



# Migration Guide for SAP Process Orchestration

Generated on: 2023-06-19 08:57:28 GMT+0000

| SHIP

**PUBLIC**

Original content: <https://help.sap.com/docs/help/90c8ad90cb684ee5979856093efe7462?locale=en-US&state=PRODUCTION&version=SHIP>

## Warning

This document has been generated from the SAP Help Portal and is an incomplete version of the official SAP product documentation. The information included in custom documentation may not reflect the arrangement of topics in the SAP Help Portal, and may be missing important aspects and/or correlations to other topics. For this reason, it is not for productive use.

For more information, please visit the <https://help.sap.com/docs/disclaimer>.

# Migration Guide for SAP Process Orchestration

Use this guide to migrate from the on-premise solution SAP Process Orchestration to the cloud solution SAP Integration Suite.

## When Should You Use This Guide?

This guide addresses customers of SAP Process Orchestration who want to move to the SAP Integration Suite. Here you can find all information needed when preparing for the migration, and during the migration process itself.



## What Is This Guide About?

This guide describes what you need to know when you want to migrate from SAP Process Orchestration to SAP Integration Suite. It covers a variety of topics related to the migration process:

- [Introduction to SAP Integration Suite](#): Before you start your migration, get to know SAP Integration Suite, learn about its advantages, and get an overview of the migration process from SAP Process Orchestration to SAP Integration Suite.
- [Landscape](#): Assess your existing integration landscape and plan your target landscape.
- [Connectivity](#): Learn how to move your connectivity options to the connectors and adapters of the SAP Integration Suite.
- [Security](#): Get an overview of the security aspects you must consider while migrating and how to manage them in SAP Integration Suite.
- [Error Handling and Logging Strategy](#): Learn about cloud-based error handling and logging strategies and understand the different approaches available to you.
- [Interface Governance](#): Understand how interfaces are governed in the cloud and how to manage them across multiple landscapes.
- [Interface Migration](#): Discover the different aspects involved in moving interfaces from SAP Process Integration and SAP Process Orchestration to SAP Integration Suite, as well as scenario and object assessment and how testing can be automated.



### → Tip

This guide is open for contributions and feedback using GitHub. This allows you to get in contact with responsible authors of SAP Help Portal pages and the development team to discuss documentation-related issues. To contribute to this guide, or to provide feedback, choose the corresponding option on SAP Help Portal:

-  *Edit*: Contribute to a documentation page. This option opens a pull request on GitHub.
-  *Feedback*: Provide feedback about a documentation page. This option opens an issue on GitHub.

You need a GitHub account to use these options.

More information:

- [Contribution Guidelines](#)
- [Introduction Video: Open Documentation Initiative](#) 
- [Blog Post: Introducing the Open Documentation Initiative](#) 

# What's New

Find out what's new in the migration guide for SAP Process Orchestration.

## May 2023

Section	Update
<a href="#">Migrating Proxy Interfaces</a> <a href="#">Setting Up XI Proxy Runtime</a> <a href="#">Message Reprocessing</a>	There's now additional information available on the migration of proxy interfaces, such as known limitations, the procedure of setting up the XI proxy runtime, and general information on message reprocessing.

## February 2023

Section	Update
<a href="#">Interface Migration Strategy</a> <a href="#">Scenario and Object Assessment</a> <a href="#">Interface Design and Implementation</a>	The capability Migration Assessment and the migration tooling for Cloud Integration are now available in SAP Integration Suite. Both support you in your migration journey to the cloud. See <a href="#">Assessing the Migration Strategy and Migrating Content</a> in the documentation for SAP Integration Suite.
<a href="#">Directory API</a>	Learn about the APIs Migration Assessment and Migration Tooling use to extract all necessary data for the migration from SAP Process Orchestration.

## July 2022

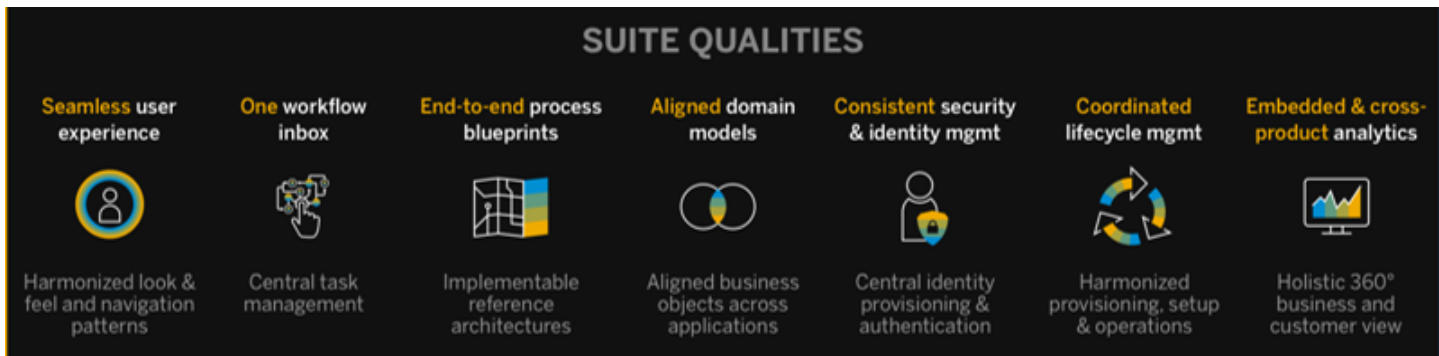
Section	Update
<a href="#">Migrating ESR Proxies</a>	In the section Interface Migration, you can now find information on how to migrate ESR proxies from the Enterprise Services Repository to the Metadata Repository.

# Introduction to SAP Integration Suite

Get to know SAP Integration Suite and its main capabilities, Cloud Integration, API Management, Open Connectors, and Integration Advisor.

## SAP Integration Suite: A Future-Proof, Hybrid Integration Platform

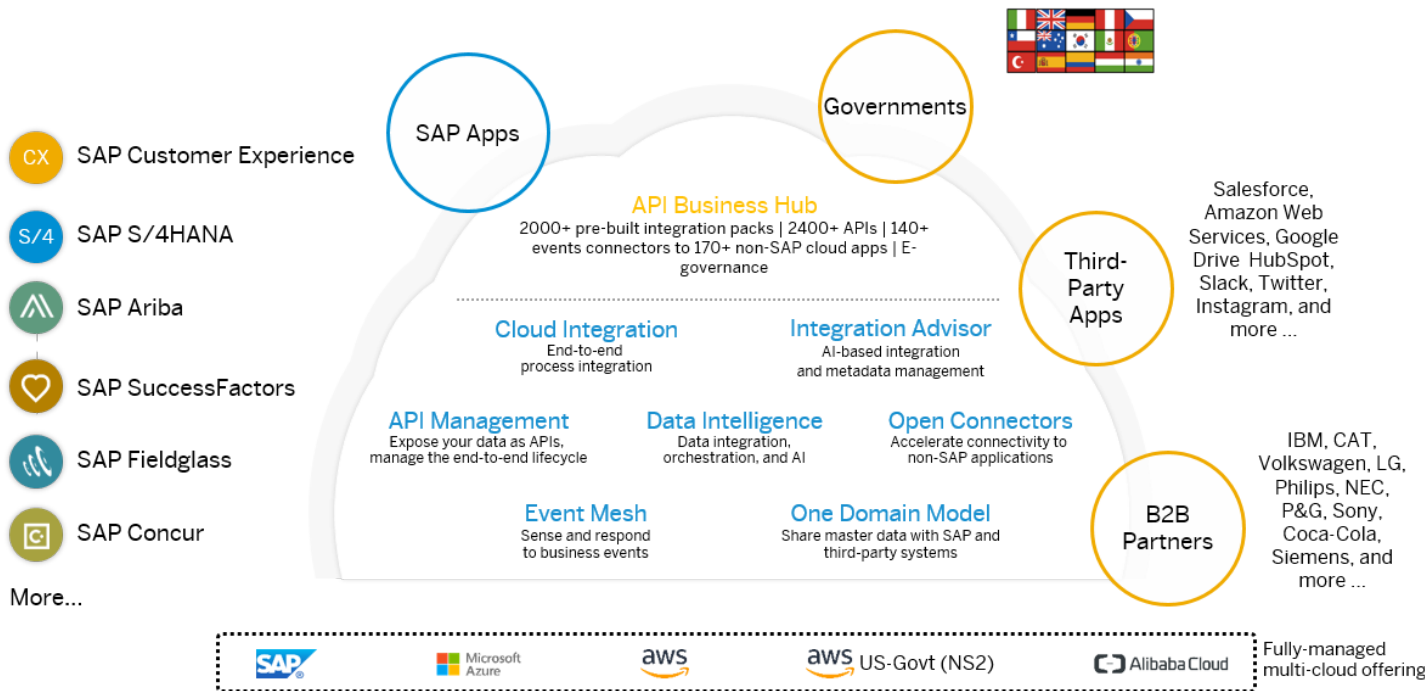
SAP Integration Suite is an open and enterprise-grade integration platform that is a fully vendor-managed multi-cloud offering. It's SAP's recommended integration platform for simplifying and accelerating integrations for both SAP and third-party integration scenarios. SAP Integration Suite supports integration in a holistic manner and provides prebuilt integration packages to enable accelerated integrations. It supports a range of integrations, including the integration of applications in the cloud and on premise, master data integration, B2B and electronic data interchange (EDI) integration, business-to-government (B2G) integration, data integration and pipelines, event-driven integration, and integration based on application programming interfaces (APIs). It also supports design and runtime governance of APIs and acts as the foundation for process excellence across the value chain. In summary, you have all the features necessary to build an Intelligent Enterprise.



**Key Highlights**

- Leader in the Gartner 2020 Magic Quadrant™ for iPaaS solutions
- Simplified integrations with capabilities of Cloud Integration, API Management, Integration Advisor, AI-based development, SAP Integration Solution Advisory Methodology, the SAP Business Accelerator Hub, Cloud Connector, and Open Connectors for more than 170 non-SAP applications
- Time-to-live is accelerated with 2200+ pre-built integrations, 2500 APIs, 160 business events
- Modernization of integrations supported integration use cases/styles: cloud to cloud, cloud to ground, A2A, B2B/G, API-driven, event-driven
- Fully capable feature set that can support cloud-to-cloud and hybrid integrations

**Capabilities of SAP Integration Suite**



SAP Integration Suite has four main capabilities: Cloud Integration, API Management, Open Connectors, and Integration Advisor.

**Cloud Integration**

The Cloud Integration capability within SAP Integration Suite supports end-to-end process integration across cloud-based and on-premise applications through the exchange of messages, from to cloud to on-premise and vice versa. It provides capabilities to process messages in integration scenarios spanning different companies, organizations, or departments within an organization.

Integration developers can use predefined integration content available on [SAP Business Accelerator Hub](#) out of the box, enhance it, or develop their own integration content from scratch. Integration content refers to all design artifacts that define how a message is to be processed during an integration scenario. Cloud Integration provides a [set of adapters](#) that allow you to specify a connection type and define, for example, which technical protocols should be used to connect a sender or a receiver system to the tenant and how this connection is protected.

Since Cloud Integration covers the majority of the capabilities provided by SAP Process Orchestration system, it's addressed as main service when planning to migrate to SAP Integration Suite.

### [API Management](#)

The SAP API Management capability allows you to publish, promote, oversee, and secure APIs in a scalable environment. SAP API Management technology helps you share digital assets and enable developer communities to consume these assets in new channels, devices, and user interfaces.

### [Open Connectors](#)

A third key capability is SAP Open Connectors, which provides out-of-the-box capabilities to connect to over 170 non-SAP services and applications on the market today. The number of standard connectors is growing and can be supplemented by the implementation of bespoke connectors. Once a particular Open Connector instance has been configured, it can be consumed in a homogeneous way within either Cloud Integration or API Management.

### [Integration Advisor](#)

The final key capability is SAP Integration Advisor, a machine learning-based design tool that can be used to accelerate the implementation of mappings between different source and target message structures. The tool uses a knowledge graph that tracks past message structure definitions and how they were previously mapped. Using machine learning algorithms, the knowledge graph can propose mapping solutions for new mapping requirements to accelerate the design and development process. The capability is particularly useful for B2B scenarios but is also applicable for use in other areas such as A2A and B2G.

## Value Proposition: Why You Should Move to SAP Integration Suite

SAP Integration Suite is a multi-cloud SaaS solution that offers a number of significant advantages over an on-premise offering like SAP Process Integration and SAP Process Orchestration. Here are a few significant points that highlight the value proposition of SAP Integration Suite compared to SAP Process Orchestration.

SAP NetWeaver Process Integration and Process Orchestration is an on-premise integration platform hosted on customer-managed infrastructure. It offers a service-oriented architecture and was one of the most widely adopted integration platforms. However, modern day IT landscapes are evolving towards a cloud-first approach. They're leveraging SaaS offerings to offload IT maintenance efforts and focus on solving business challenges and innovating on customer experience. In this digital transformation journey, it's important that the integration platform is also a cloud native solution that can also support cloud (cloud → cloud) and hybrid (cloud ↔ on-premise) integration scenarios.

Integration platforms play a strategic role for businesses coping with heterogeneous and hybrid IT landscapes. Seamless integration across applications and systems — whether they're SAP, non-SAP, on-premise, or cloud — enable organizations to digitize their end-to-end processes and make the enterprise more connected and intelligent. Forward-thinking organizations are taking a holistic and strategic approach to their enterprise integration initiatives as they pursue digital transformation, embracing the agility and simplicity of modern cloud-based integration platforms as they move to more intelligent solutions such as SAP S/4HANA and SAP's Intelligent Suite. The right integration platform can help businesses achieve end-to-end process excellence across their systems, simplify connected experiences for customers, build business-to-business (B2B) networks and digital ecosystems, and innovate with new business models, while also bringing along the best of their existing integration investments.

SAP Integration Suite is a future-proof, cloud native, fully managed integration platform as a service (iPaaS) that can help in accelerating digital transformation journeys. It supports all types of modern-day integration patterns, helping customers build an enterprise IT landscape that transforms the organization into an Intelligent Enterprise.

For more information, see [What Is SAP Integration Suite?](#)

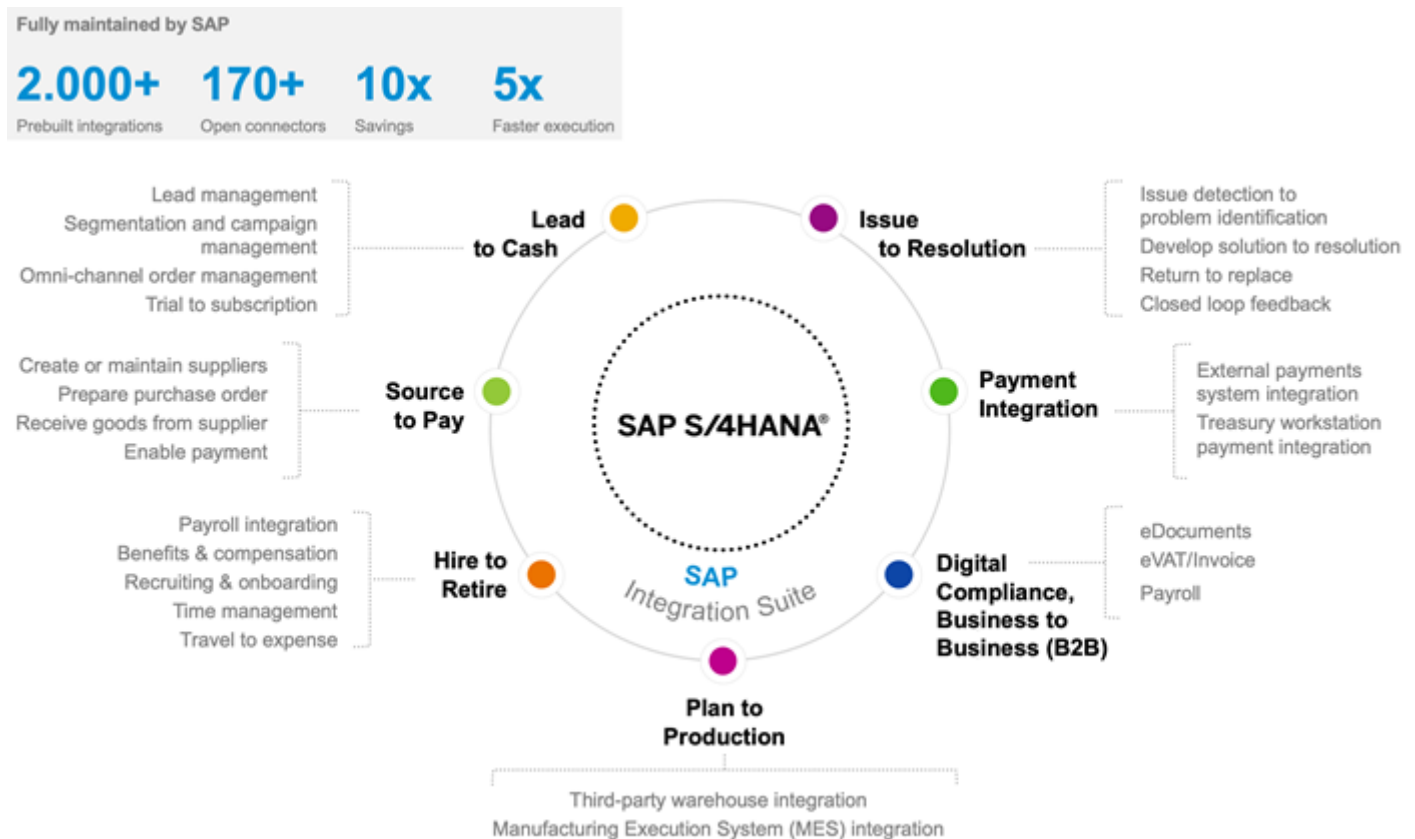
## Advantages of SAP Integration Suite

### Faster time to value

SAP Integration Suite is a fully managed SaaS offering that can help you get started with no intervention from the service provider. It also offers 2000+ prebuilt integrations, 2500 APIs, and 150 business events on SAP Business Accelerator Hub, and this list keeps growing. You can quickly get started and implement your integration projects with a combination of these features, helping you generate value in less time. In line with SAP's motto of "Run Simple", SAP Integration Suite aims to "simplify integration" as much as possible by democratizing integration using the low-code no-code approach, an intuitive graphical user interface, and intelligent technologies to improve productivity. The overall learning curve for SAP Integration Suite is significantly smaller compared to other integration solutions owing to the innovative and intuitive features it offers.

### Accelerated integration with 2000+ integration packages

SAP Integration Suite provides prepackaged business content for different line of business scenarios on [SAP Business Accelerator Hub](#) without any additional costs. These integration packages contain integration flows, message and value mappings, and scripts among other integration artifacts that can accelerate your integration projects and reduce the overall implementation time. This also results in significant cost savings, with some customers reporting as much as 60% cost savings. SAP also manages the lifecycle of these integration packages, providing updates and patches that can be consumed with a few clicks.



### Lower cost of initial deployment and ongoing cost of running the solution

Since SAP Integration Suite is hosted on the SAP Business Technology Platform (BTP), it's fully managed by SAP. You can easily get started with SAP Integration Suite as it's offered in flexible licensing models like CPEA (Cloud Platform Enterprise

Agreement) and PayG (Pay-As-You-Go). With minimal upfront effort, you have an enterprise-grade integration platform that you can leverage to implement any modern-day integration pattern. Messages are relatively inexpensive and offered in a [tiered pricing model](#) that offers a lower price per message block as the message consumption increases. This pricing model offers greater value over a flat pricing model where customers with high message loads don't get any benefit.

### Ability to absorb innovation delivered by SAP at a faster rate

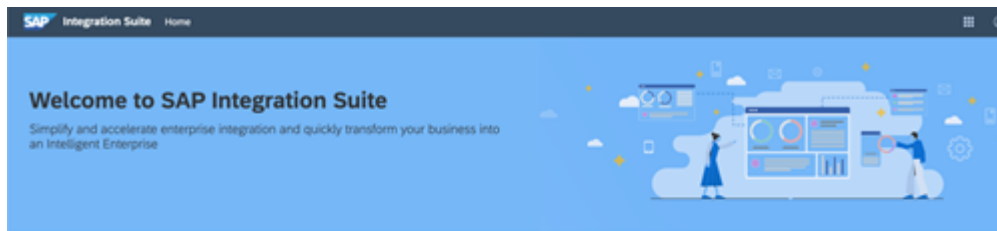
SAP fully manages and operates SAP Integration Suite, helping you focus your efforts on leveraging the service to solve your organization's problems. With no weekly downtime for updates and upgrades, you can be sure that your business-critical processes are not affected during the maintenance windows. The monthly release cycle ensures that you will receive updates and upgrades on a regular basis without any impact on your existing scenarios, helping you to improve and innovate continuously. SAP Integration Suite is also the future-proof and recommended solution from SAP where all new innovations and features will be delivered. Here's a preview of some of the features that will be delivered only on SAP Integration Suite:

- **Future-proof** your investments and move to **multicloud** (Amazon Web Services, Azure, Ali Cloud, Google Cloud) deployments.
- **Unified commercial offering and harmonized metering**
  - One license and metric for all capabilities of SAP Integration Suite
  - Free SAP-to-SAP messages via standard prepackaged content shipped by SAP via SAP Business Accelerator Hub
  - Easy scalability with instances and messages. No limitation on "Connections or bandwidth"
  - Open Connectors (170+ connectors for third-party integrations)
  - Full lifecycle API management for modern API-first strategy
  - Unlimited Async Message Queues (required for reliable delivery, e.g. if the target system isn't available)
  - Integration Advisor (60–70% faster implementation) and bundled B2B Libraries

### Lower risk of deployment and get started easily

Getting started with SAP Integration Suite has never been easier. Once you license SAP Integration Suite, you can leverage the self-service to provision your tenant and get started quickly. The entire process of provisioning a tenant takes less than 4 hours, reducing the deployment risk and helping you focus on implementing integration scenarios. You can easily set up a global integration landscape in multiple regions within a few hours. The required capabilities can be activated separately, as and when required. The common launchpad acts as the 'home page' for SAP Integration Suite, helping you seamlessly switch between capabilities.

For detailed instructions on how to quickly get started with SAP Integration Suite, see [Initial Setup - SAP Integration Suite](#).



### Capabilities

Develop and manage enterprise-wide integration across heterogeneous landscapes.

<p><b>Design, Develop, and Operate Integration Scenarios</b></p> <p>Cloud Integration</p> <p>Support end-to-end process integration through the exchange of messages.</p>	<p><b>Design, Develop, and Manage APIs</b></p> <p>API Management</p> <p>Create simple digital experiences for your consumers, partners, and employees.</p>	<p><b>Extend Non-SAP Connectivity</b></p> <p>Open Connectors</p> <p>Accelerate the connectivity with non-SAP cloud applications.</p>	<p><b>Implement Interfaces and Mappings</b></p> <p>Integration Advisor</p> <p>Develop interfaces and mappings by using crowdourcing and machine learning.</p>
<p>Manage Capabilities</p>			

### Integration Suite in a Nutshell

Accelerate your Intelligent Enterprise journey with SAP Integration Suite.



- Diverse Integration Scenarios**  
Integrate processes and data using application-to-application and business-to-business integration scenarios.
  - Multiple Endpoints**  
Connect various SAP and non-SAP applications and data sources with on-premise or cloud systems.
  - Customize Integration**  
Extend pre-configured integration scenarios to suit your custom requirements.
  - Prepackaged Content**  
Use prepackaged integration content to jump-start integration projects and get productive quickly.
- [Explore SAP API Business Hub](#)

### Resources

Explore the resources to familiarize with SAP Integration Suite.



## Success Stories of Customers Who Transitioned to SAP Integration Suite

Customers who are already ahead in their digital transformation journeys have leveraged SAP Integration Suite to build modern and future-proof IT landscapes. This has helped them to take advantage of modern approaches to not only improve their existing business processes but also innovate and deliver a world-class customer experience to their customers and partners. The following section features a preview of a few customers who reaped the benefits of using SAP Integration Suite as their integration platform of choice.

For a full list of references of customers who are using SAP Integration Suite as their primary integration platform, see [SAP Integration Suite Customer Stories on SAP Customer Story Finder](#).

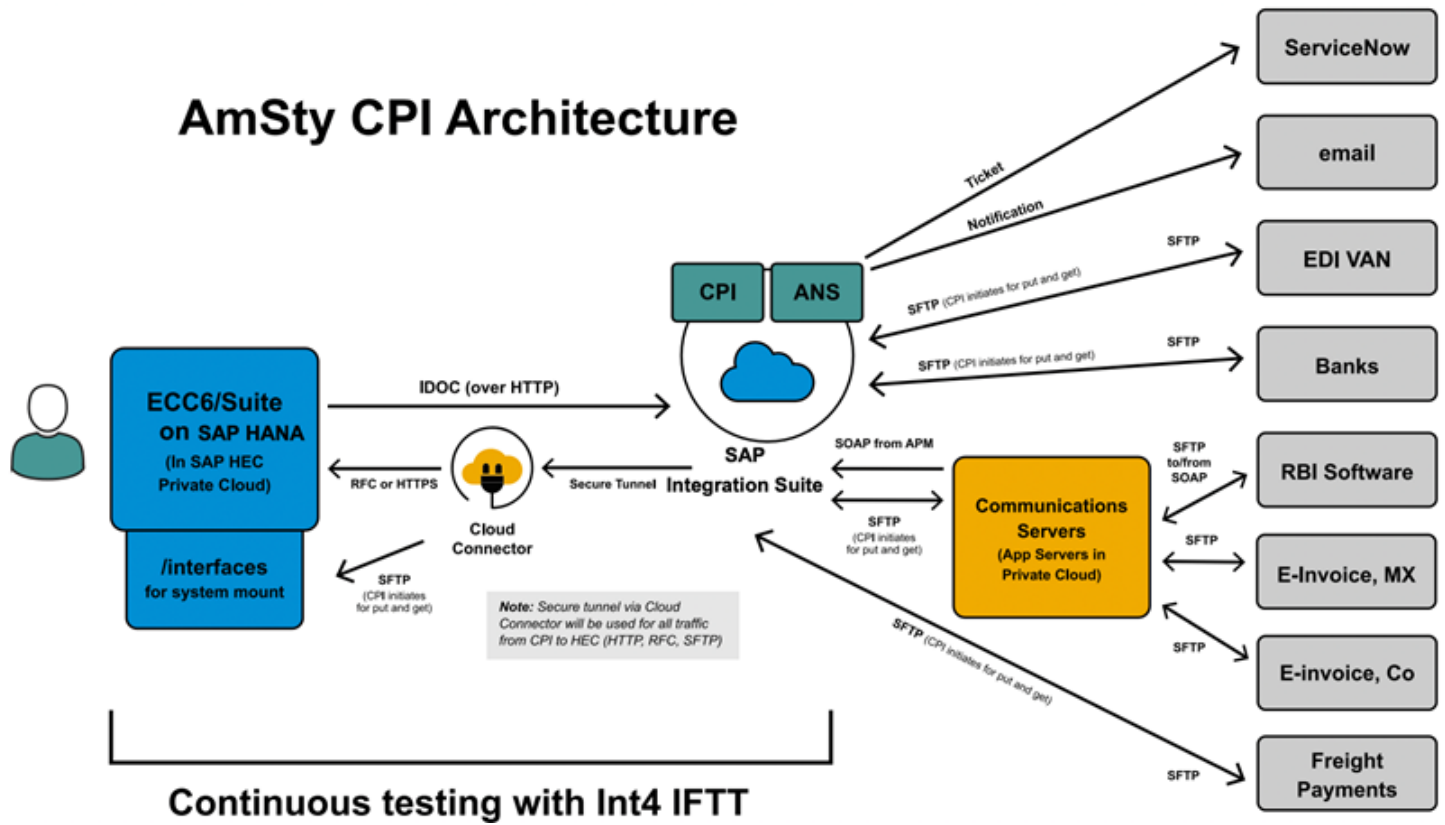
### American Styrenics LLC

In a bid to future-proof their integration platform, American Styrenics LLC (AmSty) decided to migrate their existing 40 SAP Process Integration interfaces to SAP Integration Suite. They successfully completed the migration project within 3 months, with no impact on existing business. The fact that there were no major incidents after the migration is a testament to the degree of success of their migration project. The organization also leveraged SAP Alert Notification Service to integrate alerting with ServiceNow. This enabled them to distribute the monitoring of their integrations across their support organization.

For the full story, see [American Styrenics LLC success story on SAP Innovations Award portal](#).



The following image provides an overview of AmSty's architecture.

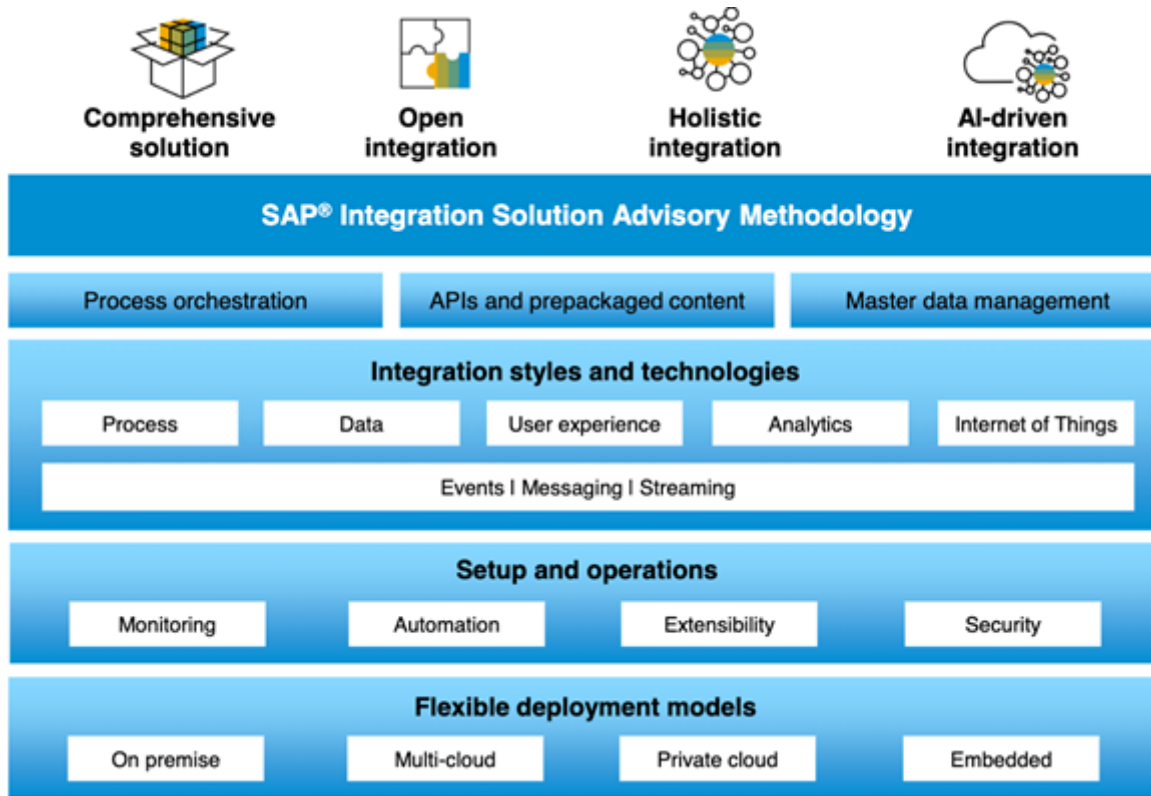


### QforIT

QforIT streamlined enterprise integrations for its customers with SAP Integration Suite. They took advantage of the prepackaged integration scenarios available on the SAP Business Accelerator Hub, improving the speed of configuring integration scenarios by 400%. They also achieved a lift and shift approach to shifting scenarios from SAP Process Orchestration and SAP Process Integration to SAP Integration Suite with minimal efforts and risks. Improved monitoring capabilities provided by SAP Integration Suite helped them identify possible problems and enable fast, reactive monitoring support. QforIT uses SAP Integration Suite for smart integration of processes, people, things, and data.

For the full story, see [SAP Customer Story Finder portal](#).

The following image is a graphical representation of QforIT's integration strategy.



## Transition to SAP Integration Suite

Review the approaches you can use to transition to SAP Integration Suite.

There are two approaches in which you can plan to transition to SAP Integration Suite:

1. **All interfaces from SAP Process Integration and SAP Process Orchestration are moved to SAP Integration Suite (recommended).**

In this approach, SAP Integration Suite is used as the primary integration platform and all the interfaces on SAP Process Integration and SAP Process Orchestration are transitioned to SAP Integration Suite. This transition is an opportunity to evaluate the integration scenarios and improve or adapt them to suit the cloud-first architecture. Modern integration approaches like event-based integration can be leveraged to build intelligent scenarios and improve the end-to-end efficiency of business processes.

2. **Deploy and maintain all cloud-to-cloud and hybrid integration scenarios on SAP Integration Suite and use SAP Process Integration and SAP Process Orchestration for a limited set of scenarios.**

In this approach, too, SAP Integration Suite is the primary integration platform. All new scenarios including cloud-to-cloud and cloud-to-on-premise are deployed on SAP Integration Suite. Only on-premise-to-on-premise integration scenarios are deployed on the existing SAP Process Integration and SAP Process Orchestration landscape.

If you prefer this approach, follow these steps as a high-level strategy for choosing the integration platform:

- Transition existing cloud and hybrid scenarios to SAP Integration Suite
- Upgrade to SAP NetWeaver 7.5
- Build all new scenarios on SAP Integration Suite and deploy on-premise scenarios on the Cloud Integration runtime available in SAP NetWeaver 7.5
- Transition to the [Hybrid Deployment Option](#) when it's released

See the article [Technical Guide: Transitioning to SAP Cloud Platform Integration Suite](#).

# Transitioning to the Cloud

The following three steps build your process of transitioning to the cloud: assess, transform, and move.

## Assess

Assess your existing landscape and identify areas of improvement and innovation for you envisioned target landscape, which can amplify the benefit of transitioning to the cloud.

The following tools and assets can help with the assessment:

- The [Integration Value Calculator](#) considering parameters like the type of integrations, average price per hour for integration consultants, duration, etc. to provide a subjective view of the estimated cost savings that can be achieved with SAP Integration Suite.
- The [SAP Integration Solution Advisory Methodology](#) tool evaluates the current integration maturity level of an organization and provides possible improvements.
- As enterprise-grade integration scenarios must be scalable, robust, and secure, the [Integration Flow Design Guidelines](#) provide best practices that can be used to build such high-quality integration scenarios. They contain patterns, tips and tricks, and development best practices that can act as guiding principles for development.
- A **future-proof integration strategy** ensures that you're prepared for the next steps in your digital transformation journeys. It's critical to factor this into your overall strategy to control costs and encourage innovation.
- Use [Migration Assessment](#), a capability of SAP Integration Suite, to estimate the technical effort involved in the migration process and evaluate how your integration scenarios might be migrated from SAP Process Orchestration to SAP Integration Suite.

## Transform

One of the primary objectives of moving to the cloud and to SAP Integration Suite is to digitally transform your organization. The key component in this transformation journey is to upgrade, refactor, and build modern business process and customer experiences that leverage the latest integration patterns, like API-first approach, event-sensing capability, and machine learning-based interface development. While a lift and shift may sound like the path of least resistance, a transformation with a KPI to improve business process efficiency and user experience results in maximum return on investment.

The following pointers can guide you in the transformation journey:

- [Accelerate Using Prebuilt Integrations from SAP Business Accelerator Hub](#): One of the key differentiators of SAP Integration Suite is the business content provided on SAP Business Accelerator Hub, which accelerates your development project and saves you maintenance effort. The content is managed by the content owners who provide upgrades and patches that can be consumed with a few clicks and no impact to running processes. If standard unedited content is used for SAP-to-SAP integration, the messages processed by this are free of cost. Customization of this standard content can be done through extensions that don't disrupt the lifecycle of the prepackaged content.
- [Leverage APIs and Events](#): A modern-day integration landscape takes advantage of APIs and events to build an intelligent enterprise. The scenarios can be configured to sense and respond to business events in real time. Communication is simplified with APIs that not only provide a greater degree of control over the way data is exposed, but also protect the mission backend systems. An API-first approach is recommended to build a landscape that is open to innovation and helps deliver world-class customer experience.
- [Secure the Cloud](#): Cloud-first strategy should be coupled with enterprise-grade security to ensure data confidentiality and protection. There are various approaches to securing cloud-based business processes, in addition to the security features provided by the SaaS service providers. SAP Integration Suite offers security policy templates that offer enterprise-grade security and ensure that APIs are protected at the highest possible standards.

## Move

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

To make your transition to SAP Integration Suite efficient, the steps involved ensure that existing integration assets are reused as much as possible to reduce costs and time, while also supporting that transformation goals are achieved.

- [Assess the Landscape](#): A detailed assessment of the existing integration landscape and the involved artifacts provides an overview of the artifacts that can be reused, refactored, and developed from scratch. This ensures optimum reuse while accelerating your move and saving costs at the same time.
- [Connectivity Migration](#): The connectivity in the on-premise SAP Process Integration and SAP Process Orchestration systems must be migrated to the respective adapters and connectors on SAP Integration Suite. In this step, the protocols used in these communication channels are evaluated and equivalent connectivity options should be chosen on SAP Integration Suite.
- [Security](#): The security aspects of a cloud-based integration platform depend on two main factors: security features provided by the platform (in this case SAP), and the security configurations that are developed to protect the integration scenarios using the features provided by the platform. This security guide provides an overview of the security aspects that should be considered when moving to SAP Integration Suite from SAP Process Integration and SAP Process Orchestration.
- [Error Handling and Logging](#): When transitioning to a cloud-based integration platform, you also require a cloud-based error handling and logging strategy. Learn about the different approaches and adapt to the error handling and logging features provided by SAP Integration Suite.
- [Interface Governance](#): Understand how interfaces are governed in the cloud and how to manage them across multiple landscapes. Learn about how transport service on the SAP Business Technology Platform can be used to implement an interface governance model.
- [Interface Migration](#): Understand the different aspects involved in moving interfaces from SAP Process Integration and SAP Process Orchestration to SAP Integration Suite. Learn about scenario and object assessment and how parts of testing can be automated with the available tools.

## Landscape

Get an overview of the important aspects regarding integration landscape, which must be evaluated before migrating from SAP Process Orchestration to SAP Integration Suite in SAP BTP, Cloud Foundry Environment.

### [Understanding Landscape Requirements](#)

Before starting a migration from SAP Process Orchestration to SAP Integration Suite, assess the current landscape and design the to-be landscape.

### [SAP Process Orchestration Architecture](#)

Get an overview of the architecture and main components of SAP Process Orchestration.

### [SAP Process Orchestration Landscape](#)

SAP Process Orchestration offers several landscape deployment solutions that can be categorized as central, distributed (model 1), or distributed (model 2) domains.

### [Landscape Architecture Use Cases](#)

Learn what landscape requirements are met with the integration architecture approaches of SAP Process Orchestration.

### [Cloud Foundry](#)

The Cloud Foundry environment allows you to create polyglot cloud applications. It contains the Cloud Foundry runtime for SAP BTP, which is based on the open-source application platform managed by the Cloud Foundry Foundation.

### [Landscape Aspects](#)

Get an overview of the main aspects of landscape, which are governance, business system communication strategy, software logistics and synchronization, and security utilization.

### [Check Connected Systems](#)

One of the first steps in your migration journey is to evaluate which systems are currently connected to SAP Process Orchestration.

### [Cloud Middleware](#)

Learn more about possible communications between Cloud Integration and an on-premise system.

### [Future Support of On-Premise Integration](#)

Many customers will still have a strong on-premise landscape and on-premise integration scenarios. This scenario is planned to be supported by a hybrid deployment option to avoid sending data to and receiving data from the cloud.

### [Identity Provider](#)

Learn more about identity providers and user governance in SAP BTP.

## Understanding Landscape Requirements

Before starting a migration from SAP Process Orchestration to SAP Integration Suite, assess the current landscape and design the to-be landscape.

For this assessment process, SAP recommends using the [SAP Integration Solution Advisory Methodology](#). The goal of SAP Integration Solution Advisory Methodology is to simplify integration and help enterprise architects to manage the complexity in their hybrid landscapes.

There are three use cases on how the methodology can support enterprise architects:



One of the first steps of SAP Integration Solution Advisory Methodology is to analyze the as-is landscape and define the to-be landscape. The focus is on the integration landscape, integration domains, integration patterns, and use cases. At this phase, there are no technologies evaluated. This provides a good overview of where the organization currently is and in which direction the organization is evolving in the next years with regards to the integration strategy.

