SAP White Paper

mySAP™ TECHNOLOGY

SAP® WEB APPLICATION SERVER: BUILDING RELIABLE BUSINESS APPLICATIONS

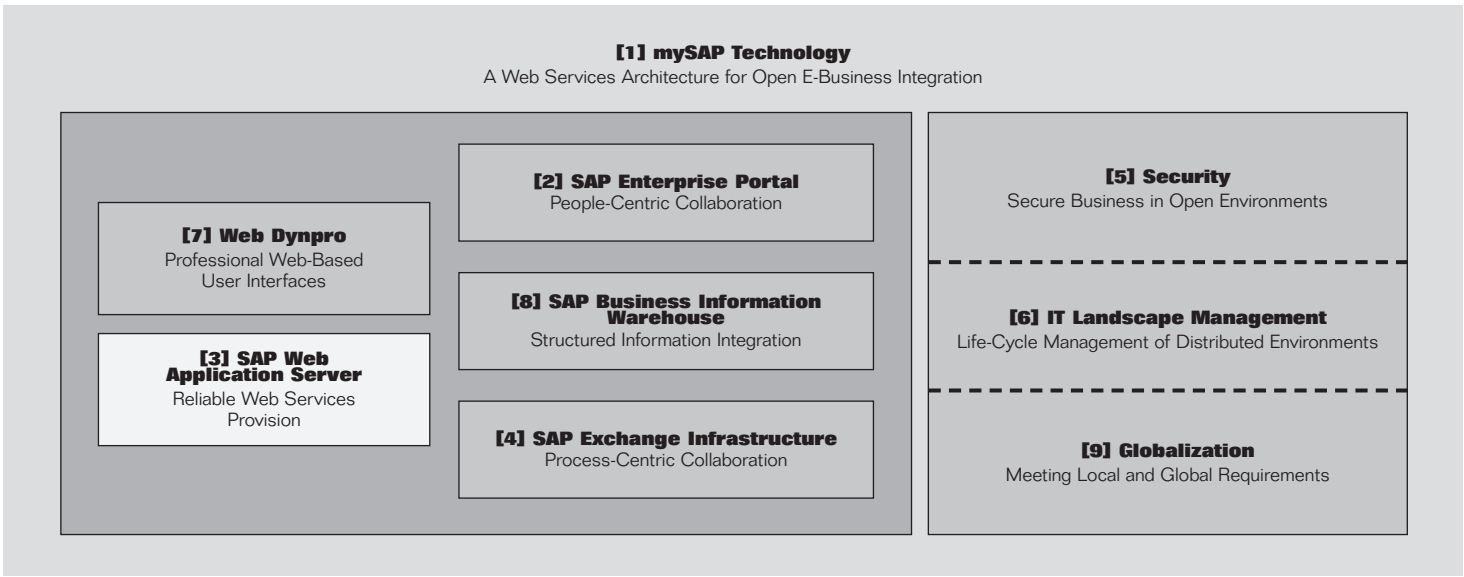
CONTENTS

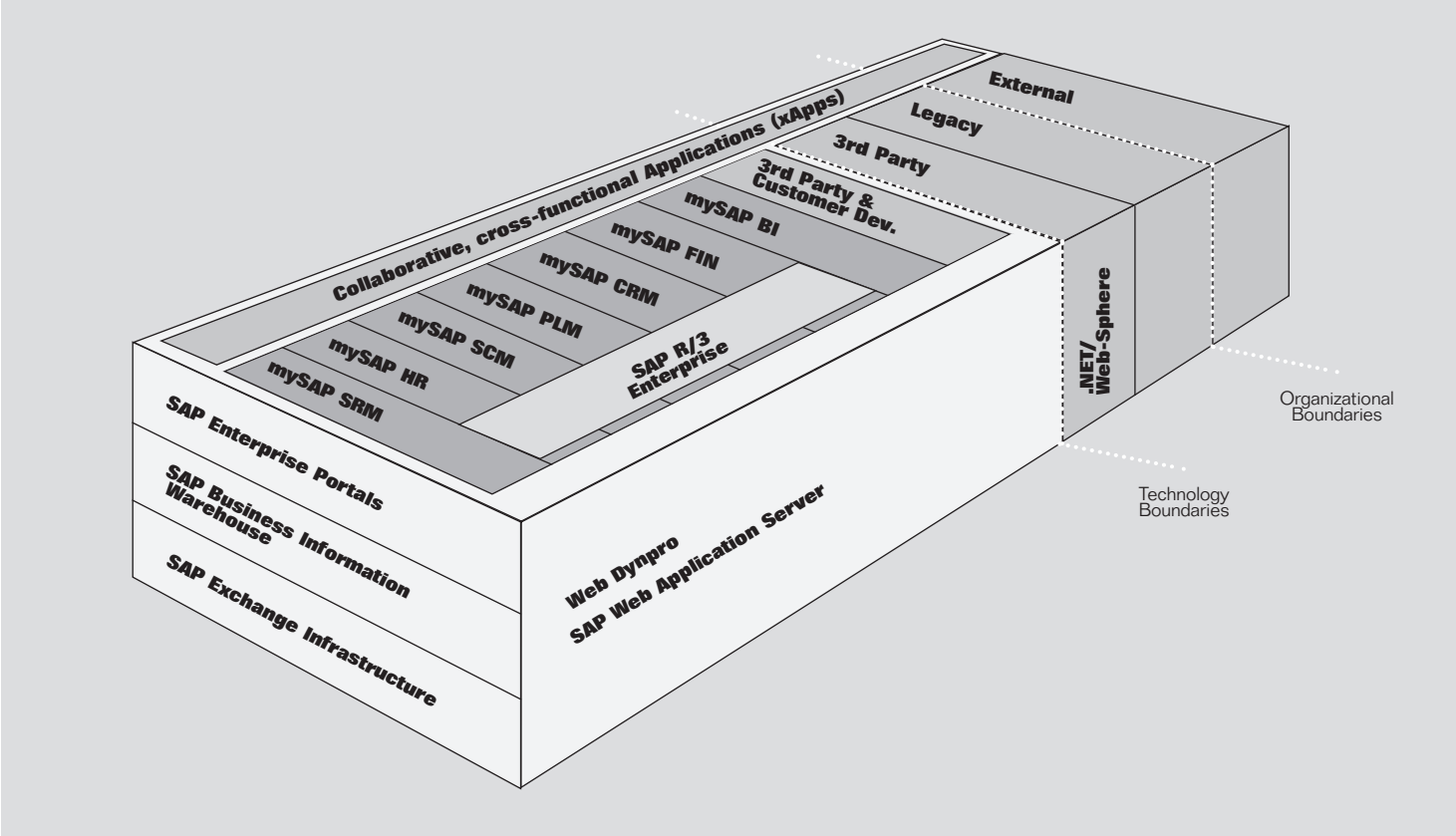
mySAP™ TECHNOLOGY FOR OPEN E-BUSINESS INTEGRATION

To gain a sustainable competitive advantage, companies need to drive collaborative business. And to implement collaborative business, companies need to integrate their existing heterogeneous IT landscapes and extend this integration to their business partners, customers, and suppliers. This integration will then create an open platform that provides existing applications as Web services and enables cross-applications – a new type of application that combines existing and new Web services that cross functional, technical, and organizational boundaries. As companies implement collaborative business, mySAP™ Technology enables them to manage heterogeneity and powers collaborative solutions. It does so for several reasons:

- mySAP Technology is a **native Web** infrastructure based on **open standards** for reliable e-business solutions. It is designed to operate in heterogeneous environments, integrating any application from any vendor based on any technology.
- mySAP Technology powers enterprise, cross-boundary applications, and collaborative business processes based on **one common infrastructure** for integration within and beyond company boundaries.
- mySAP Technology features **syndicated Web service**s, allowing for the integration of people, processes, and information based on **shared collaboration knowledge**.

