ChatGPT

Next-Generation Retail TMS: Features, Flows, and Blueprint

Trends and Technologies in Next-Gen TMS for Retail Logistics

Modern **Transportation Management Systems (TMS)** are evolving with cutting-edge technologies to meet the demands of retail supply chains. **Artificial intelligence (AI) and machine learning (ML)** are at the forefront, enabling smarter routing, dynamic pricing, and predictive analytics. AI-driven algorithms analyze real-time data (traffic, weather, delivery urgencies) to continuously optimize delivery routes and adapt to last-minute changes [1] [2]. This means faster deliveries, reduced fuel costs, and proactive responses to disruptions. ML-based demand forecasting helps shippers anticipate peaks and allocate resources efficiently [3]. Alongside AI/ML, **Internet of Things (IoT)** devices (GPS trackers, telematics sensors) provide real-time visibility of shipments' location and condition. For example, DHL has integrated IoT sensors with AI models and even blockchain to automate inventory tracking and improve decision-making [4]. **Blockchain** technology is another defining feature of next-gen logistics, used to enhance transparency and security in freight movements. Decentralized ledgers create an immutable record of each shipment's journey, reducing disputes and enabling "smart contracts" that automatically trigger payments when delivery conditions are met [5] [6]. In practice, major carriers like Maersk use blockchain-based digital contracts to eliminate paperwork and speed up cargo handoffs [4]. All these technologies contribute to a **"smarter, more dynamic" logistics network**, where the TMS can predict and preempt issues, rather than merely react [7]. Retailers also benefit from emerging tech like **autonomous delivery vehicles, drones, and robotics** in warehouses, which TMS solutions are beginning to integrate for route planning and scheduling [8] [9]. In summary, a *"next-gen" TMS* in retail leverages AI/ML for optimization, IoT for real-time visibility, blockchain for trust and automation, and cloud-based platforms for scalability and connectivity, turning transportation into a proactive, data-driven function of the supply chain.

Core TMS Modules and Features by Function

Modern TMS software is typically organized into **modules**, each covering a stage of the transportation process from order through settlement. In retail-focused TMS, certain features are especially important (like last-mile delivery support and omnichannel integration). Below we detail each primary module and its key features:

Order Management

This module handles the intake and preparation of shipment orders (the demand for transportation). It may integrate with Order Management Systems or ERP to pull in sales orders, transfer orders, purchase orders, and returns. Key features include capturing order details (origins, destinations, items, quantities), consolidating or splitting orders into shipments, and managing delivery requirements (time windows, service levels). Effective **order management ensures each order is properly tracked from creation to delivery**, serving as the bridge between sales/fulfillment and transportation [10] [11]. In a retail context, the TMS must support omnichannel order sources – e.g. e-commerce orders to consumers, store replenishment orders, and drop-ship vendor orders – combining them intelligently for transport. Real-time order visibility is crucial; a TMS provides status updates to the order system so that retailers know which orders are in transit or delivered [12]. This module often interacts with inventory and warehouse systems to confirm when goods are ready to ship. *Retail-specific needs:* The ability to pool orders by region/store, prioritize express e-commerce orders, and handle **rapid order updates** (like cancellations or address changes) is vital. A unified TMS can integrate with OMS/WMS so that order promises and transportation plans stay aligned, ensuring on-time fulfillment across channels [13] [14].

Planning & Optimization

The planning module is the "brain" of the TMS. It takes transportation requirements (shipments or orders) and determines the optimal way to fulfill them. **Route optimization and load optimization** are core functions – the TMS will automatically calculate the most efficient routes, delivery sequences, and load configurations for trucks or containers [15]. Advanced planning considers multiple factors: delivery deadlines, vehicle capacities, driver schedules, traffic, fuel costs, and any constraints like temperature control or road restrictions. For retailers, this often means optimizing multi-stop delivery runs (e.g. a truck delivering to several stores) and balancing cost vs speed for last-mile deliveries. The TMS's optimization engine may use AI/operations research techniques to perform **what-if scenarios** and find lowest-cost solutions that meet service targets [16]. Additionally, **carrier selection** (choosing the best carrier or mode for each shipment) is part of planning. A modern TMS can dynamically evaluate carriers' rates, performance, and capacity in real time to select the optimal option for each load [17] [18]. For example, it might automatically decide whether to send a store shipment via the company's own fleet or a 3PL carrier based on current costs and service levels. **Transportation network optimization** is another feature, looking at the bigger picture of network design: consolidating routes, using cross-docks or pool points, and planning backhauls (utilizing empty return trips) [19] [20]. *Retail-specific needs:* The planning module should handle **last-mile delivery optimization** – e.g., planning home deliveries with time-window promises and optimizing milk-run routes to retail stores. Machine learning enhances this by allowing continuous re-optimization up to the last minute, factoring in real-time events [21]. Also important is **omnichannel planning** – coordinating store deliveries and direct-to-consumer shipments jointly to maximize efficiency (for instance, using the same truck for store replenishment and nearby customer orders). Leading TMS solutions enable **multi-modal planning** (truck, rail, ocean, air) for global retail supply chains and can optimize international shipments as well [22].

Execution & Visibility

Execution modules put the transportation plan into action and manage it through completion. This includes **shipment dispatching**, carrier communication, real-time tracking, and managing exceptions during transit. Key features are: tendering loads to carriers (electronically via EDI/API or via carrier portals), scheduling pickups and deliveries, and providing drivers with the necessary info (routes, manifests). Once shipments are on the move, the TMS offers **real-time visibility** – capturing tracking events from carrier updates, IoT GPS devices, and driver mobile apps [23] [24]. Users (like a transportation dispatcher or a store manager awaiting a delivery) can see where a truck is and get ETA updates. The system generates alerts for exceptions (delays, temperature excursions for cold goods, etc.), enabling proactive intervention. Modern TMS execution supports a **control tower** view: a centralized dashboard of all in-transit shipments with status indicators and risk alerts [25] [26]. In retail, this is crucial for meeting tight delivery windows and providing customers with accurate delivery tracking. For last-mile execution, TMS often integrates with delivery apps or crowd-sourced delivery partners, coordinating gig-economy drivers or local couriers. **Proof**

of Delivery (POD) capture is another execution feature – drivers can upload signatures or photos, which the TMS records as confirmation for each stop. *Retail-specific needs:* **Last-mile delivery management** features are essential. This includes route deviation alerts (if a driver is running late or off route), customer notifications (text/email with live tracking links), and the flexibility to re-route or re-schedule on the fly for failed delivery attempts. Next-gen TMS solutions emphasize **customer experience** here – using AI to personalize delivery options and keep customers informed, which ultimately boosts satisfaction [27] [28]. Additionally, for brick-and-mortar retail, execution capabilities like **appointment scheduling** (coordinating dock delivery times at stores or warehouses) ensure smooth inbound flow and avoid congestion at loading docks [26].

Freight Settlement & Finance

After goods are delivered, the TMS handles the financial side of transportation. **Freight settlement** (or freight audit and payment) features ensure carriers are paid accurately and transportation costs are properly allocated. The TMS will rate each shipment using contracted carrier rates, calculate fuel surcharges or accessorial fees, and generate freight bills. It can then automate **invoice matching** – comparing carrier invoices to the expected cost, flagging discrepancies like overcharges [29]. By automating this audit, companies save time and avoid payment errors. If the retailer uses third-party carriers, the TMS can generate payment instructions to Accounts Payable or even pay carriers through an integrated platform. For in-house fleet operations, this module helps calculate internal costs per trip (driver hours, fuel usage) for accounting purposes. **Cost allocation** is another feature – assigning transportation costs to the right business unit or even splitting costs across multiple orders on a consolidated load. In a retail setting, freight settlement ensures that inbound freight costs from suppliers, inter-store transfer costs, and last-mile delivery fees are all tracked and visible to finance. Leading TMS solutions support **complex contracts and tariffs**, including international duties if applicable, and ensure **compliance with contracts** (billing exactly what was agreed) [30]. They also provide **analytics on freight spend**, feeding into cost reports (see Analytics module). *Retail-specific needs:* Retailers often ship high volumes with slim margins, so freight audit is critical to catch billing errors and optimize spend. The TMS might integrate with procurement systems for **freight procurement** (bidding and contracting carriers) and then enforce those rates during settlement [18]. Also, features to manage **claims** (for loss/damage in transit) add value – the TMS can generate claims documentation and track resolution with carriers. Finally, **reverse logistics settlement** is important: handling the cost of return shipments, which in e-commerce can be significant. A robust settlement module will support charging back return costs to the appropriate party or budget.

