

Your ERP Is Not Your Company. Your Data Model Is.

Whitepaper

Bart de Graaff · BdG Advisory · October 2026

For thirty years enterprise architecture protected the process and treated data as its exhaust. Cloud mostly moved those applications without rebuilding them. AI forces the rebuild, because it reasons over data whose meaning must survive outside the application that created it, and the largest application vendors are already buying their way into that layer.

Thirty years protecting the wrong asset

Enterprise architecture has been protecting the wrong asset for thirty years.

The asset it protected was the process. Workflow and ERP systems were built to coordinate people around predictable sequences: order to cash, procure to pay, hire to retire. Governance was wrapped around the procedure. That made sense when processes were stable, when people were the intelligence layer and when automation meant encoding what people already did.

AI does not operate on processes. It operates on data.

When AI reasons about an organisation, it does not reason about the workflow. It reasons about the transactions, the decisions, the customer history and the operational state: the permanent record of what actually happened. The process is how the organisation coordinated people. The data is what the organisation actually is.

That is the structural thesis of this paper. **Enterprise IT is moving from process-driven to data-driven architecture.** Cloud adoption did not make that move; it mostly relocated process-era applications without rebuilding them. AI forces the rebuild, because it needs data whose meaning survives outside the application that created it. In that architecture the data model, not the application, is the durable asset. Processes become temporary vehicles for decisions, rebuilt as often as the business changes. Organisations that re-architect around their data model and where their inference runs will compound advantage. Those that keep bolting AI onto process-era systems will compound drag.

This has a blunt commercial consequence. The ERP subscription is not protecting your competitive position. The data it generated is. A process platform can be replaced in a programme or two. The data model, the institutional knowledge encoded in it and the

decision history that trains your models compound over time and cannot be bought.

This paper traces how process-era architecture set the trap, why cloud did not spring it, what AI changes, how the largest application vendors are already re-positioning around data, the strongest case for staying put, and the structural moves that follow. Its companion, *Where Should Intelligence Live, and Who Controls It?*, deals with the operating model that has to govern this architecture.

The process era made sense, and it set the trap

The process era was not a mistake. It was a rational answer to the constraints of its time.

When computing was expensive and people did the thinking, the highest-value thing software could do was make work repeatable. ERP vendors codified reference processes, and buyers adopted them because standardisation was cheaper than invention and auditors liked it. The data each process produced was stored in the application's own schema, shaped by what the transaction needed, not by what the business might later want to know. Data was the exhaust of the process.

The incentives locked this in. Vendors sold process standardisation and licensed by user and module. Integrators were paid to configure processes and connect applications. Chief information officers were measured on go-live dates and budget. Every party had a reason to treat the application as the asset and the data as its by-product. Differentiation that did not fit the standard was pushed into customisation, which was expensive, or into spreadsheets, which were invisible.

The trap was that the data model got captured by the process model. What a customer is, what a product is, how stock is valued, when revenue counts: these definitions ended up inside application logic and configuration, defined differently in each system, reconciled by interfaces and month-end effort. The organisation's meaning was scattered across its process platforms.

That scattering carries a running cost that rarely appears on any budget line. A retail bank that runs lending, cards and deposits on separate platforms usually has three answers to the question "who is our customer?": an account holder in one, a cardholder in another, a household in a third. Each system is internally consistent. The bank pays for the inconsistency in integration projects, reconciliation teams, regulatory remediation and every analytics initiative that begins with months of matching records. Nobody owns the problem, because each platform owner owns a platform, not a definition. The integration tax is paid every year, in perpetuity, by an organisation that never decided to incur it.

Lidl's attempt to move onto SAP shows what happens when the two collide. Around 2011 the retailer began replacing its home-grown merchandise management system with an SAP-based platform. Seven years and an estimated €500 million later, it stopped. A central problem was not exotic. Lidl valued its inventory on purchase prices. The standard SAP retail software works on retail prices. Rather than give up a way of seeing its own business, Lidl

had the software adapted; performance fell and costs rose. Its chief executive concluded that the strategic goals were not achievable at an acceptable expense.

