



PRESS

# What's New in C\_CPI\_15?

from

SAP Integration Suite Certification Guide: Development Associate Exam



By Jaspreet Bagga

## A What's New in C\_CPI\_15?

Like SAP technology and software, SAP certifications are always evolving to keep pace with the latest industry changes. SAP has launched an updated version of the SAP Certified Development Associate - SAP Integration Suite certification exam (C\_CPI\_15) and deprecated the previous version (C\_CPI\_14).

Although this book was written before the publication of the C\_CPI\_15 exam, there is very little change between this exam and the previous exam. This certification guide will help you thoroughly prepare for the C\_CPI\_15 exam. In this chapter, we will give you an idea of what's new in the current exam, and review the few topics that haven't already been covered in the main book. We'll begin by discussing the exam structure changes in [Section A.1](#). Then we'll discuss what you need to know about the integration platform as a service, including SAP's integration approach and SAP's Integration Solution Advisory Methodology (ISA-M) in [Section A.2](#). [Section A.3](#) will cover API architecture operating modes, [Section A.4](#) will explore REST and RESTful APIs, and [Section A.5](#) will discuss Graph. We'll provide a list of important terms in [Section A.6](#) and then close this chapter with practice questions and answers related to these new C\_CPI\_15 topics.

### A.1 Exam Structure

The current SAP Integration Suite certification exam (C\_CPI\_15) is structured a little differently than its previous version. There are fewer questions in the latest exam and the passing score is higher. You also have less time for the C\_CPI\_15 exam. Here is the exam structure in more detail:

- **Exam type**

The certification exam for SAP Integration Suite at the associate level is conducted in a multiple-choice format, where you're presented with a set of answer options and you must choose the most appropriate one(s) based on your knowledge and understanding of the subject matter.

- **Number of questions**

The exam consists of a total of 60 questions covering a wide range of topics related to SAP Integration Suite (down from the previous 80 questions). These questions assess your knowledge, comprehension, and ability to apply concepts and principles in practical scenarios.

- **Cut score**

To pass the certification exam and achieve the associate-level certification for SAP Integration Suite, you must attain a minimum score of 75% (this is up from the previous 68%). The cut-off score is the benchmark set by SAP to determine the level of proficiency required to qualify for the certification.

## ■ Duration

You're allotted a total of 120 minutes (2 hours) to complete the exam (this is down from the 180 minutes/3 hours allowed for the previous exam). This duration is ample time to carefully read and analyze each question, consider the answer options, and select the most appropriate choice based on your understanding of the subject matter.

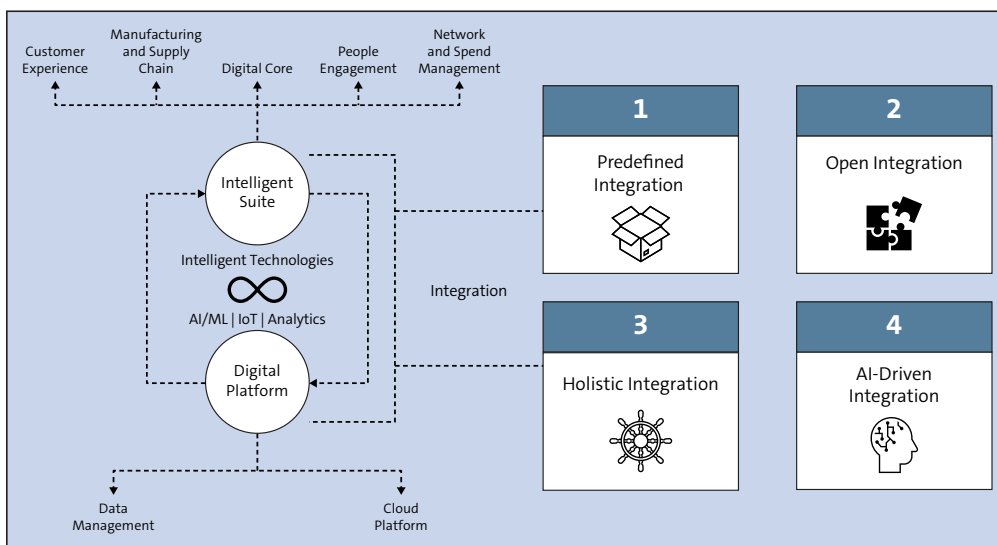
## A.2 Integration Platform as a Service

Beginning in this section, we will cover topics that are new to the C\_CPI\_15 exam and that have not been covered previously in this book. We'll begin by discussing the integration platform as a service (iPaaS).

iPaaS is a cloud-based service that gives businesses the ability to link and combine various software programs, hardware devices, and data sources. iPaaS solutions give businesses a platform for creating, deploying, maintaining, and keeping track of their integrations, making it simpler for them to streamline operations and ensuring that all of their software products can communicate with one another without any issues. As a result, productivity across the board is increased within an organization.

### A.2.1 SAP's Integration Approach

SAP's integration strategy for intelligent enterprises is designed to deliver a seamless and unified experience across its range of applications, as well as with partner and third-party solutions. This is accomplished using a variety of technologies. Figure A.1 illustrates SAP's integration approach for the intelligent enterprise.



**Figure A.1** SAP's Integration Strategy for the Intelligent Enterprise

SAP's integration strategy is based on four principles:

- **Predefined integration**

Prebuilt integrations are available for a variety of processes. This simplifies the integration of the intelligent suite's end-to-end processes (SAP systems to SAP systems, as well as SAP systems to non-SAP systems).

- **Open integration**

In addition to integrations between SAP products and between SAP products and partner products, SAP enables third-party integrations as well as bespoke extensions that make use of open APIs. For more than 170 third-party applications, SAP offers feature-rich, prebuilt connectors through the open connectors feature of the SAP Integration Suite.

