The American Journal of Management and Economics Innovations
(ISSN – 2693-0811)
VOLUME 05 ISSUE 11 Pages: 58-73
SJIF IMPACT FACTOR (2020: 5. 307) (2021: 5. 562) (2022: 6. 361)
OCLC – 1176275019

Publisher: The USA Journals

Journal Website: https://theamericanjournals.com/index.php/tajmei

Research Article

# THE ONE-WINDOW ARCHITECTURE - A SYSTEM-LEVEL MODEL FOR INTEGRATED LIFECYCLE MANAGEMENT IN EXPORT-ORIENTED B2B MANUFACTURING

**Submission Date:** November 15, 2023, **Accepted Date:** November 21, 2023,
**Published Date:** November 30, 2023
**Doi:** - https://doi.org/10.37547/tajmei/Volume05Issue11-06

**Pavlo Morozov**
**Executive Director and Legal Representative, Guangzhou San Units Trading Limited, Guangzhou, China**

## ABSTRACT

Export-oriented B2B manufacturing with multi-component products, biological raw-material variability and seasonal release cycles is conventionally organized as fragmented contract manufacturing that scatters design, engineering, production, quality control and logistics across independent contractors. Each handover leaks integration, and the burden of architectural decisions falls on a buyer who lacks cluster-side information. This paper presents a system-level overview of an alternative configuration — the one-window business architecture — developed by the author through years of operational practice in the Guangzhou bags and leather goods cluster, and applicable to the wider class of export-oriented B2B production it exemplifies. The system rests on five foundational properties: a single object of management (the collection), a single point of accountability, linear causal connectivity between phases, proactive governance through formal gate decisions (Gate 0 through Gate 6), and reproducibility as a configuration property rather than a stochastic outcome. Its anatomy is described across three levels — an architectural level of principles and decision criteria, a process level of phased lifecycle governance, and an executive level of suppliers, manufacturers and logistics operators. The paper develops six building blocks: the foundational properties, the three-level structure, the collection as object of management, the central control loop, the gate-based process structure, and the system boundaries. It then contrasts the architecture with the traditional OEM model and shows that predictability, reproducibility and margin stability emerge from the integrated configuration rather than from contractual specifications. The paper serves as the architectural reference for an intended series of companion papers.

## KEYWORDS

Architectural integration, business architecture, collection-level governance, contract manufacturing, cross-border production, gate logic, leather goods industry, lifecycle management, one-window system, product lifecycle, single point of accountability, system architect, system boundaries, transaction cost economics.

## INTRODUCTION

Contract manufacturing of bags and leather goods for the cross-border business-to-business (B2B) segment is one of the older instances of globally distributed production. Brands and large wholesale buyers source their collections from production clusters whose specialized capacity, supplier density and accumulated craft tradition are not economically reproducible at the buyer's home location [1]. The standard organizational form for such sourcing is a chain of independent contractors (an industrial designer, a pattern engineer, one or more cutting and assembly factories, hardware vendors, an inspection firm, a freight forwarder, and a customs broker), each accountable for one segment of the lifecycle. Williamson [2] read this chain as a transaction-cost configuration in which lateral coordination economizes on hierarchical authority while shifting the synchronization burden onto the buyer. In the cross-border bags and leather goods setting, where product specifications interact tightly with raw-material variability, geographic distance from the consumer market and short seasonal windows, that synchronization burden hardens into structural risk.

The risk has a recognizable signature. Designs that satisfy aesthetic and merchandising criteria turn out infeasible at the bill-of-materials level. Factories that fulfill their technical pack produce items whose packaging volume undermines logistics economics. Inspections that meet a sampling standard pass material-grade substitutions invisible at the inspection point. Henderson and Clark [3] showed long ago that this class of failure originates not in component knowledge (each contractor knows their own work) but in architectural knowledge: in the understanding of how a decision in one component cascades through the system. Where architectural knowledge is dispersed across independent firms, the chain produces no architectural decisions at all, only locally rational ones, and the buyer is left to absorb the consequences.

The one-window business architecture, the subject of this paper, is a structurally different configuration developed by the author as a response to the structural failures described above. Rather than coordinating independent contractors, it integrates the lifecycle into a single managerial contour. The collection, not the individual stock-keeping unit (SKU), is the basic unit of management. Phase transitions are formal decisions, not ad-hoc handovers. Suppliers, factories and logistics operators sit on a separate executive level, governed by a small architectural level that holds the cross-cutting decision authority. The architecture was formulated, tested and refined through the author's years of continuous operational practice in the Guangzhou production cluster, during which explicit gate criteria, role definitions and boundary rules were iteratively developed and validated across multiple production cycles and client configurations.

The paper is a conceptual paper drawing on long-term practitioner observation in a single export-oriented B2B production cluster. It uses analytical generalization rather than statistical inference: the architectural claims advanced here are descriptions of an organizational configuration grounded in the cited literature on transaction-cost economics, architectural innovation, near-decomposability and dynamic capabilities, not generalizations from a sampled population. The genre carries known limitations — single-cluster anchoring, single-observer perspective, and absence of quantitative validation — which are acknowledged here and revisited in the Limitations section before the Conclusion. The author is the originator and principal architect of the one-window methodology; he is also the legal representative of a firm that operates this architecture in the cluster discussed. The diagnosis is therefore motivated by direct authorial practice, and the authorial position is fully disclosed (see Conflict of Interest).

This paper is the first in an intended series of companion papers. The remaining papers are expected to develop, respectively, the problem the system solves, the principle of collection-level governance, and the central subject who holds the architectural authority. The aim of the present paper is narrower and prior to the others: to describe what the system is, in enough detail that the reader can read the rest of the series with the architecture already in place.

The paper is organized around six architectural building blocks. These are the structural fragmentation that the system is designed against, the five foundational properties of the configuration, the three-level structure, the choice of the collection as object of management, the central control loop and the gate-based process structure, and the system boundaries. A short contrast with the traditional OEM model and a closing note on system-level properties complete the paper.

