Frontend Master Protocol (PRD): Iron Will

Version: 1.1 (Expanded)
Status: Approved for Implementation
Scope: Frontend Only (Next.js 14, Tailwind, integration with Spring Boot Backend)

1. Design System & Visual Language

Philosophy:
The User Interface for Iron Will is not a wrapper; it is a Command Interface. It rejects the soft, friendly, dopamine-loop design patterns of modern social media in favor of a stark, utilitarian, and high-stakes aesthetic. The user is operating high-stakes machinery—their own psychology. The interface must feel precise, expensive, and slightly intimidating, like the cockpit of a stealth fighter or the terminal of a bio-research facility. Every interaction should have weight; every click should feel like a decision.

1.1 Color Palette (Tailwind Config)

We do not use standard off-the-shelf colors. We define a custom semantic palette to enforce the "High-End Biohacker" aesthetic. The goal is to create an environment that feels hermetically sealed and hyper-focused.

- **Backgrounds (The Void):**
  - bg-void: #050505 (Main app background). This is not pure black (#000), which can cause smearing on OLEDs, but a deep charcoal that consumes light. It represents the "Focus State."
  - bg-card: #0A0A0A (Card surfaces). A subtle elevation from the void, providing just enough contrast to delineate interactive zones without breaking the immersive darkness.
  - bg-overlay: rgba(0, 0, 0, 0.85) (Glassmorphism base). A heavy tint used for modals and focus states, blurring out the background to enforce tunnel vision on the active task.
- **Primaries (The "Energy"):**
  - text-neon-green: #39FF14 (Success, Active, Charging). This specific shade mimics the phosphor of retro-futuristic terminals and radium dials. It signifies "Systems Nominal" and "Willpower Active."
  - border-neon-green: rgba(57, 255, 20, 0.5) (Glow effects). Used for active borders, conveying that the component is "powered on."
- **Destructive (The "Failure"):**
  - text-crimson: #DC143C (Error, Lockout, Critical). A sharp, aggressive red that triggers an immediate "Alert" response in the user's brain.
  - bg-crimson-glitch: rgba(220, 20, 60, 0.15) (Background tint). Used during the lockout state to bathe the interface in an emergency light, simulating a containment breach.

- **Typography:**
  - font-sans: Inter or Geist Sans (UI Text). Chosen for maximum legibility at small sizes while maintaining a cold, neutral tone.
  - font-mono: JetBrains Mono or Geist Mono (Data, Scores, Timestamps). Monospace fonts imply "raw data" and "truth." All metrics—scores, time, dates—must be rendered in mono to reinforce that these are unalterable facts.
  - text-ghost: #888888 (Inactive labels). Text that recedes into the background, visible only when sought out.

1.2 Core Textures & Effects

- **Glassmorphism (The "HUD"):**
  - Standard Rule: backdrop-blur-xl bg-white/5 border border-white/10.
  - Application: Used for the Header, the Protocol Cards, and the Bottom Navigation (if any). It creates a sense of depth, implying that the data is floating above the void.
- **Scanlines (The "Terminal"):**
  - A persistent, global CSS ::before overlay with pointer-events-none.
  - Implementation: A repeating linear gradient (1px transparent, 1px black @ 3% opacity) combined with a subtle vignette. This gives the entire app a tactile, screen-like texture, reducing the sterility of flat vector graphics.
- **The "Glow" (Haptic Light):**
  - Buttons and active inputs must utilize box-shadow: 0 0 15px rgba(57, 255, 20, 0.3).
  - Logic: Interactive elements should appear to emit light. When a user hovers or focuses on an input, the glow intensity should increase, simulating a surge of power to that component.

2. Page Specifications

2.1 The Gateway (Login Page)

Visual Reference: "Elite Access / Secure Terminal" / "The Airlock"

- **Concept:** This is the transition point between the chaotic outside world and the disciplined interior of the app. It shouldn't feel like a "Signup Form"; it should feel like biometric verification for a vault.
- **Layout:** Centered vertical stack, perfectly balanced.
- **Background:** bg-void.
  - **Texture:** A subtle SVG Hexagonal Mesh pattern (opacity 5%) fades in from the bottom, suggesting a carbon-fiber reinforced structure.
- **Components:**
  1. **Logo:** "IRON WILL" in font-black (Heavy weight), tracking-widest (spaced out).
     - **Animation:** Apply a subtle CSS skew and rgb-shift animation on hover to simulate a digital glitch or signal interference. This establishes the "imperfect/hostile" tech vibe immediately.
  2. **Auth Button:**

        - **Style:** Pill shape. bg-white/10 (Glass). Border border-neon-green/30.
        - **Content:** "Sign in with Google" (White text).
        - **Interaction:** On hover, the border turns solid border-neon-green, the box shadow blooms (box-shadow: 0 0 20px ...), and the text brightens. It should feel like charging a weapon.
3. **Footer:** "SYSTEM STATUS: ONLINE" in text-xs font-mono text-neon-green at the absolute bottom. Include a blinking cursor block _ at the end to show the system is live.

