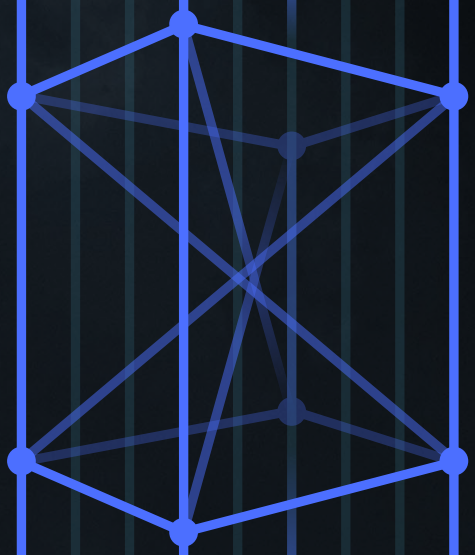
Autonomous Execution Fabric

FIVE AI LESSONS ON ENTERPRISE AI



go autonomous

Contents

3 Foreword

4 Key points

6 I – Not just an agent

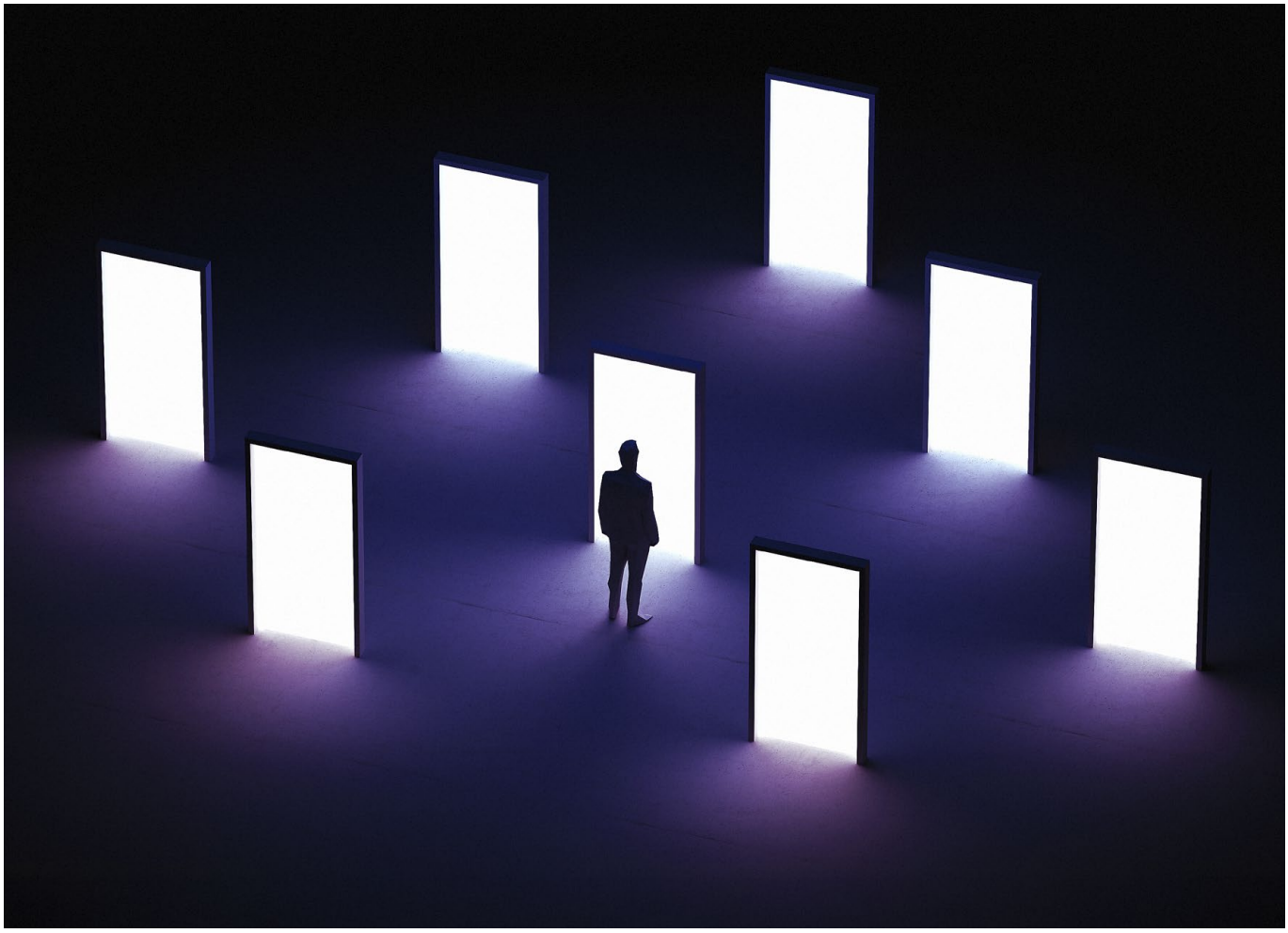
10 II – Not just input-output

12 III – You need a car, not just an engine

14 IV – Be exception-hungry

16 V – Do not forget continuous improvement

18 Conclusion



Foreword

Autonomous Commerce is a system powered by artificial intelligence that independently manages and executes the buying and selling of goods and services, enabling business operations and transactions to be completed in real time and fully autonomously.

