CASE STUDY

Leading financial organization modernizes IBM i development with multi-platform delivery pipeline using ARCAD and Azure DevOps

IBM i Azure DevOps

One of the largest multinational banking and financial services corporations in Canada sought to replace their legacy change management tool with one that supported their DevOps and automation requirements and the cloud-centric digital side of the business.

Challenge

In 2022, with a core business application developed in RPG and years of investment on IBM i, the digital banking division faced several key challenges:

"Siloed" IBM i development: IBM i development operated as a silo, using traditional IBM i tools. To accelerate software delivery overall, the bank needed to align IBM i development with the digital teams, already experienced with cloud-based Azure DevOps and Git.

Security Vulnerability: The infrastructure team had identified a critical security vulnerability due to direct connectivity between development and production systems. Complete isolation of environments would be required.

Manual Deployment: Previously with their traditional change management tool, application deployment was not working. The team had resorted to much manual effort moving source back and forth between environments while tracking the process in a different project system.

Concurrent Development: With a growing IBM i team of 40 developers, when parallel changes were necessary on the same source, the compare and merge process was manual and error prone.

Manual Handling of SQL, ILE: The previous tool lacked support for modern SQL and ILE, and was unable to manage temporal tables, user-defined functions (UDF), views, file conversions, etc. requiring extra manual effort in build and deploy.

Conflicts of Local Changes on Standard Package: Since the IBM i application was a locally customized version of the standard Fiserv package ICBS, each time a new vendor release arrived, the IBM i team needed to manually check for and resolve any conflicts in the code. The bank had already mixed original and modified objects making this difficult to manage.

Segregation of Duties: Three distinct teams were needed to complete the delivery of new IBM i application releases: development, network, and operations. In the interests of security, only the 3rd party network team had the authorization to move a release package to production. Developers prepared a package, the network team would transfer from Dev to Prod, then Operations would perform the install. This was a manual and fragmented process that slowed feature delivery.

Solution

After a thorough evaluation, the IBM i team selected ARCAD for DevOps thanks to its tight integration with Git, Azure DevOps, advanced automation of IBM i build and deploy, and built-in metadata repository which would bring a major productivity benefit for developers.
The bank already had a well-established hybrid cloud strategy based on Microsoft Azure. Mostly .NET development teams were using C# for ASP website & front-end app dev from Azure DevOps in the cloud. Azure Boards was standard for project management right across the bank, including IBM i.

The objective of the ARCAD for DevOps implementation was to integrate RPG development within the existing Azure DevOps configuration at the bank:

- Manage IBM i source code within Azure Repos/Git
- Orchestrate a CI/CD pipeline using Azure Pipelines
- Automate secure application deployment to 2 target LPARs using ARCAD DROPS and Azure Artifacts

ARCAD consultants worked remotely with the bank's IBM i and .NET teams to guide them through the DevOps journey.

