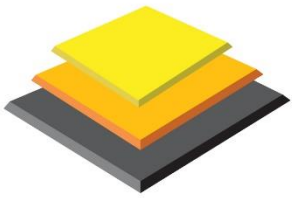
Zavanti Housing Implementation Guide

Installing Zavanti Housing

Zavanti Australia Pty Ltd



Zavanti

Technology for Advantage

This manual was produced using *MadCap® Doc-To-Help*.

Contents

Zavanti Housing Solution Installation and Architecture

Introduction

This document highlights all the components and requisites that are required to deploy Zavanti Housing solution in clients Dynamics 365 tenant. It also covers high level architecture and how the solution integrates with other Microsoft products and services.

This document showcases all dependant applications, why they are required and how they interact with the Zavanti housing system.

Below are the components the Zavanti solution requires -

- Dynamics 365 Sales Enterprise version
- Microsoft SharePoint
- Microsoft Exchange and Outlook
- Microsoft Power Platform (Power Apps)
- Microsoft Azure Functions

Please note that the contents of this document are developed in accordance with Microsoft best practice. All steps and instructions in this document are supported by Microsoft and there is no Zavanti proprietary requirements or architectural design components that's needed outside of the Microsoft framework and technology stack. Hence this is a generic document.

Where required reference and links to Microsoft websites and articles have been provided.

Zavanti Housing Solution

Zavanti Housing is a comprehensive software solution developed on the Dynamics 365 platform. The system aims to streamline and automate the management of housing, tenancies, properties and assets within an organisation by leveraging the capabilities of Dynamics 365. The system enables efficient management of properties, tenants, maintenance, and asset management.

Property and Tenancy Management:

- Maintain property records, including property details, addresses, and features.
- Manage tenancies and applications, including terms, rental rates, and renewal options.
- Handle move-in and move-out processes, including inspections and inventory management.
- Track rent payments, generate invoices, and manage bonds.

- Record and track maintenance requests and work orders associated with each property.
- Generate reports, such as rent roll, income statements, and expense reports.

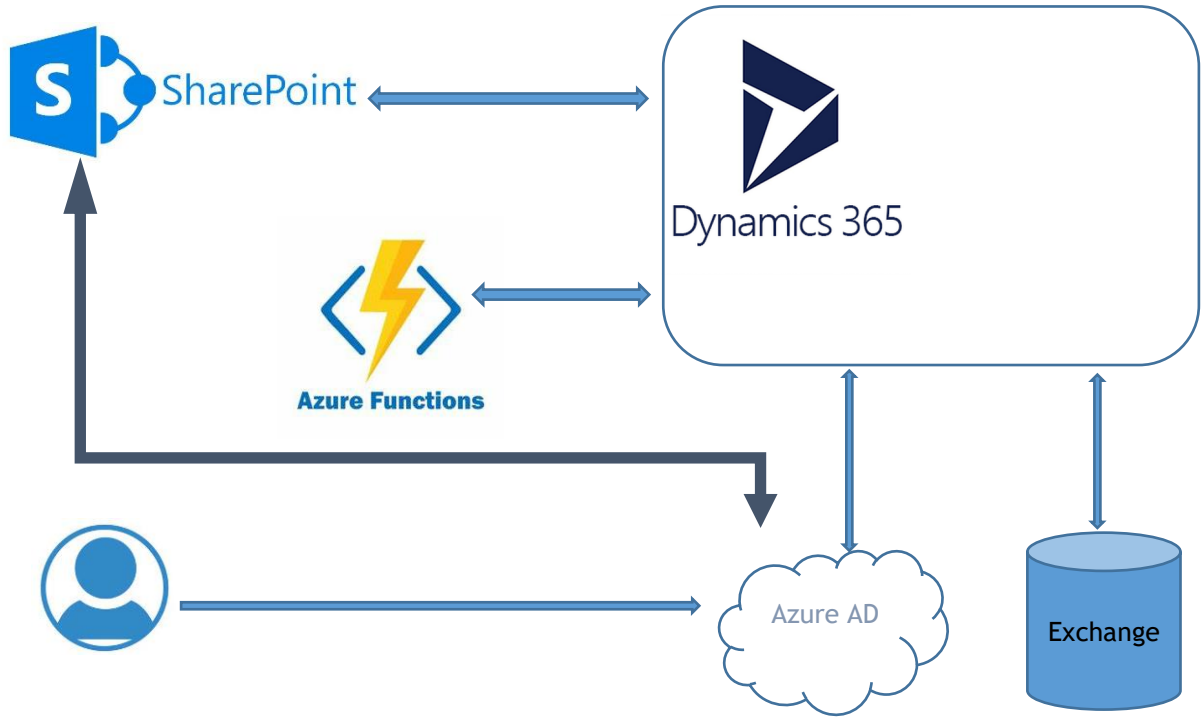
Asset and Maintenance Management:

- Maintain asset records, including details, categories, and locations.
- Schedule preventive maintenance tasks for assets.
- Record and track asset maintenance requests and work orders.
- Receive and assign maintenance requests for both properties and assets.
- Schedule and assign work orders to maintenance staff or external contractors.
- Track the progress and completion status of maintenance tasks.

Non-functional:

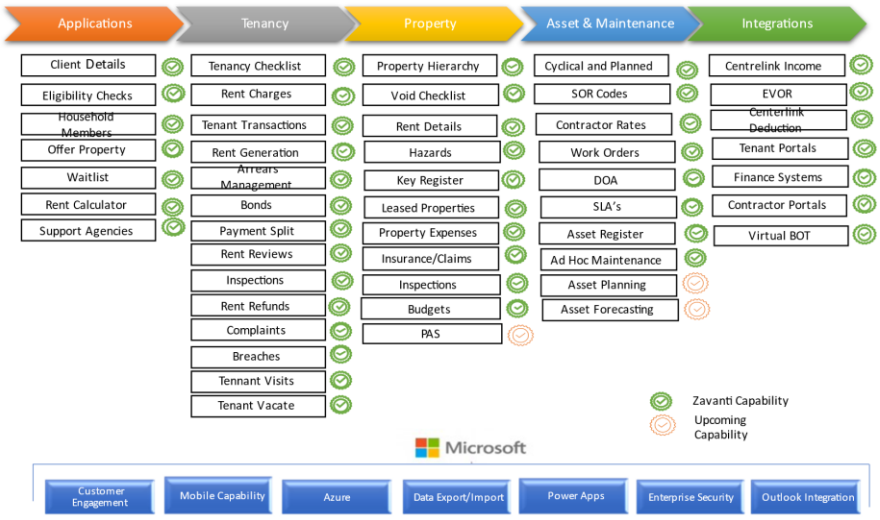
- Scalability: Dynamics 365 offers a scalable platform capable of handling a large number of properties, tenants, assets, and users.
- Security: The system leverages Dynamics 365's built-in security features, including role-based access control, data encryption, and authentication protocols.
- Performance: Dynamics 365 ensures optimal performance with its cloud infrastructure and automatic scaling capabilities.
- User Experience: The system provides an intuitive and user-friendly interface, leveraging the Dynamics 365 Unified Interface for a consistent experience across devices.

Zavanti Housing Solution Architecture Diagram



Zavanti Capability Matrix

The following diagram presents the capabilities o the Zavanti Housing solution.



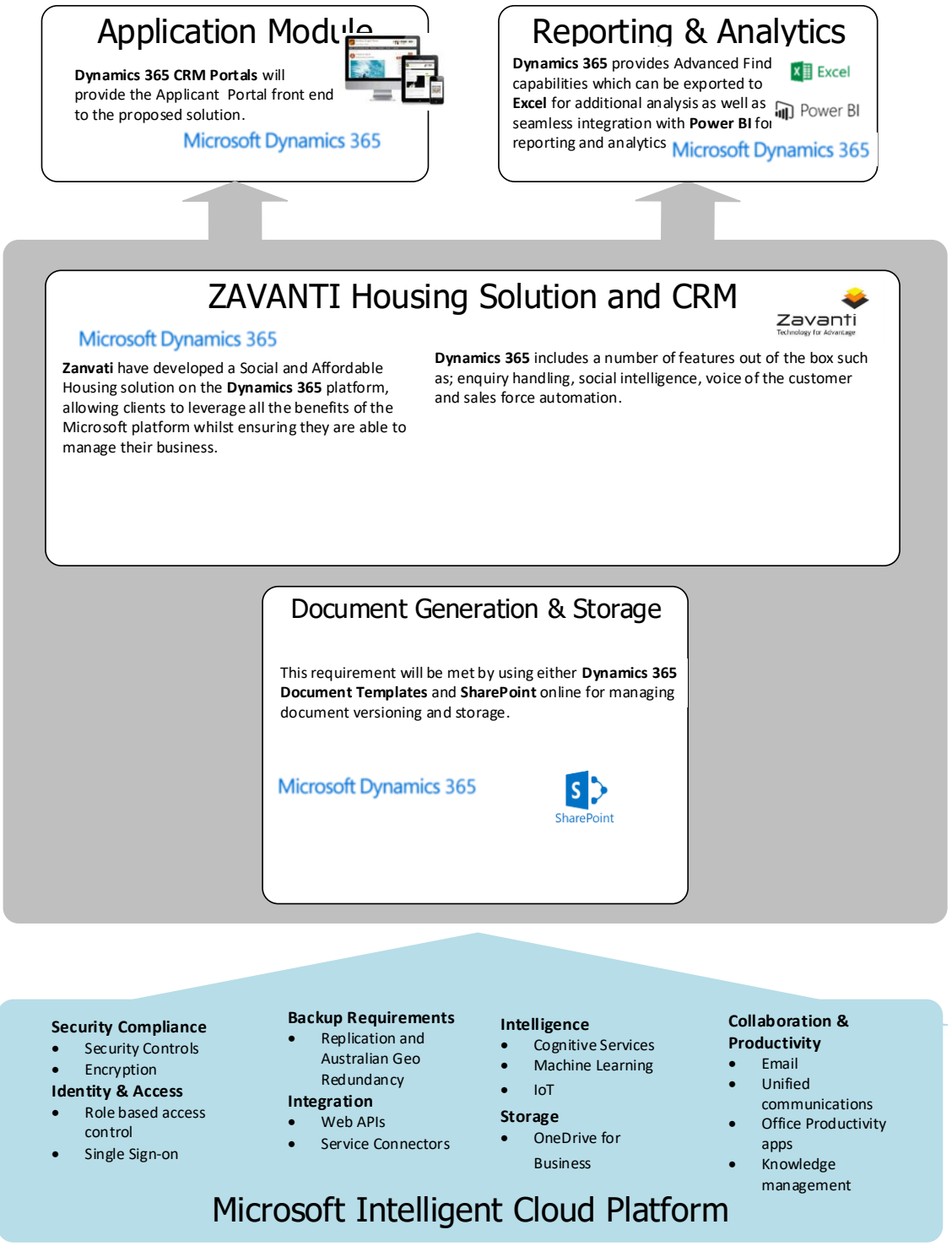
Architecture and Security

Dynamics 365 is built on a multi-tiered architecture that combines both cloud-based and on-premises components. The architecture is designed to provide scalability, flexibility, and reliability to meet the diverse needs of businesses.

Zavanti is an ISV (Independent Software Vendor) solution that is deployed within the Dynamics 365 platform. The solution extends or enhances the functionality of Dynamics 365, Microsoft's suite of enterprise resource planning (ERP) and customer relationship management (CRM) applications.

Here's an overview of how the Zavanti ISV solution operates within the Dynamics 365 platform:

- Integration: The Zavanti solution integrates with the Dynamics 365 platform to leverage its underlying infrastructure, data model, and services. The integration is achieved through various mechanisms like APIs (Application Programming Interfaces), custom entities, workflows, plugins and Zavanti specific business process flows. For example, application to tenancy business process flows.
- Installation: The Zavanti solution needs to be installed within the Dynamics 365 environment. This involves importing and configuring the solution components, such as custom entities, fields, forms, workflows, reports, and any other specific functionality provided by Zavanti.
- User Interface: The Zavanti solution introduces new screens, dashboards, forms, workflows, business processes within the Dynamics 365 user interface. These interface elements are designed to provide access to the additional functionalities that is needed for a housing and asset management solution. Users can interact with these components to perform specific tasks or access customised features.
- Functionality: The Zavanti solutions provides a wide range of functionalities tailored to housing, tenancy, and asset management. These functionalities include industry-specific processes, specific modules, document management, advanced workflows, integrations with other systems, or any other capabilities designed to enhance Dynamics 365's core functionality.
- Customization and Configuration: The Zavanti solutions may allow further customization and configuration within the Dynamics 365 platform. This is often done to enhance the solution to fit the client’s specific business process and needs. These customisations are done using tools and frameworks provided by the Dynamics 365 platform.
- Data Interaction: Zavanti solutions interact with the data stored in Dynamics 365 like Dataverse. The solution also interacts with other data platforms like SharePoint, exchange. Zavanti stores all data within Dataverse. There is no component/services or data that resides externally to the clients Microsoft 365 tenant.



Key Components of Dynamics 365 Architecture:

Presentation Layer:

Unified Interface: Dynamics 365 provides a unified interface that offers a consistent user experience across various devices, including desktops, tablets, and mobile devices. It allows users to access and interact with different Dynamics 365 applications through a web browser or dedicated mobile apps.

Application Layer:

Dynamics 365 Applications: The application layer consists of various pre-built applications within the Dynamics 365 suite, such as Sales, Customer Service, Finance, Supply Chain Management, and Field Service. Each application caters to specific business functions and comes with its own set of features, modules, and workflows.

Data Layer:

Dataverse: The Common Data Service (Dataverse) is a cloud-based data storage and management platform that acts as a centralized repository for storing and managing data across different Dynamics 365 applications. It provides a unified data model and supports integration with other data sources and external systems. All Zavanti entities are stored in Dataverse, we do not store any data or host any services outside of the Microsoft environment.

Integration Layer:

Connectors and APIs: Dynamics 365 offers a range of connectors and APIs that enable seamless integration with external systems, including third-party applications, databases, and services. These integration capabilities allow businesses to leverage existing systems and data sources within their Dynamics 365 environment.

