iTransfr KYT (Know Your Transaction) System

Technical Documentation for Development Team

Version: 1.0 **Last Updated:** December 29, 2025 **Document Type:** Technical Handoff Documentation

---

Table of Contents

---

1. Executive Summary

The KYT (Know Your Transaction) system is a compliance monitoring solution integrated into the iTransfr Admin platform. It provides real-time AML (Anti-Money Laundering) screening and transaction monitoring for cryptocurrency wallets across multiple blockchain networks (Tron, Ethereum, Solana).

Key Capabilities

- **Wallet Screening**: On-demand AML risk assessment via AMLBot API
- **Continuous Monitoring**: Subscription-based monitoring with webhook alerts
- **Deposit Detection**: Automatic detection of incoming deposits to master wallets
- **Unknown Wallet Alerts**: Detection and flagging of deposits from unregistered wallets
- **Client Mapping**: Ability to map unknown wallets to onboarded clients
- **Whitelist Management**: Control over approved outbound transfer destinations

---

2. Purpose & Business Context

Why KYT?

iTransfr operates as a cross-border payment platform handling stablecoin transactions. Regulatory compliance requires:

1. **Screening Inbound Funds**: All client wallets that send funds TO iTransfr master wallets must be screened for AML risk
2. **Identifying Unknown Sources**: Deposits from wallets not linked to onboarded clients must be flagged and investigated
3. **Risk-Based Alerts**: Compliance team needs alerts when risk scores exceed defined thresholds
4. **Audit Trail**: Full history of all screenings and alerts for regulatory reporting

What KYT is NOT

- KYT does **NOT** block transactions automatically (alerts only)
- KYT does **NOT** screen outbound transfers (that's handled by the whitelist system)
- KYT is **NOT** the same as KYC (Know Your Customer) - KYT focuses on transaction/wallet risk, not identity verification

Business Flow

```
┌─────────────────────────────────────────────────────────────────────────┐
│                       INBOUND FLOW (KYT Domain)                         │
│                                                                         │
│   [Client External Wallet] ──deposit──> [iTransfr Master Wallet]        │
│               │                                    │                    │
│               ▼                                    ▼                    │
│      ┌─────────────────┐                 ┌─────────────────┐           │
│      │  AMLBot Screen  │                 │ Deposit Monitor │           │
│      │  (Risk Score)   │                 │ (15-min poll)   │           │
│      └────────┬────────┘                 └────────┬────────┘           │
│               │                                    │                    │
│               ▼                                    ▼                    │
│      ┌───────────────────────────────────────────────────────┐         │
│      │              KYT Alerts Dashboard                      │         │
│      │  - Risk warnings (>=35%)                               │         │
│      │  - Critical alerts (>=40%)                             │         │
│      │  - Unknown deposit alerts                              │         │
│      └───────────────────────────────────────────────────────┘         │
└─────────────────────────────────────────────────────────────────────────┘

┌─────────────────────────────────────────────────────────────────────────┐
│                   OUTBOUND FLOW (Whitelist Domain)                      │
│                                                                         │
│   [iTransfr Master Wallet] ──transfer──> [Exchange Wallet]              │
│               │                                    │                    │
│               ▼                                    ▼                    │
│      ┌─────────────────┐                 ┌─────────────────┐           │
│      │ Whitelist Check │                 │ Trusted Exchange│           │
│      │ (PAXOS, Binance │                 │ (Pre-approved)  │           │
│      │  Kraken, etc.)  │                 │                 │           │
│      └─────────────────┘                 └─────────────────┘           │
└─────────────────────────────────────────────────────────────────────────┘
```

---

3. System Architecture Overview

Technology Stack

Component	Technology
Frontend	React 18 + TypeScript
Backend	Express.js + TypeScript

Database	PostgreSQL + Drizzle ORM
AML Provider	AMLBot (Silenca Tech)
Blockchain APIs	TronScan, ethers.js, Solana RPC