Read the case structurally. The fight was not about a process. It was about a data model: how Lidl defined the value of what it owned. When the standard process and the company's data model met, the data model won, because it was closer to what the company actually was. Seven years of effort went into making a process platform accept that fact.

Most organisations never run that experiment so expensively. They resolve the conflict quietly instead, by bending their definitions to fit the package and keeping the real ones in spreadsheets. That is how the process era hollowed out the asset that now matters most.

Cloud moved the applications, not the architecture

The cloud decade should have been the moment to fix this. For most enterprises it was not, and the reason is in the migration playbook itself.

The large cloud providers taught a clear method for big migrations. Amazon Web Services' own guidance lists seven strategies and says plainly that refactoring is not recommended for large migrations. It recommends rehosting, relocating or replatforming first, "and then modernizing the application after the migration is complete." Rehosting, it notes, means moving applications "without making any changes to the application."

That was sound advice for the goal most boards set: leave the data centre, convert capital spending to operating spending, reduce hardware risk. The business case was about where computing ran, not how the business was represented in data. And once the migration was complete, the modernisation step had to compete for a new budget against everything else. The migration was the project; the rebuild was a promise.

The incentives pointed the same way. The provider was paid for consumption, which a rehosted estate generates from day one. The integrator was paid to migrate, which is faster and more predictable without redesign. The CIO was paid to hit the exit date. Not one party in the room was paid to ask what the business's data model should be. Each of them behaved sensibly, and together they produced an estate that runs on new infrastructure with the old architecture intact.

The result is a familiar estate. A distributor moves its ERP and forty surrounding applications into a hyperscale cloud. The same nightly batch jobs run, now on rented servers. The customer is defined three ways in three systems, reconciled by the same interfaces. The infrastructure bill moved from capital to operating expense. The architecture did not move at all.

Cloud was never the destination. It was a decade-long detour around architectural decisions that are now unavoidable.

The next fork in that road is already scheduled. SAP has committed to mainstream maintenance for its Business Suite 7 core applications until the end of 2027, with optional

extended maintenance to the end of 2030. Every enterprise still running the previous generation of its ERP must decide what to do before those dates. The default path is to repeat the cloud pattern at the application layer: move the process platform to its successor, carry the customisations and the data definitions across, and call it modernisation. That would spend the largest architecture budget of the decade relocating the trap again.

AI reasons over data, not over workflows

AI changes the economics of this neglect, because it consumes data in a way process systems never did.

A process system needs only the data required for the next step. An approval needs the amount and the approver. A shipment needs the address and the stock. Meaning can stay implicit, because the workflow supplies the context. AI needs the opposite. To recommend a credit limit, flag a supplier risk or answer a customer, a model needs history and context across domains, with the meaning made explicit, because it has no workflow to infer it from. Data whose semantics live inside application code is close to useless to it.

Watch what happens when a mid-sized manufacturer asks an AI agent a simple question: which customers are exposed if a key supplier is late by three weeks? The answer needs supplier commitments from procurement, bills of materials from engineering, open orders from the ERP and contract penalties from the customer system. The four systems use different product codes, two definitions of "customer" and no shared notion of a delivery commitment. A planner answers the question in two days by knowing which spreadsheet reconciles what. The agent answers it in seconds, confidently and wrongly, because nobody ever wrote down the meaning the planner carries in her head. The model was not the weak link. The data model was.

Gartner has put a number on the consequence. It predicts that through 2026 organisations will abandon 60% of AI projects unsupported by AI-ready data, and found that 63% of organisations either do not have, or are unsure whether they have, the right data management practices for AI. Those projects will not fail because the models are weak. They will fail because the data model was never designed to be used by anything other than the process that wrote it.