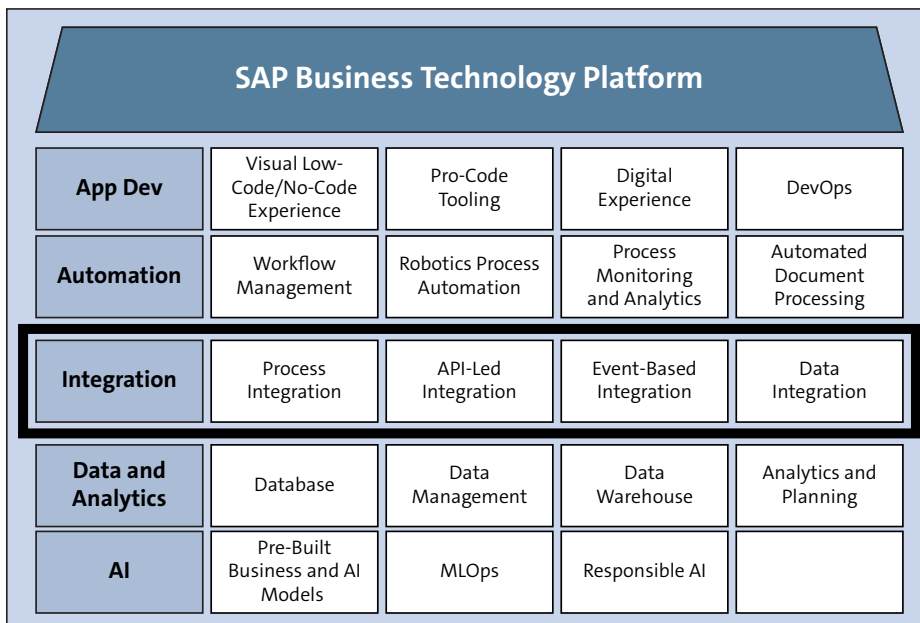
- **Holistic integration**

The majority of the types of integration needed in cloud and hybrid landscapes are covered by SAP's comprehensive integration technology portfolio. SAP supports a range of integration use cases, from process, data, user, and "thing" integration to analytics-focused integration, all of which are based on the SAP Integration Suite.

- **AI-driven integration**

SAP uses AI to streamline the creation of integration scenarios in addition to integrating intelligence into fundamental business processes. SAP Integration Suite's Integration Advisor feature is one example of AI-driven integration.

Integration is central to SAP's overall strategy. It is a key pillar of the SAP Business Technology Platform, as shown in Figure A.2.



**Figure A.2** Integration as a Pillar of SAP Business Technology Platform

Through integration, software applications, processes, and people can be securely connected whether on-premise, in the cloud, or in a hybrid architecture.

Overall, SAP supports integration in the following ways:

- By enabling the integration of SAP and other systems, including integration of third-party software, API management, B2B/B2G support, data integration, event-based integration, Internet of Things (IoT) support, and process integration.
- By supporting integration content that has already been developed, such as connectors, business events, integration packs, and APIs.
- By making best practices accessible through prepackaged SAP business content.
- By creating an AI-powered content advisor to speed up the creation of integrations and cut down on ongoing maintenance expenses.

## A.2.2 Integration Solution Advisory Methodology (ISA-M)

In this section, we delve into several key topics in relation to ISA-M. We begin by explaining what ISA-M is and illuminating its significance. Then we'll examine the ISA-M cycle more closely, breaking down its elements and features. Finally, we'll discuss how ISA-M is used, particularly for integration assessment, emphasizing its usefulness in various integration scenarios.

ISA-M is a comprehensive framework that assists organizations in defining, documenting, and implementing an enterprise integration strategy. It is founded on SAP's industry-leading techniques and expertise in assisting clients with the integration of their applications, data, and procedures. With the help of this methodology, integration requirements can be identified and mapped to particular services, and current integration solutions can be documented and defined. ISA-M offers a selection of common integration styles and patterns, assigns integration services in accordance with the context of the customer, and provides architectural solutions for integration that take decision-making into account. Roles and duties within integration scenarios are also defined with the aid of ISA-M.

In addition to these fundamental characteristics, ISA-M has the following benefits:

- **Openness to SAP and third-party integration technologies**  
ISA-M can be utilized with SAP or third-party solutions because it is not dependent on any one integration technology.
- **Use by target groups such as corporate architects and integration architects**  
The target audiences for ISA-M include corporate architects, integration architects, and other IT specialists working on integration initiatives.

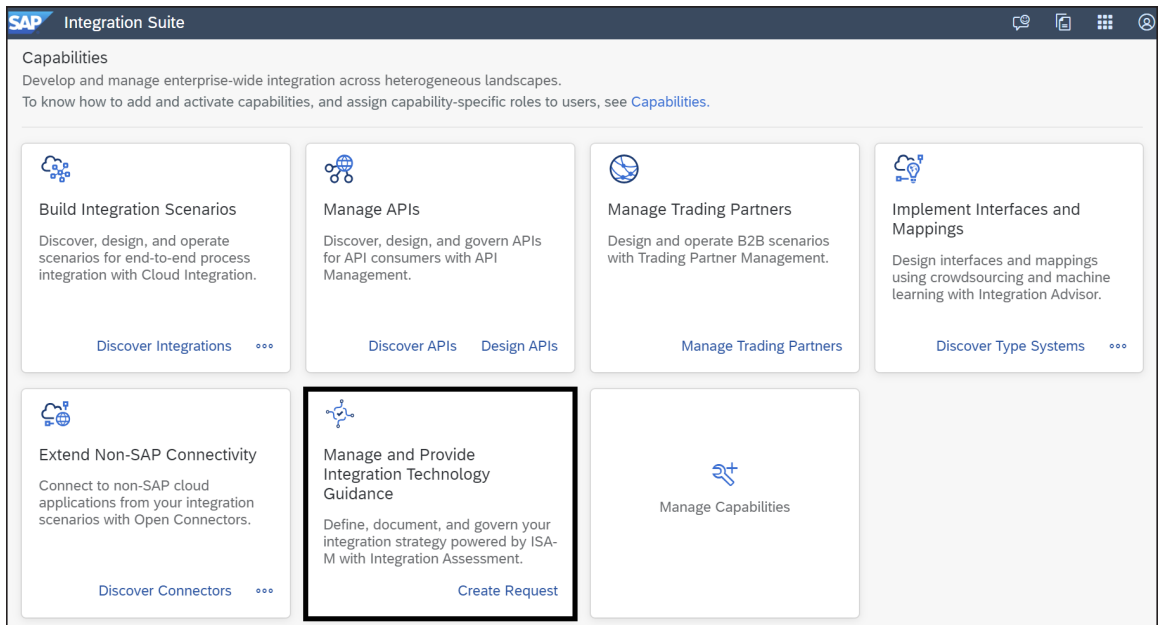
- **Use as a template for defining custom integration policies**

ISA-M can be used as a model to create unique integration policies, such as security, routing, and transformation policies.

