

# Power Platform Insights

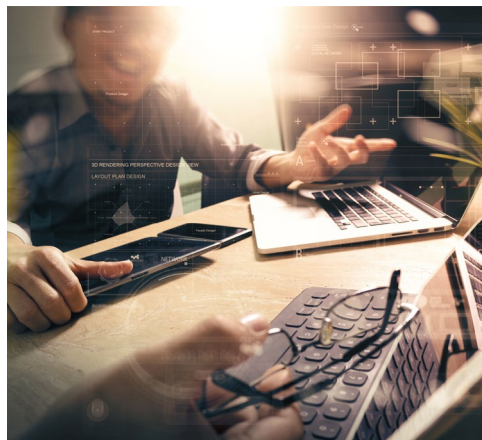


# 9 thought-provoking questions about the Power Platform



## Introduction

This month we gathered nine thought provoking questions about the Power Platform and posed them to our Hitachi Solutions' experts — people who work every day deploying applications and automations on the Microsoft Power Platform. Following are their responses.



Answers provided by and special thank you to: **David Handke**, senior technical consultant; **Jerry Martin**, senior architect and team lead; and **Ramiro Melgoza**, senior Power Platform developer.

Have any questions you would like us to answer in future newsletters?



Send them to [lgreszler@hitachisolutions.com](mailto:lgreszler@hitachisolutions.com)



## Section 1

# How do we decide if we should use low-code or traditional development to build an application?



The answer to this question comes down to several factors — a big first step is answering “who?” Determine who is maintaining the app/solution at the end of development. If you have experienced developers on staff, then there is normally no need to make them switch to low code as traditional development is way more comfortable and efficient.

But if you don't have traditional developers on staff, learning low-code development tools will be easier. Supporting a model-driven app is far easier for the programming novice than trying to manage a UI built with Angular, React, etc. Microsoft has made the tooling quite easy to understand and the need for a truly technical person is significantly reduced.

Another consideration is the ultimate complexity of the solution. Low-code tools generally are not best for intricate detail since you often have to give up control of tuning efficiency to gain the benefit of low-code creation.

This is especially true when it comes to disconnected apps. When truly granular control is required, either in processing or in UI layout, a custom solution is often a better choice. That said, most applications do not require that level of control and are best served with a solution that is easy to maintain. Deployment in low/no code is also significantly easier with predefined solution management and DevOps pipelines.

The question should really be reframed to this: “How can I use low-code and traditional development together?” The Power Platform provides a “no-cliffs” experience that enables you to use low-code tools to develop your applications rapidly, while still allowing you to leverage traditional code with custom connectors, custom controls, Logic Apps, Azure Functions, and other resources whenever necessary. These code-first components can be surfaced natively in the low-code experience to close the loop and enable developers at all skill levels to build solutions rapidly. Also, since Dataverse has a fantastic API, you can easily extend your Power Platform solutions with custom applications.



## Section 2

# How well does the Power Platform work with Oracle and SAP?

Oracle and SAP are two of the most common data sources used with Microsoft Power Platform. While there are additional steps required to connect to these, such as deployment and configuration of the on-premises data gateway, apps, flows, and reports connecting to Oracle or SAP can be highly performant.

Working with Oracle DB from an app or flow is no different than working with SQL Server. While there are differences in SQL syntax, the basic layout of the data is similar and the supported connectors are available.

The experience with SAP often depends on the version of SAP being used. The more recent **HANA** supports integration far better than did previous versions. With older versions, the data was often exported to a common platform for integrations or custom web services/APIs were created. Hana permits these to be created with greater ease.

In the event a function you require is not in these existing connectors, custom connectors or **Robotic Process Automation (RPA)** offer the perfect way to interact with your Oracle or SAP environment. The use of Desktop flows/RPA can eliminate the need for transitory data completely.



## Section 3

# When building a Power App, how do you decide whether to have the app perform an operation or use a Power Automate flow to perform the operation instead?

The question of which service should perform what action falls back to the requirements and intended user experience of the application as well as the number of records an action is being run against per instance. If an action that is performed by a user in an app must have its results immediately presented to the user (eg: user updates a record => user sees that change reflected on the next screen), then having the app perform the operation is likely the better approach. This ensures the request is handled in as-close-to-synchronous fashion as possible.

If the user action on a record can be deferred (because they don't need to see that change immediately in the application), then handing it off to a Power Automate cloud flow could be preferred. One example of this is having a Power Automate cloud flow trigger off of a record's modification to send out an Approval Request. In most cases, the approval request isn't returned to the application right away and can be handled in an asynchronous fashion. One additional area where a Power Automate cloud flow may be useful is where there are actions being taken on several records at a time.

Power Automate leverages the strength of Microsoft's cloud infrastructure to handle these types of transactions as quickly as possible, whereas an application may be limited by the users' devices and network connection. In these types of scenarios, having Power Automate do

the heavy lifting may improve the perceived performance of an application with minimal impact to the user experience/interface.

