

Trustless Work Whitepaper

🕒 Last edited time @September 26, 2025 3:03 PM

Trustless Work Whitepaper

Escrow Infrastructure for the Stablecoin Economy

1. Executive Summary

Escrow is one of the oldest and most trusted mechanisms in commerce. It ensures that funds are only released when specific conditions are met, reducing risk for all parties involved. Yet despite its importance, **escrow remains costly, complex, and inaccessible** to most online businesses and individuals. Traditional escrow providers are custodial, jurisdictionally restricted, and slow to adapt to the pace of global digital transactions.

Trustless Work addresses this gap by providing **escrow as an API**—a plug-and-play infrastructure that enables any platform to integrate **non-custodial, milestone-based escrow payments** in minutes rather than months.

Our system is built on the **Stellar Network**, chosen for its speed, low transaction costs, and compliance-friendly stablecoin ecosystem. By abstracting smart contract complexity into a simple API and SDK, Trustless Work makes programmable trust accessible to platforms, marketplaces, legal-tech solutions, and enterprises worldwide.

Key Characteristics

- **Non-Custodial:** Funds are held in Stellar smart contracts, never by Trustless Work.
- **Role-Based Security:** Each lifecycle step (marking, approval, release) requires signatures from the designated roles, ensuring least-privilege access.
- **Transparency:** Every escrow state change is recorded on-chain, creating a verifiable audit trail.
- **Flexibility:** Supports both single-release and multi-release escrows, adaptable to use cases ranging from high-value real estate transactions to phased software projects.
- **Simple Integration:** A closed-source API with open-source dApps, SDKs, and UI libraries enables rapid adoption without requiring blockchain engineering expertise.

Impact

Trustless Work democratizes escrow by providing infrastructure that was previously available only to large, resource-rich platforms. With this system, businesses can securely onboard stablecoin payment flows, reduce fraud risk, and operate with the same level of trust and transparency as global leaders—without incurring prohibitive costs.

2. Introduction & Context

Escrow is a cornerstone of secure commerce. It provides a neutral mechanism for holding funds while obligations are fulfilled, reducing counterparty risk in transactions that range from high-value real estate transfers to everyday marketplace exchanges. Despite its utility, traditional escrow has remained largely unchanged: custodial, expensive, jurisdiction-bound, and administratively slow.

The global economy is moving in the opposite direction—toward **digital-first, borderless, and instant settlement**. Stablecoins, with daily volumes exceeding hundreds of billions of dollars, are becoming the default medium for international value transfer. Yet while payments have accelerated, the infrastructure for conditional settlement—escrow—has not kept pace.

This gap creates friction:

- **For individuals**, escrow is often unavailable outside of specialized or local providers.
- **For businesses**, building custom escrow logic is prohibitively expensive, requiring months of development, specialized blockchain engineers, and external audits.
- **For platforms**, reliance on custodial third parties introduces regulatory exposure, counterparty risk, and integration bottlenecks.

Why now?

- **Stablecoin adoption** is reaching global scale, with institutions, enterprises, and individuals demanding programmable settlement rails.
- **Smart contract platforms** like Stellar's Soroban make it possible to design lightweight, auditable, and cost-efficient escrow mechanisms at scale.
- **Enterprises and platforms** increasingly need modular, API-first infrastructure to reduce time-to-market and adapt to regulatory requirements.

Trustless Work addresses this precise need. It provides **escrow as an API**: a neutral, programmable, and non-custodial layer of trust that platforms can integrate in minutes. By abstracting complexity while retaining transparency, Trustless Work enables enterprises to build secure, compliant, and cost-efficient workflows around stablecoin settlement.

The result is a system that combines the **reliability of escrow** with the **speed, cost-efficiency, and neutrality of blockchain**, bringing a centuries-old legal instrument into the digital-first economy.

3. System Architecture

Trustless Work provides a modular infrastructure for deploying and managing escrow contracts on the Stellar Network. The architecture is designed to balance **security, flexibility, and ease of integration**, allowing platforms to embed milestone-based payment flows into their products without taking on the risks and costs of custom development.

