Smart Cart Integration Architecture

Recommended CATAPULT/ WebCart integration model for smart cart vendors

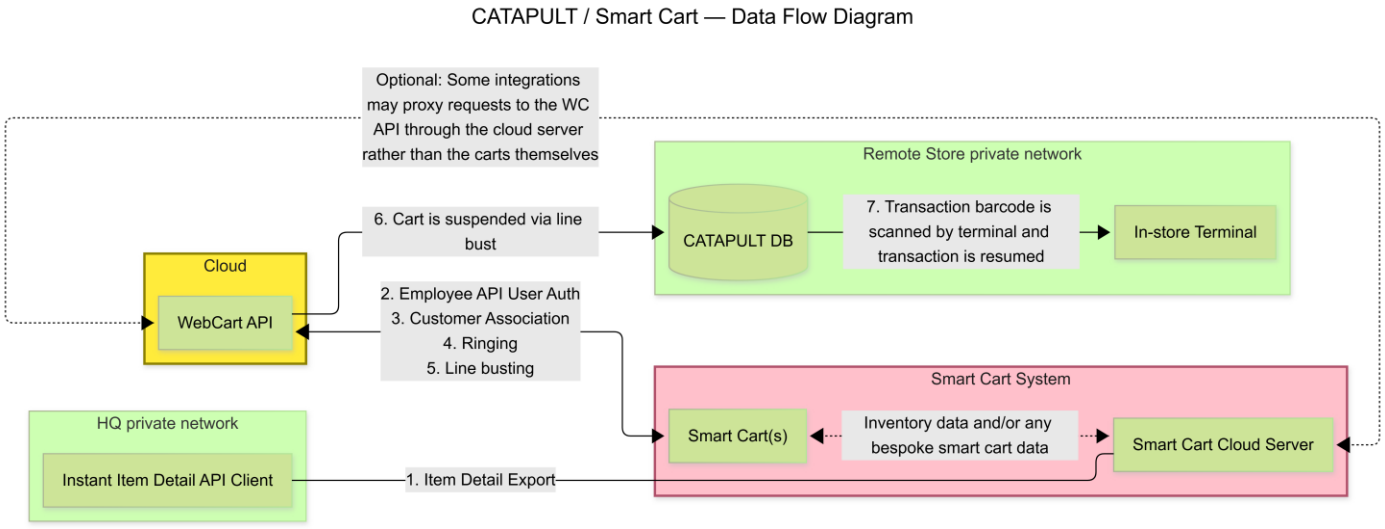
Purpose

Integrating smart carts to the CATAPULT® system requires coordination across multiple CATAPULT components including the WebCart API, point of sale (POS) or self-checkout (SCO) terminal, Instant Item Detail API Client, the smart cart, and the smart cart server. This document describes the recommended integration model for smart cart vendors that need to build an in-store basket, display running totals to shoppers, and transfer the transaction to a POS or SCO lane for payment and completion.

Integration Entities

Entity	Responsibility
WebCart API	Provides publicly available HTTPS endpoints for creating and updating a cart, authenticating as an Employee API User, optionally associating a customer, retrieving cart state, calculating real-time pricing/promotions, and suspending the cart for in-store checkout.
POS / SCO terminal	Scans or receives the transaction barcode/scan code, resumes the suspended transaction, accepts payment, and finalizes the transaction.
Instant Item Detail API Client	Sends inventory and store data from CATAPULT to the smart cart provider. A per-store license is required.
Smart Cart	In-store cart capable of communicating with WebCart API and/or the vendor smart cart platform, displaying scanned items and running totals, and presenting a checkout barcode for in-store lane handoff. CATAPULT does not currently support payment directly on the smart cart.
Smart Cart Server	Vendor-managed service responsible for receiving and storing item detail data from the Instant Item Detail API Client, including item identifiers required for WebCart cart updates. This service should maintain the vendor item/catalog cache and support recovery or re-export processes if item updates are missed.

