From Systems of Record to a System of Work

How Digital Workers Eliminate Interoperability Waste

Executive Summary

Most business "misses"—missed forecasts, missed revenue targets, missed customer commitments—are not data, analytic or decision-making problems. They are **interoperability problems**. In business terms, **interoperability** means the ability of different functions, systems, or partners to understand one another's decisions in real time and coordinate their actions. Without it, sales, marketing, finance, and supply chain each make decisions in isolation; plans are never tested against real constraints; and last-minute changes in one area create expensive surprises in another. The result is more meetings, more firefighting, and outcomes that consistently fall short.

In the pre-artificial intelligence business environment the model that was used to correct business "misses" was simple:

data problems => *cause* => **analytic errors** => *cause* => **poor decisions**.

The proliferation of data producing devices, the ease of access to data storage technologies, and rise of data science tools and algorithms certainly has created data quality issues. The time and money spent fixing data errors, making flawed decisions on wrong data, and executing strategies that miss the mark is huge. An IBM study estimated that in the U.S. alone, businesses lose roughly **$3.1 trillion annually** due to poor data and its downstream decision-making consequences. Studies reported in *MIT Sloan Management Review* show that poor data quality leads to costs equal to about **15–25% of revenue** for most companies**.**

Poor decision making caused by a reliance on gut feeling and susceptibility to bias is also a problem. Research shows that when a leader is 90% confident in a decision, they are often wrong **30-40% of the time**. Biases like **confirmation bias** (seeking out information that proves we're right) and **overconfidence** systematically distort choices.

The result? Costly missteps. According to a 2024 **McKinsey** study, poor decision-making practices erode about **20% of the annual revenue** of an average business. In other words, one-fifth of potential revenue is lost due to strategic errors, delayed responses, and failure to adapt – a huge efficiency penalty that affects companies globally. Similarly, market research firm IDC found that companies may lose **20–30% of their revenue** each year to **operational inefficiencies** and process issues which are often symptomatic of bad decisions and suboptimal management. The frequency of poor decisions is itself a concern. In one **McKinsey survey**, **72% of executives** admitted that bad decisions were just as common as good decisions in their organizations. Research by **Gartner** found that poor operational and financial

decisions by mid-level managers commonly **erode over 3% of an enterprise's profits** on average.

Executives don't always make the right decision. A long line of management research finds a consistent gap between executives' pro-data rhetoric and what actually drives decisions. Pfeffer and Sutton's work on evidence-based management shows that managers commonly default to obsolete knowledge, intuition, and tradition rather than rigorous evidence, even as they talk about "data-driven" practices. Across 356 major corporate decisions, Nutt found about half failed, often because leaders imposed preferred solutions and curtailed the search for alternatives—instead of gathering and weighing data. Scholarship indicates that executives routinely lean on gut judgment (sometimes useful, often biased), underscoring that intuition is structurally embedded in organizational decision processes. Finally, research shows organizations frequently "decouple" what they do from what they publicly profess they do—suggesting analytics programs can be more symbolic than substantive.

Nobody actually uses software systems correctly, anyway. So, data is dirty and executives make poor decisions. There is a third element that is most overlooked and that is the relationship between business worker and software system. This is where things really break down. No matter how hard enterprises try it is always 'garbage in'. In a study of CRM stakeholders, **68%** reported *incomplete* data, **65%** *missing* data, and **24%** said **less than half** of their CRM data is accurate **and complete**. Data entry errors, regardless of environment, occur **5%** of the time. The average customer contact database contains **25%+ duplicates**, compounding "I'll fill it in later" behavior. Human workers are overloaded: Salesforce's research highlights that sellers spend the majority of time on non-selling/admin work—one reason updates get skipped. Beyond non-entry, **data decays** even when entered (role changes, bounces, etc.). Benchmarks put B2B contact decay around **~20–30% per year** (≈2%/month), so yesterday's "complete" record degrades fast.

In day-to-day business tools, error and inconsistency are pervasive. Field audits and lab studies of operational spreadsheets—still the "shadow IT" behind many forecasts and reconciliations—find error rates per cell in the **0.4%–6.9%** range, with auditors reporting they had "never seen an error-free spreadsheet." In clinical repositories where data are manually keyed, researchers documented non-trivial field-level error rates and showed that even low-prevalence errors distort downstream analysis. And for judgment calls (pricing exceptions, risk reviews, lead qualification, claims), **noise**—unwanted variability between equally qualified reviewers—has been measured and popularized by Kahneman and colleagues; it is distinct from bias and

routinely large enough to affect outcomes. In short: if you assume human and data perfection, you are planning for a world you don't live in.

So what if we just get our analysis better! Most companies try to squeeze a few more points of accuracy out of their models, then wonder why nothing materially changes. In practice, small lifts rarely move the needle unless they flip real decisions at scale. Enterprises seldom evaluate models by the *value of decisions* (net benefit), but focus instead on accuracy. "Better predictions" may not actually be worth it if the benefit, including the eventual interpretation and decision by a biased executive, is not there.

