

The Test Data Imperative: Fueling the Journey towards SAP Autonomous Enterprise

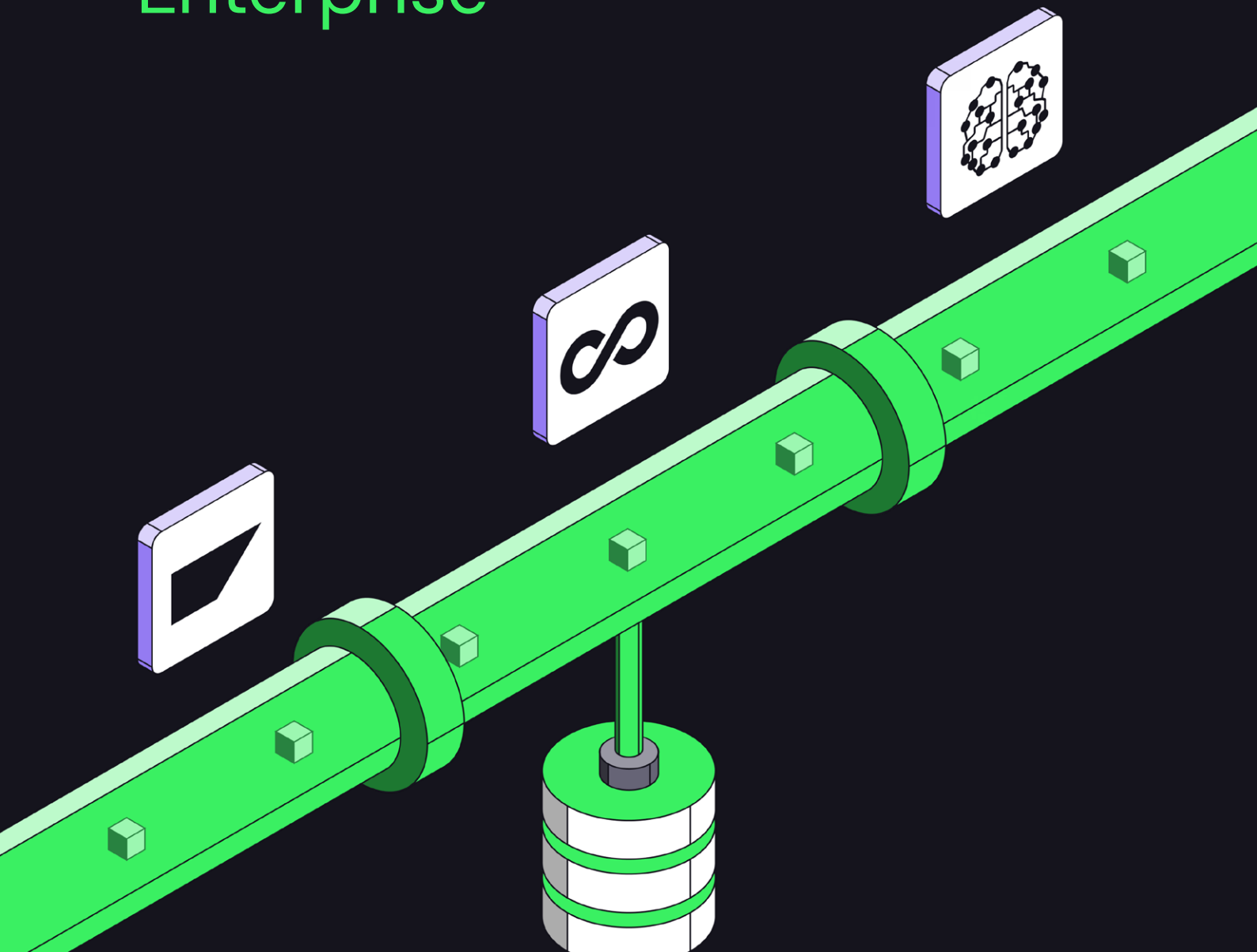


Table of Contents

About This Paper	3	The Test Data Control Plane: Making the New Playbook Real	15
The SAP Journey and Where Most Programs Actually Are	4	What the Control Plane Does	15
1. The Migration Program	4	The Control Plane Grows With the Program	15
2. The S/4HANA Stabilization Customer	4	Hybrid Human and Agentic Quality Engineering	16
3. The Autonomous Enterprise Early Adopter	5	The Partner Ecosystem Dimension	16
One Problem, Three Stages:	5	Agent Behavior Validation: The Risk Management Case	16
The Hidden Debt Holding SAP Programs Back	6	A Three-Stage Roadmap from Migration to Autonomous Enterprise	17
Legacy Data Debt	6	A Stage 1: Modernize (Migration Programs)	18
Compliance and Data Sensitivity	6	Stage 2: Accelerate (S/4HANA Continuous Delivery)	19
System Complexity and Cross-System Alignment	7	Stage 3: Autonomize (Autonomous Enterprise)	20
Delivery Velocity and the Continuous Testing Gap	7		
Agent Behavior Risk: A New Category of Testing Obligation	8		
Why Yesterday's Test Data Playbook Breaks at Today's Speed	9	Synthesized: The Control Plane for SAP Test Data in the Age of AI	22
The Manual Effort Problem	9	What That Means in Practice	22
The Stale Data Problem	10	Comparison: Traditional Approaches vs. Modern Control Plane	23
The Coverage Problem	10		
The Scale Problem	10		
Why Point Solutions Fail	11	The Fuel Layer as the Strategic Lever	24
A New Test Data Playbook for the SAP Journey	12		
Copy and Refresh: The Migration Program Foundation	12		
Masking and Scrambling: Compliance Across the Full Landscape	12		
Subsetting and Time Slicing: Right-Sized Environments for Continuous Delivery	13		
Synthetic Data Generation: Coverage That Production Cannot Provide	13		
Agentic-Ready Data Streams: The Autonomous Enterprise Foundation	14		
What Agentic-Ready Rules Out	14		

About This Paper

At Sapphire 2026, SAP laid out a compelling vision for the Autonomous Enterprise: AI assistants, specialized agents, and core business operations running with minimal human intervention. The ambition is credible, and the architecture behind it is sound.

