Sentinel AI Data Architecture Specification: Schema Blueprint for Advanced Financial Crime Detection

I. Foundational Architecture and Regulatory Context

The design of the Sentinel AI data infrastructure requires a hybrid architectural approach that simultaneously ensures transactional integrity, adheres to stringent regulatory compliance standards, and provides the necessary structure for high-performance Machine Learning (ML) featurization. This architecture must support both high-volume, real-time transaction processing (OLTP functions) and complex network analysis (Graph Analytics).

A. Sentinel AI Data Objectives: Balancing Transactional Integrity, Regulatory Compliance, and ML Featurization

The data model serves three primary, often competing, objectives. First, it must ensure **Transactional Integrity**, meaning the core tables—specifically Transactions, Accounts, and Customers—must operate with high consistency and atomicity, suitable for a relational or highly consistent analytical data store. This guarantees reliable auditing and reporting of funds movement.

Second, the architecture is fundamentally driven by **Regulatory Imperatives**. Compliance mandates, particularly Know Your Customer (KYC) and the rigorous Know Your Agent (KYA) requirements in high-growth digital banking sectors, necessitate the use of primary national identifiers. The Bank Verification Number (BVN) in Nigeria, for instance, is mandated as the unique ID number linked to every account a customer has across all financial institutions. Similarly, recent regulatory guidelines impose rigorous steps for vetting Agents, requiring the submission of BVN and National Identity Number (NIN) for individuals involved in agent banking. The schema must be structured to enforce these linkages, ensuring that every financial entity (Customer, Beneficiary, or Agent) can be traced back to its verified identity.

Third, and most critically for an advanced platform, the schema must facilitate **ML Featurization**. State-of-the-art fraud detection systems rely on both supervised learning (using labeled fraud history) and unsupervised methods (for anomaly detection). The architecture must support the rapid calculation of velocity features (e.g., transaction frequency over short time windows) and the continuous updating of behavioral baselines. The overall data model must align with common standardized approaches, such as those that organize core banking data into Party (Customer), AccountPartyLink, and Transaction entities, which serve as the foundation for risk detection. This structured collection of data is essential for accurate risk assessment.

B. Core Banking Data Model Requirements and Standardization

The Sentinel AI schema is designed around core banking data elements and incorporates risk investigation data. The structure provides a structured collection of information regarding

customers, their accounts, and their banking activities, which is necessary for the detection of financial risk.
Furthermore, the schema must accommodate comprehensive risk investigation data. The Fraud_Labels Table is the repository for this information, storing the outcome of fraud reviews (Confirmed Fraud, False Positive) which provides the necessary ground truth (the target variable, or Class) required for training and validating supervised machine learning models. Without detailed, resolved investigation data, the ability to train effective models that reduce false positives and adapt to emerging threats is severely limited.

C. Data Lineage and Temporal Integrity: Dual Timestamp Requirement

A crucial architectural requirement for preventing data leakage and overfitting in financial crime models is the disciplined management of temporal data. Models designed to operate in real-time must only use information available to the system *at the time of the transaction initiation*.
This mandate establishes a **Dual Timestamp Requirement** for all transaction events:

1. **Event Initiation Time (event_timestamp):** This captures the moment the public or an agent initiated or sent the transaction to the agent or depositary (Vchr. ChnlInitnDt). This timestamp represents the true point of decision and must be used for all calculations of velocity, frequency, and behavioral features. Calculating a feature based on the event time ensures that the model is only utilizing historical context that truly precedes the event itself.
2. **System Processing Time (processing_timestamp):** This is the date and time the system (the agent, depositary, or Sentinel AI platform) electronically received, recorded, or confirmed the transaction (Vchr. CnfrmDt or validity_start_time).

In environments where systems may experience latency or asynchronous processing, the initiation time and processing time can differ. If a feature calculation were based on the later *processing time*, but the fraud label was applied based on the earlier *event time*, the model might accidentally incorporate knowledge of future events (data leakage). This artificially inflates performance during training, leading to significant degradation in production where decisions must be made in real-time based on the true event context. Adhering to this dual timestamp mechanism guarantees temporal fidelity, a mandatory component of rigorous model risk management.

