



A FIGAF WHITEPAPER

SAP PI/PO to SAP Integration Suite Migration Guide

A practical framework for assessing, planning, migrating, testing, and governing your move to SAP Integration Suite

EXECUTIVE SUMMARY

Migrating off SAP PI/PO isn't a discretionary modernization project. SAP ends mainstream maintenance for PI/PO on December 31, 2027; extended maintenance runs only to December 31, 2030, and only at additional cost. A migration of any real size takes long enough that the deadline already sits inside your planning horizon.

For any organization still running PI/PO, the question isn't whether to migrate. It's how much of that shrinking runway you're willing to spend before you start.

This guide sets out a repeatable framework for the move to SAP Integration Suite: sizing the effort, choosing the right architecture, migrating ICOs, mappings, channels, and connectivity, governing testing and go-live so nothing breaks on the way, and putting B2B/EDI trading-partner management on a sustainable footing once you're live. Along the way, we flag where standard SAP tooling is enough, and where dedicated migration and testing tooling meaningfully cuts risk and timeline.

Contents

1.	Why migrate to SAP Integration Suite	2
2.	Sizing and planning your migration	2
3.	Architecture & platform decisions.....	4
4.	Executing the migration, object by object.....	5
5.	Governance: workflow, testing, go-live	7
6.	Choosing your migration tooling and partner	9
7.	How Figaf supports your migration	10
8.	Managing B2B & Trading Partner Management.....	11
9.	Conclusion & Next Steps.....	11

1. Why migrate to SAP Integration Suite

PI/PO's end of support is the obvious driver, but SAP has positioned Integration Suite as the successor to PI/PO since 2020. A single cloud-native iPaaS, not a port of PI/PO's own code, covering process integration, API management, and event-driven integration in one product.

- **Cloud Integration:** the direct successor for process integration, covering the large majority of PI/PO scenarios with a modern, flow-based design.
- **API Management:** a capability PI/PO never had natively, adding governance, security, and rate limiting to the interfaces you expose.
- **Advanced Event Mesh:** real-time, event-driven integration, a pattern PI/PO wasn't built for.
- **Integration Advisor:** mapping recommendations for B2B scenarios, based on a shared library of message and mapping guidelines.

According to SAP, consolidating onto one platform means fewer skill sets to maintain, one monitoring view, and lower total cost of ownership. Once you have migrated, you have a modern, supported platform for all your integration needs, one that SAP keeps developing.

2. Sizing and planning your migration

Before touching a single interface, get a realistic picture of the time, money, and staffing your migration needs. Skipping this is the single biggest cause of migrations running over budget and over schedule.

2.1 Define your target integration strategy

A migration produces integrations you will run for years, so decide what the target landscape should look like before you move anything. It is far cheaper to set that direction up front than to reshape iFlows once they are live.

SAP's Integration Solution Advisory Methodology (ISA-M) is a practical starting point. A publicly available, technology-agnostic framework for classifying your integrations and matching each to the right technology. SAP builds it into the Integration Assessment capability of Integration Suite, so your strategy can be documented and governed in the platform you are migrating to.

Working through ISA-M gives you:

- A target pattern and technology for each type of interface, rather than defaulting to "whatever PI/PO did"
- Clear criteria for which interfaces move 1:1, which get redesigned, and which are retired
- Input for your architecture decisions and developer guidelines

An outside perspective helps. Teams who have run PI/PO for years tend to see the new platform through the old one's patterns.

2.2 Assess your current landscape

For a first rough count, look at the ICOs in your production Integration Directory or at the PI/PO message overview. Treat it as a guide only: classical dual-stack scenarios don't appear as ICOs, and rarely used interfaces may fall outside the monitoring data's retention period.

SAP's own migration assessment gives an operational, system-level view. A complementary assessment goes further: it evaluates individual ICOs, flags likely migration obstacles, analyzes message volumes and system load, and supports custom queries against the ESR and Directory. Running both gives a far more complete picture than either alone.