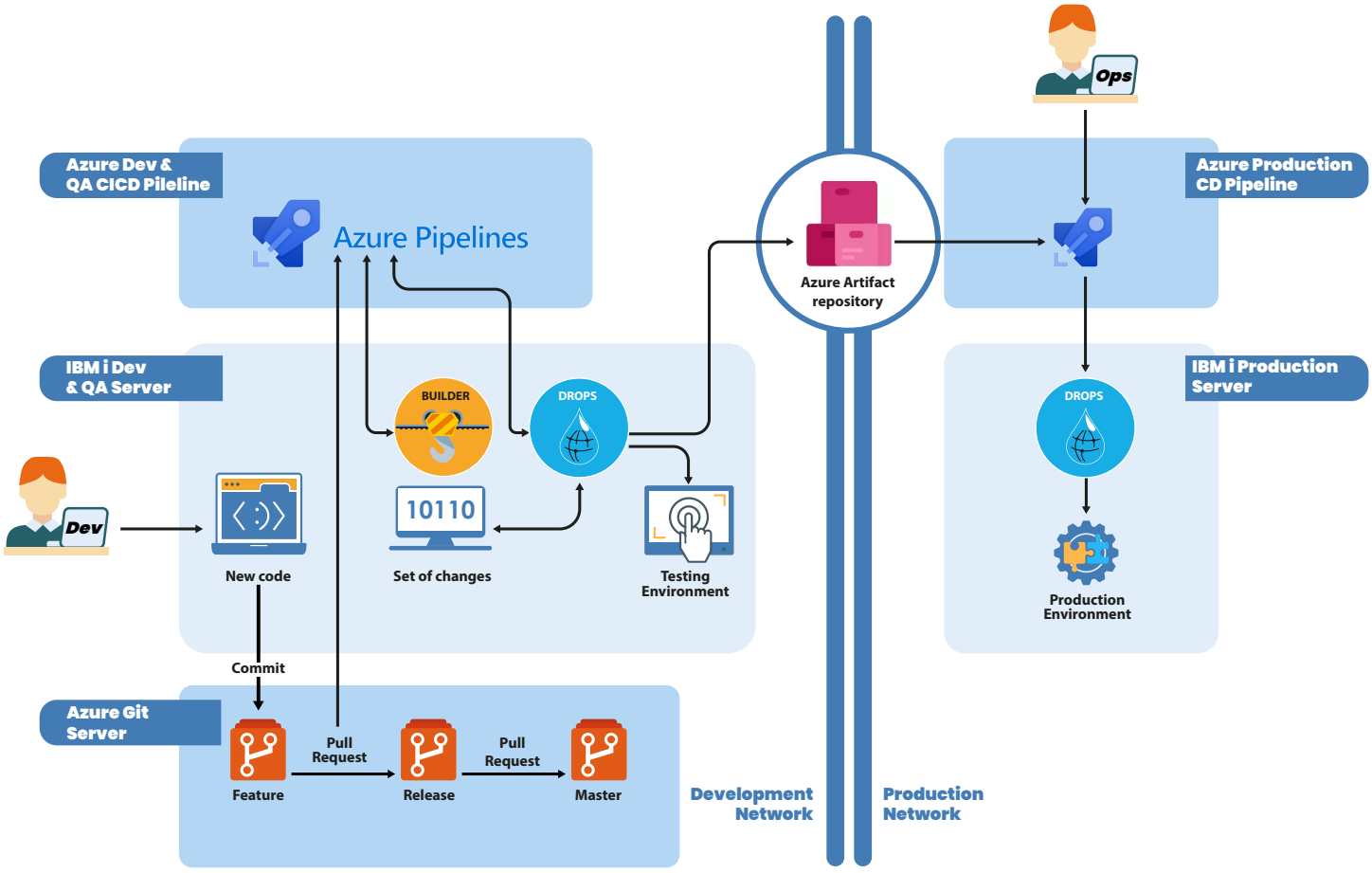
Key Features

ARCAD's built-in migration tooling ensured a smooth transition from the old technology.

Several key ARCAD for DevOps features were implemented:

Azure DevOps Integration on IBM i: The ARCAD team defined a specific IBM i workflow compatible with the Azure DevOps tool in place, using a standard Git Feature/Release model. A specific topology was implemented to manage incoming Fiserv releases and avoid comingling of code.

Deployment across Isolated Networks: Complete network isolation was achieved using ARCAD's DROPS. As no tool could span both development and production environments, to fully automate deployment and ensure consistency, Azure Artifacts was replicated on each environment and a separate copy of DROPS interacted with each of the Azure Artifacts instances controlled by the separate Azure pipelines.



Automated CI/CD on IBM i: A complete IBM i workflow was automated using ARCAD and Azure Pipelines, in respect of the isolated networks. A CI/CD pipeline now automates the build, packaging and deploy of the save file from Dev to QA, populating the Azure Artifacts instance in Dev. A second Azure pipeline in the production domain then uses the replicated Azure Artifacts instance together with ARCAD metadata repository knowledge to validate the incoming release and activate the deployment to production.

Synchronous Deployment of Java & native IBM i code: Previously, the deployment of JAR files to the IFS was an entirely manual process. Now with ARCAD for DevOps, the bank deploys Java synchronously with its related native back-end code.

Gains

The IBM i DevOps modernization project has brought significant benefits to the bank:

- **Standardization:** Automated, enterprise-wide Azure DevOps pipeline, spanning all platforms, including IBM i
- **Security:** Production and Development are now entirely isolated. Roles and responsibilities are clearly defined.
- **Cost optimization:** Application deployment is far faster thanks to automation. Deployment costs are reduced by eliminating manual effort and removing the 3rd party requirement.
- **Compliance:** Work items in Azure Boards are tightly linked to Git branches in ARCAD, giving complete traceability of code changes. Commit comments are automated with the work item ID.
- **Optimized workflow:** IBM i development is now agile with weekly iterations and scheduled deployments. Now with a Feature Release model in Git, instead of fragmented deploys of features into deployment, the bank deploys a single release into production, reducing the downtime window needed.
- **Testing Efficiency:** Instead of manually testing and deploying individual packages on deployment day, features are now tested together and deployed as a coherent release, making testing easier and more efficient.
- **Production Integrity:** The ARCAD workflow guarantees that the objects deployed and tested in QA are the same as those deployed to production – avoiding the risk of level checks and costly downtime.
- **Improved collaboration:** Now that IBM i development is fully integrated in Azure DevOps, open systems teams have complete visibility on DB2 table structure and IBM i development via the source repository.

Next Steps

Following the successful DevOps transformation, the bank is now looking to standardize on VS Code as IDE, using ARCAD's built-in extensions. Also, the team plan to reference .NET development in the ARCAD metadata repository, for cross-platform impact analysis and even closer collaboration between IBM i and open systems teams.

"The progress we've made is incredible. The team's feedback has been all positive. Initially, there was some hesitation regarding adoption, but both our onshore and offshore teams now enjoy using the new tools. ARCAD for DevOps has led to a reduction in work, and time savings. We are pleased with how it is operating and how it has leveraged our DevOps journey",

IBM i Development Manager

ARCAD SOFTWARE

www.arcadsoftware.com