II. Core Relational Schema Definition: Entity Tables

The following seven tables constitute the core relational backbone of the Sentinel AI system. They are designed for data integrity, auditability, and efficient retrieval of operational data.

A. Transactions Table

The Transactions table is the hub for all financial activity, linking the customer, device, agent, and beneficiary entities. It must capture the fundamental attributes necessary for immediate fraud scoring and compliance reporting.
Transactions Table Schema

Column Name	Data Type	Key/Constraint	Description & ML Relevance
transaction_id	UUID	PK, Indexed	Unique identifier for the transaction event.
event_timestamp	DATETIME (UTC)	Required, Indexed	The time the transaction was initiated. **Crucial for velocity feature calculation.**
processing_timestamp	DATETIME (UTC)	Required	The time the system recorded/confirmed the transaction (System Receipt Time).
sender_account_id	VARCHAR(50)	FK to Accounts	The originating account identifier.
receiver_account_id	VARCHAR(50)	FK to Beneficiaries	The destination account identifier.
amount	DECIMAL(18, 2)	Required	Monetary value of the transaction.
currency	CHAR(3)	Required	Transaction currency (e.g., NGN).
channel_type	VARCHAR(20)	Indexed	E.g., NIP, USSD, POS, ATM, Mobile App. NIP enables real-time funds transfer.
agent_id	VARCHAR(50)	FK to Agents (Nullable)	Required for agent banking transactions, linking to KYA data.
device_id	UUID	FK to Devices (Nullable)	Associated device/session ID. Crucial for Account Takeover (ATO) correlation.
ip_address	VARCHAR(45)	Indexed	IP address used for the transaction initiation, key for graph analysis.
transaction_status	VARCHAR(20)	Required	SUCCESS, FAILED, PENDING, REVERSED.
is_high_risk_flag	BOOLEAN	Indexed	Initial flag set by the real-time rules engine or basic model.
narration	VARCHAR(255)	Text	Transaction description/purpose.
anonymized_v1_v28	JSON/Vector	Optional	Placeholder for high-dimensional, anonymized ML

Column Name	Data Type	Key/Constraint	Description & ML Relevance
			features generated upstream.

B. Customers Table

The Customers table stores party demographics, KYC details, and critical aggregated risk indicators. This entity is central to establishing behavioral baselines.

Customers Table Schema

Column Name	Data Type	Key/Constraint	Description & ML Relevance
customer_id	UUID	PK	Sentinel AI internal customer ID.
bvn	VARCHAR(11)	Unique, Indexed	Bank Verification Number. **Essential for cross-account and network linkage.**
national_id_number	VARCHAR(50)	Indexed	NIN or equivalent (required for KYA/KYC).
kyc_status	VARCHAR(10)	Indexed	KYC TIER 1/2/3, Verified/Unverified.
date_of_birth	DATE	Required	Demographic data; relevant as certain age groups are targeted by fraudsters.
account_open_date	DATE	Required	Defines account tenure; short tenure is a known mule indicator.
risk_score_current	DECIMAL(5, 4)	Indexed	Latest computed entity risk score, used in dynamic risk management.
behavioral_baseline_json	JSONB		Stores key historical metrics (e.g., average transaction value, typical geo-location, frequency) required for anomaly detection.

C. Devices Table

Device intelligence is paramount for combatting Account Takeover (ATO) and SIM swap fraud, which is a significant vector allowing fraudsters to intercept one-time passcodes (OTPs). This table captures dynamic telecom and access data.

Devices Table Schema

Column Name	Data Type	Key/Constraint	Description & ML Relevance
device_id	UUID	PK	Unique device/session identifier.
phone_number	VARCHAR(20)	Unique, Indexed	The number linked to the device.
imei	VARCHAR(15)	Optional, Indexed	Device hardware identifier (if captured).
sim_swap_flag	BOOLEAN	Indexed	Real-time flag from carrier signaling recent SIM change, highly indicative of ATO risk.
last_sim_swap_date	DATETIME (UTC)		Timestamp of the last successful SIM swap event.
carrier_port_count_90d	INTEGER		Velocity feature: Number of times the number was ported to a new carrier in 90 days.
last_login_geo_lat	DECIMAL(9, 6)		Last known successful login latitude. Used to detect unusual distance velocity.
is_roaming_flag	BOOLEAN	Indexed	Flag indicating if the device is connected in an unexpected or risky location.

