SAP Cloud Integration – An Overview, Best Practices, and Implementation Steps – Part 1

[1]Deepak Kumar, [2]Maha Bhageshwara Raju Kesaboina

[1]Wilmington, USA
[2]Hyderabad, INDIA
Email: Deepak3830@gmail.com, Bhageshwar.raju@gmail.com

Abstract SAP CPI, also known as SAP Cloud Platform Integration is a cloud-based integration platform provided by SAP. It allows for connectivity, between applications, systems, and services both within and outside of an organization. With SAP CPI businesses can simplify the integration process. Optimize data flow and communication between software solutions. It facilitates the exchange of data across environments promoting interoperability and enhancing collaboration efficiency. Additionally, SAP CPI offers built-in integration iFlow and adapters to reduce the complexity of integration projects and speed up deployment timelines. Overall SAP CPI plays a role, in the SAP ecosystem by enabling the creation of agile business processes. In this paper, we will discuss SAP CPI's capabilities in connecting diverse applications, facilitating data flow, supporting different integration scenarios, and leveraging pre-built content and adapters. It will provide a step-by-step guide or overview of the fundamental steps involved in implementing SAP CPI. This paper will cover the initial setup, configuration, and the process of establishing connections between different systems. The aim is to offer a practical understanding for readers who might be considering or undergoing the implementation process. This paper will also address the monitoring aspect of SAP CPI, emphasizing the importance of keeping track of integrations and workflows. Insights into monitoring tools, key performance indicators, and best practices for ensuring the ongoing efficiency and reliability of integrated systems will be elaborated.

Keywords SAP, SAP CPI, SAP Cloud Integration, SAP On-premice to SAP cloud integration

1. Introduction

SAP CPI – SAP Cloud Platform Integration is a powerful cloud-based platform that enables seamless integration between cloud applications and other SAP or non-SAP cloud or on-premises applications. SAP CPI uses Apache Camel in its integration framework. Apache Camel is an open-source framework that offers a variety of pre-built components and patterns for creating enterprise integration solutions. SAP CPI utilizes Apache Camel to make it easier to develop integration flows and to improve the platform's flexibility and extensibility.

Integration Flow (Iflow) – An iFlow is a graphical representation of the complete integration process within SAP CPI. It outlines the steps messages must take as they travel from a source system to a target system, including data transformations and mappings along the way.

2. SAP CPI Architecture

The SAP CPI architecture consists of the following components:

- Integration Engine: The Integration Engine is the main building block of SAP CPI. It is in charge of message processing and routing them to the proper adapters.

- Message Broker: Between the Integration Engine and the adapters, communications must be stored and routed through the Message Broker.
- Adapter: The Adapters are in charge of interacting with external systems. They can be used to connect to cloud-based systems, SAP systems, and systems that aren't SAP.
- Runtime Environment: Running the SAP CPI components is the responsibility of the runtime environment. It offers a running environment for the Integration Engine, Message Broker, and Adapters.
- Management Console: The SAP CPI architecture is managed using the Management Console, a web-based interface. It can be used to set up the architecture, configure the components, and diagnose issues.
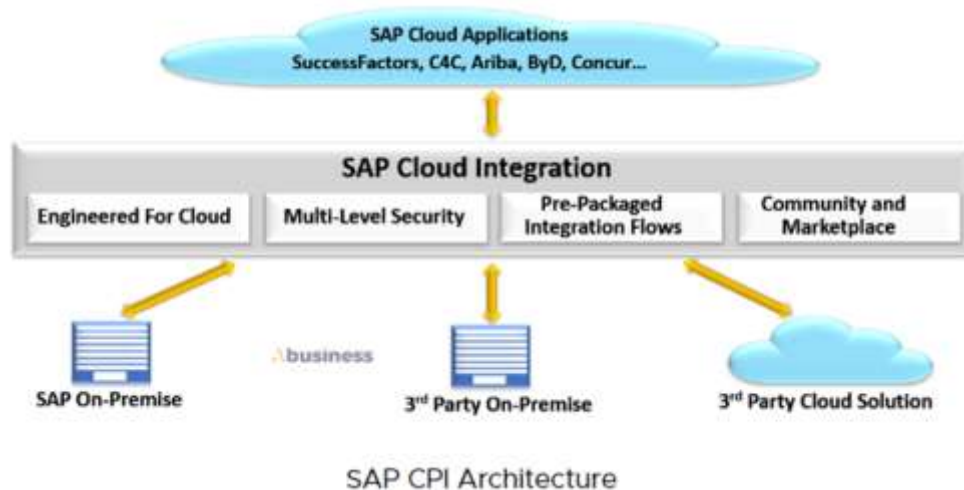

