

SAP PI/PO to CPI Migration Technical Reference: 10 Differences That Affect Real Projects

- ***Checklist by Prathmesh Sutar***

A plain-language reference for integration architects and SAP Basis teams planning or mid way through a Process Orchestration to Cloud Integration migration. Each point covers what changes, why it changes, a real business scenario to explain the impact, and how to handle it.

Based on SAP's official migration FAQ and public technical documentation. Compiled by Prathmesh Sutar. SAP Integration. Sources referenced throughout personal experiences, not proprietary or client-specific content.

SAP PI/PO to CPI Migration Technical Reference: 10 Differences That Affect Real Projects.	1
01 Message and Payload Logging Is Not on by Default LOGGING WHAT CHANGES	2
02 Custom Adapter Modules Do Not Exist in CPI.....	3
03 BPM and ccBPM Cannot Be Migrated Directly	4
04 Value Mappings Need an Extra Deployment Step	5
05 Dynamic Queues Do Not Work the Same Way.....	5
06 Java Mappings Must Be Rewritten in Groovy	6
07 ABAP Proxy Interfaces Must Be Re Pointed.....	7
08 B2B and EDI Support Is Still Catching Up	7
09 Licensing Changes From Server Based to Message Based.....	8
10 The Migration Tool Does Not Do Everything	9
How to use this reference	9

01 Message and Payload Logging Is Not on by Default

LOGGING WHAT CHANGES

PI/PO saves a copy of every message automatically. CPI does not do this by default. It saves the status and headers, but not the full message content, unless someone turns this on.

WHY THIS HAPPENS This is a cloud design choice. Storing every message costs money and raises data questions in a shared cloud service. On premise, this was less of a concern.

REAL BUSINESS SCENARIO A support engineer gets a call. An order did not reach the customer's system. In PI/PO, they open the message log and see exactly what was sent. In CPI, if logging was not switched on, there is nothing to look at. What should take five minutes now takes days, and the business team is asking for answers the whole time.

HOW TO HANDLE IT Turn on message logging where it matters and add extra logging steps in the flow for interfaces that need strong traceability. Decide this while building the flow, not after something breaks.

02 Custom Adapter Modules Do Not Exist in CPI

WHAT CHANGES

PI/PO lets you build custom adapter modules, extra Java code added to the adapter layer to handle special cases. CPI has no matching feature. It is not hidden somewhere else. It simply is not there.

WHY THIS HAPPENS

CPI runs on a different, cloud based engine. It was not built with the same low level extension points that PI/PO's on premise engine allowed.

REAL BUSINESS SCENARIO

A project team quotes ten weeks for the migration, based on a quick look at the interface list. Six weeks in, they find that twelve interfaces use custom adapter modules. Each one now needs to be rebuilt using a different method, not just copied. The go live date moves, and someone has to explain the delay to the steering committee.

HOW TO HANDLE IT

Check every interface for custom adapter module usage during the assessment phase, before the timeline is agreed. This single check prevents most of the scope surprises seen on these projects.

03 BPM and ccBPM Cannot Be Migrated Directly

WHAT CHANGES

If you used ccBPM or BPM in PI/PO to manage a multi step process, that process cannot be copied into CPI. It has to be rebuilt.

WHY THIS HAPPENS CPI

does not have a central process engine like PI/PO did. CPI flows handle messages step by step, not full business processes with waiting steps and decisions the way BPM did. REAL

BUSINESS SCENARIO

A manufacturing company uses ccBPM to collect purchase order confirmations from three systems before releasing one combined message to finance. This logic cannot be exported into CPI. A developer has to sit down and design it again from scratch, using a different approach. This is real design work, not a file import.

HOW TO HANDLE IT

Treat every BPM or ccBPM process as a redesign task from the very start of project planning, with its own time and budget, not as a simple migration item.

04 Value Mappings Need an Extra Deployment Step

WHAT CHANGES

Value mappings, the tables that convert one code to another, for example a country code or a plant code, cannot be moved into CPI the same way as other objects. They are handled separately.

WHY THIS HAPPENS

In CPI, value mappings are updated through a technical interface called an API. Updating them this way only changes the draft version. A separate step is needed to push that change live.

REAL BUSINESS SCENARIO

A team updates a value mapping to add a new plant code before a product launch. In PI/PO, this would apply right away. In CPI, they update it through the API and assume it is done. Nobody deploys the change to production. On launch day, orders for the new plant fail, because the system is still using the old mapping.

HOW TO HANDLE IT

Always treat a value mapping update as two steps: update it, then deploy it. Add this as a specific line item in your cutover checklist so it is never forgotten.

05 Dynamic Queues Do Not Work the Same Way

WHAT CHANGES

In PI/PO, you could set up a queue name automatically, based on the data in the message, so that each partner or order type gets its own queue. In CPI, queue names must be fixed in advance.

WHY THIS HAPPENS

CPI's queue system is built differently. It does not support working out a queue name while the message is being processed, the way PI/PO did.

