



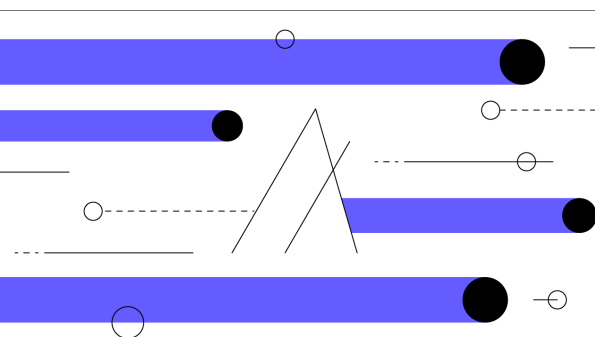
Gradian

gradian.network

Algorand Standard Asset & Decentralised App Ecosystem for the Metaverse

VERSION 1.1

April 21, 2022



Contents

1	Introduction	3
2	Fungible Token Tokenomics	3
3	Feature Roadmap	4
3.1	QR Code Wallet Connect	5
3.2	Asset to In-Game Item Translation	5
3.3	Player-Player Trading DApp	5
3.4	Marketplace DApp	6
3.5	Fixed Price Store DApp	6
3.6	Betting DApps	6
3.7	Farming/Mining DApp	6
3.8	Player vs Player Asset Battles	6
3.9	NFT Property Management	6
4	Game Engines & API Support	6

1 Introduction

Gradian is a decentralised app (DApp) ecosystem for the metaverse built on Algorand, aiming to provide DApps that securely handle common in-game transactions such as player to player trading, farming and player vs player Non-Fungible Token (NFT) battles with verifiable Random Number Generation (RNG) decided outcomes. With API support for all major game engines, developers can readily integrate Gradian into their games. Open-source and well documented APIs which interface with Gradian's DApp ecosystem are planned, with initial support for Unity3D, Unreal Engine and JavaScript.

Gradian is focused on solving the problem of scaleable and frictionless metaverse tokens, and is seeking to address the following issues with existing metaverse tokens:

- High gas fees and low transactions per second (TPS), which inhibit scalability and high transaction volume needed for expansive metaverse experiences. These problems are inherent to layer-2 metaverse tokens built on inefficient blockchains such as ERC20, which currently has high gas fees and a TPS of around 12.
- API support for a limited number of game engines. For example, [Enjin](#) lacks official Unreal Engine support at the time of writing.
- Lack of Smart Contracts specifically to handle common metaverse transactions, such as Player-Player trading, betting and asset battles with verifiable RNG.

By being a layer-2 solution built on Algorand, Gradian inherits Algorand's high efficiency (carbon neutral), high TPS, low transaction fees, and true decentralisation. Which is ideal for high volume frictionless in-game transactions. Gradian's fungible token has deflationary properties that encourage appreciation and scarcity overtime through token burning, which involves sending tokens to a wallet where they're irretrievable.

Ultimately Gradian plans to hold just 5% of tokens, with the remaining 95% in circulation. This will occur gradually over time, as the token is sold to the public and through a limited number of in-game token giveaway events.

2 Fungible Token Tokenomics

The token has deflationary properties through a fixed supply and burning events which will burn 5% of the supply within the first 3 years. Burning is achieved by sending Gradian to an address whose secret key is destroyed to act as a Gradian sink, which can be verified through viewing transaction history.

To facilitate frictionless high frequency micro-transactions, we have set the transaction fees to 1 microalgo which is the lowest fee supported by Algorand. At the time of writing, 1 microalgo is worth fractions of a United States cent. Low fees are important for micro-transactions which are on the scale of a few USD, where fees that are a more than a small ratio will deter players from participating in the transaction.

There's no clawback, freezing or reserve to remove the presence of any central authority as Gradian believes in decentralisation. Please see [ASA Documentation](#) for the definition of these terms. Since it's built on Algorand, it inherits all of Algorand's appealing properties like high throughput thanks to 46000 TPS, carbon neutrality, and high fault tolerance. In the long run, Gradian aims to hold no more than 5% of the tokens, with the remainder in circulation.

Attribute	Value
Denomination	GRAD
Total Issuance	10e9 (FIXED)
Transaction Fee	0.001 ALGO
Burn Rate	5% within the first 3 years
Target Tokens in Circulation	95% within the first 3 years
Blockchain	Algorand
Transactions Per Second	46000
Decentralised	Yes
Carbon Neutral	Yes
Clawback	No
Freezing	No
Reserve	No

Table 1: Tokenomic properties of the Gradian token at a glance.

Achieving our circulation target will happen gradually overtime as the token is purchased by the public, and accelerated through several in-game giveaway events.

