Brainium

AI Integration for Viridi Platform: Workflow and Technical Analysis

Introduction

The SaaS platform fetches invoices from multiple sources, creates past orders, and processes them. Integrating artificial intelligence (AI) enhances automation, fraud detection, data translation, user interaction, and data analysis with visualization. This document details the AI-driven components, their workflows, benefits, and a backend-focused technology stack using AWS exclusively. Vector DB is included where needed. The goal is to streamline operations, improve accuracy, and provide actionable insights while preserving core system integrity.

The key AI components are:

1. **AI-Driven Auto Transformer**: Automates adapter creation for diverse invoice APIs.
2. **Invoice Fraud Detection**: Identifies fraudulent invoices using synthetic data.
3. **Smart Translation**: Provides context-aware translation for invoice data.
4. **Chatbot for User Interaction**: Handles user queries via backend API.
5. **Smart Statistics with AI**: Enables natural language-based data analysis and visualization.
6. **Invoice Discrepancy Warning Flags**: Flags order-invoice mismatches with red, yellow, green alerts.
7. **Automatic Product Classification**: Categorizes products from invoice data.
8. **Smart Product Information Management (PIM)**: Enriches Product Detail Cards with invoice data.
9. **Detect Opportunities for Economical/Environmental Purchases**: Identifies cost-effective, eco-friendly purchases.
10. **Detect Opportunities to Reduce Waste**: Flags wasteful purchase patterns.
11. **Smart ESG Vendor Assessment**: Evaluates vendors for ESG compliance, detecting greenwashing.
12. **Smart Sustainability Reporting**: Generates GHG emission and ESG reports.
13. **AI in UX and Smart Feedback Detection**: Improves UX by analyzing user interactions.

Technology Stack

Category	Technology	Purpose
Programming Languages	Python 3.13	AI model development, data processing, and backend logic.
AI/ML Frameworks	TensorFlow/PyTorch	Training and deploying fraud detection and translation models.
	Hugging Face Transformers	NLP for Smart Translation and Chatbot.
	Pandas/NumPy	Data manipulation for Smart Statistics.
	Matplotlib/Seaborn	Data visualization for Smart Statistics (charts, graphs).
API Integration	FastAPI	RESTful APIs for invoice data and AI interactions.

	OpenAPI/Swagger	Documenting invoice provider APIs.
Databases	Amazon RDS (PostgreSQL)	Storing structured invoice and order data.
	Amazon ElastiCache (Redis)	Caching API responses and chatbot session data.
	Amazon OpenSearch Service (Vector DB)	Storing and querying embeddings for Smart Translation and Chatbot.
Cloud Infrastructure	AWS EC2	Hosting AI models and backend services.
	AWS Lambda	Serverless execution for lightweight AI tasks.
	Amazon S3	Storing synthetic data, invoice archives, and visualization outputs.
Other Tools	Docker	Containerizing AI services for consistent deployments on AWS ECS.

AI Components: Overview and Benefits of Each Individual AI Component

1. AI-Driven Auto Transformer

Introduction

The AI-Driven Auto Transformer automates adapter creation for consuming invoice data from diverse APIs without altering core serialization logic. It infers schemas and generates adapter code to map external data to the platform's schema.

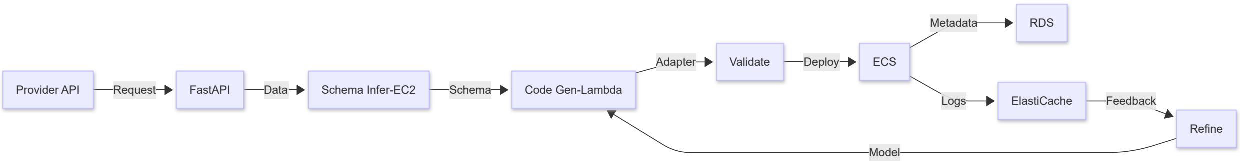
Workflow

1. **API Request Ingestion**: Receive API request from a new invoice provider via FastAPI.
2. **Schema Inference**: AI model on AWS EC2 analyzes API responses to identify fields and data types using JSON schema parsing.
3. **Adapter Generation**: A code generation model (e.g., fine-tuned CodeT5) on AWS Lambda generates Python adapter code.
4. **Validation**: Test adapter against sample data, logging results in Amazon RDS.
5. **Integration**: Deploy adapter as a FastAPI endpoint on AWS ECS, storing metadata in Amazon RDS.
6. **Monitoring**: Log performance in Amazon ElastiCache and refine model with feedback stored in Amazon RDS.