## STRUCTURAL FRAGMENTATION OF CONTRACT MANUFACTURING

A traditional cross-border OEM configuration for bags and leather goods can be drawn as a horizontal chain. The buyer, sitting in one market, transmits a request to a designer; the designer hands a concept to a pattern engineer; the engineer specifies tooling and materials and hands a technical pack to a factory; the factory contracts hardware, lining and shell suppliers; an independent inspection firm samples the output; a freight forwarder moves the goods; a customs broker clears them. Each segment is governed by a discrete contract; each segment has its own efficiency metric; each segment is locally rational.

This configuration produces three classes of failure that have been documented in the global value chain literature [1]. Information loss at handovers grows with chain length: every additional segment introduces an interpretation step, and the cumulative drift between the original buyer intent and the production parameters at the cutting table can be substantial. A second failure mode, accountability dispersion, follows from the contractual structure itself, since every contractor answers for their segment while none answers for the integrated outcome. Quality erodes more slowly: the initial reference sample is replaced over time by progressively cheaper material grades or simplified construction details, each substitution rational at the supplier level but cumulatively destructive at the collection level.

These are not failures of effort or competence. They are properties of the architecture. Point improvements such as tighter specifications, more inspection or an on-site agent change the magnitude of specific failure modes without changing the architecture that generates them; Tushman and Anderson [4] showed in a different setting that competence-enhancing adjustments inside an existing configuration cannot substitute for a competence-destroying shift to a new configuration when the structural conditions have changed. Maier's distinction between systems and systems-of-systems [5] is helpful here: the OEM chain is a system-of-systems in which each component has its own managerial and operational independence. The integrated outcome is not the responsibility of any component, and no component is positioned to deliver it. To deliver an integrated outcome reliably, the architecture itself must be different.

A second observation deepens the diagnosis. In the OEM chain the buyer occupies an architectural role for which the buyer is not equipped. The buyer holds the only complete picture of what the collection is supposed to be on the destination market, but the buyer typically lacks current information about cluster-side material grades, tooling availability, regulatory updates and supplier reputation. The buyer therefore sits at the cognitive boundary of the system without the operational reach to act on it. Whatever decisions reach the production cluster have already been filtered through the buyer's incomplete information, and any cluster-side variation that ought to update those decisions returns through a chain of intermediaries that loses fidelity at each step. Coordination overlays such as agents, inspections and weekly status meetings increase the buyer's information about the chain without giving the buyer authority over its configuration. The structural problem is not informational asymmetry alone; it is a misallocation of architectural authority.

## FOUNDATIONAL PROPERTIES OF THE ONE-WINDOW ARCHITECTURE

The one-window architecture rests on five foundational properties. Each is necessary, and absence of any one returns the configuration to the OEM baseline.

A single object of management. The unit governed by the architecture is the collection, understood as a coherent set of products designed together, sharing materials, hardware and a price logic, and released to a market on a single seasonal cadence. SKU-level management, common in the OEM baseline, optimizes individual products and fragments collection-level economics; collection-level management treats the assortment as a single economic organism. Simon [6] argued that complex systems are most tractable when they are decomposed into nearly independent subsystems whose interactions are governed at a higher hierarchical level; the one-window architecture relocates that hierarchical move from the individual product up one level, to the collection.

A single point of accountability. End-to-end responsibility for the collection's outcome — its commercial fit, its cost and its on-time delivery — sits with one subject rather than with a sequence of contracted parties. Williamson's account of hierarchical governance [2] anticipated this move at the abstract level. Hierarchical governance, in his account, localizes residual claims and residual control to a single integrating actor. Lawrence and Lorsch [7] showed empirically that organizations operating in environments of high differentiation require a corresponding investment in integration, and that the integration role cannot be diffused across the differentiated parts. In the one-window architecture, this role is held by the System Architect — the central subject to be discussed in detail in a dedicated companion paper.

Linear causal connectivity between phases. Each phase's output is the next phase's input under formal validation. Phases do not run in parallel where their outputs interact. A factory does not start cutting before raw-material qualification is closed; logistics is not contracted before packaging volumes are fixed; export documentation is not prepared before regulatory parameters are set. The principle is causal, not bureaucratic.

Proactive governance. Decisions on integration are taken before phases run, not after. Cooper's Stage-Gate framework [8] identified the cost asymmetry that motivates this property: the cost of preventing a defect at the engineering stage is a small fraction of the cost of correcting the same defect in serial production, which is itself a fraction of the cost of correcting it after export. The architecture institutionalizes that asymmetry through formal gate decisions at every transition.

Reproducibility. The configuration is designed so that the same collection, run on the same inputs, produces the same output across cycles. Reproducibility here is a property of the configuration rather than of any individual operator. It is what allows scaling without re-invention: when a new client or category enters, the system does not have to be reconstructed around them.

Table 1 summarizes the five foundational properties, their architectural functions and the consequences of their absence.

**Table 1 — Five foundational properties of the one-window architecture**

| Property | Definition | Architectural | Consequence of |
|---|---|---|---|

| | | Function | Absence |
|---|---|---|---|
| Single object of management | The collection (not the individual SKU) is the unit governed by the architecture | Unifies material, costing, logistics and calendar decisions across the assortment | SKU-level optimization fragments collection economics; cross-item synergies are lost |
| Single point of accountability | End-to-end responsibility for the collection's outcome sits with one subject (System Architect) | Eliminates accountability dispersion across contracted parties | No party owns the integrated outcome; failures are absorbed by the buyer |
| Linear causal connectivity | Each phase's validated output is the next phase's input; no parallel execution where outputs interact | Prevents unvalidated assumptions from propagating downstream | Upstream defects cascade undetected; cost of correction multiplies at each phase |
| Proactive governance | Integration decisions are taken before phases run, not after, through formal gate logic (Gate 0–Gate 6) | Institutionalizes the cost asymmetry between prevention and correction | Quality control becomes reactive; defects are detected only at inspection or after export |
| Reproducibility | The same collection, run on the same inputs, produces the same output across cycles | Allows scaling without re-invention; new clients enter an existing configuration | Each cycle is rebuilt from scratch; outcomes depend on individual operator performance |

## THE THREE-LEVEL ARCHITECTURE

The architecture distinguishes three functional levels — architectural, process, and executive — but their relationship is not a top-down hierarchy in the conventional corporate sense (Figure 1).