Figure 1: SAP CPI Architecture

3. SAP CPI Integration Capabilities

Here are some of the important integration capabilities of SAP CPI:

- Connectivity: SAP CPI offers pre-built adapters and connectors for easy connectivity with various systems, applications, and data sources, including on-premise systems, cloud applications, and third-party services.
- Integration Flows (Iflows): Users can create end-to-end integration processes using Integration Flows (Iflows). These visual representations define how messages are processed, transformed, and routed between source and target systems.
- Data Transformation and Mapping: SAP CPI provides users with data transformation and mapping tools that enable seamless data format conversion across integrated applications.
- Process Orchestration: SAP CPI automates complex processes across multiple systems, ensuring efficient and synchronized execution.
- Adapters and Protocols: The SAP CPI platform provides support for various adapters and protocols such as REST, SOAP, OData, JDBC, SFTP, and more. This enables easy integration with a diverse set of systems and technologies.
- Message Processing and Routing: SAP CPI enables advanced message processing and routing based on content, ensuring messages are directed to appropriate endpoints based on defined criteria.
- Security and Compliance: Security features are integrated into SAP CPI to ensure secure data transmission and compliance with industry regulations. This includes encryption, authentication, and authorization mechanisms.
- Monitoring: SAP CPI allows for advanced message processing and routing based on content, ensuring that messages are directed to appropriate endpoints based on defined criteria.

4. SAP CPI Overview

SAP CPI Home: The homepage of SAP CPI serves as a hub for important information and activities. It features a navigation menu that allows users to access different sections of the platform easily, such as design, monitoring, operations, and administration. Additionally, users will find an overview of key information and quick access to various functionalities such as recent items, quick links, system health and notifications, and an integration flow overview.
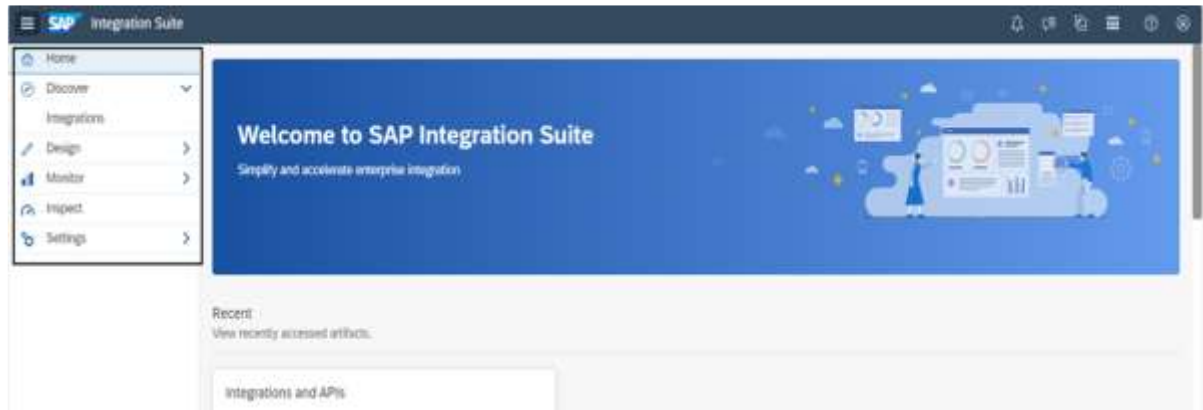