- **Reusability inside organizations, projects, or services**

ISA-M can be used again and again within organizations, projects, or services. This implies that businesses can utilize ISA-M to create a uniform integration approach that can be applied to several projects or services.

SAP provides customers and partners with a PowerPoint template for ISA-M. This template has been built for integration architects, using SAP's integration expertise, and offers a starting point for enterprise integration. Additionally, in SAP Integration Suite, an integration assessment tool was introduced in May 2022, which you can find as the capability of SAP Integration Suite, as shown in Figure A.3.

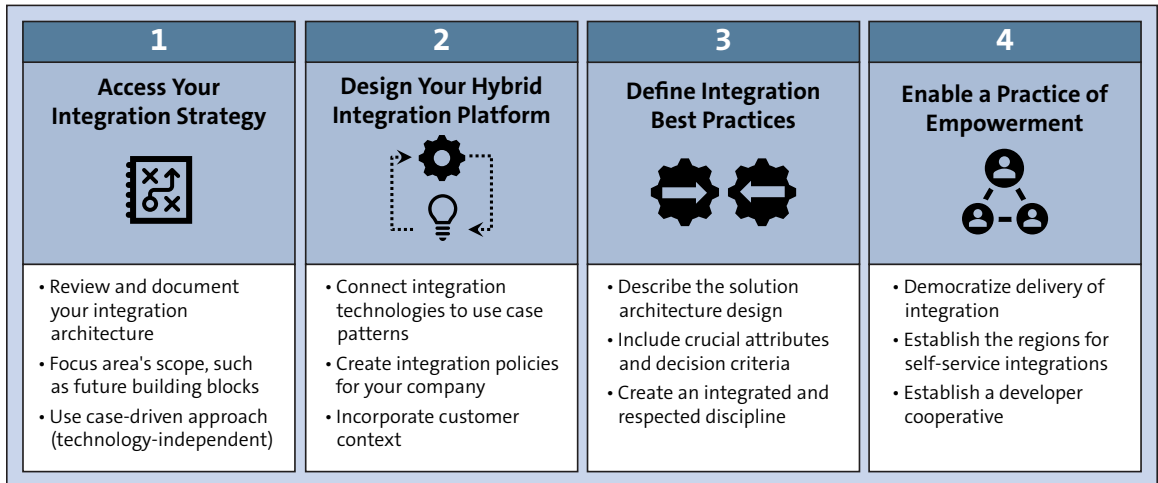


**Figure A.3** Integration Assessment in SAP Integration Suite

This tool, powered by ISA-M, offers integration patterns, architecture blueprints, and best practices, much like the ISA-M template. It is a tool-based approach to the methodology, to help integration experts define, document, and execute an integration strategy and assess current integration implementation.

## ISA-M Cycle

The ISA-M template arranges concepts within a four-staged cycle, in accordance with how SAP customers and partners are implementing the methodology. Figure A.4 shows the systematic approach of the methodology, with all four steps.



**Figure A.4** Systematic Approach to Integration in ISA-M

The various use cases for this methodology are described in this cycle:

### ■ Stage 1: Access your integration strategy

Customers and partners can evaluate their organization's existing integration plan using ISA-M's step-by-step methodology. The scoping of integration styles and related use case patterns is done after a high-level scoping on the level of integration domains. Enterprise architects can choose templates from SAP that are appropriate for their organization now and/or in the medium term.

### ■ Stage 2: Design your hybrid integration platform

The set of pertinent integration domains and use case patterns from stage one are mapped to integration technology and service categories to shape an organization's hybrid integration platform. The client context, including previous investments, business considerations, and more, has a significant impact on this mapping. A single integration platform may not always meet integration requirements. Therefore, to create a hybrid integration platform for an organization, enterprise architects may need to combine several technologies, such as an iPaaS and an IoT Platform.

### ■ Stage 3: Define integration best practices

After the hybrid integration platform's architecture is ready, best practices for integration are established to protect the execution of integration scenarios. This makes it possible to transfer integration requirements to internal or external integration developers properly and can



establish the parameters for the quality control of completed integration scenarios. Enterprise architects can produce architecture blueprints for the hybrid integration platform that address the pertinent integration use case patterns (including integration domains) for this purpose. These designs outline and detail the relationship between integration technology and business applications.

#### ■ Stage 4: Enable a practice of empowerment

Once the best practices for integration have been established, they can be implemented internally and externally (toward integration developers). Based on the established integration approach, project teams can then be given the freedom to design interfaces in an agile manner. This stage offers a feedback mechanism for adopting the integration strategy's lessons learned. This use case covers organizational aspects and procedures, such as creating an integration competency center. Additionally, this contributes to the recognition of integration as a profession inside an organization, supporting and facilitating projects related to digital transformation. Finally, this methodology allows for the scaling up of integration knowledge across the organization: Introducing self-service integration areas, for instance, with business persona implementation.

## A.3 API Architecture Operating Modes

In this section, we will explain the different operating modes of API architectures.

### A.3.1 Request-Driven Architecture

*Request-driven architecture* is a software architecture in which incoming requests from users or customers serve as the main source of the system's flow. In this architecture, clients send requests to the system, generally through a network (such as HTTP requests in online applications), to start interactions with it. Then, in response to these requests, the system's components or services carry out the required logic to process and perform the desired actions.

Figure A.5 shows how a service provider and service consumer can communicate directly in a request-driven architecture. This is synchronous communication.