D. Agents Table

The Agent entity is subject to strict regulatory oversight due to the risk of agent collusion and insider threats. The schema must be designed to enforce the latest KYA and operational guidelines, such as the mandatory use of dedicated agent accounts.

Agents Table Schema

Column Name	Data Type	Key/Constraint	Description & ML Relevance
agent_id	UUID	PK	Sentinel AI internal Agent ID.
principal_id	VARCHAR(50)	FK to Principals	The financial institution the agent is exclusive to. Non-exclusivity is a regulatory violation.
agent_bvn	VARCHAR(11)	Unique, Indexed	BVN of the individual agent or employee (Mandatory KYA data).
agent_nin	VARCHAR(50)	Unique, Indexed	NIN of the agent (Mandatory KYA data).

Column Name	Data Type	Key/Constraint	Description & ML Relevance
kya_status	VARCHAR(20)	Required	Know Your Agent verification status (Verified, Blacklisted).
dedicated_account_id	VARCHAR(50)	Unique, Indexed	Mandatory account used for all agent transactions; transactions conducted outside this account are regulatory violations.
is_exclusive_flag	BOOLEAN	Required	Should be TRUE; violation indicates high regulatory and fraud risk.
fraud_incidents_30d	INTEGER		Aggregated feature for measuring recent collusion risk.
terminal_id	VARCHAR(50)	Required	The physical or virtual terminal identifier used by the agent.

The rationale for including compliance status fields such as kya_status and is_exclusive_flag is that regulatory status is a powerful predictor of operational risk. Non-compliance (e.g., operating outside the dedicated account requirement) is not merely an audit issue but a predictive feature that feeds ML models designed to detect agent collusion and manipulation of cash flows.

E. Beneficiaries Table

This table provides metadata on receiving accounts, supporting both pre-transaction name validation (Name Enquiry) and post-transaction risk scoring. Tracking destination accounts is vital, as certain beneficiaries may be watchlisted or associated with multiple fraudulent senders.
Beneficiaries Table Schema

Column Name	Data Type	Key/Constraint	Description & ML Relevance
beneficiary_account_id	VARCHAR(50)	PK	The receiver account number.
beneficiary_bank_code	VARCHAR(10)	Indexed	Routing number or bank code.
beneficiary_name_hash	VARCHAR(64)	Indexed	Hashed name for matching and privacy protection.
is_watchlisted_flag	BOOLEAN	Indexed	Status flag based on AML sanction screening.
is_new_beneficiary	BOOLEAN		Flag if the account has never transacted with the sender before. A

Column Name	Data Type	Key/Constraint	Description & ML Relevance
			high percentage of transfers to new, unverified beneficiaries is a high-risk feature.

F. Fraud_Labels Table

The Fraud_Labels table is the indispensable source for supervised machine learning training and model performance measurement. It documents the outcome of human investigation and provides the necessary labeled data.

Fraud_Labels Table Schema

Column Name	Data Type	Key/Constraint	Description & ML Relevance
label_id	UUID	PK	Unique identifier for the fraud investigation/label.
transaction_id	UUID	FK to Transactions	Link to the specific transaction investigated.
label_status	VARCHAR(20)	Required, Indexed	Confirmed Fraud (1), False Positive (0), Under Investigation. This is the target variable (Class).
fraud_type	VARCHAR(50)	Indexed	e.g., SIM Swap, Money Mule, Account Takeover, Agent Collusion.
final_resolution_date	DATETIME (UTC)		Date the label was confirmed.
investigator_user_id	VARCHAR(50)		ID of the human analyst who validated the case.
model_prediction_score	DECIMAL(5, 4)		The score the model assigned *before* investigation (used for model tuning and calibration).