The direction of travel goes further than data readiness. The logic that process applications encode is starting to move out of them. In early 2025 Microsoft's chief executive, discussing business applications in the agent era, put it bluntly: "The business logic is all going to these agents." One need not accept the strongest version of that forecast to see the structural point. If decision logic increasingly sits in an AI layer above the applications, the applications become places where data is written and stored. What then determines quality is whether the data underneath carries its meaning, and who controls that meaning.

There is a second, slower consequence. Process systems record outcomes, not reasons. The ERP knows that a discount was granted, a supplier was approved or a claim was paid. It rarely knows why, on what evidence, against which alternative, or what happened next. That

decision history is exactly what an organisation needs to train, test and correct the models that will make those decisions in future. Every year it is not captured is a year of institutional learning the organisation cannot get back. A competitor that starts recording decisions as data today will have something in three years that no licence can buy.

The new stack, and who wants to own it

Put these pressures together and the shape of a data-driven architecture is visible. It has four layers, and they have very different economics:

- **The ledger and systems of record.** Standard transactions on standard software, kept as close to the vendor's design as possible. Reliable, auditable, replaceable. Rent it.
- **The enterprise data model and data products.** The organisation's definitions of its customers, products, assets, commitments and decisions, governed by named owners and published as reusable products. This is the durable asset. Own it.
- **The decision layer.** Models, rules and agents that act on the data model, each placed where its decision needs to run. Change it as capability improves.
- **The process and experience layer.** The workflows and screens people and agents use, assembled from the layers below and rebuilt whenever the business changes. Treat it as disposable.

The process era fused all four inside the application. The data-driven architecture separates them, so that each can change at its own speed and the asset that compounds is not trapped inside the asset that depreciates.

Some vendors have built their whole proposition on this separation. Palantir's documentation describes its ontology as representing "the decisions in an enterprise, not simply the data", and argues that traditional data architectures do not capture the reasoning behind decisions or the actions that follow, which limits learning and the use of AI. Whatever one thinks of the product, the design premise is the right one: model the business as entities, relationships and decisions, and let processes be assembled on top.

Separation is also what makes inference placement a real choice. Once decisions are made from a shared data model rather than inside workflows, where a decision runs depends on where its data is created, how fast it must act and which rules govern it. Some decisions belong next to the central ledger. Others belong in a plant, a store or a device, which is the argument of The Intelligent Edge. A process-era architecture cannot make that choice, because its logic and data are fused inside the application. A data-driven architecture can.

The largest application vendors have read the same signals. If the application were still the durable asset, the companies that sell applications would be doubling down on applications. They are buying and building data layers instead.

In February 2025 SAP, the company most associated with process standardisation, launched SAP Business Data Cloud with Databricks embedded as a native service. SAP described it as

unifying SAP and third-party data and delivering managed data products "while maintaining original business context and semantics." The phrase matters. The vendor of the process platform is telling customers that the semantics, not the transactions, are what AI needs.

In May 2025 Salesforce agreed to buy Informatica for about US\$8 billion in equity value, describing the deal as strengthening the "trusted data foundation critical for deploying powerful and responsible agentic AI." The same month, ServiceNow, whose platform is built on workflow, signed an agreement to acquire data.world, a data catalogue and governance company, to deepen its data intelligence and governance for AI agents.

Three of the largest process and application vendors made the same move within four months. The mechanism is not hard to read. Each vendor can see that control is shifting from the process layer to the data and semantic layer, and each wants that layer to sit inside its platform. That is good news for the argument of this paper and a warning for buyers.

Who owns the data model owns the decisions. An enterprise that lets its semantics be defined inside a vendor's data cloud has not escaped lock-in. It has moved the lock from the process platform, where switching was painful, to the data platform, where switching is harder still, because the definitions are now what the models were trained on and what the agents act on. The vendors' strategies confirm that the data model is the durable asset. They do not confirm that the vendor should hold it.

The test for any data platform, whoever sells it, is therefore about control rather than features. Can the organisation define its own entities and relationships, or only accept the vendor's? Can it extract its data together with those definitions, the lineage and the history, in a form another platform can use? Does a model or agent from another supplier get the same access to the semantic layer as the vendor's own? A platform that fails those questions may still be the right purchase. But the buyer should know it is choosing where its decisions will be defined, not merely where its data will be stored.

