



## Case study

# Public sector infrastructure organization learns how to choose the right solution for integration

A public sector infrastructure service organization was implementing the Dynamics 365 Customer Service app and required integration with a public-facing website, several on-premises apps, and Office 365 services such as SharePoint Online. One of the implementation objectives was to gather insights from the data that was scattered across different business applications.

The organization was working with an IT vendor that had several years of experience building complex integrations using technologies such as IBM MQ and Microsoft BizTalk.

They decided to keep a uniform integration architecture and proceeded with a microservices-based architecture to build small, reusable components using service endpoints to gain the following benefits:

- Faster and independent development, deployment, and release cycles
- Simplicity of architecture to allow decoupled and reusable components
- Fault isolation

The organization chose Azure API Management to abstract their APIs and implement a secure integration layer.

As the team started building and testing the initial components, they identified some challenges due to the architecture:

- They experienced slow performance with batch-integration scenarios because they called the services as they would have in a point-to-point integration.
- They couldn't use standard functionalities that would have been available with out-of-the-box approaches such as SharePoint Online integration with Power Platform.
- For an aggregated view, they decided to replicate all data into the Dynamics 365 Customer Service app, which led to additional storage costs.
- They encountered throttling and API limits issues, which prevented successful completion of the integration.

At this stage, the organization decided to re-evaluate their standardized architecture principle and redefine their integration strategy. They divided their needs into two categories: batch and point-to-point synchronizations. They also adopted a “Configure-before-you-customize” approach.

With a reasonable mapping of different integration needs to the patterns, they were able to redefine a suitable integration architecture for these components.

For point-to-point integrations, the organization used the previously defined microservices-based architecture and incorporated the necessary changes for aligning with Power Platform API limits.

For batch synchronizations, the organization designed the architecture to manage the overall ETL processes with the following approach:

- They extracted the data from their source application by creating SQL Server Integration Services packages and storing that data in a staging database to perform all transformations.
- They hosted their staging environment in Azure SQL, within the same datacenter as the Customer Service app.

- They sent that data as batch data to the Customer Service app to reduce the number of individual connections.
- They considered API and throttling limits and built retries as part of the design.

For an aggregated view of the data, they realigned their approach to use Azure Data Lake, as Microsoft recommended. The out-of-the-box Export to Azure Data Lake feature required simple configurations to export data from the Customer Service app into Data Lake and didn't involve writing any additional code.

These changes in their overall approach provided significant benefits:

- Their batch synchronization performance improved significantly, and they cut the duration of the daily sync by more than 70 percent.
- Their “Configure-before-you-customize” approach let them choose out-of-the-box options to reduce development and maintenance costs.
- The retry mechanism, combined with monitoring, ensured that any failure scenarios were well handled and negative impacts on end users were avoided.

Even though the architecture realignment reset the organization's timeline, the benefits outweighed the risk because they were able to align their design with product features and the recommended practices. The organization already had a change-management plan in place to take care of stakeholder communication and alignment, and to plan trainings for all affected teams in a timely manner.