Walk into most SAP programs today, and the picture looks very different. Migrations are taking longer than the business case promised, test environments are running on stale data, and QA teams are already behind. SAP's direction is clear; the gap between that direction and where most customers actually are is what this paper is about.

Most organizations are not at the Autonomous Enterprise yet, and the path to get there is longer and harder than any product announcement admits. At every point on that path, testing lives or dies on the quality of the test data. Get the test data layer right, and the critical testing that underpins program delivery accelerates; get it wrong, and it stalls, no matter how advanced the tooling on top.

This paper is for SAP transformation, testing, and QA leaders who need to make that shift real. It does not prescribe a single silver-bullet approach, because the right move depends on where your program is today and what it needs next. Instead, it shows why a modern test data foundation is one of the highest-leverage investments you can make in support of S/4 modernization and in order to realize agentic test/QA operations; and outlines practical steps to modernize and automate your Test/QA operations, whether you are running an ECC migration dry run or validating autonomous agent behavior before go-live.

The SAP Journey and Where Most Programs Actually Are

SAP is moving its customers toward an autonomous enterprise. The vision is credible: a unified AI platform, domain-specific Joule Assistants orchestrating hundreds of specialized agents, and a user experience built around outcomes rather than transactions. For customers who get there, the operational gains are substantial.

But the SAP customer base is not a monolith. The journey toward that destination looks different depending on where an organization is starting from, and the nature of the test data problem differs materially at each point.

1. The Migration Program

A substantial proportion of SAP customers are still running ECC or are in active transition to S/4HANA. The 2027 ECC maintenance deadline has concentrated minds, and for many organizations the migration is already underway, more complex and resource-intensive than the original business case suggested.

Test data problems here are immediate. Migration dry runs require environments that accurately reflect the current production landscape, but data volumes are large, compliance obligations constrain how production data can be used, and Custom ABAP code, Z-tables, and years of configuration drift mean standard tools routinely require significant manual intervention. The window for each dry run is tight. Programs that hit data preparation delays at this stage absorb them directly into the go-live schedule.

2. The S/4HANA Stabilization Customer

A second cohort has completed the migration and is operating S/4HANA at pace. SAP's quarterly release cadence applies regardless of internal readiness, and the integration complexity with non-SAP systems including Salesforce, Workday, Coupa, and specialist logistics and banking platforms was frequently underestimated during the migration phase. The regression suite that served adequately for bi-annual releases is struggling to keep up.

Test data problems here primarily concern frequency and cross-system consistency. Data refreshed several weeks prior no longer reflects the current configuration. Integration tests fail because test data across connected systems was prepared independently rather than aligned. The testing cadence has accelerated; the data layer has not.

3. The Autonomous Enterprise Early Adopter

A third, smaller cohort is beginning to deploy Joule Assistants and evaluate agentic workflows. These are typically larger, more mature organizations with the capacity to move early on new platform capabilities.

For these programs, the test data question has shifted character. Validating that a process completes correctly is no longer sufficient. What matters is whether an AI agent behaves correctly across the full distribution of data conditions it will encounter in production — the period-end anomalies, multi-currency combinations, and configuration states that rarely occur but are precisely where autonomous execution fails in consequential ways. An agent acting on incorrect data does not produce a wrong report. It executes a wrong transaction, at machine speed, before anyone notices.

One Problem, Three Stages

These are not three discrete problems. They are three stages of the same underlying SAP modernization challenge. The data foundation that addresses migration program requirements is the same foundation, appropriately extended, that supports autonomous agent validation. Establishing it correctly from the outset avoids rebuilding it at each stage of the journey.

THE FUEL LAYER METAPHOR

SAP programs operate across three interdependent layers. Infrastructure keeps systems running. Code defines process logic. Test data is the fuel layer that fuels the velocity of the testing process, and it determines whether that logic behaves correctly under conditions that reflect production reality. Most SAP programs have modernized the first two layers. The test data layer has not kept pace.

It continues to be managed manually, on schedules set years ago, by teams without the tooling to match the complexity and cadence of modern SAP delivery. The dependency is not always well understood until it becomes the reason why a delivery program runs late.

The Hidden Debt Holding SAP Programs Back

Most SAP programs inherit test data problems they did not create. Custom scripts maintained by consultants who have since left. Masking rules written for schemas that no longer exist. Compliance obligations that were never properly addressed. The starting point for fixing any of this is recognizing it as a structural problem rather than an operational inconvenience.

Legacy Data Debt

SAP programs carry a consistent operational overhead that rarely appears on a program dashboard: delayed environment preparation, test cycles that cannot start on schedule, and QA engineers spending significant time on data preparation rather than testing. The root cause is legacy data debt.