And mySAP Technology does all this while protecting existing investments and achieving the lowest possible cost of ownership. mySAP Technology is the foundation of SAP® R/3® Enterprise, all mySAP.com® solutions, and cross-applications. With mySAP Technology, SAP solves the integration challenge from a business perspective.





The key building blocks of mySAP Technology are:

- **SAP® Enterprise Portal:** Integrating people and unstructured information to empower individuals
- **SAP® Business Information Warehouse:** Integrating structured information to make smarter decisions
- **SAP® Exchange Infrastructure:** Integrating processes to drive end-to-end collaborative business processes
- **SAP® Web Application Server:** A reliable infrastructure for portals, warehouses, exchanges, and all application components, providing Web services in Java 2 Platform, Enterprise Edition (J2EE), and ABAP through open standards
- **Web Dynpro:** To design and operate powerful presentation logic for professional, highly interactive user interfaces
- **Infrastructure services:** Powerful and integrated services, including security, IT landscape management, and globalization

This white paper is part of a series of nine white papers explaining the architecture and vision of mySAP Technology. This white paper explains the Web application technology aspects of mySAP Technology on SAP Web Application Server. SAP Web Application Server is used to develop and deploy Web-based application components, providing Web services based on open standards. It is also the foundation of all mySAP.com components. The figure on the previous page lists all nine white papers.

EXECUTIVE SUMMARY

SAP® Web Application Server is a scalable and reliable component platform that supports the development and operation of Web applications and Web services based on Java 2 Platform, Enterprise Edition (J2EE), and ABAP™. SAP Web Application Server embraces native Web technologies while providing all the benefits of what has previously been referred to as SAP® Basis – including the benefits of the proven and scalable data, system, and software management capabilities that SAP is known for.

SAP Web Application Server provides Web application technologies for professional, mission-critical environments. It works with any technical infrastructure – from mainframes to small servers and all major databases – and it supports user access from all devices – from Web browsers to mobile devices – to match all company needs. SAP's participation in industry standards groups helps ensure that the technology underlying SAP Web Application Server continues to evolve as a stable and reliable platform for business solutions.

SAP Web Application Server is the infrastructure for applications that provide Web services to other applications and to SAP® Exchange Infrastructure, as well as for native Web user interfaces on any device and for integration with SAP® Enterprise Portal. SAP Web Application Server is the infrastructure for current and future mySAP.com e-business solutions and SAP R/3 Enterprise, as well as for any J2EE-based application. It also serves as the foundation of the new breed of SAP cross-functional business applications called cross-applications (xApps). With SAP Web Application Server, companies have the choice and flexibility to extend their solutions according to their available development skills and technical requirements while supporting all SAP applications.

SAP Web Application Server provides a unique combination of features that are mandatory for operating mission-critical e-business solutions: reliability and scalability, openness, and low cost of ownership.

- **Reliability and scalability**
  The proven technology of SAP Web Application Server provides best-in-class reliability and scalability that are achieved through sophisticated load-balancing mechanisms for both user access and database access. Effective memory management, sophisticated caching, highly scalable and business-optimized lock management, and multiple distributed processes provide unparalleled scalability and reliability.
- **Openness**
  SAP Web Application Server fully embraces all leading open Internet technology standards and Web services standards. It natively supports Hypertext Transfer Protocol (HTTP) and its secure version, HTTPS, as well as Internet document standards such as Hypertext Markup Language (HTML) and Extensible Markup Language (XML). SAP Web Application Server is fully J2EE compliant. It also supports the Web services standards SOAP, Web Services Description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI).
- **Low cost of ownership**
  SAP Web Application Server provides comprehensive management for all aspects of the software life cycle. Sophisticated tools for all development, deployment, continuous operation, and ongoing updates to the system landscape help to improve the manageability and reduce the total cost of ownership. Capabilities for change management, monitoring, and administration support controlled deployment and the modification of business-critical applications, enabling nonstop business operation.