Figure 2: SAP CPI Home Screen

Discover: The Discover section of CPI offers several standard integration designs in pre-built packages by SAP, which require no further development. You can import them directly into your package and configure them to match the client's system details. However, there are two types of artifacts available, namely Configurable (which can't be edited but allows for the modification of sender/receiver endpoints) and Modifiable (which can be changed in any way). Using the search option users can search specific integrations as per the user requirements. The top of the page displays the total number of available integrations as of today's date. Below the search option, the total number of search items found is also displayed.

Click on the package to view integration details. On the next page, you will find a description, version, published date, mode (editable or non-editable), artifacts (Iflows), documents, and more for the selected package.
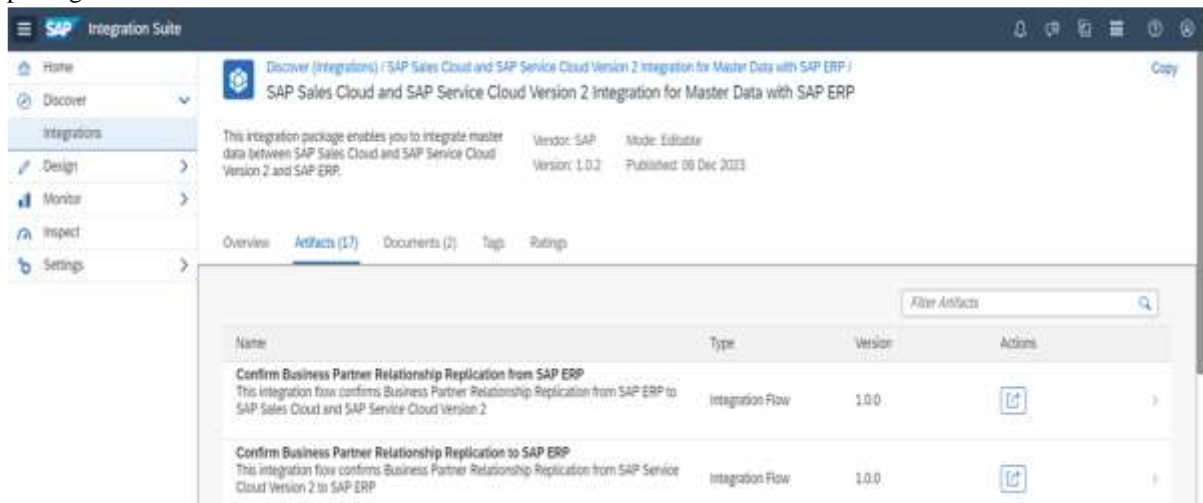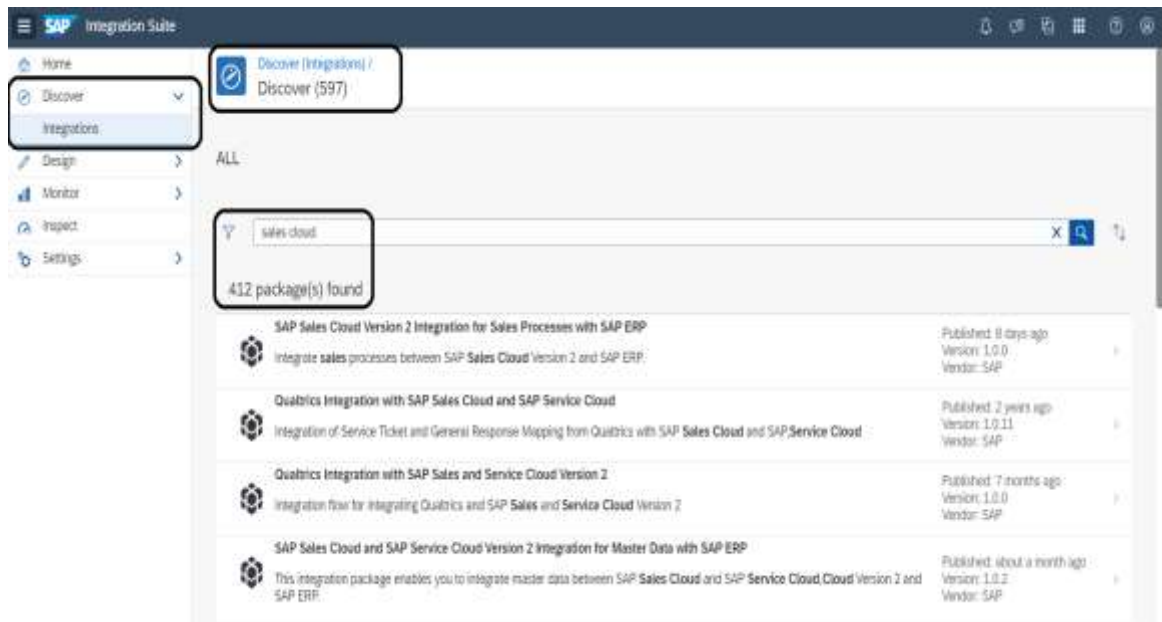

