**Introducing Integration Procedures**

So what exactly are OmniStudio Integration Procedures? They are applications used to read and write data from Salesforce and from external systems. An Integration Procedure can be called from an OmniStudio component such as an OmniScript or FlexCard, an API, or even from an Apex method.

Think of it this way: When you need to access and transform data from third-party sources and no user interaction is required, and when moving the workload from client to server is desirable, then that's when you want to use an OmniStudio Integration Procedure.

OmniStudio Integration Procedures are declarative, server-side processes that execute multiple actions in a single server call. Let’s break this down to better understand what it means.

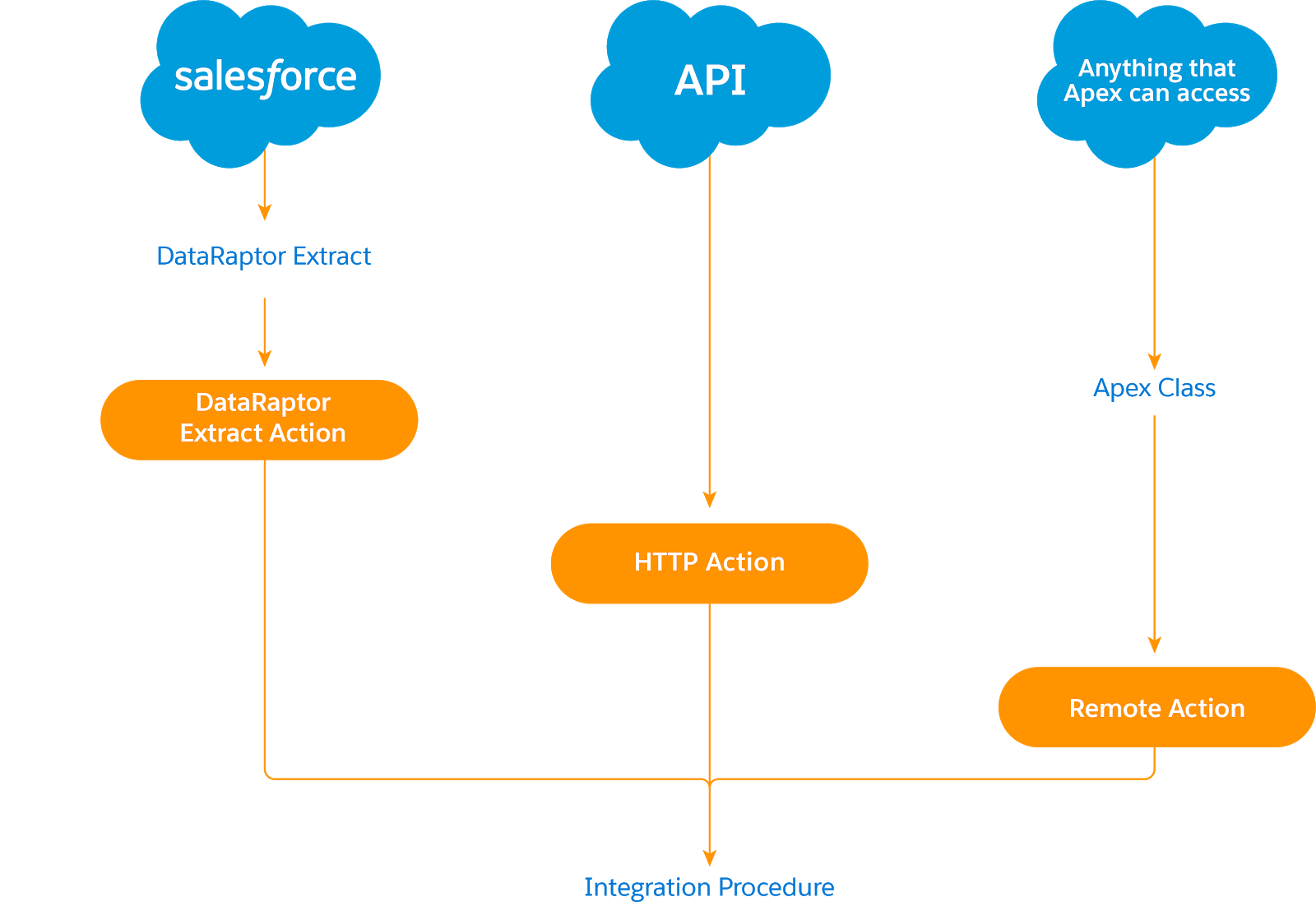
* **Declarative:**Use drag-and-drop elements in the Integration Procedure Designer to build the process structure.
* **Server-side processing:** This enables faster performance, because the server, in most cases, is faster than the client at processing data.
* **Multiple actions in a single server call:** This prevents round trips to the server. Minimizing client/server calls is beneficial, as more round trips mean slower performance.

