

TradeWatch UML Class Diagram

Contents

TradeWatch Application - UML Class Diagram	1
System Architecture UML Class Diagram	1
Architecture Overview	2
1. Presentation Layer	2
2. Page Components Layer	2
3. Dashboard Components Layer	2
4. Data Models Layer	3
5. API Integration Layer	3
6. Entity Services Layer	3
7. Utility Layer	3
Key Architectural Patterns	3
Repository Pattern	3
Observer Pattern	3
Strategy Pattern	3
Layered Architecture	4
Mobile-First Design	4
Data Flow	4
Technology Stack	4

TradeWatch Application - UML Class Diagram

VectorStream Systems - Global Trade Intelligence Platform

System Architecture Overview - Version 3.0

System Architecture UML Class Diagram

The following UML class diagram illustrates the complete architecture of the TradeWatch application, showing all major components, their relationships, and the data flow between different layers of the system.

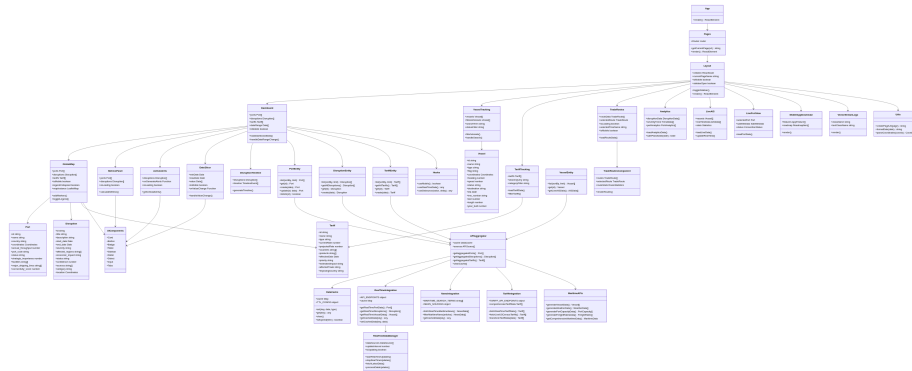


Figure 1: TradeWatch UML Class Diagram

Architecture Overview

The TradeWatch application follows a layered architecture pattern with the following key layers:

1. Presentation Layer

- **App**: Main application entry point
- **Pages**: Route management and page rendering
- **Layout**: Application layout with responsive navigation

2. Page Components Layer

- **Dashboard**: Main overview with maps and metrics
- **VesselTracking**: Real-time vessel monitoring
- **TariffTracking**: Tariff analysis and projections
- **TradeRoutes**: Trade route visualization and analytics
- **Analytics**: Advanced data analytics and trends
- **LiveAIS**: Live AIS feed monitoring
- **LivePortView**: Satellite port monitoring (coming soon)
- **MobileAppDownload**: Mobile app download and PWA setup

3. Dashboard Components Layer

- **GlobalMap**: Interactive Leaflet-based world map
- **MetricsPanel**: Key performance indicators
- **ActiveAlerts**: Critical disruption alerts
- **DateSlicer**: Time range filtering controls
- **DisruptionTimeline**: Timeline visualization
- **TradeRoutesComponent**: Route performance display

4. Data Models Layer

- **Port:** Global port entities with strategic importance
- **Disruption:** Trade disruption events with severity levels
- **Tariff:** Tariff data with projections to 2035
- **Vessel:** Vessel tracking with AIS integration

5. API Integration Layer

- **APIAggregator:** Central data aggregation with caching
- **RealTimeIntegration:** Live data from external APIs
- **NewsIntegration:** Maritime news aggregation
- **TariffIntegration:** Government tariff data sources
- **MaritimeAPIs:** Comprehensive maritime data services

6. Entity Services Layer

- **PortEntity:** Port data access and management
- **DisruptionEntity:** Disruption data operations
- **TariffEntity:** Tariff data handling
- **VesselEntity:** Vessel tracking services

7. Utility Layer

- **Utils:** Common utility functions
- **Hooks:** Custom React hooks for mobile detection and data fetching
- **VectorStreamLogo:** Company branding component
- **UIComponents:** Reusable UI component library

Key Architectural Patterns

Repository Pattern

Entity services provide a consistent interface for data access across different sources and APIs.

Observer Pattern

Real-time data updates flow through the system via reactive components and state management.

Strategy Pattern

Multiple API integration strategies with fallback mechanisms ensure reliable data access.

Layered Architecture

Clear separation of concerns across presentation, business logic, data access, and integration layers.

Mobile-First Design

Responsive components adapt to different screen sizes with touch-optimized interactions.

Data Flow

1. **User Interaction** → Page Component
2. **Page Component** → Entity Service
3. **Entity Service** → API Aggregator
4. **API Aggregator** → Real-time Integration
5. **Real-time Integration** → External APIs
6. **Response** flows back through the same layers with caching at each level

Technology Stack

- **Frontend:** React 18, Vite, Tailwind CSS
- **Mapping:** Leaflet.js with custom maritime overlays
- **Charts:** Recharts for data visualization
- **UI Components:** Radix UI with custom maritime theme
- **State Management:** React Hooks with context
- **API Integration:** RESTful APIs with intelligent caching
- **Mobile:** Progressive Web App (PWA) with React Native roadmap

© 2025 VectorStream Systems. All rights reserved. Patent Pending.