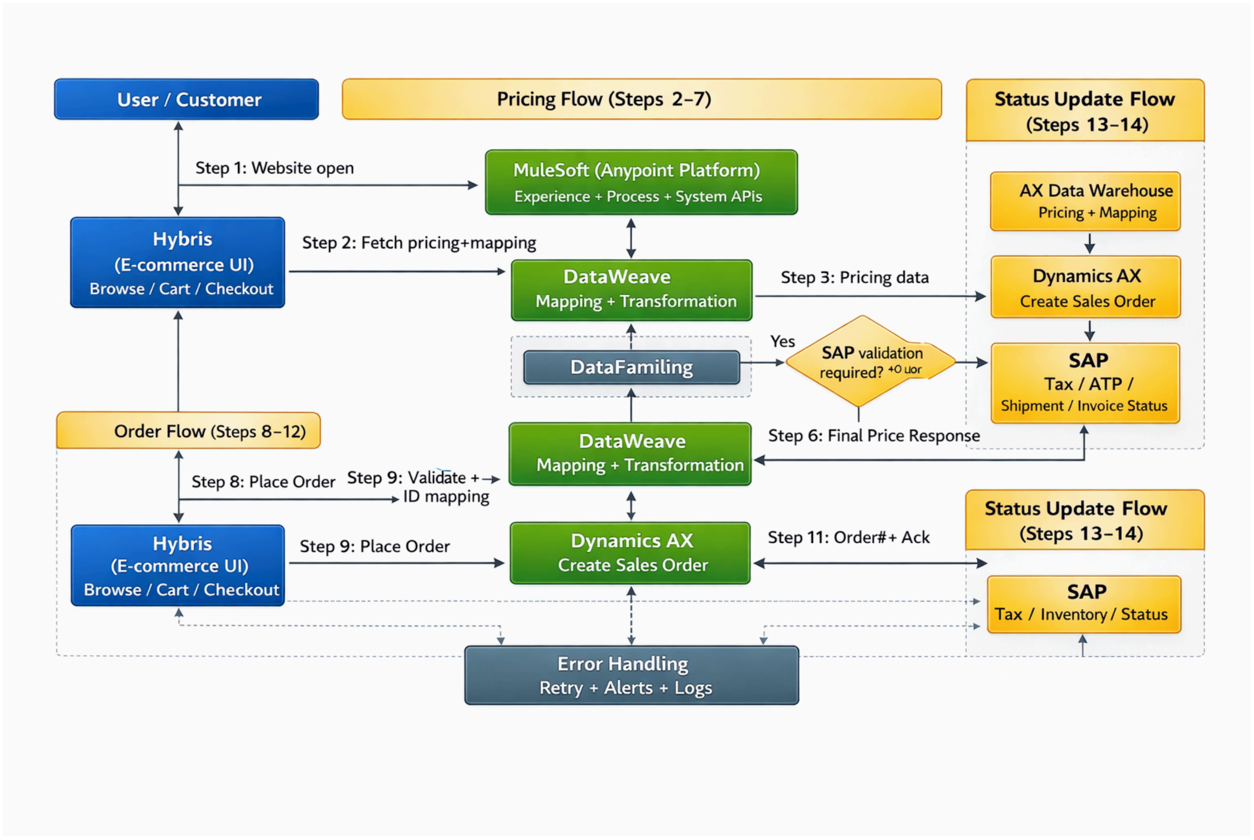
Mulesoft Project Flowchart



User / Customer
Pricing Flow (Steps 2–7)
Status Update Flow (Steps 13–14)
Step 1: Website open
MuleSoft (Anypoint Platform)
Experience + Process + System APIs
AX Data Warehouse
Pricing + Mapping
Hybris (E-commerce UI) Browse / Cart / Checkout
Step 2: Fetch pricing+mapping
DataWeave Mapping + Transformation
Step 3: Pricing data
Dynamics AX Create Sales Order
DataFamiling
Yes
SAP validation required?
SAP Tax / ATP / Shipment / Invoice Status
Order Flow (Steps 8–12)
DataWeave Mapping + Transformation
Step 6: Final Price Response
Step 8: Place Order
Step 9: Validate + ID mapping
Hybris (E-commerce UI) Browse / Cart / Checkout
Step 9: Place Order
Dynamics AX Create Sales Order
Step 11: Order#+ Ack
Status Update Flow (Steps 13–14)
SAP Tax / Inventory / Status
Error Handling
Retry + Alerts + Logs

This project is an **e-commerce integration** where **SAP Hybris** is the frontend application, **MuleSoft Anypoint Platform** is the integration and orchestration layer, and **Dynamics AX, AX Data Warehouse, and SAP** are backend enterprise systems.