In all these ways, SAP Web Application Server supports the development and operation of J2EE-based and ABAP-based Web applications and Web services for professional environments, making the benefits of its proven and reliable infrastructure available to the open technology world. The ability to use SAP Web Application Server independently from mySAP.com solutions provides unlimited opportunities to set up environments that best suit a company's needs.

SAP WEB APPLICATION SERVER ARCHITECTURE

SAP Web Application Server embraces native Web technologies while providing all the benefits of SAP's knowledge and experience through an evolutionary, standards-based approach. Companies need a professional infrastructure for developing, deploying, and executing Web applications and Web services. With SAP Web Application Server, SAP delivers a homogeneous infrastructure for J2EE-based and ABAP-based applications.

SAP Web Application Server operates all current and future components of the mySAP.com platform, SAP R/3 Enterprise, and any J2EE-based application – either third party or custom developed – as well as SAP Exchange Infrastructure, SAP® Business Information Warehouse (SAP® BW), and SAP Enterprise Portal. Companies have the choice and flexibility to extend their solutions in keeping with their available development skills and technical constraints. All existing business objects and interfaces can be used with both the J2EE environment and the ABAP environment. This approach gives companies a single infrastructure that takes best advantage of both environments.

SAP Web Application Server provides open interfaces to the market-leading Java integrated development environment (IDE) for the development of presentation and business logic. In addition, SAP offers Web Dynpro as an integral part of SAP Web Application Server for designing and implementing professional browser-based interfaces. Web Dynpro provides an easy-to-use methodology to build presentation logic – from initial modeling and design to the completed user interface. More information on Web Dynpro is available in the white paper called *Web Dynpro: Professional Web-Based User Interfaces.*

