First Week Content

Introduction of Microsoft Dynamics 365 Customer Engagement

Dynamics 365's customer engagement applications are customer focused business 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. Our 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.*

Why Microsoft Dynamics 365?

Microsoft Dynamics 365 is an innovative cloud-based enterprise software solution that brings ERP, CRM, and supporting business applications together as one streamlined product for a modern, simple, and unified experience. It helps run your entire business and deliver greater results by connecting products/services with people and data.

The real-time, predictive, and AI-driven insights that you derive will help you meet your business goals in sales, marketing, operations, services, and more. Microsoft Dynamics 365 is designed to provide you maximum flexibility and extensibility and can adapt to fit almost any business.

Define customer engagement

A customer is an individual or company that receives, consumes, or buys a product or service. The primary goal of a commercial enterprise is to attract customers and maintain a strong relationship with them over-time, by ensuring they want to continue to do business with your organization.

Organizations accomplish this through customer engagement. Customer engagement represents the emotional connection between a customer and a brand. How an organization engages their customers directly impacts their relationship with them. Highly engaged customers buy more and demonstrate more loyalty.

Many organizations leverage customer engagement software to assist them in engaging with their customers. Customer engagement applications help organizations fundamentally re-imagine how they engage customers. They help organizations create personalized marketing, sales, and service experiences using data and intelligence to improve every interaction.

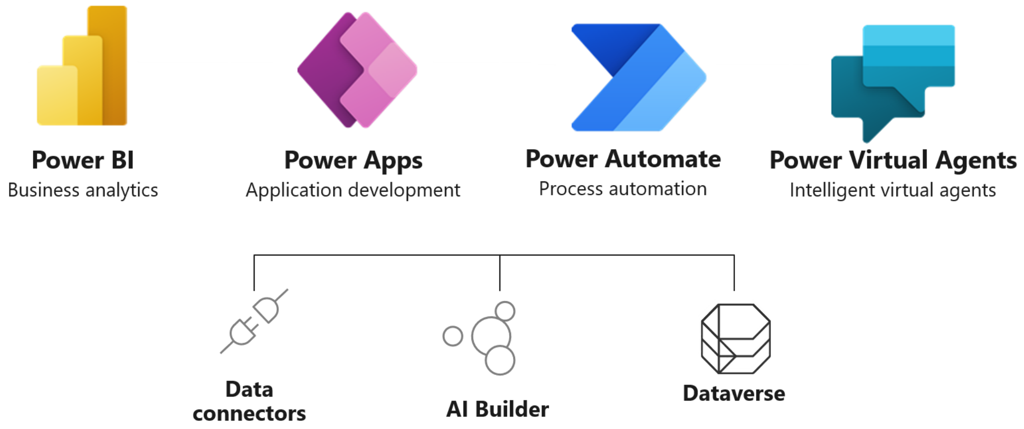
Dynamics 365’s Customer engagement applications are Microsoft’s first party applications built on the **Microsoft Power Platform**. Organizations can leverage these applications individually or together to create powerful end-to-end relationships with their customers.

Microsoft Dataverse

As mentioned previously, Dynamics 365’s customer engagement applications are built on the Microsoft Power Platform. The Microsoft Power Platform is a low-code application development platform that not only spans Dynamics 365, but also Office 365, Azure, and standalone applications. It combines the power of PowerApps, Power BI, and Power Automate into one business application platform that provides quick and easy app building and data insights.

Microsoft Power Platform

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



Power BI
Business analytics

Power Apps
Application development

Power Automate
Process automation

Power Virtual Agents
Intelligent virtual agents

Data connectors

AI Builder

Dataverse

One of the key elements to building applications on this platform is Microsoft Dataverse. Dataverse lets you securely store and manage data used by your business applications. For example, you might store all your customer data in Dataverse. Dataverse data is stored in tables. A table is a set of rows and columns. Each column in a table stores a specific type of data such as names, locations, ages, dates, salaries, and so on.

Dataverse includes a base set of standard tables that cover typical business scenarios such as accounts, contacts, and activities. The advantage to Dataverse is that organizations can also create custom tables specific to their needs and populate them with data.

For example:

- A real estate company might create tables to store the properties they sell, represent open houses, or store showings.
- A financial company might add tables to represent loan applications or bank accounts.
- An auto repair company might add tables to represent the parts they sell or the services they provide.

Application makers can use tools like Power Apps to build rich applications that leverage this data.

Leveraging Dataverse provides 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 are stored.
- Easy to secure: Data is securely stored so users can only access what they need to. Role-based security allows you to control access to tables for different users within your organization.
- 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.

Dynamics 365 first-party customer engagement applications such as Dynamics 365 Sales, Dynamics 365 Customer Service, or Dynamics 365 Marketing use Dataverse to store and secure their data. This enables you to build apps by using Power Apps and Dataverse directly against your core business data, which is already used within Dynamics 365, without the need for integration.

Examine the customer engagement apps

Dynamics 365 customer engagement applications help organizations fundamentally reimagine how they engage with customers. Organizations can create personalized marketing, sales, and service experiences using data and intelligence to improve every interaction.

There are five core Dynamics 365 customer engagement applications:

- **Microsoft Dynamics 365 Sales:** Go beyond sales force automation to better understand customer needs, engage more effectively, and win more deals.
- **Microsoft Dynamics 365 Customer** Service: Built-in intelligence delivers faster, more personalized service and adds value to every interaction.

- **Microsoft Dynamics 365 Field Service**: Built-in intelligence helps you resolve service issues before they occur, reduce operational costs, and deliver positive on-site experiences.
- **Microsoft Dynamics 365 Project Operations:** Build trusted customer relationships and deliver outstanding project experiences by delivering profitable projects on time and within budget, while increasing employee productivity.
- **Microsoft Dynamics 365 for Marketing:** Find and nurture more sales-ready leads by moving beyond basic email marketing. Connect sales and marketing, automate processes, and make smarter decisions to maximize your marketing Return on Investment (ROI).

When used alone, each application empowers organizations to manage the daily aspects related to that item. For example, Dynamics 365 Sales helps you take control of, manage, and automate the tasks related to your sales cycle. However, when combined with Dynamics 365 Customer Service, you now have a complete solution that ensures you are providing high levels of service to your customers after they are sold. Service-related events can be used to trigger sales-related events so sellers can get out and sell to customers. Even though each application provides a specific type of functionality, they each share common elements to provide a unified experience whether you are using them separately or together.

Work with customers and activities

Completed

As mentioned previously, a customer represents an individual or company that receives, consumes, or buys a product or service.

There are three ways organizations define customers:

- **Business-to-Business (B2B):** Customers are other businesses. For example, a business that sells and implements telecommunication equipment to hotels.
- Business-to-Consumer (B2C): Customers are individuals. These are businesses you likely come in contact with every day through activities such as purchasing groceries, getting a haircut, or stopping at a favorite coffee shop.
- Hybrid: Customers might be either businesses or individuals. For example, an automotive repair shop will certainly have customers who are individuals. Customers go there to get their oil changed, tires

rotated, engines overhauled, and so on. What if that automotive business has their mechanics outsourced to organizations with fleets of trucks, buses or automobiles as well? In this example, they service both organizations, and people. Other examples might include: Software Vendors, Event Centers, Insurance Companies.