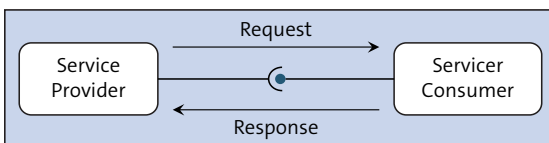


Figure A.5 Request-Driven Architecture



### A.3.2 Event-Driven Architecture

To understand event-driven architecture, we must first define events. An *event* is a brief communication that informs of a commercial occurrence. For instance, an event could be an order created in a SAP S/4HANA system. Push-based messaging (asynchronous communication) is used to fire the event to a broker. Events act as start-ups for certain responses or alerts within a system, enabling asynchronous communication between various parts or services. They convey crucial information about the event, letting event participants respond and take predetermined actions in response to the event's content. Events allow for responsive and decoupled communication between services and components.

Figure A.6 shows the architectural diagram of events produced from the producer passed through the event broker and consumed by the consumer.

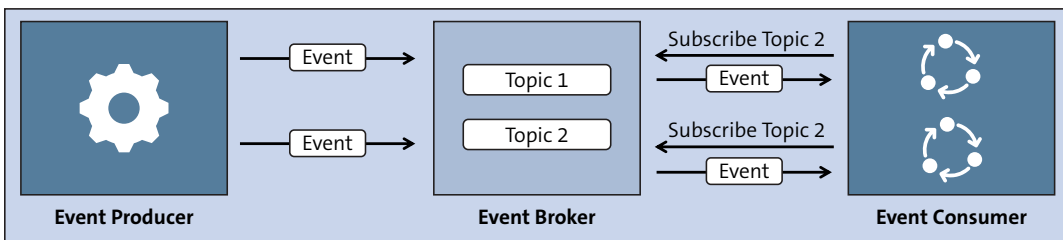


Figure A.6 Events

A software architectural design pattern called event-driven architecture places events such as user activities, system events, or communications from other components at the center of the system's operation. Components or services inside a system communicate by producing or receiving events in an event-driven architecture. These occasions set off particular processes or actions and enable loosely connected and asynchronous interactions.

#### Pull Variant

A *pull variant* refers to a specific approach to event consumption where event consumers actively request or “pull” events from a source when they are ready to process them. Figure A.7 shows the pull variant of event-driven architecture.

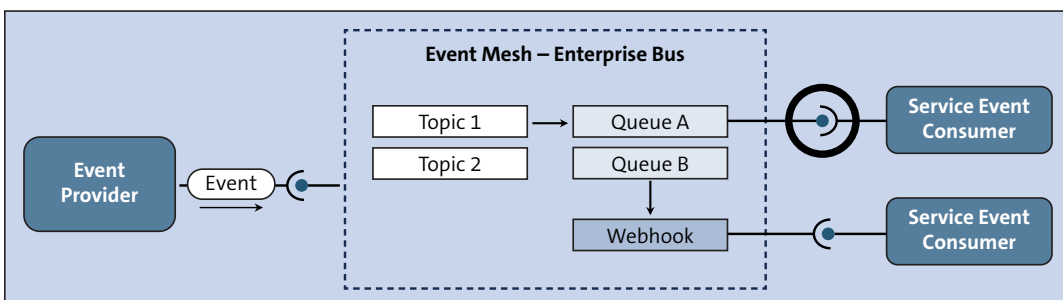


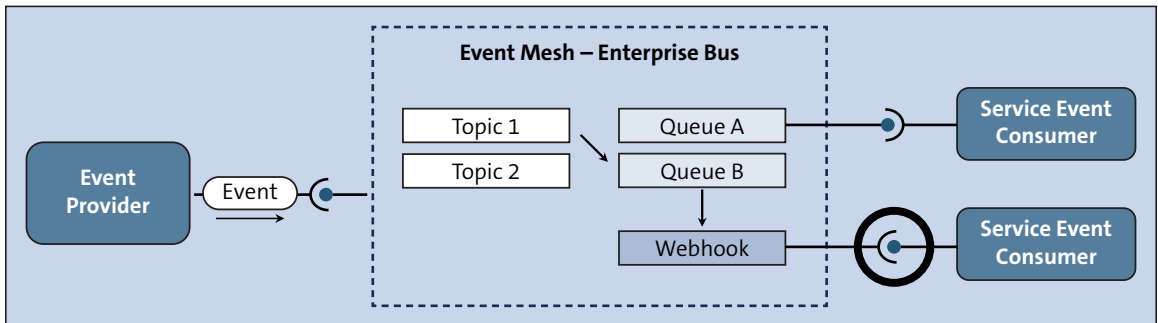
Figure A.7 Pull Variant in Event-Driven Architecture

The event provider starts the transmission of an event with a label, such as “topic”. This topic might have the name “OrderStatus\_Update.” This form of communication runs in an asynchronous manner. A queue that subscribes to the designated topic actively monitors it. The queue is referred to as **Queue A** in this instance. Using the event mesh’s given API, the event consumer can obtain the message. A request to the queue prompts the event consumer to start the interaction. The pattern of communication in this encounter is synchronous.

Overall, the event push to topic message is asynchronous whereas the pull from queue is a form of synchronous communication.

### Push Variant

The *push variant* of event communication in event-driven architectures actively sends events to event consumers as soon as event producers or a central event broker generates them. This implies that whenever a certain event takes place, it is instantly pushed or supplied to all interested customers without their actively asking for it. Figure A.8 shows the push variant in event-driven architecture.



**Figure A.8** Push Variant in Event-Driven Architecture

The event provider starts the event transmission with a label called “topic.” Such a topic might have the heading “CustomerInfo\_Updated.” The exchange of information happens asynchronously. **Queue B** has a webhook attached to it and is subscribed to the selected topic. The webhook is activated, and the event is sent immediately to the event consumer when an event with a matched topic enters the relevant queue. The event mesh facilitates this communication and incoming events start it, upholding an asynchronous communication architecture.

