DIGITAL ARCHITECTURES
STRATEGIES FOR THE UNKNOWN

By Dr. Wolfram Jost, Chief Technology Officer, Software AG

Digital architectures—blueprints for digital business

In the future, business will mean digital business, and software will be THE innovation factor. The result: The rules of the game are changing in the market. Companies are no longer exclusively consumers—now they are producers as well. They need to transform into software companies if they want to stay competitive in the digital age. Digitalization has a direct impact on a company’s IT and application architectures.

Digital use cases are as diverse as the business itself—and they cannot be purchased as off-the-shelf solutions. Digital business platforms give companies the agility and flexibility they need to succeed on the road to digitalization.

Consequently, this development means that the topic of architecture belongs on every IT manager’s agenda. While off-the-shelf solutions like ERP applications include their architecture like a “black box,” companies themselves must take charge of designing the architecture for future application scenarios. Only they have the transparency and vision to scale a suitable architecture for the new use cases based on the capabilities of digital platforms.

The time is right for a new architecture—now!

Software applications have a long history behind them. For more than 40 years, developers have been providing companies with IT programs to support their business processes. Back then, companies were considered modern and innovative if they even used software. These days, organizations simply cannot get by without software. IT is a must in the age of digitalization and, now more than ever, it takes a clever and sophisticated technological structure: To put it another way, an intelligent architecture.

A brief look at the history of software explains the demands on a digital architecture for enterprise IT.

The development of enterprise software can be divided into three main phases. During the first phase, custom software was developed for selected business functions, such as finance or human resources.
The development was always done completely in-house with external support. The first and only target system was the mainframe. The design principle was monolithic, so there was no need to discuss the mainframe architecture. It was already locked in by the choice of system. The tools used for software development, from planning to implementation, were still in the research phase and very rudimentary. The development, expansion and maintenance for these applications, therefore, required a great deal of cost and effort.

From the perspective of enterprise software architectures, the second phase was the era of packaged applications. In contrast to the software built by different companies, these programs were developed and marketed by external software providers. These off-the-shelf software packages no longer served just a single function: They began covering all the administrative processes within a business. “Enterprise Resource Planning” (ERP) became the established term for this type of application. However, packaged applications also dictated the architecture, even when the monolithic design principle was replaced as a result of technological progress by a client-server structure and web architectures. This meant that the application packages covered the limits of a company’s innovations in business models and processes. To put it another way: The external software firms dictated the direction and speed of innovation.

The third phase is happening right now and focuses on the topic of digital platforms. It differs significantly from the two previous phases. In the past, the focus was on optimizing costs and standardizing administrative processes. Now digital platforms enable businesses to create new business models through the fast development and integration of innovative, differentiating digital applications. These platform-based digital applications are shifting the focus from the back office to the front office—where the company does business with customers and partners.

New digitalized business processes are changing the way companies interact and collaborate with their customers. First, customers are at the heart of a company’s interests: Their wants, needs and purchasing behavior are analyzed in real time. Then the company translates the resulting information into new business models to create new digital customer experiences. Second, the “co-factor” is becoming increasingly important: Cooperative business models are breaking down the traditional, entirely company-centric focus. Companies are both software consumers and providers at the same time.

Consequently, businesses now need to deal intensively with the architecture of digitalization themselves—because unlike previous enterprise applications, digitalized business processes cannot be purchased off the shelf as a complete solution. Hence, the company is responsible for the design and customization of the technology basis—the technology architecture. The real importance of architecture design is evident in the fact that architectural decisions influence the nonfunctional characteristics of an application solution, such as performance, scalability, resilience, usability, updateability, security, interoperability and maintenance. Of course, this interdependency is not new. In the past, the limits of the selected software package were noticeable—for example, if a larger number of users or higher data volume had to be handled. But these days the companies themselves set the limits of their digital application architecture. Putting more effort into defining the right technology architectures takes more time up front, but it pays off afterward with greater agility and flexibility when digital business processes need to be created or adapted.
Definition and role of the technology architecture

A technology architecture describes the structure, interconnection and organization of the different software components within a larger application. It illustrates how a system is designed from the ground up. Starting from the requirements of different departments and the derived use cases, the architecture shows how the entire business logic is structured into different modules and how they are connected with each other. In addition, the architecture describes the required IT resources and how these are utilized by different modules in distributed systems.