Execution Fabric, enabling such autonomy in commercial operations, represents a pivotal innovation in the future of enterprise. However, success is not guaranteed for everyone.

Commerce in general, and B2B commerce in particular, is highly complex. Accounts, products, logistics, pricing, exceptions, data and actions in multiple systems, approvals flow – cause a myriad of decisions for every transaction.

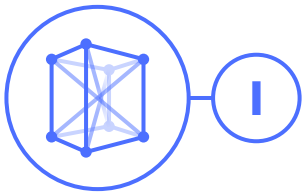
This domain demands a deep, specialized assessment that goes beyond a generic intelligent agent approach. Crucially, selecting the right intelligence hinges on thoroughly understanding the existing workflow, the people involved, and the underlying systems.

The fundamental change in how enterprises operate due to Autonomous Commerce demands a similarly transformative approach to problem-solving.

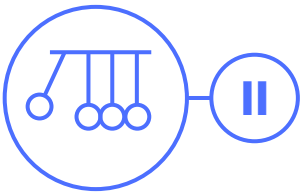
Key points

Autonomous Commerce is a new paradigm where AI-powered systems execute commercial operations end-to-end, from request to fulfillment, without human intervention.

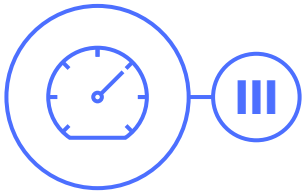
To succeed, enterprises must move beyond demos and fragmented tools, embracing unified, continuously learning systems purpose-built for commerce.



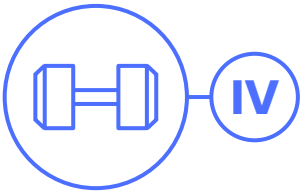
Not just an agent. Autonomous execution, particularly in complex B2B transacting, demands more than a simple, single-step "agent" approach. Real-world commercial workflows are multi-faceted and require an "ensemble of intelligence" to achieve true, end-to-end autonomy. While treating B2B transacting as a single "task" can produce impressive demonstrations, this limited view risks failing to build the necessary trust for sustained, autonomous operation.



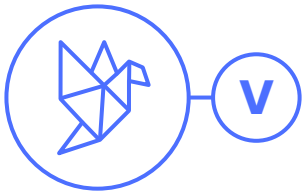
Not just input-output. Building a true execution fabric for B2B commerce requires moving beyond isolated, use-case-driven workflows that create rigidity and silos. Instead, it must enable multi-threaded execution, continuous, long-lived transactions, and layered data intelligence—allowing complex requests to run in parallel, evolve over time, and remain traceable across multiple systems for a cohesive, adaptive, and resilient operational model.



You need a car, not just an engine. An exceptional intelligence engine is merely the starting point. Just as a car requires a steering mechanism, windshield, and wheels, a true enterprise solution must incorporate a purpose-built non-disruptive AI-human interface, robust monitoring capabilities, and seamless integrations to deliver real value.



Be exception-hungry. To create a truly versatile and real-world-ready intelligence, training must prioritize data-rich edge cases. For a single organization, acquiring such a comprehensive dataset is often prohibitively expensive, time-consuming, or outright impossible, thus preventing the solution from achieving fully autonomous, customer-facing operations.



Do not forget continuous improvement. Companies that view AI implementation at the core of their operations as a singular event risk missing out on significant value or even outright failing the attempt. A truly transformative shift in operations must be managed as an ongoing internal program, integrating AI and software into comprehensive change management initiatives with executive support.

The outcome is not just efficiency — it's a structural shift toward scalable, self-improving business operations.

I – Not just an agent



Every part of the commercial process demands a different class of intelligence. Some tasks, such as ERP validation or price consistency checks, require simple logical frameworks or deterministic prediction models. Others, such as interpreting intent from a free-form email or classifying a non-standard purchase order, benefit from generative or probabilistic reasoning.

Multi-paradigm environment

The key is orchestration: knowing when to apply deterministic logic for guaranteed outcomes and when to use stochastic reasoning to infer missing context. An AI system for Autonomous Commerce must therefore be built as a **multi-paradigm environment**, combining symbolic reasoning, deterministic prediction, and generative modelling into a single operational flow.