Legacy data debt accumulates when test data environments and processes that were adequate at an earlier point fail to keep pace with the landscape they are supposed to represent. Custom ABAP scripts written for schemas that have since been extended, refresh processes held as tribal knowledge by a single Basis engineer and undocumented anywhere, masking rules that haven't been updated since the last major module rollout, and Z-tables carrying sensitive commercial data that nobody classified because they fell outside standard SAP.

This debt is self-reinforcing. Release cycles running on stale data add to it. Compliance gaps patched manually rather than resolved systematically add to it. Test environments prepared by hand rather than provisioned on demand add to it. Addressing it requires treating it as a structural issue rather than a recurring operational cost.

Compliance and Data Sensitivity

Compliance conversations in SAP test data contexts default quickly to GDPR and PII. That framing captures only part of the picture. Data sensitivity in SAP environments spans three distinct categories, each with different regulatory implications and enforcement consequences.

1. **Sensitive personal data** covers customer names, addresses, financial records, and employee and payroll information, governed by GDPR, CCPA, LGPD, and equivalent frameworks. It is most commonly found in HR modules and customer-facing systems of B2C organizations. Personal data should not appear in non-production environments without masking or anonymization.

2. **Corporate compliance-sensitive data** receives less attention but represents a broader challenge. Supplier pricing and contract terms, intellectual property, internal financial models, and commercially sensitive operational data fall into this category. The standards are set by the organization rather than an external regulator, which means they vary, are often informal, and are rarely enforced systematically across non-production environments.
3. **Government-sensitive data** applies to organizations in defense, aerospace, and other regulated industries. ITAR imposes strict controls on defense-related technical data in non-production environments, and the consequences of non-compliance are severe. This category cannot be managed through inconsistent manual masking.

All three require systematic, auditable, and consistent data protection across every non-production environment. Manual masking maintained by hand as schemas evolve cannot reliably meet that standard. The gap is most acute during migration, when data passes through more environments with less oversight than in a steady state.

System Complexity and Cross-System Alignment

Enterprise SAP environments are not single systems. A typical landscape includes S/4HANA or ECC at the core, surrounded by Ariba, SuccessFactors, and SAP BTP, connected to Salesforce, Workday, and specialist logistics and banking platforms. Business processes run across all of them simultaneously.

Testing Order-to-Cash requires consistent, aligned data across every system the process touches: from opportunity origination in Salesforce through order processing and logistics in SAP, procurement in Ariba, and financial posting in the general ledger. When test data is managed separately across these systems, by different teams using different tools on different schedules, that alignment breaks down. Integration tests produce false positives. Defects that only surface under cross-system conditions are missed until UAT or production. It is one of the most consistent sources of late-stage failures in SAP programs, and one of the areas where single-system TDM approaches fall shortest.

Delivery Velocity and the Continuous Testing Gap

SAP delivery cadences have accelerated considerably. Bi-annual release cycles are giving way to quarterly schedules, and S/4HANA Cloud programs run on SAP's release calendar regardless of internal readiness. Continuous testing, running regression, integration, and process validation as frequently as the delivery cadence demands, is becoming the operational standard.

Continuous testing requires continuous test data: data available when a testing cycle starts rather than when preparation finishes, and data that reflects current system conditions rather than a refresh

from three weeks prior. Most SAP testing teams are not yet operating at that standard. The data layer is running on a different schedule from the programs it is supposed to support.

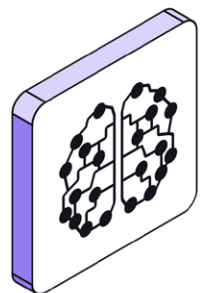
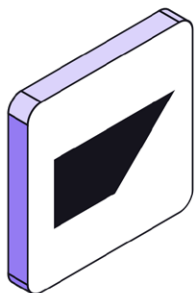
Agent Behavior Risk: A New Category of Testing Obligation

Organizations beginning to deploy Joule Assistants and agentic workflows face a testing obligation that traditional frameworks were not designed to address.

SAP's Autonomous Close Assistant illustrates the stakes clearly. By automating journal entries, reconciliation, and error resolution, it compresses financial close from weeks to days. That compression also narrows the window in which a human can identify and correct an error before it propagates. The agent does not produce a report for review. It executes transactions.

Validating that a process completes is no longer sufficient. What matters is whether it executes correctly across the full range of data conditions the agent will encounter in production: period-end anomalies, multi-currency combinations, intercompany scenarios, and configuration states that rarely occur in standard operations but are precisely where autonomous execution fails in ways that are consequential to an auditor or regulator.

Test data for agentic SAP systems is a risk management requirement, not a quality engineering consideration.



Why Yesterday's Test Data Playbook Breaks at Today's Speed

The test data management approaches most SAP teams rely on were built for a different era: simpler systems, slower release cycles, and compliance requirements that were less systematically enforced. They still work within those original parameters. Outside them, they create the bottlenecks that programs feel most acutely.

The Manual Effort Problem

SAP test data preparation remains largely manual for most organizations. The process is consistent across programs: identify what data a testing cycle requires, extract a subset from production, apply masking rules to sensitive fields, validate that relationships across SAP's interconnected tables still hold, fix the ones that broke, and load the result into the environment. Then configuration changes and the cycle starts again.

The process works when executed carefully by people who understand both the SAP data model and the specific business processes being tested. It does not scale. It concentrates critical knowledge on a small number of people, and when those people are unavailable, the test environment is not ready when the program needs it.

For many teams, the process runs on custom ABAP scripts or database extracts that represent years of accumulated, largely undocumented knowledge. When those people leave, the knowledge goes with them. When SAP upgrades change the data model, the scripts break. Each new environment requires work from scratch. This was manageable when release cycles were slower and landscapes less complex. Neither of those things describes most SAP programs today.

