B

Begin Wallet

Catalyst Fund 12 - Milestone 2 Report

Face Biometrics Integration

Executive Summary

This report documents the successful implementation of face biometrics authentication for Begin Wallet, fulfilling the requirements of Catalyst Fund 12 Milestone 2. The integration enables users to authenticate transactions and access their wallet using facial recognition technology, providing a seamless and secure user experience.

5	60%	100
Core Features	Confidence Threshold	Consensus Attempts

30s
Verification Window

Implemented Features

1. Face Enrollment

ISO/IEC 19794-5 compliant facial image capture with quality validation. Users can enroll their face through Settings > Security > Face Biometrics with real-time liveness detection.

Complete

2. Biometric Wallet Creation

Generate deterministic seedphrases from face descriptors using 100-attempt consensus algorithm. Converts 128-dimensional face descriptors to BIP39 entropy for 15-word mnemonic generation.

Complete

3. Biometric Wallet Recovery

Recover wallets using face recognition with Blockfrost API verification. Validates recovered wallets by checking for existing blockchain transactions.

Complete

4. Passwordless Transaction Signing

Face verification acts as a secure password manager. After successful verification, encrypted password is decrypted and used automatically for transaction signing.

Complete

5. Transaction Authorization

Multiple security modes: password-only, password-or-face, password-and-face. Face verification with 60% confidence threshold provides secure transaction approval.

Complete

6. Enhanced Liveness Detection

Multiple anti-spoofing measures including face area validation, eye distance checks, head tilt detection, and blink detection via eye aspect ratio analysis.

● Complete

Technical Architecture

Core Components

`src/lib/faceBiometrics.ts` - Core face detection, matching, liveness, and settings management
`src/lib/biometricSeedGenerator.ts` - Face-to-seedphrase conversion with consensus algorithm
`src/components/FaceBiometricVerify.tsx` - Verification modal for transactions
`src/hooks/useFaceBiometrics.ts` - React hook for face verification state
`src/views/user-settings/face-biometrics-setup.tsx` - Face enrollment UI
`src/views/account/generate-seed.tsx` - Biometric wallet creation
`src/views/account/biometric-recovery.tsx` - Biometric wallet recovery
`src/views/send/send-confirm.tsx` - Transaction signing with face verification

Biometric Wallet Algorithm

1. Capture face using webcam with face-api.js TinyFaceDetector
2. Extract 128-dimensional face descriptor (Float32Array)
3. Run 100 detection attempts to account for natural variance
4. Convert descriptor to deterministic entropy: `descriptor[i] * 100 → bytes`
5. Generate BIP39 mnemonic from entropy (15-word for 160-bit strength)
6. Use statistical mode (most frequent mnemonic) for wallet creation
7. For recovery: verify wallet has transactions on Blockfrost

Passwordless Signing Flow

1. User initiates transaction and selects face verification
2. Face is captured and compared against stored reference descriptor

3. If match score exceeds 60% threshold, verification is valid
4. Stored reference descriptor is used to derive AES-256-GCM decryption key
5. Encrypted password is decrypted using PBKDF2-derived key (100k iterations)
6. Decrypted password is used to sign the transaction
7. Verification token expires after 30 seconds for security

Security Model

AES-256-GCM Encryption

Wallet passwords are encrypted using AES-256-GCM with a key derived from the reference face descriptor via PBKDF2 with 100,000 iterations.

Face Verification Gate

Users must pass liveness detection and face matching before password can be retrieved. The verification is valid for only 30 seconds.

Key Never Stored

The encryption key is derived at runtime from the face descriptor and random salt. The key itself is never persisted to storage.

Liveness Detection

Multiple anti-spoofing checks including face area validation, eye alignment, head tilt detection, and eye aspect ratio (blink detection).

Security Modes

Mode	Description	Use Case
password_only	Standard password-based signing	Default mode, traditional security

Mode	Description	Use Case
password_or_face	User chooses face OR password verification	Convenience-focused, quick transactions
password_and_face	Requires face verification THEN password	Maximum security, high-value transactions

Testing & Validation

Test Scenarios Completed

- **Face Enrollment**
  Successfully capture and store reference face descriptor with quality validation
- **Password Storage**
  Encrypt wallet password using face descriptor-derived key
- **Face Verification**
  Match live face against stored reference with 60% threshold
- **Password Retrieval**
  Decrypt password using reference descriptor after successful verification
- **Transaction Signing**
  Complete Cardano transaction using retrieved password
- **Biometric Wallet Creation**
  Generate deterministic seedphrase from face using 100-attempt consensus

Milestone Deliverables

Requirement	Status	Implementation
ISO/IEC 19794-5 compliant facial image capture	Complete	face-biometrics-setup.tsx with quality validation
Consensus algorithm for seedphrase	Complete	biometricSeedGenerator.ts with statistical mode
Biometric wallet recovery	Complete	biometric-recovery.tsx with API validation
Passwordless authentication via face recognition	Complete	Encrypted password store in faceBiometrics.ts
Transaction authorization with confidence threshold	Complete	FaceBiometricVerify.tsx with configurable threshold