Exhibit 1. **The multi-paradigm environment for the Enterprise AI**

	Intent Identification	Full Context Understanding	Data Capture & Cognification	Post Reasoning	ERP Validation & Acting
Rule-Based Systems	Simple routing (sender, keywords)		Data validation rules, template parsing	Hard business constraints, approval thresholds	API safety checks, compliance gating
Discriminative AI	Document & intent classification	Entity extraction, structural parsing	Deterministic field prediction, normalization	Anomaly detection, outlier alerts, ranking	
Generative / Agentic AI	Ambiguous intent resolution, dialogue interpretation	Context reconstruction, missing link inference	Filling incomplete data, semantic enrichment	Contextual reasoning, human-like judgment	Process planning ("if missing X, create Y")
Multimodal Models	Text + layout recognition (emails, PDFs, scans)	Cross-format alignment (tables, images, spreadsheets)	Visual grounding for traceability	Visual evidence verification	Document audit trail generation
Domain-Specific Models	Commerce-specific intent recognition	Understanding of B2B entities, product logic, and pricing	Specialized extraction of POs, invoices, and contracts	Context-aware decisioning using business logic	Reliable integration with ERP & CRM schemas
Knowledge-Based Systems		Linking entities across systems and time via ontology (Tacit knowledge)	Semantic validation (taxonomy-level checks)	Rule chaining for explainable reasoning	Policy enforcement, traceable audit logic

The multi-paradigm environment describes how it is fundamental to keep all aspects of system development in mind when developing and operating large-scale AI systems for operationally critical systems, such as quote-to-cash. It's like with the human brain, if your body used all the energy on the Prefrontal Cortex, you might be the best problem solver in the world; however, unable to move your feet.

Horizontal, general-purpose GenAI models have proven powerful for open-ended reasoning, summarisation, and creative tasks. However, **they fall short when depth and precision matter**. In B2B contexts, errors have consequences: an incorrect order line, a misinterpreted delivery term, or an invalid product number can disrupt entire supply chains.

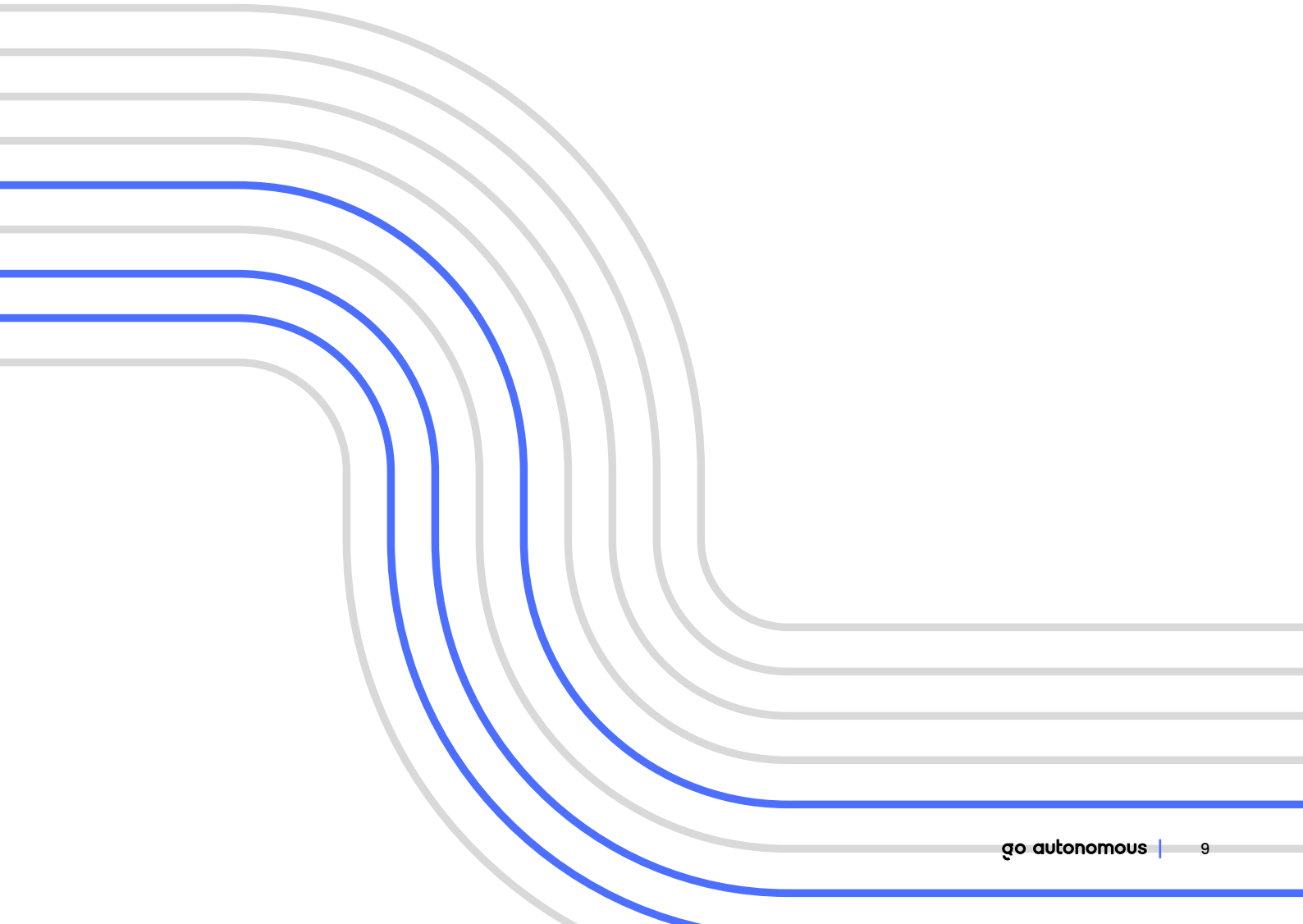
To succeed, AI in commerce must be **vertically specialized**. It must understand the semantics of part numbers, supplier IDs, contractual clauses, and currencies—not just language. It must reason within the constraints of business logic, not imagination. General models excel in generalization; specialized models excel in specialization.

