Table of Contents

:= Supply Chain Collaboration SaaS: A Business-First Guide

This guide explains, in plain language, how the Quantum framework helps you build a Supply Chain Collaboration Software-as-a-Service (SaaS). We focus on real business problems—secure data sharing, multi-party workflows, and tenant isolation—and show how the framework's building blocks solve them without requiring you to stitch together dozens of bespoke APIs.

Chapter 1. What a supply-chain SaaS needs (and how Quantum helps)

Common needs when launching a collaboration platform:

- Secure sharing across companies: Buyers, suppliers, carriers, and 3PLs must see the same truth, but only what they're allowed to see.
- Role-appropriate views: Planners, operations, and analysts look at the same orders/shipments but need different fields, filters, and actions.
- Auditability and control: You need a clear explanation of who saw or changed what, and why that was allowed.
- Fast onboarding: Each partner authenticates differently; you can't force everyone into one identity provider.
- API consistency: Your UI and integrations shouldn't learn a different API for every screen or entity.
- Data lifecycle and stewardship: Easy data imports/exports, clear status tracking, and repeatable completion steps.

How Quantum maps to these needs:

- Multi-tenancy by design: Each organization (tenant) is isolated by default; sharing is added deliberately and safely.
- Policy-driven permissions: Human-readable rules answer "who can do what" and add scoping filters so only the right data shows up.
- Consistent REST and Query: One List API and a simple query language cover most searches and reports—no explosion of bespoke endpoints.
- Flexible identity: Support for different authentication methods per organization, plus delegated admin so each tenant manages its own users.
- State + tasks: Built-in patterns for long-running, stateful processes and "completion tasks" to move work forward predictably.

Chapter 2. Why multi-tenancy is a natural fit for supply chains

Supply chains are networks. Everyone shares a process, but not a database. Quantum isolates each tenant's data by default and lets you selectively share records with partners:

- Private by default: A supplier's purchase orders are not visible to other suppliers.
- Share on purpose: Create a "collaboration bubble" around a purchase order or shipment so a buyer and a specific carrier can see the same milestones and documents.
- Central where it helps: Keep a global partner directory or product catalog in a shared domain if that reduces duplication.
- Regional and regulatory needs: Pin certain data (e.g., EU shipments) to the correct region with simple policies.

Chapter 3. Who uses the system (organizations and roles)

Organizations (tenants) - Shippers and customers: create orders, monitor shipments, approve changes. - Carriers and 3PLs: accept tenders, provide status, confirm delivery. - Suppliers: acknowledge POs, provide ASN/invoice data.

Common user types within each organization - Planners: need forward-looking visibility—capacity, forecasts, purchase orders. - Operations: need day-to-day details—stops, ETAs, exceptions. - Business analysts: need trends and history—on-time performance, cost, root causes.

Data visibility examples - Private notes: an operations note on a shipment visible only inside the shipper’s tenant. - Shared context: a delivery appointment visible to both the shipper and the carrier for a specific shipment. - Role-filtered: analysts see aggregated KPIs, planners see open exceptions, operators see actionable tasks.

Chapter 4. Identity and access: meet partners where they are

Every organization may authenticate differently: - Enterprise SSO (OIDC/SAML) for shippers and large suppliers. - Username/password for smaller partners. - Service accounts and tokens for system-to-system integrations.

Quantum supports these patterns and lets each tenant manage its own users (delegated administration). Permissions can differ by user type and tenant while staying auditable and predictable.

Chapter 5. Modeling without jargon: Areas, Domains, and Actions

To keep your platform organized, Quantum groups things into: - Functional Areas: broad business spaces like Collaboration, Finance, or Catalog. - Functional Domains: specific entity types within an area—Partner, Shipment, PurchaseOrder, Invoice. - Functional Actions: what users do—VIEW, CREATE, UPDATE, DELETE, APPROVE, EXPORT, etc.

Why this matters: - Clear menus for the UI (group screens by area and domain). - Clear policy rules (easier to say “Carriers can VIEW Shipments” or “Only Finance can APPROVE Invoices”).

