Migration to the Cloud

About this book

I had been working mostly in the on-premise Dynamics environments up until the end of 2018, so my familiarity with the PowerPlatform used to be limited to my personal experiments, and, of course, to whatever I could read/see in various posts and presentations.

However, 2019 has turned things around quite a bit, and I can now say that, even if I'm not an ultimate expert in the PowerPlatform, I can certainly appreciate the opportunities this platform is offering. That said, PowerPlatform is quite different from what we used to know as Dynamics CRM, and I had a chance to experience a lot of those differences first hand.

Therefore, this book is for those who are following the same path from the on-premise to the cloud. Although, it's not an in-depth overview. The way I think of it – it's a set of PowerPlatform-related topics which may require your attention. Some of those will be covered in greater details, and, for others, you will need to do more research on your own.

In either case, as long as it helps you understand the PowerPlatform better, I will consider my mission accomplished.

As of July 2019, this book is still work in progress…

And, of course, I wanted to mention a few folks who provided feedback and occasional contributions(you are welcome to join this list): Ben Thompson, Tarek Hannoudi, Carina Claesson

Have fun!
Alex Shlega, 2019

Last update: 18 Aug 2019

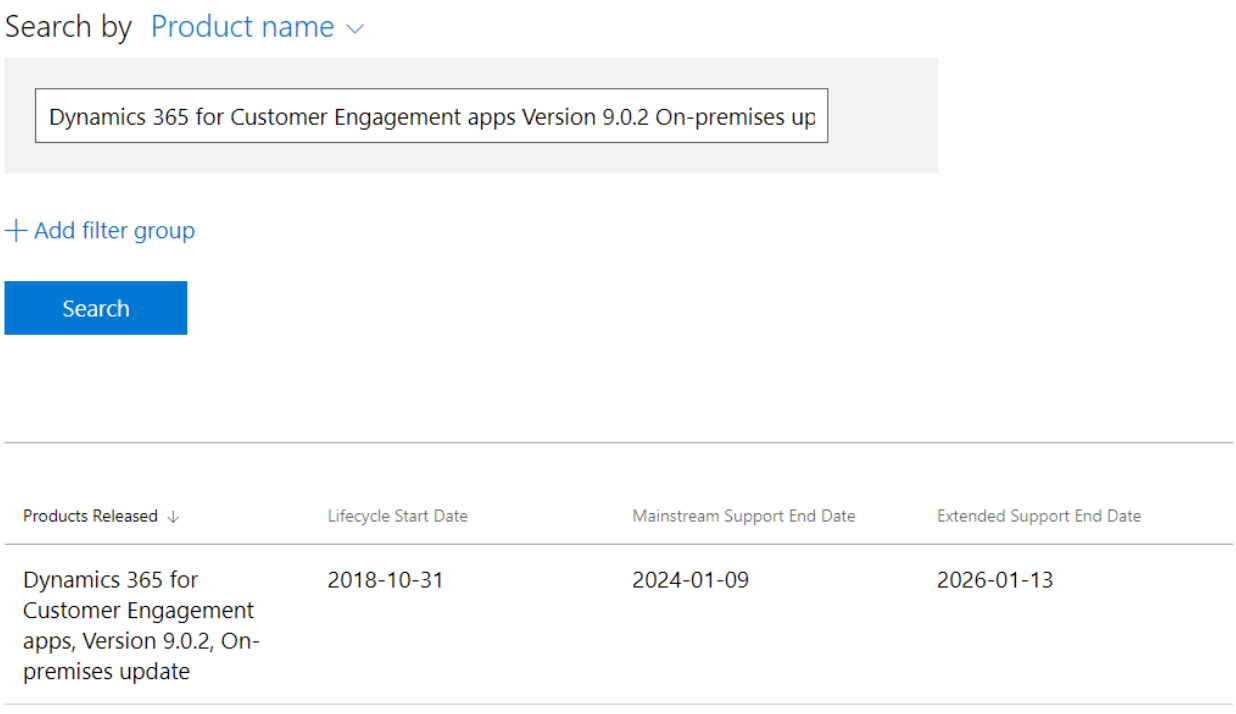
Contents

1. Reasons for moving to the cloud

As I was thinking of what should be the very first chapter in this book, I started to realise that a lot of what I'll be writing about reflects various issues that may happen on the cloud implementations. Compared to the on-premise scenarios, there are changes around licensing, reporting, data access, instance provisioning, etc.

Why would anybody still working on premise really want to move to the cloud? Of course, there is a general idea that cloud is where all the projects should be delivered in the future, but it usually helps to have more solid reasons when undertaking such projects. After all, so far Microsoft has been diligently providing updates to the on-premise version, and, even if those updates are not as frequent as they are in the cloud, it does not look like on-premise version of Dynamics is going to disappear any time soon – mainstream support for the most recent version will continue up until 2024, and, as far as I am aware, it's not yet the last version:

https://support.microsoft.com/en-ca/lifecycle/search?alpha=Dynamics%20365%20for%20Customer%20Engagement%20apps%20Version%209.0.2%20On-premises%20update



Why bother with the cloud?

That's a really interesting question, and I have to be careful not to discourage some of the clients I've been working for on the online implementation projects.

A few years ago, it looked like Microsoft had abandoned on-premise version, and it was not clear at all if we were ever going to see a new major release.

However, on-premise v9 was released in the late 2018, so the release schedule started to look better. Still, when looking at what on-premise version can deliver and what online version can deliver, you will often see that there are lots of additional capabilities in the online version which are not availaable on-premise yet. And they are probably not ever going to be available there.

Therefore, here is my short list of the reasons all Dynamics implementations should be taken to the cloud as soon as possible:

- PowerPlatform gives you much more flexibility than Dynamics ever used to be able to do. There are different licensing plans, there are two different types of applications (which still compliment each other), there are AI capabilities, there are portals, there is integration with Azure and Office 365, there is Sharepoint, there is one drive, there are Teams, there are Office Groups... all of that is integrated – sometimes those integrations work better and, sometimes, they work worse, but they are way ahead of what you can achieve with the on-premise version either way
- From what I can see, Microsoft is not investing into the on-premise anymore beyond what's required to keep on-premise clients from moving somewhere else. Most of the new developments come to the online version much earlier than they show up in the on-premise version. Some of them will never be added to the on-premise version, though, since that version is still running on the old XRM infrastructure. In the cloud, there are new developments, conferences, and announcements happening every week, but I would have a hard time finding a similar list of announcements for the recent changes in the on-premise version:

On-premises data gateway July 2019 update is now available


Arthi Ramasubramanian Iyer, Thursday, July 11, 2019

Features released as part of the July Gateway release.

» Read more

Microsoft Business Applications Summit 2019 – session list quick reference


Pratap Ladhani, Principal Program Manager, Customer & Partner Success, Thursday, July 11, 2019

Refer to this list for all the sessions at the Microsoft Business Applications Summit 2019 in Atlanta related to PowerApps, Microsoft Flow, CDS & AI Builder

» Read more

Automate your application lifecycle management (ALM) with PowerApps Build Tools (Preview)


Per Mikkelsen, Principal Program Manager, Thursday, July 11, 2019

Automate your Application Lifecycle Management with PowerApps Build Tasks for Azure DevOps

» Read more

Environment Canterbury speeds up outcome tracking with the Power Platform


Sameer Bhangar, Principal Program Manager, Wednesday, July 10, 2019

Environment Canterbury (ECan), a part of local government for the Canterbury region in New Zealand, partnered with Datacom Systems to deploy a Power Platform solution that provides a more consistent approach to how all freshwater and natural resources projects are managed and reported on. The solution has served as a proof point for moving ECan from older paper and Excel-based processes into new ways of working with digital technologies.

