PhishSafe: Behavior-Based Continuous Authentication for Mobile Banking Security

Problem Statement

- Traditional mobile banking security stops at login (PINs, passwords, OTPs, biometrics).
- Post-login sessions are vulnerable to phishing attacks, session hijacking, SIM swaps, and device handovers.
- A significant number of financial frauds occur **after login**, when the attacker masquerades as a legitimate user.
- Current security layers lack **real-time continuous identity validation** throughout the session.
- There's a growing need for **seamless, intelligent, and adaptive security** that doesn't interrupt user experience.

Objective

- To develop an **intelligent, behavior-based continuous authentication system** that detects anomalies during mobile banking sessions.
- Strengthen protection against phishing, session hijacking, and unauthorized access.
- Ensure **high user trust** without compromising on usability or accessibility.

Core Concept

- Builds a **Trust Score** for every session based on the user's real-time behavior.
- If trust score drops below a threshold, trigger adaptive responses (step-up auth, session freeze, alerts).

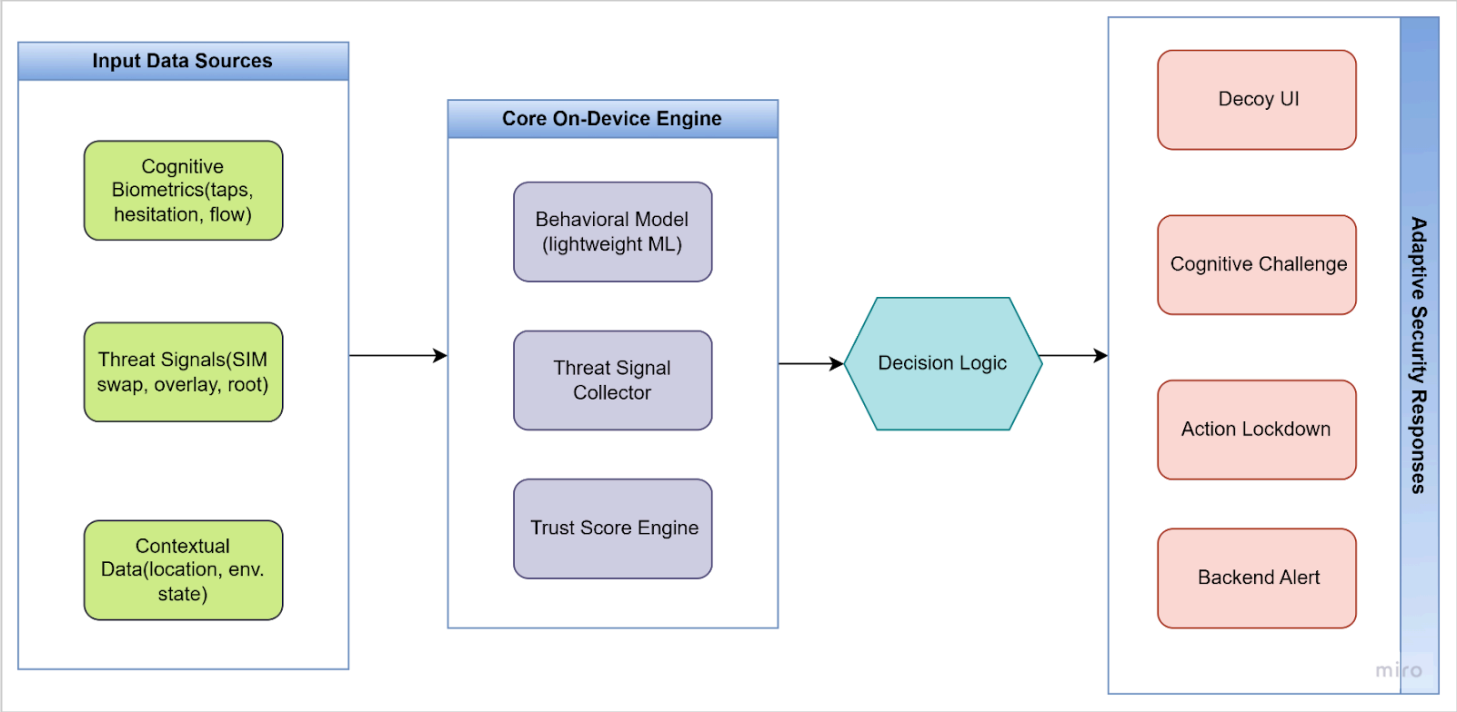
Solution Overview

PhishSafe is an advanced, context-aware fraud detection and cognitive authentication system designed for mobile banking apps. It continuously evaluates a user's behavior, decision flow, and device context in real-time to detect fraud—especially under social engineering or duress.

