

# Edge-Centric Sovereignty: A Privacy-Preserving Framework for Generative AI in Digital Mental Health

*Fabrice Colozzi, Mindologue, Miami Beach, FL, USA, fabrice@mindologue.com*

**Abstract:** The rapid proliferation of Generative AI in mental healthcare offers unprecedented accessibility but introduces critical privacy risks regarding Data Sovereignty and Personal Identifiable Information (PII) leakage. Traditional cloud-centric architectures inherently expose sensitive user affect data, such as journal entries and art therapy outputs, to third-party inference risks. This paper proposes a Local-First, Hybrid-Storage Framework designed to align with the **US Executive Order** on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence [1]. Using a production-grade Multimodal Affective Computing Framework (MACF) as a case study, we demonstrate an architecture where immediate input sanitization and output filtration occur on the device (the "Edge"), ensuring that raw, sensitive data is de-identified before transmission. We detail a "Sanitization Layer" that strips non-standard characters client-side and a "**Just-in-Time**" (JIT) consent model that grants hardware permissions only at the moment of intent. Furthermore, we explore a cryptographic approach utilizing Presigned URLs and pseudonymized **Object Storage** to decouple user identity from stored media. By shifting the locus of control from centralized servers to the user's device through strict type-safety and local caching, this framework provides a blueprint for secure, scalable, and ethically compliant digital mental health interventions. This research argues that on-device inference and strict transport security are not merely technical preferences but ethical imperatives for the future of US digital healthcare infrastructure.

**Keywords:** *Generative AI, Data Sovereignty, Edge Computing, Privacy-Preserving AI, Digital Mental Health, Local-First Architecture, Zero Trust Security, On-Device Inference.*

## I. INTRODUCTION

### 1.1 The Privacy Paradox in Digital Health

The integration of Large Language Models (LLMs) and affective computing into mobile health applications represents a paradigm shift in the delivery of psychological support. Generative AI offers the potential to democratize access to mental wellness tools, providing 24/7 responsiveness that the traditional human-centric healthcare model cannot scale to meet. However, this technological leap creates a "Privacy Paradox": the very mechanisms that make AI effective, contextual awareness, long-term memory, and emotional analysis, require the processing of deeply intimate user data. In the current industry standard, often described as "Cloud-Default," this data is transmitted raw to centralized servers or third-party "Black Box" APIs, creating an expanded attack surface that threatens user privacy and erodes public trust.

### 1.2 Alignment with National Security Standards

The risks associated with cloud-centric mental health AI are not merely theoretical; they represent a significant concern for national information security. As outlined in the **United States Executive Order 14110** on the Safe, Secure, and Trustworthy Development and Use of

Artificial Intelligence [1], there is an urgent mandate to establish technical standards that protect Americans' privacy while fostering innovation. The Executive Order explicitly calls for the implementation of privacy-preserving technologies that mitigate the risks of data leakage and unauthorized surveillance. For digital mental health platforms to operate within this ethical and legal framework, reliance on standard encryption in transit is no longer sufficient. We must rethink the architecture of data flow itself.

### 1.3 The Edge-Centric Framework (ECF)

This paper introduces an "Edge-Centric Framework" (ECF) for Generative AI applications, utilizing a production-grade affective computing platform as a primary case study. Contrary to monolithic cloud architectures, ECF posits that the mobile device, the "Edge", must serve as the primary sovereignty boundary for sensitive data. By implementing a Hybrid Storage Model and a client-side Sanitization Layer, we demonstrate how applications can perform critical data scrubbing, input validation, and "Just-in-Time" (JIT) consent checks locally, before any data touches the network.

#### 1.4 Contributions Our contribution is threefold:

We define a Hybrid Sovereignty Architecture that utilizes a "Local-First, Cloud-Sync" strategy to balance data persistence with offline security.

We present a technical implementation of Client-Side Sanitization and Presigned URL workflows that prevent credential leakage and decouple user identity from stored media assets.

We propose a standard for JIT Consent in AI apps, arguing that permissions for sensory inputs (camera, microphone) must be inextricably linked to immediate user intent to satisfy ethical informed consent requirements.

By shifting the defense perimeter from the cloud to the device, this framework offers a roadmap for developing compliant, trustworthy AI systems that protect the digital civil rights of US consumers.

## II. REGULATORY CONTEXT & RELATED WORK

### The Regulatory Landscape

The deployment of Generative AI in healthcare operates within a complex regulatory landscape defined by tension between the data-hungry nature of Large Language Models (LLMs) and the strict minimization principles of privacy laws. In the United States, the Health Insurance Portability and Accountability Act (HIPAA) [4] mandates the protection of Protected Health Information (PHI), while the European Union's General Data Protection Regulation (GDPR) [5] enforces rights regarding data sovereignty and the "Right to be Forgotten." However, the emergence of "Black Box" AI APIs challenges these traditional frameworks. Unlike static database records, generative models require dynamic, context-heavy inputs, often containing unstructured affect data, to function effectively. This creates a high risk of "**Inadvertent Disclosure**", where users, lulled by the conversational nature of an AI, voluntarily divulge PII (such as names of family members or specific medical conditions) that is subsequently transmitted to third-party inference providers.