Dynamics 365 applications support all three customer models. There are two primary records used to support this functionality:

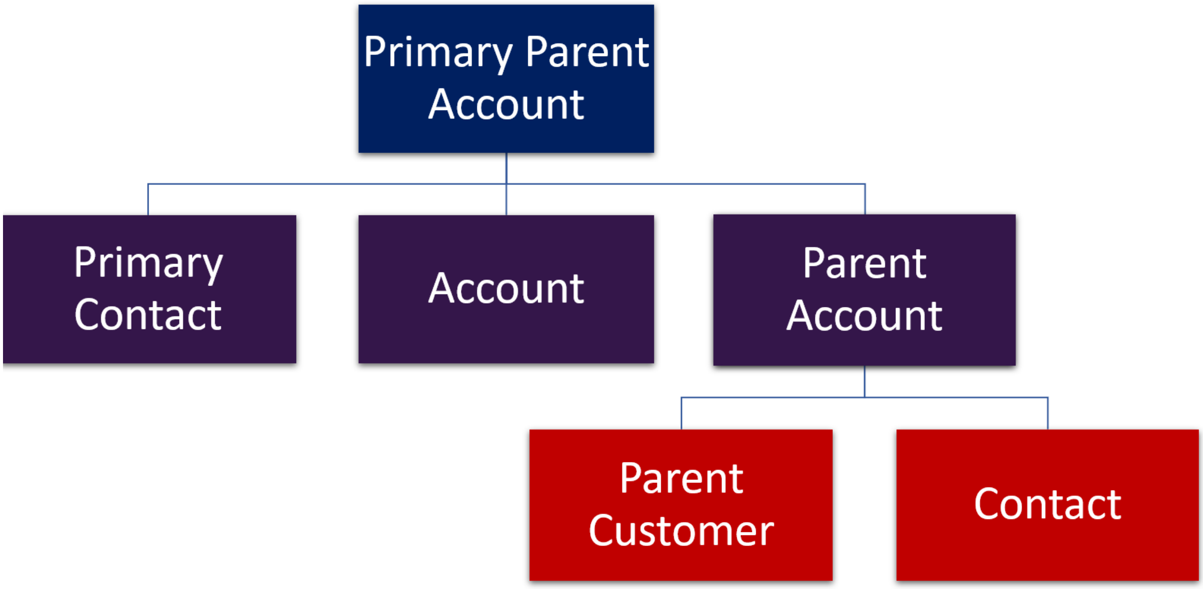
- Account: Represents a company or an organization that you do business with such as a customer, vendor, partner, or reseller.
- Contact: Represents a person. A contact can be a standalone person, or an individual at an organization.

Accounts and contacts can be related to one another in various ways. Individual contacts can be associated with an account. For example, if Microsoft were an account within the Microsoft Dynamics 365 system, then the contacts might include Bill Gates. Each contact record in Dynamics 365 includes the parent customer field that lets you link a contact to a specific account record.

While an account might have multiple contacts, it is likely that it will have a primary contact. The primary contact represents a key person at the organization that you primarily deal with. This might be the person you deal with as it relates to billing, sales, and so on.

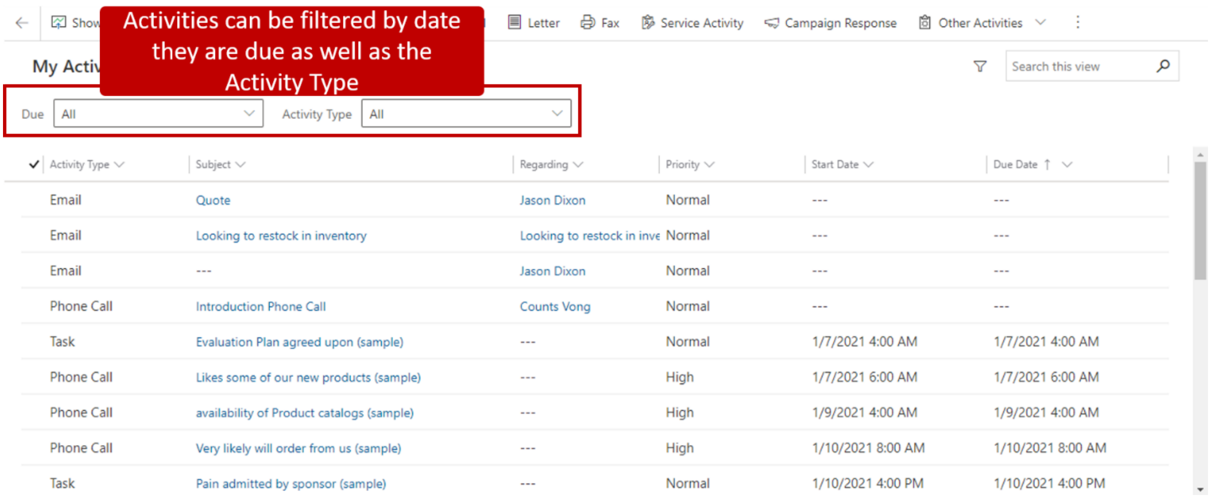
While many accounts will be stand alone accounts, there are occasions where an account can be considered as the parent account to another account. This might occur when one account is a division or subsidiary of a parent company.

The image below shows what this might look like.



Activities

Activities in Dynamics 365 help keep track of customer communications. For example, as you work with customers, you might do things like take notes, send email, make phone calls, set up appointments, or assign yourself tasks as you resolve a service case. By entering these activities in Dynamics 365, you will have a historic record of your communication with customers.



Activities can be filtered by date they are due as well as the Activity Type

Due: All
Activity Type: All

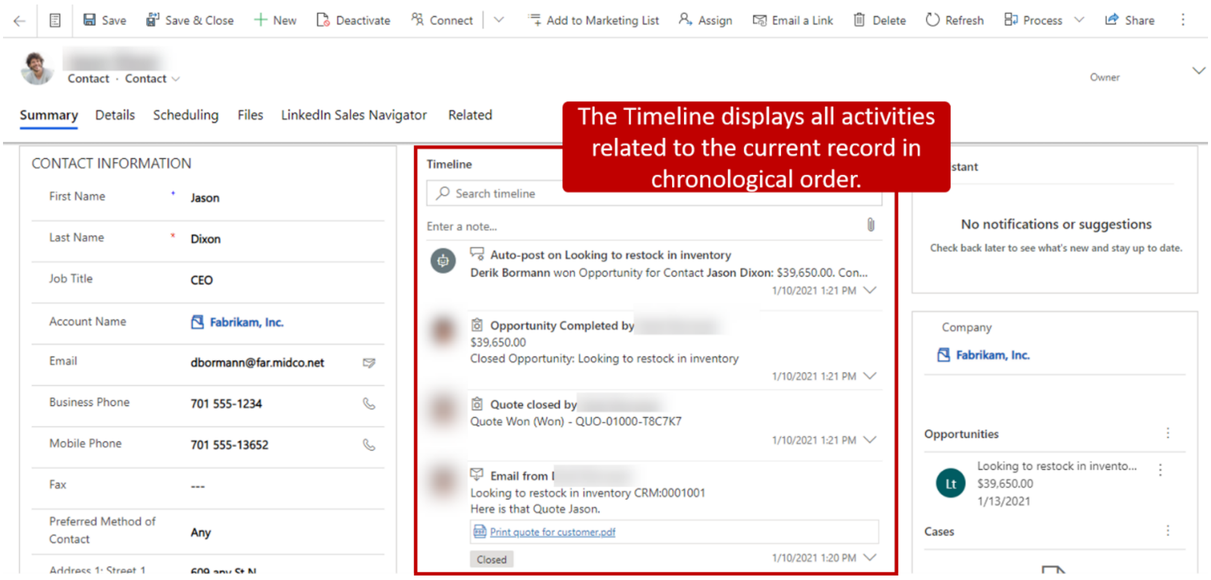
Activity Type	Subject	Regarding	Priority	Start Date	Due Date ↑
Email	Quote	Jason Dixon	Normal	---	---
Email	Looking to restock in inventory	Looking to restock in inve	Normal	---	---
Email	---	Jason Dixon	Normal	---	---
Phone Call	Introduction Phone Call	Counts Vong	Normal	---	---
Task	Evaluation Plan agreed upon (sample)	---	Normal	1/7/2021 4:00 AM	1/7/2021 4:00 AM
Phone Call	Likes some of our new products (sample)	---	High	1/7/2021 6:00 AM	1/7/2021 6:00 AM
Phone Call	availability of Product catalogs (sample)	---	High	1/9/2021 4:00 AM	1/9/2021 4:00 AM
Phone Call	Very likely will order from us (sample)	---	High	1/10/2021 8:00 AM	1/10/2021 8:00 AM
Task	Pain admitted by sponsor (sample)	---	Normal	1/10/2021 4:00 PM	1/10/2021 4:00 PM

Activities are often associated with contacts and accounts but, depending on the Dynamics 365 application you are using, they can also be associated with other types of records.

