# Technical Documentation

## Table of Contents

## 1 · System Overview

**TriX** is a blockchain-based **incentive and reward distribution system** designed for **PvP (Player vs Player)** gaming.

Its main purpose:

- Provide **trustless match staking** between two players.
- Ensure **transparent and secure** transfer of staked tokens to the winner.
- Enable players to easily purchase in-game tokens using **USDT**.

**Core Functionalities:**

1. **Game Token (GT) Purchase:**
   - Players purchase GT using **USDT only** via the **TokenStore** contract.
2. **Match Staking:**
   - Both players stake the same GT amount before a match starts.
   - Stakes are held in **escrow** until the match result is received.

3. **Winner Payout:**
    - Once the winner is confirmed by the **game server** through the API Gateway, the smart contract automatically transfers the entire staked amount to the winner.

---

# 2 · Economic Model

- **Token Purchase Rate:** 1 GT = **$1.00 USDT**
- **Match Stakes:** Example:
    - Player A and Player B each stake **1 GT**.
    - Total in escrow: **2 GT**.
    - Winner receives **100% of the total escrow** — **2 GT**.

---

# 3 · Architecture Deep Dive

## Core Components & Their Roles

| Component | Purpose |
|---|---|
| **TokenStore** | Accepts USDT for GT purchases. |
| **GameToken (GT)** | ERC-20 token representing the in-game stake currency. Minted only by TokenStore. |
| **PlayGame** | Escrows GT stakes for matches, processes results, and releases winnings. |
| **API Gateway** | Off-chain interface between the game server and blockchain smart contracts. |

---

## Contract Interaction Flow

1. **Token Purchase:**
    - Player sends **USDT** to TokenStore.
    - TokenStore calculates GT amount and mints GT to the player wallet.

2. **Match Play:**
    - Players call PlayGame to deposit their GT stakes.
    - PlayGame records the match ID and player addresses.
    - The match is played off-chain, and the result is submitted by the game server.
3. **Winner Payout:**
    - PlayGame verifies the submitted winner address.
    - Transfers all staked GT to the winner.

---

# 4 · Smart Contract Specifications

## GameToken.sol

- ERC-20 compliant.
- **Only TokenStore** can mint GT.
- Supports transfer, approval, and balance checks.

## TokenStore.sol

- Accepts **USDT** payments.
- Mints corresponding GT amount to buyer.
- Uses a fixed USDT → GT conversion rate.

## PlayGame.sol

- Accepts stakes from two players for a match.
- Stores:
    - Match ID
    - Player addresses
    - Stake amounts
- Accepts winner information from API Gateway.
- Transfers total staked amount to winner.

---

# 5 · End-to-End Functional Flow

1. **Purchase GT:**
    - Player sends USDT to TokenStore.
    - GT is minted and sent to their wallet.
2. **Stake for Match:**
    - Both players stake equal GT amounts in PlayGame.
3. **Match Execution:**
    - Game is played off-chain.
    - Game server sends winner info to API Gateway → forwarded to PlayGame.
4. **Winner Payout:**
    - PlayGame contract sends **all staked GT** to winner.

---

## 6 · Security, Gas, and Operations

**Security:**

- Access control for minting and result submission.
- Re-entrancy protection on fund transfers.
- Emergency pause in all contracts.

**Gas Optimization:**

- Use of immutable and constant variables.
- Compact storage patterns.
- Minimal external calls in payout logic.

---

## 7 · API Reference Summary

| Endpoint | Action | Trigger |
|---|---|---|
| /purchase | Buy GT with USDT | Player wants GT |
| /match/start | Start match, stake GT | Match begins |

| | | |
|---|---|---|
| /match/result | Submit match winner | Match ends |

## 8 · Glossary

- **GT:** Game Token (ERC-20) used for staking.
- **TokenStore:** Contract for buying GT using USDT.
- **Escrow:** Temporarily holding staked tokens in a contract until match result confirmation.
- **API Gateway:** Service layer that allows off-chain game servers to securely interact with blockchain contracts.