Business Logic Layer:

Dynamics 365 Business Process Flows: Business process flows define the logical sequence of steps required to complete a specific business process. They can be customized to match the unique workflows and requirements of an organization, enabling efficient automation and guiding users through standardized processes.

Workflow and Automation:

Power Automate (formerly Microsoft Flow): Power Automate is a cloud-based service that allows users to create automated workflows across different applications and services. It enables businesses to streamline processes, trigger actions based on specific events, and automate repetitive tasks within the Dynamics 365 environment.

Analytics and Reporting:

Power BI: Power BI is a business analytics tool that integrates with Dynamics 365 to provide powerful data visualization, interactive dashboards, and rich reporting capabilities. It enables users to gain insights from their data and make data-driven decisions.

Infrastructure and Deployment:

Azure Cloud Infrastructure: Dynamics 365 is built on Microsoft Azure, a highly scalable and secure cloud computing platform. Azure provides the underlying infrastructure to host Dynamics 365 applications, ensuring high availability, scalability, and data security.

On-Premises Components: In some scenarios, organizations may choose to deploy Dynamics 365 on their own infrastructure, using on-premises components. This allows businesses to maintain greater control over their environment and data.

The Dynamics 365 architecture allows organizations to leverage the power of cloud-based applications and services, enabling seamless integration, scalability, and accessibility across different devices. It provides a unified platform for managing business processes, data, and analytics, empowering businesses to drive productivity, efficiency, and customer satisfaction.

Azure AD is the primary identity and access management service used for authenticating users in Dynamics 365.It enables Single Sign-On (SSO) across multiple applications and services within the Azure

ecosystem. Users authenticate using their Azure AD credentials, such as their username and password. Azure AD supports additional authentication methods like multi-factor authentication (MFA) for enhanced security.

Delegated Authorisation and Authentication

OAuth 2.0 is an industry-standard protocol used for delegated authorisation and authentication. Dynamics 365 supports OAuth 2.0 authentication, allowing users to authenticate using their existing credentials from external identity providers. OAuth 2.0 provides a secure and standardised way to obtain access tokens, enabling users to access Dynamics 365 without exposing their credentials to the application.

These authentication methods can be combined with additional security measures like Conditional Access policies, Multi-Factor Authentication (MFA), and role-based access control (RBAC) to enforce granular access control and ensure secure access to Dynamics 365 applications and data.

The Zavanti out of the box solution includes pre-configured security roles that are applicable for tenancy managers, property managers etc and administration user access.

Zavanti Security Model

Dynamics 365 employs a robust security model to protect data and ensure the privacy and integrity of its users' information. The Dynamics 365 security model comprises various components and features that collectively establish a comprehensive security framework. Here are the key elements of the Dynamics 365 security model:

Users and Security Roles:

Dynamics 365 allows the creation of user accounts with unique login credentials that’s in AAD. Each user is assigned a security role that defines their privileges and access levels within the Zavanti system. Security roles control which actions and data a user can view, edit, or delete.

Business Units:

Business units in Dynamics 365 help organise users and data within an organization. Each business unit can have its own security roles, allowing for more granular control over access to data and functionality.

Teams:

Teams are groups of users in Dynamics 365, and they can have their own security roles. Teams are useful for managing access and collaboration within specific departments or project groups.

Security Roles and Privileges:

Security roles in Dynamics 365 define the permissions and privileges that users or teams have. These roles determine what entities (such as tenancies, properties, or tenants) and actions (such as create, read, write, or delete) a user can access or perform.

Field-Level Security:

Field-level security allows administrators to restrict access to specific fields within an entity. This feature is useful when certain fields contain sensitive information that should only be visible to specific users or roles.

Hierarchical Security:

Hierarchical security enables the enforcement of data access based on the organisational structure. Users can be granted access to their own records and records of their subordinates, depending on their position within the hierarchy.

Authentication:

Zavanti natively supports various authentication methods, including username and password, multi-factor authentication (MFA), and integration with Azure Active Directory. Users are authenticated before accessing the system, and authorisation controls determine their level of access based on roles and permissions assigned to them. Role-Based Access Control (RBAC) enables fine-grained access control, ensuring users have appropriate privileges based on their roles and responsibilities.

Data Protection:

Zavanti solution employs encryption techniques to protect data at rest and in transit. Encryption ensures that sensitive information remains secure even if unauthorised access occurs. Encryption at rest can be achieved using technologies like Transparent Data Encryption (TDE), while encryption in transit leverages secure communication protocols such as HTTPS and SSL/TLS. Its up to the client to encrypt the database and store the encryption key securely. For details on Dynamics 365 encryption refer to the below links.

https://learn.microsoft.com/en-us/microsoft-365/compliance/office-365-encryption-in-microsoft-dynamics-365?view=o365-worldwide

https://learn.microsoft.com/en-us/power-platform/admin/manage-encryption-key

Secure Integration:

Zavanti solution offers secure integration options, allowing organizations to connect and exchange data with external systems or services such as REA forms for document and contracts generation. Integration capabilities leverage secure communication protocols and authentication mechanisms to ensure the confidentiality and integrity of data during integration processes.

Audit and Compliance:

Dynamics 365 includes auditing capabilities to track changes to data, system events, and user activities. This helps in identifying any unauthorised access attempts and the detection of any anomalies. The platform also provides features to support compliance with industry-specific regulations, such as GDPR (General Data Protection Regulation).

Zavanti does not recommend Auditing for all entities. The decision to turn on auditing is up to clients business needs.

Integration with Azure Active Directory:

Dynamics 365 integrates with Azure Active Directory (Azure AD), Microsoft's cloud-based identity and access management service. This integration allows for centralized user management and authentication, enabling single sign-on (SSO) across multiple applications.

Data Encryption and Data Loss Prevention:

Dynamics 365 incorporates data encryption in transit and at rest to safeguard information from unauthorized access. Additionally, data loss prevention (DLP) policies can be implemented to prevent the accidental or intentional sharing of sensitive data.

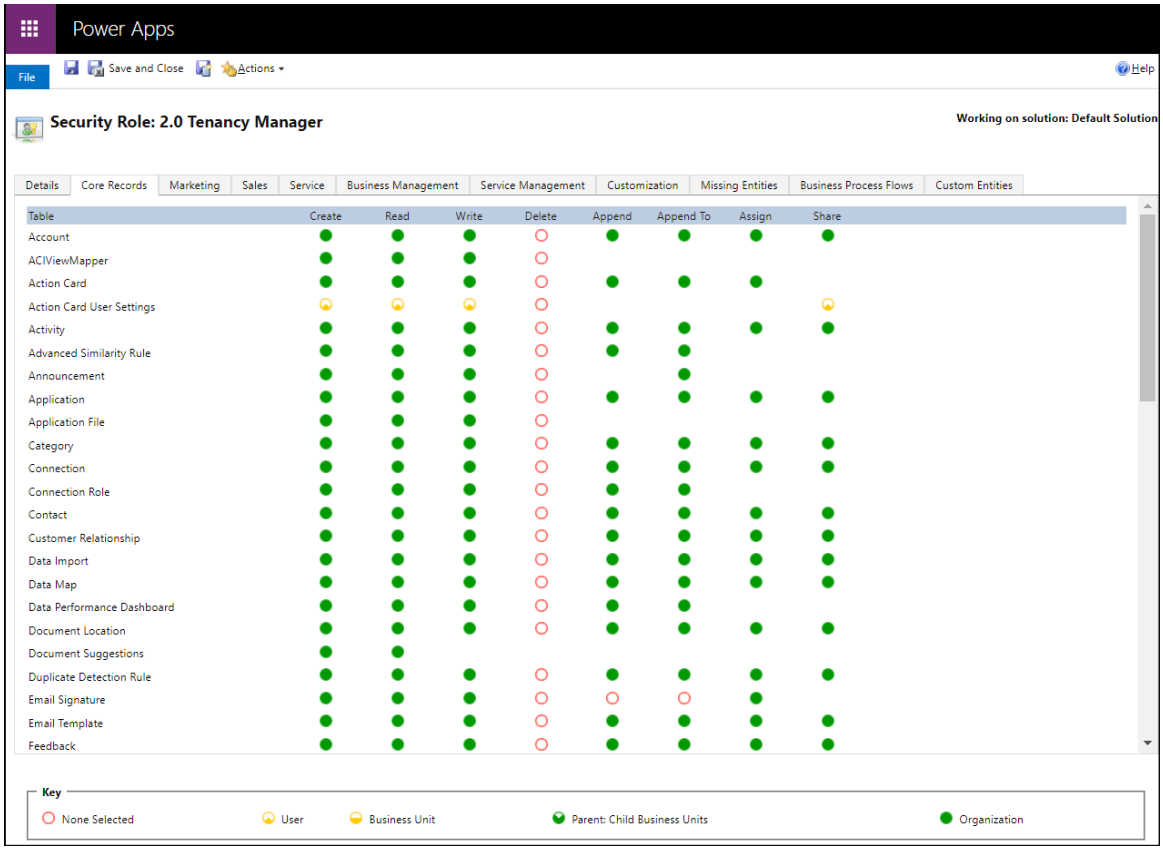
For more information, please refer to the below link –

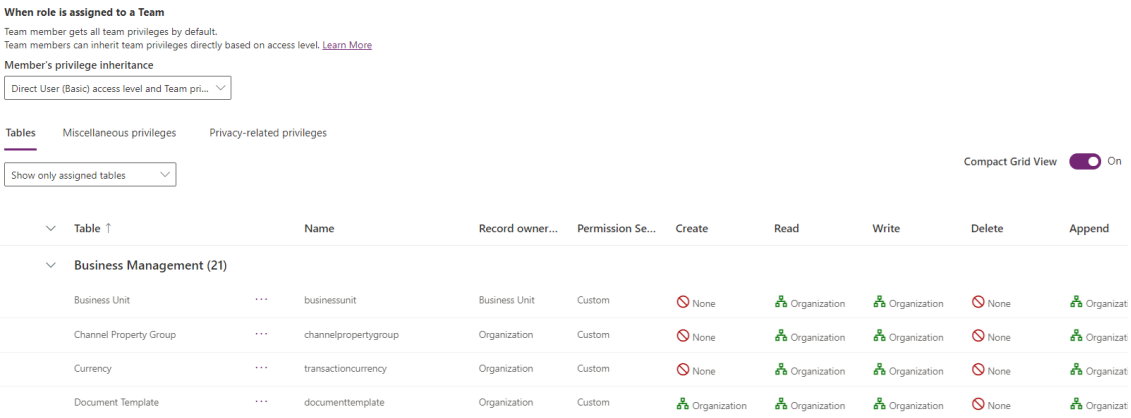
https://learn.microsoft.com/en-us/dynamics365/customerengagement/on-premises/admin/security-concepts?view=op-9-1

Record-Based Security:

Record-based security allows for the restriction of access to specific records within an entity. Administrators can define criteria and rules to determine which records a user can view or modify based on data attributes.

Following are some screen shots of the security role setting in Dynamics 365. For example tenancy manager role can be granular:





Systems Setup Requisites

Dynamics 365

To set up the Zavanti environment we require a user account set up as per the below. Zavanti can assist and do a screen share with the clients IT resource if needed.

1. Create Microsoft 365 AAD account for Zavanti user in your tenant with global administrator role preferably if not assign power platform admin user role to the Zavanti environment. Please forward us the username and temporary password. Although guest users can be created and assigned a D365 security role, there are limitations when it comes to administering D365. For example, guest users cannot be assigned Power Platform Admin roles, hence limits Zavanti for performing critical updates and customisations. For more details on power platform admin centre and roles refer to Power Platform.
2. Assistance is required from clients IT resource that has Global Admin access to set up two new D365 environments. One for production, the other for Sandbox purposes. The systems engineer is to provide D365 URLs to Zavanti. This step is not needed if the user setup in step 1 has been provided power platform admin role. Its up to the clients to name the URL's appropriately.
3. The systems engineer needs to assign a D365 Sales Enterprise license to Zavanti user created in step 1.
4. Zavanti user created in step 1 is to be provided System Administrator role in D365 and has to be an active user for the duration of the contract.
5. During the implementation process Zavanti will need additional log ins set up for consultants to work on. This will be required until the go live date. These users only need to have D365 log in with sys admin access. Zavanti are happy to comply with client security policies like MFA etc.

Dynamics 365 Secret Key and Client ID set up.

In Azure, a secret key and client ID are used to authenticate and authorise applications to access Azure resources like SharePoint and Azure Functions. This step is required because Zavanti uses Auth-Type App to connect to SharePoint. In order to acquire App Id and Client Secret we need to register an App in Azure Active Directory.

Here's a brief overview of these two terms:

Secret Key

A secret key, also known as a client secret, is a secure and confidential string of characters that is used to authenticate an application or service principal to Azure. The secret key is essentially a password that is used to prove the identity of the application or service principal. It should be kept secret and not shared with anyone else.

Client ID

A client ID, also known as an application ID, is a unique identifier assigned to an application or service principal when it is registered in Azure Active Directory (AAD). The client ID is used in conjunction with the secret key to authenticate the application or service principal to Azure.

To use Azure services, an application or service principal must be registered in AAD and must have a client ID and secret key. These credentials are used to obtain an access token, which is then used to authenticate and authorise requests to resources like SharePoint etc.