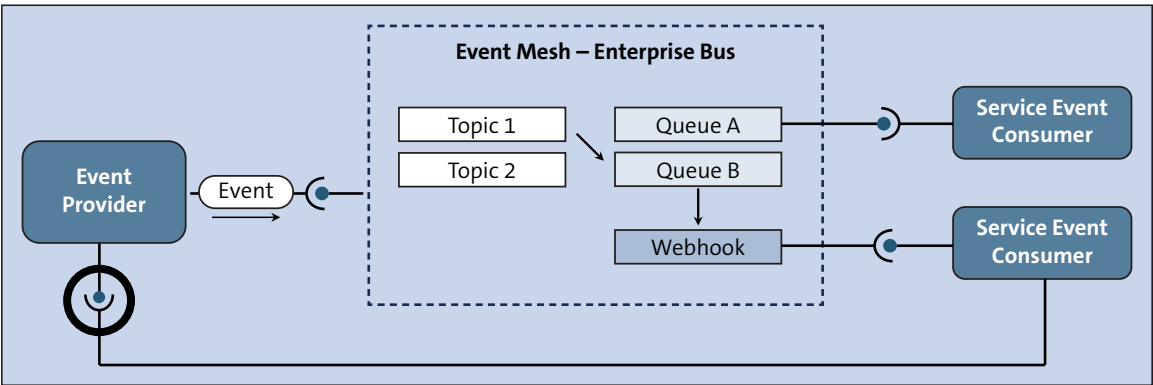
### Webhooks

A webhook is a tool for automating internet-based communication between several software programs. It performs as an HTTP callback and is often prompted by particular occurrences or adjustments within a single program. The receiving application can receive and process the data or take actions in real time when the webhook sends an HTTP POST request to a pre-determined URL in response to the occurrence of such an event. Webhooks provide seamless

and quick information sharing between platforms, services, and systems. They are frequently used for notifications, data synchronization, and workflow automation.

### A.3.3 Combining Event-Driven Architecture and Request-Driven Architecture

A useful strategy for creating intricate and responsive systems that can effectively manage both user-initiated actions and asynchronous events involves mixing request-driven and event-driven architectures. In order to make use of the benefits of each paradigm, this hybrid architecture combines elements from both. Figure A.9 shows the combined event-driven and request-driven architecture.



**Figure A.9** Combined Event-Driven and Request-Driven Architecture

The entire structure is same as the push variant, in which the event is triggered from the producer with the label “topic” and the information is exchanged asynchronously. **Queue B** has a webhook attached to it and is subscribed to the selected subject. The webhook is activated, and the event is sent immediately to the event consumer when an event with a matched topic enters the relevant queue. The event mesh facilitates this communication and incoming events start it, upholding an asynchronous communication architecture. The service event consumer submits the request to the event producer for instance, to read the recently modified product data. The data set can then be processed by the event consumer.

## A.4 REST

Representational state transfer or *REST* typically refers to a machine-to-machine interaction. REST, which is frequently referred to as *dynamic content*, enables content to be rendered when it is requested in web development. RESTful dynamic content creates a web page using server-side rendering and sends it to the user’s web browser, which interprets the server’s code and renders the page.

The REST architectural style exerts a profound influence on critical architectural properties:

- **Performance**

The statelessness and cacheability of REST improves component interactions, maximizing both network and user-perceived performance. Latency is reduced with cached answers.

- **Scalability**

REST's statelessness and layered architecture enhance scalability by making it easier to manage many components and their interactions, which is essential for systems that are subject to high loads.

- **Simplicity**

Simplicity is promoted through REST's standard interface, which provides consistent norms for resource manipulation and separates the client from the server.

- **Modifiability**

REST's decoupled architecture enables components to adjust flexibly to changing demands. Resources can change on their own, even while running.

- **Visibility**

REST makes communication between components transparent by using accepted HTTP methods and URIs, allowing service agents to monitor and control interactions efficiently.

- **Portability**

REST's reliance improves portability on common protocols and representations. Deployment across many contexts is made easier by the ability of components to travel along with program code and data.

- **Reliability**

The statelessness of REST and its layered design help to ensure system-level reliability by ensuring robustness against failures in individual components, connectors, or data sources.

The REST style also has some architectural constraints:

- **Client-server architecture**

REST separates client concerns (user interface) from server duties (processing) in a client-server architecture. Scalability and independence are promoted by this.

- **Statelessness**

Every customer request must include all pertinent information. The lack of server retention of client state between requests makes scaling possible and simplifies implementation.

- **Cache-ability**

Responses can be classified as cacheable or non-cacheable, which improves speed by using client-side caching.

- **Layered system**

REST maintains a flexible, scalable framework while accommodating intermediary layers like proxies and gateways.

- **Code-on-demand (optional)**

Although it's rarely used in practice, this constraint enables servers to send clients executable code.

- **Uniform interface**

A standardized, uniform interface with HTTP methods and status codes makes it easier to communicate with resources, which enhances interoperability and clarity.

## Restful API

Representational state transfer application programming interface, also known as a RESTful API, is a web service interface that adheres to the rules and guidelines of the REST architectural style. These APIs are made to be easy to use, stateless, and scalable. These APIs were created with the following features:

- **Base URL**

The base URL for resource access in a RESTful API is something like *http://api.example.com/*.

- **HTTP methods**

Common HTTP actions on resources are carried out using the GET, POST, PUT, and DELETE methods. Data is retrieved using GET, resources are added or updated using POST, deleted using DELETE.

- **Media type**

To specify the format of data transmitted between clients and servers, RESTful APIs use media types. Atom, microformats, or specialized media types like *application/vnd.collection+json* could all fit this bill. The media type defines the request structure that clients should use to move between application states and transitions.

## A.5 Graph API for SAP

*Graph* is an API for SAP that makes use of current open standards such as OData v4. With the help of Graph, all business data can be accessed through a single, semantically integrated business data graph.

In order to provide a standardized method of accessing and altering data, Graph makes use of the strong foundation of OData v4. Its adaptability is shown in how easily it connects to different SAP backends, enabling any of them to act as a data source. There is a repository of pre-existing, fully functional APIs available for use at <https://api.sap.com/graph> for those who are keen to get started. The adaptability of Graph also includes the ability to create customized

APIs, giving developers the option to create APIs that are adapted to their particular needs utilizing different processes. With its flexibility and wealth of resources, Graph is a handy tool for data integration and application development.

The *business data graph* is a tool that enables diverse data exploration through queries that delve into both data and their interrelated relationships. Entities, or data items, are represented as nodes in this data graph and each one has a home within a certain namespace. These entities have properties that correspond to their various fields. Edges, which represent semantic links, are crucial for navigating the complexities of data interactions. A dynamic method for understanding and utilizing complicated data structures is provided by Graph in the business data graph, which may be used to connect independent nodes through associations or connect a root node with its child nodes in a composition.