More simply, Integration Procedures are a way to get, save, and manipulate data behind the scenes. They’re especially useful in the following scenarios.

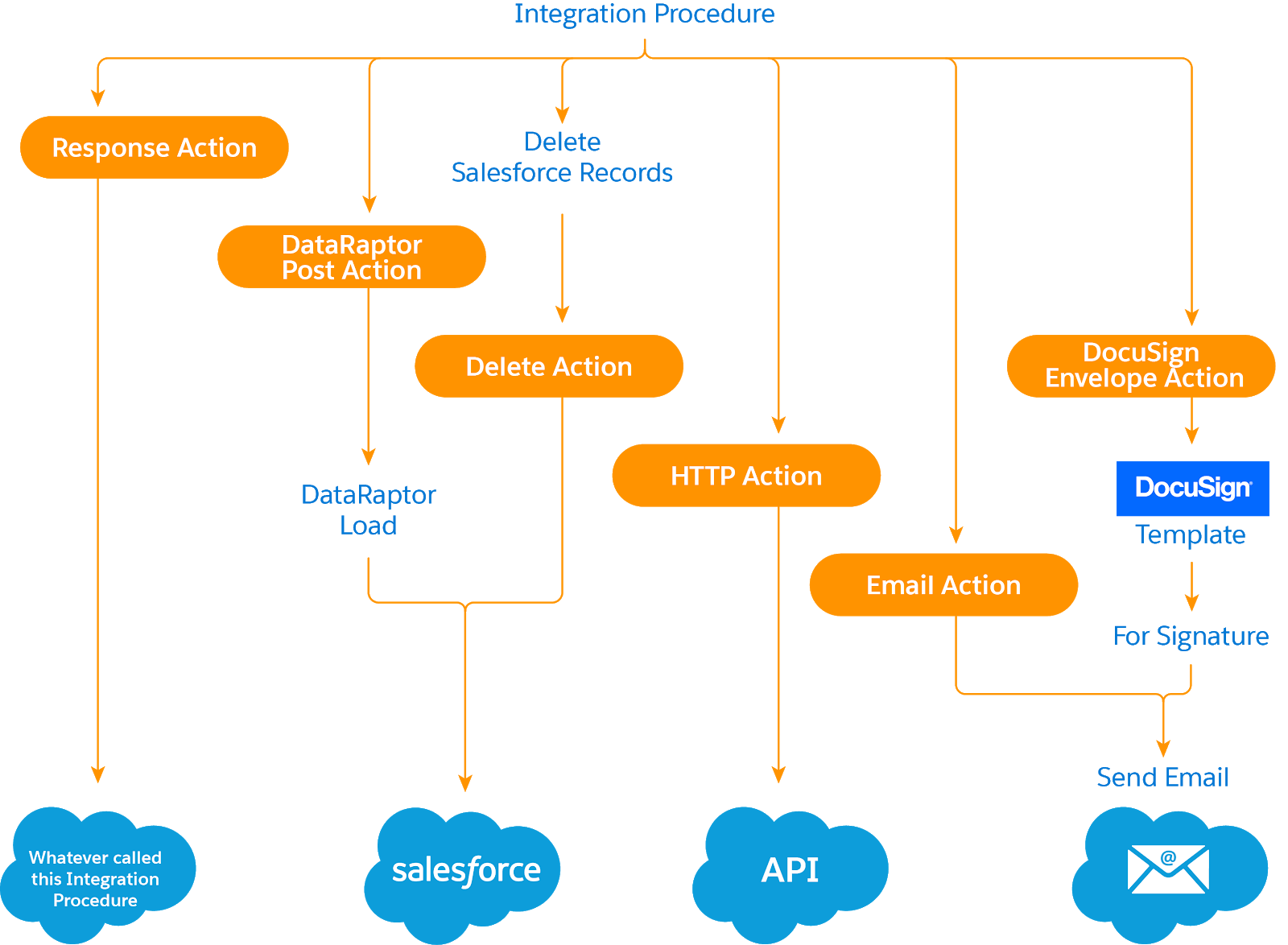
* You need to access and transform data from third-party sources.
* No user interaction is required.
* Moving the workload from client to server is preferable.

