Contract Management Process Flows

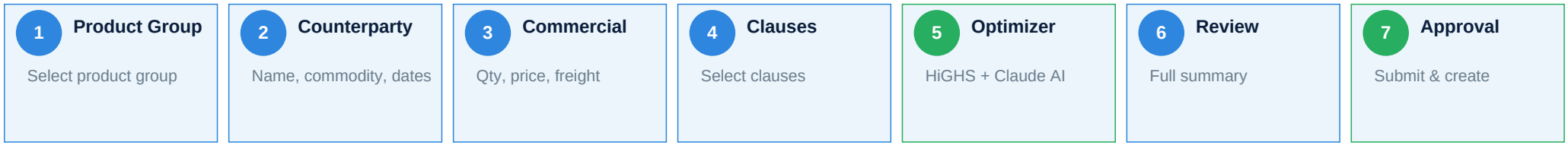
Trammo Trading Desk

7-Step Contract Wizard & Solver Integration

February 2026 | Confidential

1 Contract Management Overview

The Contract Management module is a guided 7-step wizard that takes traders from product group selection through counterparty identification, commercial negotiation, clause selection, LP solver validation, review, and final approval -- all within a single real-time LiveView interface.



KEY DIFFERENTIATOR: Unlike traditional contract management systems, Trammo integrates an LP solver directly into Step 5 -- giving traders quantitative validation of contract economics BEFORE approval.

Wizard State Machine

Step 1 -> Step 2: Validate product group selected
Step 2 -> Step 3: Validate counterparty name, commodity, delivery dates
Step 3 -> Step 4: Validate quantity and price are set
Step 4 -> Step 5: Validate at least 1 clause selected
Step 5 -> Step 6: Always valid (solver is optional but recommended)
Step 6 -> Step 7: Always valid (review is informational)
Step 7: Submit for Approval -> DB Transaction

Back navigation is always available from any step > 1.

2 Steps 1-3: Product Group, Counterparty, Commercial Terms

Step 1: Product Group Selection

Four product groups available:

* NH3 Domestic Barge (AD) -- Anhydrous Ammonia, Aqua Ammonia
* Sulphur International (SI) -- Granular, Liquid, Formed
* Petcoke (PC) -- Fuel-Grade, Anode-Grade, Calcined
* NH3 International (AI) -- Anhydrous Ammonia, Ammonia Solution

Validation: Must select one product group to proceed.

Step 2: Counterparty & Commodity

Field	Type	Validation	Example
Counterparty Name	Text input	Required, non-empty	ACME Trading Corp
Commodity	Select (filtered)	Required, matches product group	Anhydrous Ammonia
Delivery Window Start	Date picker	Required	2026-03-15
Delivery Window End	Date picker	Required	2026-04-30

Step 3: Commercial Terms

Field	Type	Validation	Solver Mapping
Quantity + Unit	Number (MT/KT/BBL)	Required	--
Proposed Price (USD)	Number (2 dec)	Required	nh3_price override
Proposed Freight	Number (2 dec)	Optional	barge_freight override
Payment Terms	Select	Default: Net 30	--
Incoterm	Select	Default: FOB	--

SOLVER INTEGRATION: Proposed price maps to nh3_price and proposed freight maps to barge_freight in the solver variable space. These overrides are applied on top of the product group defaults when the optimizer runs in Step 5.

3 Steps 4-5: Clause Selection & Optimizer Validation

Step 4: Clause Selection

Clause	Default	Description
Force Majeure	Pre-selected	Acts of God, war, government actions
Demurrage & Dispatch	--	Vessel/barge demurrage rates and dispatch rebate terms
Quality Specification	--	Product quality requirements, testing methods, rejection criteria
Quantity Tolerance	--	+/- 5% at seller's option
Price Escalation	--	Index-linked price adjustment mechanism
Payment Terms	Pre-selected	Net 30 days from bill of lading date
Insurance & Liability	--	Marine cargo insurance requirements and liability limits
Dispute Resolution	--	Arbitration -- LCIA London or ICC Paris
Termination Rights	--	Early termination triggers and notice requirements
Confidentiality	--	Non-disclosure of contract terms and pricing

Step 5: Optimizer Validation (Key Differentiator)

When the trader clicks "Run Optimizer Validation":

1. Load product group default variables (20 vars from live state)
2. Apply contract overrides: proposed_price -> nh3_price, proposed_freight -> barge_freight
3. Execute Solver.solve(product_group, vars) via Zig/HiGHS linear program
4. Display results: Status | Profit | Tons | ROI (4-card grid)
5. Show route allocations (Route 1, Route 2, ...)
6. Run PostsolveExplainer.explain_all() -- Claude AI generates 3-5 sentence explanation

Outcomes: OPTIMAL (viable), INFEASIBLE (needs adjustment), UNAVAILABLE (manual OK)
The solver step is recommended but NOT required to proceed.