Analytics & Reporting

Analytics is a cross-cutting module that aggregates data from all other TMS functions to generate insights and performance metrics. Modern TMS platforms come with built-in **dashboards and reporting tools** that help measure key performance indicators (KPIs) like on-time delivery rate, freight cost per unit, fill rate, transit time, and carrier performance scorecards [31] [32]. Users can often customize these dashboards to see the metrics that matter for their role – e.g., a logistics manager might monitor daily delivery performance and budget vs actual freight spend, while a store operations manager might look at inbound lead times and any late deliveries to the store. **Real-time analytics** provide instant visibility (for example, highlighting shipments currently at risk of delay). **Historical analysis** helps with strategic decisions: a TMS might reveal which carriers consistently perform best or which lanes have the highest costs, supporting contract negotiations and network redesign. In retail, **analytics also tie into customer service** – e.g., identifying frequent delivery failures in a region could prompt adding a new carrier or adjusting route plans to improve service. Advanced TMS analytics increasingly use **AI for predictive insights**, such as predicting which shipments are likely to be late or which upcoming orders might not fit on planned trucks, so planners can act early [33] [34]. Moreover, TMS analytics assist in **sustainability tracking** – calculating carbon emissions from transportation and suggesting greener alternatives (some next-gen systems even integrate an emissions calculator) [35]. *Retail-specific needs:* The ability to analyze **last-mile performance** (e.g., percentage of on-time home deliveries, carrier success rate by region) is key for customer satisfaction improvements. An omnichannel retailer also benefits from analytics that combine store and online fulfillment metrics – a modern TMS can highlight, for instance, how ship-from-store deliveries are performing versus DC-to-customer deliveries. Also, retail supply chains often emphasize **OTIF (On-Time In-Full)** metrics demanded by big-box customers; a TMS can track OTIF for each route and carrier. In summary, the analytics module turns the vast data in a TMS into actionable improvements, from day-to-day operational tweaks to long-term strategy shifts [36] [37].

(Note: In practice, TMS vendors sometimes name or group modules differently. For example, "Execution" and "Tracking" might be separate, or "Carrier Management" might be a distinct module focusing on carrier contracts and performance. The modules above align with common functional groupings [38] .)

User Roles and Access Hierarchies in an Enterprise TMS

A robust enterprise TMS supports a variety of **user roles**, each with specific access permissions and interfaces tailored to their needs. In a retail organization, the following stakeholders typically use the TMS:

- **Logistics/Transportation Managers:** They are power users who oversee the entire transportation program. They need access to all modules – planning, execution, settlement, analytics – to configure the system, set up contracts, and analyze performance. They often have administrative privileges to set up carriers, lanes, and policies. For example, a transportation manager might set carrier tender rules or approve exception plans.

- **Transportation Planners / Dispatchers:** These users work in the planning and execution horizon. Planners build and adjust shipment plans, so they use the **Planning & Optimization** UI heavily (e.g., a planning cockpit to manually adjust routes or consolidate loads). Dispatchers focus on **Execution** – tendering loads to carriers, scheduling pickups, and monitoring in-transit shipments. They need real-time dashboards of current shipments and the ability to intervene (reroute, expedite, etc.). Their access might be limited to day-to-day operations (not changing master data or contracts).

- **Carrier Managers / Procurement:** Often part of the logistics team, these users manage relationships with carriers. They use features like carrier scorecards, tender acceptance rates, and cost performance reports [39]. They may also initiate **transportation procurement events** (bidding for new carrier contracts) if the TMS supports it. Their access includes carrier profiles, contracts, and perhaps the ability to adjust rate tables.

- **Warehouse and Store Managers:** These are internal users in distribution centers or retail stores who interact with inbound/outbound shipments. A DC **warehouse manager** might use the TMS to see inbound delivery schedules and prepare receiving docks (often via integration with WMS). Store managers might have limited access or receive automated notifications for shipments headed to their store. For instance, a store manager could log into a portal to see what shipments are en route and their ETA, helping them plan labor for unloading. These users typically have **read-only or**

confirmation access – e.g., confirming a delivery arrival or reporting issues – rather than planning authority.

- **Customer Service Representatives:** In retail, customer service teams may use the TMS (or an integrated interface) to answer customer inquiries about order deliveries. They require **tracking visibility** – the ability to look up an order or shipment and see its status, current location, and any exceptions. Their access might be restricted to search and view shipments, plus possibly initiate a return or re-delivery request in the TMS if a customer reports an issue.

- **Finance/Accounting Personnel:** The finance team uses the TMS for **freight payment and auditing**. They typically access the settlement module to review carrier invoices, approve payments, and run reports on transportation spend. They might not use planning or execution screens at all. Permissions ensure they can't alter plans but can update payment statuses or adjust accounting codes. Sometimes a separate **audit team** role is defined to handle freight invoice disputes and claims.

- **IT/System Administrators:** These users maintain the TMS application. They manage user accounts, role permissions, and handle integrations (with ERP, WMS, etc.). They often have the highest level of access (system config), but they might not be involved in daily operations. In an enterprise TMS, administrators ensure compliance with security policies, set up EDI/API connections to carriers, and manage system updates.

In addition to internal roles, an enterprise TMS extends controlled access to **external parties** through portals or role-specific logins:

- **Carriers/Drivers:** Many TMS solutions offer a **carrier portal** where carriers can log in to accept or reject tenders, input tracking updates, and upload documents (PODs, invoices). Carriers only see their own shipments and typically only the data relevant to their execution. For last-mile or private fleet operations, drivers might use a mobile app linked to the TMS to get their route, update statuses (e.g., "delivery completed"), and capture signatures. The TMS maintains a limited profile for each driver (often through the carrier company's account). Ensuring carriers have an easy interface encourages real-time status updates, which improve the retailer's visibility.

- **Suppliers/Vendors:** In inbound retail logistics, suppliers may interact with the retailer's TMS to manage inbound shipments. For example, a vendor could have access to create or view **inbound delivery requests** (if the retailer allows vendors to book their own freight through a TMS portal). At minimum, suppliers might receive automated notifications or reports on the status of shipments carrying their goods. A TMS with a supplier portal could let suppliers see when a pickup is scheduled at their facility and if any delays occur, enhancing collaboration.

- **3PLs and Logistics Service Providers:** If a retailer outsources some transportation to a 3PL or uses a freight forwarder for international shipments, those partners might be given access to the TMS as external users. Their role could be similar to an internal planner/dispatcher but restricted to the shipments or lanes they manage. For instance, a 3PL managing store deliveries in a certain region might log into the retailer's TMS to do execution for that region only.

- **Retail Customers (Consignees):** Typically, end customers don't access a TMS directly, but they receive data from it. For B2B retail partners (like a franchise store or wholesale customer), a TMS might provide a **web tracking page** where they can input a shipment ID or purchase order and see delivery status (this is usually fed by the TMS's tracking data). In B2C scenarios, customers get tracking links via the TMS integration with last-mile carriers. The TMS ensures customer-facing updates are accurate by pushing status updates to CRM or notification systems.

To manage all these users, an enterprise TMS implements a strict **access hierarchy and role-based permissions**. Each role is configured to see and do exactly what's needed. For example, a store manager's login might only allow viewing shipments destined to their store, preserving confidentiality across a large organization. The TMS admin can often customize roles (adding a role for "Regional Logistics Manager" who can see shipments across multiple warehouses, etc.). Audit trails are also key – the system logs who made what changes, which is critical in collaborative enterprise environments to maintain accountability and security.

In summary, an enterprise TMS environment supports a multi-tier user hierarchy: from executives needing high-level dashboards, to operational staff managing shipments, to external partners feeding into the network [39] [40]. The system's design must accommodate this breadth, ensuring usability for each role (e.g., simple mobile interfaces for drivers, analytical tools for managers) while protecting data through permission controls.

Operational Flows in Retail Transportation Management

Retail supply chains involve **several distinct logistics flows** that a TMS must orchestrate: inbound goods from suppliers, outbound distribution to stores or customers, and reverse logistics for returns. We map out each of these key scenarios to illustrate how a TMS coordinates the process:

Inbound Logistics (Supplier to Distribution Center)

Inbound logistics in retail covers the movement of products from suppliers or manufacturers into the retailer's distribution network (typically into distribution centers or warehouses). The TMS inbound flow often begins with a **purchase order (PO)** or replenishment order generated in the retailer's ERP. Once a supplier is ready to ship an order, the TMS creates a **transportation request** for that inbound load.

Planning: The TMS plans how to bring the goods in: it may consolidate multiple POs from nearby suppliers onto one truck (multi-pick route) or determine optimal schedules. For international shipments, this might involve planning ocean or air shipments to a port, then drayage and inland transport. The TMS selects a carrier or arranges the retailer's own fleet for pickup. If using a vendor-managed freight model, the supplier might book a delivery slot via the TMS portal. Retailers often require high coordination here because inbound deliveries affect warehouse operations – a good TMS plan evens out the flow (avoiding too many trucks arriving at once).

Execution: The TMS communicates pickup instructions to the supplier and carrier. For example, it might send a **pickup notification** to the supplier: "Carrier X will pick up 10 pallets on DATE at your facility." The carrier receives a tender with supplier address, weight, and any special handling instructions. As the supplier loads the goods, they typically provide an **Advance Shipment Notice (ASN)** – this can be fed into the TMS to confirm what's being shipped. The TMS tracks the inbound shipment in real time; the carrier updates status (e.g., “departed supplier”, “in transit”, “arrived at DC”). Real-time visibility allows the DC to prepare receiving docks and labor in advance [41]. If delays occur (traffic, etc.), the TMS triggers alerts so the retailer can adjust warehouse staffing or notify if any stockout risk.