3.1 Core Components

Smart Escrow Contracts (Soroban)

At the foundation, each escrow is an independent Soroban smart contract. These contracts are immutable once deployed and hold funds in a **non-custodial** manner. The contract enforces role-based permissions for milestone updates, approvals, fund releases, and dispute resolution.

API Layer

The API abstracts the complexity of direct smart contract interactions. It builds, signs, and broadcasts transactions to the Stellar Network. This makes escrow deployment accessible to platforms without requiring in-house smart contract engineers. The API is closed-source to preserve transaction integrity, while the underlying smart contracts remain open-source for transparency and auditability.

SDK & Escrow Blocks

For front-end teams, Trustless Work provides a React SDK and a set of pre-built UI components ("Escrow Blocks"). These libraries allow developers to integrate escrow workflows—funding, milestone approvals, disputes—directly into their existing applications with minimal effort.

dApps (Reference Implementations)

- **Backoffice** – A production-ready interface for creating, managing, and releasing escrows.
- **Viewer** – A read-only explorer for escrow status, roles, and history, ensuring transparency for all parties.
- **Demo** – A lightweight demonstration environment for testing escrow endpoints.

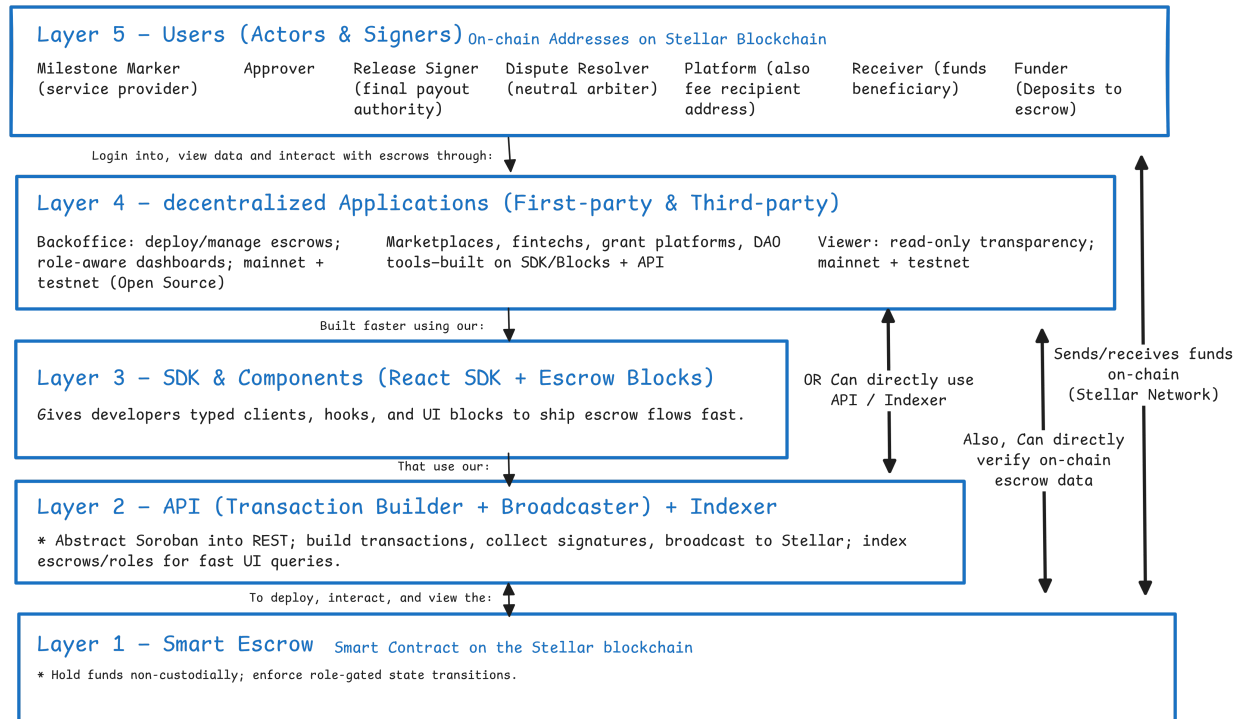
These dApps are open source and serve as reference implementations. Platforms can fork, adapt, or extend them to match their business needs.