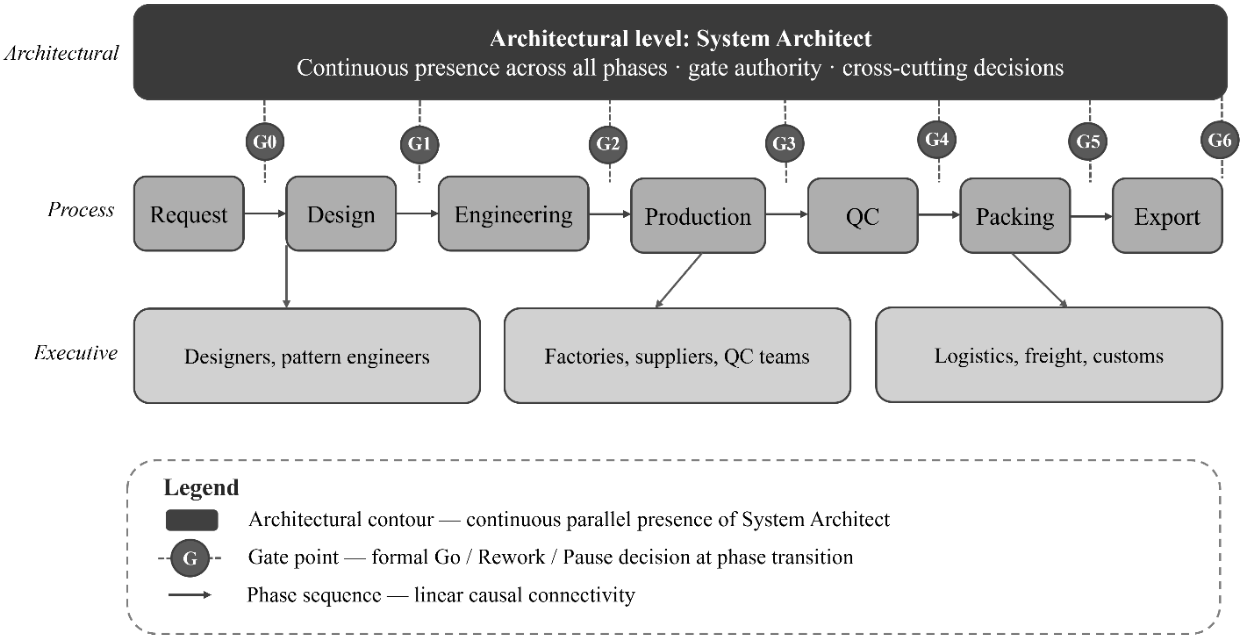


**Figure 1 — Three-level architecture of the one-window business system**

The process level is the horizontal backbone of the system: the lifecycle of a collection unfolds as a linear sequence of phases — market-request interpretation, design, engineering, prototyping, raw-material qualification, serial production, in-process control, outgoing control, and export closure. Each phase feeds its validated output into the next; the sequence is causal and irreversible under normal operation.

The architectural level runs parallel to this entire sequence rather than above it. The System Architect does not issue directives from a superior position and wait for reports; the System Architect accompanies the lifecycle from its first phase to its last, maintaining continuous operational presence at every stage. Architectural authority is exercised at the gate points — the formal transitions between phases — where the Architect verifies that the output of the closing phase meets the entry criteria of the next. Between gates, the Architect holds the cross-cutting parameters of the collection (materials book, costing envelope, regulatory constraints, calendar) and intervenes wherever a local decision would compromise the integrated outcome. The result is not hierarchical command but continuous architectural accompaniment of a horizontal process.

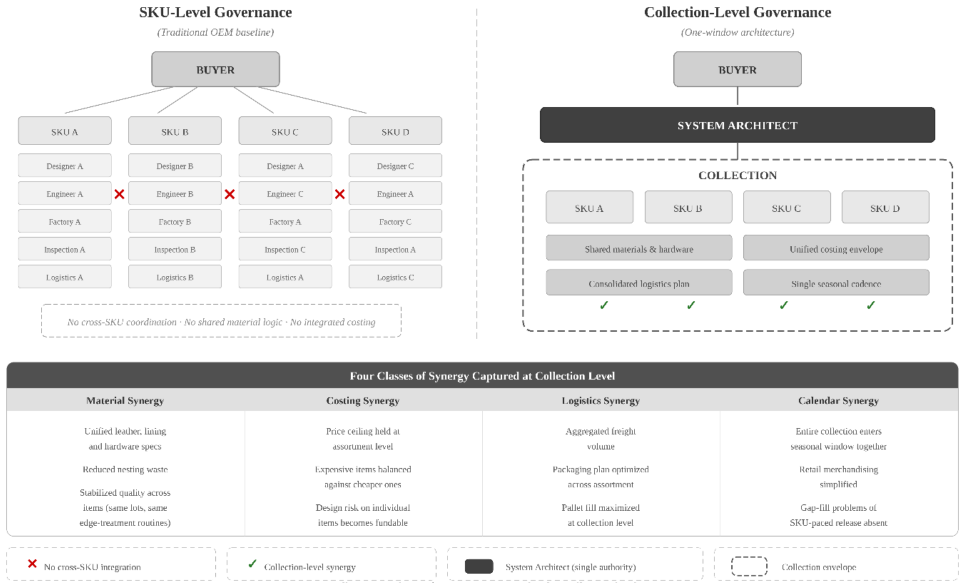
The executive level comprises the parties that physically carry out the work — cutting and assembly factories, hardware and lining suppliers, edge-treatment specialists, packaging vendors, freight forwarders, customs brokers. Each executive party is selected, qualified and managed through the architectural level; each executes within the parameters set by the process level. The executive level has no autonomous decision authority over the cross-cutting parameters of the collection.