File Structure

```
server/
├── services/
│   ├── amlbot.ts                        # AMLBot API integration
│   ├── deposit-monitor.ts               # Core deposit monitoring logic
│   ├── deposit-monitoring-orchestrator.ts # Starts all network monitors
│   ├── tron-deposit-detector.ts         # Tron network monitoring
│   ├── ethereum-deposit-detector.ts     # Ethereum network monitoring
│   ├── solana-deposit-detector.ts       # Solana network monitoring
│   ├── solana-deposits.ts               # Solana deposit service
│   └── whitelist.ts                     # Outbound whitelist validation
│
├── routes/
│   └── compliance.ts                    # KYT API routes
│
└── routes/index.ts                      # Server initialization

client/src/pages/
├── kyt-wallets.tsx                      # Wallet monitoring UI
└── kyt-alerts.tsx                       # Alerts dashboard UI

shared/
└── schema.ts                            # Database schema definitions
```

---

4. Risk Thresholds & Alert Classification

Risk Score Thresholds

Threshold	Value	Action
Clear	< 15%	No action needed
Low Risk	15% - 34%	Monitor, no alert
Warning	35% - 39%	Create warning alert
Critical	>= 40%	Create critical alert
Blacklisted	Any (flag set)	Create critical alert

Risk Status Labels

```
type RiskStatus = 'clear' | 'low_risk' | 'warning' | 'critical' | 'blacklisted';
```

Alert Types

Alert Type	Description	Trigger
`risk_warning`	Risk score crossed warning threshold	Score >= 35%
`risk_critical`	Risk score crossed critical threshold	Score >= 40%
`blacklisted`	Address is on a blacklist	AMLBot blacklist flag
`unknown_deposit`	Deposit from unmapped wallet	Wallet not linked to client
`risk_change`	Risk score changed (webhook)	AMLBot monitoring webhook

Alert Severities

Severity	Use Case
`low`	Risk change below threshold
`medium`	Unknown deposits with low risk
`high`	Warning threshold exceeded
`critical`	Critical threshold or blacklist

---

5. Core Services & Components

5.1 AMLBot Service (`server/services/amlbot.ts`)

The primary integration with the AMLBot (Silenca Tech) API for wallet risk screening.

Key Functions

```
// Screen a wallet address
checkAddress(address: string, network: string, flow?: 'fast' | 'advanced' |
'instant')
  => { success, riskScore, signals, isBlacklisted, uid, status }

// Re-check a previously screened address
recheckAddress(uid: string)
  => { success, riskScore, signals, isBlacklisted, status }

// Subscribe to continuous monitoring
subscribeToMonitoring(uid: string)
  => { success, error }

// Determine risk severity from score
determineRiskSeverity(riskScore: number, isBlacklisted?: boolean)
  => RiskStatus

// Verify webhook signature (for AMLBot callbacks)
```

```
verifyWebhookSignature(payload: any, check: string, tonce: string)
  => boolean
```

Network Mapping

```
const NETWORK_TO_ASSET = {
  'tron': 'TRX',
  'ethereum': 'ETH',
  'solana': 'SOL',
  'polygon': 'MATIC',
  'stellar': 'XLM',
  'bitcoin': 'BTC',
  'bsc': 'BSC',
  'arbitrum': 'ARB',
  'base': 'BASE',
  'optimism': 'OP',
};
```

API Authentication

AMLBot uses MD5 token-based authentication:

```
// Token generation for address checks
const token = md5(`${address}:${ACCESS_KEY}:${ACCESS_ID}`);

// Token generation for monitoring/webhook operations
const token = md5(`${nonce}:${ACCESS_KEY}:${ACCESS_ID}`);
```

5.2 Deposit Monitor (`server/services/deposit-monitor.ts`)

Core logic for processing detected deposits and determining if alerts are needed.

Key Functions

```
// Process a single detected deposit
processDeposit(deposit: DetectedDeposit, masterWallet: Wallet)
  => DepositCheckResult

// Find client by their external wallet
findClientByExternalWallet(address: string, network: string)
  => { clientId, clientName, walletId } | null

// Map an unknown deposit to a client (called from UI)
mapDepositToClient(alertId: string, clientId: string, userId: string)
  => { success, walletId, error }

// Get all active master wallets for monitoring
getMasterWallets()
  => Wallet[]
```