2.3 Run a pilot project

Size the pilot to be manageable but representative, typically 10 to 100 interfaces depending on landscape size. Use your assessment to make sure it covers:

- Message mappings with Function Libraries, and Java mappings
- A representative spread of channel and adapter types, including adapter modules with custom functions
- Routing and multi-flow scenarios
- EDI messages, if applicable

The pilot's job isn't to migrate everything. It's to surface landscape-specific limitations while changing course is still cheap. Compare SAP's tool with a third-party tool like Figaf, replay real production messages against migrated iFlows, and measure how long a batch of interfaces actually takes all the way through transport and configuration in production. Take at least some of them all the way to production, including transport and configuration. That is where migrations most often lose time. Also use the pilot to settle how you will work: error handling and retries, your testing approach, which reference iFlows become templates, and how migrated interfaces will be supported in operation.

2.4 Decide: 1:1 migration or redesign

A **1:1 migration** replicates existing PI/PO structures in Integration Suite: familiar content, mappings, and configurations carry over, tooling automates most of the work, and effort and timeline shrink substantially.

Redesign makes more sense when a message type is fundamentally changing (an IDoc-based interface becoming event-based), when native platform capabilities eliminate a former workaround, when a BPM-style process can consolidate into a single iFlow, when an interface's message volume would drive up Integration Suite licensing costs and a redesign could reduce it, or when the current architecture no longer matches where the organization is heading. A third option is replacing a custom interface with SAP's prepackaged integration content, where the connected applications are covered by standard content.

In practice, the right approach is almost always hybrid: migrate as much as possible 1:1 to control cost and risk, and reserve redesign effort for the areas where modernization delivers real strategic value.

2.5 Estimate the effort and build the business case

With your assessment, pilot results, and 1:1-versus-redesign decisions in hand, you can put numbers on the project.

Estimate the three groups (1:1, redesign, retire) separately, using the per-interface times you measured in the pilot and including development, testing, and transport. Beyond direct effort, count the value of a shorter timeline (less time running PI/PO in parallel, and less need for extended maintenance) and of lower risk at go-live.

[The Figaf migration estimator](#) runs this calculation with your own numbers and compares a manual approach with a tool-assisted one.

3. Architecture & platform decisions

3.1 Redesign iFlows where it adds value

PI/PO often required non-standard flows to support requirements the platform didn't handle natively. Integration Suite's patterns let you simplify several of these:

- ccBPM/BPMN processes used for pure technical orchestration can usually condense into a single iFlow.
- Lookups performed inside mappings to fetch data or set status can often be redesigned more cleanly.
- Custom adapters can sometimes be replaced with SAP or partner adapters from the SAP Business Accelerator Hub, or the integration can be built with the standard HTTP adapter.
- PI-to-PI bridges and similar workarounds should be re-evaluated, not replicated as is.

Check for standard content before you rebuild. The SAP Business Accelerator Hub (api.sap.com) offers prepackaged integration content from SAP and its partners for many SAP and third-party applications. It can replace a custom PI/PO interface entirely, and SAP keeps it up to date. Unedited standard content used for SAP-to-SAP integration also has a licensing advantage: SAP doesn't charge for the messages it processes. An edited standard flow, however, no longer receives SAP's updates, so use the extension options SAP provides rather than modifying it. Even where nothing fits exactly, related packages make useful reference designs.

Finally, ask whether each integrated system still needs integrating. If not, don't migrate that interface, but check whether a new need has emerged that Integration Suite is better placed to support.

3.2 Web dispatcher, reverse proxy and API management

For SOAP or REST interfaces called directly by partners, reconfiguring your Web Dispatcher (or another reverse proxy) to forward traffic to Cloud Integration minimizes friction for partners, though authentication typically has to be re-implemented at the new endpoint. Front every public-facing endpoint with API management: your existing layer if you have one, or SAP's API Management in Integration Suite. It protects your integration systems, gives visibility into partner traffic, lets you throttle to prevent overload, and lets you change the underlying platform later without disrupting partners.

3.3 Security, access, and compliance by design

Security should be designed in from the start, not added once interfaces are running.

User access. Assign access through the standard Integration Suite role collections, apply least privilege, and keep development, test, and production access separate. Review role assignments at every release, because broad rights granted "for now" tend to stay.