Migration programs feel this most acutely. A single dry run requires coordination across Basis, infrastructure, QA, and business teams — and when any part of that breaks down, the impact lands directly on the go-live schedule.

The Stale Data Problem

Test environments begin diverging from production the moment they are created. Configuration changes. New business rules are implemented. Transactional data accumulates. The test environment, fixed at the point of its last refresh, progressively misrepresents the system it is supposed to reflect. SAP customers on a yearly QA refresh cycle spend nine months testing against data that is more than three months out of date. In programs where configuration is changing weekly — which describes most active migration programs and all S/4HANA Cloud customers — the divergence becomes significant within weeks.

For agentic systems, the stakes are higher. An agent trained or tested against conditions that no longer exist will execute transactions against a system that has moved on. Stale test data does not just reduce coverage. It trains agents toward incorrect behaviors that only surface when those conditions are encountered in production — by which point the damage is already done.

The Coverage Problem

Production data is a record of what has occurred, not a comprehensive representation of what the system needs to handle. A standard production subset covers the transactions from the extraction period. It does not cover edge cases that arise under specific conditions: the unusual order type that exposes a fulfillment logic defect, the payment combination that breaks a finance posting rule, or the configuration state that occurs only during a particular business process variant.

Adequate test coverage cannot be derived from production data alone. It requires data built deliberately for the scenarios being tested.

For autonomous enterprise deployments, this limitation is critical. Joule agents operate across the full distribution of real-world conditions, not just the transactions that have historically occurred. Test data scoped only to prior production activity leaves agents unvalidated against precisely the conditions where consequential errors are most likely.

The Scale Problem

Full database copies require storage proportional to the production system. For SAP landscapes running into terabytes, maintaining separate near-production environments for development, QA, performance testing, and UAT represents a substantial and ongoing infrastructure cost.

SAP's own documentation puts client copy runtimes at several hours to several days depending on data volume. Across a full landscape with multiple environments on regular refresh schedules, that overhead accumulates across team time, infrastructure spend, and delayed test cycles in ways that rarely appear as a single line item but are consistently absorbed by the program.

Why Point Solutions Fail

Many SAP programs have tried to solve individual test data problems in isolation. A masking tool for the HR module, a subsetting script for the finance environment, and a synthetic data generator for a specific integration scenario. Each addresses the immediate problem. Collectively, they create new ones.

When test data across systems is managed by different tools on different schedules, cross-system consistency breaks down. Masking rules applied in one tool are unaware of subsetting decisions made in another, and integration tests run against data that was never built to be coherent across the full process chain. The point solution approach fixes what is visible today and creates the conditions for the next failure tomorrow.



A New Test Data Playbook for the SAP Journey

Modern SAP test data management represents a shift from manual, scheduled, effort-intensive processes to an automated, on-demand, governed capability covering every generation method programs require. The right method for any given scenario depends on the scenario itself. The role of a modern test data platform is not to prescribe which approach to use, it's to make all of them available, reliable, and fast enough to keep pace with the program.

Copy and Refresh: The Migration Program Foundation

Copy and refresh remains the most widely used and most appropriate approach for migration programs. Production data reflects actual business transactions and provides the most representative picture of how the system behaves under real conditions. The challenge is not the method itself. It is the time, manual effort, and compliance exposure required to execute it at the frequency and scale migration programs demand.

Modern provisioning addresses that overhead directly. What previously required four weeks of coordination across sixteen people now takes forty minutes on demand. Migration dry runs that previously needed days of environment preparation can start the same day. That is not an incremental improvement. It changes what is feasible within a program schedule.

Masking and Scrambling: Compliance Across the Full Landscape

Masking and scrambling protect sensitive data while preserving the structure and behavior of the underlying dataset. The differentiator in SAP environments is context awareness. Generic masking tools do not understand which fields drive SAP business logic and which cannot be modified without breaking process integrity.

Country codes, configuration-controlled reference data, and system-managed identifiers must remain unchanged. A masking approach that treats all fields uniformly produces environments that are either over-exposed or non-functional. SAP-aware masking understands which fields carry personal data, which carry corporate compliance-sensitive data, and which are functionally constrained, applying protection accordingly.

This matters across all program stages but is especially critical during migration, when data passes through more environments, more frequently, and with less governance oversight than in a steady state.

Subsetting and Time Slicing: Right-Sized Environments for Continuous Delivery

Subsetting and time slicing produce right-sized environments scoped to specific testing scenarios rather than carrying the full weight of a production copy. For S/4HANA customers on quarterly release cycles, this is what determines whether a test environment is ready when the program needs it.

The core challenge in SAP is that business processes do not respect calendar boundaries. An order placed in one period connects to a goods receipt in the next, which connects to an invoice that may fall entirely outside the specified date range. SAP-aware time slicing follows business process boundaries rather than calendar dates, extending the extraction window where necessary to preserve referential integrity. Generic time slicing produces broken datasets. SAP-aware time slicing produces environments that work.

Synthetic Data Generation: Coverage That Production Cannot Provide

Synthetic data generation addresses gaps that production data cannot. When a specific order type, payment combination, or process variant rarely occurs in production but needs to be tested, synthetic generation creates it. Where regulatory requirements or contractual obligations prohibit the use of production data in test environments entirely, synthetic generation provides coverage without compliance exposure.