### 2.1 The "Cloud-Default" Failure Mode

The dominant architecture in current mobile AI development, which we define as the "Cloud-Default" model, exacerbates this risk. In this standard workflow, the mobile client acts merely as a thin display layer, transmitting raw user inputs (text, audio, or image) directly to centralized cloud servers for processing. While these transmissions are typically protected by Transport Layer

Security (TLS), the encryption protects only the data in transit, not the data at the point of inference.

Once the data reaches the cloud endpoint, it must be decrypted to be processed by the LLM or Text-to-Speech (TTS) engine. In this decrypted state, sensitive mental health data becomes vulnerable to:

1. **Logging and Retention:** Third-party API providers may retain prompt data for model training or debugging, potentially violating the "minimum necessary" standard of HIPAA.
2. **Man-in-the-Middle (MITM) at the Provider Level:** If the API provider's internal logging infrastructure is compromised, user data is exposed in plain text.
3. **Metadata Correlation:** Even if the text is sanitized, metadata such as IP addresses, device IDs, and timestamps can be correlated to de-anonymize a user, creating a "mosaic effect" of identification.

For mental health applications, where the stigma of diagnosis remains a significant societal concern, the "Cloud-Default" model represents an unacceptable fragility in the privacy chain.

### 2.2 The Shift to Local-First Architectures

In response to these vulnerabilities, the "Local-First" software paradigm has emerged as a critical alternative. Originally coined to describe software that prioritizes local storage and offline capability, Local-First architectures in the context of AI shift the "inference burden" from the cloud to the edge device. Research by Kleppmann et al. [2] suggests that local-first software enables Data Sovereignty by ensuring the user's device remains the primary source of truth.

However, purely local execution has historically been limited by the computational constraints of mobile hardware. Our proposed framework addresses this by adopting a **Hybrid Local-First approach**. We acknowledge that while massive parameter models (like GPT or Gemini) must essentially remain cloud-based, the *pre-processing, sanitization, and guardrail*ing of data **can and must** occur locally. By treating the cloud not as a default processor but as a "hostile environment" that receives only sanitized, strictly typed, and pseudonymized data, we align technical architecture with the **US Executive Order 14110** mandate for "privacy-preserving AI," ensuring that the benefits of advanced AI do not come at the cost of the American consumer's right to privacy.

### III. SYSTEM ARCHITECTURE: THE "EDGE-CENTRIC" FRAMEWORK

To address the vulnerabilities inherent in the "Cloud-Default" model, we propose the "**Edge-Centric Framework**" (ECF). This architecture is defined by a rigorous separation of concerns where the mobile device (the "Edge") assumes responsibility for data sanitization, guard-railing, and key management, while the cloud is relegated to the role of a stateless inference engine and encrypted storage locker. The ECF is built upon three technical pillars: Hybrid Storage Sovereignty, Client-Side Sanitization, and a "No-Keys" Credential Policy.

#### 3.1 Hybrid Storage & Data Sovereignty

Data sovereignty in mobile health applications is often treated as a binary choice: data resides either locally (secure but risk of loss) or in the cloud (backed up but exposed). The ECF implements a **Hybrid Storage Model** that distinguishes between "Hot Data" (immediate operational context) and "Cold Data" (long-term records).

##### **The "Cloud-First, Local-Cache" Strategy:**

While the encrypted backend serves as the ultimate source of truth for user journals and visual analysis results, the mobile client employs a sophisticated local caching strategy to maintain functional independence. Utilizing Native Preference Storage (mapping to UserDefaults on

