

# TaskBoard

RaaS Protocol

*Decentralized Robot-as-a-Service on Solana*

**Protocol Whitepaper**

v1.0

2025

# Contents

- 01 Executive Summary
- 02 Introduction
- 03 Problem Statement
- 04 Solution Overview
- 05 Protocol Architecture
- 06 Task Lifecycle
- 07 Robot SDK
- 08 Verification System
- 09 Protocol Economics
- 10 Use Cases
- 11 Technical Specifications
- 12 Governance
- 13 Security
- 14 Glossary
- 15 Conclusion

## 01 Executive Summary

TaskBoard is a decentralized protocol that creates the first trustless marketplace for Robot-as-a-Service (RaaS) on the Solana blockchain. The protocol enables businesses to hire autonomous robots for tasks ranging from warehouse operations to last-mile delivery, with smart contracts handling payment escrow, task verification, and settlement.

### Key Highlights

- \* First decentralized RaaS marketplace built on Solana
- \* Trustless escrow system with cryptographic proof verification
- \* Sub-second finality and near-zero transaction costs
- \* Open SDK supporting all major robotics platforms
- \* Decentralized governance through DAO structure
- \* Projected \$3.2B addressable market by 2027

```
# Quick stats
$ taskboard info --summary
> Protocol: TaskBoard v1.0
> Blockchain: Solana
> TPS Capacity: 65,000+
> Avg Tx Cost: $0.00025
> Settlement Time: ~400ms
```

## 02 Introduction

The robotics industry is undergoing a fundamental transformation. As autonomous systems become more capable and affordable, businesses increasingly seek flexible access to robotic labor without the capital expenditure of purchasing equipment. This shift from ownership to access mirrors the broader trend toward as-a-service models across technology sectors.

However, the current RaaS landscape remains fragmented, inefficient, and trust-dependent. Businesses struggle to find reliable robot operators, negotiate fair pricing, and verify task completion. Robot operators face inconsistent demand, delayed payments, and limited market reach. These inefficiencies represent a significant drag on industry growth.

TaskBoard addresses these challenges by leveraging blockchain technology to create a trustless, global marketplace for robotic services. By combining Solana's high-performance infrastructure with purpose-built smart contracts, TaskBoard enables instant matching, transparent pricing, cryptographic verification, and automatic settlement.

### Vision

We envision a future where any business can access robotic capabilities on demand, and any robot operator can monetize their fleet globally. TaskBoard is the infrastructure layer that makes this vision possible - a decentralized protocol that removes intermediaries, reduces friction, and unlocks the full potential of the robotic economy.

### Mission

To build the most efficient, transparent, and accessible marketplace for robotic services, enabling businesses and robot operators to transact directly with cryptographic guarantees of fairness and completion.

## 03 Problem Statement

### Industry Challenges

The Robot-as-a-Service industry faces several structural challenges that limit growth and adoption. These problems affect all participants in the ecosystem and represent significant opportunities for disruption.

#### Fragmented Access

The RaaS market consists of thousands of small operators with no unified discovery mechanism. Businesses must contact operators individually, compare offerings manually, and negotiate custom contracts. This fragmentation creates high search costs and limits market efficiency.

#### Trust Deficit

Without standardized verification, businesses cannot objectively assess operator reliability or confirm task completion. Disputes are common and costly to resolve. The lack of transparent reputation systems forces businesses to rely on referrals or take significant risks with unknown operators.

#### Payment Friction

Traditional payment rails introduce delays of 30-90 days, creating cash flow challenges for operators. International payments add currency conversion costs and regulatory complexity. Escrow services charge 3-5% fees and still require trust in centralized intermediaries.

#### Idle Capacity

Robot operators struggle to maintain consistent utilization. Industry averages show 40-60% utilization rates, meaning expensive capital sits idle for significant periods. The lack of efficient matching between supply and demand represents billions in unrealized value.