Indexer

For usability and search, an Indexer maintains mappings of escrow IDs and role addresses. While all escrow data is available on-chain, the Indexer improves discoverability (e.g., "all escrows where this address is the Approver"). Integrators may run their own indexing service or rely on Trustless Work's hosted instance.

3.2 Design Principles

- **Non-custodial by default:** Neither Trustless Work nor the platform can unilaterally access funds. All fund movements require the signatures of the pre-defined roles.
 - **Role-based logic:** Each action in the escrow lifecycle is restricted to the party designated at creation.
 - **Transparency:** Every event—funding, milestone update, approval, release—is recorded on-chain and can be verified independently.
 - **Modularity:** Escrows can be configured as single-release or multi-release, with flexible role assignments to match the trust model of the transaction.
 - **Extensibility:** Open-source dApps and SDKs allow rapid customization, while the core API ensures consistency and reliability.
-



4. Escrow Lifecycle

Every escrow deployed through Trustless Work follows a well-defined lifecycle. This lifecycle ensures that **funds are securely locked, milestones are transparently tracked, and payments are only released when pre-agreed conditions are met**. Each phase corresponds to on-chain state transitions governed by the escrow contract and enforced through role-based permissions.

4.1 Phases

1. Initiation

- The escrow contract is created on the Stellar network.
- Parameters are defined, including:
 - Total amount (single-release) or milestone structure (multi-release).
 - Roles: milestone marker, approver, release signer, receiver, platform address, and optional dispute resolver.
 - Metadata (engagement ID, description, reference documents).
- The contract awaits funding.

2. Funding

- The payer deposits stablecoins (e.g., USDC, EURC) into the escrow contract.
- The contract balance is updated on-chain.

- If multiple funders are involved, partial contributions are aggregated until the escrow is fully funded.
- Status changes to **Funded**.

3. Milestone Updates

- The **Milestone Marker** submits updates when deliverables are completed.
- Each update emits an event that can be verified on-chain and through the Indexer.
- In multi-release escrows, each milestone has an independent status.

4. Approval

- The **Approver** reviews the milestone status.
- Options:
 - **Approve** → Moves milestone to "Approved."
 - **Dispute** → Flags milestone for resolution.
- Approval is cryptographically signed by the designated address.

5. Release

- The **Release Signer** authorizes fund disbursement.
- Upon release:
 - Funds are transferred to the **Receiver**.
 - The platform fee (default 0.3%) is deducted automatically.
 - Status changes to **Released**.
- In multi-release escrows, each approved milestone can be released independently.