The larger—and more fixable—problem in most enterprises isn't the model; it's *systems*: dozens of systems (CRM, ERP, WMS, ad platforms, payment gateways, field apps) that only partially talk to each other, run on different cadences, and are updated by fallible humans. In that reality, a single, centrally-orchestrated brain hits hard limits: as soon as you try to plan jointly across many teams and tools with incomplete views, the planning problem becomes intractable. It is the worst case. It is why systems integration projects rarely achieve their stated ROI[1]. These projects work to build a centralized data store powering a centralized insight center when the business of day to day occurs on the periphery. That's not hand-waving; it's a theorem in decision science: decentralized decision-making with partial information explodes in complexity, which is why centralized "optimal" control fails outside simplified laboratory environments.

The problem then is a **systems issue**, not a spreadsheet issue: the business culture, technological infrastructure and operations do not operate as a connected system. We spend heavily to "fix the data." First, attempting to perfect better **data in**—new fields, integrations, MDM, consultants, adoption campaigns. Then, spend to perfect **better data out**—data lakes, BI rebuilds, forecasting models and scenario tools. The spending on managing data is astronomical; the results have only been disappointing. Despite the spend, the weekly experience doesn't change: people still chase status, argue the numbers, and run the business in slides and side spreadsheets.

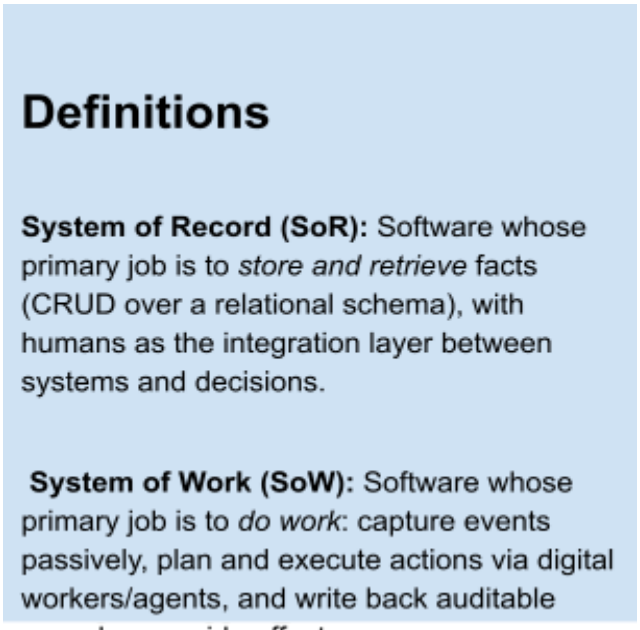
---

[1] **McKinsey & Oxford University Study:** A famous 2012 study in collaboration with the University of Oxford analyzed 5,400 IT projects. They found that:

- On average, large IT projects run 45% over budget and 7% over time, while delivering 56% less value than predicted.

From records to results. The problem isn't only that fields are wrong; it's that most enterprise software was designed as a ***system of record***: a place to type facts, store them in a relational table, and run reports later. That design assumed humans would keep the ledger current and that someone else would interpret the numbers and decide what to do. In reality, humans avoid forms, facts drift, and the gap between "what the system knows" and "what the business must do next" widens.

The database legacy in the era of artificial intelligence. Relational systems (ACID databases, forms, batch reports) won the client–server era because they guaranteed integrity and auditability. We wrapped workflows around tables: triggers, stored procedures, then low-code flows. That stack is excellent at keeping ledgers, but brittle at *running* the business in real time, especially when (a) the world is event-driven, (b) people are overloaded, and (c) many semi-independent teams must act with partial information.



What's missing is a system of work. Systems of record concentrate on *inputs* (forms, fields) and *storage* (tables). But the enterprise problem is many teams, tools, and cadences with partial views: *interoperability*. That's why you see duplicates, re-keying, status chases, and forecasts no one acts on. The software is optimized for *record-keeping*, while the organization needs *decision-making* and *execution*.

A **system of work** doesn't ask people to feed it; it *captures* activity automatically, *coordinates* actions across teams, and treats the database as a by-product with receipts (provenance) rather than the product. It turns evidence into decisions and decisions into executed steps—so the organization moves in sync even when data is imperfect. It actively 'works' rather than passively stores.

With noisy human judgment and incomplete or error-prone enterprise data, a ***system of work*** with a fabric of **digital workers** (the business-friendly term for multi-agent systems) is often a better match than a single centralized model. The reasons are (1) the world you operate in is decentralized and partially observable, (2) local, redundant decision-making cancels a surprising amount of error, and (3) there's theory and evidence that simple coordination rules among many small

decision-makers can yield stable, high-quality, more accurate insight and system behavior—even when individual inputs are messy.

In a **system of work, digital workers** operate as small, autonomous services that each see a local slice of the world (one warehouse, route, customer cohort, feeder line, or portfolio), act on it, and coordinate with neighbors through lightweight protocols. Artificial intelligence research informs us that this kind of local-rules, message-passing architecture can still produce orderly, enterprise-level behavior—and do so robustly—even when links fail, messages are delayed, novel events occur, and the business structure changes (think org changes or vendor outages).

