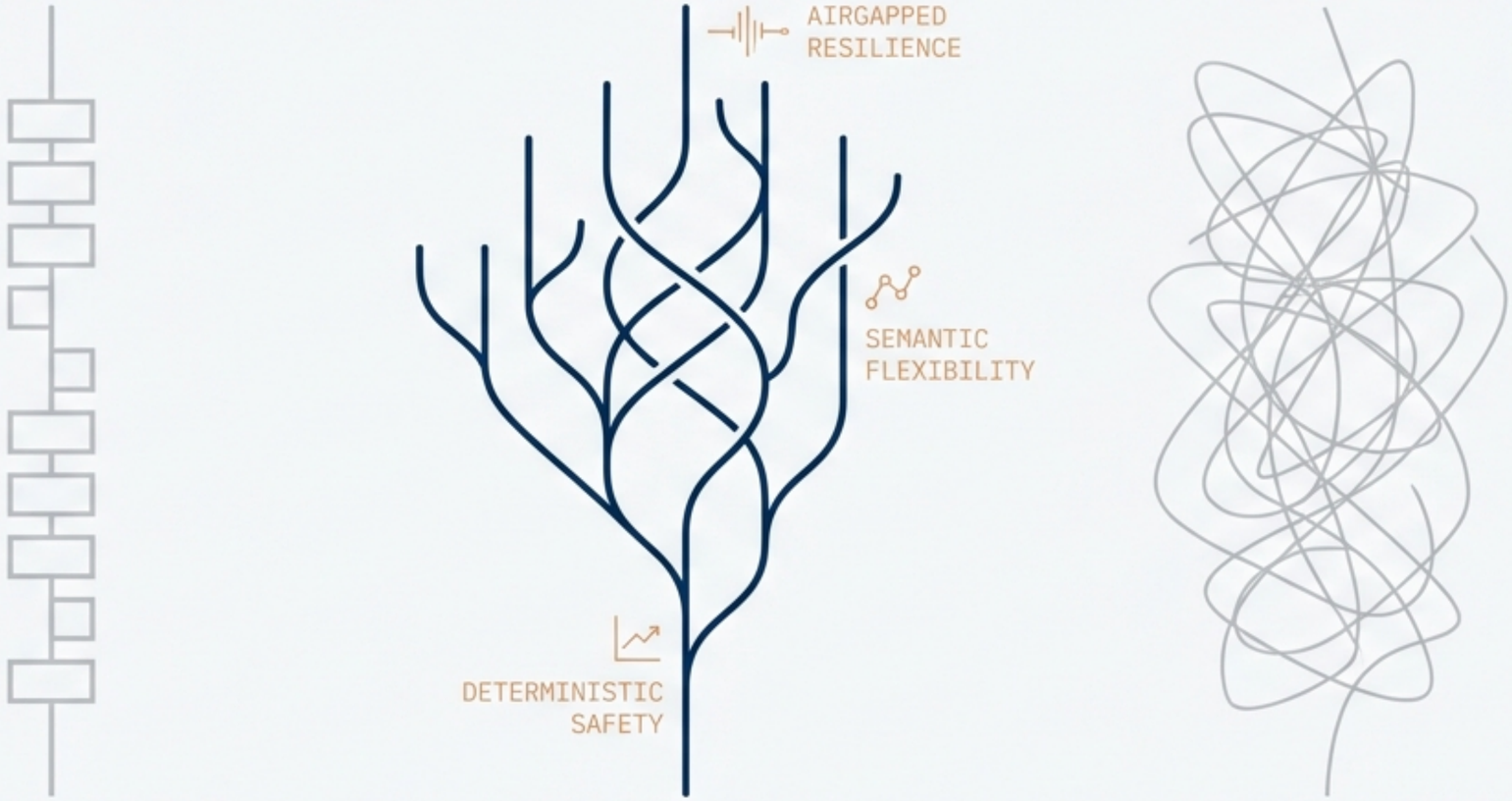
The Third Way: Engineering Resilience for an Uncertain World.

An airgapped, offline-first emergency application architecture that delivers deterministic safety and semantic flexibility



STATIC CONTENT | **THE THIRD WAY** | **GENERATIVE AI**

The Modern Resilience Paradox

Cloud-centric architectures are inherently fragile, failing precisely when they are needed most: during grid failures, network outages, and natural disasters. The prevailing solutions for offline information are a choice between two failures.

