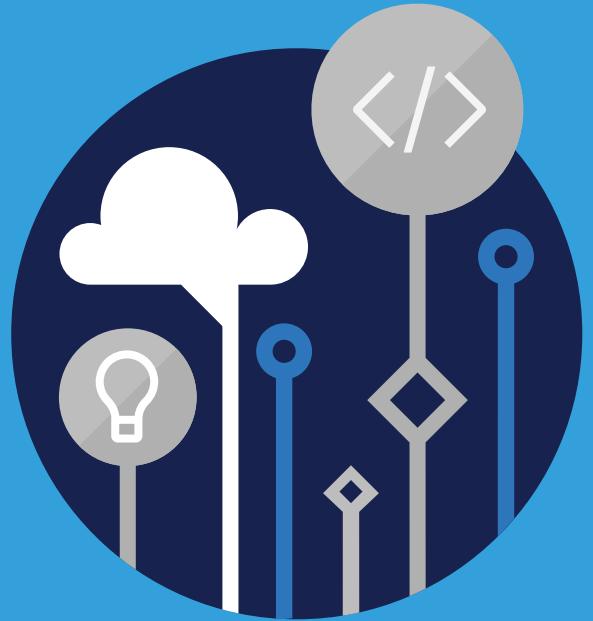


Microsoft  
Official  
Course



**PL-600T00**

Microsoft Power Platform  
Solution Architect

**PL-600T00**  
**Microsoft Power Platform**  
**Solution Architect**

---

## II Disclaimer

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2019 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at <http://www.microsoft.com/trademarks><sup>1</sup> are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

---

<sup>1</sup> <http://www.microsoft.com/trademarks>

## MICROSOFT LICENSE TERMS

### MICROSOFT INSTRUCTOR-LED COURSEWARE

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to your use of the content accompanying this agreement which includes the media on which you received it, if any. These license terms also apply to Trainer Content and any updates and supplements for the Licensed Content unless other terms accompany those items. If so, those terms apply.

**BY ACCESSING, DOWNLOADING OR USING THE LICENSED CONTENT, YOU ACCEPT THESE TERMS.  
IF YOU DO NOT ACCEPT THEM, DO NOT ACCESS, DOWNLOAD OR USE THE LICENSED CONTENT.**

**If you comply with these license terms, you have the rights below for each license you acquire.**

#### 1. DEFINITIONS.

1. "Authorized Learning Center" means a Microsoft Imagine Academy (MSIA) Program Member, Microsoft Learning Competency Member, or such other entity as Microsoft may designate from time to time.
2. "Authorized Training Session" means the instructor-led training class using Microsoft Instructor-Led Courseware conducted by a Trainer at or through an Authorized Learning Center.
3. "Classroom Device" means one (1) dedicated, secure computer that an Authorized Learning Center owns or controls that is located at an Authorized Learning Center's training facilities that meets or exceeds the hardware level specified for the particular Microsoft Instructor-Led Courseware.
4. "End User" means an individual who is (i) duly enrolled in and attending an Authorized Training Session or Private Training Session, (ii) an employee of an MPN Member (defined below), or (iii) a Microsoft full-time employee, a Microsoft Imagine Academy (MSIA) Program Member, or a Microsoft Learn for Educators – Validated Educator.
5. "Licensed Content" means the content accompanying this agreement which may include the Microsoft Instructor-Led Courseware or Trainer Content.
6. "Microsoft Certified Trainer" or "MCT" means an individual who is (i) engaged to teach a training session to End Users on behalf of an Authorized Learning Center or MPN Member, and (ii) currently certified as a Microsoft Certified Trainer under the Microsoft Certification Program.
7. "Microsoft Instructor-Led Courseware" means the Microsoft-branded instructor-led training course that educates IT professionals, developers, students at an academic institution, and other learners on Microsoft technologies. A Microsoft Instructor-Led Courseware title may be branded as MOC, Microsoft Dynamics, or Microsoft Business Group courseware.
8. "Microsoft Imagine Academy (MSIA) Program Member" means an active member of the Microsoft Imagine Academy Program.
9. "Microsoft Learn for Educators – Validated Educator" means an educator who has been validated through the Microsoft Learn for Educators program as an active educator at a college, university, community college, polytechnic or K-12 institution.
10. "Microsoft Learning Competency Member" means an active member of the Microsoft Partner Network program in good standing that currently holds the Learning Competency status.
11. "MOC" means the "Official Microsoft Learning Product" instructor-led courseware known as Microsoft Official Course that educates IT professionals, developers, students at an academic institution, and other learners on Microsoft technologies.
12. "MPN Member" means an active Microsoft Partner Network program member in good standing.

13. "Personal Device" means one (1) personal computer, device, workstation or other digital electronic device that you personally own or control that meets or exceeds the hardware level specified for the particular Microsoft Instructor-Led Courseware.
  14. "Private Training Session" means the instructor-led training classes provided by MPN Members for corporate customers to teach a predefined learning objective using Microsoft Instructor-Led Courseware. These classes are not advertised or promoted to the general public and class attendance is restricted to individuals employed by or contracted by the corporate customer.
  15. "Trainer" means (i) an academically accredited educator engaged by a Microsoft Imagine Academy Program Member to teach an Authorized Training Session, (ii) an academically accredited educator validated as a Microsoft Learn for Educators – Validated Educator, and/or (iii) a MCT.
  16. "Trainer Content" means the trainer version of the Microsoft Instructor-Led Courseware and additional supplemental content designated solely for Trainers' use to teach a training session using the Microsoft Instructor-Led Courseware. Trainer Content may include Microsoft PowerPoint presentations, trainer preparation guide, train the trainer materials, Microsoft One Note packs, classroom setup guide and Pre-release course feedback form. To clarify, Trainer Content does not include any software, virtual hard disks or virtual machines.
2. **USE RIGHTS.** The Licensed Content is licensed, not sold. The Licensed Content is licensed on a **one copy per user basis**, such that you must acquire a license for each individual that accesses or uses the Licensed Content.
- 2.1 Below are five separate sets of use rights. Only one set of rights apply to you.
    1. **If you are a Microsoft Imagine Academy (MSIA) Program Member:**
      1. Each license acquired on behalf of yourself may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.
      2. For each license you acquire on behalf of an End User or Trainer, you may either:
        1. distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User who is enrolled in the Authorized Training Session, and only immediately prior to the commencement of the Authorized Training Session that is the subject matter of the Microsoft Instructor-Led Courseware being provided, **or**
        2. provide one (1) End User with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**
        3. provide one (1) Trainer with the unique redemption code and instructions on how they can access one (1) Trainer Content.
      3. For each license you acquire, you must comply with the following:
        1. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,
        2. you will ensure each End User attending an Authorized Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Authorized Training Session,
        3. you will ensure that each End User provided with the hard-copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End

User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,

4. you will ensure that each Trainer teaching an Authorized Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Authorized Training Session,
5. you will only use qualified Trainers who have in-depth knowledge of and experience with the Microsoft technology that is the subject of the Microsoft Instructor-Led Courseware being taught for all your Authorized Training Sessions,
6. you will only deliver a maximum of 15 hours of training per week for each Authorized Training Session that uses a MOC title, and
7. you acknowledge that Trainers that are not MCTs will not have access to all of the trainer resources for the Microsoft Instructor-Led Courseware.

**2. If you are a Microsoft Learning Competency Member:**

1. Each license acquire may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.
2. For each license you acquire on behalf of an End User or MCT, you may either:
  1. distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User attending the Authorized Training Session and only immediately prior to the commencement of the Authorized Training Session that is the subject matter of the Microsoft Instructor-Led Courseware provided, **or**
  2. provide one (1) End User attending the Authorized Training Session with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**
  3. you will provide one (1) MCT with the unique redemption code and instructions on how they can access one (1) Trainer Content.
3. For each license you acquire, you must comply with the following:
  1. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,
  2. you will ensure that each End User attending an Authorized Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Authorized Training Session,
  3. you will ensure that each End User provided with a hard-copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,

4. you will ensure that each MCT teaching an Authorized Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Authorized Training Session,
5. you will only use qualified MCTs who also hold the applicable Microsoft Certification credential that is the subject of the MOC title being taught for all your Authorized Training Sessions using MOC,
6. you will only provide access to the Microsoft Instructor-Led Courseware to End Users, and
7. you will only provide access to the Trainer Content to MCTs.

**3. If you are a MPN Member:**

1. Each license acquired on behalf of yourself may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.
2. For each license you acquire on behalf of an End User or Trainer, you may either:
  1. distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User attending the Private Training Session, and only immediately prior to the commencement of the Private Training Session that is the subject matter of the Microsoft Instructor-Led Courseware being provided, **or**
  2. provide one (1) End User who is attending the Private Training Session with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**
  3. you will provide one (1) Trainer who is teaching the Private Training Session with the unique redemption code and instructions on how they can access one (1) Trainer Content.
3. For each license you acquire, you must comply with the following:
  1. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,
  2. you will ensure that each End User attending an Private Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Private Training Session,
  3. you will ensure that each End User provided with a hard copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,
  4. you will ensure that each Trainer teaching an Private Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Private Training Session,

5. you will only use qualified Trainers who hold the applicable Microsoft Certification credential that is the subject of the Microsoft Instructor-Led Courseware being taught for all your Private Training Sessions,
6. you will only use qualified MCTs who hold the applicable Microsoft Certification credential that is the subject of the MOC title being taught for all your Private Training Sessions using MOC,
7. you will only provide access to the Microsoft Instructor-Led Courseware to End Users, and
8. you will only provide access to the Trainer Content to Trainers.

**4. If you are an End User:**

For each license you acquire, you may use the Microsoft Instructor-Led Courseware solely for your personal training use. If the Microsoft Instructor-Led Courseware is in digital format, you may access the Microsoft Instructor-Led Courseware online using the unique redemption code provided to you by the training provider and install and use one (1) copy of the Microsoft Instructor-Led Courseware on up to three (3) Personal Devices. You may also print one (1) copy of the Microsoft Instructor-Led Courseware. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.

**5. If you are a Trainer.**

1. For each license you acquire, you may install and use one (1) copy of the Trainer Content in the form provided to you on one (1) Personal Device solely to prepare and deliver an Authorized Training Session or Private Training Session, and install one (1) additional copy on another Personal Device as a backup copy, which may be used only to reinstall the Trainer Content. You may not install or use a copy of the Trainer Content on a device you do not own or control. You may also print one (1) copy of the Trainer Content solely to prepare for and deliver an Authorized Training Session or Private Training Session.

2. If you are an MCT, you may customize the written portions of the Trainer Content that are logically associated with instruction of a training session in accordance with the most recent version of the MCT agreement.
3. If you elect to exercise the foregoing rights, you agree to comply with the following: (i) customizations may only be used for teaching Authorized Training Sessions and Private Training Sessions, and (ii) all customizations will comply with this agreement. For clarity, any use of "customize" refers only to changing the order of slides and content, and/or not using all the slides or content, it does not mean changing or modifying any slide or content.

- 2.2 **Separation of Components.** The Licensed Content is licensed as a single unit and you may not separate their components and install them on different devices.
- 2.3 **Redistribution of Licensed Content.** Except as expressly provided in the use rights above, you may not distribute any Licensed Content or any portion thereof (including any permitted modifications) to any third parties without the express written permission of Microsoft.
- 2.4 **Third Party Notices.** The Licensed Content may include third party code that Microsoft, not the third party, licenses to you under this agreement. Notices, if any, for the third party code are included for your information only.
- 2.5 **Additional Terms.** Some Licensed Content may contain components with additional terms, conditions, and licenses regarding its use. Any non-conflicting terms in those conditions and licenses also apply to your use of that respective component and supplements the terms described in this agreement.

3. **LICENSED CONTENT BASED ON PRE-RELEASE TECHNOLOGY.** If the Licensed Content's subject matter is based on a pre-release version of Microsoft technology ("Pre-release"), then in addition to the other provisions in this agreement, these terms also apply:
  1. **Pre-Release Licensed Content.** This Licensed Content subject matter is on the Pre-release version of the Microsoft technology. The technology may not work the way a final version of the technology will and we may change the technology for the final version. We also may not release a final version. Licensed Content based on the final version of the technology may not contain the same information as the Licensed Content based on the Pre-release version. Microsoft is under no obligation to provide you with any further content, including any Licensed Content based on the final version of the technology.
  2. **Feedback.** If you agree to give feedback about the Licensed Content to Microsoft, either directly or through its third party designee, you give to Microsoft without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft technology, Microsoft product, or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its technology, technologies, or products to third parties because we include your feedback in them. These rights survive this agreement.
  3. **Pre-release Term.** If you are an Microsoft Imagine Academy Program Member, Microsoft Learning Competency Member, MPN Member, Microsoft Learn for Educators – Validated Educator, or Trainer, you will cease using all copies of the Licensed Content on the Pre-release technology upon (i) the date which Microsoft informs you is the end date for using the Licensed Content on the Pre-release technology, or (ii) sixty (60) days after the commercial release of the technology that is the subject of the Licensed Content, whichever is earliest ("Pre-release term"). Upon expiration or termination of the Pre-release term, you will irretrievably delete and destroy all copies of the Licensed Content in your possession or under your control.
  4. **SCOPE OF LICENSE.** The Licensed Content is licensed, not sold. This agreement only gives you some rights to use the Licensed Content. Microsoft reserves all other rights. Unless applicable law gives you more rights despite this limitation, you may use the Licensed Content only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the Licensed Content that only allows you to use it in certain ways. Except as expressly permitted in this agreement, you may not:
    - access or allow any individual to access the Licensed Content if they have not acquired a valid license for the Licensed Content,
    - alter, remove or obscure any copyright or other protective notices (including watermarks), branding or identifications contained in the Licensed Content,
    - modify or create a derivative work of any Licensed Content,
    - publicly display, or make the Licensed Content available for others to access or use,
    - copy, print, install, sell, publish, transmit, lend, adapt, reuse, link to or post, make available or distribute the Licensed Content to any third party,
    - work around any technical limitations in the Licensed Content, or
    - reverse engineer, decompile, remove or otherwise thwart any protections or disassemble the Licensed Content except and only to the extent that applicable law expressly permits, despite this limitation.
  5. **RESERVATION OF RIGHTS AND OWNERSHIP.** Microsoft reserves all rights not expressly granted to you in this agreement. The Licensed Content is protected by copyright and other intellectual property

laws and treaties. Microsoft or its suppliers own the title, copyright, and other intellectual property rights in the Licensed Content.

6. **EXPORT RESTRICTIONS.** The Licensed Content is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the Licensed Content. These laws include restrictions on destinations, end users and end use. For additional information, see [www.microsoft.com/exporting](http://www.microsoft.com/exporting).
7. **SUPPORT SERVICES.** Because the Licensed Content is provided "as is", we are not obligated to provide support services for it.
8. **TERMINATION.** Without prejudice to any other rights, Microsoft may terminate this agreement if you fail to comply with the terms and conditions of this agreement. Upon termination of this agreement for any reason, you will immediately stop all use of and delete and destroy all copies of the Licensed Content in your possession or under your control.
9. **LINKS TO THIRD PARTY SITES.** You may link to third party sites through the use of the Licensed Content. The third party sites are not under the control of Microsoft, and Microsoft is not responsible for the contents of any third party sites, any links contained in third party sites, or any changes or updates to third party sites. Microsoft is not responsible for webcasting or any other form of transmission received from any third party sites. Microsoft is providing these links to third party sites to you only as a convenience, and the inclusion of any link does not imply an endorsement by Microsoft of the third party site.
10. **ENTIRE AGREEMENT.** This agreement, and any additional terms for the Trainer Content, updates and supplements are the entire agreement for the Licensed Content, updates and supplements.
11. **APPLICABLE LAW.**
  1. United States. If you acquired the Licensed Content in the United States, Washington state law governs the interpretation of this agreement and applies to claims for breach of it, regardless of conflict of laws principles. The laws of the state where you live govern all other claims, including claims under state consumer protection laws, unfair competition laws, and in tort.
  2. Outside the United States. If you acquired the Licensed Content in any other country, the laws of that country apply.
12. **LEGAL EFFECT.** This agreement describes certain legal rights. You may have other rights under the laws of your country. You may also have rights with respect to the party from whom you acquired the Licensed Content. This agreement does not change your rights under the laws of your country if the laws of your country do not permit it to do so.
13. **DISCLAIMER OF WARRANTY. THE LICENSED CONTENT IS LICENSED "AS-IS" AND "AS AVAILABLE." YOU BEAR THE RISK OF USING IT. MICROSOFT AND ITS RESPECTIVE AFFILIATES GIVES NO EXPRESS WARRANTIES, GUARANTEES, OR CONDITIONS. YOU MAY HAVE ADDITIONAL CONSUMER RIGHTS UNDER YOUR LOCAL LAWS WHICH THIS AGREEMENT CANNOT CHANGE. TO THE EXTENT PERMITTED UNDER YOUR LOCAL LAWS, MICROSOFT AND ITS RESPECTIVE AFFILIATES EXCLUDES ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.**
14. **LIMITATION ON AND EXCLUSION OF REMEDIES AND DAMAGES. YOU CAN RECOVER FROM MICROSOFT, ITS RESPECTIVE AFFILIATES AND ITS SUPPLIERS ONLY DIRECT DAMAGES UP TO US\$5.00. YOU CANNOT RECOVER ANY OTHER DAMAGES, INCLUDING CONSEQUENTIAL, LOST PROFITS, SPECIAL, INDIRECT OR INCIDENTAL DAMAGES.**

This limitation applies to

- anything related to the Licensed Content, services, content (including code) on third party Internet sites or third-party programs; and
- claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.

It also applies even if Microsoft knew or should have known about the possibility of the damages. The above limitation or exclusion may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential, or other damages.

**Please note: As this Licensed Content is distributed in Quebec, Canada, some of the clauses in this agreement are provided below in French.**

**Remarque : Ce le contenu sous licence étant distribué au Québec, Canada, certaines des clauses dans ce contrat sont fournies ci-dessous en français.**

**EXONÉRATION DE GARANTIE.** Le contenu sous licence visé par une licence est offert « tel quel ». Toute utilisation de ce contenu sous licence est à votre seule risque et péril. Microsoft n'accorde aucune autre garantie expresse. Vous pouvez bénéficier de droits additionnels en vertu du droit local sur la protection dues consommateurs, que ce contrat ne peut modifier. La ou elles sont permises par le droit locale, les garanties implicites de qualité marchande, d'adéquation à un usage particulier et d'absence de contrefaçon sont exclues.

**LIMITATION DES DOMMAGES-INTÉRÊTS ET EXCLUSION DE RESPONSABILITÉ POUR LES DOMMAGES.** Vous pouvez obtenir de Microsoft et de ses fournisseurs une indemnisation en cas de dommages directs uniquement à hauteur de 5,00 \$ US. Vous ne pouvez prétendre à aucune indemnisation pour les autres dommages, y compris les dommages spéciaux, indirects ou accessoires et pertes de bénéfices.

Cette limitation concerne:

- tout ce qui est relié au le contenu sous licence, aux services ou au contenu (y compris le code) figurant sur des sites Internet tiers ou dans des programmes tiers; et.
- les réclamations au titre de violation de contrat ou de garantie, ou au titre de responsabilité stricte, de négligence ou d'une autre faute dans la limite autorisée par la loi en vigueur.

Elle s'applique également, même si Microsoft connaissait ou devrait connaître l'éventualité d'un tel dommage. Si votre pays n'autorise pas l'exclusion ou la limitation de responsabilité pour les dommages indirects, accessoires ou de quelque nature que ce soit, il se peut que la limitation ou l'exclusion ci-dessus ne s'appliquera pas à votre égard.

**EFFET JURIDIQUE.** Le présent contrat décrit certains droits juridiques. Vous pourriez avoir d'autres droits prévus par les lois de votre pays. Le présent contrat ne modifie pas les droits que vous confèrent les lois de votre pays si celles-ci ne le permettent pas.

Revised April 2019



# Contents

■	<b>Module 0 Introduction</b>	1
	Introduction to Power Platform Solution Architect	1
■	<b>Module 1 Becoming a Solution Architect</b>	3
	Getting to know your customer	3
	Module Summary	5
■	<b>Module 2 Conceptualizing the design from requirements</b>	7
	Design from requirements	7
	Module Summary	11
■	<b>Module 3 Project governance and working as a team</b>	13
	Solution architect's role in project governance	13
	Reviewing others work	22
	Module Summary	24
■	<b>Module 4 Microsoft Power Platform architecture</b>	25
	Microsoft Power Platform architecture topics	25
	Module Summary	46
■	<b>Module 5 Data modeling</b>	47
	Model data for Microsoft Power Platform solutions	47
	Data modeling exercise	68
	Module Summary	70
■	<b>Module 6 Analytics and artificial intelligence</b>	71
	Evaluate Microsoft Power Platform analytics and AI	71
	Module Summary	96
■	<b>Module 7 Power Apps architecture</b>	97
	Explore Power Apps architecture	97
	Power Apps architecture exercise	116
	Module Summary	118
■	<b>Module 8 Application lifecycle management</b>	119
	Plan application lifecycle management	119
	Module Summary	135
■	<b>Module 9 Power Automate architecture</b>	137
	Explore Power Automate architecture	137

Power Automate exercise .....	151
Module Summary .....	153
<b>Module 10 Security modeling .....</b>	<b>155</b>
Model security for Power Platform solutions .....	155
Security modeling exercise .....	179
Module Summary .....	181
<b>Module 11 Integration .....</b>	<b>183</b>
Implement integrations with the Power Platform .....	183
Module Summary .....	200
<b>Module 12 Dynamics 365 applications architecture .....</b>	<b>201</b>
Explore Dynamics 365 application architecture .....	201
Evaluate requirements for first-party apps .....	220
Module Summary .....	222
<b>Module 13 Power Virtual Agents architecture .....</b>	<b>223</b>
Explore Power Virtual Agents architecture .....	223
Evaluate chatbot requirements exercise .....	243
Module Summary .....	244
<b>Module 14 Robotic Process Automation .....</b>	<b>245</b>
Explore Robotic Process Automation .....	245
Module Summary .....	256
<b>Module 15 Testing and go live .....</b>	<b>257</b>
Prepare for testing and go live .....	257
Module Summary .....	266

## Module 0 Introduction

### Introduction to Power Platform Solution Architect

#### Introduction

Welcome to Microsoft Power Platform Solution Architect. The Microsoft Power Platform enables organizations to optimize their operations by automating routine tasks and providing a range of solutions that enable digital transformation. A Microsoft Power Platform Solution Architect leads successful implementations and focuses on how solutions address the broader business and technical needs of organizations. The Solution Architect is a key member of the overall project team.

A solution architect should have functional and technical knowledge of Microsoft Power Platform, Dynamics 365 apps, related Microsoft cloud solutions, and other third-party technologies. A key task of the solution architect is solution envisioning. Essentially, this task involves looking at the problem and identifying which parts can take advantage of one of the Dynamics 365 apps and which parts need to be built by using Microsoft Power Platform or Microsoft Azure. Traditionally, a development-focused architect would start with custom development and low-level Microsoft Azure services. A business application-focused solution architect will instead start with Dynamics 365 and Microsoft Power Platform and then use Microsoft Azure to address any gaps.

This course will include several group exercises where students will work through challenges that Solution Architects face on projects. Modules 1, 2 and 3 content can be found on Microsoft Learn at **Architect solutions for Dynamics 365 and Power Platform<sup>1</sup>**

<sup>1</sup> <https://docs.microsoft.com/en-us/learn/patterns/become-solution-architect/>



# Module 1 Becoming a Solution Architect

## Getting to know your customer

### Exercise Overview

Split into groups of 3-5 students. Select a company to research. Choose a large, publicly traded, company. The groups will then share their findings and discuss the process of researching. Assume you need this information to bring with you to a meeting. The meeting could be the first meeting with the potential customer; it could be a meeting with your peers to discuss a potential deal.

### Learning Objectives

After completing the exercise, you will be able to:

- Find information about potential customers
- Evaluate the value of this information
- Understand how others might complete the same tasks

### Exercise 1: Learn about the customer

Find the following information regarding your selected company. Make sure to take notes so you can share what you've found with the larger group. It is not necessary to gather ALL of the items listed here. This is an example list of key details that can help you better know your customers.

#### People

Identify three separate c-level executives.

1. Name
2. Contact info and/or social media
3. Prior work

4. (Bonus) School background

**Corporate**

Identify stock symbol and value

Locate earnings report/call

Identify key divisions or lines of business

Identify headquarters and office locations

**Culture/News**

Read recent company press release

Locate social media accounts-official and unofficial

Locate what others say about company, news and social. Identify top pain points.

**Technology**

Do they have any preferred cloud providers?

Any publicly stated technology preferences?

# Module Summary

## Recap

In this module, we covered:

- Defining a Solution Architect
- Role of a Solution Architect on projects
- Project methodology
- Getting to know your customer



## Module 2 Conceptualizing the design from requirements

### Design from requirements

#### Exercise Overview

During this exercise, you will be reviewing project details for Contoso. This includes a set of requirements gathered from customer workshop sessions. You will use this information to design a high-level architecture and complete a fit gap analysis.

#### Learning Objectives

After completing the exercise, you will be able to:

- Perform evaluation of requirements to decide which approach to take
- Develop a high-level architecture for a set of requirements
- Perform a gap analysis from the Dynamics 365 apps used and the requirements

#### Exercise 1: Evaluate Project Details

Open and review the Project Details.docx file.

This contains the following information:

- Project Scenario and Overview
- Key Facts
- Requirements

As you review the document your team should take notes as you go and start to develop a proposed design to meet the requirements.

## Exercise 2: Complete the Project Workbook

Open and complete the worksheets in the Project Workbook.xlsx file.

Complete each of the following tabs:

- **High-level architecture diagram** – Create a simple high-level architecture diagram. This can be a picture of a whiteboard, a photo of a paper drawing. Simply paste it into the worksheet tab.
- **Dynamics 365 apps used** – Document which apps you used and a brief description how it will be used.
- **Fit Gap** – For each requirement complete the priority, level of effort, category and implementation details.

Your work will be reviewed by another team in a future exercise, so provide the level of detail you would want from the team you review.

## Scenario

Contoso Services provides after-market support for various manufacturer's smart beds. They use a proprietary solution today to manage their customers and their support ticket requests. They would like to move to a solution built on Microsoft Power Platform.

Some of their support requests can be handled quickly with a phone call or with a quick email response of a knowledge base article. Some may go on for days (or weeks) as the technicians dig in and research, then solve, the customers' problems. For complex challenges they dispatch technicians onsite to resolve the problem.

Today most of their support is reactive to a customer calling in. They would like to be able to become more proactive.

There are three type of cases that can come into Contoso's service center.

1. Items from customers without support agreements.
2. Items from preferred level support customers.
3. Items from premier level support customers.

Request for support can come from several sources: email, phone, online chat, social media, and so on. Requests for support coming in need to be evaluated and routed accordingly based on the requirements and policies. Some of the manufacturers have apps for customers that communicate with the beds and can then communicate the need for support to Contoso.

Contoso receives a data feed nightly from the manufacturers they support with customer information including bed serial number, bed configuration and ongoing telemetry of bed hardware wear indicators.

## Key Facts

FACT	Deteails
Number of solution users	2,500

FACT	Deteails
Number of service center agents	500
Number of field technicians	100 employees, 500 contractors
Locations	United States, United Kingdom and India
Number of customers	1.5 million
Contoso founded	2002
Contoso board of directors	9 members
Devices	Some users work with a single monitor and 1024x768 screen resolution

## Requirements

ID	Description
1	As the system, a help request shall be routed to the named team accordingly based on several factors including their service level, and type of incident so that the correct group can address the issue.
2	As a support staff, I should be able to see a list of open help requests so I can pick one to work on for which I am qualified.
3	As a customer, I don't want to wait on hold so should I be able to open a new request from Contoso's website using an interactive chat.
4	As a support staff, I need to be able to search knowledge base articles so that I can send them to customers.
5	As the system, I should process the nightly data feed from the manufacturers and keep track of the customer and bed configuration so requests can properly be associated.
6	As a support staff, I should be able to schedule a technician for onsite service.
7	As a support staff, I should see a list of customers to contact to schedule proactive maintenance based on insights from the bed wear indicators sent to us by the manufacturer so I can provide proactive service to ensure happy customers.
8	As a field technician, I should be able to view my next scheduled customer visit and receive details of the problem and directions if needed so that I can resolve their bed problem.
9	As a support staff, I need to be able to sell a support agreement to customers that didn't buy one with their original bed purchase so I can get their support request assigned to be resolved.

ID	Description
10	As a customer, I should receive an email when a support staff has closed my request so I can provide feedback on the experience.
11	As a support manager, I need to be able to see any support requests that are outside of the SLA for the customer so I can help prioritize the requests.
12	As a customer, I should be able to see the status of my support requests on Contoso's website so that I know what is happening to resolve my problem.
13	As a support manager, I want to receive a copy of any customer communication that the customer seems upset so that I can get involved to help resolve the problem.
14	As a support staff, I need to be able to see the remaining time I have to meet the SLA so that I'm not in trouble with my manager.
15	As the system, I need to track when a support request was opened, number of days open and resolved.
16	As a customer, I need to be able to take ownership of a bed, and support agreement when I acquire a used bed from a private buyer so that I can get the benefits of ownership.
17	As a support staff, regardless of my location, I should have acceptable response time for page loads.
18	As a C-level executive, I need to see real time metrics on the performance of the support team so I know what's going on.

# Module Summary

## Recap

In this module, we covered:

- How to lead the requirement collection effort
- Using fit gap analysis
- Pillars of good architecture
- Blueprinting the solution architecture



## Module 3 Project governance and working as a team

### Solution architect's role in project governance

#### Introduction to project governance

A Microsoft Power Platform Solution Architect plays a key role on a project. The solution architect will end up implementing governance, change processes and monitoring the project progress.

#### How can the solution architect help?

The solution architect specifically can make sure that requirements are in scope and nothing is over-promised. The Solution Architect therefore will need to manage the project's scope and keep the project on track. This sounds easy but in practice, requires the solution architect to be proactive in managing the project even when there is a project manager involved.

Most projects fail not due to the technical capabilities of team members but in failing to manage the scope of the project or in not addressing issues as they arise.

During the project the solution architect is a key member of a governance team. As scope or requirements change they are typically involved in review and impact assessment and are on the front lines for allowing scope creep to happen and to prevent it from happening. The solution architect may not have a seat at the governance table, but their impact will always be there.

The solution architect can help project governance by:

- Being a key member of the governance team.
- Helping manage changes in scope and requirements.
- Looking out for issues in the requirements including regulatory, compliance, auditing, and security.
- Not over-promising and helping to encourage small steps towards a big goal.

The solution architect will be seen by other project members as the technical leader of a project. The other team members will look to the Solution Architect as to how they handle changes in scope and how they should perform their own duties in relation to governance of the project.

## Keeping a project on track

Consider the projects you have had experience of as a project team member and answer these questions:

- Did the project have a governance process?
- If there was a governance process, was it followed?
- Were risks documented and were risks mitigated?
- Was there a change control process?
- How effective were these processes and could they have been improved?

These are all issues that the solution architect may have to address. Before, we get into this we need to define what we mean by project governance.

## Project governance

Project governance is not project management although there is some overlap. Project governance is the framework within which decisions about the project are made.

Project governance is a critical element of project and if performed badly, or not at all, will lead to the project's failure.

[!NOTE]

It is better to have any project governance process than none at all!

## What do we mean by governance in a project

Projects are not fixed. During a project, challenges will arise that will need to be dealt with. If ignored, then what can be a small question can turn into a major problem. If a project focuses too much on things as they happen the project can fail to move forwards. Having a project governance process in place assists a project handle concerns raised.

Project governance is about primarily about decision making within the project:

- Authority: Who makes the decision? Do they have the experience and expertise to make the right decision?
- Process: How are decisions made? How do you balance impact analysis against the need for timely decisions?
- Evidence: What information is required to make a decision? How do you perform analysis of the impact?

Any governance process should be inline with contractual terms for changes agreed to by customer. Ideally, in the Statement of work for a project, the decision making process should be defined with the ultimate authority named.

Often, your customer, whether that is internal or external, will have their own governance process but this process may have gaps or may not be relevant to the Microsoft Power Platform. Projects implemented using the Microsoft Power Platform can be very different to projects that customers have experienced before and the existing governance process may not be relevant e.g., in the speed of decision making or the level of information required. The solution architect needs to ensure the governance process meets the needs of a Microsoft Power Platform project. A solution architect will either need to adopt the existing process or build a hybrid process.

## [!IMPORTANT]

If no process is in place, the Solution Architect should push for the creation of a governance process or create one themselves.

## Governance processes

At a minimum a project should assess:

- Risks: Documentation of risks, evaluation of risks, and risk mitigation implemented.
- Issues: Documentation of issues, and a process for evaluation of issues.
- Changes: Documentation of changes, and a formal change control process.

## [!NOTE]

This module does not cover how to implement a governance process, but is concerned with making the solution architect aware that they must ensure there is a governance process in place for the project that meets the needs of the project.

## What was your experience of project governance

Consider the projects you have had experience of as a project team member and answer these questions:

- How were issues logged?
- Who was allowed to raise an issue?
- Were issues evaluated?
- Who performed the evaluation?
- How were changes raised?
- Were these process at the appropriate level for your project?
- How could you have improved them?

Let us look further into the solution architect's role in project governance.

## The solution architect's role in project governance

The solution architect will often have the role in the project of evaluating issues and changes due to their experience and expertise. However, change is easy especially in Microsoft Power Platform and the solution architect can become focussed on change rather than ensuring the project is moving forwards. The solution architect should be looking to other project team members to perform evaluation and analysis of issues and changes, under the guidance of the solution architect.

One of the challenges faced by a solution architect new to the role is the process of moving away from doing into leading and guiding. A solution architect needs to be available to other project team members and to support their growth.

## Involvement in defining governance

An issue with Microsoft Power Platform projects is that change is easy but lots of small changes and a poor governance process can kill a project. For instance, the solution architect needs to make sure the processes to evaluate change do not take longer than implementing the change itself.

It is imperative that the solution architect is involved in defining the processes and procedures for governance of the project so that they are appropriate to the Microsoft Power Platform technologies used and do not cause unnecessary overhead.

## Providing actionable feedback

The solution architect often sits between the customer and team members. The solution architect needs to provide feedback to both parties. Primarily feedback should be provided to help shape the solution.

Feedback can happen as early as the Request for Proposal (RFP)/Statement of Work (SOW) creation. However, feedback should be performed on an ongoing basis throughout the project.

The solution architect is responsible for keeping feedback constructive and actionable.

## Handling bad news

Sometimes, the solution architect has to provide feedback about something that may not be well received. Bad news does not get better with time. The solution architect must speak up early and share bad news.

Below are some examples of bad news that should not be held back by the solution architect:

- The cost of user licenses will increase by 87% if we move ahead with that requirement as written.
- That feature is being deprecated.
- With the added relationship the data import will now take 30 days.
- Data migration has identified 200 new fields and from this data three new undocumented processes have been discovered.

## Keep it actionable



The solution architect needs to keep feedback, especially bad news, actionable. There is little point in saying "Something is amiss here"; there is no clear call to action and the recipient is left trying to figure out what the problem is. Instead, the solution architect should make a clear problem statement and provide references and the potential impact on the project.

Consider projects that you have been involved in: how was bad news handled and what effect did it have on the project?

## Helping them reach the same conclusion

Appealing to authority does not work. Although, as the solution architect may have the most experience and expertise, you need to help the project team members and customer along the journey to the resolution to an issue. Simply saying "That won't work" is likely to put someone on the defensive. The solution architect should always be constructive and avoid saying "No" too often; offer options instead or negotiate the requirements.

You should ask leading questions such as "Will that cause 1,000,000 Power Automate cloud flows to be run with that configuration?" to encourage them to think about the impact of what they are proposing. Remember, the person may not have the overall view of the project that the solution architect has.

If you have concerns about a proposed solution or a change, you need to highlight your concern but encourages them to think through and resolve the concern.

Reviewing the work of others is a key task for the solution architect. There is a fine line between reviewing what someone has done and do the work yourself. When reviewing the work of others, the solution architect needs again to be constructive. The solution architect should provide suggestions on where to look for answers. For example, when it is not clear that a solid design is proposed, the solution architect might encourage the creation of a Proof of Concept (PoC) or other tests to validate the proposed solution.

The essence of the role of a solution architect is to constantly talk to the people involved in the project to ensure that the vision of the project is achieved. Poor solution architects hide away from the team and spend their time updating their architecture designs.

Let us now look at techniques a solution architect can use on a project.

## Techniques for keeping a project on track

The solution architect should ensure that the project is always moving forwards towards the project's goals. The solution architect should establish good practices that ensure the customer and team members know where the project is and are aware of any issues that affect the project's progress.

## Recipes for failure

Here are some common areas of failure in a project:

- Not documenting assumptions
- Not doing risk management
- Excessive analysis and impact assessment
- Over-promising
- Designing with incorrect assumptions or requirements
- Organizational politics
- Not having buy in from senior management
- Not being able to have a complete enterprise vision

<sup>^</sup>This list is not exhaustive and many other things can cause project failure.

## Recipes for success

Good projects have the following characteristics:

- Knowing what done looks like - from the customer's perspective not yours.
- Having a change control board to manage change.
- Manage risk proactively.
- Being more agile.
- Performing Project checkpoints – How are we doing?
- Having retrospectives and evaluating lessons learned - How did we do?

^ This list is not exhaustive and many more things that go into making a project successful.

## Project checkpoints

The solution architect should insist on a way to measure how things are going on the project. The solution architect should implement regular progress reporting and have checkpoints (tollgates) at critical points in the project.

For example, the solution architect can use simple Red, Yellow, Green coding.

### 2 Project Rating Summary

<Use trend indicators ('+' '=' '-') and specify either Red, Yellow, or Green for each category.>

Project Area	Red	Yellow	Green
Schedule/ Milestones / Deliverables		+	
Project Burn rate/schedule		+	
Technical / Other Issues/ Risks		+	
Staffing		+	
Accounts Receivable			=
Overall Rating:		+	

This screenshot highlights the Red, Yellow, Green concept. It provides a quick glance visual health of the project and what shape the areas are in. Good teams raise issues early, mitigate the issue, and escalate to red without last minute surprises.

This exact method doesn't have to be used, but the solution architect should agree on a simple health check process.

## Retrospectives

The solution architect should evaluate how the project performed at regular points in the project. There is little point waiting to the end of the project to review how the project went as that does not allow the team members to put things right. It is important to gain an understanding of how does the customer think you did so you can address any concerns the customer might have.

The solution architect should establish checkpoints to look back at progress. Before the checkpoint, the solution architect should gather feedback from inside and outside the project team.

During the checkpoint meeting, the solution architect should push for no fault discussions. The aim is to turn failures into goals for the next checkpoint review.

Let us now look further at how the solution architect works as part of a team.

## Working as a team

The solution architect will generally be the most experienced person on a project and may be the most skilled in many of the tasks in a project, but a solution architect cannot do everything alone. The solution architect needs to break down the work and allocate the work to the most appropriate team member.

The solution architect owns the overall vision of the technical solution for the project and needs to communicate that vision to the team.

## Being a role model

The solution architect is very visible in the project to other team members and will naturally be seen as a role model. The engagement manager and project manager lead the project, but often the solution architect is looked at as the "real" leader and can set the tone and pace for how the team works. The solution architect is therefore responsible for setting an example for others to follow on the project.

The solution architect may further be responsible for increasing the skills and experience of project team members, mentoring them, and encouraging them to take more responsibility and gain new skills.

## Assessing the teams' skills

A solution architect establishes the breakdown that will be used for the functional and technical design. As a solution architect it is not often that you get to choose your team and not everyone on a team can be a rock star performer.

The solution architect must learn how to assess skills of team and their weaknesses so as best to assign tasks to team members. The solution architect does not ask for a resume to learn how much Microsoft Power Platform they know, but they should have a set of probing "discussion" questions to test knowledge for example when should they use Managed instead of Unmanaged solutions, or when should they use Business rules or Power Automate cloud flows.

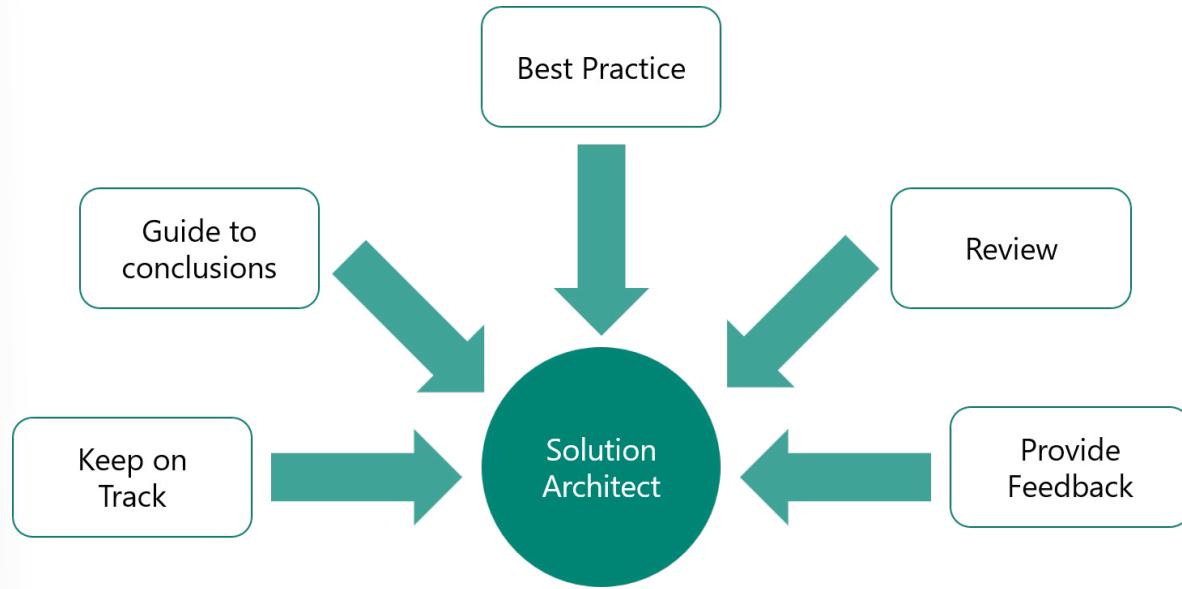
The solution architect could also assign small tasks or Proof of Concepts to team members and assess results. The solution architect should encourage the team member to explain their proposed solution so you can understand their thought processes. You may actually learn something.

The solution architect could create a skills matrix as the project evolves to enable project to leverage the strengths of the team to succeed.

## Staying on the same page

The solution architect often brings a lot prior knowledge and experience and is the person on the team who gets to see the "Big Picture". The solution architect should not assume that everyone else has the same skills, knowledge, experience, or understanding of the project.

Making assumptions and generalizations can be risky. A key role of the solution architect is to communicate the information to team members at the right time. Communication should be clear and relevant to each team member. Emails may not be enough and the solution architect should talk regularly with team members to make sure everyone is on the same page. Remember communication is two-way and to listen to the team members.



## Breaking down the work

Dividing work to assign to team members is a task undertaken by the solution architect. The solution architect should look for logical divisions of functionality in the application e.g., Customer Service vs Customer Acquisition. The solution architect should look for horizontal solution features such as document Management or computer telephony integration (CTI).

## Establishing standards

Standards are important on any project for the team to follow. There are some things in the Microsoft Power Platform that are harder to change later such as the schema prefix or table and column names.

The solution architect should define standards so that there is consistency for:

- Naming of components
- Column data types
- User Interface: The options for forms layouts, when to use multiple forms etc.
- Automation: When to use different automation options such as Power Automate cloud flows vs Classic workflow vs Business rules vs Client script
- Security: When to use roles, column security, hierarchical security etc.
- Development: for example early vs late bound, error handling etc.

The solution architect should make use of Microsoft documentation such as <https://docs.microsoft.com/powerapps/developer/model-driven-apps/best-practices/>, <https://docs.microsoft.com/powerapps/developer/data-platform/applications-best-practices-checklist> and <https://docs.microsoft.com/dynamics365/customerengagement/on-premises/developer/best-practices-sdk>.

## Working environments

The solution architect is responsible for defining how configuration and development will take place. This will depend on many factors including the skills of the team, if the team is co-located or dispersed remotely.

The development environment topology needs to be determined and some type of testing environment(s) will be needed. Remember, the Microsoft Power Platform does not provide version-controlled tracking by default.

Consider projects that you have been involved in: What else can the Solution Architect do to help teams work better together?

# Reviewing others work

## Exercise Overview

During this exercise, you will be reviewing the project workbook for the Contoso project created by another team of Solution Architects. Part of this exercise is deciding how best to manage your time. Make sure you find out from your instructor the time allotted.

## Learning Objectives

After completing the exercise, you will be able to:

- Perform evaluation of work produced by others effectively.
- Deliver constructive feedback to others.

## Exercise 1: Review Project Workbook Details and Prepare Feedback

You will be assigned another team of Solution Architects to work with and exchange feedback.

Exchange a digital copy of your project workbook from Module 2 where you architected a high-level solution for Contoso with the other team.

As a team, review the other team's workbook details and develop notes so you can go over the feedback with the other team later. Your notes should consider the following:

- High-level architecture diagram
  - Does it accurately represent the requirements?
  - How could it be improved?
- Dynamics 365 apps used
  - Did they use the appropriate Dynamics 365 apps for the requirements?
  - If you disagree with a decision, why?
- Fit Gap
  - Review the other team's choices, do any seem wrong?
- Overall assessment
  - What did the team do well?
  - What could the team have improved?

## Exercise 2: Provide feedback to the other team

Take turns with the other team providing feedback to each other based on your review.

As the team providing feedback:

- Keep the feedback constructive.
- Provide examples of how to improve the proposed architecture don't just say it's wrong
- Avoid coming across like you know everything!

As the team receiving feedback:

- Listen to the feedback, try not to interrupt.
- Try to not be defensive and listen to the ideas.
- Ask clarifying questions where the feedback isn't clear.

If you have time, identify one requirement that you think is a very poorly written requirement.

## Module Summary

### Summary

Building business solutions using the Power Platform is a team effort requiring coordination and planning to succeed. Solution architects are often the glue that holds a project team together.

In this module, you looked at the role of the solution architect in relation to project governance, including:

- Involvement in defining the project governance processes.
- Driving the project forward by implementing processes.
- Reviewing others work.
- Providing actionable feedback.

### Next steps

The next steps are to look at the architecture of the Microsoft Power Platform to learn the capabilities and how to use the strengths of the Microsoft Power Platform to design solutions to business problems.

## Module 4 Microsoft Power Platform architecture

### Microsoft Power Platform architecture topics

#### Introduction to the Microsoft Power Platform architecture

A Microsoft Power Platform Solution Architect is responsible for the overall design of the solution. Understanding the architecture of the Microsoft Power Platform, how it is designed, and its limits are very important when designing solutions based around the Microsoft Power Platform.

As you consider your solution's design, you need to understand the components and capabilities of the Microsoft Power Platform and how these control the design of your solution.

### Microsoft Power Platform Components

The Microsoft Power Platform is Microsoft's low-code, rapid business application development platform. The Microsoft Power Platform contains several different independent, yet closely related tools:

- Power Apps: Allows anyone to build custom web and mobile apps using low-code or no-code techniques.
- Power Automate: A cloud workflow tool to connect cloud and desktop applications together to automate business processes.
- Power BI: A self-service analytics service that allows users to gain insights into their data. Power BI allows data to be merged from various sources and to create models, visualizations, reports, and dashboards.
- Power Virtual Agents: Allows anyone to create chatbots without writing code from within a browser interface.



**Power BI**  
Business Analytics



**Power Apps**  
Application Development



**Power Automate**  
Workflow Automation



**Power Virtual Agents**  
Intelligent Virtual Agents

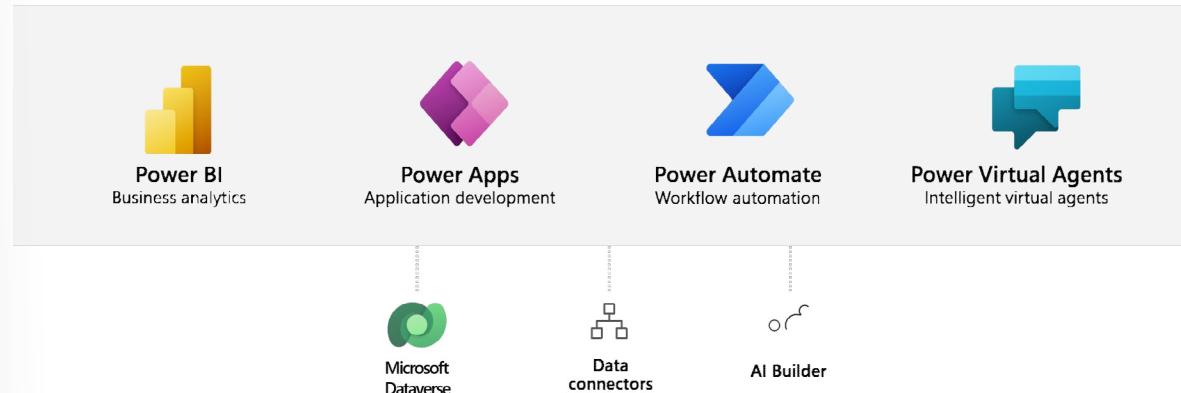
The Microsoft Power Platform includes a number of other components that support the Microsoft Power Platform tools that the solution architect can leverage:

- Microsoft Dataverse (formerly known as the Common Data Service): Provides a no-code environment to create tables, relationships, and business logic.
- Data Connectors: Connectors define the services and data sources that the Microsoft Power Platform tools can access.
- AI Builder: A set of AI model types that can use data in Dataverse to create, tailor, and train AI models that can be used by the other components of the Microsoft Power Platform.

The diagram below illustrates how the components of the Microsoft Power Platform are related.

## Microsoft Power Platform

One low-code platform that spans Office 365, Dynamics 365, Azure and standalone applications



[!NOTE]

This module describes the architecture of the Microsoft Power Platform. There are other modules that examine the details for Power Apps, Power Automate, Power BI, and Power Virtual Agents.

## Cloud

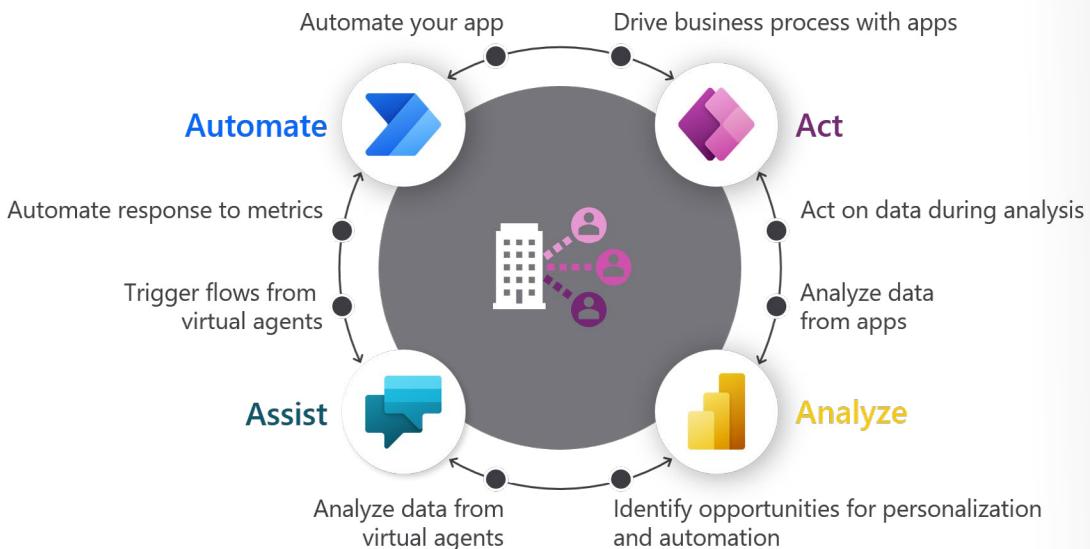
Microsoft Power Platform is a cloud based Software-as-a-Service (SaaS) service that is available from within an Azure Active Directory tenant. The Microsoft Power Platform is licensed through Microsoft 365 and secured by Azure Active Directory. The Microsoft Power Platform runs on Microsoft Azure and therefore is highly scalable and is available globally.

## Capabilities of Microsoft Power Platform

Typically solutions that a solution architect designs will make use of many of the components provided by the Microsoft Power Platform. The individual components provide great capabilities on their own but are more powerful when combined.

Power Apps allows users to act on data and drive business processes. Power Automate can automate apps and act on behalf of the user in response to metrics and events. Power BI can analyze the data captured by apps. Power Virtual Agents can assist users. Power Automate flows can be called from Power Apps, Power Virtual Agent actions, and Power BI alerts. Power BI dashboards and tiles can be displayed in Power App screens. Power Apps can be embedded in Power BI dashboards.

### Low-code tools and how they are used



## What do we mean by the platform

When most people refer to the Microsoft Power Platform, they are thinking of the four components, Power Apps, Power Automate, Power BI, and Power Virtual Agents. These tools allow you to build apps and create solutions.

Data is at the center of everything a business does today. With the Microsoft Power Platform, we can either connect to data where it lives using Connectors, or store data in Microsoft Dataverse. There are connectors for Office 365 services, such as email and SharePoint, Azure services, such as Azure SQL, as well as many non-Microsoft sources such as Twitter, SendGrid, Dropbox, and Mailchimp. Power Apps and Power Automate use the Connectors to access the data where it lives. Microsoft Dataverse is more than a database, it's a platform for building business apps and solutions.

Microsoft Dataverse controls security, implements logic, enables integrations, and has a wealth of capabilities that enable powerful business solutions to be created. These capabilities will be explored further in this module.

Before we can look at Dataverse, we first need to describe how Dataverse is accessed through environments.

## Environments

An environment is a space to store, manage, and share an organization's business data, apps, and flows. Administrators can create environments and control access to environments.

An environment has:

- Name: Name of the environment
- Location: Which Azure region the environment, its data, apps, and flows are stored in.
- Admins: Who can manage the environment.
- Security Group: Controls which users can access the environment.
- Apps: The apps created in the environment.
- Flows: The flows created in the environment.
- Bots: The chatbots created in the environment.
- Connectors: Custom connectors added to the environment.
- Gateways: On-premises gateways connected to the environment.
- Dataverse (optional): An instance of a Dataverse database.

Environments are containers that administrators can use to manage apps, flows, connections, and other assets; along with permissions to allow organization users to use the resources.

## Tenants

Microsoft 365 uses an Azure Active Directory tenant to control authentication and authorization. You do not need an Azure subscription to access the Azure Active Directory (AAD), the Microsoft 365 subscription permits access to the AAD portal (<https://aad.portal.azure.com>). Adding a user to Microsoft 365, adds the user to AAD. AAD tenants are located in an Azure region, normally the region of the user who created the tenant.

Environment are created within an Azure Active Directory (AAD) tenant. Access to the environment is authenticated by Azure AD.

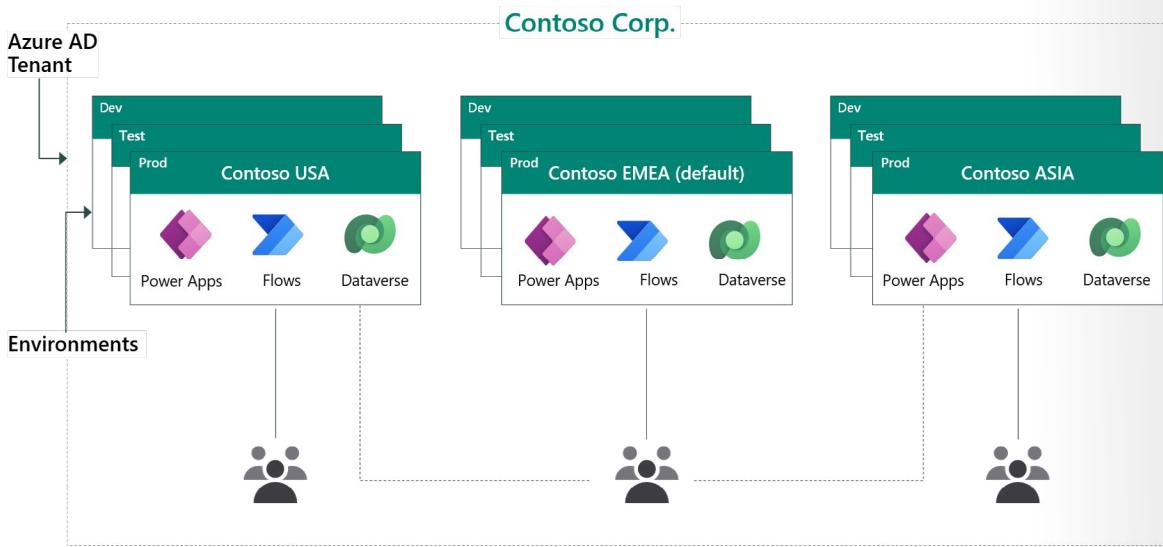
## Multiple environments

An environment is a container to separate apps that might have different roles, security requirements, or target audiences. How you choose to use environments depends on your organization and the apps you're trying to build. For example:

- You can choose to only build your apps in a single environment.
- You might create separate environments that group the test and production versions of your apps.
- You might create separate environments that correspond to specific teams or departments in your company, each containing the relevant data and apps for each audience.
- You might also create separate environments for different global branches of your company.

The following diagram illustrates how an organization may create multiple environments.

# Environments



Environments are a scope for lifecycle management and a scope for permissions.

## What is not contained in an environment

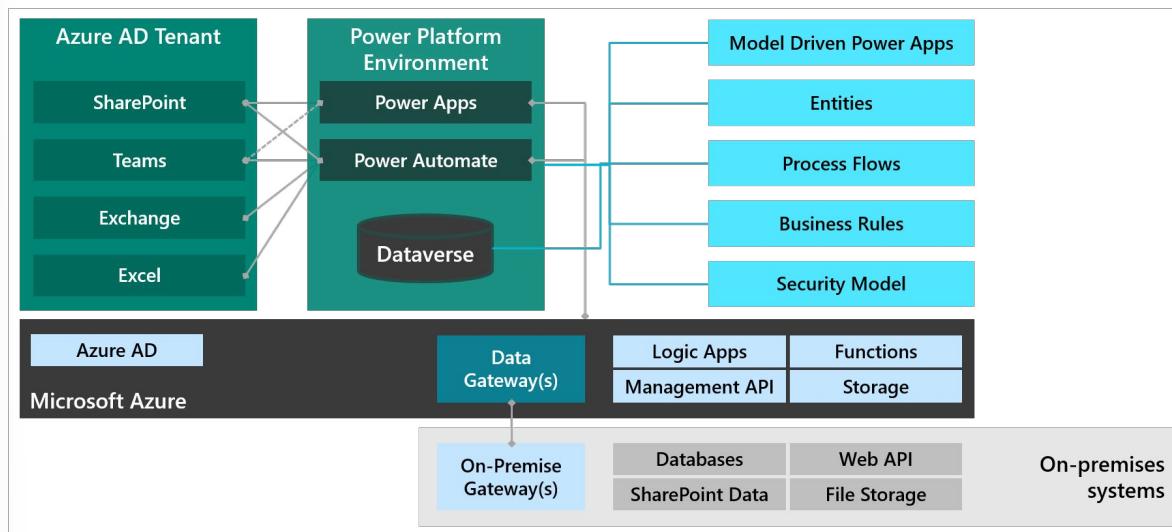
Environments contain the Microsoft Power Platform components such as apps, flows, and bots. But not everything you might build for your solution is contained in environments. Dynamics 365 customer engagement apps (Sales, Marketing, Customer Service, Field Service, and Project Operations) are contained in an environment. There are no limitations to the number of apps or flows that can be deployed within an environment.

An environment does not contain:

- Power BI workspaces, datasets, reports, or dashboards.
- Any Azure service deployed.
- Non-customer engagement Dynamics 365 apps (e.g. Finance or Business Central).

A solution architect need to consider how access to these other components is controlled.

The following diagram illustrates how an environment related to other components inside and outside the environment.

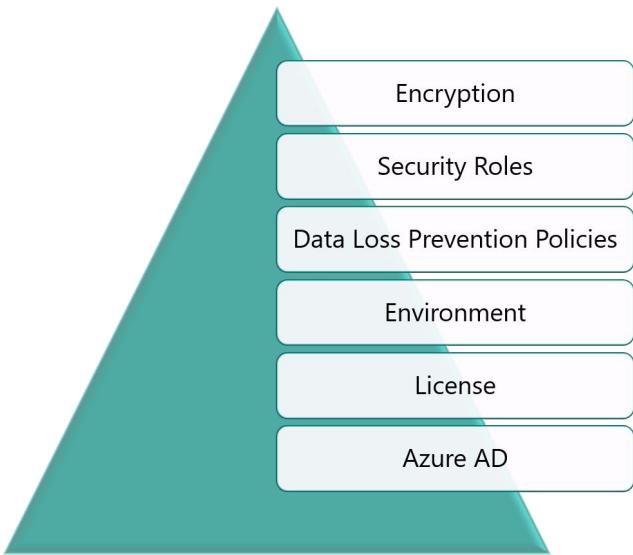


## Security layers

The Microsoft Power Platform uses multiple layers of security.

- **Azure AD:** User identities are authenticated by Azure Active Directory (AAD). With Azure AD we can restrict users using AAD features such as conditional access policies and enforce security through multi-factor authentication. Azure AD supports Single Sign-On (SSO) so users do not need to login separately to the Microsoft Power Platform. Guest access is supported for the Microsoft Power Platform.
- **Licensing:** Users require an appropriate license to access the Microsoft Power Platform.
- **Environments:** Azure AD Security groups control access to each environment.
- **Data Loss Prevention Policies:** DLPs restrict the use of connectors for environments and therefore the risk of data leakage. You can create cross-tenant inbound and outbound restrictions where you have a multi-tenant scenario.
- **Security Roles:** Access to tables and rows in Dataverse is controlled by Security Roles.
- **Encryption:** All data is encrypted at rest using SQL Server Transparent Data Encryption (TDE). All data is encrypted in transit using SSL.

The following diagram presents how security layers are applied for Microsoft Power Platform environments.



When creating an environment, the solution architect needs to decide the location of the environment.

## Environment and Data Location

Microsoft has a worldwide network of datacenters, that support the Microsoft Power Platform, set up around the world in 17 different locations. This level of localization helps organizations more easily meet data residency, sovereignty, and compliance requirements.

When creating an environment, you can specify the geographical location for the environment. The data for the environment will be held in the chosen geographical location.

[!NOTE]

An environment can be relocated after it has been created, but it is easier to select the correct region to begin with.

## Regions

Microsoft offers the option of choosing global, local, and sovereign clouds:

- Global cloud: A public internet cloud deployment option that lets customers access globally connected cloud services deployed from regional Microsoft datacenters. The global clouds are United States, Europe, and Asia Pacific.
- Local cloud: A cloud option that addresses local data residency requirements by letting all public cloud users within a country keep their data in the country. The local clouds are: Canada, Brazil, United Kingdom, France, Germany, India, Japan, Australia, United Arab Emirates, Switzerland, Republic of South Africa.
- Sovereign cloud: A cloud option that adheres to the strictest standards of operation. Data remains inside the country of residence at all times. Sovereign clouds are: US Government, China.

[!NOTE]

New locations are planned so the list above may not be complete. Check <https://docs.microsoft.com/power-platform/availability> for the latest information.

The following diagram shows the locations support for the Microsoft Power Platform.



## Data residency

There are several aspects that influence the choice of location for environments and data:

- Are the user geographically dispersed?
- What latency is acceptable?
- Does the organization have separate autonomous business units where data must not be shared?
- Are the required features available in the region?
- Are the security requirements to be considered?
- Are there specific data & compliance requirements?

If there are no other factors, you should select a location close to the majority of users that need to share data.

[!NOTE]

The location of the tenant (billing) can be different from location of an environment and its data.

## Compliance and Data Protection

The solution architect must consider regulatory and compliance requirements such as GDPR when deciding on a location for the environment and data.

Microsoft will not transfer customer data outside the selected Azure geographic location (geo) except when it is necessary for Microsoft to provide customer support, troubleshoot the service, or comply with legal requirements.

The Microsoft Power Platform stores information that is global in nature, such as user identities and profile information, in a datacenter located in the United States. All Microsoft Power Platform customer data, as well as the geo-redundant mirrors, is maintained within the selected geo.

## When to use multiple environments

In the last module we saw how the Microsoft Power Platform uses environments to separate development from production and to separate different user groups. When you further consider the location of users in a global organization there are other reasons to create multiple environments:

- To handle application lifecycle management (ALM) – Dev / Test / Prod.
- To isolate resources that are not used by same users or share data.
- To allow conflicting customizations between different regions or markets, or for compliance with legal and regulatory constraints.
- To keep environments close to users that don't share data.
- To isolate data that cannot be co-located.
- To provide places for makers to experiment.

Let us now look at working with data.

## Working with Data

Data is at the center of everything a business does today and powers the insights that can drive what it should do tomorrow. To thrive and grow, businesses need to capture, analyze, predict, present, and report data and do it all with a high level of agility.

With the Microsoft Power Platform you can either:

- Access data from the platform using Connectors.
- Store data in the platform using Microsoft Dataverse.

## APIs

The Microsoft Power Platform uses REST APIs to communicate between apps and data, and to perform management activities. REST APIs are built on open standards. You can compose HTTP requests for specific operations or use third-party libraries to generate classes for whatever language or platform you want.

Connectors and Dataverse both use REST API and OData to for requests and data.

## Dataverse vs Connectors

Connectors allow you to take advantage of the Microsoft Power Platform without moving your data. Power BI, Power Apps, and Power Automate all leverage connectors allowing you to analyze, act and automate across data and services.

With connectors:

- Can leverage existing data sources and services.
- Connect to more than 400 systems and services out of the box.
- Access data stored on-premises systems with the data gateway.
- Create custom connectors for internal and third party services.

### [!IMPORTANT]

Not all the public connectors support all the actions the underlying service supports, solution architects should investigate that the actions they require are supported otherwise a custom connector is required.

Connectors allow you to build your apps without having to migrate the data which can be time consuming or costly.

Microsoft Dataverse should be your data source of choice for new data stores, or where you want to make use of Dataverse capabilities that will be described below or you want to use features of the Microsoft Power Platform such as AI Builder or Portal Apps.

## Custom Connectors

Where there isn't a connector but the service has a REST API, a custom connector can be created. Creating a custom connector is straight forward with several different methods for defining the custom connector, including:

- Import the Open API definition to describe an existing API.
- Use Postman collection to describe an existing API.
- Create an Azure Functions and use Azure API Management (APIM).
- Create a custom API to define your own actions.

[!NOTE]

Many third party services publish an Open API definition for their full API.

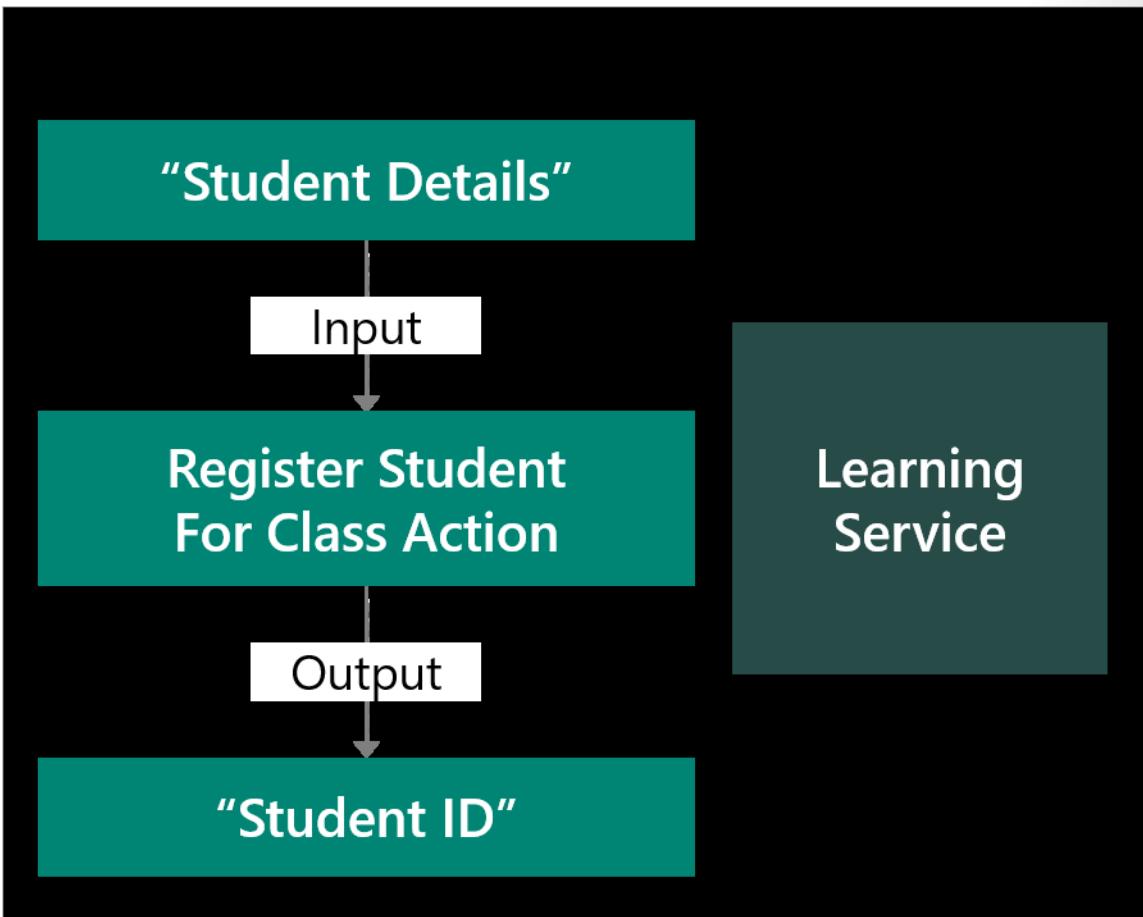
Custom Connectors supports OAuth, (including Azure AD), API Key, and basic auth.

Custom Connectors can be packaged and deployed with solutions.

Actions can make it easy for makers to invoke complex custom logic. Consider the processing to register a student for a class. The processing steps are:

- 1 Check if student exists
- 2 Create student if not exist
- 3 Lookup class by course code
- 4 Check if class over capacity
- 5 Check if student has class pre-requisites
- 6 Create class student record
- 7 Notify user of results

You could create a custom connector for the Learning service with an action called "Register Student for Class" that performs these steps as shown in the following image.



Makers without the custom connector would need to know the details of the service, by using a custom connector with an action this promotes good architectural design; loose coupling via a contract where the calling system does not need to know the details since they could change.

## Virtual tables

There is one further option for accessing data. Consider the requirement to access data from an external data source and combine it with data in Dataverse. Virtual tables allow you to create a virtual table in Dataverse, define an external data source, and map the table and columns onto the external data source.

There is an OData v4 Data Provider included with Dataverse that allows you to connect to an external OData v4 web service. There is a data provider for the SQL API of Azure Cosmos DB in preview.

[!NOTE]

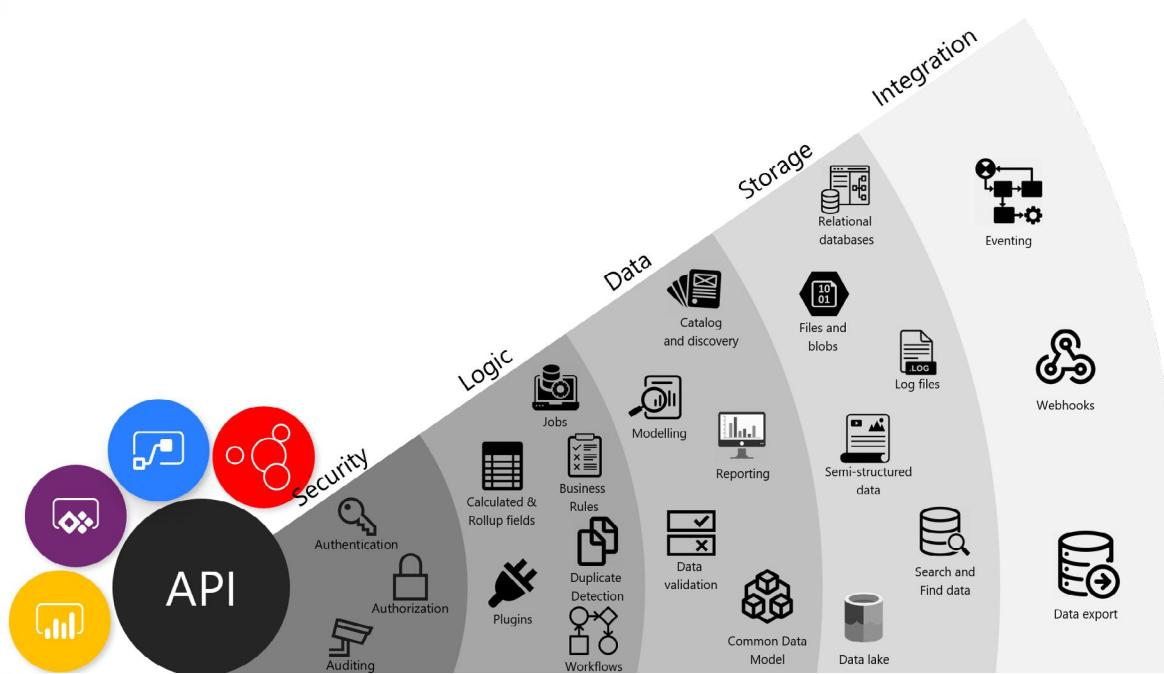
Dataverse requires that all tables have an ID attribute, this ID is known as a unique identifier and the value must be a guid. You can only map ID columns to external columns with the Edm.Guid data type. Hence the external web service must use a guid as its ID column.

Currently, virtual tables only permit read operations.

## Dataverse capabilities

Dataverse is more than just a database. Dataverse includes many features that make building business solutions easier and provide enhanced functionality.

The diagram below illustrates the capabilities provided by Microsoft Dataverse.



The solution architect needs to understand the capabilities of Microsoft Dataverse and how these can be applied when creating solutions.

Because Dataverse is built on Azure, it benefits from the Azure platform's powerful security technologies. Encryption of data, at rest and in transit, preserves confidentiality.

Dataverse uses Azure AD identity and access management mechanisms to help ensure that only authorized users can access the environment, data, and reports. Dataverse uses role-based security to group together a collection of privileges. These security roles can be associated directly with users, or they can be associated with Dataverse teams and business units. In Dataverse, individual rows can be shared on a one-by-one basis with another user. Because row-level control of access isn't adequate for some business scenarios, Dataverse has a column-level security feature to allow more granular control of security at the column level. Dataverse also includes security models that can be used for hierarchies: the manager hierarchy and the position hierarchy. The security mode that Dataverse provides is highly configurable.

There are several choices available for applying custom business logic in Dataverse including Business rules, Classic workflows, and Power Automate. Calculated columns and Rollup columns reduce the need for processing and custom code. Dataverse has built-in functionality for duplicate detection and to delete stale data.

All components created in Dataverse are held as Metadata. This means the components are discoverable and the properties for components are accessible to applications and tools. This metadata is used by Power Apps and Power Automate reducing the effort required to create apps and flows. For instance, the data type of a column and its range of possible values are held in metadata. For example, the Power Apps studio is able to access this metadata and automatically configure the control when a column is added to a form.

The data held in Dataverse is abstracted from the underlying data storage mechanism. The data could be stored in Azure SQL Elastic pools, Azure Storage, Cosmos DB, or Azure Data Lake. The maker does not have to concern themselves with the storage and access the data through the APIs provided by Dataverse. Microsoft decides how to store data different ways depending on type for files, images, and text.

Dataverse provides an event model for integrating with other systems, and there are both import and export data processing capabilities.

Dataverse utilizes Azure Search to enable data held in Dataverse to be searched.

Dataverse is not restricted to apps built with the Microsoft Power Platform, the APIs allow applications to be built with other tools such as Xamarin to build an external customer facing application.

## Why choose Microsoft Dataverse

Building out the data infrastructure to enable business insight can be both time consuming and expensive. The data originates from a variety of devices, applications, systems, services, and software as a service (SaaS). This large and growing number of sources often consists of multiple data technologies that store different types of data, expose different APIs, and use a mixture of security models. The developers needed to create these technologies can be expensive and hard to find. Developers often must have a deep understanding of how to deploy, configure, manage, and integrate these different data technologies.