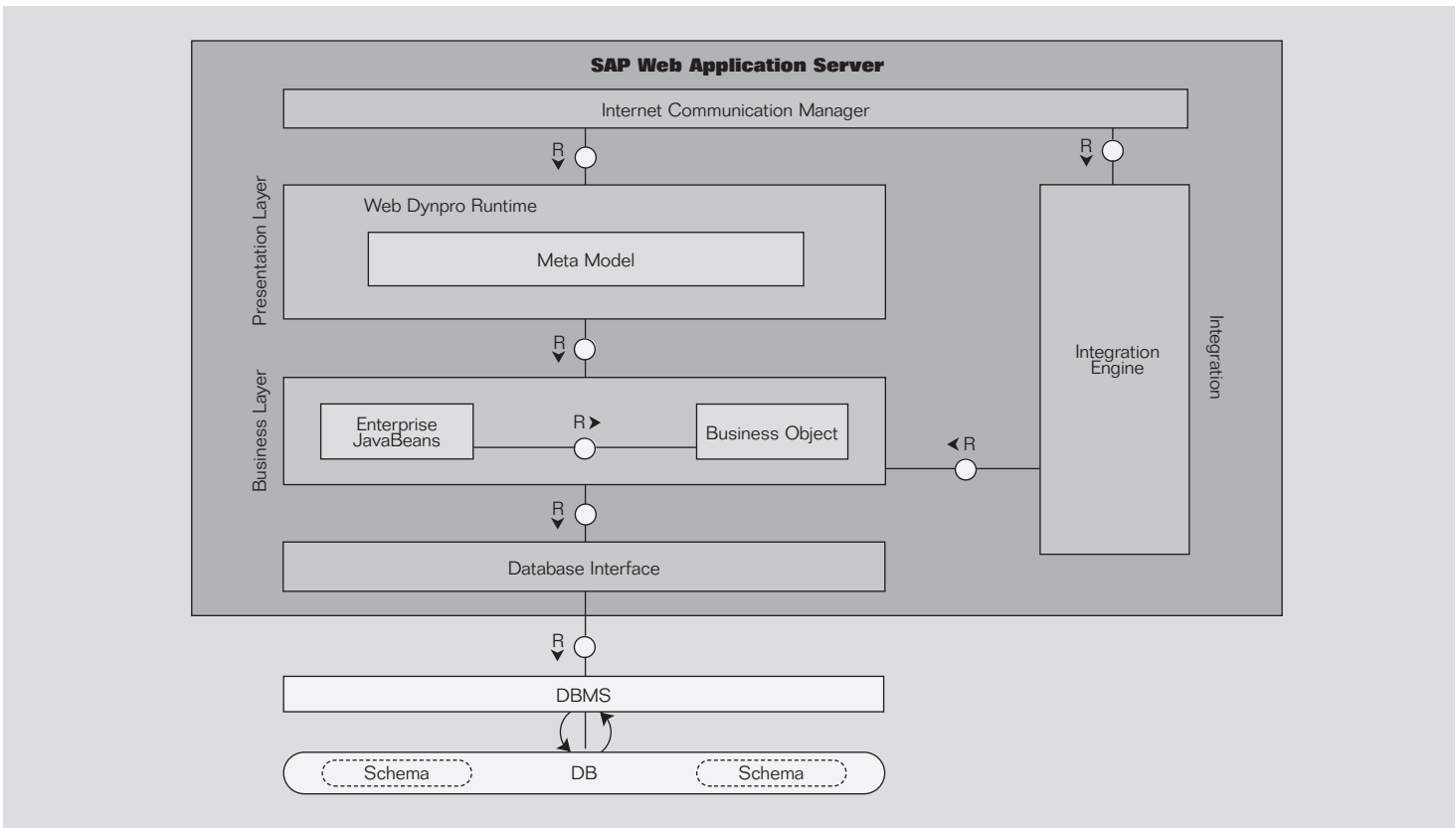


Figure 1: Overview of the SAP Web Application Server Architectures

SAP Web Application Server minimizes the effort required for change management, administration, and monitoring, thus reducing the cost of ownership. The homogeneous infrastructure of SAP Web Application Server ensures centralized and unified installation, configuration, and maintenance for J2EE and ABAP environments.

To provide the highest level of flexibility, the functional architecture of SAP Web Application Server contains three separate layers: the presentation layer, the business layer, and the integration layer. The technical architecture of SAP Web Application Server consists of two shared facilities for high scalability, reliability, and platform independence: the Internet Communication Manager (ICM) (which provides a single framework for connectivity using a variety of communication protocols), and the persistence level (which supports database independence and scalable transaction handling).

INTERNET COMMUNICATION MANAGER

The Internet Communication Manager dispatches user-interface requests (such as HTTP requests) to the presentation layer and sends Web service requests (such as XML and SOAP messages) to the integration engine.

The ICM can act as a server or a client for external communication partners. The ICM acts as a server when an application – or a client, such as a Web browser – initiates communication by connecting to the ICM. The ICM acts as a client when SAP Web Application Server initiates the communication.

An HTTP request can contain a session identifier, which also needs to be extracted from the request and passed to the relevant environment. With this session identifier, the requested environment can assign the correct session and user context.

The core ICM provides a generic framework for handling I/O independently of the protocol. Protocol-specific behavior is implemented in separate, dynamically loadable modules. Currently, modules are available for HTTP, HTTPS, Simple Mail Transport Protocol (SMTP), SOAP, and Fast Common Gateway Interface (FastCGI).