Main goal of this project is to provide **real-time pricing**, **order creation**, and **order status updates** to the customer, without exposing backend ERP complexity to Hybris.”

Minute 1–2: User & Hybris (Frontend)

“First, user opens the **Hybris e-commerce website**.
User browses products, adds items to cart, and proceeds to checkout.

Hybris is only responsible for:

- Product catalog
- Cart
- Checkout
- Showing price and order status

Hybris **does not calculate pricing**, **does not create orders in ERP**, and **does not apply tax rules**.
For all backend logic, Hybris always calls **MuleSoft APIs**."

⏱ **Minute 2–3: Pricing Flow (Steps 2–7)**

(👉 Point to Pricing Flow)

"When the user clicks **Add to Cart** or **View Price**, Hybris sends a **real-time pricing request** to MuleSoft through an Experience API.

Now MuleSoft orchestration starts.

First, MuleSoft fetches **pricing and mapping data from AX Data Warehouse**.
AX Data Warehouse contains:

- Base price
- Customer-specific pricing
- Product and customer mapping

We use AX Data Warehouse instead of direct AX tables to improve performance and reduce load on the ERP system.

Next, MuleSoft uses **DataWeave** to:

- Map fields
- Apply pricing rules
- Apply discounts
- Convert data into Hybris-required JSON format

Then MuleSoft checks whether **SAP validation is required**.

If tax or inventory availability check is needed, MuleSoft calls **SAP** and receives tax or ATP response.
This SAP response is merged with AX pricing data.

Finally, MuleSoft builds the **final calculated price** and sends it back to Hybris, where the customer sees the correct price on the UI."

⏱ Minute 3–4: Order Flow (Steps 8–12)

(👉 Point to Order Flow)

"After reviewing price, the user proceeds to checkout and clicks **Place Order**.

Hybris sends the **Create Order request** to MuleSoft.

MuleSoft performs:

- Request validation
- Duplicate check
- ID mapping from Hybris IDs to AX IDs

Using **DataWeave**, MuleSoft transforms the order payload into **Dynamics AX-compatible format**.

Then MuleSoft sends the request to **Dynamics AX**, where the actual **sales order is created**.

Dynamics AX returns:

- Order number
- Order acknowledgment

MuleSoft sends this order confirmation back to Hybris, and the user sees the **order number and success message** on the website."

⏱ Minute 4–4:30: Order Status Updates (Steps 13–14)

(👉 Point to Status Flow)

“After order creation, backend processing continues.

SAP handles shipment, invoice, and delivery processing.
Whenever order status changes, SAP sends status updates to MuleSoft.

MuleSoft normalizes these updates using DataWeave and updates **Hybris order tracking page**.

This allows the customer to track order status in real time.”

⏱ Minute 4:30–5: Error Handling & Conclusion

(👉 Point to Error Handling box)

“Across all integrations, MuleSoft implements centralized **error handling**, including:

- Retry mechanism
- Alerts
- Logging and monitoring

This ensures system reliability even if AX or SAP is temporarily unavailable.

To summarize:

- **Hybris** handles UI and customer interaction
- **MuleSoft Anypoint Platform** handles integration, orchestration, and transformations
- **DataWeave** handles all mappings and calculations
- **AX Data Warehouse** provides pricing reference
- **Dynamics AX** creates orders
- **SAP** handles tax, inventory, and shipment status

This completes the end-to-end MuleSoft integration flow.”

🔑 FINAL ONE-LINE CLOSING (Very Important)

“This project demonstrates API-led connectivity using MuleSoft to decouple Hybris from backend ERP systems while delivering real-time pricing, order creation, and order tracking.”