» Read more

June 2019 updates for Microsoft PowerApps

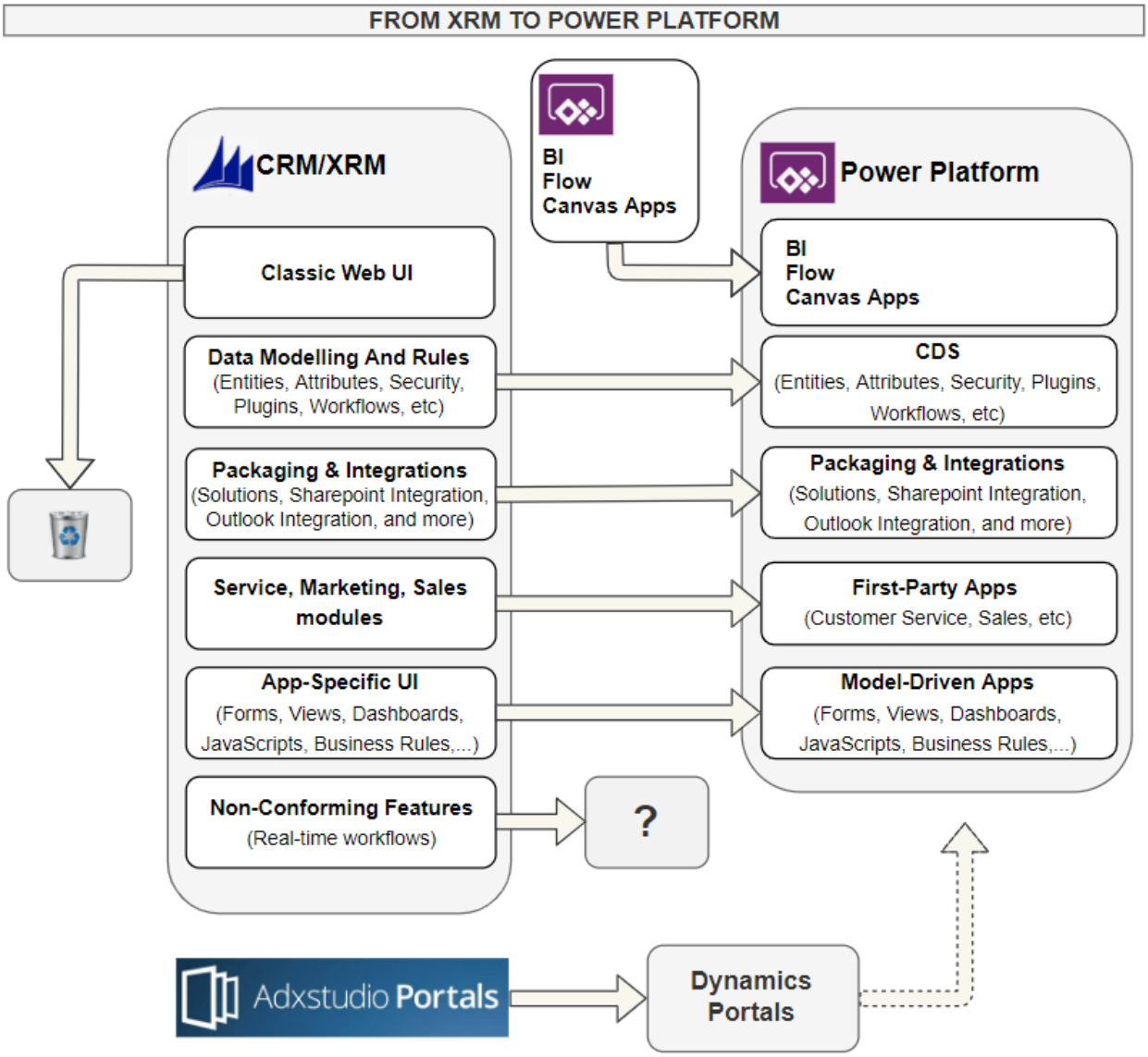
If there is any reason why you might find it difficult to move to the cloud, it's licensing. I will talk about it a few pages later, but pretty much anything in the cloud is "X as a service". And, of

course, services are not delivered for free. Some of those services will be bundled under a license plan. Others will have to be purchased separately.

Unlike in the on-premise world, your licensing costs will likely end up being quite a bit higher in the cloud, though you may be able to save on the infrastructure expenses. Is it worth moving to the cloud then?

Well, given that most of the investments in the Dynamics/PowerPlatform world are happening in the cloud these days, I don't think it would make a lot of sense to stay on-premise in the long term. There is certainly some planning to do, though, and that's where I'm hoping the rest of this book will help you in making more informed decisions.

2. From XRM to Power Platform



What is a Common Data Service?

Common Data Service lets you securely store and manage data that's used by business applications. CDS has inherited most of the capabilities that used to be provided as Dynamics XRM except, maybe, for the user interface (both client and admin).

The concept of CDS includes entities, security roles, business rules, relationships, solutions, etc. Copared to the XRM, CDS has a somewhat different administrative iterface. Also, the whole UI

piece has been separated into what's now called "applications" – there are model-driven applications and there are canvas applications.

Core CDS vs Dynamics CDS

Conceptually, CDS does not have anything to do with Dynamics. It's always been like this with the XRM, too; however, there used to be no this kind of clear separation before. In the new world Dynamics is just one example of the model-driven application built on top of the CDS, though. Actually, there are a few different applications: Customer Service, Sales, Marketing…

Since CDS is now separated from Dynamics, there is no wonder we can build applications on top of CDS which are not related to any aspect of "CRM". Even more, there are two different flavours of such applications – there can be Canvas Applications and Model Driven applications.

The only caveat to keep in mind is that all those first-party Dynamics applications that we can get from Microsoft require a slightly enhanced version of the CDS, and that's what we call Dynamics CDS. All other model-driven apps can be built entirely in the Core CDS.

Those two types of CDS ennvironments are provisioned slightly differently, but, other than that, they are very similar in what we can do with them, how we work with them, etc.

Aside from the ability to get Dynamics applications in the Dynamics CDS, there are two other notable differences. The following two features are currently available in the Dynamics CDS only:

- Dynamics App for Outlook (it's coming to the Core CDS in Wave 2)
- Sharepoint integration

And, if you would like to see some personal thoughts on the subject, here is one(feel free to send me yours):

Carina Claesson, Dynamics 365 and Power Platform Solution Architect

Still a platform for creating solutions to support all kinds of business processes, still an extendible data model and in Model-Driven Apps and the CDS we recognize the former Dynamics xRM parts. However, what once was Dynamics xRM has evolved into something new and exciting and gradually new ways of thinking has been introduced. Among my favourite new concepts is the possibility to easily utilize hundreds of different data sources through the connectors that is provided as well as the extensibility, especially the possibility to create custom connectors and PCF Controls

3. Licensing

Compared to the on-premise version where the usage is licensed, mostly, per product/application, licensing in the cloud will almost always consider cloud resources.