For autonomous enterprise programs, synthetic generation moves from supplementary to essential. Production data reflects historical transactions. Joule agents require validation against the full range of conditions they will encounter in production, including the process variants, period-end anomalies, and configuration states that production data does not naturally contain — precisely where autonomous execution is most likely to produce consequential errors.

Agentic-Ready Data Streams: The Autonomous Enterprise Foundation

As AI agents enter SAP quality engineering, the test data layer must meet a higher standard. Agentic-ready data has three defining characteristics.

1. **Machine-consumable:** structured, accessible via API, and available without manual intervention between the request and the result.
2. **Privacy-safe:** compliant by design, with no PII or regulated data accessible to agents that do not require it.
3. **Scenario-rich:** covering the full range of business process variants, edge cases, and data states that agents need to explore and validate.

An agent running continuously cannot wait for a weekly refresh cycle. It cannot work with data that was masked three weeks ago and has not been updated since. The data layer needs to run at the same cadence as the agents it supports. For human-driven testing, that layer was important. For agentic AI, it is foundational.

What Agentic-Ready Rules Out

Non-deterministic outputs that produce different results across runs cannot serve as a reliable foundation for test data. Solutions that require data to leave the network create immediate GDPR and SOX exposure in regulated SAP environments. Platforms with no persistent state, no seed registry, and no audit history cannot demonstrate compliance to an auditor. Any approach that cannot connect directly to SAP HANA, read production schemas in real time, mask in place, and write back without human intervention is not infrastructure. It is a script.

The bar for agentic-ready SAP test data is considerably higher than most current solutions were designed to meet.

The Test Data Control Plane: Making the New Playbook Real

Point-solution fragmentation cannot be resolved by a better point solution. It requires a unified control plane: a central capability covering every generation method, every system in the landscape, and every stage of the program.

What the Control Plane Does

The SAP CoE owns the platform and defines policies, data models, compliance rules, and provisioning standards. It maintains a catalog of reusable, versioned test data sets for Order-to-Cash, Procure-to-Pay, Record-to-Report, and other critical processes, available to feature teams and automation suites on-demand.

Feature teams request what they need through self-service interfaces or API calls without needing to understand the underlying SAP data model. The platform handles schema awareness, cross-system alignment, and compliance enforcement automatically.

When a team needs data to test Order-to-Cash in S/4HANA, the platform understands what that process requires across modules and non-SAP systems, including Salesforce, Ariba, and downstream logistics platforms, and provides a coherent, compliant dataset accordingly.

The Control Plane Grows With the Program

A migration program deploys the control plane to manage copy and refresh automation, masking governance, and environment provisioning. As the program matures, it extends to continuous provisioning, cross-system alignment, and reusable scenario-based datasets for quarterly release cycles. Later still, agentic sandboxes, continuous data streams for Joule agents, and scenario coverage for conditions that those agents will encounter in production.

The platform scales with the program rather than requiring replacement at each stage. It is a foundational investment that accumulates value over time rather than being rebuilt at every inflection point.

Hybrid Human and Agentic Quality Engineering

The near-term future of SAP quality engineering is a structured partnership between human judgment and AI execution rather than full autonomy. Agents run regression suites at frequencies no human team could sustain, identify high-risk scenarios, and adapt when the system changes without waiting for someone to update a script. Human QE defines the risk models, sets the guardrails, and determines what constitutes acceptable quality.

Test data is what makes this partnership work at scale. It needs to be continuous, compliant, and scenario-rich enough to support both the systematic regression humans plan and the exploratory validation agents generate.

The Partner Ecosystem Dimension

SAP's €100 million partner fund for Joule Studio builds, and AI assistant deployments signal a significant expansion of implementation activity across the ecosystem. Each of those deployments requires test data infrastructure.

Synthesized provides the layer that makes partner-built agents testable, compliant, and ready for enterprise deployment. For implementation partners, building AI assistant delivery practices on a robust test data foundation is increasingly what differentiates on quality and compliance, not just delivery speed.

Agent Behavior Validation: The Risk Management Case

SAP's Autonomous Close Assistant compresses financial close from weeks to days by automating journal entries, reconciliation, and error resolution. That compression also narrows the window in which a human can catch and correct an error before it propagates. Joule agents analyzing thousands of historical incidents to generate pre-filled maintenance work orders act on patterns derived from data. In both cases, the correctness of the agent's behavior depends entirely on the quality of the data it was tested against before it was trusted to act.

Validating agent behavior across the full range of data conditions, including the edge cases that production data does not contain, is not a quality engineering exercise. It is the risk management requirement that makes autonomous enterprise operations defensible to auditors, regulators, and boards.

A Three-Stage Roadmap from Migration to Autonomous Enterprise

Building a modern SAP test data capability does not require a multi-year transformation. Organizations that have made the most progress tend to follow the same pattern: address the highest-impact problem first, demonstrate value quickly, and expand from there. The roadmap below maps that progression across three stages.

Most programs begin at Stage 1. The platform scales through to Stage 3 without replacement.

STAGE	FOCUS AND OUTCOMES
Stage 1: Modernize (Migration Programs)	Automate copy and refresh. Implement SAP-aware masking and compliance governance. Eliminate manual data preparation overhead. Reduce migration dry-run preparation from weeks to hours. Establish the data foundation for every subsequent stage.
Stage 2: Accelerate (S/4HANA Continuous Delivery)	Move to on-demand provisioning. Extend cross-system alignment across SAP and non-SAP landscapes. Build reusable scenario-based data sets for Order-to-Cash, Procure-to-Pay, and Record-to-Report. Match data velocity to release velocity.
Stage 3: Autonomize (Autonomous Enterprise)	Stand up agentic sandboxes. Implement continuous data streams for Joule agents. Build scenario coverage for edge cases agents will encounter. Connect data generation to transport intelligence and risk-based testing.