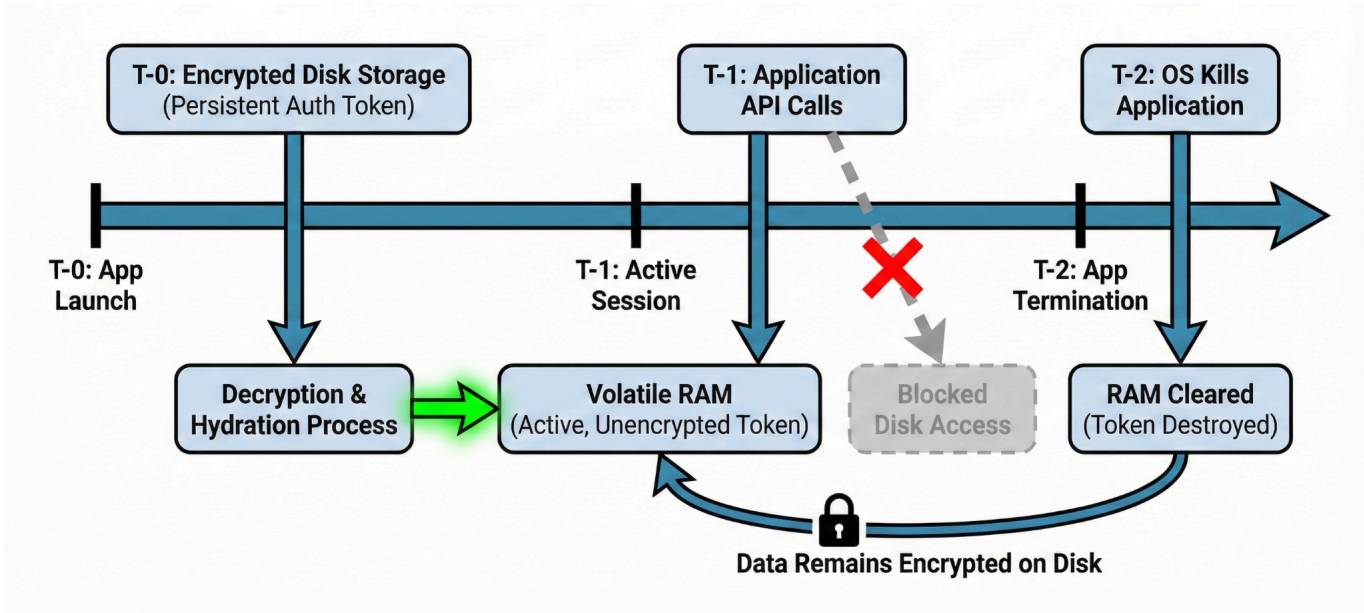
iOS), the application caches "Hot Keys", such as user settings and session state, locally. This allows the application to function in offline environments without querying the server, minimizing the frequency of network requests and reducing the metadata footprint visible to ISPs.

##### **The "Hydration" Security Pattern:**

A critical vulnerability in mobile apps is the persistence of authentication tokens in non-secure file systems. The framework addresses this via a "Hydration Pattern" implemented in the storage logic. Upon application startup, sensitive cryptographic material (specifically the `app_access_token`) is read from secure storage once and "hydrated" into a protected in-memory Random Access Memory (RAM) cache. Subsequent API calls read the token from volatile RAM rather than querying the disk. This approach serves a dual purpose:

1. **Performance:** It eliminates disk I/O latency during high-frequency WebSocket communications.
2. **Attack Surface Reduction:** By minimizing the frequency of disk reads, the window of opportunity for "side-channel" attacks or file-system snooping is significantly reduced. If the application is terminated, the RAM is cleared, and the unencrypted token ceases to exist in active memory.

**Figure 1:** The Volatile Memory Hydration Pattern



**Attack Surface Reduction:** Illustrating the secure lifecycle of the access token, which transitions from encrypted disk storage to transient RAM for active session use, minimizing the window of vulnerability to file-system scraping attacks.

**3.2 The Client-Side Sanitization Layer**

In a standard Generative AI flow, user input is sent directly to the model. In the ECF, we introduce an intermediary **"Sanitization Layer"** that sits strictly on the client device. This layer acts as a firewall for information, scrubbing data of non-semantic artifacts before it leaves the user's control.