Chapter 6. Policies that say “who can do what” (Rule Language)

Policies are simple, readable rules that match: - Who is calling (identity and roles) - What they’re trying to access (area, domain, action) - Request details (headers/body, like a shipment id or tenant)

Rules then allow or deny the action and can add filters so the user only sees data they’re permitted to see. See the Permissions guide for authoring details ([permissions]).

Write-time data placement (where a new record belongs) - By default, new records use the creator’s organization. - You can override per area/domain with a small policy—for example, keep Partner records in a shared “directory” domain while Shipments stay tenant-local. See “Data domain assignment on create” ([_data_domain_assignment]).

Chapter 7. A small, powerful API surface: List + Query

Instead of building a unique search endpoint for every screen, Quantum gives you: - List API: a single, consistent endpoint per domain to list, filter, sort, page, and project fields. - Query Language: a simple filter syntax so UIs and reports can ask precise questions. - Automatic enforcement: policies and data-domain rules are always applied server-side, so callers only receive allowed data.

Business outcomes - Faster delivery: new screens reuse the same list API. - Fewer mistakes: less custom code, more consistent results. - Safer by default: even power users can’t bypass policy enforcement.

Chapter 8. Delegated Administration (tenant-level user management)

Empower each organization to run their own house while you keep platform-wide safety: - Tenant admins invite users, reset passwords, and assign roles. - Role templates per tenant align with their org structure (Planner, Ops, Analyst, Carrier Dispatcher, etc.). - Cross-tenant boundaries are respected; global administrators can still support, audit, and troubleshoot with impersonation/acting-on-behalf-of where permitted and logged.

Chapter 9. Integrations and data management

Supply chains depend on clean, timely data. Quantum provides: - CSV imports/exports: onboard master data quickly, rerun safely, and let business users fix and re-upload. - Stateful objects: model processes (e.g., Shipment lifecycle) with clear states—Created → In-Transit → Delivered → Closed. - Completion Tasks: checklist-like steps (confirm pickup, upload POD, reconcile invoice) that drive work to done and provide accountability. - Consistent access: the same policies that protect your UI protect imports/exports and API calls.

Example uses - Bulk load a new supplier catalog with CSV import; analysts export exceptions weekly for review. - Track shipment exceptions as tasks; operations completes them with evidence (attachments/notes), all audited.

Chapter 10. End-to-end examples

1) Buyer–supplier collaboration on a Purchase Order - Create a collaboration bubble around a PO so both parties see schedule, holds, and documents. - Supplier can UPDATE promised dates; buyer can APPROVE changes. Private buyer notes remain private.

2) Shared partner directory, curated centrally - Keep one shared Partner domain so everyone finds the same carrier and facility records. - Only directory curators can CREATE/UPDATE; all tenants can VIEW.

3) EU shipment residency - Shipments created by anyone in Europe are written to an EU partition by policy. Reads remain role- and tenant-scoped.

Chapter 11. What you don’t have to build from scratch

- Data isolation and safe sharing across tenants
- A consistent CRUD and search API for every domain
- A policy engine that explains its decisions and applies filters
- A write-time placement policy (so data lands in the right partition)
- Patterns for long-running, stateful business processes and task completion

The framework gives you these foundations so your teams focus on business value—on-time deliveries, lower cost, happier customers.

Chapter 12. Next steps

- Start with siloed defaults; prove value quickly using the List API.
- Add small, targeted policies to enable collaboration bubbles and shared directories.
- Introduce delegated administration so partners self-serve.
- Use CSV imports and Completion Tasks to operationalize data stewardship.
- Deep dive: Permissions and Rule Language ([permissions]), and Data domain assignment on create ([_data_domain_assignment]).

Chapter 13. A day in the life: From Purchase Order to Delivery

This story ties the pieces together in a realistic sequence. We follow a Purchase Order (PO) from creation to delivery, with shared visibility for suppliers and carriers, a clear state graph, and checklist-like Completion Tasks guiding the work.

