data symphony

Creating Business Value, Driven by Data Intelligence

CASE STUDY

Seamless Solution for Complex Data Management Under IFRS17

CASE STUDY

Case Study

Seamless Solution for Complex Data Management Under IFRS17

Streamlined and scalable solution for financial data analysis, enhancing data accessibility and management

CASE STUDY

Overview

A financial services company sought to enhance its data management processes to meet IFRS17 requirements. We developed a sophisticated add-in that enhances Excel's functionality with SQL integration, providing a scalable solution designed to meet the rigorous data management requirements of IFRS17 while retaining ease of use.

Challenge

The implementation of IFRS17 introduced a need for significantly increased data granularity, which exposed the limitations of using spreadsheets for data management. Excel spreadsheets became unwieldy, prone to errors, and inadequate for handling large datasets or supporting multi-user access, ultimately leading to inefficiencies in both data management and analysis. In addition, the ecosystem lacked a well-defined workflow, further complicating the process. The key challenges faced were:

- Manual workflows prone to errors: The existing processes heavily relied on manual data entry and management within Excel, increasing the risk of human error and making it difficult to efficiently handle large volumes of data.
- Need for an automated workflow solution: Given the growing complexity of IFRS17 data requirements, there was an urgent need for a more automated and scalable workflow to streamline operations and improve data handling capabilities.
- Widespread reliance on Excel: Actuaries and accountants primarily used Excel for their financial analysis and reporting. Any new solution needed to integrate seamlessly with their existing tools, minimizing disruption and avoiding the need for extensive retraining.

Tools and Technologies

CASE STUDY

Framework and Libraries

- xlwings: Python library for connecting Excel with Python.
- Alpine.js: Lightweight framework for adding JavaScript behavior to the Excel add-in.
- Microsoft Yeoman JS Add-in Framework: Framework for building Office add-ins using JavaScript

Data Processing

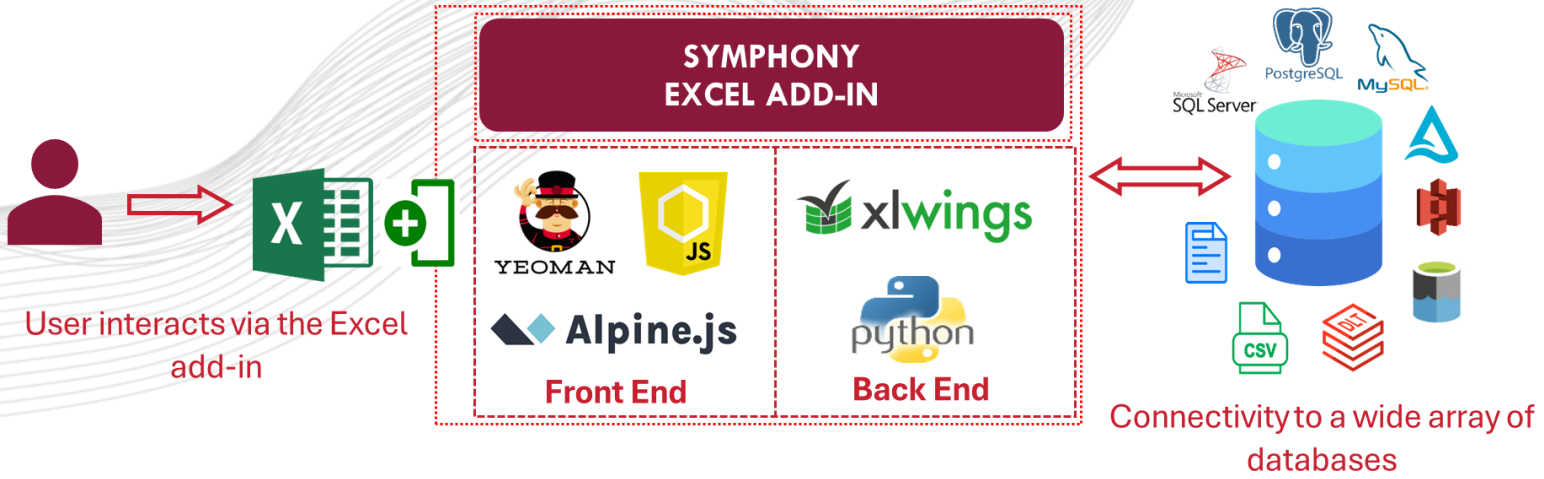
- MS SQL Server: Storing data
- SSIS: Importing and exporting data

Programming and Scripting

- Python: Used via xlwings for integration with Excel and data manipulation.
- JavaScript: Used with the Microsoft Yeoman JS add-in framework for developing the Office add-in.

Infrastructure and Deployment

- Docker: Used to containerize and deploy the add-in
- Service-Oriented Architecture (SOA): Enables scalability, adaptability, and cloud readiness of the solution.



CASE STUDY

Approach

To overcome the limitations of Excel in managing large datasets under IFRS17, we developed a scalable Excel-SQL integration add-in. This solution allowed seamless interaction between Excel and SQL databases, leveraging the strengths of both platforms while maintaining Excel's familiar interface. The key steps were:

- **Problem Identification**: We analyzed Excel's inefficiencies in handling IFRS17 data and recognized the need for SQL's scalability and multi-user access.
- **Design**: A hybrid system combining Excel's user-friendly interface with SQL's powerful data management capabilities was planned, using a modular, scalable Service-Oriented Architecture (SOA).
- **Technology Selection**: Tools like xlwings, Microsoft Yeoman JS Add-in Framework, Alpine.js, MS SQL Server and Docker were chosen for seamless integration, massive data processing, and consistent deployment.
- **Development**: A custom Excel add-in was created to allow data querying and management directly from Excel, with SQL handling large datasets efficiently.
- **Testing and Validation**: The add-in was stress-tested for large-scale data handling and security, ensuring it met IFRS17's high-volume and security demands.
- **Deployment and Training**: Deployed in cloud environments via Docker, the solution was made accessible to multiple users, with training provided to ease adoption.
- **Ongoing Maintenance**: Regular updates and improvements ensured long-term viability and adaptability to evolving business needs.

Outcomes

CASE STUDY

- **Seamless Integration:** Combines Excel's intuitive interface with SQL's powerful data management, allowing users to work within a familiar environment.
- **Scalability:** Supports handling large datasets, including petabytes of data, through integration with MS SQL Server, PySpark, Docker and a service-oriented architecture (SOA).
- **Enhanced Security:** Built with modern technologies like JavaScript and Python, ensuring adherence to contemporary security practices.
- **Cloud-Ready:** SOA design allows for cloud deployment and scalable server-side components, accommodating thousands of users.
- **Improved Data Integrity:** Provides efficient data handling, reducing errors and issues commonly encountered in large, complex spreadsheets.
- **Ongoing Support and Community:** Leveraging JavaScript and Python ensures long-term viability with a robust developer community for continuous improvements.
- **Reduced Development Costs:** Modern frameworks like JavaScript and Python reduce reliance on outdated technologies like VBA/VB.net, lowering maintenance and development costs.
- **Future-Proof:** Avoids the risks associated with using outdated and unsupported technologies such as VBA and VB.net.

data symphony

Creating Business Value, Driven by Data Intelligence

GET IN TOUCH

ask@datasymphony.com

www.datasymphony.com