Dataverse addresses these concerns with an easy to use, easy to manage, compliant, secure, scalable, and globally available SaaS data service. Dataverse empowers organizations to work with any type of data and any type of app, and use the data within it to gain insights and drive business action.

As part of Microsoft Microsoft Power Platform, Dataverse requires no or little code to be written, so it can easily be used by everyone from knowledge workers to professional developers.

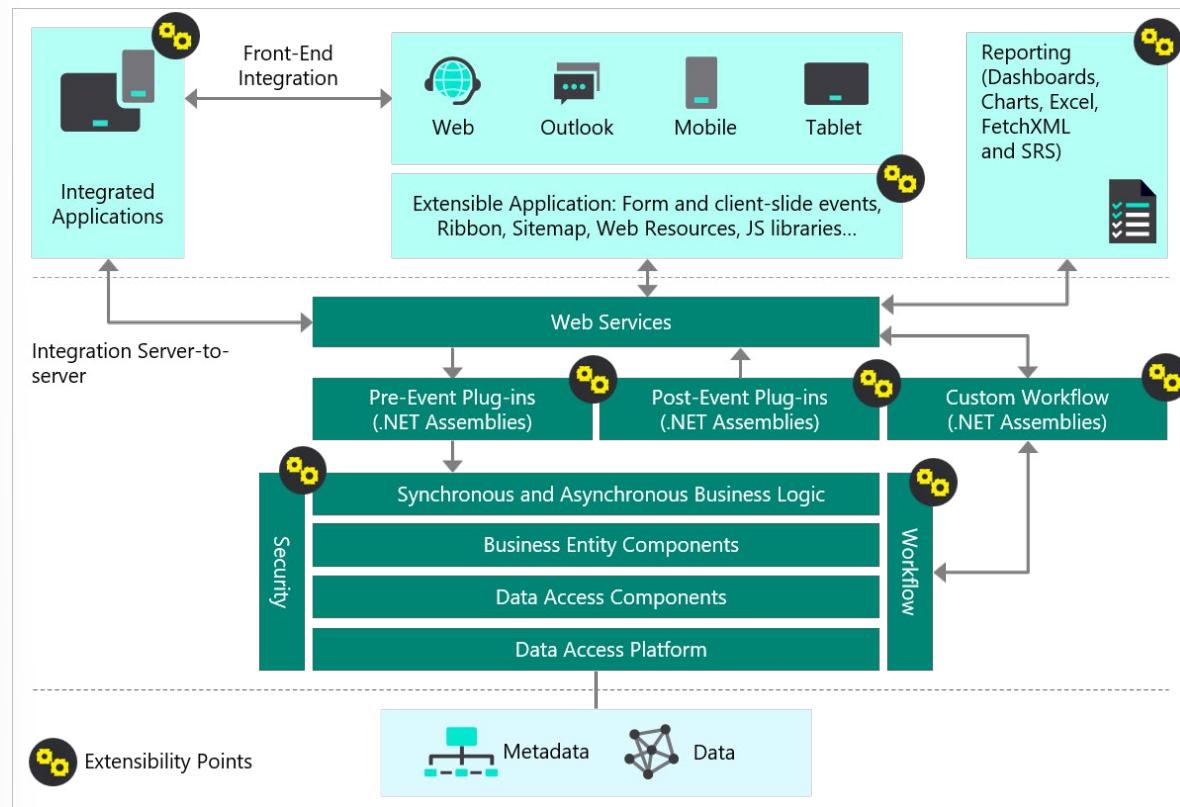
Standard and custom tables within Dataverse provide a secure and cloud-based storage option for your data. Tables let you create a business-focused definition of your organization's data for use within apps.

If you are not sure whether Dataverse is your best option, consider these benefits:

- Easy to manage: Both the metadata and data are stored in the cloud. You don't need to worry about the details of how they're stored.
- Easy to secure: Data is securely stored so that users can see it only if you grant them access. Role-based security allows you to control access to tables for different users within your organization.
- Access your data: – Data from your Microsoft Power Platform applications is also stored within Dataverse, allowing you to quickly build apps that use your data.
- Rich metadata: Data types and relationships are used directly within Power Apps.
- Logic and validation: Define calculated columns, business rules, workflows, and business process flows to ensure data quality and drive business processes.
- Productivity tools: Tables are available within the add-ins for Microsoft Excel to increase productivity and ensure data accessibility.

## Extensibility model

Everything in Microsoft Dataverse is exposed as an API call. All tools and apps all access via the Web API as shown in the following diagram.



Dataverse provides extension capabilities at every horizontal layers. As a solution architect, you need to understand the different types of extensibility paradigms. Dataverse has:

- A rich API for integrating with Web Services and OData.
- No code logic with Business Rules, Calculated columns, Rollup Columns, Business Process Flows and more.
- Developer support with full .NET SDK with access to the transactional pipeline
- Client-side scripting with TypeScript and JavaScript
- Integration with Azure Service Bus and Event Hub
- Custom actions
- Custom API

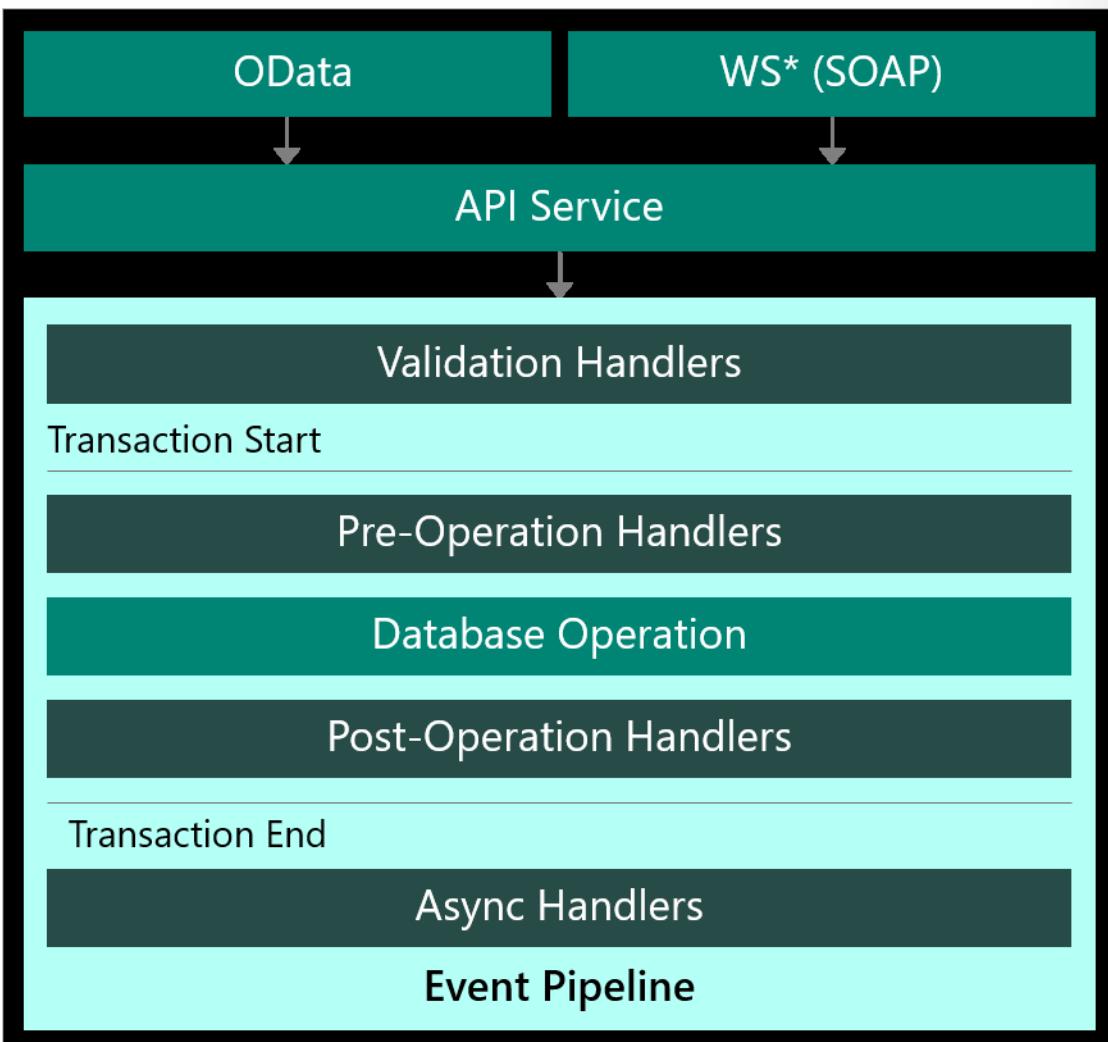
These extensibility capabilities allow complex enterprise applications to be built using the Microsoft Power Platform. Although a solution architect may not have been a developer, a solution architect needs to be familiar with each of these points and how they can be used in solutions.

## Dataverse API

The Dataverse Web API provides a development experience that can be used across a wide variety of programming languages, platforms, and devices. The Web API implements the OData (Open Data Protocol), version 4.0, an OASIS standard for building and consuming RESTful APIs over rich data sources. The Web API provides a modern, RESTful web service you can use to interact with data and metadata in Microsoft Dataverse using a wide variety of platforms, programming languages and devices.

## [!NOTE]

When you are using the Dataverse connectors in Power Automate and Power Apps they are making calls to the OData API.



A developer can add custom plug-in logic at the Pre and Post Operation stages of a transaction to perform validation, calculations, and integrations using the .NET SDK.

## [!NOTE]

The App API is also exposed to a SOAP endpoint. The SOAP endpoint is deprecated and should not directly be used for new apps.

Let us look further at creating custom logic.

## Custom Logic

Microsoft Dataverse provides a number of different ways to perform custom logic.

There are several options available for applying custom business logic in Dataverse including:

- Business rules
- Classic workflows
- Power Automate cloud flows

- Business process flow
- Calculated columns
- Rollup columns
- Plug-ins
- Custom Workflow Assemblies
- Custom Actions
- Custom API
- Client-side scripting
- Power Apps Component Framework code components
- Azure Service Bus and Event Hub integration
- Webhooks

## **Creating custom logic**

The solution architect will need to be able to decide which component of the Microsoft Power Platform is used where in the solution. There are no hard or fast rules for this as each of these options have their own pros and cons.

For instance there are the following limitations/:

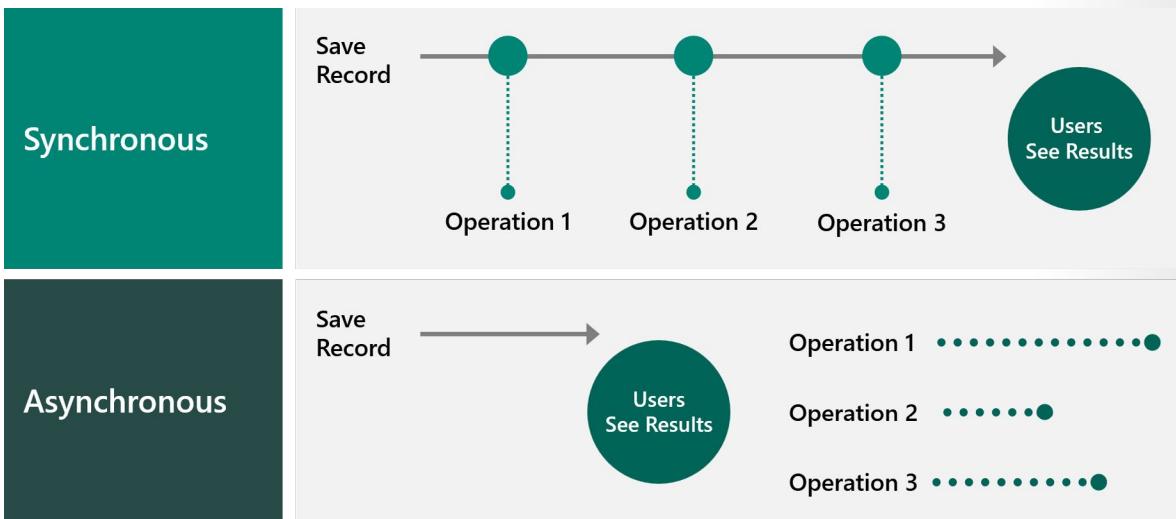
- You cannot trigger Power Automate flows from calculated or rollup columns.
- Rollup columns are only recalculated every hour.
- Business rules can only access the columns on the form for its own table.
- Classic workflows cannot access rows in 1-to-many relationships.
- Data changed by a business rule will not trigger an OnChange event created with JavaScript.

^ This list of limitations is not exhaustive.

Having a no/low code or a code-first approach is not the correct tactic, you need to decide each situation on its own merits. Having an understanding of the capabilities of the capabilities for these options is a pre-requisite for an aspiring solution architect.

## **Synchronous vs Asynchronous processing**

Custom logic can be executed either synchronously or asynchronously. The solution architect needs to consider whether custom logic operations should be performed synchronously or asynchronously.



When the operations are performed synchronously, the user's screen is blocked until all operations are completed. The operations can modify data before or after it is saved to the database. Synchronous calls add minimal overhead to handling the processing but all operations in a synchronous transaction are limited to a total of 2 minutes. This is a hard limit and cannot be changed. Dataverse Plug-ins and Classic workflows can be performed synchronously. Business rules are performed synchronously if the scope is set to table.

When the operations are performed asynchronously, the user's screen is returned after their data is saved to the database. The operation will be added to a queue and will be performed at a later point in time, this could be within a few seconds, or it could be several minutes or even hours depending on the workload. Asynchronous means that users will have to refresh their screen to see the results. There is an additional overhead in running asynchronous jobs. The platform creates rows in the AsyncOperation and WorkflowLog tables. These rows are updated when the operation starts, as it progresses, and after it completes. The rows can be set to be automatically deleted. Dataverse Plug-ins and Classic workflows can be performed asynchronously. Power Automate cloud flows are performed asynchronously.

## Client vs server processing

The solution architect needs to understand where logic is performed.

Canvas app formulas, Model-driven form script, Business rules, and Power Apps Component Framework logic happens in the user interface and the user sees the result immediately. However, this logic is only enforced in the app the logic is implemented in.

Plug-ins, Power Automate cloud flows, classic workflows, and Business rules (with scope set to Table) happens only when the data is sent to the server by an app, flow, or an API call. The user only sees results of service custom logic in their app on refresh of data. Server logic is enforced when any app, flow, or API is used.

The Microsoft Power Platform imposes limits on the number of calls that can be made. The solution architect needs to design the solution with these limits in mind.

## Platform limits

The platform does not provide an infinite amount of resources and there are limits in the number of API calls that can be made.

A solution architect needs to ensure that the solution does not exceed the limits imposed by the platform by designing appropriately and ensuring custom code can handle when limits are breached.

## What is an API Request?

Requests in the Microsoft Power Platform consist of various actions that a user makes across various products. At a high level, below is what constitute an API request:

- Power Apps: all API requests to connectors and Microsoft Dataverse.
- Power Automate: all API requests to connectors, HTTP actions, and built-in actions from initializing variables to a simple compose action. Both succeeded and failed actions count towards these limits. Additionally, retries and additional requests from pagination count as action executions as well. As a basic rule each step in a Power Automate cloud flow is an API request.
- Dataverse: all create, read, update, and delete (CRUD), assign, and share operations including user-driven and internal system requests required to complete CRUD transactions, as well as special operations like share or assign. These can be from any client or application and using any endpoint. These include, but are not limited to, plug-ins, classic workflows, and custom controls making the earlier-mentioned operations.

## Entitlement limits

These limits represent the number of API requests users are entitled to make each day. The allocated limit depends on the type of license assigned to each user. API entitlement limits are based on a 24 hour period as shown in the following table.

User license	Number of API requests / 24 hours
Dynamics 365 Enterprise apps	20,000
Dynamics 365 Professional apps	10,000
Dynamics 365 Team member	5,000
Power Apps per user plan	5,000
Power Automate per user plan	5,000
Office 365 license	2,000
Power Apps per app plan	1,000 per app pass

If a user has multiple plans assigned from different product lines, the total number of requests allowed would be the sum of requests allocated to each license type. For example, if a user has both a Dynamics 365 Customer Service Enterprise license as well as a Office 365 E3 user license then that user will have a total of  $20,000 + 2,000 = 22,000$  requests available per 24 hours.

Dataverse provides the ability to have identities that do not require any user license to interact with the service. There are four types of these users:

- Application users
- Non-interactive users
- Administrative users
- SYSTEM user

Each tenant has a base request capacity per tenant that can only be used by these users and not by users with standard licenses. This base request capacity is based on the type of subscription, as follows:

- If a tenant has at least one Dynamics 365 enterprise subscription, they will get 100,000 requests per 24 hours.
- If a tenant has at least one Dynamics 365 professional subscription, they will get 50,000 requests per 24 hours.
- If a tenant has at least one Microsoft Power Apps or Power Automate subscription, they will get 25,000 requests per 24 hours.

Power Apps and Power Automate capacity add-on allows customers to increase the limits for a given user. Each capacity add-on raises the request limits by an additional 10,000 per 24 hours.

[!NOTE]

For further information see <https://docs.microsoft.com/power-platform/admin/api-request-limits-allocations>

## Service Limits

Apart from the daily API request limit, there are service protection limits specific to each service. As with the daily limits, these limits help maintain the quality of service by protecting the service from malicious or noisy behavior that would otherwise disrupt service for all customers.

Service protection API limits help ensure that users running applications cannot interfere with each other based on resource constraints. The limits will not affect normal users of the platform. Only applications that perform a large number of API requests may be affected. The limits provide a level of protection from random and unexpected surges in request volumes that threaten the availability and performance characteristics of the Dataverse platform.

Microsoft limits the number of concurrent connections per user account, the number of API requests per connection, and the amount of execution time that can be used for each connection. These are evaluated within a five-minute sliding window. When one of these limits is exceeded, an exception will be thrown by the platform.

[!IMPORTANT]

Service protection limits cannot be increased.

[!NOTE]

For further information see <https://docs.microsoft.com/powerapps/developer/common-data-service/api-limits>

## Retry policies and patterns

Custom logic using the APIs should handle retries. When a service protection API limit error occurs, it will provide a value indicating the duration before any new requests from the user can be processed.

The Web API returns a 429 error if the limit is reached. The response will include a Retry-After with number of seconds. With the Organization Service, a TimeSpan value is returned in the Organization-ServiceFault.ErrorDetails collection with the key Retry-After.

[!NOTE]

Care should be taken to not make it worse by over retrying.

For more on service protection limits and what to do see <https://docs.microsoft.com/en-us/power-apps/developer/common-data-service/api-limits>

## How to minimize API calls

The solution design must not depend on infinite capacity being available. For normal users of interactive apps, the limits have been set high enough that users should be affected.

Applications designed to load data into Dataverse or perform bulk updates must also be able to manage service protection API limit errors. These applications prioritize throughput so they can complete their work in the minimum amount of time. These applications must have a strategy to retry operations.

Portal applications typically send requests from anonymous users through a service principal account. Because the service protection API limits are based on a per user basis, portal applications can hit service protection API limits based on the amount of traffic the portal experiences.

Integrations should be optimized to minimize the number of API calls.

The solution architect must also consider high availability in the design of the solution.

## High availability and disaster recovery considerations

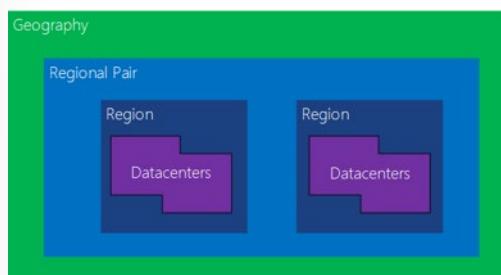
A solution architect needs to consider how to include high availability in the solution design. Microsoft handles high availability and disaster recovery for the Microsoft Power Platform and its components. The solution architect needs to focus on non Microsoft Power Platform components, custom code, and integrations.

### Principles of High Availability

When creating a solution, the solution architect should follow these principles:

- There should be no single points of failure.
- Leverage the capabilities of Microsoft Azure whenever possible.
- Use monitoring and health checks to predict and detect problems.

Microsoft deploys a Microsoft Power Platform environment within an Azure region. Within a region, services and data are replicated redundantly across the datacenter(s) to increase availability. Each Azure region has a paired region where Microsoft Power Platform resources are replicated automatically. Microsoft monitors the Microsoft Power Platform on your behalf as part of the service.



### Handling failover

The Microsoft Microsoft Power Platform components handle internal integrations, solution architects should focus on external integrations and custom code.

If there is an issue with the primary datacenter, Microsoft will initiate the failover automatically, if required, from the primary location to the regional pair. Microsoft also handles any service recovery required.

Users should not notice any interruption in service. At worst, they will have transient error when saving data. Model-driven apps for instance will handle such errors themselves.

For custom code, however, there could be issues. For example, the endpoint for the environment may change. Integrations, should use the global discovery service for endpoint discovery to get the latest environment endpoint. Custom code should have transient error handling with automatic retries.

# Module Summary

## Summary

Microsoft Power Platform provide a flexible architecture that can be tailored to support many business scenarios.

Staying up to date on the platform capabilities is essential to architecting solutions that will best leverage the capabilities

In this module, you looked at the Microsoft Power Platform Architecture, including:

- Key Microsoft Power Platform architecture components.
- Planning for environments.
- Capabilities of Microsoft Dataverse.
- How service limits can affect solution design.

A solution architect needs to make the most of the Microsoft Power Platform architecture:

- Microsoft Power Platform components are designed to work well together.
- Leverage the platform strengths, don't focus on the weaknesses.
- Use Microsoft Azure to fill gaps where you exceed the platform's capabilities, instead of pushing the platform beyond its natural capabilities.
- Avoid using unsupported techniques for customizations.

Consider the performance impact of solution choices. There are many tools in the Microsoft Power Platform that can sometimes answer the same question; choose the tool that maximizes performance and user productivity.

A solution architect needs to consider the roadmap that Microsoft publishes twice a year:

- Understand what is on the roadmap that impacts your solution architecture.
- Evaluate and decide when to adopt new emerging approaches.
- Ensure your solution architecture doesn't use deprecated features.
- Avoid using unsupported customizations to ensure fast pace updates don't impact your solution.

## Next steps

If you want to learn more about Microsoft Power Platform architecture the Power CAT team has produced a series of videos on Microsoft Power Platform architecture and best practices <https://www.youtube.com/playlist?list=PLi9EhCY4z99W2QOTgbwhFZEjpqc8YZDVH>.

To learn more about best practices to secure and govern Microsoft Power Platform environments see <https://docs.microsoft.com/learn/patterns/best-practices-environments/>

The next steps are to look at data modeling in Microsoft Dataverse.

## Module 5 Data modeling

### Model data for Microsoft Power Platform solutions

#### Introduction

The Solution Architect is responsible for the overall high-level design of the data model for the solution. Understanding the capabilities of Microsoft Dataverse is imperative when designing the data model for the solution. This module addresses the trade offs for modeling data in Microsoft Power Platform solutions.

#### What is data modeling?

A data model is a visual model showing how data flows through your system and how different entities relate to each other. Data models define the relationship types between entities and abstract a database to a visual representation that is easy to understand.

For solution based on the Microsoft Power Platform, the data model is crucial to the design of the solution as it will heavily affect app design, especially model-driven apps.

#### Solution Architect's role in data modeling

The solution architect:

- Leads the data model design effort to establish a high-level data architecture for the project.
- Establishes a data model that can be extended by the individual design teams and communicate the design with the teams.
- Reviews and provides feedback as the data model evolves during detail design of solution features.

#### Do not always have all the information

Data modeling can be hard and experience of many different projects will increase your data modeling skills. There is often not a clear correct answer and there may be many possible options.

A solution architect has to make the best choices with the limited information available, this is part of being as solution architect.

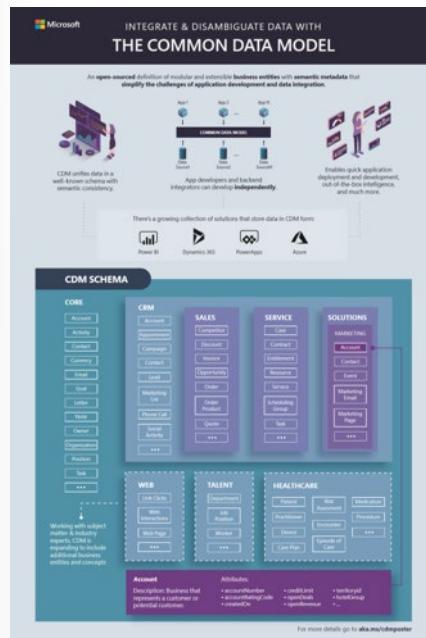
## Common Data Model

The Common Data Model standard defines a common language for business entities covering, over time, the full range of business processes across sales, services, marketing, operations, finance, talent, and commerce and for the Customer, People, and Product entities at the core of a company's business processes. The goal of Common Data Model is to enable data and application interoperability spanning multiple channels, service implementations, and vendors. Common Data Model provides self-describing data (structurally and semantically), enabling applications to easily read and understand the data.

### What is the Common Data Model

The Common Data Model is a standard and extensible collection of schemas (entities, attributes, relationships) that represents business concepts and activities with well-defined semantics, to facilitate data interoperability.

- A shared data model allows applications and data integrators to interoperate via a unified definition of data.
- Common Data Model includes a rich metadata system with standard entities, relationships, hierarchies, traits and more.
- Originated from Dynamics 365/CRM apps, open-sourced in GitHub with over 260 standard entities.
- Multiple systems and platforms implement Common Data Model today.
- Internal and external partners including Dataverse, Power BI DataFlows, Azure Data Services, Informatica and more.



## Where is Common Data Model used

When you provision a Dataverse environment, the core schema in Common Data Model is used to create the tables, columns, and relationships in the Dataverse database. Some of the core tables include: Account, Contact, Lead, and Task.

When you deploy a Dynamics 365 app such as Sales, the tables are created from the Core schema, the CRM base schema, and the Sales schema.

The storage format defined by Common Data Model is also used in other tools such as DataFlows, Power BI, and Azure Data Factory. This permits interoperability between these tools making creating enterprise solutions simpler.

## Microsoft Industry Solution Accelerators

For industry-specific tables, consider using the Common Data Model. In addition to the metadata system, Common Data Model includes a set of standardized, extensible data schemas that Microsoft and its partners have published. This collection of predefined schemas includes entities, attributes, semantic metadata, and relationships. Microsoft is working closely with representatives from various industries to make the Common Data Model more relevant to them, by creating industry accelerators.

Industry accelerators are foundational components within Microsoft Power Platform and Dynamics 365 that enable ISVs and other solution providers to quickly build industry vertical solutions. The accelerators extend Common Data Model to include new entities to support a data schema for concepts within specific industries. Microsoft is currently focused on delivering accelerators for the following industries.

- Automotive
- Financial services, including banking and insurance
- Education, including higher education and k-12
- Nonprofit
- Manufacturing
- Media and entertainment

For more information see [industry accelerators<sup>1</sup>](#).

## Data Modeling

Data modeling on the Microsoft Power Platform should look at the whole data architecture picture and include a logical look at data from Dataverse, Data Lakes, and external sources using connectors.

There are multiple types and standards for data modeling, including Unified Modeling Language (UML), IDEF1X, and others. Specific data model standards are beyond the scope of this unit, but data models for Dataverse data structures generally fall into two general categories; Logical Data Models and Physical Data Models.

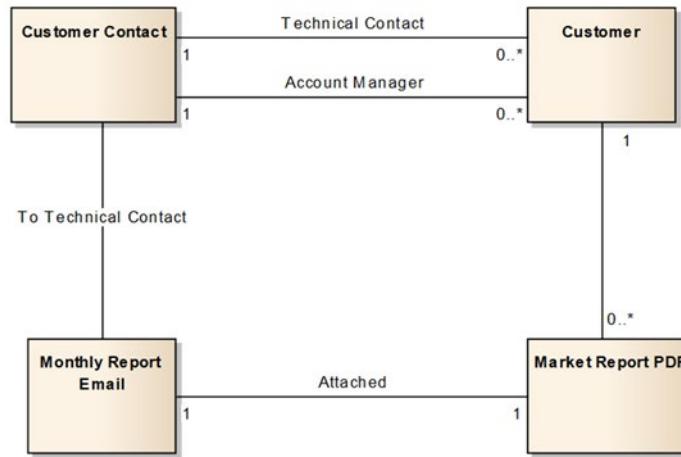
## Entity Relationship Diagrams (ERDs)

Logical data models are high-level diagrams that show the way that data flows through the system. These are frequently put together at the beginning of the project during discovery and before all of the col-

<sup>1</sup> <https://docs.microsoft.com/dynamics365/industry/accelerators/overview>

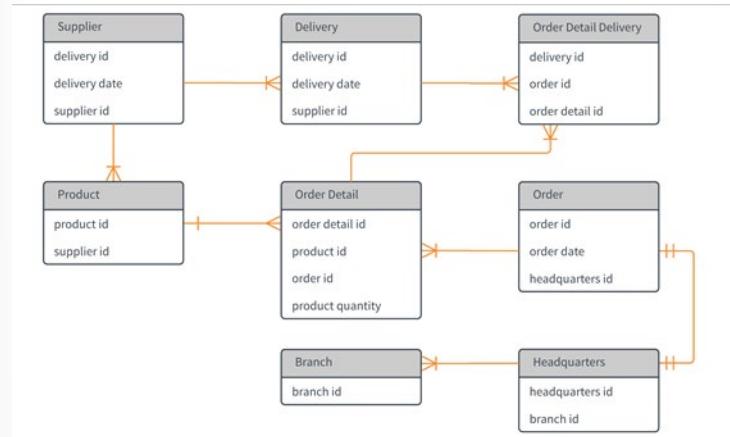
umns have been defined. Generally, the logical data model diagram uses the business names of the entities, not the schema names.

Logical data model diagrams depict the flow of data in a solution without worrying about the physical implementation.

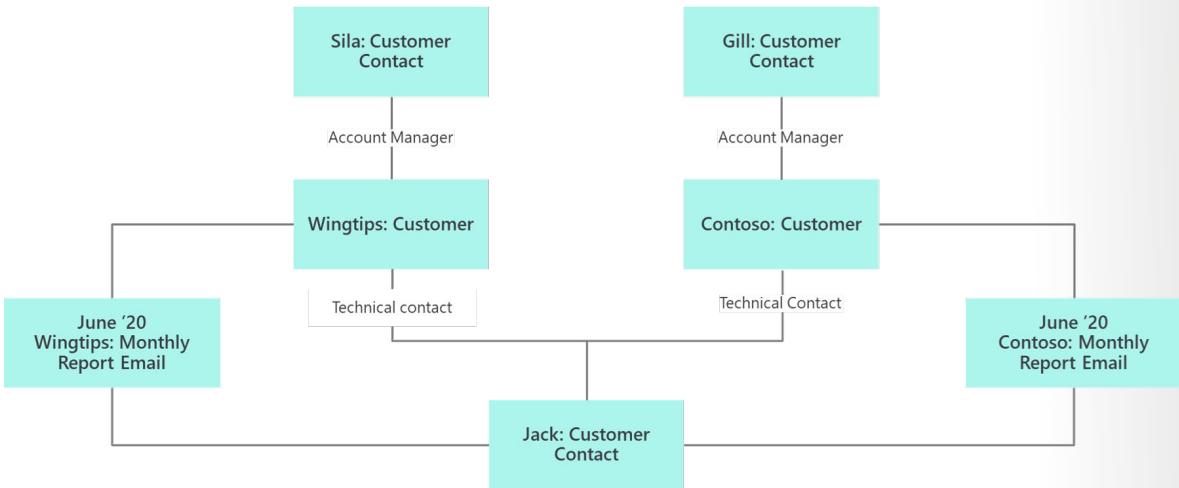


Physical data models are lower level than logical data models. They generally include column-level detail and more precisely designed relationships. The physical data model is created when the high-level logical design is translated to physical entities.

Physical data model diagrams should include showing Dataverse, Data Lake, Connector, or other data store boundaries.



You can also create object diagrams. Object diagrams really show you what you want to know and more importantly what you do not. Object diagrams need to be done in modeling sessions with domain experts.



## Data modeling strategies

The following are guidance to building a data model:

- Start with core tables and relationships: Too often teams get sidetracked with the whole problem it is a lot easier to solve small parts of the challenge and then look at it holistically later.
- Over Normalization: Teams with people with a strong data architecting background tend to try to build a Dataverse data model like they would a traditional SQL Server database. This can lead to poor user experience and requirements for extra processing. The solution architect will need to work with such people to see the cause and effect of relationships in the user experience to help them understand the goal.
- Needs today: Dataverse is great at being incrementally build using an agile process, but having some insight into the short- and long-term future helps lay the foundation. Just do not get stuck trying to nail down every future requirement you can think of.
- Proof of concepts: Dataverse makes it easy to create an environment, try a model, throw it away and try again. Sometimes even challenging two teams with the same data modeling problem can produce useful results.

## Data model influencers

The data model can be influenced by a series of factors:

- Security Requirements: The solution architect should always push for simplification, but these can drive requirements onto the data model.
- User Experience: It is easy to forget that as we add normalization and relationships we create new constructs users need to navigate in the apps.
- Data Location/Retention: Not all data is allowed to be stored. Often data from services cannot be cached and companies have internal policies that govern use of data. Some data is protected by government laws or may have specific requirements for storage, for example, PII, credit card numbers etc.
- Self-service reporting: If it takes a data architect to navigate the data model, the chances are many of the tools from Power BI and Export to Excel will be less valuable to the user. Most self-service features of Dataverse allow navigation of one level of relationship.

- Existing systems: Are these systems legacy? Is there an API? How can the data be accessed? Can it be copied?
- Localization: Is there multi-region, multi-lingual, or multi-currency requirements?

## Data Store

In this unit, we will compare data stores for your apps.

### Where's the data?

When designing your solution, it is important to understand existing data sources for your solution.

### New data

If your app is creating data that does not already exist anywhere, such as in situations where the existing business process was done using paper, we recommend storing the data in Dataverse.

### Read/write from existing system

This is a type of data where you need to retrieve the latest information from an existing database or system. In these cases, data needs to be requested at the time you need it.

### Make a copy of existing data

In situations where original data should never be modified or overwritten, you can copy the data to another data store such as Dataverse. This ensures that the data in the original system won't be changed, yet your app can work with it. This scenario is common when working with data in accounting and revenue-related systems.

### Where to store data

Data modeling on the Microsoft Power Platform should look at the whole data architecture picture and include a logical look at data from Dataverse, Data Lakes and external sources using connectors.

### Dataverse

Dataverse uses abstracts you and your apps from how data is stored. All data in Dataverse is accessed via REST APIs. Currently the tables in Dataverse are relational tables but this may change in the future. Dataverse stores its data in a mixture of stores and formats:

- Azure SQL Server Elastic Pools
- Azure CosmosDB
- Azure Storage
- Azure Data Lake in CDM folders

## Existing Data

Power Apps apps have three ways of using existing data:

- Connector: A connector allows the app to connect to various systems and sources—such as SharePoint, SQL Server, or Office 365—and directly retrieve data from them or save data to them.
- DataFlow: DataFlows extract, transform, and load data from another system to Dataverse or Azure Data Lake storage. Unlike a connector, it fetches data in a scheduled batch. Instead of just retrieving the data as-is from the data source, you can use Power Query Online to manipulate, cleanse, and transform data before you store it to the target storage.
- Virtual Tables: A virtual table is a custom table in Dataverse that contain data from an external data source. Virtual tables appear in your app to users as regular table rows, but contain data that is sourced from an external database, such as an Azure SQL Database. Virtual tables are no longer read-only and apps can create and write data to Virtual tables.

## Choosing where to store data

When choosing where to store data for your solution here are some recommendations:

Dataverse: For transactional data your apps are going to consume and manipulate.

Azure Data Lake: Good for data from other systems, read focused, and brought into a common data model structure.

Connectors: Connectors: Good for leaving existing data where it is and accessing other services that make their data available.

## Dataverse

When you're storing or viewing data with your app, an important part of the design is the data structure. Consider not just how the data will be used in one specific app or screen, but how others will use the data. Referring back to your personas, tasks, business process, and goals will help you define what data to store and how to structure it.

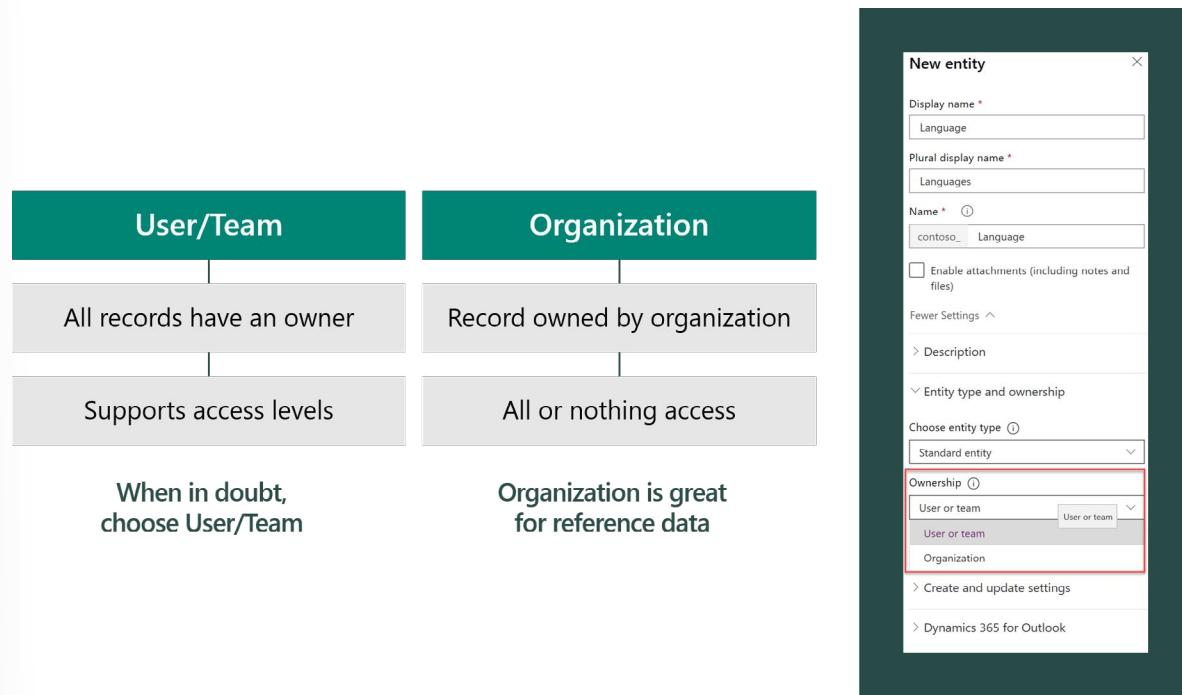
## Table types

There are three types of table in Dataverse:

- Standard: Standard tables are tables where you can store data and add to the navigation in model-driven apps. Most tables you create will be standard tables. There are several standard tables created from the Common Data Model schema in a Dataverse environment.
- Activity: Activity tables are used to store interactions such as phone calls, tasks, and appointments. There is a set of activity tables in a Dataverse database.
- Virtual: Virtual tables allow you to create the table and columns in Dataverse, but then use an external data source to store the data. To the user, the data appears in their apps like any other data.

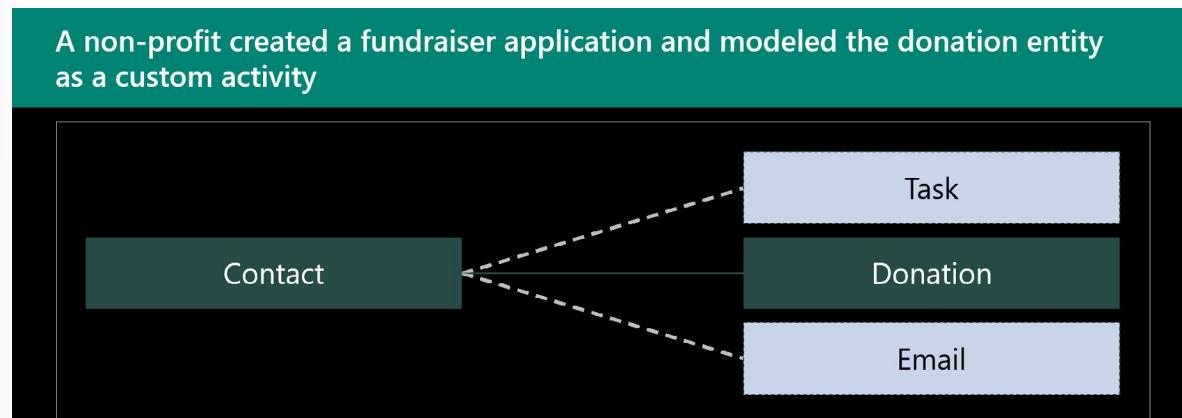
When you create a custom standard table, you must specify its ownership:

- User/Team: Default option.
- Organization: Used for reference data.



## Custom Activity tables

Activity tables are used to store interactions. They have a relationship to all tables that have Enable for activities set on their table metadata. Activity tables share the same set of columns and share the same security privileges. Rows in activity tables appear in the timeline on model-driven app forms. In this example, a custom activity table called Donation has been created.



There are some advantages for using custom activity tables:

- Show up in list with other activities.
- Can be rolled up with other activities.
- Can create a “donation” on any table that supports activities.

However, there are some major disadvantages:

- Not able to configure security different than any other activity.

- Cannot control which tables are related to a “donation”.

## Column data types

You need to choose the data type for columns wisely. This is especially true for numeric data types as you cannot compare numeric columns with different types and there are restrictions on data types for calculated and rollup columns. Once a type is chosen, it cannot be changed.

Data type	Comments
Yes/No	Ensure you never need more choices
File and Image	Allows storing file and images inline in Dataverse
Customer	Can be either contact or customer
Lookup/Choice	Make sure you choose the best one
Date/Time	Make sure you choose the appropriate behavior
Numeric	Many to choose from, choose wisely

## Choice vs Lookup

How do you decide between using a lookup or a Choice? The answer depends on the circumstances.

Choice:

- Only stores label and value as jet-value pair.
- Localization built-in.
- Treated as solution component.
- No built-in way to retire values.
- UX works to about 200 items.
- Can be filtered by using JavaScript.
- Stored as whole number on the row.

Lookup:

- Can store other data in columns on the row of the lookup table.
- You must build localization.
- Treated as reference data.
- Supports inactive state.
- UX scales to lots of items.
- Can filter by views and security.
- Stored as an entity reference.

Storing other data on the lookup table allows access when running workflows, or other customizations that reference the data. For example, a related property can be used in a check condition.

By being a solution component, Choices handle merge resolution by prefixing the value with the publisher prefix

Adding values on a Choice require administrator/customizer level access, while lookup values can be changed by a user if granted permission through security roles.

The user experience (UX) for choices is ideal for small quantity but does not work well for large sets. Lookups provide search type features not available on choices,

If you have multiple choice columns dependent on one another, this can only be achieved with Form-based script, while lookups can be filtered on other lookups using configuration.

## Storing file and image data

You have several choices where to store files and images:

Dataverse: Using the File and Image data typ

SharePoint: Good for collaboration but there is an issue with security. Security for the files follows SharePoint permissions and is not synchronized with Dataverse row permissions.

Azure Storage: Good for archiving and external access. Standalone security but can be granted for small periods of time based on link generated for consumption (valet pattern), and can handle large files.

The file and image data types:

- Are good for upload and reference.
- The security follows record permissions.
- Are limited by size.

## Calculated columns

Calculated columns allow simple calculations to be performed on data in a row.

- Calculated on retrieve of record.
- Value is read only.
- Can include columns from the same row and columns in many-to-one relationships.
- Can include rollup columns in calculation
- Cannot trigger an event for workflow, plug-in, or Power Automate.

## Rollup columns

Rollup columns allow aggregations for related rows in one-to-many relationships.

- Calculated on scheduled basis (minimum 1 hour) and can be updated on-demand by a user.
- Value is read only.
- Can roll up calculated columns.
- Can use hierarchy of related records.
- Can filter across related tables.
- Cannot trigger an event for workflow, plug-in, or Power Automate.

You can roll up “simple” calculated columns, that is, calculated columns that include non-deterministic functions cannot be rolled up.

## Relationships

Relationships define how rows are related to one another in Dataverse. Each table in Dataverse has a primary key to provide a unique reference to the rows in the table. In Dataverse, the primary key is a

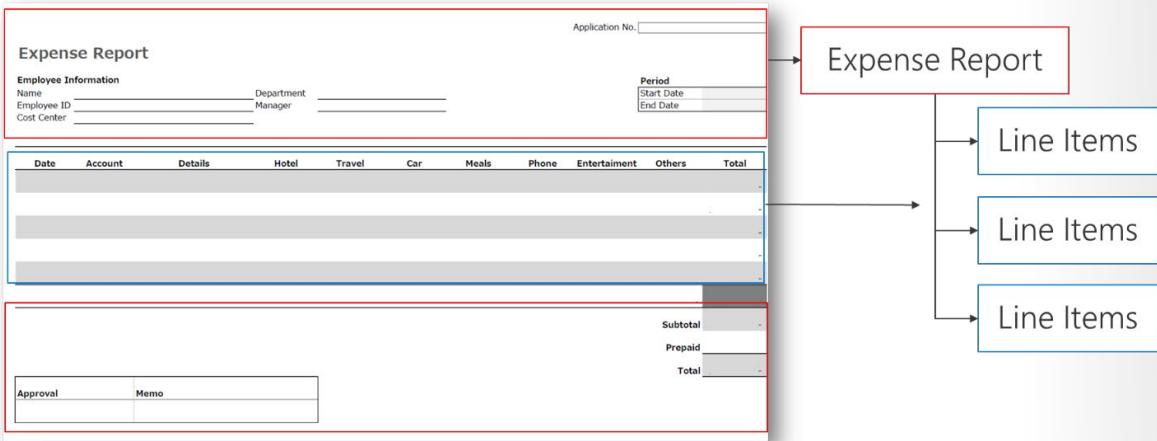
Global Unique Identifier (GUID) that is generated automatically by Dataverse when a row is created. Relationships are created by adding a reference to the primary key. This is known as a foreign key. In Dataverse, relationships are created by using a column on one table to hold the foreign key value. This foreign key is a pointer to the primary key on the other table.

Two types of relationship are supported in Dataverse:

- One-to-many (1:N)
- Many-to-many (N:N)

## One-to-many (1:N)

Let us take the following expense report as an example.

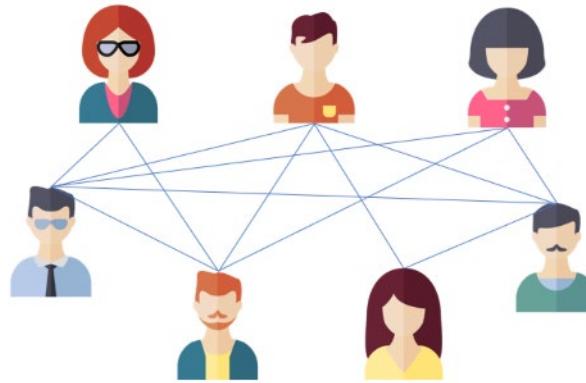


You see the main part of the expense report, which has the employee name and department details. Below the main part, you see multiple rows of descriptions for each purchased item. Let's call these line items. The line items have a different structure from the main part of the expense report. So we can say that for every expense report, there are several line items.

The relationship between the expense report and the line item is an example of a one-to-many (1:N) relationship. The main part of the expense report is linked to several line items. You can also view the relationship from the perspective of the line items: each line item can only be linked to one expense report. This is a many-to-one (N:1) relationship.

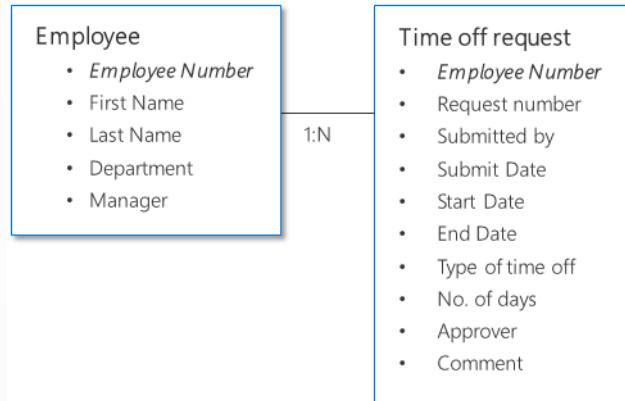
## Many-to-many (N:N)

Multiple to multiple data structure is a special type. This is for cases where multiple records can be associated with multiple set of other records. A good example is your network of business partners. You have multiple business partners (customers and vendors) that you work with, and those business partners also work with multiple colleagues of yours.



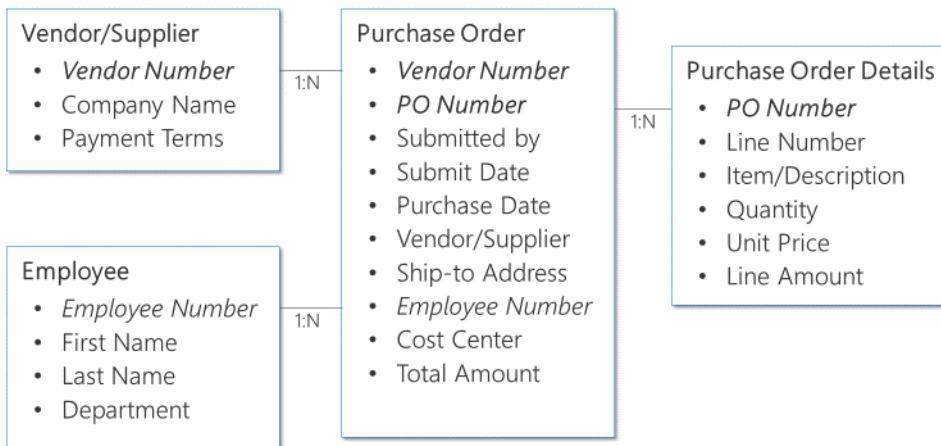
Let us look at some examples.

## Example 1: Time-off approval request



This simple example shows two sets of data. One is the employee, the other is the time-off request. Because each employee will submit multiple requests, the relationship here is one-to-many, where "one" is the employee and "many" are the requests. The employee data and time-off request data are related to one another by having the employee number as the common column (also known as the key).

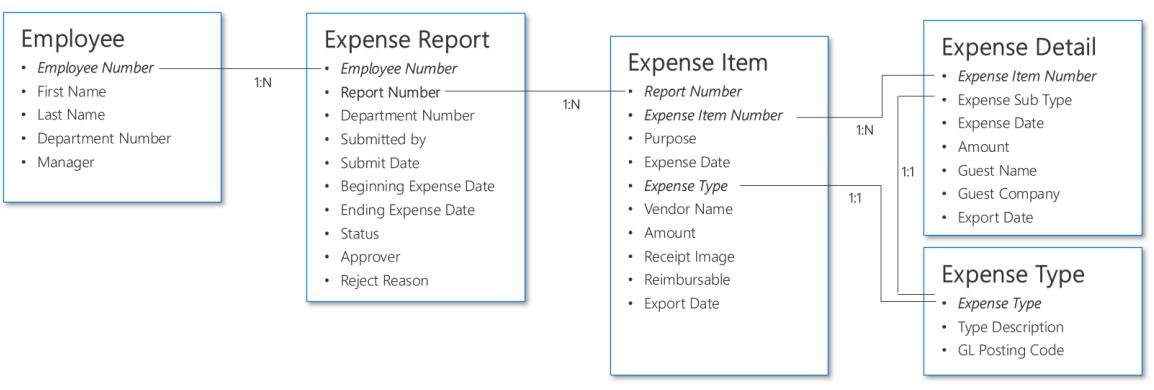
## Example 2: Purchase approval



Here, the data structure looks sophisticated but is similar to the expense report example that was discussed at the beginning of this article. Each vendor or supplier is associated with multiple purchase orders. Each employee is in charge of multiple purchase orders. Hence, both these sets of data have a one-to-many data structure.

Because employees might not always use the same vendor or supplier, vendors are used by multiple employees and each employee works with multiple vendors. Hence, the relationship between employees and vendors is many-to-many.

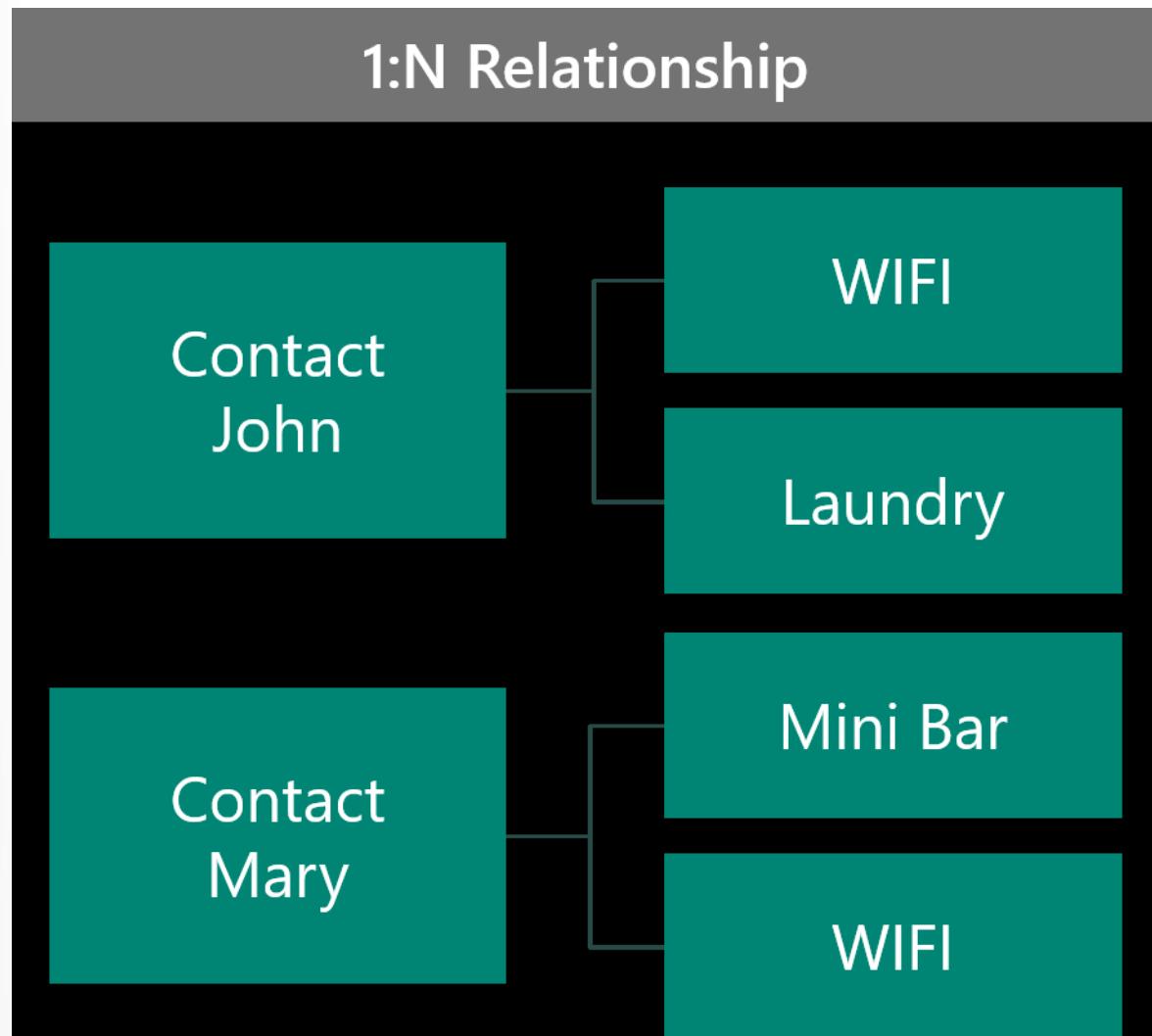
## Example 3: Expense reporting



Here we see an ERD containing multiple tables for an expense reporting solution.

## Example 4: Track which two benefits the VIP selected

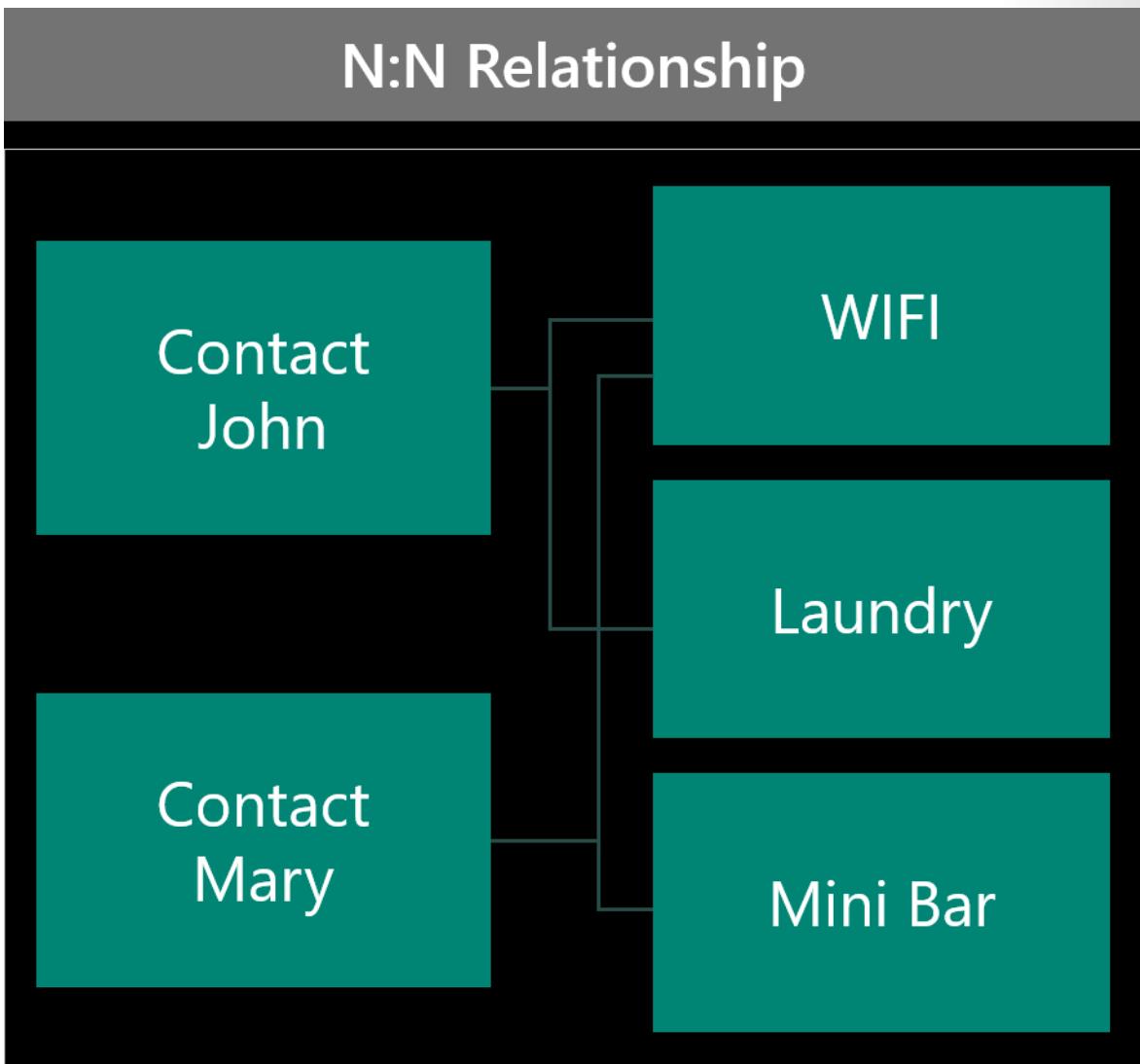
In this example we have two VIPs, John and Mary. John has chosen the WiFi and Laundry benefits, and Mary has chosen the WiFi and Mini-bar benefits. We can model this in different ways. First as 1:N relationship.



In this configuration:

- The benefit record is unique to contact.
- There is no ability to look at all contacts that choose a certain benefit.
- Can do benefit record security based on owner of contact.
- Can store more data on the benefit record that is contact-specific.
- Relationship is parental to benefit otherwise you will orphan the benefit records.

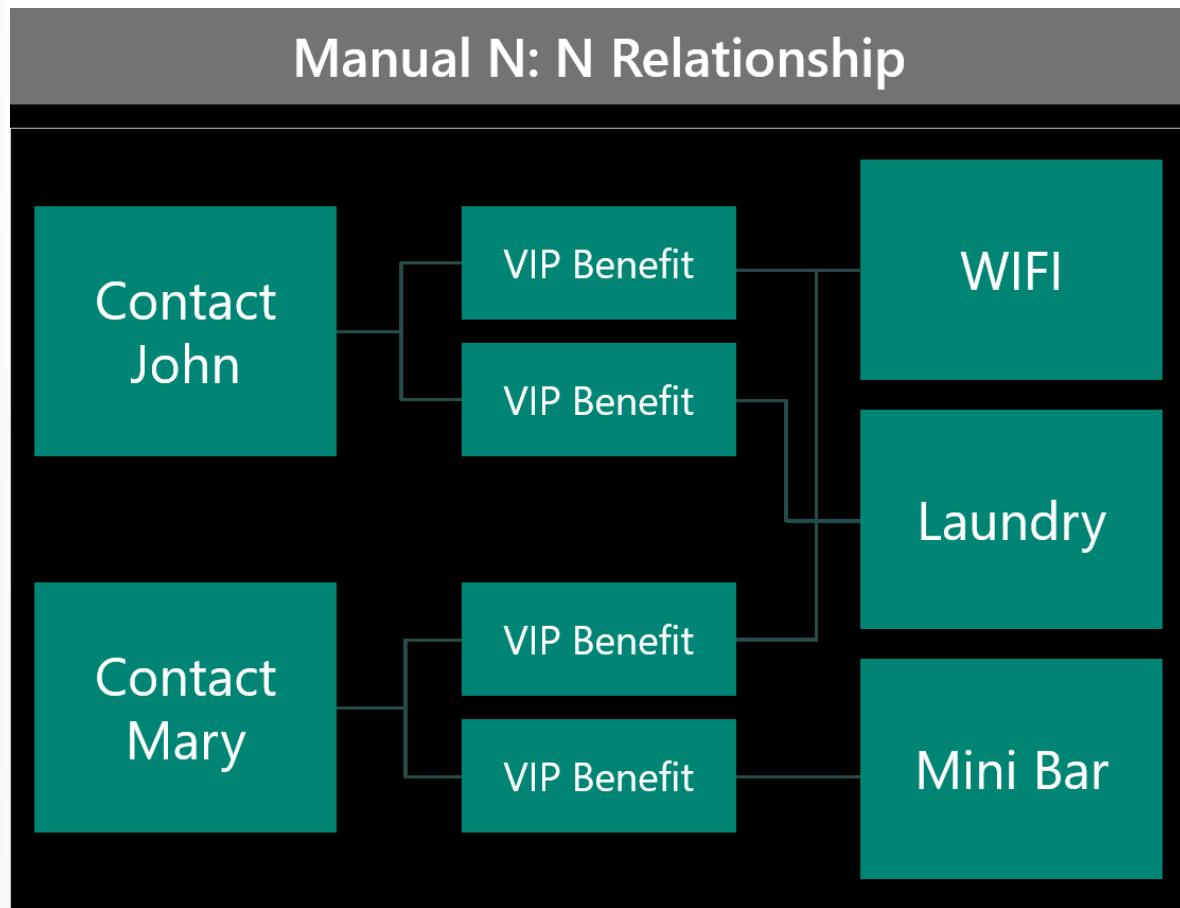
Now as a N:N relationship.



In this configuration:

- Looking at the associated records from the benefit shows all contacts that choose that benefit.
- Security on benefit is shared for all contacts no ability to tailor to each contact.
- Any attributes on the benefit are shared for all contacts, no contact-specific data
- Must use reference relationship otherwise you would be removing benefit from other contact.

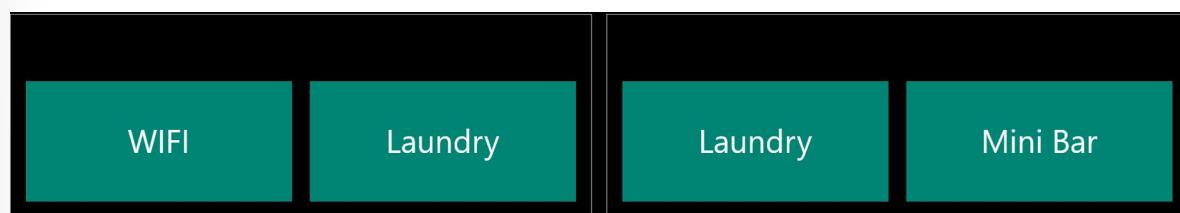
Neither of these configurations is ideal. Now let us create a custom (inter-set) table to hold the VIP's benefit.



In this configuration:

- Adds ability to store more data on the benefit table specific to that contact.
- Requires more work for the user to connect the records, they now have to create the intersect row manually
- Benefits can be secured individually
- Querying is a little more difficult you have no ability to directly access attributes on the benefit table

No let us look at just using columns on the Contact table.



In this configuration:

- Works well for Primary and Secondary benefits but would not scale to tracking lots of benefits.
- Querying and self-service BI would be easier for users.
- Security would follow that of the contact record.

- Querying all users that choose a primary benefit would require a query scanning both primary and secondary benefits.

This configuration is a good example when the benefit must be recorded for some compliance/statistical purpose but has No impact on the business or processing.

## Relationship behaviors

Relationship behaviors control how certain actions cascade down to rows related to the primary table row through the 1:N relationship. Behaviors maintain referential integrity and can prevent orphan records from being left behind.

Type of Behavior	Description
Parental	All actions use the Cascade All behavior.
Referential	Assign, Share, Unshare, and Reparent use the Cascade None behavior Delete uses the Remove Link behavior Merge uses the Cascade All behavior
Referential, Restrict Delete	The same as Referential, except that Delete uses the Restrict Delete behavior
Custom	Individual behaviors can be assigned for each action. If the choices match any of the other Type of Behavior categories, the value will change to that Type of Behavior value

The screenshot shows a configuration dialog for relationship behaviors. At the top, there's a section labeled 'Advanced options'. Below it, under 'Type of behavior \*', a dropdown menu is set to 'Referential'. Under 'Delete \*', another dropdown menu is set to 'Restrict'. At the bottom right of the dialog are two buttons: 'Done' (in purple) and 'Cancel'.

### [!IMPORTANT]

Defining relationship behaviors is important as the cascade of assign can cause related records to be assigned. If in doubt, set the behavior to Referential and Restrict.

## Alternate keys

Alternate keys are used in integrations to reduce the need to perform a query to find a record. Using an alternate key, a row can be updated without knowing its GUID.

Alternate keys:

- Are Great for use in retrieves and updates.
- Can contain decimal, whole number, text fields, dates, and lookup fields.
- Can have up to five alternate keys per table.
- Create a nullable unique index behind the scenes to enforce uniqueness for the key.

When a key is created, the system validates that that key can be supported by the platform.

## Diagram best practices

When creating ERDs for Dataverse, you should:

- Avoid data duplication; every piece of data should only have one home. Rather than duplicating the same data between multiple tables, functionality like quick view forms and displaying related table data in views should be used.
- Use the ERD relationships to review and identify potential cascading behaviors that could impact business logic. For example, with parental relationships, permissions like Assign, Share, Unshare,

Reparent, Delete, and Merge will automatically happen to related records when a parent record is updated.

## Best practices

Data modeling is a science, and there are data modeling professionals and established standards for data modeling. To be effective with Dataverse data modeling, you do not have to be a professional data modeler or use any special tools. Popular tools like Microsoft Visio can be used to quickly create a basic ERD diagram that visualizes the relationships and flow of data between tables. In this topic, we will discuss some general best practices for data modeling for Dataverse deployments. Best practices to follow are:

- Data models should be updated continuously during a deployment. It is common for a data model to be designed at the beginning of a project, but it is important that it does not stop there. As you go through the deployment, new columns and tables will be added. It is important to capture these new columns and tables in the data model and make it a living data model. Recommend to customers that they continue to update it as they enhance the system.
- Do not start from scratch. Community tools available with the **XrmToolBox**<sup>2</sup> make it easy to quickly generate ERD diagrams of your Dataverse configuration. These tools include the UML Generator and Entity Relationship Diagram (ERD) Generator. After you complete configuration updates, generate an up-to-date ERD diagram.
- Do not include every table. Some core tables, such as activities, notes, and users (record owners) are related to nearly every table in Dataverse. If you include every relationship with these tables in your data model, the result will be unreadable. Best practice is to only include the primary tables used in your configuration in your data model diagram, and only include custom relationships with the user and activity tables to maximize readability.
- Data models should include tables outside of Dataverse. If you are integrating with other systems using Dataverse data connectors or virtual tables, or if data flows outside of the Common Data Service using an integration, this data should also be represented in your data model diagram.
- Start simple, with the standard tables, then add custom table relationships to your data model.
- Experience should influence your data model. Sometimes it is easy to over-normalize your data, but in the process you can make the application more cumbersome to use.

Start with what you need now, but design the data model in a way that will support what you are going to be doing in the future. For example, if you know that in the future you will need to store more details about sales territories, using a text column for territory now will make it more difficult to implement than if you use the territory table relationship. Plan ahead for what is coming.

## Out-of-the box vs. custom tables

In this topic, we identify standard, "out-of-the-box" (OOB) tables that are used in the configuration, along with custom tables and the purpose for which they are being used. This information matters because the Microsoft Dataverse has many common tables, and as a general rule, a custom table should not be created if a standard table already exists that addresses that purpose. The reason is that if you overload your configuration with many redundant tables, you will negatively impact the performance of the system, and you will make the system more difficult to use (many redundant sounding tables confuse users in advanced find). Each custom table should serve a specific purpose. This topic will also help identify what tables are the most used and identify if you are at risk of over-loading tables.

---

<sup>2</sup> <https://www.xrmtoolbox.com>

## Should you replace standard tables with custom tables?

Sometimes makers think they should replace standard functionality with custom tables. Makers reason that, for example, if they need a sales opportunity, but they need a simpler form than the standard opportunity form, that it might be easier. However, you should consider what you may be giving up by using a custom table, instead of a standard table. Using an out-of-the-box table ensures greater alignment with core platform features. Since more features are added regularly, using standard tables makes it easy to benefit from new features when they are released. For example, if you decided to replace the standard opportunity table with a custom opportunity table, you will not be able to use Sales Insights and other AI features.

## Accounts and contacts should never be recreated

When deploying Microsoft Power Platform solutions, there are frequently multiple types of companies, organizations, and contacts that you will be tracking in the system. Some represent customer/client organizations, some may be support and advisory organizations, like accountants and legal firms, and some may be miscellaneous types of organizations, such as trade associations.

How should you manage multiple categories of company relationships? The most common approach is to use the account table for all organization types and use a column like relationship type or a custom option set to flag companies by their type or category. Views can be filtered based on the type of company, and business rules can conditionally show or hide column and form components based on type.

In order to benefit from a standard integration with Dynamics 365 Finance and Operations apps using Dual-Write, it is best to use the default tables and columns that are added by the Dual Write Core solution to your Dataverse environment.

Another approach is to create custom tables for each type of company. One reason commonly cited is "I might need to use accounts for another reason in the future, so I do not want to customize the account table."

Before recreating the account table as a custom company table, consider strongly what you are giving up by creating a custom table. Consider the following:

- Multiple addresses: Account has a unique address capability that supports multiple addresses. The first two addresses are displayed on the company form, but these address records live in the related customer address table. While you can create a custom address table tied to a custom company table, recreating the unique logic where the addresses are stored in the related table and displayed on the form and table views would require development. If you need multiple addresses, use the account table.
- Contact hierarchy: Accounts are the parent of contacts. Activities related to contacts roll up to the parent account record. This hierarchy cannot be replaced by a custom company record. You can create more relationships with custom company tables, but the standard account/contact relationship cannot be replaced. If the company has contacts with their primary company relationships to this type of company, or if you want to roll up activities from contacts to companies, use the account table.
- The standard map control in model-driven apps does not support custom company tables.
- Hierarchical relationships between parent/child accounts and the standard hierarchy visualization and roll-up of child account activities to parent account, only work with standard account tables.
- Dataverse includes a special type of polymorphic lookup column called a customer column. This column allows a row to be linked to a company/account or a contact. Dataverse does not allow for custom tables to be selected from polymorphic lookup columns.

- Marketing will not work. Marketing lists can only work with contacts, accounts, and leads, not custom tables. Dynamics 365 Marketing can send to accounts and contacts, but not custom company tables.

So, in almost every situation, the account table should be used for company records of all types. What are the exceptions?

- Minor types of companies that are not relational and have minimal attributes. Think of a type of organization with no contacts, address, and that only exists for lookup purposes.
- Unqualified or unverified companies imported from business cards or web forms that we do not want to pollute the account table. For these situations, you could use the lead table.

## Should you repurpose system tables for other purposes

Consider the scenario where you have a business requirement that is similar to opportunities, but not really a sales opportunity. Should you repurpose system tables or create new tables?

The following items should be considered before repurposing system tables:

### Consider the future

The future of the Microsoft Power Platform is moving much faster than ever before, so using tables in non-standard ways can cause problems if Microsoft makes changes to the table that you are using. Also, if you choose to repurpose a little used system table like contracts, there is a chance that Microsoft will elect to deprecate that table in the future. Custom tables do not get deprecated. Also, if you repurpose a system table, what will you do, if later, you need that entity for its intended purposes? There are customers who repurposed case, and later needed case management and had to address it with custom tables because the standard case table was already used for dramatically different purposes.

### Consider the overhead

There are many system tables that have certain columns that cannot be removed from the forms. For example, some columns on tables like opportunity, case, and campaign cannot be removed from the form. While you can hide these columns, having a bunch of locked columns on the form can add overhead to your environment configuration.

### Consider the user experience

If your use case is less than 50% inline with the standard table functionality, a custom table will typically give users a simpler user experience than scaling down a more complex system table. It is also possible to add business process flows to any table, including custom tables, which can easily make a custom table user experience as good as, or better than repurposing a system table.

## Avoiding common pitfalls

Below are common data modeling issues:

- Too many tables: Probably over normalized
- Too many columns on a table: Probably should have been a separate table.
- Use the tools: Quick view forms instead of repeating columns.
- Avoid Yes/No data type, if there might be more values, or you need to store as Unknown.
- You are stuck with data type formatting forever.

- Do not build out parts of the data model you don't plan to use.

## Proof of Concept

Dataverse makes it easy to create a trial environment and it is quick to create tables and relationships. You can build Proof of Concepts to try out your data model and see what the user experience might be like.

# Data modeling exercise

## Exercise Overview

In this exercise you will work as small groups. Review the information presented about Fabrikam Robotics and complete the tasks.

## Learning Objectives

After completing the exercise, you will:

- Evaluate customer requirements
- Make decisions for the data model

## Exercise 1

You are building a solution for Fabrikam to track visitors to a showroom and manufacturing site. Some of the visitors are potential purchasers and some are just there to see the magic of the robots working.

- Visitors must have a reserved spot to gain access.
- Visitors invited by sales staff must be tracked to a sales process, visitors just touring for fun are not tracked to a sales process.
- Visitors can bring guests.
- Each visitor must have a photo taken upon arrival and associated with their visit
- Each visitor must sign a waiver of liability each time they visit, and you must store their signature and date time of acceptance.
- Each primary visitor is assigned an engagement tracking device that tracks their location in the manufacturing area and showroom. This is for safety as well as for improving the tours. The device stores the data in its own cloud service that offers both bulk export and API access.
- You must allow for the tracking data to be viewed in the sales process user interface used by the sales staff
- Marketing has asked to be able to view visitors by day/month/quarter along with statistics on closing of sales after a visit.

## Tasks

Create a data model for the above requirements. Use whatever tool or medium that you have available to complete the data model. You can use a whiteboard, Visio or pen and paper.

- How did you handle storing of visitor photos?
- How did you handle storing of visitor waiver acceptance and signature?
- How did you accommodate for viewing of tracking data in the sales process?
- Did you use anything from the Common Data Model schema?