G. Graph_Relationships Table

This relational table acts as the high-volume repository for network edge data and shared attributes, serving as the Extract, Transform, Load (ETL) source for the dedicated Graph Database. This bridge stores transactional, device, and identity linkages that are too complex or voluminous to be efficiently queried using relational JOINs for network analysis.

Graph_Relationships Table Schema

Column Name	Data Type	Key/Constraint	Description & ML Relevance
relationship_id	UUID	PK	Unique ID for the relationship event (Edge).
source_node_id	VARCHAR(100)	Indexed	ID of the originating entity (e.g., Customer_ID, Device_ID).
target_node_id	VARCHAR(100)	Indexed	ID of the receiving entity (e.g., Account_ID, Agent_ID).
relationship_type	VARCHAR(50)	Indexed	E.g., TRANSACTED_TO, SHARES_IP, LOGGED_IN_FROM.
timestamp	DATETIME (UTC)	Indexed	Time the relationship was established or the transaction occurred.
property_value_hash	VARCHAR(100)		The shared property (e.g., Hashed Address, Hashed Email) that links the two entities.
weight	DECIMAL(18, 2)		Edge weight (e.g., Transaction Amount or frequency).

III. Feature Engineering Layer: Behavioral and Velocity Metrics

Sentinel AI's ability to detect novel and subtle fraud requires transforming raw transaction data into rich, predictive features that summarize customer behavior and operational context. These features are critical for both anomaly detection and supervised learning.

A. Designing for Real-Time Anomaly Detection and Customer Baselines

Advanced fraud prevention systems must rely on dynamic behavioral analysis to detect deviations from a user’s established norms. Anomaly detection identifies rare occurrences that are suspicious because they differ significantly from expected patterns.
The design incorporates a dedicated mechanism for baseline tracking: the behavioral_baseline_json field in the Customers Table. This field stores dynamically calculated norms, such as the mean transaction size, usual geographical locations, and typical frequency of logins or transfers. When a new transaction arrives, the system compares its attributes against this baseline. An unusual surge (spike) or drop in activity relative to the historical pattern, such as a large transfer initiated from a new city, triggers an anomaly flag. By continuously monitoring these baselines, Sentinel AI can flag subtle deviations that static,

rule-based systems often miss, leading to reported fraud rate reductions of up to 73% using advanced behavioral analytics.

B. Velocity Feature Implementation: Capturing Rapid Fund Movement

Velocity features measure the frequency and value of activities within short, critical time windows (e.g., 5 minutes, 1 hour, 24 hours). These metrics are indispensable for targeting organized financial crime patterns, specifically money mules and attempts to bypass transactional limits.

1. Value and Ratio Features (Money Mule Indicators): Money mules frequently follow recognizable transactional patterns, notably short-term account opening followed by atypical activity. A crucial engineered feature is the ratio of funds withdrawn versus funds deposited over a rolling window. For example, the Withdrawal_Deposit_Ratio_3d is derived by dividing total withdrawals by total deposits over three days. Money mule accounts are characterized by rapidly emptying incoming funds, resulting in a ratio often exceeding 90%. This feature is a primary indicator used in supervised models to target known mule activity. Similarly, the Account_Lifetime_Days (derived from account_open_date in the Customers Table) serves as a contextual feature, as new accounts with high velocity and high withdrawal ratios are acutely risky.

2. State Transition Features (Account Takeover Focus): The schema must seamlessly integrate telecom and device intelligence into transaction scoring. For instance, the **SIM Swap Status** feature, derived from the sim_swap_flag and last_sim_swap_date in the Devices Table, is critical. A high-value transaction initiated shortly after a customer's phone number experienced a SIM swap event is a definitive signal of an Account Takeover attempt, as the fraudster is attempting to monetize their control immediately after intercepting the authentication channel. Other transition features include the calculation of geo-distance velocity, which measures the distance between the current transaction's location and the customer's typical geographic baseline, flagging unusually distant activities.