A Stage 1: Modernize (Migration Programs)

Automate copy and refresh. Establish the compliance foundation. Eliminate manual overhead.

START WITH SAP DATA DISCOVERY AND RISK MAPPING

Before anything can be automated, the landscape needs to be understood. Inventory the full SAP environment: S/4HANA or ECC, satellite applications such as Ariba and SuccessFactors, and non-SAP systems that share data across business processes. Map critical schemas and table relationships, including Z-tables and custom extensions that standard tooling will not automatically detect.

Run automated schema analysis and sensitive data detection to classify fields and regulatory obligations upfront. Define custom classification rules using regular expression (regex) patterns that reflect the organization's specific data sensitivity standards, covering standard SAP fields alongside the corporate compliance-sensitive and government-sensitive categories described in Chapter 2. This classification work is the foundation on which everything else is built. Time invested here reduces rework and compliance risk at every subsequent stage.

DEFINE BUSINESS-CRITICAL SCENARIOS AND QUALITY RISKS

SAP business processes carry different levels of risk. Some are stable, well-understood, and rarely the source of significant defects. Others are complex, frequently changing, and consistently where problems surface. Prioritize the core flows: Order-to-Cash, Procure-to-Pay, Record-to-Report, and any country- or business-unit-specific variants with regulatory or operational significance.

Combine business input, production usage patterns, and historical defect data to define a risk-based test data strategy for each scenario. This shapes which data to provision first, which generation method suits each scenario, and which compliance controls apply.

THE PROOF POINT

What previously required four weeks of coordination across sixteen people now takes forty minutes on demand. Migration dry runs that previously needed days of environment preparation can start the same day. That is not an incremental efficiency gain. It changes what is achievable within a program schedule.

Stage 2:

Accelerate (S/4HANA Continuous Delivery)

Move to on-demand provisioning. Build the reusable scenario library. Match data velocity to release velocity.

DESIGN REUSABLE TEST DATA SETS FOR SAP

For each critical business process, design reusable data templates that preserve SAP business logic and cross-system referential integrity. Treat these as versioned test data products, owned by the SAP CoE and consumed across multiple workstreams rather than rebuilt for each release cycle.

Each template should be scenario-specific and process-complete. An Order-to-Cash template contains the aligned customer master, credit limit configuration, pricing conditions, delivery data, goods movement records, and corresponding financial postings — consistent across SAP and the non-SAP systems the process touches, including Salesforce and downstream logistics platforms. The same principle applies to Procure-to-Pay templates spanning SAP and Ariba, as well as Record-to-Report templates covering the full financial close cycle.

Version these templates. When SAP configuration changes or a new integration is added, the template is updated through governance review rather than being rebuilt from scratch. This is what makes continuous testing sustainable at a quarterly release cadence.

ESTABLISH THE SAP TEST DATA CONTROL PLANE

Deploy Synthesized as the unified control plane across the SAP and non-SAP landscape, covering generation, masking, subsetting, and provisioning from a single governed platform. Define policies governing who can provision which datasets, into which environments, under which privacy and governance rules.

The SAP CoE owns the control plane. Feature teams and QA engineers access it through self-service interfaces or API calls without needing to understand the underlying SAP data model. The platform handles schema awareness, cross-system alignment, and compliance enforcement, delivering compliant, production-realistic data on demand.

INTEGRATE WITH THE SAP TESTING TOOLCHAIN

Embed test data provisioning into SAP testing workflows so that regression packs, migration dry runs, and performance tests receive fresh data on demand rather than waiting on manual preparation cycles.

Wire the control plane into Tricentis Tosca so that suites can request named SAP datasets as part of standard test execution jobs. Connect to UiPath agentic testing for SAP so automated process flows run against the current configuration data. Integrate with SAP Solution Manager and SAP Cloud ALM so data provisioning is triggered automatically as part of release governance rather than managed as a separate workstream.

Where programs use custom ABAP-based test automation or SAP TAO, the control plane's API-first architecture allows dataset requests to be embedded directly into existing test scripts, removing the manual preparation steps that currently sit outside the pipeline.

The result is a testing toolchain where the data layer keeps pace with everything running on top of it.

Stage 3: Autonomize (Autonomous Enterprise)

Stand up agentic sandboxes. Build continuous data streams for Joule agents. Connect data intelligence to risk-based testing.

ADD AN AGENTIC TEST SANDBOX

Stand up a dedicated synthetic-only sandbox where agentic AI test agents can safely explore and run SAP test scenarios without PII exposure. This environment intentionally sits apart from regulated QA and UAT environments, giving agents the freedom to operate against realistic yet entirely synthetic data.

Use this sandbox to prototype self-healing, exploratory, and risk-based test workflows for S/4HANA and Fiori. Validate Joule Assistant behavior against process variants and data states that production data does not contain before those agents are trusted to execute real transactions. For the Autonomous Close scenario specifically, that means testing against multi-currency anomalies, intercompany transaction combinations, period-end exceptions, and reconciliation edge cases — conditions that standard regression cycles rarely surface but that autonomous agents will encounter.