Why does a fabric of digital workers perform better than a central brain? First, centralized optimal planning across many teams and systems with partial information is intractable in the worst case; even small decentralized decisions blow up in complexity. That's a mathematical way of saying "one brain won't scale to your whole company's uncertainty." Second, there's a large control-theory literature showing that when many small decision-makers use **simple consensus/coordination rules**, the combination of those decisions still converges—despite delays, link failures, and changing org structures. That gives you global order from local, intermittent signals rather than brittle, attempts at coordination and optimization of the entire state of the enterprise. Third, **centralized training + decentralized execution** (a standard way to train digital workers) lets you learn coordination policies with richer, historical views, what some call "institutional memory" and the system's operational process and constraints, then run them locally in production with limited messaging—matching real enterprise functions.

Why is this architecture resilient in error-filled environments?

- **Locality contains bad data.** Each digital worker acts on the slice it sees best (one warehouse, feeder line, route, region, portfolio), so a corrupted field or stale assumption in one system doesn't contaminate the whole enterprise plan before peers can correct. Convergence results for consensus with switching topologies/time-delays formalize this containment and eventual agreement.

- **Redundancy cancels mistakes.** Combining several imperfect decision-makers beats relying on one: from the **Condorcet jury theorem** (majority of >50%-competent voters approaches correctness as the group grows) to **ensemble methods** like bagging, which reduce variance and error by aggregating diverse models. Digital workers are a system-level analogue of these results.

- **Reliability from unreliable parts.** The classic result by **von Neumann** shows you can build reliable computation by arranging noisy components with voting/restoration. Modern distributed systems (e.g., **Paxos/Raft** consensus) operationalize the same idea for state and logs under failures—exactly the conditions enterprise platforms see.

- **Graceful degradation beats brittle optimality.** In fast-changing settings, the goal isn't "perfect prediction" but **robust, near-optimal actions** that can be corrected quickly when reality disagrees—precisely what local, coordinated workers deliver.

- **Institutional memory builds collective intelligence.** As digital workers act repeatedly in their domains, they accumulate and refine localized knowledge—patterns of demand in a region, recurring failure modes in a process, or negotiation outcomes with a vendor. When this knowledge is shared across the network, it forms an *institutional memory*: a durable layer of context that survives individual turnover and fragmented records. Over time, this collective memory helps the organization make faster, more consistent decisions, reduces the reinvention of solutions, and improves resilience by ensuring that lessons learned in one corner of the enterprise are not lost but embedded in the system as a whole.

What this means for business problems. Lead-to-cash, inventory-to-delivery, outage response, new product introductions, and network-wide pricing are all **system-of-systems** problems plagued by missing fields, stale records, judgment noise, and asynchronous updates.

System of systems is an end-to-end capability—like getting revenue, fulfilling orders, or restoring service—that depends on **multiple, independently owned systems and teams** working on **different cadences, different incentives and KPIs** with **different data definitions** to deliver one outcome. No single platform has full authority or complete, up-to-date information; results emerge from how these parts **coordinate decisions**, not from any one system's accuracy. This is more than "systems integration"—it's about synchronizing actions across CRM, ERP, WMS, ad platforms, finance, field service, and partners under real-world time pressure.

You see system-of-systems dynamics in everyday work: **lead-to-cash** (marketing→CRM→CPQ→billing→revenue), **inventory-to-delivery** (demand plan→ERP→WMS/TMS→3PLs→retailer portals), **omnichannel promotion and ad spend tied to in-stock**, **network-wide pricing** across marketplaces, **outage response** (monitoring→ticketing→workforce→vendor dispatch), **claims-to-payout** in insurance, **patient flow** in hospitals (ED→beds→OR→discharge), **airline disruption recovery**

(fleet, crew, gates), **bank onboarding/KYC**, and **M&A Day-1 operations**. Common symptoms: local fixes create distant side effects, handoffs stall because clocks don't match, "truth" varies by system, and small data gaps snowball into missed SLAs, stockouts, or revenue leakage.

Why system-of-systems matters? If you treat these as single-system or pure "data" problems, you'll chase diminishing returns. Treat them as **system-of-systems** problems and focus on **coordinating decisions across boundaries**—using lightweight automation where each domain acts on its local view, shares just enough context, and learns from outcomes—so the whole network moves in step despite imperfect data and constant change. A digital-worker approach puts small decision services at the seams (CRM↔ERP↔WMS↔support), has them act locally with clear decision rights, and align via simple messages (quotas, capacities, prices, risk scores). The net effect is **less propagation of local mistakes**, faster correction when reality contradicts a plan, and measurable, system-level performance—without pretending you'll ever get perfect data or a single model that "knows everything." (As the **No-Free-Lunch** results remind us, there is no universally best algorithm; tailoring structure to the problem is where the edge comes from.)

Bottom line: if your world is full of human error, inconsistent inputs, and shifting conditions, betting on ever-more-accurate single models is a diminishing-returns strategy. A network of coordinating **digital workers** is built to *absorb* noise: it localizes decisions, aggregates away individual mistakes, and still reaches reliable enterprise-level behavior—with the math, the systems theory, and real-world results to back it.

A system of work embeds **digital workers** at the boundaries of real processes—inside the inbox, on the call, in the warehouse, alongside the ad platform, in the order flow. They:

- **Capture** events passively (no extra typing).
- **Plan & act** using approved playbooks (e.g., prioritize tickets, rebalance inventory, sequence outreach, pause spend).
- **Coordinate** with peers through compact signals (capacity, risk, price), not brittle global state.
- **Record** what happened with provenance (citations to the underlying emails, calls, orders, usage).