It introduces **Cognitive Behavioral Biometrics** combined with **Micro-Threat Signal Analysis** to detect suspicious session activity and take smart, adaptive security actions.

PhishSafe is an intelligent, on-device fraud detection system for mobile banking apps that:

- Continuously analyzes **user behavior and cognitive patterns** during sessions
- Collects **contextual threat signals** like SIM swaps, device overlays, and access patterns
- Maintains a **dynamic Trust Score** that evolves during the session
- Triggers **smart responses** (decoy UI, behavioral questions, feature lockdown) when trust drops
- This keeps genuine users safe—even under social engineering attacks—without ruining the user experience.



Session Flow with Behavior-Based Authentication

Key Features

1. **Cognitive Biometrics**
- Analyze hesitation before tapping buttons
- Detect irrational or rushed navigation paths
- Detects suspicious rushes (e.g., instantly going from login to transfer page)
- Track decision-making patterns compared to normal behavior

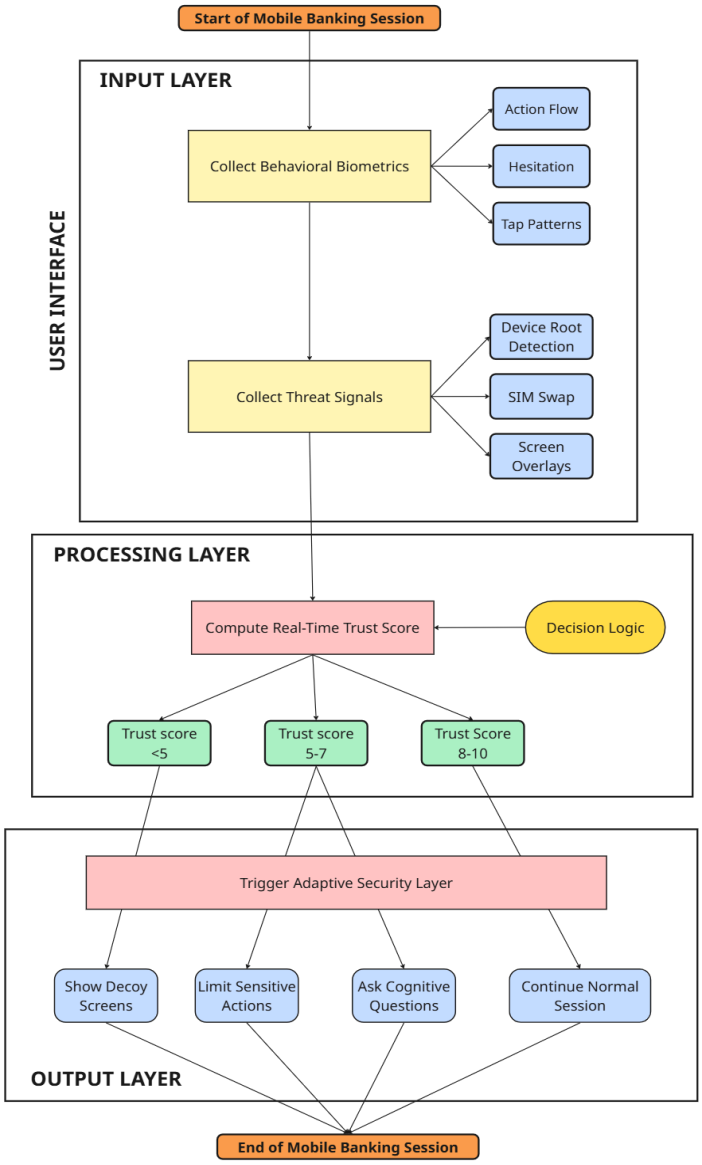
2. **Micro-Threat Signal Monitoring**
- Detect SIM swaps, screen overlay apps, rooted devices, or emulators
- Monitor location inconsistencies or device environment changes