Benefits

- **Scalability**: Automates onboarding of new providers.
- **Maintainability**: Isolates adapter logic from core system.
- **Efficiency**: Reduces manual development effort.
- **Adaptability**: Handles varied API formats.

Workflow Diagram



2. Invoice Fraud Detection

Introduction

The Invoice Fraud Detection AI identifies fraudulent invoices using synthetic pattern data for training and retraining with feedback, ensuring robust detection without extensive historical fraud data.

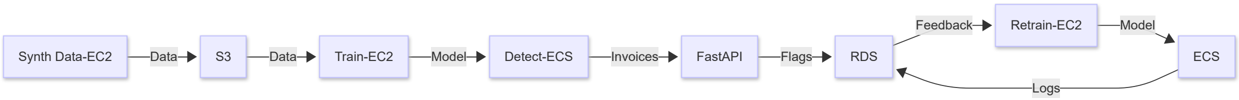
Workflow

1. **Synthetic Data Generation**: Create synthetic invoices with fraudulent patterns using AWS EC2, stored in Amazon S3.
2. **Model Training**: Train a TensorFlow model on synthetic data using AWS EC2.
3. **Real-Time Detection**: Process invoices via FastAPI on AWS ECS, flagging potential fraud.
4. **Feedback Loop**: Store user feedback in Amazon RDS and retrain model using active learning on AWS EC2.
5. **Alerting**: Log flagged invoices in Amazon RDS for admin review.
6. **Model Update**: Deploy updated models via Docker on AWS ECS.

Benefits

- **Accuracy**: Synthetic data ensures robust initial training.
- **Adaptability**: Retraining improves detection over time.
- **Security**: Mitigates financial risks from fraud.
- **Automation**: Reduces manual review effort.

Workflow Diagram



3. Smart Translation

Introduction

The Smart Translation AI provides context-aware translation for invoice data (e.g., product names), leveraging Amazon OpenSearch Service (Vector DB) for semantic understanding to overcome limitations of generic translators.

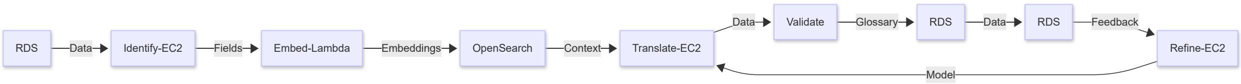
Workflow

1. **Data Identification**: Use NLP on AWS EC2 to identify translatable fields in invoices.
2. **Embedding Generation**: Generate embeddings using Hugging Face's mBART on AWS Lambda, stored in Amazon OpenSearch Service.
3. **Context Analysis**: Query OpenSearch to retrieve similar contexts for accurate translation.
4. **Translation**: Translate fields into target language via AWS EC2, preserving domain-specific meanings.
5. **Validation**: Check translations against a glossary in Amazon RDS.
6. **Feedback**: Store user corrections in Amazon RDS to fine-tune the model on AWS EC2.

Benefits

- **Accuracy**: Semantic search via Vector DB ensures context-aware translations.
- **Usability**: Supports multilingual clients.
- **Customization**: Adapts to industry-specific terms.
- **Efficiency**: Automates translation tasks.

Workflow Diagram



4. Chatbot for User Interaction

Introduction

The Chatbot handles user queries about invoices and orders via a backend API, using NLP and Amazon OpenSearch Service for semantic query understanding.

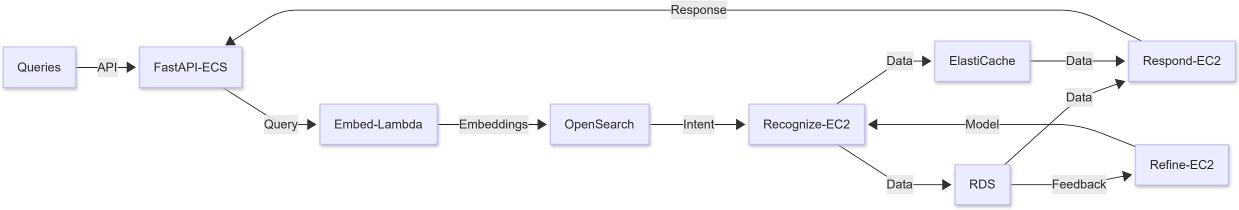
Workflow

1. **Query Ingestion**: Receive user queries via FastAPI endpoint on AWS ECS.
2. **Query Embedding**: Generate embeddings using Hugging Face's DistilBERT on AWS Lambda, stored in Amazon OpenSearch Service.
3. **Intent Recognition**: Query OpenSearch to match query to intents (e.g., "view invoices").
4. **Data Retrieval**: Fetch data from Amazon RDS or Amazon ElastiCache.
5. **Response Generation**: Generate response via AWS EC2 and return via API.
6. **Feedback**: Store user feedback in Amazon RDS for model refinement.