Receipt and Settlement: Upon arrival at the distribution center, the warehouse (via WMS) confirms receipt, and this can loop back to update the TMS that the inbound order is delivered. The TMS then moves to freight settlement: verifying the carrier’s charge for that inbound load. It will ensure the rate charged matches the contract for that lane and load. If the TMS is integrated with the retailer’s procurement, it might also update the landed cost of goods. Any discrepancies (damaged goods, quantity differences) are logged, possibly affecting claims.

In retail, inbound TMS management is critical for maintaining in-stock inventory. Retailers often operate **just-in-time replenishment** (especially grocery with perishable goods) – TMS planning optimizes these deliveries to keep products fresh and avoid overstock [42]. For example, supermarkets rely on TMS to plan daily fresh produce deliveries overnight when traffic is low, ensuring shelves are restocked each morning. Another complexity is **multi-tier inbound**: goods might go from a supplier to an interim cross-dock facility. A next-gen TMS can plan such multi-leg routes and coordinate with **dock scheduling** systems so that inbound trucks have reserved times at the DC, minimizing wait and congestion [43]. Overall, the inbound flow managed by TMS reduces lead time and transportation cost – by consolidating loads, choosing optimal modes (full truckload vs LTL vs parcel), and avoiding rush expedites. When disruptions like late supplier readiness or port delays occur, the TMS provides visibility so contingency plans (like re-routing or using alternate suppliers) can be executed proactively.

Outbound Logistics (Distribution Center to Store or Customer)

Outbound retail logistics can be split into two streams: **DC-to-Store deliveries** (replenishing brick-and-mortar stores) and **DC-to-Customer** or **Online orders** (e-commerce fulfillment, which may ship from DCs or stores). A TMS often manages both, as they share transportation resources but have different requirements.

DC to Retail Store: Typically, stores place replenishment orders through the retailer’s inventory system, which cue shipments from the distribution center. The TMS aggregates all store orders due for delivery in a given period (say, daily or weekly routes). **Planning:** It will cluster store orders into efficient delivery routes – for example, a truck might leave the DC and make deliveries to 5–10 stores on a circular route (a multi-drop route). The TMS must respect each store’s receiving hours and any delivery appointment windows. It also maximizes truck utilization (volume/weight) while meeting those constraints. This often involves **continuous route optimization** where the TMS might recompute routes each day based on order volume and traffic forecasts [21]. For long distances or different regions, the TMS may assign different carriers or use line-haul to intermediate pool points. Once plans are set, the TMS either assigns the retailer’s own fleet vehicles or tenders the loads to contract carriers that service those store routes.

Execution: The TMS generates **loading manifests and delivery sequences** for each truck. At the DC, loaders follow this plan (often orchestrated by a WMS that communicates with the TMS on the loading order corresponding to the route). As the truck leaves, the TMS begins tracking. **Real-time visibility** is important to stores: if a truck is delayed, store managers need alerts (e.g., via text or a portal) so they can adjust staffing. Modern TMS solutions provide store-facing notifications – e.g., an automated email the night before with an ETA, and updates on the day of delivery if any delay occurs [26]. During transit, if the TMS is integrated with telematics, it continuously updates ETAs. On arrival, the driver (or store receiver) might use a mobile app to confirm delivery, perhaps scanning barcodes of goods delivered, which serves as Proof of Delivery. The TMS marks the store shipment as delivered, which can trigger the inventory system to recognize the stock as in-store.

Customer E-commerce Deliveries: These can be more complex due to last-mile constraints. Orders are captured on an e-commerce platform, then either routed to a **fulfillment center (FC)** or sometimes to a store (for ship-from-store) to be picked and packed. When ready to ship, the TMS steps in to plan and execute the delivery to the customer's address. If using parcel carriers (e.g., UPS, FedEx), the TMS might perform a **carrier selection and rate shopping** function: comparing parcel rates and transit times to choose the best service for each order. It can print shipping labels and dispatch packages via integration with parcel carrier systems. Increasingly, retailers use local couriers or their own delivery fleet for same-day or next-day delivery in metro areas – for those, the TMS will **batch customer orders into delivery routes** akin to a small milk-run, similar to how it plans store routes [44] [45]. This is where AI helps: learning constraints like traffic patterns and driver speeds, to efficiently group deliveries and sequence them.

Execution and Customer Experience: For home deliveries, execution is customer-facing. The TMS triggers **customer notifications** at various stages: e.g., "Your order has shipped – here's a tracking link" (which is powered by the TMS's tracking data). On the delivery day, customers may get a narrower ETA window or real-time van tracking on a map if the TMS integrates with a last-mile visibility tool. If a delivery attempt fails (customer not home), the TMS can log that as an exception and may either reschedule automatically or flag customer service. Modern TMS platforms often integrate with customer communications channels to allow rescheduling or delivery instructions (some have self-service portals for customers to choose a delivery time). After successful delivery, the TMS records the proof (signature, photo).

In the **store delivery** scenario, after execution the TMS may feed store delivery performance metrics (was it on time? was everything delivered?) into analytics or a vendor scorecard if third-party carriers are used. In **customer delivery**, the TMS often provides data back to the customer order system to trigger "delivered" status updates, which could prompt customer feedback surveys or simply close the order.

Retail outbound flows put a premium on **last-mile efficiency and reliability**. A next-gen TMS is crucial here: by dynamically optimizing routes up to the last minute and leveraging real-time traffic data, retailers can significantly improve on-time delivery rates to stores and customers [46]. Also, by unifying store and online fulfillment in one TMS, retailers can do **omnichannel optimization** – e.g., if a truck is going to a store, maybe it can also carry a few home deliveries for nearby customers, reducing duplication of routes. Few legacy systems could handle that, but newer TMS solutions are beginning to offer such capabilities as **multi-drop, multi-segment route planning** that mixes delivery types. Additionally, sustainability comes into play outbound: the TMS might route trucks to minimize empty miles and even account for lower emissions (some retailers plan "green delivery days" where shipments are consolidated – the TMS can facilitate that by planning fewer, fuller trucks).

Reverse Logistics (Returns and Exchanges)

Reverse logistics is a significant aspect of retail, especially with the growth of e-commerce where return rates are high (e.g., apparel). A TMS supports the **efficient return of goods** from stores or customers back to a return center, warehouse, or supplier.

Customer Returns (E-commerce): When a customer initiates a return (through a website or customer service), it kicks off a return order in the system. The TMS may get involved by providing return shipping options. In many cases, retailers include a return label in the original shipment or allow customers to generate one via a portal – the TMS's carrier selection logic ensures that label is for an approved carrier service and perhaps uses the retailer's bulk return shipping rates. Once the return package is handed to the carrier, the TMS (or integrated parcel tracking) monitors it similar to an outbound shipment, so that the retailer knows when to expect it back and can update the customer. Some advanced TMS/OMS combos even trigger immediate refund or exchange when tracking shows the item in transit back, to expedite customer service. If the item is large (e.g., furniture), the TMS may schedule a pickup from the customer by a 3PL – essentially planning a small reverse pickup route.

Store Returns and Transfers: Retailers often allow customers to return online purchases in store. In that case, goods accumulate at stores and need to be moved back to a central location or a clearance center. The TMS can help plan **backhaul routes** – e.g., when trucks deliver to stores, the TMS can plan them to **pick up return cartons from the store** on the way back to the DC. This efficient use of trucks (a backhaul) turns a reverse flow into part of the forward flow, saving cost. If not picked up by store deliveries, the TMS might schedule periodic dedicated return pickups from stores, especially for high-volume locations.

For goods that need to go back to vendors (defective or unsold merchandise under return agreements), the TMS again will plan either direct vendor pickups or consolidation at DC then vendor shipment.

Execution: During returns execution, visibility is again important. The TMS tracks return shipments so that the retailer's inventory systems know when returned stock will arrive (important for potentially reselling those goods). For international returns (e.g., returning products to an overseas supplier), the TMS handles documentation similarly to outbound but in reverse, including export paperwork from the origin country.

Cost and Settlement: Reverse logistics can incur costs that need tracking – e.g., who pays for return shipping. The TMS's settlement module can differentiate customer-paid returns vs retailer-covered returns. It can invoice the appropriate party or simply record it for internal cost analysis. In cases where returns go back to a supplier for credit, the TMS might help document the freight expense for chargeback to that supplier. Modern TMS solutions help streamline returns to be more cost-effective and even *customer-friendly*; for instance, **SAP TM** has features to make the returns process cost-effective and customer-centric, treating returns as "first-class" shipments rather than afterthoughts [47] .

Analytics in Reverse: Over time, the TMS can report on return flows: which lanes see the most returns, average transit times for returns, etc. This can help retailers improve their returns networks (e.g., opening more return centers or using faster methods for high-value items).

In summary, the TMS manages reverse logistics by providing visibility and control similar to outbound: planning the route or carrier for returns, tracking their progress, and settling the costs appropriately. An optimized TMS-driven returns process can even become a competitive advantage – for example, enabling a retailer to offer rapid turn-around on exchanges because the TMS helped set up a swift returns pipeline. As retailers pursue **circular economy** initiatives, TMS-managed reverse flows (for recycling or refurbishing goods) are also increasingly important. Whether it's a customer sending back a pair of shoes or a store returning unsold inventory, the TMS ensures the journey is logged, optimized, and integrated with the broader supply chain.