Deposit Processing Flow

```
                    ┌───────────────────────┐
                    │   Deposit Detected    │
                    └──────────┬────────────┘
                               │
                    ┌──────────▼────────────┐
                    │  Already Processed?   │
                    │    (txHash cache)     │
                    └──────────┬────────────┘
                               │ No
                    ┌──────────▼────────────┐
                    │  Is Whitelisted?      │
                    │  (Exchange address)   │
                    └──────────┬────────────┘
                               │ No
                    ┌──────────▼────────────┐
                    │  Find Client Wallet   │
                    │  (client_external)    │
                    └──────────┬────────────┘
                               │
                 ┌─────────────┴─────────────┐
                 │                           │
     ┌───────────▼────────┐      ┌───────────▼────────┐
     │   Client Found     │      │  Unknown Wallet    │
     └───────────┬────────┘      └───────────┬────────┘
                 │                           │
     ┌───────────▼────────┐      ┌───────────▼────────┐
     │  Screen via        │      │  Screen via        │
     │  AMLBot            │      │  AMLBot            │
     └───────────┬────────┘      └───────────┬────────┘
                 │                           │
     ┌───────────▼────────┐      ┌───────────▼────────┐
     │  Risk >= 35%?      │      │  Create Alert:     │
     │  Create Alert      │      │  unknown_deposit   │
     │  (risk_warning/    │      │  (for mapping)     │
     │   risk_critical)   │      │                    │
     └────────────────────┘      └────────────────────┘
```

5.3 Network-Specific Detectors

Tron Detector (`server/services/tron-deposit-detector.ts`)

- Uses TronScan API to fetch TRC20 token transfers
- Checks USDT and USDC deposits
- Polls every 15 minutes

```
// Key configuration
const TRONSCAN_API = 'https://apilist.tronscanapi.com/api';
const SUPPORTED_TOKENS = {
  USDT: 'TR7NHqjeKQxGTCi8q8ZY4pL8otSzgjLj6t',
  USDC: 'TEkxiTehnzSmSe2XqrBj4w32RUN966rdz8',
};
```

Ethereum Detector (`server/services/ethereum-deposit-detector.ts`)

- Uses ethers.js with Alchemy provider
- Checks USDT and USDC ERC20 transfers
- Uses Transfer event logs

```
// Key configuration
const SUPPORTED_TOKENS = {
  USDT: '0xdAC17F958D2ee523a2206206994597C13D831ec7',
  USDC: '0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48',
};
```

Solana Detector (`server/services/solana-deposit-detector.ts`)

- Wraps existing `solana-deposits.ts` service
- Checks USDC and USDG SPL token transfers
- Uses Solana RPC with signature parsing

---

6. Deposit Monitoring System

How It Works

1. **Startup**: When the server starts, `startKytDepositMonitoring()` is called in `routes/index.ts`
2. **Orchestrator**: The `deposit-monitoring-orchestrator.ts` starts all three network detectors
3. **Polling**: Each detector polls its blockchain every 15 minutes
4. **Detection**: Incoming transfers to master wallets are detected
5. **Processing**: Each deposit is processed through `deposit-monitor.ts`
6. **Alerting**: Alerts are created for unknown wallets or high-risk clients

Configuration

```
// Polling interval (15 minutes)
const DEPOSIT_POLL_INTERVAL_MS = 15 * 60 * 1000;

// Transaction cache to prevent duplicates
const MAX_PROCESSED_CACHE = 10000;
```

Master Wallet Requirement

Each network should have exactly ONE active master wallet:

- The system queries all wallets with `walletType === 'master'` and `status === 'active'`
- Deposits TO these addresses are monitored

Unknown Deposit Workflow

```
┌──────────────────────────────────────────────────────────────────────────┐
│                     UNKNOWN DEPOSIT WORKFLOW                             │
│                                                                          │
│  1. Deposit detected from unknown address                                │
│                         │                                                │
│                         ▼                                                │
│  2. Auto-screen via AMLBot → Get risk score                              │
│                         │                                                │
│                         ▼                                                │
│  3. Create 'unknown_deposit' alert with:                                 │
│     - Source address                                                     │
│     - Risk score & signals                                               │
│     - TX hash, amount, currency                                          │
│     - Master wallet that received it                                     │
│                         │                                                │
│                         ▼                                                │
│  4. Compliance reviews in KYT Alerts UI                                  │
│                         │                                                │
│                         ▼                                                │
│  5. "Map to Client" dialog:                                              │
│     - Select client from dropdown                                        │
│     - Creates client_external wallet record                              │
│     - Links wallet to client via clientWalletLinks                       │
│     - Updates alert status to 'mapped'                                   │
│                         │                                                │
│                         ▼                                                │
│  6. Future deposits from this address are now tracked as known client    │
│                                                                          │
└──────────────────────────────────────────────────────────────────────────┘
```

---

7. Whitelist System (Outbound Transfers)

The whitelist system controls WHERE iTransfr can send funds FROM master wallets.