Benefits

- **Efficiency**: Automates query resolution.
- **Accuracy**: Vector DB improves intent recognition.
- **Scalability**: Handles multiple queries with caching.
- **Usability**: Simplifies user interactions.

Workflow Diagram



5. Smart Statistics with AI

Introduction

The Smart Statistics AI enables natural language-based data analysis and visualization of invoice data. Users can query data (e.g., "Show invoices from 2023") and request visualizations (e.g., "Plot invoice totals by month"), with AI generating Python code for analysis and charts.

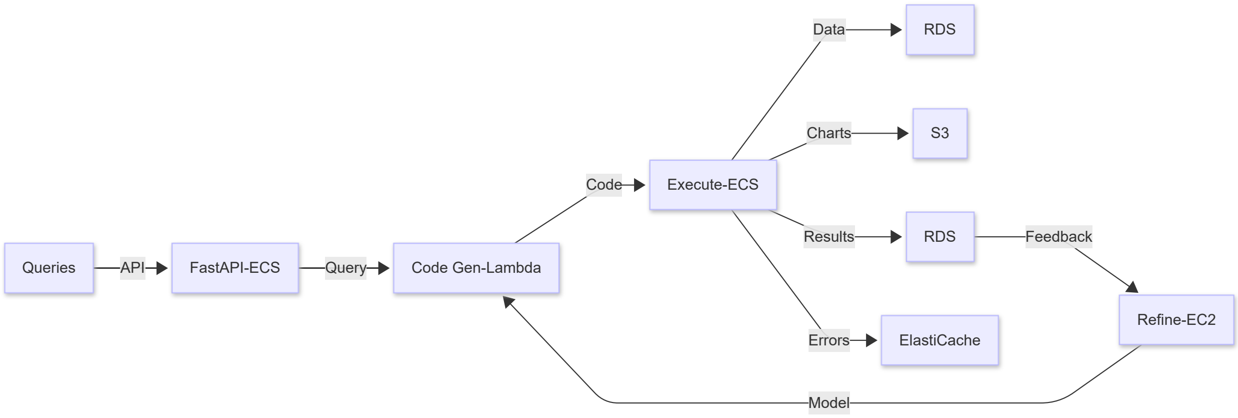
Workflow

1. **Query Input**: Receive natural language query via FastAPI on AWS ECS.
2. **Code Generation**: Use fine-tuned GPT on AWS Lambda to generate Python code, leveraging Pandas for analysis and Matplotlib/Seaborn for visualizations.
3. **Code Execution**: Run code in a Docker sandbox on AWS ECS, accessing data from Amazon RDS.
4. **Visualization Generation**: Generate charts (e.g., bar, line) and store images in Amazon S3.
5. **Result Storage**: Store analysis results and visualization metadata in Amazon RDS.
6. **Feedback Loop**: Log user feedback in Amazon RDS to refine NLP model.
7. **Error Handling**: Log errors in Amazon ElastiCache for debugging.

Benefits

- **Accessibility**: Enables non-technical users to analyze and visualize data.
- **Flexibility**: Supports diverse queries and visualization types (e.g., bar, line, pie charts).
- **Efficiency**: Automates complex analysis and visualization tasks.
- **Insights**: Uncovers trends through data analysis and visual representations.

Workflow Diagram



6. Order Discrepancy Warning

Overview

The Order Discrepancy Warning AI compares invoices with corresponding orders to detect mismatches (e.g., price, quantity) and flags them with a red, yellow, green alert system (red for critical issues like fraud, yellow for minor discrepancies, green for no issues). It integrates with the fraud detection component for enhanced accuracy.