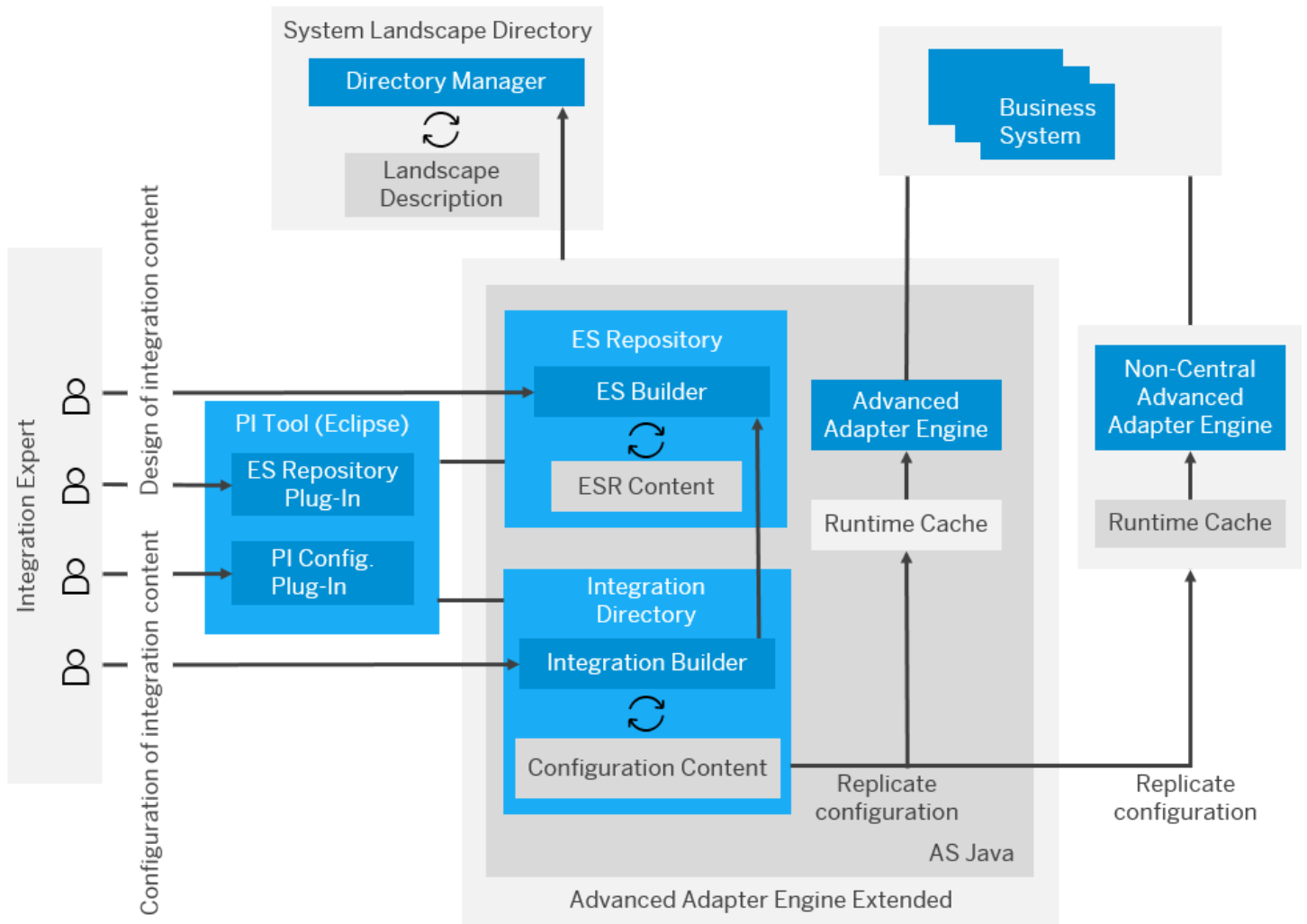
Once the direction of the integration strategy has been mapped out at a high level, a more detailed evaluation of the current technologies used within the SAP Process Integration and SAP Process Orchestration landscape can commence.

[SAP Process Orchestration Architecture](#) describes the important factors to analyze with regards to the existing SAP Process Integration and SAP Process Orchestration landscape, and the high-level steps to follow to ensure a successful migration.

## SAP Process Orchestration Architecture

Get an overview of the architecture and main components of SAP Process Orchestration.

In an SAP Process Orchestration system, there are different tools to define, design, and develop integration scenarios. The following figure illustrates its high-level architecture:



The main components of SAP Process Orchestration are the following:

- **System Landscape Directory (SLD)**: The SLD is a component of SAP NetWeaver and is implemented with Java technology. It's the central information repository for the system landscape and stores all information related to the installed components of the landscape, which means it also facilitates the easy access to the information of the different systems and software. The information is divided in three categories:
  - **Landscape**: configuration of the technical systems, landscapes, and business systems.
  - **Software Catalog**: information about the products and software components.
  - **Development**: information about the name reservations and CIM instances.
- **Enterprise Services Repository (ESR)**: ES Repository is a central repository that makes it possible to define, design, and create objects to be used in the integration scenario. The repository stores the definitions and metadata of enterprise services and business processes. It also provides a central modeling and design environment for creating and aggregating data models. ES Repository is managed using the **Enterprise Services Builder (ESB)**, which is the tool used to develop and access the objects in ES Repository and the **Services Registry**:
  - **Enterprise Services Builder (ESB)**: ESB is used to define and manage objects in ESR. It contains the namespaces, data types, message types, service interfaces (outbound/inbound), message mapping, and operation mappings of interfaces.
  - **Services Registry**: Services Registry represents a registry for Web services and centrally stores information regarding the services within SOA landscape. Contains information related to the services, with references to the WSDL of registered services and endpoints. It provides support for SAP and non-SAP applications and controls


the available services in SOA landscape. It's an important source for developers to find the available services in the system landscape that could be reused in integration scenarios.

- **Integration Directory (ID):** The Integration Directory is used to configure the end-to-end integration scenarios using the objects created in ESR and SLD.

Objects included:

- Integrated Configuration Objects
- Communication Channels
- Routing between the sender and target systems, and the mappings.

Related Toolset:

- **Integration Builder:** Edit (create, copy) and manage the objects in the Integration Directory.
- **Advanced Adapter Engine (AAE):** With the Advanced Adapter Engine, you can connect SAP systems and non-SAP systems. You use the various adapters in the Advanced Adapter Engine to convert XML- and HTTP-based messages to the specific protocol and format required by these systems, and the other way around.
- **Cloud Integration Content:** You can deploy, execute, and operate cloud integration content from the SAP Integration Content Catalog in the Advanced Adapter Engine with all the deployment options (PI-AEX, decentralize Adapter Engine, SAP Process Orchestration and SAP Process Integration dual usage type). For more information, see SAP Note [2197483](#) .
- **Configuration and Monitoring:** These components provide an overview of the status of the individual components of SAP Process Integration and SAP Process Orchestration and include the capability to perform test message execution to verify that the runtime components are working correctly, test connectivity, check the status of your communication channels or the Java Proxies, etc. For more information, see the section [Error Handling and Logging Strategy](#).

The integrated development environment SAP NetWeaver Developer Studio (NWDS) is an Eclipse-based tool that can be used to develop end-to-end scenarios (interfaces). In NWDS, it's possible to create ESR objects and Integration Directory configurations, and to graphically model the message handling within the Process Integration Designer perspective.

With NWDS, you can build end-to-end integrations using integration flows, more plug-ins for development that might be accessed by the perspectives and views, and an integrated runtime view. The SAP Process Integration Runtime perspective allows you to directly navigate into the respective monitoring environments of SAP Process Integration such as message monitoring and channel monitoring. For more information, see [SAP NetWeaver Developer Studio Basics](#).

## SAP Process Orchestration Landscape

SAP Process Orchestration offers several landscape deployment solutions that can be categorized as central, distributed (model 1), or distributed (model 2) domains.

The following section describes the three high-level architectural models in SAP Process Orchestration deployment: [Central](#), [Distributed \(Model 1\)](#), and [Distributed \(Model 2\)](#). The scope is to give an overview of architecture topology with specific reference to federation, and not to discuss scaling with respect to performance or high availability.

### Central

A single integration server communicating with all business systems. Central deployment of SAP Process Orchestration is a common architecture for most organizations as it represents the smallest possible installation footprint and holds associated TCO benefits.

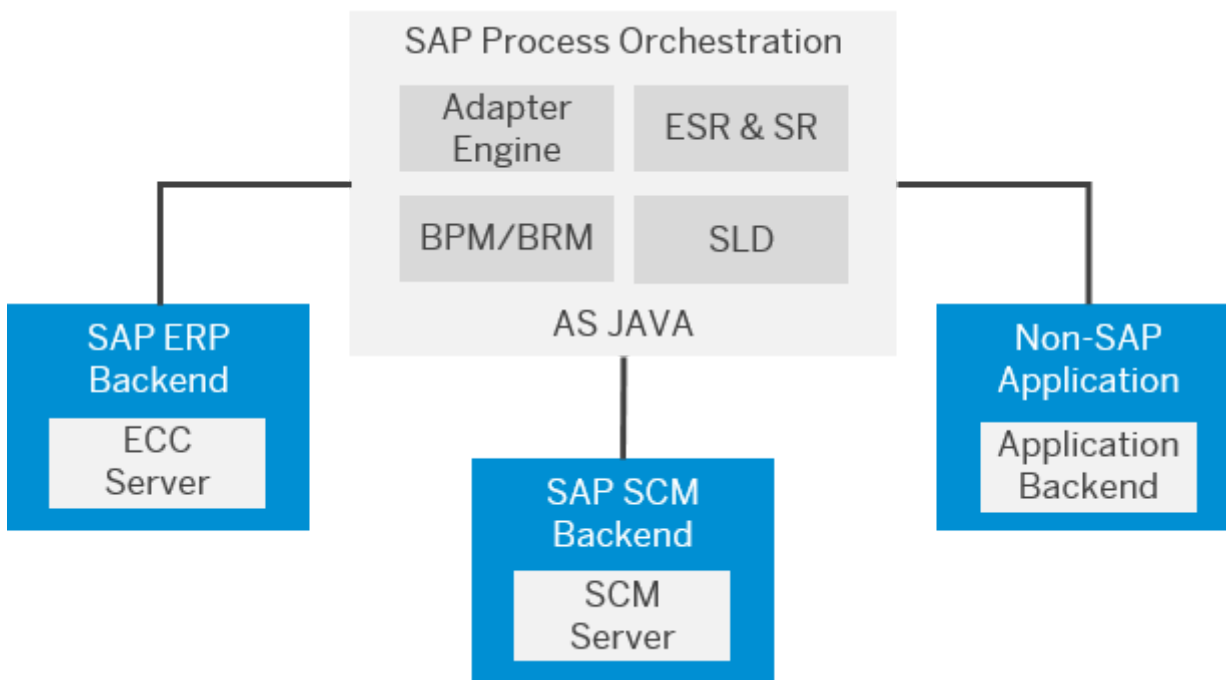
#### Benefits

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

- Less complex system architecture
- Central governance of metadata
- Central configuration, monitoring, and administration
- Support for stateless message processing, and stateful message orchestration

### Considerations

- For SAP Process Orchestration, system sizing and configuration must reflect both human-centric and integration-centric scenarios.



### Distributed (Model 1)

A single integration server with one or more decentralized adapter engines communicating with relevant business systems. It's often found in organizations where performance or security reasons dictate a more complex solution compared to centralization. This is a hybrid approach between central and federated models combining the benefits around performance and security with the benefits from centralized governance, maintenance, and monitoring.

#### Benefits

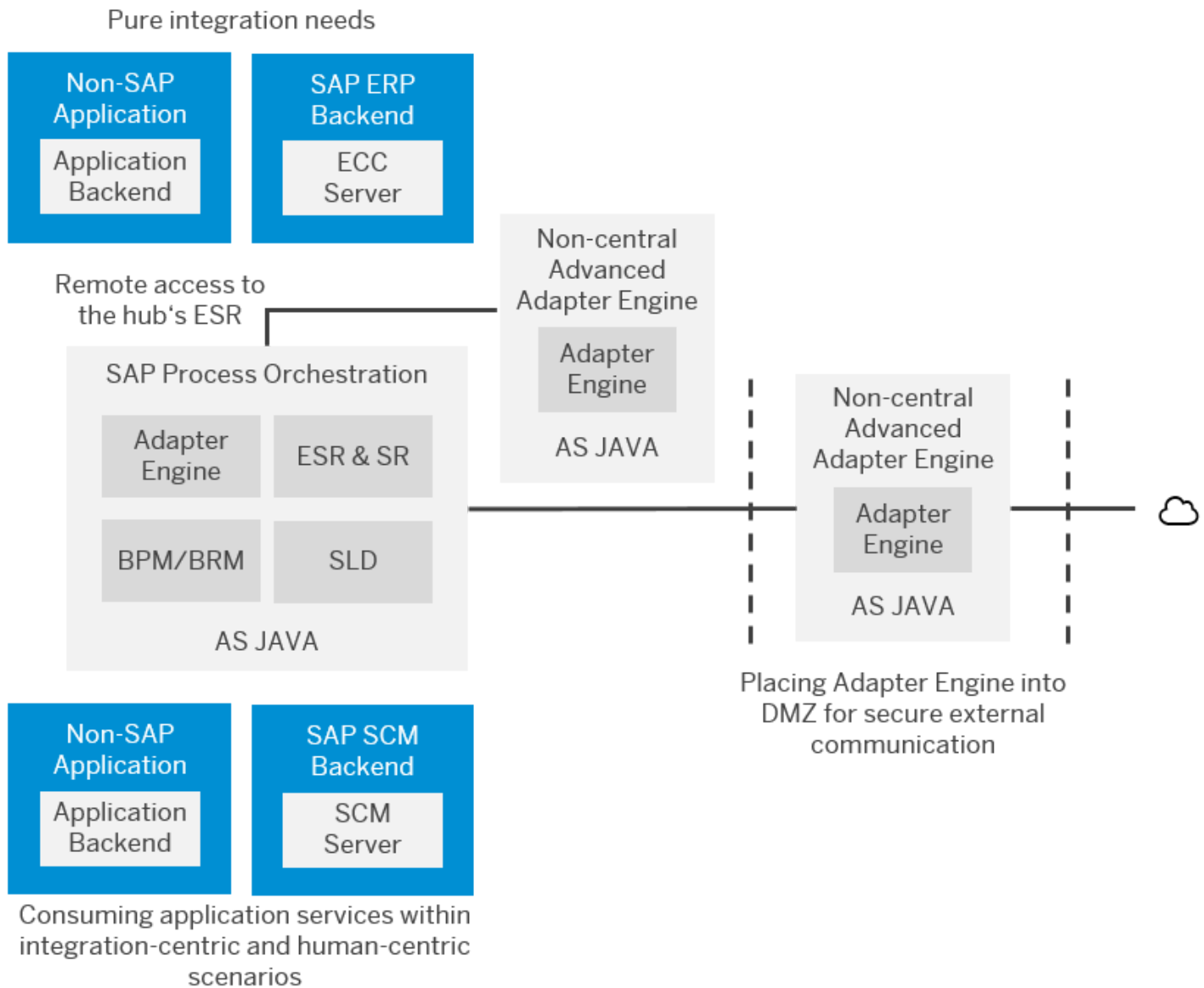
- Centralized governance, maintenance, and monitoring
- Decoupled runtimes and distributed load
- Separating pure messaging from process orchestration
- Localization: Reduced message travel time and reduced network load in global networks
- Business continuity: Minimize planned downtime via switchover
- Security: Secure deployment of AAE instances in DMZ enforcing network isolation for B2B scenarios

#### Considerations

- BPM/ccBPM scenarios require central SAP Process Integration / SAP Process Orchestration runtime.



- For SAP Process Orchestration, two Integrated Configuration Objects are required for scenarios spanning both Adapter Engines.



## Distributed (Model 2)

Two or more integration servers (domains) within a single landscape tier communicating with relevant business systems. Use of additional decentralized adapter engines per integration server is also possible and adds a further level of complexity. Due to the more complex architecture, the second distributed domain model allows greater flexibility, however at the expense of a higher TCO. Such organizations are usually larger in size and require a high degree of abstraction in their landscape. Customer-driven reasons include:

- Organization and divisional autonomy due to legal or operational reasons
- Separation of A2A and B2B integration scenarios for security reasons
- Separation by process types such as transactional versus master data
- Business continuity such as downtime minimization
- Quality of Service obligations
- Billing requirements based on volume
- Security issues around personal data

- Prioritization of messages
- Geography
- Technical Abstraction from upgrades & downtime

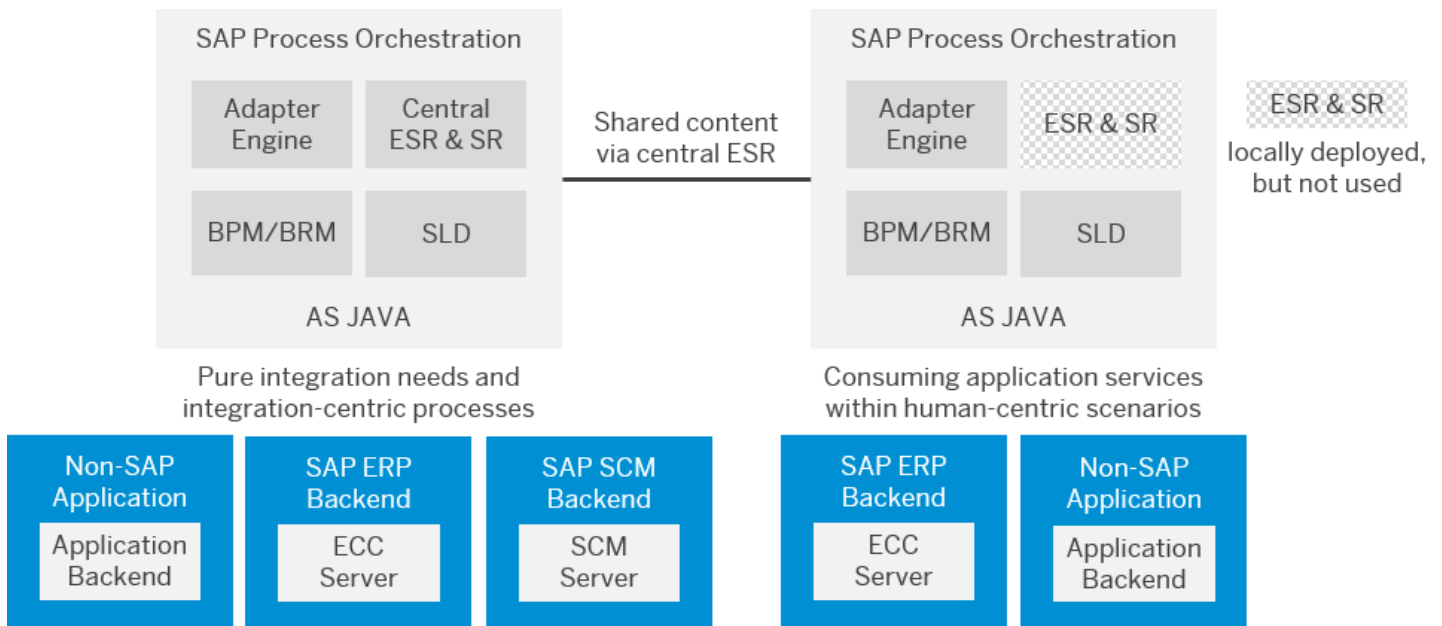
In contrast to customer-driven reasons for this model, there are also some SAP-driven reasons such as isolation for upgrade path independency or decoupling of business systems and portal UI bindings.

**Benefits**

- Separation of SAP Process Integration / SAP Process Orchestration systems due to different reasons, such as technical aspects, business requirements, or geographical distribution
- Systems can be optimized for their respective use case
- ESR can be connected to multiple systems
- Central development and service design

**Considerations**

- Central ESR requires at least SAP NetWeaver 7.31
- Increased TCO due to additional administration effort



## Landscape Architecture Use Cases

Learn what landscape requirements are met with the integration architecture approaches of SAP Process Orchestration.

The different integration architecture approaches of SAP Process Orchestration categorized in [SAP Process Orchestration Landscape](#) can be used to meet different landscape requirements. The following table lists some assessment aspects of these requirements:

Assessment Criteria	Description
Administration	Defines how difficult the overall landscape is to operate and maintain with respect to downtime, planned upgrades etc. The application of support packages, notes, and hot fixes are included in this criterion.

Assessment Criteria	Description
Availability	Relates to how available the SAP Process Integration system is for message processing. In distributed (model 1) and distributed (model 2) domain models, availability is improved through the decentralized message processing capability and technical abstraction.
Monitoring	Defines the complexity and coverage of the monitoring solution in each deployment model. With SAP Process Integration, IT professionals can configure and monitor service-enabled applications across the landscape. Using the integrated capabilities of SAP Solution Manager and SAP NetWeaver, IT administrators can monitor the availability and performance of applications, processes, and services, as well as analyze the root cause of problems, and undertake optimization measures.
Total Cost of Ownership	Defines the costs of implementing, operating, and supporting a given IT solution. This includes areas such as the technical infrastructure, software licensing, implementation costs, support costs, upgrade, and downtime costs.
Internal Performance	Covers the internal performance of the SAP application software, database, and operating system when processing a message in SAP Process Integration. It assumes that other variables such as message packaging and size of messages are constant, which is an indicate measurement for the relationship between each architectural model.
Network Utilization	Reflects the impact on or utilization of the physical communication network. It considers the geographical location of business systems, integration servers, and adapter engines, as well as latency and reliability.
Common Business Process Delivery	The ability to enforce common business processes across the SAP Process Integration landscape, including design-time object reuse and redundancy avoidance.
Globalization & Localization	The ability to meet local and global requirements both technically and on a business level. For example, a distributed (model 2)-domain solution with local governance can add flexibility and abstraction while a centralized governance model might not meet local requirements.

## Cloud Foundry

The Cloud Foundry environment allows you to create polyglot cloud applications. It contains the Cloud Foundry runtime for SAP BTP, which is based on the open-source application platform managed by the Cloud Foundry Foundation.

The Cloud Foundry environment enables you to develop new business applications and business services, supporting multiple runtimes, programming languages, libraries, and services. You can leverage a multitude of build packs, including community innovations and self-developed build packs. For more information, see the official [Cloud Foundry documentation](#) .

The following hyperscales and datacenters are available:

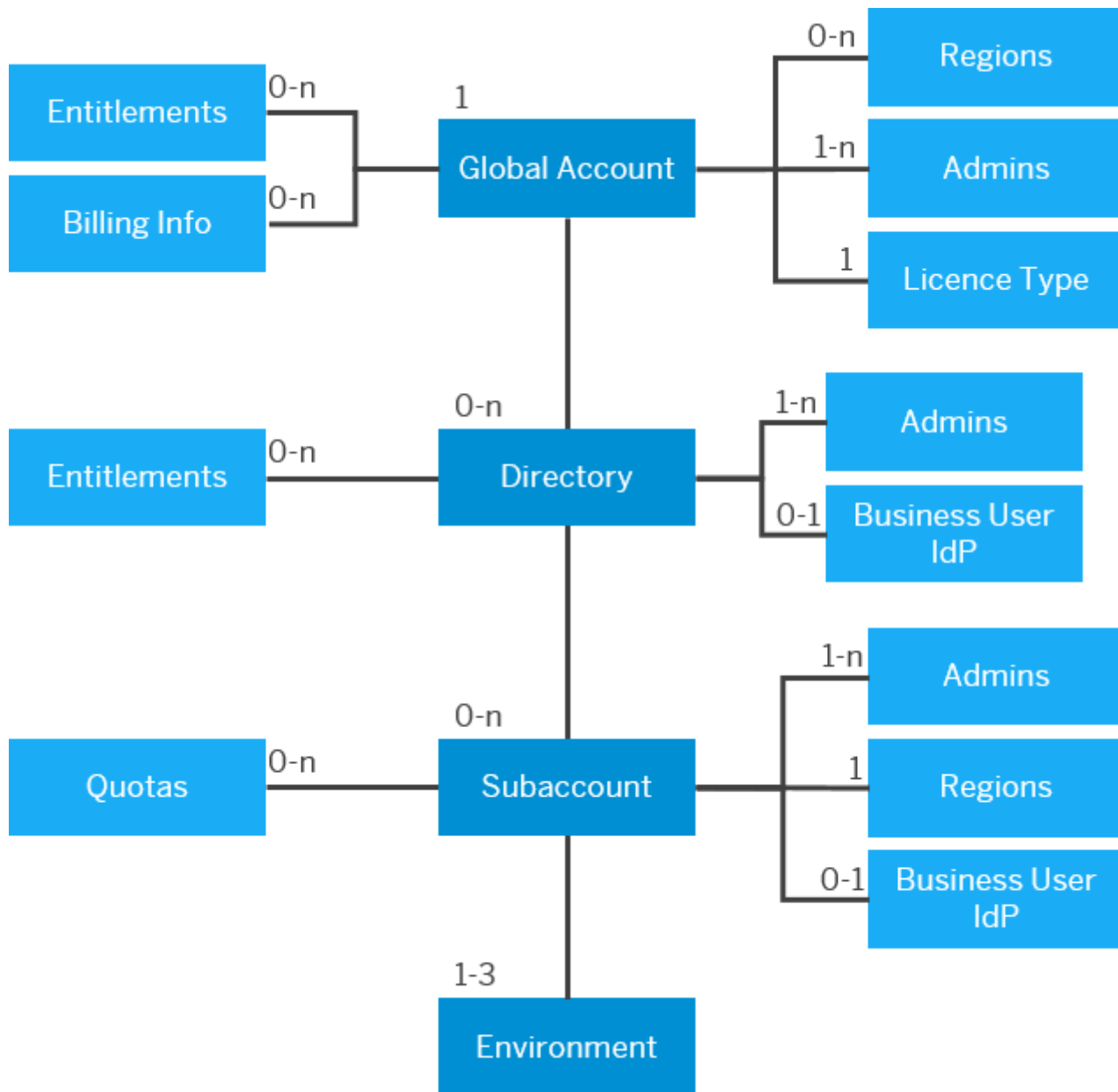
- Microsoft Azure
- Amazon Web Services
- Alibaba
- Google Cloud

### ! Restriction

The availability on Google Cloud is planned for the future and is subject to change. For more information on new features and future releases, access the [Road Map Explorer](#) and the [What's New section](#) of SAP Integration Suite.)

## Account Provisioning

With the cloud-centric model on SAP Business Technology Platform, slightly different terminology is used – **global accounts**, **subaccounts**, and **regions**. License costs are rolled up to the **global account** level which, based on the account type selected, offers different entitlements. The entitlements can then be allocated as quota to **subaccounts**. The quota ensures efficient allocation of services and resources across subaccounts.



Customers can deploy applications in different **regions**. Each region represents a geographical location (for example, Europe, US East) in which applications, data, or services are hosted. Regions are provided either by SAP or by our Infrastructure-as-a-Service (IaaS) partners Amazon Web Services (AWS), Microsoft Azure, Google Cloud, and Alibaba Cloud. The third-party region providers operate the infrastructure layer of the regions, while SAP operates the platform layer and Cloud Foundry.

Separated subaccounts can be provisioned to cater for the different environment requirements at the time so customers could have subaccounts for production, test, development environments etc. Subaccounts can also be provisioned on a temporary basis (e.g., to support a fixed-time-frame project) and deleted later, releasing any services and resources used. Since the license costs reside at the global account level, you can have as many subaccounts as needed.

Subaccounts enable companies to have a dedicated dev subaccount per development project. After the development and test phases, the app can be published in one single test subaccount and then one production subaccount. This is especially suited for companies with a focus on continuous integration and continuous delivery, as it creates isolation and independency for the different environments. For more details, see [Account Model 5: Create a Staged Development Environment Per Functional Area](#) in the Best Practices for SAP BTP.

## SAP Integration Suite on Multi-Cloud

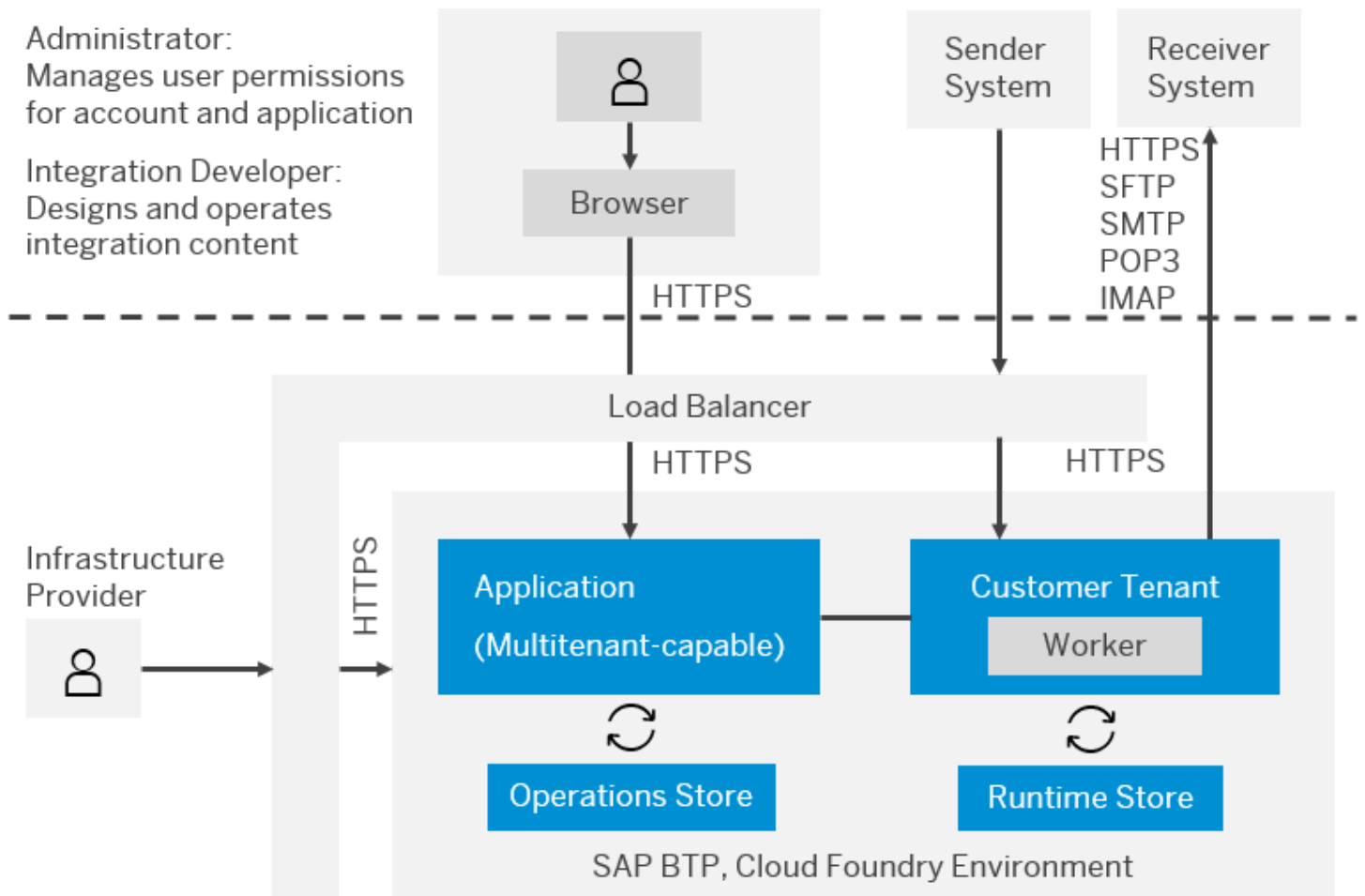
SAP Integration Suite is an enterprise integration platform as a service (EiPaaS) that allows you to connect applications, processes, and people with speed and ease. It runs on the hyperscalers and includes different capabilities like Cloud Integration, API Management, Integration Advisor, and Open Connectors to solve integration challenges.

Since Cloud Integration is the main capability used during the migration of SAP Process Integration and SAP Process Orchestration to SAP Integration Suite, it's described in the following chapters.

### Cloud Integration Landscape

The technical approach in Cloud Integration differs from the SAP Process Integration and SAP Process Orchestration landscape. In Cloud Integration, the integration platform is designed as a containerized and clustered integration platform. Messages processed by integration flows from different customers are handled on different parts of the platform (referred to as tenants). Tenants processing integration flows from different customers are strictly separated from each other in terms of CPU, data storage, and user access. For more information, see [Technical Landscape, Cloud Foundry Environment](#)

The following diagram describes the high-level architecture of Cloud Integration:



In a Cloud Integration system, you have two key components:

- Tenant Management Nodes

The tenant management node represents the web interface of Cloud Integration.

The following operations can be performed in the web interface:

- **Discover:** Discover standard content available on SAP Business Accelerator Hub and copy this content to the tenant workspace.

- **Design Content (Workspace):** In the workspace you can:
  - Configure standard content
  - Develop custom content
  - Deploy artifacts (both standard and custom) to the runtime (worker) node(s).

The content is created within integration packages. A package contains:

- Details about the package content (description, version, vendor, ...)
- Documents: Additional resources to describe the content (files or links)
- Artifacts: integration flows, value-mappings, script collections, REST API, SOAP API, and custom integration adapters. Other artifacts such as referenceable message mappings will become available shortly, and others may be added in the future.

- **Monitor:** In this view all monitoring and operation tasks can be executed, including:
  - Monitor Message Processing: Provides an overview of the processed messages, with the capability to drill down into individual message processing instances to diagnose any issues
  - Manage Integration Content: Deploy integration artifacts and monitor their deployment status
  - Manage Security:
    - Security Material (Credentials, OAuth, ...)
    - Keystore (Certificates)
    - Access Policies
    - JDBC Material
    - User Roles
    - Connectivity Test: Testing Connectivity from Cloud Integration to other systems (SSH, TLS, FTP, SMTP etc.)
  - Manage Stores: Data Storage, Variables, Number Ranges, Message Queues
  - Manage Locks: Message Locks, Design-Time Artifact Locks
- Worker Node (runtime container)

During the deployment process, an integration artifact is compiled into an executable format and copied to each runtime container. The worker nodes then process messages that are exchanged with external systems or other integration flows.

## Central Architecture

In a central landscape, many business systems are bound to a single integration server. Most customers use this landscape as a starting point as it represents the smallest possible installation footprint and holds associated TCO benefits compared to the other models. For most organizations this model is ideal and caters for central governance and a high reuse of development objects. Global requirements can be met in this architectural model by implementing appropriate security and role-based policies for design time and runtime objects.

The central approach is also the perfect starting point for entering the cloud world, as all business systems connect to a single SAP Integration Suite.

## Distributed (Model 1) Architecture

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

Distributed (model 1) models are often found in organizations where performance or security reasons dictate a more complex solution compared to centralization. The distributed (model 1) model is a hybrid approach between central and distributed (model 2) domain models as it combines the benefits of performance and security with the benefits of centralized governance, maintenance, and monitoring.

A distributed architectural model on SAP Process Orchestration is defined by one integration server integrating with one or more decentrally deployed adapter engines. Since the SAP Integration Suite doesn't offer distributed runtime nodes comparable to an Adapter Engine yet, this scenario can't be used. As an alternative, the distributed (model 2) domain model can be used with the same restrictions and guidelines as the distributed model. With the planned release of the hybrid deployment option, the distributed (model 1) architecture can also be resolved with a lightweight runtime in the customer network. The following table describes some aspects of using distributed (model 1) architecture:

Reason	Description	Behavior in the cloud
Localization	Local processing of messages reduces network load while improving performance and reliability	The business system can communicate with regional cloud data centers. On ground localization will be available in the future with the hybrid deployment option.
Business Continuity	<ul style="list-style-type: none"> <li>• Downtime minimization</li> <li>• Technical abstraction from upgrades &amp; downtime</li> </ul>	SAP Integration Suite updates (monthly) are done with zero down time. There are up to 4 major upgrades planned on the SAP BTP per year (up to 4 hours). Depending on the datacenter location, the updates occur in different time slots.

## Distributed (Model 2) Domain Architecture

Due to the more complex architecture, the distributed (model 2) domain model allows greater flexibility, but at the expense of a higher TCO. Organizations that use the distributed (model 2) domain model are usually larger in size and require a high degree of abstraction in their landscape.

Assessment Summary	
Advantages	Disadvantages
Local autonomy	TCO
Flexibility	Governance hard to enforce
Performance improvements due to local processing	Monitoring / Operation complexity
Availability and failover possible (manually)	Multi Hop Scenarios may be introduced
Network efficiency	Common Business Process Delivery

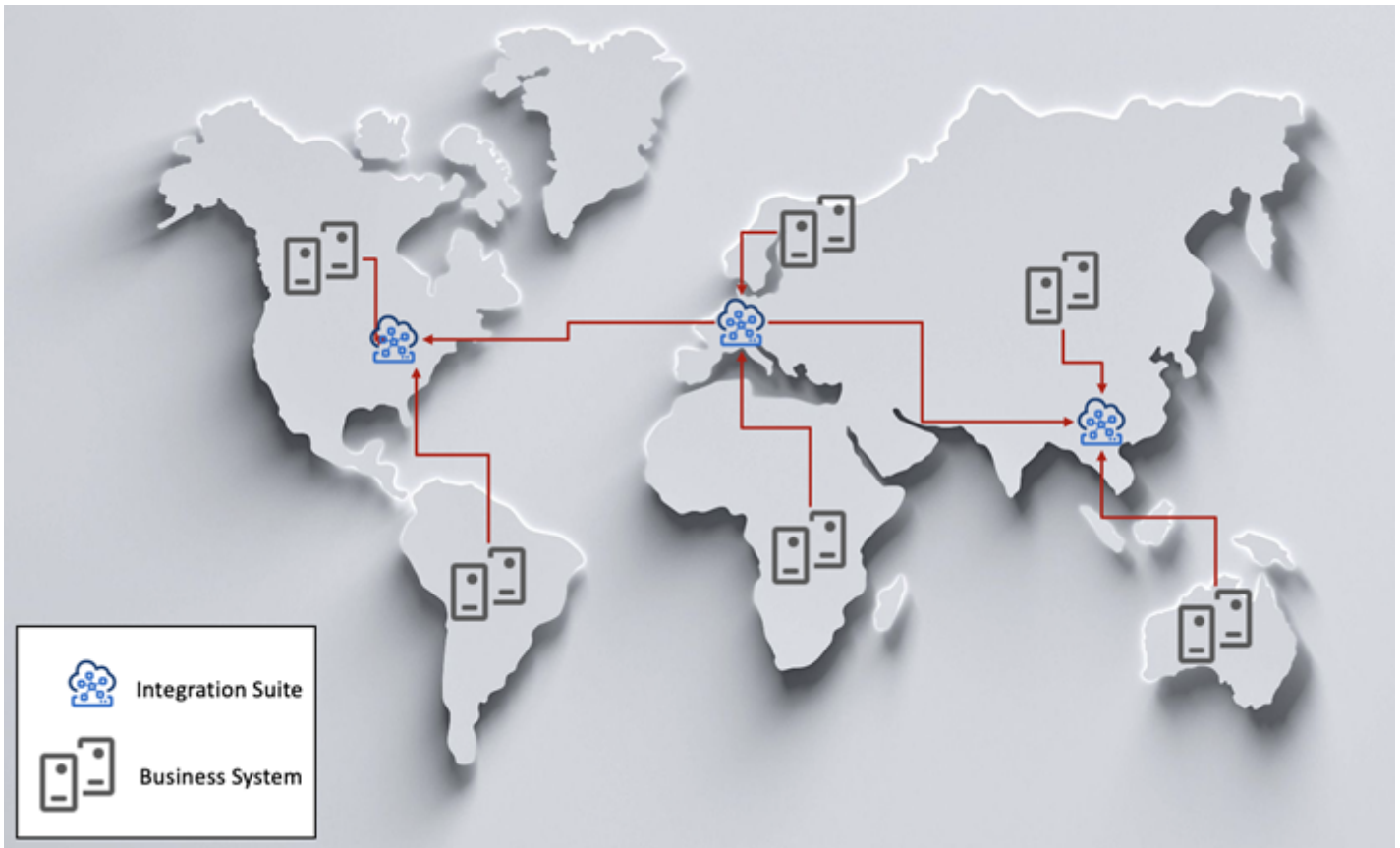
Reason	Description	Behavior in the cloud
--------	-------------	-----------------------

Reason	Description	Behavior in the cloud
Segmentation	<ul style="list-style-type: none"> <li>• Geographical</li> <li>• Organization and divisional autonomy due to legal or operational reasons</li> <li>• Separation of A2A and B2B integration scenarios for security reasons.</li> <li>• Separation by process types such as transactional versus master data.</li> <li>• Quality of Service obligations</li> <li>• Billing requirements based on volume</li> <li>• Prioritization of messages</li> <li>• Security issues around personal data</li> </ul>	<p>Segmentation use cases can be resolved with multiple SAP Integration Suite instances. However, it must be checked if the reasons are still valid.</p> <ul style="list-style-type: none"> <li>• Automatic scaling depending on resource consumption</li> <li>• Additional features to ensure data protection</li> </ul>
Business Continuity	<ul style="list-style-type: none"> <li>• Downtime minimization</li> <li>• Technical abstraction from upgrades &amp; downtime</li> </ul>	<p>SAP Integration Suite updates (monthly) are done with zero down time. There are up to 4 major upgrades planned on the SAP BTP per year (up to 4 hours). Depending on the datacenter location, the updates occur on different time slots.</p>

Within the distributed (model 2) domain architecture, you can have different governance options.