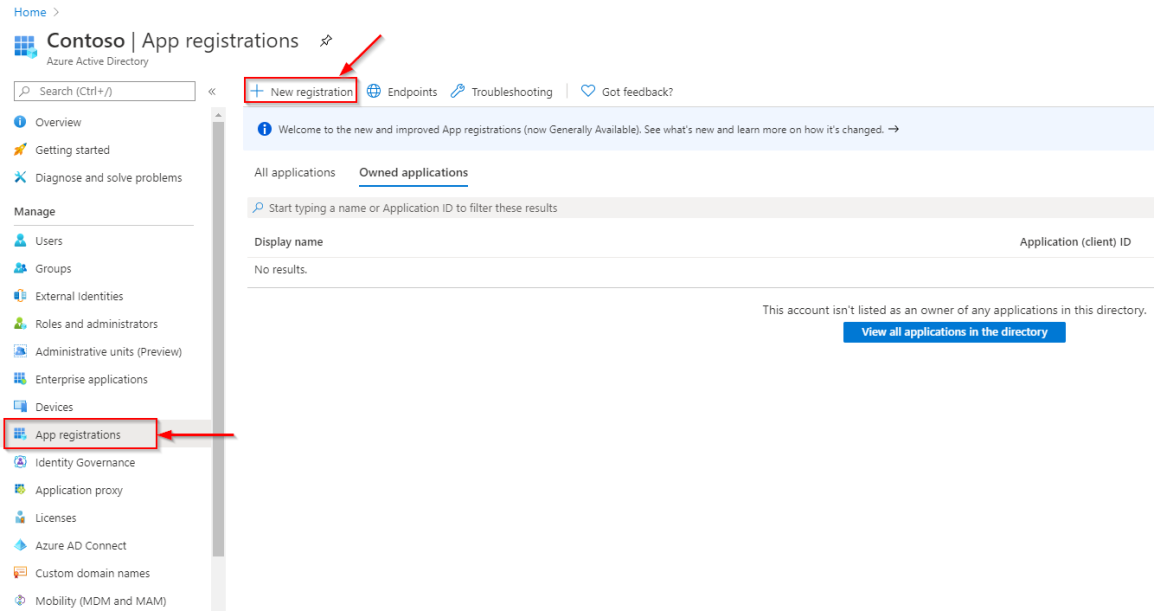
Prerequisites

Before starting this process, you must be logged into Azure portal as a Global Administrator

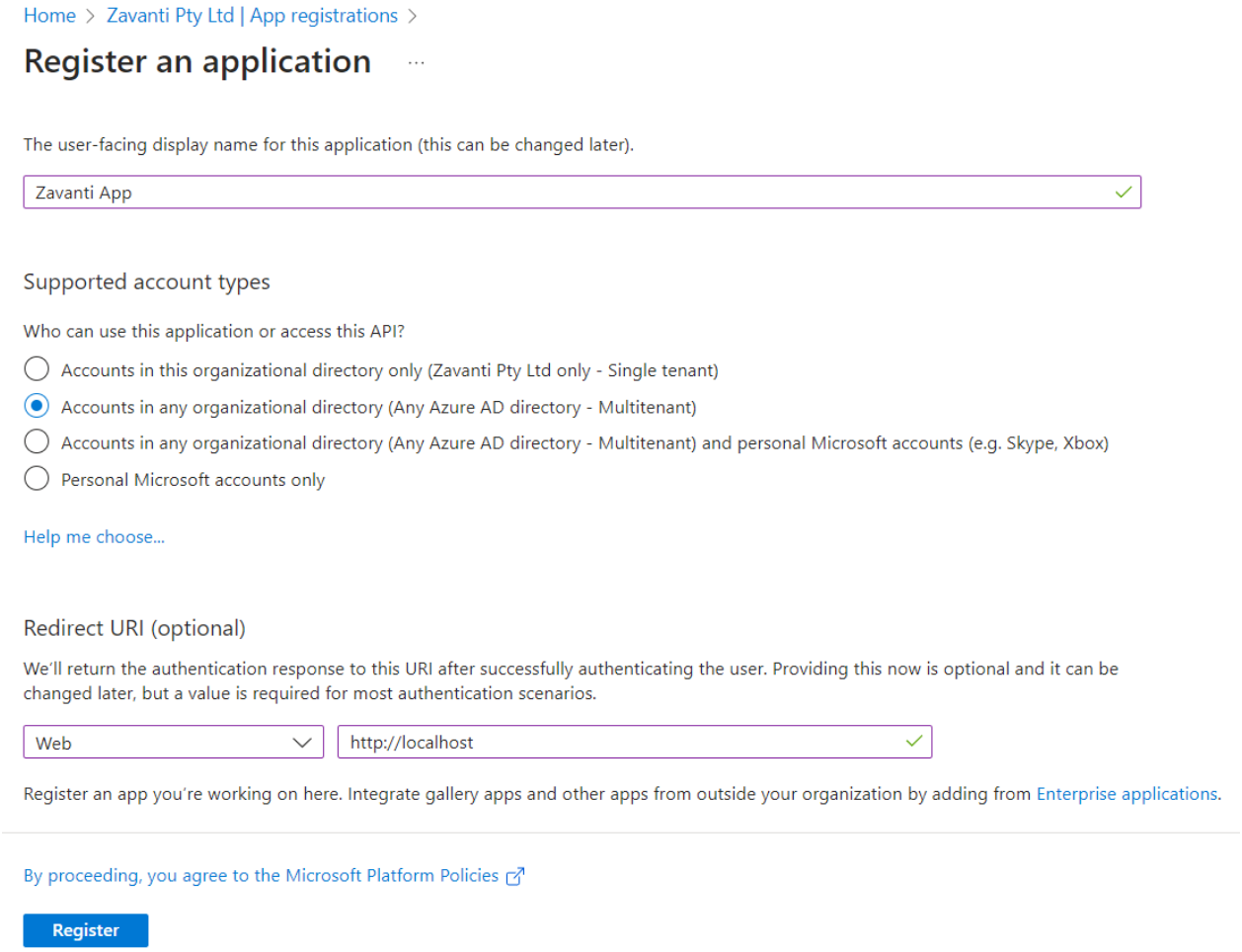
First step: Register a new Dynamics 365 App

Before starting to access API, you must register a new Dynamics 365 App. To do this you will create a unique App ID with specific right to access your dynamics 365.

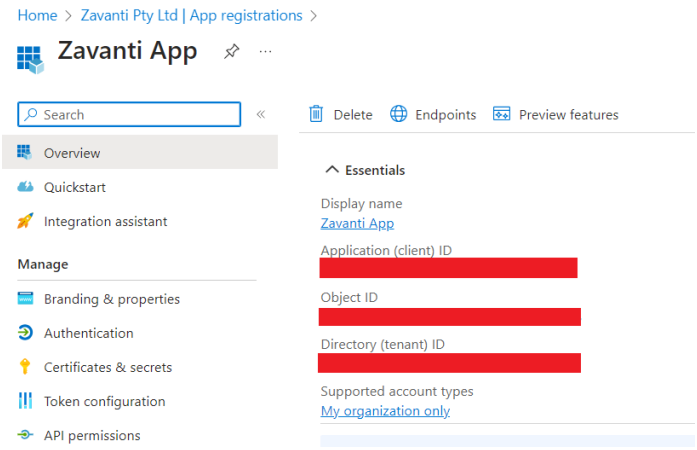
1. First you need to connect to **Azure Portal** with administrator credentials.
2. From the homepage, click on *Azure Active Directory -> App Registration-> New Registration*



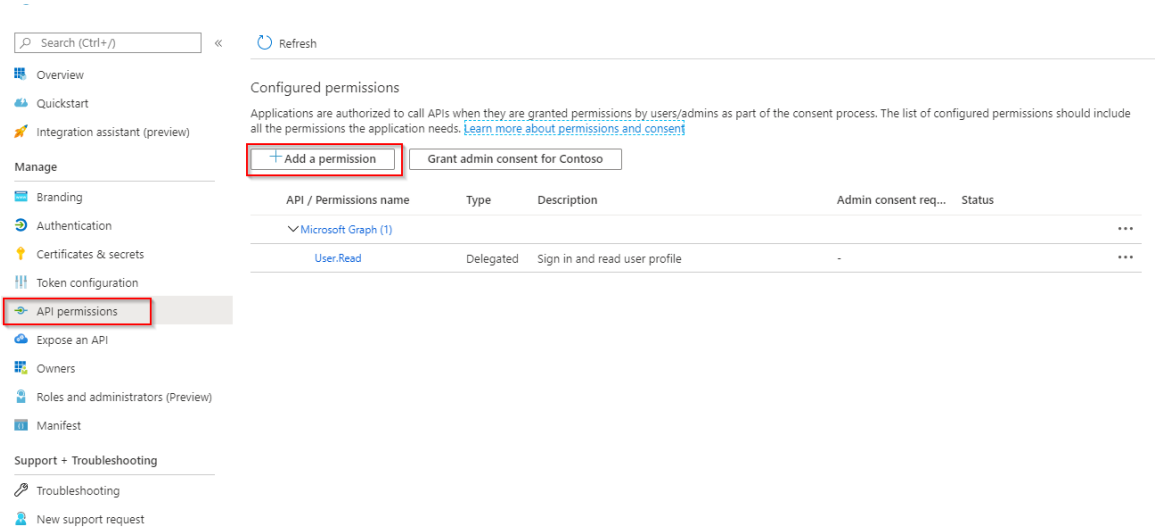
3. In the Application registration form, you must enter some information:
   - Name: this is the name of the registration app. Enter a meaningful one.
   - In the Supported account types part, select *Accounts in any organizational directory*.
   - Set the redirectUrl. In our case, let's use *Web* and http://localhost



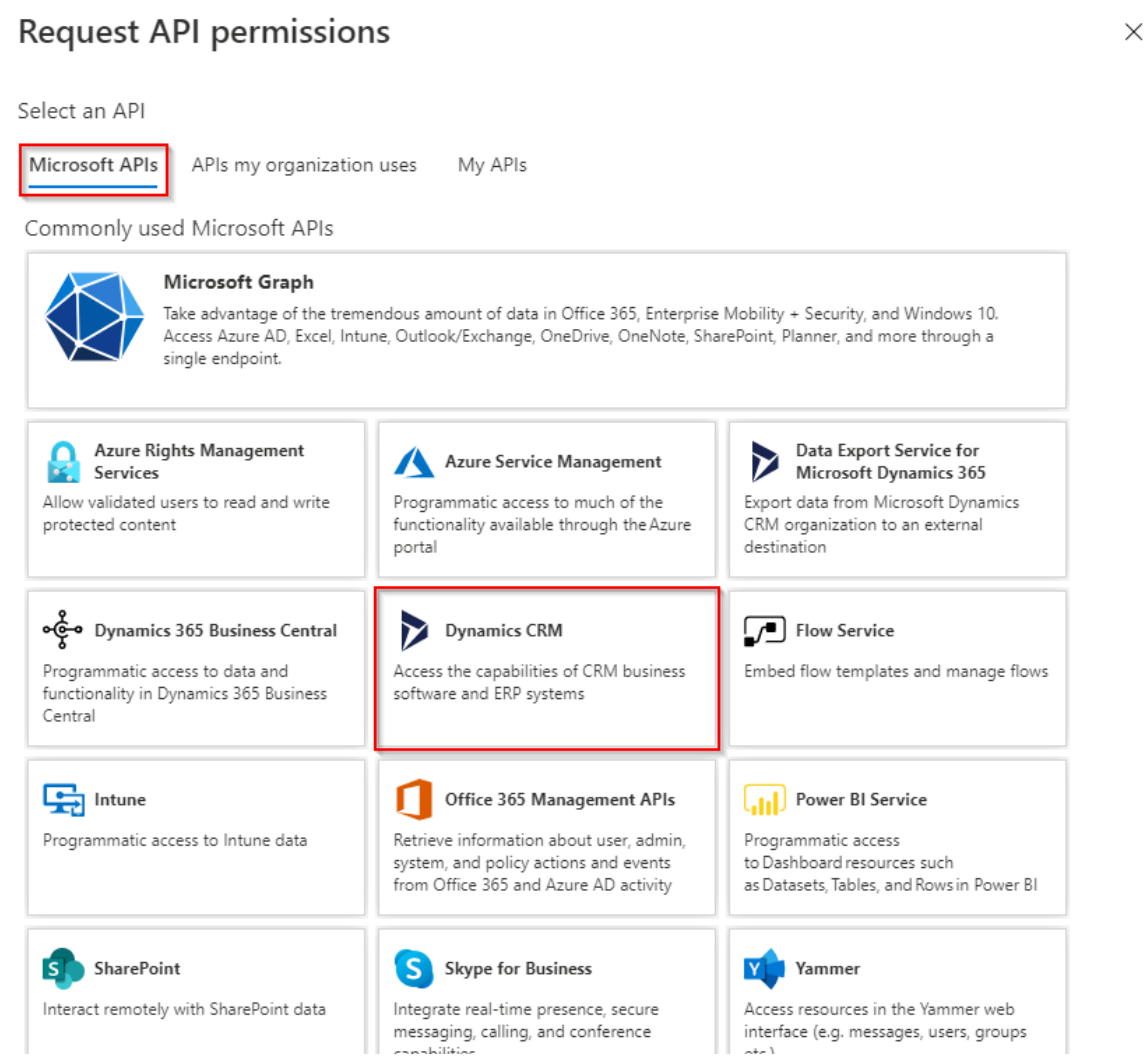
4. Then click on Register.
5. nce your application is registered, you can find the following from the overview.
   - the application id
   - The tenant id
6. Copy and save these somewhere as you will need it later.