The ICM's caching technology complements the proven database caching of SAP Web Application Server. The ICM includes sophisticated HTTP request caching that significantly enhances the performance and scalability of Web applications and Web services by performing static, dynamic, and active content caching. Caching reduces the server resources necessary to build pages and eliminates the need to repeatedly execute frequently requested pages. High performance is achieved by a multi-threaded architecture that supports concurrent read and write access. And a patented indexing algorithm especially suited for long, URL-style cache keys provides very fast access to the cache directory.

Unlike conventional Web caching, which only supports caching of static content like images, the ICM can handle Web pages with dynamic content. This is an essential capability in an infrastructure built to meet the needs of professional business Web applications. ICM caches dynamic content by incorporating the query string portion of the request URL into the cache key that is used to uniquely identify both the requested resource and any parameters submitted along with it. To support sophisticated Web applications and to prevent cache content from becoming outdated, ICM supports active caching. Active caching allows application-dependent events to trigger the invalidation of cached content. This is more sophisticated than standard HTTP caching, which provides only limited application control over the validity of cached content (typically based just on expiration times for the content). The ICM also caches unfound objects related to invalid requests. As a result, SAP Web Application Server is protected from overload situations in which the system is flooded with invalid or malicious Web requests.

PRESENTATION LAYER

In the presentation layer, the user interface of a Web application can be developed directly with JavaServer Pages (JSP) or Business Server Pages (BSP) and the corresponding JSP tag library; or with the high-level Web Dynpro technology; or with a mixture of both, if necessary. Development with Web Dynpro technology always results in Web applications with clearly separated presentation and business layers. The underlying business layer provides the business content in Java or ABAP.

BUSINESS LAYER

The most important part of any application is the business logic, which processes company-critical data. The business logic runs on SAP Web Application Server and can be propagated to the outside – either to a user interface or to a third-party application. Business logic can be exposed with a new user interface or with an enhanced version of an existing interface. The business logic can be written either in ABAP or in Java based on the J2EE standard. This provides the greatest flexibility during development. The business layer consists of a full-featured, J2EE-certified runtime environment that processes the requests passed from the ICM and dynamically generates the responses. The business logic layer is based on SAP's professional and proven application server environment.

Developers can implement business logic and persistence with Enterprise JavaBeans (EJB) using the full J2EE environment. EJB provides a component model for the application logic to process data and to generate a result for a particular page. EJB is the fundamental link in the J2EE programming model between the presentation layer and business-critical data. Developers can also access the business objects of applications running in the ABAP environment to benefit from their business logic and persistence. For example, a Web application that creates sales orders might implement the presentation logic using the Web Dynpro technology, but it can also easily benefit from the existing sales order application running in the ABAP environment by accessing the appropriate Business Application Programming Interface (BAPI®). The business logic is usually provided through business objects that access the database.

INTEGRATION LAYER

SAP Web Application Server offers a comprehensive runtime infrastructure for providing integration services. The integration engine is an integral part of SAP Web Application Server and allows instant connection to SAP Exchange Infrastructure for all mySAP.com solutions out of the box. The integration engine provides messaging services that exchange messages between the components that are connected in SAP Exchange Infrastructure. In SAP Exchange Infrastructure, shared knowledge is maintained centrally, whereas the physical communication model allows communications through the central integration engine and also peer-to-peer communication directly from component to component. For more information on the integration engine, see the white paper entitled *SAP Exchange Infrastructure: Process-Centric Collaboration*.

PERSISTENCE

Persistence is a major technical challenge in business applications. To minimize the effort required to implement a persistence framework, SAP Web Application Server has outstanding persistence capabilities. These capabilities include a highly sophisticated database interface, transaction capabilities, and caching, as well as object relational mapping. The database interface ensures optimized data access from within the ABAP environment through Open SQL, accompanied by technologies for SQL statement caching, data buffering, and tracing. Business logic can be developed completely independently of the underlying database and operating system. In addition, SAP brings its business knowledge to the technology world by supporting the evolution of open standards, such as J2EE. SAP will continue to enhance and extend the J2EE offering for full-scale business application development. Database independence is also made possible by support for open standards, such as Java Database Connectivity (JDBC). SAP propagates the outstanding capabilities of Open SQL for ABAP to Open SQL for Java and offers a variety of standard application programming interfaces (APIs) to application programmers, such as SQLJ. Other technologies like Java Data Objects (JDO) and container-managed persistence (CMP) for EJB or even the direct use of the JDBC API are supported, too.

