



Enterprise Services Architecture – An Introduction

Table of Contents

- EXECUTIVE SUMMARY 3
- CHALLENGE FACING IT ORGANIZATIONS 4
 - No Silver Bullets* 4
 - EAI only provides a partial answer..... 4
 - Web Services Begin to Address IT Challenges* 5
 - Address Issues of Heterogeneity and Reuse 5
 - Enhancement Needed to Support Enterprise Scenarios 5
- ENTERPRISE SERVICES ARCHITECTURE EXTENDS WEB SERVICES BENEFITS 6
 - What is Enterprise Services Architecture?* 6
 - Order-to-Cash Without Enterprise Services Architecture..... 6
 - Order-to-Cash with Enterprise Services Architecture..... 7
 - Key Characteristics of Enterprise Services Architecture 7
 - Enabling Enterprise Services Architecture with SAP NetWeaver* 8
- WHAT ENTERPRISE SERVICES ARCHITECTURE MEANS FOR CUSTOMERS 8
 - Leverage Existing IT Assets*..... 9
 - Extend IT Assets to Deliver New Business Value* 10
 - Innovate* 11
 - Enable Adaptive Business Solutions 11
 - Enable Flexible Organizational Structures 13
 - Enterprise Services Architecture Changes How Solutions Are Built*..... 13
 - SAP Roadmap For Enterprise Services Architecture* 14
- CONCLUSION 14

Executive Summary

Changing business models, growing competition and globalization, tighter regulation, and increasing M&A activity are combining to accelerate the pace of business change. More than ever, success rides on IT's ability to adapt rapidly to evolving business needs. CIOs need a robust, cost-effective way to leverage and extend a heterogeneous collection of enterprise applications to support new requirements and enable innovation. It is increasingly costly and inefficient to stitch together new business processes that span disparate applications or cross-organizational boundaries or that require analytics and collaboration. For IT organizations to enable business agility, they must ensure that enterprise applications are not only high-performance business engines driving efficiencies, but also become flexible building blocks of future business systems. A clear blueprint for evolving existing architectures is needed.

The mid-90s answer to these problems, enterprise application integration (EAI), has proven to be costly to implement and even harder to change over time. More recently, Web services enabled a large step forward toward flexibility across a heterogeneous landscape. However, current implementations have yet to unlock the true power of Web services. Most Web services today expose functionality of individual applications and are too fine-grained to be efficient building blocks for enterprise-wide business processes.

Creating new value from existing IT assets calls for new answers. SAP's answer is Enterprise Services Architecture, an open architecture for adaptive business solutions, enabled by SAP NetWeaver™. Building on the benefits of Web services, it delivers on the promise of services-oriented architectures, enabling both flexibility and business efficiency without increasing costs. With Enterprise Services Architecture, companies have a cost-effective blueprint for composing innovative new applications by extending existing systems, while maintaining a level of flexibility that makes future process changes cost effective.

Enterprise Services Architecture will move IT architectures, step by step, to dramatically higher levels of adaptability and help companies move closer to the vision of the real-time enterprise.

Challenge Facing IT Organizations

IT today plays a larger role than ever in driving business success. The systems created, managed, and maintained by IT now serve as the engines of the enterprise. The time required for IT to deliver solutions and the long-term structural cost have become crucial considerations in whether or not to go forward with new business initiatives. As companies focus on core competencies and outsource non-differentiating operations, even standard processes, like order management, suddenly become more complex, since they now cross enterprise boundaries. Transforming core business processes to create strategic advantage increasingly involves people, processes, and information across multiple organizations and systems.

IT organizations must enable these new business initiatives by finding a cost-effective way to leverage and extend existing systems to support new business needs. This means that traditional IT architectures that supported process automation in a departmental context must evolve. They must seamlessly integrate processes and information from disparate applications regardless of where they reside; enable collaboration between departments, suppliers, partners and customers; and manage and adapt processes to the pace of business change.

To deliver against these requirements, IT organizations must address two major hurdles posed by the current environment:

- ❑ **Heterogeneity** – The agile IT organization must support the new generation of business requirements in an increasingly heterogeneous system landscape. Today’s landscapes are dominated by a mix of packaged enterprise suites, best-of-breed applications, and legacy systems. To complicate matters, most large companies have gone through multiple acquisitions and divestitures, leaving their IT infrastructure a web of disparate, stand-alone systems carried over from these business changes. As a result, implementing even simple business processes means cobbling together processes spanning different organizational units, different systems, or even external service providers. Over time, the cost of maintaining and adapting these processes can increase exponentially, as business structures and customer needs change.

- ❑ **Applications designed for efficiency, not reuse** – Traditional application architectures that form the business backbone of today’s enterprises were designed as high-performance transactional engines driving operational efficiencies. Since they were built for efficiency around a tightly integrated set of functions, they often do not have a clear distinction between user interface, business logic, and data. Modifying these systems to support adding new channel partners, introducing new products or services, or targeting new customer segments can result in unforeseen costs and complexity. While the integrated nature of these applications serves their intended uses well, IT organizations must find a way to transform these tightly integrated business engines into the flexible, reusable building blocks of future systems. Reusability is the key to increasing productivity and flexibility while reducing TCO.