#### High Entry Barriers

Small and medium businesses cannot justify the capital expense of robot purchases. Leasing arrangements require long-term commitments and creditworthiness. The result is that robotic automation remains concentrated among large enterprises.

```
# Market inefficiency analysis
$ taskboard analytics --market-gaps
> Search Cost per Transaction: $450 avg
> Payment Settlement: 47 days avg
> Dispute Rate: 12% of transactions
> Fleet Utilization: 52% avg
> SMB Adoption: 8% penetration
```

## 04 Solution Overview

TaskBoard solves these challenges through a decentralized protocol that creates trustless interactions between task posters and robot operators. The solution combines several key innovations:

### **Unified Marketplace**

A single platform where all robot operators list their capabilities and all businesses post their requirements. Advanced matching algorithms connect supply and demand efficiently, reducing search costs to near zero.

### **Smart Contract Escrow**

Funds are locked in Solana smart contracts when tasks are created. Payment is released automatically upon verified completion, eliminating payment risk for both parties and reducing settlement time from weeks to seconds.

### **Cryptographic Verification**

Task completion is verified through a decentralized network of validators using multiple proof types: telemetry data, visual evidence, oracle feeds, and consensus mechanisms. This creates objective, tamper-proof verification without trusted intermediaries.

### **Reputation System**

On-chain history creates transparent reputation scores for both operators and task posters. Historical performance data enables informed decision-making and rewards reliable participants with better opportunities.

### **Global Accessibility**

Built on Solana, TaskBoard enables instant, low-cost transactions globally. Any business anywhere can access robotic services, and any operator can serve a worldwide market.

```
# Solution impact projection
$ taskboard analytics --solution-impact
> Search Cost Reduction: 95%
> Settlement Time: 400ms
> Dispute Rate Target: <1%
> Utilization Target: 80%+
> Fee Reduction: 70%
```

## 05 Protocol Architecture

TaskBoard consists of four interconnected layers that work together to provide a seamless robot-as-a-service experience. Each layer is designed for modularity, scalability, and security.

### Layer 1: Smart Contract Layer

The foundation of TaskBoard is a set of Solana programs written in the Anchor framework. These programs handle all on-chain logic including task creation, bidding, escrow management, verification, and settlement. Key contracts include:

- \* TaskManager - Handles task lifecycle from creation to completion
- \* EscrowVault - Manages locked funds with conditional release logic
- \* VerificationOracle - Coordinates validator consensus for proofs
- \* ReputationRegistry - Tracks and updates participant reputation scores
- \* GovernanceModule - Manages DAO voting and protocol upgrades

### Layer 2: Verification Layer

A decentralized network of validators confirms task completion through cryptographic proofs. Validators stake tokens to participate and earn rewards for accurate verification. The system uses a Byzantine fault-tolerant consensus mechanism requiring 2/3 agreement for proof acceptance.

### Layer 3: Robot SDK Layer

The open-source TaskBoard SDK enables any autonomous system to participate in the marketplace. The SDK provides libraries for authentication, task discovery, bidding, execution monitoring, and proof submission. Supported platforms include ROS2, Isaac SDK, and custom integrations.

### Layer 4: Application Layer

User-facing applications provide intuitive access to protocol functionality. This includes a web dashboard for task management, mobile apps for operators, CLI tools for developers, and API endpoints for enterprise integrations.

```
# Contract addresses (Devnet)
$ taskboard contracts --list --network devnet
> TaskManager: TBrd...7xKm
> EscrowVault: TBes...9pQn
> VerificationOracle: TBvf...3rWs
> ReputationRegistry: TBrp...2mNk
> GovernanceModule: TBgv...8jLp
> Treasury: TBtr...5yZt
```

## 06 Task Lifecycle

Every task on TaskBoard follows a deterministic lifecycle managed by smart contracts. This ensures transparency, predictability, and trust for all participants. The lifecycle consists of eight distinct stages with clear transition conditions.

```
# Task creation example
$ taskboard task create \
--type warehouse_inventory \
--location 37.7749,-122.4194 \
--budget 50 SOL \
--deadline 2025-XX-XXTXX:XX:XXZ \
--requirements ./task_spec.json
> Task created: TASK-2025-00847
> Status: CREATED
> Escrow locked: 50 SOL
> Bidding opens: NOW
```

### Lifecycle Stages

#### 1. CREATED

Task poster submits requirements, budget, and deadline. System validates parameters and generates unique task ID.

