Aptos Room: The Trustless, Proof-of-Skill Talent Layer for the Web3 Ecosystem

Virtual Connekt Lab

November 2025

Abstract

AptosRoom introduces a trustless talent protocol powered by a skill-gated decentralized Jury System that delivers fair, objective, and manipulation-resistant task evaluations. Instead of relying on subjective ratings, unverifiable CVs, or stake-weighted jurors, AptosRoom selects jurors based solely on proven ability, protected by commit–reveal cryptography and median scoring.

This jury engine is enabled by the Keycard, a mutable, non-transferable Proof-of-Skill identity that evolves with real performance. The Keycard tracks task results, accuracy, and behavior across categories, ensuring that only skilled contributors can judge work and that reputation cannot be faked, bought, or transferred.

Together, the Jury Protocol and Keycard create a merit-based coordination layer where skill not wealth or central authority determines outcome. Built on the high-performance Aptos blockchain, AptosRoom delivers the first practical, scalable implementation of true Proof-of-Skill for Web3.

Executive Summary

AptosRoom exists because Web3 still relies on Web2 trust models—résumés, ratings, admin moderators, and client discretion. Nothing in the ecosystem truly measures skill or guarantees fair evaluation.

AptosRoom solves this by introducing two core primitives:

1. **Keycard** – a non-transferable Proof-of-Skill identity that updates automatically when you complete tasks or serve as a juror. It becomes your verifiable, on-chain résumé.

2. **Jury Protocol** – a decentralized, skill-gated peer review system. Jurors are selected based on proven ability, not tokens or social influence. All voting is commit–reveal encrypted,

[1] *Disclaimer: To maintain a consistent narrative, this paper outlines AptosRoom in the present tense. As such, it includes forward-looking statements based on our current architectural design and expectations. Please note that the final deployed version of the AptosRoom protocol may vary from the concepts detailed in this document.*

scored via median consensus, and validated using a variance model that detects manipulation.

Together, they create a trustless coordination layer where:

- clients get guaranteed fairness,
- contributors get guaranteed payment,
- and the network, not a platform, determines quality.

Everything is fully on-chain with no admin overrides.
Long-term, the Keycard becomes the shared skill identity for the Web3 ecosystem usable by DAOs, dApps, and protocols that need verifiable talent.

1. Introduction

AptosRoom is a decentralized coordination protocol designed to bring verifiable skill and trustless arbitration to the Web3 labor market.

While blockchain technology has revolutionized finance (DeFi), digital labor remains stuck on Web2 rails dependent on centralized intermediaries to manage disputes, hold funds, and curate reputation. AptosRoom eliminates these bottlenecks by replacing human moderation with a code-enforced **Jury Protocol** and replacing static profiles with dynamic **Proof-of-Skill identities (Keycards)**.

AptosRoom replaces these fragmented systems with a fully on-chain coordination protocol built for verifiable skill and trustless arbitration. At its core is the **Jury Protocol**, where qualified contributors not clients or moderators evaluate work through encrypted, multi-juror scoring. Jurors are randomly selected based on their **Keycard**, a non-transferable Proof-of-Skill credential earned solely through verified contributions.

Submissions are reviewed using commit–reveal cryptography, median scoring, and variance analysis to ensure fair, manipulation-resistant outcomes. Payments are governed by **Dual-Key Consensus**: the client (Gold Key) and jury (Silver Key) must both agree before the Vault releases funds.

Every completed task updates the contributor's Keycard, forming a transparent, portable, and tamper-resistant reputation profile. AptosRoom transforms how contributors are evaluated and paid creating a merit-based, decentralized future for Web3 labor.

2. The Problem: Trust, Verification, and Reputation Are Broken

2.1 Problem Statement: The Coordination Crisis in Digital Labor Markets

The current landscape of freelance and bounty platforms both Web2 and early Web3 still relies on outdated trust models. Centralized intermediaries charge high fees, control escrow flows, and settle disputes subjectively. Delays are common, non-payment is widespread, and reputations remain trapped inside closed platforms. Even top contributors must rebuild trust each time they move to a new service, creating friction, fragmentation, and unreliable outcomes for everyone involved.

AptosRoom solves these failures with a blockchain-native, trustless coordination protocol built around verifiable skill and transparent arbitration. At the center is the Jury Protocol a decentralized peer-review system that replaces human moderation and token-weighted courts with multi-juror consensus based purely on demonstrable ability.

