Narcissus Refraction Engine (NRE): Technical Stack and Implementation Plan

Auspexi Research Team
sales@auspexi.com

July 2027

Abstract

This document outlines the technical stack and implementation plan for the Narcissus Refraction Engine (NRE), enabling developers to execute Auspexi's vision of bias-free synthetic data generation upon investment. The stack leverages React/TypeScript, Python, free Hugging Face models, Netlify, and Supabase, with a potential transition to Databricks for secure data delivery. The plan ensures modularity, scalability to 10M records/day, and compliance with GDPR, ISO 27001, and HIPAA, aligning with NRE's architecture for rapid deployment.

1 Introduction

The Narcissus Refraction Engine (NRE) generates 3M synthetic records/day with $<0.5\%$ real data reliance, targeting SMEs in finance, healthcare, logistics, and retail. This document details the technical stack and a developer-ready implementation plan to scale NRE to 10M records/day post-investment, ensuring compliance and modularity.

2 Technical Stack

The NRE stack is designed for cost-efficiency, scalability, and compliance, leveraging open-source and low-cost platforms.

2.1 Frontend: React/TypeScript

- **Purpose**: Build a modular, type-safe dashboard for SMEs to interact with NRE's synthetic data APIs and visualize performance metrics (e.g., bias reduction, accuracy).
- **Libraries**: React 18, TypeScript 5.5, Tailwind CSS for styling, React Router for navigation.
- **Features**: Real-time data visualization, API seat management, and user authentication.
- **Hosting**: Netlify ($19/month) for CI/CD, serverless functions, and static site hosting.

2.2 Backend: Python

- **Purpose**: Handle data generation, harmonic-octonion embeddings, bias pruning, and model training.
- **Libraries**: NumPy, pandas, PyTorch (for embeddings and neural validation), FastAPI (for REST APIs).

- **Features**: Implements 8-Suite SDSP, Harmonic Refraction Layer, Polyhedral Bias Pruner, and Narcissus Feedback Loop.
- **Deployment**: Supabase ($25/month) for initial database and storage; AWS EC2 c5.4xlarge ($0.68/hour) for scaling to 10M records/day.

2.3 Hugging Face Models

- **Purpose**: Generate and validate synthetic data using free, open-source models from Hugging Face, as referenced in Auspexi's SDSP whitepaper.
- **Models**: DistilBERT (for text generation), BART (for data augmentation), CLIP (for multi-modal validation).
- **Integration**: Python scripts to fine-tune models for NRE's 8D octonion space and bias reduction.

2.4 Infrastructure: Netlify and Supabase

- **Netlify**: Hosts React frontend, providing serverless functions for API endpoints and CI/CD for rapid deployment.
- **Supabase**: Manages real-time PostgreSQL database, authentication, and storage for synthetic data metadata. Costs $25/month for initial 3M records/day.
- **Scalability**: Upgrade to Supabase Pro ($100/month) or AWS EC2 (c5.4xlarge, $0.68/hour, $8,200/month for 10M records/day).

2.5 Potential: Databricks

- **Purpose**: Replace Supabase for secure data delivery/reception in regulated industries (e.g., finance, healthcare) post-investment.
- **Features**: Enterprise-grade security (SHA-256, zk-SNARKs), Delta Lake for data pipelines, and compliance with GDPR, ISO 27001, and HIPAA.
- **Integration**: Migrate Supabase PostgreSQL to Databricks SQL, leveraging Databricks' MLflow for model tracking.

3 Implementation Plan

The plan outlines tasks, timelines, and dependencies for a developer team to deploy NRE post-investment, targeting 3M records/day initially and 10M with funding.

3.1 Phase 1: Setup and Development (Months 1–3)

- **Task 1: Frontend Development**
  - Develop React/TypeScript dashboard with Tailwind CSS.
  - Implement API seat management, authentication (via Supabase Auth), and data visualization (e.g., bias metrics, KS test results).
  - Deploy to Netlify with CI/CD pipeline.
  - *Dependencies*: Netlify account, React 18, TypeScript 5.5.
  - *Timeline*: 2 months.

- **Task 2: Backend Development**
  - Implement 8-Suite SDSP in Python using FastAPI and PyTorch.
  - Develop Harmonic-Octonion Embeddings and Polyhedral Bias Pruner:

$$o = x_0 + \sum_{i=1}^{7} x_i e_i, \quad B = \sum_{i=1}^{8} w_i \cdot \mathrm{Var}(\phi_i(x)).$$

  - Integrate Hugging Face models (DistilBERT, BART) for data generation.
  - Deploy to Supabase for initial 3M records/day.
  - *Dependencies*: Python 3.11, PyTorch 2.3, Supabase account.
  - *Timeline*: 3 months.

3.2 Phase 2: Validation and Testing (Months 4–6)

- **Task 3: Model Validation**
  - Implement Hypercube Validator using Kolmogorov-Smirnov tests (KS test, $D_n < 0.0005$, $p > 0.05$).
  - Fine-tune Hugging Face models for 98% bias reduction ($D_{\text{bias}} < 0.02$).
  - *Dependencies*: Task 2 completion, NumPy, pandas.
  - *Timeline*: 2 months.
- **Task 4: Compliance Testing**
  - Verify GDPR, ISO 27001, and HIPAA compliance using SHA-256 and zk-SNARKs.
  - Test data pipelines with 1M financial records for 97% accuracy.
  - *Dependencies*: Task 2 completion, Supabase database.
  - *Timeline*: 1 month.

3.3 Phase 3: Scalability and Deployment (Months 7–9)

- **Task 5: Scalability Implementation**
  - Migrate backend to AWS EC2 c5.4xlarge ($0.68/hour) for 10M records/day.
  - Optimize TypeScript APIs for throughput, targeting $8,200/month operational cost.
  - *Dependencies*: AWS account, Task 2 completion.
  - *Timeline*: 2 months.
- **Task 6: Databricks Integration (Optional)**
  - If funded, migrate Supabase data to Databricks Delta Lake.
  - Implement secure data delivery/reception with MLflow for model tracking.
  - *Dependencies*: Databricks partnership, Task 5 completion.
  - *Timeline*: 1 month.

3.4 Phase 4: Launch and Maintenance (Months 10–12)

- **Task 7: Production Launch**
  - Deploy NRE to production, offering API seats at \$5K–\$20K/year.
  - Monitor performance (97% accuracy, 98% bias reduction).
  - *Dependencies*: Tasks 1–6 completion.
  - *Timeline*: 1 month.
- **Task 8: Maintenance**
  - Monitor Narcissus Feedback Loop for model drift ($D_{\text{KL}} < 0.03$).
  - Update models quarterly using Hugging Face's latest releases.
  - *Dependencies*: Task 7 completion.
  - *Timeline*: Ongoing.

4 Resource Requirements

- **Team**: 2 frontend developers (React/TypeScript), 3 backend developers (Python/FastAPI), 1 DevOps engineer (Netlify/Supabase/AWS).
- **Budget**: \$44/month (Netlify \$19, Supabase \$25) initially; \$8,200/month for AWS scaling; Databricks costs TBD.
- **Hardware**: Local development (8-core CPU, 16GB RAM); AWS EC2 c5.4xlarge for production.

5 Conclusion

The NRE tech stack and implementation plan enable rapid deployment of a scalable, compliant, and cost-efficient synthetic data engine. Post-investment, developers can leverage React/TypeScript, Python, Hugging Face models, Netlify, and Supabase (with potential Databricks integration) to achieve 10M records/day, targeting a \$10–20B market. Contact sales@auspexi.com for details.