FEED AGENTIC, RISK-BASED TESTING WITH THE RIGHT DATA

Combine change intelligence — transport history, configuration diffs, SAP Knowledge Graph relationship maps — with usage analytics and historical defect data to direct data generation toward the highest-risk processes first. The data layer should be intelligent about where it focuses, not just fast at generating volume.

Expose APIs so agentic tools, including Tricentis AI features, UiPath agentic testing for SAP, and SAP AI Core-based agents can pull tailored datasets per test mission rather than consuming a static shared environment. Each agent gets the data it needs for its specific objective, generated on demand, compliant by design, and scoped to the relevant process boundaries.

A continuously running agent cannot wait for a weekly refresh. The data infrastructure must run at the same cadence.

OPERATIONALIZE GOVERNANCE AND CONTINUOUS IMPROVEMENT

Establish KPIs across migration and BAU delivery: time to provision SAP test data, coverage depth per scenario, defects caught pre-production, and privacy incidents. Review them consistently with SAP CoE, QA, and platform teams.

Use KPI trends to refine data models, expand scenario libraries, and adjust compliance guardrails as the program evolves. High-risk scenarios surfaced by Joule agents feed back into the synthetic data templates built in Stage 2. Compliance gaps identified in governance reviews update the classification rules established in Stage 1. The capability improves alongside the quality engineering program it supports.

ROLL OUT BY WAVE: PILOT, SCALE, INDUSTRIALIZE

Start with one S/4HANA release train or country rollout. Establish cycle-time reduction and defect-catch improvement against a measurable baseline, then extend them to more processes, regions, and integration points. Industrialize the approach so the test data control plane becomes a standard component of the SAP Activate methodology and ongoing quarterly release governance, a permanent operational capability rather than a workstream that gets wound down after go-live.

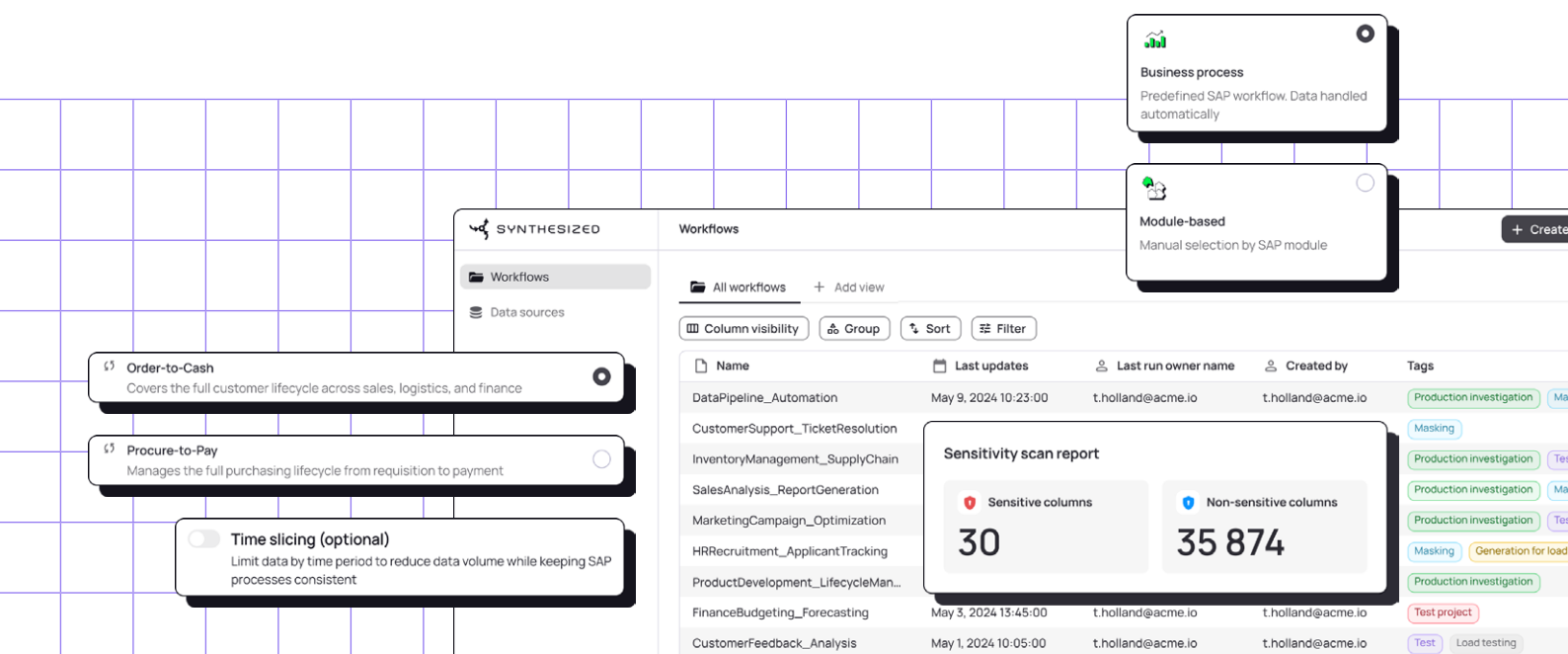
Synthesized: The Control Plane for SAP Test Data in the Age of AI

Synthesized is the only test data management platform that combines native SAP intelligence with comprehensive coverage across non-SAP systems, serving every stage of the program from ECC migration through to autonomous enterprise operations from a single control plane.

What That Means in Practice

Schema discovery that understands SAP table structures and relationships, including Z-tables and custom extensions. Module-aware masking that knows which fields can and cannot be modified without breaking business logic. Time slicing that follows business process boundaries rather than calendar dates. Synthetic data generation covering the edge cases and process variants that production data does not contain.