For IT organizations to enable business change, they must find a cost-effective blueprint to evolve existing architectures towards greater flexibility across heterogeneous landscapes.

NO SILVER BULLETS

In the mid-1990s, these constraints gave rise to (EAI, which attempted to stitch together business scenarios using specific application-to-application (“A2A”) interfaces designed for performance and reliability. But EAI has not produced an integration architecture that is cost-effective in the long run, and it has proven to have its own problems. More recently, Web services have held out great promise, but their true power remains to be tapped.

EAI Only Provides a Partial Answer

While EAI tools can successfully link individual applications, they require that programmers understand the inner workings of both sides, which creates a **tightly coupled** integration. Programmers then have to maintain these links over the useful life of the applications. Creating and maintaining these hard-wired links

is expensive and resource intensive. Every process change triggers expensive and complex programming and testing. Further, reusing functionality developed in these distributed environments becomes very difficult. EAI too often becomes a high-cost path, with companies spending five times as much in services and support as they do on EAI tools. This complexity and cost stands in the way of IT-enabling business change.

Any answer to the challenges of enabling flexible processes spanning heterogeneous landscapes must emphasize long-term adaptability and cost reduction. And it must leverage the same infrastructure, whether the integration scenarios are between applications within a department, across enterprise boundaries, or some combination of the above.

WEB SERVICES BEGIN TO ADDRESS IT CHALLENGES

Address Issues of Heterogeneity and Reuse

A **Web service** represents a self-contained, self-describing piece of application functionality that can be found and accessed by other applications using open Web standards. A Web service is **self-contained**, because the application using the Web service does not have to depend on anything other than the service itself, and **self-describing**, because all the information on how to use the service can be obtained from the service itself. The descriptions are centrally stored, and accessible through Web standards-based mechanisms to all applications that would like to invoke the service. In summary, Web services answer the need for a standardized and vendor-agnostic way to cope with heterogeneity and to create interoperability and compatibility among various applications:

- ❑ Instead of requiring programmers to establish and maintain links between applications, Web services are **loosely coupled**, making connections simpler and more flexible and allowing application architects to more easily find and understand services offered by other cooperative applications. Applications can access a Web service across a network using mechanisms based on Web standards.
- ❑ Web services provide a standards-based way for an application to expose granular functionality such as “delete order,” which would remove an order from one particular system. To do this, Web services make use of highly standardized interfaces, which create an abstract layer that conceals how the underlying functionality is implemented.

Web services begin to answer the challenges faced by IT organizations:

- ❑ **Heterogeneity** – The ability to communicate with other applications using standards-based mechanisms simplifies connectivity across heterogeneous landscapes. Web services can easily be discovered for use by applications that require them, as standardized catalogs of services are developed. Similarly, the abstraction of functionality provided by Web services is very useful in heterogeneous environments, since it serves to conceal the differences between systems on technical level. Since Web services hide implementation details of the underlying applications, a developer “using” Web services to build a new solution needs no knowledge of the structure of the applications that deliver the service. By the same token, he can feel secure that his solution using the Web service will not be impacted if the underlying applications change.
- ❑ **Reuse** – Web services also provide preliminary answers to the issues of flexibility and reuse. Changes can be made in the underlying implementation or in the program calling the Web service, as long as the behavior of the Web service stays the same. This provides the basis for combining and reusing Web services without creating a spaghetti-like maze of unmanaged complexity.

Enhancement Needed to Support Enterprise Scenarios

Today, Web services are largely being used to expose functionality delivered by single applications. An example could be a Web service to check inventory for a particular part or item, a task that might be a single step in a larger process.

However, Web services are too granular to be efficient building blocks for enterprise business scenarios. The simple process of canceling an order illustrates this point. From a business perspective, the directive to cancel an order encompasses several cross-functional and cross-application activities, including sending a confirmation to the customer, removing the order from the production plan, release materials allocated to the order, notifying the invoicing department, and change the order status to “inactive” or deleting it from various systems. Each of these activities may be a Web service offered from different systems. The ability to build a complex end-to-end scenario to cancel an order would be a very powerful enterprise-level business service. Clearly, efficiently developing new business scenarios leveraging existing applications calls for business-level building blocks that aggregate the benefits of multiple web services. The true power of Web services will be only unlocked when enterprise business scenarios can be flexibly composed from services that are defined at a business level and combine functionality across multiple applications.