The Old Way (Static Content)

Inflexible, difficult to search under duress, and fails to provide guided, context-aware assistance.

“Inadequate under the cognitive load of a crisis.”

The Risky Way (Generative AI)

Prone to factual hallucination (a potential fatality in medical contexts), non-deterministic, and imposes severe power penalties on battery-constrained devices.

Our Solution: A Hybrid Intelligence Architecture.

We reject the false choice between static and generative.

Our approach combines the **deterministic safety** of state machines with the semantic flexibility of neural networks.

We retrieve **verified information** and **guide** the user through **pre-validated logic**, we **do not generate new, potentially flawed, advice**.

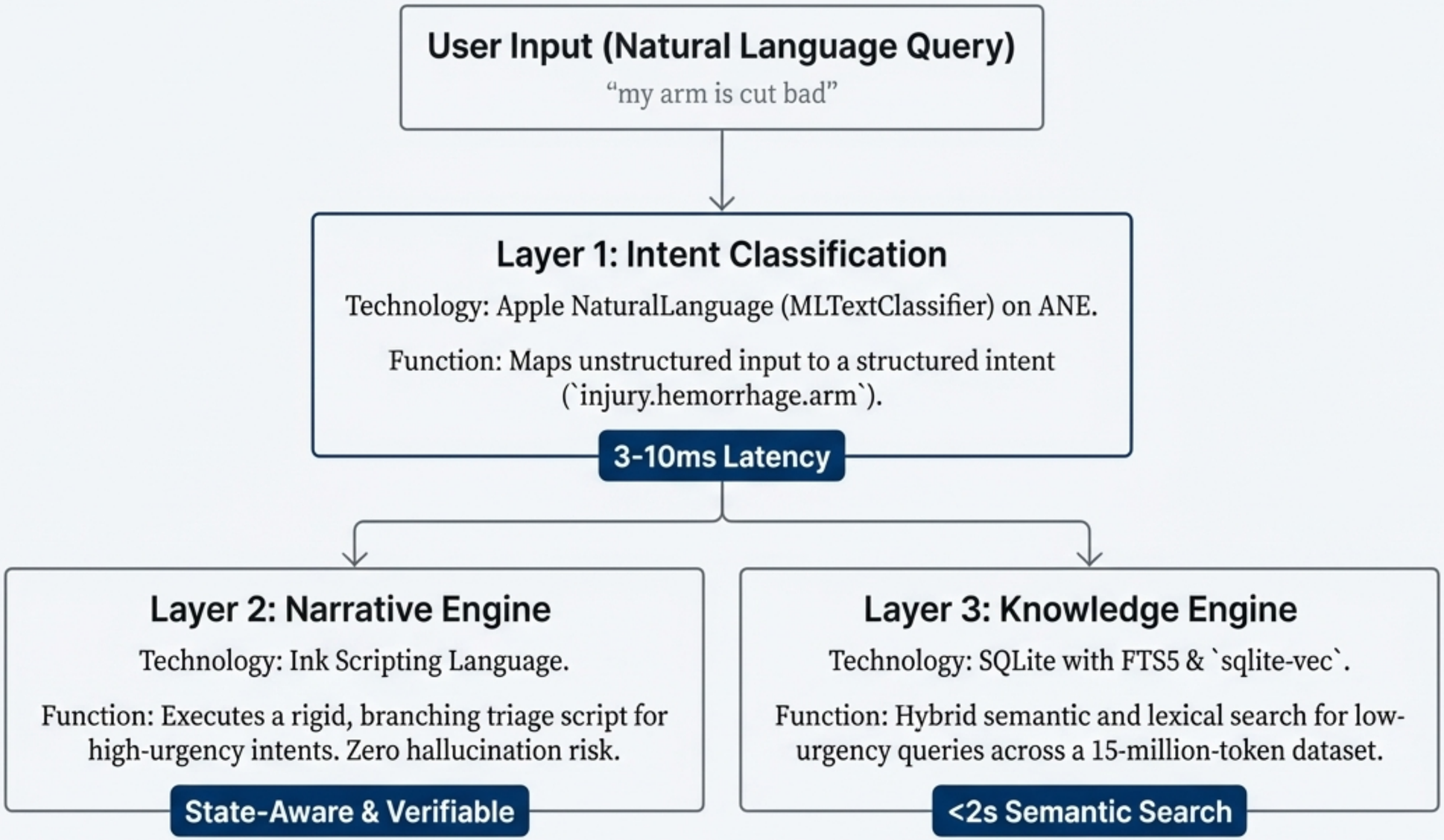
Layer 1: The Intent Router
(Acts as the Triage Nurse)
Fast & Instinctive

