Made with z Systems: Bankdata, Denmark

IN THE BEGINNING

Bankdata has thousands of core banking COBOL applications running on z Systems. New workloads in Web Services using XML and PDF formats have put pressure on Bankdata to modernize their infrastructure, as COBOL is not optimized for working with these data structures.

Initial attempts to extend their infrastructure with distributed platforms seemed to overcomplicate the architecture of the solution, resulting in increased management costs and specialist skills requirements.

THE BIG IDEA

Java was proposed as an ideal candidate to handle these workloads given the support for integration with COBOL on z Systems. Transformation between the two programming languages storage structures is very simple. Java was already a strategic programming language and application platform at Bankdata but only on distributed platforms.

Concerns were raised over Java's potential performance, which was a key requirement for Bankdata.

TAKING ACTION

Demo applications were developed by Bankdata for both CICS and BATCH. The application were a mix of Java and COBOL programs; Java handling SOAP message parsing and transformation for COBOL.

Adopting the new Java framework allowed Bankdata to employ industry standards to create a simple elegant solution, rather than extending and complicating existing architecture.

RESULT!

The Java project proved successful. Deploying a centralized solution on z Systems provided a robust and manageable architecture. The portability of Java skills was a key enabler for the project.

Bankdata were pleased with the performance of Java on z Systems. The implemented solution proved to be more than capable of handling the throughput of millions of transactions a day. Whilst economic efficiency was ensured by the zIIIP offload capability.

This project validated running Java on z Systems, opening the door for future projects to be deployed in this new architecture.

In a sentence? A new simple architecture has been founded, for using Java on z Systems, either as part of a joint venture with COBOL or as a pure Java application.