- **Remember** what occurred before and how enterprise learning should inform present decisions.

Local views + lightweight consensus let many small decisions converge on coherent enterprise behavior—even with delays, outages, or schema drift. Instead of one fragile, centralized optimizer, you get a mesh that is robust, auditable, and scalable.

The database is still there—its role changes. We don't throw out relational databases; we demote them. The event log is primary, agents compute claims (fields) from evidence, and the relational view is a materialization for compatibility and reporting. Records stay consistent because they're produced from receipts, not from memory.

"So what?" This local-first design also mirrors how effective organizations are built. Herbert Simon called it *near-decomposability*: large systems work because they're composed of loosely coupled subsystems.

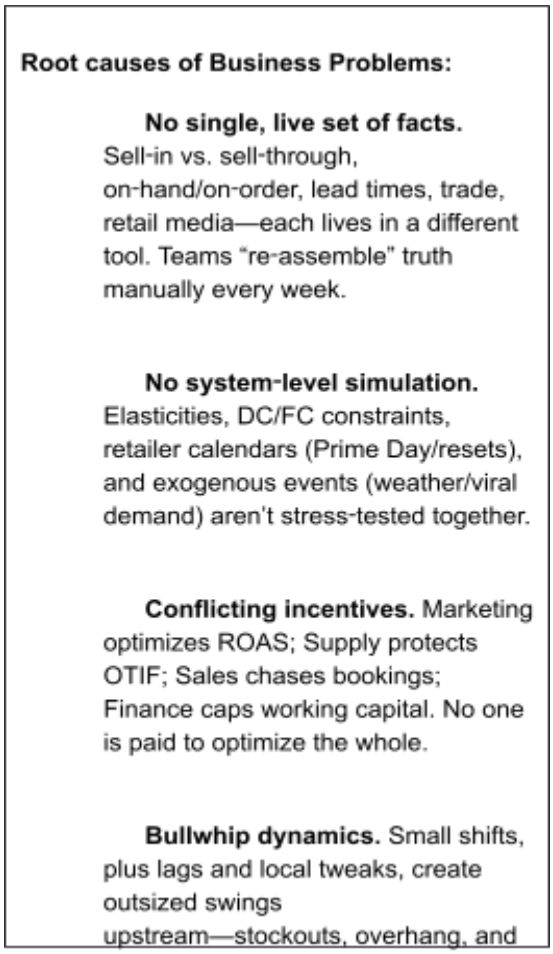
- **Less interoperability friction:** no re-keying, fewer status chases; work captures itself.
- **Better institutional memory:** shared guardrails, elasticities, and lead times are part of the fabric, not trapped in decks.
- **Actionable forecasts:** predictions are tied to *plays* ("protect these SKUs; pause these flights; escalate these accounts") with owners and approvals.

Digital workers map directly to those boundaries—regions, channels, product lines—so they interoperate naturally with existing teams and platforms rather than fighting them. Instead of one brittle pipeline, you get many coordinated "doers" that fail gracefully and scale horizontally as the business grows.

There's growing evidence that this approach pays off in messy, multi-party environments. In urban traffic networks (a classic system-of-systems), networked digital workers controlling individual intersections and sharing only local signals have cut average travel times materially on city-scale networks (e.g., **~422s → ~297s** on a 196-intersection Manhattan grid, beating strong optimization baselines). The point isn't traffic—it's the pattern: local context + minimal communication + coordination logic beats monolithic control when reality is dynamic. In power and energy, recent surveys show distributed "digital worker" architectures improving demand response, microgrid scheduling, and resilience—precisely because they tolerate partial

observability and faults while scaling across many assets. In supply chains and logistics, reviews report that multi-party reinforcement learning (trained centrally, executed locally) is competitive on inventory control, routing, and pricing in settings where information is fragmented across partners and systems.

The training pattern behind many of these wins is also business-friendly: **centralized training, decentralized execution**. You can learn coordination policies in a lab or simulation where more data is available, then deploy digital workers that run independently in production with limited messaging—so you get enterprise-level coordination without a fragile command center.



What this means for the business agenda is straightforward:

- **Solve interoperability with autonomy + coordination, not bigger models.** Put digital workers at the boundary of key systems and teams; give them narrow, auditable responsibilities (e.g., prioritize tickets, set micro-prices, rebalance stock, route a truck) and a way to align with peers on shared constraints.

- **Expect robustness and scalability out of the box.** Because they operate on local views and only exchange compact signals, digital workers degrade gracefully under outages, delays, or schema drift—unlike centralized controllers.

- **Measure success by decision value.** Instrument policies with decision-curve analysis or similar net-benefit metrics so leadership sees when coordination *changes actions* (and economics), not just dashboards moving a few basis points.

Instead of one monolithic algorithm struggling to find a solution, a swarm of digital workers, dozens, hundreds, or even thousands of simple, coordinated artificial enabled workers. Each operates on a limited set of rules, but through their collective

action, they achieve emergent intelligence. This is a proven force multiplier. Forget the lone genius. When it comes to solving complex, multi-faceted problems, the future is about managing teams of digital workers; and they will, experience tells us, consistently outperform even the most brilliant human mind. Why? Because complexity thrives on scale, speed, and the elimination of ego. A single brain, no matter how capable, is a bottleneck of bias and biological limitation. .