Layer 2: The Narrative Engine
(Acts as the Specialist Clinician)
Guided & Deterministic

Layer 3: The Knowledge Engine
(Acts as the Reference Library)
Deep & Comprehensive

This isn’t Retrieval-Augmented Generation.
It’s **Retrieval and Guided Execution.**

The Three Layers of an Airgapped Intelligence.



Layer 2: Deterministic Triage via The Narrative Engine

Medical protocols are algorithms—Finite State Machines. In a crisis, guidance must be predictable and verifiable. Generative models are probabilistic and can skip steps or 'forget' state. Ink provides a human-readable, auditable scripting layer for this deterministic logic.

Ink Script Example (`breathing_emergency`)

```
=== breathing_emergency ===
I understand you're having trouble breathing.
Let's figure out how serious this is.

* [I can barely breathe or speak]
  ~ severity = "critical"
  -> call_999_immediately

* [Breathing is difficult but I can talk]
  ~ severity = "moderate"
  -> moderate_assessment

* [I'm worried about someone else's breathing]
  -> assess_other_person
```

Zero Hallucination

Compiled JSON is deterministic; the same inputs always produce the same advice.

State Persistence

The engine tracks conversation state (e.g., `has_tourniquet`), allowing for context-aware instructions later in the flow.

Sensor Fusion

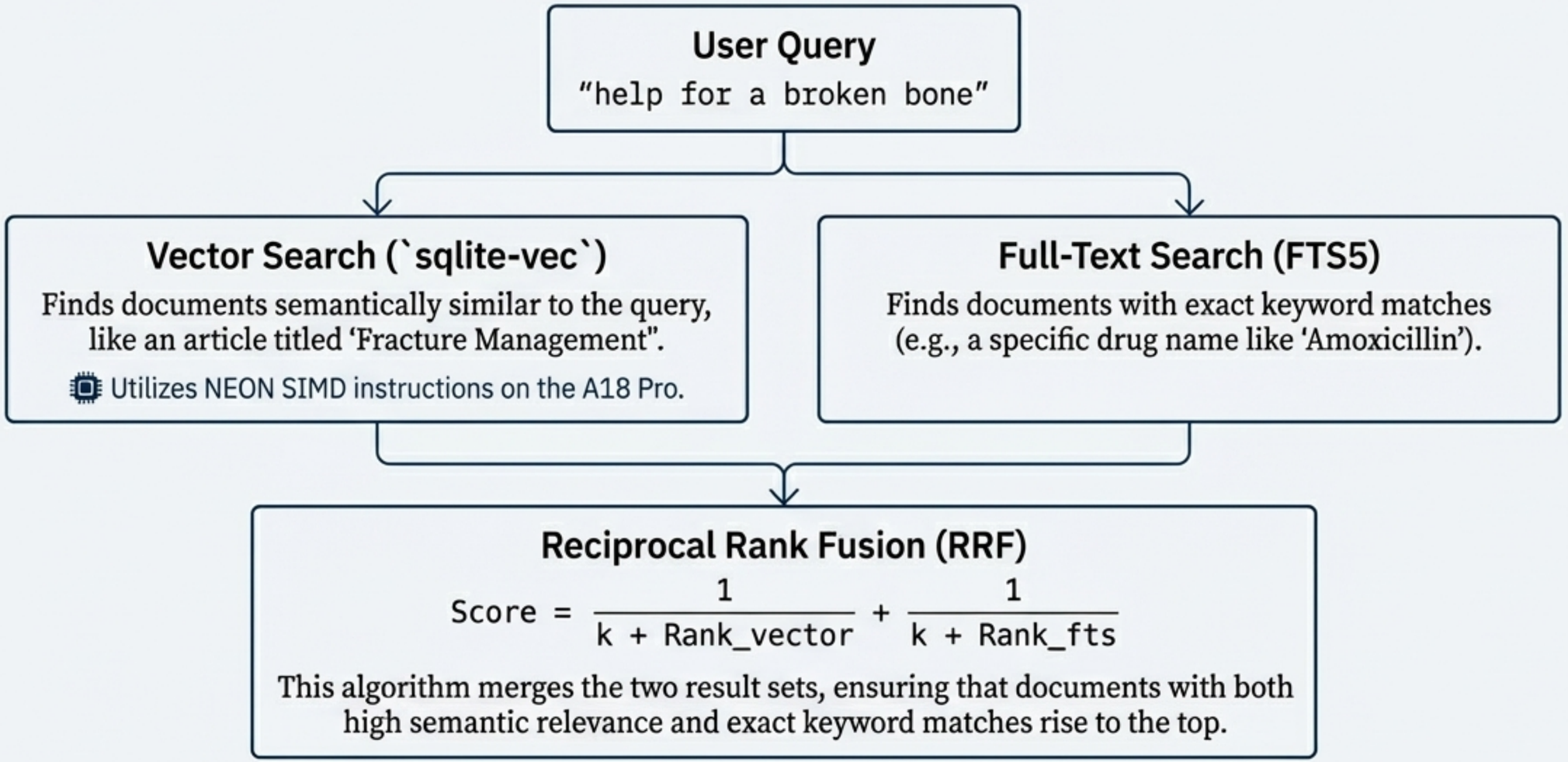
Ink variables can be bound to iOS sensors. A code example:

```
Assuming safe location? {is_moving_fast:
WARNING: You appear to be moving. Secure
yourself first.}
```