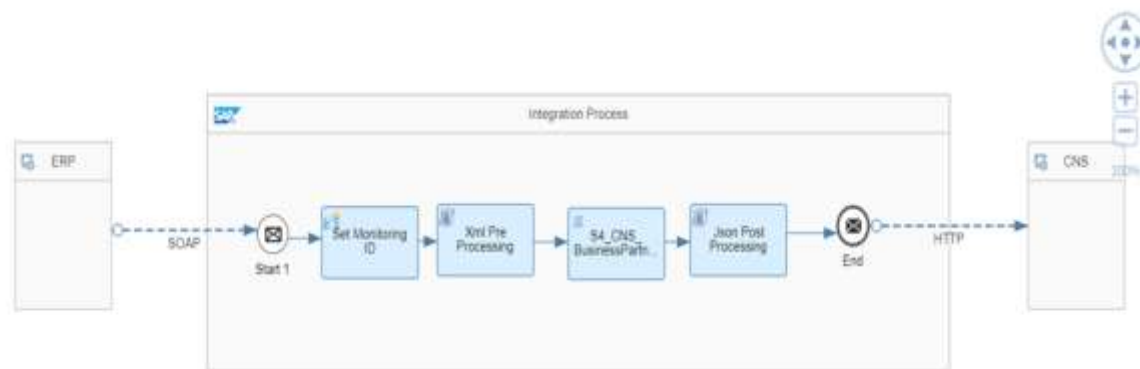*Figure 3: Package details*

Discover

Figure 4: Discover

Figure 5: Integration Flow

Click on the relevant Iflow to view further details.

Design – The CPI Design section provides the complete platform to develop your custom Integration flows (Iflows) inside the package shown. You can import packages and iflows directly to this platform if they have been downloaded from other CPI tenants. You have the option to create a new package and build an IFlow from scratch, or you can copy an existing package along with its required artifacts and customize them to suit your specific needs.