High-Level Architecture Diagram

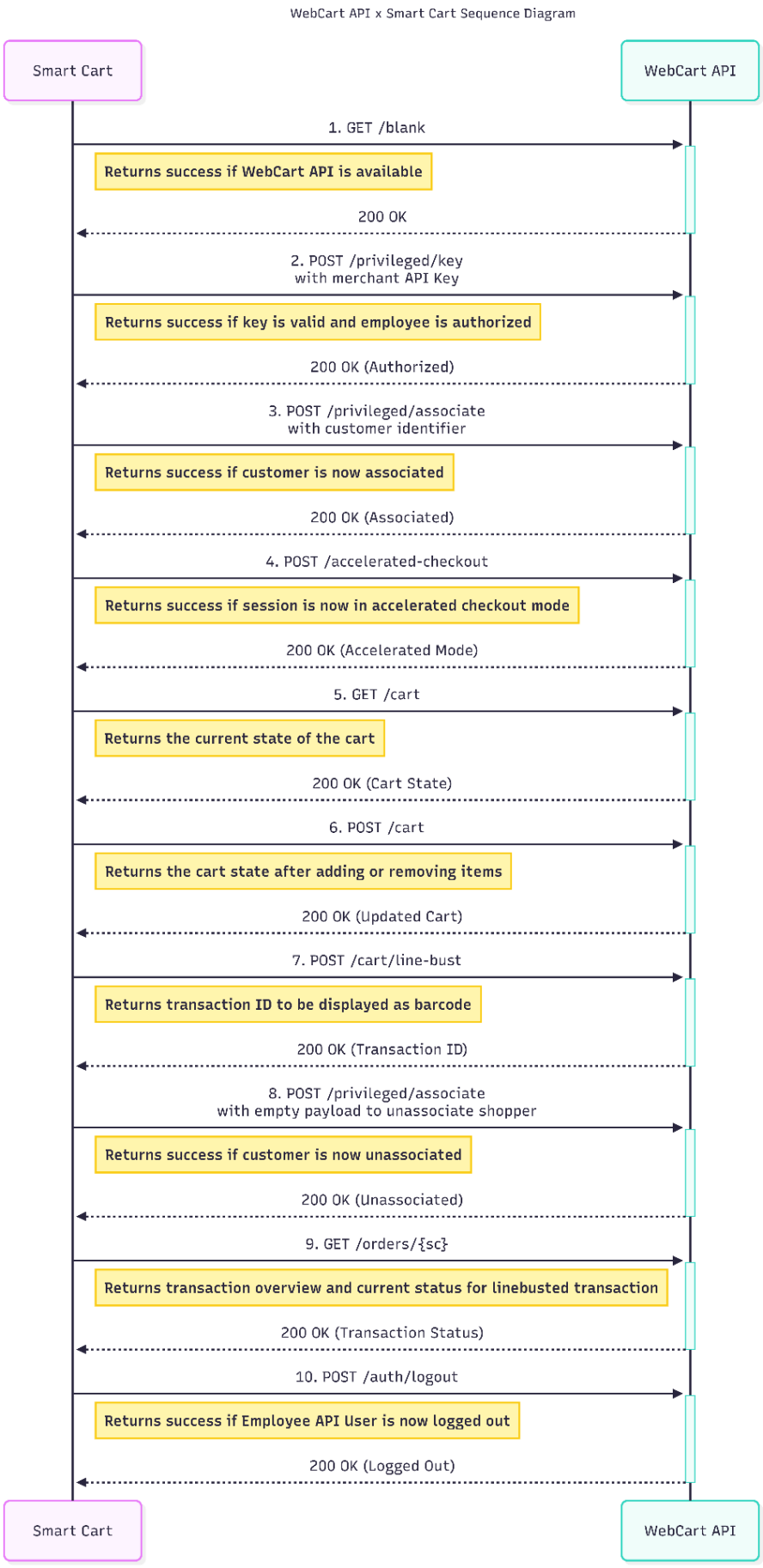


Interaction Details

- The Instant Item Detail API Client can be configured to send detailed item data, item images, and store data for each store in an organization. The client supports sending data to multiple sources. Data from this client should be stored within the smart cart system. Note: This is only required if an integrator wants access to *all* item data at once. Should item data be needed only at the time of item scanning, the WebCart API “Item detail by Scan Code” or “Item Summary by Scan Code” endpoints can be used.
- Smart cart systems should use the item identifiers supplied through the Instant Item Detail API Client when adding items to WebCart. Depending on the item and environment, these identifiers may include INV_PK/INV_CPK or supported Big PK values. The WebCart API supports both composite PK/CPK-style identifiers and Big PK formats.
- If the Instant Item Detail API Client cannot reach the smart cart provider endpoint, there is no automatic retry. Missed updates may remain missing until a new item change occurs, or a manual re-export is requested. Retailers and vendors should define a recovery process for missed item updates and an in-store process for handling unknown items at the cart.
- Before performing any action through the WebCart API, the smart cart system must authenticate as an Employee API User. The merchant must create dedicated employee account(s) and API key(s) for the smart cart provider and share the credentials through a secure channel.
    - The Employee API User should follow the principle of least privilege and be granted only the permissions required for the smart cart integration. At minimum, the Employee API User must have the “WebCart Access” and “GraphQL API Access” Authorization Security checkpoints to use the WebCart API. Customer association functionality requires an additional “Associate Customer” Authorization Security checkpoint. Retailers should grant only the checkpoints required to support the approved smart cart use case.
- Once authenticated as an Employee API User, the smart cart system may optionally associate a shopper to an existing customer account by account number or unique phone number. If a shopper does not have an account, this step can be skipped. If a customer is associated, they should be unassociated after line busting to avoid incorrectly linking future transactions. Note: If the smart cart system does not wish to associate customers, then the “Employee API User Auth” and “Customer Association” steps of this recommended integration flow can be bypassed.

- The WebCart API should be used to build the basket by retrieving the current cart and updating it by adding, removing, or changing items. Real-time pricing, promotions, discounts, and running totals should be displayed using the values returned in the WebCart cart response.
    - Smart carts should display any restriction-related information returned by the integration where applicable. However, final enforcement for restricted items should be validated when the suspended transaction is resumed and finalized at the POS/SCO terminal.