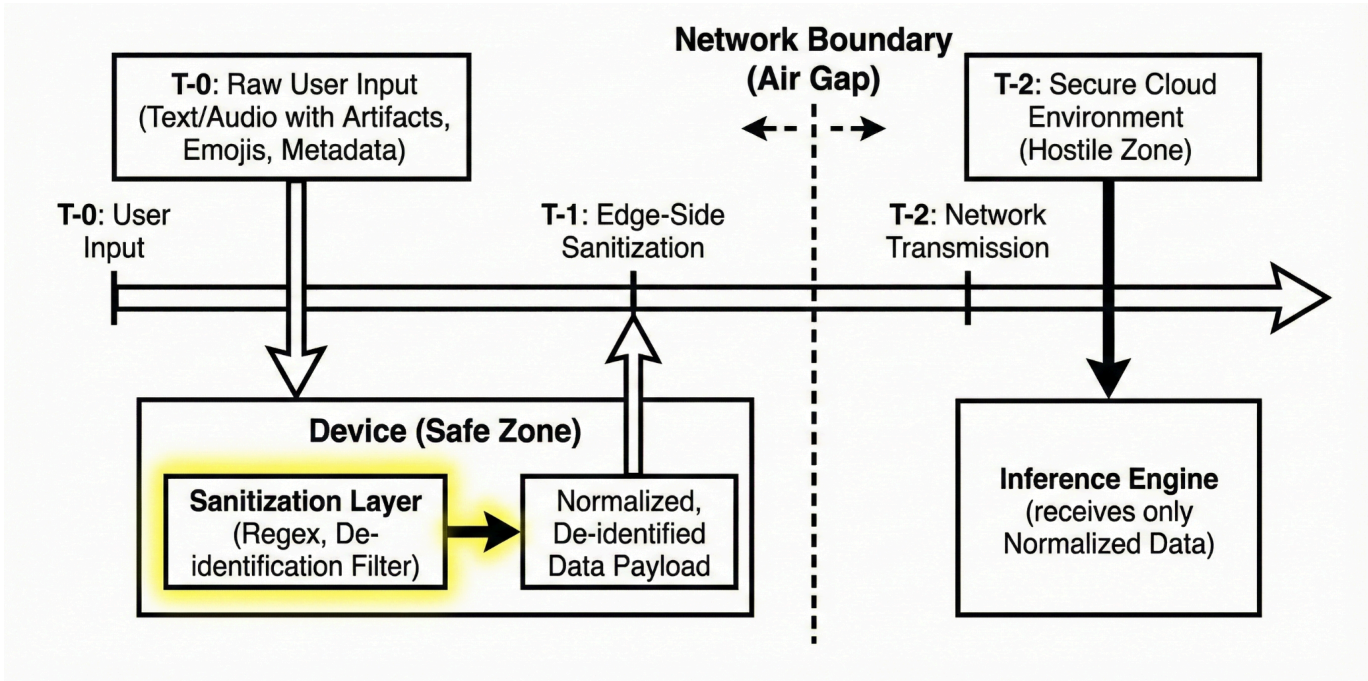
**De-Identification via sanitization\_layer:** The application implements a strict filtration logic, `sanitization_layer`, which processes all text designated for audio synthesis. Before the text payload is transmitted to the inference engine, this function strips emojis, non-standard Unicode symbols, and complex formatting characters (e.g., Markdown syntax, guillemets). While functionally designed to ensure high-quality, human-like audio output, this mechanism serves a vital privacy function: **Artifact De-Identification**. Non-textual symbols often carry metadata or unique fingerprinting characteristics that could be used to re-identify a specific user's writing style. By reducing the input to standard

alphanumeric text *on the device*, the system enforces a normalization protocol that makes user inputs generic and indistinguishable from a data-structure perspective.

**Input/Output Guardrails & Type-Safety:**

To prevent "Prompt Injection" attacks, where malicious actors attempt to manipulate the AI into revealing system instructions or generating toxic content, the ECF enforces **Strict Type-Safety** on all WebSocket connections (Strictly-Typed WebSocket Channels). Unlike REST APIs that may accept broad JSON blobs, the WebSocket parsers enforce rigid payload types (e.g., `message_start`, `message_chunk`). Any payload that does not strictly conform to these schema definitions is rejected by the client-side parser before rendering. This ensures that even if the remote AI model hallucinates or is compromised, the client device will refuse to process or display malformed or dangerous instructions.

**Figure 2:** The Data Sovereignty Boundary Caption



**Edge-Side Filtration:** Visualizing the "Sanitization Layer" where text and audio artifacts are scrubbed locally on the device, ensuring only normalized, de-identified data traverses the unsecure network layer to the cloud.

