shop for your game management needs, our web based game and content management system provides access to all game data, configuration, content and reports. Access at <http://admin.roarengine.com>.

### Dedicated game server

Unlike most 'social game platforms' who are just mashing a database and some website code together for all their clients, with Roar engine you get a dedicated, blazing fast, specially built, C++ driven Roar engine server for each game you create, along with a custom URL to access that game's API.

### Export game data

Take your data with you when you go. Our simple export to XML functionality ensures that you never have data lock-in, and can migrate your content and setup when needed.

### Admin API access to player data

You asked for it, you got it. An Admin API that allows you to build your own player data management tools to run on your own server or desktop.

### Servers powered by Joyent

Roar engine runs on the powerful [Joyent](#) network infrastructure, specifically designed for high-load, explosive-growth social applications.

### Run an in-house server

For studios who want to manage their own server clusters, we offer a custom license to install the entire Roar engine stack in-house. See <http://roarengine.com/licensing> for more details.

### **Game analytics**

Understanding the usage of your game is an essential part of optimising content, user-flow, growth and ultimately revenue. Our compelling and interactive graphs blow away the competition.

### **Revenue report**

Keep track on your daily earnings, across all your games, or drill into revenue reports on a per game basis. Monitor critical metrics such as ARPU and ARPPU, as you grow your game revenues.

### **Active users (DAU & MAU)**

Users are at the heart of your game, so it's crucial to track the performance and growth of metrics such as registrations, daily and monthly actives, and sticky factor (what ratio of your active players are playing every day).