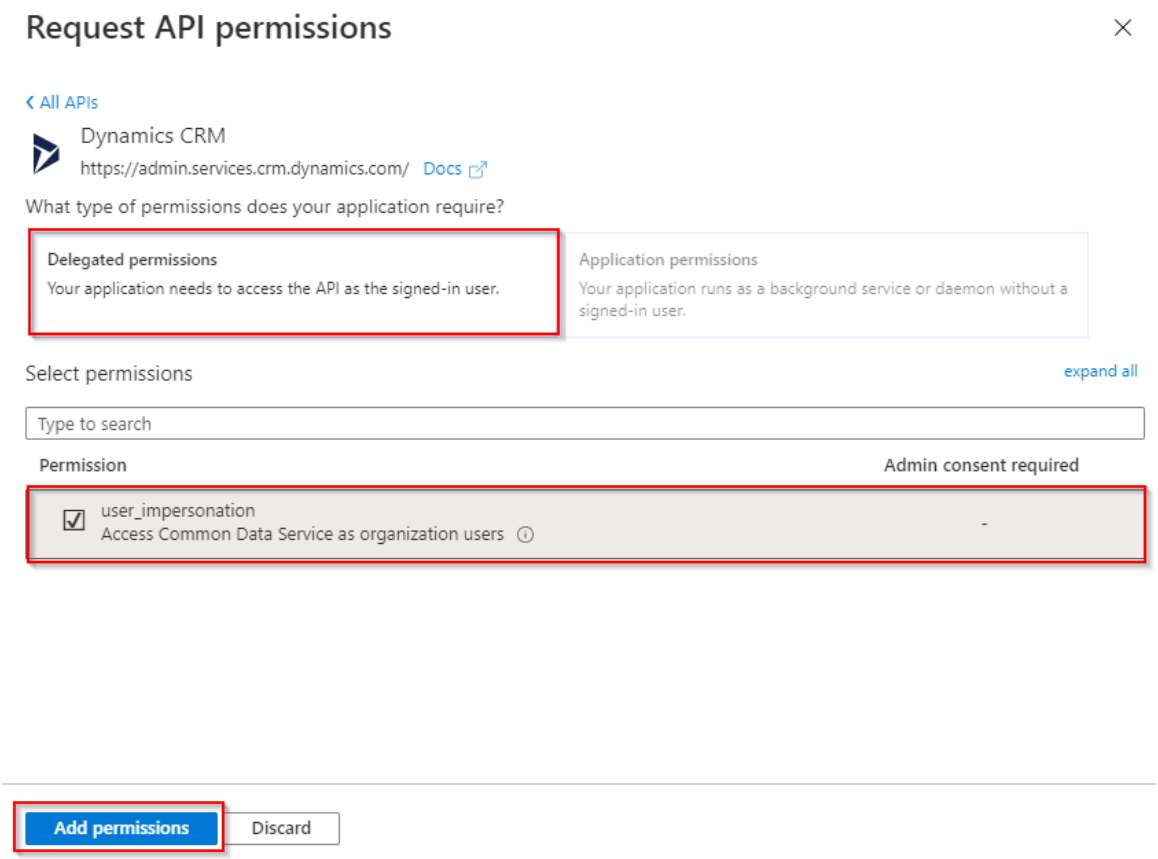
7. Now go on API permissions



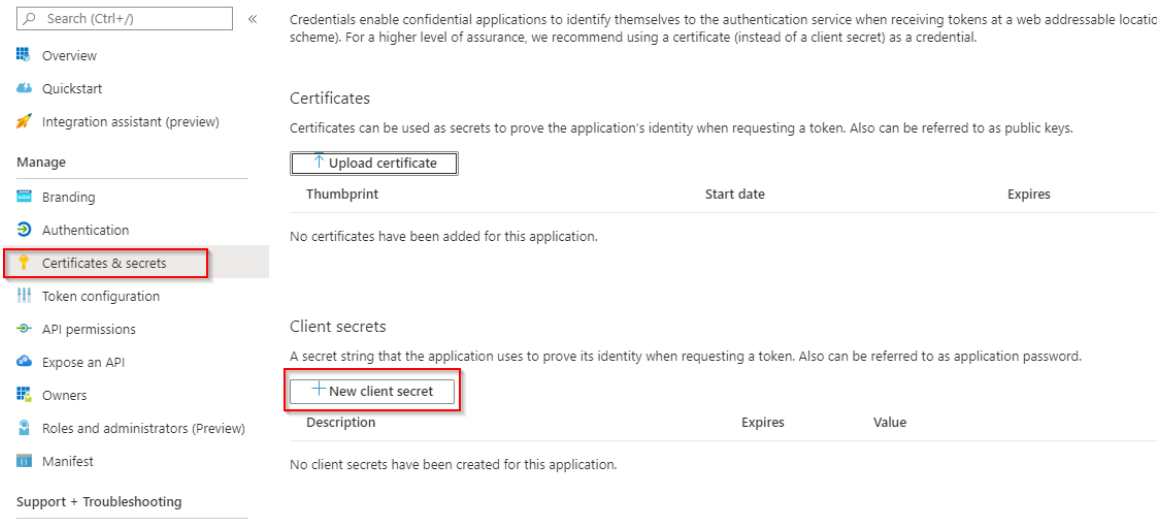
8. Click on Add a permission.



9. Select Dynamics CRM under the API Microsoft Graph tab.

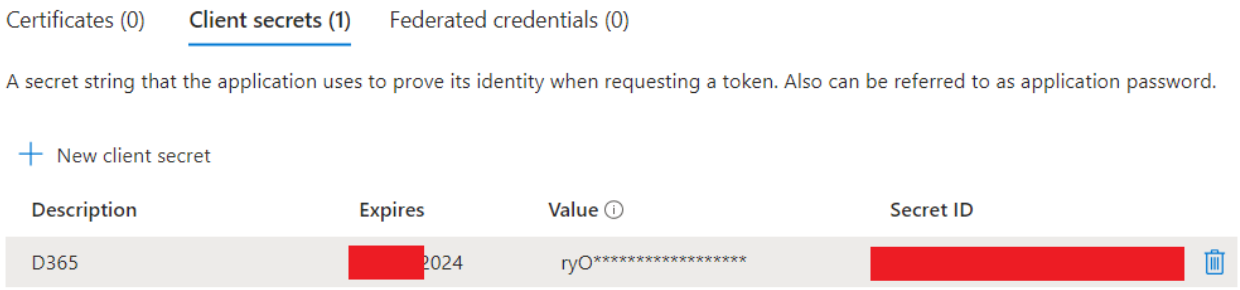


10. Click on *Delegated permissions*, check the options and click on *Add permissions*.
11. Now Click on *Certificates & Secrets* and create a new client secret.



12. Add a name and define the expiration duration of your secret value.
13. Then click onAdd.

14. Your client secret is now created. Copy and keep its value somewhere safe because you won't be able to copy again once the page is reloaded.



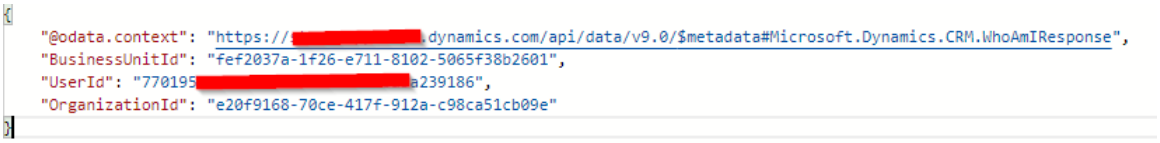
15. Your app is successfully registered and setup.
16. You will now be able to get your token and contact dynamics 365 API.

Second Step: Get the access token.

1. First, we must execute a POST web request with several information in the body in order to get a token.
   - Request URL (as POST):
     https://login.windows.net/<ID_TENANT>/oauth2/token
   - You can get the tenant id in the overview of the application created in the Azure portal
   - Body:
   - grant type = the string "client_credentials"
   - ClientID = The application id. You can get in the overview of the application created in the Azure portal.
   - client secret = The client secret generated in the portal azure
   - resource = the URL of your CRM (https://xxxxxxx.crm6.dynamics.com/)
   - Response :
     This request should return a Json string containing the token:

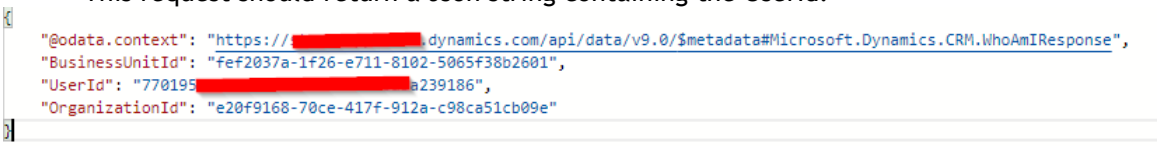
2. Test: WhoAmI
   - Then, you just have to use this token as “Bearer” in the header of any API CRM call requests.
   - Let’s take an example with the WhoAmI method which is supposed to return the id of the calling user.
   - Request URL (as GET):
     https://<CRM_URL>/api/data/v9.0/WhoAmI()
   - Header:

Bearer Token : The access token get in the second



Response :

This request should return a Json string containing the UserId:



SharePoint

SharePoint is a web-based collaboration and document management platform. It is designed to enable teams and organisations to share, store, organise, and manage information and documents in a centralised and secure manner. Zavanti housing system uses SharePoint as the document repository and leverages out of the box integration between Dynamics 365 and SharePoint. The Zavanti Housing solution has built in workflows that create a folder structure in SharePoint automatically. For example, when a new property or tenancy is created the system will generate a folder within SharePoint.

For more information on the set up of out of the box integration please refer to:

https://learn.microsoft.com/en-us/power-platform/admin/set-up-sharepoint-integration

Integration Architecture

Introduction:

The integration between Dynamics 365 and SharePoint enables seamless collaboration, document management, and data synchronization between the two platforms. This section outlines the approach, components, and workflows for integrating Dynamics 365 and SharePoint.

Integration Scenarios:

The integration between Dynamics 365 and SharePoint supports various scenarios, including but not limited to:

- Document Management: Store and manage documents related to Dynamics 365 records within SharePoint document libraries.
- Collaboration and Communication: Enable teams to collaborate on documents, share information, and communicate using SharePoint features such as document co-authoring, versioning, and comments.
- Data Synchronization: Sync data between Dynamics 365 and SharePoint to ensure consistency and eliminate data duplication.
- Workflow Automation: Trigger workflows in SharePoint based on events or changes in Dynamics 365 records.

Integration Components:

1. **Dynamics 365:**

   Entity Mapping: there are several Dynamics 365 entities and fields that require integration with SharePoint. For example, property, application, Tenancy etc

   Integration Triggers: Configure event triggers within Dynamics 365 to initiate the integration process based on specific events, such as record creation, update, or deletion.

2. **SharePoint:**

   Document Libraries: Create SharePoint document libraries to store and organize documents related to Dynamics 365 records.

   Lists and Columns: Define SharePoint lists and columns to capture additional data associated with Dynamics 365 records.

   SharePoint Connectors: Utilise out of the box SharePoint connectors to enable seamless communication between Dynamics 365 and SharePoint.

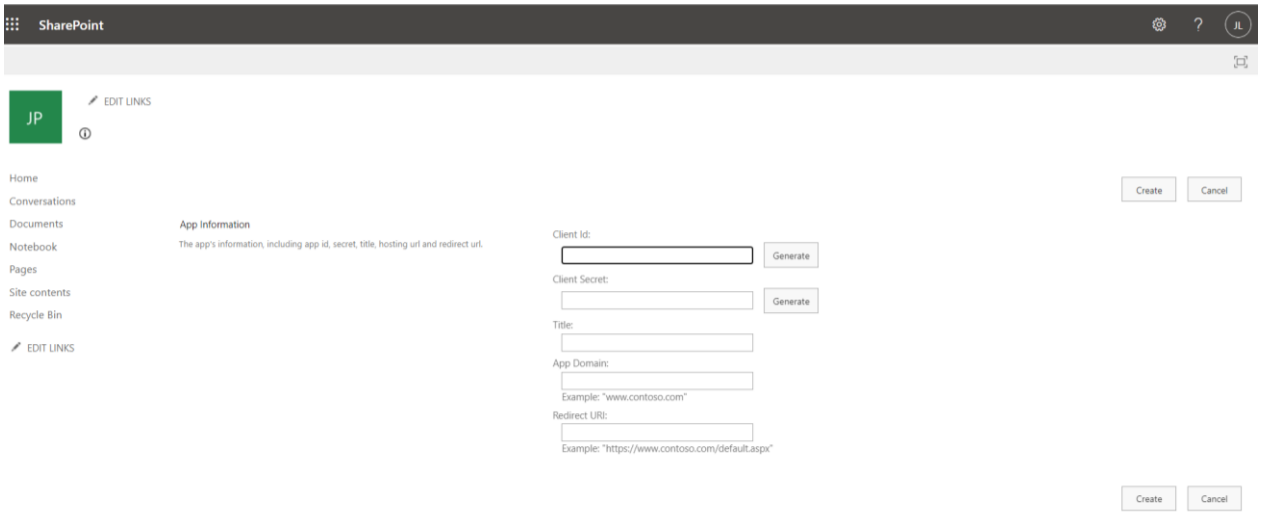
Set Up Integration

Refer to the below Microsoft URL as a reference –

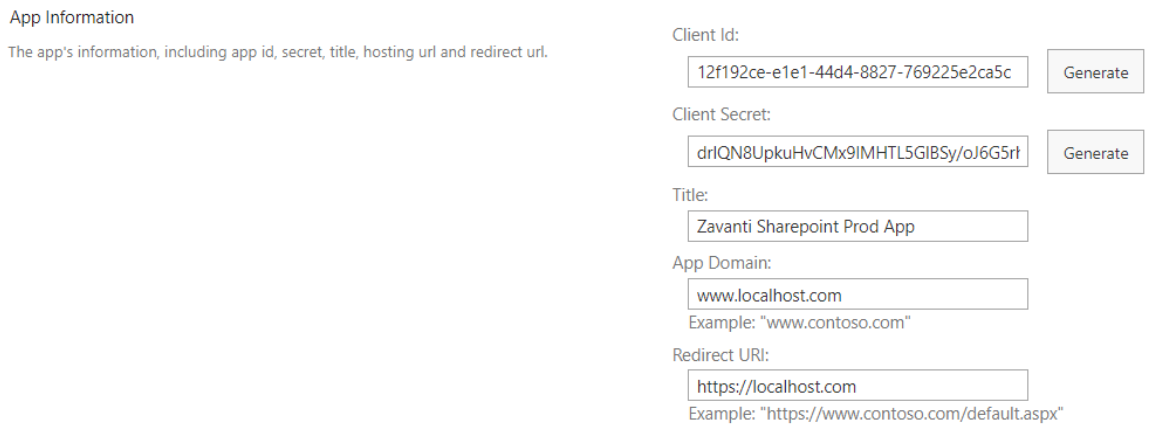
https://learn.microsoft.com/en-us/sharepoint/dev/solution-guidance/security-apponly-azureacs

1. Engineer creates 2 new sites on SharePoint, one for Production, the other for Sandbox.
2. Enable SharePoint access to Zavanti user.
3. Provide SharePoint URL to Zavanti so that integration with D365 can be set up by Zavanti.
4. Register Sharepoint Add-in using appRegNew Form
5. Navigate to – the site collection URL set up in step 1.