Enterprise Services Architecture Extends Web Services Benefits

Realizing this need, SAP has defined **Enterprise Services Architecture**. Enterprise Services Architecture takes Web services standards and services-oriented architecture principles and extends these to meet the needs of enterprise business solutions. It helps IT organizations leverage existing systems to build and deploy flexible solutions that support end-to-end business scenarios across heterogeneous landscapes. Enterprise Services Architecture addresses the business issue that most companies are facing – extending existing IT assets to support business change and innovation, while lowering total cost of ownership.

WHAT IS ENTERPRISE SERVICES ARCHITECTURE?

Enterprise Services Architecture is SAP’s open architecture for adaptive business solutions. The fundamental premise of Enterprise Services Architecture is the abstraction of business activities or events, modeled as **enterprise services**, from the actual functionality of enterprise applications. Aggregating Web services into business-level enterprise services provides more meaningful building blocks for the task of automating enterprise-scale business scenarios. Enterprise services allow IT organizations to efficiently develop **composite applications**, defined as applications that compose functionality and information from existing systems to support new business processes or scenarios. All enterprise services communicate using Web services standards, can be described in a central repository, and are created and managed by tools provided by SAP NetWeaver. The order-to-cash business scenario illustrates the benefits of Enterprise Services Architecture.

Order-to-Cash Without Enterprise Services Architecture

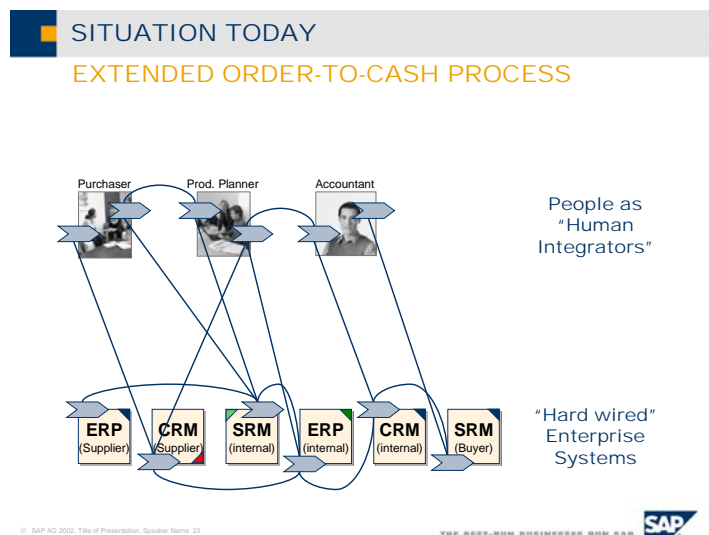


Figure 1: Order-to-Cash Scenario without Enterprise Services Architecture

As the figure shows, order-to-cash involves multiple applications – customer-facing applications (CRM), supplier-facing applications (SCM), and ERP systems where the order resides, and where all transactions and fulfillment entities are stored. In the typical order-to-cash scenario, employees act as human integrators, sitting in front of many different applications, transferring information from one to the next by copying and pasting and retyping information, making process flow decisions as needed. The applications, when they are communicating, are hard-wired through brittle connections that are expensive to maintain.

Order-to-Cash with Enterprise Services Architecture

With Enterprise Services Architecture, a composite application can use enterprise services to automate the flow of information from application to application. Each user in the business scenario has a role-based interface that provides exactly the information and functionality required to meet their goals. The process is defined, controlled, implemented, and managed at a business level, with SAP NetWeaver providing the environment to construct enterprise services to control the flow of information from one enterprise application to the next. The enterprise applications continue to support the traditional transactional scenarios that are departmental in scope, while participating in end-to-end business scenarios.

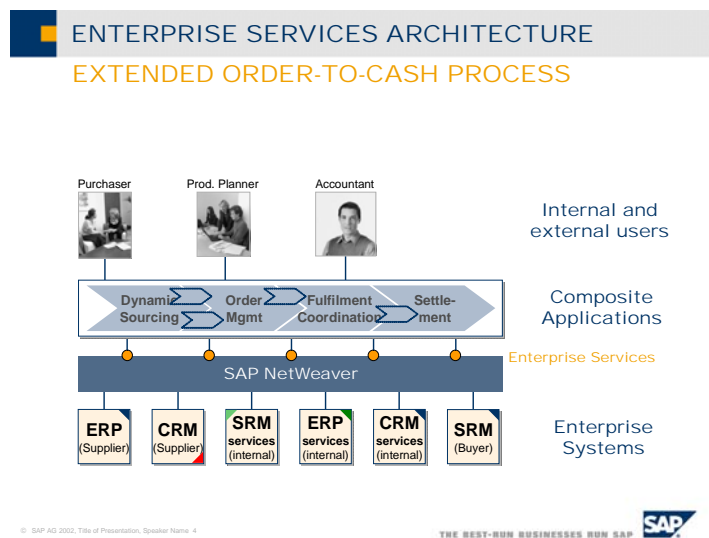


Figure 2: Order-to-Cash with Enterprise Services Architecture

Key Characteristics of Enterprise Services Architecture

The key characteristics of Enterprise Services Architecture listed are crucial to implementing business scenarios like order-to-cash:

- ❑ **Enterprise Services Architecture extends the benefits of Web services to enterprise business scenarios**, by aggregating existing systems functionality into business-level enterprise services.
- ❑ **Enterprise Services are modeled from an “outside-in” perspective.** While the core set of enterprise services identified by a company may be substantially enabled by legacy or enterprise applications (including those from SAP), they are not defined or constrained by SAP or any other vendor’s applications. In other words, Enterprise Services Architecture defines or models services “outside-in” for any application i.e. based on business events relevant to enterprise business processes, not necessarily on an existing application or implementation. SAP will evolve its applications to support enterprise services designed for each business domain or industry that it currently addresses.
- ❑ **Enterprise Services Architecture offers a blueprint for enterprise-wide business process evolution, with complete investment protection.** Enterprise Services Architecture is a blueprint for a customer’s entire IT landscape, as well as an application architecture for SAP. Leveraging its breadth

and depth of industry knowledge, SAP is defining a catalog of enterprise services that promote reuse of functionality across SAP® solutions and third-party solutions. SAP expects that customers will independently select a core set of enterprise services needed to support their key business scenarios. They will then match these with enterprise services available from SAP, develop custom enterprise services where needed, and build flexible business scenarios by leveraging existing IT systems.

- ❑ **Enabled by SAP NetWeaver, Enterprise Services Architecture offers a gradual path to flexible, service-centric system landscapes.** Enterprise Services Architecture allows for gradual and non-disruptive transition of existing applications and architecture to higher levels of flexibility and value.
- ❑ **Enterprise Services Architecture allows new business processes to be developed, deployed and changed independent of existing applications.** “Consumers” of enterprise services are isolated from changes in applications that “provide” the service. Enterprise Services Architecture leverage an abstraction layer between the way an enterprise service is used, and the way the corresponding functionality is implemented within an enterprise application. This abstraction allows composite applications or custom UIs using the service, or its so-called “consumers,” to be decoupled from the applications “providing” the service. As a result of the decoupling, IT can leverage the rich functionality and best practices of enterprise applications to support new, innovative business solutions, and yet evolve these solutions independently of changes in the underlying applications.

ENABLING ENTERPRISE SERVICES ARCHITECTURE WITH SAP NETWEAVER

SAP has designed SAP NetWeaver to support the integration and application platform needs of enterprise architectures, while enabling standards-based interoperability with other platforms that may be part of the landscape. SAP NetWeaver enables the **development, deployment, and administration** of enterprise services. SAP NetWeaver can help customers service-enable their system landscapes by supporting communication based on Web services standards. With SAP NetWeaver, customers can leverage services that are part of enterprise suites like mySAP™ Business Suite, as well as create custom services as needed.

In addition to providing tools for creating and managing enterprise services, SAP NetWeaver also supports the design, implementation and execution of applications that use those services, like composite applications and custom UIs. SAP has developed a framework for quick and easy assembly of composite applications in a repeatable fashion. SAP NetWeaver provides a unified application development platform that contains the tools, methodologies, rules, user interface patterns, and services that allow SAP, its partners and customers to build composite applications, either as products for sale or custom applications for use by one company.

What Enterprise Services Architecture Means for Customers

With Enterprise Services Architecture, the traditional challenges of system landscapes no longer stand in the way of IT support for business initiatives. Enabled by SAP NetWeaver, Enterprise Services Architecture helps customers manage heterogeneity of systems landscapes. This allows the underlying application components that supply the functionality represented by the enterprise service to be altered, without impacting applications “using” the service. Further, Enterprise Services Architecture enables customers to see their existing systems as not merely as powerful business engines for today, but also as the building blocks of future business processes.

Enterprise Services Architecture intrinsically supports an incremental development process. Its use of abstraction makes it relatively easy to combine and recombine functionality from different applications as needed, and without having to pull solutions apart and start over. This quality is mirrored in SAP NetWeaver, which supports an upgrade path that begins with a portal interface for Web services, and leads to an evolving array of services and composite applications of ever-greater complexity and scope.

Whether called upon to free up capital by reducing TCO, enable business change or support innovation, IT organizations can now leverage and extend IT assets towards these goals.

LEVERAGE EXISTING IT ASSETS

With Enterprise Services Architecture and SAP NetWeaver, customers can create predictable cost structures by consolidating their IT landscapes to reduce TCO. From a one-time investment in service enablement and SAP NetWeaver deployment, they can get repeat benefits in easily being able to build and deliver a variety of solutions. User productivity solutions are one such category that represents immediate return, helping IT organizations drive significant incremental value out of existing IT assets.

Reducing TCO through consolidation: Leveraging integrated business solutions powered by SAP NetWeaver, SAP customers can reduce TCO and create a more sustainable IT cost structure. At the same time, they can introduce a Web services-based foundation for the future. The same infrastructure can be leveraged for both A2A as well as B2B solutions, making the borders of the enterprise transparent.