- **Central:** Similar to the central architecture, development is done in only one specific tenant and then replicated to the required tenants. No changes or new developments are allowed in the other tenants.
- **Local:** Design-time objects are created and changed in any development system. Local teams are responsible for their own development objects. This limits the reuse and avoidance of redundancy of objects due to local autonomy.
- **Mixed:** Changes are made centrally and locally, but with strict control mechanisms to ensure some object reuse is possible and redundancies are minimized. This needs to be defined as an enterprise-wide practice by your organization. An example of such a mechanism is enforcing naming conventions or scenario-specific object separation. There could be common or global packages and some specific local packages.





## Landscape Aspects

Get an overview of the main aspects of landscape, which are governance, business system communication strategy, software logistics and synchronization, and security utilization.

### Governance

Governance can be defined by the policies, processes, and procedures that support any IT landscape. In the context of SAP Integration Suite and this guide, governance is defined relative to change management of design time objects in the SAP Integration Suite. Who can change what and where are of relevance in this criterion.

In a distributed (model 2) domain model, a decision needs to be made where changes are allowed, for example if changes are done in one central SAP Integration Suite only or regionally and then replicated with appropriate controls. A central governance model is highly recommended as a starting point in distributed (model 2) domain scenarios. When this can't be achieved, a mixed mode can be adhered to as described in the following:

- **Central**

Design-time objects are created and changed in one development system only and then replicated consistently throughout the landscape as required. No changes are to be made or new objects to be created in other regional domains.

- This is the recommended approach in distributed (model 2) domain scenarios when it can be applied practically. It may require strict policy enforcement and a higher level of collaboration but the benefits of object reuse and one model for everyone are significant.
- The implementation of access policies and user roles for design-time object creation and authoring is recommended.

Governance in a central deployment model is simplified as there's only one development environment for objects to be created or changed and hence central governance strategy is enforced.

- While a distributed (model 2) domain scenario may give you greater flexibility, the central model is recommended to guarantee high object reuse and consistency.
- The use of developing guidelines (naming conventions, roles, and authorizations) can support global requirements and cater to the needs of regional divisions or businesses. Defining these constraints before an implementation is imperative for success and possible future growth.

- **Regional**

Design-time objects are created and changed in any development system. Regional SAP Integration Suite instances are responsible for their own development objects and are not replicated. The ability to reuse objects and avoid redundancies is minimized due to local autonomy and resulting loss of control.

- At the very least, you should implement naming conventions per region or division so that controlled convergence is still possible in case of consolidation at a later stage.
- This mode guarantees isolation of each instance, which may be a business requirement allowing complete autonomy and agility when onboarding or decommissioning segments.

- **Mixed**

Changes are to be made centrally and locally however with strict control mechanisms to ensure some object reuse is possible and redundancies are minimized. An example of such a mechanism is enforcing naming conventions or scenario-specific object separation, like transactional relevant objects x master data.

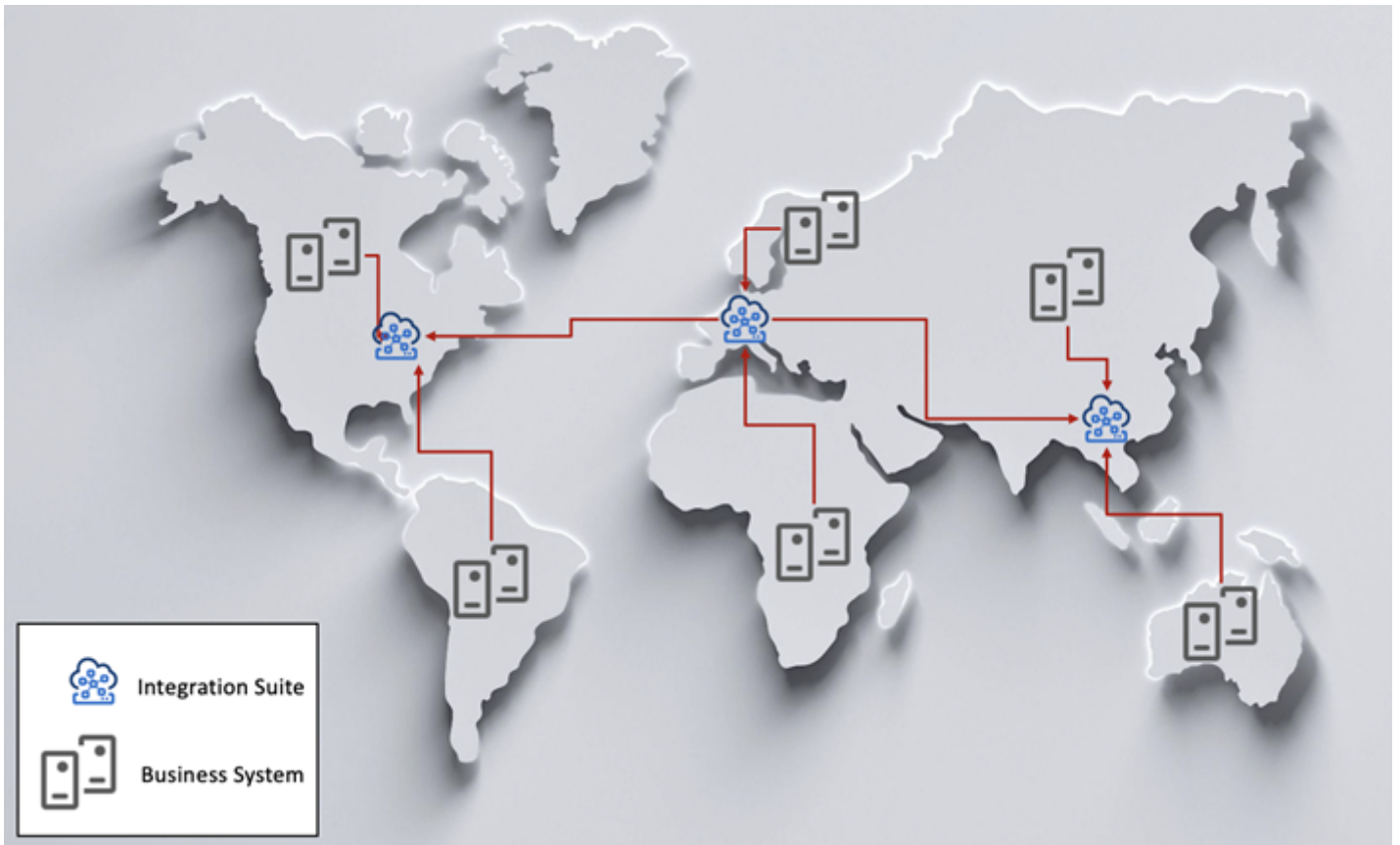
- Demarcation of object ownership is recommended. A global SAP Integration Suite instance may be responsible for master data, and local regions handle the transactional scenarios. Or possibly all SAP-centric content is managed globally and local divisions or regions manage legacy system objects and scenarios.
- The aim of this mode is to achieve some level of reuse and convention such that redundancies are avoided. This demarcation is specific to each organization.

## Business System Communication Strategy

In any architectural model, a decision needs to be made regarding business system communication strategy. What SAP Integration Suite instance can communicate with what business systems? Should a hub-to-hub or a hub-to-system model be permitted? The following diagram illustrates the differences of the approach.

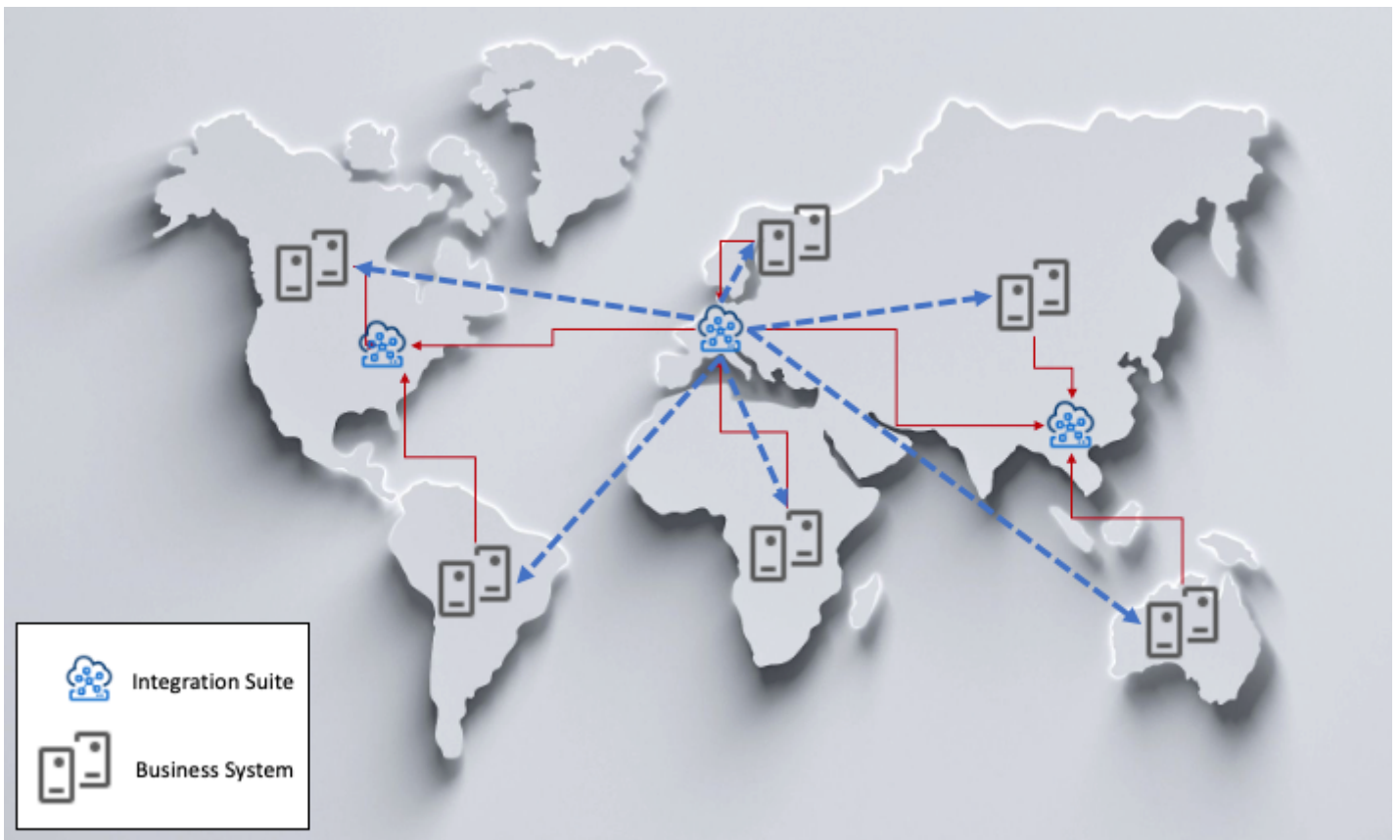
### Hub-to-Hub Model

This approach normalizes communications between regions or divisions such that each business system belongs to a specific SAP Integration Suite. Any messages or scenarios requiring integration to a given business system must be processed via its nominated integration server. This allows consistent communication paths but requires additional configuration due to the additional hops from integration server to integration server. Consideration must be given to performance implications and possible demarcation options based on protocol.



### Hub-to-Business-System Model

This approach allows each SAP Integration Suite to directly communicate with each business system, which increases the complexity of integration scenarios. Managing such an approach requires a higher level of governance and may be restricted to certain protocols. For simplicity, this diagram only illustrates one integration server communicating with each business system



### Software Logistics and Synchronization

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

## Central

In a central deployment model, the transport and synchronization of objects are optimized due to a single production instance. As a result, no synchronization is required.

### Distributed (model 1)

In a distributed (model 1) deployment model, the transport and synchronization of objects are optimized due to a single production instance. As a result, no synchronization is required.

### Distributed (model 2)

In a distributed (model 2) domain deployment model, the transport and synchronization of objects are complex and require careful consideration per integration scenario. It all depends on whether integration scenarios are local or global in nature.

- If the integration scenarios are local and don't require multihop to each integration server, the development objects are discrete in scope.
- If your integration scenarios cross domains and require multihop integration with other integration servers, an increased amount of synchronization is required when promoting scenarios within your landscape.

Generally, the level of distributed (model 2) domains and combination of governance models determines how complex your synchronization effort will be. Keep in mind that models are determined by organisation-specific architectural practices.

## Security Utilization

Although this has been covered to an extent within the governance section, it's important to reiterate the application of security in the context of SAP Integration Suite and distributed architectures. It's common for organizations to specify they need distributed (model 2) domains due to security reasons. SAP Integration Suite offers a feature to configure the access to design-time and runtime artifacts, which allows very strict control. In a global example, you can allow developers in the Europe to only access and change objects they are assigned to, and the same applies to developers in APJ. The recommendation is to thoroughly investigate the application of security as a means to enable central governance and simplify your architecture with a central approach

## Check Connected Systems

One of the first steps in your migration journey is to evaluate which systems are currently connected to SAP Process Orchestration.

The following parameters are relevant for a high-level evaluation:

- Cloud / On-Premise System
- System Description
- Direction
- Adapter types (protocols)
- Polling/Pushing

This information can be picked up from the SAP Process Orchestration System Landscape Directory through analysis of the Integration Directory (ID). With regards to the landscape, the migration to support Cloud systems should always be given except if the connected system has some specific requirements. For ground (on-premise) systems, additional checks are required as discussed in the following chapters.

# Cloud Middleware

Learn more about possible communications between Cloud Integration and an on-premise system.

## ! Restriction

Some features mentioned on this site are planned for the future and are subject to change. For more information on new features and future releases, access the [Road Map Explorer](#) and the [What's New section](#) of SAP Integration Suite.

A migration from SAP Process Orchestration to SAP Integration Suite means moving to the cloud, which requires that all on-premise systems must be able to communicate with the Internet. With the planned release of the on-premise runtime of SAP Integration Suite (hybrid deployment option), this may no longer be applicable. In that case, it means developing and operating in the cloud, and executing on-premise. Since the hybrid deployment option isn't available yet, we'll focus on the cloud-only version. The guide will be updated when the hybrid deployment option is available.

### [On-premise system to Cloud Integration](#)

- The on-premise system must be able to establish an HTTPS connection to Cloud Integration (SSL Handshake, Authorization, and Authentication).
- If any proxies or firewall changes are required, this should be considered and recorded as an infrastructure change requirement.

Since Cloud Integration is a cloud solution, security is a key aspect and therefore SSL protocol is a must.

### [Cloud Integration to on-premise system](#)

The on-premise system must be able to receive a message from the Internet, using one of several options:

- [Cloud Connector](#)
- [Reverse Proxy](#), which requires allowlisting of IP range at customer DC

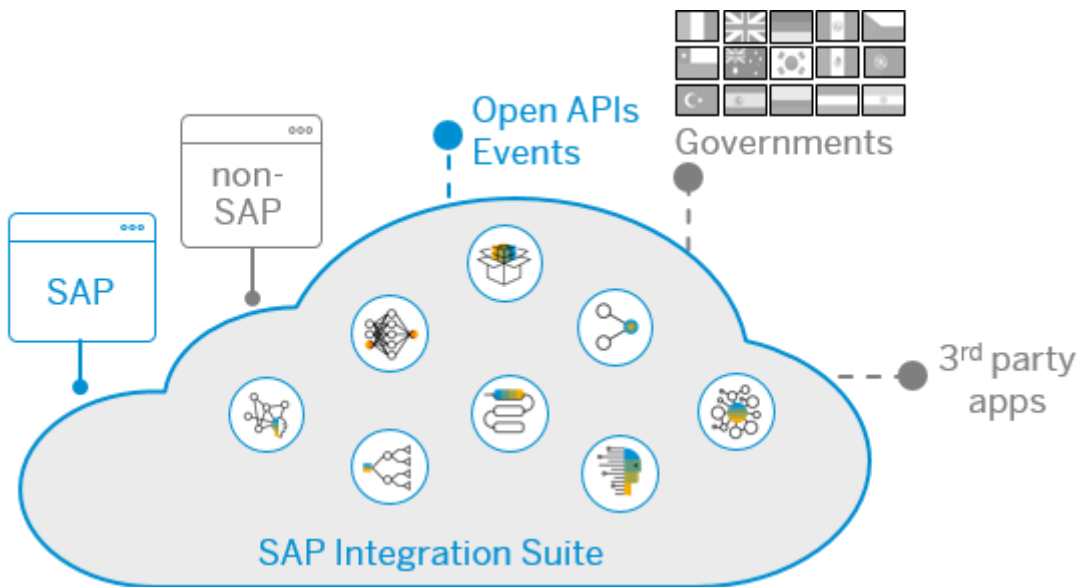
There might also be some challenges regarding security that have to be evaluated for on-premise systems and are described in [Security](#).

## Future Support of On-Premise Integration

Many customers will still have a strong on-premise landscape and on-premise integration scenarios. This scenario is planned to be supported by a hybrid deployment option to avoid sending data to and receiving data from the cloud.

## Cloud Integration Runtime

Today there's already the option to use the Cloud Integration runtime on SAP Process Orchestration. The design and implementation are done in the web UI of Cloud Integration. In the next step, the integration flow is deployed to the Cloud Integration runtime of SAP Process Orchestration. This allows leveraging the newly available features on Cloud Integration to implement integration scenarios, such as on-premise to on-premise.



Cloud

Ground

## Hybrid Deployment Option

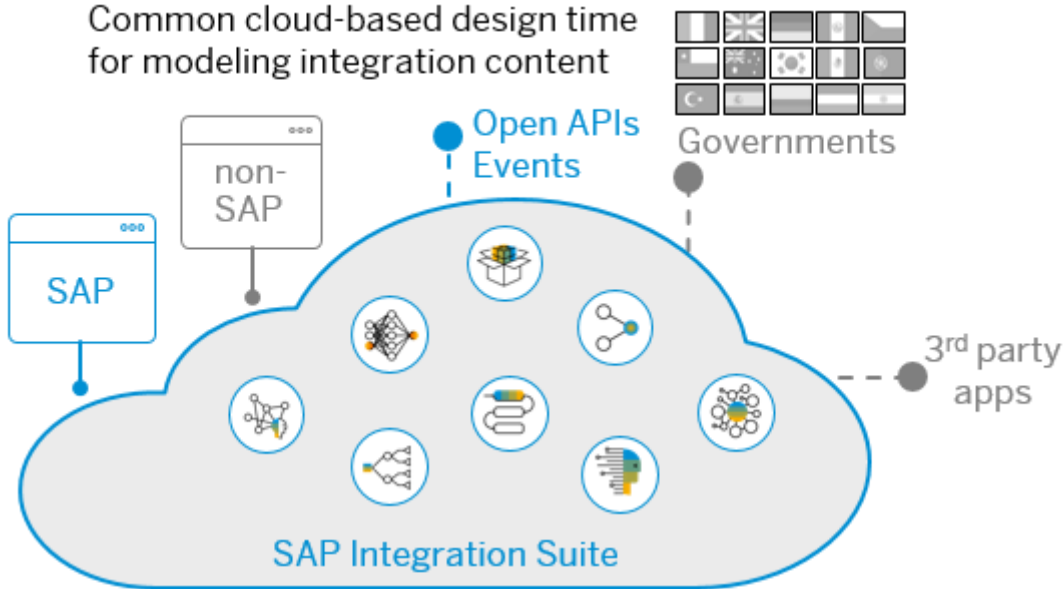
### ! Restriction

This feature is planned for the future and is subject to change. For more information on new features and future releases, access the [Road Map Explorer](#) and the [What's New section](#) of SAP Integration Suite.

SAP is working on providing a lightweight runtime allowing you to have an on-premise runtime, which can be used for all SAP Integration Suite scenarios (e.g. on-premise to on-premise integration processes).

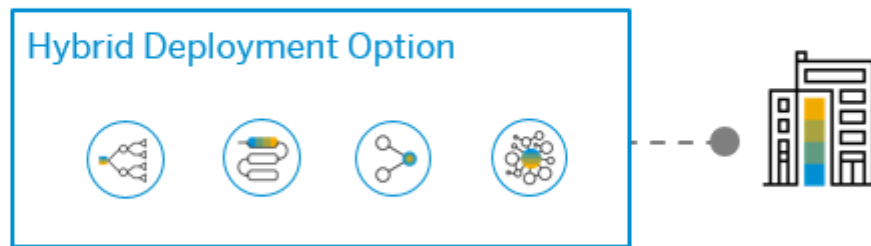
- Design, configuration, and monitoring will be on the cloud
- The runtime will leverage container technology such as Docker
- The runtime can hold any SAP Integration Suite services or engines
- The software lifecycle of the hybrid deployment option shall be in sync with the cloud delivery speed

Common cloud-based design time for modeling integration content



Cloud

Ground



SAP Integration Suite services and runtimes in a containerized environment

## Identity Provider

Learn more about identity providers and user governance in SAP BTP.

The users of SAP BTP are provided by identity providers. You can either use the default SAP identity provider (SAP ID service) with your email ID or configure your own corporate identity provider (e.g. SAP Cloud Identity Services - Identity and Authentication service or Azure Active Directory). If you use an external identity provider, you must configure a trust relationship between SAP BTP and the identity provider.

For more information on how to connect with different Active Directory services, see the tutorial [Integrate Microsoft Azure AD with SAP BTP, Cloud Foundry environment](#) and the blog [How to integrate Azure AD with SAP BTP, Cloud Foundry environment](#).

Additionally, you can enable single sign-on using the Identity Authentication service and an external Active Directory as described in the following tutorials:

- [Connect Azure Active Directory to Identity Authentication Service](#)
- [Register SAP BTP, Cloud Foundry Subaccount in Identity Authentication Service](#)

### i Note

Once the custom IDP configuration is complete, it's necessary to grant permissions to users for SAP BTP applications through the SAP BTP cockpit before they try to access them. See [Configuring User Access to the Application](#).

You can find the different personas for administering the roles in [Persona](#) in the Cloud Integration documentation and in SAP Note [2872526](#) on roles in API Management.

## User Governance

The global account and subaccounts get their users from identity providers. Administrators make sure that users can only access their dedicated subaccount by making sure that there's a dedicated trust relationship only between the identity providers and the respective subaccounts. Developers configure and deploy application-based security artifacts containing authorizations, and administrators assign these authorizations using the SAP BTP cockpit.

The SAP Integration Suite is a bundle of services running on SAP BTP. The authorization groups, roles, and their assignments of the SAP Integration Suite are managed through the SAP BTP cockpit and can be managed via web browser. There's also the option to perform Authorization Management using SAP BTP APIs.

Using [SAP BTP Authorization Management REST APIs](#) provides functionality to manage roles and their assignments to users. It's not limited to SAP Integration Suite and can be extended to other subscriptions and services under SAP BTP subaccounts. These APIs can be used in cases where user assignment needs to happen in a controlled manner to enforce security policies, audit, and compliance from SAP GRC or other user management and governance solutions.

Authorization Management API – Operations:

- Groups – Manage groups and their assignments to users and roles within the specified account.
- Roles – Manage roles and their assignments to users and groups in the specified account and application.
- Users – Manage role and group assignments to the specified user.

## Connectivity

Get an overview of the important aspects regarding the connectivity migration, for example, where it's necessary to identify the systems in use, their protocols, if custom adapters are used, and more.

### [Steps in a Migration](#)

Follows these several structured steps to migrate your communication channels from an SAP Process Orchestration format to a Cloud Integration format.

### [Connected Systems and Protocols](#)

When analyzing your SAP Process Orchestration system, extracting as many details as possible with regards to what systems it connects to and which transport and messaging protocols it uses is the first step.

### [Standard Adapter Migration](#)

SAP Process Orchestration provides many adapter types as standard. Depending on the adapter type, an equivalent may be available in Cloud Integration, or an alternative may need to be used.

### [Custom Adapter Migration](#)

If you, or your implementation partner, have implemented fully bespoke adapters for SAP Process Orchestration, you may need to have these reimplemented using the Adapter Development Kit (ADK) provided for Cloud Integration.

### [Adapter Modules](#)

Learn about the differences between adapter modules in SAP Process Orchestration and in Cloud Integration, and how you can redesign the adapter modules.

### [Linking Communication Channels to Interfaces](#)

In Cloud Integration, there's no separation between the communication channels and the other artifacts (such as Message Mapping Schema definitions) that make up an interface. It's therefore key to understand which communication channels are used by which interfaces as the same channel can be used by more than one interface.

### [Migration of Value Mappings](#)



Learn how you can import value mappings from SAP Process Orchestration to Cloud Integration.

### [Connecting to On-Premise Systems Using Cloud Connector](#)

Find out how Cloud Connector can help you to connect cloud-based services and on-premise systems.

## Steps in a Migration

Follows these several structured steps to migrate your communication channels from an SAP Process Orchestration format to a Cloud Integration format.

- [Assess and document the adapter configurations in use](#)
- [Group and classify configurations according to key criteria](#)
- [Map standard adapters from SAP Process Orchestration to Cloud Integration](#)
- [Reimplement custom adapters or explore alternatives](#)
- [Reimplement adapter modules or explore alternatives](#)

### Assess and Document the Adapter Configurations in Use

The integration artifacts in SAP Process Orchestration differ from those in Cloud Integration. In SAP Process Orchestration, there's a clear distinction between design time objects (interfaces, mappings, ...) and configuration time objects (channels, Integrated Configuration objects). In Cloud Integration, there's no distinction between those objects as everything is encapsulated within one artifact called integration flow.

In SAP Process Orchestration, adapter configurations are maintained in communication channel objects. Each communication channel defines which transport protocol to use, specified as an adapter type, and several configuration parameters that define, for example, hostname, port, user, and password to the remote system.

Each communication channel in use in a particular SAP Process Orchestration system needs to be analyzed in order to map it to a corresponding communication channel definition in Cloud Integration.

Details on this topic are discussed in the sections [Connected Systems and Protocols](#) and [Standard Adapter Migration](#).

### Group and Classify Configurations According to Key Criteria

Once the adapter configurations have been extracted as communication channel definitions, group and classify them by several key criteria. These criteria can include, but aren't limited to, the following considerations:

- **Adapter direction** (sender or receiver) – some adapter types are only available in one direction in Cloud Integration (for example the RFC adapter)
- **Adapter type** – standard SAP Process Orchestration adapter or custom adapter
- **File Content Conversion usage** (Yes/No)
- **Advanced filename selection usage** (Yes/No)
- **Quality of Service (QoS) requirements** – Is Exactly Once in Order (EOIO) required, or Exactly Once (EO)
- **Protocol** – For example, if it's an HTTP, non-HTTP or File-based service, this can help while choosing a migration strategy for a similar adapter supported in .
- **Retry strategy for asynchronous interfaces** (if any)
- **Use of custom/special adapter modules**

## Map Standard Adapters from SAP Process Orchestration to Cloud Integration

In addition, some adapters in Cloud Integration may have feature restrictions in comparison to the SAP Process Orchestration equivalent. It's therefore critical to any migration to analyze all adapter types in use on SAP Process Orchestration and evaluate their availability on Cloud Integration or seek a replacement solution where an adapter type isn't available.

Details on this topic are discussed in the section [Standard Adapter Migration](#).

## Reimplementation Strategy for Custom Adapters

If you, or your implementation partner, have implemented fully bespoke adapters for SAP Process Orchestration, you may need to have these reimplemented using the [Adapter Development Kit \(ADK\)](#) provided for Cloud Integration.

Alternatively, you can search for an OEM adapter equivalent, or use the [SAP Open Connectors service](#) to connect to non-SAP solutions.

Details on this topic are discussed in the section [Custom Adapter Migration](#).

## Reimplementation Strategy for Adapter Modules

SAP Process Orchestration supports the concept of adapter modules (generally implemented as Enterprise JavaBeans, EJB, and deployed on the SAP Process Orchestration Java Server). These provide the capability to further manipulate the incoming (for sender communication channel) or outgoing (for receiver communication channel) message payload before either processing the XML message within the SAP Process Orchestration messaging engine or sending the final message out to the target system.

Cloud doesn't support this concept, but instead supports the concept of introducing additional conversion steps into any integration flow to support any additional manipulation required before full processing, or transmission to the target system. This is arguably a much more transparent and flexible approach, but it also means that a conversion process needs to take place. Another option is to create a custom adapter providing the same features as the standard adapter and extend it with the additional functionality.

Details on this topic are discussed in the section [Adapter Modules](#).

## Connected Systems and Protocols

When analyzing your SAP Process Orchestration system, extracting as many details as possible with regards to what systems it connects to and which transport and messaging protocols it uses is the first step.

This information about connected systems and protocols can then be further analyzed to identify any issues or challenges ahead, and to allow for an estimation of the effort required to migrate to a Cloud Integration tenant solution.

### ! Restriction

Some features mentioned on this site are planned for the future and are subject to change. For more information on new features and future releases, access the [Road Map Explorer](#) and the [What's New section](#) of SAP Integration Suite.

To extract this information, the following techniques can be used:

1. Utilize the standard Web Services that are available as part of the Integration Directory API. For details on this method, see the blog [SAP PI/PO Directory API: Extract detailed Communication Channel configurations into an Excel sheet without custom codes/macros](#).

- Utilize the upcoming SAP PI/PO Assessment Tool: Data Extraction Utility. This tool provides the capability to extract not only the communication channel data, but also details regarding each integrated configuration object, as well as the design time artifacts such as operation mappings and message mappings. It uses a Java Swing-based UI to provide a more user-friendly approach to extracting the data. Connectivity to your SAP Process Orchestration system is defined using the [Settings](#) dialog.

## Standard Adapter Migration

SAP Process Orchestration provides many adapter types as standard. Depending on the adapter type, an equivalent may be available in Cloud Integration, or an alternative may need to be used.

For detailed documentation of the adapters in Cloud Integration, see [Configure Adapter in Communication Channels](#).

### ! Restriction

Some adapters mentioned on this site are planned for the future and are subject to change. For more information on new features and future releases, access the [Road Map Explorer](#) and the [What's New section](#) of SAP Integration Suite.

## Standard Sender Adapter Types

SAP Process Orchestration	Cloud Integration	Restrictions/Alternatives
AS2	AS2, AS4	
EDISeparator	Available through AS2, AS4, or SFTP adapter	Alternative – Possible use of AS2/AS4/SFTP adapters with EDI Splitter step or EDI Validator step in integration flow.
File	Available through FTP or SFTP adapter	Alternative – Possible use of FTP or SFTP adapters.  File Content Conversion can be performed in standard step CSV to XML or using a script step.
HTTP_AAE	HTTP	
IDoc_AAE	IDoc	Cloud Integration supports IDoc in HTTP protocol, which means that the traditional IDoc with packing (via RFC) can't be used.
JDBC	Planned for future	Alternative – Use timer event along with JDBC receiver adapter to achieve the same JDBC polling mechanism.
JMS	AMQP	Although Cloud Integration supports the JMS adapter, it can only be used internally and not to external JMS servers. For this scenario, the AMQP adapter should be used instead.
Mail	Mail	
Marketplace	Available through HTTP adapter	Alternative – Possible use of the HTTP adapter with implementation of XSLT to convert from MML and XML or JSON.

SAP Process Orchestration	Cloud Integration	Restrictions/Alternatives
OData	OData	SAP is working on supporting multiple cloud and on-premise database vendors.
OFTP (Odette FTP)	Available through external application	Alternative – Possible use of an external application to make a bridge between HTTP requests and OFTP.
REST	Planned for Cloud Integration Currently implemented through HTTP adapter	Alternative – Use integration flow steps to handle HTTP verbs and decomposition of URIs.
RFC	Available through Web service	Alternative – Implement as a Web service in Cloud Integration and build an outbound SOAP proxy in your ABAP system to replace the RFC call.
SFSF	SuccessFactors	
SFTP	SFTP	Limited support for advanced file selection.
SOAP	SOAP	
WS_AAE	Implemented through SOAP adapter	
X400	Planned for future	
Other adapters: BC, CIDX, Ebics, RNIF (RosettaNet Implementation Framework), RNIF11	Planned for future	

## Standard Receiver Adapter Types

SAP Process Orchestration	Cloud Integration	Restrictions/Alternatives
AS2	AS2, AS4	
EDISeparator	Available through AS2, AS4, or SFTP adapter	Alternative – Possible use of AS2/AS4/SFTP adapters with EDI Splitter step or EDI Validator step in integration flow.
File	Available through FTP or SFTP adapter	Alternative – Possible use of FTP or SFTP adapters.  File Content Conversion can be performed in standard step CSV to XML or using a script step.
HTTP_AAE	HTTP	
IDoc_AAE	IDoc	Cloud Integration supports IDoc in HTTP protocol, which means that the traditional IDoc with packing (via RFC) can't be used.
JDBC	JDBC	Some limitations to which database vendors are currently supported.

SAP Process Orchestration	Cloud Integration	Restrictions/Alternatives
JMS	AMQP	Although Cloud Integration supports the JMS adapter, it can only be used internally and not to external JMS servers. For this scenario, the AMQP adapter should be used instead.
Mail	Mail	
Marketplace	Available through HTTP adapter	Alternative – Possible use of the HTTP adapter with implementation of XSLT to convert from MML and XML or JSON.
OData	OData	SAP is working on supporting multiple cloud and on-premise database vendors.
REST	Planned for Cloud Integration Currently implemented through HTTP adapter	Alternative – Use integration flow steps to handle HTTP verbs and decomposition of URIs.
RFC	RFC	
SFSF	SuccessFactors	
SFTP	SFTP	Limited support for advanced file selection
SOAP	SOAP	
WS_AAE	Implemented through SOAP adapter	
X400	Planned for future	
Other adapters: BC, CIDX, Ebics, OFTP (Odette FTP), RNIF (RosettaNet Implementation Framework), RNIF11	Planned for future	

## Custom Adapter Migration

If you, or your implementation partner, have implemented fully bespoke adapters for SAP Process Orchestration, you may need to have these reimplemented using the Adapter Development Kit (ADK) provided for Cloud Integration.

It may be of value to assess the marketplace to determine whether an OEM partner has already implemented an adapter that provides the connectivity you require. Some of the adapters provided by SAP partners are fully SAP-certified and fully supported for several non-SAP systems. There are many adapters like Salesforce, ServiceNow, Workday, etc. available out-of-the-box in SAP Integration Suite. For a full list of available adapters, see [Configure Adapter in Communication Channels](#).

You can search on [SAP Business Accelerator Hub](#) and the internet for different OEM partners and their adapters.

If an OEM adapter isn't available, an additional alternative is to use the SAP Open Connectors service, which is one of the key complimentary components of the SAP Integration Suite. Open Connectors provides out-of-the-box connectivity to more than 170 non-SAP applications or services. It provides harmonized RESTful APIs with built-in interactive API documentation using OpenAPI Specification and normalized authentication, which provides built in security capabilities for secure robust connectivity to third-party applications. It also drastically reduces the time needed to connect to top non-SAP SaaS applications.

Once a connection instance to a particular non-SAP system has been configured, the built-in Open Connectors adapter in Cloud Integration can seamlessly connect to it using configurable authentication credentials to perform the usual CRUD (Create, Read, Update, Delete) operations.

The following blogs provide details on the topic:

- [Google Calendar's Info with API Management and Open Connectors Services](#)
- [How to create a sample integration scenario using Open Connectors Adapter](#)
- [The SAP Open Connectors service by Cloud Elements](#) (video)

If neither of these options provides a solution, the final option is the reimplementing of your custom adapter using the ADK. As both Process Orchestration and Cloud Integration are based on Java, and the adapters are, too, a significant proportion of coding can be ported easily.

The ADK is available for download from [SAP Development Tools - Cloud Integration](#). The deployment of adapters can be done via the Web UI of Cloud Integration.

The blog [Cloud Integration Adapter development using Maven Project perspective](#) walks you through the process of adapter development by illustrating the creation of a simple sample adapter. Additionally, the blog [Cloud Integration - Developing Custom Adapters to access On-Premise systems](#) illustrates the creation of a custom adapter that is able to communicate with on-premise systems via Cloud Connector.

## Adapter Modules

Learn about the differences between adapter modules in SAP Process Orchestration and in Cloud Integration, and how you can redesign the adapter modules.

In SAP Process Orchestration, you can add a standard or custom adapter module while you're defining the properties of your communication channel. This allows you to perform different transformations like zip and encrypt the message, or change and remove the namespace in the XML. Although the same concept doesn't apply to adapter modules in Cloud Integration, you can transform the message using the standard integration flow steps or creating a JavaScript or Groovy script. The following table displays some strategies that can be used to redesign these adapter modules:

SAP Process Orchestration	Description	Reimplementation strategy
AF_Modules/PayloadSwapBean	You use this module to replace the application payload of the XI message that contains the business data with another payload that is appended to the XI message as an additional attachment.	<p>This module is normally used while sending a single attachment as the application payload with Mail adapter.</p> <p>Cloud Integration offers multiple options to handle this:</p> <ul style="list-style-type: none"> <li>• map properties as attachments directly in the Mail adapter by default</li> <li>• use Groovy script to create attachments in a step before Mail call</li> <li>• read attachments and set them as main payload after sender mail channel</li> </ul>

SAP Process Orchestration	Description	Reimplementation strategy
AF_Modules/StrictXml2PlainBean	You use this module to convert an XML document that is in the main payload of the XI message to text format.	In Cloud Integration, you have different ways to convert your XML document to text format. The most common way is to use the conversion steps available (e.g., XML to CSV, XML to EDI, XML to JSON). You can also use Groovy script to convert the XML message as you want in a step before the adapter call.
AF_Modules/XMLAnonymizerBean	You use this module to anonymize XML elements and attributes by removing namespaces or namespace prefixes from the XML document of the main payload.	In Cloud Integration, you can use an XSLT in the step before the adapter call to change/remove the namespaces or prefixes.  You can also use Groovy script to achieve it.
SAP_XI_IDOC/IDOCXmlToFlatConvertor	You can use this module to convert IDoc XML to IDoc Flat format. If you add this module to the module chain, then it converts the data in the XI message payload from IDoc XML to IDoc Flat format. For example, you can add this module to the module chain of a receiver file adapter and create an IDoc Flat file.	In Cloud Integration, you can use the standard XML to CSV converter to achieve it.
SAP_XI_IDOC/IDOCFlatToXmlConvertor	You can use this module to convert the data in the XI message payload from IDoc Flat format to XML format. For example, you can add this module in the module chain of a sender file adapter and create IDoc XML from IDoc Flat format.	In Cloud Integration, you can use an external library or develop a custom Groovy/Javascript script to perform this conversion.
AF_Modules/PayloadZipBean	You use this module to compress one or more payloads or extract payloads from a compressed file.	In Cloud Integration, you can use the standard Encoder and Decoder steps to compress or extract payloads in ZIP, GZIP, Base64, or MIME.
AF_Modules/DynamicConfigurationBean	You use this module to edit the message header for adapter-specific message attributes. You can add attributes to the header or delete attributes from the header.	In Cloud Integration, you don't have the same concept of adapter-specific attributes but you can use external parameters or runtime parameters (e.g., <code>#{property.queryOptions}</code> ) while defining dynamic attributes in your adapter.  For more information about external parameters, see the blog <a href="#">Externalizing parameters using SAP Cloud Platform Integration's Web Application</a> .

You can find more information of these standard adapter modules in the help documentation on [Adding Modules to the Module Processor](#).