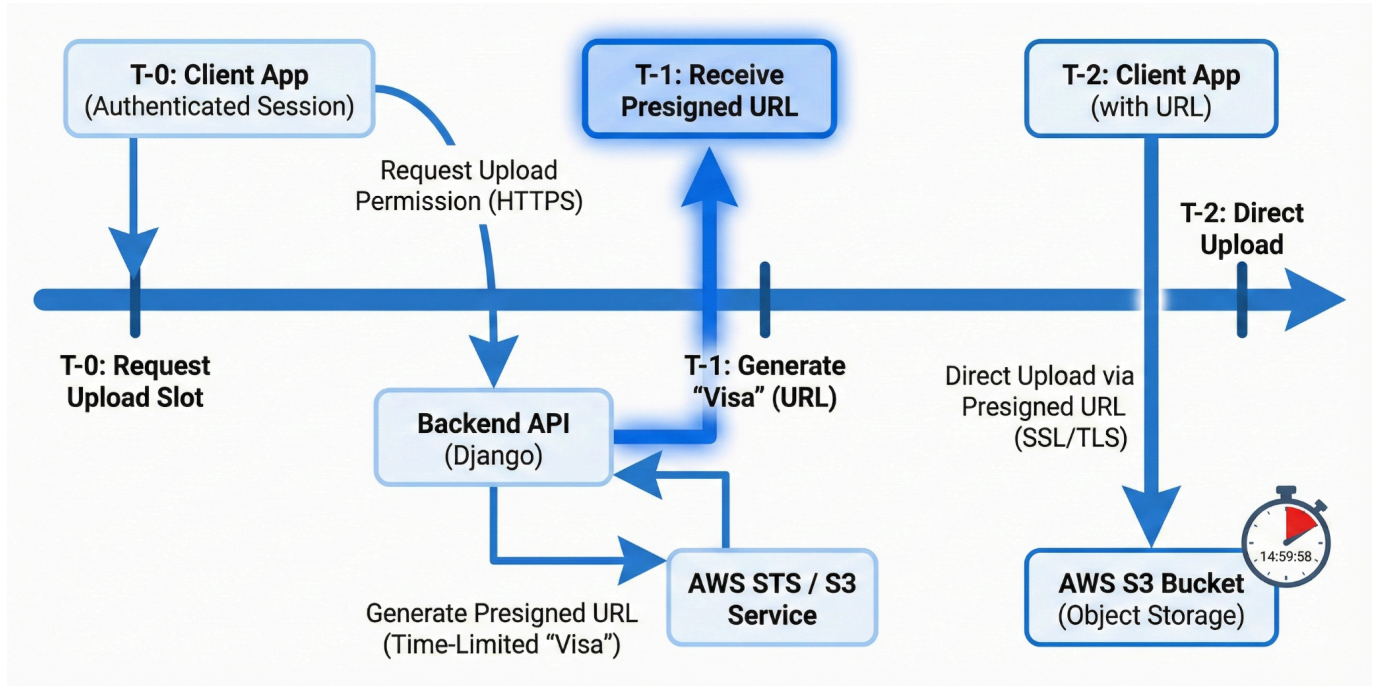
**3.3 Secure Transport & The "No-Keys" Policy**

A pervasive security flaw in mobile development is the hard-coding of cloud credentials (e.g., AWS Secret Keys) within the application binary. If an attacker reverse-engineers the app, they can extract these keys and access the entire user database. The ECF eliminates this risk through a **"No-Keys" Policy** using **Presigned URLs**.

**Ephemeral Access Control:** In the proposed architecture, the client application possesses zero permanent credentials

for the object storage service. When a user needs to upload a drawing or sync a meditation audio file, the client requests a temporary "upload slot" from the backend authority. The backend validates the request and generates a time-limited **Presigned URL**, a cryptographic "visa" that grants permission to upload a specific file to a specific location for a short window (e.g., 5 minutes).

**Figure 3:** The Ephemeral Credential Workflow



**Caption:** Zero-Trust Architecture [3]: A sequence diagram illustrating the "No-Keys" policy, where backend-generated Presigned URLs grant temporary upload authority (a "Visa") without ever exposing root AWS credentials ("the Passport") to the client application.

1. **Request:** Client authenticates via HTTPS and requests upload permission.
2. **Authorization:** Backend verifies User UUID and generates the signature.
3. **Direct Transmission:** The client uploads the file directly to the storage bucket using the Presigned URL.
4. **Decoupling:** Because the file flows directly from Client to Storage (bypassing the application server's bandwidth) but is authorized by the Backend, the architecture ensures efficiency without sacrificing security.

Crucially, this means that even if a user's device is fully compromised, there are no long-lived administrative credentials to steal. The attacker would only find an expired URL. Furthermore, all transport occurs over SSL/TLS 1.3 encrypted channels, and real-time streams utilize **Secure WebSockets (wss://)**, ensuring that data in transit is opaque to network sniffers.