Credentials don't migrate. Passwords, keys, and certificates are not carried over from PI/PO, so every authenticated channel needs its security material recreated in the target tenant. Treat this as a separate workstream with its own owner; across hundreds of channels, it is easily underestimated. It is also a good moment to replace basic authentication with certificates or OAuth where partners support it.

Encryption everywhere. Use SFTP or FTPS instead of plain FTP, TLS for all HTTP traffic, and PGP where the payload itself must be protected. On-premise systems connect through SAP Cloud Connector, which needs no inbound firewall ports. Track certificate expiry dates, since an expired certificate is a common cause of sudden outages and be deliberate about where payloads containing personal data are logged.

Compliance. If regulations such as GDPR or PCI DSS apply, confirm early which certifications and data-residency options your SAP BTP region offers.

3.4 Cloud or Edge Integration Cell: choosing where integrations run

PI/PO runs inside your own network, next to the systems it connects, so decide early where each integration should run in the new landscape.

For most scenarios, the cloud runtime is the right default. Latency rarely causes problems, especially if you choose a BTP region close to your data centers, but test time-critical interfaces during the pilot.

Some integrations shouldn't leave your private network: data that may not cross the internet, high-volume on-premise-to-on-premise traffic, and low-latency scenarios such as manufacturing. For these, SAP offers Edge Integration Cell. You design and monitor integrations in the cloud, but run them in your own Kubernetes environment, and SAP explicitly positions it as a path for PI/PO customers who need to keep processing on premise. It is included with most Integration Suite editions, but you operate the cluster yourself, and adapter availability differs between the two runtimes, so check before committing interfaces. The migrated content is the same on either runtime: the choice changes your infrastructure and operating model, not your migration approach.

Edge Integration Cell is also harder to introduce into your organization, and the Kubernetes infrastructure and setup mean significant upfront cost, so it needs a solid business reason. Needing a single adapter that only runs there is rarely enough.

4. Executing the migration, object by object

With planning and architecture settled, execution comes down to a repeatable process applied consistently across ICOs, mappings, channels, and connectivity types.

4.1 ICO migration

Migrating an Integrated Configuration Object means making deliberate choices rather than accepting defaults. Using the Figaf migration tool as an example:

- **Profile selection:** Pipeline All (Recommended for Async interfaces), local Processes or the legacy Single Process option..

- **Channel mapping:** map PI communication channels to their Cloud Integration equivalents including hostname translation between environments.
- **Routing selection:** migrate specific flows selectively, restructuring conditions across separate iFlows as needed.
- **Artifact reuse:** reuse previously migrated mappings and function libraries or regenerate the.
- **Preview and test:** preview before executing, and test immediately afterwards.

Validate every result against the expected outcome and keep manual iFlow adjustments synchronized. Troubleshooting is far cheaper here than downstream.

4.2 Classical & Receiver Determination migration

Classical (dual-stack) objects and Receiver Determinations can migrate directly, without first converting to ICO format. Select the Receiver Determination, review the generated mapping, execute, then migrate the associated mappings and function libraries.

KNOWN LIMITATION: ABAP and ABAP-XSLT mappings are not automatically convertible and require manual re-implementation.

4.3 Build reusable channel migration templates

Simple channels (a SOAP sender, say) often need little more than a credential update. Others need structural change, such as File to SFTP or JMS to AMQP. Rather than solving each from scratch, build a reference iFlow for the target channel type with its parameters externalized, generate a template from it, and use XSLT to populate those parameters from the original PI channel configuration. Add guidance for whoever runs the migration (for example, "set a new password"), test the template against real channel data, and reuse it for every channel of that type. The effort pays off quickly wherever you have many channels of the same kind.

4.4 Common connectivity challenges

Connectivity is consistently the hardest part of a PI/PO migration. Cloud Integration's channels are deliberately simpler, which sometimes pushes former adapter logic into the iFlow itself.

- **File:** SFTP/FTPS via Cloud Connector, File-to-HTTP jobs, a move to JMS/AMQP or HTTP, or cloud storage (SharePoint, OneDrive, S3).
- **JMS:** plan to replace JMS with AMQP, possibly upgrading your broker or adopting Advanced Event Mesh.
- **IDoc:** on the sender side, an IDoc dispatcher flow with value mapping instead of one SM59 connection per IDoc; on the receiver side, router flows or shared connections.
- **Proxy/XI30:** dispatcher iFlows, with separate flows for asynchronous and synchronous traffic; sender and receiver IDs maintained per interface in transaction SXMSIF let you move interfaces over one at a time.
- **REST:** the HTTP adapter needs manual additions for URL-parsing, routing, JSON/XML conversion, and error handling, typically in Groovy.