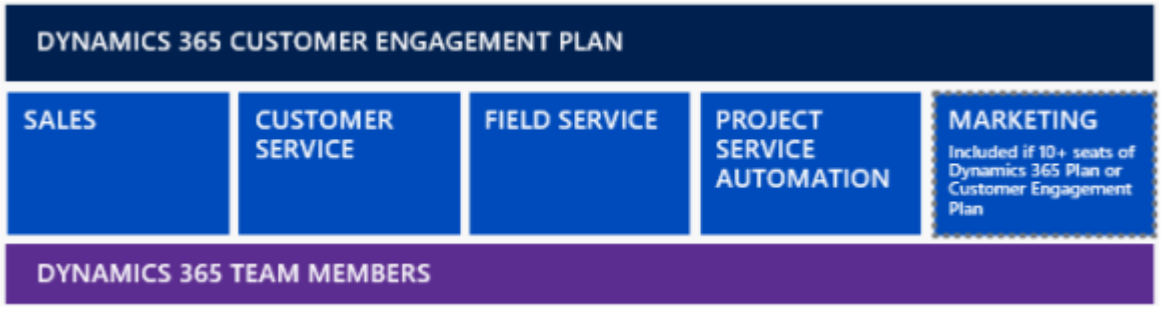
Here are just a few examples of what might be considered a "resource" in the PowerPlatform world:

- Storage space
- Transactions count
- Requests count
- Flow runs
- Execution time
- Premium components

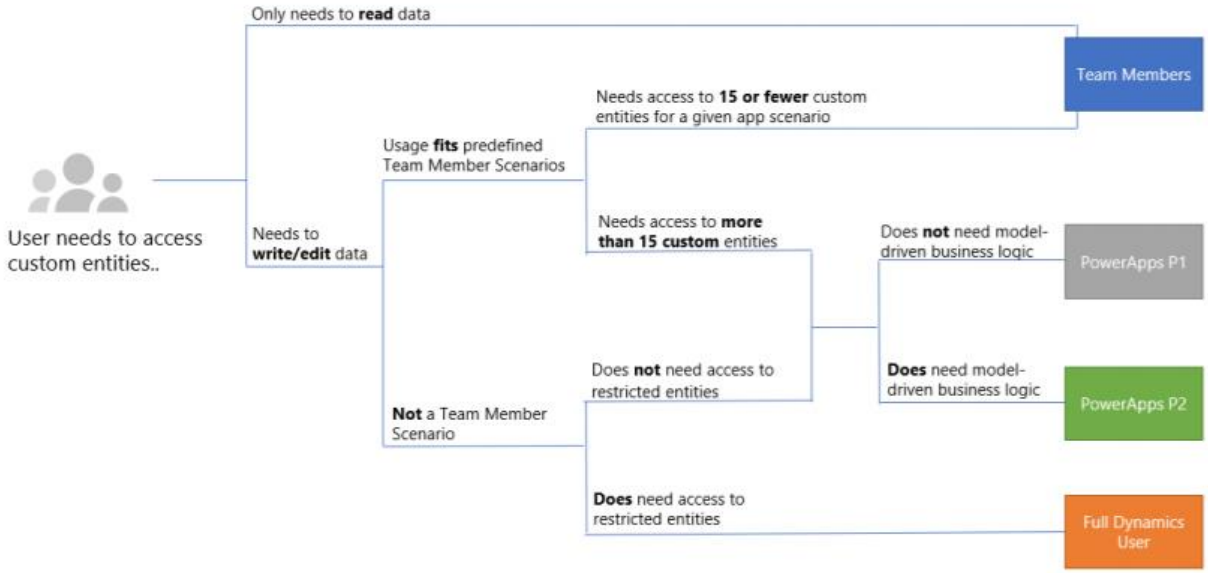
However, there are two different ways to look at the licensing:

- Getting your users licensed to use first-party applications
- Getting your users licensed to use your own custom applications

In the first scenario, the licensing is, really, rather simple. Depending on the application your users need access to, you just have to choose between the following licenses:



Team member licenses, even though they are relatively cheap, have evolved quite a bit and there are some important limitations to keep in mind. Most importantly, they only allow edit access to only 15 custom entities per first-party application, and those entities are supposed to contribute to the main application scenario. Here is a diagram from the licensing guide that may help with the decision making:



Also, team member licenses do not allow access to the custom model-driven applications, so they only make sense in the context of the first-party applications.

On the other hand, there might be users in the environment that won't need access to any of the first-party applications. What it really means is that those users might be accessing either a Core CDS environment, or a Dynamics CDS environment, but they are not supposed to have create/update/delete access to the the restricted entities, since access to those entiites is only granted through one of the Dynamics licenses:

https://docs.microsoft.com/en-us/powerapps/maker/common-data-service/data-platform-restricted-entities

(Of course, there can be no Dynamics apps in the Core CDS, and, hence, restricted entities do not exist in the Core CDS).

Either way, in case with custom applications, there are two licensing options which do not provide create/update/delete access to the Dynamics first-party applications:

- Power Apps Plan 1 (P1)
- Power Apps Plan 2 (P2)

There are a few things to remember about them:

- PowerApps is licensed on a per-user basis.
- User licenses are assigned on a named user basis, each user needs a separate license to run apps.
- PowerApps licenses do not limit the creation of apps.
- Any PowerApps license is eligible to administer apps.

- PowerApps Plan 1 provides access to Common Data Service to store and manage data. Users can run canvas apps that are built on Common Data Service, use premium connectors, access data in custom applications or on-premises data.
- PowerApps Plan 2 allows users to run model-driven apps with code plug-ins and real-time workflows. For details please visit the PowerApps pricing page.
- For details please visit the PowerApps pricing page or download the Licensing guide.

In addition to the standalone plans PowerApps capabilities are also included in certain Office 365 and Dynamics 365 plans allowing customers to extend and customize Office 365 and Dynamics 365 with PowerApps and Microsoft Flow capabilities that these offerings include. Select Dynamics 365 apps and plans include a full PowerApps P2 license, learn more here.

If, after reading so far, you have got a feeling that licensing is complicated, I think you got it right. Brian Illand described this situation in his "License Driven Development" post

Depending on the choosen licensing approach, you might be able to use first-party applications and just build on top of them, or you may have to create a custom application from scratch. You might be able to use a model-driven application (which is what Dynamics used to be), or you may have to stick to Canvas Apps (a completely new flavour of apps for somebody coming out of Dynamics background). You might be able to use Microsoft Flow for process orchestration, or you may find out that your licensing only gives you a limited number of Flow runs per month. You might be able to read from the account entity with Team Member license, but you migth not be able to write there.

Unlike in the on-premise world, licensing strategy is really becoming part of the solution architecture, whether we like it or not.

Just as I was writing this, new licensing model has started to emerge from Microsoft:

https://powerapps.microsoft.com/en-us/blog/new-licensing-options-for-powerapps-and-flow/

On the PowerApps side, here is what's going to happen in a nutshell:

PowerApps plans

New and renewing customers will have the flexibility to choose PowerApps per app or per user plans.

- ***PowerApps per app plan.*** *Allow individual users to run applications for a specific business scenario based on the full capabilities of PowerApps. Pricing will be $10 user/app/month.*

- ***PowerApps per user plan.*** *Equip individual users to run an unlimited number of apps, without any feature restrictions. This takes the place of existing P1 and P2 plans and will be $40 /user / month.*

  PowerApps users will continue to be able to run any flow that is triggered directly from an app, or from the data that app updates. However, the full standalone capabilities of Microsoft Flow will be reserved for the standalone Flow plans outlined below.

  Come to think of it, $10 user/app/month plan is offering something we have not seen in the model-driven applications world so far. Which is… inexpensive license plan. Although, if you look closer, there are a couple of things to keep in mind(again, at the moment of writing):

  - Minimum purchase requirement (30 users)
  - 1000 daily API calls allotment per licensed user

  That license plan is not, yet, available – it's what is coming in October, so we'll see whether this is final or if there will be any amendments before then.