2.2 The Dashboard (HUD)

Visual Reference: "Focus Mode" / "The Cockpit"

- **Layout:** A rigid, grid-based layout (Three-row Grid: Header, Score, Protocols). The user should feel like a pilot scanning instruments.
- **Zone A: The Header (Top 10%)**
    - Left: "IRON WILL" (Small logo, monochrome).
    - Right: Settings Gear (Lucide Icon). When clicked, it opens a mechanical-feeling drawer, not a standard modal.
- **Zone B: The Score (Top 40% - Dominant)**
    - **The Component:** "Health Bar" / "Willpower Gauge."
    - **Visual:** A massive, screen-spanning horizontal progress bar.
        - Container: h-16 bg-gray-900 rounded-none border border-gray-800. Industrial look.
        - Fill: bg-neon-green. Use a CSS linear-gradient to give it a "liquid energy" look. Add an animate-pulse effect to make it look alive.
    - **Data:** The Score (e.g., "8.4") displayed in massive 6xl font-mono text centered *over* the bar (using mix-blend-mode if possible for contrast). This number defines the user's worth in the system.
    - **The Avatar:** An HTML5 Canvas element rendering a floating "Wireframe Sphere" (Three.js or pure Canvas).
        - **Behavior:** It rotates slowly in idle. When the score drops, the wireframe turns red and spikes (becomes jagged). When the score is high, it is a perfect, smooth sphere.
- **Zone C: The Protocols (Bottom 50%)**
    - **List:** Vertical scroll of "Contract Cards."
    - **Card Style:** Minimalist bg-card.
        - Left: Habit Name ("Deep Sleep Protocol") in bold Sans.
        - Right: Status Badge.
            - "PENDING": Hollow yellow border, blinking text.
            - "DONE": Solid Green fill, black text.
            - "FAILED": Solid Red fill, glitch text.
    - **Action:** Clicking a "PENDING" card triggers a "Drawer Open" animation from the bottom (The Audit Modal).

2.3 The Contract (Create Habit)

Visual Reference: "Negotiation Phase" / "Protocol Configuration"

- **Concept:** Not a form, but a "configuration panel." The user is setting parameters for a mission.
- **Input 1: Protocol Name**
  - Style: Transparent background, border-b border-gray-700 (No rounded boxes). Large text-2xl text. The placeholder should be "ENTER PROTOCOL NAME".
- **Input 2: The Time Dial (Custom Component)**
  - **Visual:** Instead of a native OS dropdown, build a horizontal scroll wheel (like a safe dial or an industrial valve).
  - **Style:** The numbers should be Monospace. The "Selected" time is sandwiched between two glowing neon green lines in the center of the screen. Haptic feedback (vibration) should trigger on mobile as numbers tick by.
- **Input 3: Criteria Builder**
  - **Visual:** "Code Blocks."
  - **Interaction:** User inputs criteria like "Sleep Score > 85".
  - **Feedback:** As the user types, the system auto-formats the text into a <code> block styling: [METRIC: Sleep Score] [OPERATOR: >] [VALUE: 85]. This reinforces that the user is programming a rule, not just making a wish.
- **The Button:** "INITIATE CONTRACT"
  - Style: w-full bg-neon-green text-black font-bold uppercase tracking-widest h-14.
  - **Shape:** Rectangular, sharp edges (0px border-radius). It should look like a physical launch button.

2.4 The Auditor (Upload & Verify)

Visual Reference: "Scanning Interface" / "The Judgement"

- **State 1: Dropzone**
  - A dashed border area with a grid background. Text: "DROP EVIDENCE HERE."
  - Icon: A stylized "Upload" arrow that pulses.
- **State 2: Analyzing (The "Magic")**
  - **Animation:** The uploaded image is displayed in grayscale. A "Laser Line" (Green div with absolute positioning and a box-shadow glow) moves top-to-bottom over the image repeatedly (high speed).
  - **Overlay Text:** "EXTRACTING METADATA..." -> "VERIFYING TIMESTAMPS..." -> "COMPARING CRITERIA...". Text should appear with a typewriter effect in Monospace font.
- **State 3: Verdict**
  - **PASS:** A massive rubber-stamp effect "APPROVED" slams onto the screen (scale down animation). Color: Neon Green. Particle effect: Minimalist square confetti falls, signifying data fragment integration.
  - **FAIL:** A violent glitch effect shakes the entire screen viewport. "REJECTED" stamp in

Crimson Red. The Agent's avatar (the sphere) turns red, expands, and becomes jagged/noisy.