The implementation relies on pre-calculating these metrics in a dedicated feature store for low-latency retrieval during real-time model scoring. The following table summarizes key derived velocity features.

Derived Velocity Features (Conceptual)

Feature Name	**Source Data**	**Time Window**	**Mule/ATO Pattern Detected**
sender_tx_count_5m	Transactions	5 minutes	Brute-force/rapid transactional velocity to bypass limits.
withdrawal_pct_7d	Transactions	7 days	Rapid cycling of funds (a ratio of withdrawals to deposits exceeding 90% indicates mule activity).
time_since_last_sim_swap	Devices	N/A	High-risk feature measuring latency between SIM event and transaction, critical for Account Takeover.

Feature Name	Source Data	Time Window	Mule/ATO Pattern Detected
new_beneficiary_pct_30d	Beneficiaries/Transactions	30 days	High percentage suggests unusual destination pattern or potential mule coordination.
unique_ip_logins_7d	Devices	7 days	Login velocity from different locations, signaling potential credential sharing or session hijacking.

C. Handling High-Dimensional and Anonymized Features (V1-V28)

The schema design provides flexibility to incorporate complex data types. The anonymized_v1_v28 field in the Transactions Table is designated for features generated by proprietary or deep learning models, where the input variables may be numerous, high-dimensional, and often obfuscated for privacy. This structural allowance ensures that Sentinel AI can ingest vector embeddings and other advanced analytical outputs without requiring constant schema changes to the core relational tables. The inclusion of these features (V1-V28) is often crucial for models like Isolation Forest or One-Class SVM, which are effective in anomaly detection against transaction data.

IV. Graph Architecture: Network Intelligence Specification

Financial crime, particularly money laundering and fraud rings, operates as a coordinated network problem, exploiting relationships between accounts, people, and devices. Traditional relational systems struggle with the performance requirements of deep, iterative relationship queries (JOINs), which is why a dedicated Graph Database layer is mandatory for Sentinel AI. This architecture is essential for detecting patterns like circular movements of money, structured deposits, and shared attributes that reveal the true extent of criminal operations.

A. Graph Data Model (GDM) Proposal: Nodes and Edges

The Graph Data Model (GDM) defines the entities (Nodes) and their connections (Edges) within the network structure, sourced from the core relational tables.

Graph Data Model Components: Nodes

Node Label	Critical Properties	Source Tables	Purpose in Sentinel AI
Customer	customer_id, bvn, risk_score, kyc_status	Customers	Core party entity, linking identity to activity.
Account	account_id, open_date,	Transactions,	Financial instrument

Node Label	Critical Properties	Source Tables	Purpose in Sentinel AI
	currency	Customers	used in transactions.
Device	device_id, ip_address, sim_swap_flag	Devices	Tracks access points, critical for linking multiple users to single hardware.
Agent	agent_id, principal_id, dedicated_account_id	Agents	Intermediaries whose activity must be monitored for collusion.
Location	geo_hash, city, country	Transactions, Devices	Used to group activity by shared geography (e.g., ATM geolocation analysis).

Graph Data Model Components: Edges

Edge Label	Source Node	Target Node	Critical Edge Properties
TRANSACTED_TO	Account	Account	transaction_id, amount, timestamp, channel_type (The flow of funds).
SHARED_ATTRIBUTE	Customer/Account	Customer/Account	attribute_type (e.g., IP, Address, Phone), last_seen (Exposes hidden intermediaries).
LOGGED_IN_FROM	Customer	Device	login_time, success_status (Tracks device usage history).
WORKS_FOR	Agent	Principal	kya_status, is_exclusive_flag (Organizational compliance linkage).

B. Graph Feature Engineering for Financial Crime Detection

The power of the graph model is realized through pattern matching and algorithm execution, which efficiently uncover fraud rings and money mule networks by focusing on connected data.

1. Detecting Circular Transaction Rings (Money Laundering): Circular transaction patterns are a hallmark of layering in money laundering operations, where funds are moved through multiple accounts before returning to the originator to obscure the source. The GDM enables specialized graph queries (e.g., using Cypher or GQL) to find paths where the ring starts and ends with the same account, with transactions occurring sequentially in time, and where intermediary accounts retain a small percentage of the funds (e.g., less than 20%). This capability allows Sentinel AI to move beyond isolated transaction monitoring and identify the complete structure of illicit fund flows.