4. The Data

There are multiple datacenters around the world, so, from the regulations perspective it may be important to understand where your data is located.

As per the https://azure.microsoft.com/en-ca/global-infrastructure/regions/:

- A region is a set of datacenters deployed within a latency-defined perimeter and connected through a dedicated regional low-latency network.

- A geography is a discrete market, typically containing two or more regions, that preserves data residency and compliance boundaries.

Microsoft splits data into a few different categories

https://www.microsoft.com/en-us/trust-center/privacy/customer-data-definitions

Namely, there can be:

- Administrator data
- Customer data
- Customer content
- Object metadata
- Payment data
- Personal data
- Support and consulting data

From the data storage perspective, customer data and personal data are two data categories for which may require special consideration.

Customer data is all data, including text, sound, video, or image files and software, that you provide to Microsoft or that is provided on your behalf through your use of Microsoft enterprise online services, excluding Microsoft Professional Services. For example, it includes data that you upload for storage or processing, as well as applications that you upload for distribution through a Microsoft enterprise cloud service.

Personal data means any information relating to an identified or identifiable natural person. In other words, personal data is any data that is associated with a specific person. Personal data provided by our customers through their use of the service, such as the names and contact information of customer end users, **would also be customer data**. But personal data could also include certain data that is not customer data, such as the user id our service assigns to each user; such personal data is considered pseudonymous because it alone cannot identify the individual.

As far as Dynamics 365 goes, there are a few different geos and corresponding regions where customer data can be stored **at rest** (https://www.microsoft.com/en-us/TrustCenter/Privacy/dynamics365-customer-service):

Australia	Australia East (New South Wales) Australia Southeast (Victoria)
Asia Pacific	East Asia (Hong Kong) Southeast Asia (Singapore)
South America	Brazil South (São Paulo State)1
Canada	Canada Central (Toronto) Canada East (Québec City)
Europe	North Europe (Ireland) West Europe (Netherlands)
India	Central India (Pune) South India (Chennai)
Japan	Japan East (Tokyo, Saitama) Japan West (Osaka)
North America	Quincy, WA Chicago, IL San Antonio, TX Des Moines, IA Blue Ridge, VA Cheyenne, WY
United Kingdom (UK)	UK South (London) UK West (Cardiff, Durham)

Although, there are some situations when customer data can be transferred to other locations. That includes customer support scenarios, machine learning cognitive services, etc.

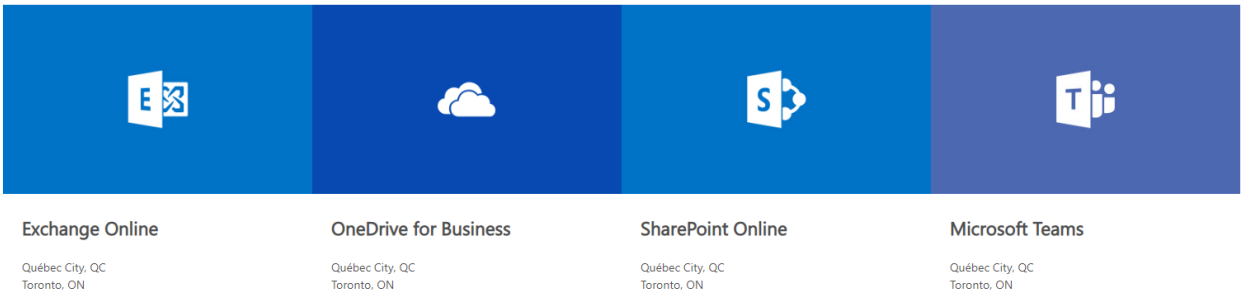
What does "at rest" means? That's, actually, when/where the data is stored. Not where it might be cached in memory if the user is accessing that data from another data region, for example.

It's worth noting that you can also find where certain types of customer data are located at rest for specific Office 365 services (other than Dynamics/PowerApps) using the page below:

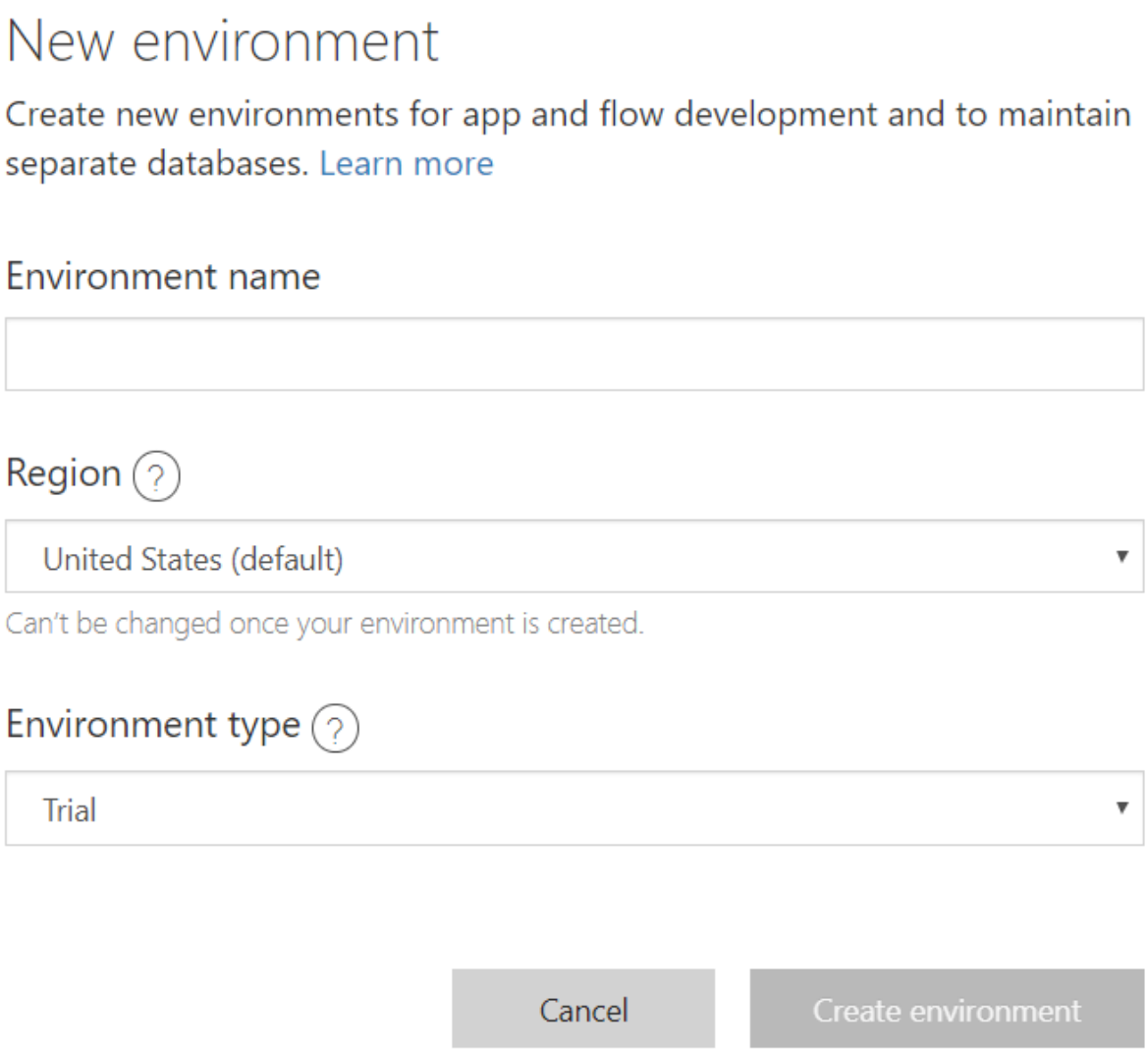
https://products.office.com/en-us/where-is-your-data-located?geo=Canada#Canada

New Office 365 tenants are defaulted to a data center geography (Geo) based on the country of the transaction associated with that tenant's first subscription.