2.5 The Lockout (Game Over)

Visual Reference: "Hostile / Punishment" / "Containment Breach"

- **Global Overlay:** z-index: 50. Covers the entire app, blocking all navigation. The user is trapped.
- **Background:** Deep Black with bg-crimson-glitch. A subtle red vignette pulses at the edges of the screen like a heartbeat in a horror game.
- **Effect:** CSS Keyframe animation shake applied to the main container every 5-10 seconds to disorient the user.
- **Centerpiece:**
  - **Text:** "PROTOCOL FAILED" (Stencil font / Military style).
  - **The Timer:** A massive countdown to unlock (e.g., 23:59:42). Font: Monospace, Red.
  - **Subtext:** "Accountability score critical. Access revoked."
- **Interaction:** All buttons are visually disabled (opacity 0.5, grayscale). Cursor set to not-allowed. Clicking anywhere plays a harsh "Access Denied" sound effect (optional, but recommended for PWA).

3. Frontend Architecture (Next.js)

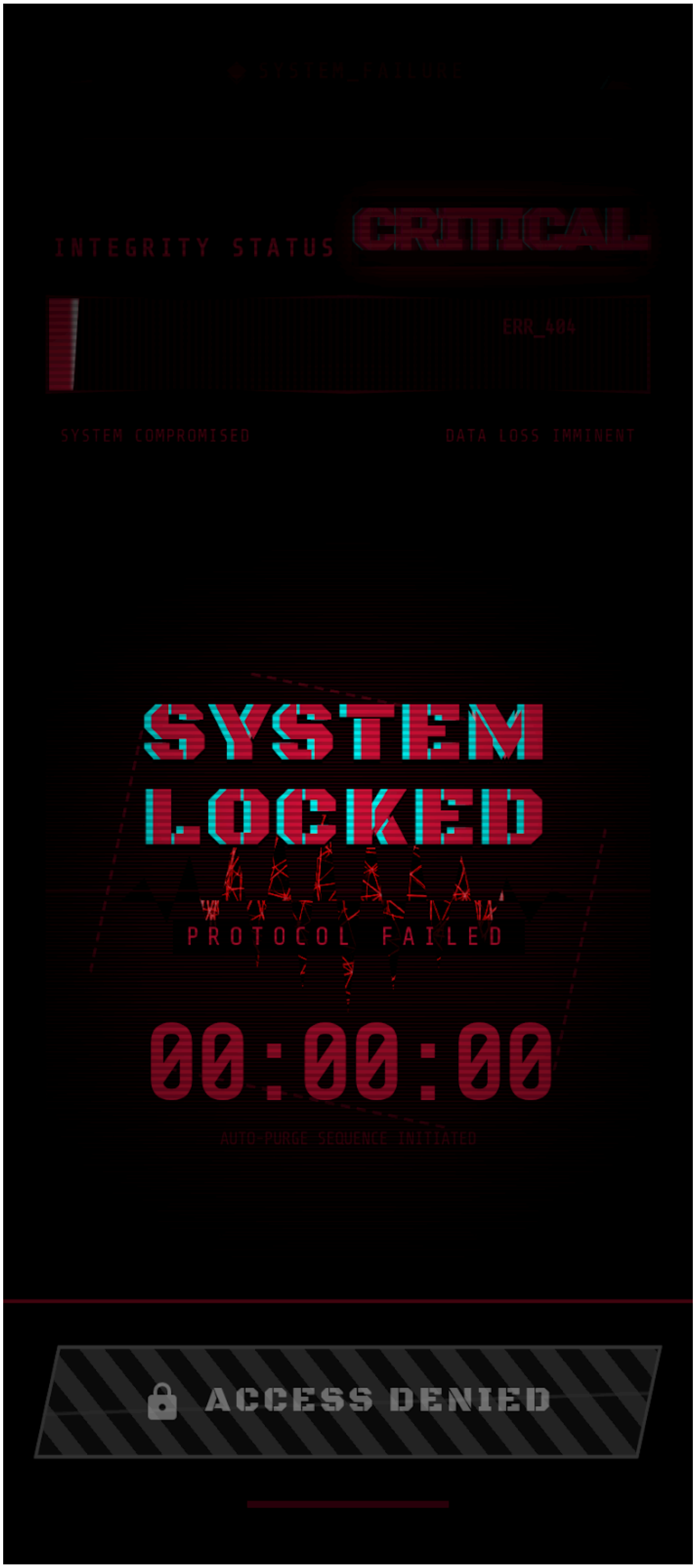
3.1 Tech Stack Justification

- **Framework: Next.js 14 (App Router)**. Chosen for its robust routing and server-side rendering capabilities, ensuring the dashboard loads instantly (crucial for maintaining the "high-performance" feel).
- **Styling: Tailwind CSS** + tailwind-merge + clsx. This combination allows for complex, dynamic class logic (e.g., changing button states based on "Lockout" variables) without messy inline styles.
- **State Management: Zustand**. We need a global store for the "User Score" and "Auth State" that is lighter than Redux but more capable than Context API. Zustand's atomic updates are perfect for high-frequency UI changes (like the score bar).
- **Data Fetching: TanStack Query (React Query) v5**. Handles caching, polling (for the "Nag" notifications), and optimistic updates (instant feedback on upload before server confirmation).
- **Animation: Framer Motion**. Standard CSS transitions are too linear. Framer Motion allows us to use "Spring" physics, making the UI feel heavy, mechanical, and reactive, fitting the Biohacker theme.
- **Icons: Lucide React**. Clean, vector-based icons that fit the minimalist aesthetic.
- **3D/Canvas: React Three Fiber** (or raw Canvas API). Needed for the "Living Avatar" sphere.

3.2 Directory Structure

```
/src
  /app
    /login        # Page: The Gateway
    /dashboard    # Page: The HUD (Protected Route)
    /contract     # Page: New Habit Configuration
    /lockout      # Page: Punishment Screen (Global Redirect if score < 3)
  /components
    /ui           # Atoms: The basic building blocks.
                  # BioButton (Custom variants), BioInput, GlassCard, NeonBadge
    /features     # Organisms: Complex business logic components.
                  # AuditScanner (Upload+Scan logic), ScoreBar (Vis logic), TimeDial
    /canvas       # Three.js/Canvas scenes
                  # AgentSphere.tsx (The 3D Avatar)
  /lib
    /store        # Zustand stores (useAuthStore, useGameStore)
    /utils        # Date formatters, Class mergers, Score calculators
    /hooks        # Custom hooks (useinterval, useLockout)
  /styles
    globals.css   # Custom scanline effects, glitch keyframes, font imports
```

Attaching the UI Design images of the different screens:



● SYSTEM_FAILURE

INTEGRITY STATUS CRITICAL

ERR_404

SYSTEM COMPROMISED

DATA LOSS IMMINENT

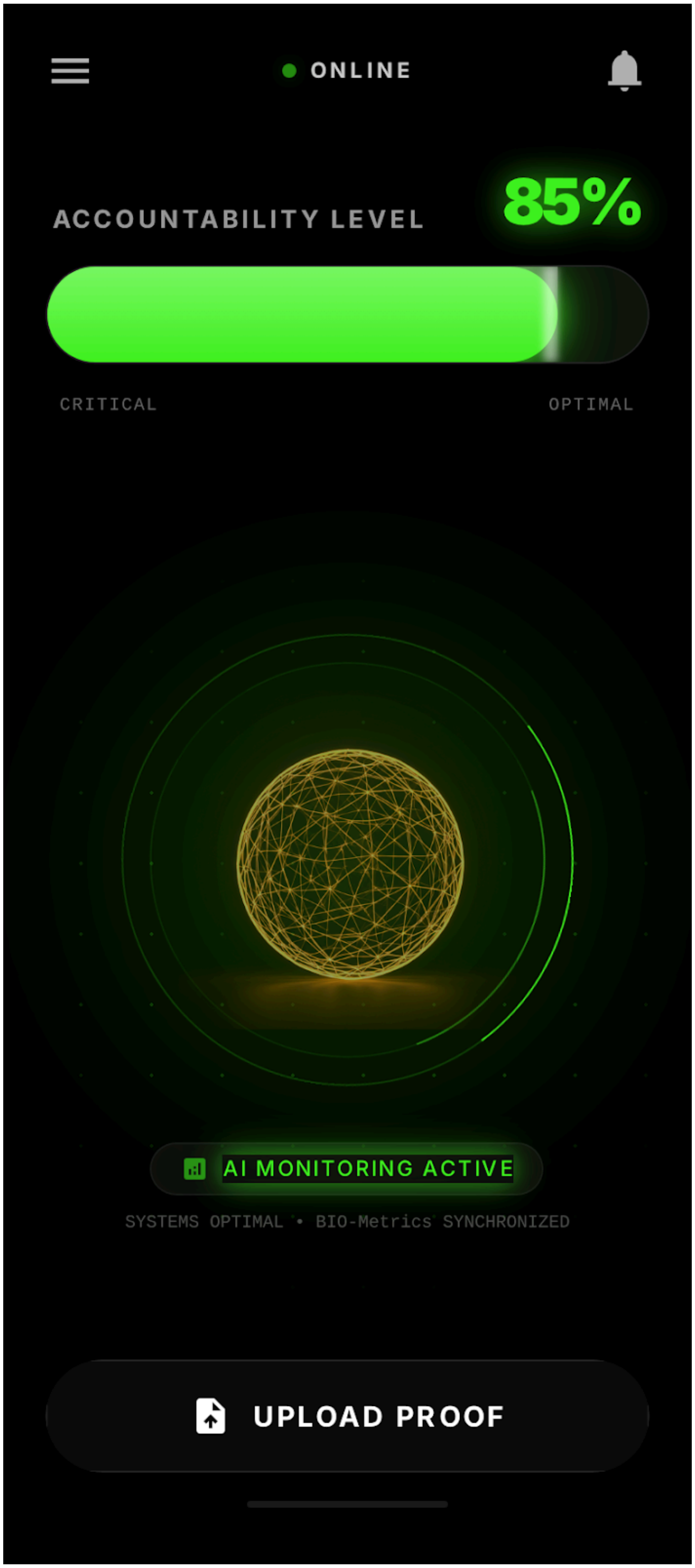
SYSTEM LOCKED

PROTOCOL FAILED

00:00:00

AUTO-PURGE SEQUENCE INITIATED