#### 2. FUNDED

Budget amount is transferred to escrow contract. Funds are locked until task completion or cancellation.

#### 3. BIDDING

Qualified robots submit bids including proposed price, estimated completion time, and capability proofs.

#### 4. ASSIGNED

Task poster selects winning bid or system auto-assigns based on ranking algorithm. Robot receives task details.

#### 5. IN\_PROGRESS

Robot executes task while streaming telemetry data. Real-time monitoring available to task poster.

#### 6. PENDING\_VERIFICATION

Robot submits completion proof. Verification request dispatched to validator network.

#### 7. VERIFIED

Validator consensus confirms task completion. Verification proof recorded on-chain.

## **8. SETTLED**

Escrow releases payment to robot operator minus protocol fees. Reputation scores updated.

## 07 Robot SDK

The TaskBoard Robot SDK enables any autonomous system to participate in the marketplace. The SDK is open-source, modular, and designed for easy integration with existing robotics software stacks. It handles all protocol interactions including authentication, task discovery, bidding, execution monitoring, and proof submission.

### Installation

```
# Install TaskBoard SDK
$ pip install taskboard-sdk
# Or for ROS2 integration
$ apt install ros-humble-taskboard
```

### Robot Registration

```
# Register a new robot
$ taskboard robot register \
--type industrial_arm \
--manufacturer Universal_Robots \
--model UR10e \
--capabilities pick_place,welding,assembly \
--location warehouse_zone_a \
--availability 24/7 \
--stake 100 SOL
> Robot registered: ROBOT-IND-0847
> Public Key: 7xKm...9pQn
> Status: ACTIVE
> Reputation: 0 (new)
> Capabilities indexed: 3
```

### Supported Robot Types

#### Industrial Arms

Manufacturing, assembly, welding, painting, quality inspection

#### Drones/UAVs

Aerial survey, infrastructure inspection, agricultural monitoring, delivery

#### Humanoids

Warehouse operations, customer service, facility maintenance

#### Delivery Bots

Last-mile logistics, campus delivery, food delivery, retail restocking

**AGVs**

Material transport, inventory management, pallet handling

**AMRs**

Flexible warehouse navigation, order picking, facility patrol

**Cobots**

Human-collaborative assembly, precision tasks, laboratory automation

## **SDK Features**

### **Auto-Discovery**

Automatically finds and bids on tasks matching robot capabilities

### **Telemetry Streaming**

Real-time status updates to task posters and validators

### **Proof Generation**

Automated collection and formatting of completion evidence

### **Fleet Management**

Coordinate multiple robots under single operator account

### **Offline Queue**

Caches tasks and syncs when connectivity restored

### **Safety Interlocks**

Emergency stop and safe state transitions

```
# SDK configuration
$ taskboard sdk config --show
> Network: mainnet-beta
> RPC Endpoint: https://api.mainnet-beta.solana.com
> WebSocket: wss://api.mainnet-beta.solana.com
> Auto-bid: ENABLED
> Max concurrent tasks: 5
> Telemetry interval: 1000ms
```

## 08 Verification System

Task completion is verified through a decentralized network of validators using multiple proof types. This system ensures objective, tamper-proof verification without relying on trusted intermediaries. Validators stake tokens to participate and face slashing for malicious or negligent behavior.

### Proof Types

#### Telemetry Proof

Sensor data from robot IoT systems including GPS coordinates, timestamps, actuator states, and environmental readings. Data is signed by robot's private key and verified against task requirements.

#### Visual Proof

Before and after images or video with cryptographic hashing to prevent tampering. Computer vision algorithms assist validators in comparing evidence against task specifications.

#### Oracle Proof

Third-party data feeds confirming real-world events. Includes IoT sensor networks, ERP systems, inventory databases, and delivery confirmation systems.

