Matter Center for Office 365

Technical specifications and system architecture

Table of contents

[Introduction 3](#_Toc438062988)

[System architecture diagram 3](#_Toc438062989)

[Matter Center components 4](#_Toc438062990)

[Component descriptions 4](#_Toc438062991)

[Deployment view 5](#_Toc438062992)

[Communication flow diagram 5](#_Toc438062997)

[Detailed communication flow for Matter Center 6](#_Toc438062999)

[System attributes 6](#_Toc438063001)

[Merits and limitations 6](#_Toc438063002)

[Dependencies 9](#_Toc438063003)

[Availability 10](#_Toc438063004)

[Reliability 10](#_Toc438063005)

[Safety 10](#_Toc438063006)

[Maintainability 11](#_Toc438063007)

[Storage requirement 11](#_Toc438063008)

[In-app security 11](#_Toc438063009)

[Confidentiality 12](#_Toc438063015)

[Integrity 13](#_Toc438063016)

[User groups 13](#_Toc438063017)

[List structure 13](#_Toc438063018)

[MatterCenterRole 13](#_Toc438063019)

[MatterCenterMatters 14](#_Toc438063020)

[MatterCenterHelpSection 15](#_Toc438063021)

[MatterCenterHelpLinks 15](#_Toc438063022)

[UserPinnedMatter 16](#_Toc438063023)

[UserPinnedDetails 17](#_Toc438063024)

[Matter Configurations 19](#_Toc438063025)

[Glossary 19](#_Toc438063026)

# Introduction

Microsoft’s Corporate, External, and Legal Affairs (CELA) Technology and Operations team has developed Matter Center for Office 365, a productivity and collaboration solution built using the Office app development model, Azure and Sharepoint.

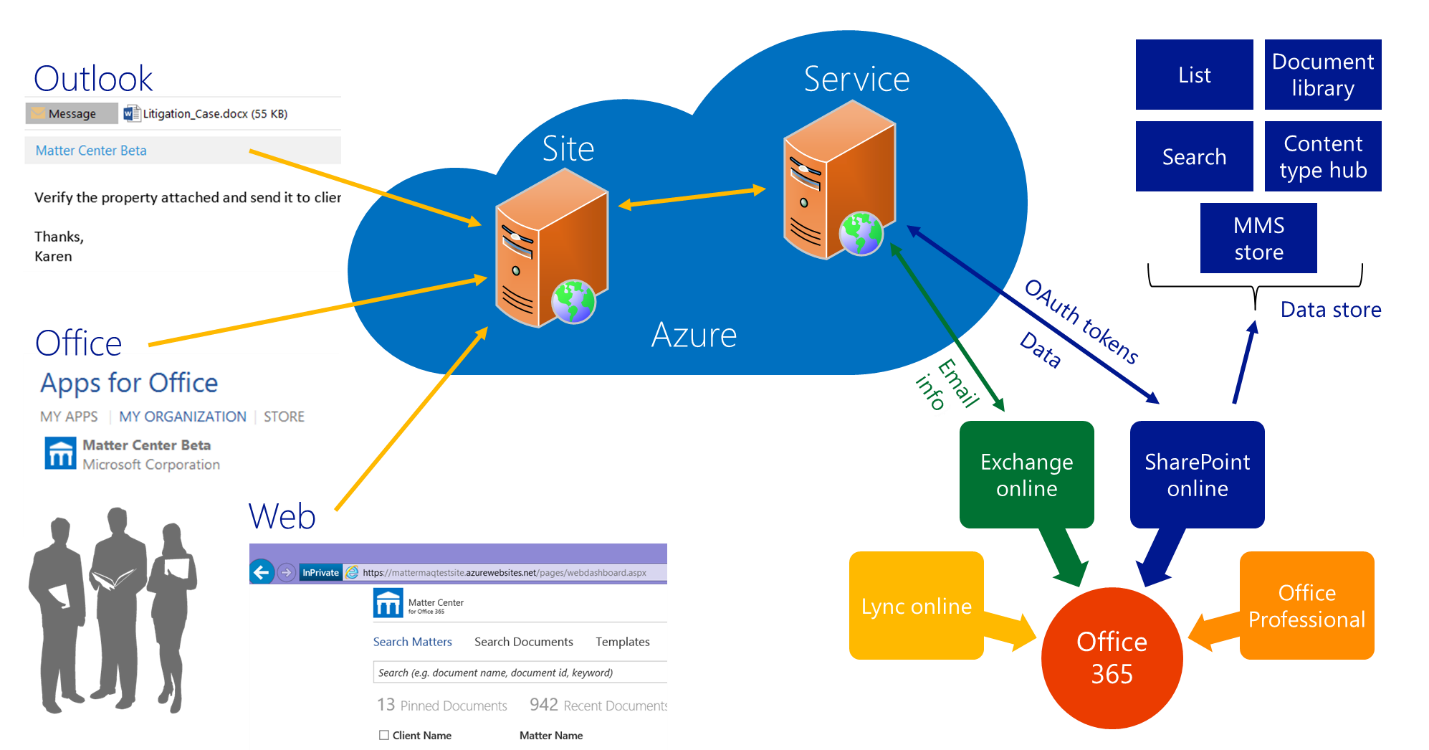
Since Matter Center is built on Office 365, it inherits a multitude of benefits from the underlying productivity platform, spanning enterprise-grade features within SharePoint, Exchange, OneDrive for Business and more. Examples include self-service business intelligence, compliance and information protection, eDiscovery, document and knowledge management, advanced federated search and discovery, and tools for enterprise management and control.