2. Shared Identifier Analysis (Exposing Fraud Rings and Synthetic Identities): Fraudsters creating fake profiles or synthetic identities often recycle limited identity information (e.g., a

phone number, email, or IP address). By leveraging the SHARED_ATTRIBUTE edge, analysts can quickly execute queries to find all customers or accounts that share a specific attribute (e.g., the same client IP address). Clustering analysis based on shared attributes quickly exposes accounts that are seemingly unrelated in the relational store but are secretly linked by the fraudster. This is effective in identifying mule accounts that operate across different locations but share an underlying digital signature.

3. Agent Collusion Modeling: The enhanced KYA requirements and the regulatory focus on dedicated agent accounts are aimed at mitigating agent and merchant collusion, where insiders or networked agents manipulate cash flows. In the graph model, agent collusion is detected by analyzing the transactional relationship between the Agent node, their associated Dedicated_Account, and the Account nodes they transact with.

Graph metrics are engineered specifically for this purpose:

- **Centrality Metrics:** Algorithms like PageRank can identify agents who act as central "hubs" in the network, exhibiting abnormally high connectivity or mediating transactions between otherwise disconnected groups (high betweenness centrality). Such agents often represent crucial nodes in collusion rings.
- **Community Detection:** Algorithms like Louvain can group agents, customers, and beneficiaries who transact frequently within a closed cluster, providing evidence of a localized fraud ring operating independently from the main customer base.
- **Transaction Imbalance Analysis:** The graph can model rapid or unusual fund movements facilitated by the agent, such as direct deposits into third-party wallets which are sometimes illegal because they violate customer verification guidelines and facilitate money laundering.

The network intelligence derived from graph analysis provides robust features for ML models that are impossible to calculate efficiently in a relational environment.

Graph Network Features

Feature Category	Graph Algorithm	Application in Sentinel AI
Centrality Metrics	PageRank, Betweenness	Identifies highly influential "hub" accounts or agents who act as central intermediaries in fraud rings.
Path Analysis	Shortest Path, All Paths	Confirms circular transaction flows, structured deposits, and measures the distance between a suspect and known fraudulent entities.
Community Detection	Louvain, Connected Components	Groups accounts/devices that interact tightly and exclusively, exposing closed fraud rings or mule networks.
Density/Sharedness	Shared Neighbors, Similarity	Measures the overlap in shared attributes (IP, Address) between entities to flag synthetic identities and linked fraud accounts.

V. Data Governance and Scalability Recommendations

The successful deployment of Sentinel AI requires strict governance protocols, particularly regarding data privacy, model training, and the management of a complex hybrid technology stack.

A. Data Security, Anonymization, and Synthetic Data Protocols

Given the highly sensitive nature of the PII and transactional data, robust security measures are necessary. All personally identifiable information (PII), including BVN, NIN, and account names, must be tokenized or pseudonymized for use in model training environments. Only hashed or anonymized versions should be used by ML pipelines (such as the V1-V28 features). Furthermore, financial crime events (Fraud, Class 1) are rare, leading to severe class imbalance, which challenges supervised machine learning algorithms. To address this and protect real customer data, Sentinel AI should incorporate advanced **Synthetic Data Generation**. This process involves creating artificial data samples based on anonymized distribution files and path structures observed in production data (e.g., exploded JSON paths). Synthetic data generation helps augment the training sets, improving model robustness against fraud rarity and mitigating the risk of model drift.

B. Compliance Reporting Requirements and Audit Trails

Sentinel AI must maintain rigorous auditability to meet regulatory standards. The system must track and store evidence of every decision, score calculation, and feature derivation, linking back to the dual timestamp fields (event_timestamp and processing_timestamp) to fulfill regulatory obligations.
The Fraud_Labels Table is critical in the compliance pipeline, as it provides the linkage to external Case Management Systems (CMS) where investigation resolutions, including the filing of Suspicious Activity Reports (SARs), are documented. This ensures that regulatory outcomes are continuously fed back into the data architecture, providing the highest quality labels for future model improvements.