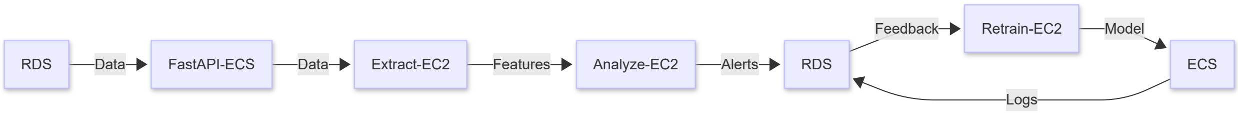
Workflow

1. **Data Ingestion**: Retrieve invoice and order data from Amazon RDS via FastAPI on AWS ECS.
2. **Feature Extraction**: Use AWS EC2 to extract features (e.g., price, quantity) for comparison.
3. **Discrepancy Analysis**: Apply a TensorFlow model on AWS EC2 to compare features and assign alerts (red: critical, yellow: minor, green: none).
4. **Alert Storage**: Log alerts with metadata (e.g., discrepancy type) in Amazon RDS.
5. **Feedback Loop**: Store user feedback on alert accuracy in Amazon RDS and retrain model on AWS EC2.
6. **Model Update**: Deploy updated models via Docker on AWS ECS.

Benefits

- **Accuracy**: Detects both mistakes and potential fraud with tiered alerts.
- **Usability**: Clear red/yellow/green system simplifies issue prioritization.
- **Integration**: Enhances fraud detection by sharing data insights.
- **Efficiency**: Automates discrepancy detection, reducing manual checks.

Workflow Diagram



7. Automatic Product Classification

Overview

The Automatic Product Classification AI categorizes products from invoice data (e.g., “laptop” as “Electronics”) using NLP and Amazon OpenSearch Service for semantic understanding, enabling better inventory and analytics.

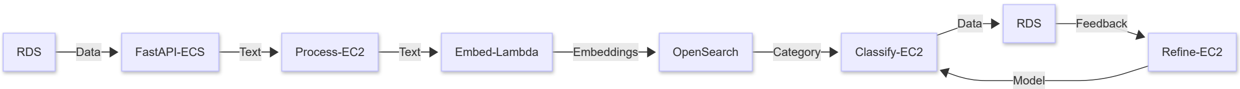
Workflow

1. **Data Ingestion**: Retrieve invoice product data from Amazon RDS via FastAPI on AWS ECS.
2. **Text Processing**: Use NLP on AWS EC2 to preprocess product descriptions (e.g., tokenization).
3. **Embedding Generation**: Generate embeddings using Hugging Face's Transformers on AWS Lambda, stored in Amazon OpenSearch Service.
4. **Classification**: Query OpenSearch to classify products into categories via AWS EC2.
5. **Storage**: Store classifications in Amazon RDS.
6. **Feedback**: Log user corrections in Amazon RDS to fine-tune the model.

Benefits

- **Accuracy**: Semantic embeddings ensure precise categorization.
- **Scalability**: Handles diverse product descriptions across invoices.
- **Insights**: Enhances inventory management and analytics.
- **Automation**: Eliminates manual product categorization.

Workflow Diagram



8. Smart Product Information Management (PIM)

Overview

Enriches VIRIDI Product Detail Cards by automating the addition of details (e.g., classification, CO2e units, origin) from limited invoice data.

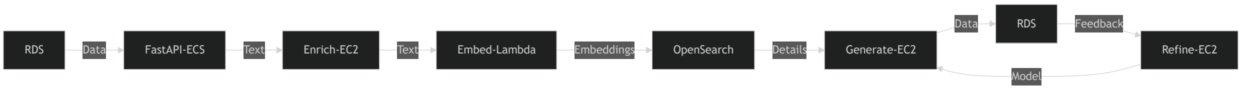
Workflow

1. **Data Ingestion**: Retrieve product data from Amazon RDS via FastAPI on AWS ECS.
2. **Data Enrichment**: Use NLP on AWS EC2 (Transformers) to extract/enrich details.
3. **Embedding Generation**: Generate embeddings on AWS Lambda, stored in Amazon OpenSearch Service.
4. **Detail Generation**: Query OpenSearch to infer details (e.g., CO2e) on AWS EC2.
5. **Storage**: Save enriched data in Amazon RDS.
6. **Feedback**: Log corrections in Amazon RDS to refine model.

Benefits

- **Automation**: Reduces manual data entry for Product Detail Cards.
- **Accuracy**: Semantic embeddings ensure precise detail inference.
- **Scalability**: Handles diverse product data.
- **Sustainability**: Supports CO2e tracking for ESG goals.

Workflow Diagram



9. Detect Opportunities for Economical/Environmental Purchases

Overview

Identifies cost-effective and eco-friendly purchase options by analyzing invoice/order data for cost and environmental impact.