## IV. USER AGENCY: DESIGNING FOR TRUST

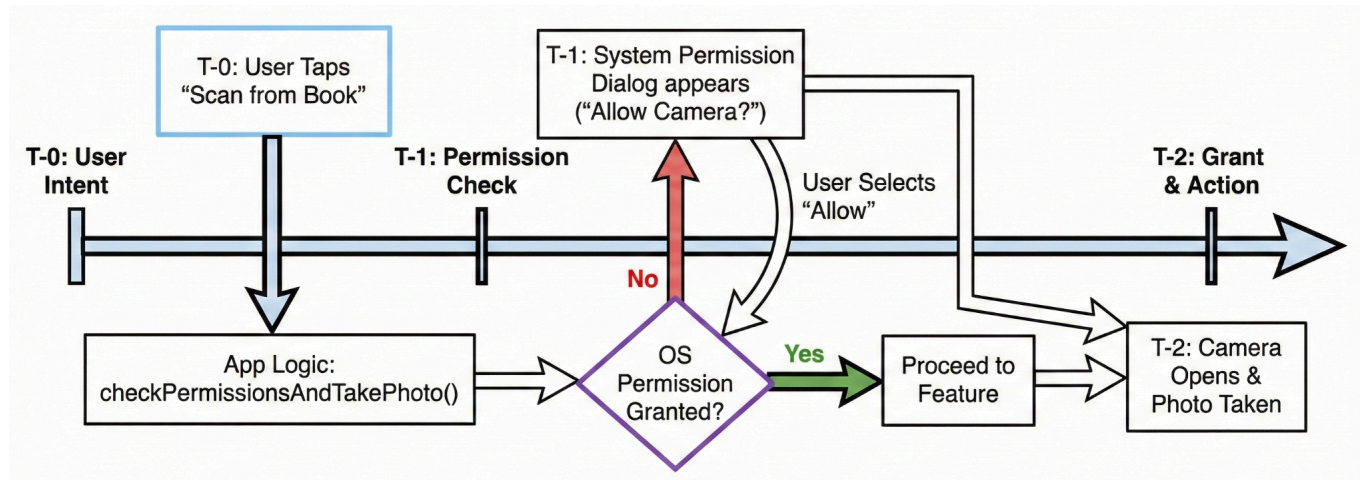
Technical security measures, no matter how robust, are insufficient if the user interface (UI) manipulates users into surrendering more data than necessary, a practice known as "Dark Patterns." The **Edge-Centric Framework (ECF)** posits that true privacy preservation requires not just encryption, but User Agency: the explicit, informed, and granular control over one's own data. We implement this through **Just-in-Time (JIT)** Consent and Pseudonymity by Design.

### 4.1 Just-in-Time (JIT) Consent

Standard mobile development practices often encourage "Permission Grabbing" at application launch, where apps request access to the camera, microphone, and photo library immediately upon first opening. This creates "Consent Fatigue," conditioning users to blindly tap "Allow" without understanding the context or necessity of the request.

The framework rejects this model in favor of **Just-in-Time (JIT) Consent**. In the ECF, permissions are never requested in the abstract; they are requested only in strict correlation with a user-initiated action.

**Figure 4:** The Intent-Based Consent Loop Caption



**Just-in-Time Authorization:** A decision tree contrasting standard "permission-grabbing" against the proposed framework, demonstrating how hardware access (Camera/Mic) is strictly coupled to immediate user intent to satisfy ethical consent requirements.

### Case Study:

The Permission Verification Flow Consider the workflow for analyzing a physical drawing. The application does not hold camera permissions by default.

1. **Intent:** The user actively taps "Scan from Book" or "Capture Image."
2. **Logic Check:** The internal logic (verifyPermissionsAndCapture) verifies if permission has been granted previously.
3. **Request:** Only if permission is missing does the OS-level dialog appear.
4. **Graceful Degradation:** If the user denies permission, the app does not crash or block other features; it simply logs the denial and disables that specific button.

This workflow ensures that the context of the data collection is clear. The user understands, "I am giving access to my camera right now, because I want to scan this specific image". This temporal linkage between intent and consent is the gold standard for ethical AI interaction, as it respects the user's autonomy and minimizes "permission creep".

## 4.2 Pseudonymity by Design

In mental health applications, the leakage of identity is often more damaging than the leakage of data. To mitigate this, the ECF employs a strategy of **Pseudonymity by Design**, effectively decoupling the user's real-world identity from their digital artifacts.