Galbraith's information-processing view of organization design [9] anticipated the need for a dedicated integration function in complex, differentiated environments: cross-cutting decisions must be concentrated at the point where the information needed to make them is most economically aggregated. The classical application of this principle is a vertical hierarchy in which an upper level aggregates information from lower levels. In the one-window architecture the same principle is applied differently: the integrating function runs horizontally alongside the lifecycle rather than vertically above it, and information is aggregated not through reporting levels but through the Architect's direct operational presence at each phase and each gate. This is the structural distinction between a corporate hierarchy and an operational architecture — and it is what separates the one-window system from conventional management models.

Maier's architecting principles for systems-of-systems [5] remain relevant in a different respect: the executive level is a system-of-systems (independent firms with their own management and operations), and the architectural level imposes coherence on that system-of-systems not through ownership but through gate authority and supplier qualification — the minimum governance required to produce an integrated outcome from independent actors.

## THE COLLECTION AS OBJECT OF MANAGEMENT

The choice of object of management is the first architectural decision. The OEM baseline manages individual products; the one-window architecture manages collections. The choice is consequential (Figure 2).



Four Classes of Synergy Captured at Collection Level

| Material Synergy | Costing Synergy | Logistics Synergy | Calendar Synergy |
|---|---|---|---|
| Unified leather, lining and hardware specs | Price ceiling held at assortment level | Aggregated freight volume | Entire collection enters seasonal window together |
| Reduced nesting waste | Expensive items balanced against cheaper ones | Packaging plan optimized across assortment | Retail merchandising simplified |
| Stabilized quality across items (same lots, same edge-treatment routines) | Design risk on individual items becomes fundable | Pallet fill maximized at collection level | Gap-fill problems of SKU-paced release absent |

**Figure 2 — Collection as object of management: SKU-level vs collection-level governance**

A collection is not a portfolio. A collection is a coherent assortment that shares materials, hardware, construction logic, price positioning and seasonal cadence. That shared substrate creates four classes of synergy that an SKU-level configuration cannot capture. The first is material. When leather, lining and hardware specifications are unified across the assortment, nesting waste falls, procurement simplifies, and quality stabilizes across items, since the same lots and the same edge-treatment routines serve a much larger fraction of the production run than they would under SKU-level specification. Costing follows from this same logic but inverts the locus of the trade-off: the collection-level costing decision balances expensive items against cheaper ones, holding the price ceiling at the level of the assortment rather than item by item, and so admits design risk on individual items that would be unfundable in isolation. Logistics is a simpler matter — aggregated freight volume gives the packaging plan a degree of freedom it does not have at SKU level. The fourth synergy is calendar. Because the entire collection enters and exits the seasonal window together, retail merchandising is simplified and the gap-fill problems characteristic of SKU-paced release are largely absent.

The SKU-level baseline is structurally unable to capture these synergies. Each contractor optimizes within its own contract; nobody is contracted to balance costs across the assortment, and the nesting plan that maximizes material yield at the collection level requires a single pattern engineer with visibility across the entire material book, which the OEM baseline does not provide. Brusoni and colleagues [10] described an analogous asymmetry at the firm level — firms that "know more than they make" — and Iansiti's [11] account of technology integration in complex environments documented the conditions under which a single integrating actor with cross-component knowledge outperforms a contractually decomposed alternative. The one-window architecture relocates the same asymmetry to the collection.

The shift from SKU to collection also changes the temporal structure of management. SKU-level management runs on the calendar of individual products; collection-level management runs on the calendar of the seasonal release. The latter is shorter, more compressive and more decision-dense, which is one of the reasons that the architecture's gate logic is centralized in a single decision authority rather than in a corporate review committee.

The collection-as-unit choice also changes the criterion of success. Under SKU-level management, an item is a success if it meets its individual specification; the collection's success is the statistical aggregation of item-level successes. Under collection-level management, an item is a success only if it serves the collection — if it strengthens the assortment's coherence, fits the materials book, balances the costing, and slots into the seasonal release. The criterion shift has direct consequences for the engineering level: design decisions that look reasonable item-by-item are screened against the collection envelope, and the technical pack carries cross-item constraints that an item-level technical pack does not. Selva, Cameron and Crawley [12] documented systematically the patterns by which architectural decisions of this kind (what is held constant across a family, what is allowed to vary, where the disciplined interfaces are placed) recur across complex system architectures, and the one-window configuration sits squarely within that catalog.

## THE CENTRAL CONTROL LOOP

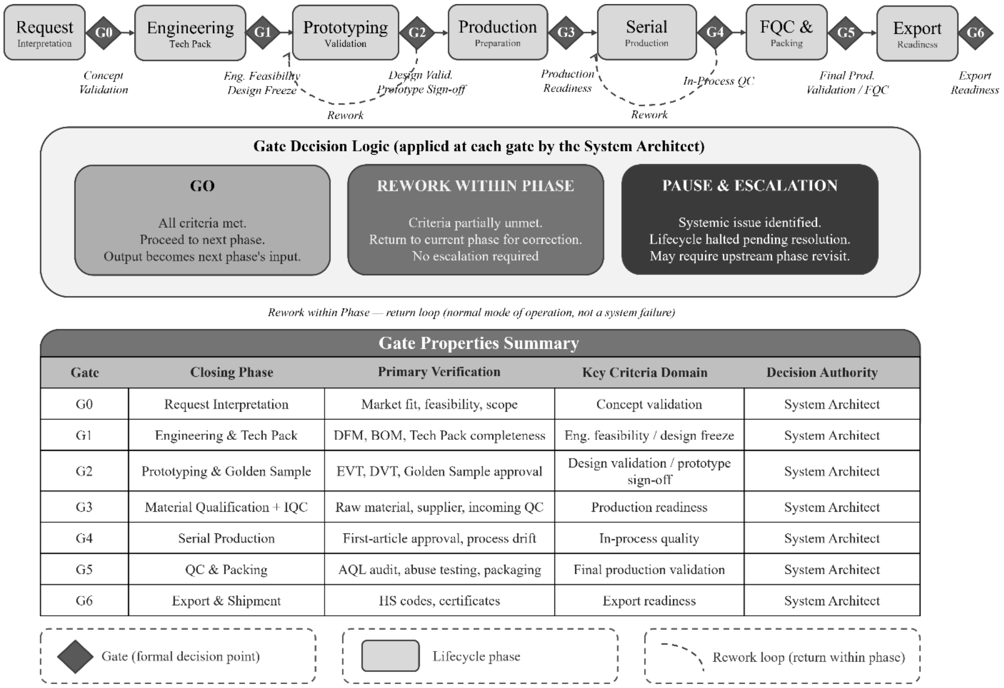
The architectural layer is operationalized through a central control loop that holds the lifecycle together. The loop has three components: a single decision authority, a fixed set of gate criteria, and a closed information flow. The decision authority is the System Architect — the role to be discussed in detail in a dedicated companion paper. For present purposes the relevant point is that gate decisions, supplier qualification decisions, and architectural-parameter decisions are concentrated in one subject; they are not distributed across executives.