Enhancing user productivity: Enterprise services allow interaction logic to be abstracted from the underlying business logic and applications, allowing IT organizations to choose the appropriate interface and technology for each type of user. This allows user productivity to be enhanced in a number of ways, including the following:

- Role-specific portals enriched by business intelligence can be developed in any technology, meeting the needs of professional as well as occasional users. Wiring together heterogeneous data and logic from existing systems like SAP R/3®, these portals provide information, alerts and monitoring tailored to the users' business needs.

- Interactive forms and other types of desktop technologies allow users to work with familiar user interfaces in occasionally connected (on-line/off-line) scenarios.

- Self-service components (e.g., customer self-service to complement call center operations) can be built to extend existing applications, and deployed both within and outside the enterprise.

PRODUCT CHANGE REQUEST SCENARIO – ENHANCING USER PRODUCTIVITY WITH ENTERPRISE SERVICES ARCHITECTURE

The Product Change Request (PCR) process provides an illustration of how Enterprise Services Architecture can transform a business scenario from a challenge to a strategic advantage, and significantly enhance user productivity. The PCR process is initiated in manufacturing firms when an important part needs to be changed because either the design, or the materials used to build it have changed. Sometimes PCRs result from the needs of a particular customer or from regulatory requirements. In all cases, a PCR means that the entire manufacturing process must be examined to make sure that the change will not adversely affect the product.

It is important to be able to audit this process as fully as possible, to ensure that all required reviews are obtained and errors prevented. The standard operating procedure for PCRs at most firms is paper-based. Each reviewer examines the PCR document and gathers information from the systems of record that they rely on, entering and reentering data as needed. The result is a process that moves along slowly with little visibility or automation. But in the end, the required approvals are all obtained.

In the next version of mySAP ERP, PCR will be implemented as a composite application based on enterprise services. The paper request will be replaced with a set of interactive forms that allow users to work on entering the required information off line, but then can be submitted and automatically transferred into underlying systems. Much of the information in the forms is automatically populated, avoiding costly and error prone entry and reentry of data. Enterprise services move the data back and forth from the many different systems needed to populate one form. All of the information moves intact from one approver to the next.

The foundation of enterprise services takes the PCR process beyond the simple automation of a paper process. Decoupling the end-user's view, the forms, from the underlying systems provides unprecedented flexibility. Fields can be added to forms, and the process can be changed on the fly. Using the Composite Application Framework within SAP NetWeaver, an end-user can decide that an additional review by an engineering expert must take place and add a process step to obtain that review on the fly. Information from legacy systems can be incorporated into the process through Web services. In a recent implementation, automation of this process reduced the time for a PCR to a mere 5% of what it had been.

EXTEND IT ASSETS TO DELIVER NEW BUSINESS VALUE

With Enterprise Services Architecture, customers can build new applications by extending existing systems, supporting business needs while maintaining investment protection. Enterprise services provide efficient, reusable, business-level building blocks with which to compose industry-specific scenarios delivered as composite applications. Rather than being forced to follow the traditional cycle of “rip and replace,” composite applications will free customers to extend the lifespan of their legacy applications, and mix and match underlying components from SAP and other vendors.

Composite applications not only enable new scenarios, but also benefit traditional processes such as invoice verification. While the invoice verification process is fairly structured, it spans multiple departments across the entire enterprise – procurement, accounting, and logistics – often relying on different enterprise applications, third-party products, legacy systems, or even external services. Invoice verification will often need to be reconfigured to adapt to business change or systems evolution, and therefore, implementing invoice verification as a composite application would deliver obvious benefits.

Composite applications can be built, deployed and adapted more quickly and cost-effectively than traditional software. First, a composite application can be built with no knowledge of the architecture, interfaces and development platforms of the underlying applications that deliver the functionality, since enterprise services are a standards-based abstraction. Second, a composite applications that “use” an enterprise service are now isolated from changes in the underlying applications and systems. In the past, when an individual piece of application functionality was changed, all interfaces and applications that touched this component would have to be

INTEGRATED EXPLORATION AND PRODUCTION

Asset maintenance for the oil industry is equivalent in stature to cash management for banks. Oil companies own assets whose value stretches into the trillions of dollars. Many of the assets, like equipment on oil rigs or at refineries are remote, complex or both. Downtime of even a few hours can cost millions.

The process of asset maintenance in the oil industry involves many specialists and experts. When the progress of drilling slows on a rig or a glitch appears in the refining process, there could be hundreds of reasons. Under current practices the field engineer assembles the evidence and calls in experts to help. Crucial information about part availability, supplier contracts, budgets and so forth are spread across different systems of record.

The current process relies on best efforts by all involved, based on a sporadic unmanaged flow of information between different participants in spreadsheets, documents, email and presentations. Attempts at automating this process have run straight into a mountain of integration problems.