Additional considerations:

- Should the operation be performed in another user context? *Use a flow*
- Is it a long running task? *Use a flow*
- Do you need to re-use the functionality elsewhere in the solution? *Use a flow*
- Does the operation need to happen synchronously (real-time)? *Have the app perform the operation*

Some developers like to have separation of responsibility within their app. It can be very confusing to do some IO operations within an app and others from a Power Automate flow. While a great deal of efficiency can be had from using the IO operation in an app directly, it is often done so at the risk of easier maintenance. Almost anyone can read a flow and follow the arrows to understand what is being done — while reading a patch statement requires more of a programmers' mind set. Solution Architect Jerry Martin's general rule is to go the way that is easiest to maintain for the folks who will be tasked and then move to the more complex as needed to gain efficiency.

“ Make it work, make it right, and make it fast — in that order.

*Jerry Martin*





## Section 4

# When would you choose Logic Apps vs Power Automate?

Power Automate cloud flows are built on the Logic Apps platform. Logic Apps and Power Automate share the same connectors. There are more similarities than differences when it comes to performance and execution of tasks. Logic Apps is positioned as an integration tool, Power Automate is an automation tool. The following are considerations when deciding if a task should be performed by Logic Apps or Power Automate:



**Supportability:** If you have people with good technical background or Azure experience, use Logic Apps.



**Licensing/cost:** Logic Apps is priced based on consumption, Power Automate pricing is more of a licensing model — users will have licenses for certain flow capabilities (such as flows that leverage standard connectors) as part of their Office 365 license, so you generally have some flow capacity that is already paid for. Power Automate does not charge per run, but does have limits for the number of runs based on the license that the owning user/service account has.



**Capabilities:** While Power Automate cloud flows are based on Logic Apps, they have a number of capabilities not available with Logic Apps. This includes the modern Dataverse connector (Logic Apps does not support the newer “current environment” connector), does not include desktop/RPA flows, approval flows, the ability to add the process to a solution, Process Advisor, and many other very useful capabilities.

Logic Apps should be leveraged primarily when the operation requires significant IO that is not easily managed by the use of individual licensing for Power Automate. And, it isn't either/or. Many of our customers leverage both Logic Apps and Power Automate strategically. Use the right tool for the job.



## Section 5

# I have an application that we built on SharePoint, but we want to move it to Dataverse — what are the steps to do that?



There are two main ways to transfer to or leverage your SharePoint-based solution in Dataverse:

1. Use Dataflows
2. Use Virtual Tables

Which of these you choose is dependent on your end-goals with the use of the data.

**Dataflows** allow you to setup one-time or recurring one-way transfers of data from your source (eg: SharePoint List) to Dataverse. If your intent is to migrate your data into Dataverse and then use that as the system of record, a one-time transfer using Dataflows may be one way to accomplish that. Once in Dataverse, you can update any existing applications to point to the new data source.

**Virtual Tables** allow you to create a virtualized Dataverse table that gets its data directly from your data source — including SharePoint Online Lists. Where Virtual Tables differ from Dataflows is in their ability to handle Create, Read, Update, and Delete (CRUD) operations against that source. This effectively enables two-way interaction between SharePoint and Dataverse, where SharePoint can continue to be the source of truth while also allowing you to build platform-native solutions (eg: Model-Driven Apps) tied to the virtual Dataverse table.



## Section 6

# How do I decide if we should address a business need using an app or using a chatbot?

A Power Virtual Agents chatbot and a Power App are different kinds of apps — one is conversational/text based, while the other is graphical. Both can be used to quickly perform an operation or provide information to a user. Here is what we look at when deciding if something is a good fit for an app or a chatbot:

**Focus:** How focused does the user experience need to be? Apps can provide a variety of features and give the user a choice of several paths to follow. With a chatbot, you are proceeding down a focused conversational tree, designed to narrow down a request or question with the goal of providing information, answering questions, or gathering input to launch an operation. For example, searching a knowledge base, creating a helpdesk ticket, answering questions are all great fits for chatbots. But a scenario like time entry might not be because for time entry, users don't always follow the same path, and they frequently jump around. For a time entry system, an app would be better, as it requires a more visual interface. However, a user requesting to cancel submission of their timesheet might be a good use case for a chatbot, as that is a more focused operation.

**Authentication:** A chatbot can be used for anonymous users, where an app cannot be used for anonymous users. If you want to add an anonymous way for your customers to request more information about your business and embed it in your website, a chatbot provides this capability and may be extended by developers (when using bot framework composer).