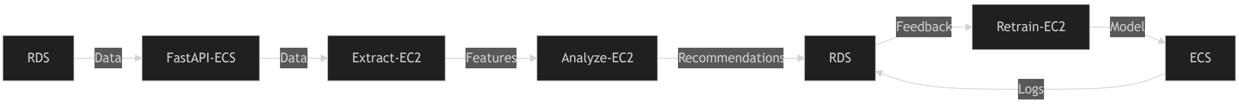
Workflow

1. **Data Ingestion**: Fetch invoice/order data from Amazon RDS via FastAPI on AWS ECS.
2. **Feature Extraction**: Extract features (e.g., price, CO2e) on AWS EC2 using Pandas.
3. **Opportunity Analysis**: Use TensorFlow on AWS EC2 to score purchases.
4. **Recommendation**: Generate recommendations (e.g., alternative vendors) in Amazon RDS.
5. **Feedback**: Log feedback in Amazon RDS for retraining.
6. **Model Update**: Deploy updated models via Docker on AWS ECS.

Benefits

- **Cost Efficiency**: Identifies cheaper purchasing options.
- **Sustainability**: Promotes eco-friendly vendors/products.
- **Automation**: Streamlines decision-making.
- **Insights**: Enhances ESG reporting.

Workflow Diagram



10. Detect Opportunities to Reduce Waste

Overview

Identifies wasteful purchases (e.g., excess quantities) to optimize resource use, analyzing purchase patterns.

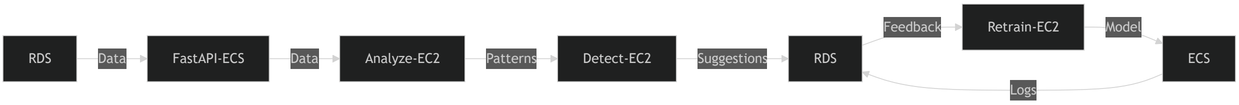
Workflow

1. **Data Ingestion**: Retrieve invoice/order data from Amazon RDS via FastAPI on AWS ECS.
2. **Pattern Analysis**: Analyze patterns (e.g., overstocking) on AWS EC2 using Pandas.
3. **Waste Detection**: Apply TensorFlow on AWS EC2 to flag wasteful patterns.
4. **Recommendation Storage**: Store suggestions in Amazon RDS.
5. **Feedback**: Log feedback in Amazon RDS for refinement.
6. **Model Update**: Deploy updated models via Docker on AWS ECS.

Benefits

- **Sustainability**: Reduces unnecessary purchases.
- **Efficiency**: Optimizes inventory management.
- **Cost Savings**: Minimizes wasteful expenditure.
- **Scalability**: Handles large datasets.

Workflow Diagram



11. Smart ESG Vendor Assessment

Overview

Evaluates vendors for ESG compliance, detecting greenwashing by analyzing vendor data and external ESG reports.

Workflow

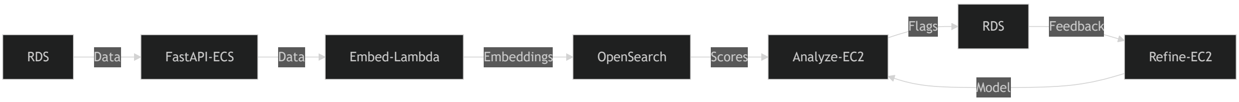
1. **Data Ingestion**: Collect vendor data from Amazon RDS and external reports via FastAPI on AWS ECS.
2. **Embedding Generation**: Generate embeddings using Transformers on AWS Lambda, stored in Amazon OpenSearch Service.
3. **ESG Analysis**: Query OpenSearch on AWS EC2 to score vendors on ESG metrics.

4. **Greenwashing Detection**: Flag inconsistencies using TensorFlow on AWS EC2.
5. **Storage**: Save ESG scores/flags in Amazon RDS.
6. **Feedback**: Log feedback in Amazon RDS for refinement.

Benefits

- **Transparency**: Identifies greenwashing for reliable ESG reporting.
- **Compliance**: Ensures vendors meet ESG standards.
- **Automation**: Reduces manual assessments.
- **Accuracy**: Semantic analysis improves scoring.

Workflow Diagram



12. Smart Sustainability Reporting

Overview

Generates automated sustainability reports, including GHG emissions and ESG metrics, from invoice/order data.