Enterprise Services Architecture sorts out the integration challenges through the use of enterprise services, composite applications, and the integration capabilities of SAP NetWeaver. SAP's composite application for Integrated Exploration and Production (xIEP) implements a business scenario for asset maintenance that gives all the project participants the tools to use the real-time data, accessing the same database to collaboratively expedite a solution potentially in a matter of hours.

xIEP uses enterprise services to bring together the technical, project management, and financial information about a potential solution into an intuitive format for all participants. Intra-company and inter-company access permissions can be controlled to allow virtual teams to collaborate in real-time. One interface and one repository for information keeps the team knowledgeable about the big picture and important technical and financial details, including integration of data from third party sources. Experts can subscribe to specific points in the remediation process so their contact information is ready when needed. Management can information captured by the system to monitor the performance of everyone involved.

With data from all systems being available in real-time, more informed decisions can be made quickly. The asset maintenance process becomes transparent and the information is accessible through the Internet in one consistent interface. The ROI case is dramatic. Avoiding even one hour in certain situation can pay for the entire solution.

The implementation of SAP xIEP takes place through a portal that is designed for maximum usability and accessibility. The Guided Procedure feature of the SAP Composite Application Framework allows processes to be adjusted on the fly by participants and configured to a company's specific needs. xIEP will continue to be expanded with functionality and services to cover further business scenarios including all aspects of the upstream value chain including Exploration Planning & Execution, Field Ticket Management, Logistics Management, Production Management, Upstream Financials, Complex Procurement. and Well Planning & Delivery.

changed as well. For B2B solutions that crossed enterprise boundaries and touched systems and processes from customers and partners, the complexity increased exponentially. These barriers to new business processes are significantly reduced. Finally, composite applications can be easily deployed across heterogeneous landscapes, following the same approach both within and across enterprise boundaries.

INNOVATE

Enable Adaptive Business Solutions

Enterprise Services Architecture enables adaptive business solutions, which can be developed rapidly, deployed easily across heterogeneous landscapes and adapted quickly and efficiently, as business needs change. As SAP customers progressively adopt Enterprise Services Architecture, it becomes possible to leverage and integrate existing IT assets into innovative new business processes, while still maintaining a level of flexibility that makes future process changes cost-effective. CIOs can at last reconcile the inflexibility of current architectures with the CEO's requirement that IT support business initiatives quickly and efficiently. IT organizations can now deliver flexible solutions with a sustainable long-term cost structure without trading off efficiency and rapid time-to-market.

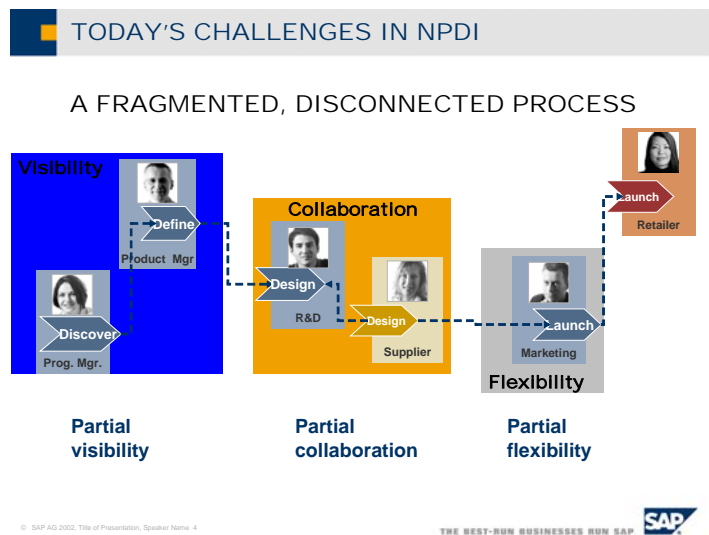


Figure 3: New Product Development Before Enterprise Services Architecture

NEW PRODUCT DEVELOPMENT AND INTRODUCTION (NPDI) – INNOVATION WITH ENTERPRISE SERVICES ARCHITECTURE

The development and launch of new products is a high stakes game in which companies bet huge sums developing ideas into prototypes and finally into products and brands for consumers. The importance of NPDI, makes it a cross-functional activity including Marketing, Research and Development (R&D), Supply Management, Manufacturing, Finance, etc. NPDI usually starts with the identification of an opportunity in the market and ends with the successful launch of the product. In between are many activities to define the requirements, develop and test a product concept, fully define and develop the product, source the suppliers involved, plan the manufacturing and supply chain and prepare marketing programs. On top of that a product strategy must be defined; a product program must be managed and all the projects needed to drive the NPDI process must be monitored.

In most markets and especially those relating to consumer products, the number of new product introductions per annum has increased dramatically. Driven by consumer demand and fuelled by advances in technology, companies have to bring more and more products to market in order to remain competitive. The companies best able to execute NPDI will clearly have an advantage as they reduce time to market and make effective use of scarce internal resources.