- How did you handle marketing's statistics needs?

# Module Summary

## Recap

In this module, you looked at data modeling with the Microsoft Power Platform, including:

- Learning about data modeling influencers
- Learning about data model strategies
- Choosing where to store data
- Choosing data types
- Defining relationships

A good data model is essential as it forms the foundation of what you build your apps on. The data model drives your app interface and design.

Problems with your data model are much harder to correct once apps have been composed using the model.

## Next Steps

Learn more with the **Design data models for Dynamics 365 solutions module<sup>3</sup>**.

---

<sup>3</sup> <https://docs.microsoft.com/learn/modules/data-models/>

## Module 6 Analytics and artificial intelligence

### Evaluate Microsoft Power Platform analytics and AI

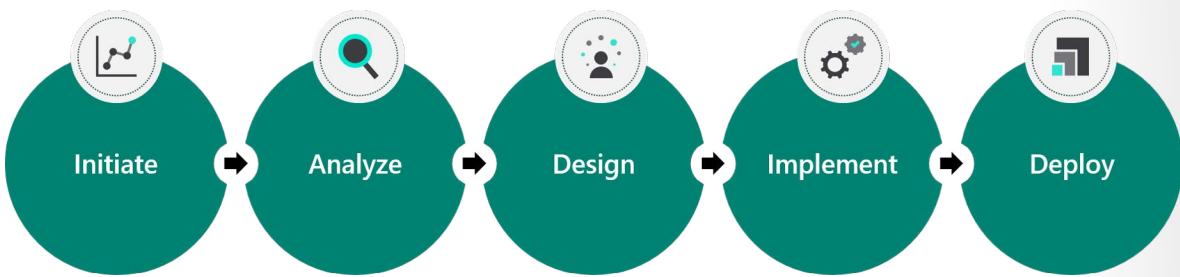
#### Introduction

Analytics play a major role in solutions that are developed for Microsoft Power Platform. Analytics and insights can be delivered with Microsoft Power Platform in several different ways. The solution architect will need to evaluate the ways for implementing reporting and analytics in the solution.

AI is becoming more important in enhancing the apps and data in business solutions. Accordingly, the solution architect will need to evaluate the options for using AI within Microsoft Power Platform solutions.

#### Reporting and insights

The solution architect needs to consider reporting and analytics in all stages of the project.



For example:

- A solution architect will often be involved in sales and presales and will use reporting to impress the customer.
- During the Analyze phase, the solution architect must clarify the actual goals for reporting and analytics.

- In the Design phase, the solution architect must ensure that data modeling incorporates the required data and data relationships to support the reporting and analytic requirements.
- Training and setting up for self-service are required at the Deploy stage.

## Solution architect role in reporting

Microsoft Power Platform offers a multitude of options for reporting, which will be covered in this module. Solution architects must evaluate the needs from the requirements and then identify the best approach to take.

[!NOTE]

Different approaches might be required for different needs. The solution architect should select the most appropriate method for each requirement and scenario.

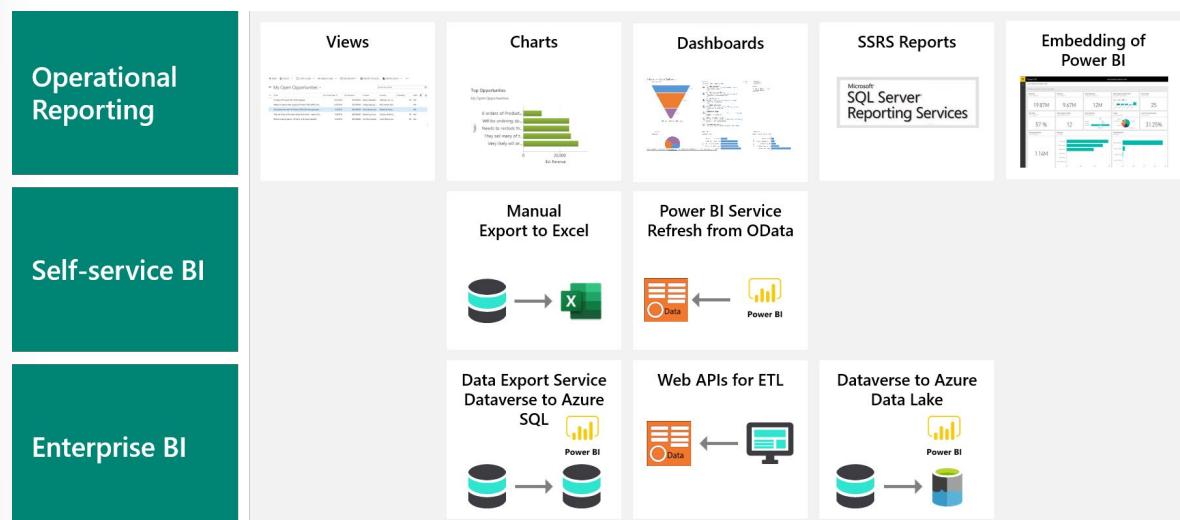
Because reporting is left until the end of a project, a key task that is often overlooked is ensuring that the data model supports the requirements for reporting and analytics. The data model must also support the required relationships for analytics. Solution architects should ensure that the data model supports application processing and required reporting.

The solution architect should look for opportunities to use proactive insights and AI instead of reactive reviewing of reports and analytics.

## Types of reporting and analytics

When looking at reporting, the solution architect needs to consider different categories of reporting and analytics:

- **Operational reporting** - Data comes from Microsoft Dataverse directly and is viewed and interacted with in the context of an app in Microsoft Power Apps.
- **Self-service BI** - Data is exported from Dataverse or could be refreshed on a schedule.
- **Enterprise BI** - Data is extracted for use in broader enterprise reporting tools, which could be done to allow integration of data from other sources.



Solution architects need to be aware of the prebuilt insights that are available with Microsoft Dynamics 365 apps, such as:

- Dynamics 365 Sales Insights

- Dynamics 365 Customer Service Insights
- Dynamics 365 Customer Insights
- Dynamics 365 Fraud Protection

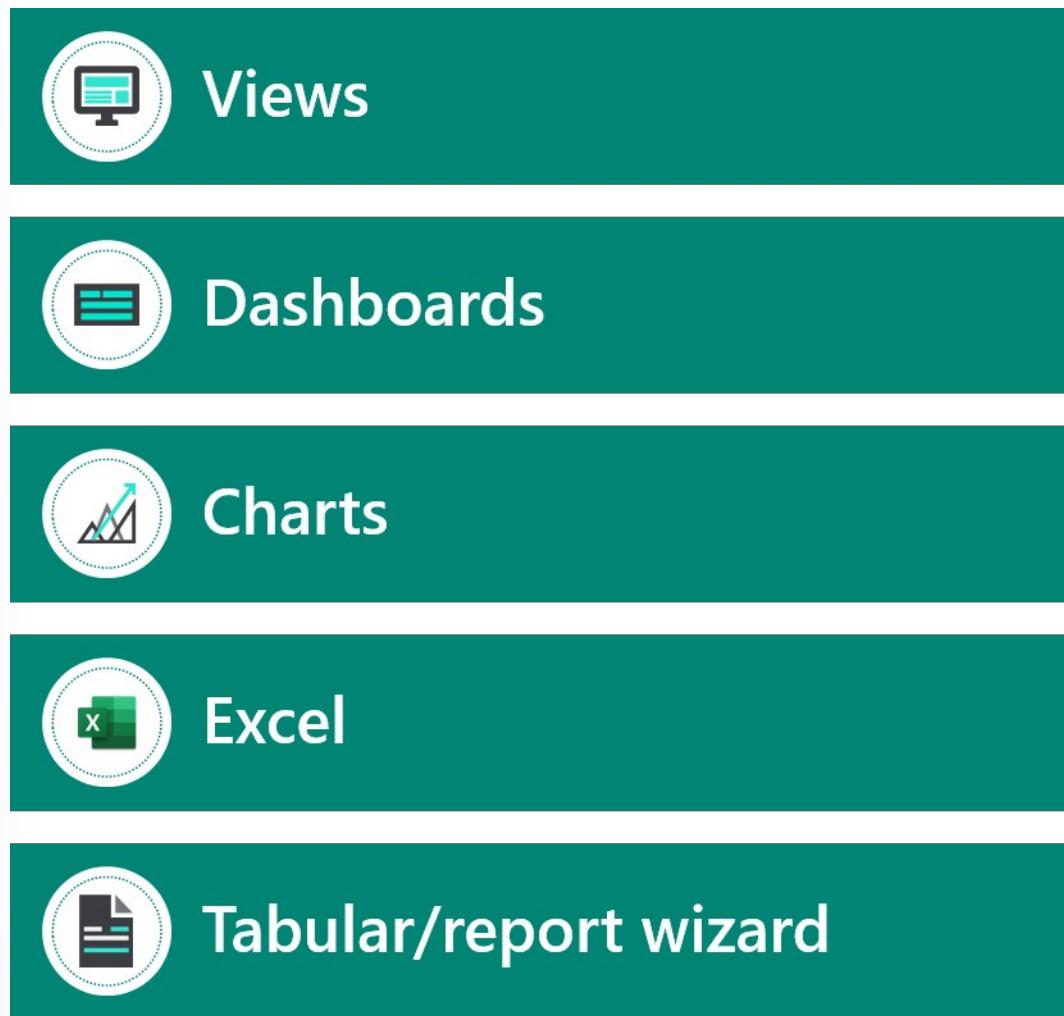
Extensive insight capabilities are available within these apps, which could be used in the solution instead of building reports. Additionally, Microsoft Power BI template apps are available for Dynamics 365 apps and could be deployed to meet reporting requirements.

Questions that solution architects should consider when evaluating requirements for reporting and visualizations:

- What data is required?
- Is data required that is external to the solution?
- Does the requirement fit with one of the prebuilt insights?
- Who consumes the report/visualization and are they users already?
- How fresh does the data have to be, or does it need to be at a point in time?
- Does something need to be built, or can an existing view or report satisfy the requirement?
- What is the expected action that users will take from reviewing the report?
- Is the action something that can be predicted, or can it be automated instead?

## Reporting capabilities

The reporting options that are provided in Microsoft Power Platform meet many operational reporting requirements.



## Model-driven apps

Dataverse provides many reporting options for model-driven apps:

- **Views** - Views are stored queries on the tables in Dataverse with selected columns and filters. Many reporting requirements are simple lists of data, and views can often meet many of those requirements.
- **Charts** - Charts are visualizations of Dataverse data in a view.
- **Dashboards** - A dashboard is a collection of views and charts. Power BI visualizations can also be added to dashboards. Standard dashboards are for viewing of data only. Interactive dashboards allow users to filter data and take action.

The advantages of these options are:

- Simple access from within apps.
- Data is always current.
- Security model is enforced.
- They are included in solution packages.

- No special skills are required.
- Users can create their own personal views, charts, and dashboards.

The disadvantages of these options are:

- Only simple visualizations are available.
- Data is limited to a single table and tables in many-to-1 relationships.
- Data is always current and cannot look back at a point in time.
- Trends cannot be analyzed.
- Users must have a license and be an app user.
- Charts and dashboards are limited to 50,000 rows.
- Views will only show the first 5,000 rows.

## Export to Excel

Dataverse provides the ability to export to Microsoft Excel. Users can export as static data or as a dynamic query. Exporting of data is from a view with columns and filters.

You can provide static data to users who are not app users. Static data is for when you need to get a snapshot of the data at the current date and time or if you want to share the data with others.

Use dynamic query to get the most up-to-date information. Additionally, you can use dynamic query to refresh your information in Excel and match what you see in your app at any time. A user must have a license to access an Excel worksheet with a dynamic query. Dynamic queries support exporting the data as either rows or a PivotTable.

The **Export to Excel** feature can only include data from Dataverse.

Data can also be edited in Excel Online and then saved back into Dataverse to give an immersive editing user experience.

A limit has been established for exporting to Excel of 100,000 rows by default. You can increase this limit to 1,000,000 rows with the **MaxRecordsForExportToExcel** setting.

## Word and Excel templates

You can use Microsoft Word and Excel templates for reporting. Word templates are for a single row and all its related rows. Excel templates are for a view, or list, of rows. Excel templates can include visualizations and other analysis that are provided by Excel.

You can control access to individual templates through security.

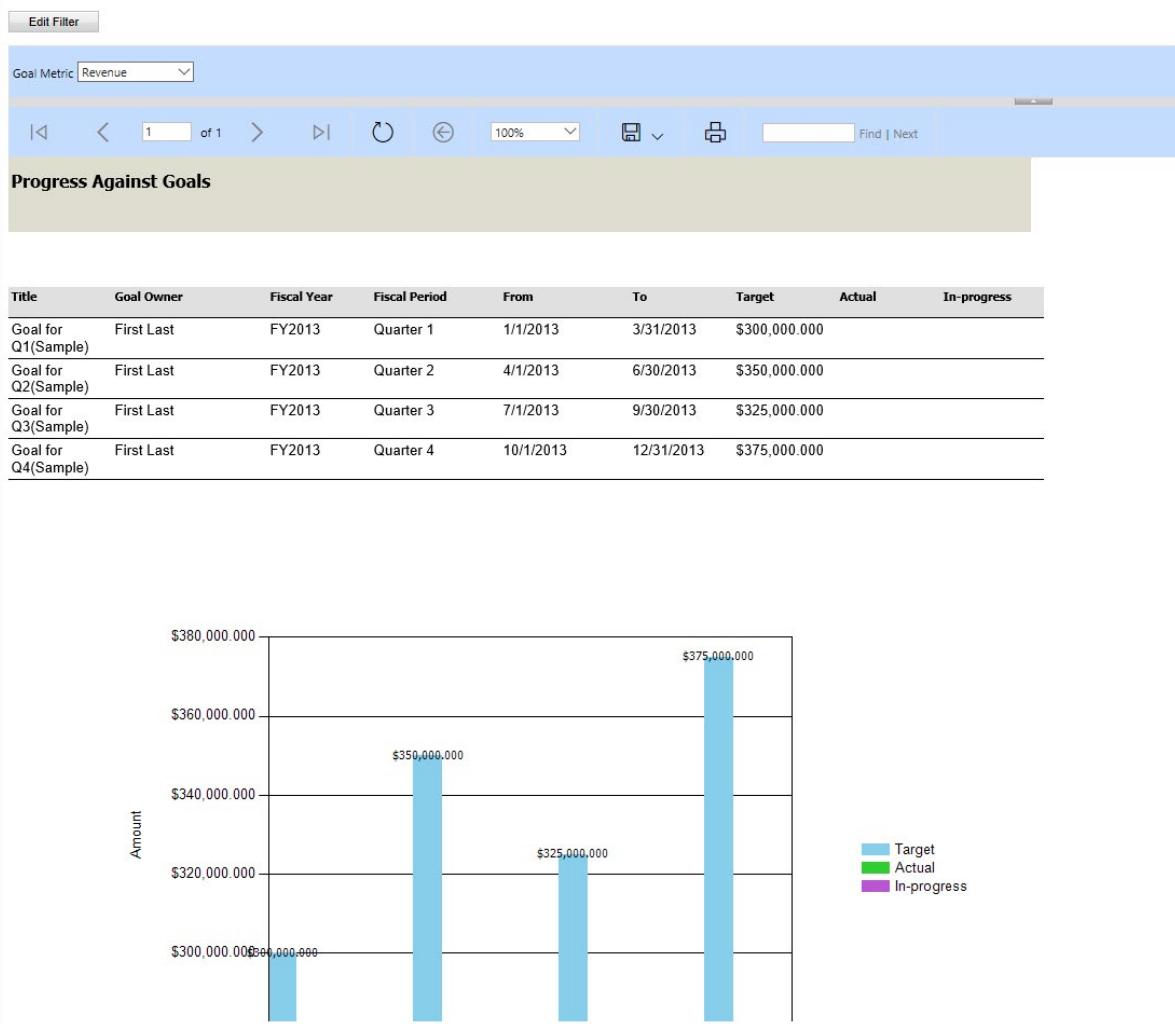
### [!IMPORTANT]

Word and Excel templates can't be included in a solution package. Document templates that are downloaded from one environment can only be used within that environment. Environment-to-environment migration for Word or Excel templates isn't currently supported.

## Report wizard

The report wizard is a user-reporting generation tool for model-driven apps. The report wizard can create a SQL Server Reporting Services report from data that is held in Dataverse. The report can be tabular or contain a simple chart.

The reports that the wizard generates have a simple layout, as shown in the following screenshot.



You can download and edit these reports to change their layout. Reports that are created by the report wizard can be included in a solution package.

## SQL Server Reporting Services

A data analyst can create reports by using SQL Server Reporting Services and Visual Studio. SQL Server Reporting Services reports can retrieve multiple datasets from different parts of the data model, which allows for more complex reports than can be created with the options that were previously described.

### [!NOTE]

Reports and queries can run for up to five minutes. When the maximum period has been reached, the report will time out and a message will be returned to the user. Within the five-minute duration, reports and queries are allowed to span large datasets that are beyond 50,000 rows.

Tips for creating reports:

- Design reports to query smaller datasets over shorter periods of time by adding a time-based filter in the report, such as the current month or quarter, to limit the results.
- Limit the number of tables that are needed to return the result. This approach helps reduce the time that is required to run the query and return the result set.

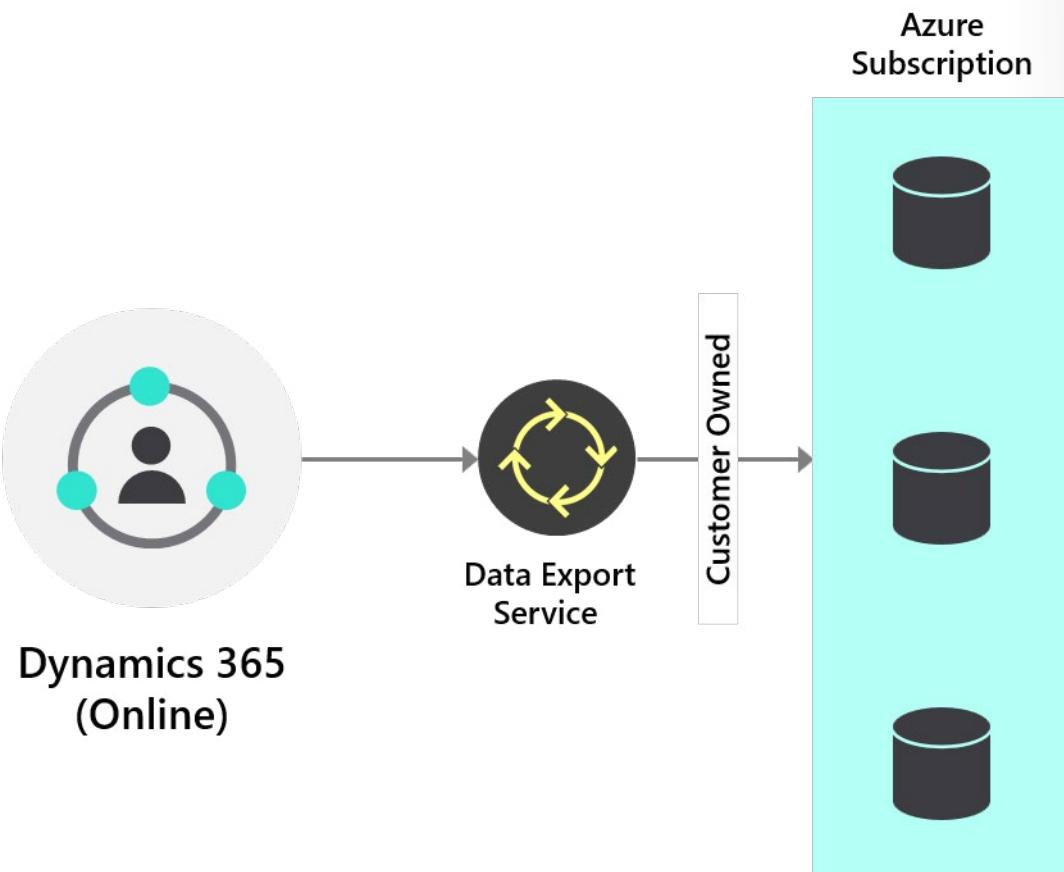
- Reduce the number of rows that are shown in detailed reports. Suitable filtering can be used to reduce the number of rows that are returned by the query to reduce time-outs.
- For aggregated or summarized reports, queries must be used to push the aggregation to the database and not fetch detailed rows and perform aggregation in the SQL Server Reporting Services report.

[!IMPORTANT]

Running large reports can affect performance for all users. The solution architect should consider offloading the reports by exporting the Dataverse data to enable more complex and deeper reporting.

## Export to Azure SQL

Data Export Service replicates Dataverse data to your own SQL database. You can use Microsoft Azure SQL or a SQL Server on an Azure VM. Data Export Service is available as an add-on from Microsoft AppSource.



[!IMPORTANT]

Data Export Service is only available for Microsoft Dynamics 365 Customer Engagement apps.

Data Export Service intelligently synchronizes the entire data initially; thereafter, it synchronizes on a continuous basis as changes occur (delta changes) in the system. This feature helps enable several analytics and reporting scenarios on top of data with Azure data and analytics services. The data in the Azure SQL database is almost real-time. Data analysts can run SQL queries natively on the Azure SQL database. Error handling and monitoring are provided.

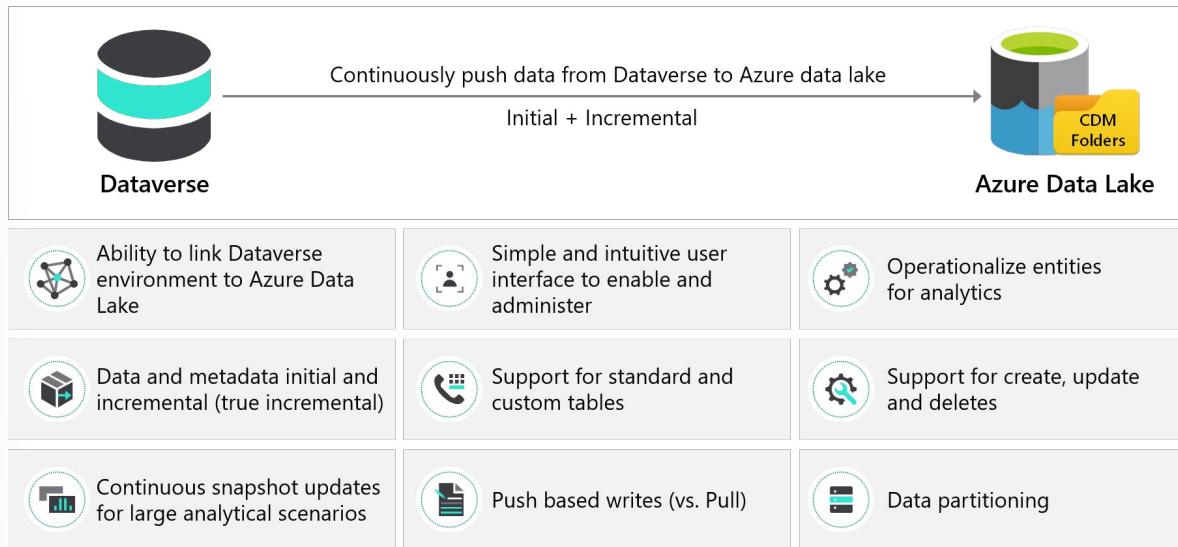
## [!NOTE]

Data Export Service uses the change tracking features of Dataverse. A table must be enabled for change tracking to be exported by Data Export Service.

An alternative export option is to use Microsoft Azure Data Lake.

## Export to Azure Data Lake

The Export to Azure Data Lake service enables continuous replication of Dataverse table data to Azure Data Lake Storage Gen 2, which can then be used to run analytics such as Power BI reporting, Microsoft Azure Machine Learning, Data Warehousing, and other integration scenarios.



Export to Data Lake simplifies the technical and administrative complexity of export tables for analytics. With a few simple selections, customers can link their Dataverse environment to a data lake in their Azure subscription and then select tables and export them to Azure Data Lake Storage. All data and metadata changes (initial and incremental delta) in Dataverse are automatically pushed to Azure Data Lake Gen 2 without additional actions.

Data is stored in Common Data Model format, which can be consumed by Power BI and other Azure analytics services.

## Alternative options

Advanced Find is one of the most useful tools that functional consultants, business analysts, administrators, and even end users should master. Advanced Find allows users to create their own queries and save them as personal views. Advanced Find queries are the basis for many other functions in model-driven apps, including Export to Excel, Excel templates, bulk delete, duplicate detection, and dashboards.

Creating a report is not always needed:

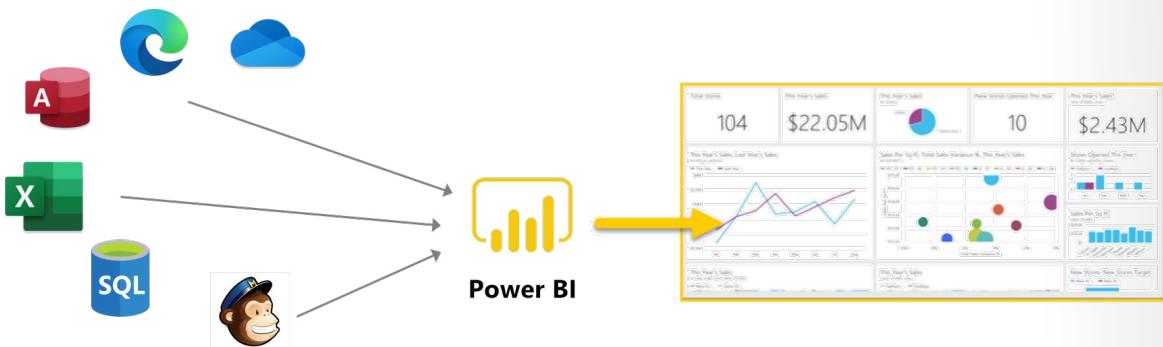
- For impromptu reporting, consider using a combination of Advanced Find and Excel.
- For users, consider using out-of-the-box dashboards and charts.
- For reporting that must be printed or exported, consider creating Word templates and Excel templates.

The solution architect will need to consider reporting tools from other sources that the customer is using.

Power BI should always be considered for reporting and analytics.

## Power BI

Microsoft Power BI is a collection of software services, apps, and connectors that work together to turn your unrelated sources of data into coherent, visually immersive, and interactive insights. Whether your data is a simple Microsoft Excel workbook or a collection of cloud-based and on-premises hybrid data warehouses, Power BI helps you connect to your data sources, visualize (or discover) what's important, and share that with anyone you want.



Power BI can be simple and fast, capable of creating quick insights from an Excel workbook or a local database. However, Power BI is also robust and enterprise-grade, ready for extensive modeling and real-time analytics and for custom development. Therefore, while it can be your personal report and visualization tool, it can also serve as the analytics and decision engine behind group projects, divisions, or entire corporations.

A key advantage of Power BI is the ability to include many data sources such as Salesforce, tables that are embedded in webpages, Microsoft Access, Excel, SQL databases, Mailchimp, and Dataverse.

Power BI is a self-service platform that helps users interact with prebuilt datasets and reports and to create their own visualizations.

The fundamental components of Power BI are:

- **Datasets** - A collection of data from one or more data sources that are cleansed, transformed, and modeled.
- **Visualizations** - A visual representation of data, such as a chart or a color-coded map. Power BI has many types of visualizations.
- **Reports** - A collection of visualizations on one or more pages.
- **Dashboards** - A collection of visuals on a single page that you can share with others. Dashboards can be embedded in Power Apps.
- **Tiles** - A single visualization on a dashboard. Tiles can be embedded in Power Apps.
- **Apps** - A collection of reports and dashboards that can be shared.

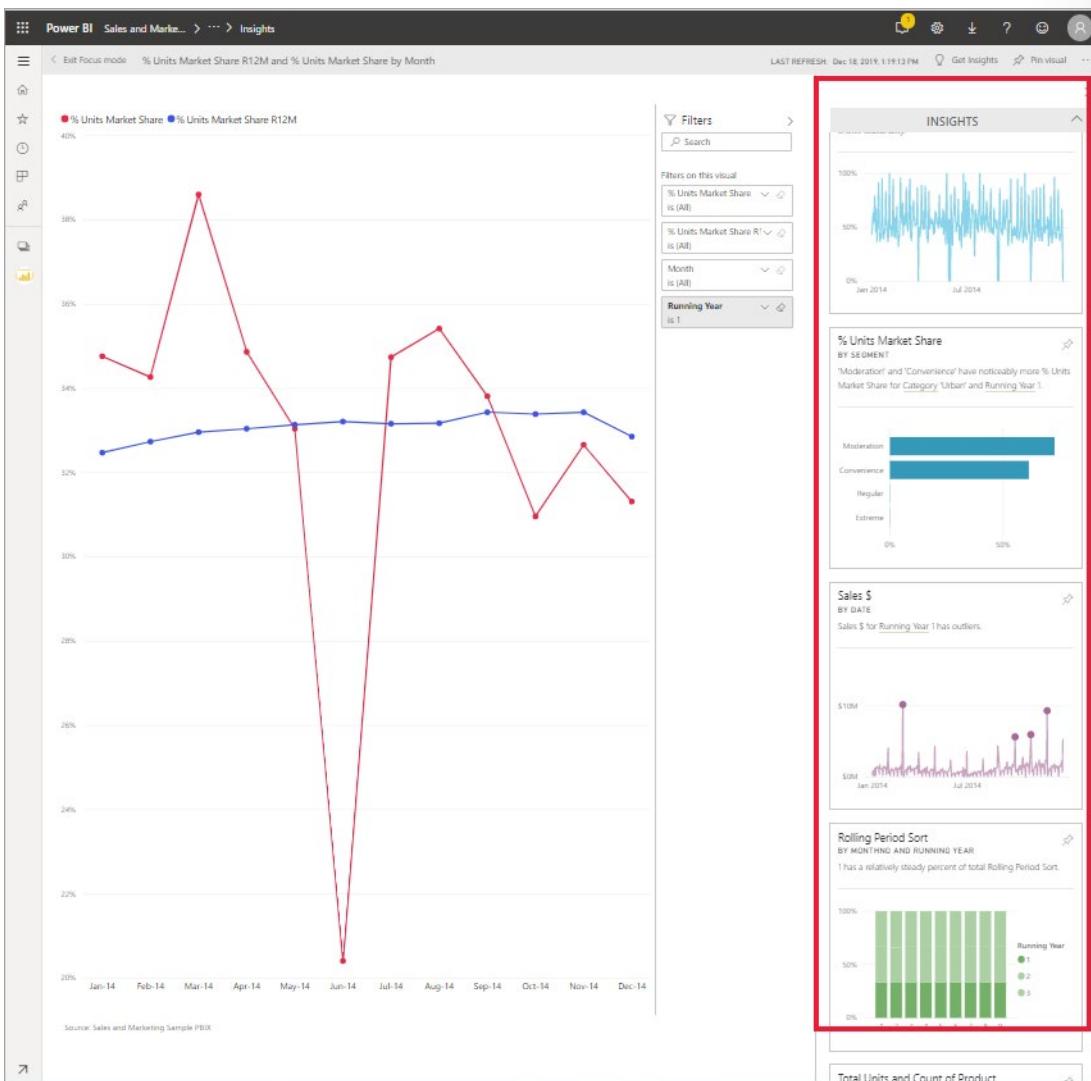
## Power BI capabilities

Power BI offers a range of out-of-the-box visualization options that are available directly from the **Visualizations** pane. When you select the fields that you want to display in a visualization, you can experiment with the different visualization types to find the one that best suits your needs.

Common Power BI visualization types:

- **Bar and column** - Various bar and column chart visualizations that present specific data across different categories in a stacked or clustered format.
- **Table** - A grid that contains related data in a logical series of rows and columns.
- **Line and area** - Help present trends over time.
- **Pie and donut** - Divide the data into segments.
- **Treemap** - Displays data as a set of nested rectangles. Each level of the hierarchy is represented by a colored rectangle (branch) containing smaller rectangles (leaves).
- **Waterfall** - Shows a running total as values are added or subtracted, which is useful in displaying a series of positive and negative changes.
- **Scatter** - Effective when comparing large numbers of data points without regard to time. You can add a play axis to animate how the data changes over time.
- **Map** - Geographic map with data shown as bubbles.
- **Card** - A single data point.
- **Gauge** - A circular arc that displays a single value that measures progress toward a goal or target.
- **Analytics** - Help you track progress toward a specific goal over time.

The quick insights feature generates visualizations based on your data. Power BI applies a set of sophisticated algorithms to discover potentially interesting trends and patterns and then generates visuals that you use.



Some AI powered visualizations that a solution architect should consider include:

- **Q&A** - Allows users to ask natural language questions and get answers in the form of a visual.
- **Key influencers** - Helps you understand the factors that are affecting a specific metric. It analyzes your data for you, ranks the factors that matter, and then displays those factors as key influencers. The visual also helps you contrast the relative importance of these factors.
- **Decomposition tree** - Automatically aggregates data and lets you drill down into your dimensions so that you can view your data across multiple dimensions. The decomposition tree can be used to conduct root cause analysis.

A common method for sharing Power BI visualizations is to create a Power BI app. Apps can be shared internally and externally. Power BI apps can be embedded in other applications. You can see an embedded Power BI app on the **Analytics** tab in the Microsoft Power Platform admin center.

## Work with Dataverse data

Two options for using Dataverse data in Power BI are:

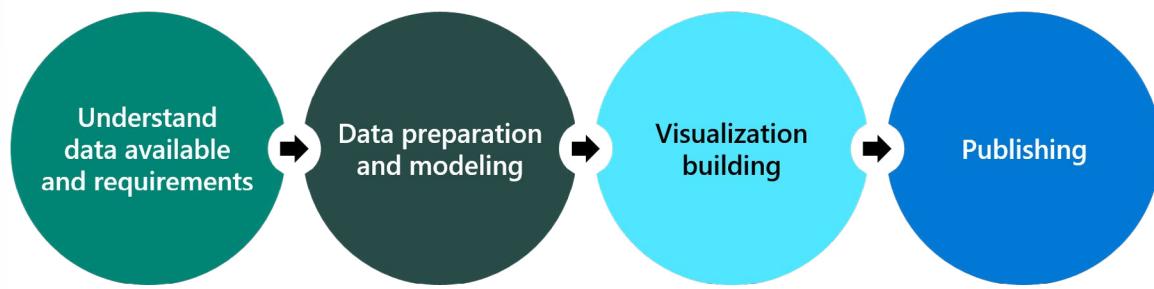
- **Common Data Service (legacy) connector** - The earlier version of the connector that uses the OData connection to Dataverse. You should use this connector for large datasets that are greater than 80 MB. This version also supports paging of the query results and building reports that use the image data type. OData connections import data into a dataset and are refreshed on a regular interval.
- **Dataverse connector** - This option uses the new Tabular Data Stream (TDS) protocol to connect to Dataverse. This connector can either import the data into the dataset or use Direct Query, which permits real-time data.

[!NOTE]

You can also use dataflows, which are described later in this module.

The process for creating Power BI visualizations is shown in the following diagram.

## From data to visualization



After you have connected to a Dataverse environment, perform the following series of activities to prepare and model the data:

- Select tables
- Set fields as the right data type
- Filter data
- Clean and transform data
- Manage relationships
- Use Data Analysis Expressions (DAX) to create analytics

Cleanup and transformation are required because it helps make data consumption easier. For instance, if you perform cleanup and transformation in your dataset by providing column names, using labels for choice columns, using lookup data, and handling dates, the dataset will be easier for others to consume.

## Handle dates

Power BI uses dates in visualizations extensively. Power BI automatically recognizes data columns and creates hierarchies by week, month, and year. You can use these hierarchies as standard. However, the solution architect should consider the creation of a date, calendar, or table, and then link dates to this

table. This approach is a best practice in Power BI. A date table creates a row for every date within a range.

A calendar table will:

- Allow visuals to show zero values when no data is provided for a date.
- Provide more options for slicing and sorting data according to dates.
- Permit different granularity. For instance, it allows a comparison between a table that is based on month with another table based on days.
- Provide more options for using DAX date formulas. The SAMEPERIODLASTYEAR or LASTQUARTER type calculations will not work without a date table.
- Allow weekends and holidays to be excluded from calculations.

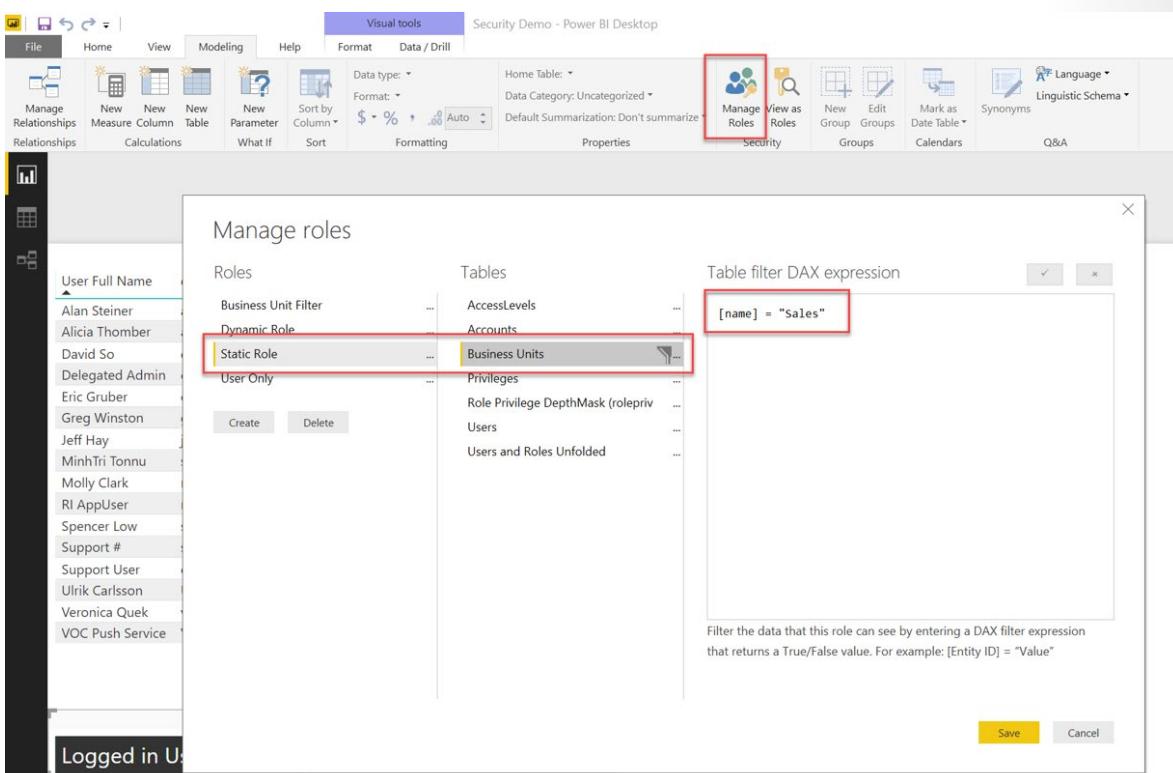
[!IMPORTANT]

Dates in Dataverse are stored in Coordinated Universal Time (UTC).

## Security

When using Power BI, you need understand how security on rows is applied. The security model is managed by Power BI; the Dataverse security model with security roles and business unit hierarchy does not apply on data that is imported into a Power BI dataset.

Power BI uses the concept of row-level security (RLS) to restrict data access for given users. Filters restrict data access at the row level, and you can define filters within roles. RLS is configured in Power BI desktop by using rules.



Characteristics of row-level security are that it:

- Allows different levels of access for different users to the same content.

- Is based on DAX formulas.
- Applies security logic at *row level*.
- Is a True/False DAX evaluation for each row.

[!NOTE]

If you use the TDS endpoint with Direct Query, then the Dataverse security model is applied to data in Power BI.

## Alerts

Power BI can generate alerts from three visuals: gauge, card, and analytics on a Power BI dashboard. In Power BI service, a rule is added that specifies the threshold for the value and how often it is checked. A notification is triggered every time that the condition is met.

Alerts show in Power BI service, display on the Power BI mobile app, are sent by email, and optionally can trigger a Power Automate cloud flow.

## Workspaces vs. environments

Power BI does not use Microsoft Power Platform environments. Power BI uses workspaces to create separate spaces for datasets, reports, and dashboards. Workspaces are similar to environments. Users can be given roles in a workspace to allow collaboration when building Power BI components. A user doesn't need to belong to a workspace to have a report or dashboard shared with them.

## Application lifecycle management

Power BI resources can't be included in Microsoft Power Platform solution packages, and deployment needs to be managed separately.

Power BI has its own application lifecycle management (ALM) process (available with Power BI Premium only) to move Power BI components from one workspace to another.

The three stages of the ALM process are:

- **Development** - Design, build, and upload new content with fellow creators. This stage is first in deployment pipelines.
- **Test** - Test your app to see how it will look for your users with larger volumes of data.
- **Production** - Share the final version of your content with business users across the organization.

## Requirements

Before creating datasets in Power BI, the solution architect should understand the data and requirements in relation to Power BI capabilities.

## Location of the data

Reporting highly depends on the data structure and volumes, in addition to user expectations. The solution architect should ask the following questions about the data that is required for reporting:

- Is all data coming from Dataverse?
- Is reference data available that might be useful?

- What is the volume of data?
- How frequently will data need to be refreshed?
- How is the data related?
- What are the key fields that are needed to relate data?
- What are the key fields that contain analytics?
- How sparse are key data fields?
- What data can be left behind?

[!IMPORTANT]

The size of data and refresh frequency can impact licensing.

## Who consumes the output

The solution architect needs to understand the requirements about who will be looking at Power BI reports and dashboards, and then they should ask the following questions:

- Who will be consuming the Power BI visualizations?
- Are data security considerations available regarding use of data?
- Will people use the Q&A feature?
- Are plans in place for using alerts?
- Are plans in place for using mobile?
- Do external stakeholders exist?
- Are licensing considerations necessary?

## Required security

Solution architects should understand what sets of data that each user is permitted to see in Power BI. A key consideration is whether the data that is shown in Power BI should be the same as the data that can be seen through the provided apps or whether the solution architect should be able to see more summarized data.

Other security considerations:

- Power BI content in shared model-driven app dashboards must also be shared in Power BI.
- Shared dashboards are refreshed in context of the report owner, not the user who is viewing the dashboard.
- Determine whether row-level security (RLS) needs to be configured or not.
- Assess if you should exclude or obfuscate secure fields and personal data.
- Examine aggregating data in the query.
- Limit the export data options in Power BI.

[!IMPORTANT]

Power BI and Dataverse have no synchronization privileges. Consequently, you could inadvertently grant other people access to confidential data in Power BI.

Consider the frequent scenario where:

- The user should only have access to their own data.
- Managers should only have access to their own team's data.
- Executives should have access to all data.
- Data has been imported into a dataset and combined with data from other systems.

To meet these requirements, you will need to limit access to rows based on assigned roles in the dataset by using RLS.

## Power Platform

Power BI can be used with the other components of Microsoft Power Platform.

Power BI content can be:

- Consumed in PowerBI.com.
- Consumed from Power BI mobile and Power BI data alerts.
- Embedded in Power Apps and Microsoft Teams.

To embed Power BI in a model-driven app, an administrator must first enable Power BI visualization embedding for the environment. You can add an entire Power BI dashboard to a model-driven app, or you can add a tile from a Power BI dashboard to a Dataverse dashboard. To add a Power BI dashboard or tile, the maker will select the workspace, dashboard, and tile.

The Power BI dashboard that is used must be shared with the users of the model-driven app. If the Power BI dashboard is not shared, the user will see errors, as shown in the following screenshot.

The screenshot shows a Power BI dashboard titled "My new Dashboard". On the left, there is a bar chart titled "Top 5 Est. Revenue by Account" with the following data:

Account	Avg (Est. Revenue) (\$)
Adventure Works	\$1,982,372.00
Humongous Insurance	\$2,095,680.00
Blue Yonder Airlines	\$2,187,500.00
Northwind Traders	\$3,482,225.00
A. Datum	\$3,750,000.00

On the right, there is a list of records with the following data:

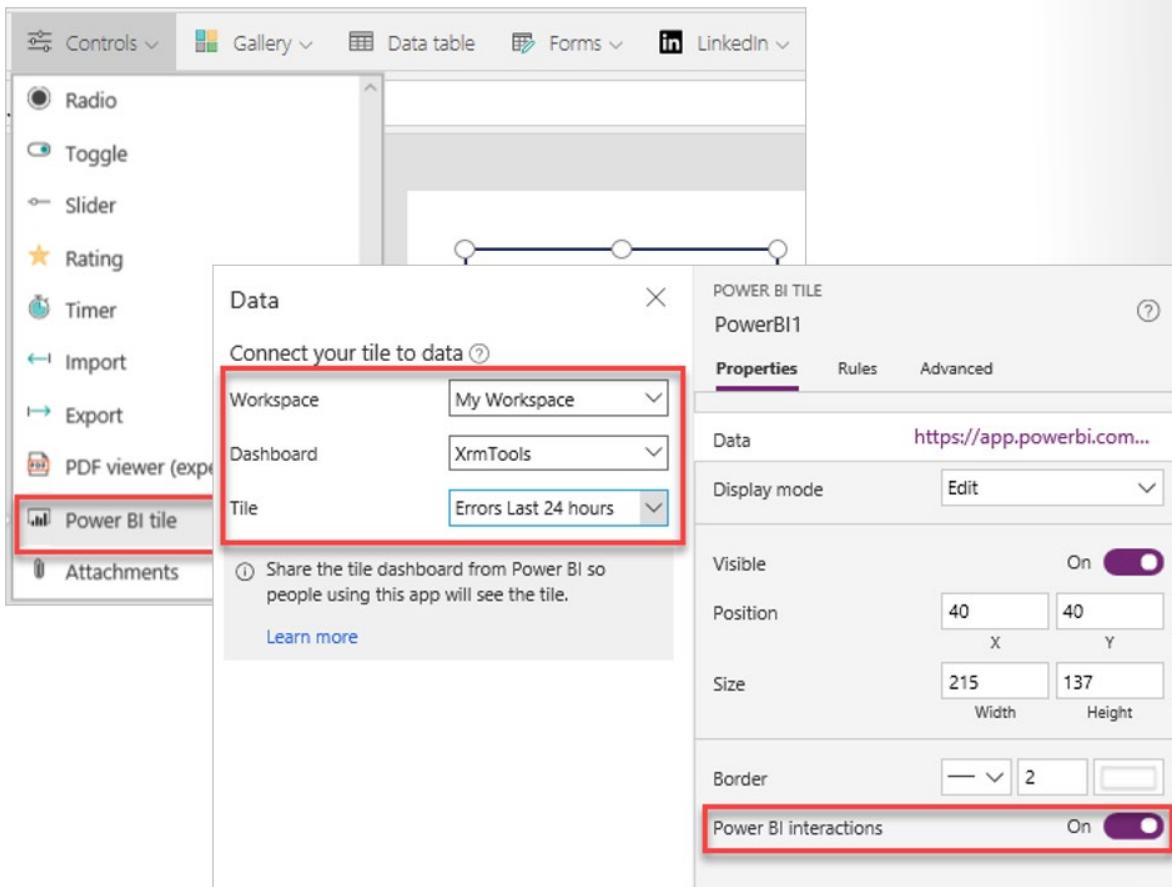
Topic	Account	Owner	Est. Close Date
Expressed interest in A. Datum X-line of printers - Edit...	Northwind Trad...	SYSTEM	12/12/
Interested in our newer offerings	A. Datum	SYSTEM	7/9/
Video Hardware Upgrade	Adventure Works	SYSTEM	11/30/
Refrigeration Smart Sensors	TailsSpin Toys	Ulrik Carlsson	12/19/
4G Enabled Tablets	Southridge Video	Spencer Low (S...	5/6/
Home Electronic Systems Upgrade	Fourth Coffee	Sven Mortensen...	5/12/
Home PC	Adventure Works	Ulrik Carlsson	8/31/
Notebooks	City Power & Li...	SYSTEM	8/31/



### [!NOTE]

Power BI dashboards and tiles can now be added to system dashboards and user dashboards.

To embed a Power BI tile in a canvas app, the maker will add the Power BI control to the screen and then select the workspace, dashboard, and tile, as shown in the following screenshot.



The Power BI dashboard that the tile is from must be shared with the users of the canvas app. If the Power BI dashboard is not shared, users will see an error.

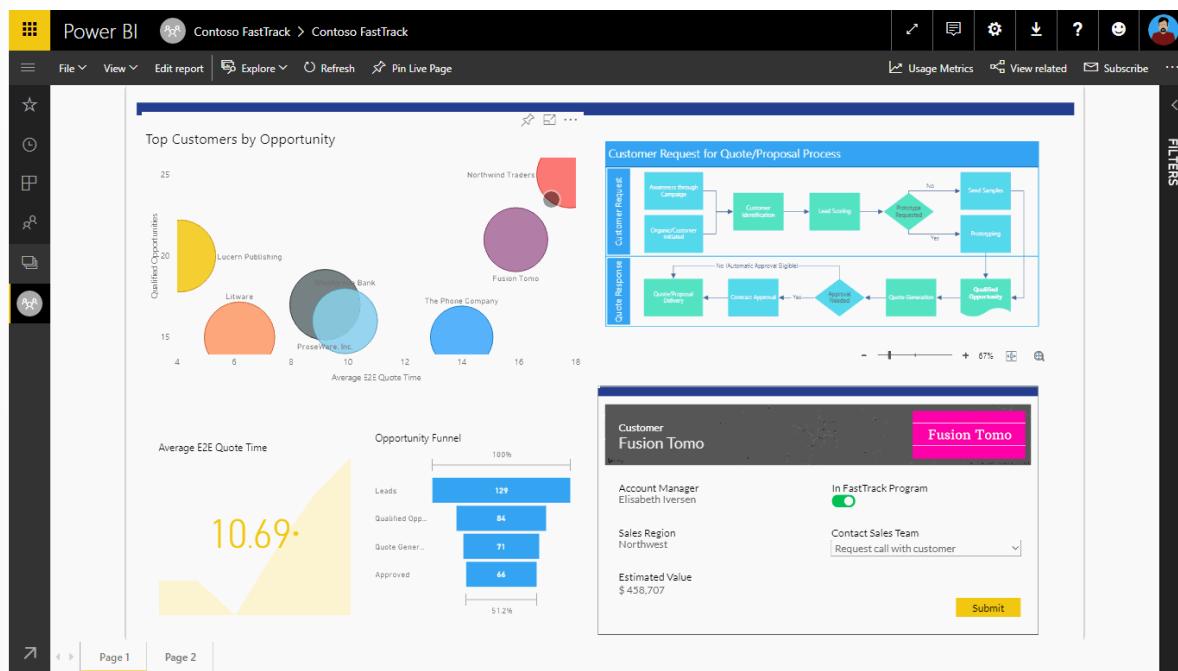
We do not recommend that you have more than three Power BI tiles loaded at the same time within an app. The **LoadPowerBIContent** property controls the loading of Power BI content. When this property is set to **true**, the Power BI content is loaded and shown. When the property is set to **false**, the Power BI content is unloaded, which releases memory and optimizes performance.

It is possible to pass a parameter of type string to the Power BI tile from the canvas app by modifying the **TileURL** property and appending

```
&$filter=<TableName>/<ColumnName> eq '<Value>'
```

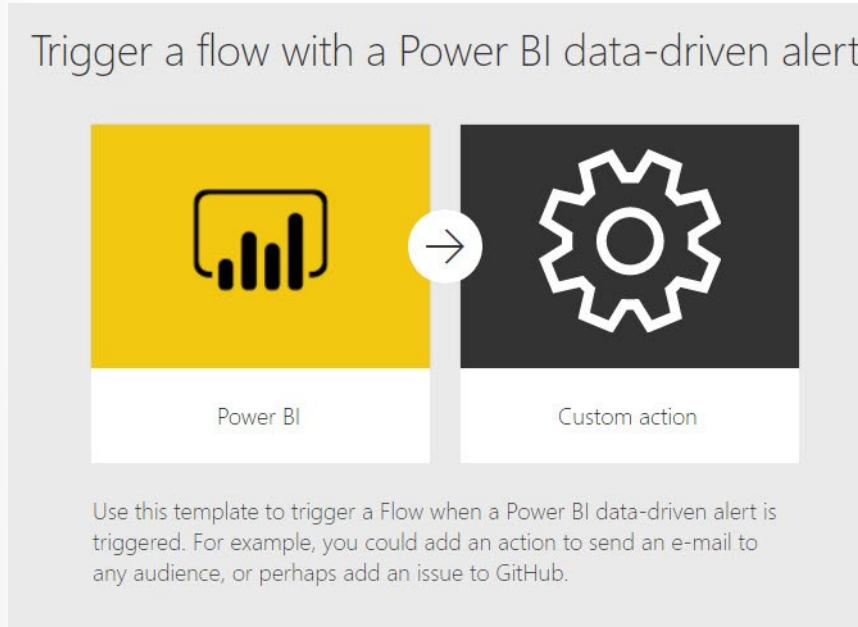
This action allows the tile to be filtered by a value in the canvas app.

A canvas app can be embedded within a Power BI report. The canvas app can interact with the other visuals in the report. A Power Apps visual can be added to the report. The canvas app uses the **PowerBIIntegration** object to integrate the canvas app with Power BI.



## Alerts and Power Automate

Power BI alerts can trigger Power Automate cloud flows. Power BI alerts are rules that trigger an alert when a value exceeds a threshold value. The Power BI alert must be created first and a Power Automate cloud flow will be linked to the Power BI alert.

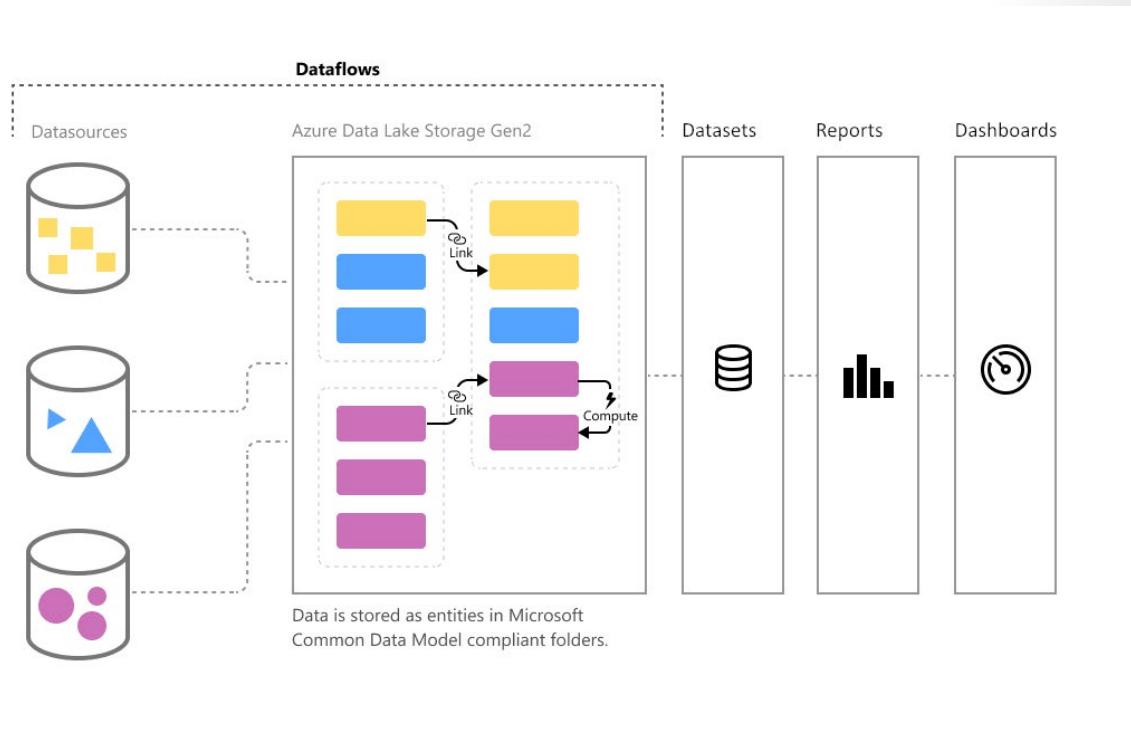


## Power Automate visual

The new visual in Power Automate is **Preview**. This new visual allows a report author to add a button to a Power BI report that can trigger a Power Automate cloud flow.

## Dataflows

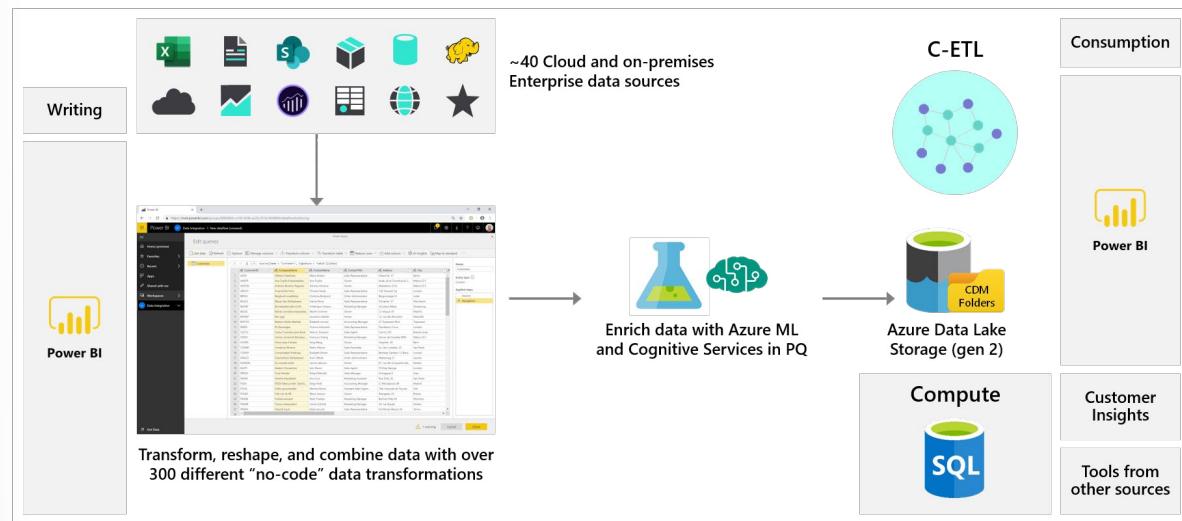
Dataflows are cloud-based extract, transform, and load (ETL) services that help you ingest data from various sources and then convert it into an analysis-ready form. Dataflows can be used by Power BI to process and ingest large volumes of data.



Dataflows are designed to support the following scenarios:

- Create reusable transformation logic that can be shared by many datasets and reports inside Power BI. Dataflows promote reusability of the underlying data elements, preventing the need to create separate connections with your cloud or on-premises data sources.
- Expose the data in your own Azure Data Lake Gen 2 storage, enabling you to connect other Azure services to the raw underlying data.
- Create a single source of the truth by forcing analysts to connect to the dataflows rather than connecting to the underlying systems, providing you with control over which data is accessed and how data is exposed to report creators. You can also map the data to industry standard definitions, enabling you to create tidy curated views, which can work with other services and products in Microsoft Power Platform.
- If you want to work with large data volumes and perform ETL at scale, dataflows with Power BI Premium scales more efficiently and gives you more flexibility. Dataflows support a wide range of cloud and on-premises sources.
- Prevent analysts from having direct access to the underlying data source. Because report creators can build on top of dataflows, it might be more convenient for you to allow access to underlying data sources only to a few individuals and then provide access to the dataflows for analysts to build on top of. This approach reduces the load to the underlying systems and gives administrators finer control of when the systems are loaded from refreshes.

Dataflows can transform data by using Power Query and enrich data by using Microsoft Azure Cognitive Services and Azure Machine Learning, as shown in the following diagram.



After you've created a dataflow, you can use Power BI Desktop and Power BI service to create datasets, reports, dashboards, and apps that use Common Data Model to drive in-depth insights into your business data.

Data that is exported from Dataverse into Azure Data Lake Storage can be imported into Power BI by using dataflows.

## Dataset vs. dataflow

Dataflows are optional but datasets are not. You can't use a dataflow directly; the dataflow populates the dataset.

If you don't use a dataflow and only use datasets, the dataset preparation is done repeatedly in each dataset. With dataflows, the dataflow performs the data preparation once, and then each dataset uses the pre-done prep. Dataflows can also have computed entities that are the needed insights for a report.

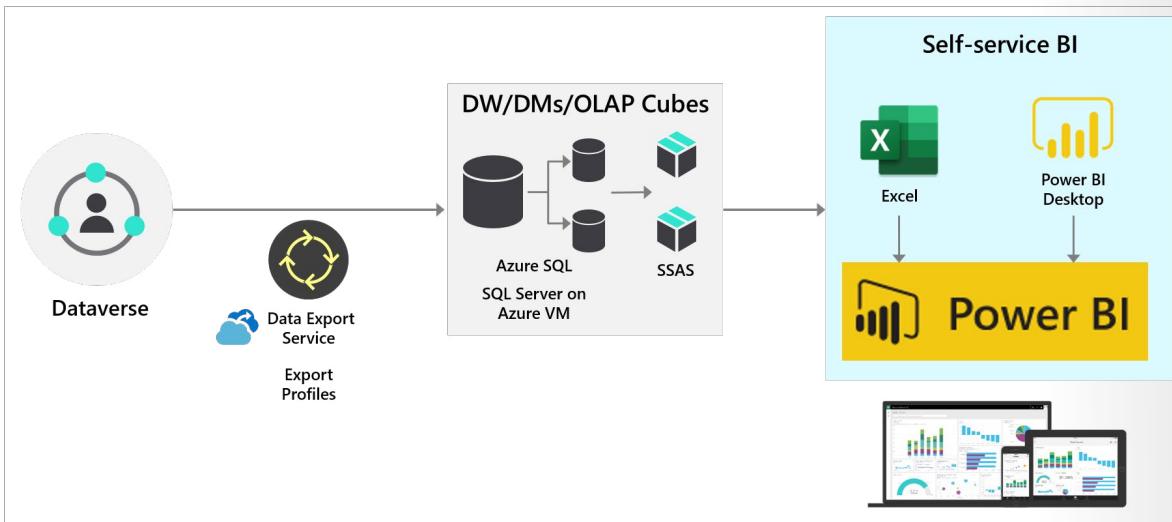
You will always use datasets, but you can choose to use a dataflow if it helps.

## Enterprise BI

Enterprise BI, where data is extracted for use in broader enterprise reporting tools, has many benefits:

- Reduces load on operational data stores
- Allows for transformation and historical analytics
- Allows for use with reporting tools from other sources

The following diagram shows use of Data Export Service to extract data from Dataverse and populate a data warehouse that reporting tools such as Power BI can use.



[!NOTE]

You could substitute Export to Azure Data Lake in place of Data Export Service.

## Artificial intelligence

AI refers to computers thinking and acting in a way that simulates a human. AI is a technology that takes information from its environment and responds based on what it learns. The goal of AI is to create a machine that can mimic human behavior.

AI is more than learning; it is knowledge representation, reasoning, and abstract thinking. Machine learning is the subset of AI that takes the approach of teaching computers to learn for themselves rather than teaching computers all that they need to know. Machine learning is the foundation for modern AI and focuses on identifying and making sense of the patterns and structures in data.

Microsoft provides a number of machine learning services to enhance data.

## Dynamics 365 AI apps

Solution architects need to be aware of the prebuilt insights that are available with Dynamics 365 apps, including:

- Dynamics 365 Sales Insights
- Dynamics 365 Customer Service Insights
- Dynamics 365 Customer Insights
- Dynamics 365 Fraud Protection

## Azure Cognitive Services

Cognitive Services is a suite of prebuilt AI services that developers can use to build AI solutions. Cognitive Services meets common AI requirements and allows you to add AI to your apps quickly without expertise in machine learning.

Cognitive Services APIs cover:

- Computer vision

- Natural language processing
- Speech
- Decision
- Web search

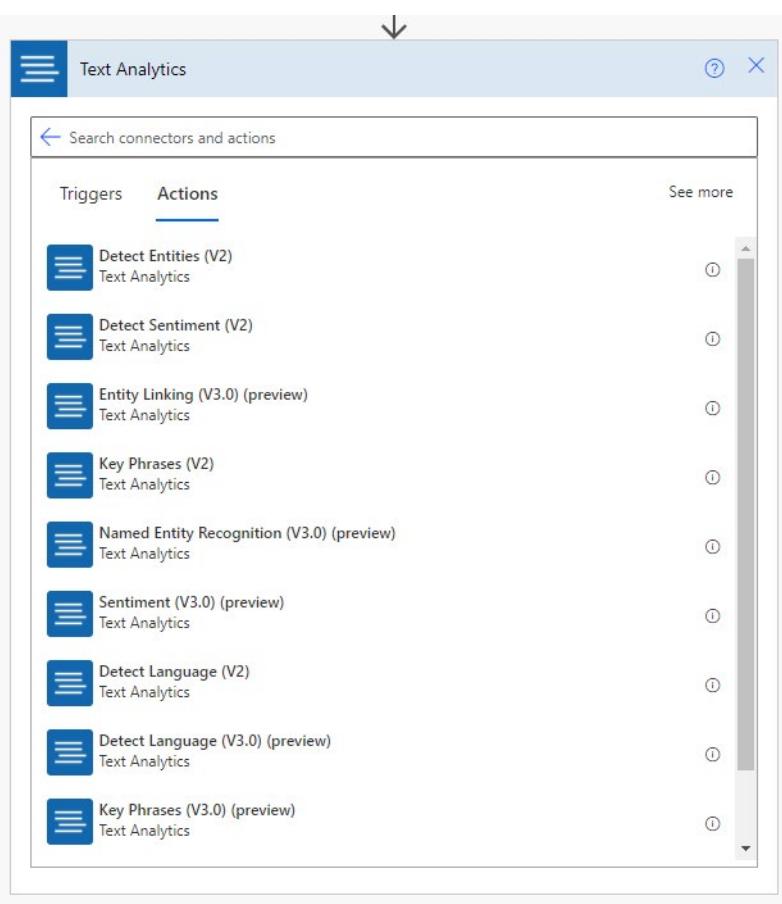
Cognitive Services is available as a set of REST APIs that can be consumed by applications. Essentially, Cognitive Services includes off-the-shelf services that help you develop an AI-based solution quickly and with less specialist expertise.

Microsoft has created connectors for Azure Cognitive Services for Power Apps and Power Automate:

- Content Moderator
- Computer Vision
- Custom Vision
- Face
- LUIS (Language Understanding)
- Text Analytics (including Sentiment, Named Entity Recognition, Key Phrase Extraction)
- Translator

[!NOTE]

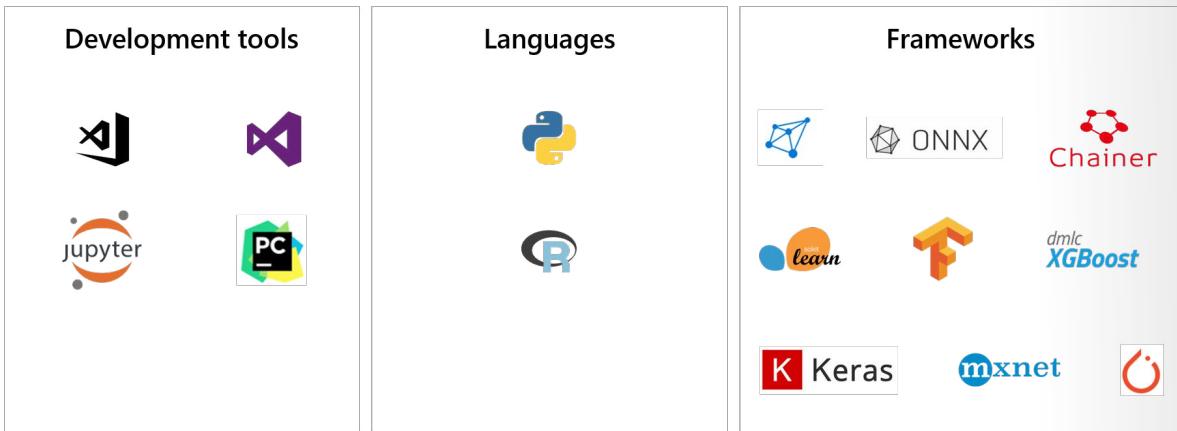
Cognitive Services connectors are premium connectors.



These connectors can be used to enhance data and application functionality.

## Azure Machine Learning

Azure provides many different services to help you create your own machine learning models when Cognitive Services doesn't meet your needs. You can build machine learning models by using many different tools, languages, and frameworks.



Machine learning is beyond the scope of this course. However, solution architects should be aware that Azure Machine Learning can allow developers to implement enterprise grade machine learning for scenarios that are not met by AI Builder or Cognitive Services.

## AI Builder

AI Builder is a component of the Microsoft Power Platform solution that allows you to add AI to predict outcomes to help improve business performance without writing code. You do not need to understand machine learning or learn Python to use AI Builder. Microsoft helps make it easier for you to create AI models and then consume those models in Microsoft Power Platform.

AI Builder takes the concept of Cognitive Services further, enabling anyone to use AI in their apps and flows and to build their own machine learning models without needing expertise in machine learning or having to write code.

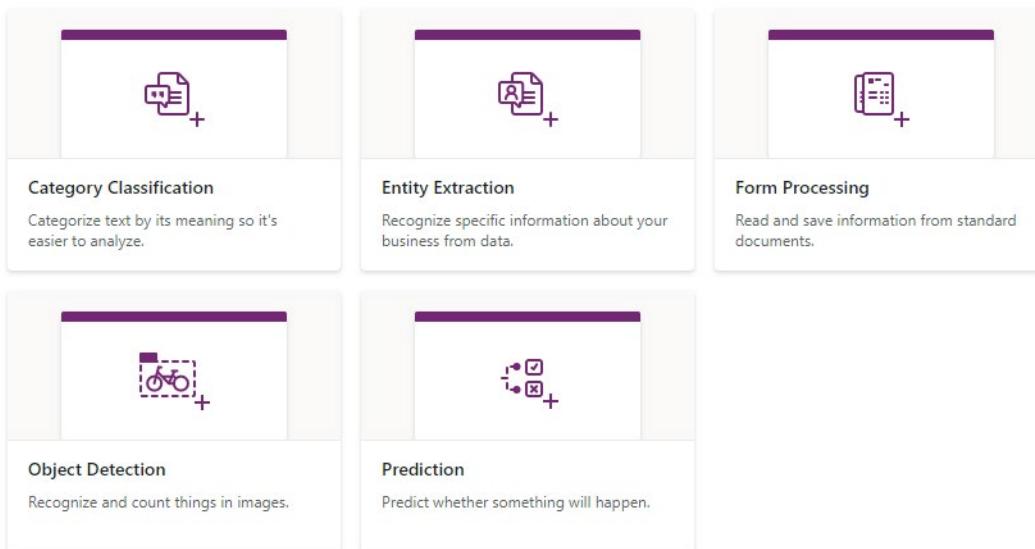
With AI Builder, you can:

- Use one of the prebuilt AI models that are supplied with AI Builder.
- Build and train your own AI model by using your own data.

AI Builder has five model types for prediction, vision, and language:

- **Category Classification** - Performs natural language processing on text data and classifies into separate categories.
- **Entity Extraction** - Recognizes specific data in text data. Entity extraction transforms unstructured text into structured data that can be used in apps and flows.
- **Form Processing** - Reads and extracts information from documents such as invoices and purchase orders.
- **Object Detection** - Finds objects within images.
- **Prediction** - Analyzes patterns in historical data to predict the outcome of new data.

Refine a model for your business needs



AI Builder has the following pretrained models:

- **Category Classification** - Classifies text into categories that are associated with customer feedback, such as compliments, issues, and pricing.
- **Entity Extraction** - Recognizes and extracts standard business objects in data.
- **Key Phrase Extraction** - Identifies the main talking points from a piece of text.
- **Language Detection** - Identifies the language that is used in a piece of text.
- **Sentiment Analysis** - Detects whether the message in a piece of text has a positive or negative emotion.
- **Text Translation** - Translates text from one language into another language.
- **Business Card Reader** - Extracts information from an image of a business card.
- **Text Recognition** - Extracts words from documents and images.
- **Receipt Processing** - Extracts details from pictures of printed and handwritten receipts.

### Get straight to productivity

<b>Business Card Reader</b>  Automatically process business card information	<b>Category Classification (preview)</b>  Categorize text by its meaning so it's easier to analyze	<b>Entity Extraction</b>  Extract entities and their types from text
<b>Key Phrase Extraction</b>  Extract the key talking points from text	<b>Language Detection</b>  Identify the language being used in text	<b>Receipt Processing (preview)</b>  Read and save information from receipts
<b>Sentiment Analysis</b>  Analyze positive/negative sentiment in text	<b>Text Recognition</b>  Automatically process text from images	

Canvas apps can use prebuilt models and custom models to enhance data. You could use an AI Builder model to analyze text that a user has entered. You can take a picture with a canvas app and then use an AI Builder model to extract the text from the image or to detect objects in the image.

You can use AI Builder models in two ways with a canvas app:

- By adding AI Builder model controls to a screen
- By using AI Builder models through the formula bar

Power Automate can use all prebuilt models and any custom models in AI Builder to enhance data. You can trigger a Power Automate flow when a record is created or when an image is stored. An AI Builder connector is available that you can add to a flow to access the models. For example, Power Automate can categorize a new record or predict what will happen to a newly created record.

# Module Summary

## Recap

Reporting and analytics are an important part of the digital feedback loop of an organization. Analytics and insights can be delivered in several different ways with Microsoft Power Platform. The solution architect needs to evaluate the ways for implementing reporting and, when possible, look for opportunities to be more predictive than reactive when looking at reports and analytics.

In this module, you reviewed the role of the solution architect in relation to analytics and AI, including:

- Planning and evaluating requirements
- Reporting capabilities of Microsoft Power Platform
- Power BI
- Enterprise BI

## Next steps

The next steps are to look at the architecture of Power Apps to learn the capabilities of Power Apps when designing solutions to business problems.

# Module 7 Power Apps architecture

## Explore Power Apps architecture

### Introduction

Solution architects are responsible for a solution's overall design. Understanding the capabilities of Microsoft Power Apps is imperative when designing the apps for a solution. This module explains the different types of apps, the differences between apps, and when and which type(s) of apps to use. Additionally, you will learn how to divide apps and discern how many apps that you need.

This module provides technical details around Power Apps but doesn't discuss best practices because they consistently change in Microsoft Power Platform. A solution architect should always check the latest information for Power Apps.

The main objective of this module is to ensure that solution architects learn how to decide which app type to use.

### Types of apps in Power Apps

The three types of apps in Power Apps are:

- **Canvas** - These apps consist of screens, forms, and controls. You have full control over all visual elements of your app through the **Properties** pane, where you can set values and properties, control navigation, and save data by using functions and expressions that are similar to Microsoft Excel.
- **Model-driven** - Apps that are built on the data model that you create in Microsoft Dataverse. Model-driven apps are constructed from separate components, such as forms, views, charts, and dashboards. Each component type has its own editor, and to build an app, you will assemble the separate components.
- **Portal apps** - These apps are customer-facing websites that are built on top of Dataverse. With a portal app, you can expose your data to external stakeholders and allow them to view and update data in a more secure way.

## Patterns

Apps have common patterns that can be beneficial to you in your project.

### Microsoft apps vs. partner apps vs. custom apps

A solution architect should consider using a partner app from Microsoft AppSource. The rest of this module will focus on the creation of custom apps.

The first decision that a solution architect needs to make is which type of app to start with:

- Microsoft Dynamics 365 apps
- A partner app from AppSource
- Custom app(s)

Dynamics 365 has many types of apps that you can use individually or together. You need an understanding of these apps and their capabilities so that you can decide whether to use the out-of-the-box functionality of the apps, customize the apps, or even build your own custom apps. Consider Microsoft Dynamics 365 apps as model-driven apps in Power Apps that are built on top of Microsoft Dataverse.

A solution architect should start with evaluating whether Dynamics 365 apps can meet the requirement or not. If not, then they need to consider AppSource next, and then finally consider custom apps.