Purpose

- Only pre-approved exchange addresses can receive outbound transfers
- Prevents accidental or malicious transfers to unauthorized destinations

Pre-Configured Addresses

```
// Example: PAXOS USDG Treasury (Solana)
const PAXOS_USDG_WALLET = "G34fLQYEu8gJ4ABbnFuMPYeGQ4swjxaxReanHj2GrhaR";
```

Whitelist Service (`server/services/whitelist.ts`)

```
// Validate if address is whitelisted
validateAddress(address: string, network: string)
  => WhitelistValidationResult
```

```
// Check if address is whitelisted (boolean)
isWhitelisted(address: string, network: string)
  => boolean

// Require whitelisted (throws error if not)
requireWhitelisted(address: string, network: string)
  => void

// Add system address to whitelist
addSystemAddress(address: string, network: string, label: string, userId: string)
  => void
```

Important Note

The whitelist is **NOT** part of KYT screening. It's a separate control for:

- Outbound transfers only
- Pre-approved exchange partners (PAXOS, Binance, Kraken, Bitget)

---

8. API Reference

Wallet Endpoints

Method	Endpoint	Description
GET	`/api/kyt/wallets`	Get monitored wallets only
GET	`/api/kyt/wallets/all`	Get all wallets

Alert Endpoints

Method	Endpoint	Description
GET	`/api/kyt/alerts`	Get all alerts (optional `?status=` filter)
GET	`/api/kyt/alerts/unread-count`	Get count of unread alerts
PATCH	`/api/kyt/alerts/:id`	Update alert status/notes
POST	`/api/kyt/alerts/:id/map-to-client`	Map unknown deposit to client

Screening Endpoints

Method	Endpoint	Description
POST	`/api/kyt/screen`	Manual address screening
POST	`/api/kyt/monitor`	Enable continuous monitoring (requires 2FA)
DELETE	`/api/kyt/monitor/:walletId`	Disable monitoring
GET	`/api/kyt/screenings`	Get screening history
GET	`/api/kyt/screenings/wallet/:walletId`	Get screenings for specific wallet

Deposit Monitoring Endpoints

Method	Endpoint	Description
GET	`/api/kyt/deposit-monitoring/status`	Get monitoring status
POST	`/api/kyt/deposit-monitoring/check`	Manually trigger deposit check
GET	`/api/kyt/clients-for-mapping`	Get client list for mapping UI

Webhook Endpoint

Method	Endpoint	Description
POST	`/api/kyt/webhook`	Receive AMLBot monitoring webhooks
GET	`/api/kyt/webhook-url`	Get webhook URL for AMLBot config

Request/Response Examples

Screen Address

```
POST /api/kyt/screen
Content-Type: application/json

{
  "address": "TLFmgAiVevcqBSX9DocpmsLYEzWR2BooLe",
  "network": "tron",
  "walletId": "optional-wallet-uuid"
}

# Response
{
  "success": true,
  "status": "success",
  "riskScore": 12.5,
  "signals": {
    "gambling": 0.05,
    "darknet": 0.02
  },
  "isBlacklisted": false,
  "screeningId": "uuid"
}
```

Map to Client

```
POST /api/kyt/alerts/:alertId/map-to-client
Content-Type: application/json

{
  "clientId": "client-uuid"
}

# Response
{
  "success": true,
  "walletId": "new-wallet-uuid",
  "message": "Unknown deposit mapped to client successfully"
}
```