REAL BUSINESS SCENARIO

A retailer processes orders from fifty different store locations, each needing to stay in the exact order they arrived. In PI/PO, each store automatically gets its own queue. In CPI, this is not possible in the same way. If the team uses one shared queue instead, a single failed order from one store can block every other store's order behind it in the queue.

HOW TO HANDLE IT

Find every interface using this kind of setup early and decide case by case whether a shared queue is acceptable, or whether the design needs to change.

06 Java Mappings Must Be Rewritten in Groovy

WHAT CHANGES

Complex Java based mappings built in PI/PO do not carry over into CPI as they are.

WHY THIS HAPPENS

SAP's own migration guidance says to rewrite Java mappings using Groovy scripts, which is the scripting language CPI is built around, instead of trying to run the old Java code.

REAL BUSINESS SCENARIO

A company built fifteen complex Java mappings over ten years, each with its own business rules. Someone assumes this is a simple copy job. It is not. Each mapping has to be read, understood, and rewritten by a developer. This is real development work and needs to be budgeted and planned like any other build task, not treated as a quick technical step.

HOW TO HANDLE IT

List every Java mapping and how complex it is before setting the project budget. Use SAP's Groovy script templates as a starting point for the rewrite.

07 ABAP Proxy Interfaces Must Be Re Pointed

WHAT CHANGES

Interfaces built using ABAP Proxies point to PI/PO's central engine by default.

WHY THIS HAPPENS

CPI does not have that same central engine. The proxy connection has to be changed so it points to CPI's adapter engine instead.

REAL BUSINESS SCENARIO

A company goes live on CPI over a weekend. On Monday, invoices stop reaching the finance system. The proxy interface was never re pointed, so it is still trying to reach the

old, switched off PI/PO system. Nobody notices until finance asks why invoices did not arrive.

HOW TO HANDLE IT

List every ABAP proxy interface during the assessment phase, and add the re pointing step as its own task in the cutover plan, checked off before go live, not after.

08 B2B and EDI Support Is Still Catching Up

WHAT CHANGES

CPI's tools for managing trading partners and EDI messages are newer and do not yet do everything PI/PO's B2B tools could do.

WHY THIS HAPPENS

SAP built CPI's B2B capability more recently, and it is still being developed. There is a tool called Intelligent Content Advisor to help speed up new B2B mapping work, but it does not close every gap yet.

REAL BUSINESS SCENARIO

A distribution company has two hundred trading partners connected through EDI. The team assumes this will migrate like any normal interface. It does not. EDI and partner management turn out to be the single biggest and most expensive part of the entire project, far more than the standard interfaces anyone was worried about at the start.

HOW TO HANDLE IT

Treat B2B and EDI interfaces as their own project stream, with a dedicated review and timeline, separate from the rest of the migration.

09 Licensing Changes From Server Based to Message Based

WHAT CHANGES

PI/PO licensing is based on server capacity. CPI licensing is based on how many messages flow through it.

WHY THIS HAPPENS

This follows the move from owning your own servers to paying for a cloud service. The cost model changes because the ownership model changes.

REAL BUSINESS SCENARIO

A company has one very high volume interface that barely mattered under the old licensing, since the cost was fixed regardless of volume. After moving to CPI, that same interface drives a large part of the monthly bill, because cost now follows message count. The IT budget owner is caught off guard by an invoice nobody planned for.

HOW TO HANDLE IT

Estimate expected message volume for your biggest interfaces before the budget is agreed, not after go live.

10 The Migration Tool Does Not Do Everything

WHAT CHANGES

SAP provides an official Migration Tool, but it only supports a defined set of adapters and patterns. It does not handle every scenario automatically.

WHY THIS HAPPENS

Complex cases such as BPM, ccBPM, custom adapter modules, and certain mapping functions sit outside what the tool can process on its own, and need manual work instead.

REAL BUSINESS SCENARIO

A project team runs the Migration Tool across their whole interface list and assumes the output is close to finished. Later, they discover that around one fifth of their interfaces need manual rework the tool never flagged. That work was never in the original project plan or budget.

HOW TO HANDLE IT

Use the tool's output as a starting point only. Manually check every interface and rate it as high, medium, or low complexity before locking in your project estimate.

How to use this reference

This covers ten of the most common differences between PI/PO and CPI. It is not everything. Every company's landscape is different, and the real risk points in your case depend on your interface count, your use of custom adapters, your B2B setup, and your use of BPM or ccBPM. All ten points come from SAP's official migration documentation and public technical sources, not from one specific client project. This is a general reference, not proprietary analysis. If it would help to walk through how these apply to your own PI/PO landscape, feel free to reply to whichever message brought you here. Happy to compare notes.

Compiled by Prathmesh Sutar. SAP Integration, PI/PO to CPI. Views shared here are based on SAP's official documentation and public technical sources and personal learning experiences

Regards,
Prathmesh Sutar
prathmesh.sutar@technolize.in | +91 9834129332