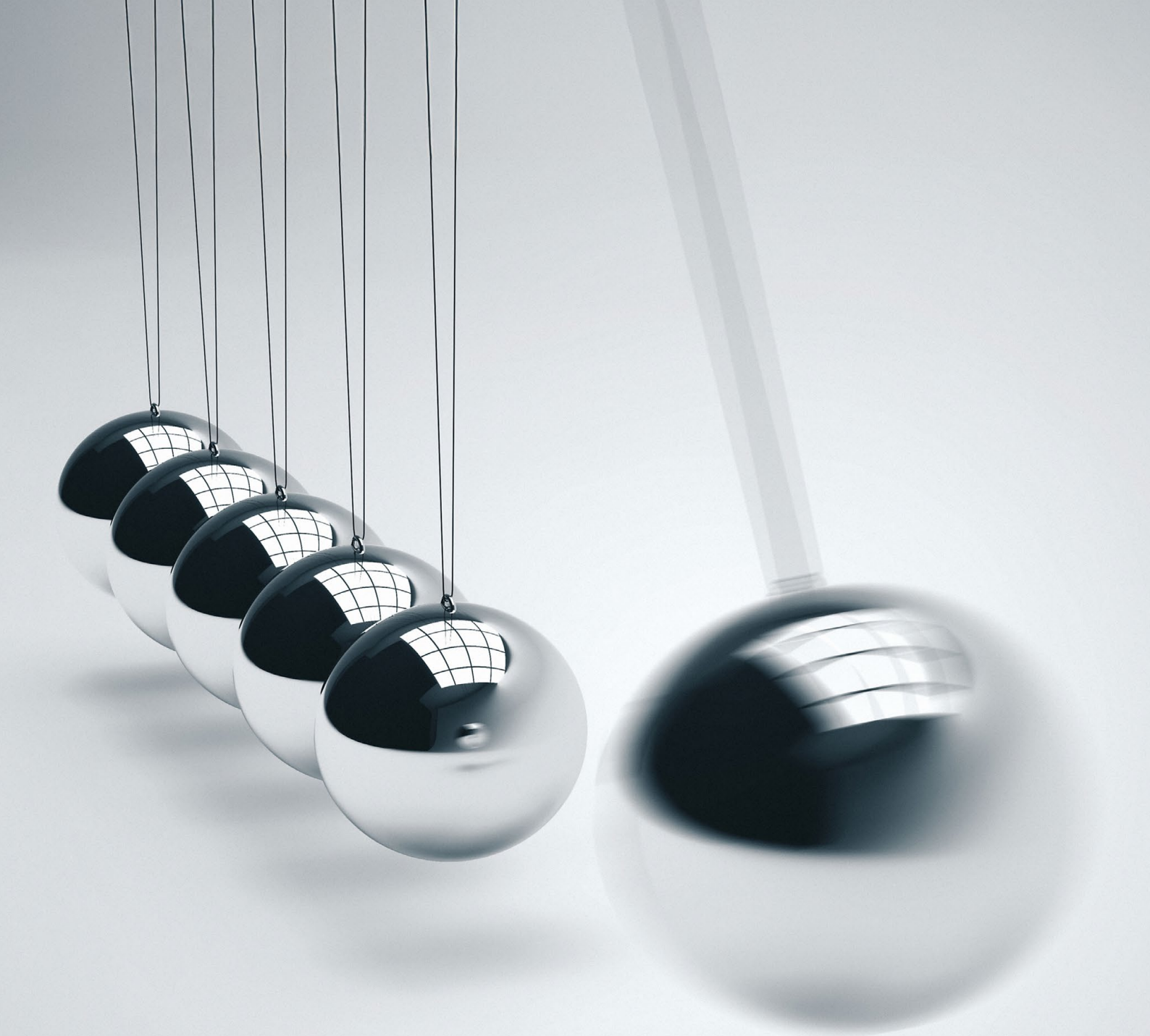
The true value of Autonomous Commerce lies in outcomes, not assistance. The perceived value of AI is often measured in "hours saved" or "emails answered faster." While useful, these metrics miss the point. The **actual value** is realized when products reach customers faster, errors disappear, and the enterprise can scale without adding people. To capture the full value, you need to stay on top of your analytics. In the modern era of Digital Commerce, understanding your bottlenecks, shifts in demands and customer sentiment is key to optimisation.