Major Enterprise TMS Solutions for Retail Supply Chains

Several enterprise-grade TMS platforms dominate in the market, each with strengths in different areas. Below is an overview of major TMS vendors and how their solutions cater to retail logistics:

- **SAP Transportation Management (SAP TM):** Part of SAP's broader supply chain suite, SAP TM is a highly integrated solution suited for large enterprises (especially those already using SAP ERP). It supports end-to-end transportation processes (inbound, outbound, domestic, international) within one system [48] [49]. For retail, SAP TM's strengths include **deep integration with SAP's order and warehouse modules**, providing real-time visibility of orders and deliveries in one place [12]. It offers advanced planning tools from strategic freight contracting down to load building and routing. Retailers benefit from its **analytics and visibility** – SAP TM facilitates carrier collaboration and network-wide visibility so retailers can track shipments and respond quickly to disruptions [41]. It has robust capabilities for **multi-modal shipping and global trade compliance**, useful for retailers importing goods worldwide. SAP TM also emphasizes **sustainability and compliance**, optimizing routes to reduce emissions and ensuring regulatory adherence (e.g., hazmat, driver HOS rules) [50]. A differentiator for SAP in retail is its ability to handle complex scenarios like **temperature-controlled shipments** and **last-mile delivery optimization** by considering traffic and delivery windows, which enhances customer service for e-commerce orders [51]. However, SAP TM is known to be **complex to implement** and often best suited for organizations willing to align with SAP's end-to-end process framework. Its UI and user experience have improved in recent versions (especially with Fiori apps), but new users face a learning curve. For SAP-centric retailers, though, the seamless data flow from ERP to TMS to warehouse is a big advantage.

- **Oracle Transportation Management (OTM):** Oracle's TMS is another top-tier solution, often chosen by large retailers and logistics providers. OTM is recognized for its **robust optimization engine and scalability** – it can handle high shipment volumes and complex global routes with ease [52] [53]. Oracle has designed OTM to manage **outbound, inbound, and inter-facility shipments all on one platform**, with strong capabilities in fleet management and freight payment. A key selling point for Oracle in retail is its **breadth of mode support and financial integration**. It excels in scenarios requiring tight control of costs and compliance: for instance, retail industries like food & beverage or pharmaceuticals (where traceability is key) often leverage OTM for its detailed shipment visibility and auditing features [54]. Oracle's solution integrates naturally with Oracle's ERP and WMS, but also can connect to other systems, and it's offered as a cloud service. Features like **Logistics Network Modeling** allow companies to simulate changes in their distribution network (helpful for retailers considering new DC locations or delivery models) [55]. OTM also provides unique visualization tools (e.g., a 3D load configuration view to maximize trailer space) which are useful for planning efficiency [56]. In the context of 'next-gen' criteria: Oracle has been updating OTM with more **AI-driven capabilities** (for ETA predictions, etc.) and has a strong **track record of reliability**. OTM is considered an **industry leader for enterprise TMS** [54], but like SAP, it can be complex and heavy – the **total cost of ownership is high**, and it's most suitable for large retailers with mature IT and the need for comprehensive functionality. Mid-sized businesses or those seeking simpler solutions might find OTM excessive for their needs.

- **Blue Yonder TMS (formerly JDA TMS):** Blue Yonder offers a TMS that is often praised for **planning and optimization strength** and its alignment with retail and manufacturing needs. Blue Yonder's TMS is part of a larger supply chain suite (now marketed with their Luminate platform), which means it can natively integrate with their demand forecasting, merchandising, and warehouse solutions. **For retailers, Blue Yonder stands out in supporting complex distribution operations and** control tower visibility**. It's known as a "powerhouse" in TMS, with an execution-focused design embedded in a comprehensive supply chain suite** [57] **. Retail enterprises (especially big-box and grocery) use Blue Yonder TMS to handle global freight as well as detailed store delivery planning. The platform leverages extensive** data partnerships for visibility**, meaning it can integrate external data (traffic, weather, carrier tracking feeds) to enhance its real-time picture** [57] **. Recently, Blue Yonder has invested in** AI and "cognitive" TMS features**: their latest release includes a multi-enterprise network for collaboration and** AI agents that provide predictive and prescriptive insights **via a conversational interface** [58] [25] **. This moves users from reactive management to a more proactive approach with decision support – e.g., suggesting re-routing around a disruption, or flagging a likely late shipment before it happens** [33] **. Blue Yonder's solution is fully cloud-based and built to scale, and the integration with the Blue Yonder Network means easier connectivity with carriers and partners** [59] **.** Strengths: **Great for retailers who want a unified planning + execution system, strong multi-modal capabilities, and cutting-edge AI features.** Weaknesses: **Some users have noted the UI and user experience of Blue Yonder TMS can lag behind modern standards (historically it had an older interface)** [60] **. Also, as part of a broad suite, it may be** costly and complex** if a retailer only wants standalone TMS functionality – it shines best when deployed as part of Blue Yonder's end-to-end supply chain solution for large enterprises [61] .

- **Manhattan Associates TMS:** Manhattan offers the **Manhattan Active® Transportation Management** system, which is a component of its supply chain execution suite (alongside WMS, labor management, etc.). Manhattan's TMS is **cloud-native and microservices-architected** in its latest incarnation, emphasizing adaptability and continuous updates. A hallmark of Manhattan's solution is its **unified platform** approach: transportation management is closely integrated with warehousing and order management in the Manhattan Active suite, which is very attractive to retailers looking for an all-in-one execution system [62] [63] . For example, Manhattan's TMS can seamlessly share data with Manhattan's WMS to coordinate when an outbound order is picked vs when the truck is scheduled – smoothing handoffs. Manhattan TMS has strong capabilities in **fleet management and last-mile** as well, partly due to Manhattan's experience with retail store distribution. It supports continuous route optimization, multi-stop routing, and has good tools for managing private/dedicated fleets (driver assignments, etc.) [64] . Manhattan is also recognized as a leader in the Gartner Magic Quadrant for TMS in recent years [65] . **Strengths for retail:** Manhattan's solution is built with omnichannel in mind; it can handle parcel, LTL, TL in one system and is **API-first** for easy integration (helpful for connecting to e-commerce platforms or carrier APIs) [66] . Its modern UI and workflow-oriented design are user-friendly, reducing training time for planners and dispatchers. The cloud model means even new features (like machine-learning based enhancements) become available to users regularly. **Potential drawbacks:** As a relatively newer re-architected platform, some extremely complex global features might not be as battle-tested as SAP or Oracle (though Manhattan has a long history in TMS as well). Also, companies not using Manhattan's WMS/OMS might not fully capitalize on the unification benefits. That said, Manhattan TMS is highly regarded for **retail and distribution-centric transportation**, and many retailers (especially in North America) leverage it for its balance of advanced functionality and modern, cloud-based delivery.

- **Descartes (MacroPoint, Aljex, etc.):** Descartes Systems Group offers multiple transportation solutions, including a robust shipper TMS and specialized products for carriers and brokers (e.g., Aljex for 3PLs). Descartes' strength is in **network connectivity and global logistics**. Their TMS is used by enterprises managing complex international, multi-modal supply chains [67]. Descartes has a global logistics network that makes integration with carriers (for EDI and tracking) relatively easier – for instance, their MacroPoint solution is an industry-leading real-time freight visibility platform, now often bundled or integrated with the TMS. For retail, Descartes is a good fit for those needing **international shipment management, trade compliance, and visibility**. It has strong customs and compliance modules (helping retailers ensure import/export rules are followed). Additionally, Descartes supports **route planning and home delivery optimization** through separate route planning tools that can tie into the TMS, which some retailers use for last-mile delivery optimization (e.g., routing delivery vans). **Strengths:** Very comprehensive for global logistics, good at **multi-leg planning** (like port to DC scenarios), and a leader in **real-time tracking** (with MacroPoint) which gives a control tower view across carriers. **Weaknesses:** Descartes' solutions can be somewhat fragmented (from acquisitions) – integration between the core TMS, the visibility module, and other pieces might require effort unless using their unified cloud platform. Also, Descartes often positions itself as a managed service or network, which might **limit flexibility** for some users who want a more DIY platform [68]. For smaller retailers, the cost might be prohibitive, and they may not need the full international capabilities. Moreover, Descartes doesn't have an encompassing supply chain suite for planning or WMS – so retailers might need to integrate it with other systems to get a full picture [69]. In summary, Descartes TMS is favored by companies with heavy-duty transportation needs (e.g., cross-border, multimodal, high regulatory requirements), including retail importers and 3PLs, but it might be overkill for domestic-only or medium-scale operations.

(Other notable TMS vendors include Echo Global (Command), MercuryGate, 3GTMS, e2Open and more, but the above are the ones specifically mentioned and are major players relevant to enterprise retail supply chains.)

