3. Cumulative Tariff Layers: Section 301 and Anti-Dumping

A fundamental requirement for the Landed Cost Calculator is the ability to handle "stacked" or "cumulative" tariffs. Section 301 duties on Chinese imports, for instance, apply on top of normal MFN rates.[12] Furthermore, the system must account for the US Department of Commerce's final determination in Feb 2026 regarding Chinese graphite anode materials, where total penalties reached 220.18% through the combination of IEEPA, Section 301, Section 232, and anti-dumping/countervailing duties.[26]

$$Total\_Duty = MFN + Sec301 + Sec232 + AD/CVD$$

The architecture must implement this formulaic summation, ensuring that each component is individually verifiable back to its originating legal proclamation.[12]

Frontend Architectural Design: Intuitive Strategic Interface

The frontend architecture is designed to empower strategic decision-makers with a high-impact, visual-first experience modeled on contemporary high-volume booking engines.[32] As per the user requirements and the analysis of the provided visual cues from turn 1, the interface is structured around a "Hero" search and calculation bar.

1. Unified Trade Search Bar (The redBus Paradigm)

The core input mechanism follows the layout of the reference image, modified for international trade parameters.

- **Origin/Destination Fields:** Instead of "From" and "To" for bus stops, these inputs represent "Exporting Country" and "Importing Country." Each field includes a country-specific icon (e.g., a flag or container port icon) and a dropdown selection menu for precise identification.[32]
- **Classification Selection Logic:** A central toggle allows the user to choose between "HS Code" (6-digit international) or "HTS Code" (10-12 digit country-specific). The architecture implements a state-driven logic gate: if the "HS Code" button is selected, the "HTS Code" input becomes non-interactive (grayed out or hidden), and vice-versa.[33] This prevents misclassification errors where a user might input a 6-digit code into a 10-digit field.[34]
- **Date and Volume Inputs:** Placeholders for "Shipment Date" and "Volume/Weight" are integrated into the capsule-shaped bar, allowing the system to query time-sensitive freight rates and seasonal agricultural tariffs.[32]
- **The Calculate Button:** A prominent, high-contrast "Calculate Landed Cost" button triggers a POST request, serializing the input state into a structured JSON payload.[37]

2. Interactive Map and Curved Visualization

The system generates a dynamic map depiction as the primary visual output. Using the React-Leaflet library with a geodesic or Leaflet.curve plugin, the frontend draws a curved line connecting the centers of the two selected countries.[40]

- **Geodesic Logic:** The line represents the real minimal "great circle" path between the coordinates, appearing curved on the 2D Mercator projection.[40]
- **Quadratic Bézier Curves:** For visual clarity and branding, the system calculates a control point P perpendicular to the midpoint of the linear distance between coordinates A and B, offset by an angle θ_{offset} (typically 25 degrees).[40]
- **Directional Rendering:** To prevent overlap in complex multi-hop routes, the architecture implements a rule where horizontal lines wider than they are tall are curved upward (left-to-right rendering), and vertical lines are curved to the right (top-to-bottom rendering).[43]

3. Chatbot Integration and Notification Architecture

The frontend includes a persistent AI Assistant icon, typically positioned in the bottom-right corner.[44]

- **The "Save Money?" Popup:** This is an asynchronous notification component triggered by the AI Assistant Langflow.[44] If the pathfinding algorithm identifies an alternative route or supplier with a lower landed cost (e.g., switching from a 25% tariff Chinese supplier to a 0% tariff Vietnamese or UAE-based partner), the system pushes a floating notification to the user.[36]
- **State-Linked Chat Initiation:** When the user clicks the notification or the chat icon, the window expands and initializes a session-linked conversation. The frontend passes the session_id and the context of the current calculation (JSON data) to the assistant to ensure it doesn't lose the context of the user's initial query.[44]

System Backend and Langflow Orchestration

The system's intelligence is distributed across two distinct Langflow instances: the **Main Trade Processor** and the **AI Assistant Advisor**. These are coordinated through a headless "Runtime" environment optimized for API-driven execution.[50]

1. Main Trade Processor Langflow (The Logic Core)

The Main Langflow is responsible for the direct execution of the user's request. It follows a sequential, deterministic workflow aimed at high accuracy and database integrity.[7]

- **Phase 1: JSON Data Ingestion:** A Webhook component receives the POST request from

the frontend.[38] A Parser component extracts the HTS code, origin, destination, and volume, converting them into structured text variables.[38]

- **Phase 2: Database and RAG Integration:** The system uses a RAG Agent to query a vector database (e.g., Astra DB) containing the 2026 tariff compendium and a relational database (PostgreSQL) for the trade_products_tariffs.csv dataset.[12]
- **Phase 3: Formulaic Computation:** A Calculator Tool is triggered to compute the baseline landed cost using the LaTeX formula provided earlier.[37] This component applies the "Cumulative Duty Logic" for Section 301 and the "Green Deduction Logic" for CBAM.[12]
- **Phase 4: Structured Output Generation:** The agent synthesizes the results into a standardized JSON response containing the "Total Landed Cost," a line-by-line "Duty Breakdown," and the "Regulatory Justification" (e.g., "Applied Japan-EU EPA 0.0% rate").[52]

2. AI Assistant Advisor Langflow (The Optimization Engine)

While the Main Processor answers "What is the cost?", the AI Assistant works in the background to answer "How can we make it cheaper?". This Langflow is non-linear and agentic.[52]

- **Phase 1: Pathfinding Algorithm Integration:** The assistant employs parallel applications of Dijkstra and A* algorithms to analyze the global shipping graph.[58] It identifies faster or cheaper routes by considering "Equivalent Tariff Delay" costs and port accessibility in Feb 2026.[60]
- **Phase 2: Sourcing Footprint Analysis:** The agent compares the current request against alternative sourcing hubs like Vietnam, Indonesia, or the UAE's Free Trade Zones (FTZs), which offer 100% foreign ownership and corporate tax exemptions.[46]
- **Phase 3: Savings Math and Threshold Trigger:** A specialized Math Agent calculates the delta between the user's current path and the optimal path.[36] If the savings exceed a pre-defined threshold (e.g., 5% of total cost), it triggers a webhook to the frontend notification component.[36]
- **Phase 4: Conversational Engagement:** Once the user opens the chat, the assistant uses a Chat History component to maintain context, explaining the logic of the suggested alternative (e.g., "Shipping via the India-Middle East-Europe Economic Corridor could reduce your CBAM liability by 15%").[12]

Mathematical Modeling of Path Selection

Logistics in early 2026 is no longer just a distance problem but a "Weighted Impedance Surface" problem.[65] The architecture calculates the "Shortest Path" where weight W is a function of both physical distance d and regulatory friction F.

$$W_{lane} = \sum(f_{freight} \cdot d) + \sum(TLC_{regulatory} + T_{delay} \cdot V_{time})$$

Where:

- T_{delay} is the border dwell time (higher for sanctioned countries).
- V_{time} is the per-day value of time savings for the specific product category.[60]

The system uses Yen's algorithm to find the "top k" shortest paths, providing the user with multiple options: the absolute cheapest, the fastest (often utilizing expensive air freight to bypass land border tensions), and the "most green" (minimizing CBAM liability).[58]