# Ustadept (USTA) Whitepaper

Version: v1.0
Date: 2026-02-16 (Europe/Istanbul)
Tagline: “USTA bilir.” *(Turkish; roughly “USTA knows.”)*

## Important Notice
USTA is a **meme/community token**. This document does **not** promise profit, utility, or future value. Nothing here is investment advice. The purpose of this paper is to document the project’s approach, verifiable references, and safety boundaries.

## 1) Summary (TL;DR)
USTA is a **fixed-supply (100,000,000,000)** ERC-20 token on Ethereum. The full supply is minted **once at deployment**; there is **no inflation** and no “owner mint / emissions” lever that can increase supply later.

Ustadept’s stance is intentionally strict: **proof over narrative**.
- Official references are maintained in `docs/OFFICIAL-LINKS.md`.
- Canonical deployments and verification references are maintained in `DEPLOYMENTS.md`.
- Anything outside these canonical records must be treated as **unofficial**.

## 2) Abstract
USTA is a fixed-supply ERC-20 token on Ethereum built for community experiments, tipping, and fun. The supply is minted once at deployment and cannot be increased. The project prioritizes verifiability: official references and canonical deployments are maintained as single sources of truth in the repository.

## 3) Motivation: where most damage happens

In crypto, the biggest attack surface is often not technical—it is **social**:
- fake “official” channels
- vague tokenomics and hidden admin knobs
- “trust me” claims
- unverifiable marketing

Ustadept is designed around the opposite:
> **Shortest path = least surprise + maximum verifiability.**

## 4) What is Ustadept?
Ustadept is an **open-source** community project built around USTA. Core idea:
- build in public
- produce evidence (code, commits, verified deployments)
- be playful, but keep engineering disciplined

USTA’s target use cases:
- tipping
- community activities
- small experiments and on-chain social coordination

## 5) Principles (the Ustadept ethos)
Ustadept prioritizes these axes over inflated promises:
- **Useful:** real benefit, not hype
- **Verifiable:** proof > statement (links, code, on-chain reality)
- **Human:** humor over hype; respect over extraction
- **Open:** open source, open discussion
- **Reliable:** boring/working beats fast/broken

This is not marketing copy; it’s an operating standard.

## 6) The “Official” Boundary (anti-scam rule)
USTA can be easily imitated. Therefore, one rule protects users and integrators:

> If a link is not listed in `docs/OFFICIAL-LINKS.md`, treat it as **unofficial**.

Implication:
- New domains, social accounts, listings, repos, or documents become “official” only via a recorded repo update (PR/commit) to OFFICIAL-LINKS.

Why so strict?

Because user safety is protected by asking **“where is it canonically recorded?”** not “who said it?”

## 7) Token Specification
### 7.1 Network / Standard
- **Network:** Ethereum Mainnet (canonical)
- **Standard:** ERC-20

### 7.2 Parameters
- **Name / Symbol:** USTA / USTA
- **Decimals:** 18
- **Total Supply:** 100,000,000,000 USTA (fixed)
- **Minting:** constructor mints `MAX_SUPPLY` exactly once to the initial recipient

### 7.3 Supply Policy (Hard Constraint)
USTA’s economics are intentionally “boring”:
- **No inflation**
- **No owner mint**
- **No emissions**
- No transfer “tax” mechanics

This “boringness” is a feature: it reduces both technical and social attack surface.

## 8) Contract Design (Implementation Notes)
USTA is implemented on OpenZeppelin ERC-20, with a minimal, no-surprises policy.

### 8.1 Fixed Supply
The contract hard-codes `MAX_SUPPLY`:
- `100_000_000_000 * 10**18`

Constructor behavior:
- reverts if `initialRecipient == address(0)`
- mints `MAX_SUPPLY` to `initialRecipient`

### 8.2 `USTA_BUILD` (build marker)
The contract includes a constant `USTA_BUILD` as a build marker.
Purpose:
- Make runtime bytecode uniquely identifiable to reduce “Similar Match” issues on explorers
- Does **not** change behavior (it simply exposes a public getter)
- Improves the likelihood of full/exact verification on explorers after deployment

This is aimed at **operational verifiability**, not “extra features.”