Comparing TMS Vendors Against Next-Gen Criteria

When evaluating these vendors through the lens of "next-gen" TMS characteristics (the trends from section 1), we see both commonalities and differences:

- **AI and Predictive Capabilities:** All major TMS providers are incorporating AI/ML to some degree, but Blue Yonder has been especially aggressive in this area. Blue Yonder's latest TMS release embeds *cognitive algorithms and generative AI* to enable predictive planning and a conversational user experience [25]. This aligns with next-gen expectations of proactive, self-learning systems. Oracle and SAP are also adding AI for route optimization and ETA predictions, though often via add-on modules (Oracle's cloud platform integrates AI for real-time traffic and delay alerts, SAP leverages AI in its broader IBP and "digital supply chain" offerings to augment TM). Manhattan's TMS uses machine learning within its optimization and analysis (e.g., it learns from historical data to improve route efficiency and demand forecasts), and the Manhattan Active platform in general touts embedded intelligence [70] [71]. Descartes is leveraging AI mainly in its visibility/telematics domain to predict delays. **In summary:** All vendors recognize AI/ML is "table stakes" for next-gen TMS, but Blue Yonder currently markets the most advanced AI-driven UX, while Oracle/SAP emphasize AI within a stable, large-scale operations context.

- **Real-Time Visibility and IoT Integration:** Next-gen TMS demands end-to-end visibility. Here, Descartes (with MacroPoint) and Blue Yonder (with its network and partnerships) stand out. Descartes offers one of the most robust real-time tracking solutions, which is a reason many shippers use it [67]. Blue Yonder's multi-enterprise network approach similarly focuses on breaking data silos and providing unified visibility across partners [59]. Oracle and SAP have visibility capabilities (Oracle has a IoT Fleet Monitoring cloud service that can complement OTM, SAP offers Event Management and IoT integrations on its cloud platform). Manhattan integrates GPS and telematics for in-transit tracking and provides real-time alerts natively [23]. All are strong here, but **Descartes likely takes the crown on pure real-time multimodal visibility** due to its purpose-built network. Next-gen criteria also include IoT sensor data (for condition monitoring like temperature/security); SAP and Oracle have modules to ingest IoT data (leveraging their IoT Cloud or Edge services) but those may be separate from core TMS. Blue Yonder and Manhattan have partner integrations for IoT truck sensors. **Bottom line:** All vendors support real-time tracking, next-gen TMS should seamlessly integrate telematics – here the difference is often execution: Descartes and Blue Yonder have more out-of-the-box networks, others might need more integration effort.

- **Blockchain and Secure Collaboration:** Not all major TMS have directly integrated blockchain features yet. However, **Oracle** has been exploring blockchain in its cloud offerings for supply chain, and SAP has done pilots (SAP TM can interface with SAP's blockchain add-ons for track-and-trace). Blue Yonder's multi-enterprise network isn't explicitly blockchain, but it serves a similar purpose of a single source of truth among parties [26]. Some logistics tech companies (and possibly Descartes labs) have tested blockchain for freight bill audit and chain-of-custody, but it's not mainstream in products. This might be a gap that custom solutions could address if needed (e.g., integrating a blockchain ledger for high-value shipments).

- **Architecture (Cloud, API, Multi-tenancy):** Next-gen TMS are typically cloud-native, multi-tenant SaaS with open integration. **Manhattan Active TMS** is a prime example of modern architecture – microservices, multi-tenant cloud, with an API-first design [63]. Blue Yonder moved its TMS to SaaS as well and highlights a modern cloud architecture eliminating data silos [72]. Oracle OTM and SAP TM, historically on-premise heavyweights, are now offered in the cloud (Oracle OTM Cloud, and SAP TM as part of S/4HANA or SAP Cloud TM), but some would argue their core architecture is not as cloud-native as Manhattan's or newer entrants. Oracle's cloud TMS is highly scalable and supports multi-tenancy for 3PL scenarios (e.g., OTM can segregate data by business unit or client), but it's essentially the same powerful engine delivered via cloud. SAP TM in S/4HANA can be multi-tenant in a sense of handling multiple subsidiaries or logistics service provider scenarios with partitioned data. Descartes, being a network provider, has a multi-tenant network platform by nature and offers its apps on that cloud – this is great for quick integration of new partners. All vendors provide extensive APIs and support standard EDI formats for carrier comms. Manhattan and Blue Yonder particularly emphasize **easy integrations** – Manhattan explicitly encourages API connections to ERP, WMS, and external apps to avoid data silos [24]. FitGap's analysis notes that API-first, flexible integration is a key consideration for modern TMS, especially to connect with legacy systems and new cloud services [73] [74]. **Overall:** All five can integrate with other systems, but Manhattan and Blue Yonder (and MercuryGate in the mid-market) are often lauded for being more open and easier to integrate in a modular fashion, while SAP/Oracle might prefer you use their ecosystem for best results (though they do have APIs). A custom solution might aim to combine the raw power of an Oracle/SAP with the nimbleness and openness of a Manhattan approach.

- **User Experience and Automation:** A next-gen TMS should have an intuitive UI, and leverage automation to reduce manual work. Here, Manhattan Active TMS and newer UI of Blue Yonder are improving user experience. Blue Yonder's introduction of a conversational AI agent for TMS is a novel way to simplify user interaction [75]. Oracle and SAP have historically complex UIs, but they have modernized some (Oracle's OTM now has a configurable UI and better dashboards, SAP Fiori offers role-based simple apps for TM). Still, user feedback often indicates **Blue Yonder's older TMS UI was clunky** [60]; Blue Yonder has been working to modernize that as part of their Luminate platform. Manhattan's UI is generally considered good after their rework. Automation-wise, all these TMS offer workflow automation – e.g., automatic tendering, automatic re-planning when a truck is delayed (if configured), auto-invoice matching. Oracle and SAP being rule-driven and highly configurable can automate many processes but require expertise to set up. Blue Yonder and Manhattan focus on out-of-the-box intelligent automation (Manhattan, for example, can auto-adjust plans and notify planners only for exceptions; Blue Yonder's AI agents aim to automatically handle routine disruptions). **In short:** Legacy leaders (SAP, Oracle) are very capable but can be heavy to configure; newer designs (Manhattan Active, etc.) push more smart defaults and guided workflows. A custom solution could aim to marry the rich functionality of a legacy TMS with a streamlined, **user-friendly interface and automation** that reduces the need for constant human intervention.

- **Gaps a Custom Solution Could Address:** After comparing, some gaps/opportunities emerge. One is **agility and customization** – large commercial TMS can be rigid or expensive to tailor to unique needs. A custom next-gen TMS could be built more flexibly to adapt to novel business models (say, integrating with crowdsourced delivery platforms or new data sources faster). Also, **cost** is a factor: all these enterprise TMS are significant investments. A custom solution might target mid-sized retailers who need advanced tech (AI, etc.) but at lower scale and cost. Another gap might be **specialized last-mile and store integration features** – while vendors have modules or add-ons, a retailer might develop a custom TMS focusing specifically on last-mile optimization (for example, tightly integrating with in-store pickup scheduling or ride-sharing delivery options) that off-the-shelf systems don't natively support yet. Moreover, some users desire an **"independent" TMS** not tied into a 3PL's network or a broader suite, to avoid lock-in or conflict of interest in data [67] [69]. A custom TMS could prioritize neutrality and tailor algorithms purely to the retailer's needs without catering to multiple clients. Finally, while the big TMS solutions are adding AI, a custom-built platform could possibly leapfrog by incorporating the *very latest* machine learning techniques or blockchain features faster, since products have release cycles. Of course, building a TMS from scratch is a major endeavor, but focusing on unmet needs (like truly seamless omnichannel transportation management or ultra-granular carbon tracking on every route) could justify it if those needs are critical and not well-served by current vendors.

Enterprise-Grade Architecture and Non-Functional Requirements

Designing a "next-gen" TMS for the enterprise means meeting stringent architectural and **non-functional requirements (NFRs)** beyond just features. Some key requirements include:

- **High Availability and Reliability:** Transportation operations run 24/7, often across global time zones. An enterprise TMS must be highly available, minimizing downtime. This implies a cloud architecture with redundant servers across regions, failover mechanisms, and possibly active-active deployments to ensure if one data center goes down, the TMS stays up. Retail promotions or holiday seasons can't afford system outages that halt load planning or tracking. Typically, enterprise TMS solutions aim for **99.9% or higher uptime**, achieved through load-balanced web servers, clustered application servers, and replicated databases. **Disaster recovery** is also planned – e.g., the system can be restored in minutes or hours in a backup environment if a major failure occurs.

- **Scalability and Performance:** Retail supply chains can generate huge transaction volumes (think of large e-commerce players shipping millions of packages daily). The TMS architecture should be scalable horizontally – able to add computing resources on demand to handle peak loads (such as peak season planning or real-time updates on millions of shipments). The system should maintain responsiveness (UI load times, optimization run times) even as data volume grows. Modern designs use microservices and distributed processing to scale specific functions (for example, running route optimization algorithms on a distributed cluster for faster results). Performance tuning (fast in-memory computations, efficient algorithms) is crucial so that even complex optimizations execute in reasonable time. For instance, planners expect route plans to compute in minutes, not hours, even for hundreds of stops and constraints.

