

Your All In One E-commerce Solution

# Dealytics

Layered Architecture

Presented by Akanta Das

# Overview

1. Presentation Layer (User Interface and API Gateway)
2. Business Logic Layer (Core Services)
3. Persistence Layer (DAO, ORM)
4. Database Layer (Relational and Time-Series Storage)
5. Integration Layer (Web scraping, API Integration, Notification and Message Services)
6. Infrastructure Layer (Cloud platform, Containerization, CI/CD Monitoring, Log Management)



# Presentation Layer

- Purpose:
  - Handles user interaction,
  - ensuring a seamless experience for buyers and sellers.

# Presentation Layer

- Components:

- Buyer :

- Personalized Deals UI
- Trending Deals UI
- Product Query U
- Deals List UI
- Product Review UI

- Admin :

- Scraping Dashboard
- Scraping Configuration View

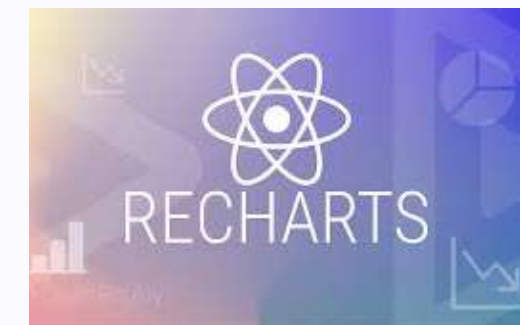
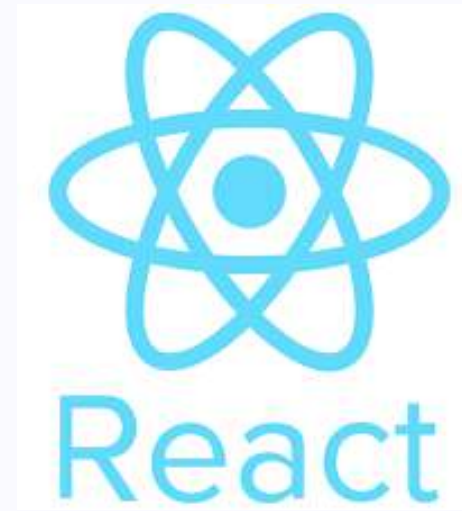
- Seller :

- Critical Inventory Product UI
- Critical Reputation Product UI
- Critical Price Update UI
- Recent Product UI
- Deal List UI
- Product List UI
- Custom Analytics UI
- Product Review UI



# Presentation Layer

- Tech Stack:
  - React.js
  - Material-UI + Tailwind CS
  - Recharts + D3.js



# Business Logic Layer

- Purpose:
  - Implements the core logic of backend
  - Data processing, storing, retrieval workflow



# Business Logic Layer

- Components:
  - Analytics Service for Buyer
  - Analytics Service for Seller
  - Subscription Service
  - Product Aggregation Service
  - ReviewService
  - Deal Metrics Service
  - User Engagement Analytics Service
  - Conversation Service
  - Product Forecast Service(Price/Reputation/Inventory)

# Business Logic Layer

- Tech Stack:
  - Backend Services:  
Spring Boot with embedded Tomcat
  - Internal Communication:  
Eureka Naming Server





# Persistence Layer

- Purpose:
  - Abstracts interactions with the database
  - Provides schemas for database tables