- **RFC:** RFC-over-SOAP for senders on the cloud runtime, or the native RFC sender adapter on Edge Integration Cell; receivers connect through Cloud Connector. RFC lookups in mappings need their channel set up manually.
- **JDBC:** supported, but queries use an XML structure, and multi-step operations must be designed explicitly; consider exposing the data as a REST or SOAP-service instead.
- **OFTP:** OFTP is supported on Edge Integration Cell, but running Edge Integration Cell for this one adapter adds too much complexity to your landscape (Section 3.4). A simpler option is a separate OFTP service that passes messages on to Cloud Integration, where they reach your TPM flows through ProcessDirect.
- **Custom adapters:** rebuild as standard flows using HTTP calls or bring in SAP or a specialist partner.
- **Adapter modules:** no direct equivalent; move the logic into Groovy scripts in shared script collections.

4.5 Migrate file channels with content converters

SAP Cloud Integration's CSV-to-XML converter works differently from PI/PO's content conversion and requires an XSD, which can force changes to file structures and downstream mappings. Figaf comes with a 95% compatible Content converter, which simplifies the migration.

4.6 The risk of modifying ICOs before migration

Pre-modifying ICOs in the source system to work around a migration tool's limitations is riskier than it looks. In our experience, there is a real chance that something gets missed and is accidentally transported to production alongside the change, on top of the cost of rolling the modification back and the extra testing and documentation. Wherever possible, choose a migration approach that lets you select which part of an integration to migrate without touching the source system at all.

5. Governance: workflow, testing, go-live

5.1 Establish your migration workflow

Plan the migration as a series of releases rather than one big cutover. Keep the first release small: a handful of interfaces covering different patterns, ideally between internal systems, where you control both ends and can switch back quickly if something doesn't work as expected (and something will). Shape later releases around risk and around how changes normally flow through your landscape.

Migrate one ICO at a time, especially early in the project. Batch-migrating 10 to 100 ICOs is technically possible, but each iFlow still needs individual review, and if a systemic problem surfaces partway through, everything migrated before the fix may need to be redone.

Everybody can migrate to iFlows. You only get the credit for it once you've actually tested the integration.

Every migrated interface still needs hands-on work before it is done. Configuration, credentials, mapping fixes, testing, and documentation. Full end-to-end automation of this workflow isn't realistic yet, so plan for human oversight.

5.2 Set guidelines and build a shared repository

Agree on developer guidelines before migration work scales beyond the pilot: naming conventions, how each integration pattern is built, what must be tested before an interface moves on, and how credentials and encryption are handled. Cloud Integration's built-in design guidelines check iFlows against SAP's recommended practices, but your own conventions still need to be written down and enforced.

Create a central repository for artifacts you intend to reuse, such as Groovy scripts, standard XSDs, and shared logic, and track every place each one is used, so updates propagate consistently and long-term maintenance stays manageable.

5.3 Testing strategy

Testing is where most real migration effort goes, and it's non-negotiable for a true 1:1 result: UDFs must produce the same result, and subtle behavior differences need to surface before go-live.

Collect test data directly from your production PI/PO system, so you test real message shapes and edge cases rather than synthetic ones. Scale effort to risk: simple mappings need little testing, while EDI structures and complex mappings warrant much more. Automated regression testing (replaying real PI/PO input messages through Cloud Integration and comparing the output with PI/PO's) is the most efficient way to validate at scale and can substantially reduce how much testing needs to happen with partners.

5.4 Transport and change management

"Export the iFlow" sounds like a two-minute job, but what travels with it is harder, and transport is the step most migration estimates leave out. An iFlow depends on script collections, message mappings, value mappings, and function libraries, and it only works once its externalized parameters (endpoints, hostnames, paths) are set for the target environment and its security material exists there. None of that environment-specific configuration travels with the content. Miss one dependency and the import fails, or worse, the interface breaks at runtime.