Concentration of decision authority does not mean centralization of execution. Execution remains distributed across the executive layer; many parties act in parallel within their phases. What is concentrated is the authority over the transitions between phases and over the configuration of the executive layer itself. This is the architectural difference between the one-window architecture and a coordinated OEM chain: a coordinator receives outputs from each segment and arranges inputs for the next, but a coordinator does not have authority to swap a supplier mid-cycle, to overrule a phase deliverable, or to redefine a phase criterion. The architectural layer does.

The closed information flow runs continuously around the loop. Phase outputs are reported to the architectural level at each gate; gate decisions are communicated to the executive level for implementation; deviations and supplier performance are recorded and accumulated as institutional memory; the institutional memory is consulted at the start of each new cycle, both to qualify suppliers and to set realistic gate parameters.. Levitt and March [13] characterized exactly this kind of accumulation: organizations encode the lessons of past cycles into routines and stored inferences that guide subsequent action without re-deriving them from scratch. Galbraith's information-processing view of organization design [9] is helpful in the same context: the architecture is, in effect, a closed information-processing loop with the architectural layer as its hub. This closure is what distinguishes the one-window architecture from a project-management overlay applied to an OEM chain. A project-management overlay carries information across phases but does not own the phases; the architectural layer owns them.

## PROCESS STRUCTURE WITH LINEAR CONNECTIVITY AND GATE LOGIC

The process layer of the architecture is a linearly connected sequence of phases, each separated from the next by a formal gate (Figure 3). Seven gates govern the lifecycle of a collection:



Gate Decision Logic (applied at each gate by the System Architect)

GO
All criteria met.
Proceed to next phase.
Output becomes next phase's input.

REWORK WITHIN PHASE
Criteria partially unmet.
Return to current phase for correction.
No escalation required

PAUSE & ESCALATION
Systemic issue identified.
Lifecycle halted pending resolution.
May require upstream phase revisit.

Rework within Phase — return loop (normal mode of operation, not a system failure)

Gate Properties Summary

| Gate | Closing Phase | Primary Verification | Key Criteria Domain | Decision Authority |
|---|---|---|---|---|
| G0 | Request Interpretation | Market fit, feasibility, scope | Concept validation | System Architect |
| G1 | Engineering & Tech Pack | DFM, BOM, Tech Pack completeness | Eng. feasibility / design freeze | System Architect |
| G2 | Prototyping & Golden Sample | EVT, DVT, Golden Sample approval | Design validation / prototype sign-off | System Architect |
| G3 | Material Qualification + IQC | Raw material, supplier, incoming QC | Production readiness | System Architect |
| G4 | Serial Production | First-article approval, process drift | In-process quality | System Architect |
| G5 | QC & Packing | AQL audit, abuse testing, packaging | Final production validation | System Architect |
| G6 | Export & Shipment | HS codes, certificates | Export readiness | System Architect |

Figure 3 — Process structure with gate logic: lifecycle phases and formal transition decisions

**Gate 0** — Concept Validation (closing the request-interpretation phase; verifying market fit, positioning and commercial viability of the collection concept);

**Gate 1** — Engineering Feasibility / Design Freeze (closing the engineering development phase; confirming that the Tech Pack is complete, DFM analysis is passed, and the design is frozen for prototyping);

**Gate 2** — Design Validation / Prototype Sign-off (closing the prototyping phase; approving the Golden Sample as the absolute reference standard for serial production);

**Gate 3** — Production Readiness / IQC & Sourcing (closing raw-material qualification and incoming quality control, evaluated simultaneously; confirming that materials, tooling and production setup are ready);

**Gate 4** — In-Process Quality Control / First-Article Approval (verifying the first article against the Golden Sample before serial production continues; monitoring process drift through periodic audits);

**Gate 5** — Final Production Validation / FQC (statistical AQL audit of the finished batch before shipment; including functional stress testing);

**Gate 6** — Export Readiness (closing the cross-border shipment; verifying HS codes, certificates of origin, chemical test reports and packaging integrity).

Each gate is a formal decision taken by the System Architect according to three-way decision logic: Go (proceed to next phase), Rework within Phase (return to the current phase for correction without escalation), or Pause and Escalation (halt the lifecycle pending resolution of a systemic issue). The decision is governed by pre-specified criteria, not by the discretion of the decision-maker; the role of the decision-maker is to ascertain that the criteria are met.

The gates therefore play three combined functions. Their first function is to prevent unvalidated outputs from propagating into the next phase, where their cost of correction would be larger [6, 8]. Beyond that, they concentrate cross-cutting verification at exactly the points where cross-cutting verification is meaningful — at transitions, where the assumptions of the next phase are formed. For the buyer, they offer a structured certification of where the collection is in its lifecycle, without exposing the buyer to the operational interior of the chain.