- In sales scenarios, activities might be added to records like leads, opportunities, or quotes.

- In service scenarios, activities might be added to records like cases, work orders, or projects.

As your organization's relationship with a customer builds over time, you and other people on your team can look through the record time-line to see the history of your interactions.



[redacted]
Contact · Contact

The Timeline displays all activities related to the current record in chronological order.

CONTACT INFORMATION

First Name	Jason
Last Name	Dixon
Job Title	CEO
Account Name	Fabrikam, Inc.
Email	dbormann@far.midco.net
Business Phone	701 555-1234
Mobile Phone	701 555-13652
Fax	---
Preferred Method of Contact	Any
Address 1: Street 1	600 any St N

Timeline

Search timeline

Enter a note...

Auto-post on Looking to restock in inventory
Derik Bormann won Opportunity for Contact Jason Dixon: $39,650.00. Con...
1/10/2021 1:21 PM

Opportunity Completed by [redacted]
$39,650.00
Closed Opportunity: Looking to restock in inventory
1/10/2021 1:21 PM

Quote closed by [redacted]
Quote Won (Won) - QUO-01000-T8C7K7
1/10/2021 1:21 PM

Email from [redacted]
Looking to restock in inventory CRM:0001001
Here is that Quote Jason.
Print quote for customer.pdf
Closed
1/10/2021 1:20 PM

stant

No notifications or suggestions
Check back later to see what's new and stay up to date.

Company
Fabrikam, Inc.

Opportunities
Looking to restock in invento...
$39,650.00
1/13/2021

Cases

Microsoft Dynamics 365 CE Architecture

Microsoft Dynamics 365 Customer Engagement Architecture:

Microsoft Dynamics 365 CE has a N-tier architecture which hasn't changed for a long time.

This N-tier architecture can be extended with the help of UI tools and code means developers can customize the standard CRM functionality by developing workflows, plugins and custom workflow activities.

Knowing the architecture of Dynamics 365 CE is quite important for the developers to extend/customize the application.

The 3 tiers (or layers) of the Dynamics 365 CE architecture are:

1. Web Tier

2. Application Tier

3. Data Tier

Web Tier:

Web Tier handles the web pages of the applications such as CRM entity forms, views, charts, dashboards and all other UI components. The web tier or web layer is configured on the IIS server which is responsible for the rendering of the web components.

Application Tier:

Application tier manages the business logic of the application such as plugin, processes, security and data elements. The access to the application layer is controlled through the web services, means to access any of these components the user has to be authenticated.

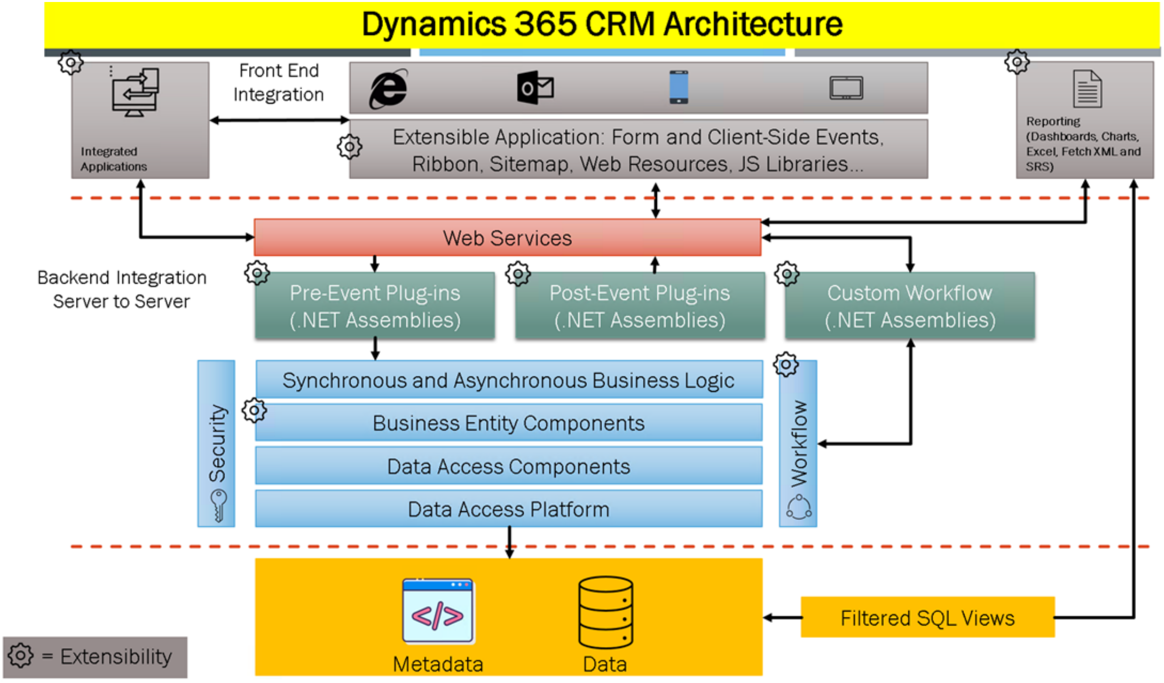
Data Tier:

The data or database layer is responsible for the storing the data and metadata of the Dynamics 365 CE application. The database used is SQL Server and the database will be configured when we install the CE application.

For every entity in the application there will be a corresponding table available in the application.

D365 – Architecture Overview

Following is an overview of D365 CRM architecture.


Dynamics 365 CRM Architecture
Integrated Applications
Front End Integration
Extensible Application: Form and Client-Side Events, Ribbon, Sitemap, Web Resources, JS Libraries...
Reporting (Dashboards, Charts, Excel, Fetch XML and SRS)
Web Services
Backend Integration Server to Server
Pre-Event Plug-ins (.NET Assemblies)
Post-Event Plug-ins (.NET Assemblies)
Custom Workflow (.NET Assemblies)
Security
Synchronous and Asynchronous Business Logic
Business Entity Components
Data Access Components
Data Access Platform
Workflow
Metadata
Data
Filtered SQL Views
= Extensibility

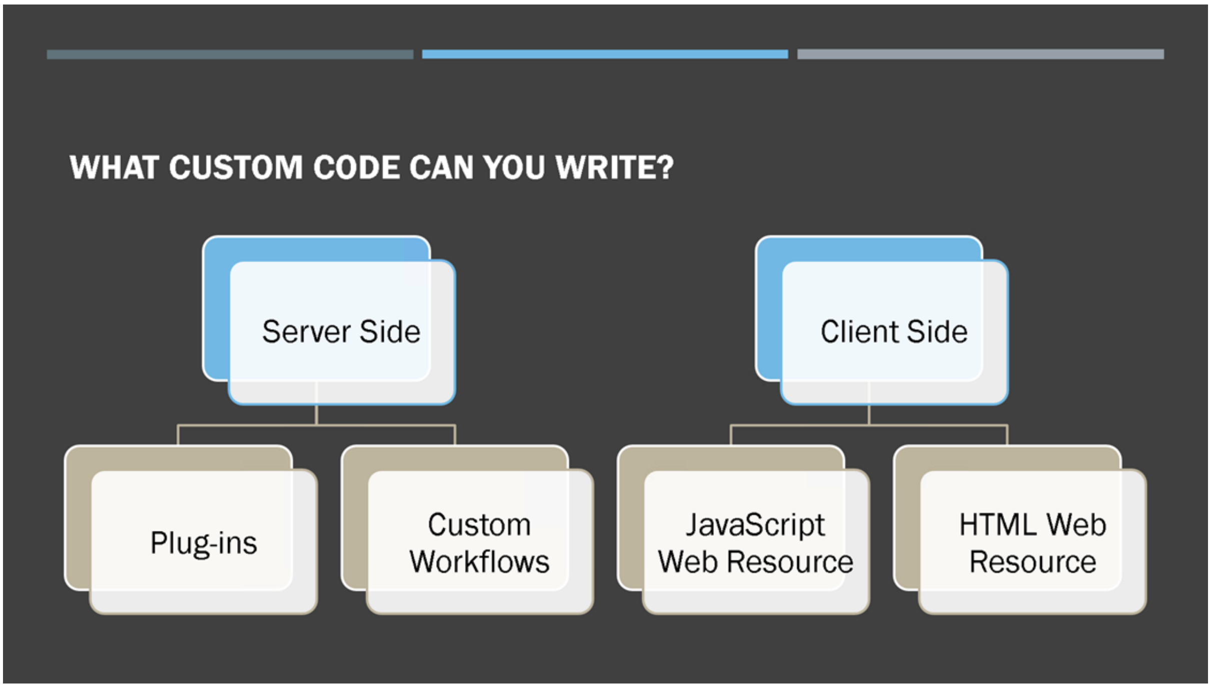
Understanding the architecture is important to understand the entire ecosystem of development tools here.