The key drivers for selecting the right technology architecture are the amount of data the application has to process, the number of users and the degree of flexibility, performance and scalability that is needed. An application that serves millions of users and processes terabytes of data requires a different architecture than an application for 50 users and a few gigabytes of data. Likewise, an application that processes data in real time (seconds or milliseconds) needs a different architecture than a no-latency critical application.

Transforming business ideas into technology architectures is a complex and challenging task. Developers always run the risk of over- or under-engineering the design, which can result in insufficient process support, a lack of agility and scalability, cumbersome operational management and low user acceptance.

Limits and change in technology architectures over time: from the mainframe to the cloud

The use cases of IT in businesses have always been and continue to be the result of technological opportunities and feasibility of software technology. From the start, innovative breakthroughs in hardware, network and storage technology, and software development have driven change in enterprise IT. The technology architecture—or more specifically, the transition of architectures—is, therefore, a reflection of technological progress, as a look at the different episodes over time makes clear.
EVOLUTION OF ARCHITECTURES

**Mainframe era**
Mainframes have been around since the 1950s, but they became the stars of the IT world in the ‘60s and ‘70s. They remain an important component of many IT landscapes to this day, although in a completely different form.

The first architectures were very simple. Software and hardware were rigid and hardwired, which meant that applications were always developed for a specific hardware system. A huge central mainframe system handled all the tasks, while “dumb” terminals were used for display and entry. A monolithic application on the mainframe contained a company’s entire business and data logic.

This type of architecture was simple to maintain, but it lacked agility and flexibility. It was easy to scale up the mainframe’s bandwidth, but a scale out was impossible. Progress in software technology and programming led to an initial standardization and logical separation of software levels. In addition to hardware producers, the first software companies, such as Software AG with its high-performance database, Adabas, became a driving force in building the software (product) industry.

**Client-server era**
The developments that emerged in the mainframe era led to the next generation of technology architectures, the client-server architecture. The major progress over mainframe computers was due to the separation of the entire application’s business logic into three layers: client (UI), application server (program, process and decision logic) and database server (data logic). This radical change was made possible by a variety of software-based interventions: UNIX® as a hardware-independent operating system, SQL as a standardized database query language, distributed object models and new object-oriented programming languages for program and data logic. At the same time, the size of hardware was shrinking, thanks to progress in chip and storage technology. Computers no longer required entire rooms and found their way upstairs from the basement into offices.

The client-server architecture was a real breakthrough. It led to a radical new way in designing and developing user-friendly interfaces as well as greater independence between hardware and software. Another key aspect was that the client-server model represented more than just a technological change: It also signified an organizational change from centralized to decentralized responsibilities. That was when business units began to engage in the topic of IT.
The webization of IT
Client-server architectures were followed by web architectures. The biggest difference from the client-server model—and at the same time the essential innovation—was that the browser became the new front-end for accessing business applications via the internet. Parts of the business logic and its operation were now managed by the browser rather than a dedicated client system.

Generally speaking, web applications are computer programs that allow users to access data in a database over the internet using a web browser. The information is generated dynamically by the web application through a web server (in a specific format, in HTML using CSS, for example). Web applications can be deployed quickly anywhere at no cost and without any installation requirements on the client side. Doing business through the internet led to completely new business models and interaction patterns between companies and their customers. This opened up an entirely new sales and communication channel for companies whose potential was described with the keyword “e-business/e-commerce.”

Service-Oriented Architecture (SOA): stylistics for distributed application architectures
In the context of the webization of the client-server model, SOA is more of an architectural design paradigm than a concrete architectural implementation pattern. The core idea of SOA was to deconstruct monolithic client-server applications into smaller, independent components called services, which communicate with each other through defined service interfaces and can be coordinated into higher-value services. Opening up these service interfaces led to external applications being able to easily consume them as well.

The way the service's business logic was implemented is insignificant. The technical implementation as well as the complexity remain hidden behind the standardized service interface, which acts as a contract between the provider and consumer of the service.

This device allowed SOA to have greater flexibility and agility with integrating, changing and maintaining the application logic. Despite the ingenuity of the concept, the success of SOA was very limited. The main reason for this was that the methodology for efficiently implementing SOA principles was there but not the tools and technologies (yet).

