

TOR: JudgePay – AI-Governed Conditional Payment System

Date: 2026-02-05

Version: 1.0.0

Status: Draft / Ready for Review

1. Background & Rationale

In the emerging Agentic Economy, payments between autonomous agents lack a programmable verification layer. Traditional "pay-then-work" creates risk for the payer, while "work-then-pay" creates risk for the worker. **JudgePay** introduces a trustless, algorithmic middle layer where USDC is escrowed and released only upon cryptographic verification of pre-agreed conditions.

2. Objectives

- To provide a **deterministic** escrow mechanism for Agent-to-Agent (A2A) commerce.
- To eliminate "blind trust" by enforcing **conditional execution** via Smart Contracts.
- To enable **automated dispute resolution** based on objective inputs (Hash, Length, Latency).

3. Scope of Work (Functional Requirements)

3.1 Smart Contract Layer (Base Network)

- **Escrow Logic:** The system must accept USDC deposits and lock them in a non-custodial contract.
- **State Management:** The contract must track task states: `Open`, `Submitted`, `Completed`, `Refunded`, `Disputed`.
- **Release Mechanism:** Funds can only move via:
 1. `evaluate(true)` → Release to Worker.
 2. `evaluate(false)` → Refund to Requester.
 3. `deadline_passed` → Emergency Withdraw (Refund).

3.2 AI Decision Layer

- **Role:** The AI acts as a "Digital Oracle" to verify off-chain data (e.g., code quality, image generation).
- **Constraints:**
 - AI **cannot** move funds arbitrarily.
 - AI can only signal `PASS` or `FAIL` to the contract.
 - All decisions must reference a specific `Job Spec` and `Submission Hash`.

3.3 Interface & Schema

- **Job Spec Schema:** JSON-based definition of work (e.g., `min_length`, `forbidden_words`, `required_file_type`).
- **Submission Hash:** Cryptographic proof of work submitted by the worker.

4. Security Requirements

- **Role Separation:** The `Requester`, `Worker`, and `Evaluator` must be distinct addresses.
- **Replay Protection:** Unique Task IDs prevent double-spending or replay attacks.
- **Time-Locks:** All tasks must have a hard deadline to prevent permanent fund locking.

5. Auditability & Deliverables

- **On-Chain Evidence:** Every decision must emit a blockchain event (`TaskCompleted`, `TaskRefunded`) with a transaction hash.
- **Validation Report:** A PDF report summarizing test cases (Success/Failure paths) must be generated for each deployment.

6. Legal Disclaimer & Liability

([REDACTED])

'[REDACTED] AI [REDACTED] JudgePay [REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED] AI'

(Translation: The AI system in JudgePay serves solely as a tool to evaluate work based on pre-agreed scope and conditions. It does not hold legal authority, nor does it act as a court or arbitrator. Fund transfers occur via smart contract mechanisms based on defined conditions, not the discretion of the AI.)

7. Approval

Authorized By: Sydear (System Architect)

Implemented By: Akay (AI Agent)