---

9. Database Schema

wallets Table

```
CREATE TABLE wallets (
  id VARCHAR PRIMARY KEY DEFAULT gen_random_uuid(),
  client_id VARCHAR REFERENCES clients(id),
  wallet_type VARCHAR(20) NOT NULL,  -- 'master' | 'client' | 'client_external'
  network VARCHAR(50) NOT NULL,
  address VARCHAR(255) NOT NULL,
  label VARCHAR(100),

  -- AML/KYT fields
  aml_risk_score DECIMAL(5,2),
  aml_status VARCHAR(50) DEFAULT 'not_checked',
  aml_monitoring_enabled BOOLEAN DEFAULT false,
  aml_monitoring_uid VARCHAR(255),
  aml_alert_threshold DECIMAL(5,2) DEFAULT '35',
  aml_critical_threshold DECIMAL(5,2) DEFAULT '47',
  aml_last_checked TIMESTAMP,
  aml_last_signals JSONB,

  created_at TIMESTAMP DEFAULT NOW()
);
```

Wallet Types:

- `master` : iTransfr-owned wallets that receive client deposits
- `client` : Sub-wallets created via Turnkey for clients
- `client_external` : Client's own external wallets used for deposits

aml_alerts Table

```
CREATE TABLE aml_alerts (
  id VARCHAR PRIMARY KEY DEFAULT gen_random_uuid(),
  wallet_id VARCHAR REFERENCES wallets(id),
  address VARCHAR(255) NOT NULL,
  network VARCHAR(50) NOT NULL,
  alert_type VARCHAR(50) NOT NULL,  -- 'risk_warning', 'risk_critical',
'blacklisted', 'unknown_deposit', 'risk_change'
```

```
  previous_risk_score DECIMAL(5,2),
  new_risk_score DECIMAL(5,2),
  risk_signals JSONB,
  severity VARCHAR(20) NOT NULL,

  amlbot_uid VARCHAR(255),
  amlbot_payload JSONB,

  -- Unknown deposit fields
  tx_hash VARCHAR(255),
  deposit_amount DECIMAL(20,8),
  deposit_currency VARCHAR(20),
  master_wallet_id VARCHAR REFERENCES wallets(id),
  client_id VARCHAR REFERENCES clients(id),

  -- Status tracking
  status VARCHAR(50) DEFAULT 'unread',  -- 'unread', 'reviewed', 'resolved',
'dismissed', 'mapped'
  reviewed_by VARCHAR REFERENCES users(id),
  reviewed_at TIMESTAMP,
  notes TEXT,

  created_at TIMESTAMP DEFAULT NOW()
);
```

aml_screenings Table

```
CREATE TABLE aml_screenings (
  id VARCHAR PRIMARY KEY DEFAULT gen_random_uuid(),
  wallet_id VARCHAR REFERENCES wallets(id),
  address VARCHAR(255) NOT NULL,
  network VARCHAR(50) NOT NULL,

  risk_score DECIMAL(5,2),
  risk_signals JSONB,
  is_blacklisted BOOLEAN DEFAULT false,

  amlbot_uid VARCHAR(255),
  amlbot_response JSONB,

  check_type VARCHAR(50) DEFAULT 'manual',  -- 'manual', 'automatic', 'monitoring'
  triggered_by VARCHAR REFERENCES users(id),

  created_at TIMESTAMP DEFAULT NOW()
);
```

client_wallet_links Table

```sql
CREATE TABLE client_wallet_links (
  id VARCHAR PRIMARY KEY DEFAULT gen_random_uuid(),
  client_id VARCHAR REFERENCES clients(id) NOT NULL,
  wallet_id VARCHAR REFERENCES wallets(id) NOT NULL,
  is_primary BOOLEAN DEFAULT false,
  notes TEXT,
  linked_at TIMESTAMP DEFAULT NOW(),
  linked_by VARCHAR REFERENCES users(id)
);
```

---

10. Environment Variables

```
# AMLBot API Credentials
AMLBOT_ACCESS_ID=your_access_id
AMLBOT_ACCESS_KEY=your_access_key

# Blockchain API Keys
ALCHEMY_API_KEY=your_alchemy_key      # For Ethereum
TRON_API_KEY=your_tronscan_key        # For TronScan (optional)
SOLANA_RPC_URL=https://api.mainnet-beta.solana.com
```