This document provides an overview of Matter Center’s underlying technical specifications, system architecture, components, communication processes, and system quality attributes. For definitions of any and all acronyms used throughout this document, refer to the [Glossary](#_Glossary).

# System architecture diagram

This section includes a diagram of Matter Center’s system architecture.

**System architecture diagram**



# Matter Center components

The components of Matter Center are as follows:

| **Component name** | **Purpose** | **Execution environment** |
| --- | --- | --- |
| Outlook/Office app | * Opens app in Outlook & Office (Word, Excel, PowerPoint) | Client machine |
| Website | * Provides UI elements for all apps | Windows Azure |
| WCF Service | * Receives and validates inputs from Website * Performs actual SharePoint operations * Sends output to Website in JSON format | Windows Azure |
| SharePoint Online (O365) | * This is the data store for entire application * It holds documents, permissions, configurations, taxonomy, search relate components etc. | SharePoint Online |
| SharePoint app | * Authenticates & Authorizes user * Provides the entry point for all apps (including Outlook and Word) | SharePoint Online |

## Component descriptions

The components of Matter Center are described below:

| **Component name** | **Description** |
| --- | --- |
| Outlook/Office app | Application wrapper within Outlook and Word client programs that allows the user to access the SharePoint system from Outlook and Word |
| Website | The frontend component which is developed using HTML, CSS & JS. This component interacts with the service to performs an action based on user interaction with other elements. This website is also responsible to perform validations whether user have provided correct information to take the action. This layer calls service methods using AJAX JS calls.  This single website serves SharePoint, Outlook and Office apps. This website is following responsive design using media queries. |
| WCF Service | This is a traditional Windows Communication Foundation (WCF) service which forms the business layer of the application. This service authenticates the calls from the frontend, interacts with SharePoint, and passes the requested data back to the frontend for display. In order to interact with SharePoint, this service makes use of [SharePoint Client Server Object Model (CSOM)](https://msdn.microsoft.com/en-us/library/ff798388.aspx) |
| SharePoint Online (O365) | This is the system where all information is stored. It holds all documents library, documents, metadata associated with each document, configuration list, permission, indexes and crawler information etc. |
| Windows Azure | Hosts Website and WCF Service as two separate websites with different domains |
| SharePoint Search API | This application depends on SharePoint Search crawl to return its data to avoid any space and performance constraints for large documents. This also enables Matter Center to efficiently make use of metadata tags and search on document content. |
| SharePoint App | This is a [provider hosted app](https://msdn.microsoft.com/en-us/library/office/fp142381.aspx) used to authenticate and authorize user with SPO environment, we make use of this app. Each SharePoint app has an appredirect URL which check if current user is allowed to access the app or not. Along with this it also provides a set of SPO tokens. These tokens are used to impersonate current user and taken action on SPO on their behalf. |