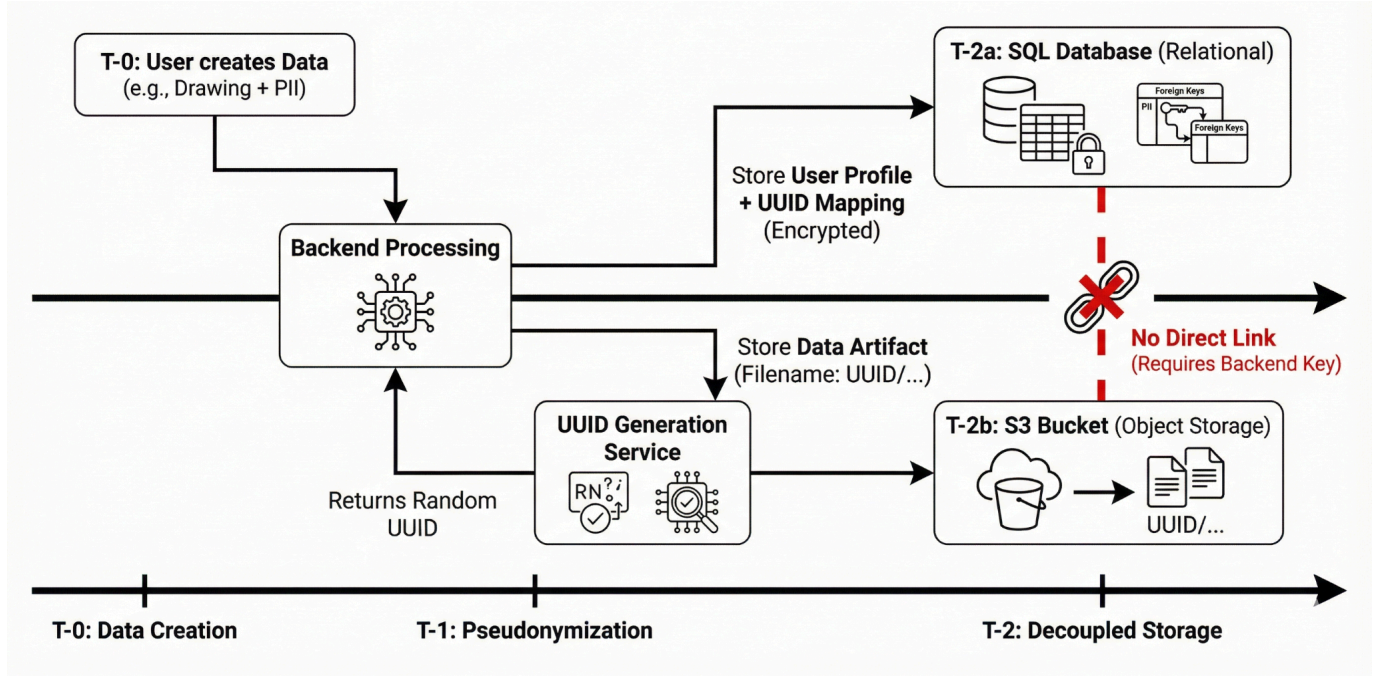
**Guest Mode Isolation:** The application supports a fully functional "Guest Mode," where users are tracked solely via a unique deviceId and an optional firstName. No email, phone number, or social login is required to access the core AI features. This allows users to test the efficacy of the mental health tools without creating a permanent, identifiable record in the database. Upgrading to a full account is an explicit, opt-in action, not a requirement for entry.

**Storage Namespacing & Data Decoupling:** On the backend, this pseudonymity is enforced structurally. Object storage keys are namespaced using random User UUIDs (e.g., user-uuid/relaxation\_tts/audio\_file.m4a) rather than semantic filenames containing names or dates. This creates a "User-Media Decoupling" effect for any potential attacker.

If the storage bucket were to be leaked, the attacker would find a collection of untraceable audio files and drawings. Without access to the separate, encrypted SQL database that maps UUIDs to user profiles, the media files are orphaned and effectively anonymous. This architectural

separation ensures that a breach in one part of the system (the storage layer) does not automatically result in a breach of privacy (identity exposure).

**Figure 5: Pseudonymous Data Decoupling Caption**



**Identity Isolation:** The structural separation of the Relational User Database (SQL) from Object Storage, utilizing random UUID namespaces to prevent metadata correlation and ensure that leaked media assets cannot be traced back to a specific user identity

## V. THREAT MODELING & RESILIENCE ANALYSIS

To validate the robustness of the Edge-Centric Framework, we subject the architecture to a "Red Team" analysis, evaluating its resilience against common attack vectors in the mobile threat landscape.

### 5.1 Attack Vector: Device Compromise & Theft

- **Scenario:** A malicious actor gains physical possession of a user's unlocked device.
- **Mitigation:** The "Hydration Pattern" serves as a critical fail-safe. Because the `access_token` is held in volatile RAM after the initial load (and not continuously read from a persistent text file), the surface area for extracting credentials via file-system dumping is minimized. Furthermore, the application enforces strict session timeouts. If the app is killed or the OS suspends the background process, the RAM is

cleared, and the token is effectively destroyed, forcing a re-authentication that the attacker cannot complete without the primary credentials.

### 5.2 Attack Vector: Network Sniffing (Man in the Middle)

- **Scenario:** A user connects to the application via an unsecured public Wi-Fi network, and an attacker attempts to intercept the data stream.
- **Mitigation:** Beyond standard TLS 1.3 encryption, the Cryptographic Signature architecture renders interception futile. Even if an attacker sniffs the upload URL, the signature is time-bounded (typically expiring within minutes) and scoped strictly to a specific resource path. It cannot be reused to access other files or upload malicious payloads elsewhere. Additionally, real-time drawing analysis occurs over Secure WebSockets (`wss://`), which establishes a persistent, encrypted tunnel that resists standard HTTP packet inspection tools.

### 5.3 Attack Vector: Prompt Injection & Malicious Payloads

- **Scenario:** A sophisticated user attempts to inject executable code or jailbreak prompts into the journal interface to manipulate the backend AI.
- **Mitigation:** The Strict Type-Safety logic on the client side acts as a pre-emptive shield. The WebSocket parsers do not blindly accept strings; they require payloads to match defined schemas (`message_start`, `message_chunk`). Malformed JSON or payloads containing unexpected control characters are rejected by the client's parsing logic before they are ever rendered or passed to the TTS engine. This effectively neutralizes injection attacks by enforcing an "allow-list" of valid data structures.