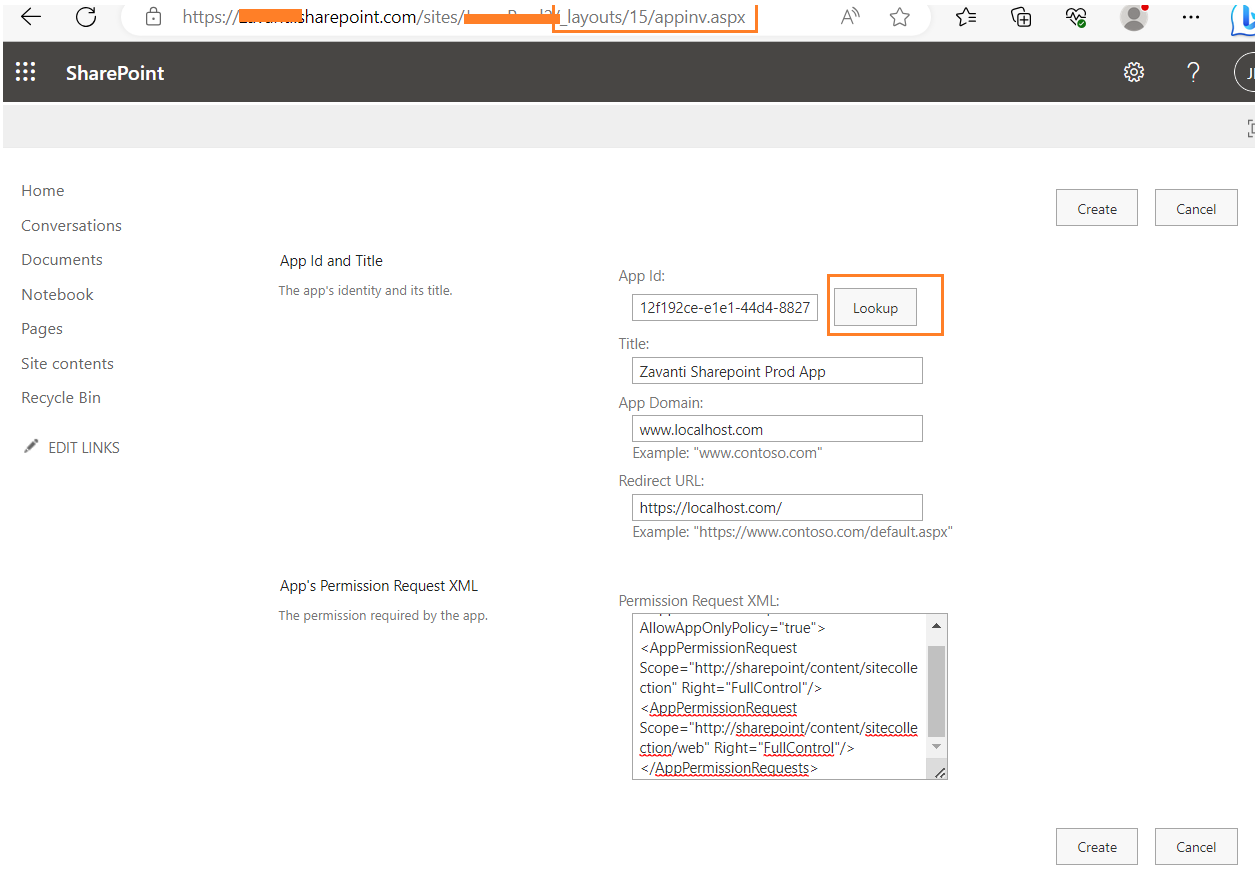
   [Site Collection URL]/_layouts/15/AppRegNew.aspx



6. Click Generate Client ID and for Client Secret. Insert appropriate Title, App Domain and Redirect URL



7. Click Create to proceed with the Sharepoint App Add-in creation process.
8. Please provide Zavanti the Client ID and Client Secret set up in step 5. We need this to authenticate SharePoint integration from Dynamics.
9. Next step is granting permissions to the newly created principal. Since we're granting tenant scoped permissions this granting can only be done via the appinv.aspx page on the tenant administration site. You can reach this site via **Error! Hyperlink reference not valid.**
10. Once the page is loaded add your client id (created in step 5) and click on look up the created principal.

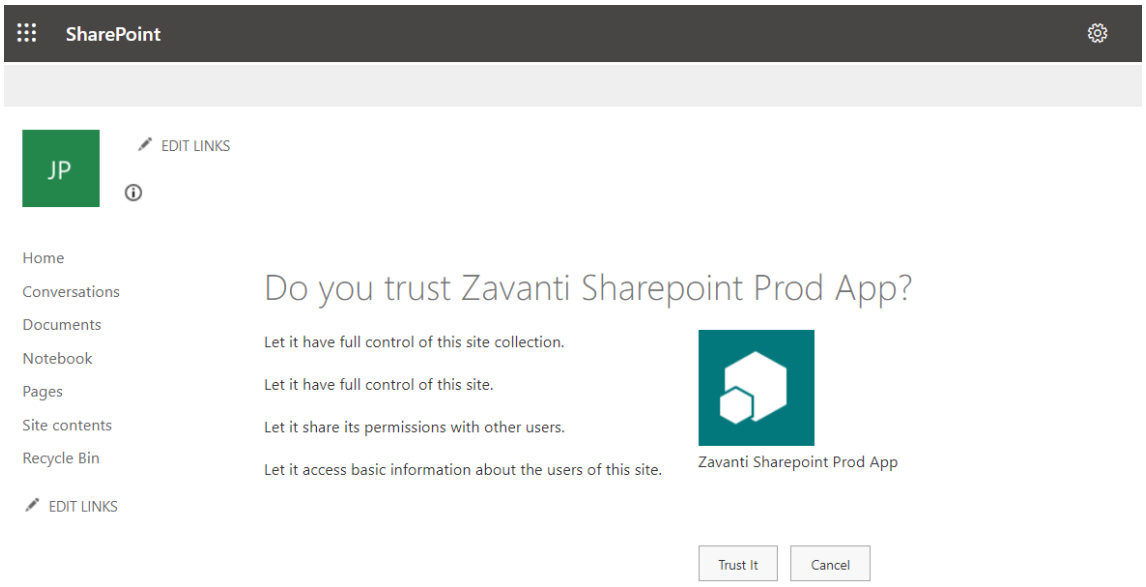


To grant permissions, you'll need to provide the permission XML that describes the needed permissions. Since this application needs to be able to access to the Zavanti SharePoint site (created in step 1)

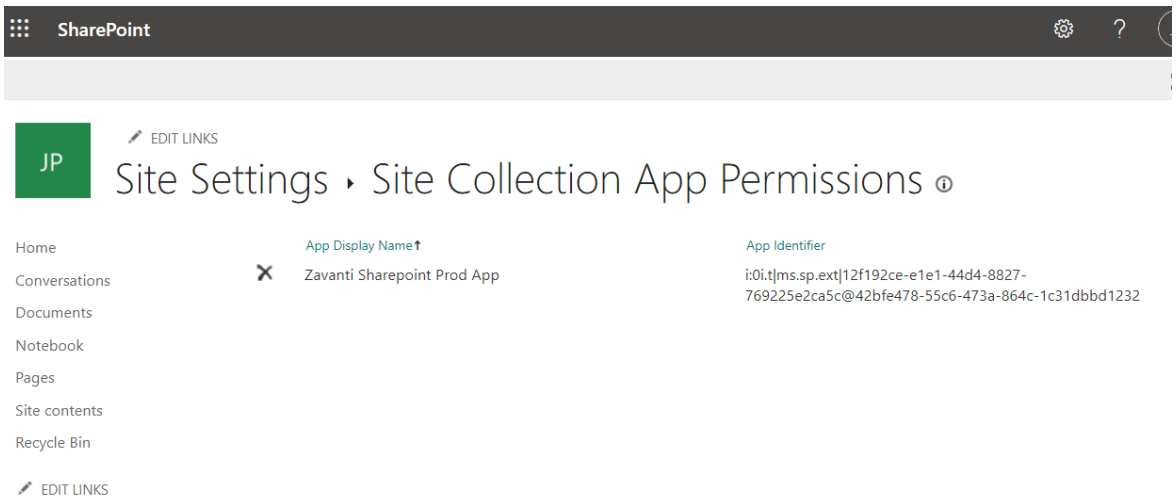
XMLCopy

```
<AppPermissionRequests AllowAppOnlyPolicy="true">
<AppPermissionRequest Scope="http://sharepoint/content/sitecollection" Right="FullControl"/>
<AppPermissionRequest Scope="http://sharepoint/content/sitecollection/web"
Right="FullControl"/>
</AppPermissionRequests>
```

When you click on Create, you'll be presented with a permission consent dialog. Press Trust It to grant the permissions:



In order to confirm the permission has been granted to the add-in.Navigate to [Site Collection URL]/_layouts/15/appprincipals.aspx



Azure Functions

Azure Functions is a serverless computing platform provided by Microsoft Azure. It allows developers to write small, event-driven pieces of code that can be executed in response to various events such as HTTP requests, queue messages, blob storage changes, and timer events. Azure Functions can be written in several programming languages, including C#, Java, JavaScript, Python, and PowerShell.

Dynamics 365 imposes a default SQL timeout of 2 minutes for executing SQL queries. In scenarios where long-running or complex queries exceed this timeout, Azure Functions can be used to overcome this limitation. This section outlines the approach and components involved in utilising Azure Functions to handle time-consuming queries on behalf of Dynamics 365. Zavanti leverages Azure Functions to overcome large data/query processes like payment imports, adding locations and assets to property based on master data etc.

Azure Functions Architecture

Integration Workflow

Query Execution:

Dynamics 365 triggers an Azure Function by passing the SQL query or relevant query parameters. Azure Function receives the query and initiates the execution.

Long-Running Query Execution in Azure Function:

Azure Function connects to the SQL database using appropriate database connectors or libraries. The SQL query is executed within the Azure Function environment.

Handling Query Results:

Azure Function receives the query results from the SQL database. If the query execution completes within the timeout limit, the results are returned to Dynamics 365 directly. If the query execution exceeds the timeout limit, a partial result or a notification indicating the ongoing query execution is returned to Dynamics 365.

Continuation and Result Retrieval:

Dynamics 365 acknowledges the partial result or notification received from the Azure Function. Dynamics 365 triggers subsequent requests to the Azure Function to retrieve the remaining query results. Azure Function continues the query execution from where it left off and returns additional results as requested by Dynamics 365. This process continues until the complete query result is received by Dynamics 365.

Dynamics 365 Dataverse

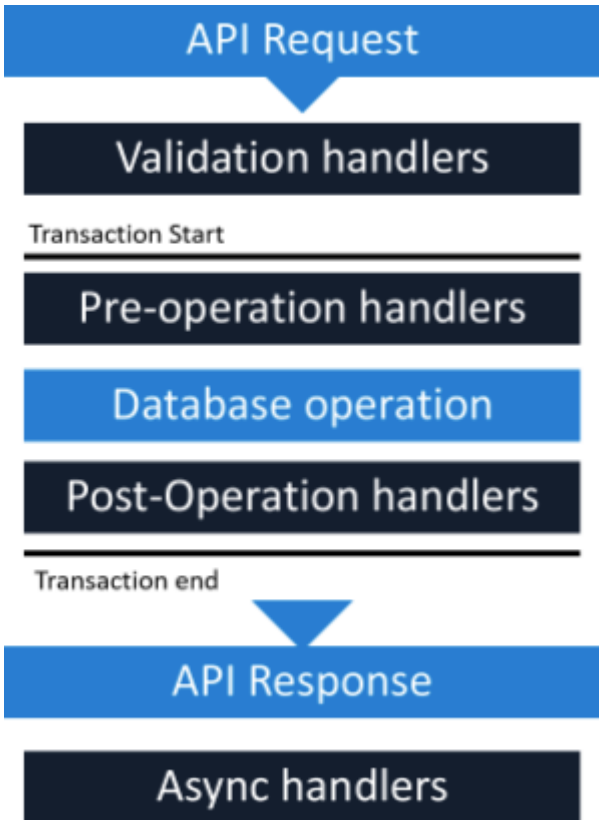
Dataverse lets you securely store and manage data that's used by business applications. Data within Dataverse is stored within a set of tables. A table is a set of rows (formerly referred to as records) and columns (formerly referred to as fields/attributes). Each column in the table is designed to store a certain type of data, for example, name, age, salary, and so on. Dataverse includes a base set of

standard tables that cover typical scenarios, but you can also create custom tables specific to your organization and populate them with data by using Power Query. App makers can then use Power Apps to build rich applications that use this data.

Dynamics 365 Plugins

Dataverse provides the ability to write code that sits between the API and the data. This code, written in .NET, is referred to as a plug-in. Because the plug-in sits between the API and the data, it enforces the same logic on every app. Plug-ins can be synchronous or asynchronous, and perform the following tasks:

- Return errors to the user.
- Query Dataverse data to evaluate logic to perform.
- Perform data operations.
- Perform outbound HTTPS requests.
- Plug-ins are registered at points in the event pipeline, which are illustrated here.



Within the event pipeline, the following events can occur:

- Requests and Responses can be examined and rejected or manipulated in several steps of the event pipeline.
- Validation handlers can throw custom exceptions to reject operations that your logic considers invalid.
- Pre-operation handlers can modify requests before the database operation.
- Post-operation handlers can modify responses.
- Async handlers perform automation after the response is returned.

One constraint with plug-ins is that they must be self-contained. If integration code requires references to other libraries, integration can be done by using Azure Functions.

Azure Functions provides a serverless code execution option for business and integration logic.

Functions are triggered by a call from an external system, service, or code such as Zavanti Property Plugin. For Dataverse, that trigger can come directly from Dataverse using Service Bus, a webhook, or a call from a plug-in. Additionally, the Azure Functions call can be initiated via a flow in either Logic Apps or Power Automate that involves the Dataverse connector.

Miscellaneous information regarding Azure Function Set up.

Creation of a new Azure Subscription, in order to support the provisioning process of Azure function please note that there will be separate billing and cost involved in creating a new subscription in Azure. The Zavanti user will need to be assigned as a co-owner of the subscription. If this is not possible, we will need your IT representative's assistance in setting this up.