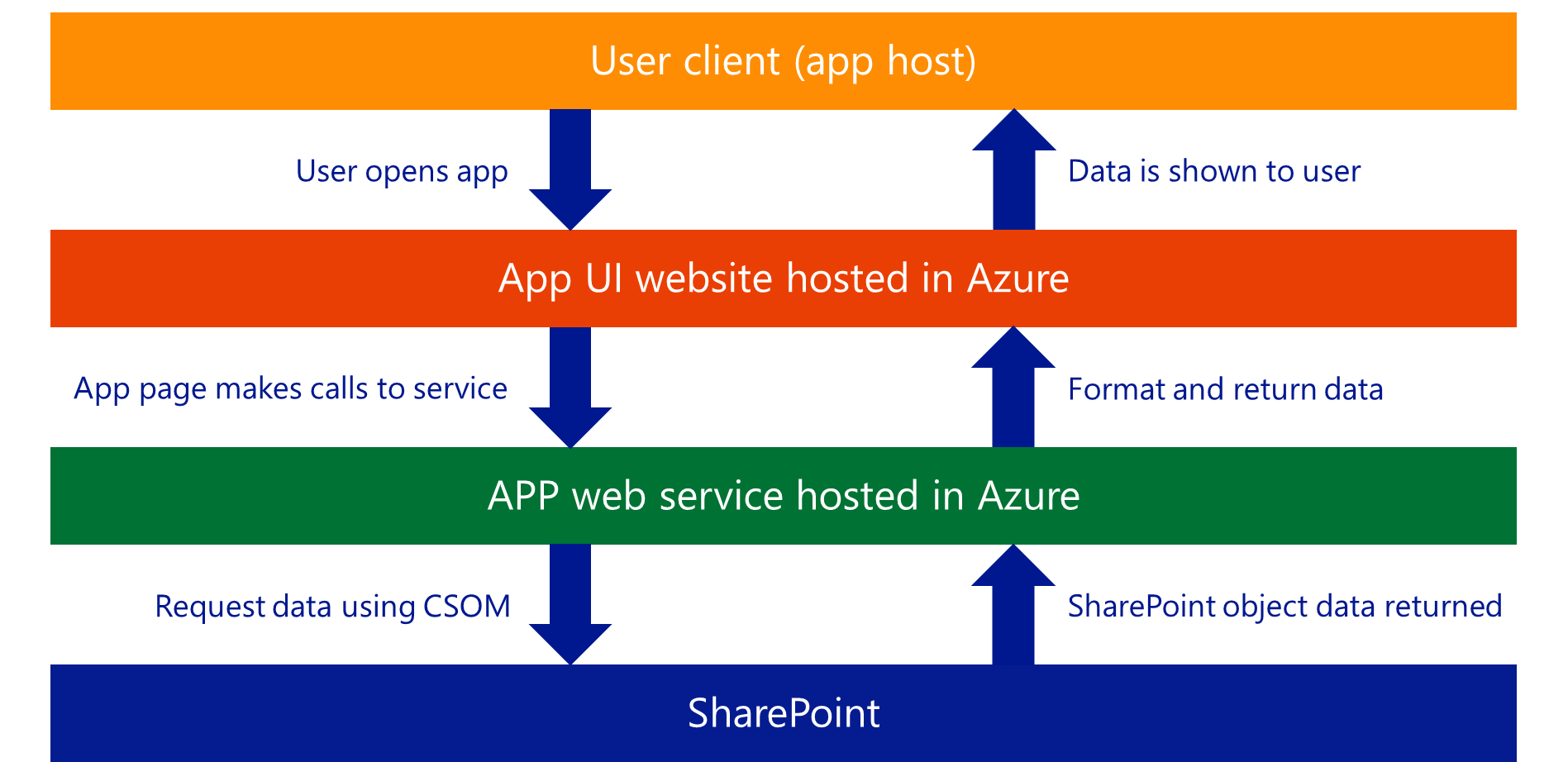
# Deployment view



## Communication flow diagram

The detailed communication flow diagram for Matter Center is as follows:

**Communication flow diagram**





## Detailed communication flow for Matter Center

The communication flow for Matter Center follows these steps:

1. A user navigates to the app page from Office/Outlook or the Web Client. The app checks if the user is already logged in; if yes, the app loads and allows the user to perform operations (see step 4).
2. If the user is not logged in, the app redirects the user to SharePoint’s login page via the app redirect page for authentication.
3. SharePoint asks the user to login and, in turn, returns the SPAppToken to the app. Using the SPAppToken, the app generates RefreshToken and saves it in cookies to specify that user has successfully logged in.
4. Every user action performs calls to WCF service using this RefreshToken for SharePoint operations:
5. In case if the user is trying to upload mail/attachments, the Azure WCF service makes a calls to Exchange web service using the current attachments or email token to get current email or attachments in SOAP format.
6. After getting current email/attachments from the Exchange service, the WCF service uploads that mail/attachments to SharePoint using RefreshToken and SPO CSOM.
7. For each SharePoint operation, RefreshToken passed to service by website is authenticated by SharePoint to get current user’s context.
8. SharePoint then provides results to service for the operation performed.
9. Service passes the results to the UI layer in JSON string format. In case of an error in the service layer, the error message is logged into Azure Cloud Table Storage and the user sees a user friendly error message on the UI layer.
10. The app saves user entered data or selections on Matter creation process only, in case, if the app closes before completing all the steps. This data is stored in the user’s local storage.

**Note**: To ensure secure communication of data, Matter Center uses HTTPS (Hypertext Transfer Protocol Secure) protocol across all communication channels.



# System attributes

## Merits and limitations

Matter Center creates a site collection for each client for a particular law firm. Each client can contain multiple legal matters. Each matter creates a separate document library within the specific client site collection. Along with this document library, Matter Center also creates following for a matter:

* OneNote library with an Empty OneNote file;
* A Calendar list;
* A Task list;
* A Matter Details page which brings all of the above elements together on a single web page; and
* Each Matters is created with unique GUID which is auto-generate by the system. This is different than the Matter ID which is currently manually entered by users by creating the matter.

The table below provides the merits and limitations of using a client as a site collection and a matter as a document library. Further information about these software limits can be viewed [here](https://support.office.com/en-in/article/sharepoint-online-software-boundaries-and-limits-8f34ff47-b749-408b-abc0-b605e1f6d498).

|  | **Client as site collection** | **Matter as document library (with contiguous memory allocation)** |
| --- | --- | --- |
| Major limitation | Each tenant can only support 2000 clients  Therefore, the maximum number of clients that can be supported is 100,000, which will require a total of 50 tenancies | The upper limit to the storage space is 1 TB for each site collection  Once 80% of the space is in use, a new site collection must be manually created as SharePoint online doesn’t provide an API to create site collections |
| Content separation | High | High |
| Ease of data access | It is easier to access data for a client as long as data resides in only one site collection  Complexity will quickly increase if the content spreads across multiple site collections and could become unmanageable | There is no limit on the number of document libraries in a site, meaning that there is no limit on the number of matters a site collection can have, as long as it’s below 1 TB overall  Custom administrative policies should be followed; no new content should be created when usage reaches 80% or 90% of the maximum allowable limit, so as to allow content to expand |
| Administration requirements | High | Medium |
| Archival | Unknown | Matter libraries can be archived with records management |
| Can contain | Anything in SharePoint such as reusable workflows, documents, custom lists, images etc. | Limited availability  Can contain only documents, images etc. |
| Permission assignment | Easier to assign at site collection level | Need to break permissions at document library level when created |
| Calculations | Size limitation:  **Note:** This calculation does not take into account any space requirement for versioning and archival  Approx. matters/client = 500,000 🡪 A  Number of sites each site collection can have = 2000 sub-sites/site collection 🡪 B  Since each site collection can only hold 2000 matters, 250 site collections/clients are needed (500,000/2000) to accommodate all matters (From A & B) 🡪 C  Each tenant in SharePoint Online can only hold 500,000 site collections  Since every client will need upward of 250 site collections to hold its matters (from C above), and because we can have only 2000 clients (500,000/250) in one tenant (each tenant = 500,000 matters) collection to hold its matters (from C above), we can have only 2000 clients (500,000/250) in one tenant (each tenant with 500,000 matters)  So, to hold 100,000 clients, 50 tenancies are required (100,000/2000)  Please note the total number of tenants is huge (2500)  This may or may not always be the case (so large) because a firm may not always have 100,000 active clients  The figure of 100,000 reflects both active and inactive clients | Size limitations:  Memory available/site collection on SharePoint Online = 1 TB  Memory available/tenant on SharePoint Online = 25 TB  Once 25 TB is used up, we will need to move onto a new tenancy |