## 9) Canonical Deployments & Verification (Single Source of Truth)
Canonical facts are maintained in:
- `DEPLOYMENTS.md` → canonical deploy records + verification references
- `docs/OFFICIAL-LINKS.md` → official references + canonical contract pointers

### 9.1 Canonical Mainnet (USTA v1)
Canonical contract (Ethereum Mainnet):
- `0x8D15C25E0fF24256401Fd4DA6d85301084FC3672`

Canonical deployment metadata (recorded in `DEPLOYMENTS.md`) includes:
- deployer / initial recipient
- creation transaction hash
- block number and timestamp
- explorer verification links

### 9.2 Testnet Canonicals (Sepolia)
The repository separates:
- “current/canonical” test deployments
- “deprecated / do not use” deployments
- notes about explorer “Similar Match” and bytecode mismatch

This separation is critical for integrators: it prevents connecting to the wrong contract.

## 10) Deploy Guards & Verification Tooling
The repository includes guardrails against operational mistakes.

### 10.1 Mainnet deploy guard (hard gates)
The mainnet deploy script refuses to run unless:
- `chainId == 1` (Mainnet only)
- `CONFIRM_MAINNET_DEPLOY=YES` is explicitly set
- `USTA_INITIAL_RECIPIENT` is set and is not the zero address

This reduces costly errors such as deploying to the wrong network or minting to an invalid address.

### 10.2 Verification helper (reads `DEPLOYMENTS.md`)
The verification helper:
- takes `network + version`
- extracts canonical contract and constructor args from `DEPLOYMENTS.md`
- runs `hardhat verify` with the proper flags (e.g., production build profile)

### 10.3 CI hygiene (npm audit gate)
A simple but effective policy:

- run `npm audit`
- fail on high/critical findings

This prevents “ignoring dependency risk” by default.

## 11) Distribution Model

The Litepaper states clearly:
- the entire supply was minted to the initial recipient at deployment
- subsequent distribution happens via normal on-chain transfers
- no public sale / ICO / IEO is claimed

This design aims to keep the token focused on community use rather than a “sales narrative.”

## 12) Community & Ecosystem

USTA’s “utility” is community-driven (not a centralized app):
- tipping tools
- bounties (docs, translation, testing, integration)
- mini-games / experiments
- education content and reproducible demos

Principle:
> 100 small verifiable contributions beat 1 big promise.

## 13) Ustadept Foundation (role)

Ustadept Foundation is a community-focused coordination layer:
- open-source development, documentation, educational support
- brand & user-safety defense (scam/impersonation reduction)
- a public “trailmap”: transparent priorities and visible progress

The Foundation’s most important operational duty:
> protect the “single source of truth” model (OFFICIAL-LINKS + DEPLOYMENTS).

## 14) Security Policy & Responsible Reporting

If you believe you found a security issue, report privately:
- Email: `ustadept@ustadept.com`
- Subject: `[SECURITY] USTA Token Vulnerability Report`

Scope typically includes:
- contracts in `contracts/`
- deployment/verification tooling

Disclosure policy note:
- no guaranteed timeline is promised

## 15) Risks (honest section)
- **Market risk:** price can be volatile or go to zero.
- **Smart contract risk:** all software has risk, even if simple.
- **Phishing/impersonation:** treat only OFFICIAL-LINKS as official.
- **Operational risk:** wrong chain/contract errors are possible if canonical records are ignored.

## 16) Non-affiliation
“USTA” here refers to the Turkish word meaning “master/craftsman.”
This project is **not affiliated** with the United States Tennis Association or any other public entity.

## 17) “Trailmap” instead of roadmap
USTA avoids bloated roadmaps. The trailmap is:
- issues/PRs
- releases
- deployment records
- verification evidence

Rule:
> If it’s not done, it’s a plan. If it’s done, it’s a commit.

## Appendix A — Canonical References
- Official links (SSoT): `docs/OFFICIAL-LINKS.md`
- Canonical deployments (SSoT): `DEPLOYMENTS.md`
- Litepaper (short): `docs/LITEPAPER.md`
- Repo: `https://github.com/ustaofficial/ustadept`
- Website: `https://www.ustadept.com/`

## Appendix B — Canonical Contract (short)
- Ethereum Mainnet (USTA v1): `0x8D15C25E0fF24256401Fd4DA6d85301084FC3672`

End of document.