### App types

A model-driven app has the following features:

- Dataverse data driven
- Data relationship navigation
- Consistent UI
- Security trimming of UI
- Responsive UI
- Consistent accessibility
- User personalization
- End-user tooling (Excel, import, export, and so on)
- Back/mid office and process focused

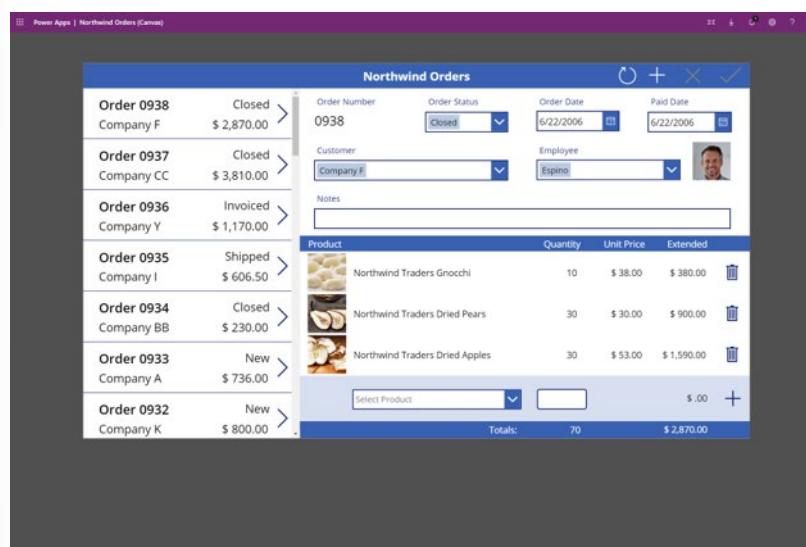
The image displays three separate screenshots of Microsoft PowerApps canvas applications, each showing a different aspect of an innovation management system.

- Innovation Challenge:** This screenshot shows a dashboard with several visualizations. At the top left is a pie chart titled "Active Challenges by Domain" with segments for Industry 4.0 (blue), IoT convergence (orange), New business model (purple), and Big data (yellow). To its right is a bar chart titled "Active Challenges" showing the "Most popular Challenges" with categories like Connected operations, Enterprise sustainability, Connected products, Big data, and IoT themes. Below these are two more charts: "Active Ideas" (Top 3 Ideas) and "Active Challenges".
- Active Challenges:** This screenshot shows a list of challenges grouped by name. The challenges listed are:
 

Name	Number of ideas	Launch date	Accepted new ideas?	Close date	Award issued?
Connected Operations	5	3/22/2018	4/29/2018	5/5/2018	No
Enterprise sustainability	4	4/16/2018	4/30/2018	5/3/2018	No
Connected products	3	10/12/2018	10/30/2018	11/4/2018	No
IoT farming	2	3/1/2018	4/30/2018	5/3/2018	No
Smarter manufacturing	2	5/15/2018	5/29/2018	6/2/2018	No
Big data	2	12/1/2018	12/30/2018	1/8/2019	No
Servitization	1	8/1/2018	8/29/2018	9/2/2018	No
Renewable energy	1	8/1/2018	8/30/2018	10/4/2018	No
Holographic computing	0	7/1/2018	7/29/2018	8/2/2018	No
- Connected Operations:** This screenshot shows a detailed view of a specific challenge named "Connected Operations". It includes tabs for Details, Timeline, and Related. The Details tab shows the challenge's name, description (Industry 4.0 is a term used to describe a collection of technologies and processes, including intelligent machines, connected systems, and the Internet of Things), owner (SYSTEM), and communication via email. The Timeline tab shows a timeline with three stages: Setup, Track (0), and Select And Execute. The Related tab shows a list of contributed ideas, such as Connected quality control, Fleet automation, Integrated service management, and Automobile fuel consumption.

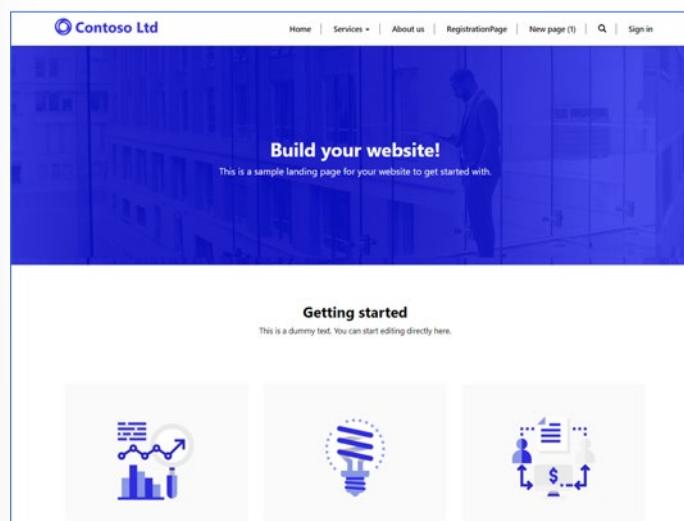
A canvas app has the following features:

- Not Dataverse data driven
- Visual presentation of info
- Custom UI
- Non-responsive UI
- Device integration
- Basic offline support
- SharePoint or Microsoft Teams embedding
- Task focused apps



A portal app has the following features:

- Dataverse data driven
- Web application
- Uses model-driven forms and views as framework to show Dataverse data
- Can be customized with standard web technologies (HTML, JavaScript, CSS, Liquid, and so on)
- External user focused



## Common pattern

When deciding on which app type to use, you should be aware that a solution will typically have multiple apps of different types. The common patterns are:

- **Model-driven** - Administration of data and processes
- **Canvas** - End user focused
- **Portal** - Externally facing portal for non-users

Canvas apps can be used to solve edge cases that would often be ignored.

## Apps in Teams

You can add canvas apps and model-driven apps to Microsoft Teams and then access them from a tab in a channel. This module will explain later how to create apps in Microsoft Teams.

## Embed apps

An embedded canvas app can be embedded in a model-driven app form. This feature enables many different scenarios that can be difficult to achieve without code, such as:

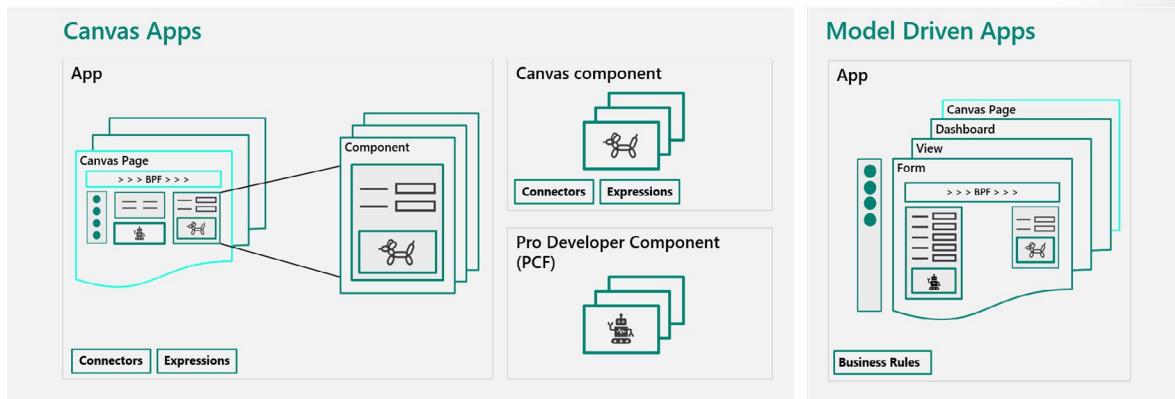
- Including visuals.
- Accessing data that is not in Dataverse by using connectors.
- Using complex logic and operations on Dataverse data.

Limitations of using embedded canvas apps are:

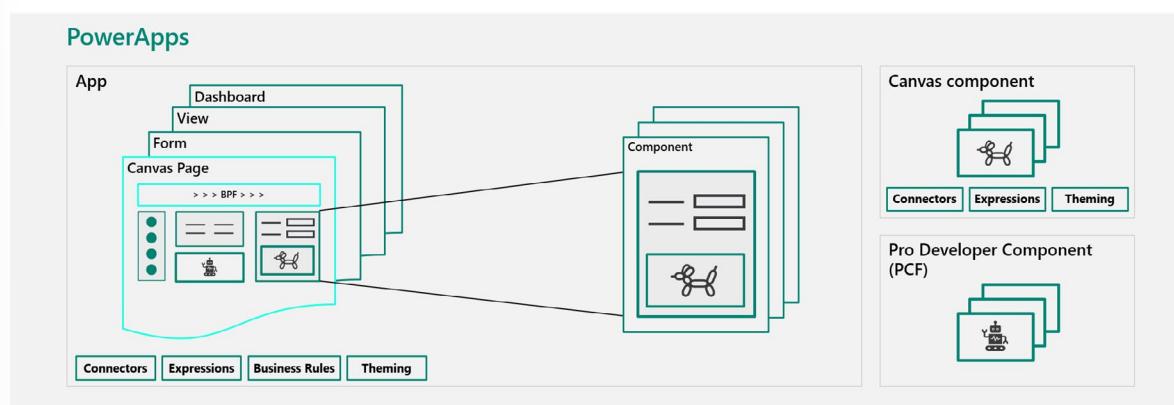
- They are only available during edit; they are not available during creation of new rows.
- The **Save of** form does not save the component data at the same time.

## Roadmap

Currently, canvas and model-driven apps have separate designers, and the integration between canvas and model-driven apps is not seamless. Not all features of each app type are available to the other, as shown in the following diagram.



Microsoft plans to merge and eliminate the differences between canvas and model-driven apps. Ultimately, everything will be merged into Power Apps, with screens made up of canvas and model components, as shown in the following diagram.



## Composition

App composition is one of the most important decisions that a solution architect must make. App composition is the process of determining the number and type of apps. Deciding on how many apps and what features to include in an app are important considerations in the process.

### Decide on the number and type of apps

Solution architects should consider the following principles when choosing which apps to build:

- Large monolithic apps should be avoided.
- Too many small apps can be overwhelming for users if they have to frequently context switch.
- Components can be used by multiple apps, allowing composition of apps that target users with specific needs.
- Offer groups of users targeted mobile apps to save time when they are away from their desks.

#### [!IMPORTANT]

A solution architect needs to consider which user communities will use which app, when they will use the app, and on which devices. It might be beneficial to map the apps in such a way that helps you decide the number and type of app.

### Extend existing apps vs. create new apps

Dynamics 365 apps come with apps like Sales hub and Customer Service hub. You can choose to use and extend these apps, or you can create your own apps.

Using and modifying an existing app might:

- Expose new features when updates are released.
- Include features that you do not need.
- Include components that you can't reuse or replicate in custom apps.

Composing a new app will:

- Provide complete control over what is included.
- Require you to manually add new features to app navigation.
- Enable you to use custom forms and views that you can control the structure of.

## Determine which type of app to make

Model-driven apps are built on top of the data model in the Dataverse environment. Views and detail screens for model-driven apps are based on the data structure. Accordingly, they offer users a more consistent appearance and behavior from one screen to the next without requiring much effort from the app creator.

Model-driven apps are useful in scenarios where the business logic is complex, such as:

- Sophisticated data models.
- Business process management.
- Tracking activities that are associated with data.

Canvas apps can be built with or without a Dataverse database. They use connectors to access data and services. Canvas apps start with a blank screen, like an artist's canvas, and then the creator can manually lay out each screen. This feature gives the creator complete control over the placement of each element on the canvas.

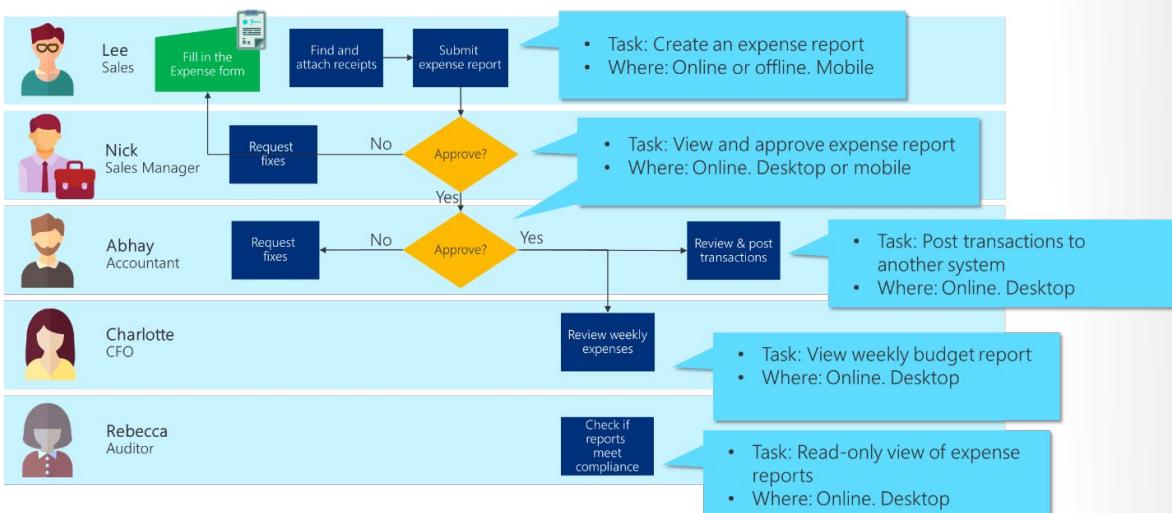
You should choose canvas apps if the user expects a customized user experience and because they offer:

- A graphical, intuitive interface.
- The ability to create a tailor-made UI based on user requirements.
- Integration that spans multiple systems by using connectors.

[!NOTE]

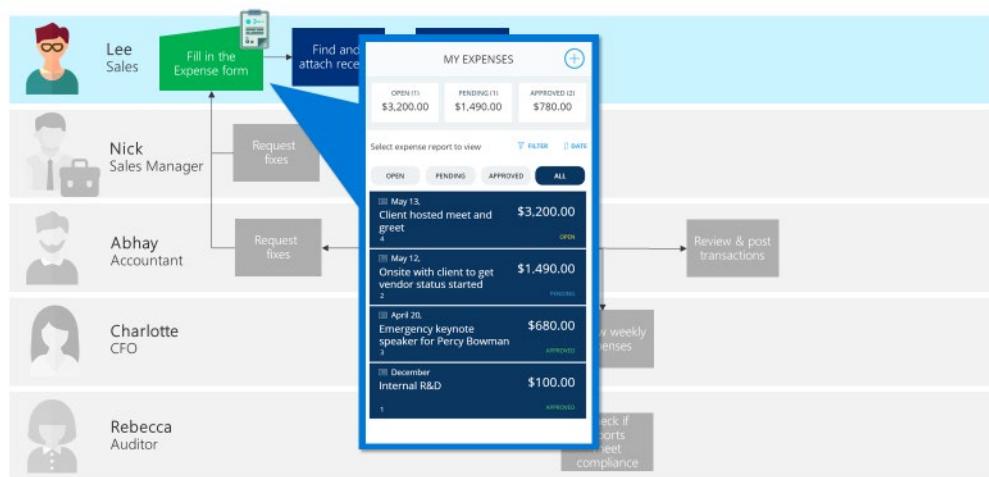
Consider creating a model-driven app unless your users have a specific need for a canvas app. Model-driven apps enable you to make your app quickly because they don't require you to build the UI yourself.

Your business process might require more than one app. In the following example scenario, a solution is required for creating and viewing expense reports. Many tasks have been identified that need to be completed by different user roles.



The expense report solution has several task sets that are different, so you should consider making several apps. The data that people use is the same, but the user experience will be tailored to the specific scenario and personas.

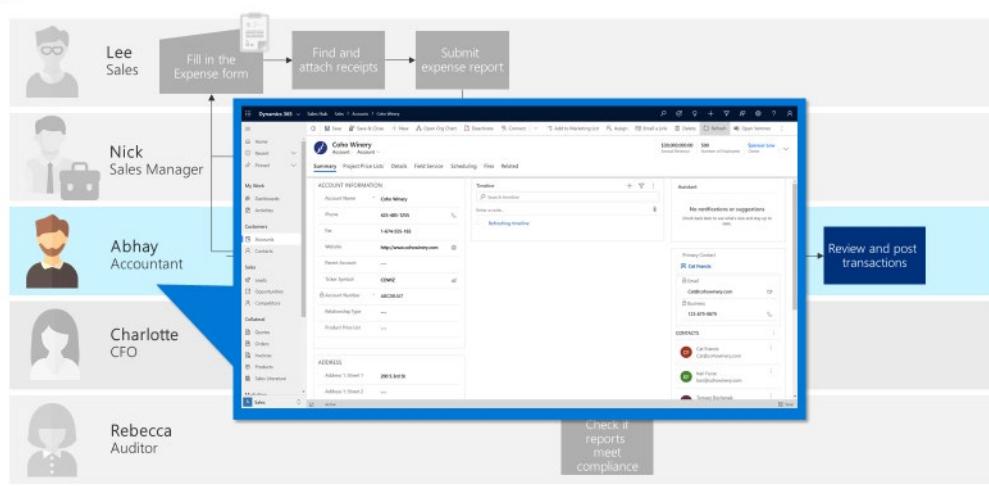
Canvas apps would be the best fit for employees who are filling in the expense form, which enables people like Lee to submit an expense report by using an attractive mobile app that is intuitive to use and can be used when offline.



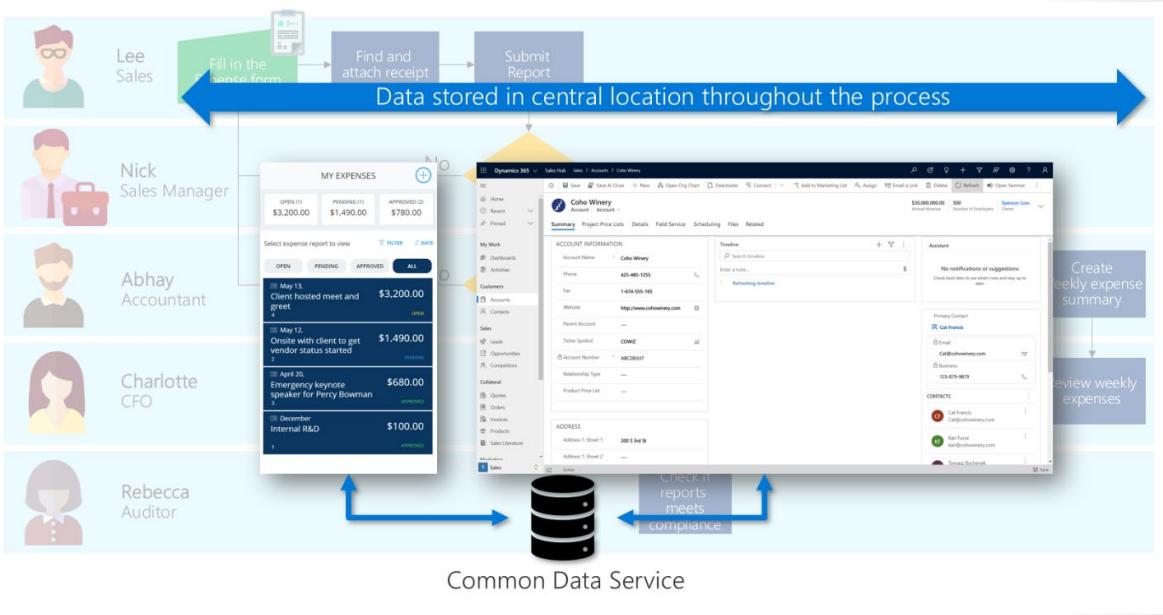
Notice in the preceding image that Abhay has the following requirements:

- Must be able to review all expense reports and receipts
- Responsible for ensuring compliance for every expense report
- Large volume of work; needs to be able to process information quickly
- Must be able to report on how expenses are balancing up to the budget

To be able to process a large volume of work and process information quickly, the best fit for Abhay would be a model-driven app. It allows Abhay to quickly view all details of the submitted expense report, assess how it affects the budget, and look for related information such as vendor details.



This example scenario shows a combination of canvas app and model-driven app. Though they are two different types of apps, all data can be centralized in one place: Dataverse.



## Components

A solution architect needs to consider creating components as part of the app architecture. Components are important because they can promote reuse, enabling multiple makers to collaborate on multiple parts of a canvas app at the build stage.

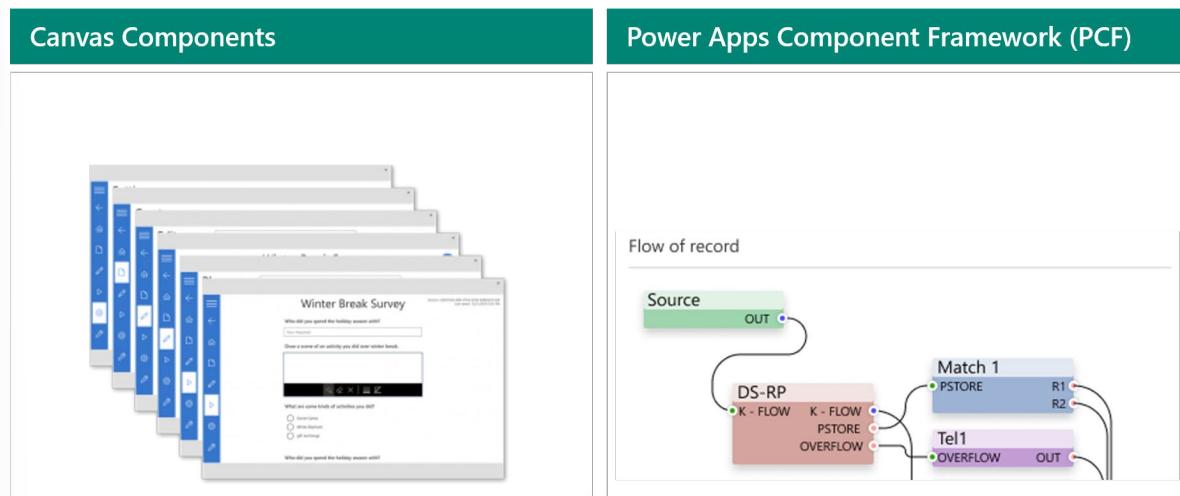
Characteristics of components are that they:

- Allow reuse within and across apps.
- Can allow multiple makers to work on building a single app.
- Help promote consistency and reduce redundancy.

A solution architect should look for controls that would benefit from being made into a component, headers, common widgets, and so on.

Two types of components in Power Apps are:

- **Canvas components** - Targeted for canvas app makers and work only in canvas apps.
- **Power Apps component framework code components** - Targeted for professional developers.



## Component libraries

Canvas components are reusable building blocks for canvas apps so that app makers can create custom controls to use inside an app or across apps by using a component library.

Characteristics of canvas components are that they:

- Are ideal for reuse in canvas apps.
- Can be used across multiple apps.
- Allow use of common maker skills, helping to simplify the process for multiple makers.
- Are limited to capabilities of canvas app formulas and connectors.

Component libraries are containers of component definitions that help make it easier for you to:

- Discover and search components.
- Publish updates.
- Notify app makers of available component updates.

### [!IMPORTANT]

Canvas components should be added to component libraries. Component libraries are the recommended way to reuse components across apps. When you use a component library, an app maintains dependencies on the components that it uses. The app maker will be alerted when the updates to dependent components have become available. Hence, all reusable components should be created within the component libraries.

The solution architect should have a strategy for managing components.

## Code components with Power Apps component framework

Power Apps component framework empowers professional developers and app makers to create code components for model-driven and canvas apps (public preview) to provide an enhanced user experience for the users to work with data on forms, views, and dashboards.

Characteristics of code components are that they:

- Can be used in model-driven and canvas apps in web and mobile.
- Can be used in multiple apps to replace fields and grids.
- Can be used by new record and edit experiences.
- Can typically use multiple components on a form without performance problems.
- Require code developers with TypeScript and HTML skills.
- Allow the use of professional developer skills, meaning that all types of developers can contribute to the solution.

Code components can be packaged in solutions. When the solution is updated, each app that consumes the component automatically recognizes the updated component.

A solution architect should purposefully use code components to create a good user experience.

A large community develops code components. You can reuse these components from the [gallery<sup>1</sup>](#) and [sample component gallery<sup>2</sup>](#).

The following screenshot is for a community control that displays a grid of counts based on calculated risk scores.



## Optimize

Two main ways of development are imperative and declarative. Imperative development focuses on how to achieve the goal, while declarative development focuses on getting the result. Imperative provides more flexibility because you can control every step in the process, but it requires more code and more complexity. Declarative is simpler and more straightforward to use, but you might lack the ability to have the complete control that you want.

Canvas apps takes your declarative "what" and optimizes the "how." You might not be able to precisely express "what," so Power Apps will help by allowing you to use imperative development. A mistake that

<sup>1</sup> <https://pcf.gallery/>

<sup>2</sup> <https://powerusers.microsoft.com/t5/Canvas-Apps-Components-Samples/bd-p/ComponentsGallery>

makers often make is to use imperative development when declarative development will be simpler to use and will perform better.

Canvas apps can be made to look attractive and, while having an attractive app is important, apps that perform well will get better user adoption.

The following sections discuss techniques for optimizing canvas app performance.

## Offload work from apps

As formulas in apps get larger and more complex, consider if work should be done somewhere else. Work can be offloaded to Microsoft Power Automate cloud flows, business rules, plug-ins, and other server-side logic in Dataverse.

### [!NOTE]

A common method is to offload logic into a Power Automate cloud flow that uses the Power Apps trigger. The flow can be called from a Power Apps expression that passes data to the flow and receives a result back from the flow.

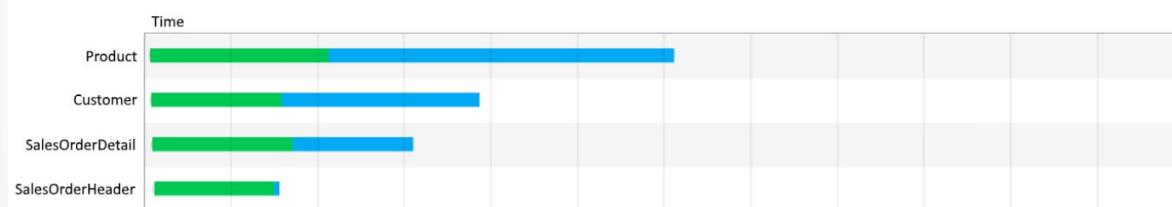
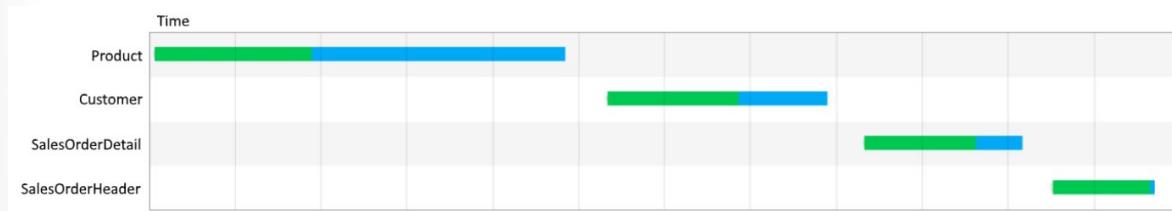
You can also create custom connectors to Microsoft Azure Functions or other custom logic. If you find that you are using imperative development inside an app, consider offloading this logic to a more appropriate feature.

## Performance

Common issues with app performance are:

- **Data access** - Initially, the app gets large sets of data into data collections and then uses the data within multiple screens over client-heavy operations like JOIN, Sort, Add Column, and Group By.
- **Formulas in OnStart** - The app triggers many unnecessary data calls in screens, and these data calls return large data records.
- **Repeatedly retrieving data from the source** - Use the **Set** function to cache data from lookup tables locally.

With OnStart, you should encourage makers to use the **ClearCollect** function to cache data locally and the **Concurrent** function to reduce the time to load the cached data. The first image shows the loading of four datasets without the **Concurrent** function, and the second image shows the process with the **Concurrent** function.



With so many options, performance needs to be considered often. Analysis and improving optimizations are ongoing efforts. You should validate best practices by referring to **slow performance sources<sup>3</sup>**, **common performance issues<sup>4</sup>**, and **performance tips<sup>5</sup>**.

The solution architect should implement a canvas app performance-tuning strategy.



A tuning strategy should:

- Avoid any work than you can.
- Defer work that is less likely to be needed.
- Parallelize work wherever possible.
- Monitor the app in operation; work might not always be obvious.

You should use a progress indicator for the user on long-running work.

## Test Studio, Azure Monitor, and Application Insights

Canvas apps should be properly tested. Microsoft provides Test Studio for regression testing of canvas apps that can be included in automated build processes.

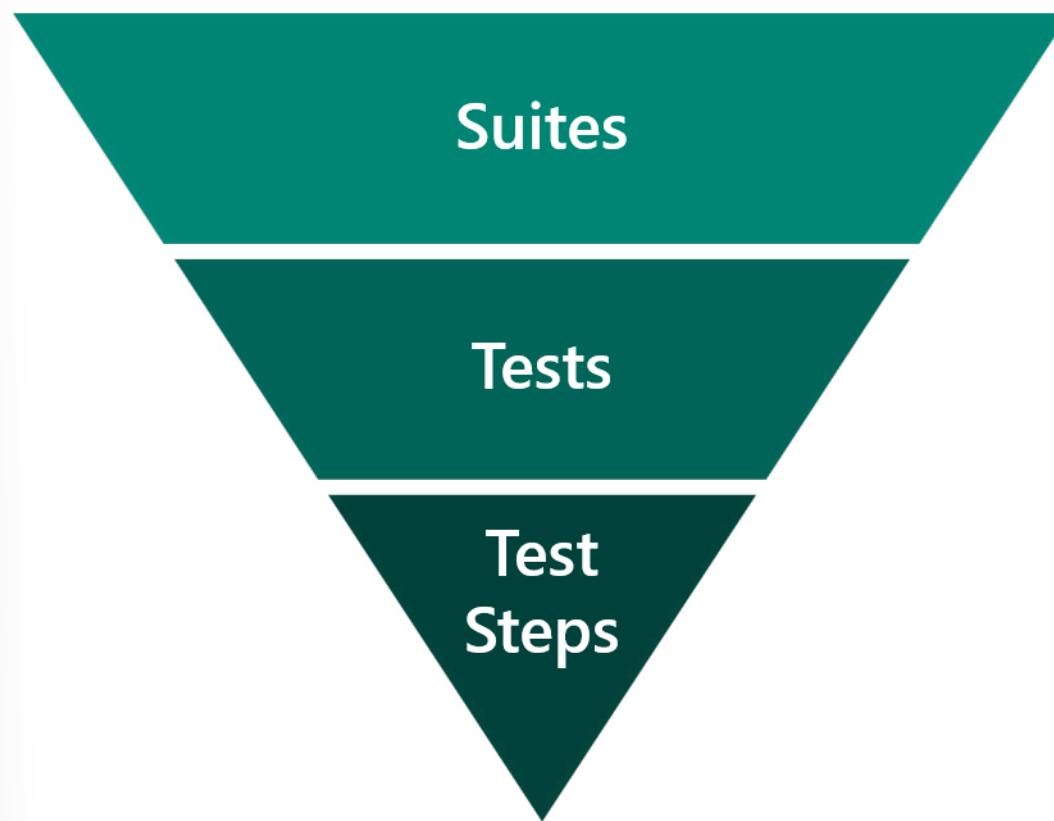
Test Studio includes the following features:

- **Suites** - Test suites are used to organize or group test cases together.
- **Tests** - Test cases are made up of a series of test steps. Test cases are run to validate that your app, or specific features in your app, works as you expect.
- **Test steps** - Instructions or actions. Test steps are written by using the Power Apps expression language.
- **Test assertions** - The expected result of a test.

<sup>3</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/slow-performance-sources>

<sup>4</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/common-performance-issue-resolutions>

<sup>5</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/performance-tips>



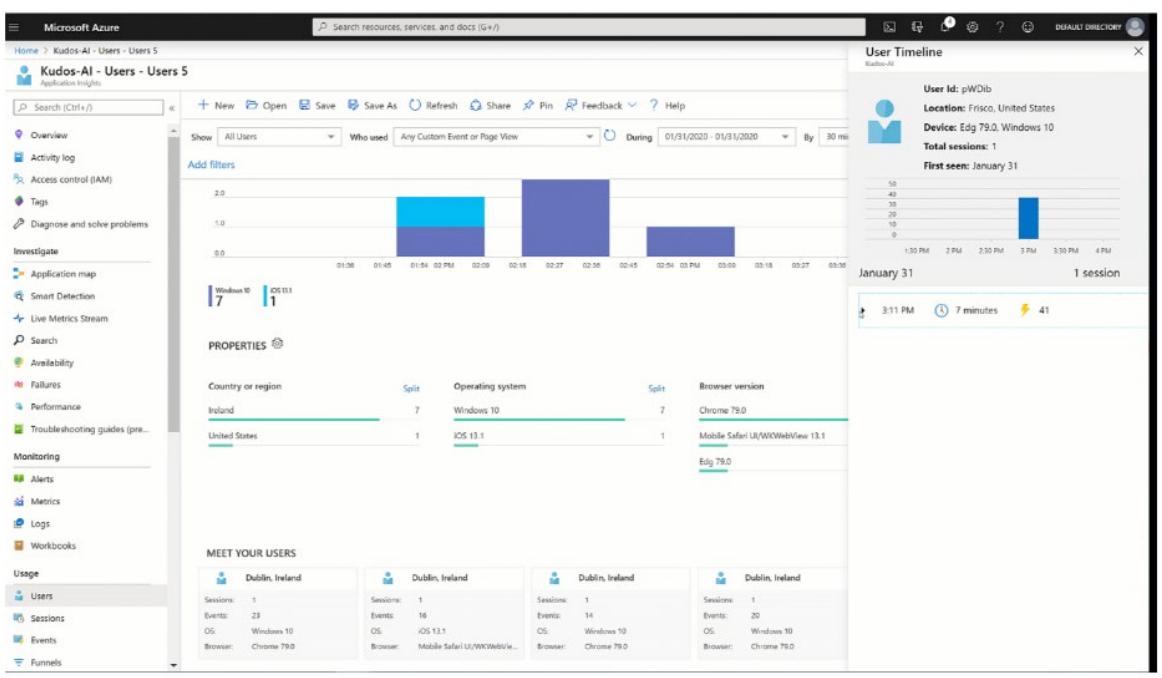
Microsoft Azure Monitor is a tool that offers makers the ability to view a stream of events from a user's session to diagnose and troubleshoot problems. Canvas app makers can use Monitor to view events while building a new app in Power Apps Studio and to monitor published apps during runtime. Model-driven app makers can monitor page navigation, command implementations, form-related issues, and other major actions to understand app behavior and make improvements.

ID	Time	Category	Operation	Result	Status	Duration	Data source	Control	Property	Response
16	17:43:18.904	ScreenLoad	Navigate (Brow...)	Success		150		IconNewItem1	OnSelect	
17	17:43:19.686	UserAction	Select	Success				DataCardValue2	OnSelect	
18	17:43:30.923	UserAction	SetProperty	Success				DataCardValue2	Text	
19	17:43:30.958	UserAction	Select	Success				IconAccept1	OnSelect	
20	17:43:31.264	Network	createRow	Success	Created	201	247	Bikes	IconAccept1	OnSelect
21	17:43:31.452	ScreenLoad	Navigate (Brow...)	Success		155		EditForm1	OnSuccess	1,493
22	17:43:31.471	Network	getRows	Success	200	123	Bikes	BrowseGallery1	Items	3,008
23	17:43:31.603	Network	getAttachments	Success	200	230	Bikes	Attachments_D...	Default	122
24	17:43:32.390	UserAction	Select	Success				IconNewItem1	OnSelect	
25	17:43:32.553	ScreenLoad	Navigate (Brow...)	Success		150		IconNewItem1	OnSelect	
26	17:43:33.887	UserAction	Select	Success				DataCardValue2	OnSelect	
27	17:43:36.620	UserAction	SetProperty	Success				DataCardValue2	Text	
28	17:43:36.670	UserAction	Select	Success				IconAccept1	OnSelect	
29	17:43:36.949	Network	createRow	Success	Created	201	238	Bikes	IconAccept1	OnSelect
30	17:43:37.121	ScreenLoad	Navigate (Brow...)	Success		151		EditForm1	OnSuccess	
31	17:43:37.130	Network	getRows	Success	200	127	Bikes	BrowseGallery1	Items	4,508
32	17:43:37.180	Network	getAttachments	Success	200	203	Bikes	Attachments_D...	Default	122

You can connect your canvas apps with Application Insights, a feature of Azure Monitor. Application Insights includes powerful analytics tools that help you diagnose issues and understand what users actually do with your app.

With your app connected to Application Insights, you can collect telemetry on how users are actually using your app to help you improve the quality of your apps. Some of the telemetry that you can gain from setting up this feature includes:

- Number of active users who are using the app.
- Location of where the app is used.
- Which screens are used most frequently.
- User flow from one screen to another.



As a solution architect, you should decide if Application Insights will be included in the apps that you create.

For more information, see [Application Insights<sup>6</sup>](#).

## Teams

Microsoft Teams is becoming a key collaboration app that is the focus of a user's daily work. To prevent the need for users to switch between applications when accessing their data, it is possible to embed existing Power Apps in Microsoft Teams channels.

Microsoft Teams includes a Power Apps application that, after it's been added, will enable canvas apps to be created with a team. This action creates an environment and Dataverse for Teams database. The maker can create an app within a team that all members of the team can access.

## Create Power Apps in Microsoft Teams

Dataverse for Teams is a built-in, low-code data platform for Teams. The Power Apps application in Teams provides an integrated experience for app makers to create and edit apps within Teams and quickly publish and share them for anyone on the team to use, without having to switch between multiple apps.

<sup>6</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/application-insights>

and services. With Power Apps Studio embedded in the Power Apps application in Teams, and the built-in data platform providing a user-friendly, editable data table, you can quickly build apps based on custom data tables that are Teams-specific and scenario-specific.

Dataverse for Teams, which is built on Microsoft Dataverse, provides relational data storage, rich data types, enterprise-grade governance, and one-click solution deployment to the Teams app store. This version of Dataverse does not include all functionality of Dataverse.

The advantage of creating canvas apps inside of Teams is that no other license is required to use the app and data. If you have access to Microsoft Teams through a Microsoft 365 license, then you can use this feature.

Using Dataverse for Teams might seem attractive initially, but as a solution architect, you need to be aware of the limitations, particularly around integration, such as:

- No API access
- Some controls are not supported
- Canvas apps only
- AI Builder is not supported
- Custom connectors are not supported

For more information, see **known issues and limitations<sup>7</sup>**.

## Portals

The ability to easily show and interact with Microsoft Dataverse data on an externally facing website is the core benefit for implementing a Power Apps portal.

Power Apps portals are designed for interaction with the internal and external audiences.

## Features of portal apps

Power Apps portals are built on top of Dataverse. This architecture comes with a major benefit. All differentiating features of model-driven apps in Power Apps are the features of Power Apps portals as well, including:

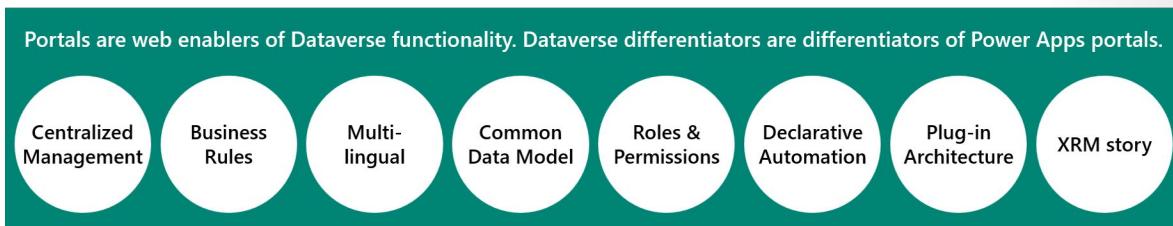
- Centralized management
- Common Data Model
- Roles and permissions
- Forms and views
- Business rules
- Declarative workflows and actions
- Plug-in architecture
- Integration with other services
- Dataverse extensibility
- Audit

---

<sup>7</sup> <https://docs.microsoft.com/powerapps/teams/known-issues-limitations>

Power Apps portals deliver a complete content management system out of the box, with all content stored in Dataverse. As a result, content can be edited through Power Apps portals Studio and also directly by using the Portal Management app. Additionally, the robust Dataverse security model can help secure the content.

Accessibility	Charts and Graphs	Entity Permissions	Polls	Sitemap
Ads	Content Publisher	Invitations	Profile Management	Web Design
Azure Integration	Discussion Forums	Knowledge Management	Redirects	Web Forms
Bootstrap Design	Entitlements & SLAs	Links	Responsive Design	Web Pages
Branding	Entity Actions	Location Finder	UX Localization	Web Roles
Case Deflection	Entity Forms	Ratings and Comments	Search and SEO	Web Templating
Case Management	Entity Lists	Open Data Protocol	Multi-lingual Content	Power Automation

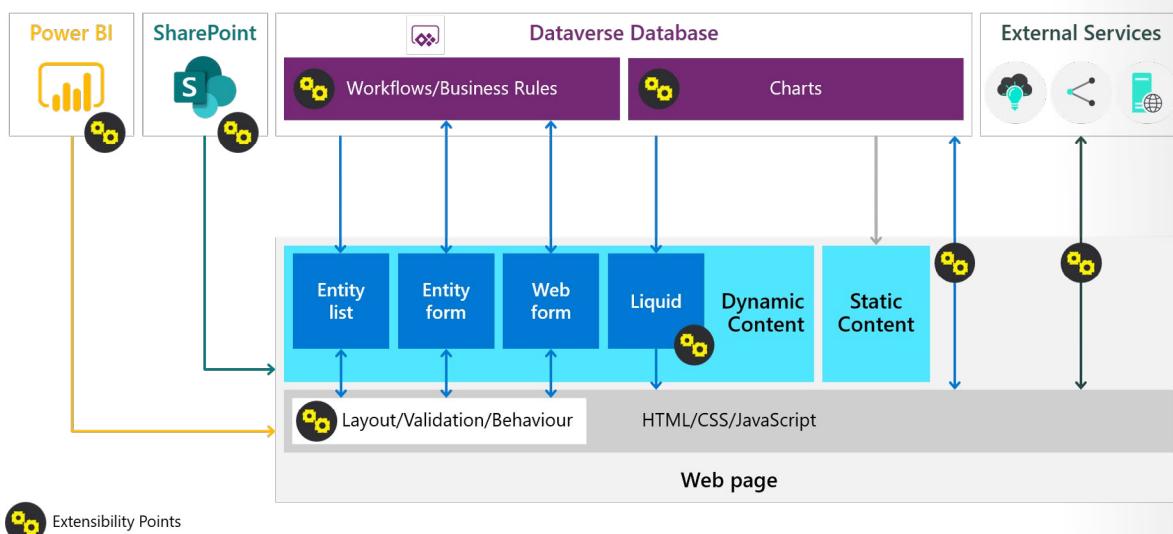


[!NOTE]

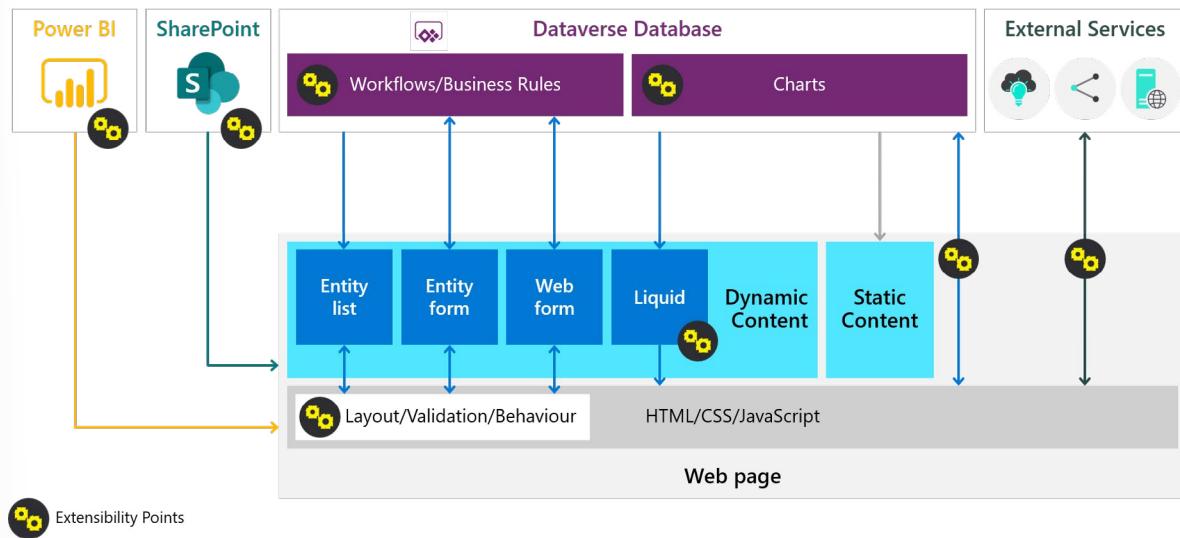
A portal requires a Dataverse database to be available in the environment to install and configure some key components. While a portal from blank can be configured in an environment without any of the Microsoft Dynamics 365 apps installed, the templates (Customer self-service, Employee self-service, Partner, and Community portals) have dependencies on Microsoft Dynamics 365 apps.

## Portals architecture

Power Apps portals transact directly with Dataverse data. You can create lists and forms over data. Power Apps portals provide built-in components that use model-driven views and forms. The portals can be customized and extended, as shown in the following diagram.



Power Apps portals extend Dataverse solutions to internal and external audiences in a more secure way. Portal visitors can access portals as either anonymous or authenticated users.



## Expose Dataverse data

The following sections describe the several different ways that you can expose Dataverse data.

### Use cases for portals

When to consider using a portal app:

- Interactions with Dataverse for external and internal users is more secure
- Community or self-service sites for customer service
- CRUD on Dataverse data
- Limited resources and budget, business-user, and no-code configuration requirements
- Responsive design, and content is accessible across all devices and browsers
- Multi-lingual implementations
- Single sign-on

When to exercise caution:

- Most data resides in an external (non-Dataverse) system
- Heavy requirements around document management, indexing, and searching.
- Large volumes of users who drive heavy traffic to the portal.
- eCommerce requirements, including processing payments and maintaining an online store.
- Use cases that are more appropriate for direct Power Apps licensed user access.

### Authentication

Power Apps portals allows authenticated and unauthenticated users. The solution architect needs to know if authenticated access will be used and how users will be authenticated, whether it's through Microsoft Azure B2C, Azure Active Directory (Azure AD), or another identity provider.

[!IMPORTANT]

You should avoid using locally stored accounts for authentication.

## Implementation considerations

When implementing a portal app, the solution architect needs to consider the following key factors:

- Whether a blank template or a Dynamics 365 template will be used
- The gap from template to requirements
- Which portal pages will require advanced Liquid template-skilled resources
- What data that authenticated users will need access to

## Deployment considerations

Portal assets, such as views and forms, can be packaged in solutions, but most portal configuration is stored as data over many tables. The Configuration Migration tool can help mitigate some of the work when you are moving from development to test to production.

## Further reading

Power Apps portals architecture <https://docs.microsoft.com/learn/modules/portals-architecture/>

Work with Power Apps portals <https://docs.microsoft.com/learn/paths/work-power-apps-portals/>

Extend Power Apps portals <https://docs.microsoft.com/learn/paths/extend-power-apps-portals/>

# Power Apps architecture exercise

## Exercise Overview

In this exercise you will work in small groups. Review the information presented about Fabrikam Robotics and complete the tasks.

## Learning Objectives

After completing the exercise, you will:

- Identify and group app users
- Discuss and propose app types to solve a requirement
- Identify and recommend use of components

## Exercise 1

You are building a solution for Fabrikam to track visitors to a showroom and manufacturing site. Some of the visitors are potential purchasers and some are just there to see the magic of the robots working.

Review the requirements here and address the concerns listed later in this document.

- Fabrikam has been rapidly moving to the cloud and already leverages Office 365 for their email. They are adopting the use of Microsoft Teams at a rapid pace.
- The sales staff handles their own leads and opportunities. They are compensated by commission on close of sales and are very competitive. The staff that works on small deals all work together and receive commission on all closed deals in their department. The large deals sales staff mostly work by themselves; except on very large deals where there is a team of them assigned.
- The sales staff that works on large deals regularly visits prospects offices and are having meetings at conferences with prospects. While they each have mobile hotspots for WiFi, in practice many buildings have little or no signal.
- To handle the sales staff viewing tour data a custom connector will be created to connect to the device service cloud API to make the data available to the app the sales staff is using.
- Reception staff must be able to check in visitors prior to their showroom and manufacturing floor tour. During the check in, they must be able to capture a picture of each visitor. Reception staff must not have access to the sales data.
- Discounts can only be approved and applied by one of the sales managers. The app should ensure the sales staff can't approve their own discounts. In the data model, there is a discount field of type currency and a required lookup field to the approver.

- Customers should be able to request a showroom visit from Fabrikam's website. The proposed solution should remind them via SMS message if they provide a phone number 24 hours before their visit.

## Discuss as a group how you would address each of the following:

1. Identify the following:
  1. Different app user groupings?
  2. What Power App(s) would you propose creating for these groups?
  3. What type would they be?
    1. model-driven
    2. canvas
    3. portal
  4. non-Power App Custom Dev
2. How are you accommodating in your app architecture for the sales staff that visit prospects' offices?
3. How will you propose to use the custom connector so the sales staff can view the tour data from their app?
4. Identify places a Power Apps Component Framework component might be helpful or required by your application architecture?
5. If any of your apps are canvas apps, what is your plan to support multiple app makers being able to work on building the app?
6. What would you offload and rely on Power Automate to handle?

# Module Summary

## Recap

App composition is one of the most important tasks for a solution architect. Take the time upfront to properly design your app composition and identify reusable components that should be built.

A user's app experience will be how they judge the solution architecture.

[!IMPORTANT]

The level of change in the Microsoft Power Platform means that best practices will change every few months.

## Next steps

See the following links for more information on Power Apps architecture.

- **Master advanced techniques for Power Apps canvas apps learning path<sup>8</sup>**
- **Create components with Power Apps component framework learning path<sup>9</sup>**
- **Power Apps guidance documentation<sup>10</sup>**

---

<sup>8</sup> <https://docs.microsoft.com/learn/patterns/understanding-advanced-topics/>

<sup>9</sup> <https://docs.microsoft.com/learn/patterns/use-power-apps-component-framework/>

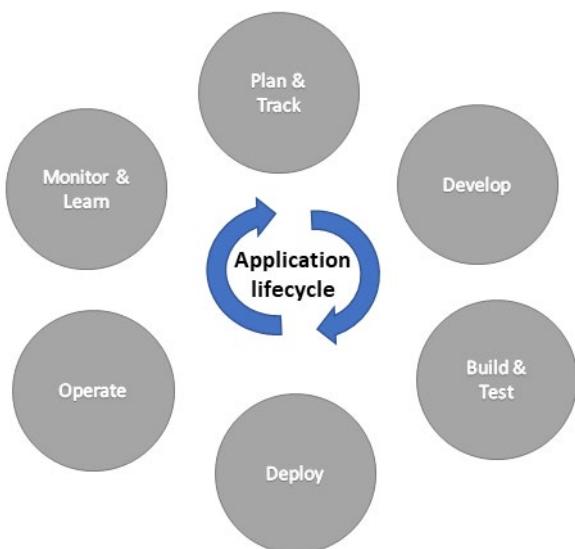
<sup>10</sup> <https://docs.microsoft.com/powerapps/guidance/>

## Module 8 Application lifecycle management

### Plan application lifecycle management

#### Introduction

The application lifecycle is the cyclical software development process that involves these areas: plan and track, develop, build and test, deploy, operate, monitor, and learn from discovery.



This module is about defining your environment structure, how you compose your solutions, your application lifecycle management (ALM) strategy, and then how to use Azure DevOps Build and Release pipelines.

## Application Lifecycle Management with the Microsoft Power Platform

Microsoft uses solutions to package apps and customizations and export from one Dataverse environment as a file and then import that solution package file into another Dataverse environment. The solution packaging supports different scenarios from simple changes to fully automated release management. Microsoft uses the same method for packaging its own first-party apps and updates.

Microsoft also provides tools to enable solutions to be exported from a development Dataverse environment to be imported to test and production environment using Azure DevOps.

The following concepts are important for understanding ALM using the Microsoft Power Platform.

- Solutions are the mechanism for implementing ALM; you use them to distribute components across environments through export and import. A component represents something that you can potentially customize. Anything that can be included in a solution is a component, such as site maps, apps, entities, fields, charts, or plug-ins.
- Dataverse stores all the artifacts, including solutions.
- Source control should be your source of truth for storing and collaborating on your components.

## Microsoft's vision

Microsoft Power Platform vision for ALM is to enable apps and customizations to be easily deployed through automated processes.

- Quick Start: Enable app builders to get set up with an environment with the latest build and connected to source control and make a change quickly
- Build: Simplify tooling, consolidate portals and speed up inner loop
- Deploy: Enable an automated repeatable (predictable) deployment methodology
- Manage: Invest in other environment management capabilities to offer more flexibility for app builders to use and dispose pre-configured environments as needed
- Monitor: Application telemetry and feedback loop by design

Solution architects should understand the vision and journey Microsoft is on to shape how ALM is done for Microsoft Power Platform projects. As this journey evolves, solution architects should continue to shape their own plans to apply the capabilities provided by the platform and tools.

## Solution Architect's role

The Microsoft Power Platform solution architect needs to define the environment strategy and application lifecycle management for transporting work from development to test to production.

The solution architect should:

- Lead the establishment of an application lifecycle management (ALM) plan.
- Evaluate and determine the amount and sophistication of the ALM that is appropriate for the project.
- Work with the various teams to support their efforts to implement execute the plan.

## Environment strategy

The solution architect needs to define the strategy for application lifecycle management on the project. This is part of the process of helping the organization put in place proper governance for the solution.

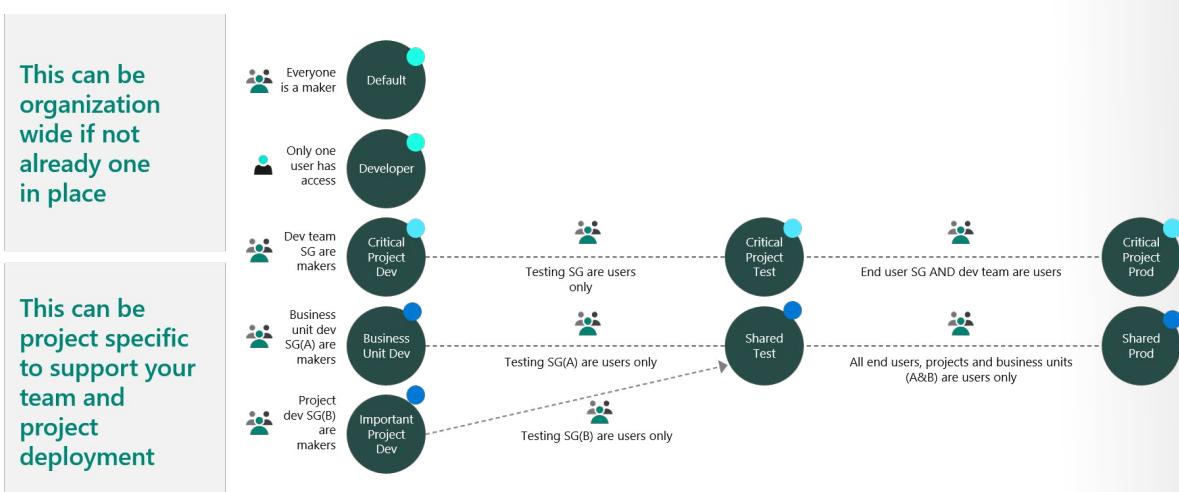
An environment is a container that stores, manages, and shares your business data, apps, flows, connections, and other assets, along with permissions to allow organization members to use the resources.

An environment serves as a container to separate apps that might have different roles, security requirements, or target audiences. How you choose to use environments depends on your organization and the apps you're trying to build. For example:

- You can choose to only build your apps in a single environment.
- You might create separate environments that group the test and production versions of your apps.
- You might create separate environments that correspond to specific teams or departments in your company, each containing the relevant data, and apps for each audience.
- You might also create separate environments for different global branches of your company.

Developing an environment strategy means configuring environments and other layers of data security in a way that supports productive development in your organization, while securing and organizing resources. A strategy to manage environment provisioning and access, and controlling resources within them, is important to:

- Secure data and access.
- Understand how to use the default environment correctly.
- Manage the correct number of environments to avoid sprawl and conserve capacity.
- Facilitate application lifecycle management (ALM).
- Organize resources in logical partitions.
- Support operations (and helpdesk) in identifying apps that are in production by having them in dedicated environments.
- Ensure data is being stored and transmitted in acceptable geographic regions (for performance and compliance reasons).
- Ensure isolation of applications being developed.



The following types of environment are available in the Microsoft Power Platform.

- Sandbox: A sandbox environment is any non-production environment of Dataverse. Isolated from production, a sandbox environment is the place to safely develop and test application changes with low risk.
- Production: The environment where apps and other software are put into operation for their intended use.
- Community (Developer): The Power Apps Community Plan gives a user access to Power Apps premium functionality, Dataverse, and Power Automate for individual use only. This environment is primarily meant for learning purposes. A developer environment is a single-user environment, and cannot be used to run or share apps. A Community Plan environment can participate in the Azure DevOps pipeline.
- Default: A single default environment is automatically created for each tenant and shared by all users in that tenant. The Default environment is used by Microsoft 365 services.
- Trial: Trial environments are for trying new features or to perform proof of concepts. Trial environments are automatically deleted after 30 days.

[!IMPORTANT]

The solution architect needs to define how many environments are required, what is their purpose, and what are the dependencies between environments. At a minimum, a healthy ALM practice should include using a test environment prior to deploying anything to the production environment. This ensures that you have a place to test your app, but also ensures that the deployment itself can be tested.

For more information, see [environments<sup>1</sup>](#) and [environment strategy<sup>2</sup>](#).

## Handling solutions and other non-solution aware code and components

Microsoft Power Platform projects consist of components that can be packaged inside solutions in environments and components that cannot be added to solutions such as components deployed in Azure, configuration data, and Power BI reports. The ALM plan must consider how to handle these non-solution aware components.

The solution architect needs to decide if application lifecycle management will be managed by using solutions or by using source code control. Traditionally, Microsoft Power Platform projects have been more environment centric, but many are now moving towards source control centric.

If you use an environment centric approach, then:

- The Dev environment is the master copy of all changes.
- Changes are promoted directly from Dev -> Test -> Prod.

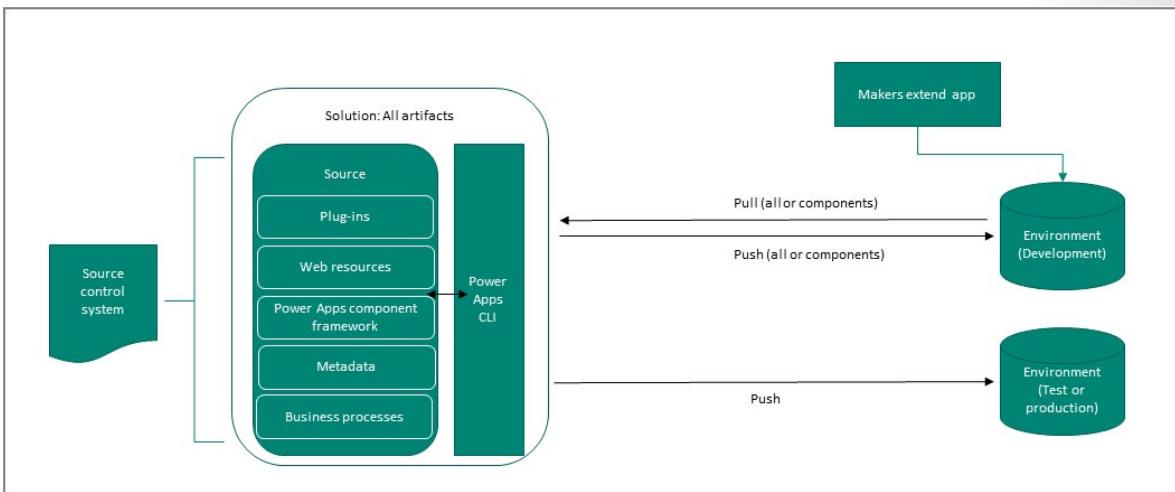
If you use a source control centric approach, then:

- Source control is the master.
- The Dev environment is re-created from source control (this can be automated and repeatable).
- Changes from Dev are checked into source control.

---

<sup>1</sup> <https://docs.microsoft.com/power-platform/admin/environments-overview>

<sup>2</sup> <https://docs.microsoft.com/power-platform/guidance/adoption/environment-strategy>



A Source control centric approach encourages having a definitive master and the ability to re-create development environments for any version tracked. Microsoft is encouraging and building tooling to support source control-centric ALM.

[!NOTE]

Ideally all environments other than production should be “throw away” i.e., the dev and test environments can be deleted and recreated without any loss.

Using a source control-centric approach will enable a DevOps approach with Build and Release Pipelines. Using an environment-centric approach means that you need to define the workflow for app makers and developers. The solution architect will need to define how, and who, will promote the app through the environments from Dev to Prod.

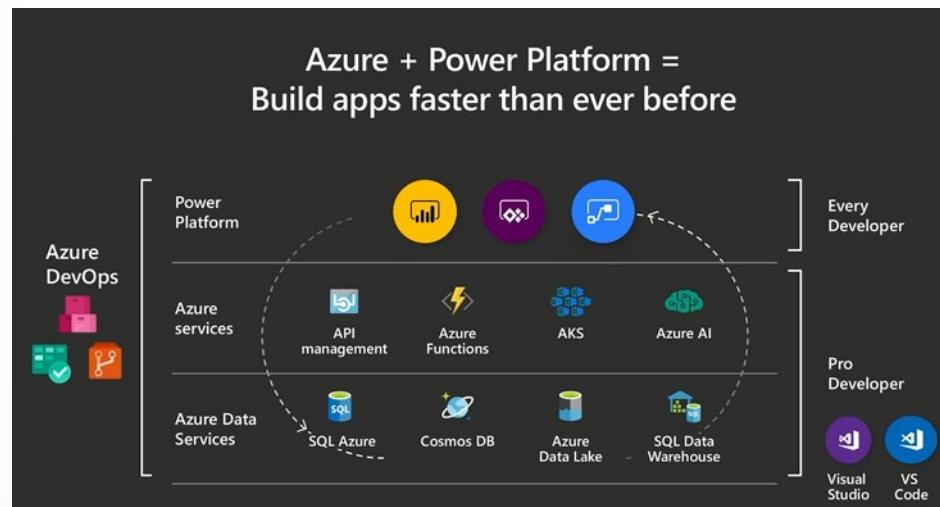
The solution architect will also need to define how to configure each environment and look for ways to make this easier.

## Team working

Compared to traditional app development, Power Apps projects are different in two key areas:

- How various members of the project team work together to create the solution.
- Development methodology.

Power Apps is a platform that benefits both pro developers and citizen developers. In a traditional development environment, only pro developers could be involved with the actual making of an app. With Power Apps, everyone has the power to build the apps they need by using advanced functionality that was previously available only to professional developers. Power Apps democratizes the custom business app building experience by enabling users to build feature-rich, custom business apps without writing code.



With Power Apps, you can quickly create a usable version of your app, because Power Apps provides a WYSIWYG (what you see is what you get) development experience. Users experience the actual working app very early in the development process, and if new requirements arise, new features can be added to the next version.



There are a number of issues with customizing and developing components within the Microsoft Power Platform:

- The Microsoft Power Platform does not support versioning of components (except for Canvas apps).
- Users cannot work on the same Microsoft Power Platform component at the same time.
- With Model-driven apps, there are multiple components each with their own editors allowing work to be divided between makers. With Canvas apps, there is only one editor and only one person can work on an app at any one time. By using Canvas components, you can allow multiple makers to work on the same app.

The solution architect should establish the workflow for how app builders will make changes and promote them. Proactive communication and work assignment should be managed to minimize conflicts between makers.

You can minimize clashes between makers by creating an individual environment for each maker. Individual maker environments offer isolation and tracking but require extra effort to merge work and resolve conflicts. A shared maker environment can be less complex but does not offer isolation between app builders and lack detail tracking of changes.

## Source control

Even if you are using an environment-centric approach you will still need to decide where the master copy of the solutions and code live.

Solution aware developer code assets (such as plug-ins, PCF code components, and Form Scripts (transpiled from TypeScript) should be “built” on a build environment and not the developer’s desktop. After being built the assets should be deployed to an environment that the master solution will be exported from, or built into a solution that will be installed.

## Tools

There are several tools and apps that Microsoft provides that can be used with Microsoft Power Platform ALM:

- Microsoft Power Platform admin center: The Microsoft Power Platform admin center provides a unified portal for administrators to create and manage environments.
- Power Apps build tools: The Power Apps build tools automate common build and deployment tasks related to Power Apps using Azure DevOps.
- GitHub: Popular example of a version control system.
- Configuration Migration Tool: The Configuration Migration Tool enables you to move configuration and/or reference data across environments.
- Package Deployer: The Package Deployer lets you deploy packages of assets to Dataverse instances. Packages can consist of not only solution files, but also flat files, custom code, HTML files, and data.
- Solution Packager: Solution Packager is a tool that can unpack a compressed solution file into multiple XML files and other files, so they can be easily managed by a source control system.
- Power Apps CLI: Microsoft Power Apps CLI is a simple command-line interface that empowers developers to create code components.
- Package deployment PowerShell module: The Package deployment PowerShell module is used to deploy packages to Dataverse environment.
- Power Apps checker PowerShell module: The Power Apps checker PowerShell module interacts with the Power Apps checker service so you can run static analysis jobs and download the results.

[!NOTE]

GitHub actions for the Microsoft Power Platform are currently in preview.

## Solutions

Solutions are containers that track and manage customizations in a Dataverse environment. Solutions are used to transport apps and components from one environment to another, or to apply a set of customizations to existing apps.

When you have multiple environments, each environment has their own set of solutions.

Solutions are a container to track the changes you make to Dataverse, Power Apps and Power Automate flows

Solutions are how you transport and install changes to target environments

Development

Test

Production

### Microsoft Dynamics 365 apps are installed using solutions

Third-party apps provided by Independent Software Vendors (ISVs) also use solutions

[!NOTE]

When you have a Dynamics 365 first-party app like Sales, the app is installed using the same solution framework. Third-party ISVs also ship their products using solutions.

Solutions have the following features:

- Solutions include metadata and certain entities with configuration data. Solutions do not contain any business data.
- Solutions can contain many different Microsoft Power Platform components, such as model-driven apps, canvas apps, site maps, flows, tables, table metadata, columns, forms, views, business rules, process definitions, custom connectors, web resources, choices, charts, and components created by developers like scripts or compiled code.
- Solutions are packaged as a unit to be exported and imported to other environments, or deconstructed and checked into source control as source code for assets.
- Solutions are used to apply changes to existing solutions.

## Types of Solution

There are two types of solution:

- Unmanaged: Used during development and to transport to other development environments.
- Managed: Used to distribute to non-development environments.

Unmanaged Solutions are to be used in development environments while you are making configuration changes to your application. Solutions are exported as unmanaged and checked into your source control system. Unmanaged solutions should be considered your source.

Managed solutions are used to deploy to any environment that is not a development environment. This includes test, user acceptance testing (UAT), system integration testing (SIT), and production environments.

Managed solutions can be serviced (upgrade, patch, and delete) independently from other managed solutions. As an ALM best practice, managed solutions should be generated by a build server and considered a build artifact.

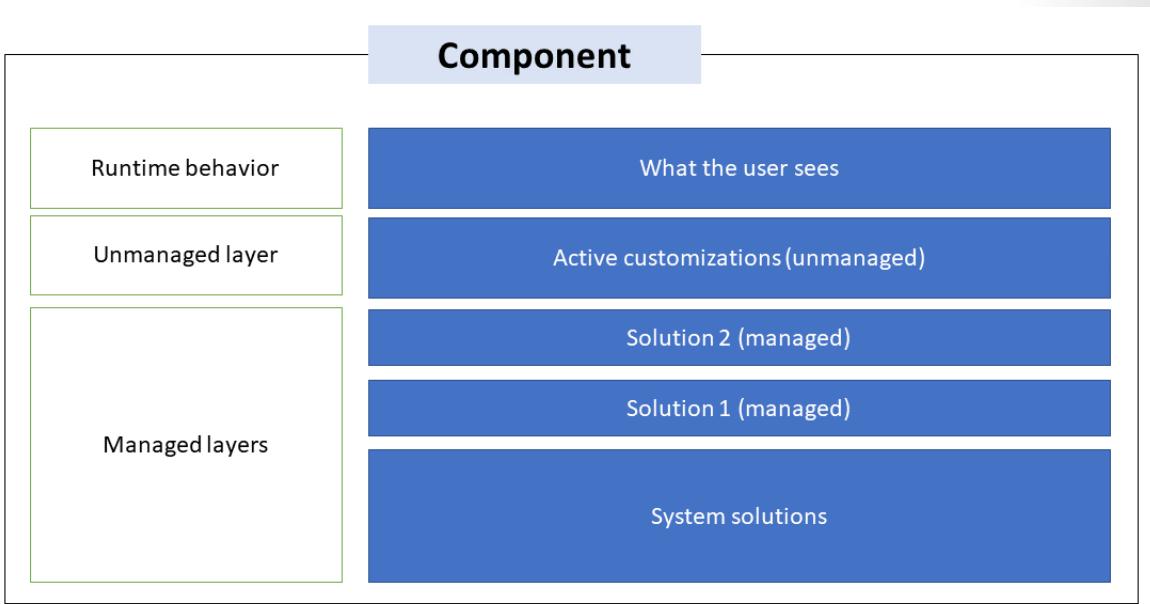
## Solution layering

Solution layers describe the dependency chain of component from the root solution introducing it, through each solution that extends or changes the component's behavior. Layers are created through

extension of an existing component (taking a dependency on it) or through creation of a new component or version of a solution.

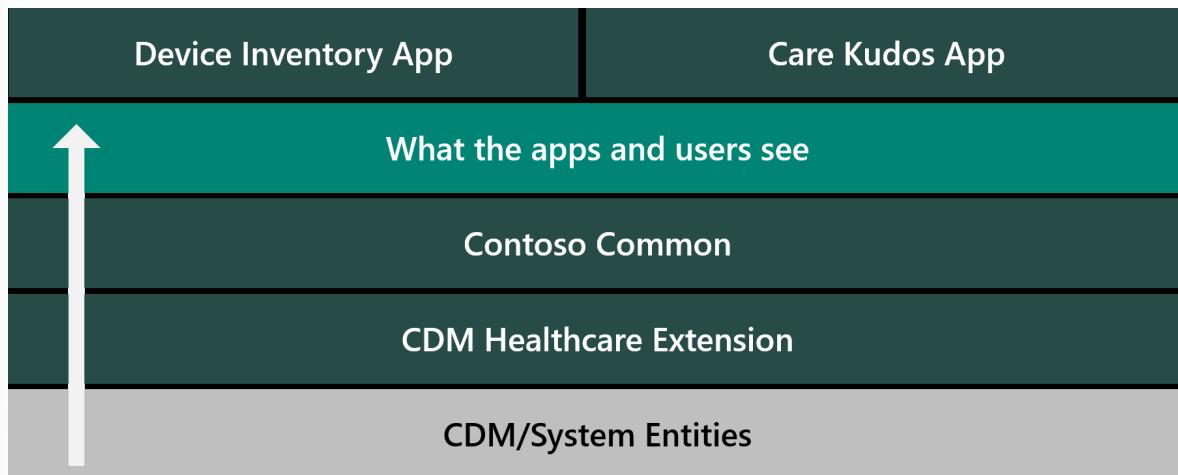
Solution layering is implemented at a component level. Managed and unmanaged solutions exist at different layers within a Microsoft Dataverse environment. In Dataverse, there are two distinct layers:

- Unmanaged layer: All imported unmanaged solutions and ad-hoc customizations exist at this layer. All unmanaged solutions share a single unmanaged layer.
- Managed layers: All imported, managed solutions and the system solution exist at this level. When multiple managed solutions are installed, the last one installed is above the managed solution installed previously. This means that the second solution installed can customize the one installed before it. When two managed solutions have conflicting definitions, the runtime behavior is either "Last one wins" or a merge logic is implemented. If you uninstall a managed solution, the managed solution below it takes effect. If you uninstall all managed solutions, the default behavior defined within the system solution is applied. At the base of the managed layers level is the system layer. The system layer contains the entities and components that are required for the platform to function.



The solution architect needs to decide on the number of solutions that will be used for their business solution. While you could work with a single solution, this leads to dependencies on releases, and large solutions can take a long time to export and import. Most projects use multiple solutions. Solution architects should understand the merge behavior when a solution is updated or when multiple solutions are installed that affect the same component.

In the following example, four solutions have been used: The CDM Healthcare accelerator, a Common Contoso solution, and two solutions containing apps. These solutions are layered. For example, the contact form is customized in the CDM Healthcare Extension. The same form elements are modified in the Contoso Common solution. The end users will see the changes from the Contoso Common solution.



## Solution structure

The following are some strategies for creating solutions listed in order from simplest to most complex:

- Single solution
- Multiple solutions
- Multiple solutions with shared components

By creating a single solution, you establish a working set of customizations. This makes it easier to find items that you have customized. This approach is recommended when you only want to create a single managed solution. If you think you may have to split up the solution in the future, consider using multiple solutions.

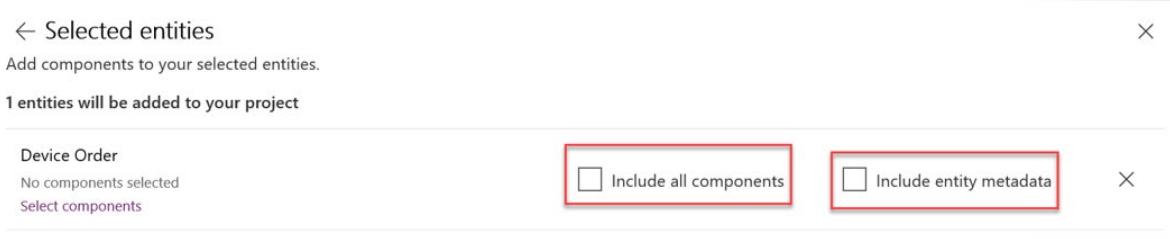
If you have two unrelated solutions that don't share components, the most direct approach is to create two unmanaged solutions.

You may have multiple solutions that share components. You may have a certain set of common functionality within multiple solutions and that common functionality is compatible with any of the other functionalities unique to each solution. Some components can be included in more than one solution as long as any changes that were made to them are compatible with all other solutions that use them. It is important that all the solutions share the same solution publisher. If the solution publisher is not identical, you will not be able to install more than one of your solutions.

There are some rules you should follow with solutions:

- Create a Solution Publisher and use for all solutions.
- Don't use the default publisher, the Default solution, or the Common Data Service Default Solution.
- Keep the solution structure as simple as possible.
- Avoid using include all components unless adding in an unmanaged table.
- Include table metadata only when you are changing table properties.
- Add the subcomponents of a table (columns, forms, views etc.) only when you are changing them.

Adding only what is required to a solution is known as segmenting a solution.



When deciding how to segment into one or more solutions consider the following:

- Use multiple solutions only for a tangible purpose, multiple solutions add complexity.
- Multiple solutions that share components should be avoided.
- Multiple solutions each require their own environments to ensure they remain independent.
- Care must be taken to manage dependencies.
- Makers must know what solution to put new components in.
- Common patterns for multiple solution splitting are horizontal and vertical partitioning.

## Horizontal solution splitting

Horizontal splitting refers to creating solutions that only contain components of the same type.

Visual Components	Processes & Plugins	Reports	Security Roles	Main
Apps	Processes	Reports	Security Roles	Tables
Canvas Components	Plug-in Assemblies			Choices
PCF Components				Client Extensions
Web Resources	SDK Message Processing Steps			Service Endpoints
				Dashboards
				Connection Roles
				Article Templates
				Email Templates
				Mail Merge Templates
				Field Security Profiles

## Vertical solution layering

Vertical layering groups components into functional areas. Often you will have a shared base/common solution with separate solutions for each key business area.



You can combine vertical and horizontal partitioning, for instance the base containing all tables and processes with separate solutions for each app.

## Configuration

As you transport solutions through environments those components may have configuration settings or refer to data that also need to be transported through the release process.