Every generation method is available from a single control plane, simultaneously across SAP and non-SAP systems. Storage footprints up to 99% smaller than those of full-copy approaches. Provisioning in minutes rather than days. Compliance governance covering classification rules, audit logs, and policy enforcement that withstands scrutiny from regulators, auditors, and internal security teams.



The screenshot displays the Synthesized control plane interface. On the left, a sidebar shows 'Workflows' and 'Data sources'. The main area is titled 'Workflows' and lists various workflows such as 'DataPipeline_Automation', 'CustomerSupport_TicketResolution', and 'InventoryManagement_SupplyChain'. A 'Sensitivity scan report' is overlaid on the workflow list, showing 30 sensitive columns and 35,874 non-sensitive columns. On the right, two callout boxes describe 'Business process' (Predefined SAP workflow) and 'Module-based' (Manual selection by SAP module). At the bottom right, a 'Tags' section lists various tags like 'Production investigation', 'Masking', and 'Test project'.

Business process
Predefined SAP workflow. Data handled automatically

Module-based
Manual selection by SAP module

Sensitivity scan report

Sensitive columns	Non-sensitive columns
30	35 874

Workflows

Name	Last updates	Last run owner name	Created by	Tags
DataPipeline_Automation	May 9, 2024 10:23:00	t.holland@acme.io	t.holland@acme.io	Production investigation, Masking
CustomerSupport_TicketResolution				Production investigation, Test project
InventoryManagement_SupplyChain				Production investigation, Masking, Generation for load
SalesAnalysis_ReportGeneration				Production investigation, Test project
MarketingCampaign_Optimization				Production investigation, Masking
HRRecruitment_ApplicantTracking				Production investigation, Test project
ProductDevelopment_LifecycleMan...				Production investigation, Test project
FinanceBudgeting_Forecasting	May 3, 2024 13:45:00	t.holland@acme.io	t.holland@acme.io	Test project, Load testing
CustomerFeedback_Analysis	May 1, 2024 10:05:00	t.holland@acme.io	t.holland@acme.io	Test project, Load testing

Order-to-Cash
Covers the full customer lifecycle across sales, logistics, and finance

Procure-to-Pay
Manages the full purchasing lifecycle from requisition to payment

Time slicing (optional)
Limit data by time period to reduce data volume while keeping SAP processes consistent

Comparison: Traditional Approaches vs. Modern Control Plane

TRADITIONAL APPROACH	SYNTHESIZED CONTROL PLANE
Manual scripts and tribal knowledge	Automated, governed, and auditable by design
Periodic refresh cycles (monthly or quarterly)	On-demand provisioning in minutes
Single-system masking tools	Cross-landscape masking across SAP and non-SAP
Calendar-based time slicing	Business process-boundary time slicing
Point solutions creating alignment gaps	Unified control plane across all methods and systems
Human-driven testing only	Supports human and agentic test workflows
Compliance as a manual checklist	Compliance by design with full audit history
Data layer rebuilt at each program stage	Foundation investment that compounds across stages

CONCLUSION

The Fuel Layer as the Strategic Lever

Most organizations reading this are already somewhere along the journey described in this paper.

- Migrations in progress
- Go-lives completed or approaching
- AI impact analysis is under evaluation or already running
- Agentic testing on the roadmap.

None of that is theoretical. It is already happening.

What it is building toward is quality engineering that is genuinely proactive rather than simply faster. Agents continuously probe SAP environments for risk rather than waiting for a scheduled regression window. Test generation driven by what changed in a transport rather than what someone had the capacity to write. Automation that adapts as the system evolves, without requiring manual intervention.

The data layer is what makes that possible or prevents it.

Agents that run on stale data develop incorrect behavioral patterns, and agents with access to sensitive data create compliance exposure that undoes months of governance work. The gap between the ambition of autonomous SAP quality engineering and the reality of how test data is currently managed is where most programs will either accelerate or stall.

Most SAP programs are committed to the journey toward S/4HANA and the Autonomous Enterprise but have not yet reached the destination. The migration, the release pressure, and the compliance obligations are all still very much in play. The data layer that served the previous operating model was not built to meet the demands of the journey now.

SAP has built the platform, deployed the agents, and begun automating the processes. What determines whether that investment delivers is not the AI agent's sophistication but the quality of the data those agents were tested against before they were trusted to act.

The programs that establish the test data foundation now, treating it as a strategic capability rather than a migration workstream or a compliance checkbox, will be the ones that move fastest and with the most confidence at every stage. The fuel layer is what makes the engine run efficiently, faster and continuously.

Now is the time to put it in place.

About Synthesized

Synthesized redefines software quality with intelligent test data automation and secure, compliant test environment provisioning. It is the only test data management platform that unifies AI-driven test data automation, test data management, and synthetic data generation for both SAP and non-SAP enterprise applications and system testing.

Its platform enables engineering teams and business users to generate production-ready test data on demand through an API-first, compliance-by-design approach, turning test data into a strategic enabler of continuous, AI-driven delivery.

Recognized by Gartner as a Cool Vendor and trusted by global enterprises across finance, telecom, and government, Synthesized provides SAP programs with the data foundation to move faster and with greater confidence at every stage of the journey.

Next Steps

LEARN MORE ↘

synthesized.io

CONTACT US ↘

Request an executive briefing & demo to understand how Synthesized eliminates test data constraints and accelerates your transformation initiatives: synthesized.io/contact-sales