Jurors are not volunteers or financial stakers. They are randomly selected from contributors who have earned Keycards AptosRoom's non-transferable Proof-of-Skill identity through successful task completions. These credentials cannot be bought or forged, ensuring that only qualified individuals evaluate work.

Every submission is reviewed by multiple domain-matched jurors who score through commit–reveal encryption to preserve independence and prevent bias. Their accuracy is continuously tracked, and outliers are detected mathematically. Honest jurors are rewarded for alignment with consensus; manipulative or negligent ones are penalized.

Task payments are secured by the Vault, a smart contract that locks funds and releases them only when both client approval ("gold key") and jury consensus ("silver key") are achieved. No intermediaries, no delays, no disputes.

Each completed task updates the contributor's Keycard, building a transparent, portable reputation profile that reflects real skill, reliability, and specialization. Over time, this creates a merit-based foundation for the Web3 labor economy—one where contributors are discovered, evaluated, and rewarded through verifiable work, not trust in centralized platforms.

2.2 The Need for a Trustless Reputation Layer

Reputation defines opportunity in any labor market—but today's systems are fragile and easily manipulated. Web2 platforms lock reputation inside centralized databases, governed by opaque scoring algorithms. Early Web3 attempts aren't much better: token-based reputations skew toward wealth, while NFT badges are static, shallow, and rarely reflect real work.

What Web3 needs is a **reputation layer that can't be gamed, bought, or inflated**—one built from verifiable proof of actual contribution.

AptosRoom introduces this through the **Keycard**, an on-chain Proof-of-Skill identity. It records every task completed, every jury decision, every variance flag, and every improvement in a contributor's skill tier. Because it is generated entirely through protocol-verified actions, the Keycard becomes a trustworthy, earned résumé that reflects competence—not popularity or capital.

This makes talent discovery, hiring, and team formation much more reliable. Any dApp, DAO, or marketplace can query Keycard data to find qualified contributors, filter applicants, or automate skill-based matching.

By turning reputation into a **composable, tamper-proof on-chain primitive**, AptosRoom lays the foundation for a meritocratic Web3 labor economy.

2.3 Why Decentralized Peer Review Is Essential for Web3

Proof-of-Skill addresses identity, but Web3 also needs a trustless way to evaluate work. Relying solely on clients leads to bias and inconsistent results, especially when clients lack the expertise to judge technical tasks. Token-based voting creates its own failures, where influence comes from wealth rather than competence.

A decentralized jury model solves this. By selecting jurors based on verified skill and pseudonymous credentials, evaluation becomes multi-perspective, unbiased, and domain-aware. Commit–reveal scoring prevents influence and copying, while statistical variance detection rewards honest judgment and penalizes manipulation.

This creates an "honesty equilibrium": jurors are incentivized to score thoughtfully because their future eligibility depends on accuracy, not popularity or capital. The system becomes self-correcting.

In AptosRoom, the jury is not only a reviewer but a governance primitive. It decides outcomes, enforces payments, updates Keycards, and ensures that quality is judged by those with the competence to recognize it. This transforms moderation and dispute resolution into deterministic protocol logic — fair, transparent, and resistant to human bias.

3. System Architecture

AptosRoom is engineered as a modular, fully on-chain protocol for trustless task coordination. It transforms every phase of freelance work task creation, submission, evaluation, payment, and reputation into deterministic logic executed on the Aptos blockchain.

3.1 High-Level Workflow

The system operates through the following sequence:

1. **Clients** publish tasks by instantiating a smart contract container called a *Room*.

2. **Contributors** submit entries, with all actions governed by on-chain logic.

3. **Jurors** are randomly selected (filtered by skill) to evaluate submissions using a blinded, commit–reveal process.

