**Технологический Стек Проекта "Sentio Engine"**

**Уровень 1: Инфраструктура и Запуск (Фундамент)**

* **Контейнеризация:** **Docker**
  + **Роль:** Упаковать сервис "Sentio Engine" и его зависимости (если они появятся, например, БД) в изолированный контейнер. Это сделает его переносимым.
* **Оркестрация (опционально, для простоты):** **Docker Compose**
  + **Роль:** Позволит одной командой запустить сам сервис и, например, небольшую базу данных SQLite/PostgreSQL для хранения долгосрочной эмоциональной памяти.

**Уровень 2: Хранилище Данных (Память и Личность)**

* **Конфигурационные Файлы:** **JSON** (или **YAML**)
  + **Роль:** "ДНК" личности. Хранение всех статических, но легко изменяемых параметров:
    - emotion\_definitions.py (можно переделать в genomes/emotions.json)
    - profiles/female\_profile.json и male\_profile.json
    - Drives.json (система влечений)
    - BeliefSystem.json (система ценностей)
* **Хранилище Состояний:** **SQLite** (или **PostgreSQL** для более серьезных применений)
  + **Роль:** "Долгосрочная память". Хранение данных, которые накапливаются и изменяются со временем:
    - **EmotionalHistory:** Таблица для логирования активных чувств с временными метками (для анализа настроений).
    - **EmotionalMemory:** Таблица для хранения "триггеров" и ассоциаций (условных рефлексов).
    - **RelationshipState:** Таблица для хранения метрик отношений с каждым пользователем (session\_id, familiarity, intimacy, etc.).
* **Хранилище Текущего Состояния (опционально):** **JSON-файл** (как вы и делали) или **Redis**
  + **Роль:** "Краткосрочная память". Хранение текущего, сериализованного состояния движка (active\_emotions, neuromediators) для быстрого восстановления после перезапуска. Redis здесь подходит лучше, так как он обеспечивает более быстрый и атомарный доступ, чем файловая система.

**Уровень 3: Бэкенд и Логика (Сам Движок)**

* **Основной Язык:** **Python 3.10+**
  + **Роль:** Идеально подходит для быстрой разработки и работы с данными.
* **API-Сервер:** **FastAPI**
  + **Роль:** "Интерфейс к душе". Предоставляет простой и быстрый HTTP API для взаимодействия с движком. Ключевые эндпоинты:
    - POST /stimulus: Для "впрыскивания" эмоций.
    - GET /report: Для получения полного отчета о текущем состоянии.
    - POST /tick: (опционально, для внешнего управления) Для вызова одного "тика" симуляции.
* **Математические Вычисления:** **NumPy**
  + **Роль:** "Калькулятор". Эффективное выполнение всех математических операций с векторами эмоций и параметрами.

**Уровень 4: Искусственный Интеллект (Интеграция с Внешним Миром)**

Этот уровень **не является частью самого движка**, но движок спроектирован для взаимодействия с ним.

* **LLM-парсер (Внешний компонент):**
  + **Роль:** "Переводчик с человеческого". Это может быть функция в вашем чат-боте, которая использует **Llama.cpp** или **API OpenAI/Anthropic** для анализа ответа пользователя и преобразования его в "стимул" для Sentio Engine.
    - *Пример:* Анализирует текст "Спасибо, ты мне очень помог!" и формирует JSON: {"стимул": {"радость": 0.8, "доверие": 0.6}}.
* **LLM-генератор (Внешний компонент):**
  + **Роль:** "Голос". Это основной LLM вашего чат-бота, который получает **отчет** от Sentio Engine и использует его как часть своего системного промпта для генерации ответа в соответствующем стиле.

**Сводная Таблица Технологического Стека "Sentio Engine"**

|  |  |  |  |
| --- | --- | --- | --- |
| **Уровень** | **Компонент** | **Технология** | **Назначение** |
| **Инфраструктура** | Контейнеризация | **Docker / Docker Compose** | Упаковка и запуск сервиса |
| **Данные** | Конфигурация | **JSON / YAML** | Описание "личности", геномов, профилей |
|  | Долгосрочная Память | **SQLite / PostgreSQL** | Хранение истории, триггеров, отношений |
|  | Краткосрочная Память | **JSON-файл / Redis** | Хранение текущего состояния |
| **Бэкенд** | Основной язык | **Python** | Вся логика симуляции |
|  | API | **FastAPI** | Предоставление внешнего интерфейса |
|  | Вычисления | **NumPy** | Математические операции |
| **Интеграция с ИИ** | Парсер Стимулов | *(Внешний)* **LLM (Llama.cpp)** | Анализ текста и создание "стимула" |
|  | Генератор Ответов | *(Внешний)* **LLM (любая)** | Использование отчета для стилизации ответа |

Этот стек является **легковесным, сфокусированным и самодостаточным**. Вы получаете полноценный микросервис, который делает одну вещь, но делает ее невероятно хорошо — симулирует сложную эмоциональную жизнь. Его легко поддерживать, тестировать и, что самое главное, — **легко интегрировать** в любой из ваших существующих или будущих мета-проектов.