---

11. Workflow Diagrams

Complete KYT Workflow

```
┌─────────────────────────────────────────────────────────────────────────────┐
│                            iTransfr KYT SYSTEM                              │
├─────────────────────────────────────────────────────────────────────────────┤
│                                                                             │
│  ┌──────────────────────────────────────────────────────────────────────┐   │
│  │                      1. MANUAL SCREENING                             │   │
│  │                                                                      │   │
│  │  Admin clicks "Screen" → API call to AMLBot → Results stored         │   │
│  │  Risk score displayed → Optionally enable monitoring                 │   │
│  │                                                                      │   │
│  └──────────────────────────────────────────────────────────────────────┘   │
│                                                                             │
│  ┌──────────────────────────────────────────────────────────────────────┐   │
│  │                      2. CONTINUOUS MONITORING                        │   │
│  │                                                                      │   │
│  │  Wallet added to monitoring → AMLBot webhook registered              │   │
│  │  Risk changes → Webhook received → Alert created                     │   │
│  │                                                                      │   │
│  └──────────────────────────────────────────────────────────────────────┘   │
│                                                                             │
│  ┌──────────────────────────────────────────────────────────────────────┐   │
│  │                      3. DEPOSIT MONITORING                           │   │
│  │                                                                          │   │
│  │   Server polls every 15 min → Tron, Ethereum, Solana                     │   │
│  │   Deposit detected → Check if known client → If not, create alert        │   │
│  │   Admin maps to client → Future deposits tracked                         │   │
│  │                                                                          │   │
│  └──────────────────────────────────────────────────────────────────────────┘   │
│                                                                                 │
│  ┌──────────────────────────────────────────────────────────────────────────┐   │
│  │                       4. ALERT MANAGEMENT                                │   │
│  │                                                                          │   │
│  │   Alerts displayed in KYT Alerts dashboard                               │   │
│  │   Filter by status → Review → Add notes → Mark resolved                  │   │
│  │   Unknown deposits → Map to Client → Create wallet record                │   │
│  │                                                                          │   │
│  └──────────────────────────────────────────────────────────────────────────┘   │
│                                                                                 │
└─────────────────────────────────────────────────────────────────────────────────┘
```

Alert State Machine

```
┌───────────────────────────────────────────────────────┐
│                                                       │
│    ┌─────────┐                                        │
│    │ UNREAD  │  ◄────── New alert created             │
│    └────┬────┘                                        │
│         │                                             │
│         │ Review                                      │
│         ▼                                             │
│    ┌──────────┐                                       │
│    │ REVIEWED │                                       │
│    └────┬─────┘                                       │
│         │                                             │
│    ┌────┴─────┬─────────────┐                        │
│    │          │             │                        │
│    ▼          ▼             ▼                        │
│ ┌────────┐ ┌─────────┐ ┌─────────┐                   │
│ │RESOLVED│ │DISMISSED│ │ MAPPED  │ (unknown_deposit)│
│ └────────┘ └─────────┘ └─────────┘                   │
│                                                       │
└───────────────────────────────────────────────────────┘
```

---

12. Pending Tasks & Future Enhancements

Immediate Tasks (Before Go-Live)

1. **Run Database Migration**

   ```
   npm run db:push
   ```

This adds the new columns to `aml_alerts` table (txHash, depositAmount, depositCurrency, masterWalletId, clientId).

2. **Configure AMLBot Webhook**
    - Get the webhook URL from `/api/kyt/webhook-url`
    - Configure it in AMLBot dashboard
    - Enable signature verification when AMLBot supports it (currently commented out)
3. **Add Master Wallets**
    - Ensure each network (Tron, Ethereum, Solana) has exactly one active master wallet
    - These are the wallets that will be monitored for incoming deposits
4. **Test Deposit Detection**
    - Send a small test deposit from an unknown wallet
    - Verify alert is created
    - Test the "Map to Client" functionality

Future Enhancements

1. **Historical Backfill**: Import historical transactions (currently starts from now)
2. **Real-time Webhooks**: Replace polling with WebSocket subscriptions where available
3. **Auto-blocking**: Optionally block deposits from blacklisted addresses
4. **Enhanced Reporting**: Compliance reports for auditors
5. **Multi-master Wallet Support**: Support multiple master wallets per network

---

Document History

Version	Date	Author	Changes
1.0	Dec 29, 2025	Claude Code	Initial documentation

This document is generated for the iTransfr development team. For questions, contact the compliance or engineering teams.