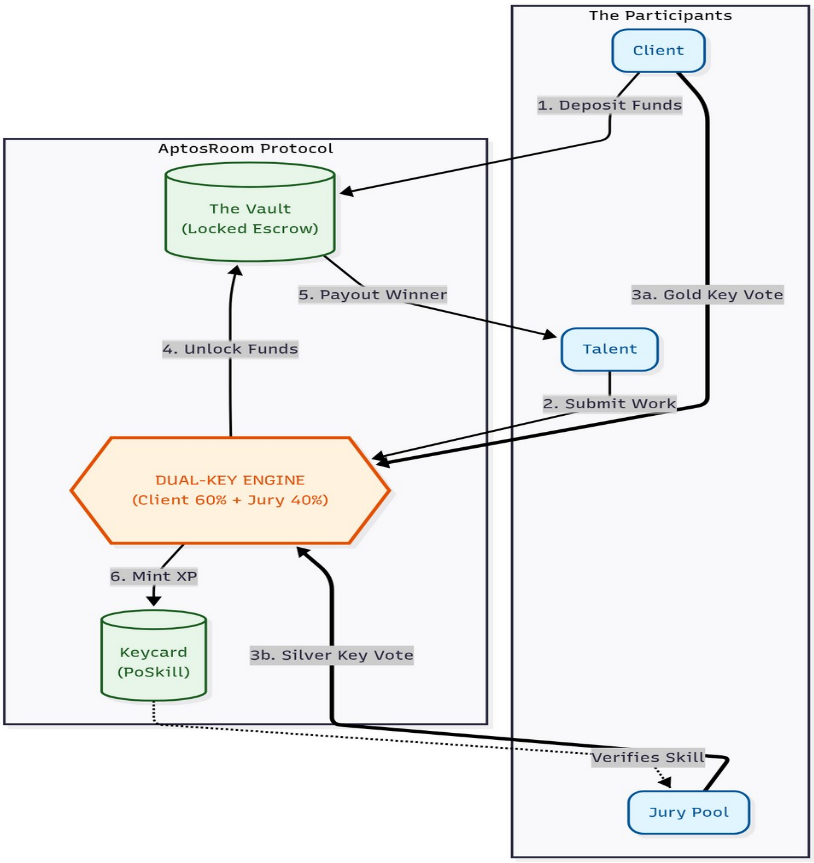
4. **Final scores** are aggregated using a hybrid model: 60% client input, 40% jury input.

5. **Payments and reputation updates** are executed automatically based on ranking outcomes.

3.2 Core Modules

AptosRoom is composed of five lightweight, composable smart-contract modules that together form the trustless coordination layer:

- **Rooms**
  Each task lives inside a Room—a self-contained container that stores metadata, submissions, deadlines, and escrowed funds. Rooms are fully permissionless and ephemeral.
- **Task Engine**
  The execution center of each Room. It manages submissions, computes scores, determines rankings, and triggers payouts based on immutable on-chain logic.
- **Jury Module**
  Controls juror eligibility, reputation gating, random selection, commit–reveal scoring, and median aggregation. It ensures evaluations are fair, independent, and skill-based.
- **Keycard Module**
  Maintains each contributor's Proof-of-Skill identity. It records task completions, juror performance, accuracy, and flags—forming a dynamic on-chain reputation.
- **Treasury Module**
  Oversees escrow, juror rewards, and protocol fees. It guarantees funds cannot be misused and distributes payments automatically once consensus is reached.



High-Level Protocol Architecture

3.3 Smart Contract Design

AptosRoom is powered entirely by Move smart contracts deployed on the Aptos blockchain. These contracts coordinate task creation, submission, scoring, fund release, and on-chain reputation without relying on intermediaries.

- **Task Lifecycle**
  - Clients create Rooms and escrow task funds on-chain.
  - Contributors submit entries as unique Move resources tied to the Room.
  - After deadline, jurors are randomly selected based on Keycard filters.
  - Jurors score entries via a commit–reveal scheme.
  - The contract aggregates scores, selects winners, and triggers payouts.
- **On-Chain PoSkill (Keycards)**
  - Each user has a Keycard resource bound to their Aptos account.
  - Keycards are updated automatically based on verified actions:
    - Task completions
    - Jury participation
    - Scoring accuracy (variance checks)
  - Keycards are tamper-resistant: they can't be copied, deleted, or faked.
- **Security and Verification**
  - Access control is enforced through Move capabilities:
    - Only jurors can score tasks.
    - Only the protocol can update ranks, release funds, or mutate Keycards.
  - Critical functions include formal invariants verified via Move Prover:
    - "Only the top submission can be rewarded"
    - "Reputation updates must follow valid scoring outcomes"
- **No Off-Chain Dependencies**

- All actions—task creation, scoring, reputation, payouts are enforced and executed on-chain.
- No centralized entity can alter results or interfere with workflows.