- **Multi-Tenancy and Segmentation:** If the TMS will serve multiple business units, clients, or regions, it should support multi-tenancy – logically (and securely) partitioning data and configurations by tenant. This is especially relevant if a retailer has sub-brands or an internal 3PL operation managing transport for external partners. A multi-tenant cloud TMS allows one software instance to serve all, simplifying upgrades while keeping data segregated [76]. Even within one company, **role-based data visibility** acts similarly: e.g., European logistics teams only see EU shipments, while HQ can see global. Multi-tenancy must be built with robust access controls so no user can see or affect another tenant's data.

- **Integration and API-First Design:** A TMS never operates in isolation – it integrates with **ERP systems (for orders, invoices)**, **WMS (for shipment release and receiving)**, **OMS/e-commerce platforms**, and with external partners (carriers, suppliers). Thus, an API-first design is critical: the system should expose a comprehensive set of web services (REST/JSON or XML) for every function – from posting new orders, to retrieving tracking updates, to rating a shipment [24]. Support for legacy integration methods like EDI is also important, since many carriers still use EDI X12 or EDIFACT messages for tendering, status, and invoicing. An enterprise TMS likely includes an integration middleware or at least configurable adapters for standard messages (204 tender, 214 tracking, 210 invoice in X12 terms, for example). Modern APIs also enable easier integration with emerging tech – for instance, connecting to a mapping service for real-time traffic, or consuming forecast data to adjust plans. **Event-driven architecture** is also a plus: the TMS can publish events (like "Shipment Delivered" or "Truck Delayed") that other systems subscribe to, enabling a responsive supply chain nerve center. In short, **smooth data flow** is a must – any custom TMS must remove data silos and ensure frictionless communication with the rest of the IT ecosystem [77] [78].

- **Security and Compliance:** Transportation data is sensitive (it can include customer addresses, product information, freight costs) and must be protected. An enterprise TMS requires robust security: **role-based access control** as discussed for user roles, data encryption (in transit via HTTPS and at rest in databases), and audit trails for who did what. Multi-factor authentication for users, especially external ones, is often required. Compliance with standards like **SOC 2, ISO 27001** for data security and **GDPR** for privacy (if handling personal data like customer info in the EU) must be built in. Additionally, **regulatory compliance** in transportation (hazmat regulations, customs requirements, hours-of-service for drivers) should be supported in the system's logic [79]. For example, the TMS might restrict planning to legal driver hours or include country-specific transportation rules. Security also extends to ensuring the system can't be manipulated – for example, if providing a carrier portal, one carrier should never be able to see another's shipments (a clear data partition). Periodic security audits and penetration testing would be part of maintaining an enterprise-grade TMS service.

- **Monitoring and Supportability:** Enterprises need tools to monitor the health of the TMS. This includes application performance monitoring (to catch integration errors, slow queries, etc.), and business activity monitoring (to ensure, say, that all expected EDI messages have been processed). A custom TMS should have a well-structured logging system and possibly a dashboard for system admins to see queue backlogs, interface status, and any errors. High supportability also means clear error messages and the ability to reprocess or fix data (for example, if a carrier tender fails, support can intervene). *On the operations side,* supportability includes the ability to configure business rules without code – e.g., an admin user can update a carrier rate or add a holiday closure for a warehouse through a UI, rather than needing a developer. This reduces maintenance burden and improves agility.

- **Configurability and Extensibility:** Retailers often need to adapt business rules (like how to assign orders to shipments, or how to prioritize carriers). An enterprise TMS should be highly configurable: supporting user-defined rules, workflows, and data fields without altering core code. For instance, one retailer might want to prioritize cost over speed in planning, another the opposite – the TMS should allow adjusting such parameters easily. Extensibility refers to the ability to plug in custom algorithms or modules. A next-gen TMS might allow custom optimization plug-ins (perhaps using a retailer's proprietary logic) or adding new integration endpoints fairly easily. For a custom-built TMS, using a microservices architecture with defined interfaces can help different teams extend functionality independently.

- **Analytics and Big Data Handling:** As mentioned, the TMS generates a lot of data. The architecture should include a strategy for analytical reporting that doesn't bog down the transactional system. This could mean a separate data warehouse or using cloud big data tools to offload and crunch large datasets (like years of shipment history for trend analysis). The system should be able to feed a business intelligence tool with data or have built-in analytics modules that can run complex queries efficiently. Given the rise of AI, an enterprise TMS might also incorporate machine learning pipelines – for example, retraining a delivery time prediction model on fresh data periodically. The architecture should accommodate this (perhaps by integrating with cloud ML services or having in-built AutoML for key use cases).

- **Testing, Quality, and Change Management:** In an enterprise environment, any updates to the TMS must be carefully managed. This means having **separate environments** (development, QA, staging, production) and automation for testing (maybe simulation of shipments to test new optimization logic). A custom TMS should have a suite of automated tests (unit, integration, performance) to ensure that adding new features or scaling up won't break existing functionality. For vendor software, retailers often use support sandboxes to test new patches or releases. In a custom build, adopting DevOps practices and continuous integration/deployment (CI/CD) can help deploy updates in a controlled, reversible manner.

In essence, an *enterprise-grade* TMS is **robust, secure, and scalable by design**. It's not just about solving routing problems; it's about doing so consistently under heavy load, with zero tolerance for critical failures, and fitting seamlessly into a complex IT landscape. Meeting these NFRs is what differentiates a prototype or basic TMS from one that can run the logistics of a Walmart or Amazon-scale operation. Any blueprint for a next-gen TMS must prioritize these architectural goals alongside functional innovation.

Blueprint for a Custom Next-Gen Retail TMS

Bringing together all the research, we can outline a **structural blueprint** for building a custom next-generation TMS tailored to retail:

1. Modular Architecture: The system should be organized into clear modules mirroring the transportation lifecycle – **Order Management, Planning & Optimization, Execution & Tracking, Settlement, Analytics** – as described earlier. Each module encapsulates its logic but shares data through a common platform. For example, an Order Management service creates freight units (shipments) that the Planning service consumes. This modularization aligns with the domain-driven design and allows focused development and scaling of each part (e.g., you can scale out the optimization engine separately if it's compute-intensive).

2. Unified Data Model & Integration Layer: At the core, have a unified data model for shipments, orders, routes, etc., stored in a scalable database. On top of this, build an integration layer (set of APIs and message queues) so that all internal modules and external systems communicate through well-defined interfaces [24]. For example, when an order is placed in the ERP, it calls the TMS API to create a shipment requirement; when a carrier sends a tracking update, it posts to a webhook/endpoint on the TMS. This API-centric approach ensures **extensibility** – new channels (a new e-commerce front-end or a new carrier API) can be connected without deep changes. Use of an enterprise service bus or cloud messaging (like AWS SNS/SQS, Azure Service Bus) can help manage event-driven flows (publish/subscribe model for events like "Shipment Dispatched" to notify stores or customers).

3. Intelligent Planning Engine: Implement the Planning & Optimization module with state-of-the-art algorithms (and make it **pluggable** for future improvements). This engine could use linear programming or constraint solvers for core routing, enhanced with machine learning for time estimations and dynamic inputs. It should support **multi-modal and multi-stop optimization**, considering costs, time, and resource constraints simultaneously [80]. A custom solution might use an open-source solver (like Google OR-Tools) as a base, extended with custom heuristics for retail (e.g., ensuring store deliveries happen during open hours, grouping deliveries geographically by neighborhoods for last-mile). The planning engine should run in a microservice that can scale out (multiple optimization jobs in parallel) and possibly utilize cloud computing (serverless or distributed computing for heavy computation). The output of planning – optimized loads and routes – flows into Execution.

4. Real-Time Execution Platform: The Execution module should be event-driven and **real-time at its core**. It will manage state of each shipment (Created -> Planned -> In-Transit -> Delivered, etc.) and update as events come. A suggested design is to use a streaming/event platform (like Kafka) where each shipment is a stream of events, and the Execution service processes those to update status and trigger actions. For instance, a "Pickup Confirmed" event from a carrier triggers an update that the shipment is now in transit, and maybe notifies the customer. Build a **control tower dashboard** for users on top of this, showing color-coded shipment statuses and allowing intervention (e.g., if a delay event comes in, a planner can choose to re-route via a different DC, invoking the planning module in partial). Incorporate **IoT integration** here: e.g., if a truck's temperature sensor goes out of range, the Execution module gets that alert and flags an exception. The system should automatically perform common tasks – e.g., auto-tender a load to the next carrier if the primary rejects it, or send ETA updates to stores and customers without human involvement. Empower it with **business rules engine** to handle exceptions (like "if delay > 2 hours AND type = store delivery, then notify store and consider split shipment").

5. Seamless User & Partner Portals: Provide tailored UIs: a web or mobile interface for internal users (planners, managers) with rich functionality (planning cockpit, analytics dashboards), and lightweight web/mobile portals for external partners (carrier portal for tendering and status input, supplier portal for ASN and pickup requests, etc.). The design should emphasize **usability** – intuitive screens, map visualizations for routes, drag-and-drop re-planning, and alert-driven workflows (the system highlights what needs user attention). Given many users will be on the go (like a transportation manager or a driver), responsive design and mobile apps are important. For example, a driver mobile app can integrate with the Execution module to send GPS positions and allow capturing signatures; a store manager might have a simplified view on a mobile device of inbound deliveries.