Linear connectivity is the second discipline of this level. Phases are not run in parallel where their outputs interact. The architectural level permits limited parallelism — for example, packaging design and serial production can run in parallel after Production Readiness — but never where one phase's output is a load-bearing input of another. Rework — return to a previous phase — is permitted and is treated as a normal mode of operation rather than as a failure of the system: when a downstream phase exposes a defect in an upstream assumption, the lifecycle returns to the upstream phase, the assumption is corrected, and the lifecycle resumes. This is the operational form of Simon's [6] near-decomposability argument: a small set of disciplined interfaces between phases gives the system a finite number of structured states and a finite number of legitimate transitions between them, which is the property that allows the configuration to produce reproducible outcomes despite local variation.

A practical consequence of the gate logic is that the lifecycle is auditable on a phase-by-phase basis. The state of any collection in flight can be reported with reference to the last gate it has passed and the next gate it is approaching, together with the criteria still outstanding. The buyer therefore receives a structured certification of progress without being exposed to the operational interior of the chain. This auditability is one of the architecture's principal exports to the buyer: it converts a process whose interior was opaque under the OEM baseline into a process whose state is unambiguous at every reporting moment. It also stabilizes the architectural level's own decision discipline: the criteria are formalized, applied uniformly across collections, and revised through institutional memory rather than through individual judgment at the moment of decision.

## SYSTEM BOUNDARIES

A common misunderstanding of the architecture is that it tries to absorb everything around the buyer. It does not. The architecture has a sharp boundary, and what stays outside the boundary is part of its design.

Inside the boundary are the activities whose integration is the architecture's reason for existing — market-request interpretation, design, engineering, prototyping, supplier qualification, raw-material qualification, serial production, quality control, packaging, export documentation, customs clearance and cross-border delivery. Outside the boundary are activities that belong to the buyer's commercial model, including market strategy, pricing in the destination market, brand communication, retail merchandising, sales-force management, and post-sale customer service. The architecture does not enter those activities; it interfaces with them through a single point of contact that translates the buyer's market parameters into the architectural level's design constraints.

Table 2 specifies the boundary position and rationale for each function.

**Table 2 — System boundary: functions inside and outside the architectural contour**

| Function | Boundary | Rationale |
|---|---|---|
| Market-request interpretation | Inside | Translates buyer intent into architectural parameters; misinterpretation cascades through the entire lifecycle |
| Design and cross-cultural adaptation | Inside | Requires simultaneous knowledge of destination-market aesthetics, production constraints and regulatory requirements |
| Engineering and prototyping | Inside | Technical pack and BOM decisions determine material yield, production feasibility and unit cost across the collection |
| Supplier qualification and raw- | Inside | Material-grade decisions interact |

| | | |
|---|---|---|
| material sourcing | | with design, costing and regulatory compliance; cannot be isolated without information loss |
| Serial production management | Inside | Process stability and drift prevention require architectural-level authority over factory operations |
| Quality control (in-process and outgoing) | Inside | Gate verification at transitions is the architecture's primary mechanism for preventing defect propagation |
| Packaging and pre-shipment preparation | Inside | Packaging volumes interact with logistics economics and export documentation; late-stage changes are costly |
| Export documentation and customs clearance | Inside | Regulatory parameters (REACH, CPSIA) must be set at the design phase and verified at export; separation creates compliance risk |
| Cross-border logistics and delivery | Inside | Freight planning depends on packaging volumes, calendar and export-readiness parameters set upstream |
| Market strategy and brand positioning | Outside | Belongs to the buyer's commercial model; inclusion would dilute architectural focus without adding integration value |
| Pricing in the destination market | Outside | Determined by the buyer's competitive environment; the architecture provides cost structure, not retail price |
| Retail merchandising and sales | Outside | Post-delivery activity governed by the buyer's distribution model |
| Post-sale customer service | Outside | Outside the production lifecycle; no integration dependency with upstream phases |

The criterion for inclusion in the architectural boundary is integration value. A function is inside the boundary if integrating it materially reduces handover loss, accountability dispersion or quality erosion, and if its integration is governable through the architectural level's authority. A function is outside the boundary if its integration would dilute the architecture's focus or duplicate the buyer's competencies. Adner's [14] treatment of the ecosystem as a structure of complementary roles makes the same point in a different vocabulary: the value proposition that an integrating actor can deliver is bounded by the activities it is positioned to align, and pulling activities inside that perimeter without an alignment requirement reproduces the costs of vertical integration without recovering its benefits. The architecture is therefore deliberately minimal — it integrates exactly what needs to be integrated, no more.

The boundary is also a defense of the architecture's own coherence. Each function brought inside the boundary expands the architectural level's scope and increases the cross-cutting decisions that must be coordinated. If the scope expands beyond the architectural level's effective bandwidth, the level degrades into a fragmented set of sub-architectures that interact informally — at which point the integration that the architecture exists to deliver is silently lost. The minimalism of the boundary is therefore not a stylistic preference but an operating constraint. It is also one of the points at which the architecture's design diverges from the project-management overlay: an overlay tends to absorb adjacent activities by accretion, whereas the architecture defends its scope deliberately and exits activities that no longer contribute integration value to the lifecycle.

## EMERGENT SYSTEM PROPERTIES

The properties that buyers ultimately purchase from the architecture — predictability of outcome, reproducibility of quality, stability of margin, calendar reliability — are not properties of any individual component. They are emergent properties of the integrated configuration. By "emergent" the paper means, precisely, system-level properties produced by the architectural-control mechanism through cross-cutting constraint applied to component decisions, rather than properties spontaneously arising without architectural authority; the term names a hierarchical-control outcome, not an unplanned aggregation. Anderson's review of complexity theory in organization science [15] frames this category of property as a product of the interaction structure rather than of the components in isolation. The wider literature on organizational adaptation and dynamic capabilities [16, 17, 18, 19] reaches the same conclusion from different starting points. March [16] in particular argued that the persistence of an organizational form depends on its capacity to balance the exploitation of established routines with the exploration of new ones, a balance that the architectural level holds at the inter-firm level.