Cloud computing—IT service as an internet service
The newest paradigm of a technology architecture is called cloud computing, the demand-based use of IT services as a rental service. All IT resources or capabilities of an application or an IT resource are provided and consumed as a service via the internet. This means that nothing is installed on the customer or user side. Key characteristics include web-based front-ends, an almost unlimited combination of IT services, dynamic scalability and multi-tenancy.

Cloud computing provides an architectural pattern, deployment concept and business model in one comprehensive approach. All data or capabilities of an application or an IT resource are provided and consumed as a flexible service through the internet. This means that the customer or user side is relieved of the complex task of setting up and implementing a physical IT infrastructure, which further eliminates the cost-intensive acquisition of hardware and software. In addition, cloud computing and its subscription-based pricing models replace the traditional perpetual software licensing models with annual maintenance fees.
Launch into the digital age—new concepts, paradigms and technologies for digital architectures

Digital Transformation is everywhere around us now. In the future, every business will also be a digital business. Everything that can be digitalized, will be—now and in the future. But why is that? To put it simply: because we can!

In concrete terms, digitalization enables the creation of new business models and opportunities by connecting the physical world with the digital. Digital business models are based on platforms, not on finished (software) products. Only platforms provide the flexibility and agility needed for rapid, continuous change and adaptation. Digital technologies like cloud, mobile, social networks, the Internet of Things (IoT) and big data form the foundation. Digitalization is a disruptive change in how companies design, develop, deliver and sell products and services to their customers and build partner ecosystems that create value.

The Digital Business Platform

Platforms like the Digital Business Platform from Software AG do not consist of a single monolithic system. Rather, they have various building blocks that are implemented individually and can be supplemented gradually. Software AG's Digital Business Platform covers the technological requirements of digitalization with the following five building blocks:

1. Building block one: for business and IT transformation. Digital Transformation is a process in and of itself. This transformation process must be professionally planned, managed and monitored. Therefore, the content transformation, the processes and organization, and the IT environment are not only a technological task but a management task as well. It must be supported with tools and methods for smart decision making in the future. Business Process Analysis (BPA) and IT management tools (like ARIS and Alfabet) are very valuable here.

2. Building block two: for in-memory data storage. In the future, data will be managed in-memory for real-time access and analysis within milliseconds. Performance and scale are the key challenges of digital business. In-memory tools (like Terracotta) deliver the technological foundation.

3. Building block three: for integration. Monolithic applications will continue to exist, but other systems will join them. These include SaaS applications, mobile applications, big data applications, IoT applications, social networks and partner applications. All of these systems need to be integrated to achieve seamless and efficient processes. Therefore, integration is the order of the day—which integration tools and API management tools deliver (e.g., with webMethods).

4. Building block four: for processes. Companies that want to change their business must change their business processes based on new business applications. This is at the core of every business model. In the end, it is the process innovations that distinguish digital companies from traditional ones. Redesigning the differentiating business processes through the quick and easy development of new digital business applications is, therefore, a key element of digitalization. Supporting components for these include agile apps and Business Process Management System (BPMS) tools (e.g., webMethods BPMS and webMethods AgileApps).

5. Building block five: for real-time data analysis and decision making. Making faster and better decisions in real time is also at the core of every digital strategy. Collecting, storing, analyzing and visualizing data that is relevant to decision making is, therefore, a crucial ability for a digital enterprise. This applies to both static as well as real-time data. Components like Apama and webMethods Operational Intelligence are available for real-time analysis and operational intelligence.
Due to the inherently disruptive effect of digital models and opportunities, the demand for IT’s service promise is growing. They need a higher degree of agility and flexibility, lower latency, greater scalability, improved user experiences, automated decision making based on real-time analytics and a hybrid transactional/analytical processing. A new generation of application architectures that supports the major software and operating concepts, such as cloud, microservices, containers, API management, DevOps and event management, is imperative.

**THE NEW ARCHITECTURAL PARADIGM**

At the same time, people need to say goodbye to the idea of a monolithic architecture that covers all business tasks within a company—comparable to the traditional ERP requirements. To move ahead of the competition and roll out differentiating innovations, companies need to design, develop and implement digital applications in an agile and flexible way. However, these application functions can only be provided at the desired speed and quality if businesses take a new path with their application architecture style.
Building agile software: microservices

Designing, developing and deploying differentiating and innovative digital applications in an agile and flexible way is a prerequisite for digital business processes. Companies must set themselves apart from the competition with their applications to compete and succeed. Delivering innovative applications and application features with the required speed and quality demands new ways of designing application architectures. With microservices, a new architecture paradigm is establishing itself as the bearer of hope for better agility, productivity and sustainability in the digital application world.