ACCESS DENIED

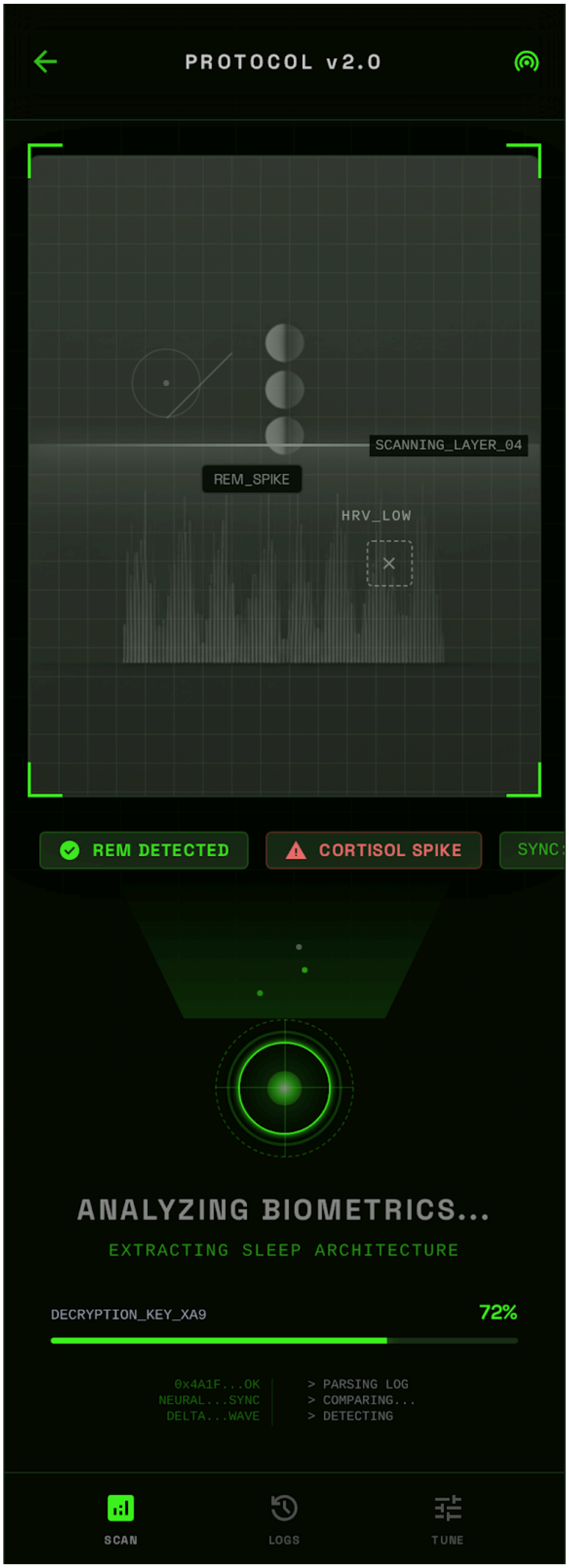


ONLINE

ACCOUNTABILITY LEVEL 85%

AI MONITORING ACTIVE

SYSTEMS OPTIMAL • BIO-Metrics SYNCHRONIZED

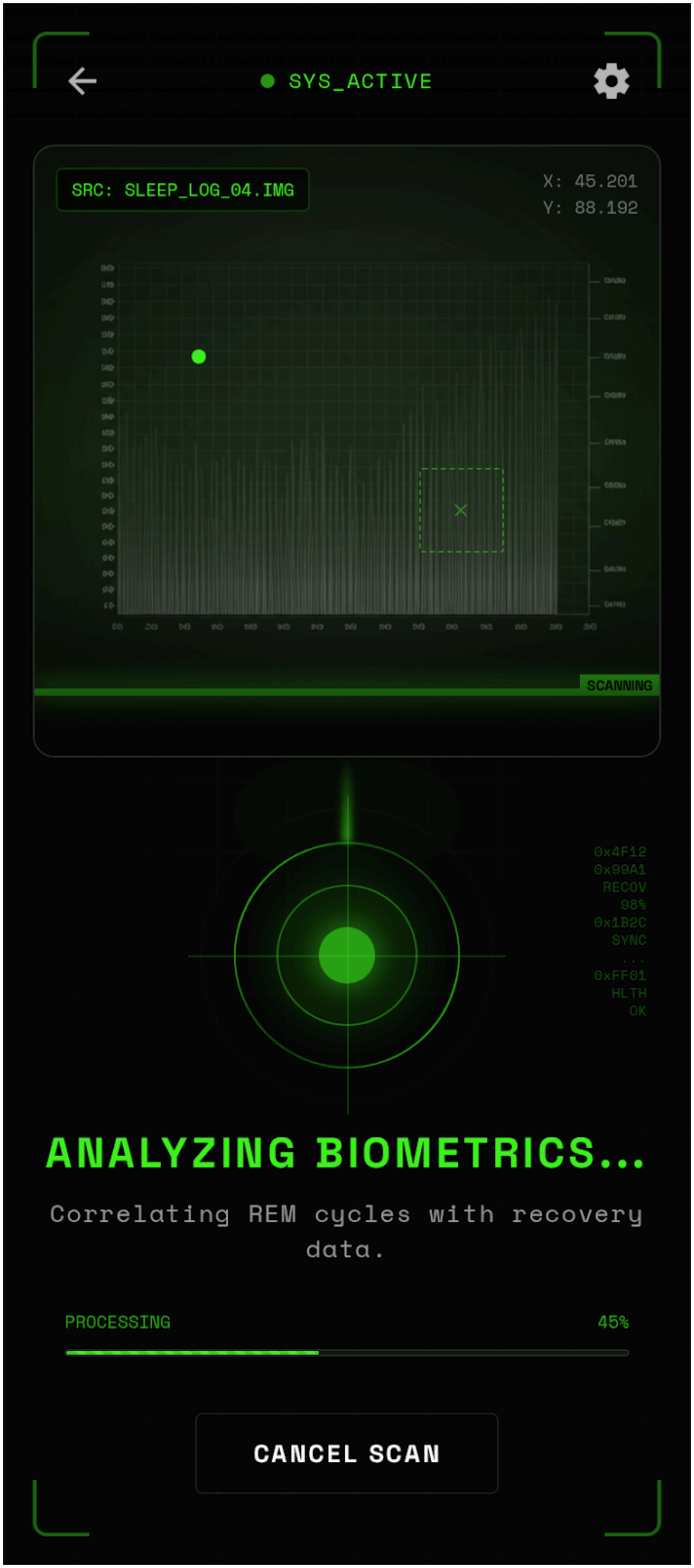