...where `is_moving_fast` is set by CoreLocation.

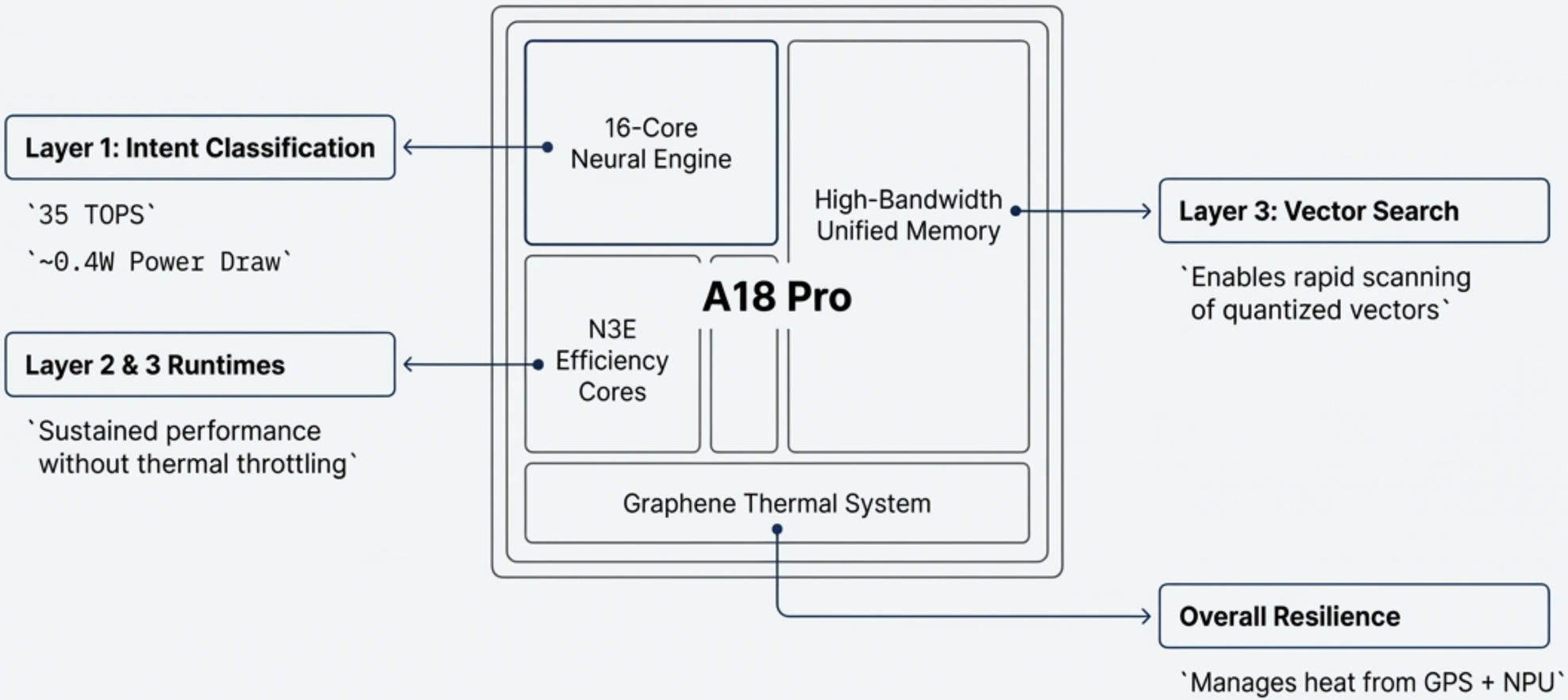
Layer 3: Hybrid Search via The Knowledge Engine.