SAP supports transport through SAP Cloud Transport Management with the Content Agent service, through CTS+, or by manual export and import. Whichever you choose, put these rules in place before the first migrated interface leaves development:

- **Make dependencies visible before transport**, so nothing is discovered missing after import.
- **Maintain environment-specific configuration systematically**, not by hand per tenant, so that development, test, and production values can't be mixed up.
- **Let approvers see both content and configuration** before signing off a move to production.
- **Link every change to a ticket and a version**, so you can see exactly what was shipped, when, and why.

Figaf Simplifies the transport process so you get more control of what is going on and reducing the effort of handling the migration.

5.5 Plan and execute go-live

Choose a go-live window outside peak hours so you can observe performance. An SFTP file interface can typically switch whenever convenient, while multi-partner interfaces are often safer switched one partner at a time.

Communicate with system owners and partners ahead of time: IP ranges, server addresses and credentials, expected performance changes, the go-live date, and the rollback procedure. Some partners are harder to reach than others, so factor that into your timeline.

Have a concrete rollback plan: who gets contacted, and exactly what switches the interface back to PI/PO. If you need it, use it before the peak period, not during it. On the day, activate the interface in Integration Suite, disable the old adapters, monitor closely, and confirm with the business that the outcome matches expectations.

5.6 Learn and repeat

After each release, capture what you learned about how the interfaces actually behave and what could be automated, then apply it to the next wave. Running and monitoring integrations on SAP Integration Suite is different from PI/PO, so use each release to adjust your operating procedures and support model, not only your development approach.

Checklist: the life of one migrated interface

1. **Understand:** pattern, and whether to migrate 1:1, redesign, replace, or retire
2. **Migrate:** routing, mappings, function libraries, and channels, from your templates
3. **Configure:** adapter settings, and security material in each tenant
4. **Fix:** message mappings, UDFs, and scripts
5. **Test:** real production messages, automated regression, and business tests where needed
6. **Document:** what changed, and which tests prove it works
7. **Transport:** through test to production, with dependencies and configuration checked
8. **Go live:** low-traffic window, partners informed, rollback plan ready
9. **Evaluate:** lessons learned into the next release

6. Choosing your migration tooling and partner

6.1 Comparing migration tools

SAP's migration tool generates iFlows from predefined patterns. You can create one iFlow per scenario, or use SAP's pipeline approach, which recreates PI/PO-style processing with a set of generic, reusable iFlows. Recent releases added support for Receiver Determinations from PI 7.5 dual-stack systems, and reuse of function libraries and message mappings.

The Figaf DevOps Suite takes a different route. It models the routing structure of each ICO or Receiver Determination and generates a dedicated local process for each receiver and receiver interface, translating XPath conditions and adding built-in retry for failed messages. In practice, that gives you:

- **Broader source-system support:** ICOs and dual-stack objects from PI 7.11 through 7.5, so you can migrate without upgrading PI/PO first.
- **Mapping clean-up:** mapping parameters and constants that standard Cloud Integration mappings don't support are updated during migration, and function libraries can be migrated either as SAP standard function library objects or through Figaf's Groovy conversion.

- **Configurable channel migration:** XSLT-based templates define exactly how each channel type converts, for example File to SFTP or FTPS.
- **Special cases handled:** file content conversion using the same configuration as PI/PO (Section 4.5), and a B2B mode that migrates existing B2B interfaces without rebuilding mappings.

Neither tool handles every scenario best: some interfaces migrate faster with SAP's tool, some more cost-effectively with Figaf, and some should be redesigned instead. The most reliable way to choose is to pilot both against a representative set of your own scenarios and compare the results.

6.2 Choosing an implementation partner

A migration is a large project to run alongside day-to-day integration work, so bringing in a consultancy or freelancers can make sense. When you evaluate a partner, ask:

- **Experience:** How many PI/PO migrations have they delivered, and of what size and complexity compared with yours?
- **Automation:** Which tools will they use, and have they compared SAP's tool with alternatives on real scenarios?
- **Testing:** How will they prove that each migrated interface behaves like the original? Do they test against real production messages?
- **Handover:** How will they document the result and transfer knowledge to your team?