- So the first dotted line separates the client side components from the server side components.
- Wherever you see the gear icon, those components are customizable. So you can write some custom code where you see the gear icons and these components are available on client side as well as on the server side.
- Every D365 CRM instance has Metadata database and Data database. We don't need to do much with it as the platform takes care of these databases.
- Also, you cannot directly access or make changes to the database with any code.

What Custom Code can you Write?

You can broadly categorize the custom components into following types:

- Server Side
- Client Side

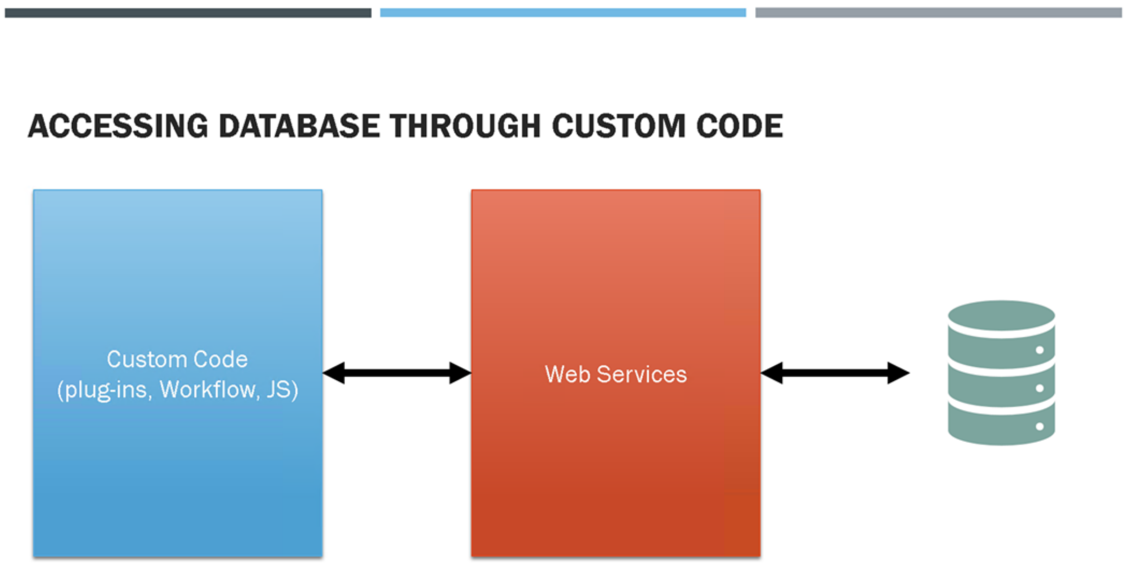


WHAT CUSTOM CODE CAN YOU WRITE?

How does Custom Code access CRM data?

Custom code cannot access data directly from the D365 CRM database.

It can access data through API (Webservices) only. The connection between web services and databases is taken care of by the platform.



Second Week Content

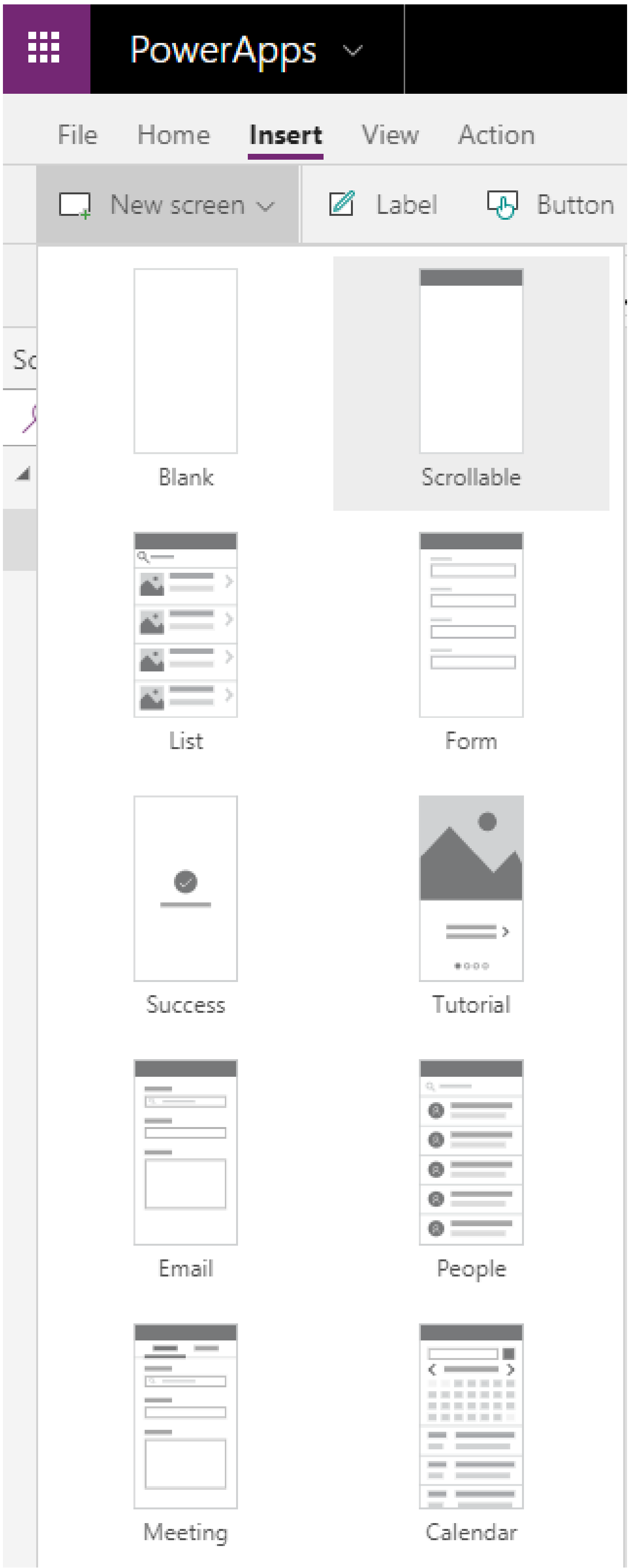
6 major components of PowerApps

POWER APPS

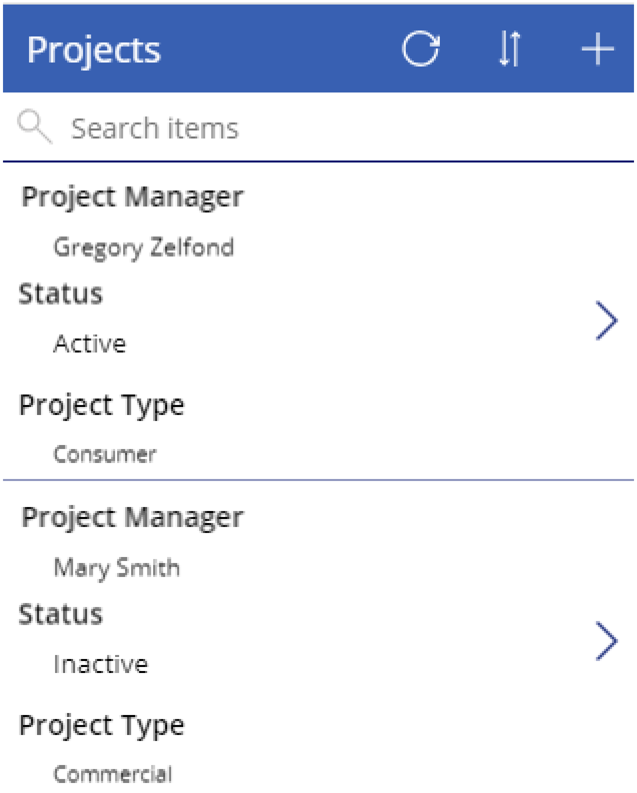
If you want to embrace PowerApps, you first need to understand its anatomy and basic elements. With this post, I would like to highlight, what are, in my opinion, the major components of PowerApps one needs to understand first before creating the first app.