Integration Procedures do what they do thanks to a host of awesome capabilities.

**Integration Procedures handle multiple data sources.** They read data from Salesforce, external systems (using representational state transfer [REST]/application programming interface [API] calls), and Apex classes. It looks a little something like this:



**Integration Procedures serve as data sources for multiple technologies.** They are called from and return data to OmniScripts and OmniStudio FlexCards. They also serve as data sources for APIs and Apex code. Check it out:



**Integration Procedures are portable.** This means you build an Integration Procedure once and use it everywhere. You can use the same one in a FlexCard as well as in an OmniScript.

**Integration Procedures only send and receive the data you need.** An often overlooked performance factor is the amount of data being sent between the browser and the server. The Integration Procedure’s Response action allows you to trim the data returning to the browser from the server. This minimizes client/server data transfer, which is a key factor if you’re on a slow network or using a mobile connection.

**Integration Procedures perform batch processing.** They process large amounts of data without causing a Salesforce timeout.

As you can see, Integration Procedures have some fantastic capabilities. Next, let's examine how they make working with data management easier.

**Integration Procedure Advantages**

We recommend using Integration Procedures as much as possible as a data source. Why? Because they have a streamlined structure that’s efficient and consistent. They’re adaptable to use with all your data sources and easy to navigate.

But that’s not all—they give developers a lot more control over the data going back and forth and:

* Provide optimal flexibility.
* Make implementations easier.
* Greatly improve the performance of FlexCards and OmniScripts.

Another huge advantage of using Integration Procedures is that they future-proof your design.

For example, when you design a FlexCard, you need some data from the server. However, you might not yet be entirely sure what that data looks like or how to get it. Here’s how Integration Procedures provide a solution.

* You create your Integration Procedure with sample (mock) data, and call the Integration Procedure from the FlexCard. You move forward in your card design, even if the backend system is not yet ready.
* When that backend system becomes ready, you simply make the changes to the Integration Procedure, and you’re up and running. You no longer have to touch your FlexCard. This approach is a good way to separate your development work on the front end from your development work on the back end.

In addition to these benefits, replacing Apex classes with Integration Procedures has several perks. For example, Integration Procedures:

* Require no Apex test code coverage, which is a time- and cost-saving technical advantage.
* Are much easier to maintain and update.
* Take up to 97% less development time.

Let’s take a closer look at that last one. Here’s a comparison of development times with OmniStudio implementations when using custom Apex classes versus using Integration Procedures.

|  |  |  |  |
| --- | --- | --- | --- |
| **Backend Service Complexity** | **Build Time Apex** | **Build Time Integration Procedures** | **% Reduction in Level of Effort and Duration Using Integration Procedures** |
|  |  |  |  |
| Easy | 2 hours | 30 minutes | 75% |
| Difficult | 6 weeks | 1 day | 97% |