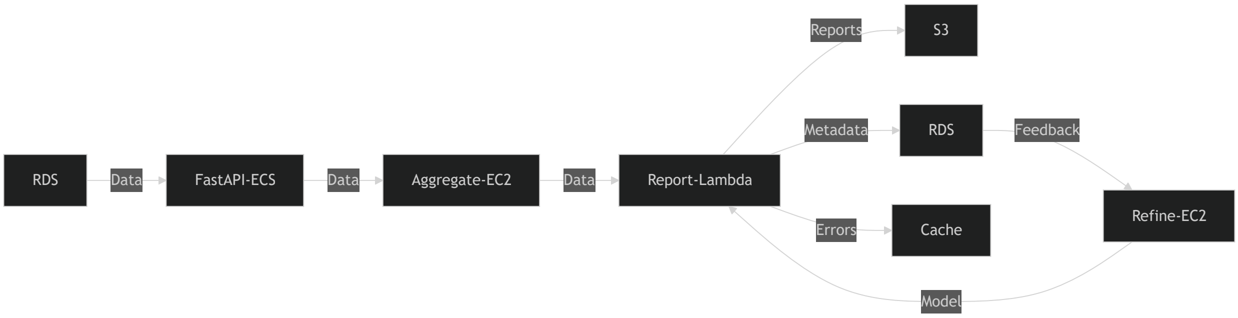
Workflow

1. **Data Ingestion**: Fetch invoice/order data from Amazon RDS via FastAPI on AWS ECS.
2. **Data Aggregation**: Aggregate data (e.g., CO2e, ESG scores) using Pandas on AWS EC2.
3. **Report Generation**: Use fine-tuned GPT on AWS Lambda for reports with Matplotlib visualizations.
4. **Storage**: Save reports as PDFs in Amazon S3, metadata in Amazon RDS.
5. **Feedback**: Log feedback in Amazon RDS for refinement.
6. **Error Handling**: Log errors in Amazon ElastiCache.

Benefits

- **Automation**: Streamlines ESG/GHG reporting.
- **Insights**: Provides actionable sustainability metrics.
- **Compliance**: Supports regulatory requirements.
- **Visualization**: Enhances reports with charts.

Workflow Diagram



13. AI in UX and Smart Feedback Detection

Overview

Analyzes user interactions to detect feedback and improve UX, supporting VIRIDI's user-centric design.

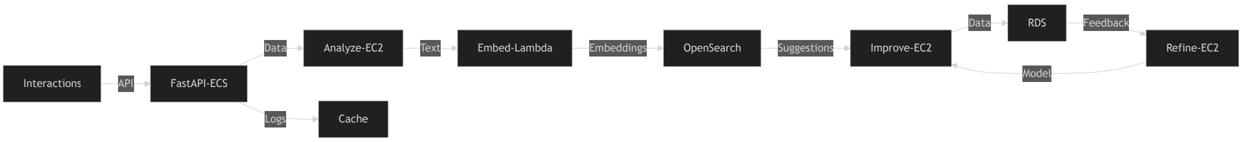
Workflow

1. **Interaction Logging**: Capture interactions (e.g., clicks, queries) via FastAPI on AWS ECS, stored in Amazon ElastiCache.
2. **Feedback Analysis**: Use NLP on AWS EC2 (Transformers) to identify sentiment/issues.
3. **Embedding Generation**: Generate embeddings on AWS Lambda, stored in Amazon OpenSearch Service.
4. **UX Improvement**: Query OpenSearch to suggest UI improvements on AWS EC2.
5. **Storage**: Save feedback/suggestions in Amazon RDS.
6. **Refinement**: Retrain models on AWS EC2 with feedback.

Benefits

- **Usability**: Enhances UX based on user behavior.
- **Responsiveness**: Quickly detects issues.
- **Automation**: Reduces manual feedback analysis.
- **Scalability**: Handles large interaction datasets.

Workflow Diagram



Conclusion

The 13 AI components enhance the VIRIDI platform’s capabilities in automation, security, translation, user interaction, data analysis, sustainability, and UX. The AI-Driven Auto Transformer streamlines API integration, Invoice Fraud Detection and Discrepancy Warning Flags ensure security, Smart Translation supports global use, the Chatbot simplifies queries, Smart Statistics provides insights, Automatic Product Classification and PIM enhance product management, Economical/Environmental Purchases and Waste Reduction promote sustainability, ESG Vendor Assessment ensures compliance, Sustainability Reporting meets regulatory needs, and UX Feedback improves usability. The AWS-exclusive stack, with Amazon OpenSearch Service as a Vector DB, ensures scalability and accuracy. Compact, rectangular Mermaid diagrams provide clear workflow visualization.