The foundation of any successful Autonomous Commerce system is to focus on the problem. It is easy to be captivated by the release of ever-larger models and novel frameworks. In enterprise automation, you need to keep a complete overview of all technologies and apply them where they provide the most value, all the way from simple symbolic rules to complex multimodal and generative models.

> We need to make sure that when we handle an order, it's made correctly. It's the right pricing, it's the right products, it's the right materials. You cannot do that if you apply general AI. You need a domain specific knowledge space in order to make sure that you get it right the first time.

Anders Liechti, CIO Nilfisk





II – Not just input-output

In B2B commerce, many organizations attempt to build their execution fabric through a use-case-by-use-case approach. While this seems practical at first—tackling specific customer needs one workflow at a time—it often leads to a fragmented and brittle architecture. Each use case becomes an isolated silo, with rigid logic and narrow integration boundaries. Over time, this patchwork of disconnected workflows limits flexibility, impedes innovation, and prevents the organization from realizing the full potential of an integrated execution fabric. The result is a system that can handle straightforward transactions but struggles with complexity, scale, and adaptability.

Multi-threading.

Real-world B2B transactions rarely follow a single, linear path. A **single customer request can simultaneously trigger multiple independent parallel actions**—such as updating a quote, retrieving product information, and initiating an approval workflow—all involving different systems and stakeholders. This demands a multi-threaded execution model: the ability to launch, track, and reconcile multiple parallel threads of activity from a single originating request. The execution fabric must not only parallelize tasks efficiently but also maintain coherence across them—ensuring traceability, synchronization, and consistent state across all processes involved.

Continuity of execution over time.