3. **Dynamic Trust Scoring**
- Combine user behavior and threat signals to calculate session risk in real-time
- Trigger adaptive responses based on trust score thresholds

4. **Smart Security Actions**
- Display **decoy screens** to block real data during suspicious sessions
- Prompt the user with cognitive confirmation questions ("What was your last transaction?")
- Temporarily lock or restrict sensitive actions like fund transfers
- Raise silent backend alerts for internal fraud monitoring

5. **Adaptive Security Response**
- Session Timeout
- Step-up re-authentication
- Transaction blocking
- Silent alert to bank systems

Flowchart (Session Flow)



Implementation Plan

Phase 1: Behavior Data Collection (Synthetic & Simulated)

- Touch interaction datasets (e.g., pressure, swipe, tap)
- Session flows, common usage paths

Phase 2: ML Model Design

- Use supervised + semi-supervised learning
- Train anomaly detection models (Isolation Forest, Autoencoders, or LSTM)
- Trust Score algorithm to assign risk levels to sessions

Phase 3: Prototype App Development

- Android-based mobile banking simulation app
- Integration of behavior tracking and Trust Score evaluation
- Frontend mock screens to show step-up auth or session alerts

Phase 4: Real-Time Simulation

- Use synthetic users and simulated attack sessions to evaluate model performance
- Tune thresholds to reduce false positives/negatives.