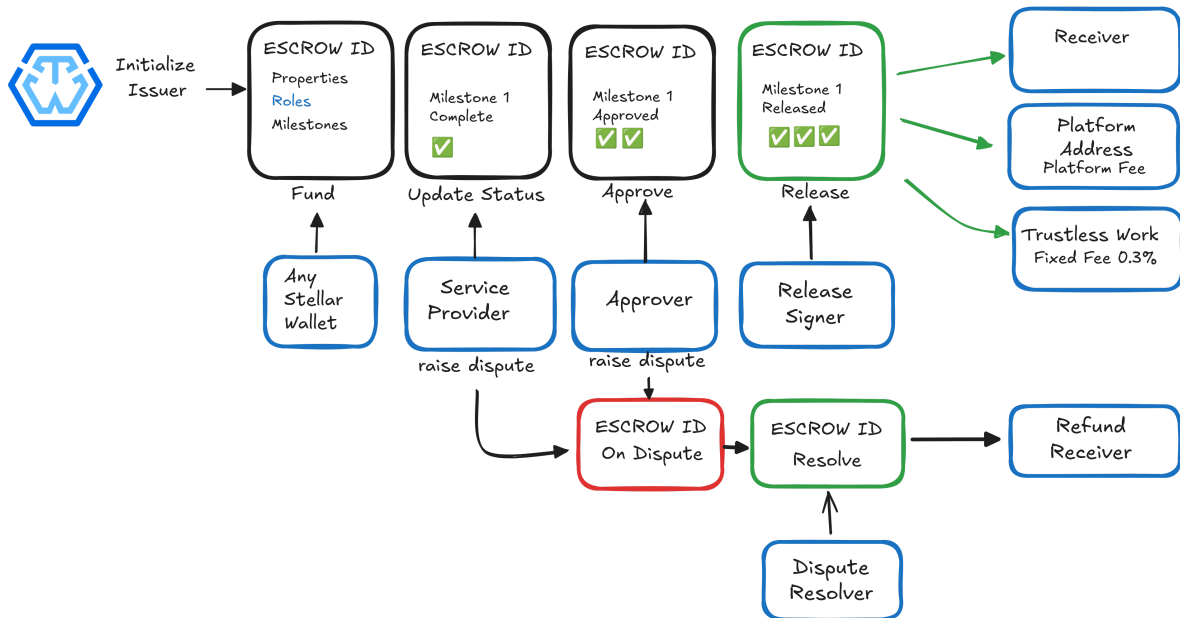
6. Closure / Dispute Resolution

- **Closure**: Once all milestones are released, the contract is finalized.
- **Dispute**: If raised, the **Dispute Resolver** issues a ruling:
 - Approve and release funds.
 - Adjust amounts.
 - Cancel and refund the payer.
- Resolution is final, recorded on-chain, and closes the escrow.

4.2 Key Security Properties

- **Atomicity**: Funds cannot be partially moved outside of defined release functions.
- **Least privilege**: Each role can only perform its specific actions.
- **Transparency**: All lifecycle transitions are logged as on-chain events.
- **Finality**: Once funds are released, the transaction is irreversible.

Stellar Network / Soroban Mainnet



5. Escrow Types

Trustless Work supports two standardized escrow structures. Both operate under the same lifecycle and role logic, but differ in how funds are allocated and released. This flexibility allows platforms and high-value transactions to choose the escrow type that matches their risk profile, commercial structure, and compliance requirements.

5.1 Single-Release Escrow

Definition

A single-release escrow is a contract where the entire escrowed amount is released in one transaction once all milestones are approved and conditions are met.

Typical Use Cases

- Security deposits (e.g., real estate rentals, equipment leasing).
- High-value asset transfers (e.g., boats, domains, intellectual property).
- One-off service contracts with a single deliverable.

Lifecycle Specifics

1. Initiation: One total amount is defined.
2. Funding: Depositor transfers full amount to the escrow contract.

3. Milestone Updates: Typically a single “completion” marker.
4. Approval: Approver confirms fulfillment.
5. Release: Release Signer authorizes transfer of the full balance (minus fee) to the Receiver.
6. Closure: Escrow is finalized.

Security Guarantees

- Funds are indivisible; no partial release is possible.
- Either the full release occurs, or the full amount is refunded in case of dispute or cancellation.
- Prevents manipulation of partial amounts in sensitive, high-value contexts.

5.2 Multi-Release Escrow

Definition

A multi-release escrow divides the total escrowed amount into multiple milestones, each with its own value and release cycle.

Typical Use Cases

- Professional services with phased delivery (e.g., design → build → deployment).
- Crowdfunding campaigns tied to progress milestones.
- Grants, bounties, or recurring project payments.
- Retainers or subscription-based services where progress triggers partial payments.

Lifecycle Specifics

1. Initiation: Escrow is created with a milestone schedule (e.g., 40% design, 40% build, 20% launch).
2. Funding: Depositor funds the full amount or funds each milestone incrementally.
3. Milestone Updates: Marker signals completion of each milestone.
4. Approval: Approver validates each milestone individually.
5. Release: Release Signer triggers payout of the approved milestone only.
6. Closure: Once all milestones are completed and released, the escrow is finalized.

Security Guarantees

- Each milestone operates as a mini-escrow, reducing risk exposure for both parties.
- Funds remain locked for unfinished milestones, ensuring alignment throughout the project.
- Disputes can be raised on a specific milestone without affecting others.

