Zimbozapp: Meal Recipe Telegram Bot

A Data Engineering Final Project by Orgad Salomon

Demonstrates modern data pipeline from ingestion to user interaction

by orgad salomon

Project Overview

Smart Recipe Suggestions

Based on groceries, prep time, meal type, dietary needs

Modern Tech Stack

Docker, Python, Kafka, PySpark, MinIO, Elasticsearch, Airflow

Complete Pipeline

Full data engineering workflow demonstration

Technology Stack

Spoonacular API
Recipe data source

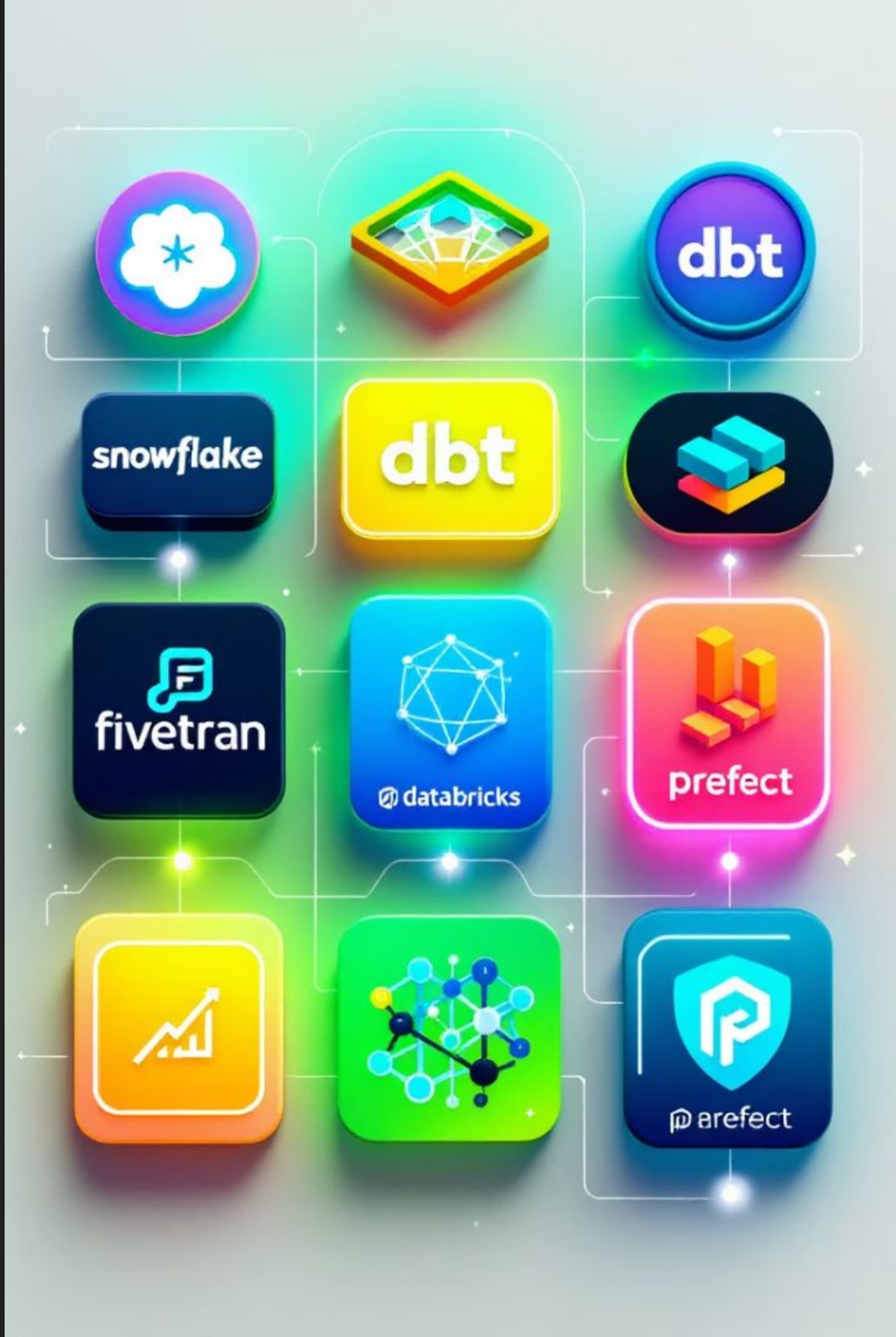
Kafka Streaming
Real-time data flow

PySpark Processing
Batch and stream handling

MinIO + Elasticsearch
Storage and search

Airflow
Orchestrate the process

Docker
Containerize



System Architecture

1

Spoonacular API

Recipe data ingestion

2

Kafka Stream

Message broker layer

3

PySpark Processing

Data transformation

4

Storage & Search

MinIO lake, Elasticsearch index

5

Telegram Bot