Here is an example for Exchange Online, OneDrive for Business, SharePoint Online, and Microsoft Teams:



PowerApps environments are not listed on the page above, though. Instead, when creating a PowerApps environment, we can choose the region:



Also, when looking at the list of environments in the powerapps admin centre, we can see which region each environment belongs to:

TYPE	REGION	CREATED BY
Sandbox	Canada	SYSTEM
Production	Canada	SYSTEM
Trial (1 day)	Canada	Alex Shlega
Trial (24 days)	Preview (United States)	Alex Shlega
Developer	United States	SYSTEM
Default	United States	SYSTEM
Production	United States	SYSTEM

5. Document Management (Sharepoint, Groups, OneDrive)

Model-driven PowerApps are all about well-structured data. There is an entity model, and there is a UI built around that entity model.

For quite a while, Dynamics used to have Sharepoint integration. In the earlier versions, it used to be Sharepoint list component. Then it became Sharepoint server-side integration. In the current version, there is, also, OneDrive integration and Office Groups integration.

Although, from the document storage perspective, all those options require SharePoint Online.

Here is what you may want to keep in mind when considering those options:

- In general, it's always been better to store documents outside of the Dynamics database, and it's still the same with PowerApps. Although, since CDS File Storage is priced differently now, and it's, actually, a different storage, I would not be surprised if there are viable scenarios where you can store documents in the notes entity attachments without having to pick up huge price tag and/or without having to deal with performance degradation
- Sharepoint storage is better suited for storing documents related to the individual records
- Sharepoint security won't duplicate Dynamics security as is – you'll still need tools like SharePoint Permissions Replicator if that's an issue
- Groups are more suitable for team collaboration. You would not want to create a separate group for each record, even though you might. If you do, those groups will be showing up everywhere, not just in Dynamics, and you'll soon might be wishing you had never done it
- OneDrive is better for storing personal and/or sensitive files that have to be shared on a "need to know" basis

Note: Occasionally, you will hear that there is no technical difference between Dynamics CDS and Core CDS. And, most of the times, this statemet holds. However, there are exceptions. For example, at the time of writing this, Sharepoint integration is not available in the Core CDS environments.

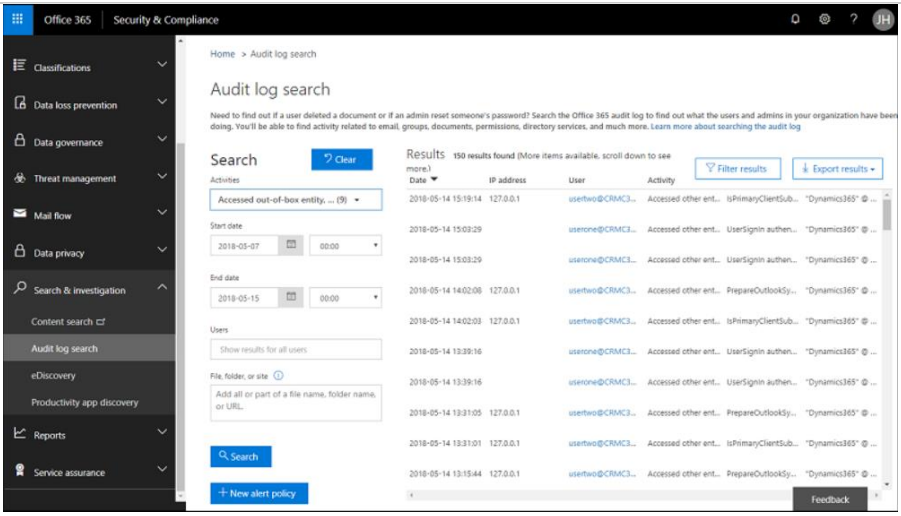
6. Technology mapping

PowerPlatform comes with a number of new concepts and technologies. The table below will help to understand how certain on-premise technologies have evolved in the cloud.

Classic Workflows	These are still available in the cloud. However, in there are a few things to keep in mind: • Eventually, Micrsoft Flow is meant to replace classic workflows, so, when designing solutions for Power Platform, the team should start building experience around the Flows • At the moment, there is no Microsoft Flow alternative to the real-time workflows • Other than Microsoft Flow, there are at least two new other cloud technologies which might be useful for the process orchestration: Azure Functions and Logic Apps. However, Flows are integrated with the Power Platform on the solution level (solution designer can add Flows to the PowerPlatform/Dynamics solutions), and, so, even though the other two might be better suited for all kinds of integrations, Flows are unique in that they can be distributed as part of solutions
XRM	There is no XRM in the PowerPlatform – instead, there are a few concepts which, together, cover XRM and offer additional capabilities. • CDS (common data service): This concept covers entities, security roles, business rules, workflows(not Flows), plugins, Web API, etc • Model-Driven Power Apps: the concept of applications was introduced in Dynamics, but it has evolved. A model-driven power apps is a combination of entities, forms, views, site map, dashboards, business processes, etc. In other words, it's the "presentation part" of the XRM, and it's an evolution of the application component that was first introduced in Dynamics • First-Party Apps: what we used to know as various modules (sales, service, marketing, etc) has evolved into first-party model-driven apps. The tricky part about those apps is that they require a little special instance of CDS (Dynamics CDS), but, other than that, those first-party apps serve as a great example of what can actually be developed as a model-driven application. And yes, first-party applications have to be licensed separately through "Dynamics" licensing.
Web Resources	Web Resource are still there. However, there are at least a couple of powerful alternatives to the classic javascript/HTML web resources now: • PowerApps Component Framework is a developer-friendly component development framework which utilizes TypeScript for

	development and which produces PowerPlatform solution files to be deployed into the PowerApps environments. Remember editable grids? PCF is still in the preview only; however, the idea is that a developer could use it to develop a custom component that could, then, be utilized in vaiours forms/views throughout the application in the similar manner we used to be able to choose an editable grid instead of the readonly grids • Embedded Canvas Apps: Canvas Apps represent a different flavour of PowerApps. They are much more UI-focused than model-driven apps, and they can work with different datasources (ex: Excel spreadsheets or SharePoint lists). Canvas Apps can be embedded into the model-driven application forms, thus offering a unique customization option not available in the on-premise world. From that standpoint, Embedded Canvas Apps might be a great solution for accessing non-CDS data from within a model-driven application (what used to be done through plugings / webresources / virtual entities before)
ADX Studio	ADX Studio is no more, of course. Instead, there are Dynamics Portals. There are at least a few things to keep in mind: • There is no access to the server-side code (although, we can use Liquid and Javascript) • There is one portal that comes with the tenant, but we have to purchase additional instances (this may change) • There are certain limits on the number of page views • A separate portal instance is required for every new site. Even though the same Dynamics instance can have multiple sites • For the proper ALM, you may need to purchase additional portal instances (for dev/qa)
SQL Database	Where we were very used to having access to the SQL Database in the on-premise world, things are quite different in the cloud. Essentially, there is no database we can access there. Behind the scene, there is still SQL running somewhere; however, we are not allowed to run SQL queries against that database and/or to analyze database statistics. It's all up to Microsoft to do. Here are a few examples of what it means in practice: • When working with the CDS data, we are limited to using Web API and/or SDK • The above also applies to the SSRS reports, so we can only use FetchXML-based SSRS reports • We can use PowerPlatform solution designer (or, possibly, SDK) to update metadata, but, for instance, we can't go to the database and create an index manually