Even with a partner, stay closely involved. You will operate these integrations for years, so your team needs to understand every design decision.

7. How Figaf supports your migration

The Figaf DevOps Suite was built to remove manual effort from PI/PO to Integration Suite migrations, while keeping humans in control of the decisions that matter. Its migration tooling started with SAP itself: in 2021, Figaf won SAP's Hack2Build, a competition sponsored by SAP to automate PI/PO migrations, and the winning prototype has since been developed with customers and partners in real migration projects. In Figaf customer projects, automating assessment, migration, testing, and transport has reduced overall migration effort by at least 30%.

- **Assessment and estimation:** a landscape and ICO assessment report, plus a migration estimator projecting time, cost, and savings versus a manual approach
- **Automated migration:** ICO, classical, and Receiver Determination migration, with automated channel mapping and content conversion (Section 6.1)
- **Automated testing:** real input and output messages captured from PI/PO, replayed against the migrated iFlows, and compared automatically, turning weeks of manual comparison into a repeatable process
- **Governance:** tracking of every place a shared artifact is used, plus change management, versioning, and documentation, with tickets linked to your ITSM tool, such as Jira or SAP Cloud ALM
- **Transport:** dependencies shown before anything moves, externalized parameters translated automatically between environments, and import, configuration, and deployment in a single step

The Suite installs locally for a quick proof of concept, or on SAP BTP for a full project rollout, so no migration budget goes on setting up infrastructure.

8. Managing B2B & Trading Partner Management

Migrating your core interfaces is only part of the story. Once you're live, B2B and EDI trading-partner integration is one of the areas where SAP's own tooling leaves the most structure up to you, and where that gap tends to show up as cost later rather than sooner.

8.1 Why native TPM struggles at scale

SAP's Trading Partner Management (TPM) framework is close to a blank canvas: ten partners built by hand is manageable, but a hundred across three environments is not. There is no out-of-the-box best practice, so most organizations invent one under time pressure, and correctness ends up living in specialists' heads and spreadsheets. Deviations then surface in production instead of before transport.

8.2 What Figaf B2B Management adds

Figaf B2B Management supplies the structure SAP leaves open, without replacing SAP as the source of truth. Standard SAP objects stay authoritative, governance sits around them rather than in a parallel runtime, and the add-on can be removed at any point without losing the underlying configuration. The result is a setup that scales, with the hundredth partner costing little more than the tenth, and that behaves the same regardless of who built it.

8.3 Core capabilities in Figaf B2B Management

- **Implementation accelerators:** partner profiles, agreements, templates, and scenarios created in bulk, with one convention enforced, plus scope builders for MIGs, MAGs, and agreement templates. Where standard TPM has you build each partner through roughly seventeen screens, Figaf collapses that into a single guided step.
- **Legacy SAP PI B2B:** existing PI/PO partner data, mappings, and test messages reused for a gradual transition instead of a rewrite, with a result that follows Integration Suite TPM best practice.
- **Change and transport:** versioned change sets moving through tickets and transports, dependencies visible before a change moves, and production import blocked until the gap report is clean.
- **Test, validate, and operate:** regression test cases per agreement built from real messages, a rules engine covering naming, identifiers, versions, and environments, one governed view across tenants, plus EDI monitoring and searchable archiving of interchanges.

In a shared TPM landscape, one partner's change can affect objects that other partners' live traffic depends on, and the usual response is to slow releases down. Figaf's approach aims instead to make releases safe enough to stay frequent, with humans staying in the approval loop as more of the work moves to guided automation.

9. Conclusion & Next Steps

A PI/PO to SAP Integration Suite migration is rarely just a technical lift-and-shift. It's a chance to simplify architecture, retire integrations that no longer earn their keep, and put a modern testing and governance process

in place. Whether you handle the migration in-house or bring in outside support, the framework in this guide should help you plan a migration that is predictable, well-tested, and low risk.

9.1. Get started

Try the free Figaf Migration Edition: figaf.com/migrationedition

The Migration Edition of the Figaf DevOps Suite is free for up to 12 months and includes the assessment and migration dashboard, a trial migration of up to 10 ICOs, and 30 days of automated testing. No credit card required.