5.3 Comparative Summary

Feature	Single-Release Escrow	Multi-Release Escrow
Structure	One total amount	Multiple predefined milestones

Feature	Single-Release Escrow	Multi-Release Escrow
Release Type	Full payout in one action	Partial payouts per milestone
Common Applications	Security deposits, asset transfers	Service contracts, grants, crowdfunding
Risk Exposure	Binary (all or nothing)	Distributed (phased risk)
Dispute Resolution Scope	Entire amount	Per milestone

Single Release vs Multi Release

Escrow ID

Balance: 0

Properties:

Flags

Trustline

Dispute

Amount

Released

Platform Fee

Resolved

Engagement ID

Roles:

Service Provider

Dispute Resolver

Approver

Platform

Release Signer

Receiver

Milestones:

Milestone 1

Description

✓✓✓

Status

Milestone 2

Description

Status

Even if Milestone 1 is approved, no release is possible until Milestone 2 is also approved.

Escrow ID

Balance: 0

Properties:

Trustline

Platform Fee

Engagement ID

Roles:

Service Provider

Dispute Resolver

Milestone Approver

Platform Address

Release Signer

Receiver

Milestones:

Milestone 1

Description

Amount

Status

✓✓✓

Flags

Dispute

Released

Resolved

Milestone 2

Description

Amount

Status

Dispute

Released

Resolved

Milestone 1 is approved, and the milestone amount can be released. Remaining funds will remain on escrow until Milestone 2 is approved and released.

6. Role Logic

Every Trustless Work escrow is governed by **roles**. These roles are cryptographically bound to public addresses on the Stellar Network and are immutable once the escrow contract is created. By separating responsibilities, Trustless Work enforces a **principle of least privilege**: no single party can control the entire flow of funds without the cooperation of others.

6.1 Role Definitions

1. Milestone Marker

- **Responsibility:** Signals when work or deliverables are completed.
- **Authority:** Can move milestones from "Pending" to "For Review."
- **Limitations:** Cannot approve or release funds; cannot move funds themselves.
- **Security Rationale:** Provides the first attestation of progress, but requires independent validation.

2. Approver

- **Responsibility:** Validates whether the deliverable meets the agreed requirements.
- **Authority:** Approves or disputes milestones.
- **Limitations:** Cannot release funds; cannot modify milestones.
- **Security Rationale:** Independent oversight ensures that Marker claims cannot trigger payouts without review.

3. Release Signer

- **Responsibility:** Executes the transfer of funds once approval has been granted.
- **Authority:** Calls the release function, sending funds to the Receiver.
- **Limitations:** Cannot act without prior approval; cannot alter escrow configuration.
- **Security Rationale:** Ensures funds are only released after both Marker and Approver steps are complete.

4. Receiver

- **Responsibility:** The beneficiary of funds.
- **Authority:** Passive; receives funds once released.
- **Limitations:** Cannot trigger release independently.
- **Security Rationale:** Receiver has no signing authority to prevent self-payouts.

5. Platform Address

- **Responsibility:** Receives the service fee (e.g., 0.3%) automatically upon release.
- **Authority:** None beyond collecting fees.
- **Security Rationale:** Platform is compensated fairly without participating in milestone or approval logic.

6. Dispute Resolver (Optional)

- **Responsibility:** Intervenes in case of disagreement.
- **Authority:** Can override milestone status or reallocate funds according to a resolution outcome.
- **Limitations:** Inactive unless a dispute is formally raised.
- **Security Rationale:** Provides a neutral adjudication path, often assigned to an independent third party, DAO, or arbitration body.

6.2 Role Assignment

- Roles are set **at contract creation** in the JSON configuration.
 - Each role maps to a Stellar public key.
 - Roles may be the same entity (e.g., Approver = Release Signer in small contracts) or distributed among multiple independent addresses.
 - Contracts emit an event confirming role assignments, providing an **auditable record**.
-

6.3 Security Guarantees

- **Separation of Duties:** Marker, Approver, and Release Signer must act sequentially, reducing risk of collusion.
 - **Non-Custodial:** No role has unilateral access to funds. Even the Receiver cannot withdraw without approvals.
 - **Transparency:** Each role action emits an on-chain event (MilestoneMarked, MilestoneApproved, FundsReleased, DisputeRaised).
 - **Flexibility:** Platforms can design different trust models by assigning roles to internal teams, external auditors, or automated agents.
-