6. Advanced Analytics & AI Integration: Alongside the operational system, build an analytics pipeline. This could involve a separate data store (data warehouse or data lake) where all historical shipments, route plans, costs, etc., are stored. Use this for generating both **standard reports** (delivery performance, freight spend by carrier, etc.) and feeding **machine learning models**. For instance, develop an AI model that predicts transit times more accurately than static estimates by learning from historical patterns [81]. Integrate that model into the Planning phase to better estimate arrival times. Another model could predict which shipments are at risk of delay (based on current factors like weather, carrier performance), allowing proactive rerouting – align this with next-gen trend of predictive logistics [3]. The blueprint should include an **ML Ops process**: continuously train models on new data, deploy updates to production, and monitor their accuracy. Also, provide analytics back to users in a friendly way, e.g., *"Analytics & Reporting"* dashboards with interactive filters (perhaps using a BI tool or custom UI). Include **sustainability metrics** (CO2 emissions estimator per shipment) in analytics, as retailers are increasingly tracking Scope 3 transportation emissions [35].

7. Integration of Emerging Tech: Keep the architecture flexible to plug in emerging technologies. For blockchain, one could incorporate an *optional* distributed ledger component for certain high-value shipment lanes or partner integrations requiring tamper-proof logs – for example, use Hyperledger or Ethereum smart contracts to log custody events and automate freight payments upon delivery [82] [83]. If this is built in a modular way (only used for those who need it), it future-proofs the TMS for when more partners adopt blockchain. Similarly, design the system to accommodate **autonomous vehicles and drones**: for instance, if in the future a drone delivery is scheduled, the TMS planning should be able to treat it as just another mode of transport with certain capacity and range. This means mode-agnostic design – a new mode can be added to the system's master data without breaking logic. The Execution module should be able to ingest telemetry from autonomous fleets directly for tracking.

8. Security, Compliance, and Controls: Make security a foundational part of the blueprint. Use OAuth2 / JWT for API security so that every integration is authenticated and authorized. Implement fine-grained roles as discussed, and perhaps include an **"audit module"** that records all changes and can produce compliance reports (e.g., a log of all deliveries and any access to their data for GDPR compliance). Enforce **compliance rules** via the system – e.g., ensure no route violates driver HOS by building that into planning constraints, ensure hazmat handling rules by having a field on orders that triggers special planning logic, etc., so the

TMS actively helps compliance [79]. Plan for **data privacy** if needed (mask personal data like customer phone numbers in the driver's app except when needed, purge data as per retention policies, etc.).

9. Testing and Simulation Environment: A next-gen TMS blueprint should include a **simulation mode** where planners can simulate changes (like what-if scenarios: "What if we use a different carrier or change a route?") without affecting live data [55]. This sandbox environment can use production-like data to let the logistics team experiment with network changes (opening a new DC, etc.) using the same algorithms that run the live system. This is invaluable for continuous improvement. Also, build automated testing harnesses – e.g., simulate a day's worth of orders and shipments to verify that the system handles it with expected outcomes – especially whenever new code is deployed.

10. Incremental Deployment and Microservices: Structure the development and deployment so that components can be updated independently. For example, if a better route optimization service is developed, it can be deployed as the new Planning microservice without affecting Execution or Settlement. Use containerization (Docker/Kubernetes) to deploy these services, enabling scaling and resilience. This approach also helps if one piece fails – e.g., if the optimization service has an issue, it doesn't crash the tracking service; the system can degrade gracefully (maybe planners switch to manual mode temporarily but execution and visibility still continue). **Continuous integration/continuous deployment (CI/CD)** pipelines will ensure new features or fixes roll out frequently in small increments, which is a hallmark of next-gen software vs. monolithic annual releases.

Competitive Differentiators: By following this blueprint, the custom TMS would position itself with several differentiators against existing vendors. It would be *truly omnichannel-focused*, treating store deliveries and e-commerce with equal priority, whereas some legacy systems were stronger in one or the other. It could offer a **modern user experience** (think a single-page web app with drag-and-drop scheduling, real-time maps, chatbots for querying shipment status) that outclasses older UIs. Its **AI-driven core** (optimization and predictive alerts) would meet the next-gen standard, possibly even pushing beyond current vendor capabilities by quickly adopting new algorithms or data sources (for example, integrating crowd-sourced traffic incident data or weather forecasts directly into planning). As a custom build, it could also be tightly tailored to the retailer's unique processes – for instance, if a retailer has a unique way to prioritize store vs online orders during capacity crunch (e.g., always favor stores for brand reputation), that rule can be encoded from the start, whereas off-the-shelf might not have that nuance. Additionally, a custom TMS could incorporate **cost-saving innovations** like dynamic load pricing – using algorithms akin to ride-share pricing to decide when to send what, which not many TMS do yet. Finally, by adhering to enterprise NFRs (scalable, secure, integrable), this blueprint ensures the system is not just innovative but also reliable for critical operations.

In conclusion, a next-gen retail TMS built on this blueprint would be a **modular, intelligent, and resilient platform**: one that uses the latest tech to optimize every mile, provides total visibility from supplier to customer, supports all stakeholders with tailored interfaces, and evolves seamlessly as the retail landscape changes. It would effectively combine the strengths of leading vendor solutions (robust functionality, global reach) with the agility of modern software development (cloud-native, AI-infused, user-centric design), truly empowering the retail supply chain with a competitive edge.

Data Flow Diagrams (Level 0, 1, 2)

To visualize the structure and data flows of the TMS, we present Data Flow Diagrams at three levels of detail: **Level 0 (Context)**, **Level 1 (Top-Level Processes)**, and **Level 2 (Drill-down of Execution process)**. These diagrams illustrate how the TMS interacts with external entities and the flow of information within the system.

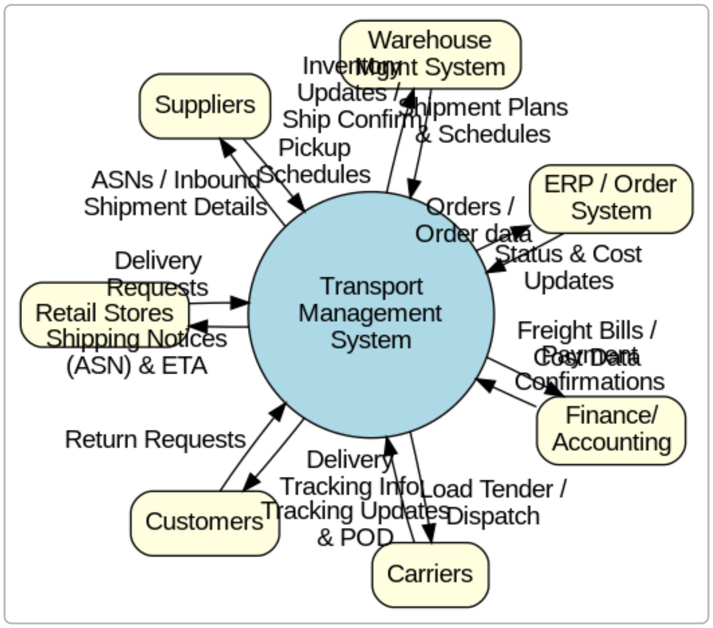

Figure 1: DFD Level 0 – TMS Context Diagram. This diagram shows the TMS as a single process (center) and its interactions with external entities. In a retail environment, the TMS exchanges data with numerous systems and partners. For example, it receives **Order data** from the ERP/Order System and sends back **status and cost updates** once shipments are executed [10] [14]. It plans inbound pickups with **Suppliers** (sharing pickup schedules and receiving ASNs), coordinates with the **Warehouse Management System (WMS)** to get inventory readiness updates and provide shipment plans [84] [14], and communicates with **Carriers** by tendering loads and obtaining tracking updates (including proof of delivery) [18] [29]. For last-mile delivery, the TMS interacts with **Customers** by sending delivery tracking information and receiving return requests. The **Finance/Accounting** system is updated with freight bills and payment confirmations once the Settlement process is completed. This context diagram highlights the TMS as the central hub that connects all logistics stakeholders, ensuring data flows seamlessly across the retail supply chain.