1) Purchase Order is created (by the Buyer) - Action: A buyer creates a PO in the Collaboration area under the PurchaseOrder domain. - Data placement: By default, the PO is written to the buyer's organization (their data domain). If you prefer a shared domain for POs, configure a small policy; otherwise, the default works well. - Programmatic sharing: A rule shares the specific PO with the chosen Supplier (or Suppliers). The Supplier can view the PO and update the fields you allow (e.g., promised date), but cannot see private buyer-only fields.

State graph (illustrative) - Draft → Open → SupplierAcknowledged → ReadyToShip → PartiallyShipped → FullyShipped → Received → Closed

Completion Tasks (examples attached to the PO) - Buyer: Provide required documents (commercial terms, incoterms) - Supplier: Acknowledge PO (due in 24 hours) - Supplier: Provide ASN (advanced shipping notice) for each shipment - Supplier: Confirm pickup window - Buyer: Approve any date changes

2) The Supplier prepares shipments (shared onward to Carriers) - Action: The Supplier creates one or more Shipments linked to the PO (and optionally to specific lines). - Data placement: Shipments are written to the Supplier's domain by default (their own organization), but are shared with the Buyer so both parties see the same timeline. - Sharing to Carriers: When the Supplier tenders a shipment, the shipment is shared with the selected Carrier so they can update movement and milestones.

Shipment state graph (illustrative) - Planned → Tendered → Accepted → InTransit → Delivered → ProofVerified → Closed

Shipment Completion Tasks (examples) - Carrier: Confirm pickup - Carrier: Update in-transit location/ETA - Carrier: Upload POD (proof of delivery) - Supplier: Reconcile quantities shipped vs. ordered

3) Status updates complete tasks and move states forward - When the Supplier marks "SupplierAcknowledged," the PO's acknowledgement task completes and the PO moves to SupplierAcknowledged. - When all lines are ready and at least one shipment is created, the PO advances to ReadyToShip. If some but not all lines ship, it enters PartiallyShipped; once all lines ship, it becomes FullyShipped. - Carrier updates (e.g., Delivered with POD uploaded) complete shipment tasks. Those completion events can also advance the PO state (e.g., all shipments Delivered → PO moves to Received). Final checks (invoices matched, discrepancies resolved) move the PO to Closed.

Why this is safe and predictable - Roles and policies ensure each party sees only what they should: the Buyer sees everything; the Supplier sees the shared PO and its related shipments; the Carrier sees only the shipments they handle. - Completion Tasks remove ambiguity: everyone knows the

next step and who owns it. Each task completion is audited. - The state graph makes lifecycle transitions explicit. Policies can require certain tasks to be completed before a state transition is allowed.

4) Business visibility and reporting (List API + Query) - Operations view: “Purchase Orders in progress” shows all POs in Open, SupplierAcknowledged, ReadyToShip, or PartiallyShipped, including late tasks and upcoming milestones. - Buyer/supplier view: Both parties see the same PO status and related Shipments, with role-appropriate fields. - Simple reporting example (illustrative): GET /collaboration/purchaseorder/list?filter=status IN ("Open","SupplierAcknowledged","ReadyToShip","PartiallyShipped")&sort=dueDate:asc&limit=50 - Add projections to include key fields and roll-ups (e.g., shipped vs. ordered quantities). Related Shipment info can be retrieved similarly via the List API on the Shipment domain, filtered by the PO id.

What made this easy (and repeatable) - Multi-tenancy by default: Each org’s data is isolated; sharing is explicit and safe. - Policies (Rule Language): Define who can see or update which fields and when. The same rules apply to UI, API, and imports/exports. - Data domain assignment on create: Defaults keep data in the creator’s org; you can configure exceptions (e.g., shared directories) with a tiny policy. - Stateful objects + Completion Tasks: Clear states and checklists turn complex collaboration into a predictable flow. - List API + Query Language: One consistent way to fetch work lists, timelines, and reports without proliferating custom endpoints.