

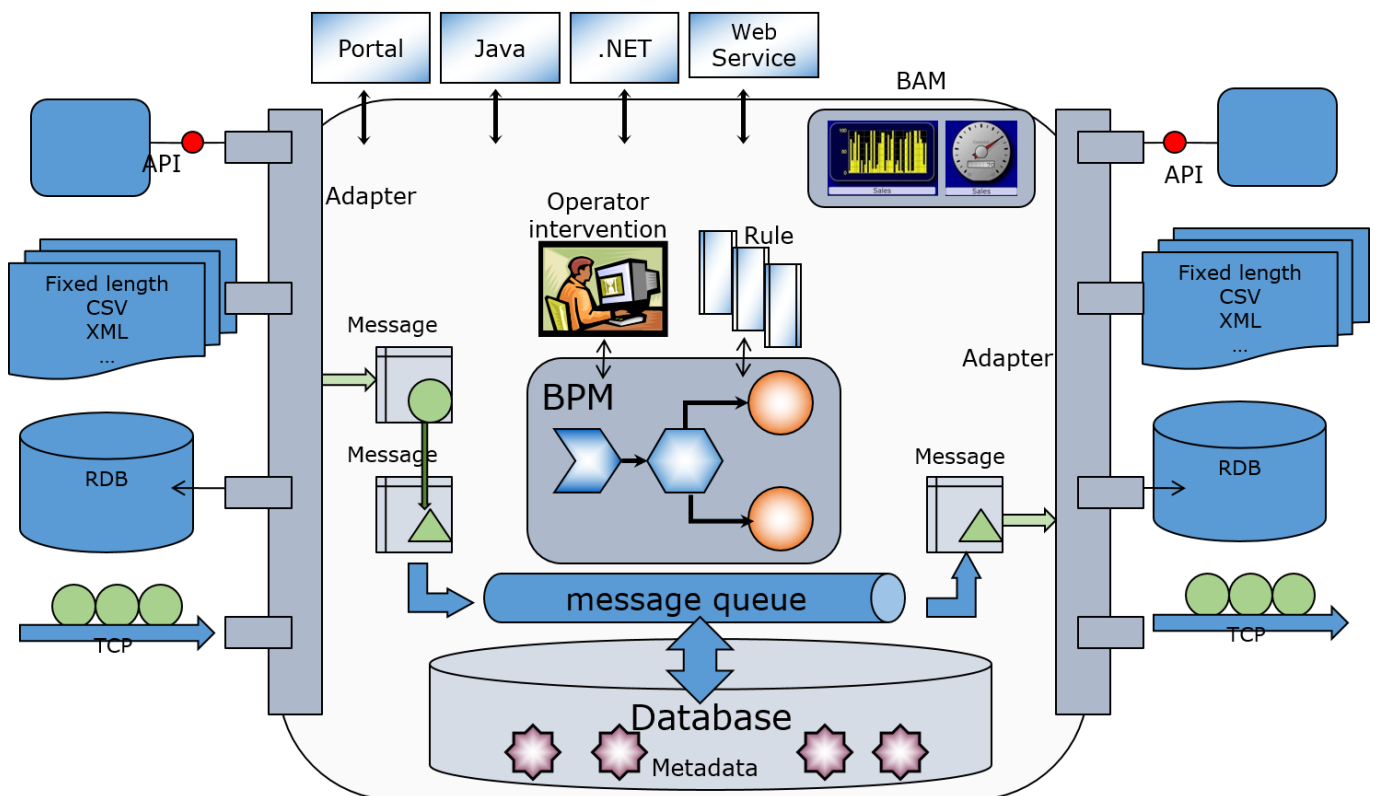
Article

[Mihoko Iijima](#) · Mar 5, 2021 3m read

[InterSystems IRIS for the First Time] Interoperability: Let's learn how it works

This article is a continuation of [this post](#).

The purpose of this article is to explain how the Interoperability menu works for system integration.



The left side of the figure is the window for accepting information sent from external systems.

There are various ways to receive information, such as monitoring the specified directory at regular intervals to read files, periodically querying the database, waiting for input, or directly calling and having it passed from applications in other systems.

In the system integration mechanism created in the IRIS Interoperability menu, the received information is stored in an object called a **message**. The **message** is sent to the component responsible for the subsequent processing.

A **message** can be created using all the received information or only a part of it.

Suppose you want to send the information contained in the **message** to an external system. In that case, you need to send the message to the component responsible for requesting the external network to process it (the right side of the figure). The component that receives the **message** will request the external system to process it.

Besides, suppose a **message** requires human review, data conversion, or data appending. In that case, the **message** is sent to the component in the middle of the diagram (BPM), which is responsible for coordinating the processing flow.

Messages are used to send and receive data between each component. When a **message** is sent or received, the message is automatically stored in the database.

Since **messages** are stored in the database, it is possible to check the difference before and after the data conversion. Check the **message** that was the source of a problem during an operation or start over (resend) from the middle of the process. Verify the status using **messages** at each stage of development, testing, and operation.

A simple picture of system integration would be divided into three components (business services, business processes, and business operations), as shown in the figure below.

There is also a definition called "**production**" that stores information about the components to be used (e.g., connection information).

The role of each component is as follows:

Business Services

Responsible for receiving information from external sources, creating **messages**, and sending **messages** to other components.

Business Processes

This role is activated when a **message** is received and is responsible for coordinating the process (calling components in the defined order, waiting for responses, waiting for human review results, etc.).

Business Operations

This function is activated when a **message** is received and has a role in requesting the external system to process the message.

Messages are used to send and receive data between components.

Components other than business services initiate processing when they receive a **message**.

The question is, what is the purpose of creating and using this **message**?

Messages are created by retrieving the information you want to relay to the external system from entered data into the business service.

Since not all external systems connected to IRIS use the same type of data format for transmission, and the content to be relayed varies, the production can freely define **message classes according** to the information.

There are two types of **messages**: request (= request message) and response (= response message). The **message** that triggers the component's activation is called request (= request message), and the **message** that the component responds to after processing is called response (= response message).

These **messages** will be designed while considering the process of relaying them.

In the following articles, we will use a study case to outline the creation of **productions, messages, and components**.

[#Beginner](#) [#Interoperability](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

Source

URL: <https://community.intersystems.com/post/intersystems-iris-first-time-interoperability-lets-learn-how-it-works>