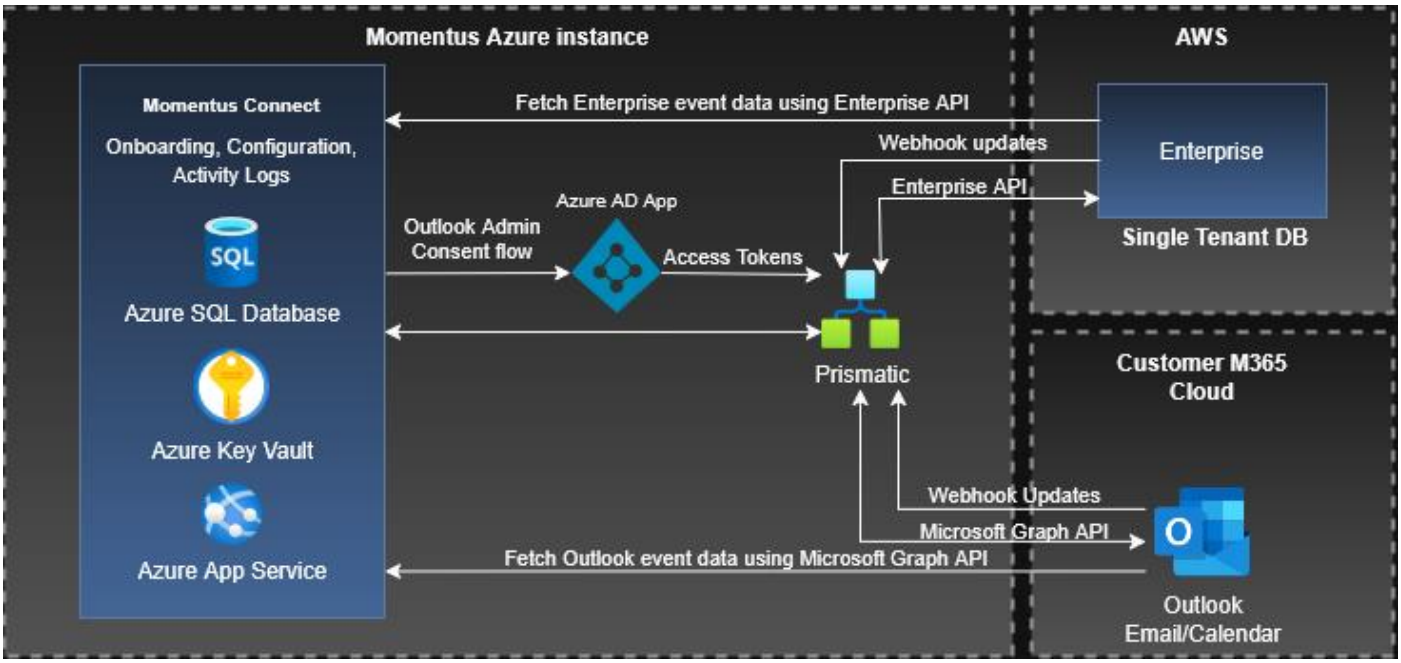
# Momentus Outlook Email Integration

## Technical Overview

## Enterprise-Outlook Email Integration Architecture

### Component Diagram:




Following is a brief description of a few of the components mentioned in the above diagram:

#### Momentus Connect

- Momentus Connect is a web application used to onboard and configure the Outlook Integration as well as to view the event sync logs. It can be accessed by visiting the 'Momentus Connect' link from Enterprise application.
- All webhook subscription creations occur within Momentus Connect.
- It is also the starting point of the initial sync flow. If initial sync is requested while activating the integration, Momentus Connect fetches the emails/meetings from the source system (Outlook) and sends them to Prismatic, where they are further processed and sent to the destination system (Enterprise).

#### Prismatic

- Prismatic is used to design and execute integration workflows.
- For active sync, the actions emails: sent, received, or meeting invites from Outlook to Enterprise are sent as webhook updates to the relevant Prismatic App which triggers a sync workflow and the source system data is processed as per the sync configuration and sent to the destination system.
- During this workflow, Prismatic communicates with Momentus Connect application and access-controlled Azure SQL database via authenticated API calls and Prismatics' standard SQL connector respectively for storing/accessing Activities and fetching access tokens of the source and destination systems.

#### Azure AD App

- Microsoft Graph API is being used to fetch email & meeting data of the customer's Outlook mailboxes that are configured to sync. We use Get access without a user mechanism which involves registering an

Azure Active Directory application configured with the required permissions (listed below). The “Azure AD App” shown in the above diagram represents this same application.

- As a part of admin consent flow, the administrator of the customer Outlook account grants permissions the Azure AD app needs after which the app can request an access token to invoke Graph API endpoints supported with the granted permissions. We use the OAuth 2.0 client credentials grant flow for this, which is commonly used for server-to-server interactions that must run in the background as per Microsoft.
- For requesting access tokens using the client credentials grant, we use the Access token request with a shared secret flow.
- Azure Key Vault is used to securely store the Client ID and Client Secret of our Azure application as Key Vault secrets. Access tokens generated through the OAuth 2.0 flow are encrypted and never reused. A unique token is generated for each webhook request from Microsoft, and it is discarded once the request is completed — no tokens are stored.

### Momentus Connect: Multi-tenancy Flow

We adhere to the standard Multitenancy .NET implementation outlined in the following article.
https://learn.microsoft.com/en-us/ef/core/miscellaneous/multitenancy

## Sync Trigger Mechanism

### Initial sync

- After configuring the Outlook Accounts and sync settings in Momentus Connect, the mail subscriptions are set on the individual Outlook Accounts.

### Active sync

- Active Sync is initiated whenever the subscriber account sends or receives an email or meeting request (based on configured settings). Each of these actions triggers a **webhook from Microsoft**, which includes an ID for the corresponding message or event.
- Momentus then uses this ID to retrieve the full email or meeting details and pass them to Enterprise for activity creation.
- Activities are created in Enterprise **only** if at least one recipient matches a contact or account, or if the Event ID or Event Opportunity ID is included in the email subject line. Emails that do not meet these criteria are not stored.

## Outlook Permission for Outlook Integration

Microsoft Graph’s Get access without a user flow is used to generate access tokens. The API application permissions that we request in this flow are:

1. User.Read.All
    - Grants permission to read user profiles for all users in the directory.
    - **Why it's needed:** Required for Momentus to confirm that the subscriber’s email account is active and valid within the Outlook environment.
2. Mail.Read
    - Grants read access to users' mailboxes.
    - **Why it's needed:** Required to create and manage mail subscriptions across users. This allows the application to monitor sent and received emails and associate those activities with users, as configured by the admin. Without this permission, email-based activity tracking would not function.
3. Calendar.Read
    - Grants read access to users’ calendar events.

- **Why it's needed:** Used for an optional feature that automatically generates activities from Outlook calendar invites. This permission allows the system to subscribe to meeting invites, enabling activity creation when meetings are booked.

## Note:

As Microsoft Graph REST API is used for communicating with Outlook system and Microsoft does not support REST APIs for mailboxes that remain on-premises from June 2023 as per the article The End of the REST API for On-Premises Mailboxes Preview, Enterprise – Outlook Email Sync Integration is not supported for Exchange on-premise mailboxes.