ANALYZING BIOMETRICS...

EXTRACTING SLEEP ARCHITECTURE

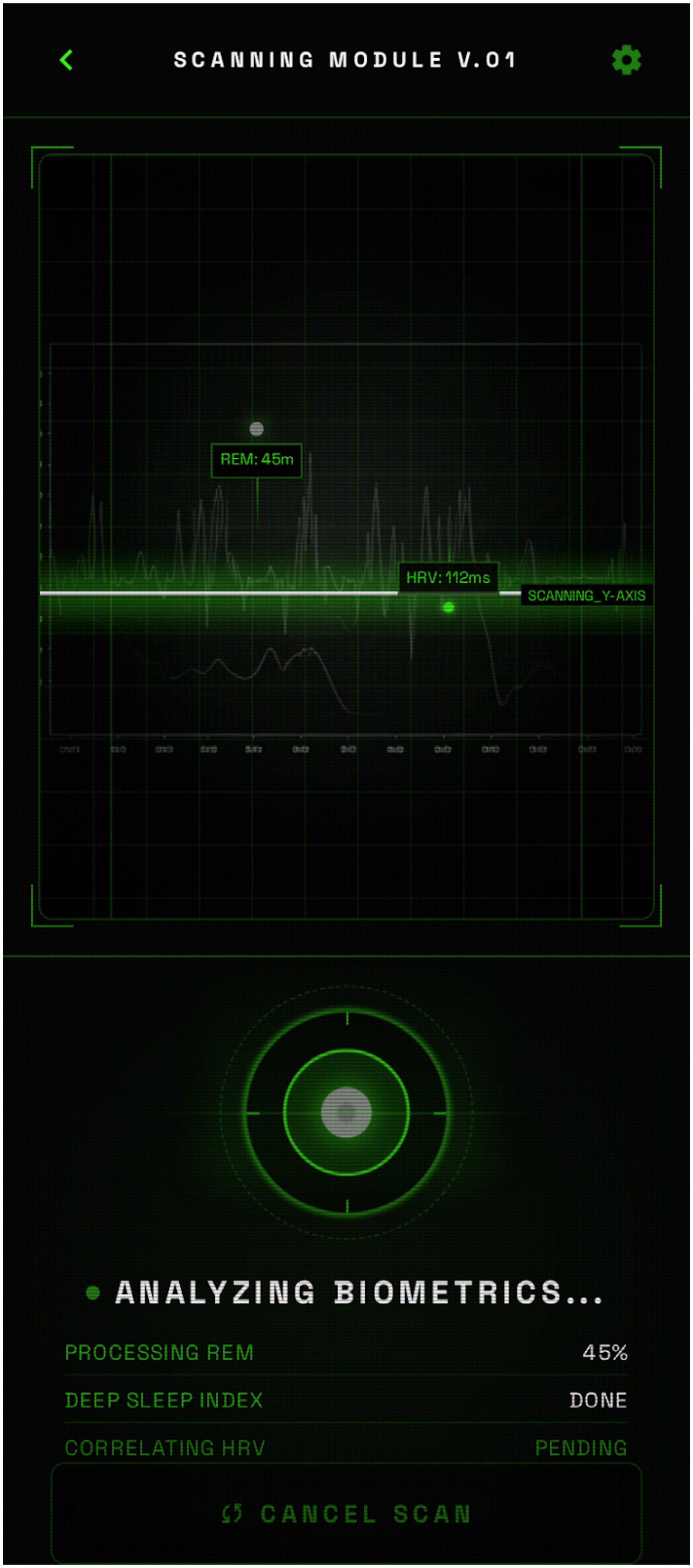
DECRYPTION_KEY_XA9 72%

```
   0x4A1F...OK | > PARSING LOG
NEURAL...SYNC | > COMPARING...
  DELTA...WAVE | > DETECTING
```