In the SAP Integration Suite, Graph is available as an extension of the API Management capability, as shown in Figure A.10. By extending conventional API management, Graph gives you the ability to access all of your company's data in the form of a semantically connected data graph using a single, robust API.

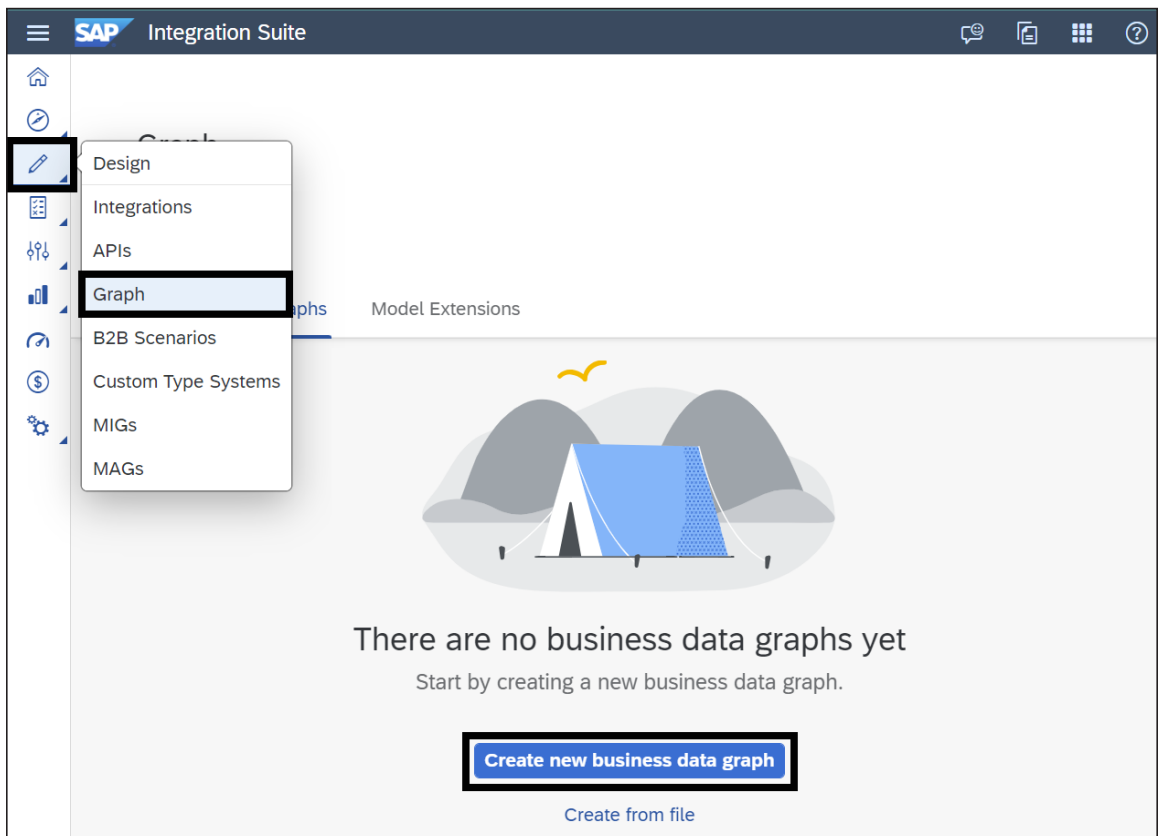


Figure A.10 Graph in SAP Integration Suite

## A.6 Important Terminology

In this chapter you have learned the following terminology:

- **Integration platform as a service (iPaaS)**

iPaaS is a cloud-based platform that facilitates the integration of applications and data across different systems, providing a unified environment for seamless communication.

- **Integration Solution Advisory Methodology (ISA-M)**

ISA-M is a methodology for guiding organizations in planning and implementing integration solutions, ensuring efficient and effective integration strategies.

- **Intelligent integration**

This refers to the use of AI and machine learning in integration solutions, allowing systems to adapt, automate, and optimize data flows and processes intelligently.

- **Event-driven architecture**

Event-driven architecture is a design approach where system components communicate through events or messages, enabling real-time responsiveness to changes and events within an application.

- **Request-driven architecture**

In this architecture, user or client requests trigger interactions within a system, allowing users to initiate and control processes through their requests.

- **Event**

An event is a significant occurrence or change in a system that can be detected and used to trigger actions or processes in event-driven architectures.

- **Webhook**

A webhook is a mechanism that allows one system to notify another system of events or updates by sending HTTP requests, enabling real-time data synchronization and integration.

- **Representational state transfer (REST or RESTful)**

RESTful refers to an architectural style for designing networked applications, often using HTTP, with a focus on simplicity and scalability in communication between systems.

- **Graph**

Graph is an SAP technology that enables the integration and access of business data from SAP systems, based on the OData v4 standard, providing a standardized and adaptable approach to data integration.

## A.7 Practice Questions

1. Which four ISA-M use cases are relevant to enterprise architects?

*Select the correct answer:*



- ☐ A. Access your integration strategy
    - Design your hybrid integration platform
    - Define integration best practices
    - Enable a practice of empowerment
  - ☐ B. Access your integration platform
    - Design your smart hybrid PaaS
    - Define your best practices via a consulting platform
    - Enable a practical guide for enhancements
  - ☐ C. Design your hybrid integration platform
    - Enable a practical guide for enhancement
    - Define integration best practice
    - Access your integration platform
2. What is the purpose of ISA-M?  
*Select the correct answer:*
- ☐ A. To provide a collection of common integration patterns and styles.
  - ☐ B. To enable SAP and third-party integration technologies.
  - ☐ C. To provide corporate architects with a technique to help them create their integration plan.
  - ☐ D. To provide integration services.
3. What are the four guiding concepts behind SAP's integration strategy?  
*Select the correct answer:*
- ☐ A. Predefined integration, open integration, holistic integration, and AI-driven integration.
  - ☐ B. Open integration, holistic integration, prepared integration of content, and integration of AI and BI.
  - ☐ C. AI-driven integration architecture, opened integration view, holistic integration view, and preconfigured integration.
4. Which of the following are commonly used HTTP methods?  
*Select the correct answer:*
- ☐ A. GET
  - ☐ B. PUT
  - ☐ C. POST
  - ☐ D. DELETE
  - ☐ E. All of the above

5. What are the four stages of ISA-M?

*Select the correct answer:*

- ☐ A. Access your integration strategy
- ☐ B. Design your hybrid integration pattern
- ☐ C. Define integration best practices
- ☐ D. Enable the practice of empowerment
- ☐ E. All of the above

6. Which of the following statements best describes a request-driven architecture?

*Select the correct answer:*

- ☐ A. It relies on automated processes to initiate interactions with clients.
- ☐ B. It is a software architecture where the system's flow is primarily determined by user or customer requests.
- ☐ C. It focuses on handling server-side events to trigger system actions.
- ☐ D. It is primarily used for batch processing in data-intensive applications.