Conclusion: A system of record is not a system of work. You can record perfect history and still miss the quarter.

Ask of any "AI" initiative:

Does the system capture activity without asking people to type more?

Can it change what teams actually do next week, with audit trails?

If a link fails or a schema changes, does it degrade gracefully—or stop the business?

If the answer is "no," you're funding another system of record with better charts.

The **Intellex system**—an intelligence operating system for your business—does exactly that. It manages a small team of **autonomous, AI-enabled digital workers** with a **shared memory**. They watch key events (a PO change, a stockout risk, a campaign going live), propose the next best step, and carry out the routine parts under your approvals. As they work, the "record" updates itself with receipts. Less time entering data. Less debating the output. More time doing the right thing.

Intellex, as a **system of work, fixes** the system, not just the cells. It **captures activity automatically** (email, calls, orders, usage, ad flights), **stores shared memory** (guardrails, elasticities, lead times), and **coordinates actions** with agents that run approved plays across teams. The record updates itself as a by-product, with receipts you can click into. Now "data in" is effortless, and "data out" is a set of **recommended actions with evidence**, not a number to argue about.

For the business leader, this shows up in simple ways: fewer status meetings, fewer surprises at month-end, fewer fire drills before a retailer window. Forecasts are still imperfect—but they are finally **useful** because they're tied to what changes if you act: which SKUs to protect, which flights to pause, which POs to pull forward, and who needs to approve. That is the practical difference between a system that **records** and a system that **works**.

Intellex's thesis: In the AI era, the tools, data, and ways of working your business relies on must evolve. Traditional **systems of record** excel at storing what happened after the fact, but they do little to help functions act together in the moment. Most companies feel the gap between what actually happens with customers and what shows up in their CRM or ERP. Record-centric tools ask people to stop doing the

work so they can **report** the work—fill fields, convert leads, update stages. That pause is costly, producing stale, incomplete, or optimistic data. Forecasts wobble, the pipeline looks healthier than it is, and churn risks go unnoticed until it's too late. Finance and RevOps build compensating spreadsheets and QA steps, multiplying effort and creating even more drift between reality and the record.

Welcome to Intellex

Intellex runs your business as a **system of work**. It watches what's happening, tests your options against real-world limits, coordinates who does what next, and keeps the receipts for every decision—without asking people to retype the work they already did.

Picture Monday morning. In one workspace, everyone sees the same live facts: what sold, what's on hand and on order, which campaigns are scheduled, which lanes are slipping. Intellex has already noticed that a tent-pole campaign starts next week while a couple of DCs are tight. It lays out three ways forward—shift inventory, reshape the flight, or pull forward a PO—and shows the impact of each on revenue, margin, on-shelf availability, and OTIF. The owner chooses. Approvals happen where they should. Intellex carries out the routine steps and links the result back to the evidence so the team can learn from it next time.

That's the product: a piece of software that **does the coordination** work humans currently do in spreadsheets, email threads, and meetings—faster, with fewer misses, and with an audit trail you can trust.

How Intellex works

Intellex starts by **capturing the facts automatically**—sell-in and sell-through, on-hand and on-order, lead times, EDI/portal events, and the calendars for retail media and trade. It remembers the operating rules you live by: ship windows, DC/FC capacity, planograms and resets, how a given retailer behaves around events like Prime Day, and the lifts and cannibalization you've seen before.

With those facts and that memory, it **tests options before you spend or ship**. If marketing wants to move a flight, Intellex checks inventory and capacity. If supply wants to rebalance stock, Intellex checks promo timing and rank protection. You see the trade-offs side by side, scored on the metrics operators manage every day.

When a direction is picked, Intellex **turns the decision into owned actions**. It drafts the concrete moves—adjust this allocation, pace that campaign, stage that NPI—routes them to the right people, and respects your approval tiers. Low-risk, reversible changes can run automatically; bigger moves wait for a click. Every step is logged with the evidence behind it.

Finally, Intellex **closes the loop**. Expected versus actual is captured without slide archaeology, and what you learn feeds the next planning cycle. Over time, the plan stops drifting because strategy, operating choices, and daily implementation live in one view.

Intellex captures what is happening automatically, tests options against real constraints, coordinates who does what next, and records the evidence behind every decision.

How you adopt it

Intellex is delivered as SaaS and sits **on top of what you already use**. We connect in read-only to your ERP/OMS/WMS, retailer portals/EDI, retail-media and trade systems, and (where permitted) email/calendar and your data warehouse. You pick a narrow starting point—one category, one retailer window, one problem worth solving—and Intellex runs in **shadow mode**. It proposes actions with receipts; your team tunes guardrails; then you turn on approvals and let it handle the routine steps. From there, you widen the scope at your pace.

Governance mirrors how you already work. Ownership is explicit, approvals are tiered (green/amber/red), and everything has a receipt. Nothing hides in a model: leaders can drill from any number to the signals underneath—what moved, what arrived, what ran, and when.

- Cross-functional coordination for demand/supply, retail media and trade, product launches/resets, and deduction/OTIF prevention.