## Dependencies

Matter Center requires the following to function appropriately:

|  |  |
| --- | --- |
| Object | **Dependencies** |
| Term store | Term store is used for saving details about clients and details of hierarchy for Practice Groups, Area of Law and Sub Area of Law  These are displayed as filters in search apps |
| Search API | SharePoint’s Search API is leveraged to get indexed search results (libraries and documents) |
| Managed properties | Managed properties are used along with Search API to filter, sort and retrieve required results |
| Site columns and content types | * Site columns and content types are used when creating matters * Site columns are added to content types so that custom properties can be associated to each document * Content types are associated to matter libraries so as to categorize the uploaded documents |
| SharePoint list | We are currently using multiple list which are used by the application which contains configuration information |
| Custom Permission groups on SharePoint | In order to get users permission, we are making use of a special SharePoint group. This group allows the SharePoint JSOM object to make remote calls from SharePoint web pages |
| Full Control for SharePoint app on Entire Tenant | The SharePoint App required full control on the entire tenant. This permission is required because we are creating SharePoint pages with the help of SharePoint app |
| Read and Write permission for Exchange App in ItemRead and ItemEdit mode | Exchange app should have ItemRead permission so that Matter Center appear in the email. At the same time, it also requires ItemEdit permission so that Matter Center app can be added while users are composing emails as well |
| Application Insights (if required) | This is a usage tracking tool available with Azure. You can use this tool to capture usage pattern for your application |
| Redis Cache | To reduce the amount of time to pull static content from SPO, we are making use of Azure Redis cache to store this information. Redis cache currently only holds data from term store & Help list |
| Azure Table Storage | This will hold all error log information in case if there are unexpected issues with the application |

The below table identifies core Matter Center Lists:

| **Object** | **Details** |
| --- | --- |
| Matter Center role list | SharePoint list where details about roles are stored  These roles are used while creating matters |
| Matter Center matter list | SharePoint list where details about matters are stored  These matter details are provided by user during creating matters  The list is populated for storing details  Otherwise, the list is not used anywhere in the app |
| Pinned matter details | SharePoint list where details about pinned matters for each user are stored  Find pinned matters, pin matter and unpin matter operations use this list |
| Pinned document details | SharePoint list where details about pinned documents for each user are stored  Find pinned documents, pin document and unpin document operations use this list |
| Matter Configurations | SharePoint list where default settings of Matter Creation page is stored |
| Matter Center help section list | SharePoint list where details about contextual help sections are stored. |
| Matter Center help links | SharePoint list where details about contextual help links are stored |

### Availability

**Windows Azure:** The availability of apps depends on the availability of websites for UI and service. These websites will be unavailable for a brief period when publishing is performed. You can even configure multiple instances for your website and service on azure to reduce downtime to your end users

**SharePoint Online** maintenance may also cause the apps to be unavailable.

### Reliability

**Windows Azure:** The reliability of apps depends on the integrity of data as handled by SharePoint. WCF service does not store any data within it and hence, if down, can be restored by re-publishing.

### Safety

**Windows Azure:** Matter Center depends on Azure for secure the communication channels and its content. Azure website and service does not store any information Apps support default safety as provided by SharePoint and/or Windows Azure.

### Maintainability

**Windows Azure:** The apps are maintainable as they follow three-layer architecture with separation of presentation, service and utility layer. Also, the SharePoint component will require minimum updates if redeployed after changes.

### Storage requirement

**SharePoint Online:** The storage space for each site collection will be dependent on the space allocated by the administrator. The max limit for the site collection depends the Office 365 SKU and the storage of quota allocated for the site collection. This is configurable and can be extended depending on the storage available in the tenant.

**Windows Azure:** In our application we are using Azure Storage to capture all error logs only. This logging will help in debugging errors with apps in Outlook, Word and SharePoint.

## In-app security

**Windows Azure:** Security in the app is handled primarily by RefreshToken which is provided by SharePoint app. This RefreshToken is retrieved from SharePoint upon successful login as part of response headers. This Encrypted RefreshToken is then saved in secure cookies for future app launches. This token is valid for ten hours. In order to perform an action on SharePoint, we use RefreshToken to generate AccessToken and then, using AccessToken, we create SharePoint’s context.

The RefreshToken is extracted from Response Headers using the TokenHelper class. This class is added by default when you create Provider Hosted SharePoint add-in in Visual Studio. Using the TokenHelper class, we generate AccessToken from RefreshToken to impersonate logged in user and perform a task on SharePoint.