The strongest objection: the ERP is the system of record

The best case against this argument comes from experienced operators, and it deserves a full answer.

It goes like this. The ERP is the system of record. Its standard processes are not bureaucracy; they are controls. They make the ledger auditable, the close repeatable and the financial statements defensible. Standardisation is how large organisations got cheaper and safer, and the last thing a finance chief needs is "data-driven" experimentation in the general ledger. The industry has heard the data story before. A decade of data lakes produced swamps nobody trusted. And the ERP vendors' own advice is now to keep the core clean, not to tear it apart. SAP's clean core strategy exists because many ERP landscapes "become weighed down by custom code, undocumented changes, and hard-to-maintain integrations". Why fight the platform?

Most of this is correct, and the answer accepts it. The ledger should stay a ledger. Standard processes should run the work that does not differentiate the business, and run it on the vendor's standard with as little modification as possible. Clean core is the right discipline.

But follow clean core to its conclusion. If customisation is removed from the process layer, the organisation's differentiation has to live somewhere else. It does not disappear because the core is clean. It moves into the data model, the data products built on it and the decision logic that uses them. Clean core is not an argument against data-driven architecture. It is the strongest argument for it: keep the process standard and rent it, and put the investment in the part that is yours.

The data lake objection is fair, and it is about ownership, not architecture. Lakes failed because data was centralised without owners: dumped in one place, defined by nobody, trusted by no one. The response the field developed, set out in Zhamak Dehghani's data mesh principles, starts from domain-oriented ownership and data as a product, with the business teams closest to the data accountable for its meaning and quality. A data-driven architecture without named owners will fail the same way the lakes did. With them, it is the first architecture in which the organisation's meaning has an accountable home.

A newer version of the objection says that none of this matters any more, because large language models can read messy data and infer its meaning. There is truth in it: models are good at reading documents, mapping fields and suggesting how records match. But inference about meaning is a guess, made afresh each time, with no owner and no audit trail. Two agents can guess differently about what "active customer" means and both sound confident. Models make it cheaper to build and maintain a data model. They do not remove the need for one. They raise the price of not having one, because every wrong guess is now executed at machine speed.

The claim, then, is not "rip out the ERP". It is "stop treating the ERP as the asset". Keep it as the reliable engine for standard transactions. Stop spending architecture budget making it the place where the business is defined.

Structural moves

The rebuild does not start with a platform purchase. It starts with a decision about what the organisation is going to own.

First, name an owner for the enterprise data model. Not for storage or pipelines, but for meaning: what a customer, a product, an asset, a contract and a transaction are, across every system. Give that owner authority over how vendors' schemas map to the enterprise model, and a seat in every architecture and procurement decision that touches it. Without a named owner, the data model will default to whatever the largest vendor defines. How that owner fits into the wider structure of authority is the subject of the operating model companion.

Second, treat the 2027 ERP decision as a data-model decision. Before choosing a migration path, decide which definitions are standard and can follow the package, and which encode how the business actually differs and must be held in the enterprise model. Migrate meaning deliberately rather than carrying it across inside customisations.

Third, stop customising process and start building data products. Keep the core standard. Put differentiation into governed data products and decision services that sit on the enterprise data model and can be reused by people, applications and AI alike.

Fourth, map decisions to data and inference placement. For each consequential decision, identify the data it needs, where that data is created, how fast the decision must be made and which rules govern it. Place the inference accordingly, centrally or at the edge, rather than wherever the application happens to run.

Fifth, contract for portability of data and semantics. In every platform agreement, secure the right to extract data together with its definitions, lineage and history in usable form. Do not move the lock from the process platform to the data platform.

Sixth, measure the architecture by how fast it feeds a decision. The useful metric is not go-lives or migrated workloads. It is how long it takes to give a new AI use case trustworthy, well-defined data, and how many definitions had to be reconciled by hand to do it.