**Licensing:** Power Virtual Agents is priced based on tiers of sessions and can be very affordable for external users.



## Section 7

### How do I decide if should build a Power App vs. a Power BI dashboard?

The primary use-case of Power Apps is to create solutions that view and modify data, be it a record or a collection of records. Power BI's primary use-case is the ability to display and analyze data to gain further insights from it through the use of traditional visualizations and advanced AI analytics (powered by Azure).

Thankfully, both services can be used together, with Power Apps supporting the ability to embed Power BI tiles within a given application and Power BI allowing you to embed a Power Apps canvas app as a Power BI visual. Depending on your scenario, choosing one or both of these options will allow you to build the exact solution you need.



## Section 8

### Am I “painting myself into a corner” by building a Power App on Dataverse? What if our needs change in the future, are we locked in?

Power Apps reduces the risk of “lock in” compared with monolithic application platforms like SAP. With loosely coupled micro apps and services, you can easily replace one or more of them without having to replace the entire system.

Dataverse is a data service that leverages Azure SQL, Azure blob, and other standard Azure services. While it is a primary data source for Power Apps, it also has a very mature API and you can easily build custom applications on Dataverse.

Regardless of choice of data storage, a well written app can be changed to use a different data store if required. It is usually easier to go in with an idea of the long-term plan for the app, but the beauty of the Power Platform is the ability to change directions mid-stream without causing a huge amount of disruption or rewrite.

When you bring your data into Dataverse, your data remains yours. Whether exporting it manually or through APIs, that same data can later be connected with systems you onboard in the future to migrate out of or integrate with Dataverse. If you do ever decide to no longer store your data in Dataverse, you can rest assured that the vast number of third party connectors (and the ability to create custom connectors) will allow you to update any existing applications to connect to your new data source provider.





## Section 9

# How do we predict and budget for future license and capacity growth?



Budgeting for future growth can be difficult, especially when you are first getting started with the platform.

One reasonable way to predict the capacity growth of a Power App is to look at the current data footprint, if migrating from a different platform. If that is not an option (maybe this is a new process), you can also do a proof of concept to get the average footprint of a record instance — for example, a client was deploying a maintenance checklist Power App and wanted to know how much storage they would need. They had some users create several checklist responses, including the average number of photos that they would typically have, and from that we got an average data size per checklist response. Then estimate the expected number of responses you will have on a weekly basis, and you can get a fair estimation of how fast your capacity will grow.

You can also mitigate potential increases with a good data retention policy. Not all records are needed in the active database, and Dataverse makes it very easy to export your data automatically to a data lake, removing it from the capacity limitations of Dataverse.

And for very high volume data, consider storing that data somewhere else — such as Azure SQL — and leveraging Virtual Tables to surface that data inside of Dataverse.

But what about inconsistent use cases? Thankfully, the Power Platform offers a “pay-as-you-go” option, which licenses the solution rather than the individual user. Using this option allows you to pay for actual utilization rather than committing to large up-front cost for individual user-license that may sit unused initially. Once utilization stabilizes, you’ll be able to make more informed decisions on who may require an individual license due to their consumption of several different applications and who can remain on a per-app model.

In addition, most Microsoft/Office 365 licenses include some access to create Power Apps, Canvas Apps, or Power Automate cloud flows using non-premium connectors. This enables you to create simple solutions on data sources you’re already familiar with such as **Excel**, **SharePoint**, and **OneDrive** — without additional cost.

## Microsoft Power Platform

# “In the news”

The following are the top news items from the Microsoft Power Platform team.

Download Power Apps onto your desktop and leverage offline functionality with the Power Apps for Windows application

<https://powerapps.microsoft.com/en-us/blog/announcing-general-availability-of-power-apps-for-windows/>

Automate data extraction from contracts, statements of work, and financial reports using AI Builder’s new unstructured document processing capabilities

<https://powerautomate.microsoft.com/en-us/blog/automate-unstructured-documents-like-contracts-with-ai-builder/>

Create Power Pages that can be accessed by external customers, vendors in the new Power Pages design studio

<https://powerpages.microsoft.com/en-us/>

Reassign a flow to a new owner from the Power Automate portal

<https://powerautomate.microsoft.com/en-us/blog/change-owner-of-a-solution-flow/>

Use an expanded set of data types for input/outputs variable in Power Automate Desktop

<https://powerautomate.microsoft.com/en-us/blog/power-automate-for-desktop-june-2022-update/>



Looking for ways to connect in person on all things Microsoft Power Platform? Check out Microsoft's first annual **Power Platform conference in Orlando, September 20-22**. Look for Hitachi Solutions — including Troy Taylor who is a featured speaker — while you're there!

### Why Hitachi Solutions?

Hitachi Solutions helps its customers successfully compete with the largest global enterprises using powerful, easy-to-use, and affordable industry solutions built on Microsoft cloud services.



**Contact Hitachi Solutions today to get started!**



[global.hitachi-solutions.com](http://global.hitachi-solutions.com)