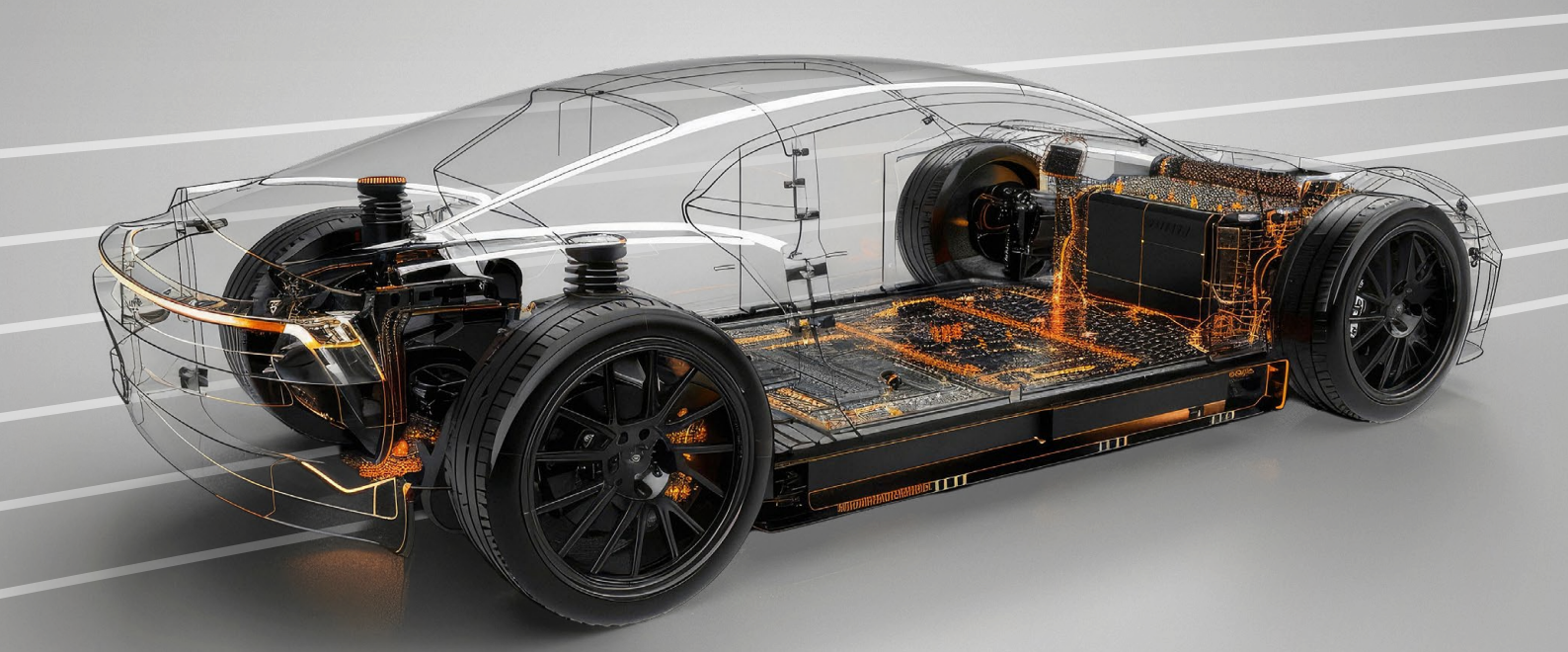
Modern B2B operations require continuity of execution over time. Unlike simple input-output transactions, these processes unfold as long-lived interactions that evolve through multiple stages—such as a quote-to-order conversion, supplier coordination, or abandoned-quote recovery. Each of these scenarios involves dynamic dependencies, asynchronous responses, and business logic that extends over days or even weeks. The execution fabric must therefore **support persistent, stateful execution**—capable of pausing, resuming, and adapting to changing conditions while maintaining complete context across all related transactions.

Multiple layers of data.

An effective execution fabric must also manage multiple layers of data representation for each transaction. It starts with **raw input data** (such as Excel files, PDFs, or free text), which must be interpreted and structured through system intelligence to get the interpreted data. Then comes the **contextualized data**, where data is refined based on historical interactions, insights from previous transactions and other systems. Finally, an **execution data** structures this information into actionable objects—like orders or quotes—ready for operational use. Maintaining all four data states in parallel enables traceability, auditability, and adaptive learning, ensuring that every execution is both transparent and context-aware.

Building an execution fabric for B2B commerce requires moving beyond transactional thinking toward orchestrated, intelligent, and resilient execution. It must integrate multi-threaded processing, continuous lifecycle management, and layered data comprehension into a unified, adaptive framework. Only then can businesses eliminate silos, handle complexity with agility, and deliver seamless, end-to-end execution that truly reflects the interconnected reality of B2B commerce. The goal is not just to automate workflows, but to build a living, evolving execution system—one that can learn, adapt, and scale with the enterprise.

III – You need a car, not just an engine



The goal is not to build another AI model; it is to create an **autonomous system capable of understanding, reasoning**, and acting across every stage and channel of a commercial transaction, with the reliability, transparency, and determinism required in enterprise environments.

Partial automation may look impressive in demos, but it rarely creates tangible business impact. Only when the **full end-to-end process**—from request to fulfilment—is automated can organisations achieve measurable efficiency gains, customer satisfaction, and working capital improvements.

Having exceptional intelligence in your AI system is just the beginning. Like a car with a powerful engine, intelligence alone cannot deliver value unless paired with steering, wheels, and a windshield. In the enterprise context, this means pairing intelligence with a purpose-built human interface, robust monitoring, and deep integrations. These three components turn raw intelligence into a true, operational enterprise machine.

Context-aware AI-human interface.

B2B commerce is inherently complex. It revolves around documents, PDFs, Excel sheets, links, databases, and structured data — not chat-style conversations. Real enterprise users don't need another standalone chatbot or dashboard; they need AI seamlessly embedded into their existing workflows, directly within the tools they already use.

A true AI-human interface must be non-disruptive and context-rich, surfacing relevant data, documents, and decisions exactly where work happens. It should enable smooth collaboration between AI and humans, so when human judgment is needed, all necessary information is already visible and accessible. Most importantly, it should reduce rather than increase cognitive load. This is not about adding another interface; it's about making AI part of the interface people already trust.

Safe Autonomy through Decision Analytics

No enterprise flips a switch and lets AI autonomously process revenue for a multi-billion-dollar operation.

The transition from human-led to AI-led decisions must be measured and closely monitored. This requires decision analytics — continuously comparing AI decisions with human decisions to ensure reliability, accuracy, and accountability.

