Modernizing Messaging in Secure Environments

Overview:

Our client produces revenue, in part, by accepting, processing and returning requests for information. A critical component of this process is a messaging queue platform that processes the requests. The replacement solution was selected by the client for Theoris to incorporate.

- ***Project Business Case:*** The purpose of the project was to address a technical currency issue for the messaging queue platform where the technology was at the end of life. A technical currency issue arose for the messaging queue platform.
- ***Project Objective:*** Address a technical currency issue by replacing the current Windows Service Bus with RabbitMQ, while maintaining or exceeding the performance.
- ***Project Benefits:*** Provide the messaging queue platform with technical longevity.

Technologies: Azure Dev Ops, C#, .NET, RabbitMQ

Solution Type: Application Development

Industry: Finance

Client Challenges:

Software development was needed to implement RabbitMQ into the message queue platform. The Theoris development team operated in a highly secure development, staging and production environments due to the sensitive nature of the data being processed. This led to issues getting the project code developed because the developers were not initially granted access to the RabbitMQ endpoint from the development server.

Solution:

To overcome the access issues, Theoris developers created a proof of concept of the integration with RabbitMQ on local, secure development laptops. This enabled the project team to keep moving forward while access was being provided.

Result:

The proof of concept was tested by Theoris and client personnel. After testing revealed an initial performance issue, the issue was quickly corrected, retested by Theoris and the client and implemented into the production environment. The project was completed on schedule and under budget.