



Business challenge

To enable member banks to launch new applications and services faster and at lower cost, Fiducia & GAD IT AG needed to make it easier and more transparent to call existing services from new software.

Transformation

IT service provider Fiducia & GAD IT AG has introduced Java alongside COBOL within IBM® IMS™ on IBM z Systems®. By Java-enabling existing IMS core banking functions and presenting them to front-end developers via APIs, the organization is accelerating the creation of new services while extending the life and value of its applications.

Business benefits

Ensures

best delivery by making code open, agile and portable

Accelerates

application delivery; cuts costs with modern frameworks and APIs

Enriches

existing applications fast and at low risk, using more easily accessible skills

Fiducia & GAD IT AG

Bringing high-speed, low-cost, low-risk development to core banking systems

Fiducia & GAD IT AG (Fiducia & GAD) provides IT services to all 1,100 cooperative banks in Germany and also to private banks and companies in other industries. With administrative headquarters in Karlsruhe and Münster, and offices in Munich, Frankfurt and Berlin, Fiducia & GAD employs almost 5,600 employees and generates annual revenues of approximately EUR1.26 billion. The company's agree and bank21 banking solutions are used by more than 164,000 bank employees to manage almost 100 million customer accounts and 38,000 ATMs nationwide.

“We see Java on z Systems as a key technology in driving competitive advantage for our clients.”

—Pascal Meyer, Senior Enterprise Architect, Fiducia & GAD IT AG

Share this



In search of speed and simplicity

To help Germany's cooperative banks maintain their market share and customer satisfaction ratings, Fiducia & GAD constantly invests in improving its own service provision and technologies. Recognizing that retail banking customers everywhere expect faster and more efficient services, delivered at their convenience through a growing variety of channels, Fiducia & GAD saw that it needed to increase the speed and ease of developing new functionality.

Pascal Meyer, Senior Enterprise Architect, explains: "In general terms, our banking clients need to bring new applications and services to market more quickly, and we always want to drive down the costs while also improving the service levels we offer. With these things in mind, our primary objective was to make our banking applications available in a way that would be more familiar to the latest generation of developers. These were the key drivers for adopting Java within our existing IMS/COBOL runtime on IBM z Systems."

The classic approach to implementing core business transactions at Fiducia & GAD was based on Java/HTML5 at the front-end, connected

to a mid-tier layer in plain Java, and IMS and IBM DB2® for z/OS® at the back-end. In this kind of hybrid architecture, it is necessary to include additional "compensation" logic within every single component, increasing the overall complexity. The compensation logic ensures that operations distributed across the various components can be rolled back to the last consistent state in the event of an error or interruption. However, by bringing Java code inside the IMS environment, Fiducia & GAD knew that it could remove the need for this extra logic, and cut complexity, costs and delays.

Thomas Bauer, Team Leader and IT Architect, adds: "In terms of the application architecture, Java opens up the possibility of using a broader range of frameworks and protocols. For example, RESTful web services are more natural in Java than in COBOL, and certain functions that are relatively hard to implement in a pure COBOL environment are available as pre-built artifacts in Java. We wanted to cut time-to-market by deploying modern, re-usable software components—and, at the same time, protect our investment in existing business logic by Java-enabling it."

As Java was already a long-standing strategic platform for Fiducia & GAD, the organization was keen to enable greater portability of its code across both distributed and z Systems environments. This would support best delivery by enabling an ongoing free choice of platform, based purely on price-performance considerations. Equally, bringing Java to z Systems alongside COBOL would enable Fiducia & GAD to enrich existing core banking functionality in a seamless, low-risk manner, as well as making it easier for software architects within the distributed environment to call core transaction services directly from their own applications.

"In the longer term, our goal is to be more platform-independent by having Java building blocks that can run anywhere," comments Pascal Meyer. "Apart from the ability to reduce architectural complexity, what was really appealing about mixing COBOL and Java inside IMS was that it would enable us to build new application components tightly integrated with existing components. In this way, we saw that we could gain the advantages of a modern hybrid app—namely, short time-to-market, increased flexibility and greater ease of development—without losing the performance benefit of IMS where it really counts in core banking transactions."

Cutting complexity with Java

Fiducia & GAD worked closely with IBM developers in a multi-year project to optimize a common runtime for Java constructs within mission-critical IMS production environments. This common runtime is available as standard within the latest release of IMS, rather than being specific to Fiducia & GAD. Today, with Java and COBOL interoperable within IMS, Fiducia & GAD has modernized and revitalized its core banking systems without impacting their performance or reliability.

There are two major focal areas for Java on z Systems at Fiducia & GAD: first, the organization is building hybrid Java-COBOL applications for classic IMS processing and batch workloads; and second, it migrates native Java code from the distributed environment to z Systems whenever this approach makes more sense.

"Many of our batch jobs were spread across multiple platforms on both z Systems and the distributed environment, with different schedulers that we needed to coordinate," says Pascal Meyer. "This approach also required data exchange and/or data sharing. Today, by executing Java jobs directly on z Systems,

we need only a single scheduler so there is less complexity. We also achieve better performance because there is no need to move or convert data, and because the Java code sits right next to its data rather than needing to access it over a network.”

Fiducia & GAD is building a set of generic services in Java that can be exposed as APIs to developers of front-end applications running in its distributed environment. For example, a developer could use an API to call up the current balance of a particular customer’s checking account in order to display it in a mobile app.