- Overlay on existing systems (ERP/OMS/WMS, retailer portals/EDI, retail media platforms, data warehouse, CRM/email).

- Human-in-the-loop execution with approvals and full auditability.

Day to day, the experience is simple. Operators review what changed and why, accept or adjust the proposed moves, and keep moving. Finance sees revenue, margin, and working-capital effects as decisions are made, not after the fact. Leadership spends more time on priorities and less on reconciliation.

Intellex isn't a rip-and-replace platform, a new place to type data, or a black box. It's a practical way to make the business behave like a connected system: **work captures itself, options are stress-tested, decisions have owners, actions leave receipts, and the next cycle starts smarter**.

Primary users & roles

- **Operators** (Sales/Category, Trade/Marketing, Supply/Logistics) — review recommendations, approve actions, monitor outcomes.

- **Finance** — view impacts on revenue, margin, working capital; approve guardrails.

- **Administrators/RevOps** — manage connections, rules, thresholds, and approvals.

- **Software coworkers (digital workforce)** — automated roles executed by Intellex (see below).

Core capabilities

1. **Unified Live Facts**
   Auto-captures and normalizes: sell-in/sell-through, on-hand/on-order, lead times, EDI/ASN events, planograms/resets, retail-media and trade calendars, and key customer signals (email/call artifacts where permitted). Provenance is attached to each fact.

2. **Shared Operational Memory**
   Stores constraints, elasticities, event effects (e.g., major retailer events), DC/FC capacities, ship windows, and policy guardrails. Versioned and permissioned.

3. **Scenario Engine (constraint-aware)**
   Runs side-by-side what-ifs for campaign timing/depth, allocation, pricing/promo, and NPI staging by retailer/SKU. Compares outcomes on

revenue, margin, on-shelf availability (OSA), and OTIF.

4. **Decision Orchestration**
   Turns chosen scenarios into concrete actions with owners and due dates (e.g., PO pull-forward, DC/FC reallocation, campaign pacing/mix, promo guardrails). Routes through green/amber/red approval tiers; tracks status to completion.

5. **Execution Connectors**
   Issues approved changes via integrations or task instructions:
   - ERP/OMS/WMS (orders, allocations, ship windows)
   - Retail media/trade platforms (flight pacing, mix, pausing)
   - Retailer portals/EDI (acknowledgments, shipment updates)
   - Collaboration (Slack/email tasks)

6. **Evidence & Audit**
   Every recommendation and decision links to underlying events and assumptions. Expected vs. actual is captured automatically and feeds the next planning cycle.

How Intellex Works

At its core, Intellex operates as an orchestration layer that turns fragmented signals into coordinated action. It doesn't replace existing systems—instead, it overlays them, pulling in data, interpreting it against shared context, and pushing back only the changes or tasks that matter. The engine follows a deliberate cycle of observing, understanding, planning, executing, and learning—ensuring the organization is always adapting based on evidence.

Operating model

The Intellex operating model is built around a repeatable loop that mirrors how effective teams already work, but with automation and discipline applied. Data and events are continuously observed, placed into context, and translated into concrete plans. Actions are carried out through existing systems and workflows, while outcomes are measured to refine future cycles. This rhythm keeps teams focused on decisions and outcomes, rather than chasing data or reconciling systems.

Observe → Understand → Plan → Execute → Verify/Learn

Every decision cycle in Intellex begins with observation and ends with learning. Live signals from multiple systems are captured on a schedule, interpreted against

business constraints and historical memory, and turned into forward-looking scenarios. The best scenarios are executed through approvals and connectors, then validated against actual outcomes. This cycle ensures that each iteration of planning is smarter than the last.

- **Observe**: Ingests live signals on a schedule (near-real-time where available).
- **Understand**: Applies shared memory (constraints, elasticities, event effects).
- **Plan**: Generates and compares what-ifs; highlights trade-offs.
- **Execute**: Creates tasks or pushes approved changes through connectors with approvals.
- **Verify/Learn**: Captures expected vs. actual; updates memory and future plans.

Interfaces & integration

Intellex is designed to slot into the tools businesses already use, not to replace them. It connects with ERP, OMS, WMS, retail portals, marketing platforms, and CRM systems—usually in a read-only mode for safety—and pushes approved actions back into those same systems. Users interact through a web workspace, with optional Slack and email integrations for ease of use. Administration is straightforward, with clear mappings, thresholds, guardrails, and roles to maintain governance.

- **Inbound connections** (read-only by default): ERP/OMS/WMS, retailer portals/EDI, retail-media and trade systems, CRM/email/calendars (where permitted), data warehouse/lake.
- **Outbound actions** (after approval): order adjustments, allocation updates, campaign pacing/mix changes, calendar/plan updates, and task assignments.
- **User surfaces**: Web workspace; optional Slack commands/notifications; email summaries.
- **Administration**: Connections, entity mappings (retailers/SKUs/DCs), thresholds, guardrails, roles, and approval tiers.