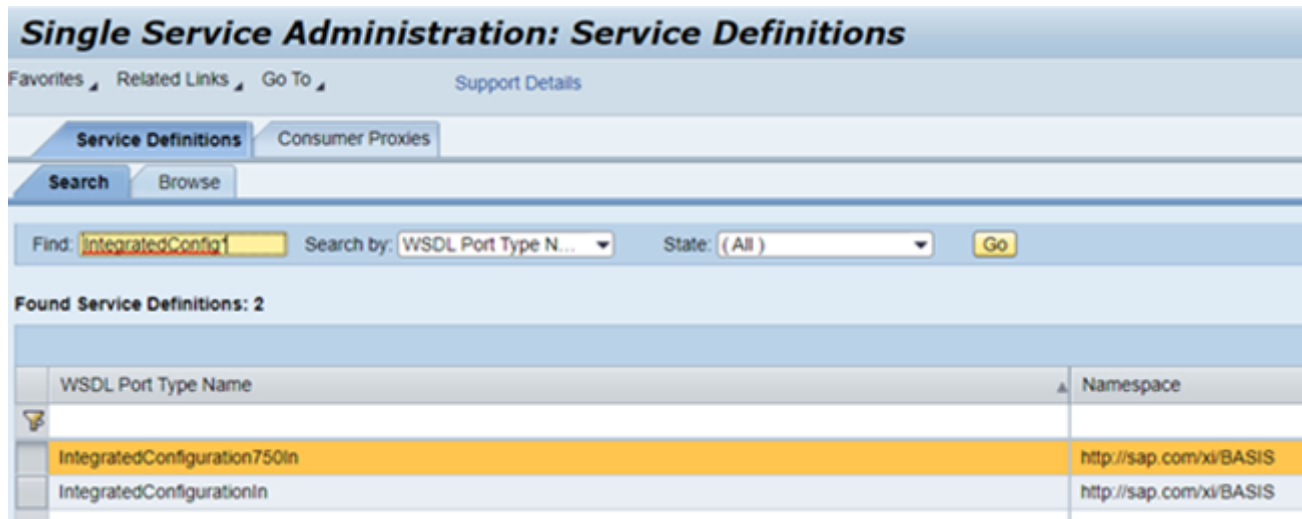
## Linking Communication Channels to Interfaces

In Cloud Integration, there's no separation between the communication channels and the other artifacts (such as Message Mapping Schema definitions) that make up an interface. It's therefore key to understand which communication channels are used by which interfaces as the same channel can be used by more than one interface.

To achieve this understanding, a list of Integrated Configuration objects needs to be extracted and analyzed to provide the linkage between the Integrated Configuration objects and the communication channels the interface uses.

Several techniques can be employed to extract this information, for example the one described in the following:

Like the communication channel extraction, utilize the standard Web services that are available as part of the Integration Directory API. The blog [SAP PI/PO Directory API: Extract detailed Communication Channel configurations into an Excel sheet without custom codes/macros](#) provides details on how to extract the information for the communication channels. A similar approach can be employed, but in this case using an alternative Web service. Go to [SAP NetWeaver Administrator > Configuration > Connectivity > Single Service Administration](#), and in [Service Definitions](#), search for *IntegratedConfiguration750In* instead of *CommunicationChannelIn*:



Some of the columns that can be found in the extraction after converting to MS Excel are:

- Sender Party ID (Interface Configuration Object key)
- Sender Component ID (Interface Configuration Object key)
- Interface Name (Interface Configuration Object key)
- Interface Namespace (Interface Configuration Object key)
- Party ID (Communication Channel key)
- Component ID (Communication Channel key)
- Channel ID (Communication Channel key)

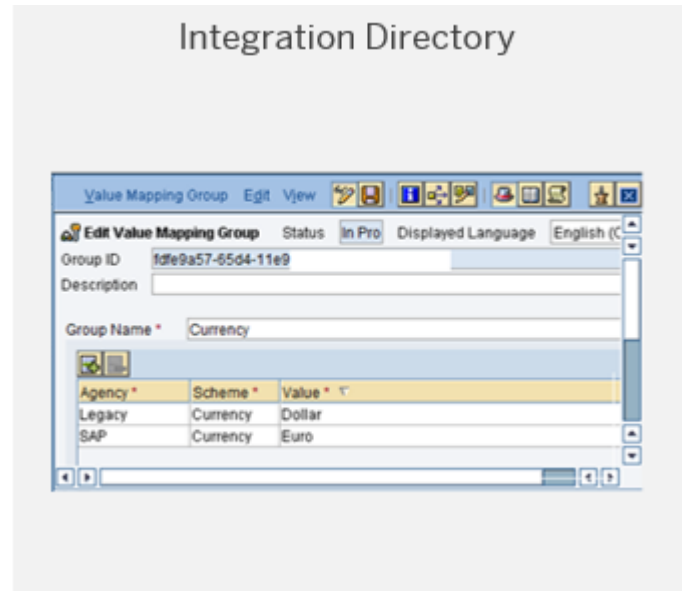
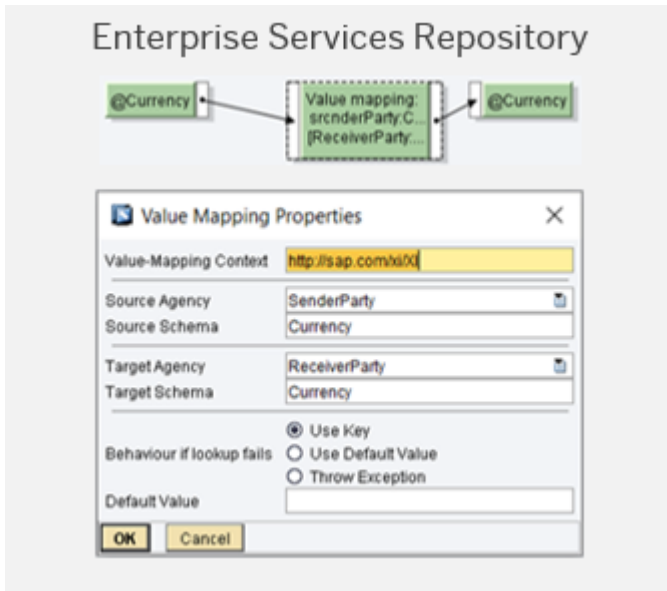
With that, you can apply pivot tables and formulas to improve the visualization of key data.

## Migration of Value Mappings

Learn how you can import value mappings from SAP Process Orchestration to Cloud Integration.

In SAP Process Orchestration, the value mapping standard function is defined in ESR and is used to map different representations of an object to each other. At the same time, it's needed to define a value mapping group (value-mapping tables) in Integration Directory. The image below shows the representation of the value mapping defined in SAP Process Orchestration.





This function can be used in different message mappings with the same key field, but the same value mapping group need to be referenced. Additionally, you can choose to use a rather static option called fixValues, which is a key-value list in which you can directly insert the value mapping in the message mapping while designing your interface in ESR. The fixValue mappings are automatically imported when importing the Message Mapping or Operation Mapping from SAP Process Orchestration.

For more information, see [Designing and Configuring Value Mappings](#).

In Cloud Integration, the value mapping configuration is similar. It's possible to import a CSV file containing the values. After the CSV file importation and deployment, on the message mapping, you need to add the value mapping function and fill in the identifier and agency. For more information, see [Creating Value Mapping](#).

It's possible to export a CSV file from SAP Process Orchestration and import it in Cloud Integration in a value mapping. If the value mapping already exists in Cloud Integration, the process only works if you import a CSV in an existing value mapping, too. The values are merged with existing Agency and Identifier entries.

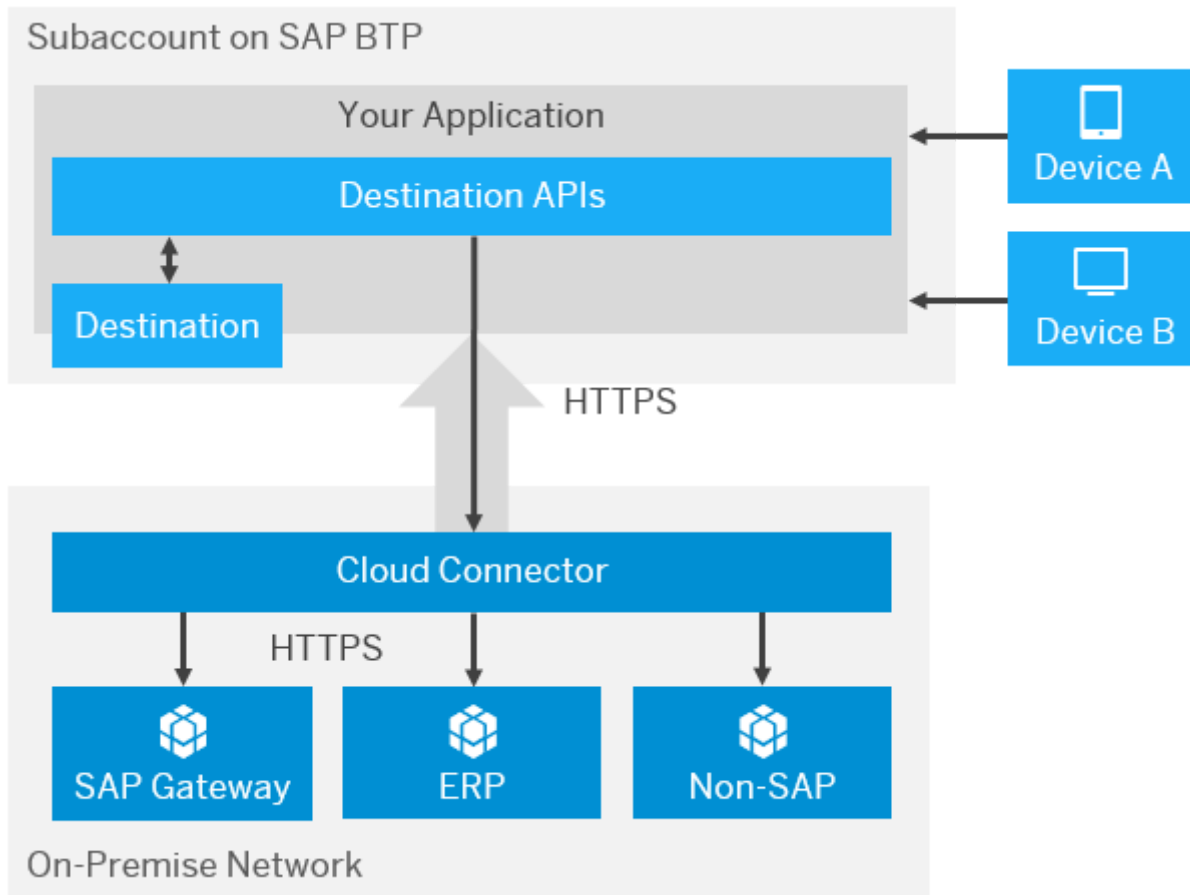
Another way to import value mappings from SAP Process Orchestration to Cloud Integration using the SOAP API (Directory APIs) from of SAP Process Orchestration. This API exports the value mappings as a XML file. After this step, the XML file must be converted to CSV format and follow the process described in the previous paragraph.

## Connecting to On-Premise Systems Using Cloud Connector

Find out how Cloud Connector can help you to connect cloud-based services and on-premise systems.

While businesses across the globe are rapidly adopting a cloud-centric approach to their IT landscapes, many will inevitably end up with a hybrid environment whereby some software systems reside on-premise and others have moved to the cloud as subscription-based services. Due to the requirements imposed by the diverse nature of many business processes and their need to interact with multiple software systems in a fully connected manner, integration between the cloud-based services and the on-premise systems is an inevitable necessity.

SAP Cloud Connector was implemented as a solution to bridge the gap between on-premise and the cloud. The SAP Cloud Connector is an on-premise piece of software that needs to be installed inside your landscape. Once configured and paired with your SAP BTP account, a secure tunnel is established between SAP BTP (and all the services and applications that run on it) and the Cloud Connector. This way, all communication between SAP BTP and the backend system gets routed via the Cloud Connector over the secure SSL tunnel. As a result, the access control now only needs to be configured in the Cloud Connector. Despite having a light footprint, SAP Cloud Connector is a secure, richly featured application that provides full logging, auditing, and traceability features together with the capability to be installed as a cluster to ensure high availability.



## Installation of Cloud Connector

SAP Cloud Connector is freely available to download and install from [SAP Development Tools](#). Several installation types are available providing support for Windows, Linux and Mac OS X.

For more information about how to install Cloud Connector, see [SAP BTP Connectivity](#).

## Setup of Cloud Connector

Cloud Connector can be used to control access for both On-Premise-to-Cloud and Cloud-to-On-Premise communication. However, the On-Premise-To-Cloud scenario is only used when connecting an on-premise application with an SAP HANA database on SAP BTP. Since the SAP Integration Suite can only utilize the feature to set up a connection to on-premise systems, the trigger of the integration scenario must always be SAP BTP.

Currently, the following protocols are supported:

- HTTP
- RFC (currently Cloud Integration only provides an RFC receiver adapter, so no RFC calls from an on-premise application to Cloud Integration are possible)
- LDAP
- TCP (to support protocols such as SFTP and FTP)

### ⚠ Caution

TCP-based connections can pose a security risk by permitting unmonitored traffic, so ensure only trustworthy applications have access. Because it uses plain TCP, the Cloud Connector can't see or log any detail information about the calls. Therefore, in contrast to HTTP or RFC (both running on top of TCP), the Cloud Connector can't check the validity of a request.

To minimize this risk, make sure you

- deploy only trusted applications on SAP BTP,
- configure an application allow list in the Cloud Connector, as described in [Set Up Trust for Principal Propagation](#), and
- take the recommended security measures for your SAP BTP (sub)account, as described in section [Security](#) in the SAP BTP documentation.

For more information on configuration, operation, and security, see the [Cloud Connector](#) help documentation.

## Example of Use in an Integration Flow

There are different use cases to establish connection to an On-Premise System. In the blog [Cloud Integration - Connect CPI with Your On-Premise Mail Server](#), a connection to an on-premise Mail server is established.

## Access Control

The Cloud Connector offers the feature to allow only specific services and paths to be exposed to SAP BTP. This can be specified with the Access Control. For help documentation on setting up access for each protocol type, see the following:

- Access control for [HTTP](#)
- Access control for [RFC](#)
- Access control for [LDAP](#)
- Access control for [TCP](#)

## Cloud Integration Using Cloud Connector

Once the appropriate access control has been set up, Cloud Integration can use the Cloud Connector with a simple configuration step in the adapter channel.

The key configuration settings are illustrated in the following graphic using an SFTP adapter as an example:

**SFTP**

General **Target** Processing

FILE ACCESS PARAMETERS

Directory: /indir

File Name: file\_1234.txt

Append Timestamp:

CONNECTION PARAMETERS

Address: sftp.localhost:22

Proxy Type: On-Premise

Location ID:

Authentication: User Name/Password

Credential Name: sftp.localhost\_user

Timeout (in ms): 10000

The first key change is to set the **Proxy Type** to **On-Premise** (rather than **Internet**), at which point the **Location ID** field becomes visible. If you have multiple Cloud Connector instances connected to a single SAP BTP tenant subaccount, you need to specify a unique location ID when setting up the connections. In this situation, you must specify the location ID of the correct Cloud Connector you wish to connect to as part of the channel configuration. If you only have a single Cloud Connector instance with no location ID, this can be left blank.

The second key information is the **Address** (the URL of the target server) and must correspond to the URL specified in the destination definition.

The last key part is to provide the correct authentication credentials.

## Security

Discover the important aspects regarding security that must be evaluated before migrating from SAP Process Orchestration to Cloud Integration.

Using a cloud-based integration platform imposes dedicated security requirements on the software vendor (SAP) that hosts the platform, as well as on the customers who use the platform. Customers who use Cloud Integration agree that a significant part of their (and their customers') sensitive data is processed by and stored within an infrastructure not owned by themselves. The core task of an integration platform is to serve as the transit place for messages that can contain sensitive customer data. First and foremost, these messages must be protected against eavesdropping and unauthorized access.

Therefore, the integration platform must fulfill the following main requirements:

- The integration infrastructure is already designed and built in such a way that it meets the highest security standards. For more information on compliance, visit the [SAP Trust Center](#).

It must be guaranteed that the technical system landscape, the communication between the components of the integration platform, and the storage locations of messages are secure.

- The processes related to the usage of Cloud Integration meet the highest security standards.

These processes include processes at SAP related to the development and upgrade of the Cloud Integration software, the processes related to the provisioning and operation of the customers' virtual environment by the infrastructure provider, and the customer onboarding process during which customers set up secure connections between their infrastructure and SAP's integration platform.

- Customers have several options to configure how messages are exchanged within an integration scenario so that the involved data is protected at the highest level.

When designing integration flows, customers can choose between several options to protect messages by establishing secure communication channels (transport-level security) and by configuring digital encryption and digital signing of messages (message-level security).

For full help information on the security capabilities of Cloud Integration in the Cloud Foundry environment, see [Security, Cloud Foundry Environment](#).

For access to the recommended Learning Journey to learn all about security from an SAP BTP perspective, go to [Security with SAP BTP](#).

## Securing User Access to Cloud Integration

A customer can decide to either use the identity provider [SAP ID service](#) or add their own compatible identity provider such as Microsoft Azure Active Directory. The SAP ID service is initially used to authenticate that a particular user is who they claim to be by the user entering credentials (such as a username and password) to provide proof. Once authentication has been verified,

the next stage (authorization check) takes place to determine what access to features and functions the user has for a particular application.

Cloud applications hosted on the SAP BTP typically use role-based access control. This means that authorizations to access specific applications (such as Cloud Integration) and execute specific tasks (for example, Create and Edit Artifacts, Deploy Security Material, Monitor Integration Flows) are provided to each user through the assignment of roles and role collections.

The provision of roles and role collections to users is a task typically performed by a Tenant Administrator using the security tools provided on the SAP BTP Cockpit. For detailed information on how to configure user access to the Cloud Integration application by assignment of roles and role collections, see [Configuring User Access to the Application](#).

The help documentation on [Tasks and Permissions](#) provides a detailed table of the roles and roles collections available as standard to control access to each specific task and feature within Cloud Integration. The table provides the corresponding information for both the Cloud Foundry and the Neo environment.

## Managing Security Material in Cloud Integration

To securely transmit messages from a source system to a target system via middleware such as Cloud Integration, a number of key types of security components must be stored and managed securely on each Cloud Integration tenant.


These security components can be categorized as follows:

- [Configure Inbound HTTP Connection](#)
- [Security Material](#)
- [Keystore Entries](#)
- [Access Policies](#)
- [JDBC Material](#)
- [User Roles](#)

Before using such security material to start transmitting or receiving messages, it can be useful to test their validity, for example by executing a connectivity test.

### Configure Inbound HTTP Connection

Various sender adapters allow you to enable a sender system to send messages to Cloud Integration over the HTTP protocol. For inbound HTTP connections, a load balancer component connects the remote sender system and the Cloud Integration tenant. There are multiple options of user authentication for user role authorization:

Authentication	Description
<a href="#">Client certificate authentication</a>	<p>Using this option, authentication of a sender is performed based on a client certificate. At runtime, the system checks if a service key that contains the client certificate provided by the sender is available. If a service key is available, the system checks if the associated service instance has a role specified that grants permissions to call the integration flow endpoint.</p> <p>An SAP BTP tenant administrator can generate the <a href="#">service key</a> used for certificate-based authentication.</p> <p>This process is also describes in the blog <a href="#">How to Set Up Secure HTTP Inbound Connection with Client Certificates</a> .</p>

Authentication	Description
<a href="#">OAuth with client credentials grant</a>	<p>The sender is authenticated based on an OAuth access token. The access token is retrieved in a preceding call from an authorization server.</p> <p>An SAP BTP tenant administrator can generate the <a href="#">service key</a> used for OAuth with client credentials grant.</p>
<a href="#">Basic Authentication for a user with client credentials from the service key</a>	<p>The sender is authenticated based on user credentials that are generated together with a service key.</p> <p>The steps to create service instance and service key are the same as for the option "OAuth with client credentials grant". However, during runtime no access token is retrieved. Instead, the values of clientid and clientsecret from the service key are used for authentication.</p>
<a href="#">Basic authentication for user registered with an identity provider</a>	<p>The sender is authenticated based on user credentials associated with a user registered at an identity provider (IDP).</p>

## Managing Security Material

On the **Operations** page of the Cloud Integration tenant, the **Manage Security** area provides an overview of security-related artifacts. Select the tile **Security Material** to open the area **Manage Security Material**, in which security materials can be created and uploaded.

The following types of security material are supported:

Material Type	Description
User Credentials	Used for Basic Authentication (with Username / Password combination)
OAuth2 Client Credentials	<p>Many Web servers nowadays use OAuth credentials for authorization purposes. Cloud Integration provides the capability to store and use such credentials to allow the system to connect at runtime.</p> <p>For deployment information, see <a href="#">Deploying an OAuth2 Client Credentials Artifact</a>.</p>
OAuth2 SAML Bearer Assertion	<p>Used for accessing OAuth protected resources by propagating the user identity</p> <p>For more information, see the blog <a href="#">OAuth2 SAML Bearer/X.509 Certificate Authentication Support in SuccessFactors Connector</a>.</p> <p>For deployment information, see <a href="#">Deploying an OAuth2 Bearer Assertion</a>.</p>
OAuth2 Authorization Code	<p>This is another form of use of OAuth2: if you use an OAuth2 authorization code, the authentication is performed via a separate authorization server as an intermediary step. The client can exchange the OAuth2 authorization code for an access token that is granted by the authorisation server and typically has a restricted Time-To-Live (TTL). This way the actual user credentials are never shared with the client, and the owner of the user credentials can revoke the authorisation code at any time.</p> <p>For deployment information, see <a href="#">Deploying an OAuth2 Authorization Code</a>.</p>
Secure Parameters	<p>Used to deploy confidential data, for example, key information such as passwords that may be required for custom adapters.</p> <p>For deployment information, see <a href="#">Deploying a Secure Parameter Artifact</a>.</p>
Known Hosts (SSH)	<p>The known hosts file contains the public keys for all hosts with which the Cloud Integration tenant intends to communicate by using the Secure Shell (SSH) protocol.</p> <p>For deployment information, see <a href="#">Deploying an SSH Known Hosts Artifact</a>.</p>

Material Type	Description
PGP Public Keyring	<p>This artifact contains one or more public keys that enable the Cloud Integration tenant to encrypt or verify messages using the Pretty Good Privacy (PGP) encryption standard as part of support for Message Layer Security (MLS).</p> <p>For deployment information, see <a href="#">Deploying a PGP Public Keyring</a>.</p>
PGP Secret Keyring	<p>This artifact contains one or more public and private key pairs for the usage of Open Pretty Good Privacy (PGP). The private key enables the Cloud Integration tenant to decrypt or sign messages as part of Message Layer Security (MLS).</p> <p>For deployment information, see <a href="#">Deploying a PGP Secret Keyring</a>.</p> <p>For details on how to create a key pair, see <a href="#">Creating a Key Pair</a>.</p> <p>For more information about importing and exporting PGP secret keys, see the blog <a href="#">Import and Export PGP Secret Key - Change PGP Secret Key Password</a>.</p>

## Managing Keystore Entries

The Cloud Integration Operations view contains the Keystore Monitor. The Keystore Monitor allows a tenant administrator to manage the tenant keystore and its entries (X.509 certificates and key pairs).

For more information, see [Managing Keystore Entries](#).

### Migration of Keystore Entries from SAP Process Orchestration to Cloud Integration

The methods to access, import, and export each certificate differ between the systems and the administrative tools. This section provides details on how to migrate certificates in SAP Process Orchestration and Cloud Integration.

#### Export Certificate from SAP Process Orchestration

In SAP Process Orchestration, you can access the keystores by going to **SAP NetWeaver Administrator** > **Configuration** > **Security** > **Certificates and Keys**.

On this page, you can manage the application server Java certificates and keys. For instance, you access the server credentials of multiple virtual keystores called keystore views. The keys and certificates in the keystore views can be used for encryption, identification, and verification purposes when using AS Java functions.

Assuming that the certificate is in **TrustedCAs**, click on that to check the entries associated.



Status	Name	Type	Description
<span style="color: green;">■</span>	SFTP_test	USER	
<span style="color: green;">■</span>	TicketKeystore	SYSTEM	Contains the key-pair to use for issuing logon and assertion tickets
<span style="color: green;">■</span>	TREXKeystore	USER	Contains keys and certificates used by the TREX service
<span style="color: green;">■</span>	TrustedCAs	USER	Template view that contains trusted server certificates
<span style="color: green;">■</span>	UMEKeystore	USER	Contains a key-pair used by the User Management Engine (UME)

In the details of **TrustedCAs**, select the certificate and select **Export Entry**. Save the key/certificate on your local file system.

**Certificates and Keys: Key Storage**

Content Security

Key Storage Views

Key Storage Views

Export Entry To File

Select export format:\*

File Type

Base64 X.509

Binary X.509

No Files for Download

Download Link

Close

Details of view "TrustedCAs"

View Entries

Export Entry

Status	Name	Entry Type	Algorithm	Valid From
Green	.microsoftonline	CERTIFICATE	RSA	Fri May 20 14:51:2

By default, SAP NetWeaver Java comes with some keystore views serving different purposes:

- TicketKeystore (used for issuing logon and assertion tickets)
- UMEKeystore (contains key-pair used by the UME provider service of the AS Java)
- ICM\_SSL\_<instance\_ID> (contains the SSL key pair and trusted server certificates for client authentication over SSL)
- TrustedCAs (contains trusted server certificates)

For more information about how to import a certificate into the keystore view, see [Uploading Certificates to SAP Process Integration](#) and SAP Note [2056672](#).

### Importing Certificates to Cloud Integration

The Keystore Monitor in Cloud Integration allows a tenant administrator to manage the tenant keystore and its entries (i.e., X.509 certificates and key pairs).

Connections between a Cloud Integration tenant and a remote system can be secured in different ways, such as basic authentication, client authentication, or OAuth2. Using client authentication, you must maintain these certificates in the keystore.

In Cloud Integration, you can access the [Keystore](#) tile in the [Manage Security](#) section.

A keystore typically contains entries that belong to the tenant administrator (customer) and entries that are owned by SAP. Each entry is uniquely identified by an alias.

The operations you can perform on a keystore entry depend on whether it's owned by SAP or the tenant administrator. As the tenant administrator, you can, for example, delete or backup entries that are owned by the tenant administrator. SAP can't access or download any keystore entries owned by the tenant administrator.

In Cloud Integration, there's the concept of security material, which is stored in the tenant and enables the reuse of some parameters such as authentication parameters (for Basic authentication), PGP private and public keys (to support encryption and decryption for Message Layer Security), and OAuth2 credentials.

## Managing Access Policies

Access Policies serve two purposes: first, to restrict the tasks that can be performed in an artifact at design-time, and second, to restrict the data that can be displayed at runtime level, such as the access to Message Processing Log Attachments, trace data, data stores, and variables.

This is custom documentation. For more information, please visit the [SAP Help Portal](#)



Access policies provide an extra layer of granularity in terms of controlling how artifacts can be accessed, and by whom, at a deeper level than the standard User Roles provide as part of the standard SAP BTP authorization model.

Currently, access policies can be used to control access to the following integration artifacts and integration content on a Cloud Integration tenant:

- Integration flows
- OData API definitions
- REST API definitions
- SOAP API definitions
- JMS message queues
- Global datastores
- Global variables

A single access policy can be used to control access to multiple types of objects. The actual names of the objects can be specified by using a specific name, or a regular expression (e.g., HR, Finance) can be used to protect multiple objects.

Access policy can help in segregating data even on a single tenant.

## Managing Access Policies, Cloud Foundry Environment

In Cloud Integration, the access policies restrict the access to selected integration artifacts and associated data, provide controlled access, and safeguard the operations from unauthorized users. Thus, access policies are associated with a role and contain references to integration artifacts. It's possible to group some integration flows and provide protection to the business data processed.

The access policies are defined in the design and runtime of integration artifacts and include the protection of integration flows and artifacts, REST and SOAP APIs, and so on.

To define the access policies in Cloud Integration, follow these steps:

1. Define a custom role using the platform cockpit.
2. Define an access policy using the Web UI (Monitor section).
3. Assign the custom role to users of the referenced artifacts.

With the access policies, the following actions are restricted: edit, copy, download, delete, configure, mass delete and download, and simulation of integration flow.

For more information, see [Managing Access Policies, Cloud Foundry Environment](#), as well as the blogs [Access policy for securing design artifacts](#) and [Access Policies: Defining Roles on Artifact Level](#).

## Managing JDBC Material

On the **Operations** page of the Cloud Integration tenant, the **Manage Security** area provides an overview of security-related artifacts. You open the **Manage JDBC Material** area by clicking the tile **JDBC Material**.

In this section, two types of JDBC-related material are stored:

- **JDBC Data Sources:** Allow you to create and manage a cluster of artifact connections to interact with a database (DB). Each data source contains information on database type, as well as database-specific configuration parameters.

For more information, see [Managing JDBC Data Sources](#).

- **JDBC Drivers:** Typically the tenant administrator configures JDBC drivers on your Cloud Integration tenant to enable you to establish connections to a database managed by a third-party vendor. Your database vendor should provide the driver or the access to download it from their official website.

For more information, see [Configure JDBC Drivers](#).

## Managing User Roles

A user account represents an individual person and can have one or more roles associated. These roles define the user's permissions in organizations and spaces. The user role management determines which features users can view, check, access, and what kind of actions users can do. In this case, each role has a set of permissions.

In order to access user roles, in the **Monitor** view, click the **User Roles** tile in the **Manage Security** section. User roles defined by SAP and by the tenant administrator are displayed.

In this app, roles can be added, changed (only the description), or deleted. The actual assignment to users in the system must, however, still be performed by an administrator that has the authorization to modify roles, role collections, and role assignment in the SAP BTP Cockpit. This process is described in the section [Securing User Access to Cloud Integration](#).

## Validating Security Content Through Connectivity Tests

Once security content has been deployed onto a Cloud Integration tenant, it's good practice to perform connectivity tests between the tenant and the business systems you wish it to connect to where possible. Performing these connectivity tests before sending any messages between the middleware and the business systems avoids any possible loss of messages.

On the **Operations** page of the Cloud Integration tenant, the **Manage Security** area provides an overview of security-related artifacts. You open the **Connectivity Tests** area by clicking the tile titled **Connectivity Tests**.

Currently, testing of the following adapter types is supported:

Adapter Type	Description
TLS	Checks HTTPS connectivity and allows you to verify that certificates are valid and trusted.
SSH	Allows you to verify access to an SFTP server and verify that the credentials used are accepted by the server. You also have the option to verify access to specific directories on the server. Supported credential types: <ul style="list-style-type: none"> <li>• User credentials</li> <li>• Public key</li> </ul>
FTP	Allows verification of access to an FTP server (on the Internet or on-premise) and also verifies that the user credentials are accepted (if authentication is required).
SMTP	Allows verification of SMTP connectivity to a mail server. Supported credential types are: <ul style="list-style-type: none"> <li>• Encrypted username/password</li> <li>• Plain username/password</li> <li>• OAuth2 authorization code</li> </ul>

Adapter Type	Description
IMAP	Allows verification of connectivity to an IMAP mail server. Supported credential types are: <ul style="list-style-type: none"> <li>• Encrypted username/password</li> <li>• Plain username/password</li> <li>• OAuth2 authorization code</li> </ul>
POP3	Allows verification of connectivity to a POP3 mail server. Supported credential types are: <ul style="list-style-type: none"> <li>• Encrypted username/password</li> <li>• Plain username/password</li> </ul>
AMQP	Allows verification of connectivity to an AMQP server via either TCP or WebSocket transport protocol. Both Internet-based and on-premise (via Cloud Connector) connectivity is supported. Connectivity is via TLS and the test can also validate Server Certificates.
Kafka	Allows verification of connectivity to a Kafka broker using either SASL or Client Certificates over TLS.
Cloud Connector	Validates that connectivity to a particular Cloud Connector instance is working. If multiple Cloud Connector instances are connected to the same SAP BTP tenant, the Location ID can be used to uniquely identify the correct target. See later section for details on Cloud Connector.

For more information, see [Performing Connectivity Tests](#).

## Using Security Material in Integration Flows

To ensure the security and protection of the message exchange, multiple solutions are possible. It's possible to use methods to encrypt and decrypt the message content, and to digitally sign and verify the message. Another option to secure the communication on transport level is by selecting the HTTPS or SFTP protocol and using specific authentication methods.

The security material deployed and managed in a Cloud Integration tenant is accessed at runtime by integration flows and APIs to implement two forms of security:

- **Transport-Level Security (TLS)**, which involves the encryption of traffic communication between two systems across a network.
- **Message-Level Security (MLS)**, which involves the encryption of actual message payload being transmitted, and they remain encrypted until the payload is explicitly decrypted using a valid key.

Cloud Integration uses the security material discussed in [Managing Security Material in Cloud Integration](#) to implement TLS and MLS.

### Transport-Level Security

Transport-level security provides communication security and protection of data integrity and privacy between two communicating applications. It's used for web browsers and other applications that require data to be securely exchanged on a network.

The following table lists some security elements:

Security Elements	Description

Security Elements	Description
Certificate-based authentication	<p>The authentication of a sender is done based on a client certificate. The system checks if a service key is available that contains the client certificate (provided by the sender). Afterwards, the system checks if the associated service instance has a role specified which has permissions to call the integration flow endpoint.</p> <p>For more information, see the blog <a href="#">How to Set Up Secure HTTP Inbound Connection with Client Certificates</a>.</p>
Securely connecting to on-premise via Cloud Connector	<p>For more information, see the section <a href="#">Connecting to On-Premise Systems Using Cloud Connector</a>.</p>
SSH	<p>Used to read files from the system (sender adapter) or to write files to the system (receiver adapter).</p> <p>See SAP Note <a href="#">2700759</a> for information on how to obtain a Host Key to establish an SSH connection between Cloud Integration and an SFTP server.</p>
OAuth2 Authentication	<p>The following receiver adapter types support the option <i>OAuth2 Client Credentials</i>:</p> <ul style="list-style-type: none"> <li>AMQP/WebSocket, OData V2, OData V4, HTTP</li> </ul> <p>The following receiver adapter types support the option <i>OAuth2 SAML</i>:</p> <ul style="list-style-type: none"> <li><i>Bearer Assertion</i>: OData V2, HTTP, SAP SuccessFactors OData V2</li> </ul> <p>For more information, see the blog <a href="#">SAP Cloud Integration - OAuth2 Client Credentials Support in OData V2 Adapter</a>.</p>
Principal Propagation	<p>The Connectivity and Destination services on SAP BTP allow the use of principal propagation to forward the identity of a Cloud user to the remote system. This assists in the use of Single Sign-On (SSO).</p> <p>For more information, see the help documentation on <a href="#">Principal Propagation</a>, <a href="#">Scenario for Cloud to On-Premise</a>, and <a href="#">Scenario for Cloud to Cloud</a>.</p>

For more information about transport-level security, see [Security Elements \(Transport-Level Security\)](#).

## Message-Level Security

Message-level security allows you to digitally encrypt or decrypt, and sign or verify a message (or both). It can be used when the transport-level security can't be implemented, which happens if a legacy communication endpoint doesn't support this option.

The following table presents some message-level security options:

Message-Level Security Options	Description
PGP Encryption and Decryption	<p>Used to encrypt/decrypt the message content and signing/verification a message.</p> <p>For more information, see the blog on using a <a href="#">HCI Encrypt with PGP</a> on how to use a PGP public key to encrypt data, and the help documentation on <a href="#">using a PGP encryption step</a> and on using a <a href="#">PGP decryption step</a>.</p>
PKCS#7 Encryption	<p>Used to encrypt the content of the message while it's being sent to the cloud.</p> <p>For more information, see the help documentation on <a href="#">encrypting and signing with PKCS#7</a>, and on <a href="#">verifying PKCS#7 signatures</a>.</p>

Message-Level Security Options	Description
XML Signatures	Used for signing and verifying a payload, which ensures the authenticity of a message and guarantees the identity of the signer and the message was not changed after signing. It's possible to digitally sign and validate a message based on the XML signature standard. In this case, the digital signature of a document is stored as an XML element.  For more information, see the help documentation on <a href="#">signing message content with an XML digital signature</a> and on <a href="#">verifying XML digital signatures</a> .
WS Security	Used for signing and verifying a SOAP body and encrypt/decrypt message content (and the other way round to verify a message and to decrypt the message content).  For more information, see the help documentation on <a href="#">Sender SOAP Adapter</a> and <a href="#">Receiver SOAP Adapter</a> .

For more information about message-level security options, see [Message-Level Security](#).

## Managing Destinations in SAP BTP Cockpit

Destinations are key building blocks in SAP BTP and are used to define connections for outbound communication from your application to remote systems. These remote systems can be on-premise or in the cloud. Typically, Cloud Integration can directly access the virtual systems exposed by the Cloud Connector, except for the RFC channel. In Cloud Integration, for the RFC channel it's necessary to configure a destination in SAP BTP Cockpit.

Similar to destinations in SAP BTP, in SAP Process Orchestration, for some communication channels you can maintain RFC/HTTP destinations to reuse them as much as necessary, for example, when you want to connect SAP Process Orchestration to your ECC receiver.

A destination has a name, a URL, authentication details, and other configuration details specific to the requirement and type of destination.

For full help documentation regarding the use of the Destinations Editor (and other methods to maintain destinations), as well as the setup of the different destination types, see [SAP BTP Connectivity - Initial Setup](#).

## Error Handling and Logging Strategy

Before migrating from SAP Process Orchestration to Cloud Integration, important aspects of the error handling and logging strategy that must be evaluated.

### [Error Handling](#)

Discover the different types of error messages and how they're handled in SAP Process Orchestration and Cloud Integration.

### [Logging](#)

Learn about different ways to do logging and tracing in SAP Process Orchestration and Cloud Integration.

### [Simulation](#)

While developing the interface, you need to test your message mapping, or in Cloud Integration, test part of your integration flow. To do so, both SAP Process Orchestration and Cloud Integration provide features to support this development activity.

### [Monitoring](#)

Find out which monitors you can utilize when working with SAP Process Orchestration and Cloud Integration, and discover offerings for application lifecycle management.

### [Operation](#)

Learn more about the ways both SAP Process Orchestration and Cloud Integration operate, for example when deploying interfaces, performing connectivity tests, logging files, or working with APIs to automate operations.

## Error Handling

Discover the different types of error messages and how they're handled in SAP Process Orchestration and Cloud Integration.

### Error Messages

There are two types of errors that you can encounter when handling errors in SAP Process Orchestration and Cloud Integration:

- **Technical Errors:** Errors triggered by the system because of technical issues. These can be a wrong password, the Web Service is down, scheduling error, mapping error, etc.
- **Business Errors:** Errors triggered by the target application itself. These errors can be, for example, an error while creating a business partner, or a response error while creating a sales order because the material number doesn't exist in the target system.

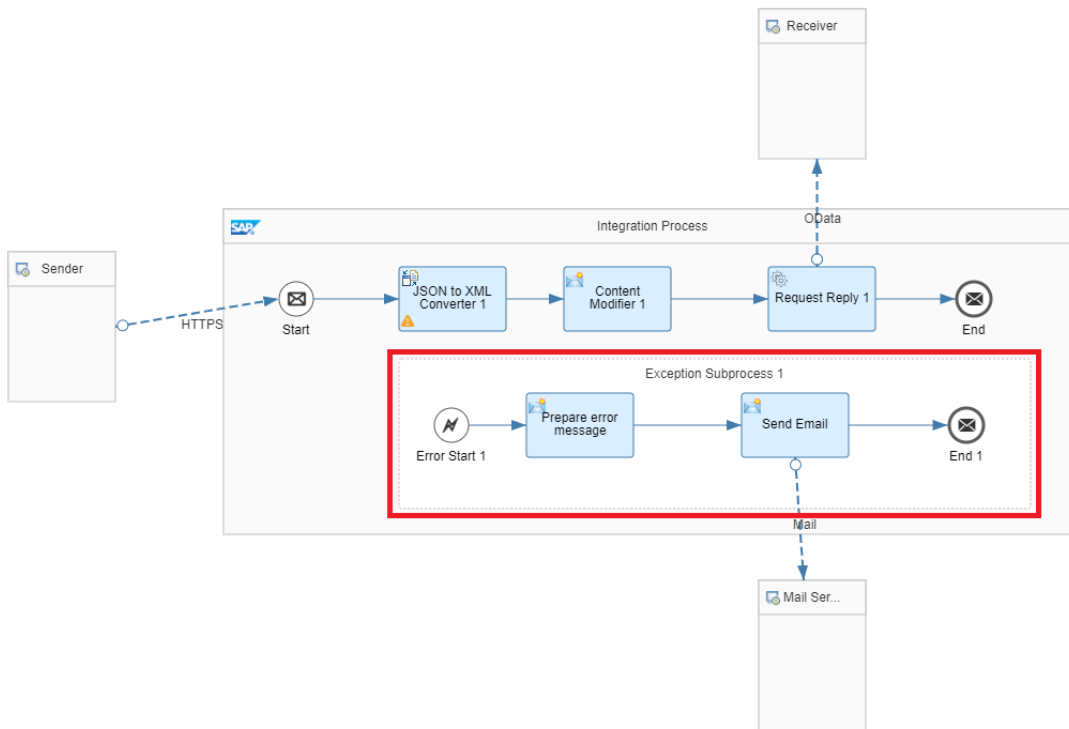
In the middleware layer, you normally provide support for technical errors, while for the business errors you only need to assure that the error message is propagated for the source/caller system. Alternatively, you can also use [SAP Application Interface Framework](#) to process the error in the business layer.