Data within the app is displayed based on SharePoint’s context created from AccessToken. This SharePoint context is user based and hence, if the user has permission on that SharePoint object, he/she will able to view that object (note that this holds true even when SharePoint’s search API is used). Hence, the authorization within the apps is handled by SharePoint’s default behavior.

**Encryption and decryption of RefreshToken for SharePoint online**

After RefreshToken is retrieved from SPAppToken, RefreshToken is encrypted using an AES algorithm. The key used for the encryption is stored in the app settings of the Azure website for the UI. RefreshToken is exchange between UI and Service with each call to ensure the authenticity of the request.

RefreshToken is decrypted in service before retrieving the access token to generate SharePoint context. The key used for decryption is again stored in the app settings of the Azure for the service. Please note, this key has to be same as the one which is present in UI. This key should be changed after a duration. To keep track of the old key used for decryption, it is stored in the app settings of the Azure website for the UI and service.

When the user reloads the page or switches from one app to another, RefreshToken is checked whether it is encrypted using the new encryption key. It will be encrypted with the new key if not already done. The new encrypted RefreshToken is then stored in the cookie for future purposes.

In the service, RefreshToken is first decrypted with the new encryption key. If the decryption fails, the service will try to decrypted it with an old encryption key. In case if service is still not able to decrypt the Token it will simply throw and error. RefreshToken will remain encrypted with the old encryption key until the user reloads the page or switches from one app to another.



### Confidentiality

Matters can be open (accessible to everyone) or secure/confidential (accessible to only selected individuals (due to conflict of interest or sensitivity of information)). Office 365 currently does not support explicit deny (i.e. the mechanism to specifically block an individual’s access to a document or matter, irrespective of their role in Office 365). As a work around to maintain the confidentiality of matters and its document, Matter Center explicitly grants permissions only to users assigned as part of the matter team during the matter creation process.

To elaborate on more on this point, consider an example where a law firm is working on a matter with the client **Northwind Traders**. John Doe, Jane Doe and Sara Davis are the only members allowed to work with Northwind Trader’s client.

If an open matter is created for Northwind Traders, John, Jane and Sara all will be able to view everything associated with this matter because permissions will have been inherited from the client site collection.

If the matter is determined confidential, and only John and Jane are assigned to the team, then Matter Centers breaks the permission to all matter components (document library, page, calendar, OneNote library) and grants permissions to only John and Jane. However, tenant and site collection administrators will still be able to view all the information related to the Northwind Traders matter.

In addition to above, each matter created via Matter Center, is created using a system generate GUID. For e.g. the document library created for the matter is created using this GUID. This GUID is generated using ASP.NET GUID class. This GUID is different from Matter ID which should be manually entered by users. This implementation will later allow to easily integrate with automated Matter ID generation systems as well. This same GUID is used for other Matter Components as well like, calendar, OneNote library, task list and Matter Details Pages as well.

### Integrity

All legal documents present in each matter will be available and editable for only those users who have been provided explicit access when the matter was created.

### User groups

In order to access the app two SharePoint groups are required to provide appropriate permissions. The details for these groups are as follows:

1. **Matter Center users** – These are the users who will be accessing the Matter Center apps, but will have no access to the create matter app. Users in this group will have read permissions on the app catalog and will have contribute permissions on the configuration lists used by the Matter Center apps.
2. **Provision matter users** – These are the users who will have access to the Matter Center create matter app. Users in this group will have read permissions on the app catalog.

# List structure

As described in above sections, Matter Center makes use of multiple configuration list which help with functionality of Matter Center. This section describes purpose of each list which is used by Matter Center. All lists mentioned in this section are available under you App Catalog site collection.

## MatterCenterRole

This list contains all roles currently belonging within the law firm. These roles are used on the matter creation process. The user must select specific role and identify users for those roles which assigning team for a matter.

Following are default roles in Matter Center’s.

|  |
| --- |
| **Role** |
| Responsible Attorney |
| Paralegal |
| Working Attorney |
| Legal Admin |
| Attorney |

A user role of Responsible Attorney is mandatory while creating a new matter. Each role can be assigned specific permissions and, based on these permissions, the user can access specific actions.

Depending on the business need, the roles are specified as mandatory or non-mandatory.

When a matter is being created, at least one user must be assigned to each of the mandatory roles. By default, we have set Responsible Attorney as the mandatory role.