C. Technology Stack Implications (Relational vs. Graph)

The data architecture necessitates a tiered, hybrid stack to optimize for different data access patterns:

1. **Relational Layer (Source of Truth):** PostgreSQL or a similar high-consistency database should house the Transactions, Customers, Agents, and Beneficiaries tables. This layer is optimized for transactional integrity, high-volume ingestion, and standardized reporting.
2. **Feature Store Layer (ML Readiness):** This layer, which could leverage technologies like Redis or specialized feature stores, caches the pre-calculated velocity and behavioral metrics derived from the relational layer (e.g., the features documented in Section III). This separation is vital for providing low-latency feature retrieval necessary for real-time model scoring.
3. **Graph Layer (Network Intelligence):** A dedicated, scalable graph database (e.g., Neo4j) is required for housing the network topology derived from the Graph_Relationships table. This layer performs the heavy-duty link analysis and pattern matching necessary to expose fraud rings and hidden connections that traditional databases cannot efficiently handle.

This architectural choice ensures that the system achieves maximum performance: speed for

handling instantaneous transactions (NIP) and analytical depth for investigative pattern matching.

VI. Conclusions and Recommendations

The schema proposed for Sentinel AI successfully integrates transactional integrity, rigorous compliance requirements (KYC/KYA), and advanced featurization capabilities necessary for state-of-the-art financial crime detection.

The analysis confirms that compliance status is an intrinsic risk feature; mandating the tracking of Agent exclusivity, dedicated accounts, and BVN/NIN compliance across the Agents and Customers tables provides a powerful, legally-grounded input for ML models attempting to detect collusion and synthetic identities.

The implementation of the **Dual Timestamp Requirement** (event_timestamp vs. processing_timestamp) across all transaction records is non-negotiable for maintaining temporal fidelity and preventing model bias, ensuring the platform adheres to critical model risk management standards.

The adoption of a **Hybrid Relational-Graph Architecture** is the only feasible method for detecting modern, coordinated fraud. While the relational layer handles volume and speed, the graph layer enables rapid traversal of connected data to identify complex schemes like circular fund movements and shared identifiers, dramatically improving the ability to find fraud patterns 1000 times faster than traditional databases.

Key Recommendations for Implementation:

1. **Prioritize Feature Store Development:** Immediate development must focus on the feature engineering pipeline (Section III) to calculate and cache velocity metrics (e.g., withdrawal_pct_7d) based exclusively on the event_timestamp. These derived metrics are the most critical inputs for both anomaly detection and supervised models targeting money mule networks.
2. **Enforce Strict KYA Data Discipline:** Ensure that the data ingestion pipeline validates and enforces the population of bvn, nin, and dedicated_account_id for all entities in the Agents Table as mandated by regulatory guidelines. Any missing data points must immediately translate into a higher risk score for the associated transactions.
3. **Establish Graph Synchronization:** Implement a robust ETL process to project data from the core relational tables (Transactions, Devices, Graph_Relationships) into the dedicated graph store in near real-time, allowing the network intelligence layer to remain current and effective in detecting emerging fraud rings.

Works cited

1. Bank Verification Number(BVN) - NIBSS, https://nibss-plc.com.ng/bank-verification-numberbvn/ 2. Understanding the New 2025 Agent Banking Guidelines from the CBN, https://openafricapod.substack.com/p/understanding-the-new-2025-agent 3. Detecting Fraudulent Transactions: A Guide to Building an Advanced Fraud Detection System | by Nafisa Lawal Idris | Medium, https://medium.com/@nafisaidris413/detecting-fraudulent-transactions-a-guide-to-building-an-advanced-fraud-detection-system-9e7506af55a4 4. Anomaly Detection: What You Need To Know - BMC Helix, https://www.helixops.ai/info/anomaly-detection.html 5. Understand the AML data