7. What are the two variants in event-driven architecture?

*Select one or more answers:*

- ☐ A. Push
- ☐ B. Jump
- ☐ C. Pull
- ☐ D. Turn

8. What is a webhook primarily used for in software integration?

*Select the correct answer:*

- ☐ A. Generating random HTTP requests
- ☐ B. Real-time data sharing and automation between software programs
- ☐ C. Running web applications
- ☐ D. Managing database queries

9. How does the request-driven architecture communicate?

*Select the correct answer:*

- ☐ A. Asynchronous communication
- ☐ B. Synchronous communication
- ☐ C. Binary communication
- ☐ D. None of the above

10. What does RESTful stand for in the context of a web service API?

*Select the correct answer:*

- ☐ A. Real-time state transfer application programming interface
- ☐ B. Resource state transmission application protocol
- ☐ C. Representational state transfer application programming interface
- ☐ D. Responsive state transformation application protocol

11. Which critical architectural property is significantly influenced by the REST architectural style, allowing for reduced latency with cached responses and maximizing network and user-perceived performance?

*Select the correct answer:*

- ☐ A. Scalability
- ☐ B. Simplicity
- ☐ C. Modifiability
- ☐ D. Performance
- ☐ E. Visibility

12. Which technology is described in the provided information for accessing and integrating business data from SAP systems?

*Select the correct answer:*

- ☐ A. OData v4
- ☐ B. SAP HANA
- ☐ C. RESTful API
- ☐ D. Graph

13. Which technology serves as the foundation for Graph, enabling data access and integration within the SAP ecosystem?

*Select the correct answer:*

- ☐ A. SAP HANA
- ☐ B. GraphQL
- ☐ C. OData v4
- ☐ D. RESTful API

## A.8 Practice Answers and Explanations

### 1. Correct answer: A

The enterprise architect chooses these four use cases because accessing integration strategies allows for informed decision-making, designing hybrid integration platforms ensures efficient system integration, defining best practices promotes consistency, and enabling empowerment empowers teams to execute integration strategies effectively, making it a comprehensive and relevant set of use cases for enterprise architects.

### 2. Correct answer: C

The purpose of ISA-M is to assist corporate architects in developing their integration plans. In order to ensure that integration efforts are in accordance with business objectives and technological needs, it provides a structured methodology and guidelines for building integration solutions inside an organization. ISA-M supports effective and efficient integration planning by assisting architects in making educated judgements on integration patterns and styles.

### 3. Correct answer: A

SAP's strategy revolves around these four guiding concepts. Predefined integration refers to preconfigured solutions, open integration emphasizes interoperability, holistic integration aims for a comprehensive approach, and AI-driven integration integrates artificial intelligence for enhanced functionality.

### 4. Correct answer: E

For communication between clients and servers, HTTP (hypertext transfer protocol) typically employs the following four methods: GET (retrieve data), PUT (update or create a resource), POST (submit data for processing), and DELETE (delete a resource). In online applications, these techniques allow for a variety of interactions.

### 5. Correct answer: E

ISA-M covers a wide range of integration solutions, such as (A) accessing your integration strategy, (B) creating hybrid integration patterns, (C) outlining integration best practices, (D) and facilitating the practice of empowerment. A variety of crucial use cases are covered by this comprehensive technique for effective integration design and implementation.

### 6. Correct answer: B

Request-driven architecture is a software architecture in which user or client requests determine the system's flow. It indicates that the system interacts primarily with incoming client requests, launching activities and processes in response to those requests to meet user demands and requirements.

**7. Correct answer: A, C**

The two primary variations of event-driven architecture are *push* and *pull*, where events are actively pushed from producers to consumers and consumers request events when they are prepared to process them, respectively. These variations control the delivery and processing of events inside the architecture.

**8. Correct answer: B**

Webhooks are mostly utilized to facilitate real-time data sharing and communication across various software applications or services. They enable seamless integration and process automation, which are essential for contemporary software integration scenarios, by enabling one application to deliver automated notifications or data updates to another.

**9. Correct answer: B**

The request-driven architecture uses synchronous communication. This method of communication is real-time and interactive because when a client submits a request to the system, it anticipates an instant and direct answer. Binary communication (C) refers to the format of data exchange and is not a style of communication in this context. Asynchronous communication (A) often entails delayed or non-blocking interactions.

**10. Correct answer: C**

The term “RESTful” refers to a web service API design philosophy that places an emphasis on representing and manipulating resources as they actually are in a system. A popular and effective method for creating web-based APIs, it interacts with these resources via normal HTTP methods.

**11. Correct answer: D**

Performance is greatly impacted by the REST (representational state transfer) architectural style, which permits cached answers. In order to reduce latency with cached data and improve network and user perception of speed, it encourages statelessness, which requires that each request from a client to a server contain all the information necessary to interpret and process the request.

**12. Correct Answer: D**

The provided information makes Graph stand out as a method for integrating and retrieving corporate data from SAP systems. It highlights how Graph functions as a powerful tool for data integration and application development within the SAP ecosystem by offering a standardized approach for data access, flexibility with various SAP backends, and the capacity to establish customized APIs.

**13. Correct Answer: C**

The core technology for facilitating data access and integration inside the SAP ecosystem, OData v4, serves as the base around which SAP Graph is developed. This standard is crucial to the capabilities of Graph since it enables the easy interchange of data between various systems.