Controls & governance

Because orchestration without control is risky, Intellex is built with governance at its core. Approvals follow a tiered model, with automatic, owner, or management checkpoints depending on risk. Permissions are strictly role-based, ensuring the right people see and do the right things. Every decision and action is logged

immutably, with evidence linked for audit and compliance. This makes the system transparent and trustworthy. Key functions are:

- **Approvals**: Green (auto), Amber (owner approval), Red (management approval).
- **Permissions**: Role-based access to data, actions, and approvals.
- **Audit**: Immutable decision and action log with links to evidence; exportable.

Deployment & operation

Intellex is delivered as a SaaS platform, with onboarding focused on quickly mapping entities and connecting systems for a chosen domain (like one category × one retailer). Once live, data refreshes follow the cadence of source systems, while planning runs on a weekly rhythm and event alerts trigger in real time where possible. Extensibility through APIs and adapters means the platform can grow with the business without heavy customization.

- **Delivery**: SaaS.
- **Onboarding**: Map entities and connect systems for chosen domain (e.g., one category × one retailer).
- **Cadence**: Data refresh aligned to source systems; weekly operating rhythm for plans; real-time alerts for events where available.
- **Extensibility**: APIs/webhooks for custom connections; action adapters for additional systems.

What is included

Out of the box, Intellex provides everything needed to run coordinated decision cycles. Business users and admins get access to the workspace, a library of digital workers to automate recurring tasks, a scenario engine for testing trade-offs, and connectors to common enterprise and retail systems. Audit and export tools are also included, ensuring visibility and compliance from day one.

- Access for business users and admins to the workspace.
- A series of connectors into event producing systems.
- Library of digital workforce.
- Scenario engine and decision orchestration.
- Connectors for common ERP/OMS/WMS, retailer portals/EDI, and retail-media platforms.

- Audit/export utilities.

What Intellex is not

Just as important as what Intellex is, is what it is not. It does not replace ERP, OMS, WMS, or data warehouses—those systems remain the systems of record. It does not ask users to enter data manually, nor is it a custom system-integration project. Instead, Intellex sits above the existing stack, coordinating decisions and execution across systems with lightweight integration and strong approvals.

- Not a replacement for ERP/OMS/WMS or a data warehouse.
- Not a manual data entry system.
- Not a custom SI project; Intellex operates as an overlay and coordinates execution with approvals.

How to start with Intellex

Pick one category and one retail window. In 30 days we will connect, simulate, approve, and run one coordinated plan—so you see fewer stockouts, less waste, and lower costs in the next quarter. If you want, tell me the 2–3 workflows where interoperability hurts you most (e.g., lead-to-cash handoffs, inventory-to-delivery, outage response). I can sketch a digital-worker blueprint for each: placement, decision rights, coordination signals, and the right success metrics to prove business value in quarter-scale pilots.

The solution isn't more data or more meetings. It's about creating **interoperability**—breaking down silos to ensure every function operates from a single source of truth. This allows leaders to replace gut calls with coordinated, data-driven actions that are tested against real-world constraints, turning missed targets into delivered results.

Citations

- Bloch, M., Blumberg, S., & Laartz, J. (2012). Delivering large-scale IT projects on time, on budget, and on value. McKinsey & Company.
- Bromley, P., & Powell, W. W. (2012). From smoke and mirrors to walking the talk: Decoupling in the contemporary world. *The Academy of Management Annals, 6*(1), 483–530.
- Brynjolfsson, E. (1993). The productivity paradox of information technology. *Communications of the ACM, 36*(12), 66–77. https://doi.org/10.1145/163298.163309
- Business Insider. (n.d.). Report on unproductive meetings costing $37B (U.S.).

- Dane, E., & Pratt, M. G. (2007). Exploring intuition and its role in managerial decision making. *Academy of Management Review, 32*(1), 33–54.
- Dehning, B., & Richardson, V. J. (2002). Returns on investments in information technology: A research synthesis. *Journal of Information Systems, 16*(1), 7–30. https://doi.org/10.2308/jis.2002.16.1.7
- DeLone, W. H., & McLean, E. R. (1992). Information systems success: The quest for the dependent variable. *Information Systems Research, 3*(1), 60–95. https://doi.org/10.1287/isre.3.1.60
- Gallup. (2021). *State of the Global Workplace 2021*.
- Gartner, Inc. (2018). Study on operational decision impact to profit.
- Gartner, Inc. (2023). *Predicts 2024: Data and analytics strategies build trust and accelerate value*.
- Grammarly & The Harris Poll. (2024). *State of Business Communication 2024*.
- IDC (International Data Corporation). (n.d.). Finding on revenue lost to inefficiencies.
- Keil, M., & Montealegre, R. (2000). Cutting your losses: Extricating your organization when a big project goes awry. *Sloan Management Review, 41*(3), 55–68.
- Markus, M. L., & Tanis, C. (2000). The enterprise systems experience—From adoption to success. In R. W. Zmud (Ed.), *Framing the domains of IT research: Glimpsing the future through the past* (pp. 173–207). Pinnaflex.
- McKinsey & Company. (2024). Research on decision-making effectiveness.
- McKinsey Quarterly. (n.d.). Surveys on strategic decision quality.
- Meyer, J. W., & Rowan, B. (1977). Institutionalized organizations: Formal structure as myth and ceremony. *American Journal of Sociology, 83*(2), 340–363.
- MIT Sloan Management Review. (n.d.). Analysis of bad data impacting 15–25% of revenue.
- Nutt, P. C. (1999). Surprising but true: Half the decisions in organizations fail. *Academy of Management Executive, 13*(4), 75–90.
- Nutt, P. C. (2002). *Why decisions fail: Avoiding the blunders and traps that lead to debacles*. Berrett-Koehler.
- Panorama Consulting Group. (2023). *2023 ERP report*.
- Pfeffer, J., & Sutton, R. I. (2006a). Evidence-based management. *Harvard Business Review, 84*(1), 62–74.
- Pfeffer, J., & Sutton, R. I. (2006b). *Hard facts, dangerous half-truths, and total nonsense: Profiting from evidence-based management*. Harvard Business School Press.
- Project Management Institute. (2023). *Pulse of the Profession 2023: Ahead of the curve: Forging a future-focused culture*. Project Management Institute, Inc.
- Quartz. (2019). Cost of bad managers and global productivity loss (reporting Gallup data).
- Rehder, T., Janssen, M., & Lang, N. (2021). Why technology transformations fail. Boston Consulting Group.
- Reuters. (n.d.). Volkswagen “Dieselgate” scandal cost estimated at €30B.
- Society for Human Resource Management. (n.d.). Data on communication issues.
- Standish Group International. (2020). *CHAOS Report 2020: Beyond Infinity*. Standish Group International.
- Wade, M., & Hulland, J. (2004). The resource-based view and information systems research: Review, extension, and suggestions for future research. *MIS Quarterly, 28*(1), 107–142. https://doi.org/10.2307/25148626