Today, approximately 160 out of 350 categories of IMS transactions are Java-enabled at Fiducia & GAD. The organization expects ultimately to Java-enable 80 to 90 percent of the total; the 160 already achieved represents approximately 70 percent of the total workload, as many of the remaining transactions are called less frequently.

“Developers in the distributed world will not know if they are calling Java services or original IMS transactions,” says Pascal Meyer. “Everything will be accessed in the same easy and consistent manner, making it faster

and simpler to build new front-end applications that call the robust underlying transactions on z Systems. This supports our banking clients as they look to reach out to their employees and customers with innovative services delivered through web, mobile and whatever new channels may emerge in the future.”

“Adding Java to our z Systems environment makes it easier to present new services based on existing functionality,” says Thomas Bauer. “It also gives us the option to migrate our older COBOL code, though in practice we are only doing so in cases where we are building new business logic. There is no pressure to modernize just for the sake of modernization. We continue to value the performance and robustness of COBOL and IMS at the back-end, and we continue to plug those technologies into graphical front-ends running on distributed systems, as we have done for more than 20 years. In this way, the z Systems platform running IMS continues to be the focal point for business logic, feeding directly into a variety of different channels for the consumption of that business logic: teller applications, ATM systems, customer-facing web applications and mobile apps.”

Faster, simpler, more sustainable

By combining COBOL and Java code in IMS, Fiducia & GAD is reducing complexity and thereby increasing the efficiency and performance of end-to-end business transactions. Before, enterprise batch processing was handed by a spread of multiple platforms. In contrast, now all processing—regardless of the language—can simply be integrated into a single batch step by combining COBOL and Java.

Pascal Meyer comments: “In addition to reducing the complexity, we are achieving higher throughput by minimizing input/output—after all, the best I/O is no I/O! We also benefit from moving workload from GP to zIIP specialty processors, on which the licensing fees are lower. So although there is an increase in the total MIPS required—because Java consumes more resources than COBOL—the overall cost is lower.”

The introduction of Java on z Systems also provides development efficiencies thanks to its rich libraries, which enable programmers to avoid writing their own custom code for common functions such as compressing and decompressing data.

“Adding Java to our z Systems environment makes it easier to present new services based on existing functionality.”

—Thomas Bauer, Team Leader and IT Architect, Fiducia & GAD IT AG

Similarly, it is now possible for Fiducia & GAD to integrate third-party Java software into its z Systems environment, potentially enabling it to avoid in-house development altogether where a commercial solution already exists.

Additionally, Fiducia & GAD can easily port Java code across its distributed and z Systems environments, supporting best delivery by enabling it to choose the most appropriate platform according to price-performance considerations. As new hardware-based cryptographic, compression or networking features become available on the z Systems platform, Java for z Systems will support these enhancements immediately—enabling applications to leverage the new capabilities transparently.

Thanks to the pioneering cooperation between IBM and Fiducia & GAD, Java is now an established, production-ready component of the z Systems toolbox. At Fiducia & GAD, this significantly refreshes the IMS environment, preserving its traditional qualities of performance and robustness while enabling faster development using more widely available programming skills. It also allows the organization to enrich and modernize existing core banking functionality in a seamless, low-risk manner.

“As a provider of core banking systems to more than 1,000 banks with tens of millions of end users, we cannot afford to compromise on performance and reliability,” says Pascal Meyer. “Neither can we afford to introduce risk in the development practices. The z Systems platform running IMS remains our strategic choice because it is the most secure and reliable platform that we know of,

and because it offers exceptional performance for our core banking systems. With Java in IMS, we have the best of both worlds.”

Thomas Bauer adds: “By introducing Java directly into the IMS environment alongside COBOL, we are bringing the back-end closer to the front-end of the core banking systems—which run on Java on distributed platforms—and improving both efficiency and performance by reducing the architectural complexity. We also gain access to a larger pool of skilled developers, in particular among more recent graduates, so we reduce our long-term risk around the sustainability of code.”

Carsten Pfläging, member of the board and CIO of Fiducia & GAD, comments: “Combining IMS, Java and COBOL technologies is an effective strategic way of modernizing existing mainframe applications with minimal disruption, operational risk and costs.”

By revitalizing the z Systems environment and making it more accessible to new developers, Java is helping Fiducia & GAD to retain and add to the embedded value of decades of past development work. Existing functions can be extended through the addition of Java code alongside COBOL code, and presented to developers as APIs that make it easier to call common transactions from new applications.

Pascal Meyer concludes: “We see Java on z Systems as a key technology in driving competitive advantage for our clients. With easier development and re-use of existing components, it enables us to deliver new functionality at higher speed and lower cost, supporting our banking clients as they seek to bring new services to market more rapidly.”

Solution components

- IBM® DB2®
- IBM IMS™
- IBM Developer for z Systems®
- IBM Support Assistant
- IBM z Systems
- IBM z13™
- IBM z/OS®
- IBM z/OS Problem Determination Tools

Connect with us



Take the next step

To learn more about z Systems, please contact your IBM representative or IBM Business Partner, or visit the following website: ibm.com/systems/z

© Copyright IBM Corporation 2016. IBM Systems, Route 100, Somers, NY 10589.

Produced in the United States of America, December 2016. IBM, the IBM logo, ibm.com, DB2, IMS, z/OS, z13, and z Systems are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at ibm.com/legal/copytrade.shtml. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates. This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates. It is the user’s responsibility to evaluate and verify the operation of any other products or programs with IBM products and programs. THE INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.



ZSC03341-USEN-00