1. GALLERY

A **gallery** is a way of visualizing data in the app. It is a template of screens that allows you to see and navigate the data. For example, you might have a gallery that contains a screen to see all records, then a screen to view a given record and a screen to edit a record.



All of the screens combined make up a **gallery**. You have different galleries to choose from when you create a new app. When you create an app from SharePoint list, you get a default gallery already pre-built for you.



Projects

Search items

Project Manager

Gregory Zelfond

Status

Active

Project Type

Consumer

Project Manager

Mary Smith

Status

Inactive

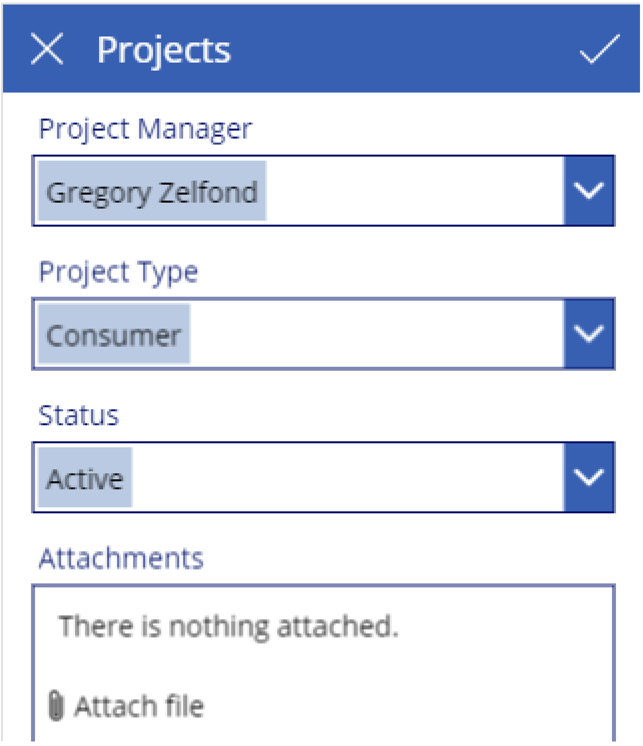
Project Type

Commercial

Example of a Screen based on a certain Gallery template

2. SCREEN

As alluded to in the previous section, a **screen** is a way to visualize a particular data set or record on a screen (mobile, iPad, Desktop). You typically have one to view all records, one to view a particular record, one to edit. Once again, when you create an app from the template, you have screens already pre-built for you (they are part of a gallery you choose above). But you can add additional screens if necessary.



Projects

Project Manager

Gregory Zelfond

Project Type

Consumer

Status

Active

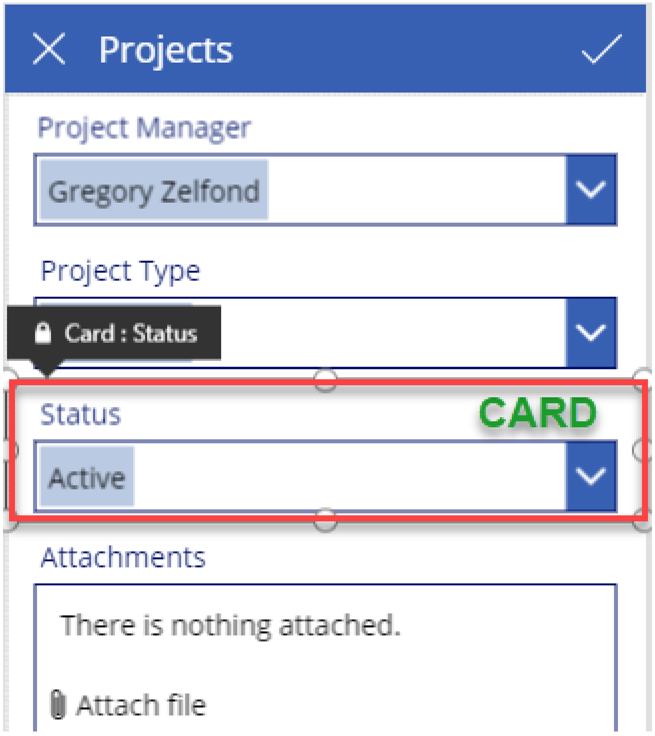
Attachments

There is nothing attached.

Attach file

3. CARD

A screen consists of **cards**. A **card** is an area on the screen that shows a given record from your SharePoint list or any other database you used to build an app. For example, if you had a SharePoint list storing project names and built a PowerApp from this list, a single card might contain a **Project Name** field or **Project Manager** field or **Project Type** field or a **Status** field.



Projects

Project Manager

Gregory Zelfond

Project Type

Card : Status

Status

Active

Attachments

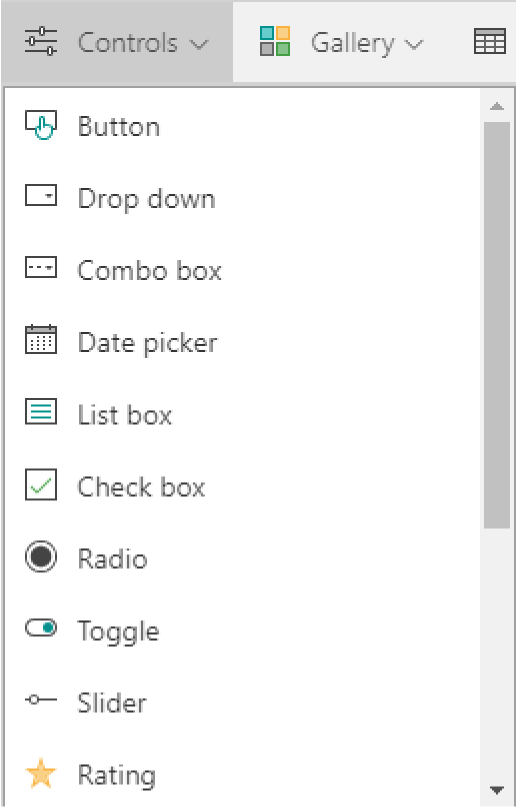
There is nothing attached.

Attach file

A card would contain all various attributes (called **Controls** in PowerApp – described below) related to the display of the record. For example, for **Status** field depicted in the image above, a card contains a **Drop-Down control** to enter Status as well as **a label** called "Status".