3 Feature Roadmap

Gradian's ecosystem of decentralised apps target common in-game use cases, where they can ensure transaction fairness and decentralisation. The transactions Gradian aims to handle with our ecosystem of specialised DApps are those which commonly arise in metaverse experiences, and include:

- Player-Player Trading Transactions
- Fixed Price Store Transactions
- In-Game Item Marketplace Transactions
- Questing Transactions
- Farming and Mining Transactions
- In-Game Betting Transactions
- Player vs Player Battles with an RNG outcome influenced by traits of the participating NFTs

Decentralised apps that handle each of these identified types of transaction will be implemented with Algorand's official PyTeal Smart Contract language. All implementations will be made open-source, giving transparency into how they operate.



The following subsections expand on planned deliverables in our roadmap which may not be immediately clear to the reader.

3.1 QR Code Wallet Connect

Users can connect their wallets to games with our dedicated wallet connect API, which when called presents a QR code on screen which can be scanned to connect a users wallet to the game. Transactions can then be initiated by the game engine on the users behalf, where the user must authorise each transaction with their wallet per Algorand's transaction protocol. Gradian will rely on Algorand's official Wallet Connect SDK which our game engine APIs will interact with.

3.2 Asset to In-Game Item Translation

A translation layer is built into our game engine APIs, which translates users NFTs into in-game assets through a map defined by the game developer. For example, you may own an NFT that represents sunglasses which is translated into 3D sunglasses which your avatar can wear in-game. This gives the developer the ability to populate players inventory with NFTs in players connected wallet, and translate these into in-game items.

3.3 Player-Player Trading DApp

A common mechanic in MMO games is the ability to trade items between players, which is prone to fraud. In most games, this transaction involves trust that the other player will give the offered item and that the game engine itself will execute the trade fairly. Gradian intends to create a DApp to perform this transaction securely, without the need for trust.

3.4 Marketplace DApp

Player collected NFTs that map into the metaverse can be traded on a Marketplace which we plan to create. Since the NFTs are based on the ARC69 NFT standard, they will also be compatible with most existing marketplaces. This is how players can move their assets between the metaverse economy, and the real world economy.

3.5 Fixed Price Store DApp

Games commonly feature in-game stores, where players can purchase items at a set price. In a metaverse experience which is like the real-world, these stores can be player owned and controlled. Gradian will provide a DApp that can handle transactions through these stores, transferring assets between the purchaser and the store owner in a secure way. Imagine purchasing a beer from a metaverse pub, owned by another player for 100 Gradian tokens.

3.6 Betting DApps

Betting is an experience Gradian aims to support with its dedicated suite of betting DApps, that includes verifiable RNG so gamblers know the games aren't rigged in the houses favour. Our focus will be on simple even chance simulations, where assets can be staked by participants and then paid out by the DApp.

3.7 Farming/Mining DApp

Gradian will provide a DApp for farming and mining, where players enter into a pool of participating wallets and receive a payout nominated by the game developer. Participation in the pool can be token based, where the wallet must provide an NFT that acts as a token to participate. The payout can be proportional to NFT provided as the token, for example you might have a diamond pickaxe NFT which has a higher reward to a bronze pickaxe.

3.8 Player vs Player Asset Battles

Gradian will facilitate in-game battles that involve contestant NFTs that are entered into battle by players. The outcome of battles will be decided by a verifiably fair RNG event which is based on properties of the participating NFT assets. The better the traits of a participating asset, the higher the chance they will win the battle. The owner of the winning asset receives both assets. This entire transaction will be handled by our battles DApp.

3.9 NFT Property Management

The ARC69 NFT standard includes a mutable property field which Gradian DApps will change to change their state, reflecting in-game events like taking damage, upgrades and leveling up. Only the manager of the asset has this ability, which the DApps will execute as. This assumes that the game developer is the managing address of the NFT, so that players are prevented from arbitrarily setting the mutable field.

4 Game Engines & API Support

Gradian will provide open source libraries for all major game engines. We intend to include a greater coverage of game engines compared to other metaverse offerings like [Enjin](#), which doesn't support Unreal Engine natively.

- Unity3D package asset, available via the Asset Store
- Unreal Engine plugin, available via the Unreal marketplace
- Generic JavaScript library, available via the Node Package Manager

PyTeal is chosen as the language for implementing smart contracts, as it is recommended by Algorand and has the greatest support for developers. All smart contracts will be visible on the blockchain, and their source code available on Gradian's [Github](#). This will provide transparency into how our smart contracts operate. In the long term, we intend to have [CertiK](#) audit our smart contracts to certify their correctness who are a 3rd party blockchain auditor.

Any REST APIs will be specified with the [OpenAPI](#) specification, which means they will support client generation for over 50 languages out of the box.