### Error Handling

In SAP Process Orchestration, there's a built-in mechanism to reprocess asynchronous error messages, or, depending on the business requirements, enable alerts via e-mail and customize a response error message to the source system (in case of synchronous messages). For point-to-point interfaces, you have the possibility to configure an error message while designing the interface in Enterprise Services Repository. Some communication channels also count with specific parameters to handle errors (e.g., REST, SFTP). When using SAP Business Process Management (SAP BPM), you can implement other types of error handling, such as generating a log file to an SFTP server and/or sending an asynchronous message to a third system. The following are the general recommendations while handling errors in SAP BPM:

- Make the error as specific as possible and provide all the necessary information to facilitate the process for the future administrator who has to handle it.
- Use boundary events for automated/human activities.
- Use multiple boundary events to handle different types of exceptions.
- Create reusable exception processes, if possible, for different processes that share the same error handling.

In Cloud Integration, if you don't use an external system or service to generate alerts via e-mail (SAP Solution Manager, SAP Cloud ALM, SAP Alert Notification service for SAP BTP) or use an asynchronous adapter with native support for reprocessing (AS2, AMQP, JMS), you can achieve both capabilities implementing exception subprocesses directly into the integration flow:



For more information about how to use exception subprocess, see [Define Exception Subprocess](#) in the documentation for Cloud Integration.

For documentation about best practices while handling errors in Cloud Integration, see [Handle Errors Gracefully](#).

If you want to access the delivered package for Cloud Integration with some examples of integration flows with error handling, go to the [SAP Business Accelerator Hub](#) or access it directly from the Cloud Integration tenant.

## Logging

Learn about different ways to do logging and tracing in SAP Process Orchestration and Cloud Integration.

## Message Staging and Logging

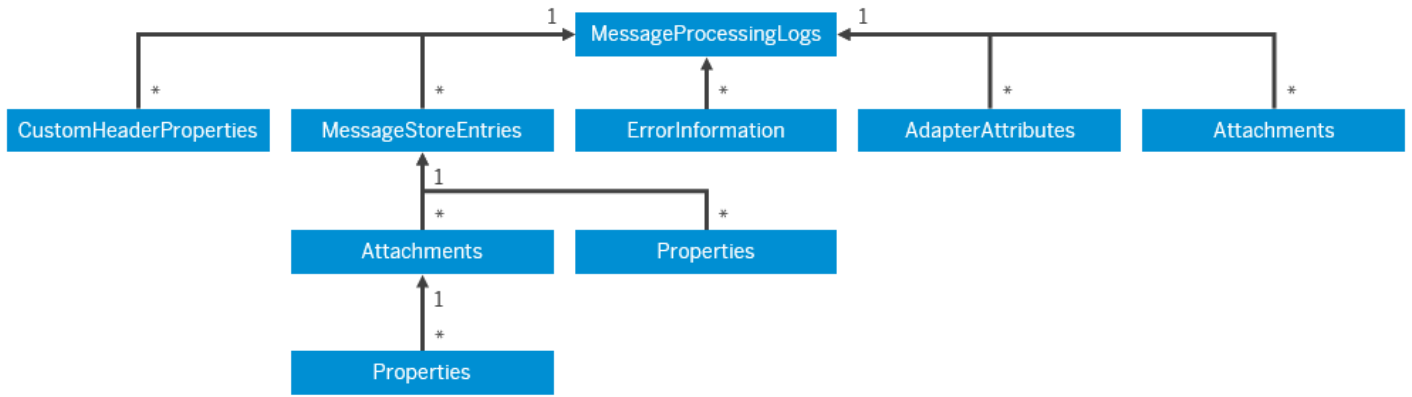
While Cloud Integration is hosted on hyperscalers with a restricted database per tenant and not administered by the customer, an SAP Process Orchestration on-premise installation varies from customer to customer and is usually managed by the customer or a partner. This means that in SAP Process Orchestration, you must maintain parameters for housekeeping, archiving, etc. For more information on how to change the message logging periods in SAP Process Orchestration, see the following help documentation:

- [Configuring Message Logging](#)
- [Persist duration time at the Adapter Engine for synchronous and asynchronous messages](#)
- [Configuring Message Retention](#)

In Cloud Integration, there are three different types of logs:

- **Audit Log:** Stores information on system changes. The retention time in database is 30 days.
- **System Log:** Stores information on HTTP requests or default trace files. The retention time in database is 7 days.
- **Message Processing Log (MPL):** Stores information on every message processed on a tenant and all their relative details such as attachments, processing steps, etc. This is the data model for the Message Processing Log entity:

This is custom documentation. For more information, please visit the [SAP Help Portal](#)



The retention time in the database for the messages is 30 days. The retention time and method may change for each log type in the future. For more information, see [Specific Data Assets](#).

[Data Store](#) is another option for developers in Cloud Integration who need to keep messages or attachments for a longer period. Data Stores are collections that can be created to store or read temporary messages. Possible use cases are:

- Collect error messages in a data store. A scheduled integration flow can collect all those error messages and send a summary of those errors via email to the IT Team or automatically create support tickets.
- Receive different asynchronous messages (with different data types) and wait before composing a full enriched target message. This is common in Commerce Cloud integrations. In SAP Process Orchestration, you could only do something similar using SAP BPM.

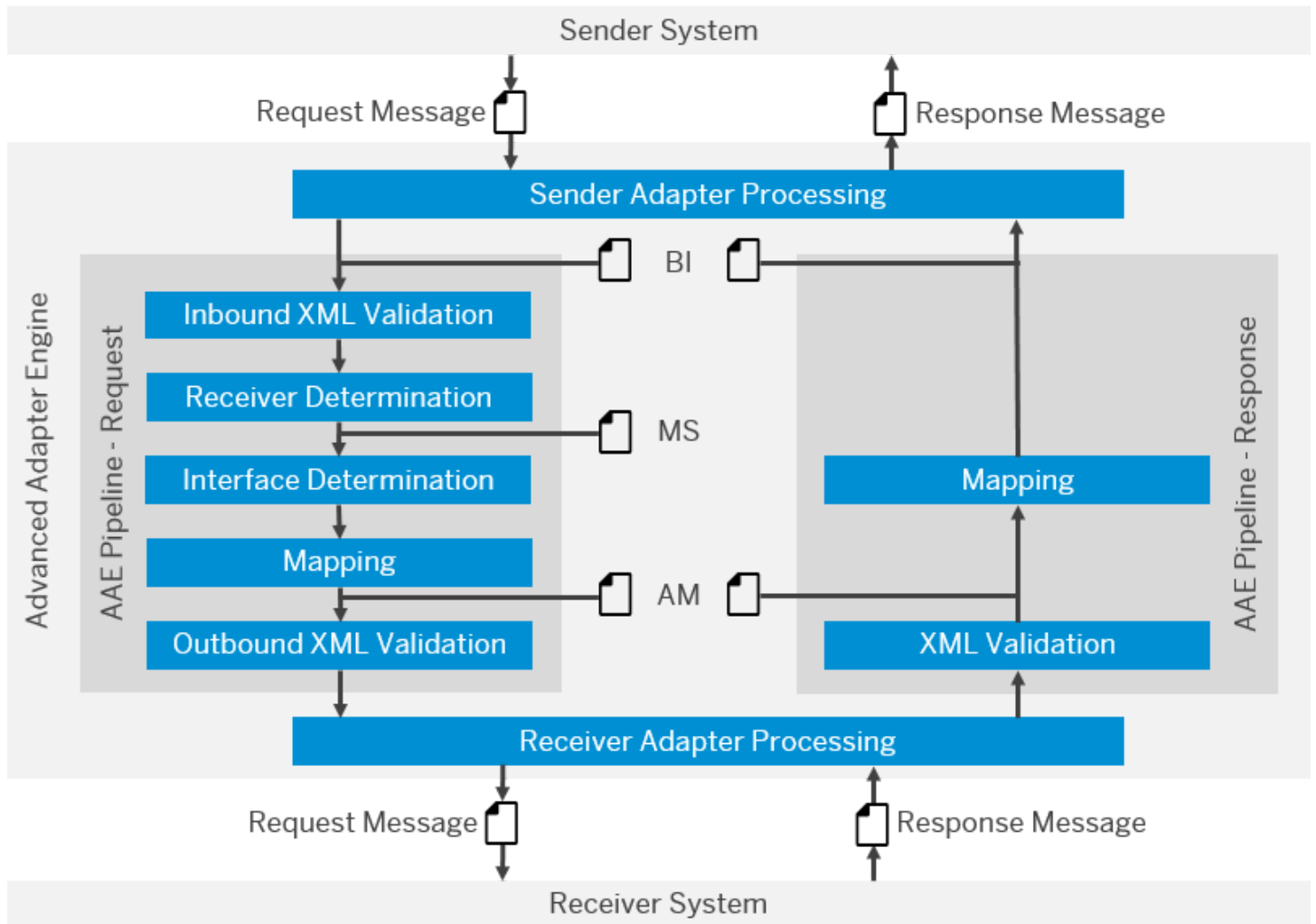
Although the data stores are a powerful resource, they must not be used as a database to keep data indefinitely. For more information about the guidelines while using data store, see [Data Storages](#) in the Cloud Integration documentation.

If long-term storage of message logs or data is required (for example, for legal or compliance reasons), it's recommended to use a separate storage mechanism such as a separate SAP HANA database instance and publish APIs that can be invoked from Cloud Integration to post any data. However, keep in mind that the introduction of these capabilities increases the processing time for integration flows.

## Trace

When doing troubleshooting in SAP Process Orchestration, you may need to access the message payload in all its phases, for example, right after SAP Process Orchestration receives the message from the sender, before and after the internal operation mapping, and more. The following overview provides each message log step:

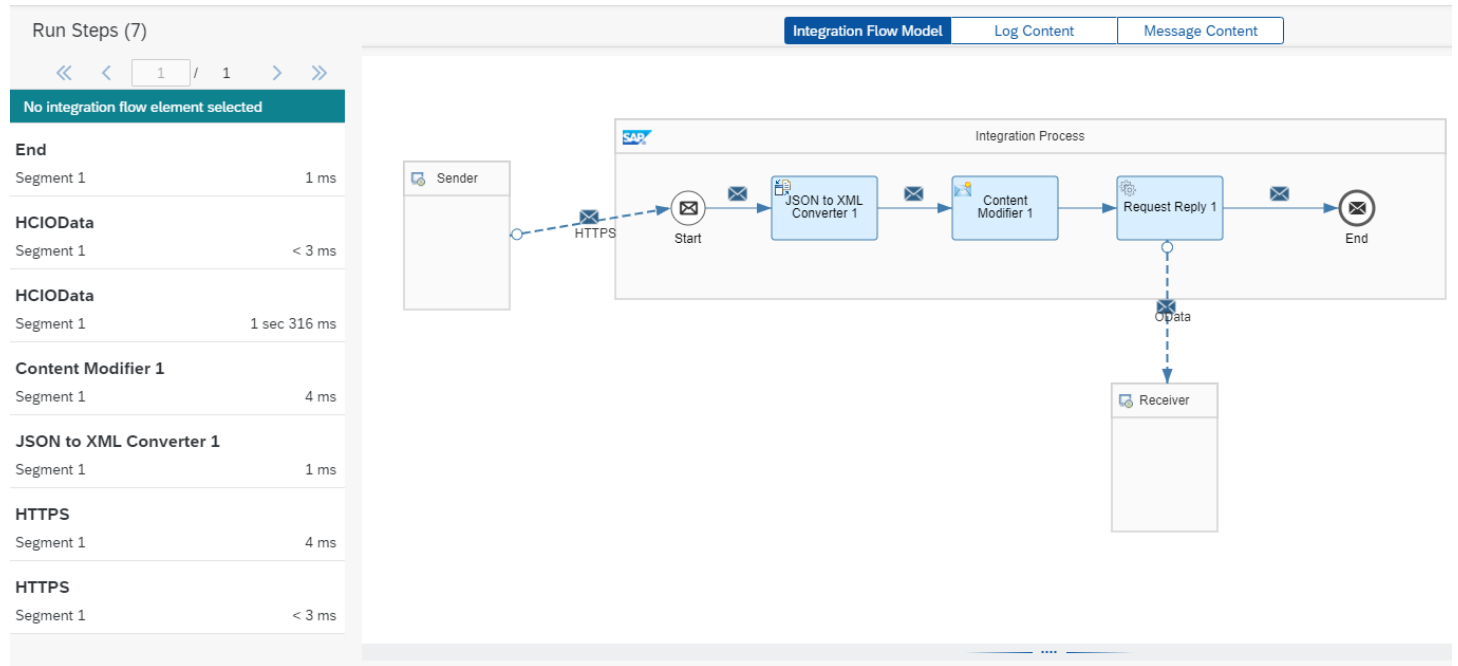




You can enable the log of each message trace level as described in the section [Message Staging and Logging](#). Alternatively, you can change it directly in your Integrated Configuration object by going to **Edit Integrated Configuration > Advanced Settings**. In section **Staging**, select **Use scenario-specific configuration**, and for **Message Preparation**, select **Store**. In section **Logging**, select **Use scenario-specific configuration**, and for **Receiver Determination** and **Mapping**, select **Log**.

After activating it, you can access the message details page in message monitor and display all the relevant steps.

In Cloud Integration, if you need to find out how a message is processed and transformed at runtime, an option is to run the related integration flow with the logging level set at Trace. Unlike SAP Process Orchestration, in Cloud Integration you don't need to redeploy your integration flow to activate the trace feature. With this level of logging turned on, you can then use the log monitor to examine the Message Payload, as well as the Properties and Header values that exist in each step in the integration flow execution:



For more information, see [Tracing the Execution of an Integration Flow](#).

If the traces contain sensitive data, they can be protected with access policies. For more information, see [Managing Access Policies, Cloud Foundry Environment](#).

## Simulation

While developing the interface, you need to test your message mapping, or in Cloud Integration, test part of your integration flow. To do so, both SAP Process Orchestration and Cloud Integration provide features to support this development activity.

In SAP Process Orchestration, you can test your message directly in the monitoring page as described in SAP Note [2367889](#). In Enterprise Services Repository, you can also test your message in the message mapping and operation-mapping level. For more information, see [Test Environment for Operation Mappings](#).

A more robust tool that allows you to set up automatic tests for existing SAP Process Integration scenarios and to reduce the business downtime is the SAP Process Integration Test Tool. For more information, see [SAP Process Integration Test Tool](#).

In Cloud Integration, there are two different options to test an integration scenario: **simulating a mapping** or **simulating an integration flow**.

### Simulate Mapping

Similar to the SAP Process Orchestration mapping test, within the **Design Environment** of an integration flow open a Message Mapping and click **Simulate**. In the next step, it's required to upload a source file as input for the simulation. Afterwards, the simulation can be executed with the **Test** button, and the result is displayed.

Test Input:  Browse

Structure	Value
BusinessPartner	
ID	0012000020
Name1	TES
Name2	TEST2
BirthDate	1989-01-01
CreditRating	0000
Telephone1	+43223620010
Telephone2	+4315968310
CellPhone1	+4367681032686
CellPhone2	+4366481022111
FaxNumber	+432236200590
EMail	test@test.com
> Account	
> Adress	
> Info	
> BonusPoints	
> BonusPoints	

Test Output

Structure	Value
BusinessPartner	
ID	0012000020
Name1	TES
Name2	TEST2
BirthDate	1989-01-01
CreditRating	0000
> Account	
> Adress	
> Info	
> BonusPoints	
> BonusPoints	
NextExpirationDate	2021-01-03T00:00:00.0Z
NextExpirationDatePoints	121
PointValue	2.42
TotalPoints	121
VKNumber	18474200

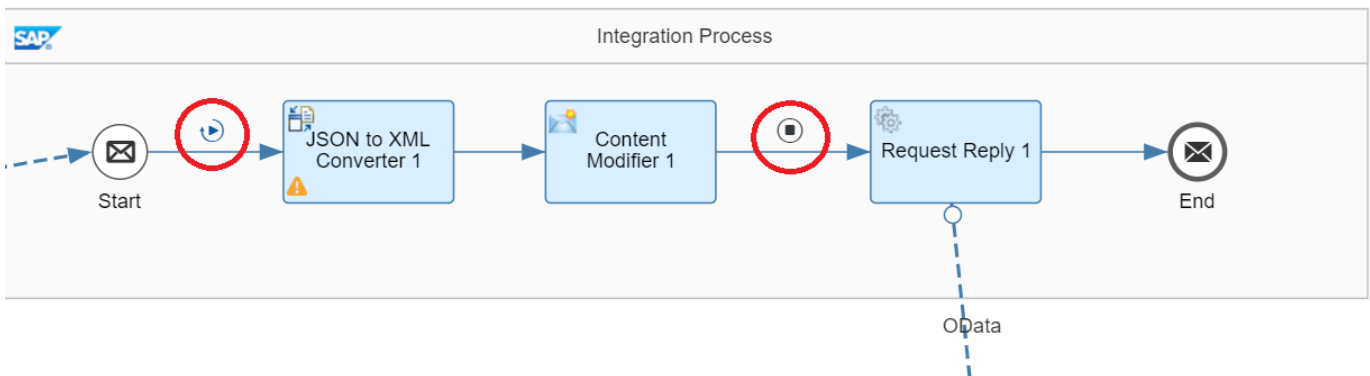
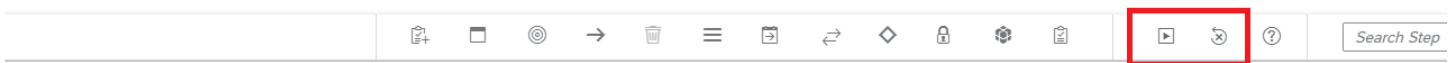
Test Download Output Close

In Cloud Integration, you can use the **Display Queue** functionality, previously available in SAP Process Orchestration, to examine the context queue of any target element. The result mapping can also be downloaded if needed.

For more information, see the blog [Message Mapping Simulation in SAP Cloud Platform Integration](#).

### Simulate Integration Flow

When developing your integration flow, you can use a simulation feature that allows you to simulate parts of your integration process or all of it. At various steps in your integration flow, you can set a **Simulation Start** point and inject the input data. If you don't want to execute the integration flow until the end, you can also specify a **Simulation End** point:



For more information, see [Simulation of an Integration Flow](#) and the blog [Integration Flow Simulation in Cloud Integration](#).

# Monitoring

Find out which monitors you can utilize when working with SAP Process Orchestration and Cloud Integration, and discover offerings for application lifecycle management.

## SAP Process Orchestration Monitoring

In SAP Process Orchestration, the following monitors support you in your daily operations and troubleshooting:

### Message Monitor

Using the message monitor in SAP NetWeaver Administrator, you can monitor the processing of messages on the Advanced Adapter Engine. You can use message monitoring in the following cases:

- To create an overview of message processing. Message overview also provides two different modes for searching messages.
- To track the status of messages.
- To find errors that have occurred and to establish what caused them.

You can also search for messages that were persisted in the database, or for messages that have already been archived.

### Communication Channel Monitor

Use the communication channel monitor to obtain information about communication channels that are set up for the Advanced Adapter Engine. You can also perform some administrative activities.

If the Advanced Adapter Engine runs on a server cluster, the communication channels comprise multiple instances for the various cluster nodes.

### IDoc Adapter Monitor

Use the IDoc message monitor to search for IDoc messages processed in the adapter. Based on the various parameters of the IDoc message you provide as search criteria, the monitor retrieves the messages and displays information about them.

For more information, see [Displaying IDoc Messages](#).

It's also possible to use the IDoc metadata monitor to display IDoc metadata information and clear the metadata for the selected IDoc type from the cache.

### Adapter Engine Status

Use the adapter engine status to get information about the adapter engine, queues, message traffic (e.g. number of asynchronous or synchronous messages), database locks, overview of messages (being) processed per sender, and receiver components.

### Channel-Independent Logs

Use the channel-independent logs to display execution steps of adapters that can't be assigned to a particular communication channel.

### Cache Monitor

Use the cache monitor to display objects that are currently in the runtime cache of the Advanced Adapter Engine and the Mapping Runtime. You can enter selection criteria to search for current objects in the runtime cache and review the details of a selected object.

### Performance Monitor

Use the performance monitor to inspect the processed data by intervals of time and their processing time for individual adapters.


### CPA Cache History

Use the CPA cache history to display an overview of cache refresh actions and their statuses.

### Background Job Monitor

Displays an overview of background jobs (adapter engine, adapter framework scheduler jobs, etc.) and offers the ability to manage them.

### More Information



- You can also [centralize the Process Integration monitoring with SAP Solution Manager](#) .
- SAP Process Orchestration has a standard functionality called [Component-Based Message Alerting](#) which allows you to create alert rules to your interfaces.
- For more information, see [Monitoring the Advanced Adapter Engine](#).


## Cloud Integration Monitoring



In Cloud Integration, a message monitor allows you to display the processed messages and their statuses, logs, and traces. Compared to SAP Process Orchestration, there are less monitors because you don't have such components like Adapter Engine, or you don't perform as many administrative tasks since some of these are part of SAP's platform support service. You also don't have monitors such as Cache Monitor, Metadata Monitor, Adapter Engine Monitor, and Communication Channel Monitor. For more details about the message monitor and its capabilities, see [Guidelines and Best Practices for Message Monitoring](#).

Cloud Integration isn't delivered with pre-built out-of-the-box alert capabilities, but you can use a predelivered package that uses the SAP Alert Notification service for SAP BTP to send alerts of failure of integration flows. You can also create your own integration using the available APIs to read the Cloud Integration tenant content.

For more information, see the following blogs:

- [Creating Custom Alert Notifications in SAP Integration Suite](#) .
- [Receive Notifications for Failed SAP Cloud Integration Flows via Any Channel with Alert Notification](#) .

For a comprehensive walkthrough of the capabilities of the SAP Alert Notification service, review this [blog series about SAP Alert Notification](#) .

If the standard capabilities of the SAP Alert Notification service aren't sufficient to meet customer requirements, or there's no external system or service like SAP Solution Manager to generate alert notifications, integration flows that invoke calls to the standard APIs published by the SAP Alert Notification service can be built within Cloud Integration. Full documentation of these APIs can be found on the [SAP Business Accelerator Hub](#) . Another option is to combine the exception subprocesses while developing your integration flow with some procedure to collect the error/success message and send it via email. For details, see the blog [How to send a mail from SAP Cloud Platform Integration](#) .

# Application Lifecycle Management (ALM)

## SAP Cloud ALM

SAP Cloud ALM is an offering for application lifecycle management. It's intended for customers who use cloud solutions provided by SAP, and who don't want to use their own ALM on-premise platform for managing those solutions. SAP Cloud ALM runs on SAP Business Technology Platform and is optimized for SAP HANA. It provides extensive implementation and operation capabilities for cloud solutions with the following key features:

- Implement cloud-centric solutions with a preconfigured, content-driven guided workspace based on SAP Activate methodology and SAP Best Practices.
- Run fit-to-standard workshops and manage all implementation, testing, and deployment activities.
- Accelerate team member onboarding, define business process scope according to project milestones, manage requirements, and track project progress.
- Ensure smooth business operations without disruptions with proactive end-to-end monitoring and alerting.
- Increase business-process execution quality and performance by finding and analyzing issues on business process, integration, user, and application level.
- Understand solution health and efficiency with advanced analytics and intelligence.

For more information, access the openSAP courses [SAP Cloud ALM in a Nutshell](#) and [Accelerate Cloud Implementations with SAP Cloud ALM](#).

You can also access the SAP Support pages [SAP Cloud ALM for Implementation](#) and [SAP Cloud ALM for Operations](#).

## SAP Solution Manager

SAP Solution Manager is an application lifecycle management platform used to implement, maintain, and integrate SAP systems, troubleshoot issues, and keep operations running securely, cleanly, and smoothly. This tool was originally designed to support on-premise systems only (including SAP Process Orchestration), but the scope of capability has been increased to include cloud-based services, including Cloud Integration.

For detailed information on the capabilities and setup procedures to connect SAP Solution Manager to Cloud Integration, see the [SAP Support page for Cloud Integration](#).

## SAP Cloud ALM vs SAP Solution Manager

### ! Restriction

Some features mentioned on this site are planned for the future and are subject to change. For more information on new features and future releases, access the [Road Map Explorer](#) and the [What's New section](#) of SAP Integration Suite.

Though SAP Solution Manager can currently be used for cloud products and SAP Cloud ALM can also be used for hybrid scenarios as part of future roadmap, it's important to understand the positioning of these two different products. SAP Solution Manager is recommended for on-premise products and SAP Cloud ALM is engineered keeping the cloud applications in mind. Both of these products have different use cases, and both serve a different market.

For more information, see the blog [SAP Cloud ALM vs SAP Solution Manager](#).

## Operation

Learn more about the ways both SAP Process Orchestration and Cloud Integration operate, for example when deploying interfaces, performing connectivity tests, logging files, or working with APIs to automate operations.

## Deploying Interface

In SAP Process Orchestration, once you activate your [Integrated Configuration](#) object, your interface is up and running.

In Cloud Integration, once you finish the design of your integration flow, you must deploy the artifact. After deployment, you can perform some tasks such as restarting the artifact, undeploying the artifact, and downloading the deployed artifact to your PC. The last option is useful when you need to recover the deployed version of your integration flow.

## Connectivity Tests

A feature introduced in Cloud Integration is the connectivity tests, which allow you to perform some simple test connections in different protocols to assure the connectivity with the target system. In TLS protocol, for example, you can download the full certificate chain that should be imported in Cloud Integration keystore most of the times. For more information, see [Performing Connectivity Tests](#).

In SAP Process Orchestration, after activating your communication channel in Integration Directory, for some of them (see SAP Note [1890633](#)) you can also navigate to the Communication Channel Monitor and try the ping channel, which provides the result of a basic connectivity test. Another option, especially for TLS protocol, is to use [XPI Inspector tool](#) to have access to the certificate chain that should be imported to the keystore. XPI Inspector offers a full trace and dump during a connectivity test, which helps to analyze and solve connectivity issues. The only exception concerns the RFC adapter, for which you need to maintain an HTTP destination in the SAP BTP tenant as described in [Creating an RFC Destination](#).

## Log Files

Both SAP Process Orchestration and Cloud Integration provide logging mechanisms to support the monitoring and troubleshooting of interfaces that are executed on the middleware systems. As Cloud Integration is a cloud-based service, the persistence of logs and any attachments to logs differs from SAP Process Orchestration in that the duration of persistence is much shorter and the deletion of logs lies outside of the control of the customer.

SAP Process Orchestration stores activities of system, integration, and database events. In the Log Viewer, you can see the technical detail system status. Based on information in system logs, you can check, monitor, and investigate any issues and possible root causes.

The following system logs are stored with multiseverity:

- Fatal error messages
- Error messages
- Warning messages
- Information messages
- Path messages
- Debug messages

You can access the Log Viewer by going to ► [SAP NetWeaver Administrator](#) ► [Troubleshooting](#) ► [Logs and Traces](#) ► [Log Viewer](#) ►

For Cloud Integration, you must use the SAP Audit Log service to monitor the security-relevant chronological records such as events, activity logs, and application logs. For more information, see [Audit Logging in the Cloud Foundry Environment](#). You can

also use the Audit Log Retrieval API to extract possessed by the tenant. For more information, access the [SAP Business Accelerator Hub](#).

## Automate Operations Using APIs

Most of these administrative tasks described for Cloud Integration and SAP Process Orchestration, such as read integration messages, security artifacts, log files, and communication channels, are available in APIs to be used externally. This way, you can automate some procedures using custom script or even build your own toolset. In Cloud Integration, the most common usage is to build your own alert notification mechanism, but you can even create a procedure to automatically maintain some objects in the tenant keystore (e.g. key pair, certificate chain).

To access the most updated list of APIs for Cloud Integration, visit the [SAP Business Accelerator Hub](#). The capabilities include:

- Manage and query integration artifacts at design time and runtime.
- Access and query HTTP and trace log files on a tenant.
- Get an overview of the messages processed on a tenant and get the details for individual messages.
- Access message store entries, JMS resources, and data store entries.
- Access the [Partner Directory](#) to maintain and manage data.
- Access and manage security content to configure secure connections with remote systems.

For SAP Process Orchestration, there's official documentation for some available APIs, for example for [Integration Directory](#) and [SAPControl Web Service](#), and also some blogs that explore other types of automation, for example [An overview on SAP PI/PO APIs](#).

## Interface Governance

Discover important aspects regarding the tools and services available that support the management, auditing, and governance surrounding the transport of integration artifacts within a Cloud Integration landscape.

### [Transport Management Options](#)

Learn more about the different transport management options that are available to you, including Enhanced Change and Transport System (CTS+), SAP Cloud Transport Management service, and SAP Solution Manager.

### [Setup Guidelines for Transport Solutions](#)

Discover information and guidance on different transport solutions.

### [Services Registry to API Management](#)

Find out more about the API Management capability.

### [Interface Documentation](#)

SAP Solution Manager 7.2 makes it possible to manage the interface documentation and document it sustainably.

## Transport Management Options

Learn more about the different transport management options that are available to you, including Enhanced Change and Transport System (CTS+), SAP Cloud Transport Management service, and SAP Solution Manager.

## Multi-tier Environments

This is custom documentation. For more information, please visit the [SAP Help Portal](#)



For SAP Process Integration and SAP Process Orchestration, it's recommended to set up separate landscapes for development, test, and productive system. In any organization using Cloud Integration as their integration middleware service, there's typically what is commonly referred to as a multi-tier environment, like in PI/PO. In both cases, this concept supports the capability for an organization to have a 'live' production middleware service in place that handles the actual movement of business messages between production systems, while still being able to develop, perform quality assurance, and fix and test new or revised integration content before deploying to production where the real business processes are performed.

The minimum number of tiers possible is a 2-tier system comprised of a development (Dev/Test) tenant and a production tenant.

Many larger organizations, however, opt for a much larger multi-tier environment such as:

- 3-tier: **Development (Dev) → Test (or Quality Assurance System) → Production (Prod)**
- 4-tier: **Development (Dev) → Test → Preproduction (Pre-Prod) → Production (Prod)**

Preproduction is typically closer in size to production (in terms of resources such as memory and CPU) and traditionally would be used for performance and volume testing, analysis of any bugs detected in production. It's often deemed as part of the production landscape and used for live support.

In some situations, organizations may opt for even more development and test tenants to support either multiple projects on the go, depend on the dimension of the project, segregate development by geographical region, by business function (e.g. separate HR, Finance). One of the benefits of multi-tier landscape is the possibility to perform tests in parallel with the developments or numerous other reasons.

For more information, see the blog [Multitenancy Architecture on SAP Cloud Platform, Cloud Foundry environment](#) .

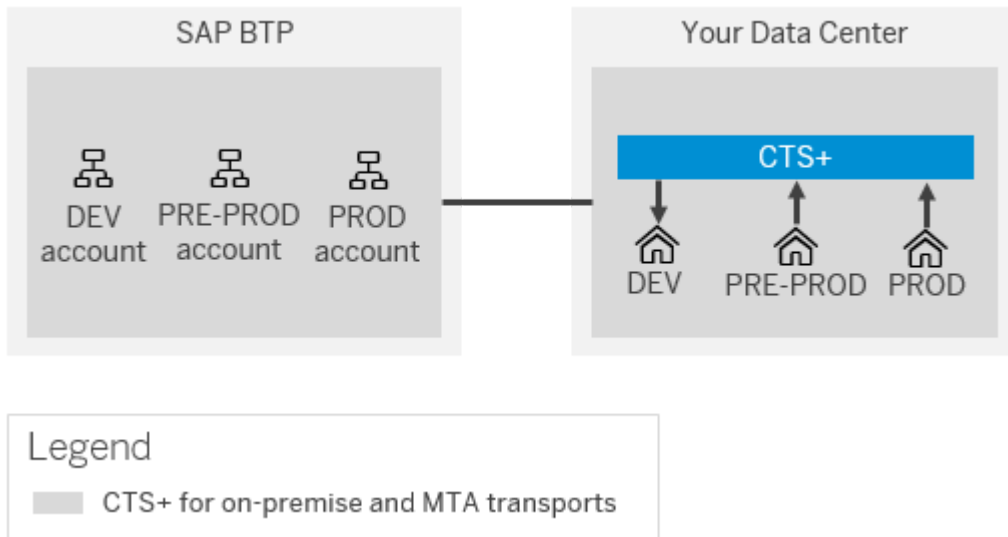
## Managing Environments

Governing, and managing the proper testing and audited transport of integration objects between the different tier systems can become complex. It remains equally critical that only properly tested and signed-off integration objects reach the production tier in a synchronized manner to ensure business continuity and prevent any breakdown in business processes. To aid in this governance process, SAP provides tools, products, and services to manage this governance process. The current toolset available is summarized in the next sections.

## Enhanced Change and Transport System (CTS+)

Enhanced Change and Transport System (CTS+) was originally designed to handle ABAP-based objects in on-premise systems as well as non-ABAP content such as the java-based objects from SAP Process Integration and SAP Process Orchestration. CTS+ is a tool that supports all development workbenches and applications.

It allows to attach the objects developed/created to a CTS+ transport request. When the development objects are ready to be tested in quality environment, for example, the request should be released. After this step, it's not possible to add or change the objects already released and for other changes, a new change request must be created. CTS+ provides a unified transport tool and simplifies the execution of imports in a new environment.



Since all transports are logged in CTS+, it's possible to have an overview of the status of the change requests, like the errors that could happen during the transport, which allows you to quickly solve the issues. You can also check if a change request was already executed and the ones that are planned.

For more information, see [Transporting Non-ABAP Objects in Change and Transport System](#).

CTS+ has been extended to handle transports in SAP BTP:

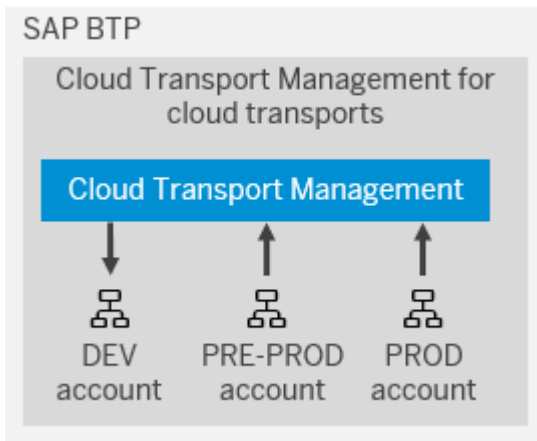
- CTS+ can transport in Cloud Foundry environment.
- The transports are based on Multitarget Applications (MTA) and cover:
  - Development artifacts
  - SAP Integration Suite
- CTS+ can only be used for organizations that intend to adopt a hybrid approach in which some systems remain either on-premise or choose to opt for the Private Cloud Edition (PCE) for SAP Solutions such as SAP S/4HANA and SAP Solution Manager.
- Note that the ABAP server acting as CTS+ control system (in many cases an SAP Solution Manager) resides in your on-premise data center even for the SAP BTP transports.

For more information, see [Transporting Multitarget Applications in Cloud Foundry using CTS+](#), the [support page for CTS+](#), and the blog [Setting up a CTS+ enabled transport landscape in SAP Cloud Platform](#).

## SAP Cloud Transport Management Service

The SAP Cloud Transport Management service is a new service that has been built directly on the SAP BTP to support the transport of not only integration artifacts and application-specific content for Cloud Integration, but also other cloud-based development objects built using the various technologies (e.g., Java, Node.JS, SAPUI5) available on SAP BTP.

SAP Cloud Transport Management allows you to manage software deliverables between accounts of different environments (like Cloud Foundry environment) by transporting them across runtimes. It's a multi-tenant application and provides a tenant separation.



Cloud Transport Management has the following capabilities:

- It's the officially recommended solution for transporting in the cloud
- It can transport development artifacts (via MTA) and application-specific content
- It will cover further content types in the future (see the [roadmap for SAP Cloud Transport Management](#))
- It covers the SAP BTP, Cloud Foundry environment and is planned to be extended to further multi-environments and to SAP SaaS solutions (such as SAP Cloud Price Quote)
- Cloud Transport Management and CTS+ can be used in parallel: Cloud Transport Management for cloud transports, and CTS+ for ABAP and non-ABAP on-premise transports.
- Cloud Transport Management is not planned to cover on-premise transports.

For more information, see the [SAP Cloud Transport Management documentation](#).

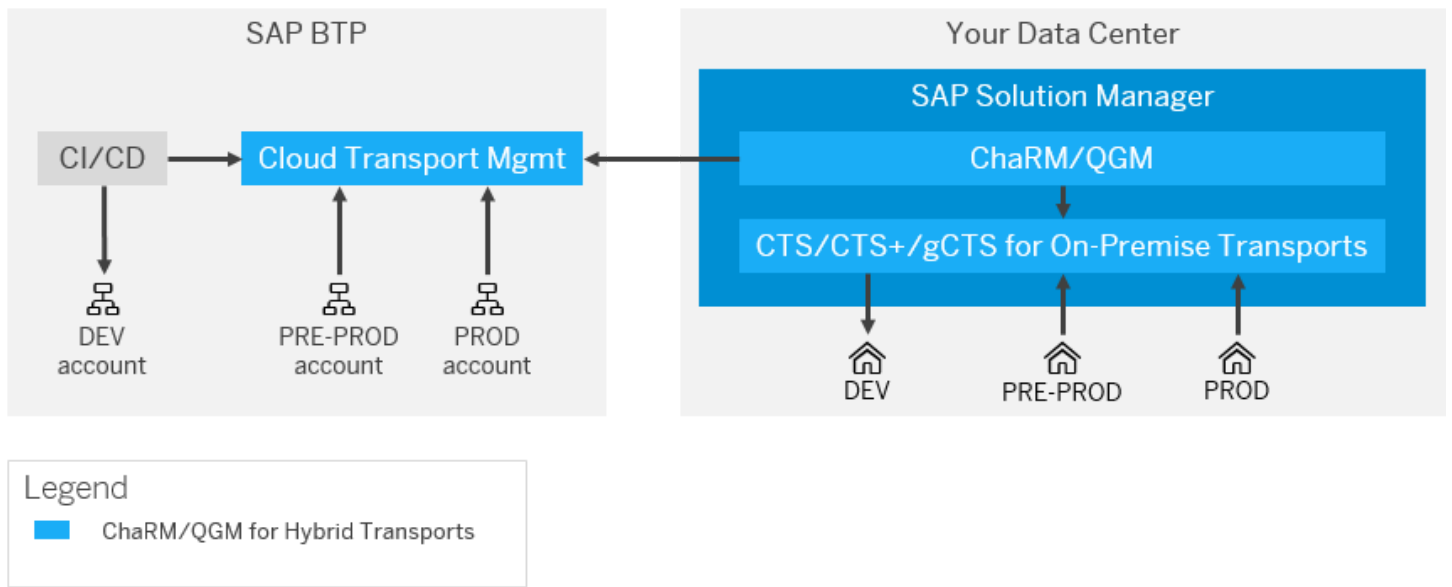
## SAP Solution Manager

SAP Solution Manager is a centralized application management and administration solution used to implement, operate, and monitor SAP solutions. It's useful to manage the lifecycle of an SAP solution and provide integrated content, tools, methodologies, and access to SAP systems. It ensures reliability and stability of solutions.

Some benefits of using SAP Solution Manager are the efficient project administration, control of cross-component implementations, minimizing the risk and increasing the reliability of solutions, and speeds up test preparation and execution. SAP Solution Manager can store the testing materials and test results to support cross-component tests.

It's also possible to manage documentation of interfaces, as describes in [Interface Documentation](#).

For more information, see the [support page for SAP Solution Manager](#).



In a hybrid environment, any changes or new development can involve development objects that span both on-premise and cloud-based services or components. For example, there may be changes required to ABAP objects in an on-premise system, while integration artifacts to support the hybrid application may be required as well. It can be critical to ensure that all objects are verified and moved in a synchronized manner to ensure the hybrid application doesn't fail. By using tools such as Change Request Management (ChaRM) in SAP Solution Manager, this synchronized movement of objects can be managed.