## Environment variables

Solution Environment variables are used for app configuration data that is environment-specific. Applications often require different configuration settings or input parameters when deployed to different environments. Environment variables store the parameter keys and values, which then serve as input to various other application objects. Separating the parameters from the consuming objects allows you to change the values within the same environment or when you migrate solutions to other environments.

The alternative is leaving hard-coded parameter values within the components that use them. This is often problematic; especially when the values need to be changed during application lifecycle management (ALM) operations.

Because environment variables are solution components, you can transport the references (keys) and change the values when solutions are migrated to other environments.

Environment variables can be of type Decimal number, Text, JSON, Two options, or Data source. Each environment variable can have a default value and a current environment value.

The following diagram illustrates how environment variables can be used to tailor values for each environment.

 Development	Var – From Email Value – dev@contoso.com	Var – Auto Approve Default – Yes
 Test	Var – From Email Value – test@contoso.com	Var – Auto Approve Default – Yes
 Production	Var – From Email Value – prod@contoso.com	Var – Auto Approve Value – No

## [!NOTE]

You can provide a default value that is used if a current value is not specified in a particular environment.

Power Apps, Power Automate and developer code can retrieve and modify the values of environment variables.

## Connection references

Connection references are used for connections that are environment-specific. A connection reference contains information about a connector. Both Canvas app and operations within a Power Automate flow bind to a connection reference.

Because connection references are solution components, you can transport the references and change the connection when solutions are migrated to other environments. Connection references enable you to change a connection associated with a canvas app or flow without editing the app or flow.

## Configuration Migration Tool

Solutions do not contain data. Often, your application will rely on reference, or configuration data. This data also needs to be transports from one environment to another.

The Configuration Migration Tool can help move data between environments. Importantly, the Configuration Migration Tool can maintain the same primary record identifier (GUID) for the rows in this data. This prevents issues with components that refer to specific data from having to be updated each time it is imported.

## Release

Solutions applied to test and production will be managed solutions. There are some options when importing managed solutions that the solution architect needs to be aware of.

## Importing managed solutions

It is not a good idea to create new solutions for each change or release of your application. This will lead to dependencies between components that will prevent you from importing the solution. You should therefore make changes to your solution(s) and import the new version.

## [!NOTE]

Solutions have version numbers. The version numbers are automatically incremented when you export a solution. You cannot import a solution with lower version number if the solution has already been imported with a higher version number.

When importing a new version of a managed solution, there are three import options:

- Update: Applies the changes in the solution.
- Upgrade: Imports changes and applies them immediately including removing components not in new solution, the old solution, and any patches are removed.
- Stage for Upgrade: Similar to upgrade but it pauses after new solution is imported before removing components so you can do data migration etc. You then manually trigger the final apply of the solution.

Updates to a managed solution are deployed to the previous version of the managed solution. This doesn't create an additional solution layer. You cannot delete components by using an update.

Upgrading a solution installs a new solution layer immediately above the base layer. Solution upgrades will delete components that existed but are no longer included in the upgraded version.

Upgrade is the default option.

## [!IMPORTANT]

Using clone a patch and clone solution to update a solution is no longer recommended because it limits team development and increases complexity when storing your solution in a source control system.

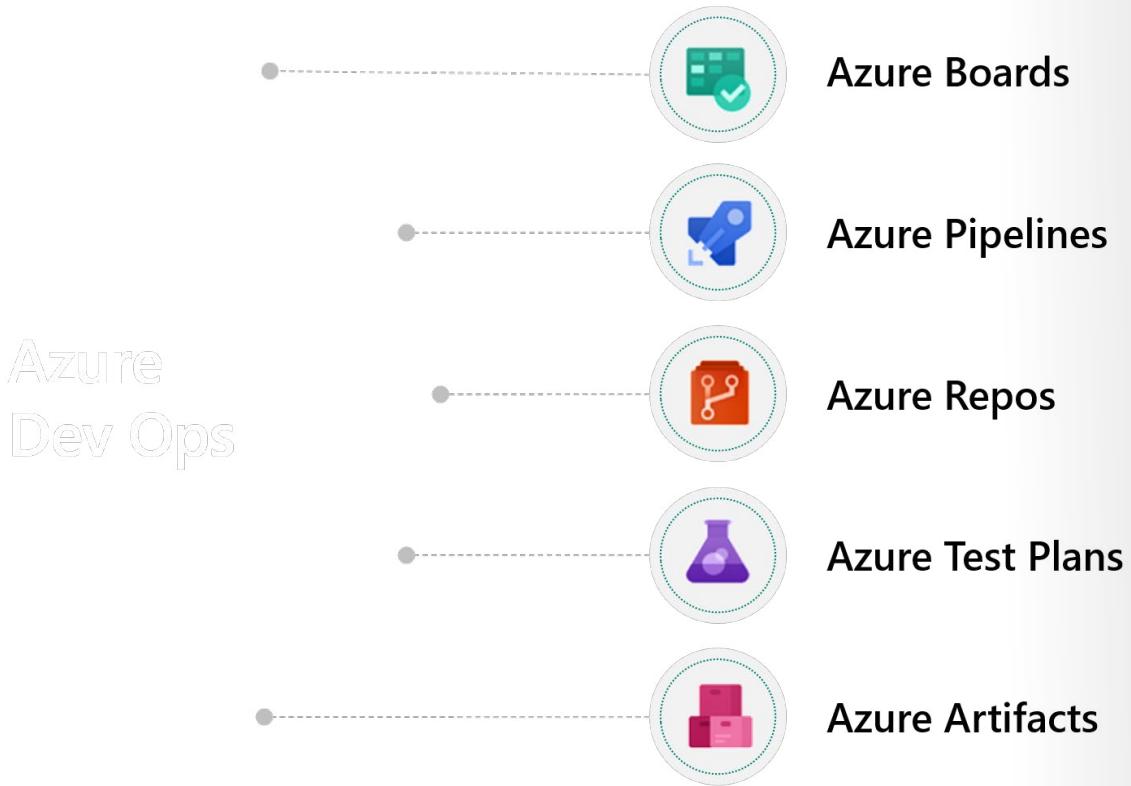
## Azure DevOps

The solution architect leads the effort to define the process for how changes will be promoted from dev to production. This includes defining the number of stages for example, Dev -> Test -> Prod and the processes to do the promotion regardless if manual or automated.

Microsoft is building tools to support doing this with Azure DevOps using continuous integration (CI) and continuous deployment (CD).

This section provides an overview of Azure DevOps and how DevOps can be used with the Microsoft Power Platform to automate deployments.

Azure DevOps provides developer services for support teams to plan work, collaborate on code development, and build and deploy applications.



Azure DevOps contains many features to help the development of applications:

- Azure Boards: Plan, track, and discuss work across your teams
- Azure Pipelines: Use to automate continuous integration and continuous deployment (CI/CD) builds and releases.
- Azure Repos: Source Control to store and track changes.
- Azure Test Plans: Plan, execute, and track scripted tests.
- Azure Artifacts: Publish solutions built by build pipelines.

## Pipelines

The Power Apps builds tools automate common build and deployment tasks related to Power Apps using Azure Pipelines.

Build pipelines can be used to:

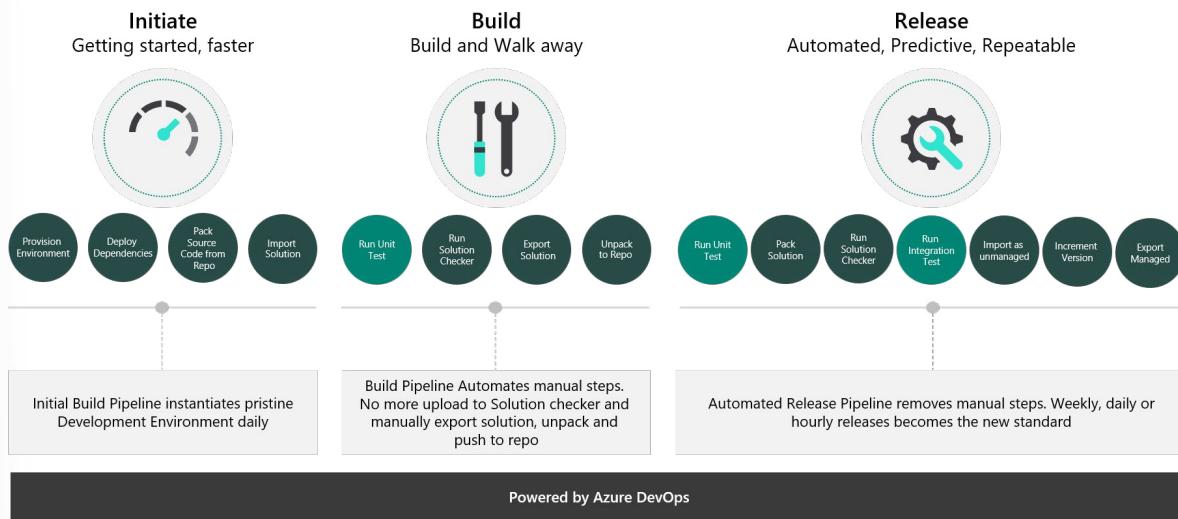
- Create dev environments.
- Commit changes from dev to source control.
- Solution Checker.
- Automated Testing.
- To build output solutions from source control (for example, managed/unmanaged).

Release pipelines can be used to:

- Take solutions from build pipelines and deploy them to one or more test/prod environments.

- Perform automated testing as part of release process.
- Pause for approvals before progressing to next environment.

The Microsoft Power Platform Build Tools tasks can be used along with any other available Azure DevOps tasks to compose your build and release pipelines. Pipelines that teams commonly put in place include Initiate, Export from Dev, Build, and Release.



## Alternative automation tools

There are alternatives for automating deployments without using Azure DevOps:

- Dataverse and Admin APIs can be used to automate from any supported language.
- PowerShell can be used instead of build tasks for more control.
- Power Automate can also be used with the platform admin connectors to automate deployments.
- GitHub actions are currently in preview.

# Module Summary

## Summary

In this module, you looked at application lifecycle management with the Microsoft Power Platform, including:

- Learning about Microsoft vision and Solution Architect's role in ALM
- Creating an environment strategy
- Defining a solution structure for your deliverables
- Automating deployments with Azure DevOps

As a solution architect you should understand how to:

- Establish ALM process and procedures. Establishing application lifecycle management process should be done at the start of the project/
- Create repeatable process. Having a repeatable process saves time and ensures consistent results.
- Designing ALM to fit your project. ALM is not a one size fits all, you must size it to fit your project.

## Deployment Checklist

Before progressing with building and deploying your solution, you need to ensure that the following items have been addressed:

- Proper environment access has been established, e.g., Security Groups.
- Data loss prevention policies are correct for solution environments.
- Service Principals and service accounts necessary for connections are in place and configured.
- Environment settings have been documented and configured in each environment.

## Next Steps

Learn more about ALM using the **Evolve your DevOps practices learning path<sup>3</sup>**.

For more information, see **Application lifecycle management (ALM) with Microsoft Power Platform<sup>4</sup>**.

<sup>3</sup> <https://docs.microsoft.com/learn/patterns/evolve-your-devops-practices/>

<sup>4</sup> <https://docs.microsoft.com/power-platform/alm/>



## Module 9 Power Automate architecture

### Explore Power Automate architecture

#### Introduction to Power Automate

The Solution Architect is responsible for the overall design of the solution. Understanding the capabilities of Power Automate is imperative when designing the automation for the solution. This module is concerned with Power Automate and its capabilities.

#### Options for automation and custom logic

Dataverse provides many options for automation and custom logic

- Business rules
- Dataverse Classic workflows
- Dataverse Plug-ins
- Power Automate cloud flows
- Power Automate desktop flows

How does the solution architect decide when they should use each of these options. We will compare these options.

#### Business rules

Business rules have two purposes. The first way that business rules are used is within model-driven app forms, to change how columns are displayed. The second way that business rules are employed is at the data layer to validate data and calculate values.

Business rules:

- Are good for simple validation or setting of values.
- Are optimized to run as part of the transaction for modifications that occur on the rows.
- Can be configured to run in model-driven apps for basic UX operations like hide/show of columns.

- Cannot access related records.
- Cannot use connectors.

Business rules can be configured to run on different scopes: single form, all forms, or on table create and update operations.

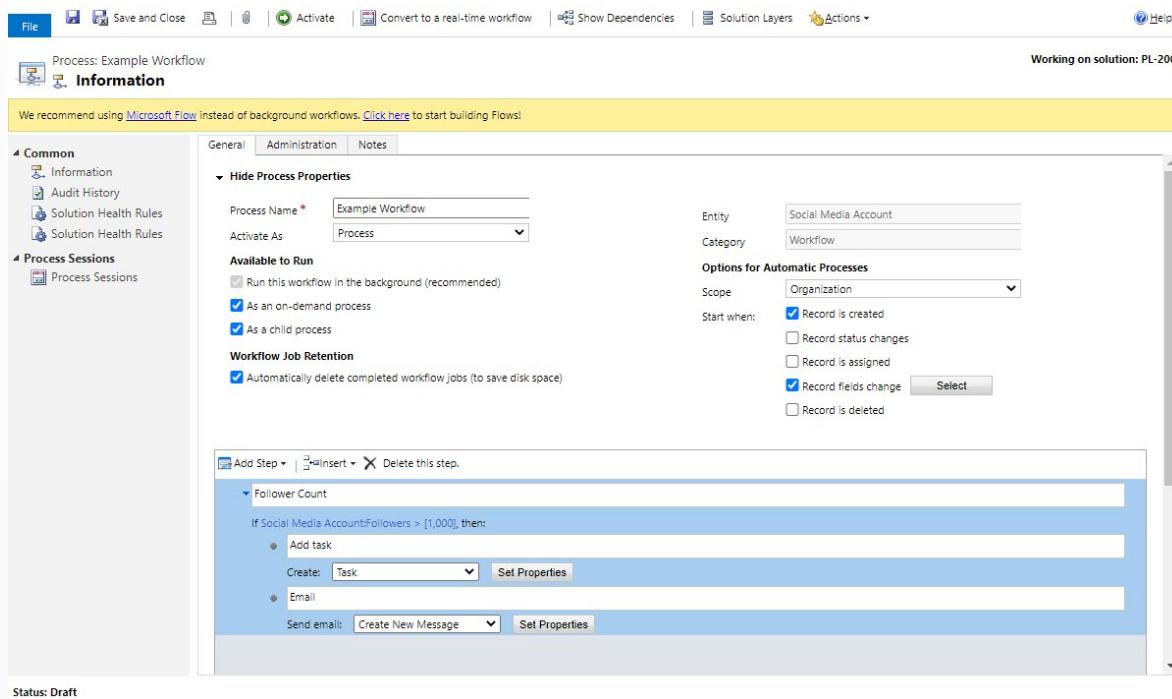
## Classic workflows

Classic workflows can be used to remove the need for users to manually perform complex tasks in sequence and control the activities required to deliver business processes consistently and repetitively.

The primary scenario for Classic workflows is when real-time processing is required.

Classic workflows:

- Cannot only access related records in many-to-one relationships.
- Are limited to operations on Dataverse data



[!IMPORTANT]

Power Automate should be the first choice for background operations

## Plug-ins

A plug-in is a .NET assembly that you can upload to the Microsoft Dataverse. Classes within the assembly can be registered to specific events (steps) within the event framework. The code within the class provides a way for you to respond to the event so that you can augment or modify the default behavior of the platform.

Plug-ins:

- Are custom logic is an extension of the Dataverse operation.
- Have the ability to modify the request and response on the fly.

- Are able to handle complex logic.
- Require developer skills.
- Can be either synchronous or asynchronous.

## Power Automate cloud flows

Power Automate cloud flows are workflows that automate repetitive tasks and streamline processes both within and across systems.

You can use Power Automate for the following:

- Personal productivity
- Sending notifications
- Handling approvals
- Gathering data
- Automating processes
- Integrating systems
- Orchestration across systems

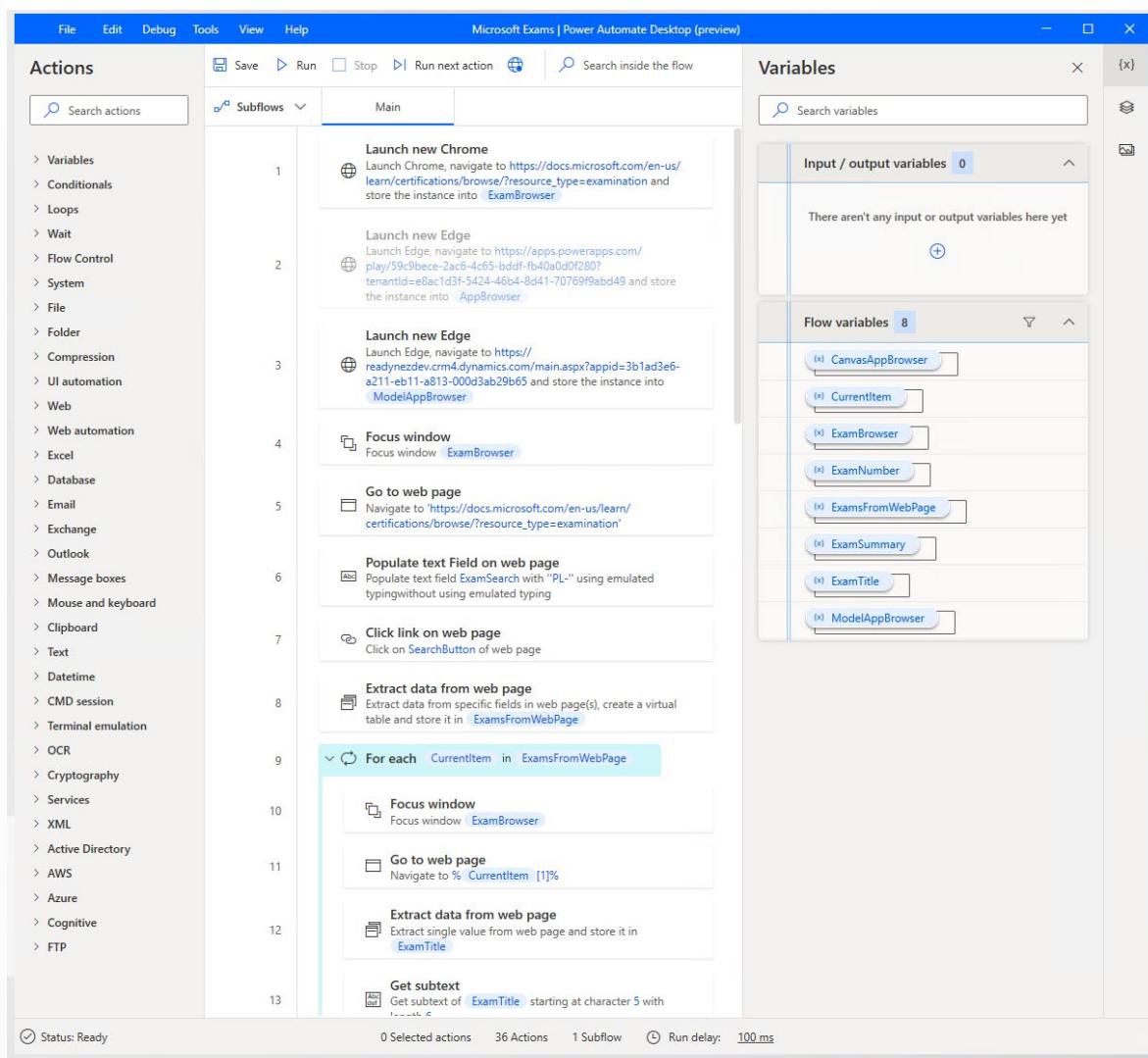
Power Automate cloud flows are the primary choice for non-real time automation. Power Automate cloud flows can be triggered in near real-time triggering by Dataverse events.

Power Automate cloud flows can use over 400 connectors to other cloud services and data sources. Power Automate can connect to any cloud service that has a REST API by creating a custom connector.

## Power Automate desktop flows

Power Automate desktop flows are for automation where there is no connector or API available to use. There are many legacy applications that do not have a method for accessing their data or functionality except through their user interface. Power Automate desktop flows use Robotic Process Automation (RPA) techniques to automate user actions on these legacy applications.

Power Automate desktop flows can automate desktop and web applications. Power Automate desktop flows can be run attended with the user manually initiating the flow, or unattended with desktop flows running on Virtual Machines in Azure.



Power Automate desktop flows are a valid way to perform integrations and automation when there is no other alternative, or when developing an integration would be expensive and time consuming.

## Connectors for Dataverse

There are three connectors for Microsoft Dataverse:

- Dynamics 365: This connector is deprecated and should not be used.
- Common Data Service: This connector is used when you are not using solutions. It allows connection to any Dataverse environment.
- Common Data Service (Current Environment): This connector is used when you use solutions. This is the connector of choice and has more options than the other connectors and is more flexible in the triggers available.

## The cost of doing nothing

A solution architect should consider the effort it will take to build an automation and decide whether it is justified. This involves weighing business value against the cost of automating the process. Business value is the ongoing benefit that the business receives from the project.

To figure out whether it is worth automating the process, you first must understand the cost of not solving the problem. As a part of defining the business value that you hope to achieve from the Microsoft Power Platform solution, you should get a better understanding of what it is costing the organization to solve the problem in the current manner. In other words, measure the cost of doing nothing.

If the business value you will receive by automating the process does not compare favorably to the cost of doing nothing, you must ask yourself whether this is the right business problem to focus on.

However, if the business value you receive by solving the business problem is greater than the cost of doing nothing, plus development time, and the monthly cost of any software licenses, it makes sense to automate the process.

## Triggers

Power Automate defines a trigger as an event that starts a cloud flow. These events can be initiated by a user, are scheduled, or are generated by a connector, including custom connectors.

The triggers used in Power Automate are defined by connectors. Each connector has its own set of triggers. Connectors falls in to two groups:

Tabular: Tabular connectors are for data sources where the data is stored in tables. Dataverse is a tabular connector. Tabular connections will have triggers around creating, updating, and deleting of records.

Function-based: Function-based triggers are for cloud services such as Twitter or Outlook. The triggers are related to functions in those services. For instance the Outlook connector has a trigger When an email is received.

## Types of Trigger

There are three types of trigger for initiating Power Automate cloud flows:

- Automated: A flow that is started by an event defined in a connector, such as a record being created or a file being added to a file location.
- Instant: A flow that is run manually by a user, such as a button being pressed.
- Scheduled: A flow that is run on a recurring basis, such as at 9 a.m. every workday, or every hour

We will focus on using the triggers for Dataverse. Automated flows can be triggered from data events in the platform's Event Framework. The Common Data Service (current environment) connector has a single automated trigger, When a record is created, updated, or deleted.

In the trigger step, you have to specify the trigger condition:

- Create
- Create or update
- Create or delete
- Create or update or delete
- Delete
- Update

- Update or delete

The Common Data Service (current environment) connector has a single instant trigger, When a flow step is run from a business process flow. This allows a Power Automate cloud flow to be manually initiated by a user as a step in a Business process flow.

The Common Data Service connector has a single instant trigger, When a row is selected. This allows a Power Automate cloud flow to be manually initiated by a user from within a form for a row in a Model-driven app.

A common pattern used with Power Automate cloud flows is to use the Scheduled trigger to run, for example daily, retrieve all the rows due for the day and then loop through and process them.

## Poll vs Push

Triggers represent notifications to Power Automate cloud flows that an event has occurred. They are generally grouped into two types of triggers: Poll and Push triggers. A polling trigger makes a call into the API at a reoccurring frequency to check for new messages. When new data is available, the trigger will run the flow. Examples include timer triggers. A push trigger responds to a push of new data from the service.

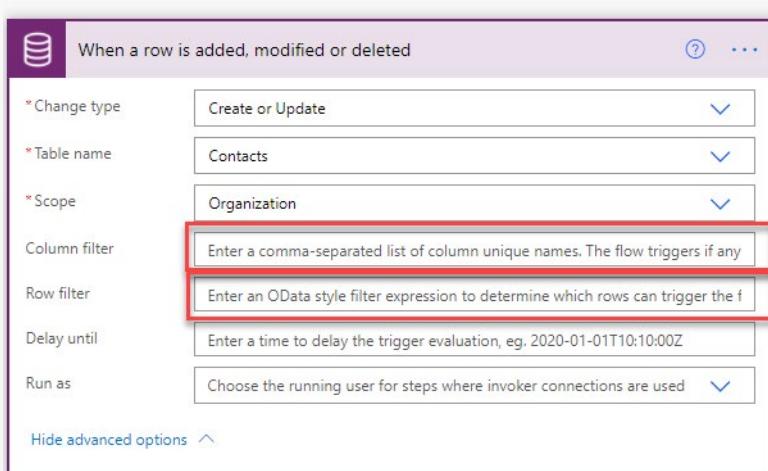
Whether a triggers polls or pushes is normally invisible to you when using the triggers. The deprecated Dynamics 365 connector used polling, the more modern Common Data Service triggers use push triggers. Push are more efficient and are more real-time.

Not every connector has a trigger. In such cases you can use a scheduled trigger and retrieve the data changed since the previous run, i.e., use a polling pattern. If you use this pattern, you need to ensure you do not run the flow too often and should leave several minutes between each iteration.

## Using filters

The solution architect should look to minimize the number of flows runs for each cloud flow. A common mistake that many makers make with Power Automate on create and update triggers is to a) retrieve the row and b) have a condition that checks if the flow needs to do anything. With the Common Data Service connectors you do not need to retrieve the row, the trigger step contains the new/changed data.

Additionally, like other triggers, the Common Data Service connectors can have a filter applied to the trigger. A filter will prevent the flow run being executed, reducing unnecessary executions of the flow.



In the previous diagram, the When a row is added, modified or deleted trigger for the Common Data Service (current environment) trigger has two filters that can be used:

- Column filter: If columns are specified, the flow will only execute if any of the columns listed are modified.
- Row filter: If an OData query is specified, the filter expression determine which rows can trigger the flow.

In Dataverse, if a column is included in a trigger's output, this does not mean that the value of the column has actually changed. If you need to ensure that you only process changed values, you need to use a plug-in with Pre and Post Images.

[!NOTE]

Instant triggers are not currently visible from apps if the flow is in a solution.

## Common Actions

The actions used in Power Automate are defined by connectors. Each connector has its own set of actions. We will look at common actions with Power Automate and Dataverse.

The Common Data Service connector has the following actions:

- Create a new record
- Get record
- List records
- Update a record
- Delete a record

The Common Data Service (current environment) connector has the same actions but also many more:

- Create a new row
- Get a row by ID
- List rows
- Update a row
- Delete a row
- Relate rows
- Unrelate rows
- Perform a changeset request
- Download a file or an image
- Upload a file or an image
- Perform a bound action
- Perform an unbound action
- Predict using AI Builder models

The Common Data Service (current environment) connector can also allows the creation and use of child flows.

## [!NOTE]

Common Data Service (current environment) has many advantages over the other connectors and should be your first choice with Power Automate.

## Retrieving data

When using "Get a row by ID" or "List rows" actions to retrieve data, specify only the column fields that are required for the flow.

With most Dataverse triggers they provide all the fields and you don't need to explicitly retrieve the record right after the trigger runs. However:

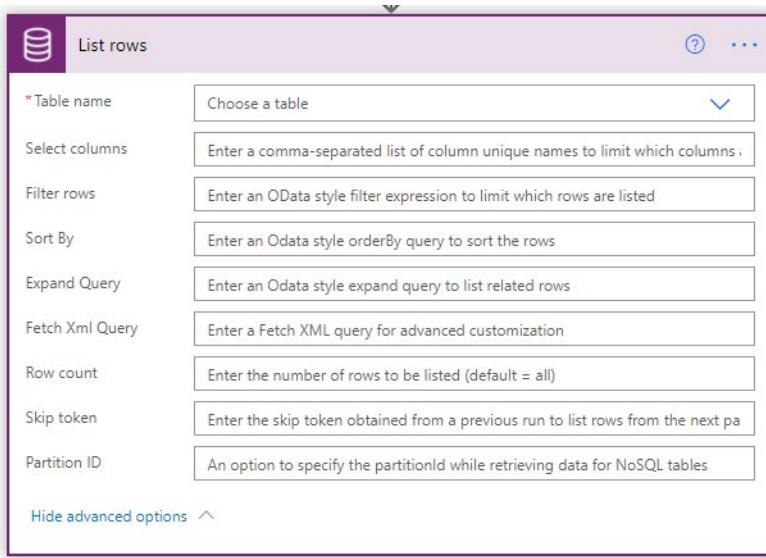
- The "When a record is selected" trigger doesn't include N:1 lookup fields and a Get a row by ID action will be required.
- Delete only includes the ID of the row no other columns.

## [!NOTE]

When referencing data previously retrieved earlier in a flow, the data is not updated automatically after the original step executes.

## Getting lists of data

The List rows actions can filter data using an OData query or a Fetch XML query to retrieve data.



When using the "List rows" action:

- Always filter the data on the List rows action not in a later step with Data Operation actions.
- FetchXML query filters can do more advanced criteria including related entities.
- Use Expand Query to include related rows in the result.
- Use the FetchXML Builder tool in XrmToolBox to create your OData and Fetch XML queries.
- Enable pagination if you want more than one page of data. The page size is determined by the connector e.g., Dataverse is 1024. Pagination is enabled via the action settings. The limit is 100,000 rows.

**[!NOTE]**

Expand Query can help reduce the complexity of your flows, reduce the number of API calls made, and speed up your flows.

## Updating data

When using the "Update a row" action:

- Only include the columns that have changed. This avoids triggering other automation and audit logging.
- Use the null expression to clear values.
- If you are using an Alternate key, providing a GUID for the primary ID will perform an Upsert operation using the other data columns you supply.

## Calling custom actions

Microsoft uses the term action for several different things in processes; for a type of process and for steps within processes and flows. Custom actions are processes similar to classic workflows in their capabilities in terms of what can be performed with steps; in other words, action processes can use conditions, and can create and update rows.

Cutsom actions are reusable processes that can be initiated from code and Power Automate. Microsoft also includes several built-in actions such as:

- Set Word Template: Creates a Word document from a Word template.
- Add To Queue: Adds a record to a queue.

Power Automate cloud flows can call custom actions using the Common Data Service (current environment) connector. You should use "Perform a bound action" for action processes associated with a table, and use "Perform an unbound action" for action processes set as Global.

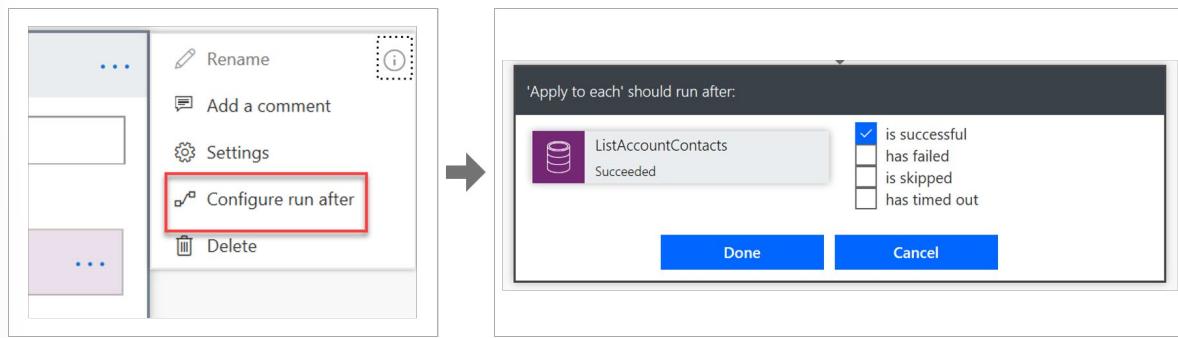
## Error handling

When a failure happens in a Power Automate cloud flow, the default behavior is to stop processing. You may want to handle errors and rollback earlier steps in the case of failure.

A solution architect should specify how errors are to be handled within cloud flows.

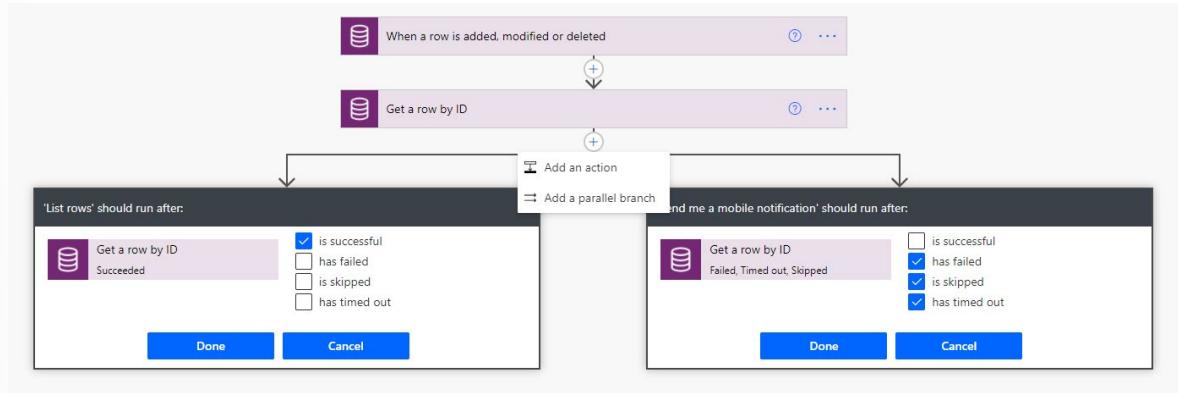
## Run after

The way errors are handled is by changing the run after settings in the steps in the flow as shown in the following image.



## Parallel branches

When using Run after you can have different actions for success and failure by using parallel branches.



## Changesets

If your flow needs to perform a series of actions on Dataverse data and you must ensure that all of these steps work or none of them work, the you should use a Changeset.

The screenshot shows a list of actions for a changeset:

- Executes a changeset request
- Update Available Cases
- Create Case
- Create Scheduled Call

Each action has a three-dot ellipsis icon on the right. Below the list is a button labeled "Add an action" with a plus sign icon.

If you define a changeset, the operations run in a single transaction. If any of the steps error, the changes made by the prior steps will be rolled back.

[!NOTE]

Changesets are only available with the Create, Update, and Delete actions in the Common Data Service (current environment) Connector.

## API limits

The platform has API limits and service protection limits. Power Automate observes these limits; each step in a flow consumes an API call.

There are also limits for certain actions in Power Automate and for Power Automate itself:

- Many operations like Apply for Each loop only work up to 100,000 iterations. You might require partitioning work when working with large sets of items.
- The Do Until loop has a default of 60 loops or 1 hour of execution. If it exceeds these, it exits the loop without error. You can increase the number of loops and time in the settings for the action
- Flows can execute for a maximum of 30 days. You should not use long running flows; use scheduled flows that check if the row needs to be processed instead.
- Connectors have throttling limits, for instance the Common Data Service (current environment) permits only 6,000 API calls per connection every 300 seconds.

## Best practice for cloud flows

There are several things your team do to make flows they build more maintainable:

- Use meaningful naming conventions – consider both admins and users.
- Rename each action e.g. List wows to List Account Contacts.
- Use comments on actions to document purpose.
- Use child flows to prevent overly complex flows or repeated logic.

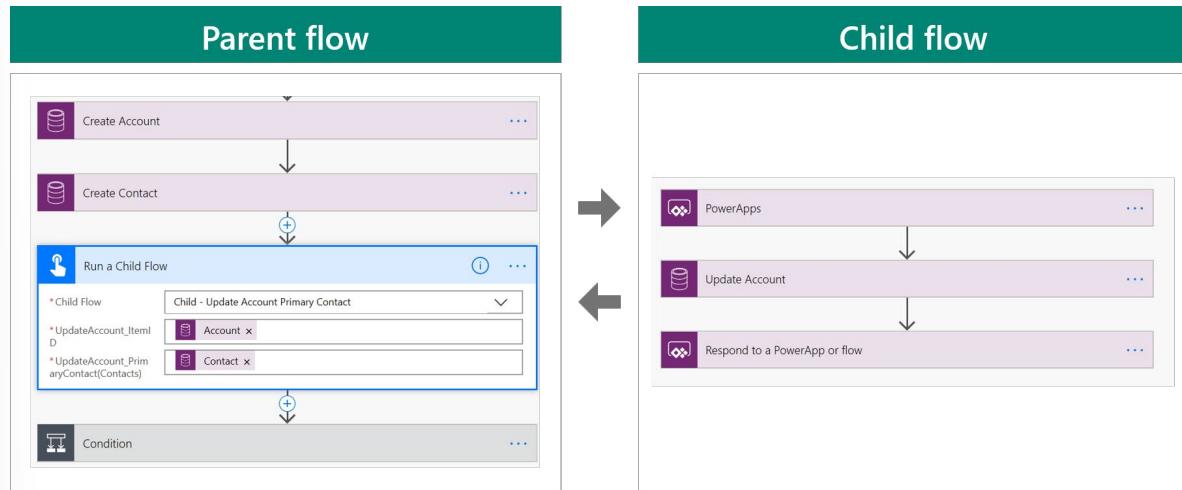
## Parent and Child flows

Using child flows allows breaking out parts of a flow into reusable child flows.

To create a child flow, you must create the flow in a solution and use the Common Data Service (current environment) trigger. The child flow can use the following triggers.

- Manually triggered button
- Power Apps
- HTTP Request

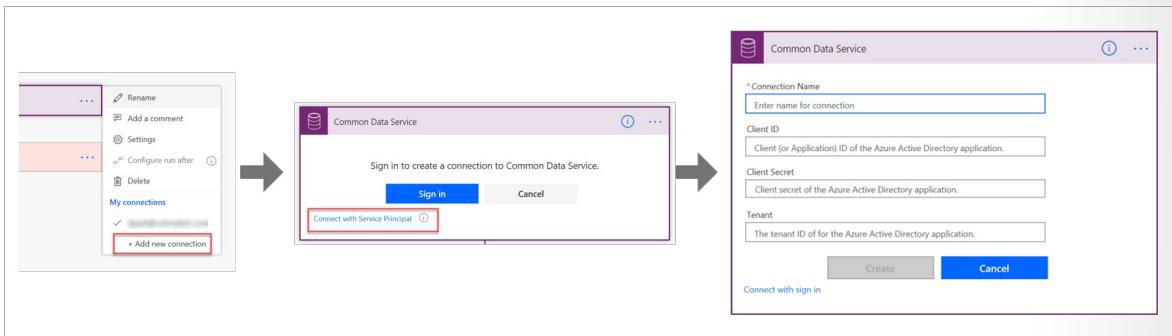
The parent flow must also be in a solution, and then can use the Run a child flow action. The child flow can pass data back to the parent flow with the Power Apps or HTTP response as shown in the following image.



## Using Service Principals

When running flows, you need to understand the user context the flow runs in. For Automated and Scheduled flows, the flow runs as the owner of the flow and the existing connections in the flow are used. An owner can edit and change the connection(s) used. For Instant flows, the owner can define that the flow runs as the owner, or will run as the user and require the user to create connections using their own credentials.

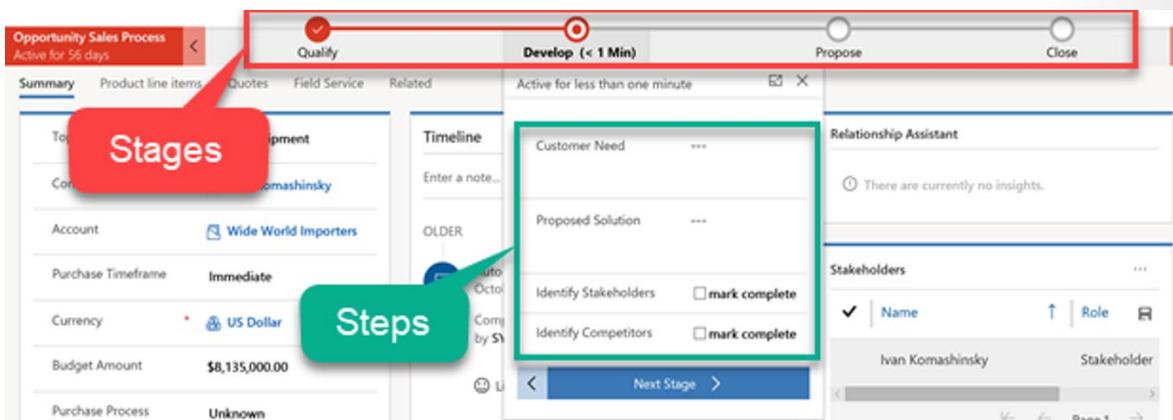
By default, when adding an action and signing into a connector uses the current interactive user's credentials and thus will execute in that user's context. Alternatively, you can sign in with a service principal using an app user created in Azure AD. This allows actions to be executed using a Dataverse app user context.



## Business process flows

Business process flows are a capability of the Microsoft Power Platform that helps users observe a defined business process, find out where they are in the process, and know what data is required to complete the next stage. Business process flows are used within model-driven apps to help guide the user through multiple steps and stages.

A Business process flow consists of stages and steps. A stage is linked to a Dataverse table, and steps are typically linked to columns in the table for the stage. The business process flow is shown at the top of a form in a model-driven app.



## Capabilities

Business process flows are interactive guides for users to get work done and to track major milestones in long running business processes. Business process flows have several capabilities that can be used in your solution to aid business processes:

- Stage gating: The steps in a stage can be mandatory, preventing the user from moving to the next stage in the process until the step has been completed.
- Conditional branching: The business process flow stages can alter which is the next stage in the process based on values entered.
- Multiple tables: A business process flow can include up to 5 tables in the process.
- Switch: A user can switch from one business process flow to another at any time.
- Security: Business process flows can be linked to security allows different users to use different business process flows.

When a Business process flow is created, a table called the process table, is created. This table is used to track the instances of the business process flow and their current stage. You can add this table to a model-driven app navigation and can create reports, charts, and dashboards to show usage of business process flows. There is a Power BI template app that provides visualizations for running of business process flows.

#### Automation and Developer APIs

Process management can be automated using the API or by Power Automate to start/stop the process as well advance stages automatically.

## Use cases

Good candidates for using Business process flows:

- Encourage outcomes, not wizard data capture.
- Link between related table; the form changes automatically from one table to another. This hides the data model from the user.
- Triggers automation based on progression between stages.

## Branching vs Multiple processes

A decision that may need to be made with Business process flows is to have one process with multiple branches or to have multiple processes. You can answer this by examining these issues:

- Do the processes need to run concurrently?
- Does the process need to return to same place after the branch?
- How will you decide which business process flow is used for new records?

## Immersive business process flows

Business process flows be used standalone outside the context of a model-driven app using a per process license. These are known as immersive Business Process flows.

You create an immersive business process flow by selecting None for the table to associate the flow with. The process table created then becomes the table where columns can be added, a form created, and data is stored.

# Power Automate exercise

## Exercise Overview

You will be working in small groups. Discuss each scenario and address the question. Each group takes a turn leading the larger group discussion on the proposed approaches.

During this exercise, you will be reviewing some scenarios and determining how to architect them using Power Automate.

## Learning Objectives

After completing the exercise, you will be able to:

- Evaluate requirements and propose a Power Automate flow design
- Evaluate a proposed design that includes Power Automate

## Scenario 1

When working on a help request for a smart bed customer, Contoso's support staff can request a replacement bed from the manufacturer. Each manufacturer provides a contact that can authorize the replacement. Once authorized, the support staff can inform the customer.

What would your design look like to implement this process?

## Scenario 2

When a repair is required, the system needs to look up the customer's support plan, check if they have available repair allowance / warranty, and if they do decrease the cost of the repair from their lifetime warranty allowance and prioritize the help request.

Your development team has proposed a plug-in – is that the best choice or how might you handle this in Power Automate?

## Scenario 3

When a manufacturer has a defect, they issue a recall to fix the problem. They will provide Contoso a file that contains the serial numbers of the affected beds. Each serial number must be looked up to see if there is an active customer and the customer record must be tagged with the recall. For some large recalls this could exceed 200,000+ beds. The process must be designed so it can be restarted if the process fails without creating any duplication.

What would your design look like to implement this process?

## Scenario 4

On create of a help request Contoso needs to calculate the customer deductible which is based a percentage determined by how old their bed is. The support staff needs this value displayed as soon as they create the record.

What would your design look like to implement this?

## Module Summary

### Summary

Power Automate should be your go to tool for automation and custom logic on Microsoft Power Platform projects.

Solution Architects should work with their teams to ensure consistency in use and use flow reviews to improve quality.

Always strive to use solutions for your automation.

### Next steps

Power Automate guidance documentation <https://docs.microsoft.com/power-automate/guidance/>



## Module 10 Security modeling

### Model security for Power Platform solutions

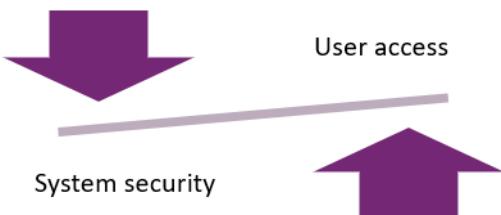
#### Introduction

The Microsoft Power Platform has many different security features in environments, in data, and in application deployment. Security is a fundamental component in the design in a Microsoft Power Platform solution. Too often security requirements are ignored until later on in a project. By including security modeling in designing of the solution, the need for significant change late in a project can be avoided.

[!IMPORTANT]

Security must be considered up front and not as an afterthought just before deployment.

The right security strategy balances legitimate security requirements with the need for system access and cross-business collaboration. When implementing Microsoft Power Platform solutions, the solution architect needs to balance two concerns; user access and system security.



On one hand, data and system security is important. The data drives the business, data that you do not want to fall into the hands of a competitor. With regulations surrounding personal data, organizations can be liable for data breaches that include personal data.

On the other hand, is system usability and user adoption. If the security model is too lax, users can view and change data that they should not be able to access; creating a perception that the data in the system is not reliable. If the security model is too stringent and you lock down everything so users can only see a small subset of the data in the system or can only perform limited operations, you diminish the value the solution, and users will revert to their old data silos and user adoption, the most important aspect of measuring the success of a solution, will suffer.

As a functional consultant or developer, you may not have been exposed to all the security features in the Microsoft Power Platform. As a solution architect, however, you will need to understand each security feature and when to apply it in different scenarios. This module is a fairly deep dive into all the security aspects of Microsoft Power Platform solutions and contains important information not found in modules for other roles.

## Security Architecture Process

To define security for a solution, the solution architect needs to break down the requirements and drive towards a clear picture of what the solution needs to look like. The solution architect will need to understand the organization's environment and their requirements around:

- Authentication
- Network Security
- Authorization

## Discovery

Discovery is about learning the organization's environment, procedures, and policies.

The solution architect needs to discover what is currently in use for authentication within the organization:

- Do they have Single Sign-On (SSO)?
- Are they using third-party products or just Azure Active Directory (Azure AD)?
- Do they use multi-factor authentication (MFA)?
- Do they use Conditional Access?

A single Microsoft Power Platform project is unlikely to change organization authentication approach and the solution architect will need to:

- Map security policies and requirements to the design.
- Create an initial authentication blueprint.
- Review the design with the organization's security representatives.

The solution architect needs to understand how authorization will be applied to the solution and will need to:

- Extract security-related requirements.
- Clarify security requirements for simplification.
- Create an initial authorization blueprint.
- Review with business analysts and security teams.

The solution architect needs to understand how the organization manages security:

- How is security managed?
- What security policies must be followed?
- What is the approval process for security architecture?
- How are application level entitlements managed?
- Which team will edit Microsoft Power Platform security?

- How are users assigned to applications?

When learning about the user environment, the solution architect also needs to discover:

- How does the organization structure influence security?
- Do people work in teams that cross organization boundaries?
- Is there a data classification system?
- What are the data retention policies?
- What are the privacy policies?
- What data access regulations apply such as GDPR?

## Solution Architect's role in security modeling

The solution architect leads the efforts to build a comprehensive security model that covers from authentication to data column level access.

The solution architect's role in relation to security consists of:

- Leading the efforts to build a comprehensive security model.
- Being able to communicate the options for security architecture at a high level and help guide the team members through the architecture design choices.
- Being an advocate for simplicity, keeping the security from being overcomplicated while at the same time ensuring necessary protections.

A solution architect also needs to consider if there are any security, regulatory, or compliance requirements that will impact the solution design.

## Security requirements

The security model will be different for each solution, but here is some guidance to consider when modeling security:

- Restrict: Users should only be able to edit data relevant to their role. However, it makes sense to be less restrictive about what data users can read as this helps users see their data in context. A solution architect should remove the ability to delete data unless necessary.
- Simplify: There are many security features in the Microsoft Power Platform. A solution architect should consider the impact of the security design on how complex the management of the system will be and how difficult it will be to change.
- Use: Often the security requirements that will be provided will come from a position of fear or lack of understanding of the capabilities of the Microsoft Power Platform. A solution architect should ensure that the security design is based in legitimate business requirements. This may require the solution architect to understand where the security requirements have originated and to discuss alternative approaches. Trying to lock down a Microsoft Power Platform solution to prevent users from performing actions can be difficult, if not expensive, if the capability to do so is not provided by the platform. The solution architect should use the security capabilities of the platform when designing security for their solution.
- Layer: The Microsoft Power Platform has security features on apps, data, and processes. Security should ideally be implemented at the platform layer for easier implementation and management.

- Review: The usage of the solution, when implemented, will not be as originally envisaged, and the patterns of usage will evolve over time. Sometimes the initial security design decisions are no longer valid and need to be adjusted.

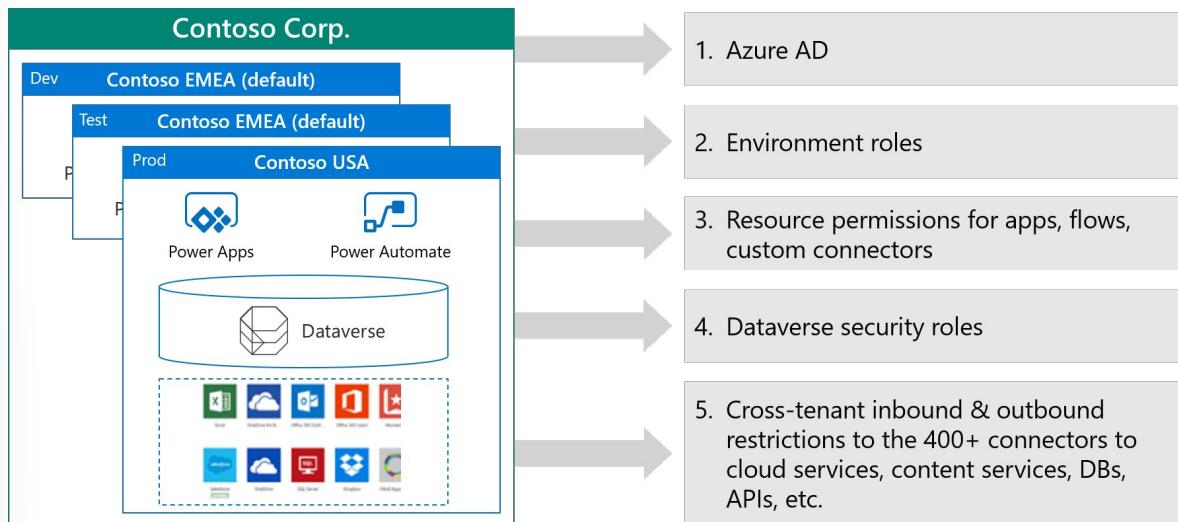
## Environment

Environments are containers that administrators can use to manage apps, flows, connections, and other assets; along with permissions to allow organization users control access to the environment and its resources.

Power Apps apps and Power Automate flows do not provide users with access to any data assets that they do not already have access to. Users should only have access to data that they really require access to.

Access to environments is multi-layered. Environments are contained within an Azure Active Directory (AAD) tenant. Access to the environment is authenticated by Azure AD. Users must have a user account in Azure AD to access any environment and have a license to use the Microsoft Power Platform. To access an individual environment user needs a Microsoft 365 role if there is no Dataverse database, and a Dataverse security role if there is a Dataverse database. Users also need to be given permission on resources in an environment such as apps, flows, and connectors.

An administrator can also control access to environments from other Azure AD tenants in B2B scenarios.



## Azure AD

The Microsoft Power Platform uses Azure AD to authenticate users for the Microsoft Power Platform.

Azure AD has several features that the Microsoft Power Platform can utilize:

- Identity and Access Management (IAM): Manage access to Power Apps apps
- Authentication: Verify credentials when a user signs in to an App or users a flow.
- Single sign-on (SSO): Access apps without having to log in separately,
- Multi-Factor Authentication (MFA): Prompted a user during sign-in for other forms of identification.
- Business-to-Business (B2B): Manage your guest users and external partners, while maintaining control over your own corporate data. Guest access is supported for the Microsoft Power Platform.

- Conditional Access: Manage access to your apps using signals to make decisions on allowing access.
- Device Management: Manage how your devices access your corporate data.
- Enterprise users: Manage license assignment.
- Hybrid identity: Azure AD Connect provide a single user identity for authentication and authorization, regardless of location (cloud or on-premises).
- Identity governance: Manage your organization's identity through employee, business partner, vendor, service, and app access controls.
- Identity protection (MSIP): Detect potential vulnerabilities affecting your organization's identities.
- Reports and monitoring: Insights into the security and usage patterns in your environment. The Microsoft Power Platform admin center provides more detailed insights than provided by Azure AD.

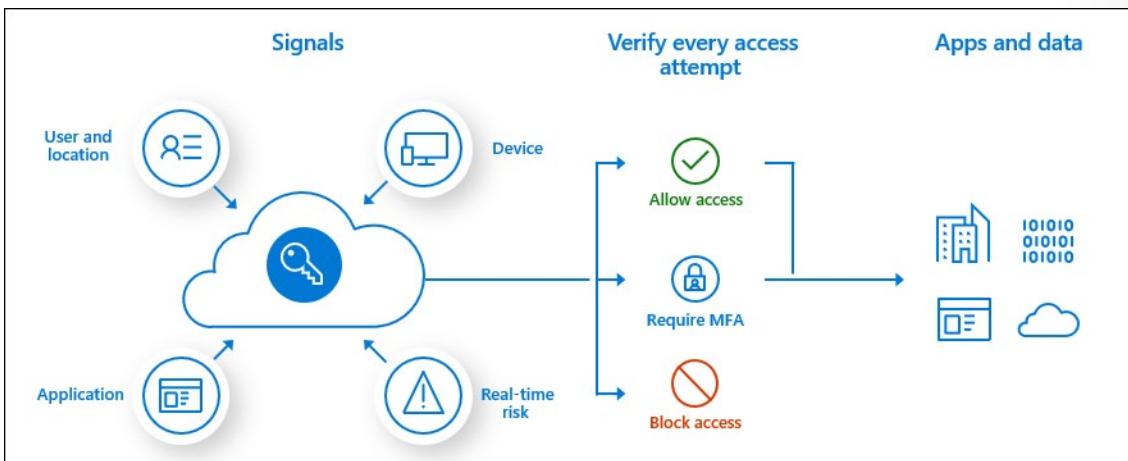
There are other features of Azure AD that are not currently supported with the Microsoft Power Platform such as:

- Business-to-Customer (B2C): Control how customers sign in using their preferred social, enterprise, or local account identities.
- Managed identities: An identity in Azure AD that can authenticate any Azure AD-supported authentication service. Currently not supported in the Microsoft Power Platform and service principals must be used instead.
- Privileged identity management (PIM): Provides elevated administrator access in a just-in-time manner.

An organization is likely to have chosen how it implements Azure AD and a Microsoft Power Platform solution is unlikely to change this fundamentally. However, the Microsoft Power Platform may provide access to users who have never had IT app access before such as mobile frontline workers, business partners, and external contractors who do not have identities in Azure AD. Therefore you may need to consider Conditional access, B2B, and Guest users.

Conditional Access required an Azure AD Premium license. Conditional access policies can grant or block access to apps and data based upon:

- User
- Device
- Location



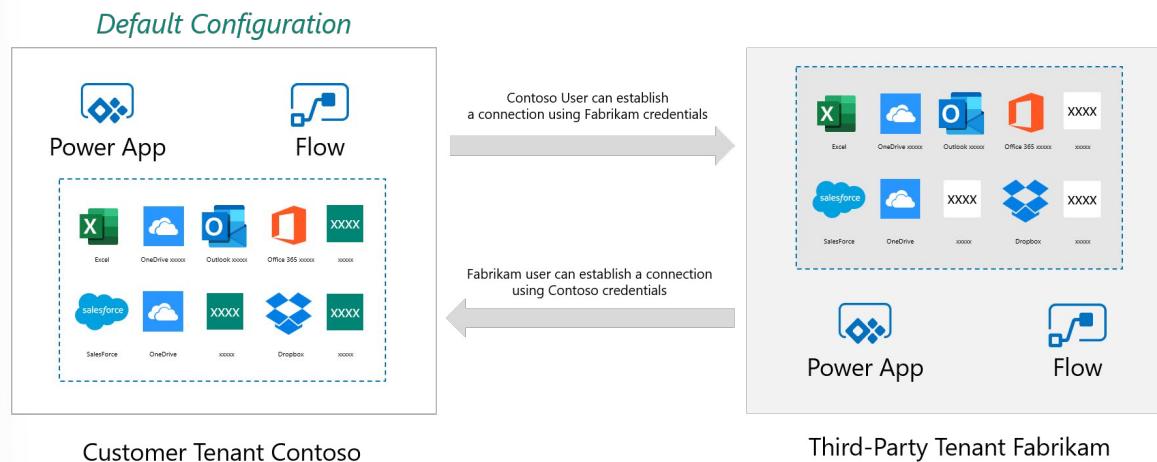
Azure AD B2B collaboration is a feature that lets you invite guest users to collaborate with your organization. You can assign licenses to guest users to run apps built with the Microsoft Power Platform. Guests cannot currently create or edit Power Apps.

For more information, see [enabling guests to access Power App<sup>1</sup>](#) and [Power Apps per app<sup>2</sup>](#).

## Cross Tenant isolation

In B2B collaboration scenarios, you securely share the apps and data in your tenant with a third party's tenant. In the default configuration, each tenant can access the other tenant's resources.

## Cross-tenant inbound & outbound restrictions



Azure AD has a feature called tenant restrictions. With tenant restrictions, organizations can specify the list of tenants that their users are permitted to access. Azure AD then only grants access to these permitted tenants. Cross-tenant isolation lets you restrict the other tenants your users can connect to. Cross-tenant isolation blocks inbound or outbound connections for canvas apps and flows.

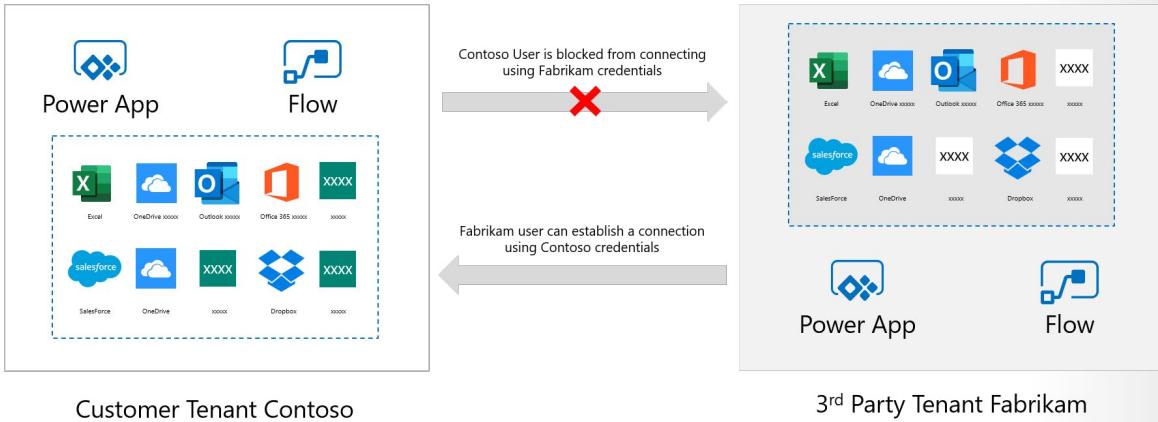
Restricting outbound connections means a user in your tenant is blocked from connecting to a third-party tenant. Restricting outbound cross-tenant connections can be done using tenant restrictions that apply to all Azure AD cloud apps that would block outbound connections for Power Apps and Power Automate flows.

<sup>1</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/share-app-guests/?azure-portal=true>

<sup>2</sup> <https://docs.microsoft.com/power-platform/admin/about-powerapps-perapp/?azure-portal=true>

## Cross Tenant Access – Restrict outbound

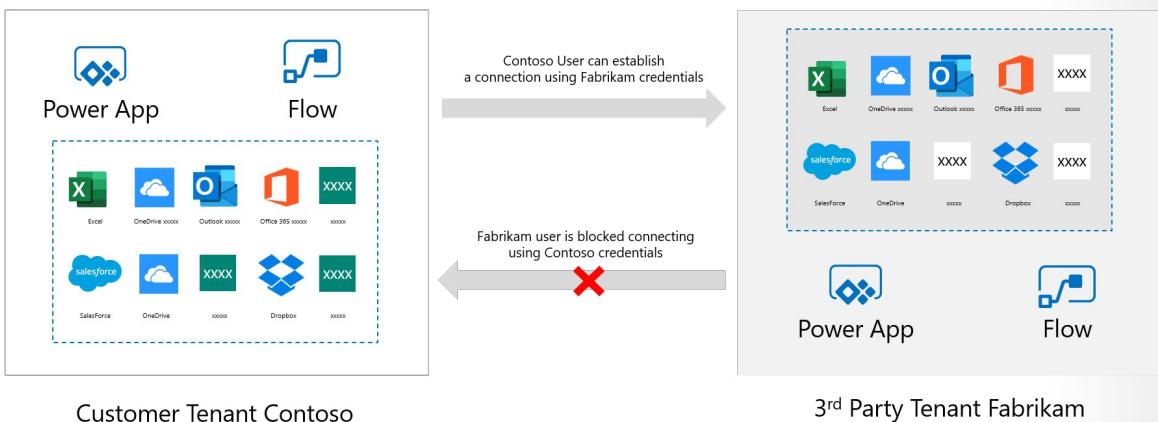
<https://aka.ms/adtenantrestrictions> => this applies to all Azure AD Cloud SaaS app



Restricting inbound connections means a user in a third-party tenant is blocked from creating a connection to your tenant.

## Cross Tenant Access – Restrict inbound

Requires support ticket today => this restriction only applies to Power Apps and Power Automate



[!IMPORTANT]

Restricting inbound cross-tenant connections requires a support ticket. This restriction then only applies to Power Apps and Power Automate flows.

## Securing Environments with Security Groups

By default a user with a Power Apps license can potentially access all environments in a tenant and you can see all licensed users in the user table in an environment. Azure AD Security groups can be used to limit access to environments. This has the added advantage of keeping the user table streamlined to just the users of the environment.

A security group can be associated with an environment when the environment is created, or it can be linked to the environment. In the following screenshot, a security group for the developers has been associated with the development environment.

The screenshot shows the 'Details' page for a Microsoft Power Platform environment named 'Development'. The page includes fields for Environment URL (redacted), State (Ready), Region (United Kingdom), Refresh cadence (Frequent), Type (redacted), and Environment ID (redacted). A red box highlights the 'Security group' field, which contains the value 'Developers'.

The following behavior is observed:

- When users are added to the security group, are added to Dataverse as users.
- When users are removed from the security group, they are disabled in Dataverse.
- When a security group is associated with an existing environment containing users, the users in the environment that are not members of the security group will be disabled.
- When no security group is associated with an environment, all users with a Power Apps license will be created as users and enabled in the environment.

[!NOTE]

Distribution groups and Office 365 groups cannot be associated with environments.

## Roles and Admin accounts

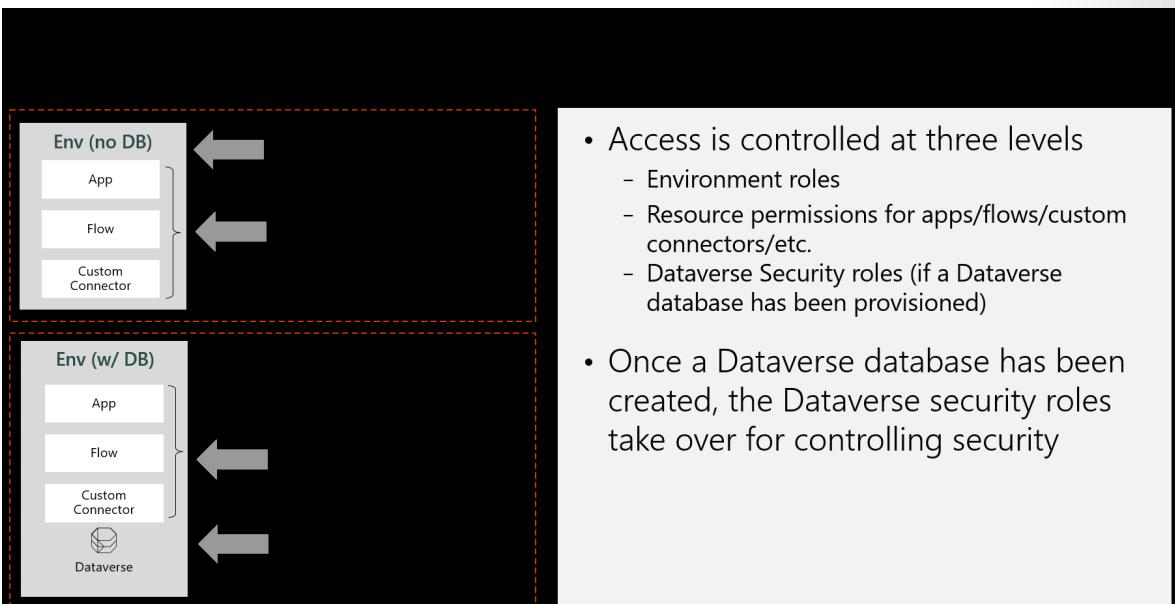
Managing environments and performing some administration activities requires one or more of these administrative roles:

- Global admin: A user with role has full administration access to all services in the tenant. Global admins have access to all Microsoft Power Platform environments and will be added to all Dataverse environments with the System Administrator security role.
- Microsoft Power Platform admin: A user with the Microsoft Power Platform admin can create environments, manage all environments even if the users do not belong to the security group for the environment, manage Power Apps, manage Power Automate, and manage Data Loss Prevention policies. A Microsoft Power Platform admin can also manage support tickets and monitor service health.
- Delegated admin: Delegated administration is used by Microsoft Cloud Solution Provider (CSP) partners to manage their customers through the Microsoft Partner Portal. A Delegated admin has full administration to all services in the customer's tenant.

## [!NOTE]

The Dynamics 365 admin role is deprecated. A user with the Dynamics 365 admin role can create environments, and manage environments but only if they belong to the security group for that environment. A user with the Dynamics 365 admin role will have the System Administrator security role in those environments.

These roles are not the same as security roles in an environment. Non-administrative users can be assigned a role for an environment. The roles that are used are different depending whether the environment contains a Dataverse database or not.



The Environment Admin role is only used when there is no Dataverse database. Environment Admin for a specific environment and allow the user with the role to add and remove users with the Environment Maker role. The Environment Maker role allows a user to create apps in that environment.

When a Dataverse database is added to the environment, a user with the Environment Admin role is given the System Administrator security role in the Dataverse database and a user with the Environment Maker is given the Basic User security role.

## Data loss prevention

Data loss prevention (DLP) policies act as guardrails to help prevent users from unintentionally exposing organizational data and to protect information security in the tenant. DLP policies enforce rules for which connectors are enabled for each environment, and which connectors can be used together.

The purpose behind DLPs is to prevent unauthorized data access and leakage from the Microsoft Power Platform. As solution architect, you need to be concerned with DLPs because:

- DLPs ensure proper data flow in an organization following organization policies.
- DLPs can prevent a developed solution from working properly if not included in your plans.
- DLPs Can break your solution after you deploy, if they are added later.

## Key facts

- Data loss prevention policies (DLP) enforce rules for which connectors can be used together.

- Connectors are classified into groups.
- A connector in one group can only be used with other connectors from that same group in an app or a flow.
- Tenant admins can define policies that apply to all environments.
- By default, no DLP policies are implemented in the tenant.

Connectors are classified as either business data only, no business data allowed, or blocked. A connector in the business data only group can only be used with other connectors from that group in the same app or flow. The names "business" and "non-business" do not have any special meaning, they are simply labels. The grouping of the connectors themselves is of significance, not the name of the group they're placed in. Connectors that are blocked cannot be used at all.

Name	Blockable	Class	Publisher	About
10to8 Appointment Scheduling	Yes	Standard	10to8 Ltd	<a href="#">Learn about 10to8 Appointment Sched...</a>
Act!	Yes	Standard	Swiftpage ACT!	<a href="#">Learn about Act!</a>
Animatira	Yes	Premium	Animatira	<a href="#">Learn about Animatira</a>
Adobe Creative Cloud	Yes	Premium	Adobe Inc	<a href="#">Learn about Adobe Creative Cloud</a>
Adobe PDF Tools	Yes	Premium	Adobe Inc	<a href="#">Learn about Adobe PDF Tools</a>
Adobe Sign	Yes	Standard	Adobe Inc	<a href="#">Learn about Adobe Sign</a>
Africa's Talking SMS	Yes	Premium	Africa's Talking	<a href="#">Learn about Africa's Talking SMS</a>

[!NOTE]

The Common Data Service connectors cannot be blocked.

## DLP and Environments

DLP policies can be scoped at the tenant level and at the environment level.

The screenshot shows the Power Platform admin center interface. On the left, there's a navigation sidebar with sections like Environments, Analytics, Resources, Help + support, Data integration, Data (preview), Data policies (which is selected and highlighted in purple), Admin centers, and Power BI. The main content area is titled 'DLP Policies > Edit Policy'. It shows a list of policy steps: 'Policy name' (set to 'Test'), 'Connectors', 'Scope' (selected), and 'Review'. To the right, there's a section titled 'Define scope' with the sub-instruction 'Choose the environments to add to this policy. [Learn more](#)'. Below that is a section titled 'I want to:' with three radio button options: 'Add all environments' (selected), 'Add multiple environments', and 'Exclude certain environments'.

It is possible to apply multiple DLP policies to an environment. At design and runtime, all policies that are applicable to the environment in which the app or flow resides are evaluated together to decide whether the resource is in compliance or violation of DLP policies. If multiple policies are configured for one environment, the most restrictive policy applies to the combination of connectors.

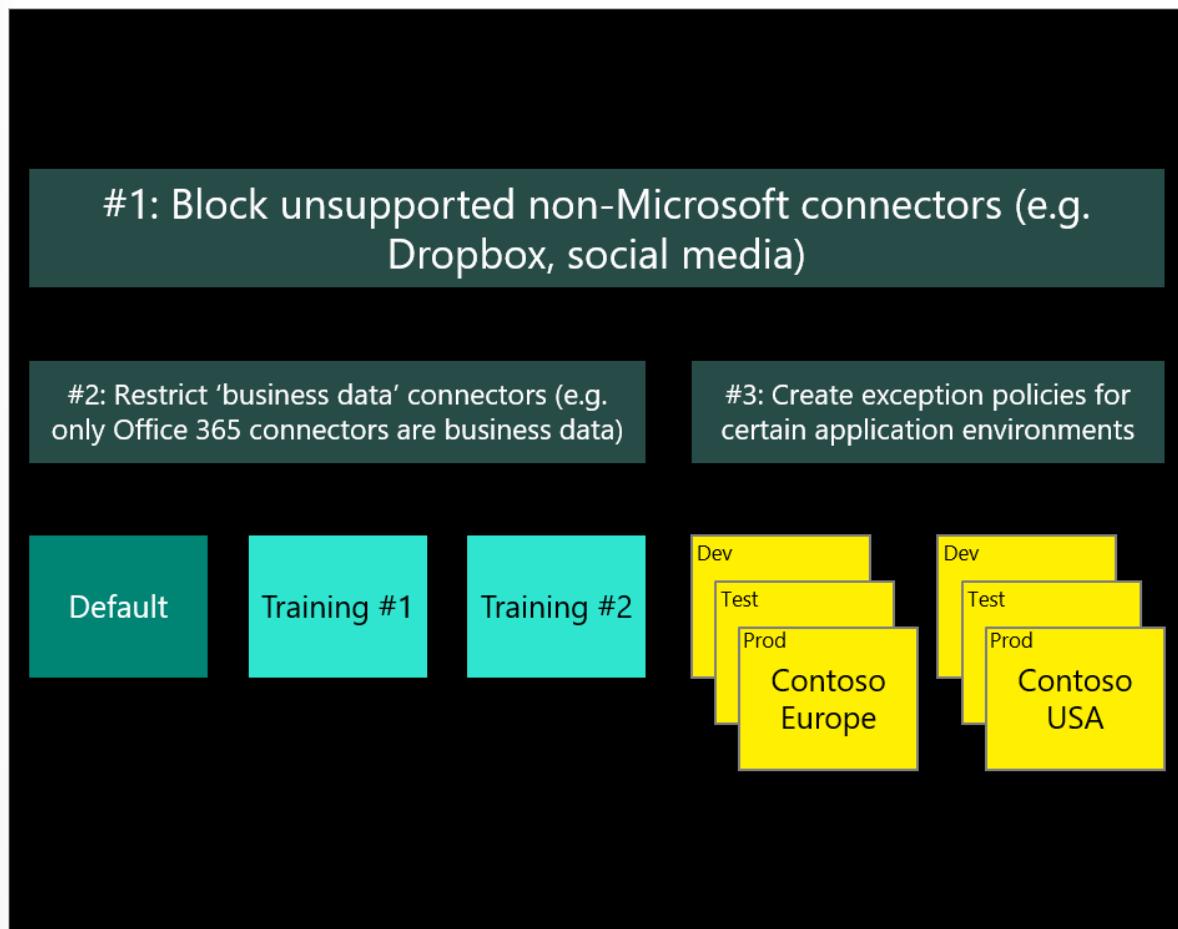
[!NOTE]

Environment DLP policies cannot override tenant-wide DLP policies.

The solution architect should aim to define the minimal number of policies and avoid, if possible having multiple policies applied to an environment.

The following approach is recommended:

1. Create a policy spanning all environments that block all unsupported non-Microsoft connectors and classifies all Microsoft connectors as Business Data.
2. Create a policy for the default environment (and other training environments) that further restricts which Microsoft connectors are classified as Business Data.
3. Create other policies, or exclude those environments from policies #1 and #2 above, that permit certain connectors to be used for specific environments.



## Deploying policies

The solution architect should consider the following with data loss prevention policies:

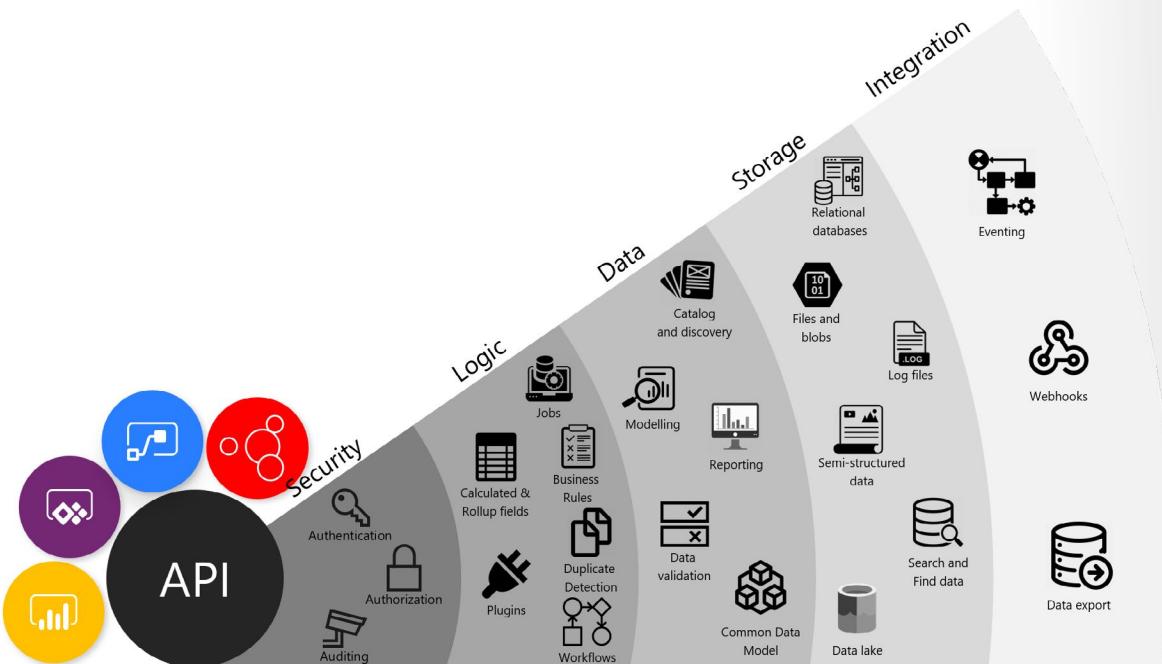
- It is best to establish a default policy early on and then grant exceptions.
- New and updated restrictions can disable existing apps and flows.
- Changes can take a few minutes to take effect.
- Policies cannot be applied at the user level only at the tenant or environment level.
- PowerShell and admin connectors can manage policies.
- Users of resources in environments can view policies that apply.