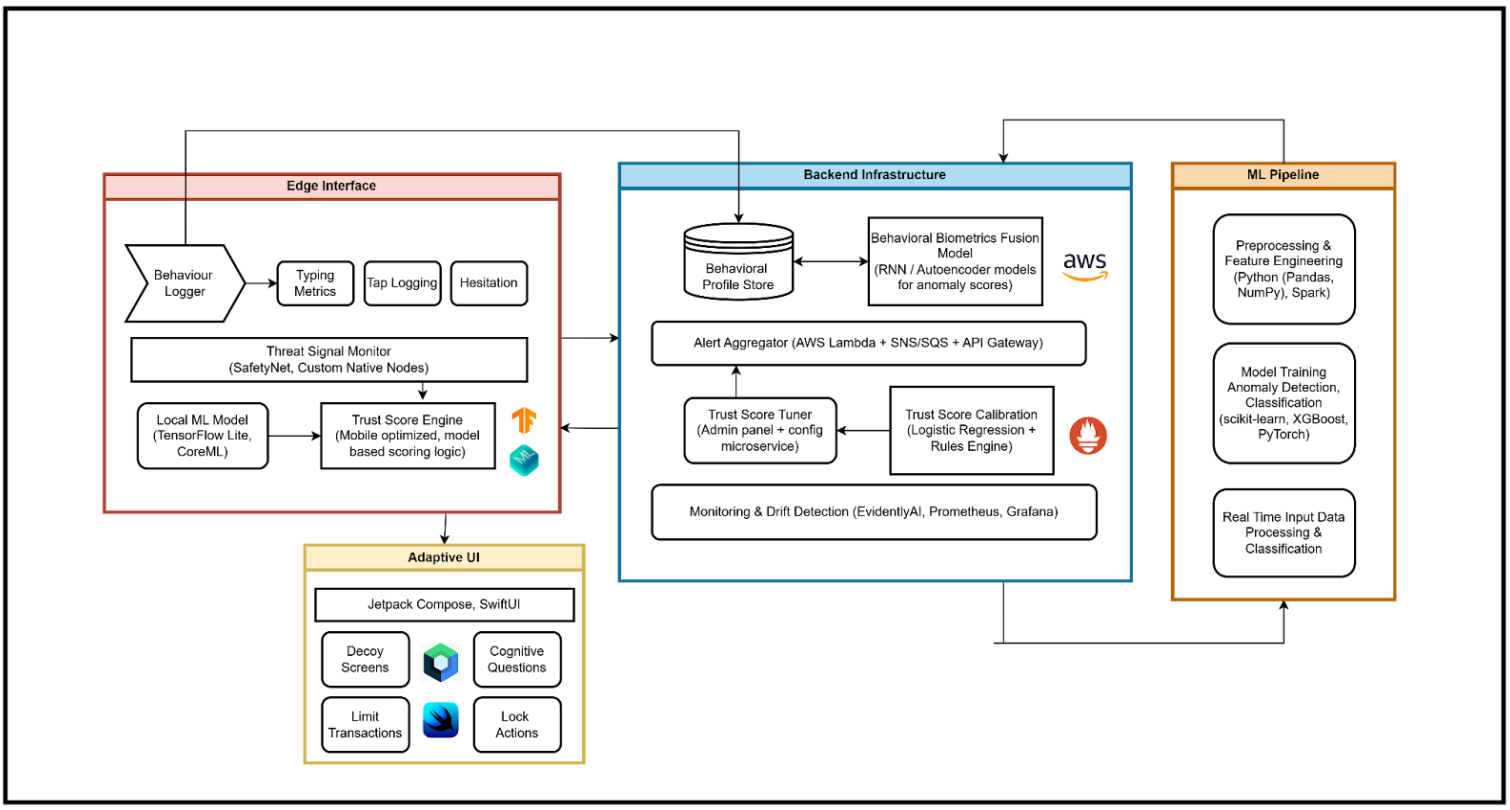
Real-World Applications

- Integrated as an SDK into mobile banking apps for continuous verification.
- Can be extended to UPI apps, e-wallets, investment apps, and even corporate banking platforms.
- Helps prevent authorized push payment (APP) frauds, device-theft abuse, and phishing-based takeovers
- Security Layer: AES Encryption for local data, secure enclave API for trusted operations

Technologies Used

- Frontend: Flutter / Kotlin (for mobile UI)
- Backend: Python FastAPI / Node.js for alert handling
- ML/AI: Scikit-learn, PyTorch or TensorFlow Lite for behavior modeling
- Storage: SQLite / LocalStorage (minimal user data)
- Security Layer: AES Encryption for local data, secure enclave API for trusted operations.

System Architecture



Value Proposition

- **Banks get**: Lower fraud rates, better session control, reduced losses, regulatory compliance.
- **Users get**: Invisible protection, peace of mind, and reduced false-positive account freezes.
- **Regulators get**: A privacy-friendly model compliant with **DPDP Act** and **zero-knowledge principles**
  .

Constraints & Known Issues

- **Cold start problem:** New users won't have enough behavior data use default profile with conservative risk policy.
- **Device permission issues:** Some devices may restrict touch data APIs fallback to context-only mode.
- **Elderly or differently-abled users:** May need special models or lower sensitivity thresholds.

Reusability & Scalability

- Can be deployed as a **modular SDK** usable across banking, insurance, and fintech platforms.
- **Plug-and-play API** design for custom Trust Score logic and risk policies.
- Easily extensible to wearables or biometric+behavior fusion in the future.

Future Enhancements

- Incorporate gesture recognition (with user consent).
- Develop bank dashboard to view risk scores, alerts, and logs.
- Build an open benchmark dataset for behavior-based continuous authentication in finance