User interface

Key Components

Airflow DAG

Weekly recipe fetch with fallback to mock data

Kafka Streams

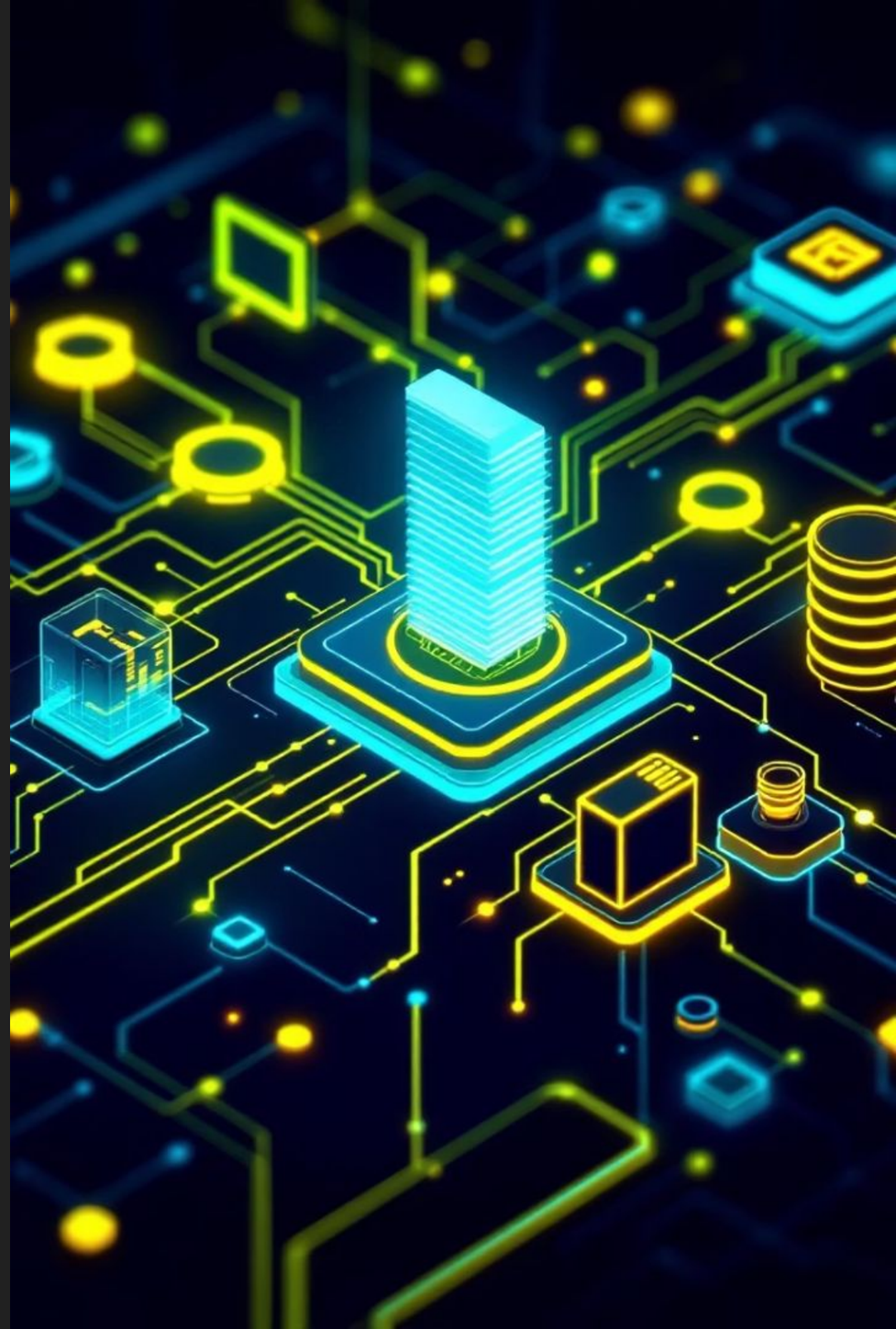
Real-time recipe data distribution

Spark Processing

Batch and streaming data transformation

Search Interface

Elasticsearch fast recipe lookup and filtering



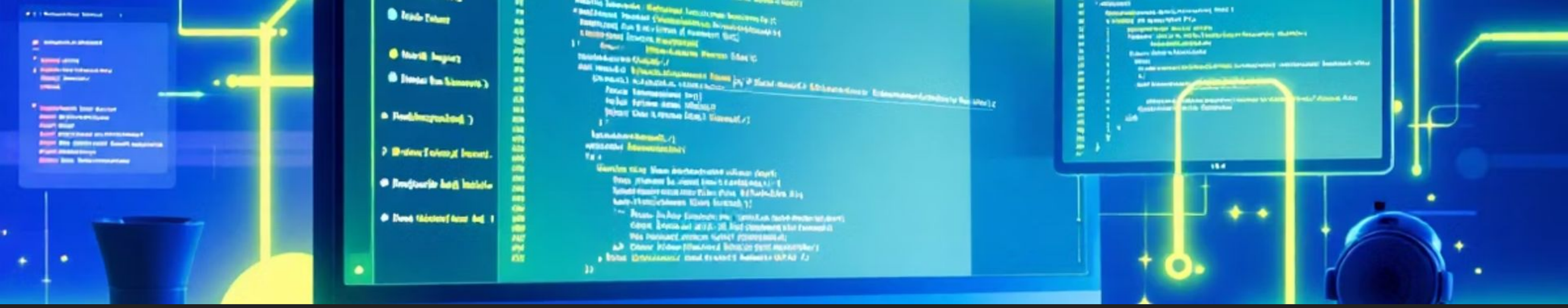
Project Structure

Core Services

- airflow/dags/fetch_spoonacular.py
- bot/bot.py
- spark/spark_streaming.py

Configuration

- elasticsearch/setup_index.py
- docker-compose.yml
- .env and config files



Setup Instructions

Clone and Configure

Clone repository and create .env file

Launch Services

Run docker-compose up --build command

Access Interfaces

Airflow UI at localhost:8081, start Telegram bot

Project Highlights

100%

Containerized

Docker Compose deployment

M1/M2

Apple Silicon

Compatible architecture

2

Data Modes

Real and mock data support

∞

Scalable