6.4 Role Configurations in Practice

- **Peer-to-Peer Transaction:** Marker = Seller, Approver = Buyer, Release Signer = Buyer.
 - **Freelance Contract:** Marker = Freelancer, Approver = Client, Release Signer = Platform.
 - **Real Estate Security Deposit:** Marker = Tenant, Approver = Landlord, Dispute Resolver = Neutral Arbitrator.
 - **Crowdfunding Campaign:** Marker = Project Team, Approver = Backers' Committee, Release Signer = Platform DAO.
-

7. Fees & Economics

The Trustless Work fee structure is designed to be **transparent, predictable, and directly aligned with value creation**. Unlike traditional custodial escrow providers, fees are not hidden in service charges, wire fees, or variable commissions. Instead, a single protocol fee is applied only when funds are released successfully.

7.1 Fee Model

- **Escrow Creation:** Free.
 - No cost to set up or configure an escrow.
 - Encourages experimentation, testing, and multiple escrow creations without financial friction.
- **Escrow Funding:** Free.

- Depositing stablecoins into an escrow contract incurs no protocol fee.
- Users may only incur **standard Stellar network fees**, which are fractions of a cent.
- **Escrow Release: 0.3% Trustless Work fee.**
 - Automatically deducted from the released amount.
- **Platform Fee: configurable % platform fee.**
 - Configurable amount at escrow creation
 - Automatically deducted from the released amount.
 - Collected by the **Platform Address** role configured at escrow creation.

Net amount is transferred to the **Receiver**.

7.2 Fee Enforcement

- Fees are hardcoded into the smart escrow contract.
 - It is executed **at the time of release**, ensuring that the fee can never be bypassed.
 - The fee logic is **non-custodial**: funds flow directly from the escrow contract to the Receiver, TW and Platform Address simultaneously.
 - No external intervention (API, dApp, or Indexer) can alter or avoid this deduction.
-

7.3 Multi-Release Specifics

- In **multi-release escrows**, the 0.3% fee is applied **per milestone release**.
 - Each milestone is treated as an independent transaction for the purpose of fee calculation.
 - Example:
 - Total Escrowed: 1,000 USDC
 - Milestone 1: 400 USDC → 398.8 USDC to Receiver, 1.2 USDC fee
 - Milestone 2: 400 USDC → 398.8 USDC to Receiver, 1.2 USDC fee
 - Milestone 3: 200 USDC → 199.4 USDC to Receiver, 0.6 USDC fee
 - **Total Fee = 3 USDC (0.3% of 1,000)**
-

7.4 Legal & Compliance Considerations

- **Transparency:** The fee is applied automatically on-chain and can be independently verified by auditors, regulators, or counterparties.
- **Predictability:** Fixed at 0.3% (subject to change only by contract upgrades and communicated to all integrators).
- **No Hidden Costs:** Trustless Work does not charge maintenance, custody, or monthly service fees.

- **Jurisdictional Neutrality:** Because Trustless Work never takes custody of funds, the fee does not classify as a financial intermediary spread; it is a service fee for protocol use.
-

8. Security Model

Security is the foundation of Trustless Work. By design, the system ensures that **no central party—including Trustless Work—can access escrowed funds**. All value flows are governed by immutable smart contracts, role-based logic, and Stellar’s consensus mechanism.

8.1 Design-Level Security

Non-Custodial Architecture

- Funds are never deposited into accounts controlled by Trustless Work.
- Escrow contracts hold stablecoins directly on the Stellar Network until release conditions are met.

Role-Based Access Control

- Every action (funding, milestone marking, approval, release) requires a valid signature from the specific role configured at escrow creation.
- Roles cannot overlap unless explicitly assigned (e.g., Approver = Release Signer).
- Enforces **least-privilege design**: no single role can complete the full lifecycle unilaterally.

Immutable Contracts

- Escrow contracts are deployed on Soroban and cannot be modified post-deployment.
- Ensures that terms and conditions agreed at contract creation cannot be tampered with later.