- Both Cloud Transport Management and CTS+ can be integrated with the Change Management tools on SAP Solution Manager:
  - Change Request Management (ChaRM)
  - Quality Gate Management (QGM)
- This allows synchronized transports of ABAP and non-ABAP on-premise assets via CTS+, and cloud artifacts via Cloud Transport Management in hybrid projects.
- This setup is important for customers with strict audit requirements such as banking or pharmaceuticals.
- The prerequisite for integrating Cloud Transport Management with SAP Solution Manager is release 7.2 with SPS10 (available as of December 2019) or higher.

### **i Note**

gCTS is the Git-based CTS solution that's currently not relevant to integration artifacts and hence isn't covered further here.

For more information, visit the blog [Interplay of SAP Cloud Platform Transport Management, CTS+ and ChaRM in hybrid landscapes](#).

## Setup Guidelines for Transport Solutions

Discover information and guidance on different transport solutions.

For detailed guidance on the options available to transport integration artifacts, and how to set up the transport service, it's recommended to read the sections on [Content Transport](#) in the Cloud Integration documentation.

### Decision Help

We provide a [decision help](#) that looks at cloud-based transport options and describes the advantages and disadvantages of each:

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

Transport Option	Recommendations
Manual export/import	<ul style="list-style-type: none"> <li>• Use only in exceptional cases, for example for urgent fixes in case the transport system is down.</li> <li>• This option is easy-to-use (no configuration required).</li> <li>• No control and logging of the transport events and the involved users.</li> <li>• Risk of losing control of who is transporting what and when. Using this option, you don't have any advantage of the tool support provided by a transport system, like, for example, logging. For this reason, manual exports and imports should be avoided in an integration project or while operating Cloud Integration.</li> </ul>
CTS+	<ul style="list-style-type: none"> <li>• CTS+ is a recommended option.</li> <li>• This option provides a fully automated transport mechanism (allows the assignment of roles, definition of transport routes, approvals, and comes with additional useful features like logging).</li> <li>• If an on-premise CTS+ is already installed and used for other systems or objects, it's an option to use CTS+ also for integration content transport.</li> </ul> <p><b>i Note</b></p> <p>CTS+ has been available for many years to transport on-premise non-ABAP content. This concept has been extended some time ago to support SAP BTP. However, only cloud content that can be packaged in MTAR files can be transported. For integration content transport, this constraint is fulfilled. Therefore, if you use MTAR-based content only in SAP BTP, CTS+ is a valid choice.</p>
SAP Cloud Transport Management service	<ul style="list-style-type: none"> <li>• SAP Cloud Transport Management is a recommended option.</li> <li>• It provides a fully automated transport mechanism (allows the assignment of roles, definition of transport routes, approvals, and comes with additional useful features like logging).</li> <li>• If there's no on-premise CTS+ available in your landscape, a good option is to go for SAP Cloud Transport Management. SAP Cloud Transport Management is the solution for transporting content in SAP BTP. Compared to using CTS+, this option initially implies a smaller investment in implementation and operation. Note that SAP Cloud Transport Management is a cloud service (subscription-based with maintenance and updates by SAP). It's therefore the way to go for customers planning to implement projects using SAP BTP services.</li> </ul>
MTAR Download	<ul style="list-style-type: none"> <li>• MTAR Download is a semiautomated option (in between manual export/import on the one hand and using CTS+ or SAP Cloud Transport Management on the other hand).</li> <li>• It allows you to download an integration package manually from a source tenant as an .mtar file. In a subsequent step, you can upload this file to the transport system and, finally, transport it to the target tenant.</li> <li>• Using this option, you benefit of having certain advantages of a transport system. However, there's still a manual step. As manual steps should be avoided as far as possible, it's recommended to use this option only in exceptional cases, for example for urgent fixes in case the connection between source tenant and transport system is down.</li> </ul>

### i Note

The manual export/import and the MTAR download options are not discussed any further in this document due to their lack of full automation and inherent lack of ability to support any governance model.

## Prerequisites

For the two automated approaches specified above, there are prerequisites that need to be performed depending on the environment your SAP BTP is deployed in.

For more information, see [Enabling Content Transport, Cloud Foundry Environment](#).

## CTS+

For more information on enabling the use of CTS+, see [Content Transport Using CTS+](#).

## SAP Cloud Transport Management Service

For setup instructions for using SAP Cloud Transport Management, see [Content Transport Using Transport Management Service](#).

For more information on using the SAP Cloud Transport Management solution, see also the blog [Introducing SAP Content Agent service: Enhanced Transport Capabilities for Cloud Integration Content](#)

## SAP Solution Manager

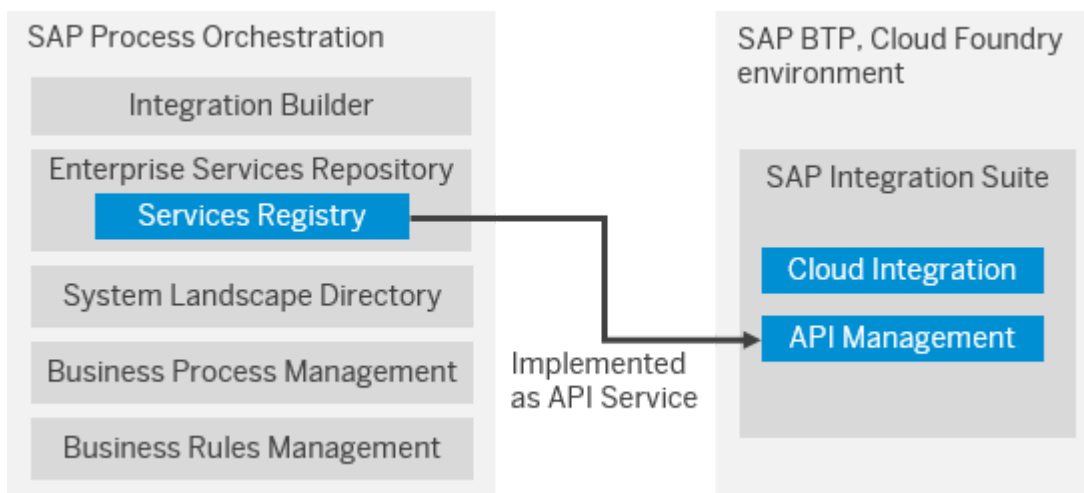
For initial information on connecting your SAP Solution Manager system to various cloud services (including Cloud Integration), see [Integrating Cloud Services in SAP Solution Manager](#).

## Services Registry to API Management

Find out more about the API Management capability.

The API Management capability comes with the objective to create an enterprise-wide governance, secure and harmonized experience for customers, partners and employees while consuming exposed APIs. API Management isn't only a service to gateway some particular API, but it also counts with a try-subscribe developer portal for external parties, real-time insights, and analytics on the API traffic and meter/monetization possibilities.

The following image shows how you can position the API Management while migrating from SAP Process Orchestration:



You can find more details about the API Management capability and how to provision it under the following links:

- [What is API Management](#)
- [Initial Setup for API management](#)
- [Enable SAP Cloud Platform API Management in Cloud Foundry Environment](#) ➔

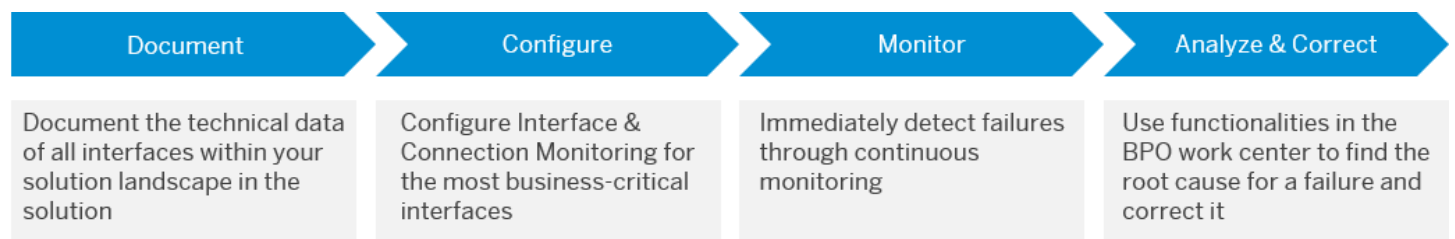
## Interface Documentation

SAP Solution Manager 7.2 makes it possible to manage the interface documentation and document it sustainably.

Interface Documentation in SAP Solution Manager 7.2 as part of Solution Documentation provides the possibility to centrally document interfaces in the solution landscape. All interface technologies that exist in an SAP landscape are available, including SAP PI interfaces where the processing is done through different kinds of adapters. Each interface can be specified individually, as every interface technology provides a set of so-called interface attributes that come out-of-the-box. In case further interface attributes are needed, it's possible to create custom attributes and use them in the same way as the attributes delivered by SAP. Moreover, custom-specific interface technologies can be defined in case the standard technologies are insufficient to describe the interface properly.

Moreover, many customers still use offline documents like spreadsheets to list their interfaces, which are cumbersome to handle and often pose the risk that data gets lost or outdated information is used because of improper document management. The Interface Documentation application aims at replacing this type of maintenance mode. Although it's some initial effort to transfer the offline documentation into the SAP system, it's worthwhile to do so. Interfaces and their attribute data can then benefit from all features Solution Documentation provides. For example, different versions of the same interface can exist in parallel, and attribute data for the same interface can vary per site or system role.

In addition, integration with the Interface & Connection Monitoring application is available: interface attribute data maintained in Interface Documentation can be used to configure Interface Channels for monitoring & alerting, without having to enter the same data into the system again. This integration reflects the best practice-like approach for Interface Management in SAP Solution Manager as indicated in the following figure.



For more details and setup instructions, see the wiki page [Interface Documentation for Solution Manager 7.2](#) ➔ and the blog [Interface Documentation with SAP Solution Manager](#) ➔.

## Documenting APIs with OpenAPI

For a more developer-oriented method to generate interface documentation, you can create API specifications that are available for all the developers that will use it in the future.

In SAP API Management, you can design you own RESTful API definitions using the API Designer UI and [OpenAPI specification](#) ➔ (formerly known as Swagger Specification).

With that, you can describe and document different aspects of your API using a language-agnostic syntax, including:

- Available endpoints and methods (e.g., GET, POST)

- Input and output for each operation
- Authentication methods
- Description of usage, license, terms, and other information.

```

1  swagger: "2.0"
2  info:
3    version: "1.0.0"
4    title: "PartnerBanking Information API"
5    x-targetEndpoint: "https://trial.apim1.hanatrial.ondemand.com"
6    host: "trial.apim1.hanatrial.ondemand.com:443"
7    basePath: "/v1/PartnerBankingInformation"
8    schemes:
9      - "http"
10     - "https"
11    consumes:
12     - "application/json"
13    produces:
14     - "application/json"
15    paths:
16     /partnerBankingInformation:
17       get:
18         description: "\nReturns all the banking information of the partner specified\
19         \ through its name or ID\n\nExample of use:\n\n/partnerBankingInformation?apiKey=123&partnerId=123456\n"
20         operationId: "getPartnerBankingInformation"
21         parameters:
22           - name: "apiKey"
23             in: "query"
24             description: "The API key provided by the Developer Portal upon API subscription"
25             required: true
26             type: "string"
27           - name: "partnerName"
28             in: "query"
29             description: "The name of the partner"
30             required: false
31             type: "string"
32           - name: "partnerId"
33             in: "query"
34             description: "The partner id number"
35             required: false
36             type: "string"
37         responses:
38           200:
39             description: "Success"
40             schema:
41               $ref: "#/definitions/partnerBankingInformation"
42             default:
43               description: "Error"
44               schema:
45                 $ref: "#/definitions/ErrorResponse"

```

With the OpenAPI specification, you can import it in different tools for different purposes (test generation, client code generation, server code generation) or you can also import into SAP API Management to generate the APIs proxy.



**PartnerBanking Information API** ▼

---

Version: 1.0.0

default ▼

**GET** /partnerBankingInformation Try out

**POST** /partnerBankingInformation Try out

**PUT** /partnerBankingInformation Try out

Updates a partner banking information.

**Parameters** Try out

Name	Description
<b>apiKey</b> * required string <i>(query)</i>	The API key provided by the Developer Portal upon API subscription <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;">apiKey - The API key provided by the Develo</div>
<b>partnerID</b> * required string <i>(query)</i>	The partner id number <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;">partnerID - The partner id number</div>
<b>partnerName</b> * required string	The partner name <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"></div>

For more information about how to use API Designer, see [Create an API from API Designer](#).

For more information about how to import OpenAPI (Swagger) files into API Management, see [Import an API](#).

For more about how to create API services using OpenAPI, see the blog [Develop and Manage API-first Enterprise Microservices in API Management](#).

## Interface Migration

Learn about the different aspects involved in migrating your interfaces from SAP Process Orchestration to SAP Integration Suite.

### [Interface Migration Strategy](#)

When migrating from SAP Process Orchestration to SAP Integration Suite on Cloud Foundry, there are different options and strategies for the migration process itself.

### [Scenario and Object Assessment](#)

Get an overview of the scenarios and objects that are common scope of migration projects, and learn about the coverage aspects between the two solutions, SAP Process Orchestration and Cloud Integration.

### [Test Automation](#)

As part of the testing phase in every migration project, several products on the market help you to create automatic testing scenarios. Test automation reduces risks to critical business processes and lowers the human effort to cover the test tasks of your interfaces.

### [Migrating Proxy Interfaces](#)

When working with ABAP proxies in the context of SAP Process Orchestration, you need to follow certain steps to ensure successful integration. The service interfaces are defined in the Enterprise Services Repository (ESR), which then generates the ABAP objects using the SPROXY transaction on the SAP side.

### [Directory API](#)

The SAP Process Orchestration Directory API is a set of SOAP APIs to read and change objects from the Integration Builder and Integration Directory.

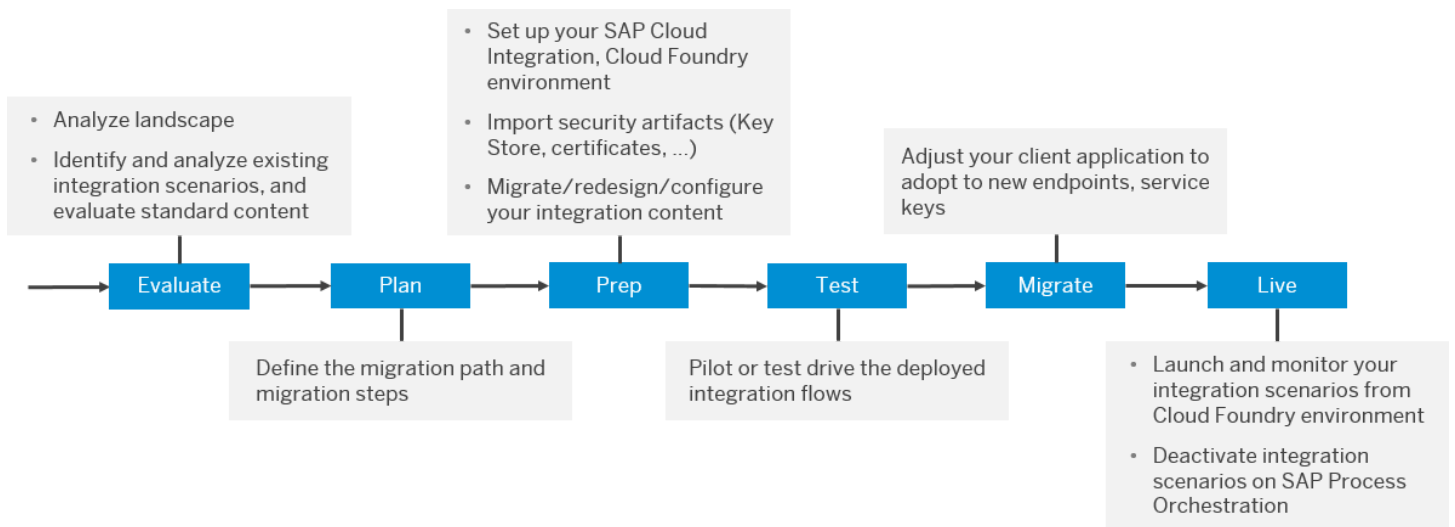
### [Interface Design and Implementation](#)

In a migration project, you may need to redesign a particular integration flow or develop one from scratch in an implementation project.

## Interface Migration Strategy

When migrating from SAP Process Orchestration to SAP Integration Suite on Cloud Foundry, there are different options and strategies for the migration process itself.

The migration process itself can be covered by a migration project, which can contain different phases as illustrated in the following diagram:



The migration phases can be adjusted to your needs and the preferred project methodology. There are different options to execute the phases.

### 1. Evaluate

- Estimate the effort involved in the migration process and how your integration scenarios can be migrated using [Migration Assessment](#).
- Analyze the current and the to-be landscape, for example, using the [SAP Integration Solution Advisory Methodology](#).
- Create a high-level architecture for the connected systems and a list of these systems with the protocols, adapters, triggers, and authentication used.
- Check if there are other options to integrate the systems on the [SAP Business Accelerator Hub](#), such as standard integration packages or standard API documentation.

### 2. Plan: create a migration strategy

- Migration of integration scenarios
  - Automatically migrated (depending on the features of the [Migration Tooling](#))
  - Semiautomatically migrated ([Migration Tooling](#), import of mappings, import of artifacts)
  - Redesign scenarios (standard content, standard APIs, interfaces not covered by the Migration Tooling)
- Side-by-side migration: Scenarios can be converted and run in parallel to SAP Process Orchestration.
- Prioritize integration scenarios depending on customer-specific aspects, for example business or functional requirements, and availability of features on Cloud Integration.

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

### 3. Prepare/Implement

- Set up Cloud Foundry
  - Provision the tenant and SAP Integration Suite
  - Configure SAP Integration Suite (authorization and roles)
- Import or configure artifacts (certificates, user credentials, known-host file, public and private keyring)
- Setup Cloud Connector (if required)
- Test the connectivity to the connected applications (optional)
- Setup implementation guidelines and architecture (if you plan to redesign integration scenarios)
- Migrate content
  - Configure standard content
  - Migrate automatically using the [Migration Tooling](#)
  - Reimplement redesigned interfaces and objects not supported by the Migration Tooling

### 4. Test

Run test phases to test the migrated content (e.g. connectivity, developer, unit, process, system, acceptance)

### 5. Migrate Connected System / Go Live

- Activate integration content on the productive tenant
- Configure the connected system to switch over to Cloud Integration

### 6. Live

- Monitoring of integration processes
- Decommission plan for SAP Process Orchestration

## Migrating Cloud Integration from the Neo Environment to the Multi-Cloud Foundation

The [guide for migrating integration scenarios in the Cloud Integration tenant from Neo environment to the Multi-Cloud Foundation](#) addresses Cloud Integration customers who have at least one existing integration scenario in the Neo environment that they would like to move to a tenant in the multi-cloud foundation.

You can also use this guide to migrate integration scenarios from one tenant to another tenant within the same environment.

Migrating Cloud Integration from Neo to Cloud Foundry allows you to stay up to date with the newest updates and offerings of SAP BTP. Benefits include the following:

- Availability on hyperscale environments, such as Amazon Web Services, Microsoft Azure, and Alibaba Cloud
- Scalability
- Asynchronous JMS message queues
- B2B libraries
- Different services in SAP Integration Suite in a unified commercial offering
- No downtime for upgrade and maintenance activities

Restrictions you may experience while migrating from Neo to Cloud Foundry include the following:

- The audit log access via Cloud Integration monitoring UI is not supported.
- Access to system log files is not supported.
- RFC adapter with principal propagation on a tenant that is hosted in Cloud Foundry environment is not supported.
- Customers can move their Cloud integration license from Neo to Cloud Foundry and perform the technical migration of the Content tenant afterwards.

### **i Note**

Such “as-is” migration is not available for Enterprise Edition, App Edition, and Enterprise Messaging Upsell (8005999). SAP Integration licenses shall be considered for such cases. For more details, contact your SAP representative.

For the full list, see SAP Note [2752867](#).

## Scenario and Object Assessment

Get an overview of the scenarios and objects that are common scope of migration projects, and learn about the coverage aspects between the two solutions, SAP Process Orchestration and Cloud Integration.

### **i Note**

Before performing the necessary analyses of your migration project, access the [SAP Business Accelerator Hub](#) to find predelivered templates, packages, and blueprints for various market cloud solutions.

Since SAP regularly delivers updates and features to SAP Integration Suite, follow the previous and future releases in the [Road Map Explorer](#) and [What's New in SAP Integration Suite](#).

You can use the SAP Integration Suite capability [Migration Assessment](#) to estimate the technical effort involved in the migration process and evaluate how your integration scenarios might be migrated.

## Scenario and Object Overview

This table highlights some scenarios and objects with the scope of migrating from SAP Process Orchestration:

Scenarios / Objects	Cloud Integration
Message Mapping	Cloud Integration supports various types of mapping methods, such as Operation Mapping (from SAP Process Orchestration), Message Mapping (XML and JSON), XSLT, and Groovy Script.  See <a href="#">Import Mapping Artifacts</a> for strategies on migrating each type of object.
Adapters	See <a href="#">Connectivity</a> .

Scenarios / Objects	Cloud Integration
B2B	<p>There are some limitations compared to SAP Process Orchestration regarding adapters and features.</p> <p><b>i Note</b></p> <p>For more information on new features and future releases, access the <a href="#">Road Map Explorer</a> and <a href="#">What's New in SAP Integration Suite</a>.</p> <p>Regarding B2B Scenarios, SAP recommends using the <a href="#">Integration Content Advisor</a>.</p> <p>For more information, see the blog <a href="#">Partner Directory - Step-by-Step example</a>.</p>
B2G	Standard Content available via <a href="#">SAP Business Accelerator Hub</a>
OP2OP	<p><b>! Restriction</b></p> <p>This feature is planned for the future and is subject to change. For more information on new features and future releases, access the <a href="#">Road Map Explorer</a> and <a href="#">What's New in SAP Integration Suite</a>.</p> <p>For ground-to-ground scenarios, SAP plans to provide a <a href="#">hybrid deployment option</a> with which you can leverage all the capabilities of the Cloud Integration, but in a local deployment.</p> <p>You can also use the Integration Gateway Component to execute and operate cloud integration content from the Cloud Integration in the Advanced Adapter Engine with all the deployment options (PI-AEX, decentral Adapter Engine, SAP Process Orchestration and SAP PI dual usage type).</p> <p>See <a href="#">Enable the Integration Gateway Component</a> and SAP Note <a href="#">2428801</a>.</p>
Message Monitoring	See <a href="#">Monitoring</a> in <a href="#">Error Handling and Logging Strategy</a> .
Deployment	See <a href="#">Interface Governance</a> .
Security	See <a href="#">Security</a> .
User Governance	See <a href="#">Interface Governance</a> .
Standard Content	See <a href="#">SAP Business Accelerator Hub</a> for new standard content released for SAP Integration Suite.
Debug/Test	See <a href="#">Error Handling and Logging Strategy</a> .
Alerting	See <a href="#">Error Handling and Logging Strategy</a> .
Job Scheduler	When developing an integration flow to create a scheduled execution, you can use timer events. See <a href="#">Add a Timer Start Event</a> .

Scenarios / Objects	Cloud Integration
SWCV	<p>Cloud Integration doesn't have the concept of software component (defined in SLD) and namespace.</p> <p>Instead, you can create a package into your tenant and create some design artifacts, such as integration flow and value mapping. The following artifacts can be created inside a package:</p> <ul style="list-style-type: none"> <li>• Integration Flow</li> <li>• REST API</li> <li>• Message Mapping</li> <li>• SOAP API</li> <li>• Value Mapping</li> <li>• OData API</li> <li>• Script Collection</li> <li>• Integration Adapter</li> </ul>
Service Registry	<p>In Cloud Integration, there's no concept equivalent to Service Registry.</p> <p>Instead, combine your interfaces created in Cloud Integration with the API Management capability. This way, you can create an omni-channel experience for customers and employees (mobile, web, devices) with benefits such as efficient API publishing, broad analytics, and new revenue streams (Data-as-a-Service). See <a href="#">What is API Management</a>.</p>

## Import Mapping Artifacts

To import or migrate import mapping artifacts from SAP Process Orchestration to Cloud Integration, you can use two features included in the Cloud Integration capability:

- Use the **Migration Tooling** feature to migrate your integration artifacts to SAP Integration Suite. For details about the feature, supported components and scenarios, and known limitations, see [Migration Tooling](#).
- You can also **import mappings from the Enterprise Services Repository**. This feature lets you connect your cloud instance to the Enterprise Services Repository using Cloud Connector and move these mapping artifacts to your integration flow. See [Importing Content from ES Repository](#).



The following limitations and possible workarounds exist when importing content from the ES Repository:

Mappings	Limitations and Workaround
Graphical Mapping (UDF)	<p>User-Defined Functions (UDFs) with function libraries, imported archives, or parameters are not supported when importing the message mapping.</p> <p>However, local Java UDFs are supported and can be viewed and edited.</p> <p><b>i Note</b></p> <p>Lookups (e.g., RFC, JDBC) are not supported. Use additional integration flow steps instead.</p>
XSLT	XSLT files are downloaded as .xsl under <code>src.main.resources.mapping</code> .

Mappings	Limitations and Workaround
Value Mappings	<p>Message mapping containing reference to value mapping is not supported at runtime. Instead, redesign your message mapping and use the value mapping of Cloud Integration.</p> <p>See also <a href="#">Migration of Value Mappings</a>.</p>
Fixed Values	<p>Although it's supported in the automatic import feature, it's highly recommended to change the Fixed Values reference to a Value Mapping in Cloud Integration. This way, you can reuse this artifact and maintain the values without changing the integration flow when necessary.</p> <p>For more information about how to import a CSV file as a Value Mapping, see <a href="#">Formatting Guidelines for CSV Files used in Value Mapping</a>, or <a href="#">Connectivity</a>.</p>
Libraries	<p>The JAR files that contain the external Java libraries can be imported manually in Cloud Integration.</p>
Function Libraries	<p>Message mapping containing reference to function libraries is not supported. You need to redesign these mappings, using Javascript/Groovy script. The scripts can be created locally in the integration flow or you can create a script collection to make it available for different integration flows inside your package. See the blog <a href="#">Script collection reusable artifact in SAP Cloud Integration</a>.</p> <p><b>i Note</b></p> <p>Lookups (e.g., RFC, JDBC) are not supported. Use additional integration flow steps instead.</p>
Java Mappings	<p>The JAR files can be imported and reused in Cloud Integration.</p> <p>You can also work with Javascript/Groovy scripts, which is the preferable approach when working in Cloud Integration. In this option, you can edit the script without having to reimport a compiled version.</p> <p>See the blog <a href="#">Using Java Mappings in SAP Cloud Platform Integration</a>.</p>
ABAP Mapping	<p>ABAP mapping is not supported. The logic must be reimplemented using any of other mapping methods in Cloud Integration: Graphical Mapping, XSLT, or Javascript/Groovy script.</p>

For more information about how to set up the connection between SAP Process Orchestration and Cloud Integration, see [Configuring Connectivity to ES Repository](#) or the blog [How to connect CPI and PI system to import mappings via SAP Cloud Connector](#).

## Test Automation

As part of the testing phase in every migration project, several products on the market help you to create automatic testing scenarios. Test automation reduces risks to critical business processes and lowers the human effort to cover the test tasks of your interfaces.

Some of these tools support different SAP products (e.g. SAP Process Orchestration, Cloud Integration) and can be used not only in the project but also in the future developments and maintenance. Note that it's important to verify the pricing conditions for each product.

For example, regression and test tools for your migration to the cloud are provided by our partner companies Int4 and Figaf. Read about their test automation tools in the blog [Leverage free test automation tools to accelerate your migration from SAP Process Orchestration to SAP Integration Suite](#).

Additional tools are available from SAP's partner ecosystem at [SAP Store](#).

## Migrating Proxy Interfaces

When working with ABAP proxies in the context of SAP Process Orchestration, you need to follow certain steps to ensure successful integration. The service interfaces are defined in the Enterprise Services Repository (ESR), which then generates the ABAP objects using the SPROXY transaction on the SAP side.

If you're migrating from SAP Process Orchestration to SAP Integration Suite, there are some key differences to be aware of when it comes to working with ABAP proxies. For example, the way that ABAP proxies are deployed and managed, and you need to adjust your existing processes and configurations accordingly. Additionally, there may be differences in the way that ABAP proxies interact with other systems and technologies within the SAP Integration Suite.

In order to successfully proceed with these changes, you need to have a solid understanding of the underlying principles and best practices as outlined in [Setting Up XI Proxy Runtime](#), [Message Reprocessing](#), and [Migrating ESR Proxies](#).

## Known Limitations

The following limitations apply when migrating proxy interfaces from SAP Process Orchestration to SAP Integration Suite:

Limitations	Workaround
ABAP Proxy generation using transaction SPROXY on the SAP system is not supported if you're using Cloud Integration.	<ul style="list-style-type: none"> <li>Use the Enterprise Services Repository to create or update ABAP proxies, but run these interfaces on Cloud Integration.</li> <li>Migrate your existing proxy, if possible, or create a new proxy interface using Metadata Repository (MDR).</li> <li>Maintain your WSDL on your preferred tool and import them through transaction SE80.</li> </ul>
Acknowledgement is not supported.	There's currently no workaround for this limitation.
ExactlyOnceInOrder (EOIO) is not supported.	A custom solution can be built to read the messages from a data store and based on their entry sequence, using for instance a timestamp and/or sequence ID, deliver the message in order.
There is a limit on the JMS resources.	Analyze your landscape and according to the expected usage of the resources and decide on the number of queues, consumers, etc. See <a href="#">JMS Resource Limits and Optimizing their Usage</a> .

## Setting Up XI Proxy Runtime

Perform the following steps to set up the XI proxy runtime.

### Prerequisites

If it's your first time setting up the outbound connectivity from SAP ERP to Cloud Integration, import the server certificates as described in SAP Note [3193513](#).

### Context

While SAP Process Orchestration uses a single RFC destination for all outbound communication from the SAP system to the SAP Process Orchestration system, Cloud Integration requires the creation of a unique destination for each interface. The destination must be configured with the target endpoint of the integration flow once it's been deployed on the Cloud Integration tenant.



You can also implement a router integration flow to receive multiples messages in a single-entry point. See [Cloud Integration - Configuring Scenario with XI Sender handling Multiple Interfaces](#).

## Procedure

1. To create an RFC destination, in your SAP system, run transaction **SM59** and choose **Create**.
2. In the dialog, enter following details:

Field	Value
<b>RFC Destination</b>	Enter an RFC destination name, for example, CI_XI_<Interface Identification>.
<b>Connection Type</b>	G HTTP connection to external server

3. In the tab **Logon & Security**, enter the required security settings.

If you're connecting to Cloud Integration using basic authentication, see [Creating Service Instance and Service Key for Inbound Authentication](#).

4. Next, you need to create a sender/receiver definition in the sender backend system. It determines which interfaces are sent through the RFC destination you created previously.

Run transaction **SXMSIF** and choose **New Entries**.

5. Enter the following details for your new sender/receiver definition:

Field	Description
<b>Sender/Receiver ID</b>	Identification of your interface, for example BP_BULK
<b>Description</b>	Description of your settings
<b>Agency</b>	Sender agency identification, if applicable.
<b>Schema</b>	Sender schema identification, if applicable.
<b>Party</b>	Sender party identification, if applicable
<b>Component</b>	Sender business system/component name

6. In section **Message Category**, select **Request** and enter the following details:

Field	Description
<b>Interface Name</b>	Identification of the sender service interface, for example BusinessPartnerSUITEBulkReplicateRequest_Out
<b>Interface Namespace</b>	Identification of the service interface namespace, for example, <a href="http://sap.com/xi/SAP_BS_FND/MDG/Global2">http://sap.com/xi/SAP_BS_FND/MDG/Global2</a>

### i Note

You can use wildcard characters, like \*, to indicate multiple possible characters when filling out the fields.

7. Next, you can continue with configuring the interface-specific runtime.

Run transaction **SXMB\_ADMIN** and, in section **Configuration**, choose **Integration Engine Configuration**.

8. To view the list of parameters for the integration engine, choose **Configuration**.

Choose **New Entries** and enter the following details:

Field	Value
Category	RUNTIME
Parameters	IS_URL
Subparameter	Enter the sender/receiver ID
Current Value	dest://<Destination ID>

9. Finally, if you use the Cloud Integration runtime and no SLD business system is connected to your SAP system, verify that an entry in table LCRT\_CLNTCACHE represents the business system of your SAP system.

You can do so by running transaction **SE16N** and maintaining an entry like the following:

ID	Creation Date	Business System	Role	Caption
<SID><CLIENT>, for example PRD100	Date of creation	Identification of your choice, for example, PRDCLNT100	LOC	Caption of your choice, for example, PRDCLNT100

You only have to perform this action once.

## Message Reprocessing

To ensure a successful migration, you can use reprocessing functionalities to retry processing your ABAP proxies in case of errors.

SAP Process Orchestration provides a message monitoring feature that enables users to monitor messages passing through the system. This feature allows you to view and manage messages that have failed to process, messages that are currently being processed, and messages that have been successfully processed. SAP Process Orchestration also supports a retry mechanism that allows you to define the number of times a message should be retried if it fails to process. You can also define the time interval between each retry attempt.

In Cloud Integration, the reprocessing of messages is done by using either the temporary storage data store or a JMS queue. When a data store is used, the message is stored temporarily in the tenant's database and can be monitored in the data store monitor. When a JMS queue is used, the message is stored in a JMS queue on the connected JMS broker. It's recommended to use a JMS queue because it's more performant.

For more information, see [Cloud Integration - Configuring Scenario Using the XI Sender Adapter](#).

## Migrating ESR Proxies

Migrate existing ESR proxies from the Enterprise Services Repository (ESR) to the Metadata Repository (MDR).

Use transaction **SPXNMIG** and report **SPROX\_MDR\_MIGRATION** to migrate existing ESR proxies to the MDR. The web services are then available locally in the backend system where they can be changed and extended.

### i Note

You don't have to migrate all your existing proxies from ESR to MDR. You only need to migration the proxies that you need to maintain the structure later.

## Prerequisites

Before the migration, define which namespaces should be considered by assigning them to the MDR. Since MDR proxies and ESR proxies share the same namespaces, the system checks which namespaces should be assigned whenever a proxy object is to be generated.

1. Open transaction [SPXNGENAPPL](#) to customize the default settings for namespace assignment.

By default, ESR is considered for Business Suite systems and MDR for Business By Design.

2. Maintain the list of namespaces by selecting [Backend Metadata Repository](#) as the [Generation Source](#).

Since wildcards are allowed, you can create a generic entry \* that considers all valid namespaces.

3. Save your changes.

## Steps

1. In transaction [SPXNMIG](#), define a namespace range and select [Execute](#). If no namespace range is provided, all entries previously maintained in [SPXNGENAPPL](#) are used as selection.

2. The system displays a list of all proxy objects that can be migrated, with information if additional steps are needed. It cross-checks the ESR object/namespace and MDR namespace assignment.

Some object types cannot be migrated and are therefore not included in the list. See [Restrictions](#).

3. Select [Migrate](#) to trigger the migration for the complete list, or a previously selected subset. Once the migration is finished, the results are displayed.

## Restrictions

The following objects cannot be migrated and are therefore not displayed in the list of proxies to be migrated:

- **External definitions**

Migrating ESR interfaces based on external definitions is not supported. They can be replaced by an external consumer/provider generated from WSDL directly in backend. A new ABAP name for the class is then generated so the previous usage should be reviewed.

- **SAP namespace (\*.sap.com)**

Objects from SAP namespace cannot be migrated.

- **Inactive proxy objects**

Additionally, **inline types** cannot be migrated to the MDR automatically. You can migrate them manually by using one of the following options:

- Change inline types to global types in the ESR and regenerating them to become global types. Instead of defining dependent elements inline, you need to create these elements separately and refer to them within the parent element. The following examples show the difference between an inline type and a global type:

- For the **inline type**, all XSD elements are contained within one sequence:

```
<xsd:complexType name="dt_inline">
  <xsd:sequence>
    <xsd:element name="tier_1">
      <xsd:complexType>
        <xsd:sequence>
```

```

        <xsd:element name="tier_2">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="integ
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

```


- o However, the **global type** consists of three different XSD parts:

```

<xsd:complexType name="dt_global">
    <xsd:sequence>
        <xsd:element name="tier_1" type="dt_tier1"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="dt_tier1">
    <xsd:sequence>
        <xsd:element name="tier_2" type="dt_tier2"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="dt_tier2">
    <xsd:sequence>
        <xsd:element name="integer" type="xsd:integer"/>
    </xsd:sequence>
</xsd:complexType>

```

- For systems classified as **SAP only**:

Using the migration tool, you can convert inline types semiautomatically with the button **Convert Inlines**. This means that inline types can be converted into global types by specifying a global name for each inline type the tool needs to generate a global type for. See SAP Note [3313582](#) .

### i Note

If this option is used, a migration back to ESR is not possible.

## New Interfaces

For new interfaces, the creation of objects should be done directly in the backend. You can use wizards for this task.

For more information, see:

- [Modeling Web Service Objects in the Backend](#)
- [Enterprise Services Wizard](#)
- [MDR: Defining web services from ABAP without requiring PI](#) 

## Directory API

The SAP Process Orchestration Directory API is a set of SOAP APIs to read and change objects from the Integration Builder and Integration Directory.

## Context

SAP Migration Assessment and the Migration Tooling use the Directory API to extract all necessary data for the migration from SAP Process Orchestration. See, for example, [Add an SAP Process Orchestration System](#) in the Migration Assessment documentation.

There are several operations possible for different objects, for example, delete integrated configurations, or read detailed information from each communication channel. You can also make mass changes in the Integration Directory. Using the programming interface, you can carry out almost all those actions that you can also carry out using the user interface (Integration Builder).

### i Note

Make sure that the user used on the API has the assigned roles: SAP\_XI\_API\_DISPLAY\_J2EE and SAP\_XI\_API\_DEVELOP\_J2EE.

## Procedure

### Using the Web Services Navigator

1. Open the Web Services Navigator using one of the following options:
  - Open SAP NetWeaver Application Server and choose **Web Services Navigator**.
  - Navigate directly to the URL `http(s)://<host>:<port>/wsnavigator`.
  - Open the SAP NetWeaver Administrator (`http(s)://<host>:<port>/nwa`) and navigate to **Configuration > Connectivity > Single Service Administration**.
2. Change the **Search Type** to **Provider System**.
3. In **Search for**, enter an API and choose **Search**.

For a list of the APIs used in Migration Assessment and Migration Tooling, see [APIs in the Directory API](#).

4. Select an API and choose **Next**.
  5. Select an operation and choose **Next**.
  6. Enter the input parameters according to the API selected in the previous step. Fill the user credentials at **Invocation Parameters**.
  7. The result is shown. You can check the execution result and download it at **Result > XML Content > Download as File**.
- You can use the request message while testing it on an HTTP client tool.

### Saving the WSDL

Save the WSDL from any directory API using the following format call:

- **WSDL URL:** `http(s)://<host>:<port>/<ServiceName>InService/<ServiceName>ImplBean?wsdl`
- **Example:** `http(s)://<host>:<port>/IntegratedConfigurationInService/IntegratedConfigurationInImplBean?wsdl`

## APIs in the Directory API

The following APIs are used in Migration Assessment and Migration Tooling:

- `CommunicationChannelInService`

- `IntegratedConfigurationInService`
- `IntegratedConfiguration750InService`
- `SenderAgreementInService`
- `ReceiverAgreementInService`
- `ValueMappingInService`
- `ConfigurationScenarioInService`
- `AlertRuleInService`

Service / Base URL: `http(s)://<host>:<port>/<ServiceName>InService/<ServiceName>ImplBean`

### **CommunicationChannelInService**

With the **CommunicationChannelIn** API, you get detailed information about a communication channel.

An XML file is created with the administrative data, adapter metadata, transport / message protocol, all adapter-specific attributes, and some other information.

For the read request, `Component ID` and `Channel ID` are mandatory.

**Endpoint URL:** `http(s):// <host>:<port>/CommunicationChannelInService/CommunicationChannelInImplBean`

**Possible Operations:** Change, Check, Create, CreateFromTemplate, Delete, OpenForEdit, Query, Read, Revert

See [Example for CommunicationChannelInService](#).

### **IntegratedConfigurationInService**

With the **IntegratedConfigurationIn** API, you get detailed information about an integrated configuration.

An XML file is created with the administrative data, inbound and outbound processing, communication channel with all adapter-specific attributes, the receiver, and some other information.