3.4 Storage and Off-Chain Components

- **User Interaction**: All inputs flow through decentralized frontends; none can modify protocol state off-chain.
- **File Storage**: Large assets (e.g. designs) are stored via IPFS or Arweave, with integrity anchors written on-chain.

The result is a hybrid architecture on-chain for logic and security, off-chain only for large data blobs.

3.5 Security Assumptions and Model

AptosRoom is built under clear security assumptions:

- **Sybil Resistance via Proof-of-Skill**
  Keycard eligibility ensures only reputationed contributors can review or influence outcomes.
- **Commit–Reveal Scoring**
  Scoring is blind and time-locked, preventing manipulation based on peer input.
- **Immutable Audit Trail**
  Every task, juror vote, and payout is recorded on-chain and linked to a specific contributor identity.
- **Escrow Protection**
  Vault logic is non-upgradeable and cannot be externally drained or rerouted. Funds are disbursed only based on ranked outcome logic hardcoded in the Room contract.

4. AptosRoom Jury Model (Decentralize Peer Review)

The AptosRoom Jury Protocol establishes a decentralized, tamper-resistant mechanism for evaluating contributor submissions in an unbiased and technically rigorous manner. In contrast to traditional platforms that rely on centralized moderators or staking-weighted courts, AptosRoom leverages a multi-juror consensus model where reviewers are selected based on demonstrated skill, not wealth or tenure.

The protocol is defined by five foundational principles:

- **Proof-of-Skill Eligibility:** Jurors must possess on-chain credentials relevant to the task domain.

- **Reputation-Gated Random Selection**: Jurors are chosen through an unbiased, verifiable randomness source.
- **Commit–Reveal Scoring:** Cryptographic voting ensures privacy, prevents bias, and enforces accountability.
- **Median-Based Aggregation:** Consensus is derived through mathematically stable median scoring.
- **Dual-Key Evaluation:** Final outcomes blend both client feedback and jury consensus, mitigating unilateral influence.

4.1 Juror Eligibility Criteria

Eligibility to serve as a juror is determined solely by Proof-of-Skill, which is cryptographically recorded via the AptosRoom Keycard system.

Minimum requirements include:

- **Domain-Specific Experience:** Completion of tasks within the same category (e.g., smart contract development, UI/UX design).
- **Credential Thresholds:**
    - ≥ 3 successfully completed tasks
    - ≥ 70% historical accuracy
    - ≤ 3 variance flags
    - Positive aggregate reputation
- **Zero Financial Gatekeeping:** No staking or token ownership is required. Juror selection is fully independent of capital.

4.2 Genesis Jury (Bootstrapping Phase)

During the initial bootstrapping phase, AptosRoom utilizes a curated **Genesis Jury Pool** to establish the first layer of verifiable skill data.

Participants include:

- Verified ecosystem contributors
- Early AptosRoom builders
- Reputable developers selected through off-chain evaluation

These members receive **Genesis Keycard Badges** and are responsible for adjudicating early submissions until sufficient decentralized reputation data is accumulated.

4.3 Juror Selection: Reputation-Gated Randomness

Selection Process:

1. Filter by category-specific skill
2. Filter by reputation (accuracy, history)
3. Use `0x1::random` to select jurors anonymously
4. Notify selected jurors → they must accept

4.4 Consensus: Median-Based Jury Scoring

Final Jury Score = **Median of all juror scores**

Why median?

- Immune to extreme values
- Resistant to one bad actor
- Forces score clustering → honesty equilibrium

This output becomes the **Silver Key** (40%) in Dual-Key Consensus.

4.5 Dual-Key Consensus

Each task is evaluated by:

- **Gold Key** = Client Score (60%)
- **Silver Key** = Jury Score (40%)

Final Score =(Client×0.60) + (Jury×0.40)

Example: Dual-Key Final Score Calculation

To determine a contributor's final task score using the Dual-Key Consensus model:

Step 1: Calculate Client Evaluation (Gold Key)

ScoreClient Score=100×0.60=60

Step 2: Calculate Jury Evaluation (Silver Key)

ScoreJury Score=50×0.40=20

Step 3: Final Consensus Score

ScoreFinal Score=60+20=80

4.6 Variance Detection: Nearest-Neighbor Model

To ensure the reliability and integrity of juror evaluations, AptosRoom implements a statistical validation layer that detects anomalies in scoring behavior.