Event Logging

- Each lifecycle transition (funded, approved, released, disputed) emits an on-chain event.
 - Provides an **immutable audit trail** for regulators, auditors, and counterparties.
-

8.2 Threat Model

Trustless Work has undergone multiple rounds of threat modeling and external audits. The following risks are explicitly addressed:

- **Unauthorized Release**
 - Mitigation: Only the Release Signer can call release, and only after an Approver has validated milestones.
- **Double Spending / Duplicate Release**
 - Mitigation: Escrow contracts enforce idempotency—once released, funds cannot be released again.
- **API Compromise**

- Mitigation: The API only facilitates contract interaction. Funds are controlled exclusively by on-chain contracts, not by the API. Integrators may bypass the API entirely and interact directly with Soroban contracts.
 - **Indexer Compromise**
 - Mitigation: Indexer is a convenience service. Even if compromised, it cannot alter escrow states or release funds, as all data and transitions are validated on-chain.
 - **Dispute Resolver Abuse**
 - Mitigation: Dispute Resolver is defined at escrow creation. Their authority is limited to resolving disputes within contract boundaries; they cannot raise a dispute and redirect funds arbitrarily outside the escrow scope.
-

8.3 External Audits

Trustless Work has undergone professional third-party audits:

- **Almanax**
 - Audit 1: April 2025
 - Audit 2: May 2025
 - Audit 3: June 2025
 - Scope: Soroban escrow contract logic, role-based permissions, release conditions, fee enforcement.
- **Runtime Verification**
 - Engagement initiated mid-2025.
 - Scope: Formal verification of escrow logic, invariants, and potential edge cases.
 - Status: Work in progress.

Audit reports are publicly available and form part of Trustless Work's transparency commitment.

8.4 Stellar as the Security Foundation

Trustless Work is built exclusively on the **Stellar Network** because of its:

- **Consensus Protocol (SCP):** Byzantine Fault Tolerant, low-latency, high-throughput.
 - **Transaction Finality:** Typically 5 seconds or less; no probabilistic settlement risk.
 - **Stablecoin Ecosystem:** Circle's USDC and other regulated assets already live on Stellar, providing compliance-grade liquidity.
 - **Cost Efficiency:** Transactions cost fractions of a cent, enabling escrow logic without prohibitive overhead.
 - **Compliance Readiness:** Stellar's design aligns with payment and financial regulatory frameworks, making it suitable for cross-border settlement.
-

9. Compliance & Legal Considerations

Escrows sit at the intersection of finance, law, and technology. For this reason, Trustless Work was designed not just as a technical product but also as an **infrastructure that withstands legal and regulatory scrutiny**. The system enforces fairness, neutrality, and transparency—core principles of escrow—while avoiding the risks associated with custodial services.

9.1 Neutrality and Non-Custodial Design

- **Trustless Work does not hold funds.** All funds are deposited directly into Stellar smart contracts, never into Trustless Work-controlled accounts.
 - By design, **no single entity**—including Trustless Work—can unilaterally move funds. Every release requires role-based authorization, with signatures tied to predefined addresses.
 - This structure distinguishes Trustless Work from traditional escrow providers, who often act as custodians and therefore assume legal liability for safeguarding client funds.
-

9.2 Transparency and Auditability

- **On-Chain Records:** Every action—funding, milestone marking, approval, release, dispute—is recorded as an on-chain transaction or event.
 - **Audit Trail:** Regulators, auditors, or legal parties can independently verify the full lifecycle of an escrow without relying on Trustless Work’s internal systems.
 - **Immutable History:** Once recorded, escrow events cannot be altered or removed, ensuring that disputes can be resolved with objective evidence.
-

9.3 Legal Clarity

Escrows created on Trustless Work provide clear and enforceable logic:

- **Role Assignment:** Responsibilities are codified at contract creation (e.g., who approves, who releases, who resolves disputes).
 - **Objective Conditions:** Fund releases are triggered only by cryptographic signatures, eliminating ambiguity.
 - **Configurable Trust Models:** Depending on the use case, roles can be assigned to independent arbitrators, DAOs, or platform operators, allowing parties to align the escrow with their jurisdictional or contractual requirements.
-

9.4 Jurisdictional Considerations

- **Stellar as Settlement Layer:** Built on Stellar, which is designed to be compliance-friendly and already supports regulated stablecoins such as USDC.
- **Cross-Border Transactions:** Escrows can be created between parties in different jurisdictions, with settlement occurring on a neutral global ledger.

