Robinhood Post-Hiring PoC

January 29th 2026

Clarity

Post-Hire POC: Identity Orchestration

Objective: To validate Clarity's orchestration engine and its ability to support the end-to-end employee identity lifecycle for Robinhood's global workforce.

1. POC Scope: Key Test Scenarios (Post-Hire)

- **Scenario A: High-Assurance Account Recovery (MFA Reset)**
  - **The Operational Goal:** Enable "immediate decisions" for high-risk events (like MFA resets) while minimizing manual escalations by providing clear failure context to the requester.
  - **The Trigger**: A simulated MFA reset or "Suspicious Activity" signal in Okta.
  - **The Action**: Clarity initiates a verification session using `useCaseId: CREDENTIAL_CHANGE` or `SUSPICIOUS_ACTIVITY`. The orchestration engine retrieves the user's existing biometric profile - established during Phase 1 (Pre-Hire, if exists) enrollment - and triggers a `LIVENESS_CHECK` + `DEEPFAKE_DETECTION` + `BIOMETRIC_COMPARISON`.
    - **The Logic:** Clarity provides an automated `PASS/FAIL` based on Liveness and Biometric match to keep the employee moving.
    - **The Deepfake Layer:** The `DEEPFAKE_DETECTION` service runs in parallel. If the automated model flags a high risk, the Human Expert is triggered as an asynchronous audit.
    - **The Result:** The Robinhood Security Team (the "requesters") receives an immediate alert if the later Human Audit contradicts the initial automated pass.
  - **The Goal:** To prove that Clarity can verify an existing employee's identity using persistent biometric data, eliminating the need for the user to present a physical ID document for high-frequency security events.
  - **Success Criteria:**
    - The system performs a match against historical enrollment data.
    - Verification results are pushed to Okta/ServiceNow with a unique `internalUniqueId` (Employee ID).
    - The system returns granular, actionable failure codes (e.g., `EXPIRED_DOCUMENT`, `LIVENESS_FAILED`) to enable the Robinhood internal team to understand the failure and determine the appropriate next steps (e.g., re-initiate check vs. security escalation) without needing to contact Clarity support.
- **Scenario B: Historical (Lifecycle) Identity Correlation**
  - **The Operational Goal:** Preserve and track the "historical context" for identity flows across the end-to-end employee lifecycle, improving security and minimizing friction for users (eg. no document verification needed).

    - **The Action**: Execute a "Cross-System Thread" test. We will initiate a verification for "User A" via a mock **Greenhouse** (ATS) trigger and then perform a follow-up "Step-Up" check via **Okta** using the same `internalUniqueId`.
    - **The Goal:** To demonstrate Clarity's ability to act as a Unified Identity Ledger, correlating verification events from disparate systems (ATS, IAM, ITSM) into a single, persistent, and searchable audit trail for each employee.
    - **Success Criteria**:
        - The Clarity Admin Console displays a unified chronological timeline for "User A".
        - Each event is correctly labeled with its `useCaseId` (e.g., `ONBOARDING` vs. `CREDENTIAL_CHANGE`).
        - Filtering by `internalUniqueId` returns all relevant forensic evidence from both the hiring and employment phases.
- **Scenario C: IAM Infrastructure Integration**
    - **The Operational Goal:** Validate the integration of Clarity as a custom Identity Provider within Robinhood's existing Okta and RBAC architecture.
    - **The Action**: Demonstrate the Okta Custom IDV Integration and Administrative RBAC. This involves setting up the "Identity Provider" in a mock Okta environment as per the guide we provided.
    - **The Goal:** To validate that Clarity is able to supportRobinhood's internal security requirements for Single Sign-On (SSO), provisioning, and granular administrative roles.
    - **Success Criteria**:
        - A Robinhood Admin can log into Clarity's console via Okta SSO.
        - Admin permissions are restricted based on roles (e.g., a Recruiter cannot see the "Fraud Evidence File" meant for the Insider Threat team).
        - The "Clarity IDV" appears correctly as a custom vendor within the Okta Identity Provider settings.
        - The API response includes actionable failure codes (e.g., EXPIRED_ID, LIVENESS_FAILED) for the requester

2. Technical Readiness: Administrative Experience

- **Single Sign-On (SSO):** Validate that Robinhood admins can log into Clarity via Okta SSO with roles restricted to their specific department .
- **Manual Verification Link Generation:** Demonstrate the admin capability for support teams to manually initiate an IDV session via the Clarity console for ad-hoc user verification.

3. Infrastructure Commitment: Flexibility & Extensibility

The following section outlines the architectural principles of the Clarity platform and our commitment to supporting Robinhood's evolving needs.

Clarity

- **Modular Orchestration Engine**
    - Clarity’s infrastructure is built as an Identity Service Bus. While the POC focuses on specific triggers, our system is architected to allow Robinhood to define and deploy new identity "Stage-Gates" by simply updating a JSON configuration, rather than refactoring code .
- **Vendor Neutrality**
    - Clarity acts as the Identity Service Bus (Orchestration Layer). Our architecture supports routing to secondary IDV sub-processors (e.g., Clear, Onfido) via a unified API, ensuring business continuity and regional compliance
- **Policy-Based Friction Control**
    - Our infrastructure supports the implementation of Adaptive Friction. This allows Robinhood to programmatically adjust verification requirements—such as toggling between automated LIVENESS_CHECK and "Human-Expert" review - based on the specific risk level of the useCaseId being triggered.
- **Configurable Escalation Policies**
    - Robinhood can toggle between 100% automated 'Immediate Decisioning' for high-velocity flows (MFA resets) and 'Human-Expert Supervision' for high-stakes events (suspicious activity detected)