Traditional models based on mean or median deviation are vulnerable to manipulation by extreme outliers—a flaw known as the *"Poison Pill"* problem. In response, AptosRoom adopts a more robust and context-aware method: **Nearest-Neighbor Variance Detection**.

Motivation

In multi-juror systems, a single malicious vote e.g., a juror scoring 10 when others agree on 90 can distort average-based detection systems, resulting in *honest jurors* being incorrectly flagged. This not only reduces fairness but also weakens trust in the reputation system.

AptosRoom's refinement addresses this issue by shifting the evaluation focus from group-wide deviation to *peer confirmation* logic.

Model Overview

Instead of evaluating how far a juror's score deviates from the *entire group*, AptosRoom asks a simpler and more resilient question:

> **"Is this juror's score reasonably close to at least one other juror?"**

This logic preserves diversity of opinion while identifying statistical isolation.

Mathematical Expression

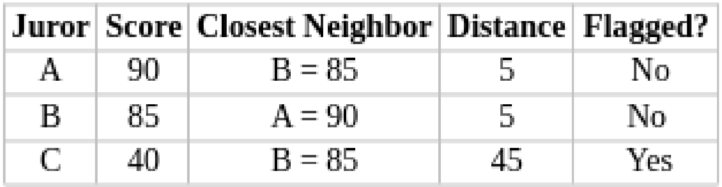
Let (s_j) be the score submitted by juror (j). Then for every juror:

$$\text{Validation Score}_j = \min\left(|s_j - s_i|\right) \quad \forall i \neq j$$

- S_j : The score submitted by **juror j**.
- S_i : The score submitted by any **other juror i** (where i =j).
- **$|S_j—S_i|$**: The **absolute difference** between juror j's score and that of juror i.
- **min**: We take the **smallest distance** between juror j and all other jurors.

- The system computes the minimum absolute difference between a juror's score and all other jurors' scores.
- If this minimum distance exceeds a configurable **Threshold (T)**, the juror is flagged as an outlier.

Example



Juror	Score	Closest Neighbor	Distance	Flagged?
A	90	B = 85	5	No
B	85	A = 90	5	No
C	40	B = 85	45	Yes

Result: Only the statistical outlier (Juror C) is flagged. Jurors A and B, who form a valid opinion cluster, are safe.

Flagging Threshold

The deviation tolerance cap (T) is configurable. A recommended value for general-purpose tasks is:

$$\mathbf{T=15\ \ points\ \ (on\ \ a\ \ 100\text{-}point\ \ scale)}$$

This threshold strikes a balance between strictness and flexibility. Jurors who submit guesses or malicious scores far from consensus will be flagged, while those offering honest yet diverse evaluations are protected.

Enforcement Mechanism

- Jurors flagged **twice consecutively** are temporarily suspended from jury eligibility.
- A slashing mechanism may be applied if flagging is accompanied by other anti-patterns (e.g., missed reveals or poor task records).
- Flags are recorded in the Keycard history and used for future jury selection filtering.

4.7 Juror Accuracy Calculation

AptosRoom measures juror “accuracy” as a long-term reliability metric used to determine future jury eligibility. Accuracy reflects how often a juror’s past evaluations were considered *valid*—meaning they fell within acceptable variance boundaries defined by the protocol’s Nearest-Neighbor model.

Accuracy is computed as the percentage of valid votes across all historical jury tasks:

$$\text{Accuracy}_j = \left(\frac{\text{Valid Jury Decisions}_j}{\text{Total Jury Participations}_j}\right) \times 100$$

Example

- Juror participated in 10 tasks
- Passed variance detection 7 times

$$\text{Accuracy} = \frac{7}{10} \times 100 = 70\%$$

Eligibility Threshold

Jurors must maintain:

Accuracy≥70% to remain eligible for future jury selection.

Jurors falling below the threshold enter a temporary retraining tier, where they must complete contributor tasks to rebuild skill credibility before re-entering the jury pool.

4.8 Commit–Reveal Scoring

AptosRoom ensures fair, bias-resistant voting by using a two-step **commit–reveal cryptographic scheme**, where jurors' scores remain hidden until the voting window closes.