	For the Dynamics CDS environments, there is a backup and restore process which covers both automated and on-demand backups: https://docs.microsoft.com/en-us/dynamics365/customer-engagement/admin/backup-restore-instances Automated backups are retained for 3 days. On-demand backups are retained for 28 days. For the Core CDS environments, it seems there is no backup and restore at the time of writing this. Also, it might be worth noting that there is a way to get a copy of your data in the Azure SQL Server database by setting up the Data Export Solution to configure the data to be exported (Also see "Data Integrations" below).
Xrm.Page	Client-side scripting has changed in the last two years in the online version. It has not just changed – it has imporved. As in "a lot". You will need to have a closer look at the documentation (here) to see what exactly has changed, but, just to give you and idea of what to expect, here is a short list: • Xrm.Page was deprecated. As of now it's still available, but you will need to start using getFormContext instead • Gone are the times when we needed to use JQuery or HttpContext. There are wrappers in the Xrm.WebApi namespace now that we can use to create, update, or delete records. There are, also, wrappers for retrieveMultiple, executeRequest, etc. Of course, nothing is stopping you from continuing to use HttpRequest, but you don't have to do it anymore • There is a bunch of useful methods in the Xrm.Utility namespace. We can show a progress indicator, we can get entity metadata, we can invoke a custom action, and more And there are some other changes/additions, so do spend some time reviewing the latest documentation
Audit Log	Audit Log has been available in Dynamics for a long time now; and, for just as long, it used to be impossible to audit read access. However, as GDPR came into force, those additional auditing options have become not just desirable, but, rather, required. So, we can, now, enable and use Activity Logging, though we can only do it in the cloud, and only in the production environments:

	 For a more complete description, you will find a lot more details here: https://docs.microsoft.com/en-us/dynamics365/customer-engagement/admin/enable-use-comprehensive-auditing
SOAP	It's been a while since Microsoft has announced the deprecation of SOAP enpoint: The .NET assemblies for the Organization service currently use a 2011 SOAP endpoint which has been deprecated. The SDK assemblies will eventually be migrated to internally use the Web API instead of the 2011 SOAP endpoint. When this happens, any code written using the SDK assemblies will continue to be supported as it will automatically transfer from the 2011 SOAP endpoint to use the Web API. This update will be fully transparent to you; additional details will be published in future SDK releases. However, it might be helpful to understand that it's not the OrganizationService that's been deprecated – it's the SOAP endpoint only. Internally, Web API will keep using organization service code to process the requests. Mind you, that has nothing to do with what our client code should be using, and that's definitely Web API. However, as far plugins go, for instance, they will still be using IOrganizationService interface. Although, the plugins may have to be rebuilt at that point.
Reporting	In the on-premise environments, reporting usually meant some combination of the following: • Advanced Find • Excel export • Dynamics excel spreadsheets • Dynamics views, charts, and dashboards • SSRS (Fetch or SQL based) • External reporting built off the data being frequently exported into an external database through various ETL tools

	In the cloud, all those options above still exist with the exception of SQL-based SSRS reports since, again, we don't have access to the database, so SQL is a no-go. However, Power BI has taken over as the preferred reporting solution in the PowerPlatform world. There are a few important considerations as far as Power BI goes: • PowerPlatform licensing does not cover Power BI. Your users must be licensed for Power BI separately if they want to start building reports there • Power BI reports are not built off the real-time data. Instead, the data is cached on the Power BI server, and it's refreshed with a certain frequency. Which is good since you are not running reports against the operational database anymore, but it might be unexpected at times since you are not getting real-time results • Users can share Power BI reports with each other. That actually means sharing the underlying dataset, too. So, if a system admin creates a Power BI report on all invoices in the system and shares it with a sales person who is only supposed to see invoices for "their" clients, that sales person will actually see all invoices in such a report. In other words, Power BI does respect user security roles, but only at the time of preparing the dataset

7. Capacity Planning

I mentioned it before, and just wanted to emphasize it here: unlike in the on-premise environments, you will soon find that cloud licensing model is often built around resource usage. Even if not everything has been translated into the “resource-based” licensing yet, this is where the PowerPlatform is, likely, moving.

Although, there is a notable example of creative packaging, too. Flows are, essentially, running on the Logic Apps engine. However, while Logic Apps are (or charged for) based on the number of individual actions/connectors executions, Flows are licensed based on the number of Flow runs. Even so, when accessing CDS data, Flows will still be relying on the Web API calls, and there are limits there as per the following page:

API Limits

03/20/2019 • 2 minutes to read • +1

We limit the number of API requests made by each user, per organization instance, within a five minute sliding window. Additionally, we limit the number of concurrent requests that may come in at one time. When one of these limits is exceeded, an exception will be thrown by the platform.

https://docs.microsoft.com/en-us/powerapps/developer/common-data-service/api-limits

There is a bunch of other limits you may need to keep in mind – you will see some of the most important ones below.

CDS Database Storage per licensed user	**PowerApps Plan 1: 20 MB** **PowerApps Plan 2: 250 MB**
CDS File Storage per licensed user	PowerApps Plan 1: N/A PowerApps Plan 2: 2 GB
API call limits	At the time of writing, the exact limits seem to be undisclosed (see the screenshot and related link above for more explanations)
CDS Log per licensed user	N/A
Flow Runs per licensed user per month	PowerApps Plan 1: 4500 PowerApps Plan 2: 15000 PowerApps for Office 365: 2000 PowerApps for Office 365 Enterprise F1: 750
CDS Database Storage per tenant	PowerApps Plan 1: 1 GB PowerApps Plan 2: 10 GB
CDS File Storage per tenant	PowerApps Plan 1: 20 GB PowerApps Plan 2: 20 GB
CDS Log per tenant	PowerApps Plan 1: 2 GB PowerApps Plan 2: 2 GB

The capacities included in the per-user licenses are pooled at the tenant level and, when the tenant's capacity is exhausted, customers may purchase additional capacity through add-on licenses.

The cap on the amount of database capacity that may be earned or purchased per tenant is subject to the technical limit of 4TB/instance, no maximum on file or log capacity.

Also, while the table above is talking about PowerApps P1/P2 plans, here is how it works with the Dynamics licenses:

These Dynamics 365 apps can be customized using PowerApps and Microsoft Flow capabilities

Dynamics 365 for Sales Professional
Dynamics 365 for Customer Service Professional
Dynamics 365 for Talent: Attract
Dynamics 365 for Talent: Onboard
Dynamics 365 Team Member

These Dynamics 365 apps and plans include PowerApps P2

Dynamics 365 Business Central
Dynamics 365 for Sales Enterprise
Dynamics 365 for Customer Service
Dynamics 365 for Field Service
Dynamics 365 for Project Service Automation
Dynamics 365 for Talent
Dynamics 365 for Retail
Dynamics 365 Customer Engagement Plan
Dynamics 365 Unified Operations Plan
Dynamics 365 Plan

Why does it all matter? In the on-premise environments, the question of capacity was never an issue. Hard drives are cheap, there are no API limits, so it was more a question of getting a more powerfull server/setting up load-balancing, etc.