model and requirements | Anti Money Laundering AI, https://cloud.google.com/financial-services/anti-money-laundering/docs/understand-data-model-requirements 6. AML input data model | Anti Money Laundering AI - Google Cloud, https://cloud.google.com/financial-services/anti-money-laundering/docs/reference/schemas/aml-input-data-model 7. Specification for CIR XML Extract Files (XML Schema Version 5.0.1) - Fiscal.Treasury.gov, https://fiscal.treasury.gov/files/cir/XML5.0.1ExtractFilesSpecificationv201.pdf 8. AML Model Validation in Compliance with OCC 11-12: Supervisory Guidance on Model Risk Management - ACAMS, https://www.acams.org/sites/default/files/2020-08/AML%20Model%20Validation%20in%20Compliance%20with%20OCC%2011-12-%20Supervisory%20Guidance%20on%20Model%20Risk%20Management.pdf 9. Real-Time Risk Scoring for Payments With Identity Data | Vonage, https://www.vonage.com/resources/articles/risk-scoring-for-payments/ 10. Fraud Prevention | TransUnion, https://www.transunion.com/business-needs/fraud-prevention 11. How AI and Analytics Are Revolutionizing Fraud Detection in Mobile Money, https://www.subex.com/blog/how-ai-and-analytics-are-revolutionizing-fraud-detection-in-mobile-money/ 12. Mobile money and organized crime in Africa - Interpol, https://www.interpol.int/content/download/15457/file/2020%2007%2015%20PUBLIC%20VERSION%20-%20Strategic%20Analysis%20Report%20-Mobile%20Money%20in%20Africa%202020.pdf 13. Agent Banking Under Watch: Nigeria's New Rules to Block Money Laundering Networks, https://fincrimecentral.com/nigeria-reform-agent-banking-aml-guidelines/ 14. NIBSS Instant Payment, https://nibss-plc.com.ng/nibss-instant-payment/ 15. Graph Databases for Fraud Detection & Analytics | Neo4j, https://neo4j.com/use-cases/fraud-detection/ 16. How Graph-Based Rules Outsmart Modern Fraud - Blog - Unit21, https://www.unit21.ai/blog/the-network-strikes-back-how-graph-based-rules-outsmart-modern-fraud 17. Behavioral Analysis for Fraud Prevention: How It Works - Disputifier, https://www.disputifier.com/post/behavioral-analysis-for-fraud-prevention-how-it-works 18. What is Anomaly Detection? - AWS, https://aws.amazon.com/what-is/anomaly-detection/ 19. Enhanced Money Mule Detection - LexisNexis Risk Solutions, https://risk.lexisnexis.com/insights-resources/article/money-mules 20. Fighting Financial Crime: AI & Data Analytics in Money Mule Detection, https://www.namlcftc.gov.ae/media/nqhlu4av/acpf_dwg_money_mules_mar2025-v4-0.pdf 21. Graph Database Use Cases for Financial Services Companies, https://blogs.oracle.com/database/post/graph-database-use-cases-for-financial-services-companies 22. Combatting Money Mule Networks: The Power of Graph AI in Financial Crime Detection, https://datawalk.com/combatting-money-mule-networks-the-power-of-graph-ai-in-financial-crime-detection/ 23. Combating Money Laundering: Graph Data Visualizations - Neo4j, https://neo4j.com/blog/fraud-detection/combating-money-laundering-graph-data-visualizations/ 24. Neo4j Fraud Demo - Developer Guides, https://neo4j.com/developer/demos/fraud-demo/ 25. Fraud in Mobile Financial Services: Protecting Consumers, Providers, and the System - World Bank Documents, https://documents1.worldbank.org/curated/en/249151504766545101/pdf/119208-BRI-PUBLIC-Brief-Fraud-in-Mobile-Financial-Services-April-2017.pdf 26. Synthetic Data Generator for Financial Contracts - Goldman Sachs Developer, https://developer.gs.com/blog/posts/synthetic-data-generator 27. Best Practices in Financial Crime Data Management - Flagright, https://www.flagright.com/post/best-practices-in-financial-crime-data-management