Step 1: Commit Phase

Each juror submits a **commitment hash** on-chain:

$$\mathbf{Commit_j = Hash(s_j \parallel salt_j)}$$

Where:

- (s_j) is the juror's score.
- ($salt_j$) is a unique, secret random string.
- The hash prevents reverse-engineering the vote.
- Only the juror who knows the salt can later unlock the vote.

Step 2: Reveal Phase

After the commit window ends, jurors must publish:

- Their original score (s_j)
- The salt ($salt_j$)

The Move smart contract verifies:

$$\mathbf{Hash(s_j \,||\, salt_j) = Commit_j}$$

The Ghosting Problem

Risks:

- A juror can commit but not reveal → their vote is lost.

- If quorum ((N = 3, 5, 7…)) is not met, scoring fails.
- Worse: A malicious juror could wait, then refuse to reveal if their vote might get penalized **avoiding slashing**.

The Auto-Reveal Keeper Model

To fix these liveness failures, AptosRoom introduces a **non-custodial Keeper system** for reveal automation.

How it Works:

At Commit Time:

- The juror signs **two messages** in one session:

    1. `Tx_Commit` → submitted to Aptos chain.

    2. `Tx_Reveal` → sent **privately** to the **AptosRoom Keeper Node**.

After the Voting Window Ends:

- The Keeper Node automatically submits the Reveal on behalf of the juror.

Security & UX Design:

- All cryptographic keys are client-side; the Keeper stores signed reveal only.
- Jurors **don't need to be online** post-commit.
- Honest users are fully covered by automation.
- Malicious ghosting is **no longer possible**.

5. Keycard: On-Chain Proof of Skill

The **AptosRoom Keycard** is a non-transferable, programmable credential that records verifiable work history and task performance directly on-chain.

It serves as the protocol's **Proof-of-Skill identity layer**, enabling transparent, Sybil-resistant jury selection and portable professional reputation.

5.1 Purpose

Keycards resolve a core limitation in both Web2 and Web3 labor ecosystems: the absence of **trustless, portable merit**. In traditional systems, user reputations are siloed, platform-bound, and often opaque. In staking courts, influence derives from wealth, not skill.

The Keycard flips this paradigm by encoding **decentralized proof-of-work history**, enforceable by smart contracts, not platforms.

5.2 Architecture

Each Keycard is implemented as a **soulbound NFT-like resource** in the Aptos Move VM. Key features:

- **Non-Transferable**: Keycards are permanently bound to the contributor's address; they cannot be traded, sold, or delegated.
- **Modular Metadata**: Skill categories, performance metrics, and flags are represented as structured fields.
- **Mutable via Protocol Logic**: Only AptosRoom contracts can modify scores, update tiers, or append new accomplishments.

5.3 Skill Attribution Model

Each time a contributor completes a verified task, the Keycard is updated with:

- Task Category (e.g., Frontend, Smart Contracts, Branding)
- Performance Score (Dual-Key Final Score)
- XP Points (used for jury tiering)
- Variance Flags

This creates an **auditable trail** of a contributor's domain-specific expertise.

5.4 Keycard and Jury Eligibility

he Jury Module queries the Keycard to validate the criteria established in Section 5.3, ensuring only qualified contributors enter the pool.

5.5 Reputation Portability

Keycards enable contributors to carry their AptosRoom-verified skills beyond the platform:

- **Intra-Aptos Compatibility**: dApps within the Aptos ecosystem can read Keycard credentials directly via on-chain calls.
- **Cross-Chain Proof Export**: Keycard snapshots can be signed and attested for off-chain use by Gitcoin Passport, EAS (Ethereum Attestation Service), or other credential systems.
- **Future Expansion**: Long-term plans include optional bridge layers and zero-knowledge proofs to validate credentials on other chains without compromising Aptos-native control.

5.6 Anti-Manipulation Safeguards

- **Juror Privilege Suspension**: Accumulating multiple variance flags results in temporary suspension from review eligibility.
- **Immutable Audit Log**: All updates (votes, flags, task completions) are recorded on-chain.
- **No Wealth Override**: Holding tokens or NFTs cannot boost Keycard rank or overwrite history.

6. Security Architecture

AptosRoom is designed to deliver trustless, tamper-resistant task execution. Its security model replaces subjective human judgment with verifiable computation and statistically robust mechanisms. This section outlines the key security primitives that make the platform Sybil-resistant, collusion-resistant, and cryptographically enforceable.