#### Consensus Proof

Multiple independent validators review evidence and vote on completion status. Byzantine fault-tolerant consensus requires 2/3 agreement for proof acceptance or rejection.

```
# Verification submission
$ taskboard verify submit \
--task TASK-2025-00847 \
--proof-type composite \
--telemetry ./sensor_log.json \
--images ./before.jpg..../after.jpg \
--oracle-ref INV-SYS-CONFIRM-847
> Proof submitted: PROOF-9f8e7d6c
> Hash: 0x7a8b...3d4e
> Validators assigned: 5
> Consensus required: 4/5 (80%)
> Est. verification time: 2 minutes
```

### Validator Requirements

- \* Minimum stake: 500 SOL
- \* Uptime requirement: 99.5%
- \* Response time SLA: <30 seconds
- \* Hardware: GPU recommended for visual proof analysis
- \* Slashing: Up to 20% for incorrect verdicts

## 09 Protocol Economics

TaskBoard uses a sustainable fee model that incentivizes all network participants while keeping costs competitive with traditional alternatives. The protocol captures value from successful task completions and redistributes to stakeholders.

### Fee Structure

```
# Fee calculation
$ taskboard fees --calculate 100
> Task Value: 100 SOL
>
> Protocol Fee (2.5%): 2.5 SOL
> - Treasury: 1.5 SOL
> - Token buyback: 0.5 SOL
> - Insurance fund: 0.5 SOL
>
> Validator Rewards (0.5%): 0.5 SOL
> - Split among 5 validators
>
> Network Gas: ~0.00025 SOL
>
> Robot Receives: 97 SOL
> Effective Take Rate: 3%
```

## Staking Requirements

### Robot Operators

Minimum 100 SOL stake as collateral. Higher stakes unlock premium features and priority matching.

### Validators

Minimum 500 SOL to join verification network. Stake weighted toward higher-performing validators.

### Task Posters

No stake required. Optional stake for priority task listing and faster matching.

### slashing

Up to 50% of stake for fraudulent proofs, failed tasks, or validator misbehavior.

## 10 Use Cases

TaskBoard enables a wide range of robotic service applications across industries. The following examples illustrate how different participants can benefit from the protocol.

### **Warehouse Operations**

An e-commerce company needs additional robot capacity during peak season. Through TaskBoard, they post inventory counting tasks specifying warehouse location, SKU requirements, and deadlines. Available warehouse robots bid on tasks, and the company selects based on price and reputation. Tasks are verified through inventory system integration and telemetry data.

### **Last-Mile Delivery**

A restaurant chain wants to offer robot delivery without fleet ownership. They integrate TaskBoard API into their ordering system. When customers select robot delivery, the system automatically posts a task with pickup location, destination, and time window. Nearby delivery robots bid in real-time, and completion is verified through GPS confirmation and customer signature.

### **Infrastructure Inspection**

A utility company requires regular inspection of solar panel installations across multiple states. They schedule recurring inspection tasks on TaskBoard specifying sites, inspection criteria, and reporting requirements. Drone operators throughout the service area claim tasks in their region. Visual proof and automated defect detection verify task completion.

### **Manufacturing Support**

A contract manufacturer experiences equipment failure and needs temporary robot capacity. They post urgent welding tasks with detailed specifications. Industrial robot operators with matching capabilities and nearby locations bid with premium pricing for rush service. Quality verification uses visual inspection and dimensional measurement data.

### **Agricultural Services**

A farm cooperative needs crop monitoring and spraying services across member properties. They post seasonal service contracts on TaskBoard with field boundaries and treatment specifications. Agricultural drone operators bid on regional packages. Telemetry proof confirms coverage area and application rates.

# 11 Technical Specifications

## Blockchain

```
# Network specifications
$ taskboard specs --blockchain
> Chain: Solana
> Consensus: Proof of History + Proof of Stake
> Block Time: 400ms
> TPS: 65,000+
> Finality: ~400ms
> Transaction Cost: ~$0.00025
> Smart Contract: Anchor Framework
> Language: Rust
```

## API Specifications

```
# API details
$ taskboard specs --api
> REST API: /api/v1
> WebSocket: /ws
> GraphQL: /graphql
> Rate Limit: 100 req/min (free), 1000 req/min (pro)
> Authentication: JWT + Wallet Signature
> SDK Languages: Python, JavaScript, Rust, Go
```

## Data Storage

```
# Storage architecture
$ taskboard specs --storage
> On-chain: Task state, escrow, verification proofs
> IPFS: Large proof files, images, telemetry logs
> Arweave: Permanent archival of completed tasks
> Indexer: TheGraph subgraph for queries
```

## 12 Governance

TaskBoard will transition to community governance through a DAO structure. Token holders will vote on protocol upgrades, fee adjustments, and treasury allocation. The governance model is designed to balance efficiency with decentralization.