Information needed to create a new Function App

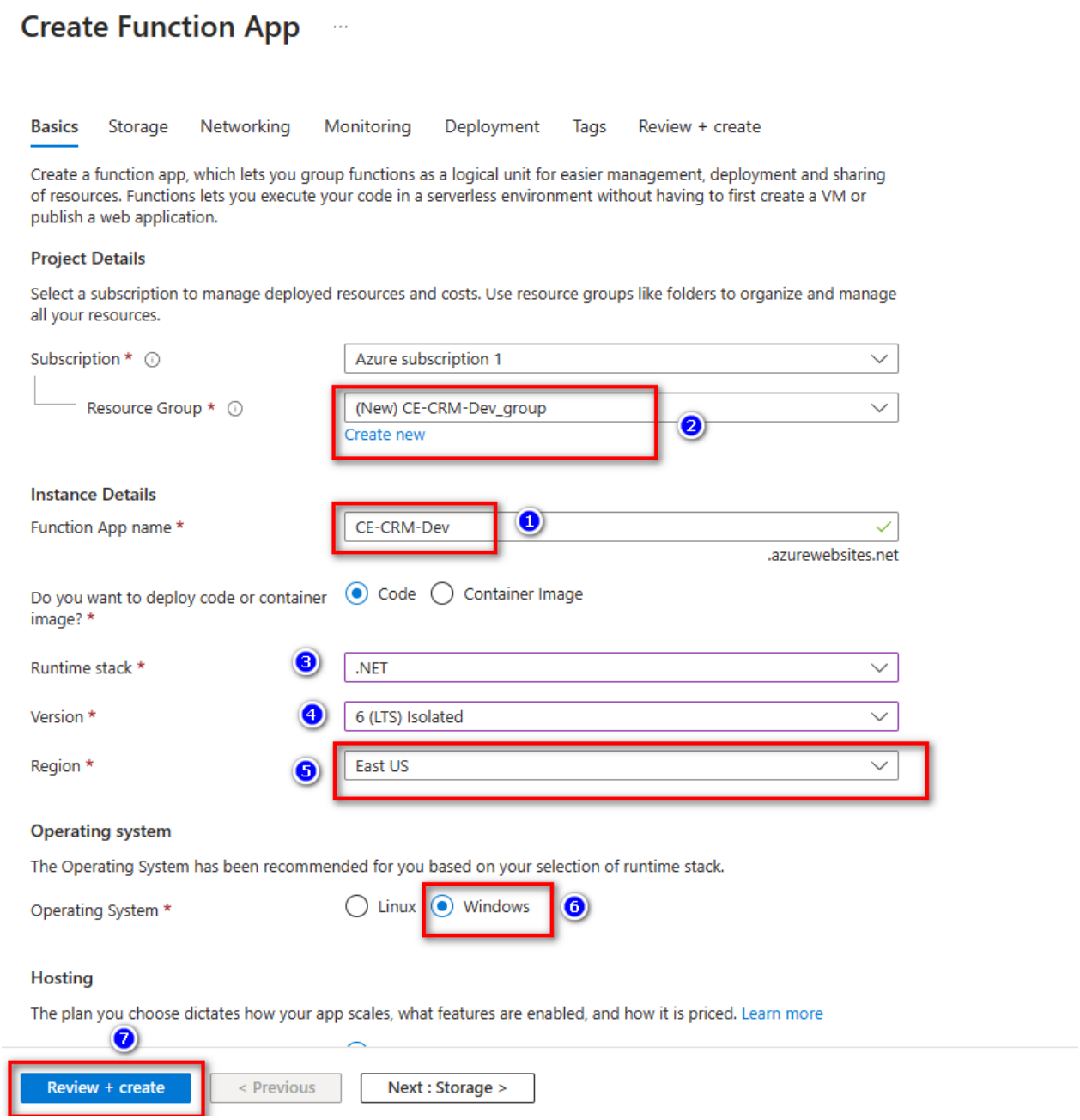
Please note its best practice to create separate Function App for production and sandbox. Naming conventions is up to the client.

Function App creation process

Prerequisites:

- Have accessible Microsoft Dynamics 365 instance.
- Have access to Azure environment with sufficient privileges to create Azure Function

1. Step 1: Create Function App
    - Resource Group -> Please select existing or create new. If not, please enter Function App name (1) -> then the Resource Group will be auto populated.
    - Region -> Please select the appropriate region.
    - Please click button "Review + create"
    - You don't need to enter information in other tabs (storage, networking...) leave as default.



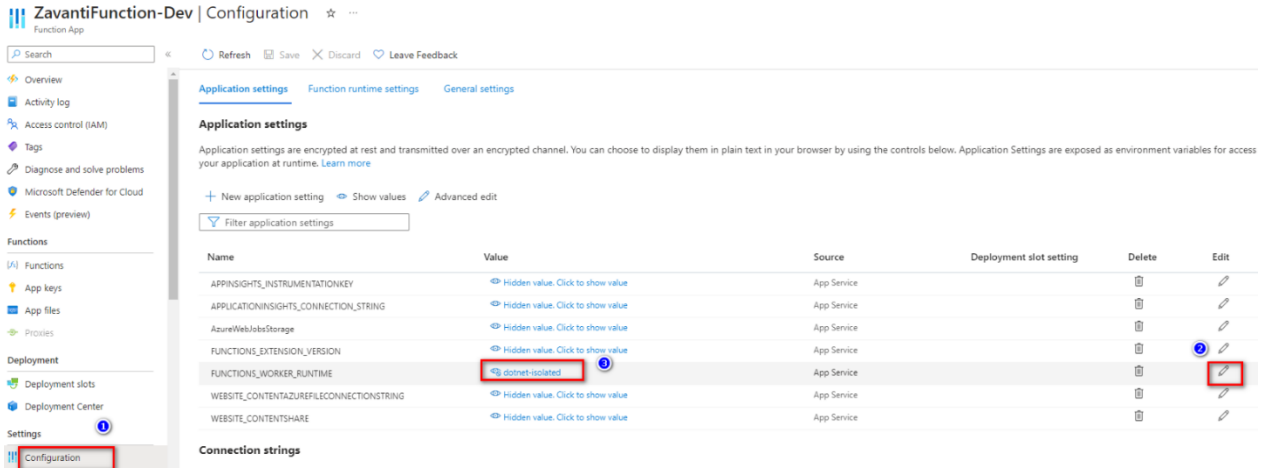
Relevant Microsoft Reference –

https://learn.microsoft.com/en-us/dynamics365/customerengagement/on-premises/developer/use-webhooks?view=op-9-1

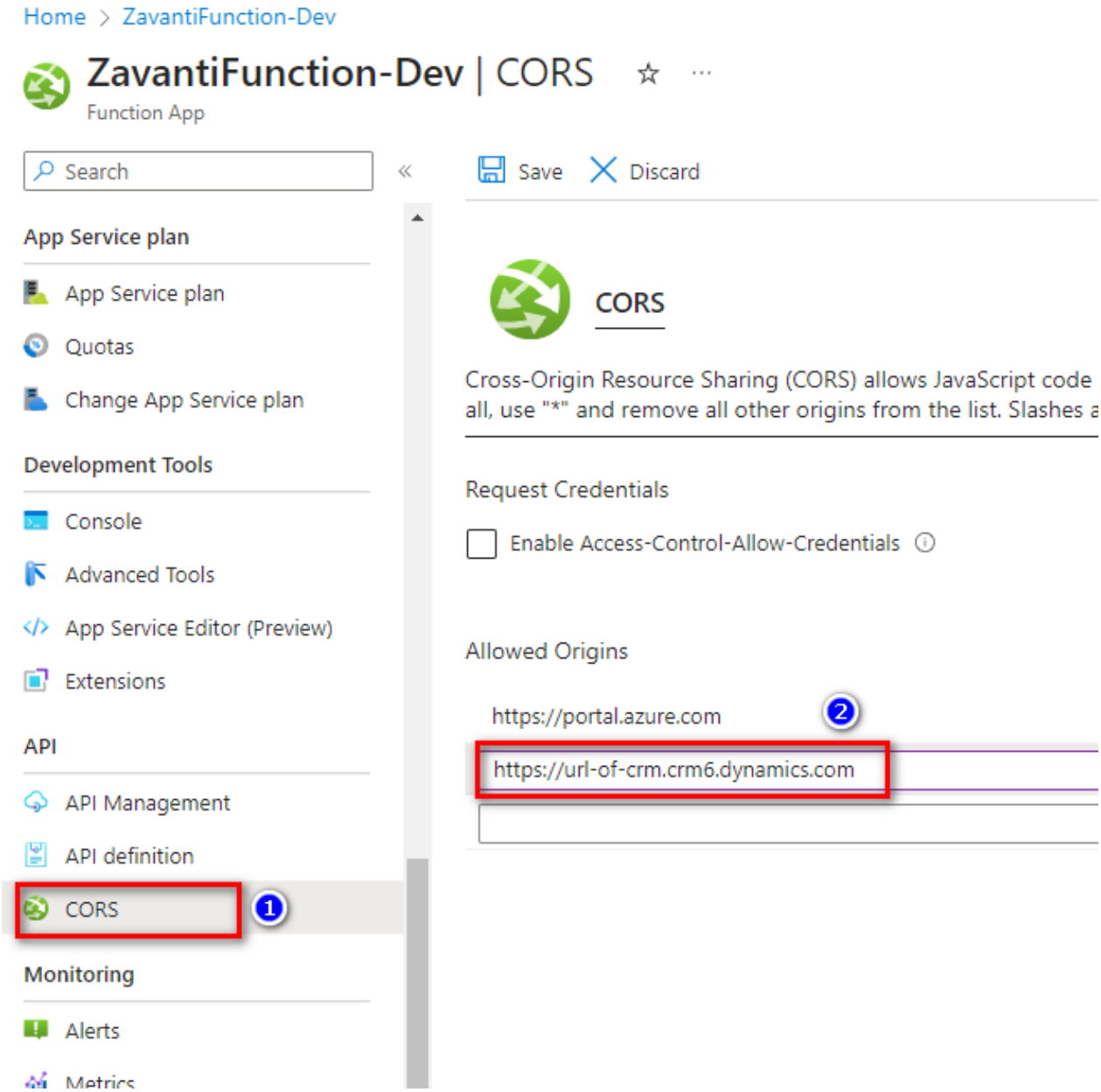
https://learn.microsoft.com/en-au/azure/azure-functions/functions-scale?WT.mc_id=Portal-WebsitesExtension

2. Step 2: Configuration of Azure Function after creation process.
   - On the left side of Function App menu, navigate to Settings → Configuration.
   - Under Application Settings, locate the Functions Worker_RunTime.
   - Click Edit.

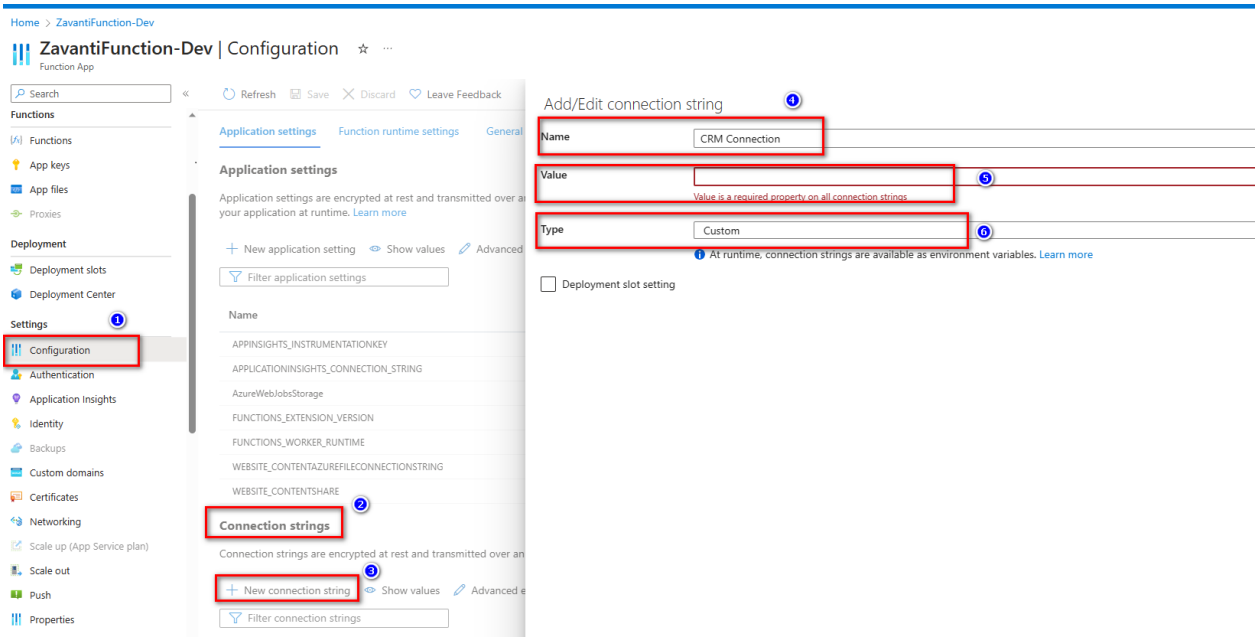
- Set the FUNCTIONS_WORKER_RUNTIME: dotnet-isolated



3. Step 3: Add Dynamics CRM URL to CORS
   - Navigate to API section under the function App.
   - Select CORS settings.
   - Specify Full CRM URL to be part of the Allowed Origins.