The emergence is produced through three architectural operations working in combination. Local optimizations of each executive party are constrained by the cross-cutting parameters of the collection: a factory that would otherwise lower its cost by substituting a leather grade is held to the technical pack and to Production Readiness verification, and a logistics provider that would otherwise optimize on container fill is held to the export-readiness parameters set at Concept Validation. Causal connection across phases is the second operation, ensuring that design choices are taken with awareness of nesting consequences, hardware choices with awareness of regulatory consequences, and packaging with awareness of unit-economics consequences. Residual responsibility for the integrated outcome then sits with the architectural level, which owns the result even when no single executive party has produced it alone [10].

A consequence is that the emergence does not survive architectural substitution. The system is not a stack of artifacts that can be handed to another holder; it is an ongoing operational practice, reproduced cycle by cycle by the continuing presence of the architectural level. This is why product lifecycle management literature [17] treats PLM as a discipline rather than as a tool: the discipline holds the architecture in place across cycles. Cross-cultural translation of buyer intent into producible specifications is one of the architecture's most demanding duties for the same reason — it is a function of the architectural level's continuous presence in the cluster, not of any artifact transferable to a representative.

The same logic explains why the architecture's predictability is aggregate rather than absolute. Any individual cycle can be disrupted by a raw-material shock, a regulatory change, or a logistics event. What the architecture guarantees is not the absence of disturbance but the system's capacity to absorb disturbance without losing the integrated outcome. When a leather lot fails Production Readiness, the architectural level redirects the cycle to a verified alternative within the supplier book; when a shipping route closes, the export-readiness parameters are re-set and the calendar is re-anchored at the architectural level rather than at the freight forwarder. The predictability that the buyer experiences is the predictability of the system's response to disturbance, not the predictability of an undisturbed environment. This is the operational meaning of treating reproducibility as a configuration property — the configuration is what stays stable when the inputs vary. Eisenhardt and Martin [18] and Helfat and Peteraf [19] developed at the firm level the related notion that dynamic capabilities are the routines by which an organization reconfigures its operational set in response to environmental change; Helfat and Winter [20] sharpened the distinction between operational capabilities, which produce stable outputs in stable conditions, and dynamic capabilities, which absorb disturbance and reconfigure the operational set. The architectural level of the one-window system is the inter-firm analog of the dynamic capability.

## CONTRAST WITH THE TRADITIONAL OEM MODEL

Table 3 summarizes the architectural contrast between the one-window architecture and the traditional OEM model. The contrast is at the level of architecture, not of effort.

**Table 3 — Architectural contrast between traditional OEM and the one-window business architecture**

| Dimension | Traditional OEM model | One-window architecture |
|---|---|---|
| Object of management | Individual SKU | Collection |
| Accountability | Segmented by contract | End-to-end, single point (System Architect) |
| Phase connectivity | Independent, parallel | Linear, gate-validated (Gate 0–Gate 6) |
| Decision authority | Distributed across contractors | Concentrated in architectural layer |
| Quality management logic | Reactive: point inspection, post-factum | Proactive: gate logic with built-in control at every phase |
| Quality assurance | Inspection at output | Gate verification at transitions |
| Gate decision logic | Pass/fail or ad hoc | Go / Rework within Phase / Pause & Escalation |
| Deviation management | Reactive: detect and correct post-factum | Proactive, architectural: prevention through gate logic |
| Export readiness | Formed at final stage as a separate process | Embedded at the design level of the collection |
| Cross-cultural adaptation | Separate task, handled ad hoc | Built-in competence, applied at the design phase |
| Institutional memory | Dispersed across participants, not accumulated systemically | Concentrated in system archive, applied in subsequent cycles |
| Speed of decision | Committee-paced | Seasonal-paced, single authority |
| System integrity | Emergent failure | Emergent success |
| Scaling logic | Add contractors | Verify and integrate suppliers |
| Buyer cognitive load | High | Low |

The one-window architecture is not a universal replacement for the OEM model. Where production is a single product run, where the seasonal calendar is irrelevant, where the buyer has the in-house architectural capacity to coordinate the chain, the OEM model is efficient and the one-window architecture is overhead. The architecture is specifically suited to repeated cross-border collections with short seasonal windows, multi-component products, raw-material variability and cultural distance between production and consumer market — the conditions under which fragmentation has been documented to produce the structural risks that motivated the architecture in the first place [1].

A frequent informal alternative is to overlay project-management discipline on a traditional OEM chain by adding tighter specifications, more inspections, and an on-site representative who reports to the buyer. As Henderson and Clark's [3] analysis of architectural failure already implies, adding bureaucratic discipline to a chain that lacks an architectural authority does not produce architectural outcomes; it produces a more elaborate version of the same chain. The architectural difference is a matter of where the cross-cutting decisions are made, not of how much process is wrapped around them.

## LIMITATIONS

The argument advanced in this paper is grounded in the author's long-term operational practice and draws on conceptual analysis rather than statistical methods. Four limitations should be stated plainly. First, the architectural description is anchored in a single production cluster (Guangzhou bags and leather goods) and the configuration's portability to other clusters or product categories is asserted analytically rather than demonstrated comparatively. Second, the paper does not present disconfirming cases — production environments where the one-window configuration was attempted and failed, or where the OEM baseline outperformed it — and the absence of such cases means the boundary conditions of the architecture's effectiveness are inferred rather than mapped. Third, the diagnosis rests on the author's longitudinal observation and operational practice rather than on independent multi-observer fieldwork, which constrains the kinds of error a single observer is positioned to detect. Fourth, the paper offers no quantitative validation: the system properties claimed (predictability, reproducibility, margin stability) are not measured here against control configurations, and the testable propositions stated below in the Conclusion are advanced as hypotheses for subsequent empirical work, not as verified findings.