Microservices are “smaller” software modules of a bigger, more complex application that can be implemented, deployed, updated and managed independently. As such, they take up the SOA concept, but go beyond it. While SOA was more about external deployment of service interfaces, microservices deals more with the inner design of a modern enterprise application. Using microservices, monolithic applications can be broken down into small program packages that function autonomously in their technology environment. As a result, the development can be distributed across an unlimited number of teams. Independent of the other teams, each team can implement, improve or expand its own microservice.

This type of architecture design is appealing because the agility and speed of software development increases, thus fulfilling a fundamental requirement for digital applications. Companies are able to deliver new innovations within days instead of months or years, creating new business opportunities faster and speeding up productivity growth. Moreover, development becomes more sustainable because the individual services are largely independent of each other. Consequently, a standardized technical basis is unnecessary. Unlike monolithic applications, microservices can go into production independently of each other. The alignment and coordination procedure that would otherwise be needed, for example, if a team would like to update a tool’s program library, is eliminated.

**Figure 5: Microservices Architecture**
Agile deployments: containers
The advantage of rapid microservice deployments takes full effect when combined with the (operating system) container concept. These containers represent an important virtualization and abstraction technology in infrastructure automation, for example, when application services need to be transferred quickly from the development environment into the testing or production infrastructure. Unlimited copies can easily be deployed if an application service must scale up to meet higher demand. Containers encompass the entire process environment of a microservice. Unlike the technology of virtual machines, they enable OS kernel and some library sharing while maintaining separate application spaces. For this reason, a container package can be provisioned within seconds or milliseconds, when it might otherwise take several minutes. Like their namesake in transport logistics, containers offer a predictable, reliable environment regardless of content. Developers are in the pleasant position of being able to concentrate exclusively on programming the business logic, while technical deployments are largely automated. Container management solutions like Docker® make agile development of applications much easier, because changes are implemented and activated with the speed provided by the digital world.

Agile connection management: APIs
The individual microservices are typically connected through Application Programming Interfaces (APIs). An API is a software component that provides functionality to access an underlying software resource (endpoint). The program interface describes the data (input/output) and operations that the resource (application) provides. As such, an API is an abstraction layer (virtualization layer) on top of a software resource providing the business logic. APIs act as a doorway to functionalities and data sources without their technical implementation playing a role. The consumer of an API can rely on the fact that the interface is stable, independent of changes in the microservices.

Every application and architecture component can benefit from the features of the API concept. API management technology is fundamental to digital architectures for managing interface invocations and transparency of the available APIs. It prevents losing oversight of invoking services in the mesh of communication channels. API management technology with gateway and portal components support IT teams in designing, implementing, deploying, operating, discovering, consuming, governing and monitoring service interfaces. The API gateway primarily acts as a facilitator to control and manage communication between the invoking system and service API. The portal is best considered a window into available APIs as well as a collaboration platform for sharing and developing new interface proposals.
API MANAGEMENT PLATFORM

Figure 6: API Management Platform

Agile process chain: continuous delivery and the DevOps paradigm
Paradigms like microservices, containers and APIs, including the new cloud operator model, contribute to a new process improvement method called DevOps, which is quickly growing in popularity. The word was coined as a combination of the terms (software) development and (IT) operations and applies the idea behind agile development methods to IT operations.

WEB SCALE ARCHITECTURE
DEVOPS AND CONTINUOUS DELIVERY

Figure 7: From Development to Deployment

Simply formulated, tools, test scenarios, deployment configurations and organizational processes reach into the entire process chain—from development to deployment. A powerful continuous deployment approach is implemented that accelerates time to market for new digital business models and process changes with shorter release cycles. The independently functioning microservices, container virtualization and automated deployment in a cloud environment create convincing technological prerequisites.

Integration—reinterpreted
In the case of digital business models and processes, companies start out in agile mode. To move ahead of the competition and roll out differentiating innovations, companies need to design, develop and deploy digital applications in an agile and flexible way. At the same time, the fundamental idea underlying new application architectures is that they follow an open, heterogeneous approach, not a monolithic one as in the past.
Therefore, a digital architecture must have the ability to integrate data and processes from standard applications for administrative business processes, legacy applications based on solid and highly stable platforms, SaaS and PaaS applications, data lakes, mobile applications and B2B applications into new digital business processes on demand.