ANALYZING BIOMETRICS...

Correlating REM cycles with recovery data.

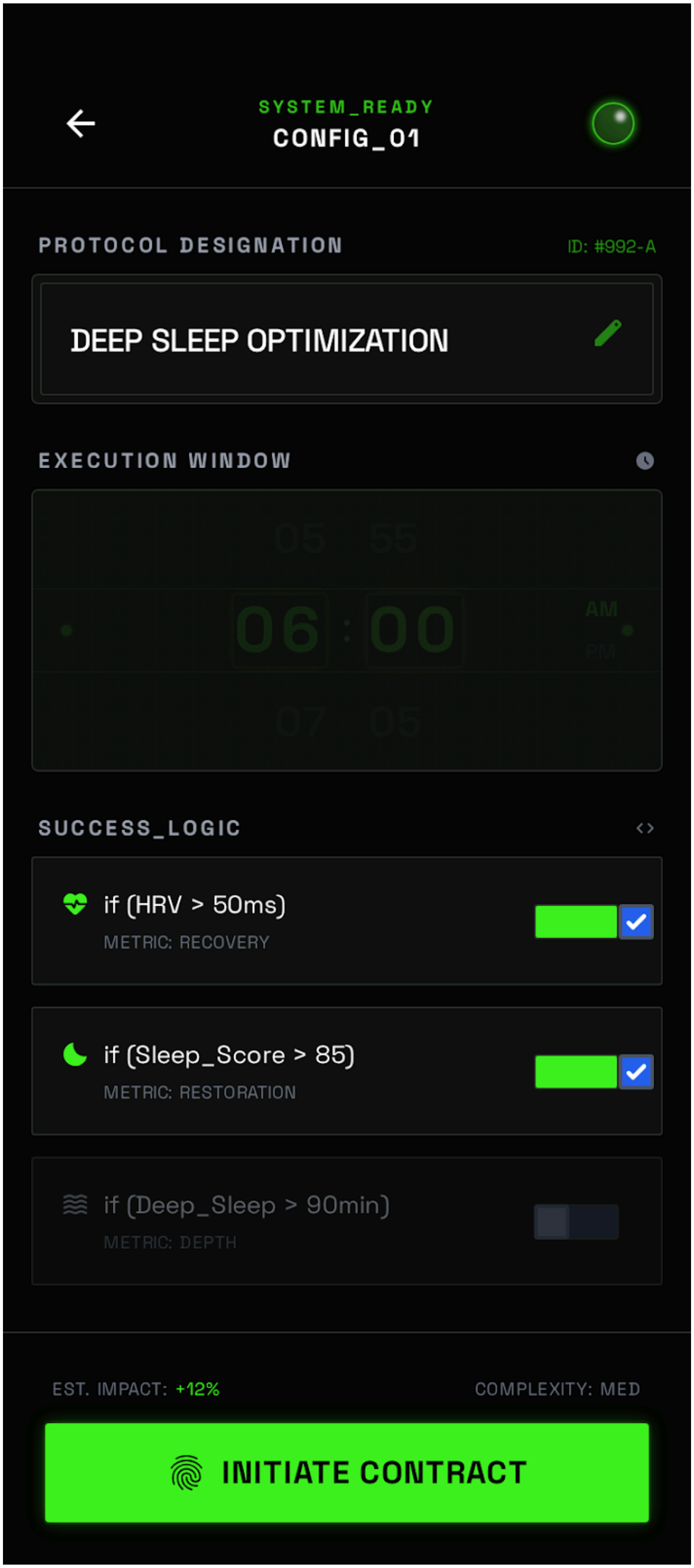


● ANALYZING BIOMETRICS...