Many companies lack an overall process for NPDI. Instead, the new product passes through a number of organizational “silos” with no single individual or group being responsible for the outcome. Best practices for new product development involve creating a set of increasingly demanding gates that each idea must pass through to move to the next stage. Healthy processes for new product development save money by identifying problems as early as possible and using a cross functional team in all stages of the process. Assembling information from systems of record, managing flexible collaboration within the company as well as with partners in the overall value chain, adjusting the process to meet the needs of identifying and processing different sorts of ideas are all key to success.

Classic barriers to an effective new product development process include the difficulty of assembling information from existing systems, the challenge of reconciling the need for strict evaluation criteria and a flexible process, the assembly and coordination of a cross functional team, and the management of large amounts of unstructured sales, marketing, market research and product data.

Automation of NPDI process frequently is stopped cold by the disconnect between the loose collaborative systems use in research & development and the more structured transactional systems in manufacturing and finance, where much of the relevant information resides.

Enterprise Services Architecture bridges these gaps by encapsulating systems and processes in enterprise services, and then orchestrating flexible processes, collaboration, and document management using the capabilities of SAP NetWeaver.

The new Composite Application Framework built into SAP NetWeaver enables SAP to create new applications targeting cross-functional business processes. Two of SAP's new xApps (cross applications) are part of the NPDI offering: The xApp Resource and Program Management (SAP xRPM) is a solution designed to improve a multi project environment by focusing on project portfolio monitoring and management.

The xApp Product Definition (SAP xPD) is designed to address the hurdles and inefficiencies of the first phases of the NPDI process: idea and concept management, concept evaluation and product definition. It supports NPDI that is implemented with a collection of Enterprise services. Information from systems of record about costs, budgets, financial projections, and bills of materials are extracted and linked to less structured material from marketing and market research. The process of evaluation starts with a set of steps and criteria that can be modified if needed by the process owners. The result is that a process that was a mélange of presentations, spreadsheets, and documents becomes a flexible, harmonized flow of information. Bad ideas are flagged and killed early, minor changes sail through, and the amount of money spent on evaluation is focused on taking as much risk out of the process as early as possible. When it comes time for development of a prototype or limited production runs, the information flows back into the systems of through the Enterprise services.

INNOVATING NPDI WITH xPD and xRPM

END TO END, INTEGRATED AUTOMATION



Figure 4: New Product Development Before Enterprise Services Architecture

Enterprise Services Architecture will drive a virtuous circle – once core applications are abstracted into reusable components by a layer of enterprise services, creating composite applications will carry less risk and take less time, leading to new services which also become reusable throughout the enterprise. As early composite applications become the building blocks for their more complex successors, they create the potential for modeling business scenarios without the long development costs of previous generations of software. New categories of applications that leverage services-oriented architectures, business intelligence and collaboration become possible, once SAP NetWeaver is deployed as an integration platform in addition to a next-generation application platform.

Now instead of just enabling efficiency, IT can now become an enabler of change. Solution implementation then falls away as a barrier to change, enabling innovation.

Enable Flexible Organizational Structures

Enterprise Services Architecture creates a clear hierarchy of components that provide the building blocks (ERP/GL/HR) from the high-value composite applications. Since it no longer matters where the applications physically reside, more flexible deployment models become possible. This enables off shoring and outsourcing of any components that are found to be non-differentiating, allowing a more rational and cost-efficient organization of IT inside and outside of the enterprise.

ENTERPRISE SERVICES ARCHITECTURE CHANGES HOW SOLUTIONS ARE BUILT

“Companies whose competitive edge is defined by their ability to quickly deliver new IT systems to meet the businesses needs will need to begin evaluating, planning for and migrating development staff to at least one of the two alternative and more efficient forms of development over the next 18 to 24 months or risk losing their competitive edge to their competition.”

– “Beyond Code-centric Development,” Forrester Research, 2003

Enterprise Services Architecture changes the way solutions are built and deployed, for both SAP and its customers. The current development approaches involve creating hard-wired one-to-one integrations and heavily customizing elements of core applications like ERP and CRM to meet the requirements of new business scenarios. This approach comes with significant drawbacks – it requires one-off integration efforts to bring together underlying applications and systems; it calls for teams of business analysts and highly skilled technical resources who must translate business requirements to the language of IT; it is expensive and time-consuming, requiring that IT organizations traversing the entire development lifecycle from requirements definition to roll-out; it is costly to maintain and change, driving up costs in the long run.

With Enterprise Services Architecture, enabled by SAP NetWeaver, the language of business becomes the language of IT. Enterprise services are defined at a granularity where they can be understood by business analysts, rather than requiring a developer’s intervention. With the model-driven approach of Enterprise Services Architecture, and SAP NetWeaver tools that support the entire lifecycle from business process modeling to code generation, a business analyst can “assemble” enterprise services into composite applications that enable new business scenarios. This approach is clearly more efficient, and helps customers build, deploy and maintain solutions with significantly greater agility, cost efficiency and speed.