Leaders should also stop:

- **Funding migrations that move applications without moving meaning.** A relocated estate is not a modernised one.
- **Building interfaces on interfaces** to reconcile definitions that should exist once.
- **Accepting a vendor's data model as the enterprise's data model** by default.
- **Measuring architecture programmes by go-live dates** rather than by what the data can now support.

Processes will keep changing; that is what they are for. The data model is what remains when they do: the record of what the organisation is, has done and has decided. AI rewards organisations that know the difference.

Rent the process. Own the data model.

What to do about it

- **Name an owner for the enterprise data model.** Give one owner authority over meaning, not storage: what customers, products, assets, contracts and transactions are across every system, and how vendor schemas map to them.
- **Treat the ERP migration as a data-model decision.** Decide which definitions can follow the standard package and which encode real difference and belong in the enterprise model, before choosing a migration path.

- **Keep the core standard and build data products.** Stop customising process. Put differentiation into governed data products and decision services that people, applications and AI can all reuse.
- **Map decisions to data and inference placement.** For each consequential decision, identify the data it needs, where that data is created and how fast the decision must run, and place inference there.
- **Contract for portability of data and semantics.** Secure the right to extract data with its definitions, lineage and history in usable form, so the lock does not simply move from the process platform to the data platform.
- **Measure time to trusted data.** Judge architecture by how quickly a new AI use case can get well-defined, trustworthy data, not by go-lives or migrated workloads.

Sources

- Consultancy.uk, Lidl cancels SAP introduction having sunk €500 million into it, 2018: <https://www.consultancy.uk/news/18243/lidl-cancels-sap-introduction-having-sunk-500-million-into-it>
- AWS Prescriptive Guidance, About the migration strategies (large migrations): <https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-guide/migration-strategies.html>
- SAP, SAP extends its innovation commitment for SAP S/4HANA, provides clarity and choice on SAP Business Suite 7, 2020: <https://news.sap.com/2020/02/sap-s4hana-maintenance-2040-clarity-choice-sap-business-suite-7/>
- Gartner, Lack of AI-ready data puts AI projects at risk, 2025: <https://www.gartner.com/en/newsroom/press-releases/2025-02-26-lack-of-ai-ready-data-puts-ai-projects-at-risk>
- Pure AI, Don't read too much into the 'collapse' of apps (Satya Nadella on business applications in the agent era), 2025: <https://pureai.com/articles/2025/02/25/dont-read-too-much-into-the-collapse-of-apps.aspx>
- Palantir, Why create an Ontology? (Foundry documentation): <https://palantir.com/docs/foundry/ontology/why-ontology/>
- SAP, SAP debuts Business Data Cloud with Databricks to turbocharge business AI, 2025: <https://news.sap.com/2025/02/sap-business-data-cloud-databricks-turbocharge-business-ai/>
- Salesforce and Informatica, Salesforce signs definitive agreement to acquire Informatica (SEC filing, Exhibit 99.1), 2025: <https://www.sec.gov/Archives/edgar/data/1868778/000119312525126272/d104080dex991.htm>
- ServiceNow, ServiceNow enhances its Workflow Data Fabric; agreement to acquire data.world, 2025: <https://newsroom.servicenow.com/press-releases/details/2025/ServiceNow-Enhances-Its-Workflow-Data-Fabric-With-New-Ecosystem-to-Power-AI-agents-and-Workflows-With-Real-Time-Intelligence/default.aspx>

-
- SAP, How to extend SAP S/4HANA Cloud the right way: clean and clear, 2025: <https://news.sap.com/2025/08/extend-sap-s4hana-cloud-right-way-clean-clear/>
 - Zhamak Dehghani (martinfowler.com), Data Mesh Principles and Logical Architecture, 2020: <https://martinfowler.com/articles/data-mesh-principles.html>

Read online: bdgadvisory.com/perspectives/data-driven-architecture