## VI. DISCUSSION & FUTURE ROADMAP

The transition from Cloud-Default to Edge-Centric architectures is not without its trade-offs. We must balance the imperative of privacy against the constraints of mobile performance.

### 6.1 The Latency vs. Sovereignty Trade-off

Implementing client-side sanitization and strict token management introduces a marginal latency overhead. The sanitization logic and the cryptographic handshake for authentication add milliseconds to the user interaction loop. However, we argue that this latency is a negligible cost for the assurance of Data Sovereignty. In the context of mental health, where trust is the primary currency, users consistently prefer a secure, private interaction over instantaneous but insecure data transmission.

### 6.2 The Path to Fully Local Inference

The current iteration of the Edge-Centric Framework is a Hybrid solution: sanitization occurs on the device, while heavy inference (LLM processing) occurs in the cloud. The ultimate goal of our research roadmap is **Zero-Knowledge Architecture** via fully on-device inference.

Advancements in mobile hardware, specifically the Neural Processing Units (NPUs) in modern mobile architectures (iOS and Android), are making this viable. Our future development focuses on porting optimized, quantized models directly to the application bundle:

- **Text-to-Speech:** Replacing cloud APIs with high-fidelity, on-device neural models, which can generate emotional speech locally without internet access.
- **Generative Text:** Utilizing 4-bit quantized versions of Small Language Models (SLMs) such as emerging open-weights architectures.

By running these models entirely on the user's hardware, we aim to eliminate the need for cloud data transmission entirely for core interactions. This would transform the platform into a "Data Bunker," where the user's drawing analysis and journal reflections effectively never leave the device, satisfying the most stringent interpretations of the US Executive Order on AI Safety.

## VII. CONCLUSION

### The Industry Imperative

As Generative AI becomes an intrinsic component of the US healthcare infrastructure, the industry faces a critical choice: continue with "**Cloud-Default**" architectures that centralize risk, or adopt "**Edge-Centric**" frameworks that distribute and neutralize it. This paper has argued that for high-stakes domains like digital mental health, the latter is not just preferable, it is imperative.

### Proof of Concept & Contributions

Using the **Multimodal Affective Computing Framework** (MACF) as a working proof-of-concept, we have demonstrated that Data Sovereignty can be achieved without sacrificing the transformative potential of AI. By implementing a Hybrid Storage Model that isolates sensitive authentication tokens in volatile memory, a Client-Side Sanitization Layer that de-identifies artifacts before transmission, and a "No-Keys" Policy that prevents credential leakage, we have outlined a robust defense against the most common vectors of data compromise. Furthermore, by enforcing Just-in-Time consent, we realign the user interface with ethical principles, ensuring that data collection is always a function of explicit user intent.

### National Impact

This "**Privacy by Architecture**" approach directly supports the objectives of **Executive Order 14110**, safeguarding the privacy of American consumers while fostering a trustworthy environment for AI innovation. Ultimately, the security of our digital mental health tools will define the public's willingness to adopt them. The Edge-Centric Framework provides the necessary foundation for that trust.

## REFERENCES

[1] **The White House.** (2023). *Executive Order 14110 on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence.*

[2] **Kleppmann, M., et al.** (2019). *Local-First Software: You Own Your Data, in spite of the Cloud.* Onward! 2019 Proceedings.

[3] **NIST.** (2020). *Zero Trust Architecture (ZTA) (NIST Special Publication 800-207).* National Institute of Standards and Technology.

[4] **U.S. Department of Health and Human Services.** *Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule.*

[5] **Voigt, P., & Von dem Bussche, A.** (2017). *The General Data Protection Regulation (GDPR): A Practical Guide.* Springer.

## BIOGRAPHY

**Fabrice Colozzi** is the **Lead Research Architect of a dedicated mental health AI mobile platform** based in Florida. A former Space Engineer with a Master of Science in Engineering from the École Nationale d'Ingénieurs de Tarbes (ENIT), France, Mr. Colozzi has spent over two decades architecting high-reliability systems where failure is not an option, from orbital propulsion to financial data reporting.

Transitioning from aerospace to the digital health frontier, Mr. Colozzi now pioneers "**Trustworthy Edge AI.**" His work addresses the "Privacy Paradox" of modern healthcare: how to leverage the immense power of Generative AI without exposing sensitive patient data to the vulnerabilities of the cloud. He is the architect of a **modern mental health Framework**, a novel software ecosystem that shifts heavy neural inference mostly to consumer mobile devices.

His research focuses on solving the thermodynamic and computational constraints of running "server-grade" AI and TTS on smartphones. By treating mental health software with the same rigor as critical flight systems, Mr. Colozzi is defining a new standard for **Sustainable, Private, and Resilient Digital Infrastructure.**