Through this visibility, organizations can calibrate trust, manage calculated risk, and enable a safe, stepwise transition to autonomy in their commercial operations. Monitoring provides the guardrails that ensure the AI is doing the right thing at the right time, creating a foundation of confidence before scaling autonomy.

Cross-Platform Execution Fabric

AI doesn't operate in a vacuum. The actions it takes — generating quotes, processing orders, approving terms — must happen within the enterprise technology stack, across systems like ERP, CRM, procurement, and data platforms. This requires cross-platform intelligence that can execute workflows across multiple systems in a transparent and auditable way.

Humans must be able to trace what was done, where, and why, ensuring every AI action is anchored to enterprise reality. Integrations transform AI from an isolated assistant into a true operational participant in the enterprise ecosystem.

A great engine of intelligence is only the starting point. The real breakthrough comes when intelligence is embedded, monitored, and integrated into the enterprise — creating a machine that safely and intelligently moves commerce forward. This is how AI becomes not just smart, but truly enterprise-ready.



IV – Be exception-hungry

One of the most common pitfalls in AI implementation is **demo-optimisation**—tuning a model or system to perform perfectly on a handful of curated examples to prove a case to stakeholders. In practice, this creates fragile systems that fail under scale.

An execution fabric designed to operate in the real world must be **exception-hungry**—that is, it must be fundamentally built to seek out, ingest, and learn from the unexpected. The real world is not a controlled environment; it's messy, adversarial, and filled with rare, unpredictable events that can derail systems trained only on narrow or sanitized datasets. Building models or infrastructures that only perform well on "known" scenarios creates the illusion of robustness, while in reality such systems are brittle and prone to failure when encountering out-of-distribution data. A real-world-ready execution fabric must therefore embrace exceptions as its most valuable learning opportunities rather than as nuisances to be ignored or filtered out.

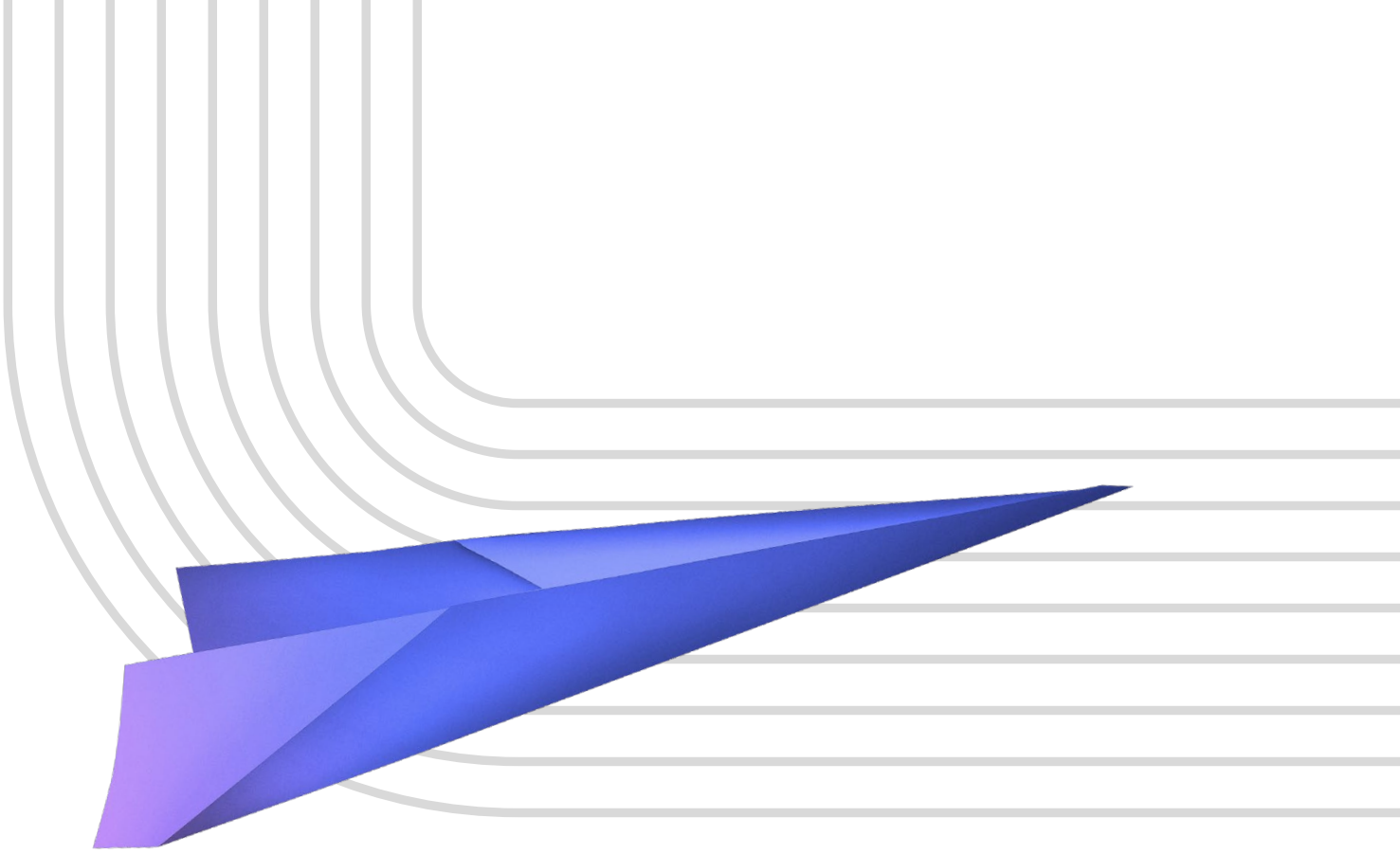
However, for a single organization, acquiring the kind of comprehensive and diverse data required to achieve this level of resilience is prohibitively expensive and often impossible. A company's internal data, no matter how large, represents only a small fraction of the real-world variation that exists across industries, geographies, and use cases. Furthermore, the data available internally is typically biased toward what the organization has already seen—it lacks representation of the rare or unseen scenarios that define true operational complexity. Without exposure to external data or collaborative learning ecosystems, such systems are condemned to repeat the blind spots of their creators.