Integration technologies are not fundamentally new, but their character is changing significantly in the context of digitalization. While in the past integration was seen as a type of middleware connecting a few internal applications, today it is at the heart of a digital architecture because of the vast number of data sources to be integrated. Naturally, this has major effects on the architecture of integration technology. The first generation was called an Enterprise Service Bus (ESB) and was essentially a monolithic system that provided all the functionality to design, develop, deploy and monitor integration logic. Routing, mapping, transformation, orchestration, security and mediation were all key capabilities. The capabilities remain, but the internal architecture is changing. Based on the specifications for the agile design model for digital architecture, those capabilities will be provided as microservices. The previous monolithic ESB concept is transforming into a modern microservices container, providing the needed integration capabilities in a very agile and flexible way.

**FROM ESB TO MICROSERVICES CONTAINER**

![Diagram: From ESB to Microservices Container]

**Digital application architectures—supporting technology concepts**

In addition to the previously mentioned methods, paradigms and technologies, additional core technology concepts support developers in designing and building agile digital architectures. In particular, these include business process management in application-oriented development, in-memory computing as an “accelerator” in data storage, and streaming analytics for handling real-time demands.

**Business process management**

The term “Business Process Management” (BPM) has a long history in the IT field. BPM was born as a discipline back in the 1990s and has continuously evolved in recent years. The use cases range from workflow and process automation to composite applications, process integration and application development. Even if BPM failed to keep all of its promises, there are many positive examples where it delivered more efficiency, flexibility and agility in process execution. These are precisely the features that distinguish digital application architectures.

Therefore, there is no doubt that BPM plays a key role in a digital architecture. Digital business models can only be implemented on new and innovative processes enabled by modern process applications. To offer a digital customer experience, processes must become more flexible, faster and simpler. Likewise, all IoT use cases (smart insurance, predictive maintenance, etc.) also require new process flows. In the context of the digital enterprise, BPM is increasingly shifting its focus toward application development.
Workflow, process automation and composite applications will remain important use cases. But the fast and easy development of digital process applications will become the predominant use case. To provide the needed flexibility in development and agility in deployment, process flows and process applications are also integrated in the microservices architectures.

**In-memory computing**
The concept of in-memory computing is an outstanding example of how new technology options are clearing the way for new use cases in IT. For many decades, it was clear that data had to be stored on hard drives. There were two main reasons for this. First, main memory was not big enough to hold large volumes of data permanently. Second, main memory was expensive. Both of these considerations have changed drastically in recent years. Main memory has become cheaper with higher storage volumes (multi-terabytes). And thus, the idea of in-memory computing was born.

In-memory computing describes an architecture where all the application data required for processing is stored in the main memory. The basic underlying idea is that the computer's main memory can store large amounts of data for sharing across several, possibly distributed applications. The key advantages include greater speed, better scalability and faster insights. These specific features are indispensable for digital architectures, which is why they cannot manage without in-memory data. For example, only web applications with in-memory technology can manage the workload from multi-channel scenarios where millions of users are accessing data and expect a low-latency response.

**Streaming analytics—IoT data stream management**
The digital world is a real-time world, meaning that data and events have to be processed at the moment they occur. So decisions cannot be delayed because data needs to be collected, stored, analyzed and visualized before a decision can be made. When a car accident happens, the crash data must be processed immediately to mitigate the risk for the driver or other people involved in the accident. When a manufacturing machine is close to overheating, the relevant event data must be processed instantly to avoid a malfunction. When a patient's heart rate or blood pressure exceeds the safe threshold, the hospital has to be informed without delay.

The streaming analytics platform provides the capability of processing event streams in real time. This platform can consume and provide data from any internal or external data source. The streaming engine can trigger alerts, send data to the BPM process engine and kick start other processes, or simply make the data available for further use through API management.

**Facelift for digitalization—integrating traditional applications**
An application and technology architecture for digital use cases is rarely developed from the ground up. Ultimately, every company uses applications that were built many years ago and still are an important part of their entire IT environment. These legacy applications support core business processes in a way that no other application ever could. The enormous value that enterprises derive from these applications comes from the company-specific business logic that has grown over decades. This unique business logic is the key value for a company and cannot be replaced easily. Rebuilding these applications would be time-consuming, risky and add limited value at best.