This allows composite applications supporting business scenarios to be built rapidly, with a minimum of coding. However, this approach is distinguished from older, rapid development paradigms in that it is not merely about creating new modules of code quickly. It is about modeling business processes that can then be assembled by wiring together existing data and business logic across a heterogeneous landscape.

As SAP’s own development processes evolve from a traditional development paradigm towards solution assembly, customers can also expect to see a shift in the speed in which SAP delivers new solutions.

SAP ROADMAP FOR ENTERPRISE SERVICES ARCHITECTURE

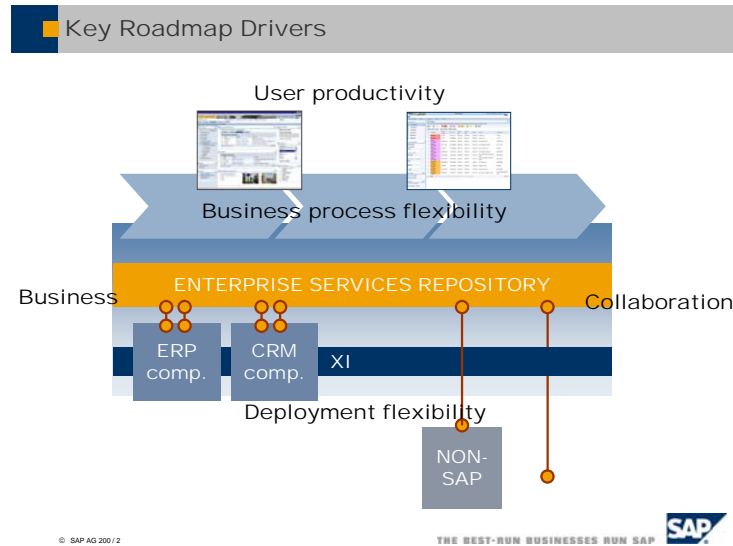


Figure 5: Key Drivers for Enterprise Services Architecture Roadmap

SAP NetWeaver already enables enterprise services, as well as applications leveraging these services, and will continue to enhance support for these. SAP is progressively evolving its business applications such as mySAP Business Suite towards Enterprise Services Architecture, increasingly abstracting their rich functionality and best practices into services that will support the needs of future IT landscapes. The SAP roadmap to Enterprise Services Architecture will be driven by customer value, and will occur in a phased fashion, as discussed below:

- ❑ **2004:** Service-enablement efforts for this year are focused on enhancing user productivity and collaboration. Service-enabled scenarios that will be available as part of mySAP Business Suite include the Product Change Request scenario discussed earlier. In addition, SAP is also carrying out a systematic inventory and prioritization of enterprise services, based on the enterprise business scenarios featured in the new industry solution maps.
- ❑ **2005:** In 2005, a full inventory of enterprise services will be available for planning by partners and customers. An **Enterprise Services Repository** will be part of SAP NetWeaver '05. All major cross-industry scenarios will be service-enabled, with specific emphasis on the business user and increased flexibility of processes.
- ❑ **2006:** All relevant enterprise services will be actively available from the repository for use by select partners and customers. In addition, 75% of mySAP Business Suite scenarios will leverage enterprise services for business and deployment flexibility.
- ❑ **2007:** All mySAP Business Suite solutions will be fully compliant with Enterprise Services Architecture.

Conclusion

Enterprise Services Architecture is a flexible set of rules like a grammar. Learning how to speak the language of Enterprise Services Architecture is a gradual process. The alphabet of Web services must first be assembled into the words of enterprise services, and then from these, sentences form. To use the grammar of Enterprise Services Architecture effectively, an enterprise must know what needs to be said – a business scenario that demands implementation, and will deliver business value. Then, even the simplest “sentences” can deliver tangible results and lower TCO.

Enterprise Services Architecture gives IT organizations the key to enabling business agility. With Enterprise Services Architecture, IT organizations can make the leap from driving operational efficiencies to delivering on the promise of the adaptive, real-time enterprise.

© Copyright 2004 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® are registered trademarks of Microsoft Corporation.

IBM®, DB2®, DB2 Universal Database, OS/2®, Parallel Sysplex®, MVS/ESA, AIX®, S/390®, AS/400®, OS/390®, OS/400®, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere®, Netfinity®, Tivoli®, Informix and Informix® Dynamic Server™ are trademarks of IBM Corporation in USA and/or other countries.

ORACLE® is a registered trademark of ORACLE Corporation.

UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of the Open Group.

Citrix®, the Citrix logo, ICA®, Program Neighborhood®, MetaFrame®, WinFrame®, VideoFrame®, MultiWin® and other Citrix product names referenced herein are trademarks of Citrix Systems, Inc.

HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA® is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.