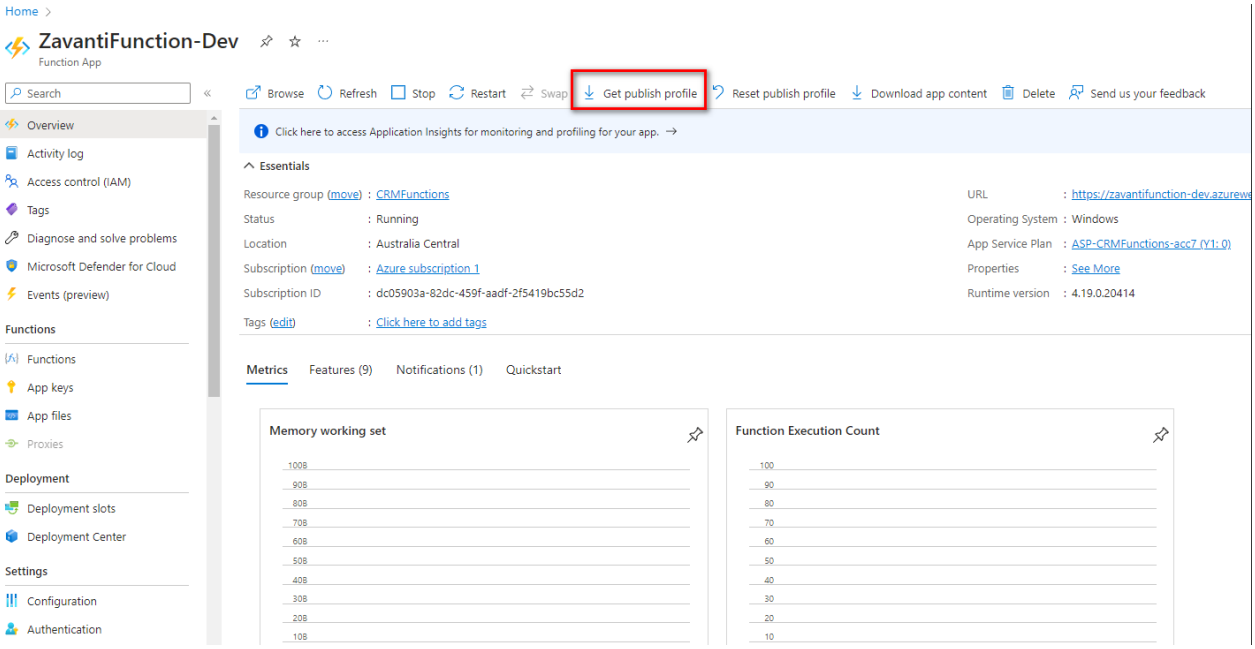


4. Step 4: Create CRM Connection String
   - Navigate to Configurations under the Settings menu as indicated at screenshots below.
   - Locate the Connection String section on the page
   - Select “+ New Connection String”
   - Specify a new Connection name
   - Provide a Connection Value. Please request connection value from Zavanti.
   - The connection string value might look something like the following
   - AuthType=ClientSecret;url=https://xxxx.crm6.dynamics.com;ClientId=xxxxxxxxxxxxxxxxxx ;ClientSecret=xxxxxxxxxxxxxxxxxxxxxxx;
   - Leave the Type = Custom.

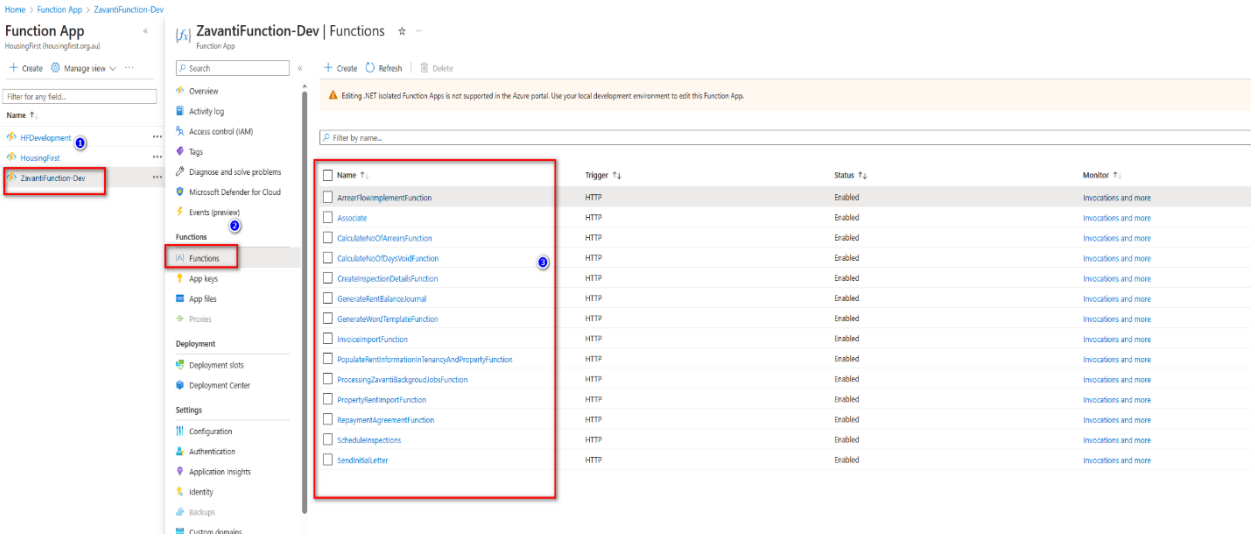

Home > ZavantiFunction-Dev
ZavantiFunction-Dev | Configuration
Function App
Search
Functions
Functions
App keys
App files
Proxies
Deployment
Deployment slots
Deployment Center
Settings
Configuration
Authentication
Application Insights
Identity
Backups
Custom domains
Certificates
Networking
Scale up (App Service plan)
Scale out
Push
Properties
Refresh
Save
Discard
Leave Feedback
Application settings
Function runtime settings
General
Application settings
Application settings are encrypted at rest and transmitted over
your application at runtime. Learn more
New application setting
Show values
Advanced
Filter application settings
Name
APPINSIGHTS_INSTRUMENTATIONKEY
APPLICATIONINSIGHTS_CONNECTION_STRING
AzureWebJobsStorage
FUNCTIONS_EXTENSION_VERSION
FUNCTIONS_WORKER_RUNTIME
WEBSITE_CONTENTAZUREFILECONNECTIONSTRING
WEBSITE_CONTENTSHARE
Connection strings
Connection strings are encrypted at rest and transmitted over an
New connection string
Show values
Advanced e
Filter connection strings
Add/Edit connection string
Name
CRM Connection
Value
Value is a required property on all connection strings
Type
Custom
At runtime, connection strings are available as environment variables. Learn more
Deployment slot setting

5. Step 4: Get Publish Profile

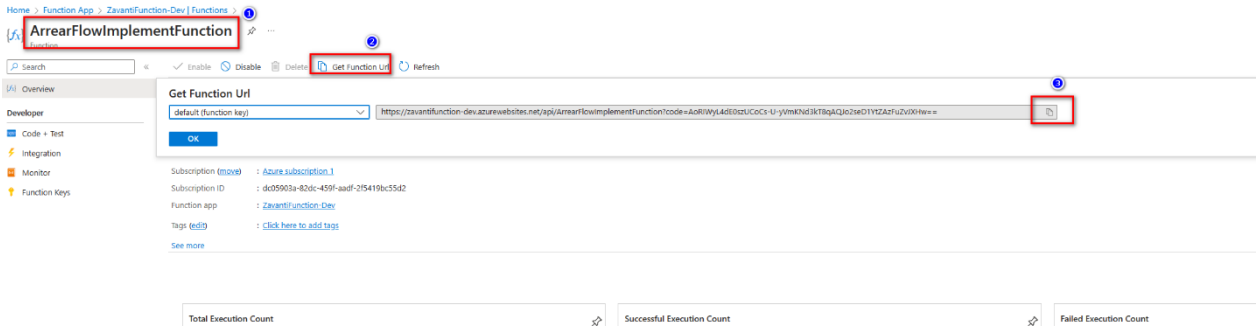
   - Go to the function app that was just created -> click “Get publish profile” button -> please download the profile file and send to Zavanti to deploy.



- We need this file to deploy and establish the connection between azure function and Dynamics 365. Once this is done, Zavanti will inform the client to proceed the next step below.
- Log on to the azure portal and select azure function, then select the function app created, on the right under functions click on Functions, click on the function itself.



- click on “get function URL” for prod and sandbox. Please send this URL to Zavanti.



Please repeat this process of getting the function URL for all the remaining functions listed. Zavanti will deploy multiple functions as per business requirements. This could include process like adding assets to locations, payment import process etc.

Please send all the URL to Zavanti.

Microsoft Exchange

Zavanti Housing also provides the functionality to send out email notifications to intended recipients for example tenant communications, work orders to be sent to contractors, rent review letters sent out to tenants. These email notifications will be generated within Dynamics 365 as a result of events being triggered and the incoming or outgoing flow of the email are handled through queue management in Dynamics 365.

Integration Architecture: Dynamics 365 and Exchange

The integration between Dynamics 365 and Exchange enables seamless communication, email tracking, and synchronization of emails and appointments between the two platforms. This integration architecture outlines the approach, components, and workflows for integrating Dynamics 365 and Exchange.

Integration Scenarios:

The integration between Dynamics 365 and Exchange supports various scenarios.

- Email Tracking: Capture and track emails sent from Dynamics 365 and associate them with relevant records for example tenants.
- Calendar Synchronization: Sync appointments and meetings between Dynamics 365 and Exchange calendars for example tenant visits and inspections.
- Contact Synchronization: Ensure consistency between Dynamics 365 and Exchange contact information.
- Workflow Automation: Trigger actions or workflows in Dynamics 365 based on events or changes in Exchange for example work orders sent to contractors, rent review letters sent to tenants.

Integration Components:

Dynamics 365:

- Email Configuration: Configure the email settings within Dynamics 365, including SMTP server details and mailbox configurations.
- Integration Triggers: Configure event triggers within Dynamics 365 to initiate the integration process based on specific events, such as email send or receive events.

Exchange:

- Email Server Configuration: Configure Exchange server settings, including authentication details and connection parameters.
- Email Routing: Establish rules or configurations within Exchange to route emails between Dynamics 365 and Exchange.

- Exchange Web Services (EWS): Leverage EWS APIs or connectors to facilitate communication between Dynamics 365 and Exchange.

Integration Workflow:

Email Tracking:

- When an email is sent from Dynamics 365, capture relevant metadata and content.
- Associate the email with the corresponding record in Dynamics 365 using unique identifiers or metadata.
- Log the email activity, including recipients, subject, and content, within the Dynamics 365 record.

Calendar and Appointment Synchronization:

- Sync appointments and meetings between Dynamics 365 and Exchange calendars.
- When an appointment is created or updated in Dynamics 365, trigger an event or action to sync the corresponding event in Exchange.
- Ensure that changes made in either Dynamics 365 or Exchange calendars are reflected in both systems.

Contact Synchronization:

- Establish a synchronization mechanism to ensure contact information consistency between Dynamics 365 and Exchange.
- When a contact is created, updated, or deleted in either Dynamics 365 or Exchange, trigger an event or action to sync the changes to the other system.
- Map and align the contact fields between the two platforms to ensure accurate synchronization.

Workflow Automation:

- Configure integration triggers within Dynamics 365 to initiate actions or workflows based on events or changes in Exchange.
- For example, trigger a workflow in Dynamics 365 when a specific email is received, or update a record based on changes in Exchange contacts.

Security and Authentication:

- Implement appropriate security measures to ensure secure communication and data transfer between Dynamics 365 and Exchange.
- Configure authentication mechanisms, such as OAuth or Azure Active Directory, to authenticate and authorize access to the platforms and resources.

Set up of Exchange and D365

Emails that are queued up in D365 will be sent to the integrated exchange server for delivery.

Assumptions: The email model below uses Dynamics 365's Server-side Synchronization method as opposed to using email router set up.

Below are the requisites which the client engineer team will need to be fulfilled.

1. Client to create a AAD username with a mailbox and email address of the user. This will be the sending email address of all email communications sent from Zavanti housing to external and internal recipients. Client to provide appropriate Exchange licenses to the user created.
2. Content of these emails are configured in D365 and will be covered in our workshop and design sessions.
3. Engineer to assign an Office 365 license to the above user.

4. Engineer to send login details to Zavanti (username and password)
5. Zavanti will be creating a queue in D365 which will be used to configure incoming and outgoing profiles for the email notifications to be sent or received.

For integration between Dynamics 365 and Exchange on premise deployment please refer to the below link

https://learn.microsoft.com/en-us/power-platform/admin/connect-exchange-server-on-premises

Zavanti Inspection App

The Zavanti out of the box solution includes an inspection app developed in Power Apps. This is a canvas app that will be deployed within the client power app environment. The app lets tenancy and property managers to conduct property and tenant related inspections using a mobile device. To deploy the app into client's environment we will need power platform admin access.

Features of the app include –

- iOS and Android device compatibility.
- Ability to view tenant and property details.
- Ability to create a maintenance flag if needed.
- Ability to take pictures and upload them to SharePoint.

Canvas App Architecture

App Components:

- Screens: Canvas apps are composed of multiple screens that represent different user interfaces or views within the app. Each screen contains controls and elements that enable user interaction, for example inspection screens, location and asset screens.
- Controls: Controls are the building blocks of a Canvas app and include buttons, labels, text inputs, galleries, data tables, and more. They allow users to interact with and manipulate data within the app.
- Data Sources: Canvas apps connect to various data sources, such as SharePoint, Dataverse. These data sources provide the app with the necessary data to display and manipulate. Zavanti has real time connections to Dataverse and SharePoint to store pictures.
- Formulas: Canvas apps use formulas, written in the Power Apps formula language, to define logic and behaviour. Formulas are used for calculations, data manipulation, navigation, and control interactions.

Data Integration:

Canvas apps integrate with data sources using connectors or custom APIs. Connectors provide out-of-the-box connectivity to data sources like Office 365, Dynamics 365, SharePoint, and more. Zavanti uses connectors to Dataverse and SharePoint. Data integration allows Canvas apps to retrieve, create, update, and delete records from connected data sources.

User Interface and Experience:

Canvas apps provide a rich user interface (UI) design capability, allowing designers to customize the app's appearance, layout, and branding.

Designers can use various UI elements like buttons, labels, images, galleries, and forms to create visually appealing and intuitive user experiences.

User interactions, such as button clicks or form submissions, can trigger actions and navigate between screens.

Logic and Business Rules:

Canvas apps can include business logic and rules to enforce data validations, perform calculations, apply security, and automate processes.

Formulas written in the Power Apps formula language enable developers to define logic, conditions, and functions.