## A.9 Key Takeaways

The idea of an API-first approach has made great headway in the world of contemporary software design and integration tactics. With this strategy, APIs are at the forefront of system design and development, serving as the main channels for the exchange of data, services, and functionality. Several important concepts and technologies are at play under the API-first paradigm, each of which contributes to the effective and adaptable transmission of information in the digital environment.

The decision between request-driven and event-driven architectures is one of the key differences between API-first approaches. In a request-driven architecture, systems interact synchronously and communicate in real time with one another. This indicates that a client expects a prompt response to a request it delivers. When using event-driven architecture, which functions asynchronously, the sender (transmitter) and receiver are decoupled from one another in terms of time and content. In this configuration, messages or events start processes or activities without requiring a prompt reaction. Event-driven architectures' loose coupling has a number of benefits, including improved scalability, resilience, and flexibility, which makes them particularly well-suited for systems with fluctuating workloads and real-time processing needs.

REST, a software architectural approach that prioritizes simplicity, scalability, and the usage of common HTTP methods, is closely related to API-first design. A RESTful web API has a base uniform resource identifier (URI) that clients can use to interact with resources while adhering to the restrictions and characteristics of REST. It uses common HTTP operations on resources like GET, POST, PUT, and DELETE and supports a number of media formats for data representation, including JSON and XML. The foundation of the API-first strategy are RESTful web APIs, which enable developers to build effective, interoperable, and accessible interfaces for data and services and facilitate seamless communication between systems.

An integral part of the API-first environment, Graph is an SAP solution that provides a uniform and standardized way to access business data in SAP systems. Graph, which is based on the OData v4 standard, serves as a bridge to join dissimilar entities from different sources into a unified API. For example, it can integrate product data from various SAP systems, such as SAP S/4HANA and SAP Sales Cloud, and then give a unified interface for searching and modifying this data. Additionally, Graph gives developers the freedom to work with pre-existing Graph

APIs or design new APIs from scratch to meet their unique needs. Graph is positioned as a potent tool for data integration and application development within the SAP ecosystem by its versatility and the abundance of resources accessible.

SAP has established four guiding principles for its integration strategy: predefined integration, open integration, holistic integration, and AI-driven integration. This is in line with its commitment to provide thorough and reliable integration solutions. Together, these guidelines make sure that SAP's integration solutions are made to accommodate enterprises' changing needs. Prebuilt connectors and predefined integration templates streamline the integration process, making it simpler to connect different systems. Open integration encourages the use of open standards and interoperability while embracing them, giving businesses the freedom to select the technologies that best suit their requirements. Integrating people, processes, and technology all go under the umbrella of holistic integration, which addresses the wider integration concerns. Finally, AI-driven integration uses machine learning and artificial intelligence to intelligently automate and improve data flows and processes.

SAP provides ISA-M to organizations to help them navigate the complexities of integration projects and effectively implement these principles. This thorough structure acts as a guide for step-by-step project planning for integration. It assists in identifying crucial services, tools, roles, use cases, and other components needed for a successful integration. Additionally, organizations can profit from SAP's integration evaluation feature, which offers helpful guidance and insights all along the integration journey to guarantee a smooth and effective procedure.

## A.10 Summary

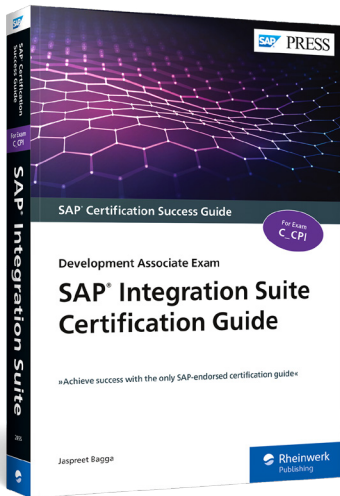
Two key concepts are at play in the world of API-first approaches: request-driven and event-driven. The participants in the former process engage in real-time interaction because it relies on synchronous communication. In contrast, the event-driven system operates asynchronously, separating the transmitter and the receiver's temporal and content domains. This loose connection has many benefits and enables systems to react to events and changes more quickly. Request-driven and event-driven paradigms are frequently used in succession, allowing systems to accommodate a variety of circumstances and needs.

RESTful web API development, which abides by the REST principles, is essential to the API-first design. These APIs have a base URI, use standard HTTP methods, support for different media formats, and have been carefully created to adhere to architectural features and limitations. An API-first strategy's core relies heavily on RESTful Web APIs, which are the main channel for system interaction, data exchange, and service provision.

On the other hand, Graph is a specialized API constructed on top of OData v4. By smoothly integrating elements from several sources into a unified API, it acts as a uniting factor. Graph, for instance, can combine product data from several SAP systems, such as SAP S/4HANA and



SAP Sales Cloud, inside a uniform interface. Developers can use Node.js to interact with the standard Graph APIs that are already available for a variety of entities or use the API Management capability in SAP Integration Suite to build unique APIs, even in low-code mode. The objective of SAP is to offer customers a complete, secure, and all-inclusive integration solution. Four fundamental tenets—predefined integration, open integration, holistic integration, and AI-driven integration—support this goal.



## SAP Integration Suite Certification Guide

### Development Associate Exam

Preparing for your C\_CPI\_14 exam? Make the grade with this SAP Integration Suite certification study guide! From API Management to the Integration Advisor, this guide will review the key technical and functional knowledge you need to pass the test. Explore test methodology, key concepts for each topic area, and practice questions and answers. Your path to SAP Integration Suite certification begins here!

417 pages, pub. 08/2023

E-Book: \$74.99 | Print: \$79.95 | Bundle: \$89.99

[www.sap-press.com/5735](http://www.sap-press.com/5735)



#### The Author

Jaspreet Bagga is an executive consultant and SAP integration subject matter expert with more than 200 successful projects delivered to Fortune 500 companies. Jaspreet is experienced in SAP integration project delivery, team management, IT transformation strategy, go-to-market strategy, SAP package selection, program budget and estimates, change control, business migrations, and managing stakeholder expectations. He has hands-on SAP skills in solution architecture, development, leading the delivery of complex integration programs, managing dispersed teams, and ensuring successful project go-live/goals.