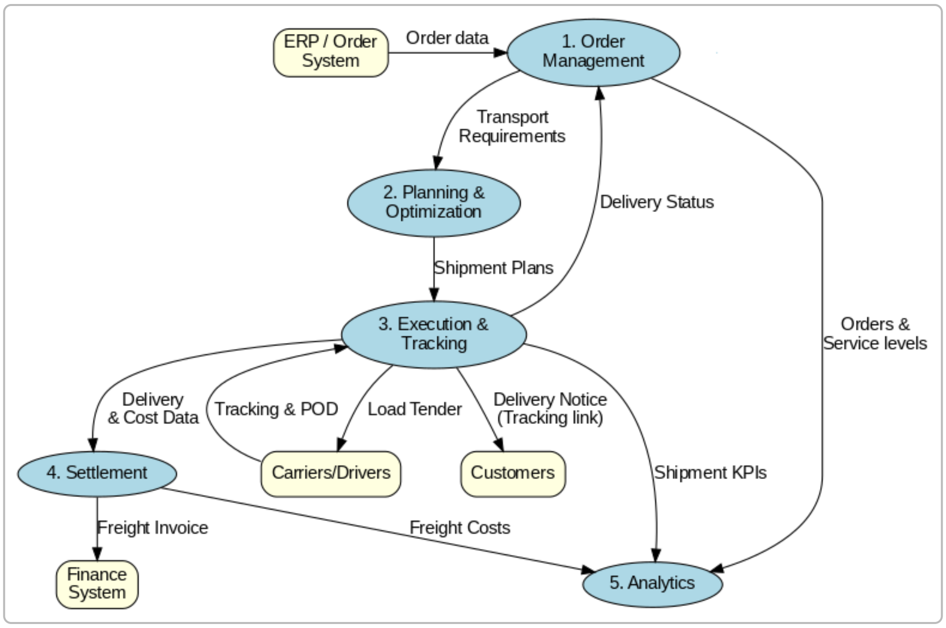

Figure 2: DFD Level 1 – High-Level TMS Processes and Data Flows. This diagram breaks the TMS into its primary functional processes and shows the data stores/flows between them and external entities. The main processes (blue ovals) correspond to modules: **1. Order Management, 2. Planning & Optimization, 3. Execution & Tracking, 4. Settlement, 5. Analytics**. Arrows indicate key data flows. For instance, the Order Management process receives **Order data** from the ERP and outputs **Transport Requirements** (shipment orders) to the Planning module. Planning then produces **Shipment Plans** (routes, carrier assignments) for Execution [85] [86]. During Execution, the TMS communicates with **Carriers/Drivers** – sending **Load Tenders** and receiving **Tracking updates & POD** (proof of delivery) events in return [18] [29]. Execution also sends **Delivery Notices (ASN) & ETA** to Retail Stores (or customers) to inform them of incoming shipments [23]. Once delivery is complete, the Execution process provides **Delivery & Cost Data** to the Settlement process, which then generates a **Freight Invoice** for the Finance system. Note the feedback loop: Execution sends **Delivery Status** updates back to Order Management (to update orders as shipped/delivered) and also feeds performance and event data to the Analytics process. The Analytics module aggregates data like **Orders & service levels** from Order Mgmt, **Shipment KPIs** from Execution, and **Freight Costs** from Settlement to produce reports and insights [31] [87]. This Level 1 DFD illustrates how each module interacts in an enterprise TMS: it's a continuous flow from orders to planning to execution, with monitoring and financial closure, all underpinned by feedback loops (status and analytics) that drive continuous improvement.

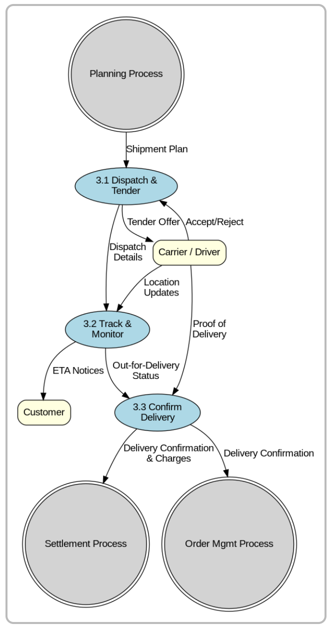

Figure 3: DFD Level 2 – Detailed Flow within the Execution & Tracking Process. This diagram zooms into the **“3. Execution & Tracking”** process (from Level 1) to show sub-processes and interactions in managing shipments. We’ve split Execution into three main sub-processes: **3.1 Dispatch & Tender**, **3.2 Track & Monitor**, and **3.3 Confirm Delivery**. The Execution phase kicks off with a **Shipment Plan** input from the Planning process (e.g. a planned route or load) arriving at Dispatch & Tender. In **3.1**, the TMS dispatches the shipment by sending a **Tender Offer** to the external Carrier/Driver and receiving an **Acceptance** (or rejection) [18]. Once accepted, Dispatch provides **Dispatch Details** (like pickup times, manifests) to the carrier. Next, in **3.2 Track & Monitor**, the TMS continuously receives **Location Updates** from the Carrier/Driver (via telematics or carrier status messages) and sends out **ETA Notices** to Customers or stores awaiting the delivery. The system monitors these updates for exceptions – if a delay is detected, it could trigger an alert or even invoke re-planning. As the delivery nears completion, **3.2** passes an **Out-for-Delivery Status** to the Confirm Delivery stage. In **3.3 Confirm Delivery**, the final proof of delivery comes in (the driver sends a **POD** or the customer signs on a mobile device). This sub-process then outputs **Delivery Confirmation & Charges** to the Settlement process and a **Delivery Confirmation** back to Order Management, closing the loop that the order was fulfilled. The gray double-circle entities represent external or higher-level processes interfacing here: the **Planning Process** feeding in the plan, and the **Settlement/Order Mgmt processes** receiving the outcomes. This Level 2 DFD highlights the real-time, event-driven nature of execution: constant communication with carriers, status updates to customers, and immediate handoff of delivery data to financial and order systems. It also underscores how **exception handling** is integral (though not explicitly drawn, one can imagine a loop from 3.2 back to 3.1 if re-dispatch is needed, or an alert out of 3.2 to a planner in case of issues). Overall, the Level 2 view provides a granular look at how a TMS orchestrates the complex choreography of getting a shipment from dispatched to delivered in the retail world, ensuring every step is tracked and communicated.

Sources: The diagrams above are based on standard TMS process flows and the specific retail-focused interactions discussed in the analysis [84] [29] [31]. They serve as a visual blueprint tying together the system modules, external touchpoints, and data movements that have been detailed throughout this report.

---

1 2 7 8 9 21 27 28 34 46 81 Next-Gen Logistics: AI and Innovation Trends in 2024 - Inbound Logistics
https://www.inboundlogistics.com/articles/next-gen-logistics-ai-and-innovation-trends-in-2024/

3 4 5 6 82 83 Navigating Tomorrow's Supply Chain: How AI and Blockchain are Revolutionizing Freight Solutions in 2024 - Advanced Logistics
https://advancedlogistics.us/navigating-tomorrows-supply-chain-how-ai-and-blockchain-are-revolutionizing-freight-solutions-in-2024/

10 11 13 14 32 64 84 What is a Transportation Management System (TMS)? | Manhattan
https://www.manh.com/solutions/supply-chain-management-software/transportation-management/what-is-a-transportation-management-system

12 Managing Your Supply Chain with SAP Business Suite | Clarkston Consulting
https://clarkstonconsulting.com/insights/supply-chain-with-sap-business-suite/

15 18 20 23 24 29 30 31 36 37 38 43 73 74 77 78 87 Essential Features of a Modern Transportation Management System | Manhattan
https://www.manh.com/our-insights/resources/articles/key-modules-and-features-transportation-management-system

16 17 39 40 76 79 80 Best transportation management systems (TMS) 2025 | FitGap
https://us.fitgap.com/search/transportation-management-systems-tms/advanced?page=3

19 85 86 Oracle Transportation Management Features, Modules & Capabilities [2025] | TEC
https://www3.technologyevaluation.com/selection-tools/features-list/46101/oracle-transportation-management?srsltid=AfmBOooNPKqaUTto7-FqApNuu__cP7w8k6BlJkrFqgYt9gS5jEEXO4SA

22 48 49 Explaining the Transportation Management Features and Functions
https://learning.sap.com/courses/business-processes-in-sap-s-4hana-transportation-management/explaining-the-transportation-management-features-and-functions

25 26 33 35 58 59 72 75 Blue Yonder Enhances Transportation Management Solution | Supply & Demand Chain Executive
https://www.sdcexec.com/transportation/wms-tms/news/22950479/blue-yonder-blue-yonder-enhances-transportation-management-solution

41 42 47 50 51 SAP TM Implementation for the Retail Industry | LeverX
https://leverx.com/newsroom/sap-tm-for-the-retail-industry

44 45 New Technology in Logistics Trends to Keep an Eye on in 2024
https://relevant.software/blog/new-technology-in-logistics-trends/

52 53 55 56 What Is Oracle Transportation Management? : Argano
https://argano.com/insights/articles/what-is-oracle-transportation-management.html

54 57 61 67 68 69 Top 10 TMS Systems In 2024
https://www.elevatiq.com/post/top-tms-systems/

60 Blue Yonder Reviews, Ratings & Features 2025 | Gartner Peer Insights
https://www.gartner.com/reviews/market/transportation-management-systems/vendor/blue-yonder

62 63 66 70 71 Retail and Supply Chain Software Solutions | Manhattan
https://www.manh.com/

65 Manhattan Associates Named a Leader in 2024 Gartner® Magic ...
https://www.businesswire.com/news/home/20240402575322/en/Manhattan-Associates-Named-a-Leader-in-2024-Gartner-Magic-Quadrant-for-Transportation-Management-Systems-for-the-Sixth-Consecutive-Year