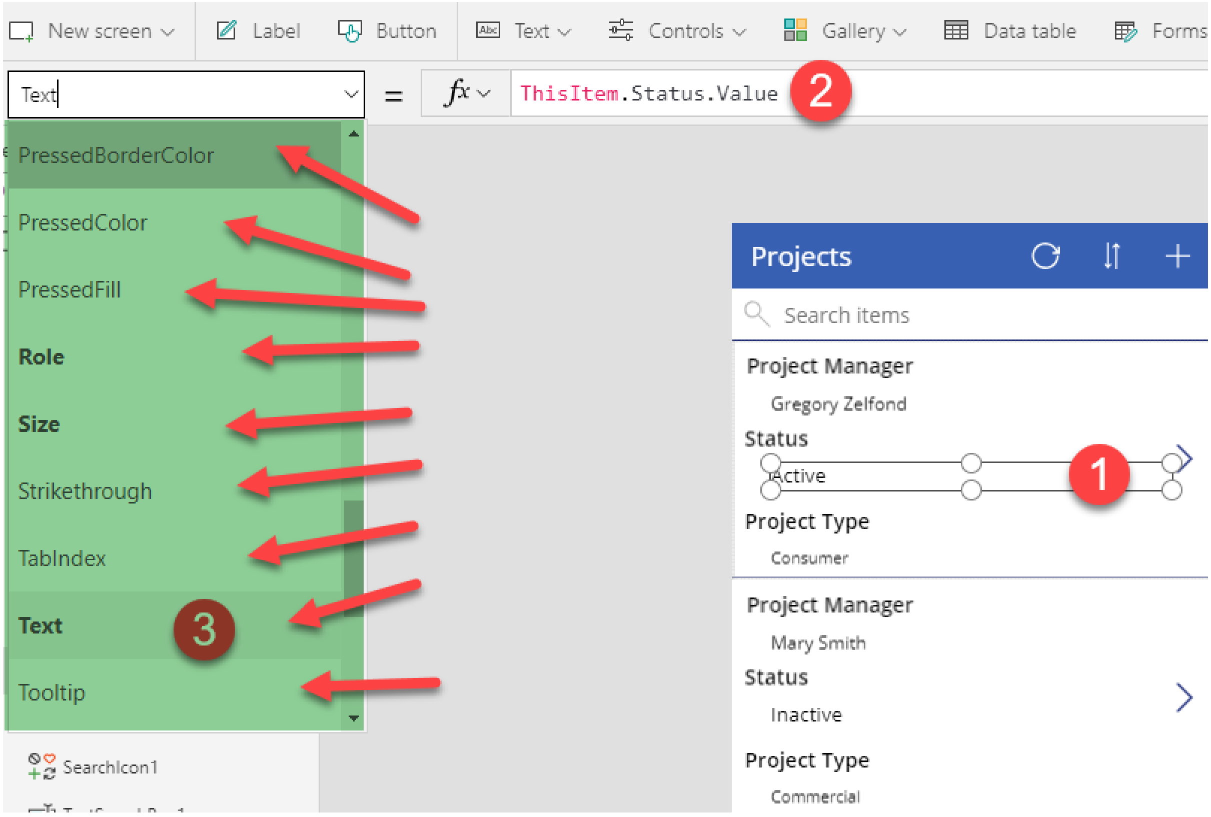
4. CONTROL

Controls is what allows you to visualize and interact with your records. Using the above **Status** field example, the **Drop-Down** to choose Project Status **is a control**. Depending on the type of your field, you can have different types of controls. For example, instead of a drop-down, I could have a radio button or toggle switch to make selection a bit more elegant on the phone app.



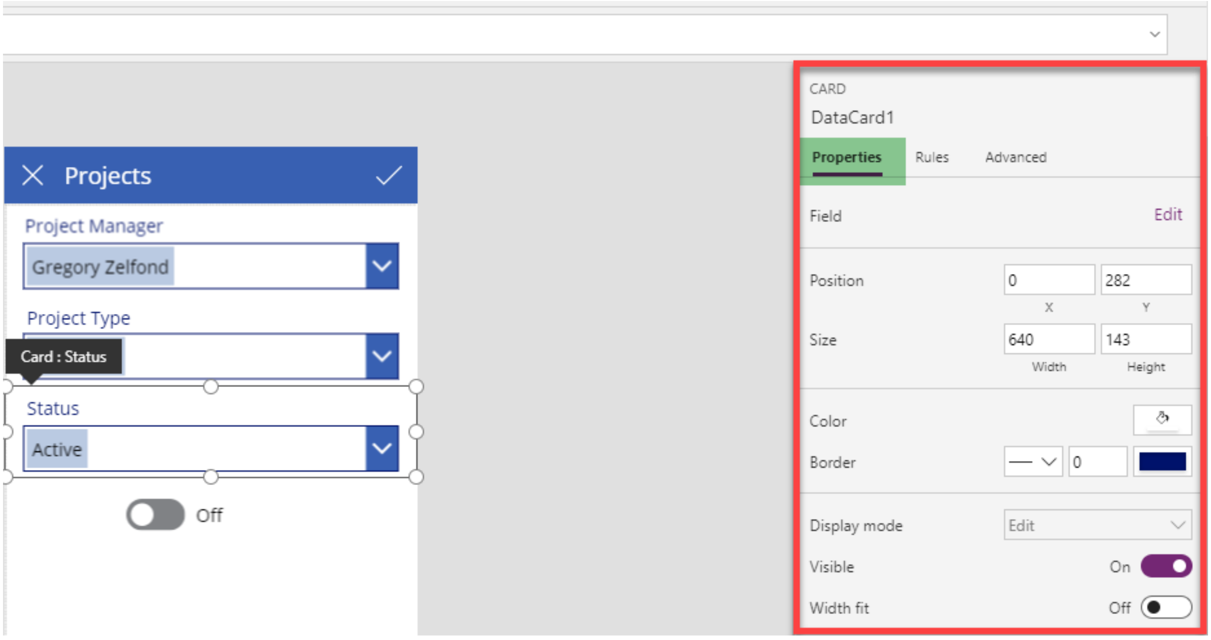
5. PROPERTY

Each control has **properties**. For example, a text entry box has a property for font size, text color, text box fill color and so on. Properties can be accessed and changed from the Properties drop-down on the left-hand-side of the screen as well as on the panel on the right-hand-side once the property is selected.



```
Text = ThisItem.Status.Value
```

All the arrows point to properties of a certain control

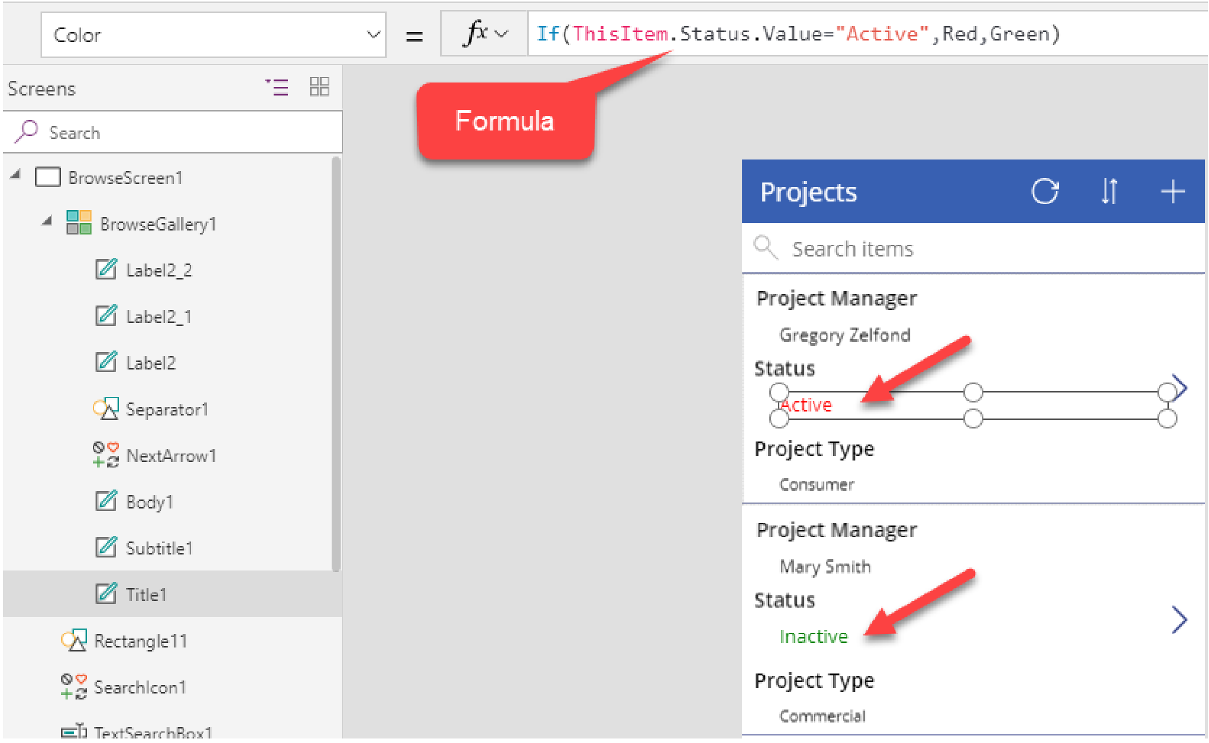


Field: Edit
Position: 0 (X), 282 (Y)
Size: 640 (Width), 143 (Height)
Color
Border: 0
Display mode: Edit
Visible: On
Width fit: Off