For broad information retrieval, users need to search by meaning, not just keywords. We use `sqlite-vec` to bring vector search on-device without external dependencies, and combine it with FTS5 for precision.



This Isn’t Just Software. It’s Hardware-Software Symbiosis.

The feasibility of this intelligent offline architecture is inextricably linked to the specific capabilities of the Apple A18 Pro silicon. The design turns the platform’s constraints (power, thermal) into advantages through deliberate optimization.



Exploiting the A18 Pro's Architectural Advantages.

The Neural Engine (ANE)

`3-10ms` latency for intent classification.

`~0.4W` power draw during inference vs. 3-5W on CPU/GPU.

Instantaneous response crucial in panic situations with negligible battery impact.

N3E Process & Efficiency Cores

We avoid the 'race-to-sleep' strategy, which is suboptimal for sustained interaction.

By targeting the high-efficiency cores, we minimize thermal output, preserving screen brightness and battery life under stress.

Unified Memory Architecture

`17%` increase in memory bandwidth.

Eliminates costly copy operations between CPU and ANE, vital for feeding data to the vector search and re-ranking algorithms.

Thermal-Aware Degradation Protocol

The app monitors `ProcessInfo.thermalState`.

Nominal	→	Fair/Serious	→	Critical
Full hybrid search.		Disable vector search, fallback to FTS5 only.		Suspend map renderer, switch to high-contrast text-only UI.

De-Risking Execution Through Principled Decisions.

A resilient architecture is not enough. We have proactively identified and solved the three most significant non-technical challenges that could derail the project: content licensing, UX validation, and the quality of guided interactions.

Decision 1: Content Licensing

The Problem: The conflict between NHS UK-only content and our ‘use anywhere’ value proposition.

Decision 2: NL-to-Lookup UX

The Hypothesis: Can a simple classifier reliably map panicked user queries queries to the correct intent without causing frustration?

Decision 3: Conversation Quality

The Risk: Poorly designed conversation trees feel robotic and fail under the stress of a real emergency.

Solving the Licensing Paradox with an Open Data Stack

Problem Recap
NHS content requires geographic verification, which fundamentally breaks the airgapped design principle.

Recommended Path
We bypass this by creating a superior, globally-licensed content bundle.

Recommended Content Bundle (~300–450MB)				
Content Type	**Source**	**License**	**Rationale**	**Size**
Medical Reference	WikiProject Medicine (ZIM)	CC-BY-SA	High-quality, globally licensed alternative to NHS content.	~200MB
Legal Rights	legislation.gov.uk	OGL v3.0	Authoritative UK emergency legislation (e.g., Civil Contingencies Act).	~15MB
Mapping	OS VectorMap District via OpenMapTiles	OGL v3.0	Detailed, 1:25,000 scale UK vector map data.	~150MB
Emergency Guidance	OpenWHO, CDC Protocols	CC-BY-NC, Public Domain	Clinically validated international emergency protocols.	~20MB

Validating the Core Experience with Rapid Prototyping

Before committing to the full architecture, we run short, focused experiments to validate our most critical UX hypotheses.

Section 1: The Weekend Spike: NL-to-Lookup Validation

Goal

Prove MLTextClassifier is "smart enough" for emergency queries.

Process

Train on 20-30 intents with synthetic data, including ambiguous phrases ("I feel weird").

Success Criteria

- **Accuracy:** **>95%** (test set), **>85%** (ambiguous queries)
- **Fallback Rate:** **<15%** of queries require manual search
- **User Rating:** **>4/5** on "I would trust this in an emergency."

Section 2: The One-Week Sprint: Conversation Quality

Goal

Ensure triage flows feel human and are effective under stress.