When deploying to an existing tenant, there are further considerations that a solution architect needs to liaise with other groups in IT:

- Confirm DLP policies are configured to support your deliverable.
- Know the lead time required for getting changes approved and implemented.
- Know what you can do, and what you require other groups to do.
- Use Azure AD Security Groups for controlling access to environments and resources.

## Dataverse

Data should always be considered as a valuable asset. Therefore designing a security model to ensure the proper usage and access to that valuable asset is important in security design. Microsoft Dataverse stores data for the Microsoft Power Platform and provides services for apps. The Microsoft Power Platform has layered security to control access to a Dataverse environment, but once a user has access to the environment, the solution architect needs to control access within the database. One of the key features of Dataverse is its rich security model that can adapt to many business usage scenarios.



## Authorization

Authorization is the function of specifying access rights, or privileges, to resources. Security design implements authorization to access data and features. Deciding who can get access to what data and what services is an essential part of any solution's design.

Bad security design can lead to poor system design and these can have far-reaching impacts and can be costly to fix:

- Manageability: Difficulty in managing individual access.
- Performance: For example, PrincipalAccessObject (POA) table growth when sharing.
- Usability: Cumbersome procedures to grant access.
- Visibility: Impossible to tell just by looking at the record who has access to it)

Every solution has unique requirements, but common capabilities and patterns can be identified and reused.

## Common patterns

Here are some common usage patterns within many business apps:

- Active involvement: Regular, significant involvement directly with the customer/data. Informed, with existing knowledge of the customer/data and current related activity, and personal actions based on a direct relationship with the people involved.
- Secondary involvement: Informed involvement, maintaining active knowledge of activity but not directly participating or acting on the data or with the customer, such as providing cover for absence of actively involved staff. Supporting others who have a personal relationship with customer such as providing advice or support to the people actively involved, providing specialist knowledge to a specific piece of data or customer.
- Transactional interaction: Specific activity-oriented involvement, for example, receiving and acting on a request to update a customer's address. No personal or on-going engagement, such as in a contact center.
- Management oversight: Managerial or governance responsibility across a business or geographical area □ Viewing and directing involvement of others rather than specific involvement.
- Reporting: Aggregated business reporting Data organized to preserve anonymity rather providing direct access to customers/deals.
- Compliance: Oversight read-only access to all records for a business area.

When designing security, the solution architect needs to understand how users work that is, do they work by themselves, work as part of a static team, or work in dynamic teams that change based on certain rules. These will affect the security design. Then the solution architect needs to understand how do other staff provide support. Do they have assigned support staff and are the support staff shared between other users? What happens if support staff are unavailable and overnight and at weekends? Also how is oversight handled?

## Design Principles

When designing security, there are some principles to follow:

- Assigned responsibility does not always need restricted access.
- Treat exceptions as exceptions, favor frequent access patterns.
- You cannot take back access for one record where you granted access to a broader set of data containing it.
- Use business units for manageability and containment, not for matching organizational structure.
- Use the simplest, performant design that meets the requirements.

[!NOTE]

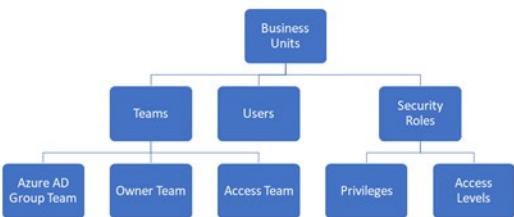
There are times where the organizational structure aligns with the required security structure but this is rare. When designing security, do not just use the organizational structure as the default.

## Security features

Dataverse provides many security features for accessing data:

- Table ownership
- Users

- Teams
- Teams
- Business units
- Security roles
- Sharing
- Azure AD Security groups
- Column-level security
- Hierarchical security
- Auditing



The solution architect needs to understand each of these features and how they combine to create the security model for their solution. This topic does not go into detail about each of these security features, but aims to explain how they can be used together to create a security model for your solution.

## Table ownership

Tables in Dataverse can either be User/Team owned or Organization owned. User/Team owned tables have an owner field and each row in the table can be assigned an owner. The owner of a row is used with the other security features to determine who has access to the row that is, access to rows is granular and can be horizontally partitioned. On the other hand, access to individual rows on Organization owned tables cannot be restricted and users will have access to all records or none.

[!NOTE]

You cannot change the ownership type after creation of a table. If in doubt, specify User/Team ownership when creating tables.

## Teams

Teams is collections of users that are used to allow users access to data. Teams is a powerful way of granting permissions to users in a broad way without micromanaging access at the individual user level. Users can be members of multiple teams.

There are three types of team:

- Owner team: Owner Teams can own rows, which give any team member direct access to that row.

- Access team: Access Teams is a method for users to easily share a row with another from within a form.
- Azure AD group team: Similar to Owner Teams but membership of the team is controlled in Azure AD.

## Security roles

Security roles are the cornerstone of data security in Dataverse. You do not grant discrete privileges to users, you create security roles. Dataverse uses role-based security to group together a collection of privileges.

Security roles can be assigned to users and to teams. Teams and users can have security roles that provide aggregate privileges.

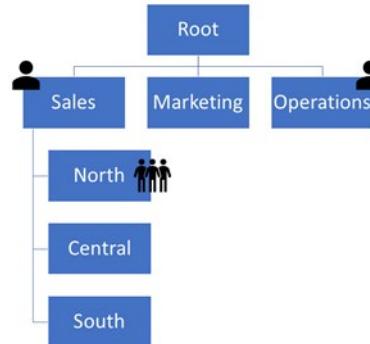
[!IMPORTANT]

All privilege grants are accumulative with the greatest amount of access prevailing. If you gave broad organization level read access to all contact records, you cannot go back and hide a single record.

## Business units

Business units contain users and teams and act as security boundaries. Business units are the primary method for controlling access to subsets of data within a table that is, horizontal partitioning of data. With Business units, you can segment users and their data.

Every Dataverse database has a single root business unit. You can create child business units to represent groups of users in a strict hierarchy.



[!NOTE]

Business units have a default team containing all users in that business unit. You cannot manually add or remove users from default teams so they can become blockers for some scenarios as the solution evolves.

[!IMPORTANT]

It is the combination of security roles and business units that is the foundation of the security model for Dataverse.

Business units provide an efficient way of managing large number of users and record access. Business units are not visible to the users in the applications, only to administrators. You should not simply mirror an organization's org chart but design the business unit hierarchy to meet the security requirements. This can sometimes mean creating business units to achieve a security requirement such as create a business

unit to parent the Sales and Marketing business units to allow some users to access those units and preventing access to the Operations business unit.

## Sharing

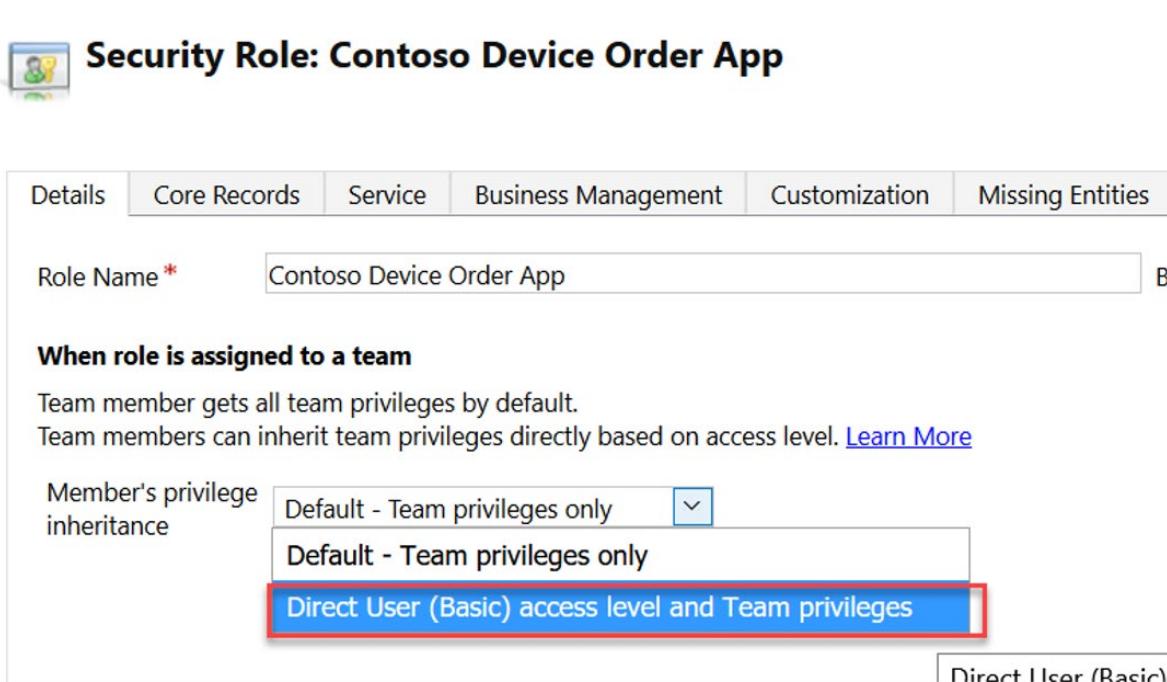
Individual rows can be shared with both users and teams. This permits users to access records that are restricted by the security model generated by business units. Sharing allows rows to be accessed outside of the strict business unit hierarchy.

## Security Roles and Teams

Security roles can be associated with teams. Users can then be associated with the team, and therefore all users associated with the team will benefit from the role. Users will have access to the rows owner by the team and depending on other security features, may have access to rows owned by other users in the team.

There is an option on the security role that controls how the security role operates with a team. There are two options:

- Default - Team privileges
- Direct User (Basic) access level and Team privileges



**Security Role: Contoso Device Order App**

Details	Core Records	Service	Business Management	Customization	Missing Entities
---------	--------------	---------	---------------------	---------------	------------------

Role Name \*: Contoso Device Order App

**When role is assigned to a team**

Team member gets all team privileges by default.  
Team members can inherit team privileges directly based on access level. [Learn More](#)

Member's privilege inheritance:

- Default - Team privileges only
- Direct User (Basic) access level and Team privileges**

Using the Direct User option treats privileges as if they were directly assigned directly to the user. Using this option you can avoid having to assign security roles to users and just use teams and security roles assigned to teams.

[!NOTE]

Owner Teams belongs to Business Units. A user can only belong to one business unit at a time, but a user can be added to a team in another business units. This allows users to access data in a business unit that is not in their hierarchy.

## Column-level security

The privileges provided by security roles operate at the row level. If a user has update privilege on a row, they can update all columns on that row. Sometimes row-level control of access is not adequate such as columns containing personal identifiable information (PII). Dataverse has a column-level security feature to allow more granular control of security at the column level.

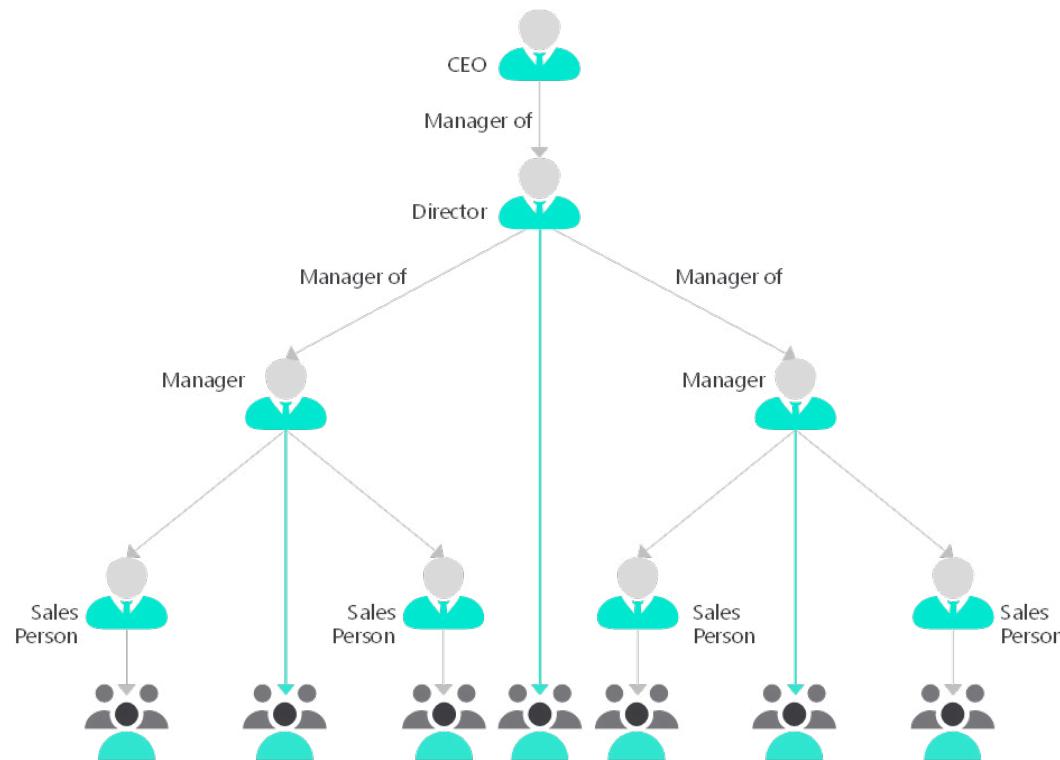
Column-level security operates separately to security roles. Column security profiles define the read/write privilege on the columns and these profiles are assigned to users and teams.

[!NOTE]

If a user does not have the privilege to read a secured column, the can still see the column on the form but will not be able to see its contents i.e., the value of the data. When using code to access secured field, the value will be null if the user does not have read privilege.

## Hierarchical security

One of the issues with the use of business units is their strict hierarchy. Access to data can only follow the hierarchy of the business units, , in the diagram of business units a user in Sales can be given access to the North, Central, and South business units, conversely a user in the Operations business unit cannot access data that belongs to Sales without being given access to all data in the organization.



In hierarchical security, the user in the position above can read and write all the data for their subordinates. They can then read data for the users lower in the hierarchy.

Hierarchical security is an alternative security model that is designed for scenarios where a user that requires management oversight is not within the same part of the business unit hierarchy. Hierarchical

security is useful where managers are in different countries or departments from their direct reports. Hierarchical security has two options:

- Manager hierarchy: This uses the user hierarchy on the systemuser table. There is a restriction that a manager is in the same business unit or in the direct chain of business units above.
- Positional hierarchy: This uses the Position table. This is more flexible and ignores the business unit structure. Positional hierarchy also allows for more than one person in a position.

Only one of the hierarchical types can be used at the same time.

## Audit

Audit captures all data changes. Auditing can help post monitor who has made what changes and can also help analyze how users are actually using the system. The audit functionality within Dataverse does not capture read of data or actions such as exporting to Excel.

There is other auditing available, called activity logging, in the Microsoft 365 Security and Compliance Center that includes data reads and other operations. Activity logging must be enabled and helps meet compliance objectives.

## Managing security across multiple environments

Security roles and Column-security profiles can be packaged into solutions and transported to other environments. Access Team templates are part of the table metadata and are included with the table when added to the solution. Other security features, such as Business units and Teams, cannot be packaged in solutions and the solution architect will need to plan for their population on environments.

## Strategies for defining security roles

There are three basic strategies for building security roles:

- Position specific
- Baseline + Position
- Baseline + Capability

The position specific is the creation of a single security role that contains all the privileges that the job role requires. The out-of-the-box roles are position-specific and are named after job roles, for example, Salesperson. You can follow this model of roles for specific job roles, but you may end up with many security roles that are similar and that you must maintain. For example, adding a new custom table will mean changing many, if not all, the roles.

A more modern model is to use a layered security model. In this model, you would copy an out-of-the-box role, such as Salesperson or Basic User, and change the access levels to the common, or minimal, levels that all users require. You might name this role Baseline. You would then create new roles and just set the access levels for the few privileges that each group of users would need in addition to the base role. These minimal roles contain the other privileges required for the position or the capability required.

In the following diagram, the baseline role is named All Staff and all users will be assigned this role. All users in Sales will also be assigned the role named Sales, a position role, with some sales users being assigned the Mobile role, a capability role, that just has the Go Mobile privilege, and manager being assigned the Sales role and the Manager role.



Security roles, although this is not obvious are linked to business units. Roles created at the root business unit are inherited by all child business units. You can create a security role for a specific business unit if you want to limit that role to the users in that business unit. Be aware, however, that security roles in solutions are always added to the root business unit when the solution is imported.

## Azure AD Group Teams

An Azure AD group team is similar to an Owner team in that it can own records and can have security roles assigned to the team. The difference is that team membership in Dataverse is dynamically derived from the membership of the associated Azure AD group. Azure AD group membership can be manually assigned or can be further derived from rule-based assignment based on the user's attributes in Azure AD.

Combining Azure AD Group Teams with assigning security roles to teams with the Direct User option can significantly simplify management of adding new users.

Both Security groups and Office 365 groups can be used for Azure AD Group Teams.

## App security

When setting up security, there are four different layers of security you can set up for an app.

- App-level security: App-level security restricts access to the app.
- Form-level security: For model-driven apps, form-level security allows you to allow only specific security groups to access specific forms. This is useful if you want to restrict how people enter or view data by their job role.
- Row-level security: The Dataverse security model controls access to rows.
- Column-level security: Column level security controls access to individual columns in a table.

Security should happen at the platform layer, not the app layer. There are numerous ways to control reading and writing data in an app. You can set columns to read-only on your model-driven form, you can use JavaScript to mask columns from the user experience, and you can hide columns fields from forms and views. None of these approaches really implement security. They do not secure the data and users can still get to the data in other ways such as with Advanced Find or Edit in Excel Online.

Additionally, all users are entitled to use the API and can use third-party and community tools to access the data. For proper security, the security features of Dataverse should be employed.

## Elevated privileges and impersonation

You should avoid giving users high levels of privileges. Plug-in .NET assemblies, Classic workflows, and Power Automate cloud flows can all run with elevated access to perform actions on behalf of the user. API code can impersonate another user if necessary.

## Automation

A solution architect should consider automating aspects of managing users and security. For instance, you can control many aspects from creating teams to sharing rows via the API.

You could trigger plug-ins or Power Automate flows based on events that occur in the system to change a user's security for example, Elevate the backup account manager automatically or share their data so they can manage account while the primary account manager is on vacation.

## Performance

Solutions with larger number of users and/or data must be aware of impact of choices. For example:

- Too much sharing can create a lot overhead.
- Too many business units can cause slow access.
- Too many processes running on events.
- Poor plug-in design.

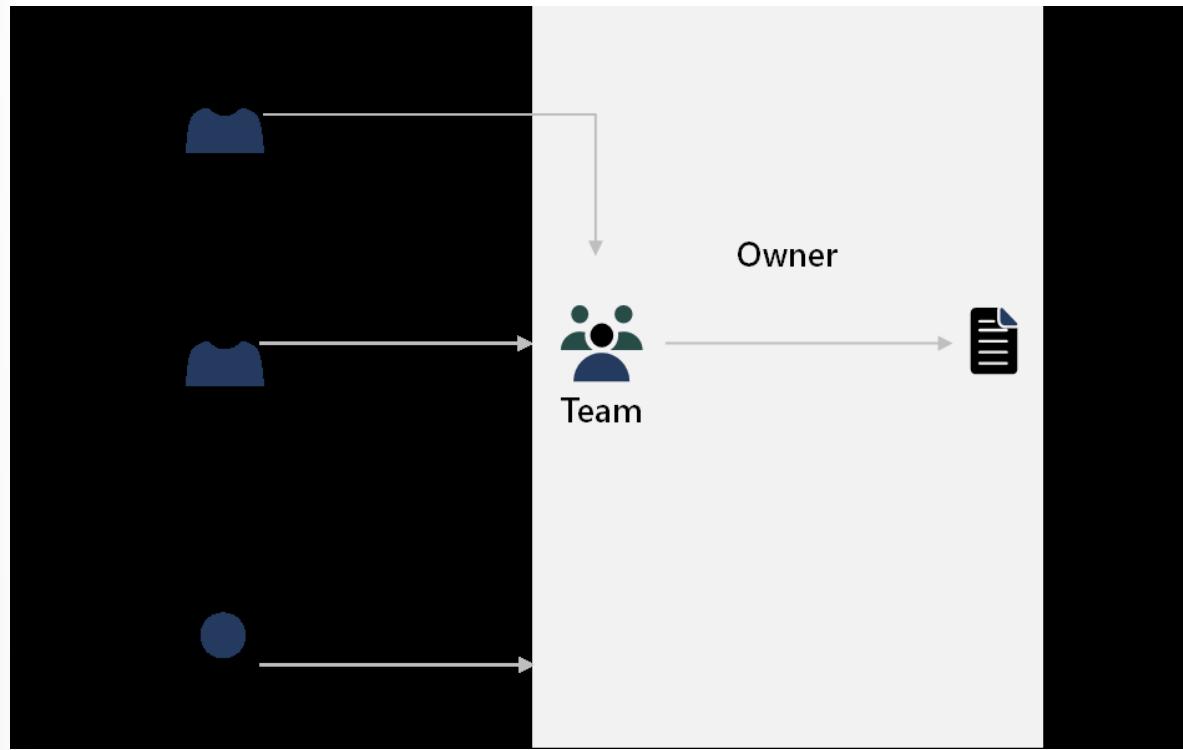
A poor security design can lead to poor performance. There are techniques you can use such as:

- Looking for ways to escalate security, for example sharing with a team instead of a user.
- Minimize the number of business units.
- Using access teams instead of owner teams.
- Testing with real volumes and real security scenarios to validate the design.
- Using the Analytics tools in the Microsoft Power Platform Admin center to view API calls and high volume processes.

The best way to improve performance is to keep the security design as simple as possible.

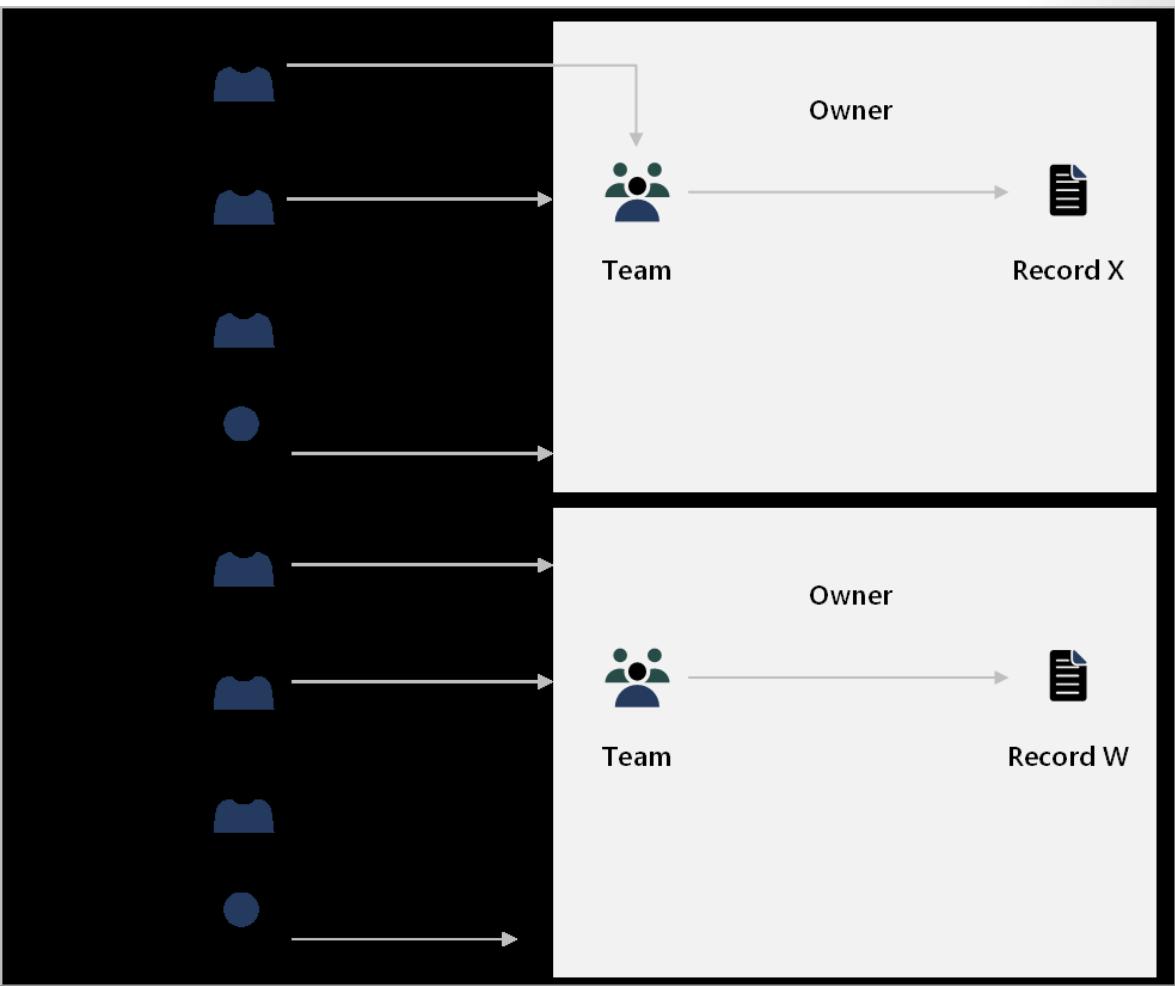
## Separate and optimize usage patterns

The solution architect should optimize for different usage patterns. This means using the different security model features to provide necessary access as shown in the following diagram with different users having different ways to access the same record.



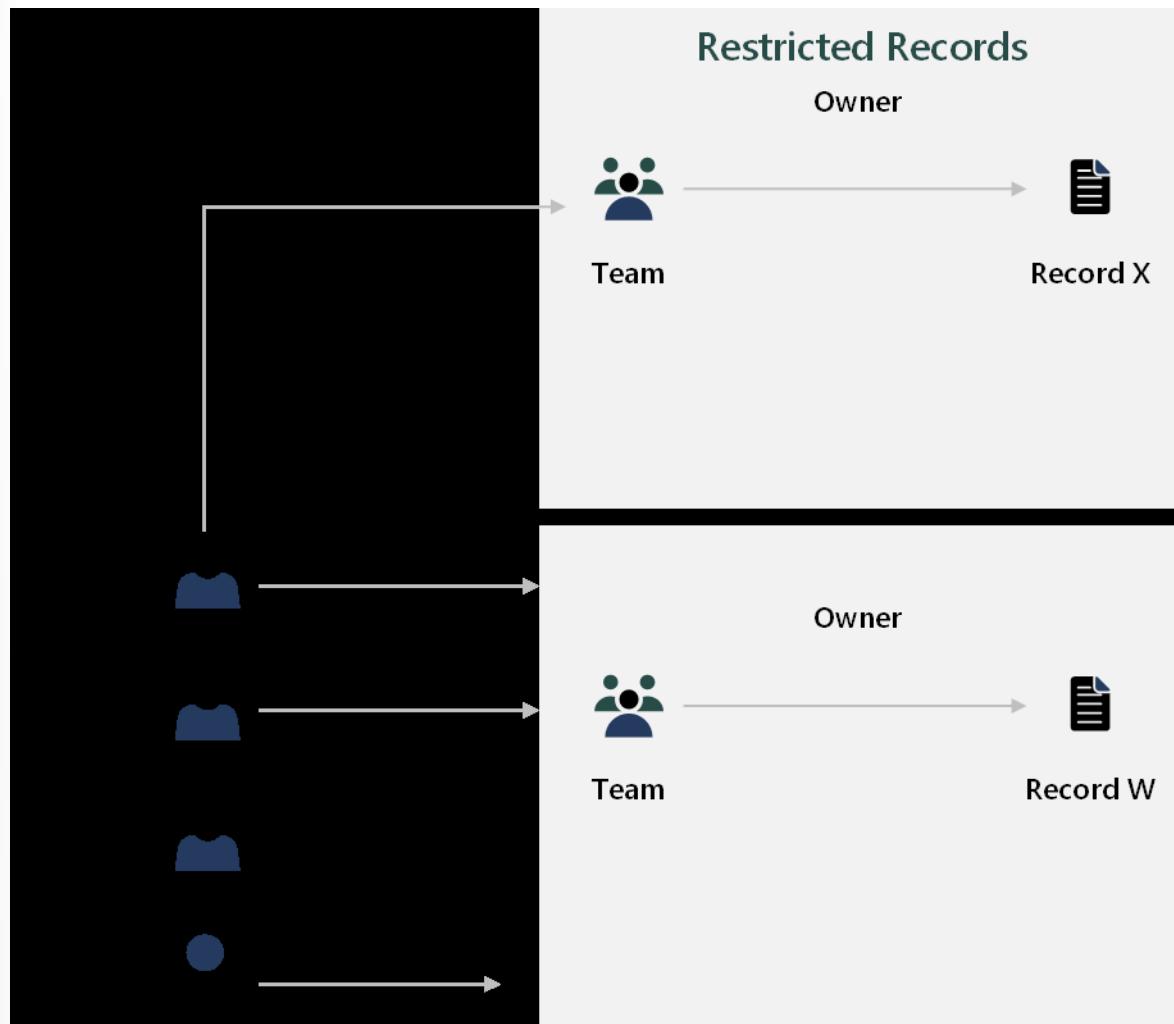
## Different business areas can be modeled differently

Not all users work in the same way. The solution architect should reflect different working models of different parts of the business and model each area differently to get an optimum solution as shown in the following diagram.



## Model exceptions as exceptions

The solution architect should aim to model the common access patterns as efficiently as possible and use a more granular model where complex access is required. Sharing is a good example of exception processing. Sharing should be with teams whenever possible.



## Separate historical from active data

Infrequently accessed but high-volume historical data can impact current data access. The solution architect should consider partitioning data in the security model in separate tables and provide a secondary mechanism for occasional access

## Review data model to help security

Some data model adjustments can make security modeling easier. The solution architect should determine if defining new data boundaries simplifies security modeling e.g.m instead of having individual access defined to Account records, moving financial reporting info from Account to a separate Financials table allowing the Account table viewable by all, with Financials table limited to managers only.

# Security modeling exercise

## Exercise Overview

In this exercise you will work in small groups. Review the information presented about Fabrikam Robotics and complete the tasks.

### Learning objectives

After completing the exercise, you will:

### Exercise 1

You are building a solution for Fabrikam to track visitors to a showroom and manufacturing site. Some of the visitors are potential purchasers and some are just there to see the magic of the robots working.

Review the requirements here and address the concerns listed later in this document.

- Fabrikam has been rapidly moving to the cloud and already leverages Office 365 for their email. They are adopting the use of Microsoft Teams at a rapid pace.
- You have already architected the solution to include the following Power Apps:
  - A model-driven app named Sales Central to be used by the sales staff and sales managers.
  - A canvas app to check in guests visiting the showroom to be used by the reception staff.
  - A portal that visitors can use to request to visit.
- Fabrikam is going to use Power Apps per app licenses for reception staff and Dynamics 365 Sales app licenses for the sales staff.
- Fabrikam's IT security team wants to make sure the new solution doesn't open up any security exposure from potential data leaks. You must ensure that your security setup prevents users from building their own flows in Power Automate that send data to outside services that aren't approved. You could almost see the panic on their faces when your sales team did the demo of how easy it was to build a flow using random 3rd party connectors.
- The sales staff handles their own leads and opportunities. They are compensated by commission on close of sales and are very competitive. The staff that works on small deals all work together and receive commission on all closed deals in their department. The large deals sales staff mostly work by themselves; except on very large deals where there is a team of them assigned.
- Reception staff must not have access to the sales data.
- Sales managers need to have access to the deals of their team members.

- Discounts can only be approved and applied by one of the sales managers. The app should ensure the sales staff can't approve their own discounts. In the data model, there is a discount column of type currency and a required lookup column to the approver.
- Fabrikam's organization has the following divisions and departments:
  - Sales
    - Small Deals
    - Large Deals
    - Wholesale
  - Marketing
  - Operations
    - Manufacturing
    - Showroom (has all the reception staff)
    - Shipping
  - Customer Service

## **Discuss as a group how you would address each of the following:**

1. How do you calm the concerns of the security staff and the rogue connector use?
2. How do you handle ensuring only sales managers approve and apply discounts?
3. How would you control access to each of the Power Apps?
4. What do you need to do to accommodate that Power Apps per app licenses are used for the reception staff?
5. What CDS Security Roles would you have the team create?
6. What business units can you begin to define to support your solution?
7. How will you handle so portal users can only see their own visit request records to check the status and potentially cancel the visit?
8. How would you ensure the small sales department can see all the data for all small sales staff, but the large deals sales staff can only see their own except when a team is put together to handle very large deals?

# Module Summary

## Summary

In this module, you looked at security modeling with the Microsoft Power Platform, including:

- Learning about different aspects of security architecture
- Learning about securing environments
- Learning the security features of Microsoft Dataverse
- The impact of security on performance
- Modeling different usage scenarios

Security must be considered from the start of the project. The solution architect must ensure that security is designed into the data and app models. It can be hard to change later on after the data model has been created and apps composed. Security modeling is unique to each solution; there is no one size fits all.

Different industries and business models follow different approaches to granting access. The solution architect should look for common patterns that emerge from requirements, often these emerge for how the relationships with the stakeholders of the system are managed.

Above all, the solution architect should keep things a simple as possible while still meeting the business needs to secure data and optimize the user experience.

## Next Steps

Learn more with the **Review the security model for your Dynamics 365 solutions module<sup>3</sup>**.

Read more **Securing the app and data documentation<sup>4</sup>**.

Read more **Security concepts in Dataverse<sup>5</sup>**.

<sup>3</sup> <https://docs.microsoft.com/learn/modules/fast-track-security/?azure-portal=true>

<sup>4</sup> <https://docs.microsoft.com/powerapps/guidance/planning/security/?azure-portal=true>

<sup>5</sup> <https://docs.microsoft.com/power-platform/admin/wp-security-cds/?azure-portal=true>



## Module 11 Integration

### Implement integrations with the Power Platform

#### Introduction

The solution architect leads the identification of integrations both in and out of the Microsoft Power Platform. The solution architect will guide how implementations are designed as part of the overall architecture.

In this module, you will learn the different ways to integrate with the Microsoft Power Platform and the solution architect's role in integration. You will learn about:

- What is integration and why do we need it
- Microsoft Power Platform features that enable integration.
- Leveraging the capabilities of Azure.

The business app you are building is often just a part of an overall bigger picture. While in the user's perspective the app is connecting to various other enterprise systems, the app could just be a participant in a much larger enterprise process flow that spans multiple enterprise systems.



## What is integration

Integration is the connecting one or more parts or components of systems to create a more unified experience or to ensure a more consistent outcome of a process. Integration results in a system that acts as one, not as individual parts executing on their own agenda.

Integration is often thought of as stitching together pieces to create a greater whole. Integration allows for things like data integrity, better user adoption, higher ROI etc.

Different components can be connected or disconnected and integration is the process of determining how best to get them to work together in a coordinated way.

## Why do we need integration

Here are six common variables involved in determining the necessity of integration. Each variable defines an inherent problem, which allows for rectification through integration.

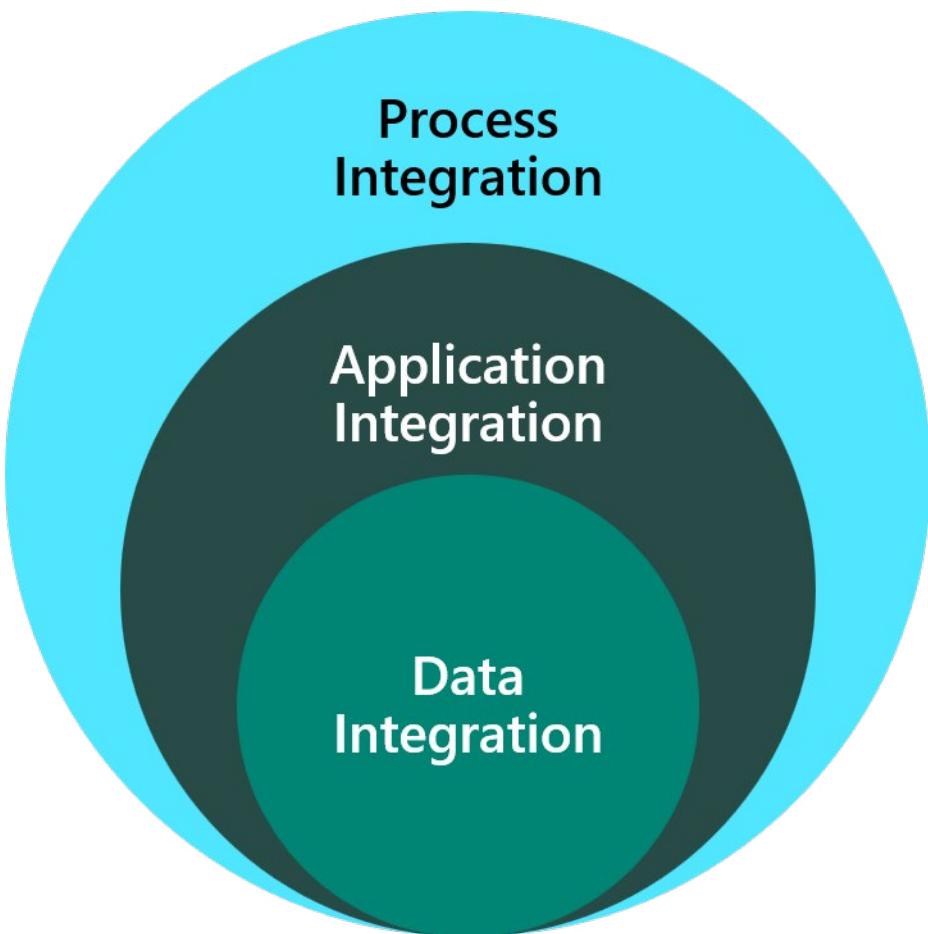
- **Usability:** From the user's perspective, a common problem in enterprises is the variety of different systems they need to interact with to perform a job. Through integration, the tasks a particular user needs to perform can be provided in a single, seamless, user interface. Training costs and time to perform a task can be decreased. This can also lead to greater consistency and customer satisfaction.
- **Volume:** Where data volumes are large, or are changing regularly, it can be problematic to duplicate. Through integration, rather than by copying or migration, the data can be accessed from a common place.
- **Real-time:** It is important to have access to up-to-date information about customers. Because customer data can be managed by different teams or as part of a regulated process, it may not be possible to use the same system to support both business needs. Integration can make it possible to access up-to-date data in real-time, ensuring accuracy every time.
- **Cost:** Some functionality is cheaper to access externally rather than reproduce. One example is address lookup. Integration to an external provider can be cheaper than replicating the capability from the raw postal service source data within the system.

- Duplication: Consistency of data is critical. One example is allocation of service resources to tasks. Duplication may end up in double-booking, resulting in an inability to deliver the service required. Although this capability may be required across multiple business areas, it is not uncommon for a single system to manage the allocation and offer that consistent service to other systems, an ability that integration can provide.
- Reuse: Reimplementing common functionality is expensive, particularly when ongoing maintenance and regression testing of enhancements are considered. An approach where common functionality is reused, rather than being reimplemented, can often be cheaper and lead to greater consistency. Integration can provide this vital undertaking.

## Types of integration

There are three types of integration:

- Data Integration: Combining data from different sources and presenting the user a unified view.
- Application Integration: A higher-level integration connecting at the application layer.
- Process Integration: Combining multiple disparate systems, each of those systems is part of an overall business function.



## How can the solution architect help?

The solution architect:

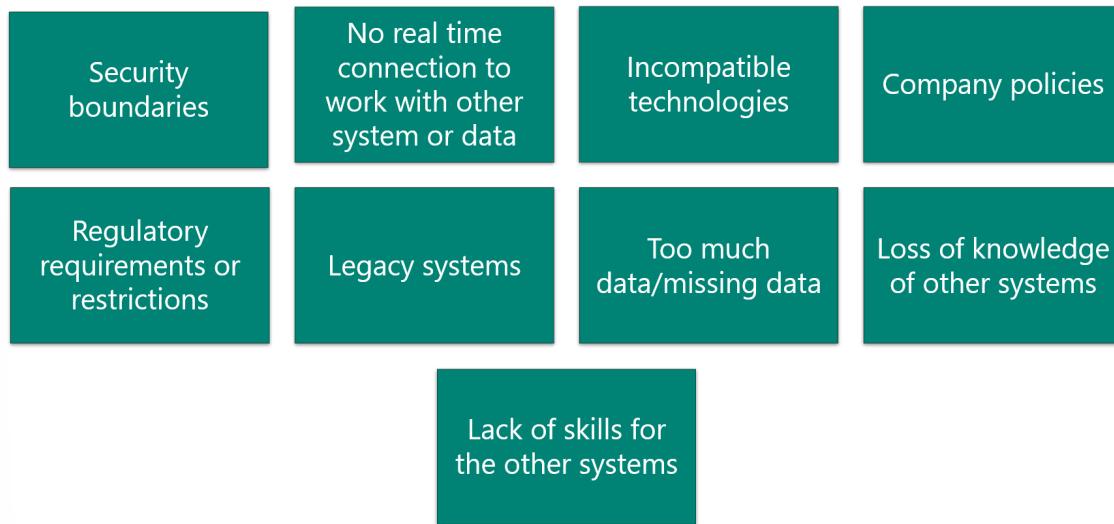
- Identifies where integration is required.
- Leads the design effort on how integrations are implemented and ensuring that integrations fit within the overall architecture.
- Leads the evaluation efforts of third-party integration tools.
- Ensures that the integration doesn't make the solution too fragile.
- Considers integrations as part of overall disaster recovery plan.

The solution architect even evaluating integrations should always consider "The cost of doing nothing". To figure out whether it is worth integrating data, applications, or processes, the solution architect must first understand the cost of not solving the problem. As a part of defining the business value that you hope to achieve from integrating with your Microsoft Power Platform solution, you should get a better understanding of what it is costing the organization to solve the problem in the current manner or what it would cost to perform manually. In other words, measure the cost of doing nothing. A key question to ask is how often does this happen. You may have seen situations where an integration has taken six months to develop, only for it to be synchronize a small number of customer details a day.

## Common challenges

Integrations can be expensive and complex, but just as real are the perceived challenges.

Below are some common integration challenges that a solution architect may face:



The solution architect must ensure that integrations are not brittle; tightly coupled systems are hard to change.

## Influencers

These are some items that will influence how you design integrations:

- Volume of data being moved or accessed.
- Quality of data.

- Latency to access or work with other system.
- Security requirements.
- Reliability requirements.
- Impact of duplication of data or functionality.
- Fit with existing Microsoft Power Platform capability.
- Cost, time, and resources.
- Internal politics.

## Causes of failure

Consider the projects you have had experience of and seen issues with integrations. What caused the integrations to fail?

The following factors will destine integrations to fail:

- Underestimating the complexity of integrating.
- Poor user experience using the integrated solution.
- Increasing cohesion of components creating a fragile system.
- Not knowing what Microsoft Power Platform does or does not do well.
- Not knowing what the other system does or does not do well.
- Source data is of poor quality, or is full of duplicates and dirty data.
- Not being clear as to what the system of record is.
- Having multiple parties involved and not coordinating the building of the integrations.
- The other parties building integrations not knowing the Microsoft Power Platform.

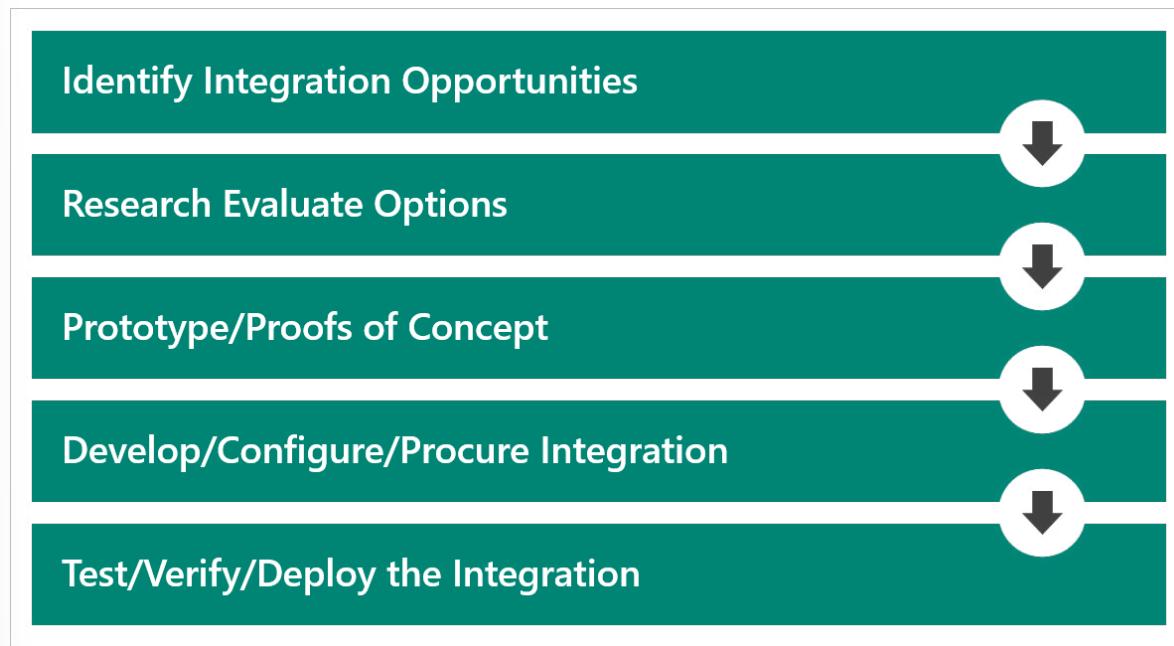
## Designing for resilience

The solution architect needs to ensure that integrations are designed to be resilient:

- Expect transient errors with your integrations.
- Include escalating retry logic with the circuit breaker pattern to eventually fail.
- Use queuing or other loosely coupled techniques to increase resiliency.
- Include in your designs how to handle common expected failures.

## Integration design process

Every situation is unique you will encounter a wide variety of integration scenarios on projects. While it is useful to be introduced to a technology and how to integrate, it is more important that you develop skills to evaluate real project needs. However, using the principals, ideas, and concepts you can navigate through complex integration challenges. The following diagram shows the steps you should follows:



The design process is a balance and there is often no right or wrong answer or solution. When considering the options for integration the solution architect needs to take into account the skills available in the team and the breadth of capabilities of the Microsoft Power Platform.

 App Builders	 Code Developers
Business Processes Data synchronization Canvas apps Embedded canvas apps	Power Automate flows Custom Connectors ETL Tools

In some cases, it is cheaper to hire staff than to build the integration. As technologists, we often overlook non-technical solutions that are still viable. You should also be asking is the integration needed and does it need to be real time.

[!IMPORTANT]

The solution architect should also consider using Power Automate Desktop flows to perform integrations at the user interface level.

## Data integration

When evaluating integrations, the solution architect should categorize each piece of data as this will guide the solution architect towards the appropriate integration solution. The following are ways to categorize data:

- Volatility: Is the data highly volatile that is, is it rapidly changing.
- Volume: How large is the volume of data?
- Time sensitive: Does the data need to be real time.
- Batch: Can the data be processed in batch or must it be processed on a transaction by transaction basis?
- Regulated: Does the data contain personal information or are there restrictions on where the data can be stored.
- Licensed: Is the data licensed and are there limitations on the data use?

## Inbound data integration

Inbound data integration is concerned with getting data into Microsoft Dataverse so that it is available to apps and flows.

## API

The Web API is one of two web services you can use to work with data and metadata in Dataverse. The other is the Organization Service.

The Dataverse Web API provides a development experience that can be used across a wide variety of programming languages, platforms, and devices. The Web API implements the OData (Open Data Protocol), version 4.0, an OASIS standard for building and consuming RESTful APIs.

All data operations that use the Dataverse APIs, whether using the Web API or the Organization Service, are converted into messages that follow the platform's event framework. This framework allows processes such as classic workflow and Power Automate cloud flows to be initiated, and allows developers to add custom plug-in steps that can be executed to perform validation and further processing.

## Event vs Batch

The solution architect should categorize the data that is required in Dataverse. A key category is event or batch based. The following diagram compares these two approaches.

Event Based	Batch Based

## Push pattern

When considering getting data into Dataverse, there the solution architect will need to determine if data is pushed into Dataverse by another system, or is pulled into Dataverse.

The general push pattern for inbound integration into Dataverse is to use the Web API with the other system making the Web API calls. However, allowing other systems to write directly into Dataverse requires the other systems to understand the data model in Dataverse and how the processes in the Microsoft Power Platform solution operate. It is often a good idea to create a layer for external systems to access:

- Event-based processing: Power Automate and Azure Logic Apps are a good approach for individual transactions triggered by changes in the source system.
- Batch processing: Batch is often abstracted by third-party tooling such as KingswaySoft or by using Azure Data Factory.
- Azure Functions: Azure Functions can abstract the need to implement business logic within your enterprise integration layer.
- Custom API: Create your own API for other systems to call.

[!NOTE]

Power Automate is often used to synchronize data between Dataverse environments.

When designing integration solutions consider using multiple threads to overcome latency effects and service limits.

## Pull pattern

The Pull pattern can be effective for data augmentation. You can use the pull pattern to get the data from external system on-demand when rows are retrieved in Dataverse. Virtual entities can be a good fit for this pattern.

## Alternate Keys

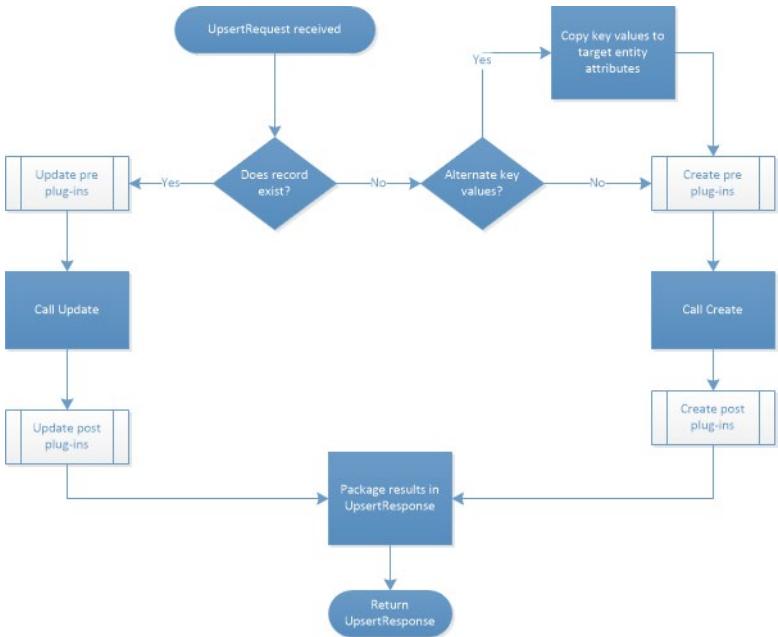
In Dataverse tables, rows are uniquely identified using a GUID. Other systems that need to integrate with Dataverse will either need to record the GUID in the database or have to query Dataverse to find the row that needs to be updated. This is inefficient. Dataverse provides the capability to create alternate keys on tables.

An alternate key allows external systems that need to read and write rows to efficiently access the rows without having to first run a query to find the GUID. For example, accounting systems often have an alphanumeric account number that uniquely identifies the account. You can define the account number field in the Dataverse table to be an alternate key so that the accounting system can read and write the account using the data it holds in its own system.

## Upsert

You can reduce the complexity involved with data integration scenarios by using the Upsert message. When pushing data into Microsoft Dataverse from an external system, you may not know if a record already exists in Dataverse. In such cases, you will not know if you should use an Update or a Create operation. You first need to perform a query to determine if it exists before performing the appropriate operation. You can now reduce this complexity and load data into Dataverse more efficiently by using the Upsert message.

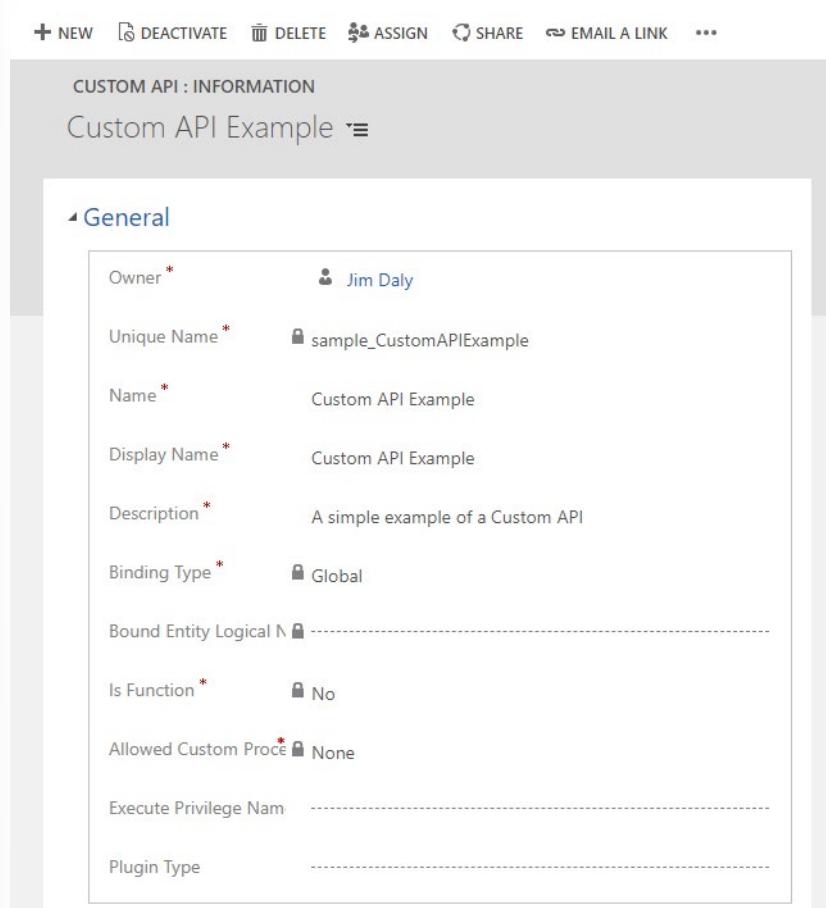
Upsert is used with Alternate Keys. You provide enough information in the Upsert call. Dataverse will look up the row and either create or update the row as shown in the following diagram.



## Custom APIs

Custom APIs are newly released functionality that you can abstract and consolidate a group of operations into an API that the other systems can call.

Custom APIs are defined by creating a custom API record as shown in the following screenshot.



A Custom API can be a Function or an Action. You should use a Function, which is a GET request, to get information, and an Action when you want to modify data.

A plug-in is used to perform the actual data operation for the Custom API.

[!NOTE]

For more information, see **custom APIs<sup>1</sup>**.

## Azure Functions

Azure Functions enables developers to create complex reusable custom logic and integrate with the other systems. Azure Functions can be consumed with Webhooks or wrapped inside a Custom Connector.

Using Azure Functions, developers can create reusable components for functional consultants and app makers to use in their apps and flows. Azure Functions can also be accessed by other applications to push and pull data into Dataverse. The Azure Function can connect to Dataverse and access the data.

You can create an API for your solution using Azure Functions to create with custom server-side logic and expose the API securely through Azure API Management (APIM).

## Outbound data integration

Outbound data integration is concerned with taking data from Microsoft Dataverse and making it available to other systems.

<sup>1</sup> <https://docs.microsoft.com/powerapps/developer/data-platform/custom-api>

## Dataverse Event publishing

Dataverse provides an event model for integrating with other systems. Dataverse supports triggering of custom code and external actions when events are detected in the platform. This is known as the Event Framework. The Event Framework provides the capability to register custom code to be run in response to specific events. The Event Framework is typically used to trigger custom plug-in code but can also be used for integrations with other systems.

To leverage the Event Framework for your solution, you must understand:

- What events are available
- How the event is processed
- What kind of data is available when the event occurs
- What time and resource constraints apply
- How to monitor performance

You can specify different stage

- PreValidation: For the initial operation, this stage will occur before the main system operation. This provides an opportunity to include logic to cancel the operation before the database transaction. This stage occurs before any security checks are performed to verify that the calling or logged-on user has the correct permission to perform the intended operation.
- PreOperation: Occurs before the main system operation and within the database transaction. If you want to change any values for an entity included in the message, you should do it here.
- MainOperation: For internal use only except for Custom API and Custom virtual table data providers.
- PostOperation: Occurs after the main system operation and within the database transaction. Use this stage to modify any properties of the message before it is returned to the caller.

[!NOTE]

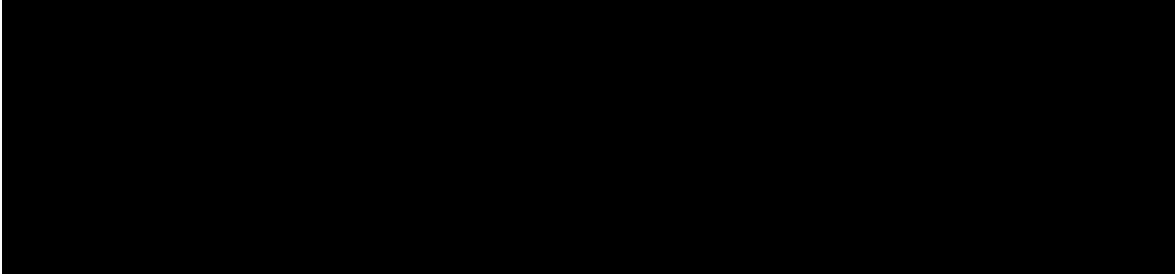
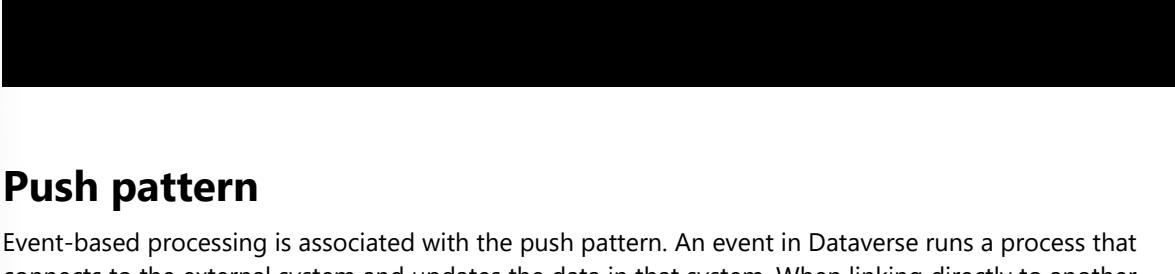
Integration typically use the PostOperation stage and the asynchronous execution mode.

The event framework can trigger:

- Plug-ins
- Classic workflows
- Power Automate cloud flows
- Messages to Azure Service Bus and Azure Event Hub
- WebHooks

## Event vs Batch

The solution architect should categorize the data that is required in Dataverse. A key category is event or batch based. The following diagram compares these two approaches.

Event Based	Batch Based
	

## Push pattern

Event-based processing is associated with the push pattern. An event in Dataverse runs a process that connects to the external system and updates the data in that system. When linking directly to another system, the solution architect needs to ensure that they are creating performance issues for both systems and are not creating a tightly coupled system.

[!IMPORTANT]

If using plug-ins, the solution architect needs to be aware of the 2-minute time limit for plug-in processing.

## Pull pattern

The pull pattern uses external events from the other system or a scheduled trigger to retrieve a set of data from Dataverse and process the set of data returned.

[!NOTE]

The Recurrence trigger in Power Automate is often used in the pull pattern.

## Change Tracking

The change tracking feature in Dataverse provides a way to keep the data synchronized in an efficient manner by detecting what data has changed since the data last synchronized. Without change tracking, it is difficult to build a reliable and efficient mechanism to determine what rows have changed in Dataverse.

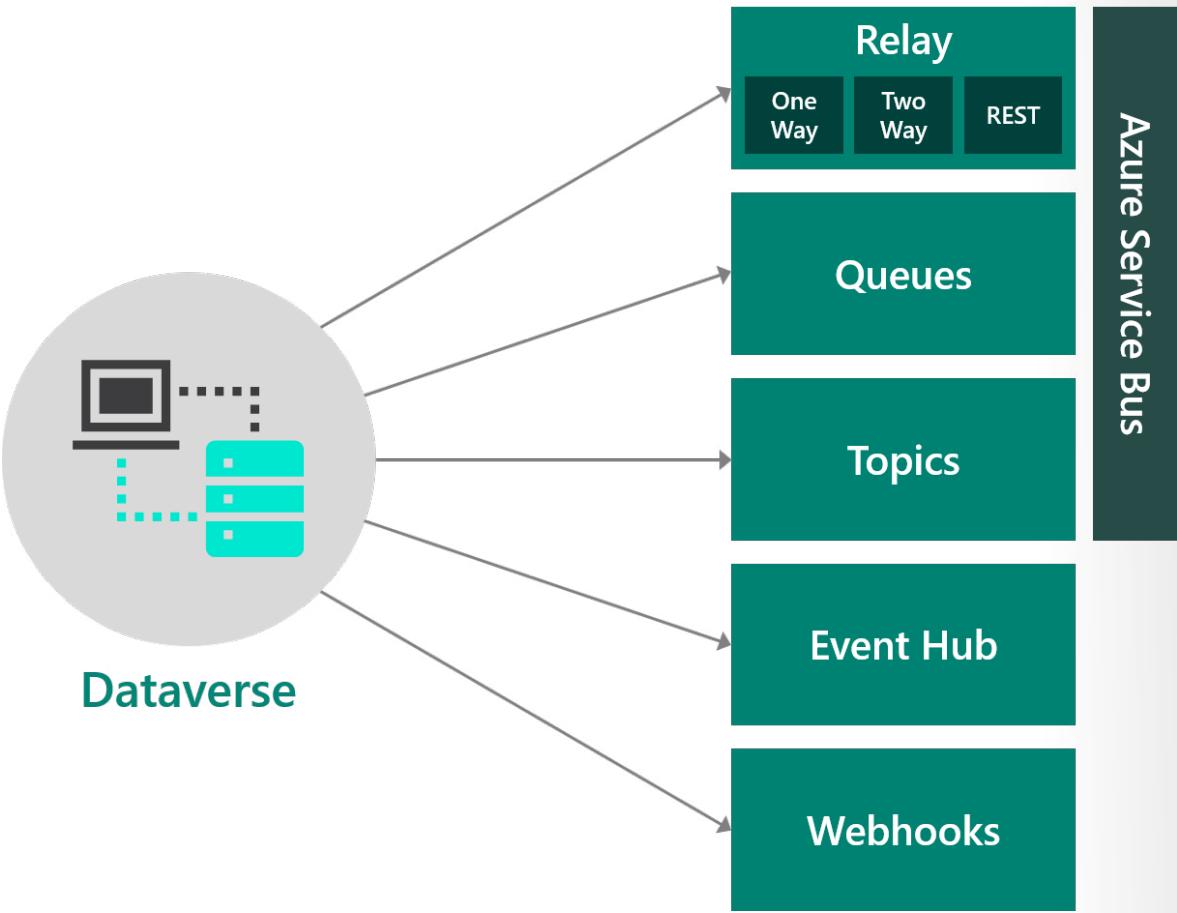
When retrieving data with change tracking, a set of delta changes is returned by Dataverse.

[!NOTE]

Change tracking must be enabled on the table.

## Azure integration

The Dataverse platform supports outbound integration with Azure. Dataverse can send messages to Azure services using the Event Framework as shown in the following diagram.



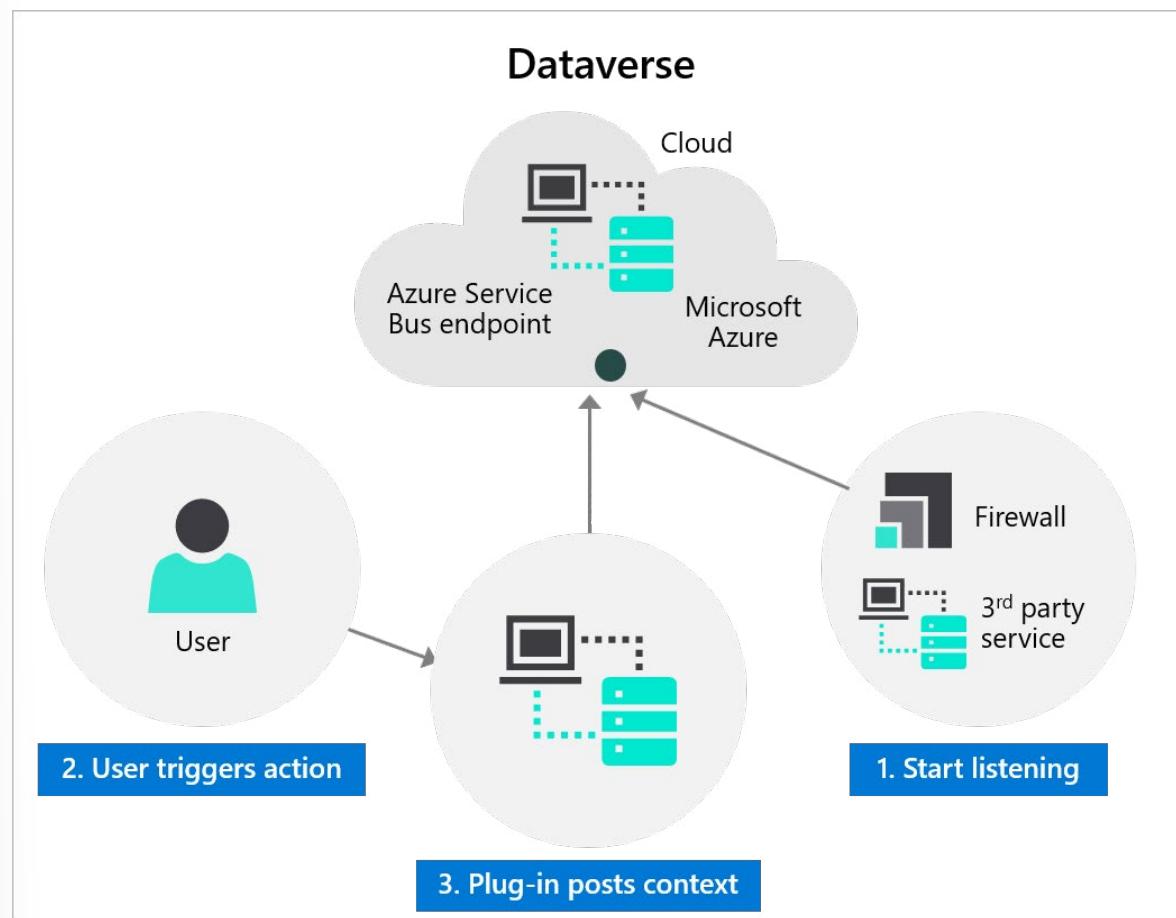
## Service Bus

One of these services that Dataverse can integrate with is the Azure Service Bus.

The Azure Service Bus is used to decouple applications and services from each other, providing the following benefits:

- Load-balancing work across competing workers.
- Safely routing and transferring data and control across service and application boundaries.
- Coordinating transactional work that requires a high-degree of reliability.

The Azure Service Bus can provide a secure and reliable communication channel between Dataverse runtime data and external cloud-based or on-premises line-of-business (LOB) applications as shown in the following diagram.



In this image, an Azure Service Bus resource has been created in Azure. A listener process has been developed that waits for messages to be posted onto an Azure Service Bus. In Dataverse, the Azure Service Bus endpoint has been registered using the Plugin Registration Tool, and a step defined that posts the plug-in context to the Service Bus as a message when an event occurs such as a row being created in Dataverse.

The listener process will automatically be able to read the message and extract the details of the event from the context (which table, ID of the row, user who triggered the event, the list of data changes, etc.) and perform the appropriate processing on this data.

The listener process can be:

- a C# program running on an on-premise system that polls the Service Bus for new messages.
- an Azure Logic App that is triggered automatically when a new message is posted.
- An Azure Function that is triggered automatically when a new message is posted.

The integration with Azure Service Bus is useful when there is the likelihood that the other system is not available or is limited in its ability to process high volumes of messages as the messages can be queued, allowing the receiving system to process the messages as fast as they can.

You can post a message to the Service Bus in one of two ways:

- No code: Create a step for the event in Dataverse against the Azure Service Bus endpoint. The plug-in execution context is posted to the Service Bus.

- Code: Create and register a plug-in in Dataverse that calls the Azure Service Bus endpoint. The message posted to the Service Bus can be customized.

Azure Service Bus can be used to:

- Build reliable and elastic cloud apps with messaging
- Protect your application from temporary peaks
- Distribute messages to multiple independent backend systems
- Decouple your applications from each other
- Ordered messaging scaled out to multiple readers

## Relay

Azure Relay is a service that used to be part of Azure Service Bus but has been separated into its own service.

The Azure Relay service facilitates tight integration between systems by helping you more securely expose services that reside within a corporate enterprise network to the public cloud. You can expose the services without opening a firewall connection, and without requiring intrusive changes to a corporate network infrastructure.

Azure Relay supports the following scenarios between on-premises services and applications running in the cloud or in another on-premises environment.

- Traditional one-way, request/response, and peer-to-peer communication
- Event distribution at internet-scope to enable publish/subscribe scenarios
- Bi-directional and unbuffered socket communication across network boundaries

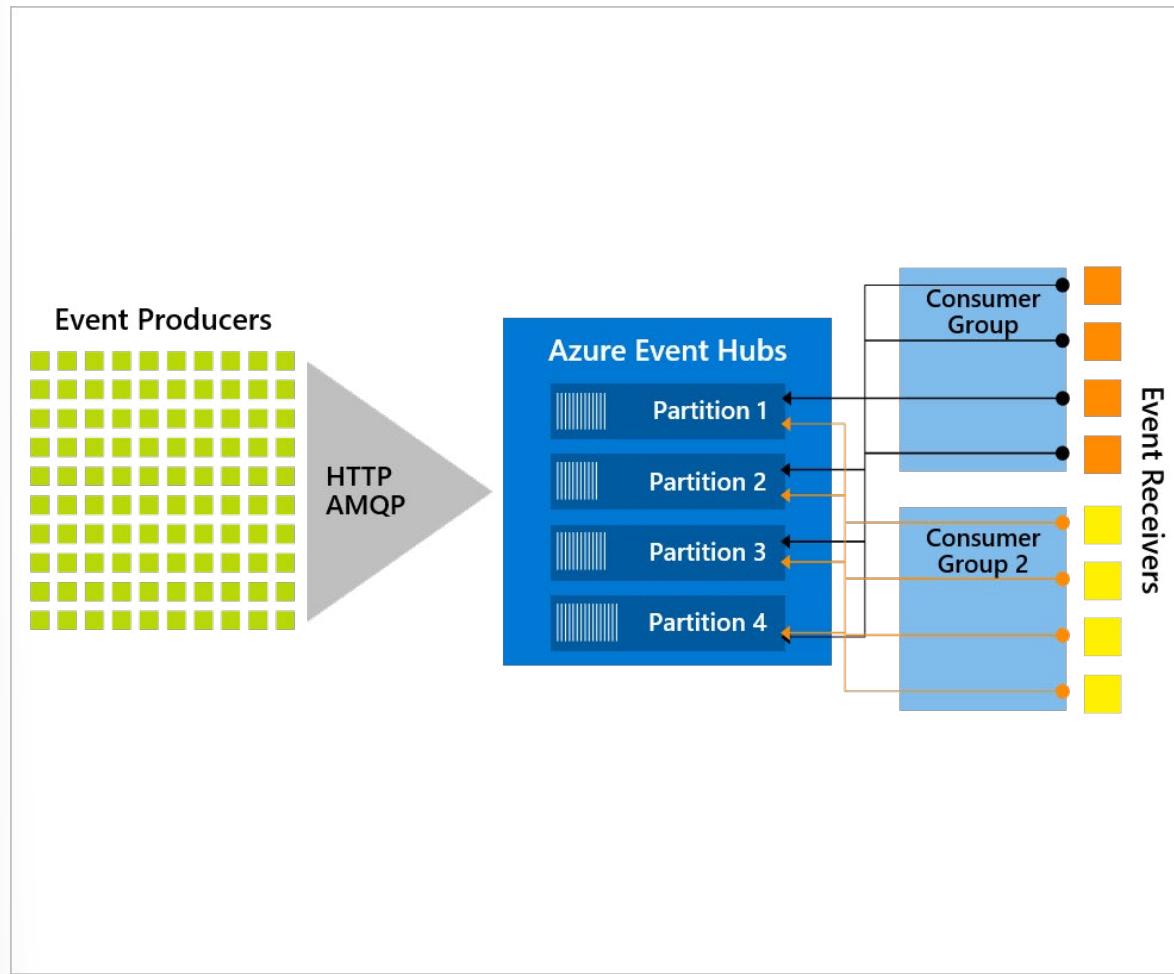
[!!IMPORTANT]

Azure Relay allows two systems to be connected without needing a direct connection. Azure Relay follows a request/response pattern so that the calling system can receive a response such as a piece of data or a success message from the other system.

## Event Hub

Azure Event Hub is a big data streaming platform and event ingestion service. It can receive and process millions of events per second. Data sent to an event hub can be transformed and stored by using any real-time analytics provider or batching/storage adapters.

Multiple other systems can subscribe to events processed by the Event Hub. The Event Hub can filter events using Consumer Groups so that systems only receive the events that are relevant to them at their own pace.



[!NOTE]

Topics in the Azure Service Bus service provide a similar method for systems to subscribe to filtered messages.

You should consider using the Azure Event Hub if you require multiple subscribers.

An example of using Event Hub is to publish events to stream analytics that in turn populates a Power BI dataset for visualization.

## Webhooks and Azure Functions

Dataverse supports the calling of a Webhook with the Event Framework. Webhooks are registered with the Plugin Registration Tool and can be triggered by a specified event in a step.

Webhooks is a lightweight HTTP pattern for connecting Web APIs and services with a publish/subscribe model. Webhook senders notify receivers about events by making requests to receiver endpoints with some information about the events. Webhooks are simply a pattern that can be applied using a wide range of technologies. There are no required frameworks, platforms, or programming languages you must use.

Webhooks enable developers to integrate Dataverse data with their own custom code hosted on external services. By using the WebHook model, you can secure your endpoint by using authentication header or

---

query string parameter keys. This is simpler than the SAS authentication model used by Azure Service Bus integration.

Azure Functions provides an excellent way to deliver a solution using WebHooks.

When deciding between the WebHook model and the Azure Service Bus integration, here are some items to keep in mind:

- Azure Service Bus works for high scale processing, and provides a full queueing mechanism if Data-verse is pushing many events.
- Webhooks can only scale to the point at which your hosted web service can handle the messages.
- Webhooks enables synchronous and asynchronous steps. Azure Service Bus only allows for asynchronous steps.
- Webhooks send POST requests with JSON payload and can be consumed by any programming language or web application hosted anywhere.
- Both WebHooks and Azure Service Bus can be invoked from a plug-in or custom workflow activity.

## **Process Integration: Power Automate vs Logic Apps**

Power Automate cloud flows are based on Azure Logic apps and can generally both can meet the same requirements. However, there are some differences. As a solution architect you should consider when to use Power Automate cloud flows or Logic Apps:

Power Automate:

- Dataverse Connector has more capability.
- Be packaged as part of a solution.
- Can perform RPA with desktop flows.
- Use Approvals connector.
- Send Notifications connector.
- Have a limit on the number of flow runs per month.