Figure 6: Design

Monitor - The CPI Monitor section offers all the necessary features to monitor both standard and custom IFlows. It also provides debugging capabilities in case of errors, management of credentials and certificates, runtime variables, and other essential functions. It may include summary charts, key performance indicators (KPIs), and visual representations of message flows. You can view the status of message processing in real-time, including information about successful and failed messages. This helps with the quick identification and resolution of integration problems. The monitor also provides access to detailed message logs, allowing you to inspect message content and context individually. This is crucial for troubleshooting and understanding the data flow within integration processes. You can visualize the flow of messages through Integration Flows using graphical representations, which helps in comprehending the processing sequence and identifying bottlenecks or issues. The monitor alerts you in real-time to errors or exceptions, and you can set up alerting mechanisms to receive notifications when specific conditions are met, such as the failure of critical integration processes.
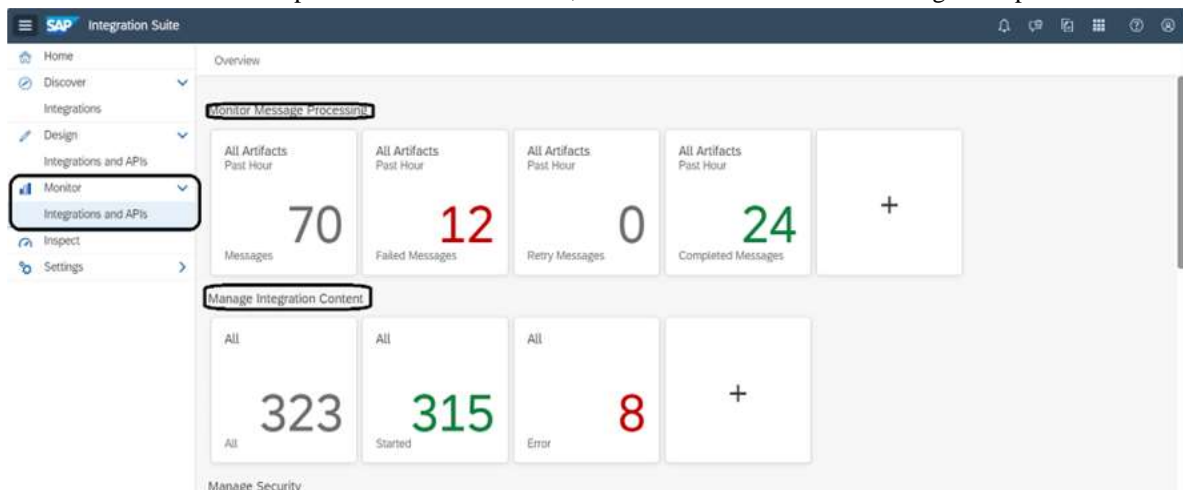

Figure 7: Monitor

Settings – The CPI Settings section provides tenant settings that aren't required in daily-based integration development.

5. Conclusion

The this discussion we have included an overview of the SAP CPI architecture, which involves understanding how different components within SAP CPI interact to facilitate seamless integration between various systems and applications. This paper cover daspects such as message processing, connectors, runtime, and the overall structure of the integration platform. This paper covers how SAP CPI enables users to connect different systems, orchestrate processes, and manage the flow of data. Integration capabilities may include pre-built connectors, data mapping, content-based routing, and support for various protocols. Also, we have provided a glimpse into the user interface of SAP CPI, including key screens such as the homepage, design tab, monitoring tab, and settings tab. This gives users an understanding of where they can perform different tasks related to designing, monitoring, and configuring integration processes. The upcoming sections will focus on implementation steps along with practices and a case study to offer valuable insights and guidance, for users working with SAP CPI.

Declarations

Ethics approval and consent to participate: Not Applicable

Consent for publication: All authors have consent to submit this paper to the Journal of Cloud Computing. Also, we confirm that this paper or any part of this paper was not submitted anywhere.