## CONCLUSION

This paper has presented the one-window business architecture as an integrated managerial system for cross-border B2B production of bags and leather goods. The system is organized around a single object of management — the collection — and a single point of architectural accountability held by the System Architect. It runs the lifecycle as a linearly connected horizontal process, governed by three functional levels, with seven formal gates (Gate 0 through Gate 6) as the only legitimate phase transitions and with reproducibility as a configuration property rather than a stochastic outcome. The architecture has a sharp and deliberately minimal boundary: it integrates exactly what needs to be integrated to reduce handover loss, accountability dispersion and quality erosion, and it interfaces cleanly with the buyer's commercial activities, which remain outside.

The architecture is not a universal replacement for the OEM model. Its proper domain is the repeated cross-border collection: short seasonal windows, multi-component products, raw-material variability, cultural distance between production and consumer market. Within those conditions, it produces predictability, reproducibility and margin stability as system-level properties of the integrated configuration rather than as the sum of contractual specifications.

To make the argument empirically tractable, three testable propositions are stated for subsequent research. P1: gate-validated lifecycle chains exhibit lower variance in lead time than gate-free chains of comparable length, controlling for product complexity and seasonal cadence. P2: collection-level objects of management produce lower per-unit landed cost than SKU-level objects of management on assortments above a threshold of shared materials and hardware, controlling for order volume. P3: configurations in which architectural authority is concentrated in a single subject exhibit faster recovery from raw-material or logistics disturbance than configurations in which architectural authority is distributed across contractually independent parties, controlling for disturbance magnitude. Each proposition specifies a directional claim, a control variable and an observable outcome, and each can in principle be tested by comparative fieldwork across cross-border production clusters.

This paper is the first in an intended series of companion papers. Subsequent papers are expected to develop, respectively: the structural problem the architecture is designed against, the architectural decision to manage at the collection level, and the central subject — the System Architect — who holds the architectural authority. Each paper is designed to be read independently of the others, but each presupposes the architectural overview given here.

## CONSENT FOR PUBLICATION

Not applicable. The work does not involve human or animal subjects, and no patient or third-party consent is required. The author consents to the publication of the paper in its present form.

## CONFLICT OF INTEREST

The author is the originator and principal architect of the one-window methodology described in this paper. He developed the architecture through years of continuous operational practice in the Guangzhou bags and leather goods production cluster and currently serves as the Executive Director and Legal Representative of Guangzhou San Units Trading Limited, a company that operates this architecture. This relationship constitutes a non-financial competing interest with respect to the conceptualization and exposition of the architecture. No financial competing interests are declared.

## ACKNOWLEDGEMENT

The author thanks the production teams, suppliers and B2B partners in the Guangzhou bags and leather goods cluster whose long-term collaboration within the author’s one-window architecture provided the operational context and empirical basis for the present work.

## REFERENCES

1. G. Gereffi, J. Humphrey, and T. Sturgeon, "The governance of global value chains," Rev. Int. Polit. Econ., vol. 12, no. 1, pp. 78-104, Feb. 2005.
2. O. E. Williamson, The Economic Institutions of Capitalism. New York: Free Press, 1985.
3. R. M. Henderson and K. B. Clark, "Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms," Adm. Sci. Q., vol. 35, no. 1, pp. 9-30, Mar. 1990.
4. M. L. Tushman and P. Anderson, "Technological discontinuities and organizational environments," Adm. Sci. Q., vol. 31, no. 3, pp. 439-465, Sep. 1986.
5. M. W. Maier, "Architecting principles for systems-of-systems," Syst. Eng., vol. 1, no. 4, pp. 267-284, 1998.
6. H. A. Simon, "The architecture of complexity," Proc. Am. Philos. Soc., vol. 106, no. 6, pp. 467-482, Dec. 1962.
7. P. R. Lawrence and J. W. Lorsch, "Differentiation and integration in complex organizations," Adm. Sci. Q., vol. 12, no. 1, pp. 1-47, Jun. 1967.
8. R. G. Cooper, "Stage-gate systems: A new tool for managing new products," Bus. Horiz., vol. 33, no. 3, pp. 44-54, May 1990.
9. J. R. Galbraith, Designing Complex Organizations. Reading, MA: Addison-Wesley, 1973.
10. S. Brusoni, A. Prencipe, and K. Pavitt, "Knowledge specialization, organizational coupling, and the boundaries of the firm: Why do firms know more than they make?," Adm. Sci. Q., vol. 46, no. 4, pp. 597-621, Dec. 2001.
11. M. Iansiti, "Technology integration: Managing technological evolution in a complex environment," Res. Policy, vol. 24, no. 4, pp. 521-542, Jul. 1995.
12. D. Selva, B. Cameron, and E. Crawley, "Patterns in system architecture decisions," Syst. Eng., vol. 19, no. 6, pp. 477-497, Nov. 2016.
13. B. Levitt and J. G. March, "Organizational learning," Annu. Rev. Sociol., vol. 14, pp. 319-340, 1988.
14. R. Adner, "Ecosystem as structure: An actionable construct for strategy," J. Manag., vol. 43, no. 1, pp. 39-58, Jan. 2017.
15. P. Anderson, "Complexity theory and organization science," Organ. Sci., vol. 10, no. 3, pp. 216-232, Jun. 1999.
16. J. G. March, "Exploration and exploitation in organizational learning," Organ. Sci., vol. 2, no. 1, pp. 71-87, Feb. 1991.
17. J. Stark, Product Lifecycle Management (Volume 1): 21st Century Paradigm for Product Realisation, 4th ed.

Cham, Switzerland: Springer, 2022.

18. K. M. Eisenhardt and J. A. Martin, "Dynamic capabilities: What are they?," Strateg. Manag. J., vol. 21, no. 10-11, pp. 1105-1121, Oct.-Nov. 2000.
19. C. E. Helfat and M. A. Peteraf, "Understanding dynamic capabilities: Progress along a developmental path," Strateg. Organ., vol. 7, no. 1, pp. 91-102, Feb. 2009.
20. C. E. Helfat and S. G. Winter, "Untangling dynamic and operational capabilities: Strategy for the (n)ever-changing world," Strateg. Manag. J., vol. 32, no. 11, pp. 1243-1250, Nov. 2011.