Logic Apps:

- Perform Enterprise Integration including EDI.
- Have higher performance.
- Can be more easily monitored using Azure tools.
- Have better error handling.
- Cannot be packaged in solutions.
- Consumption-based or fixed pricing model via an Azure subscription.

# Module Summary

## Recap

Integrations help build a more complete solution that provides a better user experience and better value than if the solutions were used individually.

A solution architect must balance the integration techniques used to ensure a good user experience and availability of the solution.

A solution architect must ensure integrations do not make the solution less stable, do not introduce points of failure, and actually add value.

## Next steps

For more information, see the following:

- **Data integration<sup>2</sup>**
- **Integration with Azure<sup>3</sup>**
- **Integrate with Microsoft Power Platform and Dataverse learning path<sup>4</sup>.**

---

<sup>2</sup> <https://docs.microsoft.com/powerapps/developer/data-platform/data-synchronization>

<sup>3</sup> <https://docs.microsoft.com/powerapps/developer/data-platform/azure-integration>

<sup>4</sup> <https://docs.microsoft.com/learn/patterns/integrate-power-platform>

## Module 12 Dynamics 365 applications architecture

### Explore Dynamics 365 application architecture

#### Dynamics 365 app specific architecture

Dynamics 365's customer engagement applications are customer focused business first-party applications that are built on the Microsoft Power Platform. They are designed to help organizations more effectively engage their customers. The sales and marketing applications help identify sales prospects and turn them into long lasting relationships. The service focused applications help your organization service your customers, whether you have a call center, service center, or are providing service in the field. Dynamics 365 customer engagement applications ensure that you have the tools necessary to fully engage your customers across all aspects of your business.

This module is all about choosing the right Dynamics 365 app and any particular issues with deploying the apps.

#### Dynamics 365 customer engagement apps

The apps that will be considered in this module are:

- Dynamics 365 Sales
- Dynamics 365 Marketing
- Dynamics 365 Customer Service
- Dynamics 365 Field Service
- Dynamics 365 Project Operations
- Dynamics 365 Customer Voice
- Dynamics 365 Customer Insights

These Dynamics 365 apps use Microsoft Dataverse and share many features.

## Common features of the Dynamics 365 customer engagement apps

The Dynamics 365 Dynamics 365 customer engagement apps are model-driven apps that run on Microsoft Dataverse. These apps inherit the capabilities described in the other modules for the Microsoft Power Platform and Power Apps. They have additional features specific for that business domain. The individual differences will be the focus of this module.

<sup>^</sup> Customer Voice and Customer Insights have their own user interface.

Below are features common to the Dynamics 365 customer engagement apps:

- Customers: Accounts and Contacts to represent organizations and people.
- Activities: Present the interactions user shave with customers.
- Searching for data.
- Data management including the import and export of data.
- Reporting including views, charts, and dashboards.
- Integration with Outlook for email.
- Mobile app

Dynamics 365 apps focus on collaboration rather than lots of features:

- Office 365 Tools: Take advantage of embedded tools like Excel and Word that sellers use every day to help build details, analyze deal details, and communicate with customers.
- SharePoint and OneDrive integration: Take advantage of the document storage and versioning capabilities of SharePoint and OneDrive for Business to help with document collaboration. The capabilities let you see who is working on and editing documents. It also gives you deeper insights into what documents customers are accessing when you use them to store Share attachments.
- Microsoft Teams integration: Engage subject matter experts (SMEs) and collaborate with colleagues across business functions and geographies. Sellers can access data directly from within a Team workspace, providing access to all of Microsoft Teams collaboration abilities. Gain an understanding of customer engagement and manage marketing at every touchpoint social, mobile, chatbots, web, or catalogs.

## Solution Architect's role in identifying apps and gaps

Using Dynamics 365 apps can reduce the amount of development required for your solution. The solution architect should consider if Dynamics 365 apps can be used to meet the requirements and should consider the mix of Dynamics 365 apps to use.

The solution architect should must lead the identification of gaps in the built-in capabilities and lead the design of how these gaps will be closed. Each Dynamics 365 app has specific functionality and may have different processes. This requires an overview of the capabilities of all of the first-party apps.

The solution architect must consider the architecture for first-party apps. In particular, how environments are used and how config data is created and propagated through the application lifecycle.

The solution architect must ensure processes are in place to handle ALM of customizations and configurations that support the apps and ensure data migration plans incorporate bringing in existing data and accommodate for any app constraints in the data model.

Finally, the solution architect must ensure integrations do not make the solution less stable.

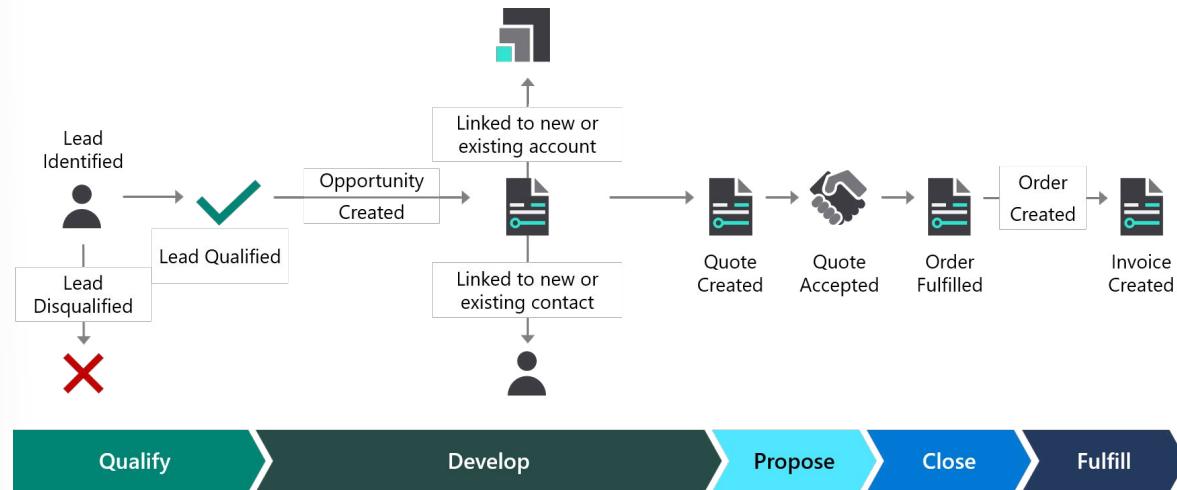
# Dynamics 365 Sales

Dynamics 365 Sales is an app that helps salespeople manage relationships with accounts and contacts, tracks sales activity, and helps users close opportunities faster. Dynamics 365 Sales manages the lead to opportunity to quote, order and invoice process.

## Overview of functionality

Dynamics 365 Sales has a wealth of functionality to support the seller and selling process:

- Pipeline management: Building a healthy pipeline can be one of the most difficult and time-consuming tasks that sellers and managers do. Sales pipelines affect everything from manufacturing schedules to staffing decisions. You can use Dynamics 365 Sales built-in features to support sales functions, such as qualifying leads and managing sales opportunities, to build a healthy sales pipeline. You can configure its features to support your sales organization's unique and changing needs easily.
- Product catalog: Build sales quotes by adding products directly from a dedicated product catalog that can function stand alone or be integrated with enterprise resource planning (ERP) applications, such as Dynamics 365 Finance or Dynamics 365 Supply Chain Management. Automated pricing options can update pricing details automatically.
- Product families: Extend the product catalog functionality with a central area to define common properties such as sizing and coloring options for all products in a product family. You can link products together to provide greater support in common sales scenarios such as cross selling, product substitution, or upselling.
- Customized pricing and discounting: Attach the most appropriate pricing based on who the customer is, current promotions, or the sales scenario with multiple price lists.
- Quote management: Generate quotes directly from details stored in a sales opportunity. You can manage the entire sales negotiation process with easy quote revisions, and quote delivery using familiar formats such as Word documents or PDFs.
- Sales orders: Create sales orders directly from quotes when deals move forward. With sales automation, you can open or close related supply chain items as the attached quote record is closed.
- Invoices: Create invoices from fulfilled sales orders. Dynamics 365 tracks the status of the invoice when the customer pays it.
- Sales playbooks: Apply repeatable proven winning sales techniques and contextual reference materials based on details defined in a sales opportunity such as the size of the deal, estimated close date, and more.



There are two versions of Sales: Sales Enterprise and Sales Professional. Sales Professional is aimed at smaller businesses and is missing some of the functionality of Enterprise.

## Implementation considerations

When implementing Dynamics 365 Sales, the solution architect needs to perform gap analysis on the following processes:

- Lead to Opportunity
- Product Catalog
- Line of business differences (E.g. wholesale vs. retail)
- Accounting and ERP integration

The product catalog (products, price lists, unit groups, and discount lists) must be configured if populating items on Opportunities, Quotes, Orders, or Invoices. Larger product catalogs could use third party tools for CPQ (Configure, Price, Quote) configuration.

Product catalog configuration and data is not included in solutions. The Configuration Migration tool can help mitigate some of the work when moving from development to test to production.

Integrating Sales with ERP systems can be complex mainly due to account and product catalog structures. The solution architect should look at the following issues when integrating with an ERP system:

- Which fields need to be added?
- What does the integration look like?

## LinkedIn Sales Navigator

Microsoft is leveraging its investment in LinkedIn with a separate app, Microsoft Relationship Sales. This is a combination of Dynamics 365 Sales Enterprise and LinkedIn Sales Navigator.

LinkedIn Sales Navigator has many features and capabilities including:

- Surfacing LinkedIn information on existing leads and accounts in Dynamics 365 Sales through the Sales Navigator widgets.
- Synchronizing between Sales Navigator and Dynamics 365 Sales daily, so that leads, contacts, and accounts present in Dynamics 365 are visible within LinkedIn Sales Navigator.

- Writing select Sales Navigator activities to Dynamics 365 Sales.

There are several issues that the solution architect need to be aware of:

- Integration features must be enabled from the Sales Hub Settings app.
- The sync to Dynamics 365 Sales is configured in LinkedIn Sales Navigator.
- Only can sync with one environment at a time.
- Only Accounts & Contacts that already exist in Dataverse will be synced.
- Records that only exist in Sales Navigator will not be pushed into Dataverse.
- The Navigator widgets added to forms are not Dynamics 365 aware and each user must sign and link their user to their LinkedIn account.

[!NOTE]

LinkedIn Sales Navigator does not permit system or test accounts which can make development and testing challenging.

## Dynamics 365 Sales Insights

Dynamics 365 Sales Insights provides AI powered insights on sales data. Sales Insights helps organizations in the following four areas:

- Guided selling: Guide sellers on their next course of action with timely and actionable insights.
- Productivity intelligence: Free up sellers' time by minimizing manual data entry with contextual, real-time suggestions for updating existing records and creating new records, making sure your data is always up to date.
- Connection insights: Build stronger relationships with customers and move relationships forward with AI-guided selling. Take preemptive steps to mitigate risks with a relationship health score.
- Predictive models: Increase conversion and win rates by using AI to prioritize leads and opportunities with the highest likelihood to convert and buy.

When implementing Dynamics 365 Sales Insights, the solution architect needs to be aware that:

- Sales Insights must be enabled per environment.

- Relationship Health is calculated based data in current environment only. Having large amounts of historical data is critical and activity Records have the biggest impact. It can take 24 hours before results will start to be visible.
- Models for Record Scoring are based on data in current environment only. Models are not part of the solution and must be configured in each environment separately.
- Who knows whom uses the Microsoft Graph and must be turned on for each environment separately.

## Dynamics 365 Marketing

Dynamics 365 Marketing is an app for organizations that require B2B marketing. Dynamics 365 Marketing allows you to create marketing forms and pages, email campaigns, and through segmentation to target contacts. It also enables you to run nurture campaigns, using customer journeys, to drive leads to Sales. Dynamics 365 Marketing includes functionality for running events and webinars.

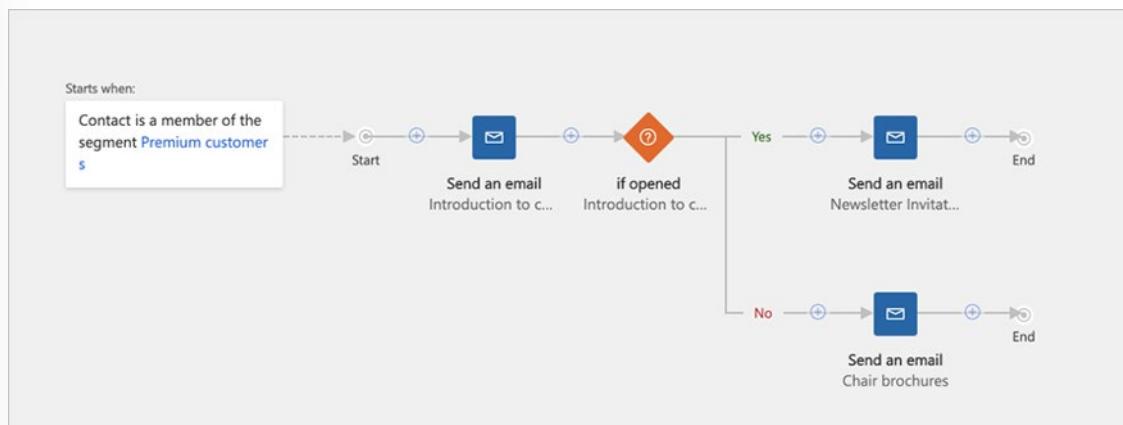
### Overview of functionality

Dynamics 365 Marketing has functionality to support marketing departments and coordinate with sales:

- Core marketing: Provides email marketing, customer journeys, behavior tracking, lead scoring, marketing pages, and more. These features form the core of the Marketing app and are unique to Dynamics 365 Marketing.
- Event management: Organize and promote in-person or webinar events, which include an online portal that attendees can use to review the event schedule and speakers, and to register for the event. This feature is currently available only with Dynamics 365 Marketing.
- Dynamics 365 Customer Voice: Create online surveys and analyze the results.
- Dynamics 365 Connector for LinkedIn Lead Gen Forms: Import leads generated with LinkedIn's lead tools into Dynamics 365.

All these features are fully integrated and work together to create a comprehensive marketing solution.

In Dynamics 365 Marketing segments can be created to represent contacts with similar characteristics. Dynamics 365 Marketing can send these potential buyers targeted marketing emails. As buyers open and interact with those emails, links to specific marketing content will provide them with more targeted information based on what they have clicked on and access. This content is provided automatically through customer journeys. As the buyer interacts further, their behavior is scored until they are classified as being sales ready.



## [!NOTE]

Dynamics 365 Marketing is primarily aimed at B2B marketing and is focussed on generating leads for sales. A lot of investment is being made by Microsoft in Dynamics 365 Marketing but if you require enterprise marketing functionality or B2C marketing then the solution architect should consider Microsoft's partner, Adobe Marketing Cloud.

## Implementation considerations

Dynamics 365 Marketing typically requires little customization. Customization should generally be avoided due to the complexities of the processing in the application. Apart from initial setup, most activities are undertaken by the users of the application and the solution architect's role will be focussed on guiding users to use the correct components of the application in the right way.

When implementing Dynamics 365 Marketing, the solution architect needs to perform gap analysis on the following four processes:

- Segmentation
- Customer journeys
- Subscriptions, Double Opt-in, and Unsubscribes
- Event management

When implementing Dynamics 365 Marketing, the solution architect needs to consider the following key factors:

- Project scope: Which components of marketing are used. Which webinar provider will be used. Is LinkedIn Campaign manager used.
- Forms: How are forms to be used; natively, embedded within existing websites, or using capture of existing forms.
- Quotas: Dynamics Marketing is licensed on the basis of the number of contacts that been involved with marketing activities in the previous 12 months.
- Deployment Support: Marketing is a complex application that requires training for all users in the marketing department.

## Deployment considerations

Dynamics 365 uses Azure to capture interactions. This is a Microsoft subscription that you do not have access to. The interaction data is synced into Marketing so that you can gain insights and use interactions to created segments and trigger customer journeys. This link to an Azure subscription makes it difficult to copy environments.

Dynamics 365 Marketing has a different licensing model than other Dynamics 365 applications. Some important aspects of Marketing's licensing model are:

- Integration with Customer Voice must be enabled for each environment.
- Using Microsoft Teams for webinars must be enabled for each environment.
- You do not purchase user licenses of Marketing. You purchase the Marketing app and bundles of marketing contacts.
- One Marketing app license permits deployment on only one Dynamics 365 environment. The environment could be a sandbox or production. Deploying on multiple Dynamics 365 environments requires multiple Marketing app licenses.

- Marketing has different pricing for production and non-production usage. A sandbox-only license entitles you to set up a test environment for trying out new features or customizations. Sandboxes include both solutions and services, and are fully functional, but have much stricter quotas and can't be used to run production campaigns.

If you do not want to purchase a license for development, Marketing offers a solution-only license. The solution-only license does not support Marketing processes such as segmentation and email sending. It does support, however, Marketing metadata such as marketing entities that you can use for extensibility purposes. The solution-only license also allows you to enable ALM operations across your environments.

Settings, emails, forms, and other marketing collateral is captured as data. For this reason it will not be included in solutions. The Configuration Migration tool can help mitigate some of the work when moving from development to test to production.

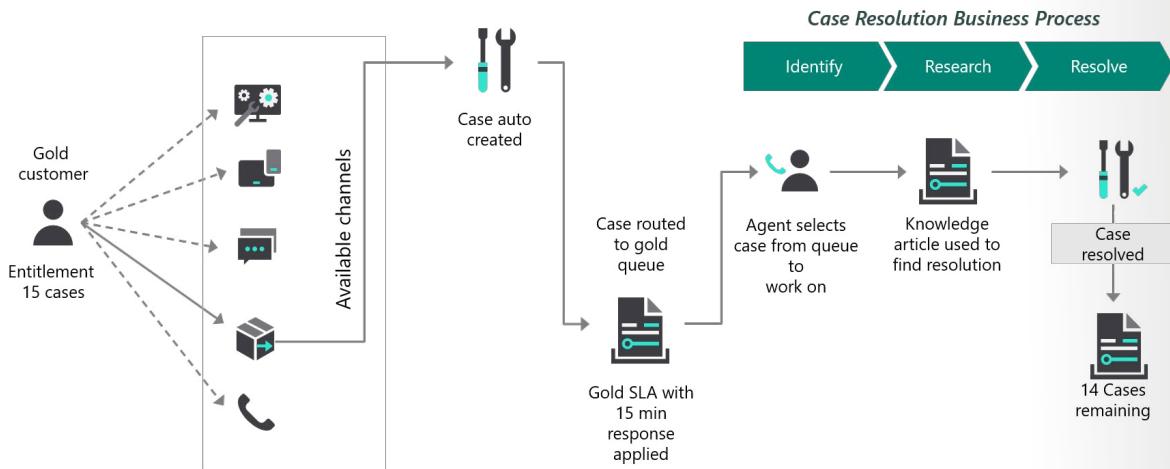
## Dynamics 365 Customer Service

Dynamics 365 Customer Service is an app that helps organizations manage relationships with accounts and contacts by capturing, tracking, and resolving service issues (cases) that customers have. Dynamics 365 Customer Service provides functionality around Cases e.g., entitlements, SLAs, queues, knowledge base, and customer satisfaction.

### Overview of functionality

Dynamics 365 Customer Service has functionality to support the customer service agent in resolving cases:

- Cases: Cases are items that you need to resolve. Cases can include questions, problems, or anything else a customer may ask about.
- Queues: Queues are containers for similar cases that are waiting on a response from an agent. You can organize cases by similar topics, levels of difficulties, or other custom rules.
- Entitlements: Entitlements are numeric- or time-based contract records you use to manage how products are serviced, who from a customer account can open cases, and the channels through which you receive a case.
- Service-level agreements: Service-level agreements define the level of service or support promised to a customer. They include the KPIs that you need to reach to meet that service level.
- Knowledge article: Knowledge articles are support articles available either internally or externally that customers or agents can reference to resolve a problem.
- Routing rules: Routing rules are criteria that decide which queue a case belongs in.
- Automatic Record Creation rules: The automatic record creation and update rules primarily help administrators define rules to create any entity record from a set of available activities. For example, creating cases from the emails sent by customers.



## Implementation considerations

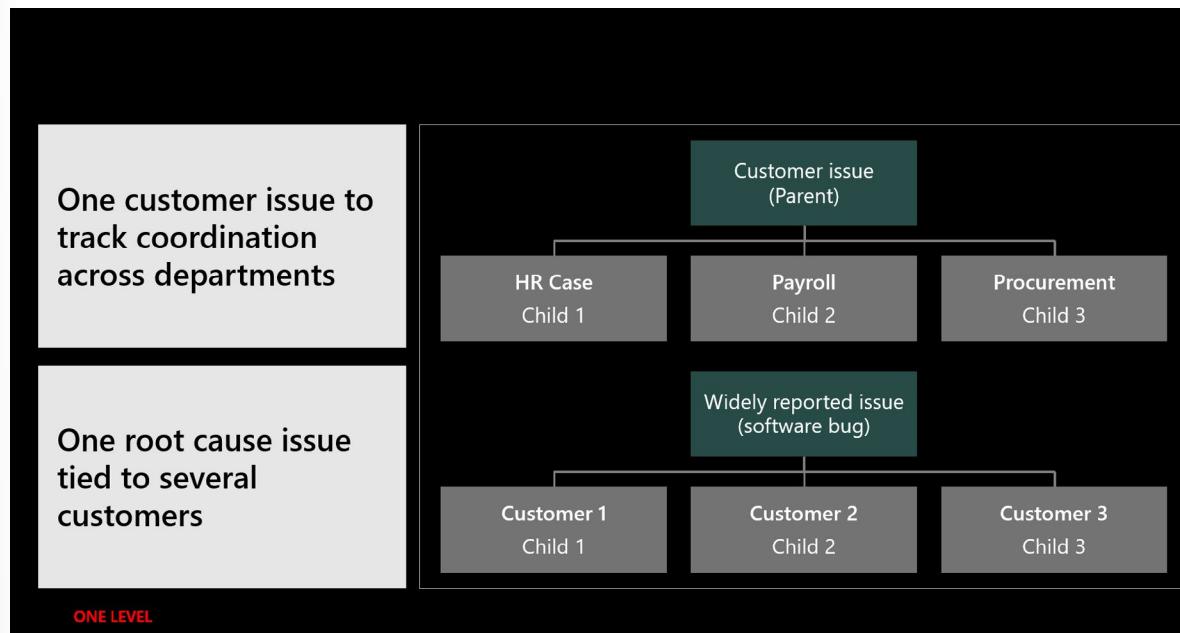
When implementing Dynamics 365 Customer Service, the solution architect needs to perform gap analysis on the following processes:

- Automatic record creation: Use the record creation and updates rule or use Power Automate.
- Customer Service Hub: The standard Account, Contact, and Case forms used by other apps are not used in Customer Service Hub. Each form has an Interactive Experience version used by Customer Service Hub app. Plan Accordingly when customizing and deploying.
- Knowledge Articles are not solution components.
- SLA items used Power Automate to perform the actions.
- Be aware of SLA behavior on import of historic case data.

Dynamics 365 Customer Service has special processing for parent and child cases. The case hierarchy supports two scenarios:

- When a parent case is resolved, all child cases are resolved.  
or
- The parent case cannot be resolved until all child cases are resolved.

The scenarios are outlined in the following diagram.



[!NOTE]

A parent case can have no more than 100 child cases attached to it.

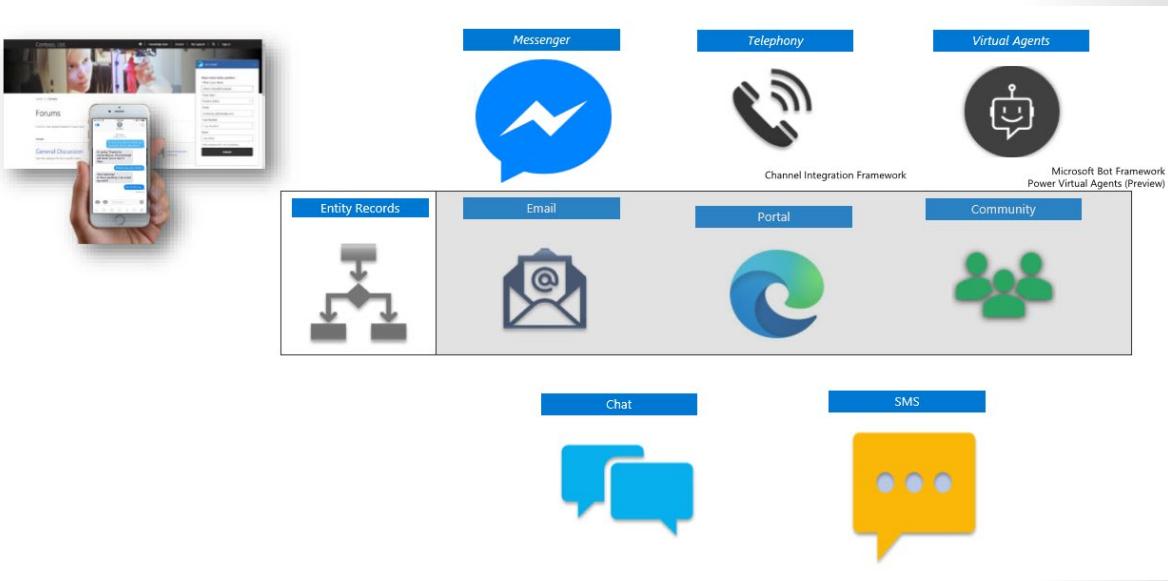
The case hierarchy only supports one-level in the hierarchy. If you need to have multiple levels or require different processing you will need to create a self-referencing relationship on the case table and define this relationship as hierarchical. You will then need to add custom code or Power Automate flows to handle the cascade behavior for case resolution.

## Omnichannel

Service focused organizations understand the need to adjust their support strategy to provide customers with the best support options based on their needs. Customers want to engage with service organizations on their terms. They want to contact you when and how they feel most comfortable. Customers also want to know that when they do contact you, you will have the information required to solve their issue.

One way that you can help customers is to offer various support channels for your customers. You can provide support via phone, email, live chat, virtual agents, social media, forums, and self-service knowledge bases. We call this range of options multi-channel support. When you give your customers multiple support options, you make it easier for them to start support requests. The increased request volume can be a challenge. It is a challenge to give customers a comprehensive and equally satisfying experience across any channel they are using.

The following graphic illustrates the various channels through which agents can communicate with their customers.



Omnichannel for Dynamics 365 Customer Service provides the following capabilities:

- **Session management:** You can work with multiple customer sessions simultaneously. You can also switch between sessions without losing the context of the conversation or any customer details.
- **Customer interaction:** You can engage with customers through the channels your customers prefer, offering a personalized experience. Tools such as real-time sentiment analysis, quick replies, and knowledge base access help you understand your customers' needs, and help to resolving issues faster.
- **Customer context:** You have a single contextual view of the customer called the customer summary. The customer summary screen helps you understand the context of the conversation by giving you the important details in a single location. The details include:
  - **Customer:** Lets you access account or contact details about the customer you are communicating with.
  - **Case:** Links your conversation to a new or existing case record to keep all relevant information together.
  - **Recent cases:** Helps you understand a customer's history with quick access to all recent cases that your organization helped the customer with.
  - **Timeline:** You can easily access the customer's historical activities like phone conversations, emails, and tasks. This combined view eliminates the need to open additional screens to search for this information.
- **Conversation summary:** Allows you to know your customers better by giving you additional contextual details that help service the customer quickly and effectively. These details might include answers to pre-chat survey questions, browser or machine information, or even details about what customers were doing before they started the conversation.
- **Access to familiar Dynamics 365 tools:** You can take advantage of existing Dynamics 365 functionality, such as record searching, quick record creation, and agent presence information.
- **Real-time notification:** You can route items to qualified agents who are available to help the customer. They will be notified of incoming communication from customers automatically.

- Proactive chat: You can engage customers while they are visiting customer portals by inviting them to chat conversations automatically, based on predefined rules that consider factors such as end-user data or time spent on a web page.
- Skill-based routing: Skill-based routing lets you route conversations to the best agent based on agent skills. This skill alignment improves automatic work distribution efficiency by ensuring the agent has the right skills to resolve a customer issue most effectively.
- Agent scripts: Agents can take advantage of pre-configured scripts for step-by-step guidance to resolve issues. You can configure and automate these steps based on session types to ensure adherence to the appropriate processes.

When implementing OmniChannel, the solution architect needs to be consider:

- OmniChannel requires Dynamics 365 Chat and or Digital Messaging Service.
- SMS provider licensing such as Telesign or Twilio is required.
- Social messaging providers will require external configuration and access to APIs from within Omnidchannel.
- It can take up to 15 minutes for widget changes to be reflected in portals.
- External applications that will be surfaced will require customer development and will need to be deployed using Omnichannel framework.
- Skills based routing must be enabled for each environment.

Omnichannel is very configuration heavy and is not a straightforward application to deploy. Several aspects of OmniChannel will need to be addressed:

- What queues will be required to support functionality.
- How many total channels the organization supports.
- How many items per channel typical agents can do at one time.
- When and how conversations are routed to agents.
- What information needs to be visible to an agent for conversations.
- What data can be transmitted in a conversation.
- The resolution process agents use for different scenarios.
- How Virtual Agents will be used.

The following components are not included in solutions and the solution architect will need a process to manage their deployment:

- Channels
- Work Streams
- Application Config Settings
- Agent Scripts
- Session & Notification Templates
- Quick Responses
- Presences
- Sentiment Models

The Configuration Migration tool can help mitigate some of the work when moving from development to test to production.

## Power Virtual Agents

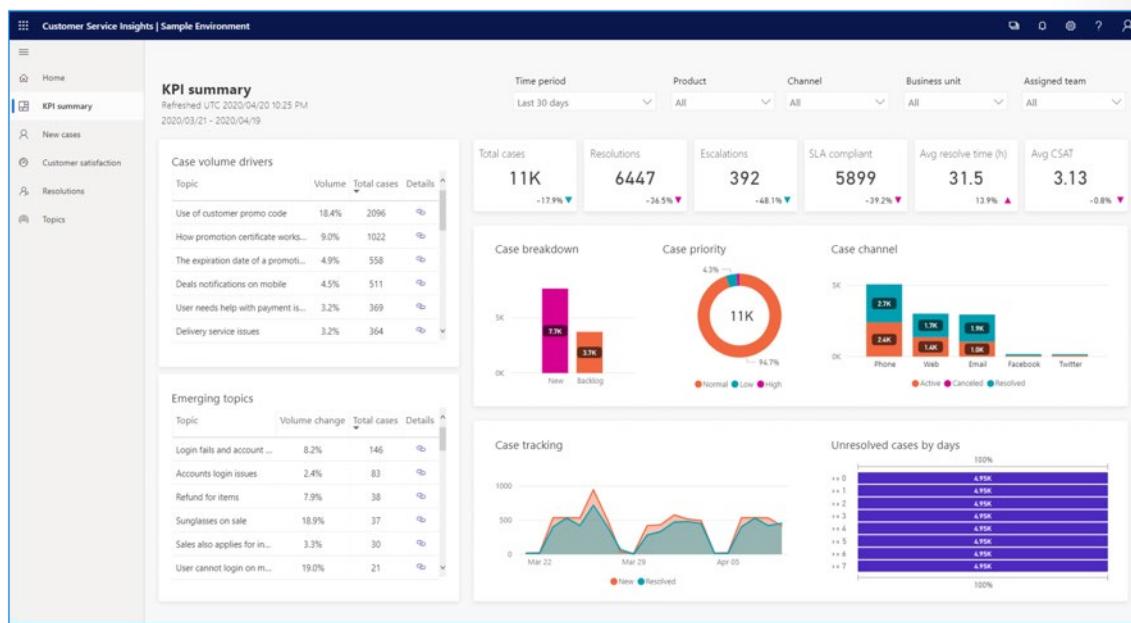
OmniChannel supports bots created with Power Virtual Agents. You can route customers to bots automatically, which can help you triage and handle specific types of customer inquiries. Through triggers, virtual agents will know when to redirect conversations to live agents automatically. When agents receive a conversation from a virtual agent, Dynamics 365 Customer Service includes all earlier communication with the bot.

Additionally, you can take advantage of bots in the user interface to give context-based recommendations to help agents in resolving issues faster.

## Dynamics 365 Customer Service Insights

Dynamics 365 Customer Service Insights provides AI powered insights on service data. Customer Service Insights contains several key components:

- Workspaces: Display service data for a specific customer service data environment. A single Customer Service Insights instance can contain multiple workspaces.
- Insights dashboards: Filterable dashboards that group together the graphical and AI-driven charts into a single space based on different KPIs.
- Topics: Group related support cases together by using AI for easier analysis. Topics are used throughout the different Customer Service Insights dashboards.



When implementing Dynamics 365 Customer Service Insights, the solution architect needs to be aware that:

- Customer Service Insights must be enabled per environment.
- Customer Service Insights reports use combination of BI analytics and AI insights.

- Data refresh occurs every 24 hours. You might see a warning icon beside the Last refresh (UTC) label in the upper-right corner of each report when there is a delay or issue in the data refresh.

## Dynamics 365 Field Service

Dynamics 365 Field Service is an app for organizations that deliver service at customer locations. The aim of Dynamics 365 Field Service is to ensure that the best resource, with the right skills, is at the right location, at the right time, with the right parts. Dynamics 365 Field Service provides functionality around the Work Order e.g., resources, scheduling, assets and inventory.

### Overview of functionality

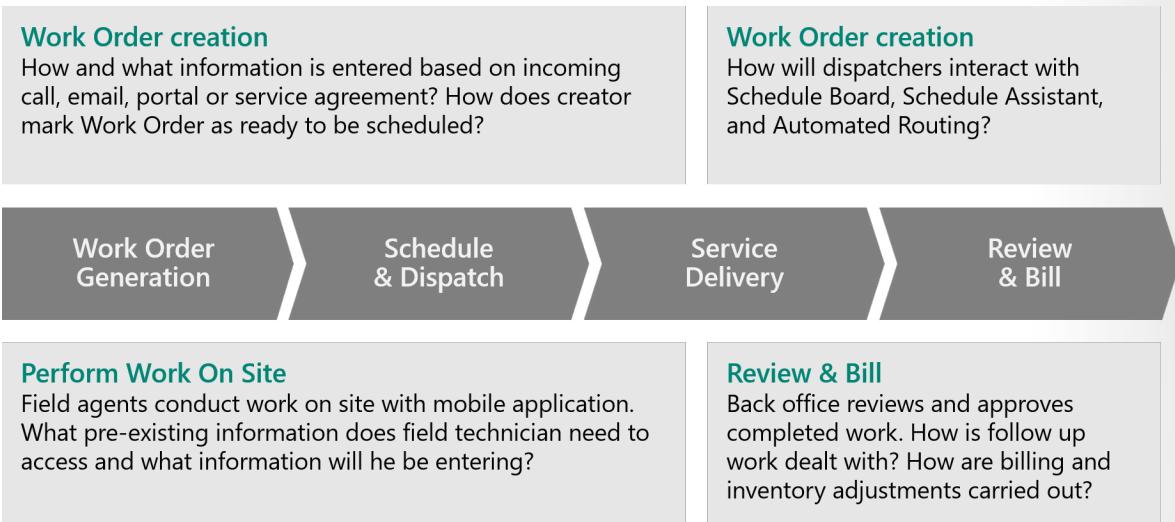
Dynamics 365 Field Service has the following key capabilities to support an organization in delivering service at remote customer sites:

- Work orders to define the service work needed primarily (but not exclusively) at customer locations.
- Scheduling and dispatch tools to manage resources and equipment needed for customer service, visualize onsite appointments, and optimize service schedules with efficient routing and resource skill matching.
- Communication tools to enhance collaboration between customer service agents, dispatchers, field technicians, customers, and other stakeholders.
- An easy-to-use mobile application that guides technicians through schedule changes and service work.
- Asset management capabilities to keep track of customer equipment and service history.
- Preventive maintenance by automatically generating recurring maintenance appointments for equipment.
- Inventory, purchasing, and returns capabilities to manage truck stock, purchase order requests and fulfillment, and product returns.
- Billing capabilities to generate invoices based on products and services delivered to customers.
- Time tracking to help you track how resources are spending their time, whether they're traveling, on break, or working.
- Analytics for reporting on key performance indicators for managing work orders, scheduling activities, and interacting with customers.

### Implementation considerations

Dynamics 365 Field Service is not like the other Dynamics 365 customer engagement apps. Deploying Field Service more like an ERP implementation than a CRM implementation and is very configuration heavy. The solution architect will need to plan how to transport these configurations across environments.

The solution architect needs to understand the Field Service process and address each of the stages in the process outlined in the following diagram.



When implementing Dynamics 365 Field Service, the solution architect needs to perform gap analysis on the following four main processes:

- Work order creation
- Schedule and dispatch
- On-site work with mobile application
- Review and completion of the work order

There are many configurable items that need to be setup before scheduling of work orders can commence. Some configurable items can have a significant impact on the functionality of Field Service.

When implementing Dynamics 365 Field Service, the solution architect needs to consider the following key factors:

- Project scope: Which features to include such as Agreements, Invoicing or Inventory management. How will be mobile app be deployed and how are messaging and notifications sent. What optimizations should be used.
- Project complexity: Field Service projects can be complex. How mature is the customer. What experience does the customer and project team have of Dynamics 365.
- Custom development: Custom development can be risky in Field Service. Are custom plug-ins required to support business processing such as Approvals. Are portals for customers and technicians required.
- Integration: Integrating with ERP and other third-party systems. Integrations can be complex with many integration points.
- Data conversion: Conversation of legacy data can be complex as the legacy data may not fit the requirements for Field Service. How are future scheduled work orders going to be migrated
- Deployment support: Field Service requires training not only of schedulers but all technicians. How is this training to be carried out? A deployment and go-live plan will be required. Should a pilot be included?

Field Service comes with multiple pre-configured workflows and plug-ins out of the box. It is important to be aware of what these do, and how they could impact customizations you make to the environment. Ensure they have not been disabled after new functionality is added.

## Deployment considerations

Most of the configuration data that will be used to support the bulk of the application functionality is captured as data. For this reason it will not be included in solutions. The following are not included in solutions and the solution architect will need a process to manage their deployment:

- Schedule Boards
- Settings
- Resource
- Characteristics
- Proficiency Models
- Products & Services
- Incident & Service Task Types
- Resources, Territories, Organization Units
- Sub-statuses
- Requirement Group Templates

<sup>^</sup> This list is not exhaustive.

The Configuration Migration tool can help mitigate some of the work when moving from development to test to production.

## Universal Resource Scheduling

Universal Resource Scheduling (URS) is a foundational component that Field Service solution is based on. Many of the items that need to be configured for Field Service are really URS components.

Features of URS are:

- Bookable Resources: the individual resource to be scheduled.
- Characteristics: The skills or certifications for the resource.
- Proficiency models: The level of skill of the resource.
- Categories: The role of the resource:
- Working Hours templates: The working hours for the resource:
- Booking statuses: Sub-statuses mapped to the organizations business processes.
- Fulfillment preferences: Controls how time slots are offered by the schedule assistant.
- Requirement Group templates: Controls how groups of resources are scheduled together to meet different requirements.
- Crews: Resource crews allow dispatchers to search and schedule multiple resources at once.
- Pools: Assemble groups of similar resources to manage capacity and give schedulers the option to assign specific resources at a later time.
- Facilities: For scenarios where a customer is expected to travel to the company's location.
- Schedule boards

[!NOTE]

URS is also used in Customer Service and Project Operations.

Resource Schedule Optimization (RSO) automatically schedules work orders to the resources that are most available and best qualified. Many field service organizations that perform work orders on site at customer locations benefit from automatic scheduling because it optimizes the routes and travel times of field technicians as they travel from work order to work order.

RSO uses algorithms developed by Microsoft and uses Azure to optimize the routes. RSO must be configured for each environment separately.

## Connected Field Service

Dynamics 365 Field Service combines monitoring and preventative maintenance with the IoT to replace your traditional break-fix service model. It is a proactive, never-fail model. You capture the information transmitted by your IoT devices and turn that information into actionable items that you execute from a Dynamics 365 instance.



Key benefits of Connected Field Service:

- Reduce downtime by connecting to IoT devices to diagnose problems before customers are aware of an issue.
- Address issues faster by remotely monitoring devices and keeping customers in the loop.
- Device data and service maintenance data help to make intelligent decisions around dispatching technicians with the right expertise, availability, and proximity to the job.

Connected Field Service uses the following services in Azure:

- Azure IoT Hub: Connected Field Service for Azure IoT Hub is an add-on solution that uses Azure IoT platform-as-a-service (PaaS) offering to connect to Dynamics365 Field Service. This option uses a number of Azure services including Azure Stream Analytics, Logic Apps, Azure Queues, and Azure storage.
- Azure IoT Central: An IoT software-as-a-service (SaaS) solution that uses Power Automate to communicate with Dynamics 365 Field Service.

## Dynamics 365 Customer Data Platform

Microsoft's Customer Data Platform (CDP) offering consists of two products:

- Customer Insights: Creates a customer profile from multiple data sources and unlocks insights to build a deeper understanding of customers.
- Customer Voice: Captures surveys to capture and analyze customer feedback.

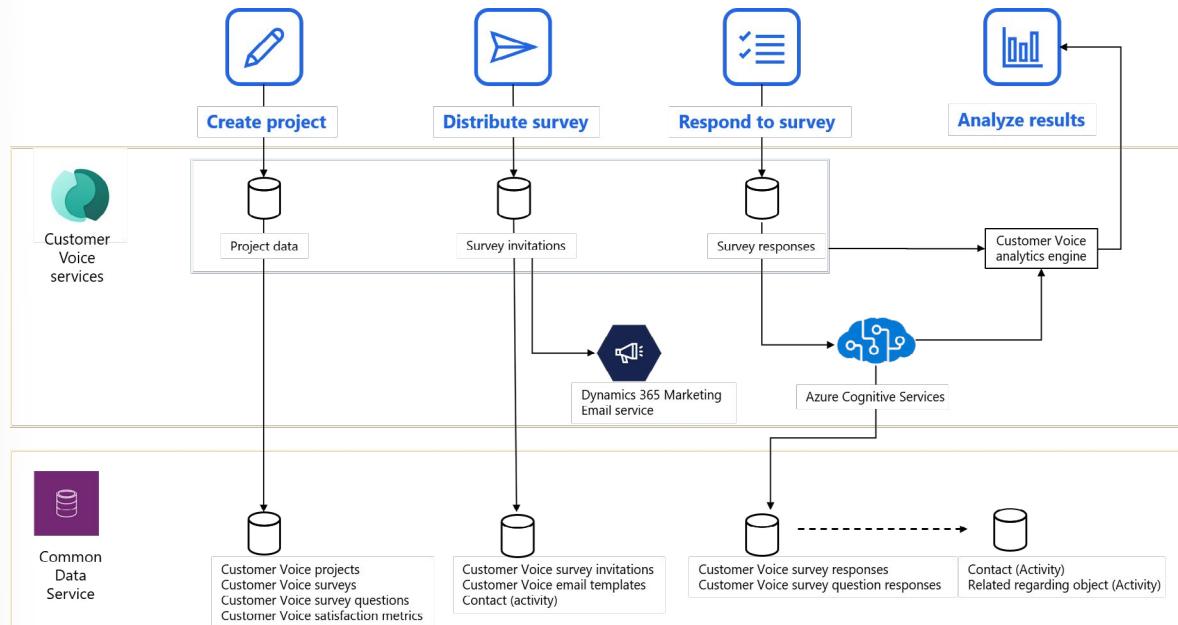
## Customer Voice

Customer Voice captures, analyzes, and acts on customer and employee feedback.

The key features of Customer Voice are:

- Projects: A container for related surveys. Each project is linked to a Dataverse environment.

- Surveys: A set of questions. A survey is distributed and survey invitations are created. Survey invitation emails are sent to the recipients by using Dynamics 365 Marketing email service internally.
- Analysis: When a respondent responds to a survey, the survey response is first stored in Customer Voice services and then sent to Azure Cognitive Services and the Customer Voice analytics engine for further processing. After responses are processed by Cognitive Services, they're stored in Dataverse.
- Reporting: Customer Voice has built-in dashboards and reports.



Customer Voice has its own user interface and is not a model-driven app. You can add the tables that Customer Voice uses to your own model-driven app, for instance to show responses such as NPS scores in your app and dashboards.

[!NOTE]

Customer Voice has its own data which is synced to Dataverse. If you delete any survey related data directly from Microsoft Dataverse, it is not synchronized with Customer Voice services. If you want to delete any data, you must delete using the Dynamics 365 Customer Voice interface.

As Customer Voice data is copied into Dataverse you can use Power Automate cloud flows. There is also a connector for Customer Voice that allows you to distribute surveys and process survey responses. The solution architect will need to determine how to distribute survey invitations and what processing is required on responses.

Customer Voice is bundled with Dynamics 365 Marketing but can be installed and used with any of the Dynamics 365 customer engagement first-party apps.

## Customer Insights

Dynamics 365 Customer Insights helps deliver personalized customer experiences. The platform's capabilities provide insights into who your customers are and how they engage with your platform. Customer Insights unifies customer data across multiple sources to get a single view of customers and has two components:

- Audience insights helps you transform your business into a customer-centric organization. Marketing, sales, and service professionals have the insights they need to personalize experiences. Connect data

from transactional, behavioral, and observational sources to create a 360-degree customer view. See results faster with a CDP designed to deliver insights that can be acted upon.

- Engagement insights enables you to understand interactively, how your customers are using your services and products – both individually and holistically – on websites, mobile apps, and connected products. Combine behavioral analytics with transactional, demographic, survey, and other data types from Microsoft Dynamics 365 Customer Insights. Maintain full control over your customer data to ensure the highest level of data governance and compliance.

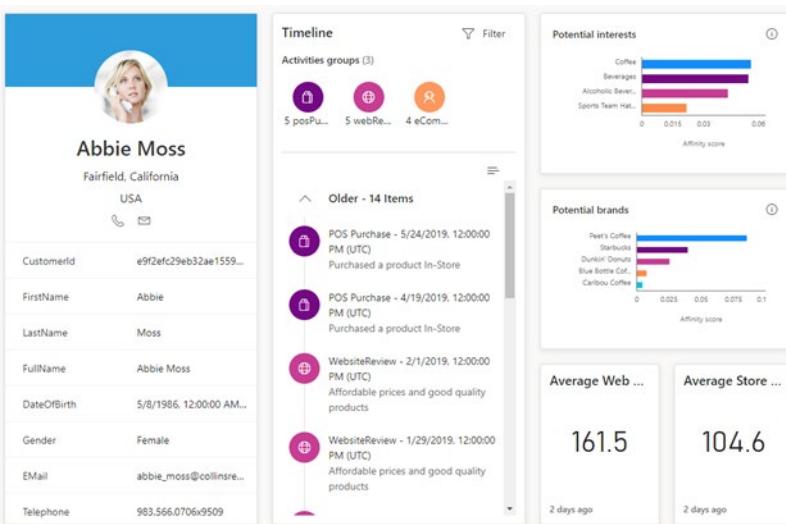
The key features of Customer Insights are:

- Ingest: Import data from multiple data sources including Microsoft Dataverse and Dynamics 365 apps. PowerQuery is used to import data.
- Unify: Data is mapped, matched, and merged to create a single customer profile.
- Enrich: Data can be enriched using Machine Learning models. Measures (KPIs) can be created based on attributes of the customer or their behavior.
- Segments: Segments of similar customers can be specified.
- Export: Customer Insights lets you export data such as measures, segments, and activities to other applications so the data can be leveraged as and used as part of the application functionality.

[!IMPORTANT]

Customer Insights integrates with Dynamics 365 Marketing. Segments created in Customer Insights can be used in Customer Journeys.

Customer Insights is a standalone subscription and has its own user interface; it is not a model-driven app.



The customer profiles in Customer Insights can be added to Dynamics 365 apps using a solution in AppSource solution. Visuals must be added to the forms and users must be added as Viewers in Customer Insights.

[!NOTE]

Contact records must be present in Dynamics 365 Sales before you can export a segment from Customer Insights to Sales.

There is a connector for Power Automate to interact with Customer Insights.

## Evaluate requirements for first-party apps

### Exercise

#### Evaluate requirements for first-party apps

Review the requirements for each app. Decide how best to address a requirement. Some items might be out of the box functionality, some might have nothing to do with our first-party applications, some might need third-party apps, and more.

#### Dynamics 365 Sales

- While viewing a contact record, sales managers need to see detailed information about the parent account record of that contact.
- Preferred pricing is only valid for a period of time and reverts back upon agreement expiration.
- Sales reps need to have the ability to configure products (color, size, components, etc.) and track them to an opportunity. We currently use a custom-built product configurator. It was built by a developer that is still on staff. It's a WPF application built using .NET.
- Some users expect the opportunity form to show and hide columns based on their column choices on the opportunity form.
- Display real-time information of an account's top current events on the account form.
- When a user creates a new opportunity valued at greater than \$1 million, change the owner of the record to the user's team. Notify the original record owner of the change.
- You must display the current stage of the lead in the lead grid.

#### Dynamics 365 Customer Service

- When a customer calls in for a new support request, if they do not have available pre-purchased support tickets available, support agents need to be able to sell them support tickets without leaving the case record they are currently viewing.
- When a preferred customer is out of available support incidences we need to reach out and offer to sell them more.
- Users must not access the application on mobile devices.
- Once a support request is opened for an on-site service, we need to dispatch to an available qualified field resource.
- The top support agent (the one with the highest number of closed case records) has created a personal chart to show their open cases from the beginning of the week. You need to make this available to all support agents to see their own support cases from the beginning of the week.
- You must display the current stage of the case in the case grid.
- Some users expect the case form to show and hide sections based on their column choices on the opportunity form.

## Dynamics 365 Field Service

- We have implemented a new training program and need resources' skills automatically updated when they pass a class.
- All screens must render for the user in less than 5.25 seconds.
- Resources must be certified prior to completing field tickets without a supervisor.
- Each work order requires a work order type.
- Dispatchers perform 70% of the booking, technicians can perform self-booking via mobile.
- Many customers do not accurately represent what is actually the problem that requires service. We need to make better use of our technicians' time and only send them on a call when it's needed.

# Module Summary

## Summary

The solution architect should determine which Dynamics 365 apps to use in the solution, identify gaps and resolve those gaps. The solution architect need to decide whether to extend or integrate the apps.

## Combining Dynamics 365 Customer Engagement Apps

Each Dynamics 365 app is independent of the others however data might be shared between apps such as accounts, contacts, addresses, activities, etc. Table columns or functionality added by an app will be visible in all apps that include that table. For instance Field Service adds tabs, and columns to the account form. The solution architect needs to consider leveraging different forms per app if functionality should not be shown in another app .

[!NOTE]

The Customer Service Hub uses different main forms by default.

## Using Dynamics 365

Deploying Dynamics 365 apps is the fastest path to go live only requiring tailoring for gaps. The solution architect should work with app functional consultants to identify and design how gaps will be closed.

In this module, you looked at the Dynamics 365 customer engagement apps:

- Dynamics 365 Sales
- Dynamics 365 Marketing
- Dynamics 365 Customer Service
- Dynamics 365 Field Service
- Dynamics 365 Project Operations
- Dynamics 365 Customer Voice
- Dynamics 365 Customer Insights

For each of these apps you learn about:

- Their individual architectures
- How environments are used
- Key processes to analyse
- Implementation considerations
- How config data is created and propagated through the application lifecycle.

## Next steps

The next steps are to look at integrations with other system and Microsoft Azure.

## Module 13 Power Virtual Agents architecture

### Explore Power Virtual Agents architecture

#### Introduction to Power Virtual Agents

We now use multiple communications channels such as email, webchat, telephone, and platforms such as Slack, Teams, and Facebook. Organizations need to be able to contactable by their customers over these multiple channels and they need to respond consistently no matter which channel a customer decides to use.

Users and customers are becoming more demanding, they require 24x7 response, responses that are tailored to their behaviors and characteristics, and to respond to more and more complex requests.

Conversational AI is the use of AI powered agents, known as chatbots, to respond to human questions and requests in an intelligent way. Organizations use chatbots to provide the first line of response to customers, handling common enquiries across multiple channels. Chatbots instigate a conversational dialog with the human whether that is over the web or another channel such as the SMS text service.

#### Issues around building chatbots

Organizations are going through complex challenges dealing with both employees and customers. Here are some statistics:

- 66% of customers try to use self-service first before contacting an agent. Why? Because contacting an agent takes a long time, you end up in a queue, when you just want an answer straight away.
- 90% of customers expect consistency across channels – on the phone, website, retail store – they want the same treatment.
- 59% of channels are managed in silos; call centers have different incentives than retail stores, the website is managed separately, and so on.

Chatbots can help address all these challenges, but chatbots can be hard to create and expensive to maintain. Traditionally, it takes a lot of skill and effort to create a chatbot, you need developers to code the chatbot, Subject Matter Experts (SMEs) to define the conversational flow, AI specialists to handle the language processing, and the chatbot needs to be able to connect to your back-office systems to access relevant data.

## What are Power Virtual Agents chatbots

Power Virtual Agents is designed to create chatbots whether the user is external (your customers) or internal (your employees). SMEs can build conversational flows using an easy graphical interface, with no code, in a browser. Developers and administrators can assist SMEs in building conversational flows and integrating chatbots with line of business applications, using Power Automate cloud flows. Power Virtual Agents include AI language processing to create natural, conversational AI.

Power Virtual Agents make it possible for anyone to build a chatbot and greatly reducing the time and cost involved in deploying a chatbot.

## Use cases for chatbots

Conversational AI can be added into application to increase the user experience. Here are some common uses for external chatbots:

- Answering questions or enquiries
- Process return and exchange requests
- Resolving a complaint or problem
- Making a purchase / check inventory / make recommendation / track shipping
- Product onboarding / help desk
- Making a reservation / booking a ticket to an event
- Manage email subscription preferences
- Paying a bill / processing a claim
- Finding a qualified person (e.g. professional services)
- Triage for healthcare / crisis communication
- Customer feedback / quizzes, surveys and competitions

A popular use for chatbots is in customer support scenarios. For example, if you visit the HP support site <https://support.hp.com> a chatbot will assist you with troubleshooting, warranty, and repairs.

Most people's experience of chatbots in customer support scenarios, but there are potentially more powerful use cases for using chatbots inside of an organization such as:

- HR support
- Employee onboarding
- Employee ideas
- Change management
- IT support
- Sales support
- Case management
- Issue management
- Inspections
- Internal process information

Why should you consider building an internal chatbot?

- Consistent internal communication.
- Convenient, quick, and easy.
- Available 24/7.
- Can act on behalf of the user for routine tasks.
- Can replace intranet or internal email for frequently accessed information.

## Solution Architect's role in determining the need for chatbots

A solution architect should consider using chatbots as part of the solution:

- Consider chatbots to replace web forms, including lead generation forms and enquiry forms.
- Consider a chatbot to help customers navigate and find a person or information, instead of using a website search.
- Chatbots can be used to redirect enquiries to the right person in the organization – customer service agents or professional services staff.

Chatbots are not just for websites. Power Virtual Agents can also be easily deployed to:

- Facebook
- Slack
- Twilio
- Email
- Mobile apps

If the solution architect does decide to use chatbots, the solution architect will need to ensure that the project team follows the principles and best practices for building chatbots as discussed in the remainder of this module.

## Responsible AI

Microsoft has defined six principles for the responsible use of AI. As chatbots are AI based services, the solution architect need to keep these in mind when designing chatbots. The Transparency principle is one of the principles that particularly applies to chatbots.

The Transparency principle is concerned with ensuring that the human understands that they are interacting with a chatbot. When a chatbot conversation is started, the chatbot should clearly state that it is a chatbot. The chatbot should state its purpose and limitations for instance by listing out what the scope of what the bot can answer or do. A chatbot should enable the user to escalate or transfer to a human.

Chatbots work well when they are limited solely to their purpose and do not try to be too generic.

## Chatbot building options

Microsoft provides many options for building chatbots:

- Azure Bot Framework
- Azure Bot Composer

- Power Virtual Agents

## Azure Bot Framework

The Azure Bot Service is part of the Azure Bot Framework, a series of SDKs and tools that allow developers to create and deploy custom bots and virtual assistants using code. The Azure Bot Service is a managed service for developing bots.

A bot communicates by receiving messages and sending responses using an Azure Web App to handle the communications. Bots are like web applications; they take requests and return responses. A bot can perform operations like other applications, they can access databases, call APIs to other services, read files, and perform calculations.

The process of receiving a message and sending a response back to the user is known as a “turn”. Think how conversations between humans’ work, each person speaks one at a time i.e., in turn. Bots operate the same way, responding to user input in turn.

The screenshot shows the 'Test in Web Chat' page for a bot named 'Web App Bot'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Settings (Bot profile, Configuration, Channels, Pricing, Test in Web Chat), Encryption, Properties, Locks, Monitoring (Conversational analytics, Alerts, Metrics, Diagnostic settings, Logs), Automation (Tasks (preview), Export template), and Support + troubleshooting (Resource health, New support request). The main area features a large blue circular logo with a white 'L' shape and two dots. Below it, a section titled 'Welcome to Bot Framework!' provides instructions: 'Now that you have successfully run your bot, follow the links in this Adaptive Card to expand your knowledge of Bot Framework.' It includes three buttons: 'Get an overview', 'Ask a question', and 'Learn how to deploy'. At the bottom, a message input field says 'What can I help you with today? Say something like "Book a flight from Paris to Berlin on April 1, 2021"' with a 'Type your message' placeholder and a send arrow icon.

A developer can connect an Azure Bot Service bot with the Language Understanding Cognitive Service (LUIS) to add language understanding to a bot to allow for a conversational experience. A developer can also enable their bot to use a knowledge base created in QnA Maker.

An advantage of the Azure Bot Service is that once developers have built bot, the bot can be deployed to one or more channels, such as Facebook or Slack, without having to change the bot's code. The Azure Bot Service takes care of the communication between these channels and your bots. The Azure Bot Service adapts the messages your bot generates to the format of the channel it is connected to.

Bots created with the Azure Bot Framework can be integrated with Power Virtual Agent bots.

## Bot Service templates

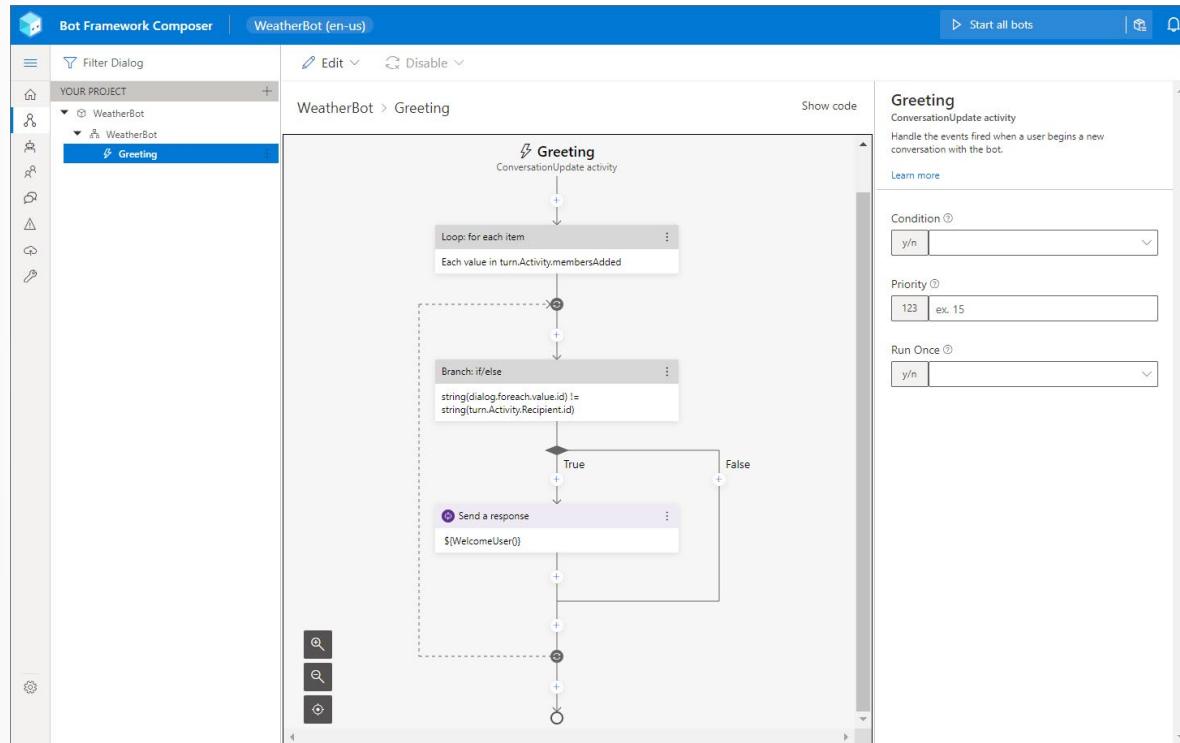
The Bot Service includes templates to help developers get started with building bots. If you create a bot in the Azure Portal you can select either the Echo bot template that just returns the user input, or the Basic bot template that includes LUIS.

## Bot Framework Composer

The Bot Framework Composer is a tool to build bots. The Bot Framework Composer uses a visual user interface to create the conversational flow and generate responses. Composer is a recent addition to Azure Bot Services and is the subject of ongoing development to add further features. Bot Framework Composer includes:

- A visual editing canvas for conversation flows
- Tools to author and manage language understanding (NLU) and QnA components
- Powerful language generation and templating system
- A ready-to-use bot runtime executable

Bot Framework Composer can be used to build bots without the need to write code and supports both LUIS and QnA Maker.



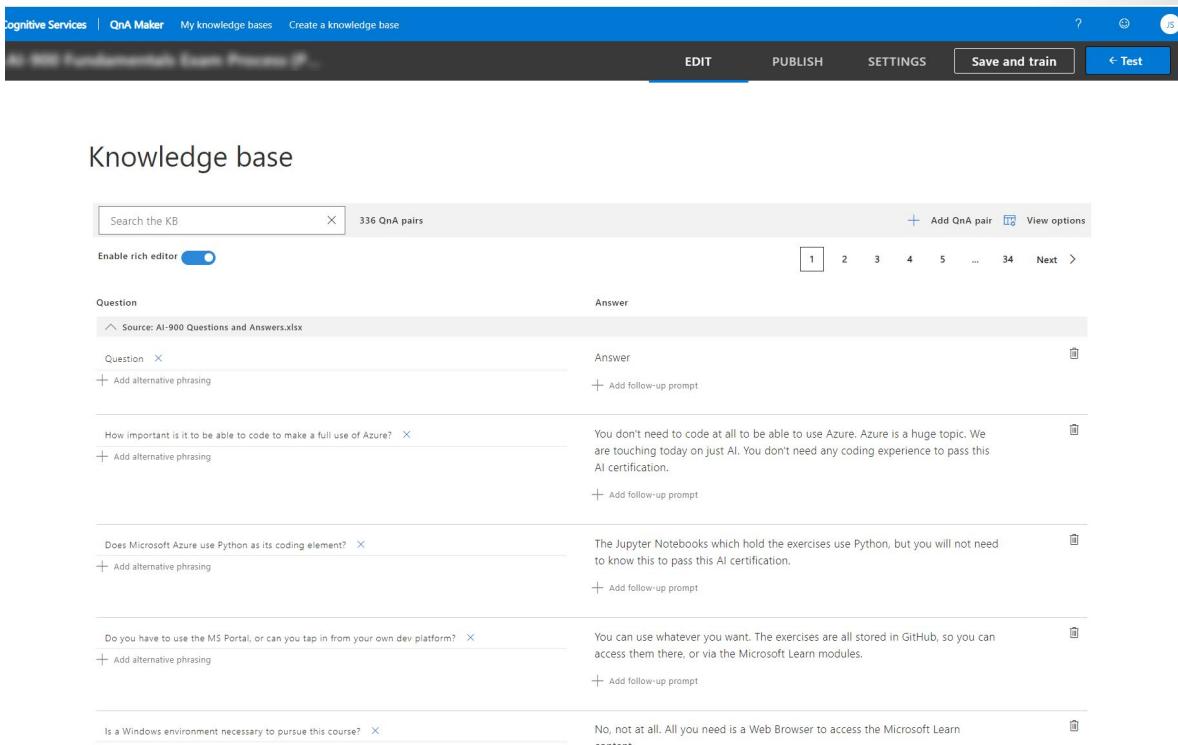
There are similarities between Power Virtual Agents and Composer; both provide a no-code authoring canvas for users to build bots. However, there are key differences in the functionality and look of both technologies. For instance, topics in Power Virtual Agents are similar to dialogs in Composer.

The Bot Framework Composer is open source and is multi-platform with support for Windows, Linux, and MacOS. You can find Bot Composer on GitHub at <https://github.com/microsoft/BotFramework-Composer>.

Bot makers can use Bot Framework Composer to create custom content and add it to Power Virtual Agents.

## **QnA Maker**

QnA Maker is a service that creates a searchable knowledge base from existing documents and websites. This knowledge base contains built in natural language processing. The QnA Maker knowledge base can then be used in bots and other applications to respond to FAQ type questions.



The screenshot shows the QnA Maker interface. At the top, there's a navigation bar with 'Cognitive Services', 'QnA Maker', 'My knowledge bases', and 'Create a knowledge base'. Below the navigation bar, the title 'AI-900 Fundamentals Exam Practice QnA' is displayed. The main area is titled 'Knowledge base' and shows a search bar 'Search the KB' with '336 QnA pairs' found. An 'Enable rich editor' toggle switch is turned on. A table lists several QnA pairs:

Question	Answer
How important is it to be able to code to make a full use of Azure?	You don't need to code at all to be able to use Azure. Azure is a huge topic. We are touching today on just AI. You don't need any coding experience to pass this AI certification.
Does Microsoft Azure use Python as its coding element?	The Jupyter Notebooks which hold the exercises use Python, but you will not need to know this to pass this AI certification.
Do you have to use the MS Portal, or can you tap in from your own dev platform?	You can use whatever you want. The exercises are all stored in GitHub, so you can access them there, or via the Microsoft Learn modules.
Is a Windows environment necessary to pursue this course?	No, not at all. All you need is a Web Browser to access the Microsoft Learn content.

At the bottom right, there are buttons for 'Add QnA pair', 'View options', 'Save and train', and 'Test'.

QnA Maker does not provide a bot itself, but you can easily generate an Azure Bot Service bot from your QnA Maker knowledge base in a few minutes without writing any code.

A knowledge base consists of question-and-answer pairs. You can create the questions and answers in a number of different ways:

- Extraction from existing documents.
- Extraction from web pages.
- Manual input.

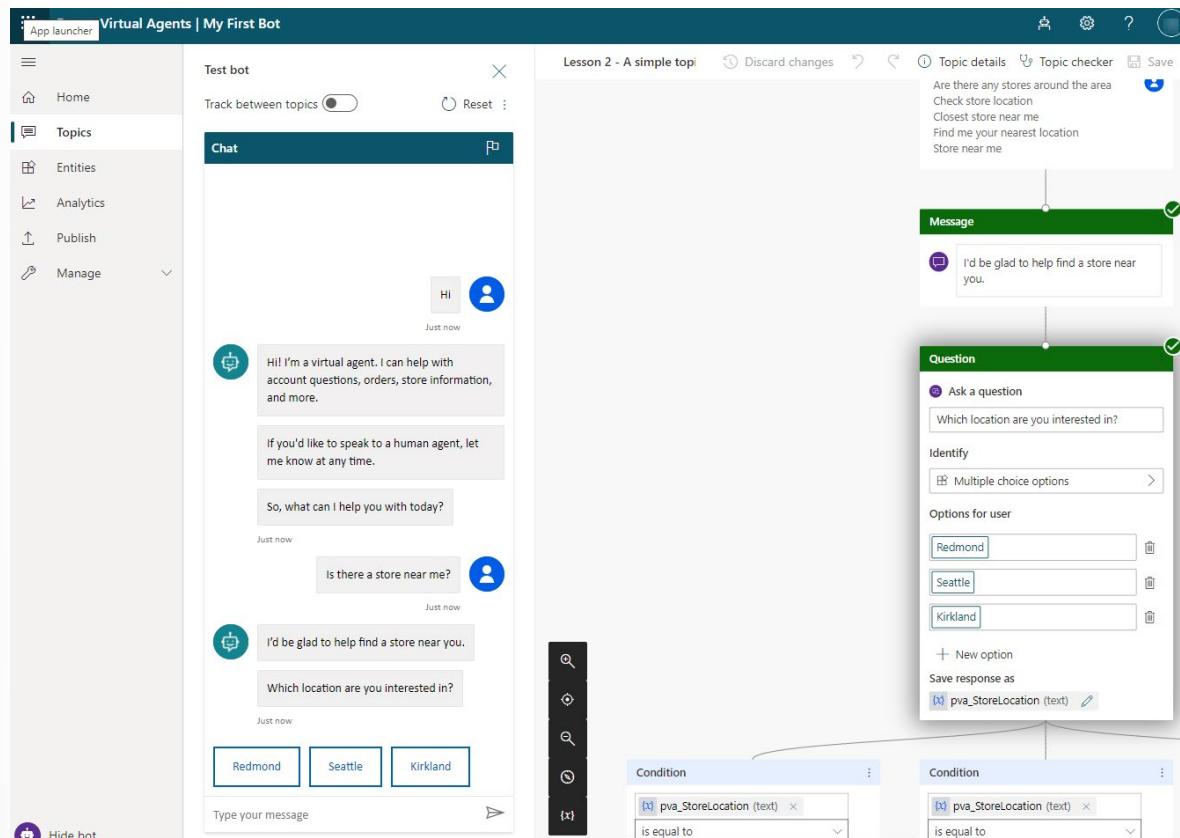
QnA Maker knowledge bases can be integrated with Power Virtual Agent bots with Power Automate.

## **Power Virtual Agents**

Power Virtual Agents chatbots are chatbots powered by artificial intelligence (AI) that allow users to focus on more complex and higher-value work while the chatbots themselves handle simple repetitive interactions. With Power Virtual Agents, you can create chatbots using an easy to use, no-code graphical interface from within a web browser.

Power Virtual Agents chatbots interact with customers and employees, answer questions, and provide information. You can deploy Power Virtual Agents chatbots to multiple channels, including your own website, Facebook, and Microsoft Teams.

Power Virtual Agents chatbots can be created easily without the need for data scientists or developers.



Power Virtual Agents is built on top of the Azure Bot Framework and leverages the capabilities of the Bot Framework in a no-code environment.

Power Virtual Agents can meet many simple requirements for chatbots, both internal and external. Power Virtual Agents are designed to be built with, or by, Subject Matter Experts (SMEs).

Power Virtual Agents are licensed per tenant and are charged based on the number of billed sessions per tenant per month. There is an initial entitlement of 2,000 sessions and additional capacity can be purchased.

The solution architect needs to determine which bot technology, or combination of technologies to use in the solution. This will depend on the skills available, if there are existing bots already deployed, and the complexity of the bots required.

## Chatbot Concepts

When creating chatbots there are some important concepts to understand before building your chatbot.

### Language Understanding

When humans interact with a computer application using text or speech, a human should not be expected to use the application's internal instructions. A computer application must be able to handle language

provided in a natural way and be able to react accordingly to the meaning that the human has expressed. Natural Language Processing, an AI capability, bridges the gap between human and computer language, it allows humans and computer applications to work together in a natural way.

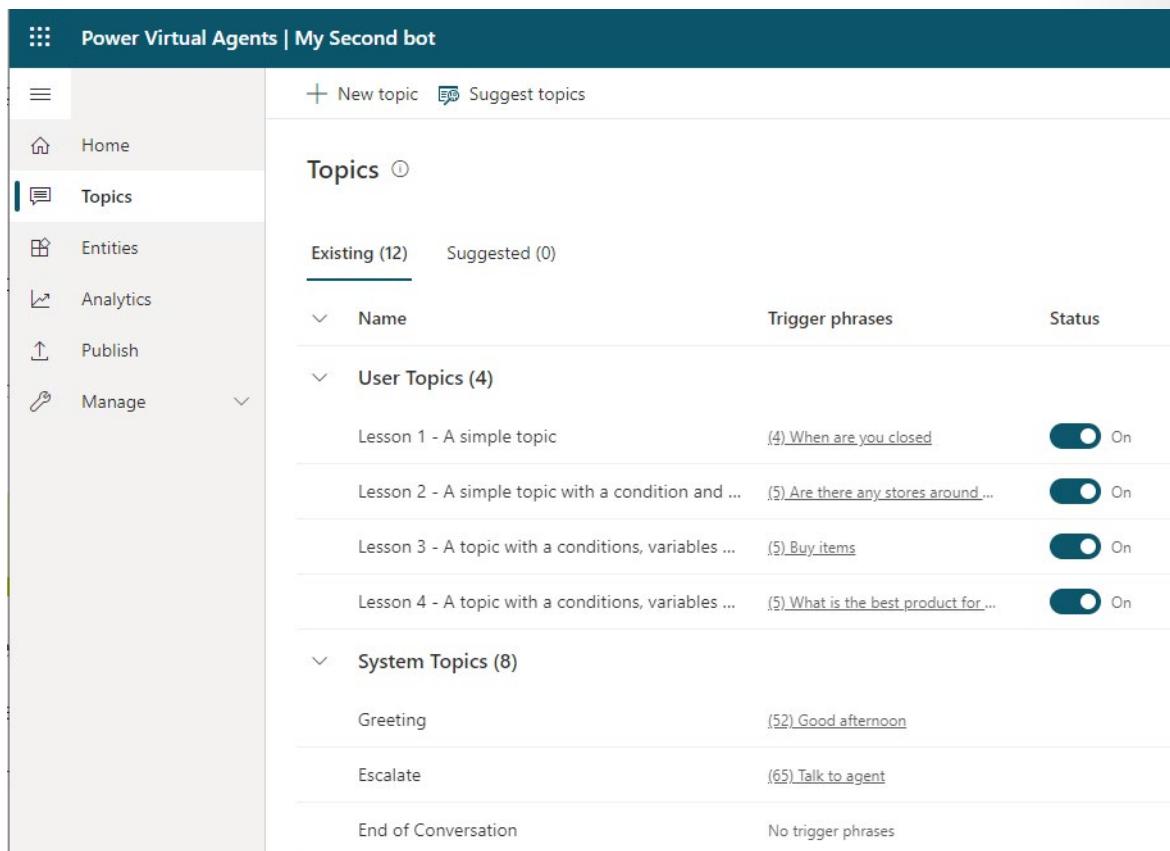
Power Virtual Agents hosts multiple AI models and AI capabilities, the core of which is a transformer-based natural language understanding (NLU) model. Power Virtual Agents leverages the natural language processing of the Language Understanding service (Luis) which is part of Azure Cognitive Services.

Power Virtual Agents, employs a language understanding model that uses an example-based approach, powered by a deep neural model. This type of large-scale model only needs to be trained once with large amounts of data, and can then be used for specific tasks with few examples without further training. Specifically for Power Virtual Agents, the use of this model allows for an intuitive way for bot makers to work on their bot content confidently, without having to involve AI experts.

## Topics

Power Virtual Agents work by identifying the subject that the user is asking about and then having a conversation about that subject.

Topics are the main subjects of the conversation. A Power Virtual Agents chatbot can have up to 1,000 topics. Each topic is a separate conversation path. It is the combination of topics in a chatbot that provides a natural conversational flow. You create topics for the tasks or requests that you need your chatbot to respond to.



The screenshot shows the 'Topics' section of the Power Virtual Agents interface for a bot named 'My Second bot'. The left sidebar includes options for Home, Topics (which is selected), Entities, Analytics, Publish, and Manage. The main area displays a list of topics under 'Existing (12)' and 'Suggested (0)'. The 'User Topics (4)' section contains four entries:

Name	Trigger phrases	Status
Lesson 1 - A simple topic	(4) When are you closed	On
Lesson 2 - A simple topic with a condition and ...	(5) Are there any stores around ...	On
Lesson 3 - A topic with a conditions, variables ...	(5) Buy items	On
Lesson 4 - A topic with a conditions, variables ...	(5) What is the best product for ...	On

The 'System Topics (8)' section lists three entries:

Name	Trigger phrases
Greeting	(52) Good afternoon
Escalate	(65) Talk to agent
End of Conversation	No trigger phrases

Topics define the purpose of your chatbot and are the first step in authoring your chatbot. A topic has two parts:

- Trigger phrases: The keywords, phrases, or utterances that the user will enter
- Conversation nodes: How your bot should respond

Each topic has its own conversation flow with the bot. When a bot identifies a trigger for a topic, the conversation for that bot is initiated.