Power Automate (formerly Flow) can be used to create workflows and automate tasks within the app. Zavanti has out of the box flows that automate some of the inspection process including saving pictures and images in SharePoint.

Security and Permissions:

Canvas apps can leverage the security features provided by the underlying data sources, such as SharePoint or Dynamics 365 to control access to data.

Permissions can be configured to ensure that only authorised users can view or manipulate data within the app.

Appropriate authentication methods, like Azure AD or OAuth, can be employed to secure access to the app.

Deployment and Scalability:

Canvas apps can be deployed to various environments, such as development, testing, and production, using the Power Apps deployment process.

Deployment options include sharing the app with specific users or distributing it to broader audiences via app stores or web links.

Canvas apps can scale to accommodate increasing user loads and can be optimized for performance by implementing techniques like data caching and asynchronous loading.

For further information on Deploying Power Apps please refer to the below link

https://learn.microsoft.com/en-us/power-apps/guidance/fusion-dev-ebook/08-protecting-deploying-app

https://learn.microsoft.com/en-us/power-platform/admin/admin-powerapps-enterprise-deployment

Zavanti Housing Solution Deployment

Customisations to the housing solution will be deployed by Zavanti Admin when all the above requisites on Dynamics 365, SharePoint and Azure Functions have been completed.

Deployment Strategy

Sandbox Environment:

Zavanti will need a separate sandbox environment for development and testing purposes. This allows us to develop and test customizations without impacting the production environment.

Development Lifecycle:

Zavanti adheres to a clear development lifecycle process that includes development, testing/sandbox, and production environments. This ensures proper testing and validation of customizations before they are deployed to the production environment.

Version Control:

Zavanti uses an internal version control system called GitHub to manage customizations and source code. This allows us to track changes, collaborate with our clients, and easily revert to previous versions if needed.

Customisation Documentation:

Zavanti will document customizations thoroughly, including the purpose, functionality, and any dependencies. This documentation serves as a reference for future deployments, troubleshooting, and knowledge transfer. This may be in the form of release notes etc.

Solution Packaging:

We package customizations into logical units using solutions in Dynamics 365. This allows for easier deployment and management of customizations across multiple environments and client sites. Solutions enable us to group related customizations together and deploy them as a single unit. This simplifies deployment and supports easy management and version control.

Incremental Deployments:

When possible Zavanti will deploy customizations incrementally rather than all at once. This minimizes the risk of disruptions and makes it easier to identify and address any issues that arise. These

incremental deployments can be clearly identified in the solution management framework of Dynamics 365.

Test and Validate:

Zavanti will conduct comprehensive testing of our customisations in a separate environment (sandbox) before deploying any customisations to the production environment. Zavanti will perform functional, integration, and user acceptance testing to ensure all the customisations work as intended before deploying into Production. This will require the client to sign off on functionality testing before any deployment to production.

Backup and Restore:

Before deploying any customizations to the production environment, we take a backup of the system. This ensures that you can restore the environment to a known working state in case of any issues during deployment. For further information regarding Dynamics 365 back-up and restore please refer to

https://learn.microsoft.com/en-us/power-platform/admin/backup-restore-environments

User Communication and Training:

Zavanti will communicate with client, end users and stakeholders about upcoming customizations and their impact. We will provide training to help users understand the changes and adapt to the new functionality.

Rollback Plan:

Zavanti will work with our clients individually to develop a well-defined rollback plan should any issues arise during deployment. This includes identifying the steps needed to revert to the previous version and communicating the plan to all relevant stakeholders.

Post-Deployment Support:

Provide ongoing support and monitoring after the deployment to address any issues or questions that may arise. This helps ensure a smooth transition and user adoption of the customizations and change.

Support and Upgrades

The following section outlines **Microsoft's** approach to delivering new features, updates, and enhancements to the Dynamics 365 suite of business applications. The strategy aims to provide customers with regular and predictable updates to keep their systems current and leverage the latest capabilities. Here are the key aspects of the Dynamics 365 release strategy.

Release Cycle:

Microsoft follows a consistent release cycle for Dynamics 365, typically consisting of two major updates per year. These updates are named as the "Spring Release" April and the "Fall Release" October. The exact timing of these releases may vary, but Microsoft provides advance notice to customers about the release schedule.

Release Waves:

The major updates are organised into release waves. Each release wave encompasses a set of new features and enhancements that will be made available to customers. The release wave timeline is typically shared by Microsoft well in advance, allowing customers and ISV partners like Zavanti to plan and prepare for upcoming changes to their solutions.

Early Access and Testing:

Microsoft provides Early Access programs, such as the Early Access Preview and the Early Access Early Adopter Program (EAAP), allowing customers and partners to test and evaluate the new features before they are generally available. This early access helps customers and partners understand the impact of upcoming changes on their business processes and gives them an opportunity to provide feedback to Microsoft and also amend their solutions. Zavanti are partners of this early access program.

Release Preview:

Before the general availability of a major update, Microsoft publishes a Release Preview guide. This guide provides detailed information about the new features, enhancements, and changes included in the upcoming release. It helps customers and partners prepare for the update by understanding the scope and impact of the changes.

Release Deployment Options:

Microsoft offers customers flexibility in adopting new releases. Customers can choose to opt-in to the release immediately or opt for a specific release deployment schedule that suits their business needs. The options include "Early Access," where customers get early access to the release, and "General Availability," where customers receive the release during the standard release schedule.

Release Documentation and Communication:

Microsoft provides comprehensive documentation, including release notes, guides, and videos, to help customers and partners like Zavanti understand the new features and changes introduced in each release. Additionally, Microsoft communicates updates and release-related information through various channels, such as the Dynamics 365 blog, community forums, webinars, and events.

Lifecycle Support:

Microsoft provides lifecycle support for Dynamics 365 releases, ensuring that customers and partners have access to critical updates, bug fixes, and security patches. Microsoft communicates the support timelines, including the availability of new updates and the end-of-support dates for older versions, allowing customers to plan their system upgrades accordingly.

Customer Engagement and Feedback:

Microsoft actively encourages customer engagement and feedback throughout the release cycle. Customers and partners can provide feedback through various channels, such as the Dynamics 365 Community, user groups, and direct interactions with Microsoft teams. This feedback helps Microsoft understand customer needs, address issues, and improve the product.

In addition to the D365 platform upgrades, Zavanti release its own upgrades. Where possible Zavanti upgrades will coincide with the D365 platform upgrades i.e. twice a year. One in April and the other in October. These upgrades may include new functionality and improvements to existing functionality. All Zavanti upgrades are optional, and we will inform our clients in the form of release notes. Upgrades follow the deployment methodology described in this document.

To comply with regulatory requirements Zavanti may on an ad hoc basis provide updates. This is mostly to do with regulatory reporting requirements or compliance. Zavanti will communicate this to clients prior to the upgrade.

Support Access

Technical support may be obtained by Customer's nominated contact person. Customer must designate one employee to serve as their primary contact person, they may nominate an alternate contact person. Customer may change their contact person so long as they provide notice to Zavanti of such change.

Support Requests

Customer must report issues by logging on to the Zavanti website: http://zavanti.com

In the event of technical difficulties in accessing the online support site, calls may be logged with Zavanti through the following:

- crmsupport@zavanti.com
- Phone 1 300 302 358
- Hours 9 a.m. – 5:00 p.m. AEST, Monday through Friday on non-holidays

After the Service Desk hours, a message bank service will take voicemail messages. These will be retrieved and entered into our Service Desk Incident Management system on the next business day.

Severity	Initial Response Time	Effect	Target Resolution Time
1 (Critical)	2 business hours	Defect prevents all useful work from being	If a Severity 1 Defect occurs during normal operating hours

		done, material defects in essential functions for which no non-manual workaround exists; or Defects that cause a material loss of data, high-risk security breach or Hosted Service disruption.	(8:30 am – 5:30 pm AEST weekdays), Zavanti will begin immediate and continuous efforts to reproduce and resolve the Defect, and will carry out those efforts until the Defect is resolved. Zavanti will use reasonable efforts to resolve all Severity 1 Defects in the shortest time possible, and will review status with Customer on a daily basis or more frequently, if requested.
2 (Major)	Next Business Day	Defects that disable essential functions but for which a non-manual workaround exists, defects that block systems test or deliverables, or defects that violate the material specifications in the Documentation, or Service component malfunction or low-risk security breach.	If the Defect is a Severity 2 issue, Zavanti will begin efforts to reproduce the problem no later than the opening of the next business day after receipt of the issue by Customer. Zavanti will use reasonable efforts to resolve Severity 2 problems as rapidly as practical, but no later than the next Update after reproduction of the Defect.
3 (Minor)	2 Business Days	All other issues.	Severity 3 Defects will be addressed in Service Provider's normal Update. For Hosted Service the consultant reports by the end of the next working day.
4 (Enhancement request)	N/A	Functionality requested which is not available in standard operations of the software. Item will be considered for future releases.	Resolution is discussed and agreed to by both parties as to a reasonable timeframe for completion.

Database Growth

When it comes to storage for Dynamics 365, Microsoft splits what's currently known as "Database Storage" into three separate buckets:

1. Database capacity,
2. File capacity, and
3. Log capacity –

and categorising it under the Dataverse, which is the underlying service for Dynamics 365.

- Database capacity consists of metadata.
- File capacity refers to attachments.
- Log capacity refers to log files that are created in the system.

Users will get 10 GB of database capacity, 20 GB of file capacity, and 2 GB of log capacity by default and receive incremental capacity for each full access user added: 250 MB of database capacity, 2 GB of file, and no additional log capacity per full access user. Additional storage can be purchased at $40/GB/month for database, $2/GB/month for file and $10/GB/month of log capacity, on an as-needed basis.

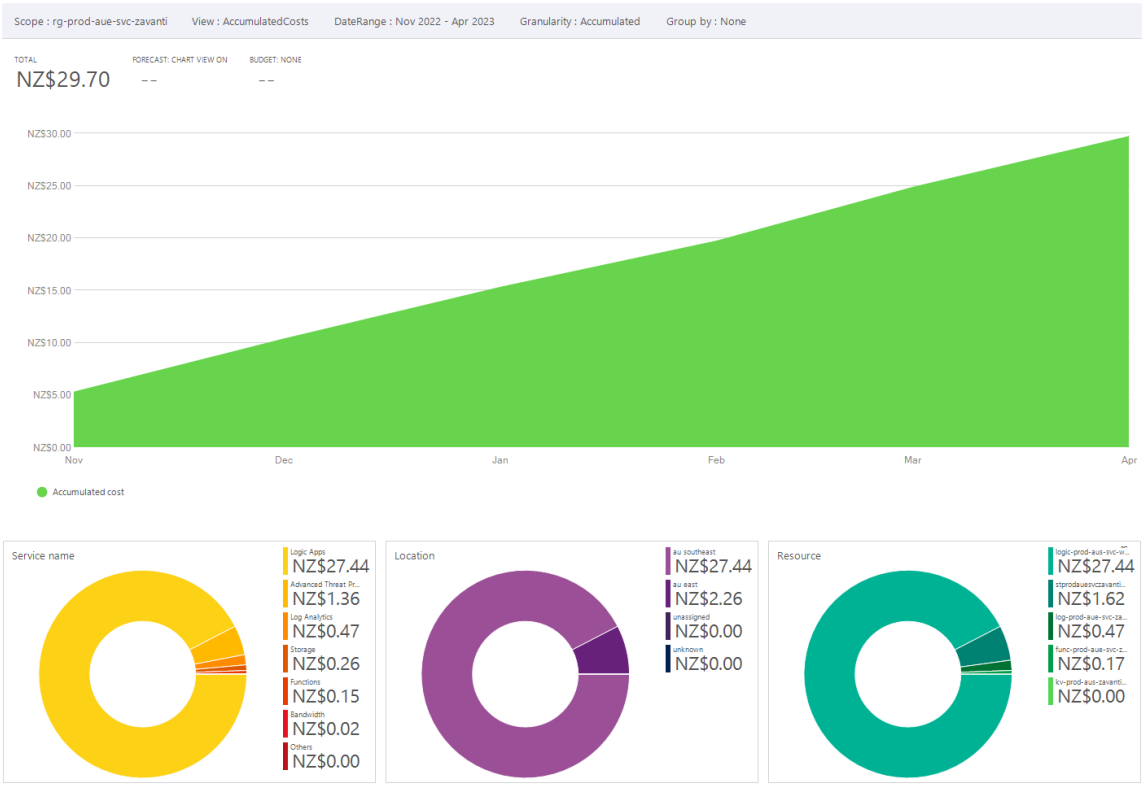
	Database	File	Log
Default	10 GB	20 GB	2 GB
Entitlement	250 MB/full-access user	2 GB/full-access user	None
Additional storage cost	$40/GB/month	$2/GB/month	$10/GB/month

Scenario: 10 Enterprise USLs

It's hard to accurately estimate the storage growth requirements for Zavanti. It depends on the amount of data that needs to be migrated before go live and the anticipated addition on new properties in the system. As an estimate based on previous implementations client that has approx. 2500 properties who has used Zavanti for over a year have the below statistics.

- Database – 15-20 Gb
- File – 20 -30 Gb
- Log File – 4 – 5 Gb

Below is an estimate cost of Azure Functions consumption based on a 6-month period.


Scope : rg-prod-aue-svc-zavanti
View : AccumulatedCosts
DateRange : Nov 2022 - Apr 2023
Granularity : Accumulated
Group by : None
TOTAL
NZ$29.70
FORECAST: CHART VIEW ON
--
BUDGET: NONE
--
NZ$30.00
NZ$25.00
NZ$20.00
NZ$15.00
NZ$10.00
NZ$5.00
NZ$0.00
Nov
Dec
Jan
Feb
Mar
Apr
Accumulated cost
Service name
Logic Apps
NZ$27.44
Advanced Threat Pr...
NZ$1.36
Log Analytics
NZ$0.47
Storage
NZ$0.26
Functions
NZ$0.15
Bandwidth
NZ$0.02
Others
NZ$0.00
Location
au southeast
NZ$27.44
au east
NZ$2.26
unassigned
NZ$0.00
unknown
NZ$0.00
Resource
NZ$27.44
NZ$1.62
NZ$0.47
NZ$0.17
NZ$0.00

Index