Another way to edit properties for the card

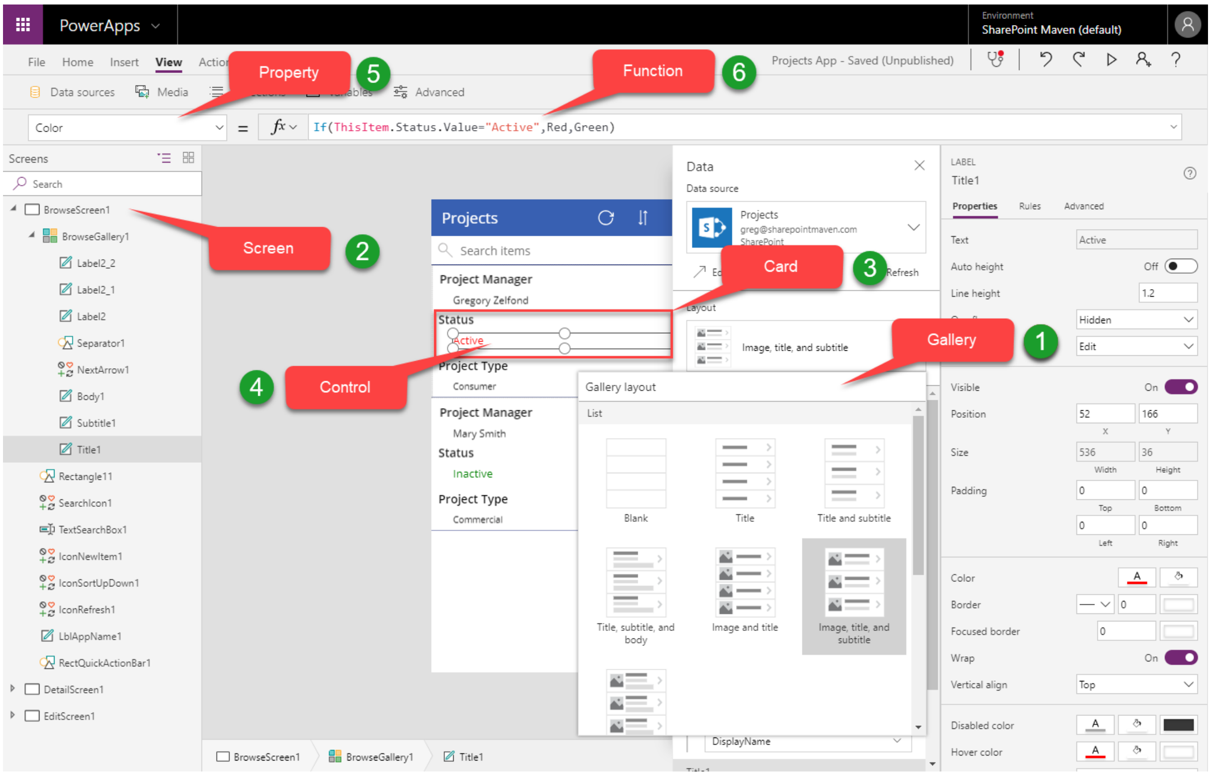
6. FUNCTION

Functions is how you interact with and change the properties. If you are familiar with Excel, you are in luck as the syntax for PowerApps functions is kind of similar to Excel. Below is an example where I use the **Formula** (If Statement) to change **Property** (Color) of a **Control** (Text) depending on the value of the field.



Color = fx If(ThisItem.Status.Value="Active",Red,Green)

And now, putting it all together:



```
If(ThisItem.Status.Value="Active",Red,Green)
```

Tweet

Custom Components in PowerApps

PowerApps can be a powerful tool for creating fast prototypes, small business apps, and proof of concepts. There are a lot of built-in components that are useful but often require a lot of configuration if you want more than just the default functions and style.

Enter PowerApps components. Components let you create your own reusable components which can be exported and imported to different projects, and make maintenance a lot easier for developers.

Components are useful in building larger apps that have similar control patterns. If you update a component definition, all instances in the app reflect your changes. You can also improve performance by using one or more components because you don’t copy and paste controls, which duplicates overhead.

Components are isolated instances that are decoupled from your app. This means that a custom component can’t access variables declared in your app unless they are parsed as properties into the component.

custom connectors in Power Platform

Power Platform has become one the most set of tools that many organizations are leveraging. Power Platform has the robust power of **Power Apps, Power Automate** and **Power BI** into one powerful business application platform that helps citizen developers as well as pro developers in providing quick and easy app building and data insights.

Microsoft Power Platform is low-code platform that spans Office 365, Azure, Dynamics 365, and standalone applications a shown below

Microsoft Power Platform is low-code platform that spans Office 365, Azure, Dynamics 365, and standalone applications a shown below

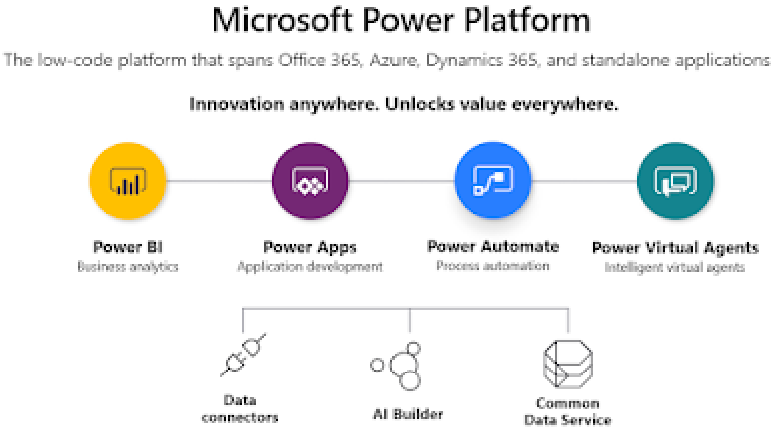
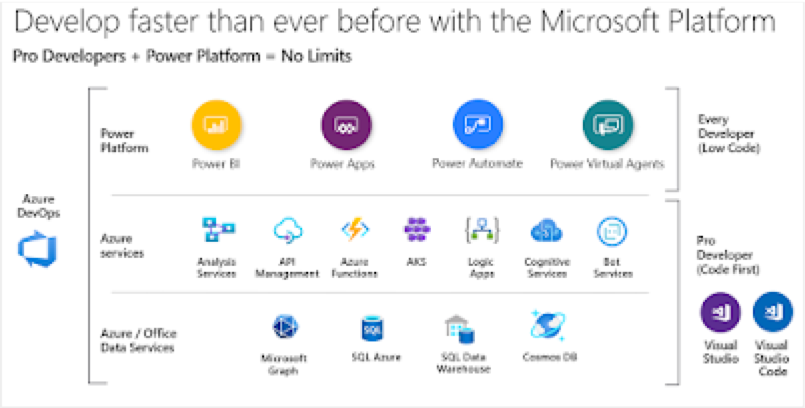


Figure 1 Credit: Microsoft

The best thing about Microsoft Power Platform is that is caters not only for **citizen developers** but also for **professional developers** as shown:



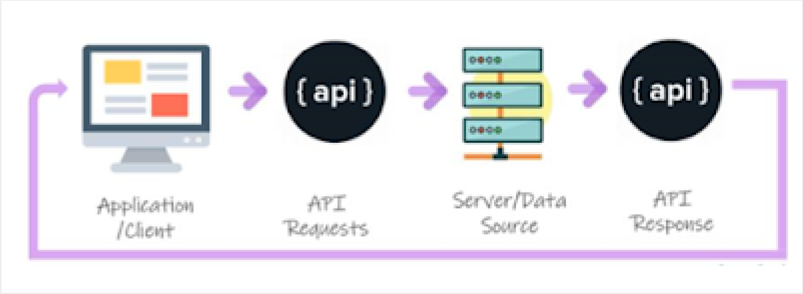
Why needs an API in Power Platform?