The use of the natural language understanding (NLU) model means that in Power Virtual Agents, when you craft trigger phrases for a topic, you only need to provide a few examples, usually in the range of five to ten phrases for a single topic. Shorter trigger phrases are better, and you should aim for two to ten words. You just need to make sure trigger phrases are semantically different: changing a single verb or noun could be enough to expand a topic's coverage. Adding things like new articles (changing or adding 'the' or 'a' or 'an'), changing capitalization, adding contractions (you're or don't), or adding plurals won't improve the triggering because contractions are already accounted for in the natural language understanding model.

## Generating topics

Many organizations will have procedures, product information, frequently asked questions, and other information in documents or on websites. Power Virtual Agents can extract information and create topics with trigger phrases using the Suggest topics option.

[!NOTE]

If you are using Dynamics 365 Customer Service Insights, you can select the topics from within Customer Service Insights and add the topics and trigger phrases to your Power Virtual Agent chatbot.

## Entities

Power Virtual Agents attempts to extract information from the phrases entered by the user. This extracted information can be used to control the conversation's path. Power Virtual Agents uses entities to identify information in a textual phrase, for instance, names, dates, and numbers. Your chatbot can then use this information to decide on the appropriate next step in the conversation.

Entities are people, places, and things that a chatbot can identify from the phrases entered into a chatbot. Power Virtual Agents includes a set of pre-built entities for the most commonly used objects and you can create custom entities for the domain of the business solution you are building.

The screenshot shows the 'Entities' section of the Power Virtual Agents interface. On the left, there's a sidebar with icons for Home, Topics, Entities (which is selected), Analytics, Publish, and Manage. The main area has a header 'Entities' with a help icon. Below it is a table with columns: Name (sorted by ascending), Description, and Method. The table lists nine prebuilt entities:

Name ↑	Description	Method
Age	Age of a person, place, or thing, extracted as a number	Prebuilt
Boolean	Positive or negative responses, extracted as a Boolean	Prebuilt
City	City names, extracted as a string	Prebuilt
Color	Primary colors and hues on the color spectrum, extracted as a string	Prebuilt
Continent	Continent names, extracted as a string	Prebuilt
Country or region	Country and region names, extracted as a string	Prebuilt
Date and time	Dates, times, days of the week, and months relative to a point in time, extracted as strings	Prebuilt
Duration	Lengths of time, extracted as a string, in standard TimeSpan format	Prebuilt
Email	Email addresses, extracted as a string	Prebuilt

## Channels

The Azure Bot Framework separates the logic of the bot from the communication with different services. When you create a bot, the bot is only available for use embedded on websites with the Web Chat channel. You can add channels to your bot to make the bot available on other platforms and services, known as channels.

One of the major benefits of the Azure Bot service is that you develop your bot once and connect to multiple channels without needing to change the code for each channel to handle the specific requirements and formats of that channel. The Azure Bot Service takes care of those requirements and converting the formats.

The following channels are available for connection to bots:

- Alexa
- Direct Line
- Direct Line Speech
- Email
- Facebook
- GroupMe
- Kik
- Line
- Microsoft Teams
- Skype
- Slack

- Telegram
- Telephone
- Twilio (SMS)
- Web Chat

Connect to channels

The screenshot shows the Microsoft Bot Framework portal interface. At the top, there is a table with three columns: 'Name', 'Health', and 'Published'. A single row is present, showing 'Web Chat' under 'Name', 'Running' under 'Health', and two small dots under 'Published'. To the right of this row are 'Edit' and 'Preview' buttons. Below the table, there is a button labeled 'Get bot embed codes'. Underneath the table, there is a section titled 'Add a featured channel' with two icons: a globe and a Microsoft Teams logo. Below this, there is a section titled 'More channels' containing a grid of ten icons, each with a name: Alexa, Direct Line Speech, Email, Facebook, GroupMe, Kik, LINE, Skype, Slack, Telegram, Telephony, and Twilio (SMS). Each icon has a small description text next to it.

Power Virtual Agents can be deployed to the same channels.

## Best practices

Chatbots, if used, are a very powerful tool for increasing customer interaction with your solution. A solution architect must ensure that the chatbots are correctly built and meet the requirements.

This section contains recommendations around designing and deploying chatbots.

## Planning

When planning a chatbot the solution architect should:

- Define the scope of the chatbot
- Define the purpose of the chatbot
- Define which channels the bot will be deployed to
- Define the key metrics and success criteria
- Verify the topics, entities, and conversational flows.

A chatbot supports three different types of topic:

- Informational
- Tasks
- Troubleshooting

A good Power Virtual Agents chatbot will have a high business impact i.e., a high level of traffic, a low level of integration complexity, and will achieve a high level of conversation completion without the need for escalation to a human agent.

A well designed chatbot will have a well defined set of goals. Each topic is linked to a business process and has trigger events, a clear set of rules, a set of documents, and a series of tasks that will be performed.

The solution architect may need to provide guidelines for creating topics and trigger phrases for chatbot authors to follow.

Watch this video for more details <https://community.powerbi.com/t5/MBAS-Gallery-2020/Best-practices-for-building-an-intelligent-bot-with-Power/td-p/1100291> and read this blog post <https://powervirtualagents.microsoft.com/blog/best-practices-when-planning-to-use-chat-bots/>.

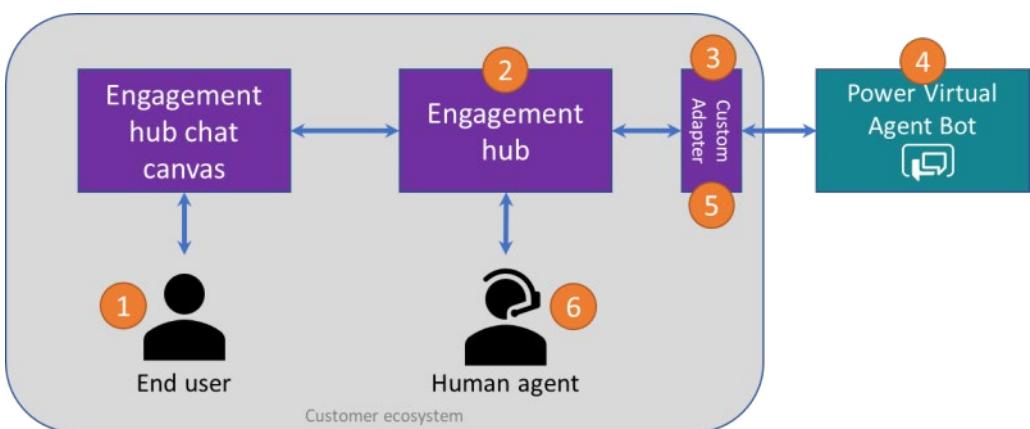
## Escalations

With Power Virtual Agents, you can hand off conversations to live agents seamlessly and contextually.

When you hand off a conversation, you share the full history of the conversation (the context) as well as all user-defined variables. Having access to this context means live agents that are using any connected engagement hub can be notified that a conversation requires a live agent, see the context of the prior conversation, and resume the conversation.

[!IMPORTANT]

You need to have an engagement hub that is being used by live agents, such as Omnichannel for Customer Service, and you need to configure the connection.



The solution architect should determine when escalation should occur and how escalation will handled.

Power Virtual Agents chatbots come with telemetry built-in so that you can monitor how your chatbots are being used. Key KPIs are the rates of abandonment and escalation to a human agent. You should monitor your chatbots and change your chatbot to increase its effectiveness.

## Variables

Variables let you save responses from your users in a conversation so that you can reuse them later in conversations.

The response for each question asked in a conversation is stored as a variable. You can then pass the variable to a Power Automate flow or use the variable later in the topic, or even in other topics, to control

the questions being asked. For example, you can use a variable to decide to skip a question if you have the information you need at that point.

Variables can be defined as either of the following:

- Topic: The variable can only be used within its topic.
- Bot: The variables can be used by any topic.

The solution architect should encourage the use of variables by chatbot authors to improve conversational flow.

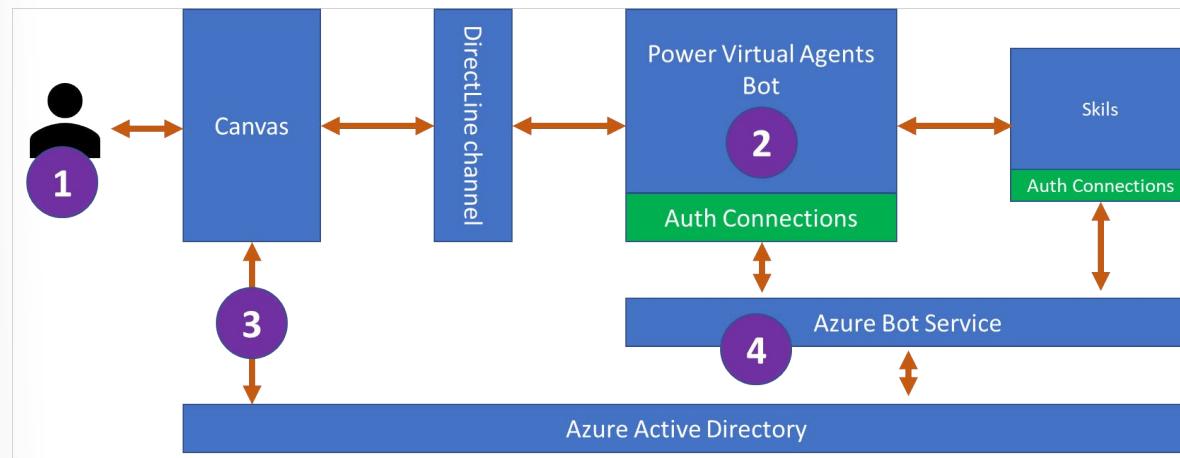
## Authentication

You can enable user authentication directly within a Power Virtual Agents bot conversation. User authentication means you can get a user's basic properties such as name and ID in bot variables, but also prompt a user to sign in using an authentication node, retrieve a user token for that user, and then use that token to retrieve the user's information from a back-end system.

Power Virtual Agents supports the following authentication providers:

- Azure Active Directory.
- Any identity provider that is compliant with the OAuth2 standard e.g., Microsoft Account or Facebook.

Power Virtual Agents supports single sign-on (SSO), which means chatbots can sign the user in if they're in to the page where the bot is deployed. You will need to register the web app in Azure Active Directory to enable SSO.



[!NOTE]

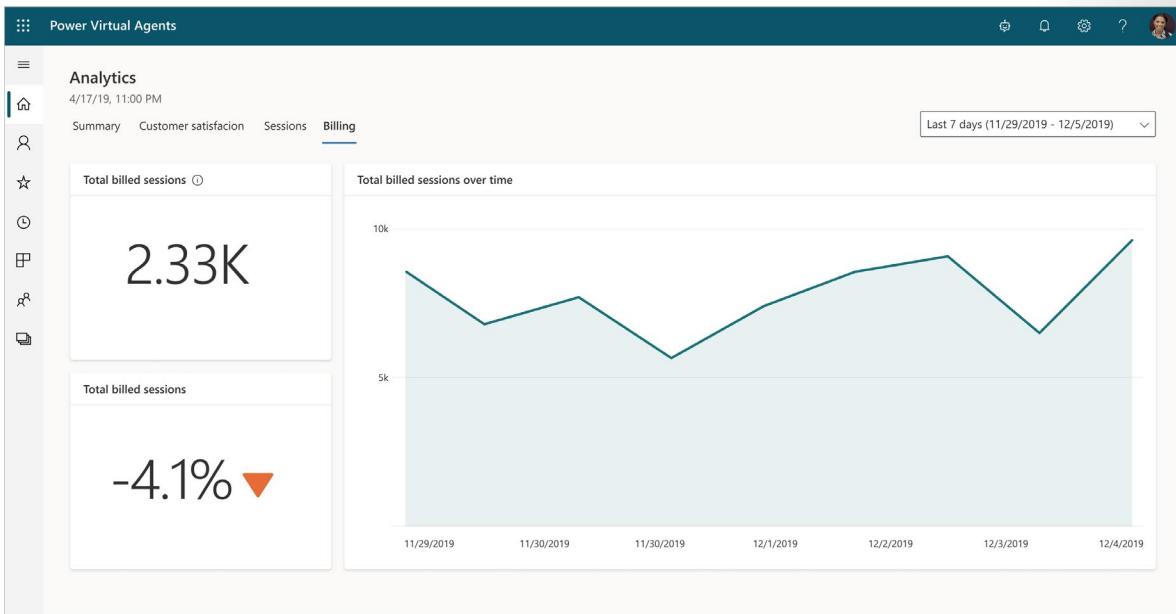
SSO is only supported on the live website publication channel and the Teams channel

The solution architect should determine if authentication is required for bots and the identity provider used. In many situations, the organization may already have identity providers configured in Azure Active Directory. If creating chatbots for Microsoft Teams then this is simple to configure with the "Only for Teams" option, however if creating chatbots for customers you may need to consider Azure B2B and Azure B2C for authentication.

## Capacity

When you purchase a Power Virtual Agents license, you gain capacity for the specified number of billed sessions. Power Virtual Agents pools this capacity across the entire tenant.

You can monitor how many billed sessions have been used from the Analytics tab in the Power Virtual Agents portal.



The solution architect will need to estimate the number of sessions required and ensure that monitoring is implemented to track bot usage and costs.

## Rate limits

Quotas are applied to chatbots to limit how often messages can be sent to the chatbot. The purpose of quotas is to throttle the service load and protect the service from being overloaded.

Quotas for Power Virtual Agents chatbots are defined as requests per minute (RPM). A request is a message from the user to the chatbot, or a message from an Azure Bot Framework Skill, in a single chat session.

The Quota is 600 RPM in the North America region and 800 RPM for the rest of the world.

## Solutions

Power Virtual Agents are solution aware and can be included in solutions and application lifecycle management processes.

### [!IMPORTANT]

Bots contain many sub-components, such as Topics, that must all be exported and imported together. You should consider creating segmenting your solution and have bots and their sub-components separate in a separate solution from other components.

### [!NOTE]]

You can only import and export bots with the Power Virtual Agents web app. The feature is not available in the Power Virtual Agents app in Microsoft Teams.

## Deployment

Power Virtual Agents are created in a selected environments. You should ensure that you are using the correct environment for development, test, and production purposes when creating chatbots.

If you are using skills then you need to define environment variables for each skill.

Post deployment of your bot via a solution there are manual tasks that may need to be performed:

- Power Automate cloud flows: Configure any connections for the first time and visit the Power Virtual Agents portal and select the bot.
- Skills: Add the values for the skills' environment variables.
- End-user authentication: Configure end-user authentication in the bot so it can take actions on the user's behalf.
- Escalations: Configure external services that hand off bot escalations to a human agent.
- Multichannel: Configure external channels, such as Facebook, and internal non-Power Virtual Agents services, such as Microsoft Teams.

The solution architect should ensure that these steps are included in the deployment plan for the solution.

## Integrating chatbots

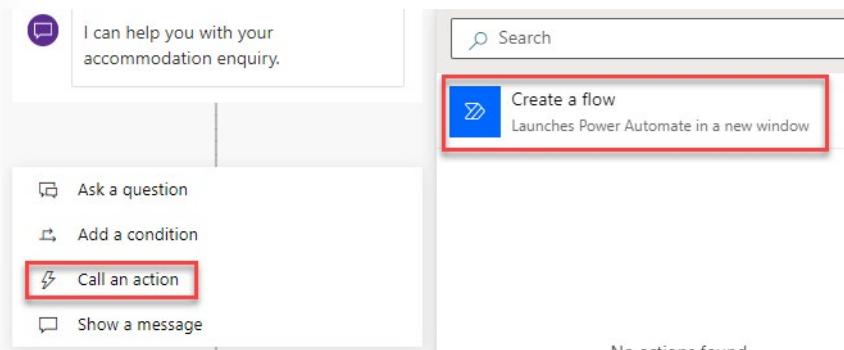
Power Virtual Agent chatbots will often require access to other services and systems to be able to complete their goals. Power Virtual Agents can also be integrated with other Microsoft AI services to create an enhanced customer experience.

## Power Automate

The primary integration for Power Virtual Agent chatbots is through Power Automate cloud flows.

Power Virtual Agents can pass parameters to a Power Automate cloud flow and receive data back from the flow and use that data in the chatbot's conversation flow. For instance, a chatbot may handle customer order updates and so will use a Power Automate cloud flow to fetch the order's status from the relevant system.

Power Automate cloud flows can be initiated from the Call an action node in a topic.



## Skills

Power Virtual Agents enables you to extend your bot using Microsoft Bot Framework skills. If there are existing Azure Bot Framework bots deployed you can define those bots as a skill and embed the skill within a Power Virtual Agents bot. You can then use the Azure Bot Framework bot from within your Power Virtual Agent chatbot.

A developer can also use a Power Virtual Agent chatbot from an Azure Bot Framework bot. The Power Virtual Agent becomes the skill and the Microsoft Bot Framework dispatcher tool can integrate with the Power Virtual Agents bot.

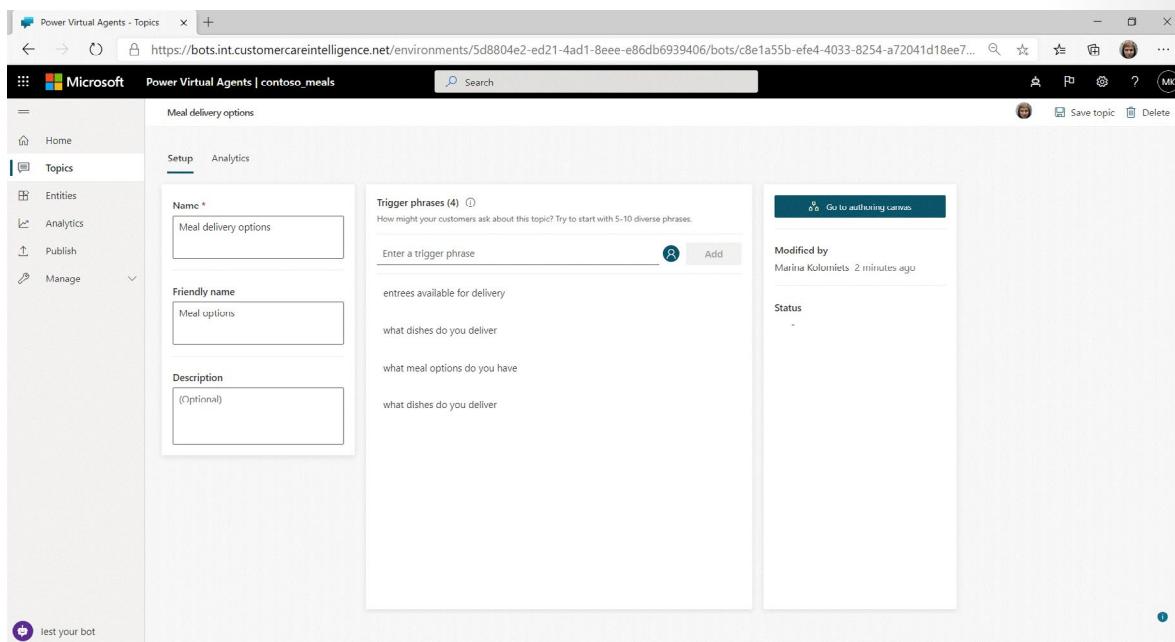
The solution architect needs to oversee the design of bots and decide when to leverage existing bots and when to create new bots and with which toolset.

## Bot Composer

Bot makers can use Bot Framework Composer to create custom content and add it to Power Virtual Agents.

For instance, by using Composer with your Power Virtual Agents chatbot you can:

- Show an Adaptive card.
- Use Bing Search as a fallback.



See <https://docs.microsoft.com/composer/pva/overview-composer-pva> and <https://docs.microsoft.com/power-virtual-agents/advanced-bot-framework-composer> for more details.

## QnA Maker

You can extend a Power Virtual Agents chatbot to provide answers from a QnA Maker knowledge base. You need to add a fallback topic that uses Power Automate to run a cloud flow that connects to the QnA Maker via its connectors to respond to questions.

Generate answers using QnA Maker knowledge base from Power Virtual Agents

The diagram illustrates a flow from an 'Unknown' state to a 'Response and QnA Maker' state. On the left, a grey box contains a plug icon and is labeled 'Unknown'. An arrow points to the right, leading to a teal and blue box labeled 'Response and QnA Maker'. The teal section contains a globe icon, and the blue section contains a speech bubble icon. A 'PREMIUM' badge is in the top right corner of the main box.

Using Power Virtual Agents, call QnA Maker knowledge base to generate answers.

This flow will connect to:

QnA Maker my qna kb Continue

See <https://docs.microsoft.com/azure/cognitive-services/qnamaker/tutorials/integrate-with-power-virtual-assistant-fallback-topic> for the steps required.

## Cognitive Services

Azure Cognitive Services can be used by bots for a number of purposes such as:

- Sentiment analysis
- Image classification
- Form recognition

You can use Azure Cognitive Services with Power Virtual Agents using Power Automate cloud flows and the associated Cognitive Services connector.

## Power Virtual Agents in Teams

Power Virtual Agents is available as both a standalone web app, and as a discrete app within Microsoft Teams. Most of the functionality between the two is the same. However, there are different reasons to choose one version or the other based on your requirements.

## Power Virtual Agents Web app

You will use standalone Power Virtual Agents if:

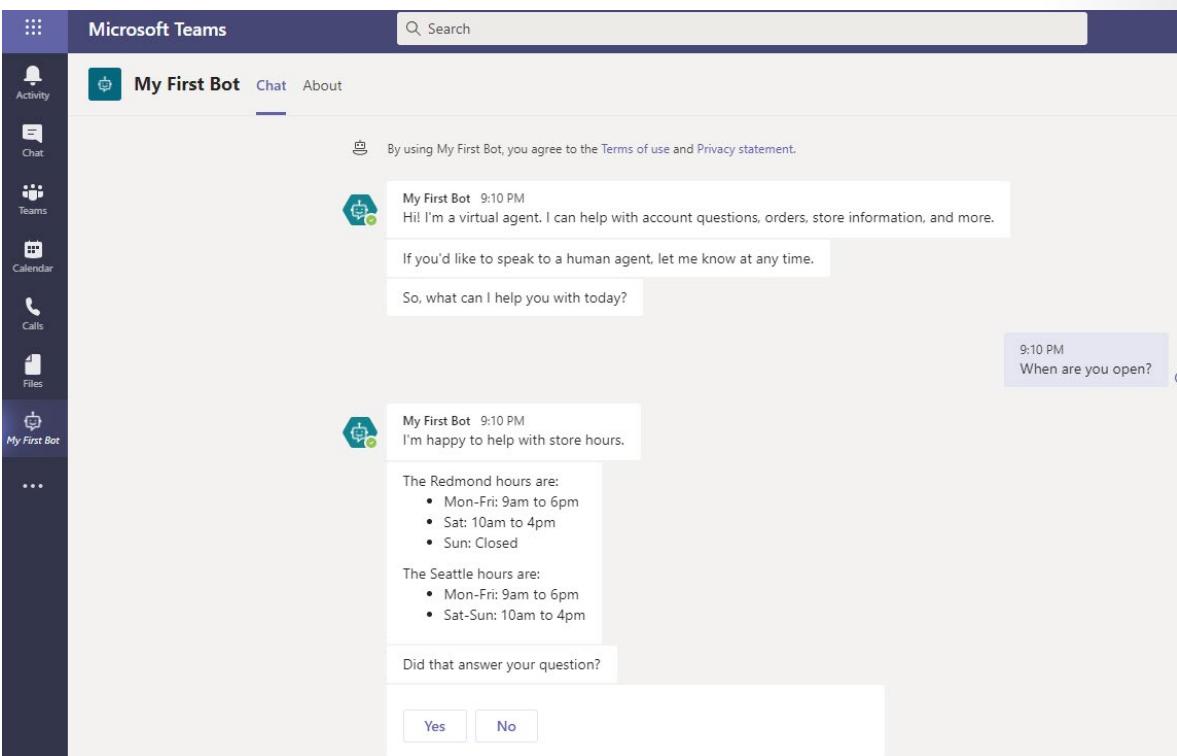
- You are an IT administrator or consultant and want to create bots for your customers to engage with.
- You want to create bots for external customers.
- You have used chatbot services in the past, and want to trial or test Power Virtual Agents.
- You are familiar with advanced chatbot concepts, such as entities and variables, and want to create complex chatbots.

The standalone web app is available at <https://powerva.microsoft.com>

## Power Virtual Agents in Microsoft Teams

You will use Power Virtual Agents in Microsoft Teams if:

- You are an employee or member of an organization or team and want to create chatbots to answer common questions posed by other employees or teammates.
- You want to use advanced concepts, such as entities and variables, but have the chatbot available only internally.
- You want to create and distribute a chatbot in the shortest time possible.



The Power Virtual Agents app in Microsoft Teams supports single sign-on (SSO), which means chatbots can sign the user in silently, without having the user enter their credentials.

Power Virtual Agents in Microsoft Teams does not require additional licensing as the license is included with Microsoft 365 subscriptions.

**[!IMPORTANT]**

Access is limited however to the membership of the Microsoft Team that the chatbot is deployed to. The chatbot cannot be deployed to other channels.

## Limitations

Power Virtual Agents in Microsoft Teams have a number of limitations such as:

- Power Virtual Agents in Microsoft Teams are limited to the standard Power Automate connectors available for flows triggered from Power Virtual Agents.
- Power Virtual Agents in Microsoft Teams can be integrated with Azure Bot Framework skills but require a Power Virtual Agents standalone license.
- Power Virtual Agents in Microsoft Teams cannot integrate Microsoft Bot Framework dialogs, and cannot be used with Azure Bot Framework Composer.
- Power Virtual Agents in Microsoft Teams cannot be included in solutions.

## Service limits

Power Virtual Agents in Microsoft Teams are limited to 10 sessions per user every 24 hours.

# Evaluate chatbot requirements exercise

## Exercise

### Evaluate requirements for chatbots

Review the requirements for each scenario. Decide how best to address the requirement with chatbots.

#### Scenario: Product and pricing

The organization has a complex set of products and prices. Sales users find it difficult to find information about which products can be sold together and how to determine the correct pricing.

You have the following set of requirements:

- Sales people must be able to find details about products easily.
- Sales people need to be shown the pricing rules for each product.
- The rules for products and prices vary by country.

What should you use to meet these requirements?

#### Scenario: Customer service

The customer service department is overwhelmed with calls. You have been tasked with improving productivity of the department and increasing customer satisfaction.

You have the following current systems:

- Dynamics 365 Customer Service is installed for basic ticket management.
- Legacy website that allows customers to login and see their tickets and submit new tickets. New tickets are created in Dynamics 365 but there is no further integration.
- SharePoint library with support processes and other documentation.

You have the following set of requirements:

- The website only shows the basic ticket details that were submitted. Customers must be able to easily obtain the status and latest action for their tickets.
- Customers have requested a search capability for support information on the website.
- Easily answered questions should be handled by a bot.
- Customers should be able to be transferred to a human agent with the bot conversation available to the human agent.

What should you use to meet these requirements?

## Module Summary

### Summary

Power Virtual Agents are a very powerful tool for creating great customer experiences. The solution architect should consider using chatbots as part of the solution:

- Replace web forms.
- Help customers find information, instead of using a website search.
- Employee assistance and information retrieval.

If the solution architect does decide to use chatbots, the solution architect will need to ensure that the project team follows the principles and best practices outlined in this module.

### Next steps

Read the following:

- Power Virtual Agents best practices blog post <https://powervirtualagents.microsoft.com/blog/best-practices-when-planning-to-use-chat-bots/>
- Power Virtual Agents Create bots with Power Virtual Agents Learning Path <https://docs.microsoft.com/learn/patterns/power-virtual-agents/>

## Module 14 Robotic Process Automation

### Explore Robotic Process Automation

#### Introduction to Robotic Process Automation

This unit introduces Robotic Process Automation (RPA) and the use cases for Power Automate desktop flows.

There are many legacy applications that do not have a method for accessing their data or functionality except through their user interface. RPA solutions operate by generating a list of actions by capturing the steps a user performs in an application's user interface. The RPA solution then performs the automation by repeating the steps in the application's UI, effectively simulating the user's actions in software. RPA enables the automation of applications that do not have an API that can be used for such automation tasks.

Robotic Process Automation (RPA) is garnering a lot of interest and attention. By automating simple tasks, RPA solutions promise to lower costs while reducing errors.

#### Legacy Automation issues

When analyzing work performed by users the following has been observed:

- 60% of all occupations have at least 30% automatable activities.
- Almost 50% of work activities globally can be automated using current technology.
- Data collection and processing times can improve by 64% with automation.

The best organizations are powered by strategic and creative people, but they are often forced to spend nearly half their time on repetitive tasks that could be automated with current technology.

A major roadblock in automating systems and workflows can be that legacy and other third-party applications have no modern API data access that allow tools such as Power Automate to include them in their cloud flows.

It is not always possible to replace the legacy applications either because of difficulty or the costs to replace them. In addition, many legacy applications are on-premises and it can be hard to link to the on-premise applications and combine them with cloud services in an end-to-end automation process.

A number of RPA solutions have been developed to solve this problem. Power Automate desktop flows are Microsoft's answer to this problem.

## Automation with Power Automate

There are many SaaS services that can be accessed with modern REST APIs that automation tools such as Power Automate can connect to and orchestrate for the purposes of integration and automation of tasks.

Using Power Automate, a solution architect can create innovative solutions. However, what happens when there isn't a connector available. If the service has a modern REST API then you can create a custom connector. If there isn't an API then Power Automate desktop flows are available.

Microsoft's approach to automation is:

- Accelerate productivity: Minimize repetitive, manual, time-consuming tasks and create more time for your teams to focus on strategic work.
- Automate at scale: Allow everybody in your organization to automate workflows using connectors for their favorite on-premises and cloud-based apps and services. From end users, professional developers, to IT.
- Apply intelligent automation: Streamline how you work by combining the power of AI with automated workflows and business process.
- Integrate automation, more securely: Enable end users to build automated workflows that comply with established policies and focus your skilled IT resources on more complex, strategic work.

This flowchart can help you determine how to architect your automation.

Decision-making flowchart for your design.](..../media/1-decision-making-flow.png)

## Use cases for Power Automate desktop flows

Power Automate desktop flows automate repetitive tasks and there are many scenarios that Power Automate desktop flows can be employed in. Some use cases for desktop flows are as follows:

- Invoice processing: Processing invoices includes many repetitive tasks, which, if performed incorrectly, can lead to delayed or incorrect payments. For instance, invoices need to be checked against the corresponding purchase orders. Desktop flows can process invoices and automatically perform the required validation checks.
- Recruitment: Desktop flows could gather and collate applications from multiple job portals into a single applicant list.
- New user onboarding: New joiners to an organization must be set up on many systems. While you can use tools such as PowerShell to perform some setup, there are some applications for which there is no automation available. Desktop flows can be used to add users and configure their settings automatically through the applications' user interfaces.

## Solution Architect's role

By using Power Automate cloud and desktop flows, a solution architect can design end-to-end processes that encompass modern cloud services, legacy desktop, and web applications.

The solution architect should look for opportunities for automation. Good opportunities exhibit the following characteristics:

- Standard process: Is the process well understood and consistent in how it needs to be executed?

- High frequency: Is this a process that runs frequently and the ROI exists to spend build cycles on it?
- Predictable processes: Are the outcomes predictable or do they require any judgement calls by humans?
- Prone to human error: Is there a risk that a human may introduce errors as a result of manually completing this process? Are there a lot of steps that are easy to perform out of order?
- High risk of failures and impact: Is the risk and/or impact of a failure high that could be mitigated through automation?

There are many uses for desktop flows. You can consider any scenario where there is the processing of forms, extracting data from systems, or processing claims where the data needs to be verified for automation with desktop flows. If you have rule-based processing, desktop flows are a viable candidate for automating those processes.

The introduction of Power Automate Desktop opens up many scenarios that would not have been possible or would have been extremely difficult and expensive to achieve.

Desktop flows can also be used for situations where there is an API but the API does not expose all of the operations available in the application's UI. In such circumstances, Power Automate Desktop maybe the answer.

During design, the solution architect should separate flows into smaller automations so that you don't have a single automation that covers the entire process. There are several reasons why you should make multiple, smaller flows:

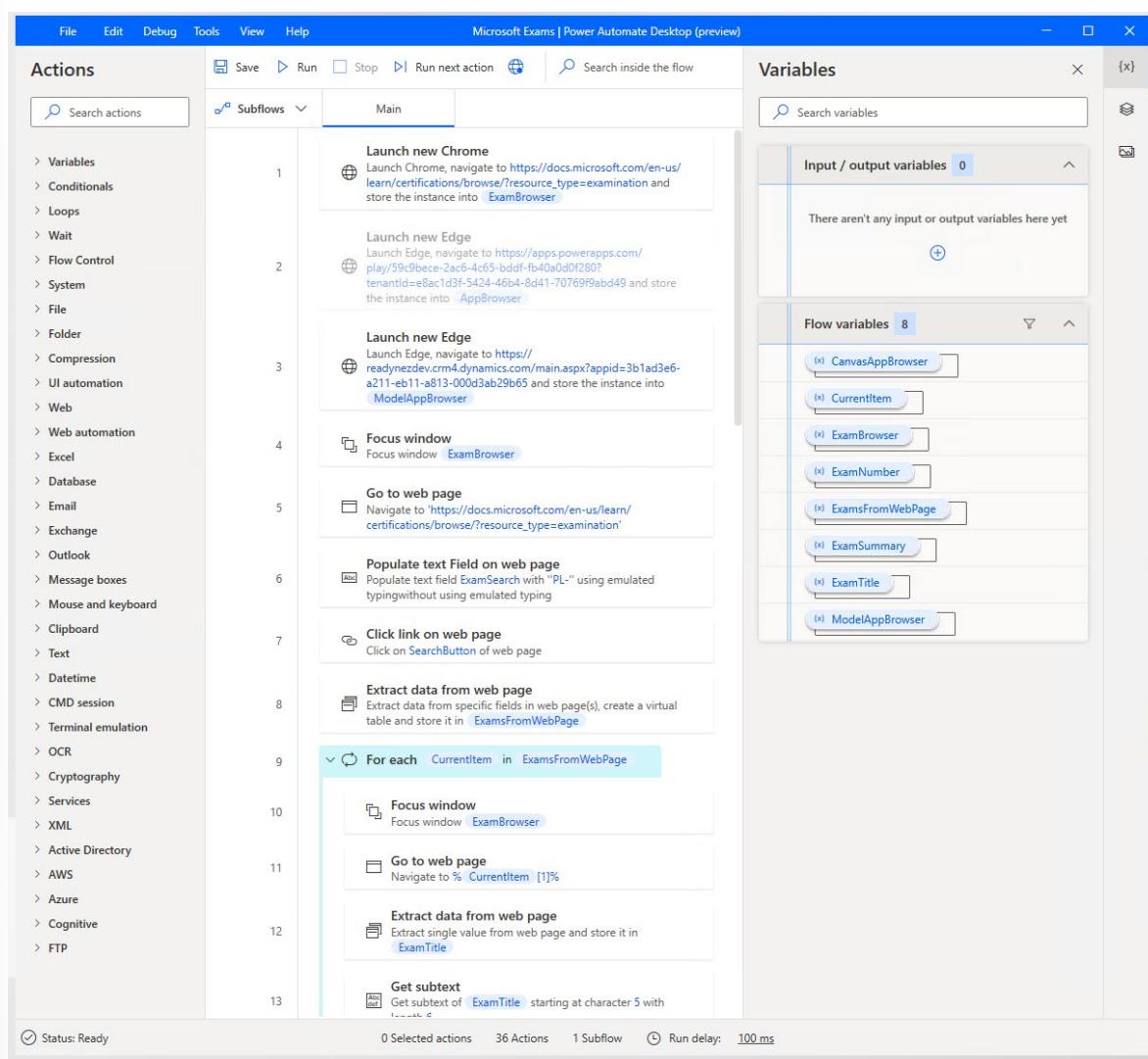
- Multiple people can work on the automation.
- Small flows with common steps may be able to be reused.
- Error handling does not need to be as sophisticated.
- Maintenance is easier.
- Easier to manage the automation if a step fails.

## Power Automate Desktop

Power Automate desktop flows are for automation where there is no connector or API available to use. There are many legacy applications that do not have a method for accessing their data or functionality except through their user interface.

Power Automate Desktop flows use Robotic Process Automation (RPA) techniques to automate user actions on these legacy applications. These techniques require software to be installed on the computers where the applications are installed. The software required to create, edit, and run desktop flows is called Power Automate Desktop.

Whether working with modern or legacy apps, on-premises or in the cloud, Power Automate Desktop can automate rule-based UI tasks by recording mouse clicks, keyboard inputs, and data entry. As well as entering data, Power Automate Desktop can also extract information from the application user interface. Once the actions have been recorded, Power Automate Desktop has a rich editor for adjusting these user actions to create complex desktop flows.



Power Automate desktop flows can be run attended with the user manually initiating the flow, or unattended with desktop flows running on Virtual Machines in Azure.

Power Automate desktop flows are a valid way to perform integrations and automation when there is no other alternative, or when developing an integration would be expensive and time consuming.

[!NOTE]

Power Automate cloud flows are explained in the Power Automate Architecture module.

## Desktop software

Unlike the rest of the Microsoft Power Platform tools, Desktop flows require software to be installed on your local computer. You need this software because you need to interact with software applications that are not accessible from the cloud.

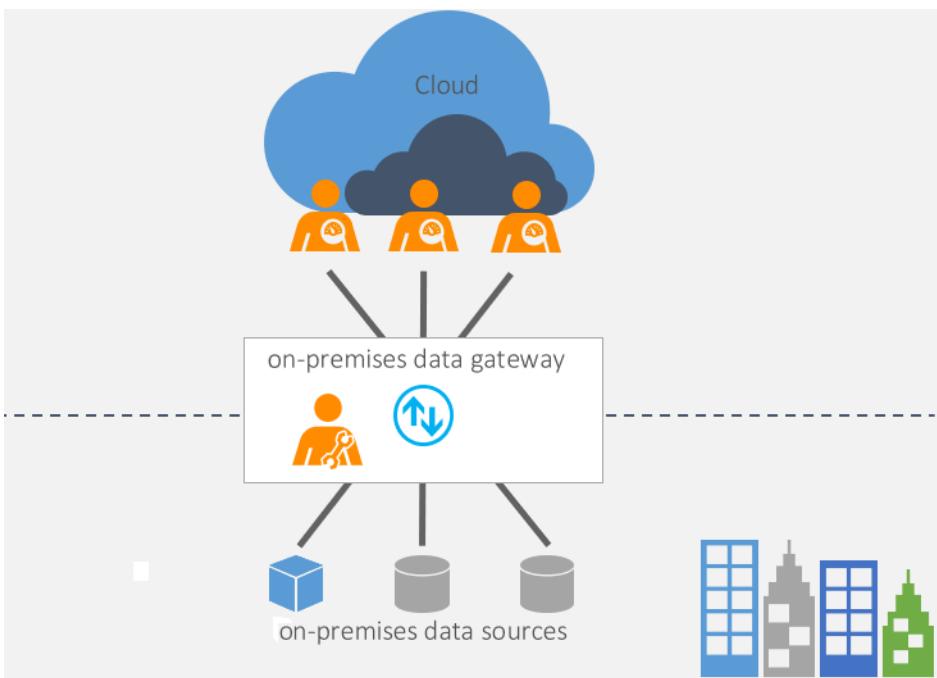
The Power Automate Desktop app is a Windows application that you use to create, edit, and run desktop flows. You can download this tool from <<https://go.microsoft.com/fwlink/?linkid=2102613>> or from the Power Automate portal. Microsoft has announced that Power Automate Desktop will, in future, be included with Windows 10 and is free to use on a Windows 10 computer.

You will need a computer running Windows 10, Windows Server 2016, or Windows Server 2019 to install Power Automate Desktop.

[!NOTE]

For unattended flows Windows 10 Home is insufficient and Windows 10 Pro or Windows 10 Enterprise is required.

To initiate a desktop flow from the cloud, a second application is required to be installed on the computer where Power Automate Desktop is installed. This is the On-premises data gateway. This tool allows desktop flows to be triggered by a Power Automate cloud flow. You can download a gateway from <https://docs.microsoft.com/data-integration/gateway/service-gateway-install> or from the Power Automate portal.



As a solution architect, you will need to liaise with the IT department to arrange for these software applications to be installed. Local software installation can often take much longer to plan and implement than you may think due to IT policies and procedures. If you are planning to use RPA in your solution, you should look to address the deployment of this software as early as possible.

## Browser software

To interact with web applications, a modern web browser is required. The latest version of the Google Chrome browser, Microsoft Edge, and Mozilla Firefox are supported.

A browser extension is required to be installed. These can be installed by the Power Automate Desktop installer but may require additional configuration. Again, the solution architect will need to liaise with the IT department as installing browser extensions maybe blocked by corporate IT policies.

There are some further settings for browsers that should be configured. These are described in <https://docs.microsoft.com/power-automate/desktop-flows/using-browsers> and the solution architect will need to arrange for these settings to be applied.

## Requirements

Power Automate Desktop requires access to a Microsoft Power Platform environment that has a Data-verse database. You may need to create a database for the default environment if one does not already exist.

## Solutions

Desktop flows are solution aware and can be included in solutions and application lifecycle management processes.

You should make use of Environment variables for any property used in a Desktop flow that might vary between environments.

## Deployment

Post deployment of a desktop flow via a solution there are manual tasks that may need to be performed:

- Gateways: Configure desktop flows to connect using the gateway(s) for the environment
- Environment variables: Set the values for the environment.
- End-user authentication: Configure end-user authentication so the desktop flow can take actions on the user's behalf.

The solution architect should ensure that these steps are included in the deployment plan for the solution.

## Recording and editing tasks

Power Automate Desktop can both capture the steps performed by the user and edit the steps in desktop flows.

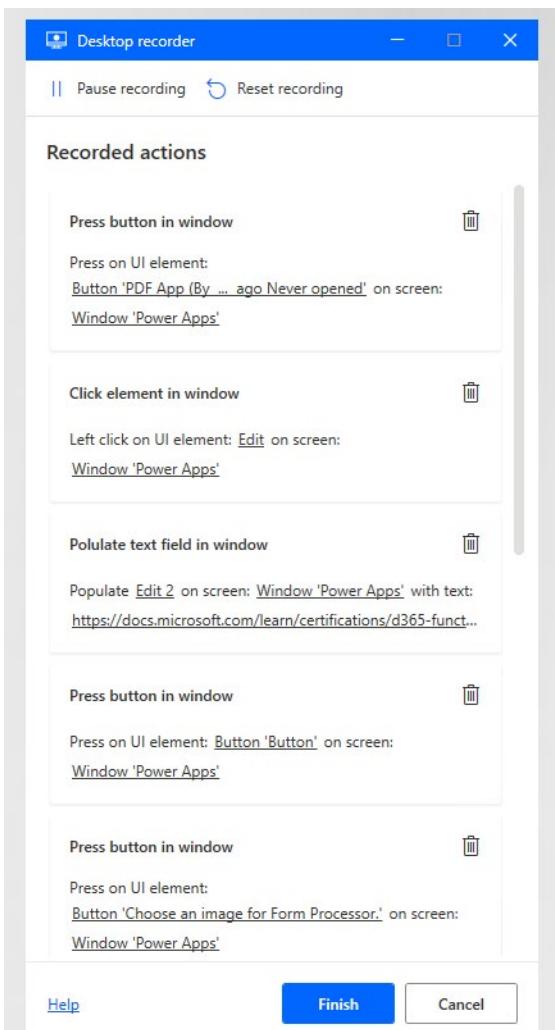
the user can record tasks with

- Desktop recorder: Records the steps on Windows native applications.
- Web recorder: Records steps in a browser for web applications.



## Desktop recorder

The Desktop recorder identifies and highlights objects in the application user interface such as buttons, menus, and fields. It then records the mouse clicks and keyboard entries against that object.

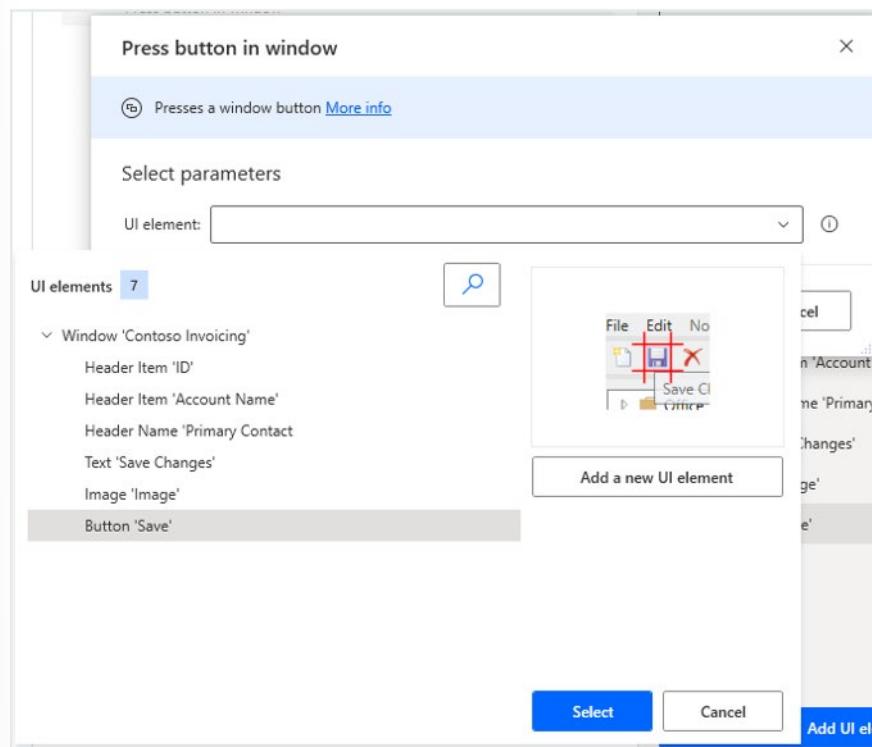


## Web recorder

The web records identifies what is clicked and entered into a web page. The process is similar and is tailored towards web applications.

## Editing steps and actions

Once steps have been recorded, you can edit to remove unnecessary actions and add further actions. You can also edit the objects selected in the application and add new UI objects.



Power Automate has a very wide range of capabilities and can perform many actions including:

- Running SQL queries against a SQL Server database.
- Manipulate the contents of an Excel worksheet.
- Process emails in Outlook.
- Run Powershell scripts on the local computer
- Copy to and from the clipboard
- Access Active Directory objects
- Access cloud services

Power Automate Desktop is very powerful with many possible actions. The solution architect should decide the most appropriate technology for performing actions. For example, should a SQL query be performed using a desktop flow, a cloud flow, or by an Azure function.

## Variables

Flow variables allow information to change in every run of a desktop flow. If you are using a desktop flow for any type of data entry, you will need flow variables to define the fields that you need to enter. The solution architect should encourage the use of flow variables and ensure that there is a naming convention for variables.

Input variables are information that you pass to a desktop flow. Output variables are outcomes that the desktop flow passed back after the desktop flow has run. Power Automate Desktop provides the ability to receive input values from cloud flows and return values back to the cloud flow by using output variables. As a result, your automations can be seamlessly integrated. The solution architect should ensure that the input and output variables are specified at the design stage as it is likely that cloud and desktop flows will be built by different makers.

If input variables are different for each environment, then you should define Environment Variables for input variables in the flows.

## Running Desktop flows

Desktop flows can be run in one of two modes:

- Attended
- Unattended



### Attended

Attended flows are aimed at automating individual tasks across desktop and web applications. The user can interact with the desktop flow if required, for example if a decision is required.

Attended flows are initiated on demand. A user can start an attended desktop flow from their local computer.

The user must be logged in to run an attended desktop flow.

### Unattended

Unattended desktop flows are used for high volume automation where no interaction is required.

Unattended desktop flows are initiated from Power Automate cloud flows. The cloud flow sets the input variables and receives the output variables.

[!IMPORTANT]

If a user is logged in, an unattended desktop flow cannot run.

## Virtual machines

One option for running unattended desktop flows is to use the user's computers when they are not working e.g., overnight or at weekends. This does require that the user actually logs off and that the software is configured the same on each of the teams' computers. This is not always the case.

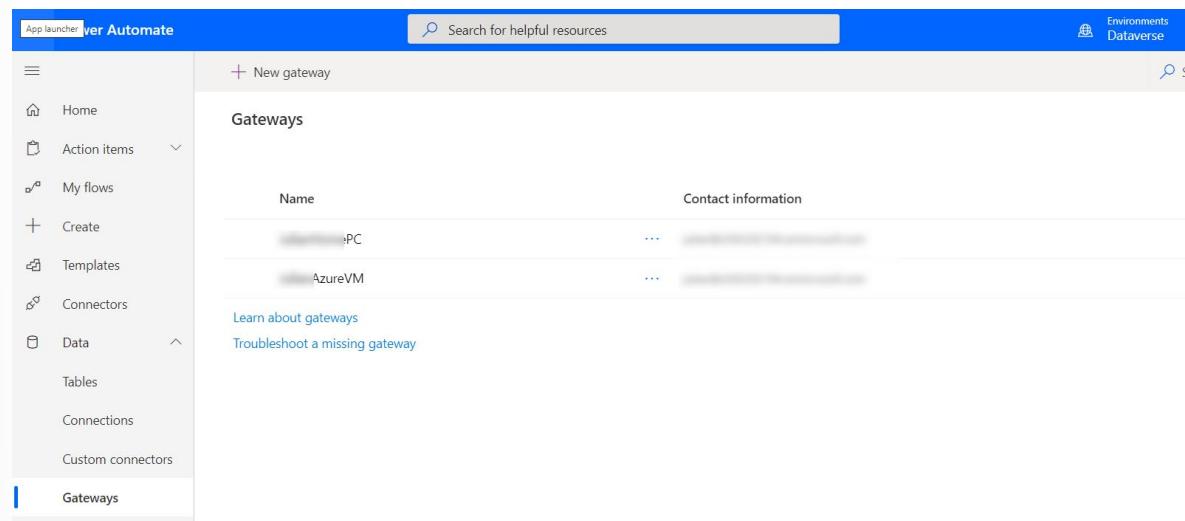
A more robust option is to create Virtual Machines in Azure. This has several advantages:

- The software configuration can be consistent across the machines.
- The number of Virtual machines can be scaled as necessary.
- The desktop flows can run during working hours.

If using virtual machines, the solution architect will need to define the specification for the virtual machines and determine the costs of running the virtual machines. The solution architect should ensure that auto-scaling rules are defined to reduce costs when the virtual machines are not required and to maximize hardware productivity.

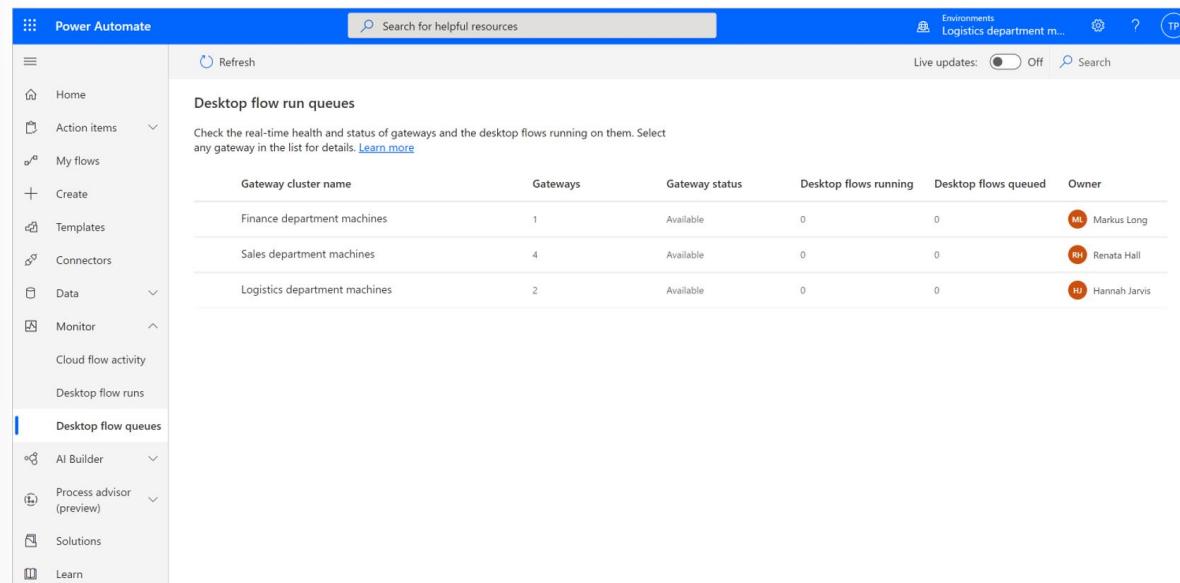
## On-premises data gateway

An on-premises data gateway must be installed on each computer to allow the computer to take part in an unattended desktop flow. On-premise data gateways once installed, are managed from the cloud.



The screenshot shows the Power Automate interface with the 'Gateways' section selected in the left navigation bar. The main area displays a table of gateways with columns for 'Name' and 'Contact information'. Two entries are listed: 'HP PC' and 'AzureVM'. Below the table are links for 'Learn about gateways' and 'Troubleshoot a missing gateway'.

Gateways can be grouped into clusters that will run instances of a desktop flow.



The screenshot shows the Power Automate interface with the 'Desktop flow run queues' section selected in the left navigation bar. The main area displays a table of gateway clusters with columns for 'Gateway cluster name', 'Gateways', 'Gateway status', 'Desktop flows running', 'Desktop flows queued', and 'Owner'. Three clusters are listed: 'Finance department machines' (1 gateway, 0 flows), 'Sales department machines' (4 gateways, 0 flows), and 'Logistics department machines' (2 gateways, 0 flows). Each row includes a small profile icon for the owner.

The solution architect should define the clusters that are required and the priorities for desktop flow jobs.

## Process advisor

Process advisor is a tool that records and analyzes your business tasks. Process advisor is a process mining tool that identifies bottlenecks in business processes. You use this tool to identify opportunities for automation and optimization.

Process advisor allows you to:

- See improvement opportunities.
- Track important metrics.

- Create work efficiencies.

Process advisor is a tool the solution architect should consider using when gathering requirements and when mapping business processes.

## Record processes

Process advisor uses Power Automate Desktop to record business tasks. Users record the tasks they perform.

You can view the actions and must group actions into activities.

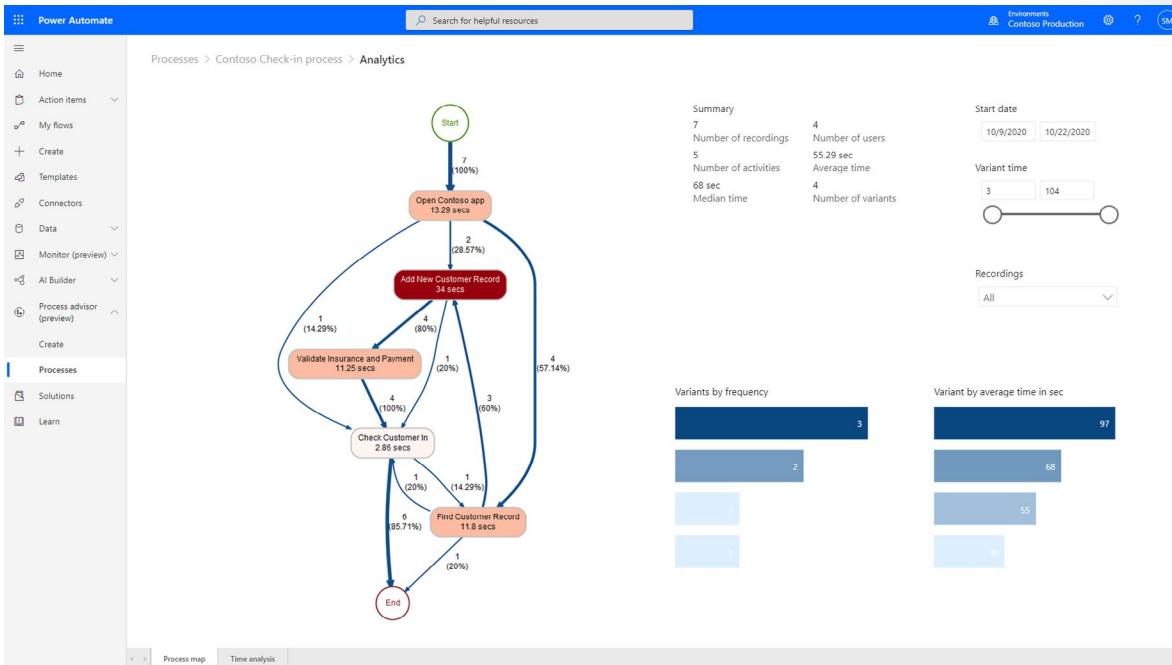
## Analyze processes

Process advisor then analyzes each of the activities and identifies the frequency of paths through the process and the variants.

Analyze discovers inefficiencies and shows optimization and automation opportunities. Process advisor automatically generates a process map that visualizes the activities process.

## Visualize processes

In the process map you can see which activities take the longest, how many variations of the there are, and what variations and actions take the most time.



Using this information, you can identify which automations are required and the type of automation to create.

# Module Summary

## Summary

The solution architect should understand how to use RPA to create end-to-end automation across web, legacy, and on-premise systems. Power Automate Desktop enable more options for automating and integrating solutions and its use should be carefully evaluated by the solution architect.

## Next steps

There are several learning paths in Microsoft Learn for Power Automate Desktop including:

- Automate processes with Robotic Process Automation and Power Automate Desktop <https://docs.microsoft.com/learn/patterns/work-automation-flow/>
- Work with different technologies in Power Automate Desktop <https://docs.microsoft.com/learn/patterns/pad-work-different-technologies/>
- Build expertise with Power Automate Desktop <https://docs.microsoft.com/learn/patterns/pad-build-expertise/>

You should also review the Power Automate guidance documentation <https://docs.microsoft.com/power-automate/guidance/>.

# Module 15 Testing and go live

## Prepare for testing and go live

### Introduction

A solution architect's work continues after the system has been designed. Their next task is to ensure that the system is deployed and used by real users.

### Solution architect role in testing and go-live

Typically, the solution architect is one of the people who knows best how the solution works and can guide the test team in how to test it.

The solution architect has a key role in testing and should:

- Stay involved until the end of testing to ensure success.
- Help educate the testing team on the solution architecture to ensure that they adequately test all components and integrations.
- Triage complex problems that arise during testing, dry runs, and after go-live.
- Participate with the go-live team in planning and implementing the go-live strategy.

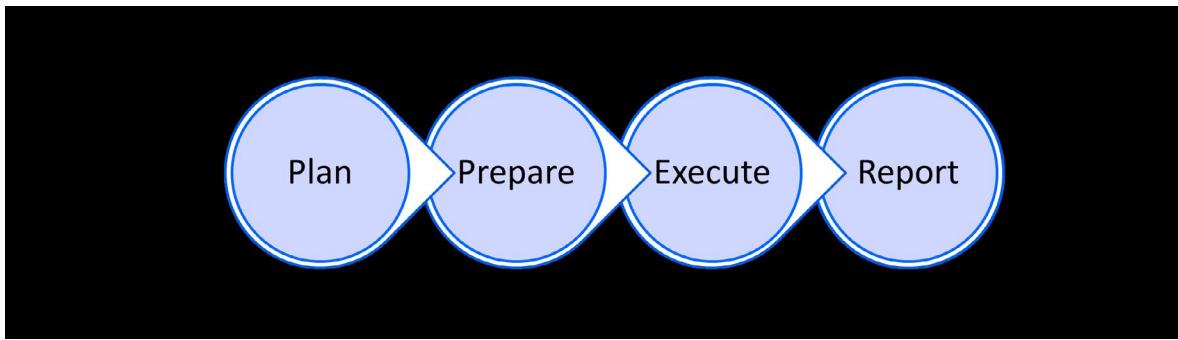
### Overview of testing

Proper testing is important to help ensure the project's success.

[!NOTE]

Testing must be an ongoing effort from the first component that is built until go-live; it can't be a one-time, big exercise.

Testing is more than the process of mapping requirements to functionality. While it's important to build and implement these types of tests, more aspects of a solution should be tested as well. Regardless of the specific metric that is being tested, the process is similar.



The testing process includes the following steps:

- **Plan** - Review the overall test strategy, develop the test plan, and perform needed analysis for baseline metrics. Identify key business scenarios that are in and out of scope. Document the requirements, if that step has not already been completed.
- **Prepare** - Set up needed environments for performance testing, user acceptance testing, and so on. Review data that is received for migration, before and after the migration testing. Validate high-level system requirements, and then develop needed scripts.
- **Execute** - Run test scripts, analyze results, identify potential bottlenecks, and then review failures and behaviors.
- **Report** - Prepare a detailed assessment of the reporting plan, results, and plan of action.

## Types of testing

The solution architect should be part of the discussion regarding the amount and type of testing that is required for a project.

Common types of tests in Microsoft Power Platform include:

- **Unit tests** - Performed by the app builder, business analyst, functional consultant, or developer.
- **Functional tests** - Verify that the implementation meets requirements.
- **Acceptance tests** - Performed by users to give formal approval.
- **Regression tests** - Tests the unchanged functions for regression and are typically performed whenever a system update has occurred.
- **Integration tests** - The goal is for all integrated systems to work in harmony. Integration testing verifies that everything works together, including integrated services and data from other sources.
- **Performance tests** - These tests are verified with expected peak load and peak transaction volume and are typically automated and run before go-live.
- **Migration tests** - Practice data migration to ensure data quality. These tests are performed in close consultation with subject matter experts that know the customer data. These experts should understand the data transition and transformation and can confirm that the migrated data is valid with proper context.
- **Disaster recovery tests** - A disaster recovery plan is useless if it doesn't work.
- **Go-live tests** - Dry runs of the complete solution and go-live process. These tests are typically performed before go-live.

Not all types of testing will be required; it's determined by the size and scope of the project.

## Unit testing

You can use a unit test to check whether a specific function or feature of your app is working correctly. Typically, app makers and developers perform unit testing. Each team member should check their own work before the handoff.

Unit testing is recommended but not required. When you are getting started, or if the amount of code in your solution is relatively small, you might perceive that you are spending more time writing tests than creating the functionality that is included in your solution. The benefits of unit testing begin to accrue when your solution becomes larger and more complex, particularly with server-side development, where you could see significant benefits in debugging locally by using mock or fake data with a testing framework.

Manual testing can be done with all apps, business rules, and plug-ins. Some tests can be automated by using the Microsoft Power Apps Studio and Visual Studio. A popular unit test framework for server-side development is Fake Xrm Easy.

The solution architect should decide on the tools that will be used for unit testing and the level of automation that should be used.

## Integration testing

The solution architect needs to help the test team understand how to test the integrated components.

One advantage of Microsoft Power Platform is its strong integration capabilities. Integration is one of the most important aspects of the business process implementation because it ensures that the implementation functions correctly and has a strong impact on overall adoption.

The solution architect and customer should review the testing scenarios that involve integrations with other line-of-business applications to ensure that end-to-end testing scenarios are created. This review will likely require that the customer plans to have test environments for their other applications.

Each integration will likely have its own test approach, and it needs to be defined. The testing team should be involved early to determine how they'll test each integration scenario. The teams need to ensure that the necessary integrations can be configured to support testing.

A key aspect of integration testing should be focused on the data that flows in and out of the integration. Much of the discussion in the data validation testing section can also apply to the data that is involved in integrations.

Because integration testing can involve other systems, you need to ensure that test environments are used for all components. You don't want integration testing to communicate with a live system and then modify the production data by accident. Occasionally, test scenarios will drive configuration options in the application integration to make it testable. Having the ability to turn off integrations can help you test without invoking the integration.

The process of building integrations is now more approachable and is therefore more accessible to a greater number of people on project teams. Often, integrations might become hidden inside of Power Apps canvas applications or Microsoft Power Automate flows. These hidden integrations can often go unnoticed because the application is using a surface connector from another source. The plan should consider these integrations like any other integration and then test them accordingly.

## User acceptance testing

User acceptance testing (UAT) is performed by users to give formal approval and to test the usability of the system. Acceptance testing is typically performed as a final check before you roll out functionality.

This test is to ensure that what has been built by the makers will match the requirements that are initially requested by the user.

Tips for getting good results from UAT:

- Test with real users.
- Choose users with diversity in terms of IT skill levels. Consequently, you can get various feedback.
- Don't give the user instructions; see whether they can understand the app intuitively.
- Observe how the users navigate the app without assistance, and then determine where you can improve the design.
- When the user is stuck on a screen, ask them to explain their expectation.
- Experiment with different devices to make sure that the test cases behave similarly.
- Ideally, test the app in the user's actual environment or location if the app uses offline capabilities.
- Ask your users to attempt "breaking" your app, such as by entering unusual characters in text fields.
- Users will typically test the "happy path" (the path that a user takes when everything is going perfectly). Ask them to also test scenarios such as canceling an expense report instead of submitting it or denying an expense report instead of approving it.

Users might not be familiar with testing software. Inform them of what type of feedback that you're looking for. It's often helpful to provide a template for bugs to make sure that testers explain exactly what they were doing, what happened, what they expected to happen instead, and relevant information about their testing environment (such as device type and browser).

The solution architect will be required to help triage issues that are raised by the users during testing.

## Security testing

Security testing is important to ensure the security of the application and alignment with regulatory requirements. This testing should include a vulnerability assessment to ensure that the application is secure. It should also include testing from the security context of different types of users to ensure that an appropriate level of data and features are available.

Security testing should also look at the security model inside the application for access to its data and features. Testing should include scenarios with different users who have different roles and access characteristics to ensure that the security model holds up. Security model testing should make sure that over sharing or under sharing isn't happening. One way to approach security model testing is to create a dedicated test user account for each major role set.

The solution architect should help define the level of security testing to perform.

## Planning

Several activities should be performed prior to go-live:

- Performance testing
- Deployment planning
- Risk assessment

## Performance testing

Performance testing helps ensure that the application is performing as designed and can handle the rigors of daily use. Strong performance is vital for user adoption. Users will be reluctant to use an application that takes too long to load pages or go through business processes. Performance testing helps the customer identify if they need to revisit certain customizations and perform tuning activities.

Many customers don't include performance testing to save on cost and effort. Consequently, they will run into user adoption problems quickly after the application goes live. The solution architect needs to make the customer aware of the risks of not doing performance testing.

The results of this testing might require that the solution architect helps plan the remediation steps and guide them through submitting support requests. Performance testing needs to be completed well in advance of the application going live to remediate issues that occurred in performance testing.

Key questions that the solution architect should address for performance testing:

- Do you have a dedicated environment for performance testing?
- Have you identified the required master or reference data for performance testing?
- Have you identified the key business scenarios and the baseline for these scenarios?
- Have you identified the concurrent load for performance testing?
- Have you identified locations to perform latency tests for each location?
- Do you have a plan to populate the required data before performance testing?

The solution architect should:

- Identify potential hotspots in the app that should be performance tested.
- Know what the peak volume might be and always plan for slightly higher.
- Ensure that contractual performance service-level agreements (SLAs) are tested to ensure compliance.

The solution architect should monitor network traffic for the different office locations. In particular, latency and bandwidth need to be checked to ensure that app performance is not adversely impacted by network issues. You can use Microsoft Azure Monitor and Azure App Insights to monitor performance of your apps.

## Deployment planning

Deployment of a solution will go more smoothly with some preliminary planning. The deployment plan consists of many activities to ensure successful deployment of the solution. Deployment plans vary on a case-by-case basis, but a deployment plan can include:

- Environment setup
- Types of testing
- User training
- Data migration
- Rollout strategy
- Support during deployment

Depending on project size, the solution architect might own the deployment plan or act as an advisor to a dedicated planning team. Typically, the solution architect does not create the deployment plan but provides input and review.

The solution architect is often the first call when the customer is unhappy with the progress on deployment.

The solution architect should:

- Ensure that the sequence of events for go-live are in place.
- Look for risks consistently and have an alternate plan.
- Ensure that the team is in place to support the deployment.

## Risk assessment

The solution architect knows the system better than anyone; therefore, they should perform their own risk assessment for go-live. The solution architect should examine the system from top to bottom and then consider the following questions:

- What could break?
- What might not work as designed?
- What if the other system goes down?
- Do we have the proper deployment sequence?

A solution architect should always plan for the worst and celebrate later when it doesn't happen.

## Data migration

Data is important when you are deploying Microsoft Power Platform solutions. Your users can't help customers when no data is in the system. Typically, when you are replacing another business application with a solution based on Microsoft Dataverse, some data is migrated into the new system during the deployment so that users can see relevant business data when they start using the application.

## Data migration approaches and tools

Multiple types of tools and approaches are available for you to use when migrating data to Microsoft Dataverse. The following list is not exhaustive, but it includes some of the most common options:

- **Get data from Excel** - In the maker portal, you can select the **Get data** option when viewing a table and then import data from a Microsoft Excel or a .csv file.
- **Legacy data import utility** - You can import data to tables from .csv, .xls, .xml, and zip files.
- **Microsoft Power Platform dataflows** - In the maker portal, you can select **Dataflows** under the **Data** menu and then configure an import from various cloud data sources. Data from these sources can be transformed prior to import by using Microsoft Power Query.
- **Extract, transform, and load (ETL)** - You can extract, transfer, and load (ETL) software such as Microsoft Azure Data Factory. Data Factory has a connector for Microsoft Dataverse.
- **Outsourced tools** - Use tools from other sources such as Kingswaysoft.
- **Custom applications** - Use custom applications that are developed by using the API.

## When to perform data migration

Make sure that you perform data migration multiple times during a project for the following reasons:

- Developers and functional consultants will require data when building apps.

- Users will require data for user acceptance testing (UAT).
- Testers will require data for integration and performance testing.
- Subject matter experts require data for data validation testing.

## Sample data

Data migration will depend on your ability to get data from the source system. Often, this process can be delayed, so it might be necessary for you to create sample data for use in early development and testing activities.

## Test data migration

Data validation is a critical part of assessing the successful completion of data migration. Two aspects of data validation testing are data quantity and data quality. Data quantity refers to the number of rows that were expected to be created in Dataverse, and data quality is the correctness of data values as defined by data mapping.

Data quantity might not be a one-to-one match of records between the source environment and the test environment, especially for transaction data because the data model design might be different in the target environment. As another example, some data quality issues are systemic with customer's data. If a company has many different customer records for the same organization and you are integrating with their financial system, you're limited in how much you can improve their data. The testing team needs to be aware of similar issues to be effective at implementing their testing plan.

Data, such as record ownership and the **Created On/By** and **Modified On/By** information, will likely differ. Likely, this factor will be affected by users who no longer exist and different ownership. For that reason, you can't do a straight table compare. However, testing should validate that this situation is being handled correctly and is mapped as expected.

The expectations for data quantity and quality should be clearly identified as part of the data migration strategy and testing strategy. Similarly, the data values might not show up as exactly the same value due to the difference in the new schema or due to applied transactional logic. Key business users must be engaged in this testing activity, especially for quality. Customers should also consider using a separate environment for testing data rather than mixing it with UAT testing because it risks UAT users modifying the migrated data. The number of runs must also be planned with an initial small subset and then a significant 70 to 80 percent of full volume to ensure the correctness of migrated data.

Data validation testing is important to ensure that the data migration is completed properly because it will have a significant impact on correct functioning and adoption of the application. It is vital to raise awareness of the importance of data quality that is imported into the production system.

The solution architect needs to identify if the customer is missing certain aspects of this critical activity and then confirm that the customer is planning to validate imported data before going live to ensure data quality.

Key questions to ask yourself during data validation testing:

- Have you identified the scope for data validation?
- Have you identified subject matter experts to perform data validation testing? Does this list include users and managers?
- Have you considered integration/migration with a smaller record set and performance testing?
- Have you determined the strategy, outcome, and schedule for data validation testing?

- Have you identified the number of runs and data volume for each testing cycle?
- Have you defined the criteria for validating data quality and data quantity?
- Have you maintained separate environments for data validation?
- Are key test users aware of the transition logic that is being applied during data migration?

## Prioritize data migration

Don't underestimate the amount of effort that is required for data migration.

Data migration planning should take place immediately after the project has started. Data migration might identify data elements and business processes that have not been captured in the requirements analysis. It is crucial that these omissions are raised with the customer so that decisions regarding what to do with this newly identified data can be taken without jeopardizing the project.

## Go live

The solution architect has a unique role in customer implementations of Microsoft Power Platform solutions. Solution architects are among the key people who are responsible for the success of the deployment.

Go-live is the process through which the Microsoft Power Platform solution becomes operational. This process goes beyond turning on the production environment and the deployment of the solution files; it also includes the onboarding of users. If you are migrating from a legacy customer relationship management system, the process can include the final production data migration, decommissioning of the old system, and post-deployment support of the new system.

To prepare for a successful project go-live, the solution architect should plan and deliver a Go-live Readiness review. The review can be a meeting or a workshop, but the solution architect should prepare a go-live document for review. The review's goal is to evaluate how prepared the customer is for go-live. The Go-live Readiness review should be scheduled and completed before the customer's go-live date, ideally with enough time to make necessary corrections that the review has revealed.

Getting ready for go-live is one of the busiest times of the project, and you should be respectful and understanding of the pressure that the team is probably experiencing and keep the meeting to its primary purpose. Be prepared to participate in the go/no-go decision process criteria and to help guide the customer toward successful criteria to avoid escalations later.

Go-live is one of the most critical milestones during a deployment. The Go-live Readiness meeting will ensure that the customer is ready for a successful go-live and won't need to postpone the go-live date due to unforeseen issues, bad strategy, or project risks that were not previously identified. Specific activities occur during this phase that you'll want to keep track of and follow up on proactively with the customer to avoid surprises. This approach includes activities such as final user acceptance testing, user training, final data migration, installation of apps for Microsoft Outlook and mobile, and migration of the client configuration to production.

## Navigate to go-live

Rarely will you have issues that are unresolved prior to go-live. The solution architect's role is to work with customer to guide them toward a go-live decision with outstanding issues.

The solution architect will need to evaluate the impact of fixing an issue prior to go-live or fixing after go-live. Additionally, the solution architect will need to plan how to resolve outstanding issues during the post go-live support period.

## Common go-live problems

Common go-live problems include:

- No plan for how to roll back if the deployment goes wrong.
- Incorrect assumptions about a user's workstation or network configurations.
- Insufficient real-world testing, which can lead to customizations that don't meet real users' needs or performance issues with real user load.

The solution architect should push to reduce the risk of problems with go-live, for example:

- Look for places to streamline and simplify the plan, such as pre-deploying mobile apps.
- Prioritize data migration so that the important data gets in first.
- Have all users access a dummy production environment so that you can resolve access issues before go-live.
- Determine whether you can run the old system and new system in parallel and then slowly move groups of users.

## Automate go-live

The solution architect should consider automating go-live activities to reduce effort and risks. By using automation, you can script and test go-live activities before going live. For instance, you can consider automating the following parameters:

- User, team, and business unit creation
- Reference data creation and updates
- User configuration
- Data migration deltas

All automation must be well tested.

## Deal with problems

The solution architect is often the first call when issues arise with go-live and in the post go-live period. The solution architect will need to proactively triage, isolate, and simplify issues. Often, Microsoft Power Platform tools can prove beneficial to you when fixing problems by using its built-in features.

Before you make changes, consider the immediate impact and long-term impact of mitigation activities that you apply.

## Module Summary

### Recap

The solution architect has a unique role in the deployment activities of a Microsoft Power Platform solution. The solution architect is one of the people who knows best how the solution works and is among the key people who are responsible for the success of the deployment. The solution architect will be a focal point for the project during the implementation phase.

Often, deployment activities such as testing, data migration, and go-live planning are left for the last minute and are frequently understaffed. The solution architect must champion the need for these efforts to start as early as possible within the project. Solution architects must ensure that they have shared the knowledge and guidance on the solution architecture to effectively plan and implement go-live.

In this module, you reviewed the role of the solution architect in relation to testing, data migration, and go-live, including during the following tasks:

- Performance testing
- Data migration
- Deployment planning
- Go-live