Modernizing the existing applications promises to be far more successful. This type of facelift is possible in different segments of the application. One of these areas for modernization is the user interface. In this case, the legacy application will be equipped with a new, modern web-based user interface through API management to provide better usability and ease of use. The data side could provide another area, moving application data continuously to an in-memory data grid for advanced data analytics as part of new processes. Integrating legacy applications with cloud or mobile applications is another option for adding value during a Digital Transformation. Real-time analytics offers another potential area of improvement by integrating streaming analytics per API that analyze data manipulations at the moment they occur to prevent fraud, for example.
All the above-mentioned opportunities for modernizing legacy applications provide a much more predictable and efficient road to success than attempts to completely replace a proven solution with an entirely new application. Therefore, legacy applications are an integral part of a digital architecture—even if they come from the pre-digital era and have to be managed in a stable mode unlike agile digital processes. At the same time, the integration offers further proof that designing an application and technology architecture lies in the company’s hands and can no longer be automatically purchased in a packaged application. And a digital business platform like the Digital Business Platform from Software AG that enables the implementation of digital architecture designs is, by nature, suited to such a bimodal deployment.

**Use case: IoT architecture**

**INTERNET OF THINGS ARCHITECTURE**

These days, the Internet of Things (IoT) is without a doubt the most important use case for a streaming analytics platform. The streaming analytics platform is the foundation for building IoT applications. IoT applications are developed to collect data from physical endpoints like machines, cars and motorcycles, analyze it in real time and store it for further analyses. The objective here is to improve the availability, utilization capacity and service life of physical assets to improve a company’s profitability. The IoT is the pivotal element of Digital Transformation. With the ability to collect and analyze data from physical devices in real time, businesses can create new digital services for their customers and optimize the usage of their products. Moreover, IoT data can also help them improve product design and capabilities, which leads to a better customer experience. Predictive maintenance, smart insurance and smart manufacturing are typical IoT use cases.
Digital business models and processes require digital architectures. Only digital architectures provide the agility, flexibility and speed that businesses need to survive in the digital world. Implementing a digital architecture should be a high-priority objective for every company and public institution. The days where architectures were covered up by monolithic off-the-shelf applications are over. The foundation for implementing a digital architecture is a digital business platform. Digital business platforms provide all the essential capabilities and components to design, implement and monitor digital architectures. The new concepts, paradigms and technologies that are available for designing, building and implementing a digital architecture are described in this white paper to give CIOs, IT managers and enterprise IT architects fundamental guidance and support their decision making.
**One last thing: Enterprise modeling and IT portfolio management**

It has already been indicated that Digital Transformation is associated with a change in business models, business processes, organizational structures and application landscapes. These business changes give it that “something extra.”

Digital architectures can only live up to their full potential if the underlying application concepts bring the necessary degree of differentiation and innovation. To ensure this is the case, the planned business innovations should be modeled and analyzed in advance using appropriate methods. Different scenarios can only be discussed and analyzed based on this type of model so the necessary decisions can be made systematically. Examples of these methods include customer journey mapping, process mapping, business model canvas, IoT modeling and customer experience modeling.

This applies in the same way to managing changes in the IT portfolio. The transformation from current status to the target status is generally executed in a variety of projects. The project portfolio also needs to be modeled to ensure continuous progress monitoring. And this applies equally to the application and service portfolio as well. The current and target status also need to be modeled here in order to manage the transition process in a structured way.

Therefore, enterprise modeling and IT portfolio management are not part of a digital architecture, but are nevertheless a key element of a Digital Transformation. Without knowing what the actual business objectives are and how to manage the transition from current to target status, the Digital Transformation will inevitably fail.

**DIGITAL ARCHITECTURE FOR THE 21st CENTURY**

![Diagram](image-url)

*Figure 11: The Digital Architecture of the 21st Century*
The digital transformation is changing enterprise IT landscapes from inflexible application silos to modern software platform-driven IT architectures which deliver the openness, speed and agility needed to enable the digital real-time enterprise. Software AG offers the first end-to-end Digital Business Platform, based on open standards, with integration, process management, in-memory data, adaptive application development, real-time analytics and enterprise architecture management as core building blocks. The modular platform allows users to develop the next generation of application systems to build their digital future, today. With over 45 years of customer-centric innovation, Software AG is ranked as a leader in many innovative and digital technology categories. Learn more at www.SoftwareAG.com.