Power Platform leverages Microsoft services to connect to other services such as SharePoint, OneDrive as 3rd party services such as **MailChimp**, Adobe Sign etc.

What is an API?

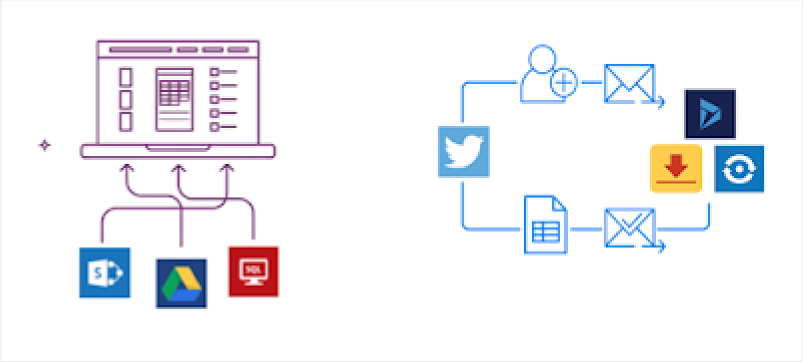
API (Application Programming Interface) that allows **two applications** to talk to **each other**.

for e.g. A user initiates **Power Apps** (Application) to get a signature **leveraging Adobe Sign**, this will be the API call from Application (Power Apps) and **Adobe Sign as a data source** as shown below.



What is connector?

A connector is a **wrapper around an API** that allows the underlying service to talk **to Power Automate** and **Power Apps.**

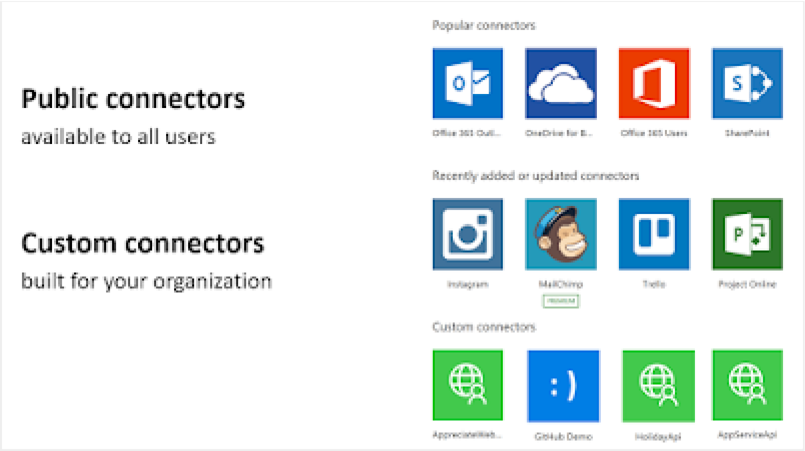


Type of connectors

You have 2 types of connectors.

1. **Public connectors (available to all)**

2. **Custom connectors (built for your organization)**



Introduction of Variables

With the introduction of PowerApps, more and more users are moving into PowerApps development for easy and no-code development tools. However, users from a coding background found it difficult, as there were no variables in PowerApps. Not now, with the introduction of variables in PowerApps, it has become a Powerful no-code tool. So let's understand PowerApps Variables and how they works.

Types of Variables

Just as we have different scopes of variables in the programming language in PowerApps, we also have different scoped variables the three different types of variables which are available in PowerApps, as mentioned below:

Global variable	App	This is similar to a global variable in programming language and it can hold a number, text string, Boolean, record, table, etc as a data value	Set

Context variables	Screen	This is similar to the parameter which we pass to methods or procedure and this variable can be referenced from only one screen	UpdateContext Navigate
Collections	App	This variable can store table which can be referenced across anywhere in the app	Collect ClearCollect

Creating and Removing Variables

In all programming languages, we declare a variable before using the variable but such is not the case with PowerApps. So, in PowerApps we never declare variables explicitly. In terms of typing, PowerApps has implicit typing. So, we do not require to specify the type of variable for example int a = 0 in programming language indicated the variable is of type integer but in PowerApps we do not require to specify the type of variable.

To create a variable, we just need to run the function:

- `Set(global Variable, "Hello World")`
- `Update Context({context Variable:"Hello World"})`
- `Collect( collection Variable, "Hello World" )`

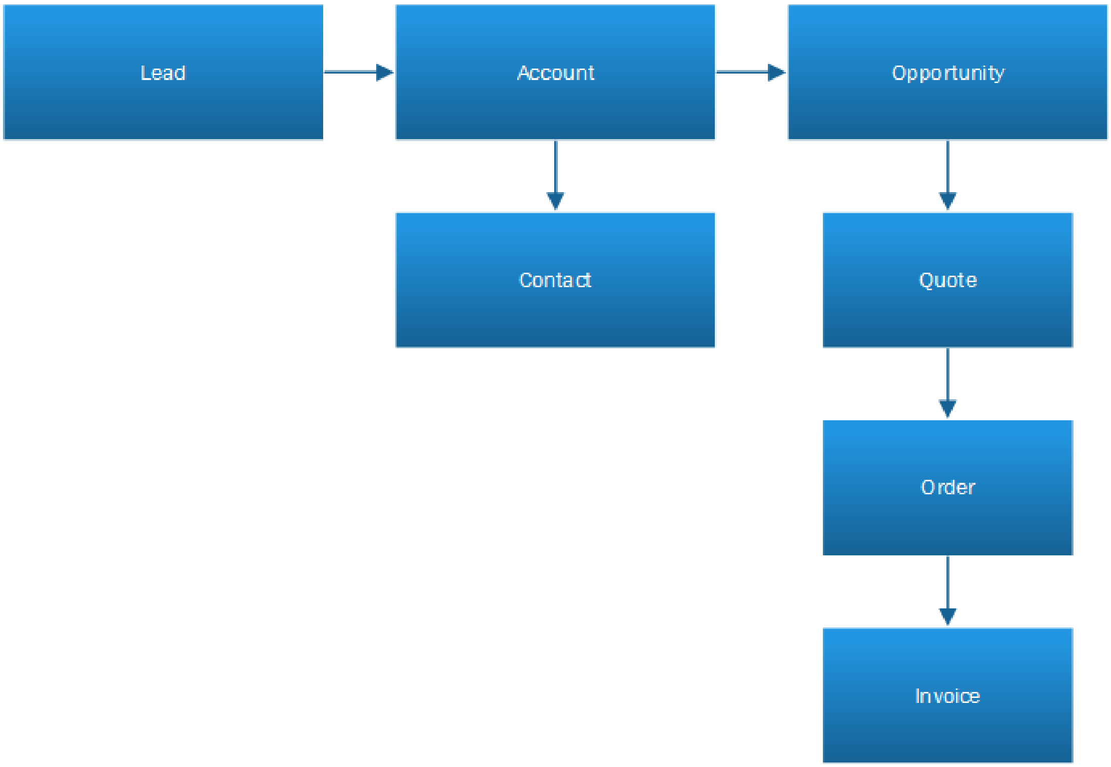
Where X is the variable name and Hello World is the value of that variable.

For removing the variable, we are required to remove all Set, UpdateContext, Navigate, Collect, or ClearCollect functions that implicitly establish the variable. If these functions are not present then there is no variable. We must even remove the variable reference if used in other places.

3rd Week Content

Sales life cycle

Generally, the sales life cycle starts with Lead generation. A lead is your potential customer. When an individual or a company shows any interest in your product/service and provides some details for further communication, then it is considered a lead:



The salesperson either manually enters the details of the lead into the CRM or captures the details through mails, websites, phone calls, campaigns, and so on. It requires a lead name (first name and last name) and a topic that describes the lead.

After lead creation, the salesperson does multiple activities to convert it to a customer. This can be done in form of appointments, ...