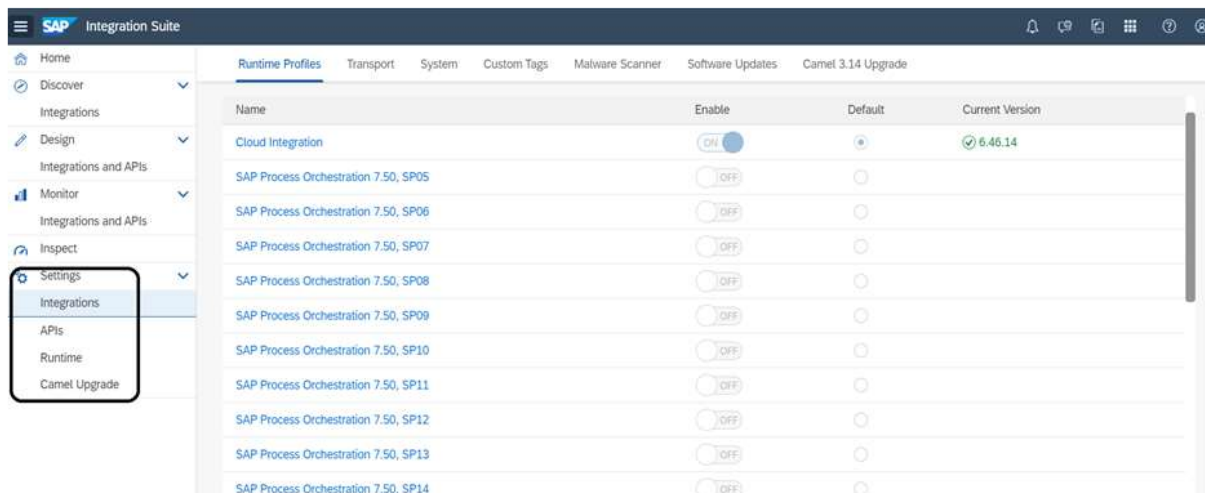**Availability of data and materials:** Not Applicable

Figure 8: Settings

References

[1]. "SAP Business Accelerator Hub," api.sap.com. https://api.sap.com/integrations/packages

[2]. Sookriti_Mishra, "My adventure in learning CPI: Part 1 | All about SAP Cloud.," SAP Community, Sep. 10, 2019. https://community.sap.com/t5/technology-blogs-by-members/my-adventure-in-learning-cpi-part-1-all-about-sap-cloud/ba-p/13393526

[3]. Sookriti_Mishra, "My adventure in learning CPI: Part 2 | Deployment Models," SAP Community, Sep. 15, 2019. https://community.sap.com/t5/technology-blogs-by-members/my-adventure-in-learning-cpi-part-2-deployment-models/ba-p/13394902

[4]. Sookriti_Mishra, "My adventure in learning CPI: Part 3 | Cloud Security," SAP Community, Sep. 15, 2019. https://community.sap.com/t5/technology-blogs-by-members/my-adventure-in-learning-cpi-part-3-cloud-security/ba-p/13398951

[5]. "SAP CPI | SAP Blogs," blogs.sap.com. https://blogs.sap.com/tag/sap-cpi/

[6]. STALANKI, "Comprehensive SAP CPI Guide for Standards & Best Practices," SAP Community, Jan. 16, 2020. https://community.sap.com/t5/technology-blogs-by-members/comprehensive-sap-cpi-guide-for-standards-best-practices/ba-p/13457873

[7]. "SAP Help Portal," help.sap.com. https://help.sap.com/docs/cloud-integration/sap-cloud-integration/sap-cloud-integration

[8]. "Home - Apache Camel," camel.apache.org. https://camel.apache.org/

[9]. "SAP Help Portal," help.sap.com. https://help.sap.com/docs/cloud-integration/sap-cloud-integration/integration-capabilities

[10]. engswee, "CPI's Groovy meets Spock – To boldly test where none has tested before," SAP Community, Jan. 24, 2018. https://community.sap.com/t5/technology-blogs-by-members/cpi-s-groovy-meets-spock-to-boldly-test-where-none-has-tested-before/ba-p/13367099

[11]. "Steps to learn SAP CPI, what you must know before your first project | SAP Blogs," blogs.sap.com.https://blogs.sap.com/2019/12/06/steps-to-learn-sap-cpi-what-you-must-know-before-your-firstproject/

[12]. "The Apache Groovy programming language - Syntax," groovy-lang.org. https://groovy-lang.org/syntax.html#_unicode_escape_sequence