Process

Author a complete medical triage script (e.g., breathing difficulties) in Ink.

Assessment

Conduct read-aloud testing with 3+ emergency response professionals, measuring time-to-critical-action.

Engineering for a 500MB Airgapped Payload

Fitting a comprehensive medical library, offline maps, and multiple ML models into a strict size budget requires aggressive optimization at every level.

Size Budget Allocation Table

Component	Size	Notes
App binary + UI assets	15 MB	Core app functionality
Intent classifier (MLTextClassifier)	5-10 MB	NaturalLanguage framework model
SQLite DBs (FTS5 + vectors)	150 MB	Primary reference storage
UK OpenStreetMap tiles (z0-12)	150 MB	Offline mapping capability
Conversation tree JSON	25 MB	Emergency scenario scripts
Headroom for growth	~50 MB	Critical buffer

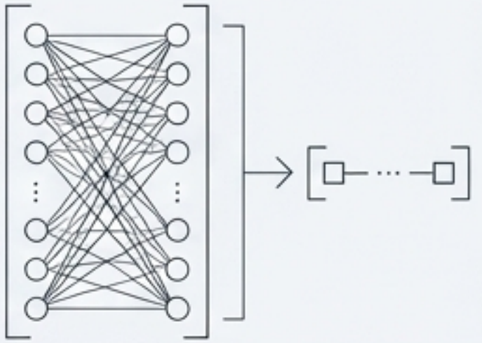
The Key to Compression: Vector Quantization

Problem

A standard 384-dim float32 vector for 100,000 documents would require ~150MB.

Solution

We use Binary Quantization, storing vectors as 1 bit per dimension.




```
100,000 articles * 384 dimensions * 1 bit = ~4.8 MB
```

Result

A **32x reduction** in storage for the vector index with negligible impact on retrieval accuracy.

The Data Pipeline: From Raw ZIM to a Searchable Knowledge Base.

Raw ZIM files are optimized for sequential reading, not random-access semantic search. We built a custom pipeline to transform this data.

1. Ingest

Start with `wikem_all_nopic.zim`.

2. Extract & Filter

Use `libzim` to iterate through the archive. Exclude non-content pages ('Talk', 'Admin').

3. Parse & Clean

Strip HTML tags to get plain text. Preserve structural hierarchy like headers and lists.

4. Semantic Chunking

Split long articles into smaller, ~512-token chunks. This allows the search to pinpoint a specific paragraph (e.g., 'tourniquet application') within a long article ('Trauma Care').

5. Embed & Quantize

Pass each chunk through the `all-MiniLM-L6-v2` embedding model, then quantize the resulting vector to binary format.

6. Load to SQLite

Populate the `documents` and `docs_vec` tables in the final SQLite database.

A Disciplined, Phased Implementation Plan.

Our development is sequenced to resolve the highest-risk items first, ensuring the project is built on a validated foundation.

Phase 1: Foundation

(2 Weeks)

Key Activities: Resolve NHS licensing decision. Build MLTextClassifier & Conversation Tree prototypes and conduct UX validation.

Outcome: All major strategic and UX risks mitigated.

Phase 2: Core Architecture

(3 Weeks)

Key Activities: SQLite schema design. ZIM file preprocessing pipeline. Background Assets integration.

Outcome: Data backbone and content delivery mechanism in place.

Phase 3: Feature Implementation

(4 Weeks)

Key Activities: Integrate the three layers: NL Classification, Conversation UI, and Search functionality.

Outcome: Core application features are functional.

Phase 4: Polish & Testing

(3 Weeks)

Key Activities: End-to-end emergency scenario testing. Performance and battery optimization.

Outcome: A production-ready, robust application.

Success is a Resilient Digital Lifeline.

We will measure success not by features shipped, but by meeting stringent performance metrics that translate directly into user trust and effectiveness in a crisis.

Technical Success Metrics

App Size
<480MB (final bundle)

Launch Time
<2 seconds cold start

Intent Latency
<500ms end-to-end

Battery Impact
<1% additional drain per hour of active use

User Experience Success Metrics

Scenario Success
>90% of users find critical information within 30 seconds.

User Confidence
>4/5 rating on “I would trust this app in a real emergency.”

Frustration Rate
<10% of sessions require abandoning natural language for manual search.

This is not just an app. It is a definitive tool for an era of uncertainty, providing a lifeline that remains operational when the connected world fails.