Even when internal data is abundant, it rarely comes labeled with explicit "exception" tags or information on how to resolve those edge cases. The process of labeling exceptions—understanding what went wrong, why it happened, and how to correct it—is itself labor-intensive and often requires human judgment, which does not scale easily. This creates a paradox: the more autonomous you want your system to be, the more you need rich, diverse, and exception-labeled data—but the harder that data becomes to obtain or annotate within the confines of a single enterprise.

A truly capable, autonomous execution fabric must transcend organizational boundaries and operate on a global scale of data aggregation. Only by learning from billions of real-world interactions and edge cases—those encountered by many actors across the ecosystem—can it develop the adaptive intelligence necessary to handle novel situations with minimal human oversight. In other words, exception-hungry systems are not just technically superior; they are economically inevitable, as they reduce the endless cycle of retraining and reinvestment that plagues isolated models. By continuously integrating and learning from the world's collective edge cases, such a fabric becomes not just robust, but alive to the evolving complexity of reality.

V – Do not forget continuous improvement





True scalability requires rigorous data science discipline: extensive testing on representative datasets, continuous monitoring for data drift, and retraining mechanisms that maintain performance as conditions evolve. Without these foundations, systems lose reliability—and with it, user trust.

Scaling is not about showing what the model can do once; it is about proving what it can do continuously. An autonomous system cannot be delivered as a one-time implementation. It must be **designed as a continuous learning program**. Over time, customer formats change, markets evolve, and tacit knowledge shifts as employees gain or lose expertise.

The system must therefore be capable of learning from these changes—understanding new data distributions, adapting to new layouts, and capturing evolving operational logic. This is achieved through layered learning: foundation-level pretraining, global algorithmic refinement, and enterprise-specific adaptation.

> Without a long-term learning framework, autonomy quickly decays into automation.

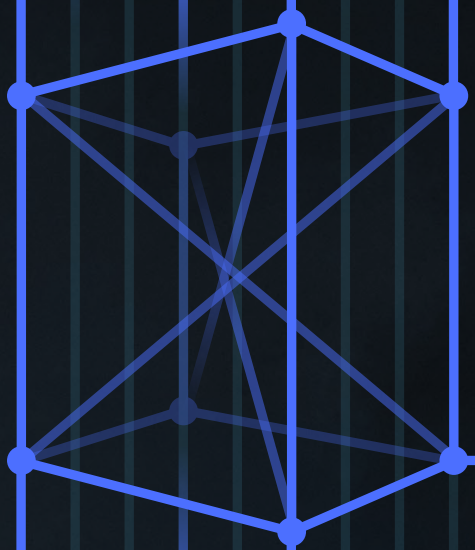


Conclusion

Autonomous Commerce marks a fundamental evolution in how enterprises operate. Success depends on building adaptive, domain-specific systems that integrate capturing, reasoning, and execution across every stage of a transaction. It requires rigorous engineering, not just innovation theatre. Those who treat autonomy as a living program — continuously learning, refining, and orchestrating multiple AI paradigms — will define the next era of commercial performance. Those who don't will remain trapped in partial automation, where potential is showcased but never scaled.



go autonomous



Accelerating Revenue Flow,
Decoupling Cost