

# Non-stop platform change

The Metzler Bank migrates to a modern technology



The Metzler Bank developed the Merian application suite nearly 15 years ago and has been improving it ever since. Merian now consists of about 70 applications and modules for managing securities, funds and bonds. Not only did more than 500,000 lines of code have to be migrated from Gupta Team Developer to C#.NET without disrupting normal application operation, but it had to have minimum effect upon ongoing development. The team from Metzler IT-Services GmbH accepted this challenge with the help of the tools from the Ice Tea Group, which specializes in such cases.

**I**n 1992, the Metzler Bank made the far-sighted decision to replace the old mainframe applications. Their

own IT development team converted all of the transaction processing for securities into a client-server application based on PCs. Gupta supplied the technological framework with two purely PC based products - the 4GL development tool previously known as SQLWindows and the SQLBase relational database. In this way the team, who were very familiar with the technical requirements, were not only able to create a working solution, but one that used very few system resources. Constantin Nicolaidis, who is now managing director of Metzler IT Services GmbH, remembers this wryly: „At first, when Merian became operational, no one would believe that a bank's securities depot could be run on a PC environment.“ However, further development of Merian (soon more than 100 were using it at Metzler) placed greater demands on the data management. In

1998, when three other banking firms decided to license the Merian software, Metzler replaced the Gupta database and converted the application to Oracle. SQLWindows, which Gupta in the meantime had renamed Team Developer, was not replaced.

## Future outlook: mediocre

As requirements grew, the Gupta development platform increasingly proved to be a bottleneck. It became evident that the Team Developer interpreter was not run-time optimized, there was very limited support for new Windows functions and, most importantly, it had become more difficult to find developers who knew how to work with a technology that was showing its age. More and more of the development time that was originally saved was now being invested in increased maintenance. Metzler IT Services distanced itself from Team Developer and from 1999 onwards developed new modules in Delphi. Up until 2003, the modules that were written using both technologies ran smoothly in parallel. After version 2.1 of Team Developer there were no fundamental improvements in the technology, which meant that technical problems became a common occurrence. At this point, Metzler made the decision to find a remedy for this „long-standing encumbrance“ to avoid winding-up in a cul-de-sac.

## Take this opportunity to start afresh

Firstly, a project group defined the requirements for the new platform: modern software architectures such as MDA should be supported, object orientation is essential, and the software should no longer be message-based but event-driven. Java and current Windows technologies were investigated in detail. Eventually, .NET was short-listed because development had previously been in a Windows environment. But how should

the existing applications be ported to this platform?

This question was easy to answer for the Delphi applications because Borland, the manufacturer, already supported .NET. The Ice Tea Group supplied Metzler with a tool to estimate the amount of work needed to migrate the complete Merian suite. This tool calculates the relevant measures such as lines of code or different item counts. The quotation for porting was then produced from this information. Porting with the „Porting Project“ proved to be considerably cheaper than the alternative, the then overdue task of programming the applications again, and so Metzler awarded the contract to the Ice Tea Group.

## The hot phase

Metzler started sending the source code of the modules over an encrypted ftp connection to the Ice Tea Group at the beginning of 2005. The first step involved using the „Ice Porter“ to compile the source code into C#.NET. The second step generated code, which was manually reworked if necessary, until it compiled perfectly under Visual Studio. NET. As Eberhard Fecher, a Premium Porting Partner of the Ice Tea Group explained, „The big advantage of the Ice Porter is that it leaves the structure of the applications intact. As a result, the generated code is easy-to-read and the developer, who previously worked with Team Developer, finds it ideal to maintain“. Metzler usually received the final C# code after a few days, but in some cases after one or two weeks, depending on the size of the module. They could then start the technical rework and validation. The tests proved to be the greatest time sink. „Our automated test procedures, built up over many years, no longer worked,“ said Nicolaidis „Recreating these was one of the biggest tasks of the whole migration“. By comparison, the more than 100 analyses and reports created under Team Developer were not much of

a problem: they were ported by the Ice Tea Group using Crystal Reports, a tool already deployed by Metzler in connection with the Delphi applications.

## Normal operation - step by step

After successfully completing the tests, each of the ported modules is deployed under .NET. Each module was replaced until, in September 2006, the complete application suite had been ported. Parallel operation of Team Developer, Delphi and .NET applications ran smoothly. „On the whole, our users don't notice any difference,“ said Nicolaidis. „The only thing that we sometimes hear is that the applications run faster after porting.“ Following on from the positive experiences at Metzler, the other banks that use Merian are now changing over to the new modules. However, this does not mean that Metzler can ‘rest on its laurels’. The Merian development team is now being trained in advanced .NET programming techniques. „What we have now is Team Developer code ported to C#, which works perfectly and can be maintained by our team with minimal effort. The next challenge is to modernize this code and make use of the latest possibilities offered by .NET“, summarized Nicolaidis. ■



**Michael Ihringer,**  
Freelance Journalist,  
Darmstadt