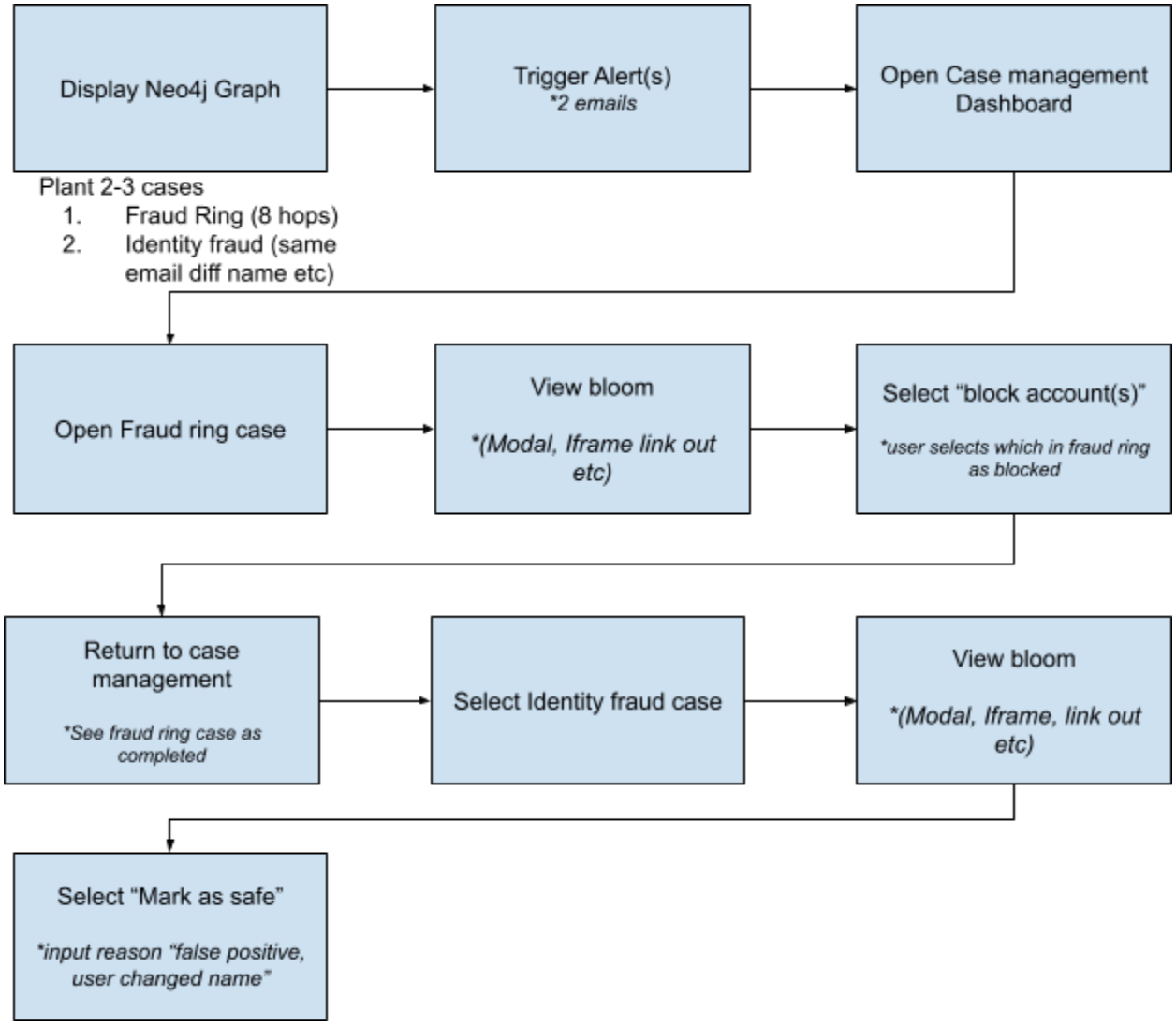
High level demo script



Module 1: Graph Model

User Story 1.1: Unified Fraud Graph Ingestion

> As a Data Engineer, I want to ingest a representative dataset of Transactions, Accounts, and Devices into Neo4j, so that I can establish the "Fraud Graph" required for fraud analysis.

- **Acceptance Criteria:**
  - The graph must contain at least **three distinct node types**: :Account, :Transaction, :Device.
  - The graph must contain relationships linking them: (:Account)-[:PERFORMS]->(:Transaction), (:Transaction)-[:TO]->(:Account), (:Account)-[:USES]->(:Device).

- **Specific Scenario:** The data must effectively model a **"Money Mule Ring"**: A cluster of multiple accounts sharing 1-2 devices (or IP addresses) and transferring funds in a circular or layered pattern. The data must also have a potential identity fraud case in the data

---

Module 2: The Alerting Framework

User Story 2.1: Configurable Graph Rule Execution

As a fraud analyst, I want to configure a detection rule based on a Cypher query (e.g., "Mule Ring Detection") So that the system automatically informs users on a new case for investigation.

- **Acceptance Criteria:**
  - System must be able to store a "Rule Definition" containing: Rule Name, Cypher Query, and Description.
  - System must execute the Cypher query against the POC dataset.
  - **Success Indicator:** The execution must result in a JSON/Object output representing an "Alert" linked to the specific Accounts/Transactions found in the pattern.

User Story 2.2: Prioritized Analyst Queue

As a Fraud Analyst, I want to view a centralized dashboard of generated alerts sorted by Risk/Severity, so that I can prioritize investigating the most critical threats first.

- **Acceptance Criteria:**
  - **UI Requirement:** A simple dashboard list view displaying columns: Alert ID, Rule Name (e.g., "Mule Ring Suspected"), Severity (Critical), and Timestamp.

Module 3: Integrated Case Management

User Story 3.1: Automated Context Population

As a Fraud Analyst, I want to click on an alert and immediately see a Case View populated with the relevant linked entities (Accounts, Devices, Previous Flags),

So that I don't have to manually query the database to understand the context.

- **Acceptance Criteria:**
  - **UI Requirement:** Clicking an Alert ID opens a "Case Detail" modal/page.
  - **Context Fetch:** The page must display the "Subject" (the Account flagged) *and* its immediate network (e.g., "Also linked to 3 other accounts via Device ID X").

    - **Visual Integration:** Embed (or link out) a specific **Neo4j Bloom** visualization for that case ID.
        - Bonus points* if able to do additional queries on bloom around the impacted nodes to demonstrate even more enhanced investigation capabilities

User Story 3.2: Case Resolution & Audit Trail

As a Compliance Officer,

I want to ensure that every case resolution (e.g., Block, False Positive) is logged with a timestamp and user ID,

- **Acceptance Criteria:**
    - **Workflow:** The Case View must have a "Take Action" dropdown with options: Block Account, Mark as Safe, Escalate.
    - **State Change:** Selecting an option updates the Case Status (e.g., Open -> Resolved).