4 Steps 6-7: Review & Approval

Step 6: Review Summary

Consolidated view of all contract data: product group, counterparty, commodity, delivery dates, quantity, price, freight, payment terms, incoterm, selected clauses (as tags), and optimizer results (if run). No edits here -- navigate back to any step. Always valid to proceed.

Step 7: Approval Submission

Single Repo.transaction creates 5 records:

1. CmContractNegotiation -- Full step_data, solver snapshot, trader ID
2. CmContractNegotiationEvent -- type: submitted_for_approval, actor: trader email
3. CmContract -- ref: CTR-{PREFIX}-{YYMMDDHHMM}-{hex}, status: pending_approval
4. CmContractVersion (v1) -- Full terms snapshot
5. CmContractApproval -- approver: trading_manager, status: pending

Emits event: contract_submitted_for_approval -> EventEmitter -> VectorizationWorker -> pgvector

Data Model

Table	Key Columns
cm_contract_negotiations	product_group, reference_number, counterparty, commodity, status, step_data, solver_snapshot, trader_id
cm_contract_negotiation_events	event_type, step_number, actor, summary, details (jsonb)
cm_contracts	contract_reference, counterparty, commodity, status, terms, selected_clause_ids, approved_by
cm_contract_versions	version_number, terms_snapshot (jsonb), change_summary, created_by
cm_contract_approvals	approver_id, approval_status, notes, decided_at

5 Event Pipeline & Vectorization

Every step transition emits events through the EventEmitter -> Oban -> pgvector pipeline, enabling semantic search over contract history.

Trigger	Event Type	Source	Key Payload
Step 1>2	contract_product_group_selected	contract_mgmt	product_group, user
Step 2>3	contract_counterparty_set	contract_mgmt	counterparty, commodity
Step 3>4	contract_commercial_terms_set	contract_mgmt	quantity, price, freight
Step 4>5	contract_clauses_selected	contract_mgmt	selected clause count
Step 5>6	contract_optimizer_complete	contract_mgmt	solver status, profit, ROI
Step 6>7	contract_review_complete	contract_mgmt	review confirmation
Optimizer	contract_optimizer_validated	contract_mgmt	full solver results
Submit	contract_submitted_for_approval	contract_mgmt	negotiation_id, contract_id, all terms

Pipeline: Contract Wizard -> EventEmitter.emit_event/3 -> event_log table -> Oban VectorizationWorker -> Embeddings.generate() -> pgvector vector_embeddings -> Semantic Search via VectorQuery

6 Competitive Positioning vs. sea.live

sea.live provides contract management for commodity trading. Trammo adds quantitative solver validation and AI-powered decision support that transforms contracting from a legal/administrative function into a decision support tool.

Capability	sea.live	Trammo Trading Desk	Advantage
Contract Workflow	Multi-step wizard	7-step guided wizard	Comparable
LP Solver Integration	Not available	HiGHS validates during contracting	Trammo
AI-Powered Analysis	Not available	Claude explains solver output	Trammo
Real-Time Data	Market data feeds	20 live vars from 10+ APIs	Trammo
Pre-Contract Optimize	Optimize after signing	Optimize BEFORE approval	Trammo
Event Vectorization	Not available	pgvector semantic search	Trammo
Counterparty Mgmt	Full KYC directory	Inline entry (extensible)	sea.live
Solver Constraint	Contracts standalone	Approved = live constraint	Trammo
Approval Workflow	Multi-level approvals	Role-based with audit trail	Comparable

KEY INSIGHT: sea.live treats contract management as a legal/admin workflow -- contracts are created, reviewed, and approved without quantitative validation. Trammo inserts an LP solver BEFORE approval, giving traders projected profit, ROI, and route allocations computed against 20 live variables. Combined with Claude AI, this transforms contract approval from a compliance checkpoint into a DECISION SUPPORT MOMENT.

WHERE sea.live LEADS: Deeper counterparty management (KYC, credit), more mature document management, broader clause library with template inheritance. These can be added to Trammo over time -- but sea.live cannot easily retrofit real-time solver integration into their architecture.