In the cloud, this is a classic example of where we don't need to be concerned with that kind of infrastructure issues anymore; however, it comes at the cost of having to pay for the resource usage. So, where we could easily store email attachments in the Dynamics on-premise database before (and I'm intentionally ignoring related performance issues here), we could try doing it in the online version but it would clearly cost us $2/GB. Which is not, really, too bad. Although, if we also tried to store lots of textual information in the CDS storage, it would cost us $40/GB. To that point, if you are thinking "email attachments", they are actually supposed to be stored in the File Storage automatically now.

Those increments are described in more details here: https://docs.microsoft.com/en-ca/power-platform/admin/pricing-billing-skus#powerapps-for-office-365

8. Cloud Innovations

The purpose of this chapter is to emphasize the point that PowerPlatform is more than re-branded Dynamics-in-the-cloud, so I am going to quickly go over a few elements of the PowerPlatform which are not available on-premise at all.

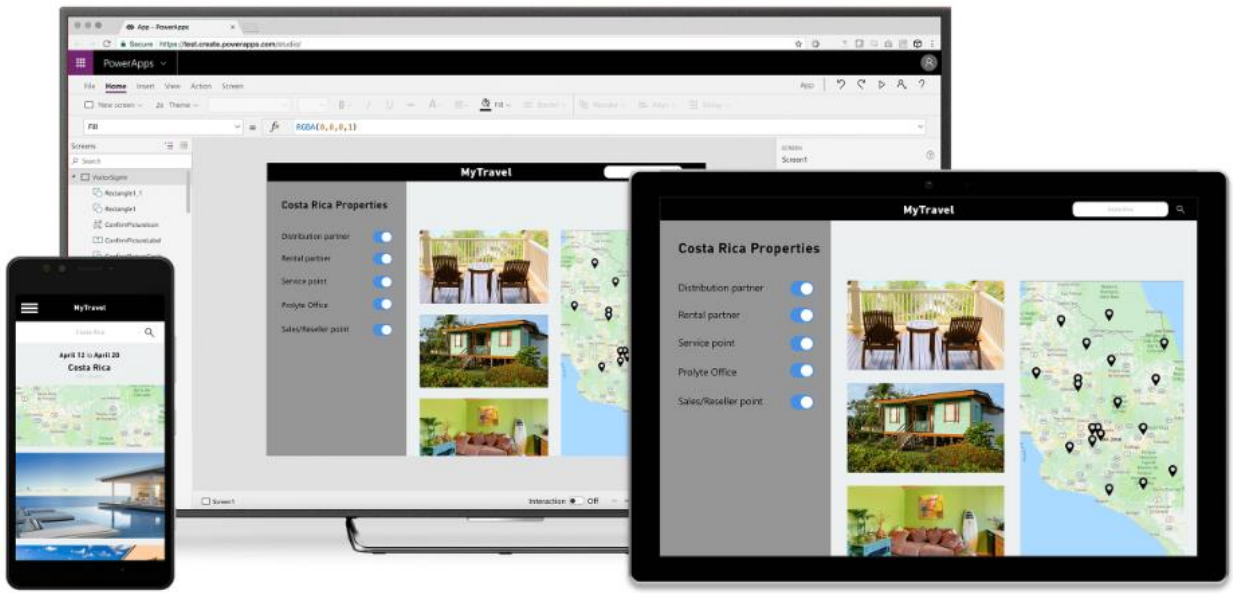
8.1. Canvas Apps

Unlike your classic Dynamics (or Dynamics XRM) applications, Canvas Apps do a couple of things differently:

- They do allow much more specific UI customizations (Color? Font? Location? Size? It's all possible)
- They do bring in the ability to connect to different data sources (Sharepoint? OneDrive? Online Excel Spreadsheet? Apparently CDS is on this list, too)

So, you can build applications that look like this:

https://powerapps.microsoft.com/en-us/build-powerapps/



There are a few ideas around them:

- They are based on the formulas, so creating a CanvasApp may look somewhat familiar if you know how to work with Excel spreadsheets
- They do not require coding

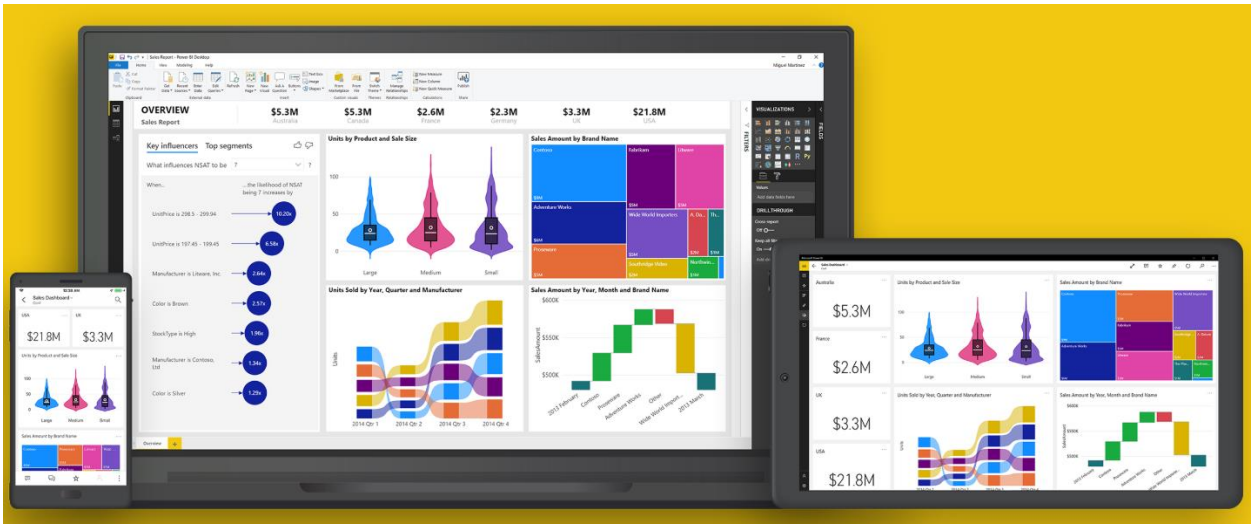
- Considering all the above, a CanvasApp developer does not have to be a professional developer who knows how to code

I would probably agree with most of that; however, there is something to consider:

- CanvasApps do not exist in isolation – unless it's your own personal canvas app that you are never going to share with anybody else, it is still an application that needs application lifecycle management, source control, solution management, etc. Yes, there might be no traditional coding, but all the other dev practices might still be applicable
- That you don't need to know coding does not always mean that it's easier to build an application like that – think of having to dynamically resize all the controls as the screensize changes, for example. PowerApps team has been working to close this kind of gaps, but it's, likely, still work in progress (https://docs.microsoft.com/en-us/business-applications-release-notes/april19/microsoft-powerapps/create-canvas-apps-responsive-layout)

8.2. Power BI

You have probably heard of the PowerBI:



https://powerbi.microsoft.com/en-us/

WOKRING ON THIS

8.3. AI Builder

WOKRING ON THIS

8.4. DevOps

WOKRING ON THIS

9. Cloud Solution Development Strategies

Cloud solution development is not that different from the on-premise solution development. Seriously. We are still dealing with the same concepts:

- Instances (what used to be organizations on-premise)
- Solutions
- Source control
- CI/CD

However, there is more now.