- When the shopper is finished shopping, the cart should be line-busted through the WebCart API. This action suspends the transaction and returns the transaction scan code that can be displayed as a barcode on the smart cart.
    - When a transaction is line-busted, it is suspended and the Employee API User is automatically assigned a new transaction.
    - The barcode can be scanned or entered at the POS/SCO lane to resume the suspended transaction. Payment and final transaction processing occur at the lane. The standard smart cart flow does not capture payment directly on the cart.
    - The status of the order can be found by calling GET /orders/{sc}.

Primary WebCart Checkout Sequence Diagram



Primary WebCart Flow Breakdown

- Calling the session endpoint without a session cookie returns a new token cookie representing a guest session. This allows a smart cart client to manage multiple guest sessions independently. Note: The smart cart is expected to manage the session cookies in the event that multiple sessions are supported.
- The smart cart provider authenticates with a merchant-provided API key linked to an Employee API User with the required Authorization Security checkpoints.
    - The smart cart may optionally associate a shopper with an existing customer account using the customer account number or unique customer phone number. If this step is skipped, the transaction continues as a guest session.
- Entering Accelerated Checkout mode allows the WebCart API to treat weights as entered when updating the cart. If this step is skipped, item weights may be rounded to the nearest configured increment. In addition, this mode prevents items from being blocked due to the store's configured out-of-stock behavior.
- GET /cart retrieves the current cart state and should be called when starting a transaction to collect relevant cart context for POST /cart.
- POST /cart updates the cart by adding, removing, or changing items and returns the updated cart response, including pricing and promotion information.
- POST /cart/line-bust suspends the transaction and returns the transaction number/scan code. The smart cart displays this scan code as a barcode for POS/SCO handoff.
    - If a customer is associated with the transaction, the customer association should be removed after the transaction is line busted to prevent subsequent transactions from being incorrectly linked to the previous shopper.
    - GET /orders/{sc} may be used to retrieve the status of a transaction by scan code. If the transaction was associated with a customer before it was line-busted, that same customer must be associated with the current cart to call this endpoint.
- The logout endpoint may be called after each shopping trip, after an inactivity timeout, or when required by a vendor-specific security workflow.

Employee API Users do not receive customer PII through the GET /cart, POST /cart, or GET /orders/{sc} endpoints. Additionally, the disp and dispShort claims will not be returned in the token JWT claims.