6.1 Sybil Resistance via Proof-of-Skill

Traditional staking-based mechanisms are vulnerable to Sybil attacks, where adversaries create multiple accounts to gain disproportionate influence. AptosRoom avoids this by requiring jurors to hold non-transferable, domain-specific Keycard credentials. Juror selection is based on:

- Successful completion of past tasks
- Accuracy in prior judgments
- Stable variance history

This form of Proof-of-Skill acts as a cryptographic filter, ensuring that only competent, real contributors can participate.

6.2 Random Jury Selection

Jurors are selected using Aptos' native randomness module `0x1::random`. The selection process is both reputation-gated and cryptographically unpredictable. This guarantees:

- No pre-coordination between jurors
- No deterministic control by clients or contributors
- Uniform distribution of review responsibilities

This randomization ensures task verification is free from favoritism and insider manipulation.

6.3 Commit–Reveal Voting Integrity

To prevent front-running and copy-cat voting, the protocol relies on the mechanism detailed in Section 4.8. Security is guaranteed by the irreversibility of the SHA3-256 hash and the client-side custody of the salt.

6.4 Tamper-Resistant Finalization

AptosRoom ensures that no human or DAO can override task results. Settlement is governed entirely by:

- **Dual-Key Consensus**: 60% client score + 40% jury score
- **Vault Logic**: Rewards are unlocked and paid out only when valid consensus is reached

7. Protocol Incentives: How Each Role Earns

To maintain long-term sustainability and fairness, AptosRoom defines transparent reward flows for all participants in the ecosystem.

7.1 Contributors (Task Submitters)

Earning Source:

- **Primary:** Task reward escrowed by client
- **Optional bonuses:** If clients assign stretch rewards or bounties

Logic:

- Contributors are paid automatically when:
    - They win the task based on final score
    - Dual-Key Consensus is successfully reached (Client + Jury)
- Payouts are **non-custodial** and released by smart contract logic.

7.2 Clients (Task Creators)

Not direct earners, but benefit via:

- Access to **skilled, vetted labor** without needing to vet manually
- **Lower dispute costs** through trustless, jury-based resolution
- Optional: Long-term staking-based discount model (future)

7.3 Jurors

(As outlined in Section 4)

Earning Source:

- Fixed % of each task's escrow, split among jurors who:

  - Submit valid, revealed scores
  - Pass variance detection

7.4 Keeper Nodes (Auto-Reveal Agents)

Earning Source:

- A small **reveal execution fee** (e.g., a small % of task escrow or fixed micro-payment)

How it works:

- Keeper Nodes store juror-signed reveal payloads and broadcast them on time
- Ensure liveness of commit–reveal protocol

7.5 Protocol Treasury (AptosRoom)

Earning Source:

- Small protocol fee from each task (e.g., 2–5%)
- Used to:
  - Fund ecosystem grants
  - Pay developer contributors
  - Subsidize juror reward pools in early phases

8. References

[1] **Shaikh, M. et al.**, "The Aptos Blockchain: Safe, Scalable, and Upgradeable Web3 Infrastructure," *Aptos Labs Whitepaper*, 2022. [Online]. Available: https://aptos.dev/

[2] **Blackshear, S. et al.**, "Move: A Language With Programmable Resources," *The Diem Association*, 2019. (Foundation for Section 2.3: Resource Safety).

[3] **Weyl, E. G., Ohlhaver, P., & Buterin, V.**, "Decentralized Society: Finding Web3's Soul," *SSRN Electronic Journal*, 2022. (The theoretical basis for Non-Transferable "Keycard" Identity).

[4] **Buterin, V.**, "SchellingCoin: A Minimal-Trust Universal Data Feed," *Ethereum Blog*, 2014. (Game theoretic basis for rewarding consensus in Section 5.6).

[5] **Ast, F. & Lesaege, C.**, "Kleros: The Justice Protocol," *Kleros Whitepaper v2.0*, 2019. (Industry standard for decentralized dispute resolution models).

[6] **Cover, T. & Hart, P.**, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, Vol. 13, Issue 1, 1967. (Mathematical foundation for Section 5.7: Variance Detection).

[7] **Douceur, J. R.**, "The Sybil Attack," *First International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002. (The core problem addressed by Proof-of-Skill in Section 7.1).