Input Streams

Communications & activity capture

- Email & calendars: Gmail / Google Workspace, Microsoft 365/Exchange/Outlook
- Meetings & notes: Zoom (recordings/notes), Microsoft Teams, Google Meet
- Chat & notifications: Slack, Microsoft Teams

CRM, sales & revenue

- CRMs: Salesforce (Sales/Service), HubSpot, Zoho,
- Sales engagement: Outreach, Salesloft
- Billing/RevRec: Zuora, Chargebee, Stripe Billing

Marketing, CDP & advertising

- Marketing automation: Marketo, HubSpot Marketing, Pardot/Account Engagement, Mailchimp
- Ad platforms: Google Ads/DV360, Meta, LinkedIn, TikTok, Amazon Ads, The Trade Desk

Commerce, retail & EDI

- Ecommerce: Shopify, BigCommerce, Adobe Commerce (Magento), Salesforce Commerce Cloud
- Marketplaces & retailer portals: Amazon Seller/Vendor Central, Walmart Retail Link/Marketplace, Target Partners Online, Kroger 84.51°, Costco, Home Depot, Lowe's, Instacart
- EDI networks/VANs & standards: SPS Commerce, TrueCommerce, OpenText; EDI X12/EDIFACT

Planning, pricing, PIM/MDM

- Planning: Anaplan
- Pricing/promo: Pricefx, PROS, Zilliant, Revionics
- PIM/MDM: Salsify, Akeneo, Stibo, Reltio, Informatica MDM

Work management, knowledge & HR

- Projects/tasks: Jira, Asana, Monday.com, Trello
- Knowledge/Docs: Confluence, Notion, SharePoint, Google Docs
- HR/HCM & WFM: Workday, SAP SuccessFactors, BambooHR, ADP, UKG/Kronos

AI/MLOps (optional for advanced use)

- Platforms: Databricks, AWS SageMaker, Google Vertex AI, Azure ML

Supply chain, ops & logistics

- OMS: Manhattan Active Omni, Blue Yonder, IBM Sterling, NetSuite OMS
- WMS: Manhattan, Blue Yonder, SAP EWM, Oracle WMS, Körber/HighJump
- TMS & visibility: Blue Yonder TMS, Oracle TMS, MercuryGate, project44, FourKites
- Carriers/3PLs: UPS, FedEx, USPS, DHL (APIs), major 3PL portals

Support, ITSM & contact center

- Support/ITSM: Zendesk, ServiceNow, Freshdesk, Jira Service Management
- Contact center/telephony: Twilio/Flex, Five9, RingCentral

Analytics, BI, observability & security

- BI/analytics: Tableau, Power BI, Mode, Superset

Data platforms & integration (databases, streams, files)

- Cloud data warehouses: Snowflake, BigQuery, Redshift, Azure Synapse
- Data lakes/lakehouse: AWS S3 + Glue/Athena, Azure Data Lake Storage, Google Cloud Storage, Databricks (Delta Lake)
- DBs: PostgreSQL, MySQL, SQL Server, Oracle, MongoDB, DynamoDB, Couchbase, Redis
- Time-series: InfluxDB, TimescaleDB, Azure Data Explorer (Kusto)
- Graph: Neo4j, JanusGraph
- Search/analytics: Elasticsearch, OpenSearch, Solr
- Streaming/messaging: Apache Kafka/Confluent Cloud, AWS Kinesis, Google Pub/Sub, Azure Event Hubs; RabbitMQ, ActiveMQ, NATS
- File/object & docs: SharePoint/OneDrive, Google Drive, Box, Dropbox; S3/GCS/ADLS
- Common formats: CSV, JSON, Parquet, Avro, ORC, XML, EDI X12/EDIFACT