Following is the list structure of this list:

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| Role name | Single line of text | Name of the role. This columns is used in the Matter Creation page |
| Mandatory | Yes/no | Flag to specify if the role is a mandatory role |

## MatterCenterMatters

This list holds all Matters which are created by the app. Only purpose of this list is to help during debugging in case if any Matters goes missing. A new item is always added in this list when a matter is successfully created

Following is the list structure for this list:

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| Title | Single line of text | This column holds client name and matter name |
| ClientName | Single line of text | This column holds the client name for which matter is created which is selected in the matter creation form |
| ClientID | Single line of text | This column holds the client ID for which the matter is created which is automatically selected in the matter creation form |
| MatterName | Single line of text | This column holds the Matter Name which is entered by user in the matter creation form |
| MatterID | Single line of text | This column holds the Matter ID which is manually entered by user in the matter creation form |
| ConflictConductedBy | Person or Group | This column holds who performed conflict check for this matter which is entered by user in the matter creation form |
| ConflictCheckOn | DateTime | This column holds when the conflict check was performed which is entered by user in the matter creation form |
| ConflictIdentified | Yes/No | This column identifies whether conflict was found for this matter which is manually selected by user |
| BlockUser | Person or Group | This column hold name of the individual who have conflict of interest and are blocked from accessing this matter |
| ManagingAttorney | Person or Group | This column holds name of users who are identified in Responsible Attorney role |
| Support | Person or Group | This column holds name of users who are identified in role other than Responsible Attorney |

## MatterCenterHelpSection

This list is used in the in solution help for creating sections. Using this list, you can configure different sections on different page. Using this you can also configure how the section should be shown in the help.

Following contains list structure of this list:

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| Section ID | Single line of text | This column holds unique Section ID which help identify which section is this |
| Section Title | Single line of text | This column holds the title of the section |
| PageName | Choice | This column hold pre-configured list of pages present in the app |
| Section Order | Number | This column allows you to determine the order in which sections should show on the page |
| NumberofColumn | Choice | This column determines who many columns the section should span. It contains values as either 1 or 2 |

## MatterCenterHelpLinks

This list is used for creating Help link within each section within the in-solution help. Following is the list structure:

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| LinkTitle | Single line of text | This column holds the link title which will be shown under a section |
| LinkOrder | Number | This column allow you to determine the order in which links will appear within each section |
| LinkURL | Hyperlink | This column contains the URL to which user will redirect on clicking this link |
| SectionID | Lookup | This column is a look up field to the Section ID field in MatterCenterHelpSection list. This will allow to create the mapping between which links should appear under which sections |

## UserPinnedMatter

This list is used to hold all pinned matters by users. This list is security trimmed so that if users gets access to this list, they can only view content associated with their alias only. Tenant and catalog admin will still be able to view content for all users in this list. For each user there will always be single line item in the list

Following is the list structure:

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| UserAlias | Single line of text | This column holds the user’s alias who has pinned matters. This alias is also used to only show data associated with logged in user in case if they directly landing on this list |
| UserPinDetails | Multiple line of text | This column holds list of all matters which user has pinned in a JSON format. This format allows us to quickly show pinned information. In this JSON structure we also store some metadata of the matter as well which is required to be displayed in Pinned section |

Following is the JSON structure on how the information is stored in UserPinDetails column:

{ "Doc library url": {

"MatterName": "MatterName",

"MatterDescription": "MatterDescription",

"MatterCreatedDate": "CreatedDateTile",

"MatterUrl": "document library url",

"MatterPracticeGroup": "semi colon separated list of practice groups",

"MatterAreaOfLaw": "semi color separated list of area of law",

"MatterSubAreaOfLaw": "semi color separated list of sub area of law",

"MatterClientUrl": "Client Site collection URL",

"MatterClient": "Client Name",

"MatterClientId": "Client ID",

"HideUpload": "Does user have upload file permission",

"MatterID": "Matter ID",

"MatterResponsibleAttorney": "semi color separated list of Responsible Attorney Name",