API Inventory

API Source	API / Endpoint	Purpose	Required?	Notes / Dependencies
WebCart API	Guest/Session token behavior, such as /blank	Creates a new guest session token when called without a session cookie.	Optional	Used to start or isolate smart cart shopper sessions.
WebCart API	POST /privileged/key	Authenticates the smart cart system as an Employee API User using a merchant-provided API key.	Required	Requires secure API key storage and required WebCart authorization checkpoints.
WebCart API	POST /privileged/associate	Optionally associates a customer with a transaction using account number or unique phone number.	Optional	Requires WebCart Associate Customer checkpoint. If associated, remove customer association after line busting.
WebCart API	POST /accelerated-	Enters Accelerated Checkout mode	Optional	Recommended for smart cart

	checkout	so weighted quantities are treated as entered and to bypass logic in/out of stock cart preventions.		flows that add weighed items. This also bypasses the in-stock add to cart restrictions that may be configured in the store's WebCart Profile.
WebCart API	GET /cart	Returns the current cart state.	Required	Use when starting or retrieving the state of a transaction.
WebCart API	POST /cart	Adds, removes, or updates items and returns updated cart totals.	Required	Primary source for real-time pricing, discounts, and promotions.
WebCart API	POST /cart/line-bust	Suspends the transaction and returns the transaction scan code/barcode value.	Required for POS/SCO checkout	Enables in-store lane checkout handoff.
WebCart API	GET /orders/{sc}	Returns order data and transaction status by scan code.	Optional	Customer association may be required when the transaction is linked to a customer.
WebCart API	Logout endpoint	Logs out the Employee API User session.	Recommended	Useful for inactivity timeout, or security-sensitive workflows.
Instant Item Detail API Client	Item detail export/feed	Sends item detail, images, store data, and item identifiers to the smart cart system. Note: This is only required if an integrator wants access to all item data at once. Should item data be needed only at the time of item scanning, the WebCart API "Item detail by Scan Code" or "Item Summary by Scan Code" endpoints can be used.	Optional	Requires per-store license. No automatic retry if delivery fails.
POS/SCO	Scan or enter transaction barcode	Resumes the suspended transaction.	Required	Payment and finalization occur at POS/SCO.
Passport API	N/A	Not required for the presented smart cart flow.	Not required	Future Passport dependencies should be reviewed separately.

Source of Truth by Function

Function	Source of Truth	Notes
Item/catalog detail	Instant Item Detail API Client / Catapult item data	The smart cart vendor should store received item detail, item images, store data, and item identifiers.
Cart state	WebCart API	Use WebCart cart responses as the current transaction state.
Pricing	WebCart API cart response	Real-time pricing is returned as the cart is built.
Promotions and discounts	WebCart API cart response	Promotions and discounts are included in cart response pricing.
Customer/loyalty association	WebCart API customer association	Optional; only applies when shopper association is performed.
Checkout handoff	WebCart API line busting	Suspends transaction and returns transaction scan code.
Payment/finalization	POS/SCO	Standard flow completes payment and finalizes the transaction at the lane.

Testing Environment Guidance

Smart cart integrations should be validated in the environment appropriate for the current phase of work. It is recommended to validate the integration in a retailer-specific lab environment before deploying to production. Item data, pricing, promotions, loyalty, checkout workflows, POS behavior, licensing, and store configuration are retailer/store specific, making end-to-end testing in the target environment essential.

Environment	Primary Purpose	Guidance
Retailer lab	Retailer-specific integration validation	Recommended for development and testing. Must be configured separately from production. Merchant should open a support ticket to request a lab environment and confirm pricing/licensing details.
Production	Live store deployment	Requires production licenses. API keys and endpoints should be production-specific and securely managed.

Security and PCI Considerations

- Merchant-provided API keys must be stored securely by the smart cart provider and shared only through secure channels. API keys should not be embedded directly on smart cart hardware if the hardware cannot protect secrets appropriately.
- CATAPULT Employee Records should be dedicated to the smart cart integration where possible, and their Authorization Security checkpoints should be reviewed periodically.

- The smart cart flow explained in this document does not capture payment on the cart. Payment card handling currently remains in the POS/SCO checkout flow after the suspended transaction is resumed.
- Any future integration that captures payment directly on the smart cart would require separate review and approval from Product, Engineering, Security, and PCI stakeholders before it could be supported.

License Notes

Both the WebCart API and the Instant Item Detail API Client require per-store licenses.

Supporting Documentation

- WebCart API: ECRS® APIs - ECRS
- Instant Item Detail API Client: ECRS® APIs - ECRS