- **Regulatory Neutrality:** Since Trustless Work never takes custody of funds, it does not operate as a financial intermediary under most legal frameworks. Instead, it functions as a **technology service provider**.
-

9.5 Dispute Resolution Framework

- **Flexibility:** Role can be assigned to a neutral third-party arbitrator, a legal firm, or an on-chain governance process (DAO).
 - **Scope of Authority:** Limited to the escrow contract; cannot access unrelated funds or alter the escrow's fundamental parameters.
 - **Legal Parallel:** Mirrors the role of an arbitrator or mediator in traditional agreements but enforces decisions through immutable smart contracts.
-

10. Appendices

The following appendices provide technical and legal reviewers with detailed reference material: API categories, JSON schemas, and glossary definitions. These ensure that Trustless Work escrows are not only conceptually sound but also **implementable, verifiable, and auditable**.

10.1 API Categories

Trustless Work exposes a REST API that abstracts direct Soroban smart contract interactions. The API is divided into logical categories:

- **Deployment Endpoints**
 - Create Single-Release Escrow
 - Create Multi-Release Escrow
- **Funding Endpoints**
 - Deposit stablecoins into escrow
 - Check escrow balance
- **Milestone Management Endpoints**
 - Mark milestone as complete
 - Retrieve milestone status
- **Approval Endpoints**
 - Approve milestone
 - Raise dispute
- **Release Endpoints**
 - Release full escrow (single-release)
 - Release specific milestone (multi-release)

- **Helper Endpoints**

- Get escrow by ID
- Get escrows by role (e.g., all escrows where an address is Approver)
- Retrieve event logs

- **Authentication**

- API key-based authentication, issued per integrator.
- Escrow contract logic remains accessible directly on Soroban for parties who prefer direct interaction.

Reference: [Trustless Work API Documentation](#)

10.2 Example Escrow JSON Schema

Single-Release Escrow (illustrative):

```
{
  "type": "single",
  "asset": { "code": "USDC", "issuer": "G...ISSUER" },
  "amount": "10000",
  "metadata": {
    "engagementId": "ENG-001",
    "description": "Real estate deposit for property ID 12345"
  },
  "roles": {
    "milestoneMarker": "G...MARKER",
    "approver": "G...APPROVER",
    "releaseSigner": "G...SIGNER",
    "receiver": "G...RECEIVER",
    "platform": "G...PLATFORM",
    "disputeResolver": "G...DISPUTE"
  },
  "milestones": [{ "id": "M1", "title": "Final Closing", "status": "PENDING" }]
}
```

Multi-Release Escrow (illustrative):

```
{
  "type": "multi",
  "asset": { "code": "USDC", "issuer": "G...ISSUER" },
  "metadata": {
    "engagementId": "ENG-002",
    "description": "Software project with three deliverables"
  },
  ,
```

```
"roles": {
  "milestoneMarker": "G...MARKER",
  "approver": "G...APPROVER",
  "releaseSigner": "G...SIGNER",
  "receiver": "G...RECEIVER",
  "platform": "G...PLATFORM",
  "disputeResolver": "G...DISPUTE"
},
"milestones": [
  { "id": "M1", "title": "Design phase", "amount": "4000", "status": "PENDING" },
  { "id": "M2", "title": "Build phase", "amount": "4000", "status": "PENDING" },
  { "id": "M3", "title": "Launch phase", "amount": "2000", "status": "PENDING" }
]
}
```

10.3 Glossary of Terms

- **Escrow Contract:** A Soroban smart contract that holds funds until predefined conditions are met.
- **Role:** A specific function within the escrow lifecycle (e.g., Marker, Approver). Each role is tied to a Stellar address.
- **Release Signer:** The role authorized to finalize payouts once conditions are satisfied.
- **Dispute Resolver:** An optional role empowered to resolve conflicts according to contract parameters.
- **Platform Fee:** The 0.3% fee deducted automatically at release and transferred to the Platform Address.
- **Indexer:** A supporting service that improves discoverability of escrows by mapping IDs to role addresses. Does not have authority over funds.
- **Immutable Contract:** Once deployed, the logic of the escrow contract cannot be altered.
- **Event Log:** On-chain record of escrow lifecycle events (funding, approvals, releases).