"MatterModifiedDate": "Last Modified Date of the matter" } ,

"Doc library url": {

"MatterName": "MatterName",

"MatterDescription": "MatterDescription",

"MatterCreatedDate": "CreatedDateTile",

"MatterUrl": "document library url",

"MatterPracticeGroup": "semi colon separated list of practice groups",

"MatterAreaOfLaw": "semi color separated list of area of law",

"MatterSubAreaOfLaw": "semi color separated list of sub area of law",

"MatterClientUrl": "Client Site collection URL",

"MatterClient": "Client Name",

"MatterClientId": "Client ID",

"HideUpload": "Does user have upload file permission",

"MatterID": "Matter ID",

"MatterResponsibleAttorney": "semi color separated list of Responsible Attorney Name",

"MatterModifiedDate": "Last Modified Date of the matter" }

}

## UserPinnedDetails

This list is used to hold all pinned documents by users. This list is security trimmed so that if users gets access to this list, they can only view content associated with their alias only. Tenant and catalog admin will still be able to view content for all users in this list. For each user there will always be single line item in the list

Following is the list structure of this list:

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| UserAlias | Single line of text | This column holds the user’s alias who has pinned documents. This alias is also used to only show data associated with logged in user in case if they directly landing on this list |
| UserPinDocumentDetails | Multiple line of text | This column holds list of all documents which user has pinned in a JSON format. This format allows us to quickly show pinned information. In this JSON structure we also store some metadata of the document as well which is required to be displayed in Pinned section |

Following is the JSON structure on how the information is stored in UserPinDocumentDetails site column:

{

"Document URL": {

"DocumentName": "Document Name",

"DocumentVersion": "Document Verison",

"DocumentClient": "client Name",

"DocumentClientId": "client id",

"DocumentClientUrl": "Client Site collection URL",

"DocumentMatter": "Matter Name",

"DocumentMatterId": "Matter ID",

"DocumentOwner": "Document Owner",

"DocumentUrl": "Document URL",

"DocumentOWAUrl": "Document Office Web App URL in case if this an Office document",

"DocumentExtension": "Document Extension",

"DocumentCreatedDate": "Created Date",

"DocumentModifiedDate": "Modified Date",

"DocumentCheckoutUser": "Document Checked out to",

"DocumentMatterUrl": "Matter Document library URL",

"DocumentParentUrl": "Absolute URL to Parent folder of this document",

"DocumentID": "Document ID generated by Document ID service"

},

"Document URL": {

"DocumentName": "Document Name",

"DocumentVersion": "Document Verison",

"DocumentClient": "client Name",

"DocumentClientId": "client id",

"DocumentClientUrl": "Client Site collection URL",

"DocumentMatter": "Matter Name",

"DocumentMatterId": "Matter ID",

"DocumentOwner": "Document Owner",

"DocumentUrl": "Document URL",

"DocumentOWAUrl": "Document Office Web App URL in case if this an Office document",

"DocumentExtension": "Document Extension",

"DocumentCreatedDate": "Created Date",

"DocumentModifiedDate": "Modified Date",

"DocumentCheckoutUser": "Document Checked out to",

"DocumentMatterUrl": "Matter Document library URL",

"DocumentParentUrl": "Absolute URL to Parent folder of this document",

"DocumentID": "Document ID generated by Document ID service"

}

}

## Matter Configurations

In matter center we have a Settings page which allows you to configure default values for a client for Matter Creation process. These default values are stored in this list which are later used by Matter Creation process to pre-populate default values. In the future any site collection specific settings will be stored in this list itself. This list should be present under each client site collection.

Following is the list structure:

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| Title | Single line of text | This column holds the static values as Matter Configurations |
| ConfigurationValue | Multiple line of text | This column holds the default values for the Matter creation process in a JSON format |

# Glossary

This section identifies any and all acronyms used throughout this document for the reader’s reference.

| **Acronym** | **Identification** |
| --- | --- |
| AES | Advanced Encryption Standard |
| API | Application Programming Interfaces |
| CSOM (diagram 5.1) | Client-Side Object Model |
| DMS | Document Management System |
| HTTPS | Hypertext Transfer Protocol Secure |
| IIS | Internet Information Services |
| IT | Information Technology |
| JSON | JavaScript Object Notation |
| MMS Store | Managed Metadata Store (Term Store) |
| OAuth tokens | Open Authorization tokens |
| OOB | Out of the Box |
| SKU | Stock Keeping Unit |
| SOAP | Simple Object Access Protocol |
| SPApp Token | SharePoint App Token |
| TB | Terabyte |
| UI | User Interface |
| WCF Service | Windows Communication Foundation Services |