### Governance Scope

- \* Protocol fee adjustments (within predefined bounds)
- \* Staking requirements and slashing parameters
- \* New robot type and capability approvals
- \* Treasury fund allocation and grants
- \* Smart contract upgrades and migrations
- \* Validator requirements and rewards
- \* Emergency protocol actions

### Proposal Process

```
# Create governance proposal
$ taskboard governance propose \
--title 'Reduce protocol fee to 2%' \
--description proposal.md \
--type parameter_change \
--voting-period 7d \
--quorum 10%
> Proposal created: PROP-0042
> Proposer stake locked: 10,000 TASK
> Discussion period: 3 days
> Voting opens: TBD
> Voting closes: TBD
```

### Voting Power

Voting power is determined by staked TASK tokens. Delegation is supported, allowing token holders to assign their voting power to trusted representatives. Quadratic voting is used for certain proposal types to prevent plutocratic dominance.

## 13 Security

Security is paramount for TaskBoard. The protocol implements multiple layers of protection for funds, data, and network integrity. Our security approach combines preventive measures, monitoring, and incident response capabilities.

### Security Measures

#### Smart Contract Audits

Multiple independent audits by leading security firms (OtterSec, Neodyme, Halborn) before mainnet launch. Ongoing audit program for all upgrades.

#### Formal Verification

Critical contract logic verified using formal methods to prove correctness properties mathematically.

#### Multi-signature Treasury

Protocol treasury controlled by 4-of-7 multisig with geographically distributed keyholders and hardware security modules.

#### Rate Limiting

Protection against spam, denial-of-service attacks, and flash loan exploits through rate limiting and circuit breakers.

#### Stake-based Sybil Resistance

Economic cost to participate prevents fake identities and spam. Minimum stakes scale with privilege level.

#### Bug Bounty Program

Up to \$500,000 rewards for responsible disclosure of critical vulnerabilities. Immunefi-hosted program.

#### Insurance Fund

Protocol-owned insurance fund to cover user losses from smart contract bugs or oracle failures.

## 14 Glossary

### RaaS

Robot-as-a-Service - Business model where robotic capabilities are accessed on-demand rather than purchased

### Task

A unit of work posted on TaskBoard with defined requirements, budget, and deadline

### Escrow

Smart contract holding funds until task completion is verified

### Validator

Network participant who verifies task completion proofs

### Stake

Tokens locked as collateral to participate in the network

### Slashing

Penalty mechanism that reduces stake for protocol violations

### Proof

Evidence submitted to demonstrate task completion

### Telemetry

Sensor and status data streamed from robots during task execution

### DAO

Decentralized Autonomous Organization - Community governance structure

### SDK

Software Development Kit - Tools for integrating robots with TaskBoard

### AMR

Autonomous Mobile Robot - Self-navigating robot for material transport

### AGV

Automated Guided Vehicle - Robot following predetermined paths

### Cobot

Collaborative robot designed to work alongside humans

### Oracle

External data source providing off-chain information to smart contracts

## 15 Conclusion

TaskBoard represents a fundamental advancement in how businesses access robotic services. By combining the efficiency of blockchain technology with the growing capabilities of autonomous systems, we enable a new economy where robots and businesses interact directly, efficiently, and trustlessly.

The Robot-as-a-Service market is projected to reach \$41 billion by 2030, yet current infrastructure remains fragmented and inefficient. TaskBoard provides the missing layer that connects supply and demand, ensures fair compensation, and verifies quality - all without trusted intermediaries.

Our protocol is designed for scale, security, and sustainability. With Solana's high-performance infrastructure, battle-tested smart contracts, and aligned incentives for all participants, TaskBoard is positioned to become the standard infrastructure for the robotic economy.

We invite robot operators, businesses, validators, and developers to join us in building the future of work. Together, we can unlock the full potential of autonomous systems and create unprecedented value for the global economy.

TaskBoard Protocol

*RaaS on Solana*

2025