For the read request, the `Sender Component ID`, `Sender Interface Name`, and `Namespace` are mandatory.

**Endpoint URL:** `http(s):// <host>:<port>/IntegratedConfigurationInService/IntegratedConfigurationInImplBean`

**Possible Operations:** Change, Check, Create, Delete, OpenForEdit, Query, Read, Revert

See [Example for IntegratedConfigurationInService](#).

### **IntegratedConfiguration750InService**

With the **IntegratedConfiguration750In** API, you get detailed information about an integrated configuration.

#### **i Note**

This API is only available on SAP Process Orchestration, SAP NetWeaver 7.5.

#### **i Note**

In contrast to `IntegratedConfigurationIn`, `IntegratedConfiguration750In` has `ReceiverWildcardIndicator` in the XML output.

An XML file is created with the administrative data, inbound and outbound processing, communication channel with all adapter-specific attributes, the receiver, and some other information.

For the read request, the `Sender Component ID`, `Sender Interface Name`, and `Namespace` are mandatory.

**Endpoint URL:** `http(s):// <host>:<port>/IntegratedConfiguration750InService/IntegratedConfiguration750InImplBean`

**Possible Operations:** Change, Check, Create, Delete, OpenForEdit, Query, Read, Revert

See [Example for IntegratedConfiguration750InService](#).

### SenderAgreementInService

With the `SenderAgreementIn` API, you get detailed information about the sender agreement.

An XML file is created with the administrative data, sender and receiver information, communication channel with all adapter-specific attributes, and some other information.

For the read request, the `Sender Component ID`, `Sender Interface Name`, and `Namespace` are mandatory.

**Endpoint URL:** `http(s):// <host>:<port>/SenderAgreementInService/SenderAgreementInImplBean`

**Possible Operations:** Change, Check, Create, Delete, OpenForEdit, Query, Read, Revert

See [Example for SenderAgreementInService](#).

### ReceiverAgreementInService

With the `ReceiverAgreementIn`, API you get detailed information about the receiver agreement.

An XML file is created with the administrative data, sender and receiver information, communication channel, and some other information.

For the read request, the `Sender Component ID`, `Receiver Interface Name`, `Namespace`, and `Receiver Component ID` are mandatory.

**Endpoint URL:** `http(s):// <host>:<port>/ReceiverAgreementInService/ReceiverAgreementInImplBean`

**Possible Operations:** Change, Check, Create, Delete, OpenForEdit, Query, Read, Revert

See [Example for ReceiverAgreementInService](#).

### ValueMappingInService

With the `ValueMappingIn` API, you get detailed information about the sender agreement.

An XML file is created with the administrative data, value mapping ID, and group name.

For the read request, the `Value Mapping ID (Group ID)` is mandatory.

**Endpoint URL:** `http(s):// <host>:<port>/ValueMappingInService/ValueMappingInImplBean`

**Possible Operations:** Change, Check, Create, Delete, OpenForEdit, Query, Read, Revert

See [Example for ValueMappingInService](#).

## ConfigurationScenarioInService

With the **ConfigurationScenarioIn** API, you get detailed information about the configuration scenario and all containing elements, for example, communication channel or integrated configuration.

An XML file is created with the administrative data, business components, communication channel, integrated configuration, and some other information.

For the read request, the `Configuration Scenario ID` is mandatory.

**Endpoint URL:** `http(s):// <host>:<port>/ConfigurationScenarioInService/ConfigurationScenarioInImplBean`

**Possible Operations:** Change, Check, Create, Delete, OpenForEdit, Query, Read, Revert

See [Example for ConfigurationScenarioInService](#).

## AlertRuleInService

With the **AlertRuleIn** API, you get a list of all alert rules with detailed information.

When `AlertRuleID` is empty in the query, a list of all rules is extracted.

No field is mandatory.

**Endpoint URL:** `http(s):// <host>:<port>/AlertRuleInService/AlertRuleInImplBeanHTTPS`

**Possible Operations:** DeregisterConsumer, Query, RegisterConsumer, RetrieveRuntimeComponents

See [Example for AlertRuleInService](#).

# Examples for Directory API

This chapter collects examples for the APIs described in [Directory API](#).

## Example for CommunicationChannelInService

### Request Example (Read)

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xs="http://www.
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <yq1:CommunicationChannelReadRequest xmlns:yq1="http://sap.com/xi/BASIS">
      <CommunicationChannelID>
        <PartyID></PartyID>
        <ComponentID>ComponentID</ComponentID>
        <ChannelID>ChannelID</ChannelID>
      </CommunicationChannelID>
    </yq1:CommunicationChannelReadRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



**Response Example (Read)**

```

<SOAP-ENV:Envelope xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-ENV="http://schemas.xmlso
  <SOAP-ENV:Body>
    <ns2:CommunicationChannelReadResponse xmlns:ns2='http://sap.com/xi/BASIS'>
      <CommunicationChannel>
        <MasterLanguage>EN</MasterLanguage>
        <AdministrativeData>
          <ResponsibleUserAccountID>User</ResponsibleUserAccountID>
          <LastChangeUserAccountID>User</LastChangeUserAccountID>
          <LastChangeDateTime>2022-12-16T08:46:11.560+01:00</LastChan
          <FolderPathID>/</FolderPathID>
        </AdministrativeData>
        <Description languageCode='EN'></Description>
        <CommunicationChannelID>
          <PartyID></PartyID>
          <ComponentID>ComponentID</ComponentID>
          <ChannelID>ChannelID</ChannelID>
        </CommunicationChannelID>
        <AdapterMetadata>
          <Name>REST</Name>
          <Namespace>http://sap.com/xi/XI/System</Namespace>
          <SoftwareComponentVersionID>SoftwareComponentVersionID</Sof
        </AdapterMetadata>
        <Direction>Receiver</Direction>
        <TransportProtocol>HTTP</TransportProtocol>
        <TransportProtocolVersion></TransportProtocolVersion>
        <MessageProtocol>REST</MessageProtocol>
          <MessageProtocolVersion></MessageProtocolVersion>
        <AdapterEngineName></AdapterEngineName>
        <AdapterSpecificAttribute>
          <Name>UseProxy</Name>
          <Namespace></Namespace>
          <Value>0</Value>
        </AdapterSpecificAttribute>
        <AdapterSpecificAttribute>
          <Name>ProxyHost</Name>
          <Namespace></Namespace>
          <Value>ProxyHost</Value>
        </AdapterSpecificAttribute>
        <AdapterSpecificAttribute>
          ...
        </AdapterSpecificAttribute>
        <AdapterSpecificAttribute>
          ...
        </AdapterSpecificAttribute>
        ...
      </ModuleProcess>
        <ProcessStep>
          <ModuleName>sap.com/com.sap.aii.adapter.rest.app/RE
          <ModuleType>Local Enterprise Bean</ModuleType>
          <ParameterGroupID>rest</ParameterGroupID>
        </ProcessStep>
      </ModuleProcess>
      <SenderIdentifier schemeAgencyID='' schemeID=''></SenderIdentifier>
      <ReceiverIdentifier schemeAgencyID='' schemeID=''></ReceiverIdentif
    </CommunicationChannel>
    <LogMessageCollection></LogMessageCollection>
  </ns2:CommunicationChannelReadResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

**Example for IntegratedConfigurationInService****Request Example (Read)**

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xs="http://www.
  <SOAP-ENV:Body>
    <yq1:IntegratedConfigurationReadRequest xmlns:yq1="http://sap.com/xi/BASIS">
      <ReadContext>User</ReadContext>
    </yq1:IntegratedConfigurationReadRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

```

        <IntegratedConfigurationID>
          <SenderComponentID>ComponentID</SenderComponentID>
          <InterfaceName>InterfaceName</InterfaceName>
          <InterfaceNamespace>InterfaceNamespace</InterfaceNamespace>
        </IntegratedConfigurationID>
      </yq1:IntegratedConfigurationReadRequest>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>

```

## Response Example (Read)

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xs="http://www.
  <SOAP-ENV:Body>
    <ns2:IntegratedConfigurationReadResponse xmlns:ns2="http://sap.com/xi/BASIS">
      <IntegratedConfiguration>
        <MasterLanguage>EN</MasterLanguage>
        <AdministrativeData>
          <ResponsibleUserAccountID>User</ResponsibleUserAccountID>
          <LastChangeUserAccountID>User</LastChangeUserAccountID>
          <LastChangeDateTime>2022-12-01T12:21:05.328+01:00</LastChan
          <FolderPathID>/</FolderPathID>
        </AdministrativeData>
        <IntegratedConfigurationID>
          <SenderPartyID/>
          <SenderComponentID>SenderComponentID</SenderComponentID>
          <InterfaceName>InterfaceName</InterfaceName>
          <InterfaceNamespace>InterfaceNamespace</InterfaceNamespace>
          <ReceiverPartyID/>
          <ReceiverComponentID/>
        </IntegratedConfigurationID>
        <InboundProcessing>
          <SenderInterfaceSoftwareComponentVersion>1234</SenderInterf
          <CommunicationChannel>
            <PartyID/>
            <ComponentID>ComponentID</ComponentID>
            <ChannelID>hannelID</ChannelID>
          </CommunicationChannel>
          <SchemaValidationIndicator>false</SchemaValidationIndicator>
          <VirusScan>UseGlobal</VirusScan>
          <AdapterSpecificAttribute>
            <Name>wsse.namespace</Name>
            <Namespace/>
            <Value>http://docs.oasis-open.org/wss/2004/01/oasis
          </AdapterSpecificAttribute>
          ...
        </InboundProcessing>
        <Receivers>
          <ReceiverRule>
            <Receiver>
              <PartyID/>
              <ComponentID>ComponentID</ComponentID>
            </Receiver>
          </ReceiverRule>
          <NoReceiverBehaviour>ErrorMessage</NoReceiverBehaviour>
        </Receivers>
        <ReceiverInterfaces>
          <Receiver>
            <PartyID/>
            <ComponentID>ComponentID</ComponentID>
          </Receiver>
          <ReceiverInterfaceRule>
            <Operation>Operation</Operation>
            <Mapping>
              <Name>Name</Name>
              <Namespace>Namespace</Namespace>
              <SoftwareComponentVersionID>1234</SoftwareC
            </Mapping>
            <Interface>
              <Name>Name</Name>
              <Namespace>urn:Namespace</Namespace>
              <SoftwareComponentVersionID>1234</SoftwareC
            </Interface>

```

```

        </ReceiverInterfaceRule>
        <QualityOfService>E0</QualityOfService>
    </ReceiverInterfaces>
    <OutboundProcessing>
        <Receiver>
            <PartyID/>
            <ComponentID>ComponentID</ComponentID>
        </Receiver>
        <ReceiverInterface>
            <Name>Name</Name>
            <Namespace>urn:Namespace</Namespace>
            <SoftwareComponentVersionID>1234</SoftwareComponentVersionID>
        </ReceiverInterface>
        <CommunicationChannel>
            <PartyID/>
            <ComponentID>ComponentID</ComponentID>
            <ChannelID>ChannelID</ChannelID>
        </CommunicationChannel>
        <SchemaValidationIndicator>>false</SchemaValidationIndicator>
        <VirusScan>UseGlobal</VirusScan>
        <HeaderMapping>
            <Sender/>
            <Receiver/>
        </HeaderMapping>
    </OutboundProcessing>
    <Logging>
        <UseGlobal>>true</UseGlobal>
    </Logging>
    <Staging>
        <UseGlobal>>true</UseGlobal>
    </Staging>
</IntegratedConfiguration>
<LogMessageCollection/>
</ns2:IntegratedConfigurationReadResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## Example for IntegratedConfiguration750InService

### Request Example (Read)

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xs="http://www.
    <SOAP-ENV:Body>
        <yq1:IntegratedConfigurationReadRequest xmlns:yq1="http://sap.com/xi/BASIS">
            <ReadContext>User</ReadContext>
            <IntegratedConfigurationID>
                <SenderComponentID>BC_PIMAS_Sender</SenderComponentID>
                <InterfaceName>SI_P2P_ASYNC_CSV_0002_Out</InterfaceName>
                <InterfaceNamespace>urn:P2P_ASYNC_CSV:0002</InterfaceNamespace>
            </IntegratedConfigurationID>
        </yq1:IntegratedConfigurationReadRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

### Response Example (Read)

```

<?xml version="1.0" encoding="utf-8"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org
    <SOAP-ENV:Body>
        <ns2:IntegratedConfiguration750ReadResponse xmlns:ns2="http://sap.com/xi/BASIS">
            <IntegratedConfiguration>
                <MasterLanguage>EN</MasterLanguage>
                <AdministrativeData>
                    <ResponsibleUserAccountID>User</ResponsibleUserAccountID>
                    <LastChangeUserAccountID>User</LastChangeUserAccountID>
                    <LastChangeDateTime>2022-12-01T12:21:05.328+01:00</LastChan
                    <FolderPathID>/</FolderPathID>
                </AdministrativeData>
            </IntegratedConfigurationID>
            <SenderPartyID/>
        </ns2:IntegratedConfiguration750ReadResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

```

        <SenderComponentID>SenderComponentID</SenderComponentID>
        <InterfaceName>InterfaceName</InterfaceName>
        <InterfaceNamespace>urn:InterfaceNamespace</InterfaceNamesp
        <ReceiverPartyID/>
        <ReceiverComponentID/>
    </IntegratedConfigurationID>
</InboundProcessing>
<InboundProcessing>
    <SenderInterfaceSoftwareComponentVersion>1234</SenderInterf
    <CommunicationChannel>
        <PartyID/>
        <ComponentID>ComponentID</ComponentID>
        <ChannelID>ChannelID</ChannelID>
    </CommunicationChannel>
    <SchemaValidationIndicator>>false</SchemaValidationIndicator
    <VirusScan>Use Global</VirusScan>
    <AdapterSpecificAttribute>
        <Name>wsse.namespace</Name>
        <Namespace/>
        <Value>http://docs.oasis-open.org/wss/2004/01/oasis
    </AdapterSpecificAttribute>
    ...
</InboundProcessing>
<Receivers>
    <ReceiverWildcardIndicator>>false</ReceiverWildcardIndicator
    <ReceiverRule>
        <Condition/>
        <Receiver>
            <CommunicationParty>
                <TypeID>Constant</TypeID>
                <Value/>
                <Datatype>xsd:string</Datatype>
                <ContextObjectName/>
                <ContextObjectNamespace/>
            </CommunicationParty>
            <CommunicationPartySchema>
                <TypeID>Constant</TypeID>
                <Value>XIParty</Value>
                <Datatype>xsd:string</Datatype>
                <ContextObjectName/>
                <ContextObjectNamespace/>
            </CommunicationPartySchema>
            <CommunicationPartyAgency>
                <TypeID>Constant</TypeID>
                <Value>http://sap.com/xi/XI</Value>
                <Datatype>xsd:string</Datatype>
                <ContextObjectName/>
                <ContextObjectNamespace/>
            </CommunicationPartyAgency>
            <CommunicationComponent>
                <TypeID>Constant</TypeID>
                <Value>Value</Value>
                <Datatype>xsd:string</Datatype>
                <ContextObjectName/>
                <ContextObjectNamespace/>
            </CommunicationComponent>
        </Receiver>
    </ReceiverRule>
    <NoReceiverBehaviour>Error Message</NoReceiverBehaviour>
</Receivers>
<ReceiverInterfaces>
    <Receiver>
        <PartyID/>
        <ComponentID>ComponentID</ComponentID>
    </Receiver>
    <ReceiverInterfaceRule>
        <Operation>Operation</Operation>
        <Mapping>
            <Name>Name</Name>
            <Namespace>urn:Namespace</Namespace>
            <SoftwareComponentVersionID>1234</SoftwareC
        </Mapping>
    </ReceiverInterfaceRule>
    <Interface>
        <Name>Name</Name>
        <Namespace>urn:Namespace</Namespace>

```

```

                <SoftwareComponentVersionID>1234</SoftwareC
            </Interface>
        </ReceiverInterfaceRule>
        <QualityOfService>E0</QualityOfService>
    </ReceiverInterfaces>
    <OutboundProcessing>
        <Receiver>
            <PartyID/>
            <ComponentID>ComponentID</ComponentID>
        </Receiver>
        <ReceiverInterface>
            <Name>Name</Name>
            <Namespace>urn:Namespace</Namespace>
            <SoftwareComponentVersionID>1234</SoftwareComponent
        </ReceiverInterface>
        <CommunicationChannel>
            <PartyID/>
            <ComponentID>ComponentID</ComponentID>
            <ChannelID>ChannelID</ChannelID>
        </CommunicationChannel>
        <SchemaValidationIndicator>false</SchemaValidationIndicator>
        <VirusScan>Use Global</VirusScan>
        <HeaderMapping>
            <Sender/>
            <Receiver/>
        </HeaderMapping>
    </OutboundProcessing>
    <Logging>
        <UseGlobal>true</UseGlobal>
    </Logging>
    <Staging>
        <UseGlobal>true</UseGlobal>
    </Staging>
</IntegratedConfiguration>
<LogMessageCollection/>
</ns2:IntegratedConfiguration750ReadResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## Example for SenderAgreementInService

### Request Example (Read)

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xs="http://www.
    <SOAP-ENV:Body>
        <yq1:SenderAgreementReadRequest xmlns:yq1="http://sap.com/xi/BASIS">
            <ReadContext>User</ReadContext>
            <SenderAgreementID>
                <SenderComponentID>SenderComponentID</SenderComponentID>
                <InterfaceName>InterfaceName</InterfaceName>
                <InterfaceNamespace>InterfaceNamespace</InterfaceNamespace>
            </SenderAgreementID>
        </yq1:SenderAgreementReadRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

### Response Example (Read)

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xs="http://www.
    <SOAP-ENV:Body>
        <ns2:SenderAgreementReadResponse xmlns:ns2="http://sap.com/xi/BASIS">
            <SenderAgreement>
                <MasterLanguage>EN</MasterLanguage>
                <AdministrativeData>
                    <ResponsibleUserAccountID>User</ResponsibleUserAccountID>
                    <LastChangeUserAccountID>User</LastChangeUserAccountID>
                    <LastChangeDateTime>2016-04-29T11:58:35.034+02:00</LastChan
                    <FolderPathID>/</FolderPathID>
                </AdministrativeData>
            </SenderAgreement>
        </ns2:SenderAgreementReadResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

```

<Description languageCode="EN"/>
<SenderAgreementID>
  <SenderPartyID/>
  <SenderComponentID>SenderComponentID</SenderComponentID>
  <InterfaceName>InterfaceName</InterfaceName>
  <InterfaceNamespace>InterfaceNamespace</InterfaceNamespace>
  <ReceiverPartyID/>
  <ReceiverComponentID/>
</SenderAgreementID>
<CommunicationChannel>
  <PartyID/>
  <ComponentID>ComponentID</ComponentID>
  <ChannelID>ChannelID</ChannelID>
</CommunicationChannel>
<SoftwareComponentVersion>2171ef81-4319-11db-ae76-c6090a4240fd</SoftwareComponentVersion>
<ForIntegratedConfiguration>true</ForIntegratedConfiguration>
<SchemaValidation>No Validation</SchemaValidation>
<VirusScan>Use Global</VirusScan>
<AdapterSpecificAttribute>
  <Name>propagatePrincipal</Name>
  <Namespace/>
  <Value>0</Value>
</AdapterSpecificAttribute>
</SenderAgreement>
<LogMessageCollection/>
</ns2:SenderAgreementReadResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## Example for ReceiverAgreementInService

### Request Example (Read)

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xs="http://www.
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <yq1:ReceiverAgreementReadRequest xmlns:yq1="http://sap.com/xi/BASIS">
      <ReceiverAgreementID>
        <SenderPartyID></SenderPartyID>
        <SenderComponentID>SenderComponentID</SenderComponentID>
        <InterfaceName>InterfaceName</InterfaceName>
        <InterfaceNamespace>InterfaceNamespace</InterfaceNamespace>
        <ReceiverPartyID></ReceiverPartyID>
        <ReceiverComponentID>ReceiverComponentID</ReceiverComponentID>
      </ReceiverAgreementID>
    </yq1:ReceiverAgreementReadRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

### Response Example (Read)

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-ENV="http://schemas.xmlso
  <SOAP-ENV:Body>
    <ns2:ReceiverAgreementReadResponse xmlns:ns2='http://sap.com/xi/BASIS'>
      <ReceiverAgreement>
        <MasterLanguage>EN</MasterLanguage>
        <AdministrativeData>
          <ResponsibleUserAccountID>User</ResponsibleUserAccountID>
          <LastChangeUserAccountID>User</LastChangeUserAccountID>
          <LastChangeDateTime>2022-03-18T14:42:17.353Z</LastChangeDate
          <FolderPathID></FolderPathID>
        </AdministrativeData>
        <ReceiverAgreementID>
          <SenderPartyID></SenderPartyID>
          <SenderComponentID>SenderComponentID</SenderComponentID>
          <InterfaceName>InterfaceName</InterfaceName>
          <InterfaceNamespace>InterfaceNamespace</InterfaceNamespace>
          <ReceiverPartyID></ReceiverPartyID>

```

```

        <ReceiverComponentID>ReceiverComponentID</ReceiverComponent
    </ReceiverAgreementID>
    <CommunicationChannel>
        <PartyID></PartyID>
        <ComponentID>ComponentID</ComponentID>
        <ChannelID>ChannelID</ChannelID>
    </CommunicationChannel>
    <SoftwareComponentVersion>0050568f-0aac-1ee8-94ef-d636a5cec849</Sof
    <SchemaValidationIndicator>>false</SchemaValidationIndicator>
    <VirusScan>Use Global</VirusScan>
    <HeaderMapping>
        <Sender></Sender>
        <Receiver></Receiver>
    </HeaderMapping>
    </ReceiverAgreement>
    <LogMessageCollection></LogMessageCollection>
</ns2:ReceiverAgreementReadResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## Example for ValueMappingInService

### Request Example (Read)

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xs="http://www.
    <SOAP-ENV:Body>
        <yq1:ValueMappingReadRequest xmlns:yq1="http://sap.com/xi/BASIS">
            <ReadContext>User</ReadContext>
            <ValueMappingID>GroupID</ValueMappingID>
        </yq1:ValueMappingReadRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

### Response Example (Read)

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xs="http://www.
    <SOAP-ENV:Body>
        <ns2:ValueMappingReadResponse xmlns:ns2="http://sap.com/xi/BASIS">
            <ValueMapping>
                <MasterLanguage>EN</MasterLanguage>
                <AdministrativeData>
                    <ResponsibleUserAccountID>User</ResponsibleUserAccountID>
                    <LastChangeUserAccountID>User</LastChangeUserAccountID>
                    <LastChangeDateTime>2018-11-27T06:18:24.949+01:00</LastChan
                    <FolderPathID></FolderPathID>
                </AdministrativeData>
                <Description languageCode="EN"/>
                <ValueMappingID>ValueMappingID</ValueMappingID>
                <GroupName>GroupName</GroupName>
            </ValueMapping>
            <LogMessageCollection/>
        </ns2:ValueMappingReadResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## Example for ConfigurationScenarioInService

### Request Example (Read)

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xs="http://www.
    <SOAP-ENV:Body>
        <yq1:ConfigurationScenarioReadRequest xmlns:yq1="http://sap.com/xi/BASIS">
            <ReadContext>User</ReadContext>
            <ConfigurationScenarioID>PI_Elevation_PIMAS_PIMG</ConfigurationScenarioID>
        </yq1:ConfigurationScenarioReadRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

```

    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## Response Example (Read)

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xs="http://www.
  <SOAP-ENV:Body>
    <ns2:ConfigurationScenarioReadResponse xmlns:ns2="http://sap.com/xi/BASIS">
      <ConfigurationScenario>
        <MasterLanguage>EN</MasterLanguage>
        <AdministrativeData>
          <ResponsibleUserAccountID>User</ResponsibleUserAccountID>
          <LastChangeUserAccountID>User</LastChangeUserAccountID>
          <LastChangeDateTime>2023-01-13T14:34:01.971+01:00</LastChan
          <FolderPathID>/</FolderPathID>
        </AdministrativeData>
        <ConfigurationScenarioID>PI_Elevation_PIMAS_PIMG</ConfigurationScen
        <BusinessComponent>
          <PartyID/>
          <ComponentID>ComponentID</ComponentID>
        </BusinessComponent>
        <CommunicationChannel>
          <PartyID/>
          <ComponentID>ComponentID</ComponentID>
          <ChannelID>ChannelID</ChannelID>
        </CommunicationChannel>
        <CommunicationChannel>
          ...
        </CommunicationChannel>
        <IntegratedConfiguration>
          <SenderPartyID/>
          <SenderComponentID>SenderComponentID</SenderComponentID>
          <InterfaceName>InterfaceName</InterfaceName>
          <InterfaceNamespace>urn:InterfaceNamespace</InterfaceNamesp
          <ReceiverPartyID/>
          <ReceiverComponentID/>
        </IntegratedConfiguration>
        <IntegratedConfiguration>
          ...
        </IntegratedConfiguration>
      </ConfigurationScenario>
      <LogMessageCollection/>
    </ns2:ConfigurationScenarioReadResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## Example for AlertRuleInService

### Request Example (Query)

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:bas="http://sap.c
  <soapenv:Header/>
  <soapenv:Body>
    <bas:AlertRuleQueryRequest>
      <AlertRuleID></AlertRuleID>
      <Description languageCode=""></Description>
      <AdministrativeData>
        <ResponsibleUserAccountID></ResponsibleUserAccountID>
        <LastChangeUserAccountID></LastChangeUserAccountID>
        <LastChangeDateTime></LastChangeDateTime>
        <FolderPathID></FolderPathID>
      </AdministrativeData>
    </bas:AlertRuleQueryRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

### Response Example (Query)



```

<SOAP-ENV:Envelope xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-ENV="http://schemas.xmlso
  <SOAP-ENV:Body>
    <ns2:AlertRuleQueryResponse xmlns:ns2='http://sap.com/xi/BASIS'>
      <AlertRule>
        <AlertRuleID>AlertRuleID</AlertRuleID>
        <AlertRuleTechnicalID>dff69b116cba3bee900f564757a3c339</AlertRuleTe
        <Description languageCode=' '></Description>
        <AdministrativeData>
          <ResponsibleUserAccountID>User</ResponsibleUserAccountID>
          <LastChangeUserAccountID>User</LastChangeUserAccountID>
          <LastChangeDateTime>2022-07-12T07:52:04.603Z</LastChangeDat
        </AdministrativeData>
        <Severity>LOW</Severity>
        <State>ENABLED</State>
        <PayloadInAlert>DISABLED</PayloadInAlert>
        <AlertConsumers></AlertConsumers>
      </AlertRule>
      <AlertRule>
        <AlertRuleID>AlertRuleID</AlertRuleID>
        <AlertRuleTechnicalID>cc6dae12dc7d363b8d351651a46804a3</AlertRuleTe
        <Description languageCode=' '></Description>
        <AdministrativeData>
          <ResponsibleUserAccountID>User</ResponsibleUserAccountID>
          <LastChangeUserAccountID>User</LastChangeUserAccountID>
          <LastChangeDateTime>2021-11-10T16:22:27.602Z</LastChangeDat
        </AdministrativeData>
        <Severity>LOW</Severity>
        <State>ENABLED</State>
        <PayloadInAlert>ENABLED</PayloadInAlert>
        <AlertConsumers>RS-ALERT</AlertConsumers>
      </AlertRule>
    </ns2:AlertRuleQueryResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

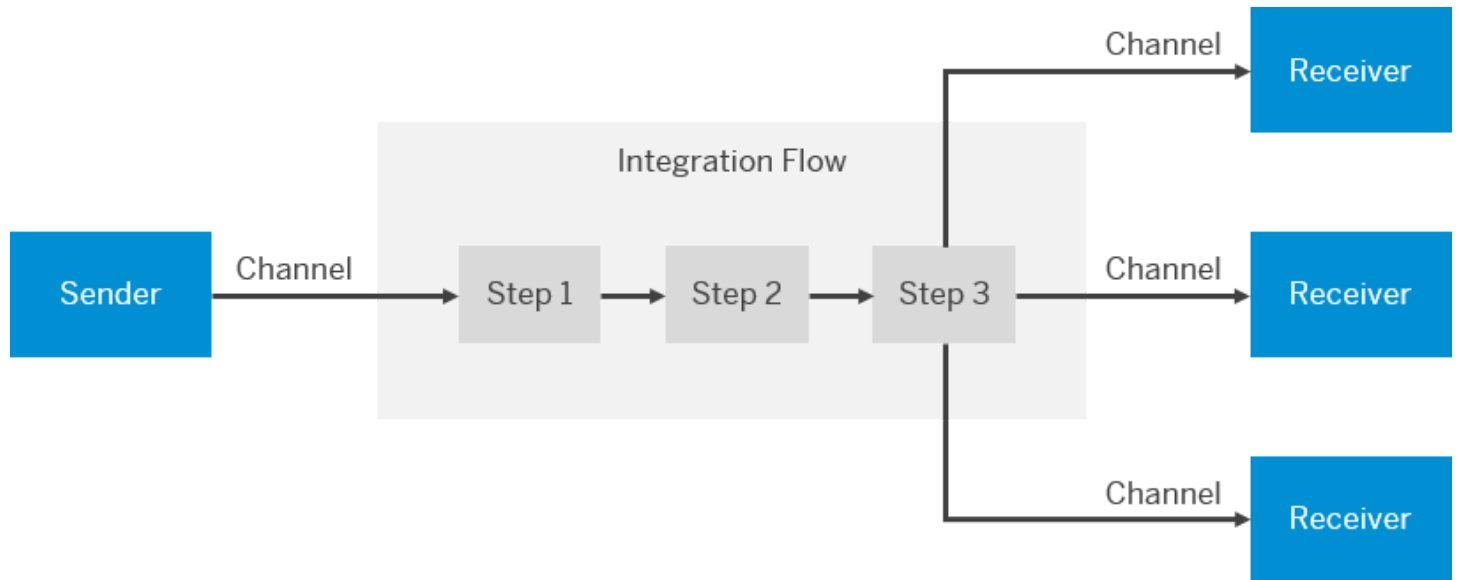
## Interface Design and Implementation

In a migration project, you may need to redesign a particular integration flow or develop one from scratch in an implementation project.

The Cloud Integration web UI provides a modeling environment that allows you to design the details of message processing (its senders and receivers as well as the individual processing steps) with a graphical user interface. For a comprehensive description of the main aspects when starting the development of a new interface, see [Getting Started with Integration Flow Development](#).

In general, an integration flow contains the following aspects:

- The senders and receivers of the message
- How the senders and receivers are connected to the tenant (adapters)
- The steps that define the message processing



Migration Tooling, a feature in the Cloud Integration capability, can help you migrate your integration artifacts from SAP Process Orchestration to SAP Integration Suite. See [Migration Tooling](#).

## Related Information

[Guidelines to Design Integration Flows](#)

[Integration Patterns](#)

[Additional References for Interface Development](#)

## Guidelines to Design Integration Flows

To start to develop an integration flow, the developer needs to make sure that the integration flows design is robust to avoid critical business processes for the project and for the company.

The integration flow designed should avoid errors and shouldn't break which could lead to errors in a critical business process. The integration flows must be kept readable and easy to understand, which is why the integration developer needs to know some characteristics to design an enterprise-grade integration flow. The following table lists some of the main aspects that need to be followed while developing integration flows:

Guidelines	Description
<a href="#">High Availability</a>	Carefully build an integration flow considering certain patterns that never break the business process.
<a href="#">Resource Management</a>	To correctly process messages in Cloud Integration, some resources are needed, like the capabilities used in the integration flow, on the computing system, database, and messaging service broker to process the messages.
<a href="#">Readability</a>	When an integration developer is designing an integration flow, it's important take the best practices and the readability of an integration flow into account. During the maintenance, it's easier and simple to quickly check the message processing flow if the integration flow is well designed.
<a href="#">Handling Failures Gracefully</a>	Even well-designed integration flows can break during message processing, but this shouldn't disrupt the business process. To avoid this situation or increase the impact of the failure, it's important to extend the integration flow with error handling. It's convenient to have supervision during the integration flow execution to check the message flow, keep the data to test, and analyze the causes of the error.

Guidelines	Description
<a href="#">Prepackaged Integration Content</a>	<p>Cloud Integration has prepackaged integration content available that contains integration flows, value mappings, documentation, and more. This is useful for the integration developer since these packages are a quick start into an integration project. Cloud Integration also allows to extend the capabilities of prepackaged integration content provided by SAP. In this case, the integration developer can design custom integration flows without changing the content.</p>
<a href="#">Highest Security Standards</a>	<p>The integration flow ultimately specifies how a message is processed and which options are applied to protect the content of the messages on the way between the sender and the receiver. Therefore, the integration developer has to make sure that the integration flows are designed in a way that ensures that the highest security standards are applied.</p>
<a href="#">Use Scripting Appropriately</a>	<p>You can use the Script step to apply script operations on the message content. The Script step type supports the following script languages:</p> <ul style="list-style-type: none"> <li>• Groovy</li> <li>• JavaScript</li> </ul> <p>If you don't consider the proposed guidelines during integration design, there's the risk that your scenario is affected in the following way:</p> <ul style="list-style-type: none"> <li>• The scenario runs into an out-of-memory situation.</li> <li>• It's hard for the integration expert to understand the logic of the integration flow.</li> <li>• Your scenario runs into compatibility issues if there are Java upgrades or Cloud Integration updates.</li> </ul>
<a href="#">Use the Partner Directory Appropriately</a>	<p>When designing business-to-business scenarios, you can use the Partner Directory to store information on business partners that are connected to the tenant in the context of a larger business network.</p> <p>The Partner Directory contains information on partners that are connected to a tenant in the context of a larger business partner network. The information stored in the Partner Directory can be used to parameterize integration flows.</p> <p>There's no dedicated user interface to access the Partner Directory. You can access its content only based on APIs.</p>

## Integration Patterns

Gregor Hohpe and Bobby Woolf have detailed a catalog of 65 common patterns in Enterprise Application Integration (EAI) scenarios in the book titled *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions* (Addison-Wesley, 2003). SAP published a blog about implementing [Integration Patterns on SAP Process Orchestration](#).

An **Integration Flow Design Guidelines – Enterprise Integration Patterns** package is available on the [SAP Business Accelerator Hub](#), where you can find more information on how to configure and use different integration patterns. It also contains sample integration flows, which can be used to test integration patterns on a Cloud Integration tenant or on SAP Process Orchestration 7.5.

The next chapters describe the following patterns and how to implement them in SAP Process Orchestration and Cloud Integration:

- [Aggregator](#)
- [Splitter](#)

- [Scatter-Gather](#)
- [Composed Message Processor](#)
- [Content-Based Routing](#)
- [Recipient List](#)
- [Message Filter](#)
- [Content Filter](#)
- [Content Enricher](#)

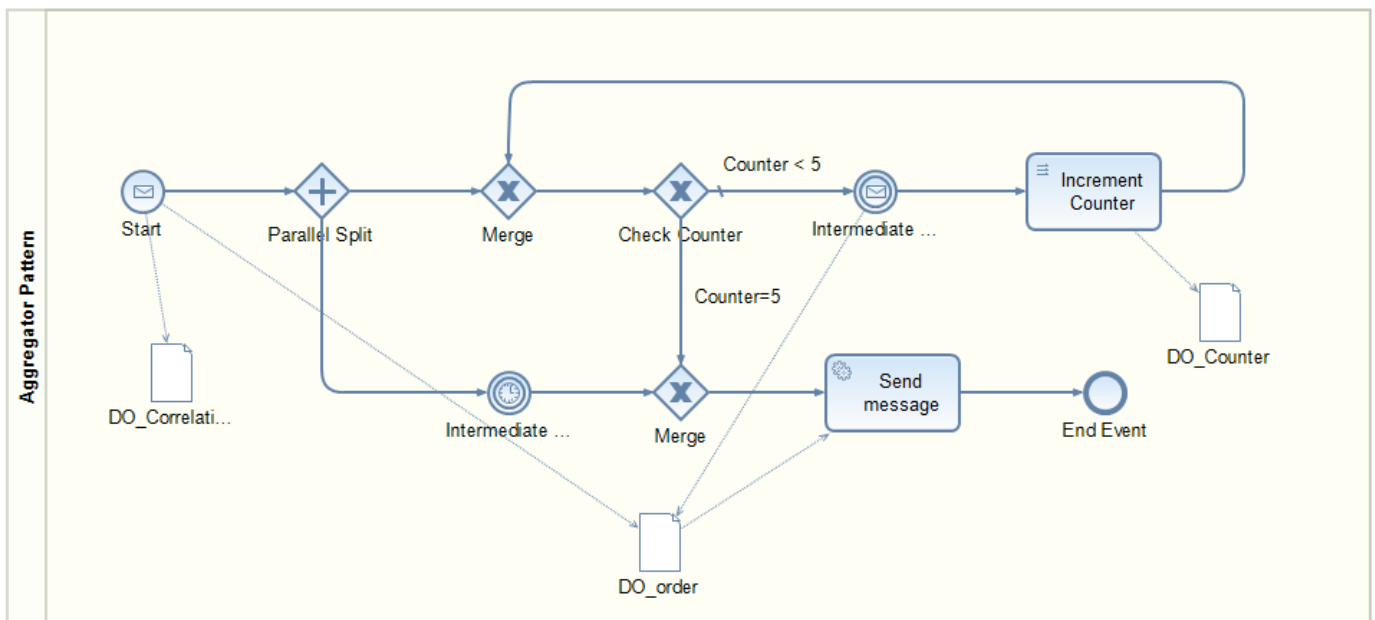
## Aggregator

The [Aggregator](#) pattern is a special filter that receives a stream of messages and identifies messages that are correlated. Once a complete set of messages has been received, the Aggregator collects information from each correlated message and publishes a single, aggregated message to the output channel for further processing.

The following sections describe the usage of this stateful pattern in SAP Process Orchestration and Cloud Integration.

### SAP Process Orchestration

Since the Aggregator pattern keeps a state, Business Process Management (BPM) is required to orchestrate the message flow. For the actual exchange of messages with the involved systems, the Process Integration runtime of SAP Process Orchestration is used. In the following, the focus is on the BPM process only. For a detailed description of the overall implementation, see the blog [Enterprise Patterns in Process Orchestration – Aggregator](#).



In this example, the process collects messages within a specific period, and each bulk should have 5 items at most. This model is a combination of an intermediate timer event and a loop. The incoming messages meeting the correlation condition are collected. An incoming message triggers the process instance. In the Start event, the counter is reset and the correlation condition is stored. A parallel split gateway is used to branch into the intermediate timer event path and the path containing the loop. The loop waits on further incoming messages with the same correlation condition as the start event. If a message arrives, the respective data object is appended, and the counter is incremented. Depending on which event occurs first, either the maximum number of messages or the waiting time has been exceeded, the aggregated message is sent, and the process

instance is ended. Optionally, the model can be enhanced to wait for a dedicated message stopping the collection (not shown here).

## Cloud Integration

For a detailed description of the Aggregator pattern on Cloud Integration, see [Aggregator](#).

The integration flow model presented in the help documentation illustrates the use of a dedicated Aggregator flow step. It's not required to model the pattern in a loop as seen for SAP Process Orchestration. In the Aggregator flow step, the correlation condition, as well as the completion conditions, can be defined. Right now, the last message condition and the completion timeout are supported in this model step. Unfortunately, maximum number of messages isn't supported yet. The rest of the flow steps are used to map the collection of items to the right message format. In this case, an order with order header information and all items, hence removing redundant information.

## Splitter

The [Splitter](#) pattern is used when it's required to split up a bulk or composite message into individual messages. For the following example, there's an order document with multiple lines. The message needs to be split into individual messages each containing the data related to one item. Furthermore, it must be ensured that the order header information is available in every message. You can find the XML files from before and after transformation in the [Splitter](#) help documentation.

The following sections describe the usage of this stateless pattern in SAP Process Orchestration and Cloud Integration.

## SAP Process Orchestration

On SAP Process Orchestration, there's the possibility to use either Business Process Management (BPM) or implement the split with a message mapping.

An example of the first option via BPM is described in the blog [Enterprise Patterns in Process Orchestration – Splitter](#). The BPM process is picked if further message orchestration steps are required.

This document illustrates the message mapping option since this option is more often used compared to the BPM process. In the message mapping, the occurrence of the source message type is set to 1, and for the target message type, it's unbound.