PROCESSING REM 45%

DEEP SLEEP INDEX DONE

CORRELATING HRV PENDING



SYSTEM_READY

CONFIG_01

PROTOCOL DESIGNATION — ID: #992-A

DEEP SLEEP OPTIMIZATION

EXECUTION WINDOW

05 55

06 : 00 AM

PM

07 05

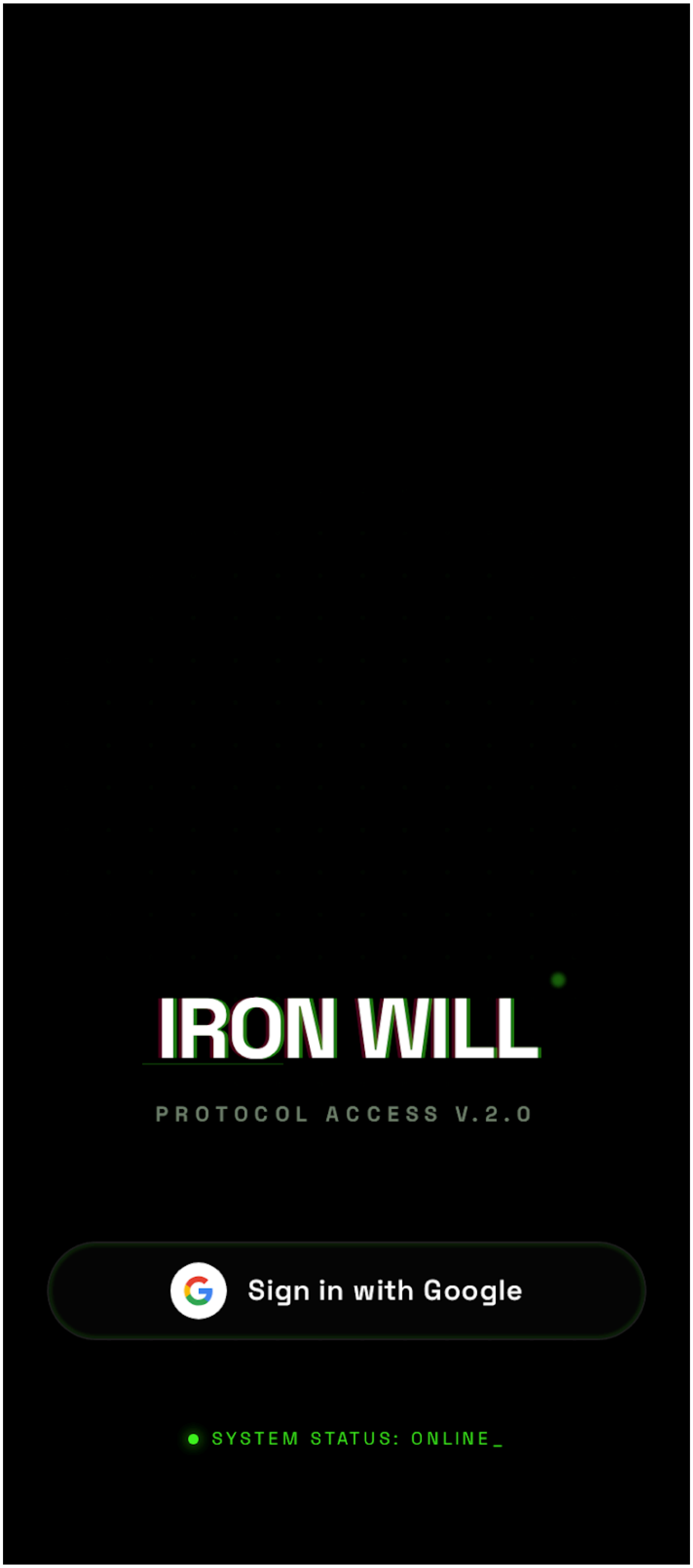
SUCCESS_LOGIC <>

if (HRV > 50ms) ☑
METRIC: RECOVERY

if (Sleep_Score > 85) ☑
METRIC: RESTORATION

if (Deep_Sleep > 90min) ○
METRIC: DEPTH

EST. IMPACT: +12% COMPLEXITY: MED



IRON WILL

PROTOCOL ACCESS V.2.0

● SYSTEM STATUS: ONLINE_