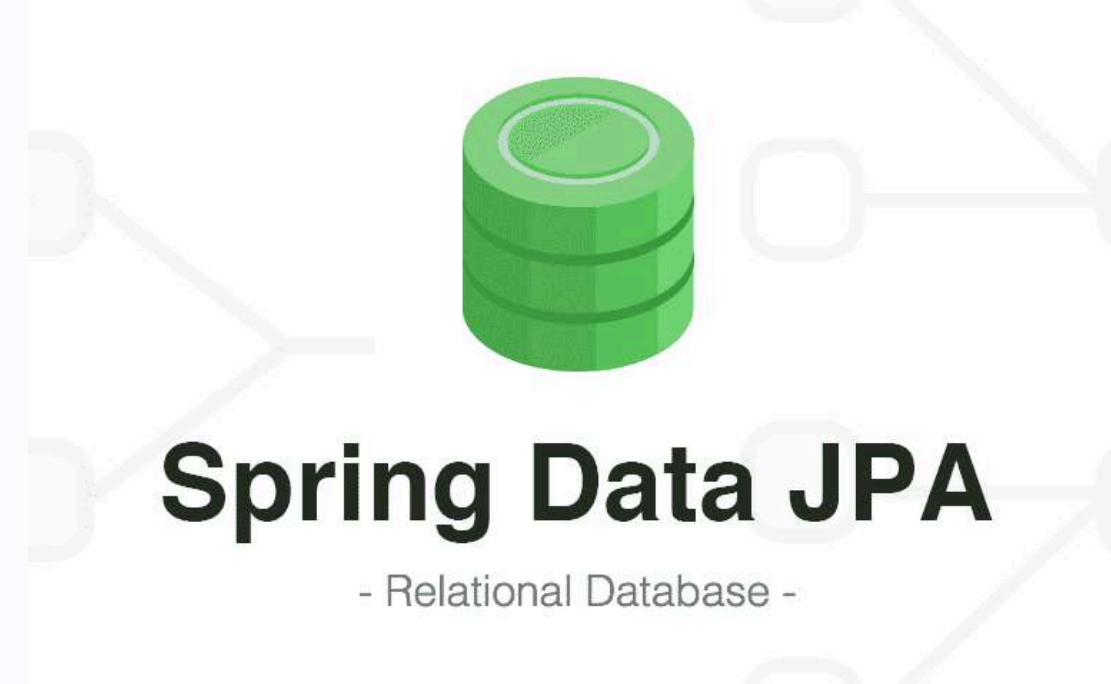
# Persistence Layer

- Components:
- DAO:
  - UserEngagementProfileDAO
  - BrowsingSessionDAO
  - BrowsingEventDAO
  - ProductUpdateDAO
  - NotificationDAO
  - ConversationModelDAO
  - MessageDAO
- ORM



# Persistence Layer

- Tech Stack:
  - ORM:  
Java Persistence API (JPA)
  - Cache:  
Redis



# Database Layer

- Purpose:
  - Actual storage and management of data
  - Handles relations between data
  - Enables querying data efficiently
  - Enables semantic search



# Database Layer

## Primary DB(Relational):

### Stores

- Product
- Buyer
- Seller
- Deal
- Review
- Subscription
- Scraping Rule
- Scraping Target
- Scraping Session
- Scraped Product

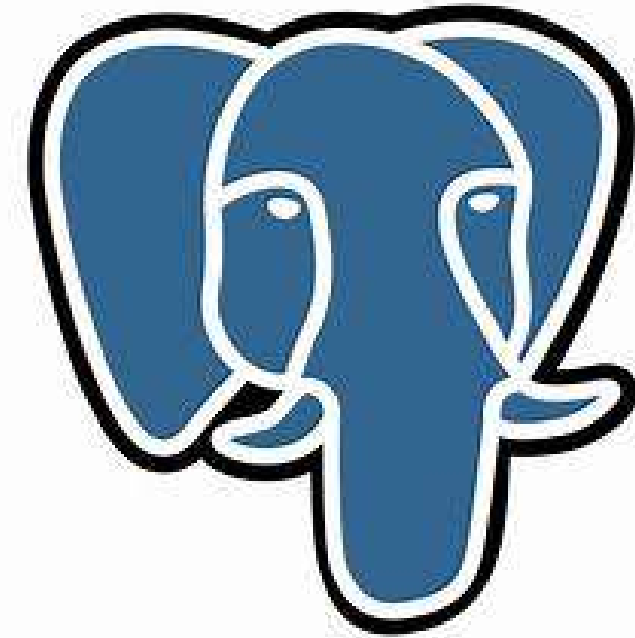
## Time Series DB:

### Stores

- UserEngagementProfile
- BrowsingSession
- BrowsingEvent
- ProductUpdate

# Database Layer

- Tech Stack:
  - PostgreSQL
  - InfluxDB





# Integration Layer

- Purpose:
  - Abstracts communication with external APIs
  - Enables usage of external services with ease

# Integration Layer

- Components:
  - Web scraping
  - API integration
  - Notification and Messaging Services
    - Notification
    - ConversationModel
    - Message



# Integration Layer

- Tech Stack:
  - Scrapy , BeautifulSoup
  - Retrofit
  - Firebase Cloud Messaging
  - SendGrid



# Infrastructure Layer

## Purpose:

- Provides foundational support for hosting and deployment the backend APIs, models and user's data
- Storing temporary data for different services
- Storing resources like images, pdf in CDN



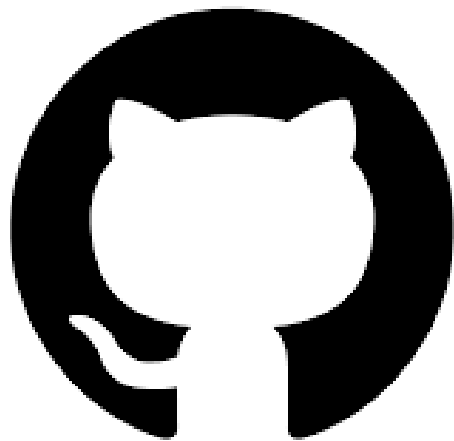
# Infrastructure Layer

- Components:

- Cloud platform
- Containerization
- CI/CD
- Monitoring
- Log management

# Infrastructure Layer

- Tech Stack:
  - AWS
  - Kubernetes
  - Jenkins
  - Github Actions
  - Prometheus
  - Splunk





Dealytics | Layered Architecture

# Thank You

Presented by Akanta Das