The screenshot shows the 'Message Mapping Overview' for the message mapping 'MM\_PurchaseOrder\_2\_Item\_Split'. The interface is divided into several sections:

- General:** Provides general information about the message mapping. Fields include:
  - Name: MM\_PurchaseOrder\_2\_Item\_Split
  - Description:
  - Namespace: http://demo.sap.com/eip/splitter
  - Software Component Version: EIP\_SWCV, 1.0 of demo.sap.com
  - Person Responsible:
- Documentation:** Includes links for 'Edit Documentation' and 'View Documentation'.
- Signature:** Selects the source and target messages.
  - Source Messages:** Software Component Version: EIP\_SWCV, 1.0 of demo.sap.com. Table:
 

Name	Namespace	Occurren...	
PurchaseOrder	http://demo.sap.com/eip/splitter	0..1	Add, Import..., Remove
  - Target Messages:** Software Component Version: EIP\_SWCV, 1.0 of demo.sap.com. Table:
 

Name	Namespace	Occurren...	
PurchaseOrder	http://demo.sap.com/eip/splitter	0..Unbound	Add, Import..., Remove
- Parameters:** A section for defining parameters, currently collapsed.

For each item within the source structure, a new message should be created. Every item node is mapped to the PurchaseOrder node.

MM\_PurchaseOrder\_2\_Item\_Split

### Message Mapping Definition

ns1:Messages

- ns1:Message1
  - ns0:PurchaseOrder
    - PurchaseOrderNumber
    - OrderDate
    - Address
    - DeliveryNotes
    - Items
      - Item
        - ItemNumber
        - ProductId
        - ProductName
        - Category
        - Quantity
        - CurrencyCode
        - Price

ns1:Messages

- ns1:Message1
  - ns0:PurchaseOrder
    - PurchaseOrderNumber
    - OrderDate
    - Address
    - DeliveryNotes
    - Items
      - Item
        - ItemNumber
        - ProductId
        - ProductName
        - Category
        - Quantity
        - CurrencyCode
        - Price

Overview | Definition | Dependencies

Properties | My Subscriptions | Problems | History | ESR Search Result | Where-Used Result | Transport Log | Where-Used Result

Expression

Item → PurchaseOr...

Enter Filter Text

Functions

The order header information such as the order number should be added to each individual target message. So, this information needs to be replicated. This can be achieved by using the CopyValue standard message function.

MM\_PurchaseOrder\_2\_Item\_Split

### Message Mapping Definition

ns1:Messages

- ns1:Message1
  - ns0:PurchaseOrder
    - PurchaseOrderNumber
    - OrderDate
    - Address
    - DeliveryNotes
    - Items
      - Item
        - ItemNumber
        - ProductId
        - ProductName
        - Category
        - Quantity
        - CurrencyCode
        - Price

ns1:Messages

- ns1:Message1
  - ns0:PurchaseOrder
    - PurchaseOrderNumber
    - OrderDate
    - Address
    - DeliveryNotes
    - Items
      - Item
        - ItemNumber
        - ProductId
        - ProductName
        - Category
        - Quantity
        - CurrencyCode
        - Price

Overview | Definition | Dependencies

Properties | My Subscriptions | Problems | History | ESR Search Result | Where-Used Result | Transport Log | Where-Used Result

Expression

@Purchase... → CopyValue → @Purchase...

Enter Filter Text

Functions

The item node and all child nodes can be mapped one by one. For the item, it's required to insert a context change to ensure that each message contains one item only. Here, the SplitByValue standard message function is used.

The screenshot displays the SAP Message Mapping Definition interface. It shows two message structures side-by-side. The left structure is 'ns1:Messages' containing 'ns1:Message1', which includes 'ns0:PurchaseOrder' and an 'Items' list. The right structure is 'ns1:Messages' containing 'ns1:Message1', which includes 'ns0:PurchaseOrder' and an 'Items' list. A 'SplitByValue' function is applied to the 'Item' node in the right structure. Below the mapping, a flow diagram shows 'Item' -> 'SplitByValue' -> 'Item'. The bottom right shows a 'Functions' panel with 'Enter Filter Text' and 'Functions' buttons.

## Cloud Integration

On Cloud Integration, depending on the use case, several options are available. In the Splitter documentation, two use cases are illustrated:

- Splitting a bulk order message into multiple orders
- Splitting a single order with multiple items

For the first use case, an Iterating Splitter flow step can be used. For a detailed description, see the [Variant with Iterating Splitter](#) documentation.

For the second use case, we describe two options: either reusing the [Message Mapping](#) or using a [General Splitter](#) flow step.

When the General Splitter flow step is added to the integration process, it takes care of splitting up the message into individual messages based on the items. On the processing tab of the General Splitter configuration, the XPath is selected as expression type and an XPath expression is maintained. The XPath expression is pointing to the location of the item node. The benefit of this option is that the General Splitter automatically duplicates the header information of the order for each individual message.

For more information, see the blog [SAP Cloud Platform Integration – Splitter](#).

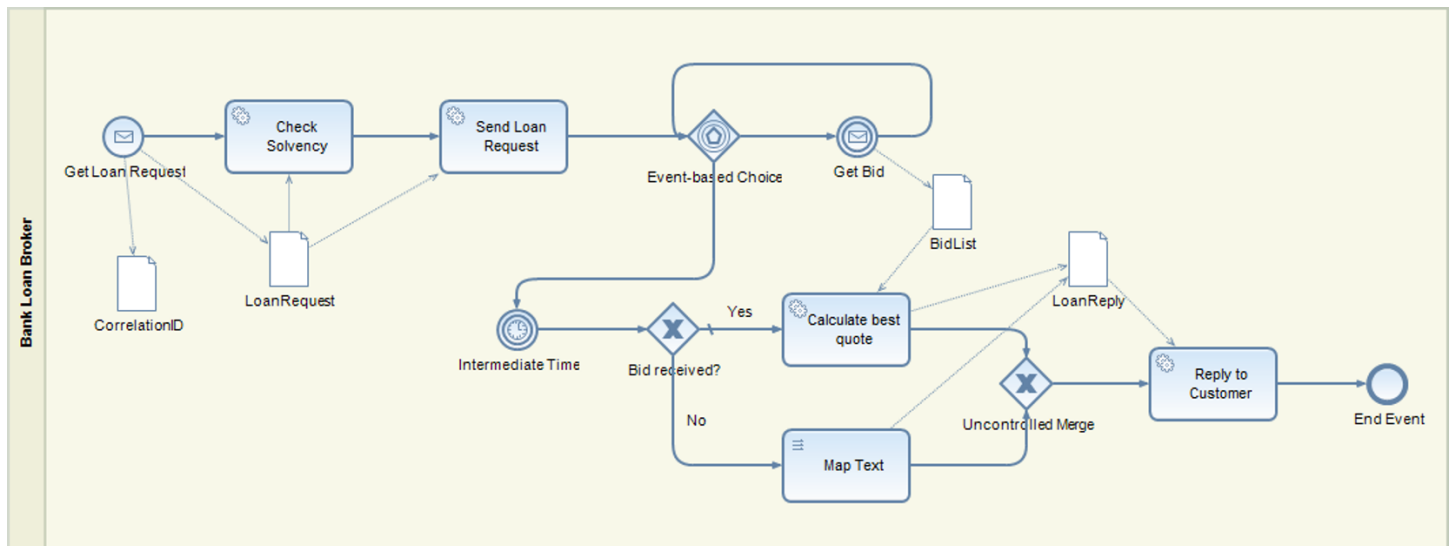
## Scatter-Gather

The [Scatter-Gather](#) pattern is used when it's required to send a message to multiple recipients and collect the individual responses in an aggregated message.

## SAP Process Orchestration

In the following, the process model of a typical bidding process that is implemented in SAP Process Orchestration are explained briefly. The process itself is modeled using BPM, and the message exchange with the various systems is done using SAP Process Integration as part of the SAP Process Orchestration solution bundle. In this document, the BPM process only is illustrated. For a detailed description of the overall implementation, see the blog [Enterprise Patterns in Process Orchestration - Scatter-Gather](#).

The process model starts with a message start event. For a new loan request, a new process instance is created. After having carried out a solvency check, the request is broadcast to multiple banks. Here, this is a service call, the distribution of the request to the banks is done within SAP Process Integration (not shown here). The bidding process should be open for a limited time, for example 2 days. Within this time frame, all bids from the banks are collected. Once the time has been exceeded, the bidding is closed, so no further bids are accepted. This is implemented via an event-based choice gateway with two branches. One branch contains an intermediate message event with a correlation condition so that the bids are assigned to the right process instance. The other branch contains an intermediate timer event, which is triggered once the time is up. In this case, the best quote is calculated, and a response is sent to the original requester. If a bid arrives after the time has been exceeded, the message is discarded because there's no active intermediate message event that meets the correlation condition.



## Cloud Integration

For a detailed description of the Scatter-Gather pattern on Cloud Integration, see [Scatter-Gather](#). In the following, only the rough model is outlined.

Since an intermediate message event is available on Cloud Integration, the Scatter and the Gather parts are modeled as separate integration processes. This is not a disadvantage, as the distribution and the collection don't have to be modeled in the same process. If the information from one part is used in the other, an exchange via a data store is modeled. Furthermore, to be able to correlate both parts in the message monitor, a common Application ID based on the request ID for instance can be defined.

The **Scatter** part takes the request and broadcasts it to multiple banks. Before, the timer is triggered via a separate message sent to the Gather part.

The **Gather** part collects the bids using an aggregator flow step, calculates the best quote, and sends the response to the original requester. The best quote calculation is done via a message mapping that can be imported from the Enterprise Service Repository (ESR) of SAP Process Orchestration.



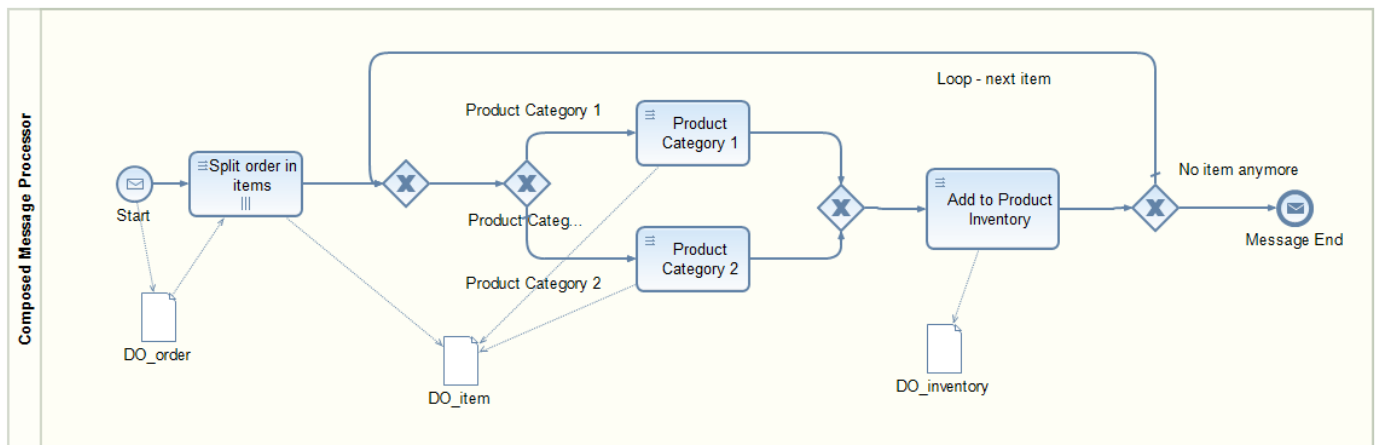
By default, if a bid arrives after the time has been expired, a new process instance is created. To avoid this, a third integration process that checks if the bidding process is still active and eventually forwards the bids to the Gather part must be added. The check is based on a respective entry in the data store. Once the request has been received, the request is stored in the data store with the request ID as the key. This is done in the Scatter part, as described before. Once the time has been exceeded, the data store entry is deleted. This is done in the Gather part, as described before. In the model below, it's checked if the data store entry still exists. If not, the exception subprocess is triggered, and the message processing is stopped, hence discarding the message. Otherwise, the message is sent to the Gather part.

## Composed Message Processor

The [Composed Message Processor](#) pattern describes how to handle a message with multiple elements, but each element requires different processing. You use a Composed Message Processor pattern to split the message into multiple submessages, route the split messages to different destinations, and then reaggregate the responses back into one single message.

## SAP Process Orchestration

The process model of the Composed Message Processor pattern on SAP Process Orchestration is explained in more detail. The process itself is modeled using Business Process Management (BPM), and the message exchange with the various systems is done using the Process Integration runtime of SAP Process Orchestration. This document focuses on the BPM process only. For a detailed description of the overall implementation, see the blog [Enterprise Patterns in Process Orchestration - Composed Message Processor](#).



The BPM process is modeled following the BPMN (Business Process Model & Notation) specification. The process model starts with a message start event receiving an order with multiple items. In a mapping, the order is split. For the mapping, looping is defined as part for each although the individual split messages actually run in sequence. For each item, a condition is carried out, and the processing is routed to either of the branches depending on the respective product category. A router condition checks if all items have been processed. If not, the processing is looped back, and the next item is processed. Otherwise, the aggregated message is returned.

## Cloud Integration

For a detailed description of the Composed Message Processor pattern on Cloud Integration, see [Composed Message Processor](#). In the following, only the rough model is outlined.

In the sample integration flow presented in the help documentation, a General Splitter is used to split an order into multiple individual messages according to the number of items. In the General Splitter property, the Parallel Processing check box is selected to ensure concurrent processing of the split messages. The Router forwards the items to different inventory systems

depending on the product category where the items are enriched. Finally, the Gather step reaggregates the multiple messages into a single message.

## Content-Based Routing

The [Content-Based Routing](#) pattern is used to route a message to the correct recipient based on its content. The message is routed to exactly one channel.

## SAP Process Orchestration

In the integration flow, you can add a recipient list and maintain a list of potential receivers. You can select one of the following routing techniques:

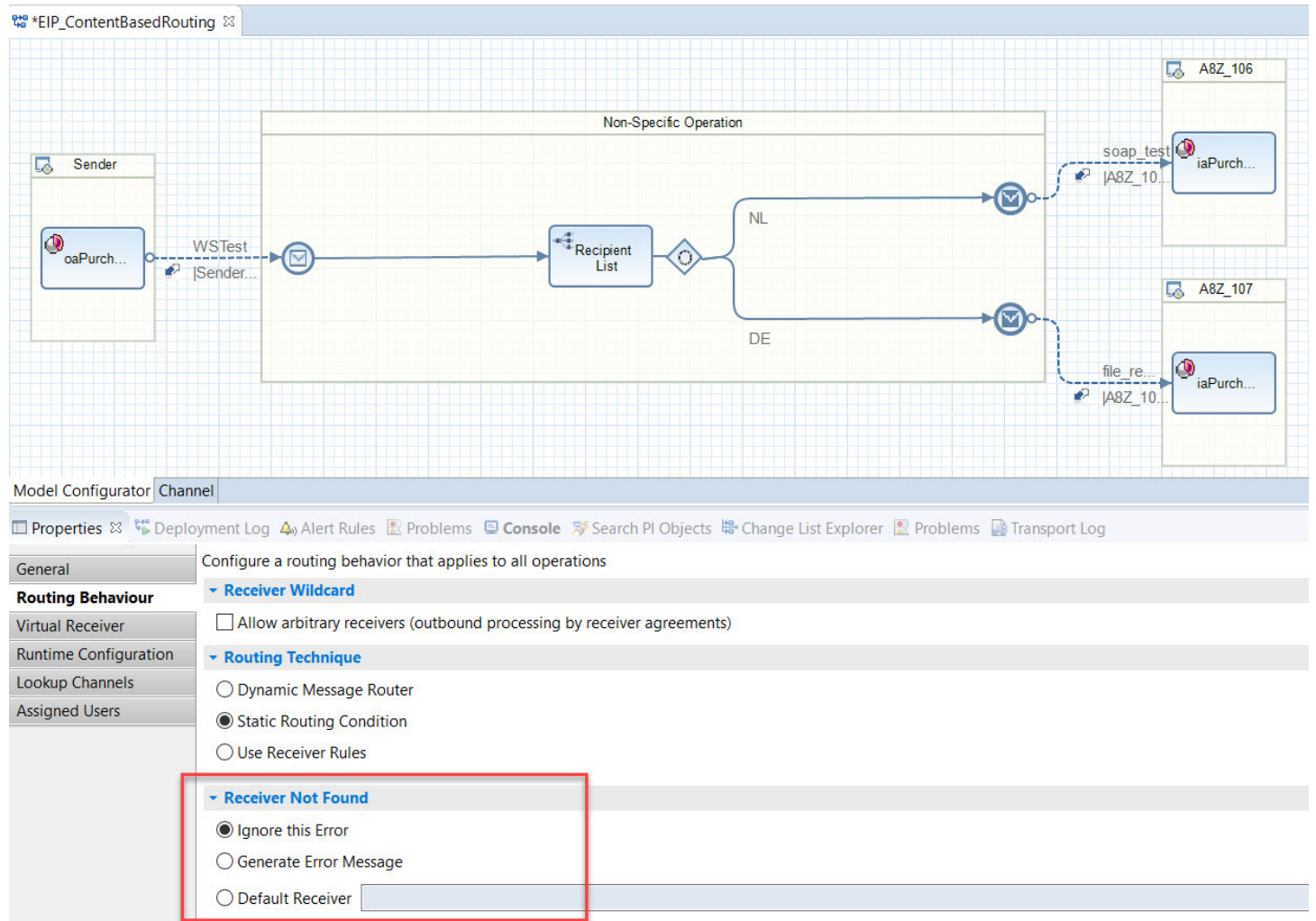
- dynamic (via message mapping),
- static (an XPath condition either based on the payload data or the message header), and
- receiver rules (reusable routing conditions).

SAP Process Orchestration doesn't distinguish between the Content-based Router and the Recipient List patterns. While the former message is routed to exactly one receiver, the latter can broadcast the message to multiple receivers. However, if used routing conditions are disjoint, each message is routed to exactly one receiver strictly following the definition of the Content-based Router. In the example, a static routing condition is used.

Here, the focus is on the Receiver Not Found configuration options. During message processing, the condition is evaluated, and if met, the message is routed to the respective receiver. If no receiver could be determined, various options to proceed are available:

- raise an error,
- send to a default receiver, or
- ignore the message.

In the following example, the latter option is used. The message is discarded without raising an error if neither condition is met.



## Cloud Integration

The Content-based Router pattern is modeled using a router flow step. Like for SAP Process Orchestration, the XPath condition can be either based on the payload data or the message header. For each of the Receiver Not Found options described above, SAP has shipped an integration flow variant. For a detailed description of all three variants of the Content-based Router pattern on Cloud Integration, see [Content-Based Routing](#). The figure in the documentation shows the variant where the message should be discarded if neither condition is met. For all variants, it's recommended to store the XPath expression as an Exchange property, which is then used in the routing conditions. This way, it's easier to trace the value of the XPath condition in case of errors. Furthermore, a better runtime behavior can be expected since the message needs to be parsed only once. For each receiver, the routing condition is configured accordingly. In addition, we define a default route that points to an End event.

## Recipient List

The [Recipient List](#) pattern is used to route a message to a list of recipients. It's similar to the Content-Based Routing pattern, but the receiver determination doesn't rely on the content of the message.

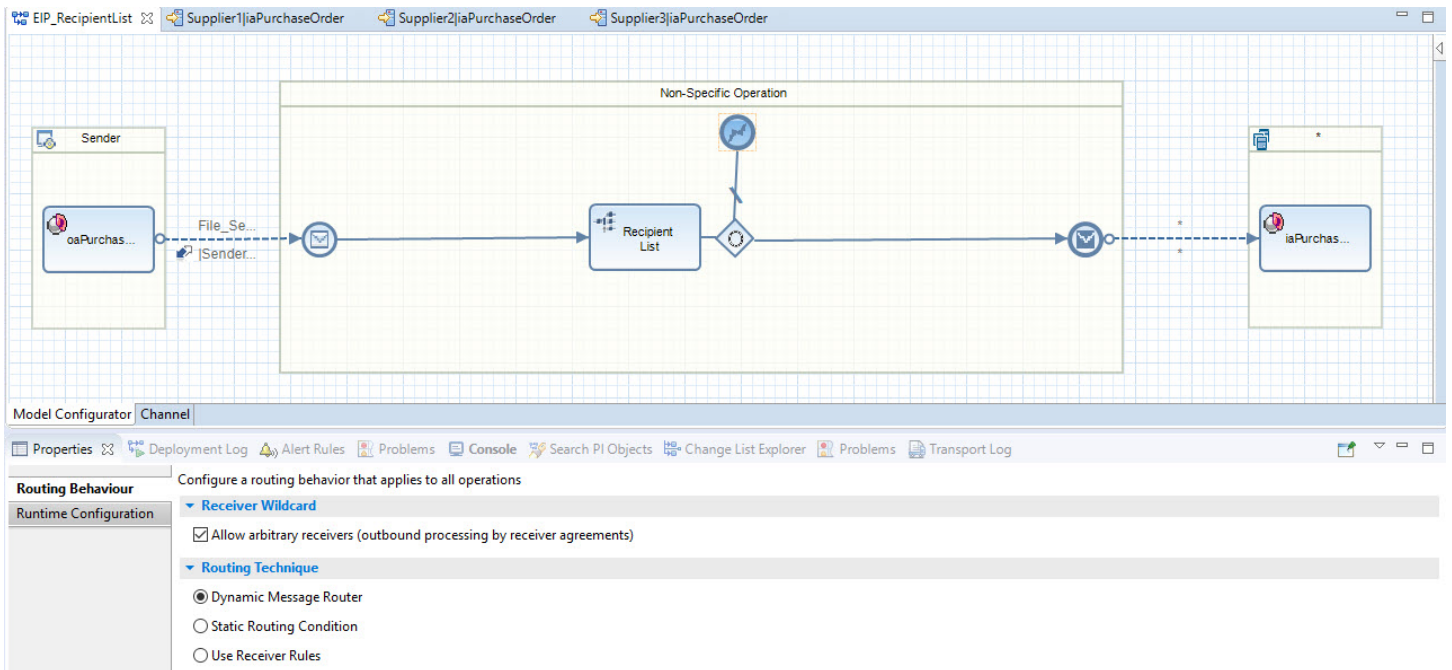
## SAP Process Orchestration

Similar to the Content-Based Routing, you model the Recipient List pattern using a Recipient List within the integration flow. When maintaining the list of potential receivers, you have the possibility to select one of the following routing techniques:

- Dynamic (via message mapping)
- Static (an Xpath condition either based on the payload data or the message header),
- Receiver rules (reusable routing conditions).

Other than for the Content-Based Router, here the routing conditions don't have to be disjoint, and hence the same message may be routed to multiple receivers.

In the following example, we stick to the dynamic routing condition in combination with the receiver wildcard. As you can see in the figure, we've chosen the **Dynamic Message Router** as routing technique and have selected the **Allow arbitrary receivers** flag.



On the **Dynamic Routing** tab, a message mapping is selected which is carried out to determine the receivers the message should be routed to.

The screenshot shows the 'Dynamic Routing' configuration in the SAP Integration Suite Model Configurator. The 'Lookup Channels' section is active, showing a message mapping named 'PurchaseOrderItem\_to\_Receivers\_OM' with namespace 'http://demo.sap.com/eip/recipient-list' and software component version 'EIP\_SWCV, 1.0 of demo.sap.com'. Below this, there is a table for parameters.

Name	Type	Value

For each potential receiver, we need to create a receiver agreement. This way you can add new receivers without the need of changing and redeploying the integration flow. When selecting the receiver in the integration flow model, in the tab **Receivers** you get a list of receiver agreements displayed which fit the particular integration flow.

The screenshot shows the 'Receivers' tab in the SAP Integration Suite Model Configurator. The table displays possible receiver agreements for the integration flow.

Receiver Agreement	Communication Channel	Adapter Type	Adapter Engine
Supplier1   iaPurchaseOrder	REST_Receiver	REST	Central Adapter Engine
Supplier2   iaPurchaseOrder	REST_Receiver	REST	Central Adapter Engine
Supplier3   iaPurchaseOrder	SOAP_Receiver	SOAP	Central Adapter Engine

## Cloud Integration

The Recipient List pattern is modeled using the same message mapping that we've used for SAP Process Orchestration before. The message mapping can be imported from SAP Process Orchestration. The message mapping returns a list of receivers,

which is then split using an Iterating splitter flow step. For each split item, the message is then routed to another integration flow via Process Direct adapter. The Process Direct address is dynamically determined based on the mapping outcome.

For each potential receiver, we create an own integration flow that would correspond to a receiver agreement in SAP Process Orchestration. By decoupling the main integration flow from the receiver-specific message delivery, we don't need to maintain a fixed number of receivers in the main integration flow. Furthermore, other than for the receiver wildcard in SAP Process Orchestration, each receiver can have a different receiver-specific mapping or other receiver-specific flow steps.

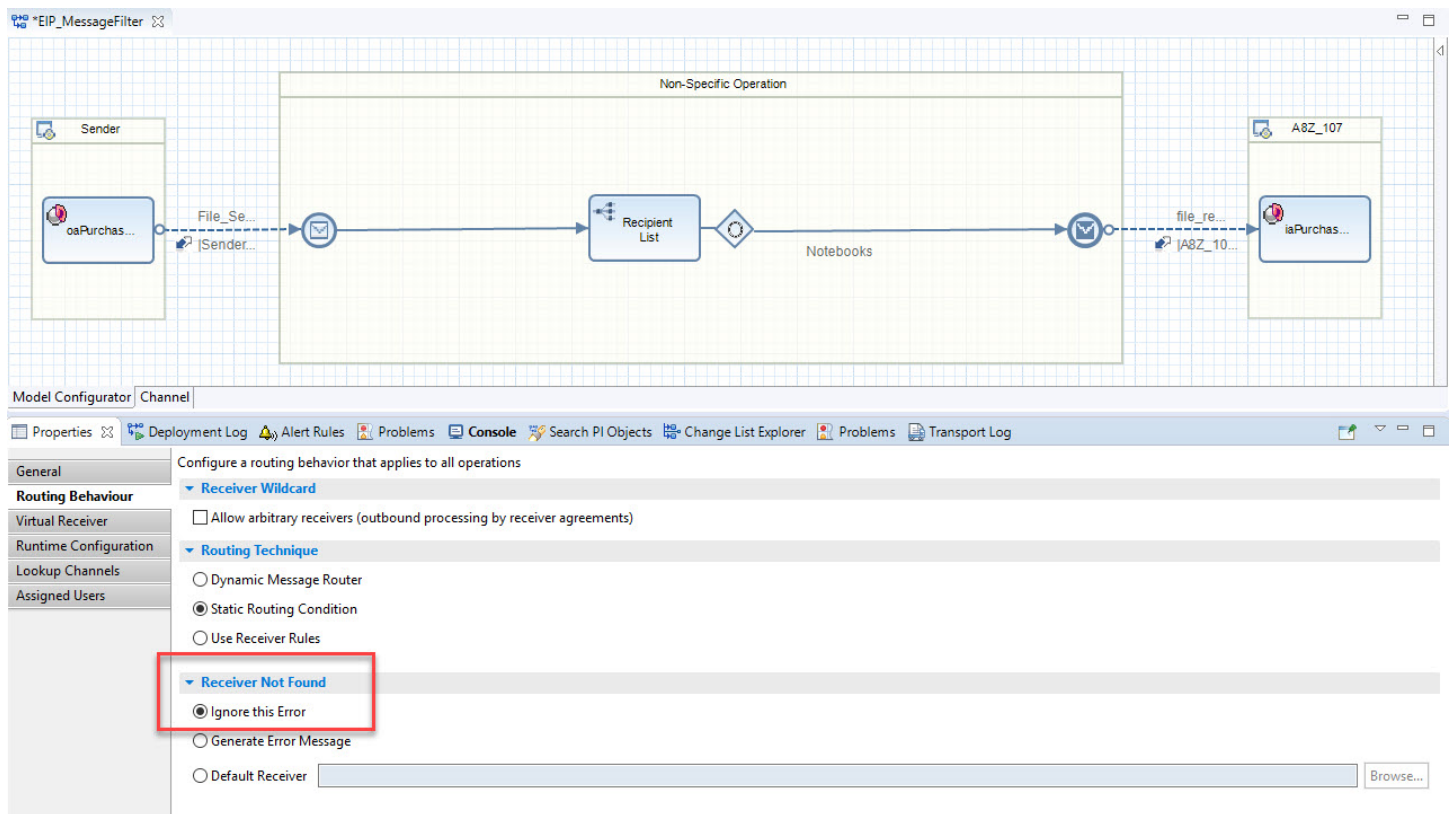
For a detailed description of the Recipient List pattern on Cloud Integration, see [Recipient List](#), which also describes another variant of the Recipient List pattern with static routing.

## Message Filter

The [Message Filter](#) pattern is similar to the [Content-Based Router pattern](#). For the Message Filter pattern however, you only have one single receiver. Any incoming message is evaluated, and if it meets the routing criteria specified by the filter, the message is routed to the receiver, or is discarded otherwise.

## SAP Process Orchestration

The Message Filter can be modeled similarly to the Content-Based Router. It corresponds to the Content-Based Router variant where the message is ignored if the receiver can't be determined. The only difference is that it only has one single receiver, see the following figure.



## Cloud Integration

On Cloud Integration, the Message Filter pattern is modeled using a router flow step with one route pointing to the actual receiver and a default route pointing to an End event. In our case, messages with the product category Notebooks are routed to the receiver. Otherwise, the messages are discarded. For a detailed description, see [Message Filter](#).

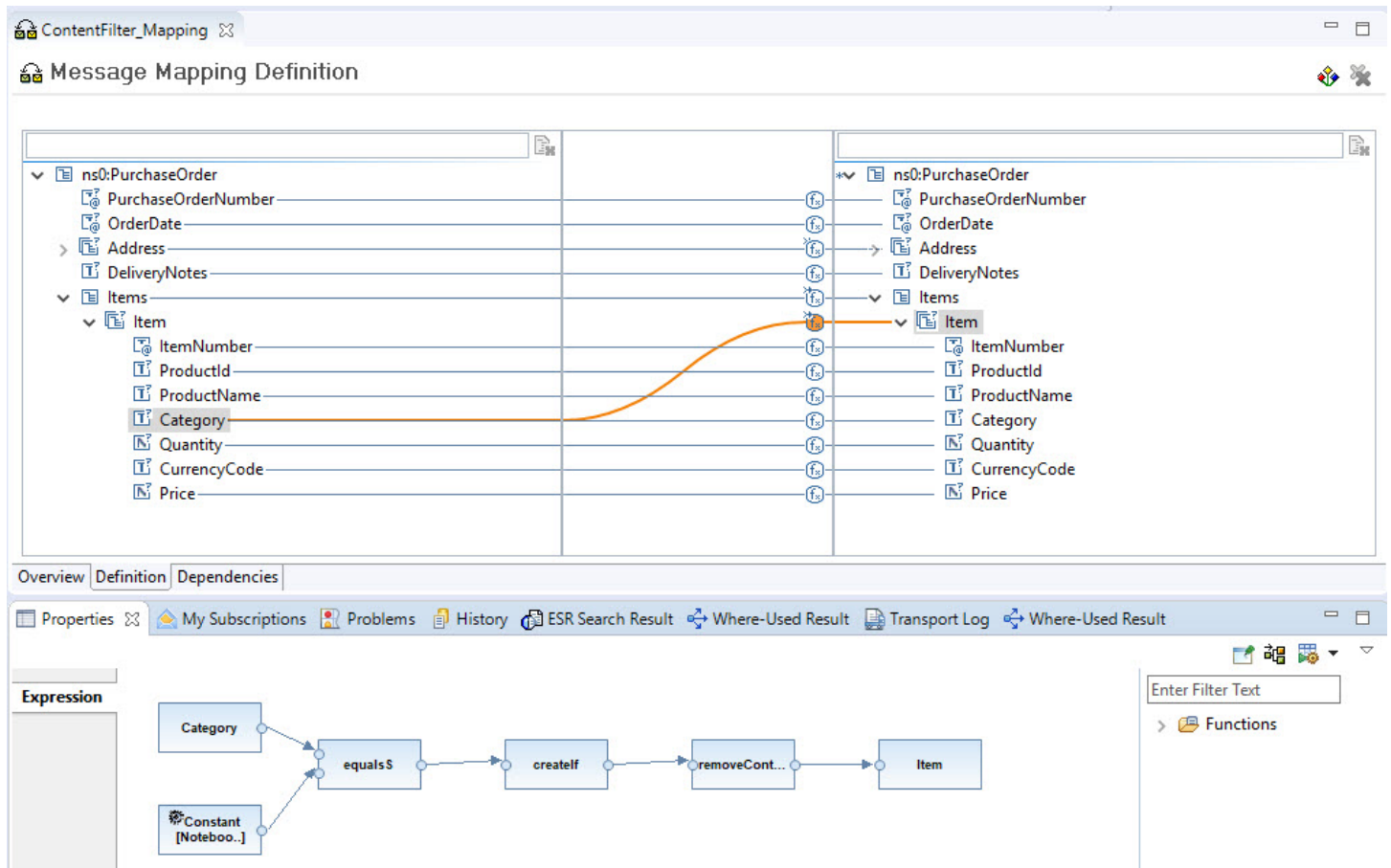
## Content Filter

The [Content Filter](#) pattern is used to remove data from a message that you don't need in your application system. You only keep the important information.

## SAP Process Orchestration

In the example, a purchase order contains multiple items of different product categories. The Content Filter pattern is implemented using a message mapping. In the message mapping, all items within the purchase order that aren't intended for a particular receiver are removed, so only those items of a specific product category are kept. In this example, those items are only the notebooks.

As illustrated in the following figure, all the header fields are mapped one by one. For the Item node, we check if the Category equals Notebooks. If so, we create the node, otherwise, the item is discarded. All fields below the Item node are mapped one by one.



## Cloud Integration

On Cloud Integration, there are two options:

- Use the dedicated Filter flow step.
- Reuse the message mapping from the earlier example.

Both options are described in [Content Filter](#). The latter option can be resolved by importing the message mapping from Process Orchestration or modeling it the same way as described in the earlier chapter. In this chapter, the focus is on the second option.

In the integration flow model in the documentation, you can see that we use a Filter flow step where we maintain the XPath expression. Here, we keep all items with category Software. Value type needs to be nodelist since the XPath expression may apply to multiple items. The filter removes the purchase order header. So, we need to save this part in an exchange property using a content modifier, and in another content modifier, we construct the message based on the item list and the stored header fields.

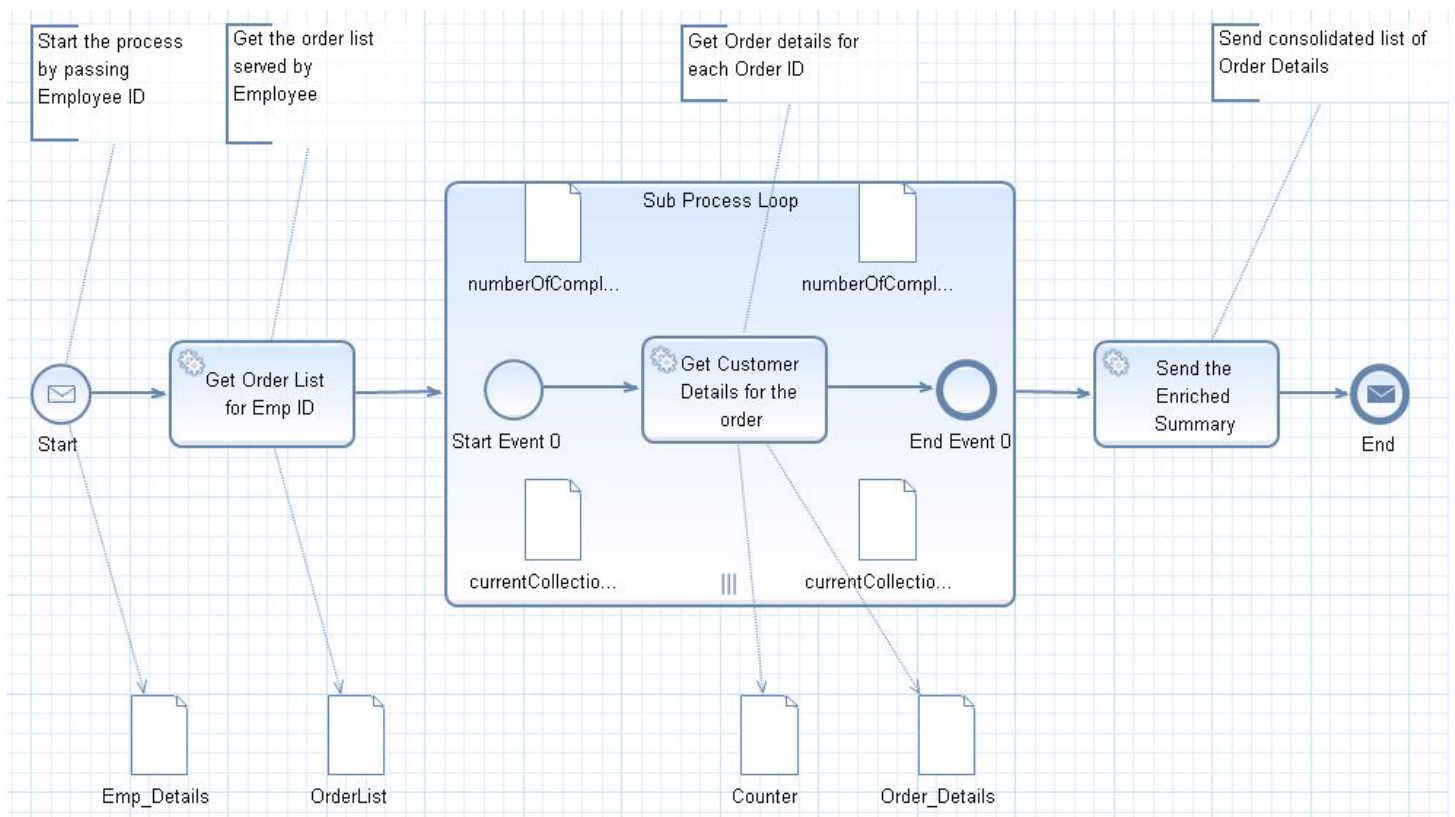
# Content Enricher

The [Content Enricher](#) pattern is used to access an external system in order to add missing data to a message. This process is also called look-up.

## SAP Process Orchestration

- Model the lookup within a Business Process Management (BPM) process
- Implement a lookup within a message mapping

The first option using a BPM process is described in the blog [Enterprise Patterns in Process Orchestration – Content Enricher On SAP Process Orchestration, there are two options:](#). In this example, you first gather an order list for a specific employee. Here, customer details are missing which you need to read from a different system. Therefore, you loop over the list of orders, and within each loop pass, you do a Web Service call using an automated activity. Within the automated activity, the additional information is then mapped to the respective order item, hence enriching the message.



For RFC and JDBC lookups, the message mapping tool within the Enterprise Services Repository (ESR) of SAP Process Orchestration supports standard functions to configure the lookups. Those functions use the message types stored in the ESR so that you can simply select the input and output fields when configuring the lookup.

## Cloud Integration

On Cloud Integration, you can model the Content Enricher:

- via the request-response pattern
- via a dedicated Content Enricher flow step

For RFC and JDBC lookups, then in the example described, a dedicated Content Enricher step was used. The benefit of the Content Enricher flow step is that it automatically matches the responses to the different items of the original message. Therefore, it's suited especially for enriching bulk data. Using this option is better in terms of performance compared to a request-response pattern. Here, you don't need to loop over all items carrying out individual lookup calls, but instead do one call, and the matching is done automatically based on your key.

In the example in the documentation, there's an order with multiple items, including product and product category information. We perform a lookup to gather the main category names (OData query not displayed), and automatically match them to the respective item based on the category as the key element.

For more information, see [Content Enricher](#).

## Additional References for Interface Development

The following table provides additional resources for developing interfaces.

Blog	Category	Description
<a href="#">SFTP Poll-Enrich</a>	Interface Development	How to utilize this feature to enhance the SFTP Features



Blog	Category	Description
<a href="#">Cloud Integration Developer Corner</a>	Interface Development	Updated list on new enhancement and guidelines from SAP
<a href="#">Asynchronous Messaging</a>	Interface Development	How to setup and configure asynchronous delivery on Cloud Integration
<a href="#">CI/CD for SAP Integration Suite</a>	Interface Development	Integration flows recipes
<a href="#">Guidelines to Migrate/Integrate Data Into Cloud from/to On-Premise Systems</a>	General	Comprehensive article discussing about Digital Transformation, SAP cloud solutions, and methodologies
<a href="#">B2B Capabilities</a>	B2B	Blog collections about B2B Integration
<a href="#">Partner Directory</a>	B2B	Blog about B2B Partner Directory