There is a term you will find well explained in the following post:

https://community.dynamics.com/365/b/dynamics365fordevelopers/posts/licence-driven-development-ldd

It's "license-driven development". Why am I making a point of it?

See, depending on how you go about licensing, you will have access to different features within PowerPlatform. Some licensing plans will give you unlimited Flows; other licensing plans will give you unlimited applications; you will need to be licensed for Dynamics to have access to the first-party apps in the Dynamics family, and that includes such entities as cases, for example. Some licneses will limit the number of daily API calls, other licenses will limit the number of applications your users can access, etc.

In other words, you'll need to have a pretty good grasp on the licensing model when designing solutions for PowerPlatform.

What else?

- Number of instances
- Managed vs unmanaged
- More options (Flows, Canvas apps, Azure Functions, Logic Apps)
- Less options (no db access for example)

10. Cloud Migration Options

When thinking of migration to the cloud, it's worth keeping in mind that PowerPlatform is not CRM. You can still use various first-party applications from Microsoft which are built on top of the PowerPlatform and which will give you fmiliar Dynamics CRM experience (of course enhanced and improved now); however, if you had been using Dynamics mostly to develop custom XRM applications before, there might be other options.

Technically, most of the times you should be able to bring over your on-premise solutions to a Dynamics CDS instance. Of course, you'll have to ensure required first-party applications have been installed if there are any dependencies on them, and you may need a license to cover this scenario.

If, however, your solutions are not relying on the first-party applications (or, in the old terminology, on the CRM functionality), you might want to consider re-building your applications from scratch and using a Core CDS instance instead.

The reason I am talking about re-building is that you might not be able to easily migrate your existing solutions from Dynamics CRM to Core CDS because of all the relationships on the first-party components that might have sneaked into the on-premise solution over time. For example, solutions like Activity Feeds (with walls etc) might easily get in the way of this kind of migration since related component would not be available in the Core CDS.

That's why it may have to be a re-built if you want your solution to work in the Core CDS.

Why would you want it to work in the Core CDS, though? From the licensing standpoint, if you are not using first-party entities, it may be cheaper to just go with Dynamics CDS, pay for an enterprise license to get that instance, and, then, license all other users through the PowerApps P2 plan.

That would work just fine, since P2 users would have access to the custom entities. In other words, from the costing perspective, you are looking at having to pay extra annual premium for that enterprise license instead of investing into the re-developmeng of your solution upfront. If your on-premise solution is somewhat complex, this may easily work in favour of using a Dynamics CDS instance.

Your costs might turn out to be different if your on-premise solution is using some of the restricted entities, since P2 would not be enough in those cases:

https://docs.microsoft.com/en-us/powerapps/maker/common-data-service/data-platform-restricted-entities

If that's your situation, you may be looking at licensing everyone with Dynamics licenses which are more expensive than P2, so, when comparing the costs, you may find out that re-building the solution may make more sense now.

There is, also, one more reason you may want to rebuild, and that's if you are building ISV solutions. Some of those solutions can be suitable for both Core CDS and Dynamics CDS; however, if they have dependencies on the first-party entities since they had been developed on-premise originally, that may limit your ability to distribute such solutions to the Core CDS clients.

You might be wondering why have not I mentioned one other scenario – can't you just bring over the database to the cloud and start using it with the most recent version of Dynamics/PowerPlatform?

Technically, it might be possible. You will not be able to do it yourself, but there are certain offerings in the Fast Track (more on the Fast Track in the next chapter) that might help:

https://azure.microsoft.com/en-ca/resources/videos/ignite-2018-dynamics-365-automate-the-migration-of-on-premise-dynamics-crm-to-the-cloud/

Here is a quick summary of what you may need to consider in each scenario:

Rebuild	**As Is**	**As Is with Database**
• Up-front development work • More flexible licensing (can build with the licensing plan in mind) • An opportunity to improve your solution by incorporating new features such as Power BI dashboards, AI, and Canvas Apps • More complicated data migration (new entities, new relationships, new attributes)	• Probably less/none development work up-front • May be limited to only a few licensing options • May still have to make changes to the solution to accommodate changes in the platform (deprecated scripting features, for example) and, also, to adapt your solution to the online requirements (such as sandboxed plugins, fetch-based SSRS reports, etc) • Data migration will still be required, but data mapping should be relatively simple *Note: all of the above really depends on which Dynamics version you are using in on-*	• Same pros and const as "without database" • Might be able to bring over the audit log • Might be able to bring over all data • May have to watch the database size since CDS storage is more expensive • This may be the only viable solution if you do need to migrate lots of data due to the limitations on the WebAPI

	premise. If you are still on Dynamics 2011, for example, then "as is" migration would not be an option at all. You will have to upgrade your on-premise instance first. If you are on 8.2 or 9.0, chances are it may work well.	thoughput imposed by the PowerPlatform

What about the data migration?

Unless you are bringing over the database through the Fast Track/Lifecycle Services, you may still need to figure out what to do with your on-premise data. From this standpoint, the tools you would be using are, likely, the same you had always been using in the on-premise deployment for data integration and/or for data migration:

- SSIS with some kind of CDS/Dynamics connector
- Scribe
- There are probably other ETL tools
- And/or you may have to write custom code in certain cases

Depending on the amount of data you are planning to migrate, you may realize that there is only so much you can really do about the data migration. PowerPlatform will limit your ability to push huge amount of data through the Web API service, and, since it's the only way to get data to the CDS, you may realize that the performance of your data migration scripts is not sufficient to migrate all the data you have. In which case data migration may become even more complicated, since, on the one hand, you may need to meet data retention policies imposed by the business and by various regulations, and, on the other hand, you may have to work within the limits of the PowerPlatform framework.

11. Fast Track

Microsoft is offering a special FastTrack service to the customers who are planning to move to the cloud:

FastTrack is a service provided by Microsoft that helps customers onboard Microsoft Cloud solutions and drive user adoption. Customers with eligible subscriptions to Microsoft 365, Azure or Dynamics 365 can use FastTrack at no additional cost for the life of their subscription.

There are eligibility requirements, so, if you are thinking of using FastTRack, you are supposed to have at least 250 Dynamics Plan or User subscription licenses. FastTrack services include tech talks, workshops, regular touchpoints, and a dedicated FastTrack engineering resource.

I've seen all kinds of Dynamics training workshots/engagements, and a lot of them are very basic, but, if you are eligible for FastTrack, do take advantage of it. Those tech talks are not your regular trainings which are following the same script all the time. They are built more around the specific needs of your organization, and you can use them to get answers, suggestions, and guidliens to the very specific questions and concerns which are unique to your organization.

Here is a quote from the FastTrack FAQ page:

The FastTrack engagement is delivered remotely by FastTrack engineers and architects in your time zone who are part of the broader Dynamics 365 engineering organization. The service is delivered largely through the following types of interactions:

Technical talks: Customized technical briefings on a select set of topics, designed to help customers optimize the value of the solution, as well as to address commonly asked questions. Example topics include data migration, solution development, and tenant administration.

Workshops: Highly interactive sessions providing best practices and guidance based on the context and unique needs of customer projects.

Regular touchpoints: Regular sync meetings to check in with customers and partners on project status and progress and to discover blocking issues and help address them in a timely manner.

Thank you for reading!

I'll be happy to hear your feedback, notes, suggestions, comments, or, well, complaints if you have any about the content of this book. There are at least a few ways you can get in touch:

Email: ashlega@gmail.com

Blog: www.itaintboring.com

Linkedin: https://www.linkedin.com/in/alexandershlega/

Wish you all the best and hope to see you around in the community!