OPENNESS BASED ON TECHNOLOGY STANDARDS

PUBLISHING, DISCOVERING, AND ACCESSING WEB SERVICES

SAP Web Application Server is an infrastructure for developing and deploying Web services. The Web services can be used in the enterprise and across enterprise boundaries, with any SAP or non-SAP application component and on any technical platform using any technical protocol.

Web services encapsulate the relevant aspects of business interaction independently from the technical architecture. For example, a Web service designed to request process details that are relevant to a particular industry can use high-level specifications, business scenarios, and business profiles as the entry point for the Web service. Yet the Web service can be used to publish, discover, and access business functions across an entire network using open standards.

Technology standards that are used in Web services and are completely supported by SAP Web Application Server are XML, SOAP, WSDL, UDDI, and the newly invented Web Service Choreography Interface (WSCI). Every participant in the Web service scenario has a WSDL interface to describe the Web service. For example, SAP makes its BAPI interfaces available as Web services through WSDL. The call for a Web service is transferred via SOAP in standard XML-based protocol for exchanging information. This allows Web services provided by SAP to be accessed in a well-defined manner that is independent of the platform and the programming language and allows the seamless integration of SOAP-based, externally available Web services.

To locate business partners and services, companies need access through a central instance. This central instance is a UDDI business directory. Through UDDI, Web services can be made available to other internal or external users of the Web service, or they can be retrieved for use as part of a business process.

SAP helps to drive the standardization of Web services by serving as a founding member of the Web Services Interoperability (WS-I) Organization and as a UDDI business registry node operator.

DEVELOPING, DEPLOYING, AND INTEGRATING WEB SERVICES

SAP Web Application Server provides a scalable and reliable Web service infrastructure that delivers Web services at high performance. SAP Web Application Server supports Web access using Web browsers, a range of mobile devices, and Web-enabled client programs.

With SAP Web Application Server, companies can extend their solutions by exposing and integrating Web services based on their business needs and available development skills. Companies can also integrate Web services to create business processes that involve multiple business partners over various systems. SAP Web Application Server allows companies to develop and deploy such Web services, giving them optimal flexibility in responding to fast-changing markets.

Companies can implement Web services with SAP Web Application Server in one of two ways, depending on whether they base Webs services on existing capabilities or developing brand-new functions. Companies can implement new capabilities in Java or ABAP and expose them as a Web service. Or they can expose existing capabilities as a Web service by first defining the Web services interface in WSDL and then implementing the capabilities against a generated proxy.

Every self-contained, modularized function – whether provided by SAP as part of a mySAP.com solution or developed by a company or a partner – can be made available as Web service and accessed over the Internet using standard protocols. This includes BAPIs, the proven and established interfaces from SAP. That means that companies can very easily expose the full range of business functions as Web services, thanks to SAP's native integration of Internet technology and support of the Web service paradigm.

In addition, SAP Web Application Server allows Web services to be integrated regardless of where they reside or how they were implemented. Thus a company can easily realize business processes that span different systems within or across that company. The variety of supported tools and their seamless integration into the development process allow companies to fulfill their company's requirements rapidly and with high quality using commonly accepted and published Internet standards.

The use of Web services is not limited to a specific communication technology stack. Companies can choose to optimize communication through other transport protocols and formats, including Remote Function Call (RFC), in a highly optimized LAN-based synchronous or asynchronous communications behind the firewall. Companies can therefore use Web services to integrate enterprise applications behind the firewall, as well as collaborative scenarios that cross company boundaries.

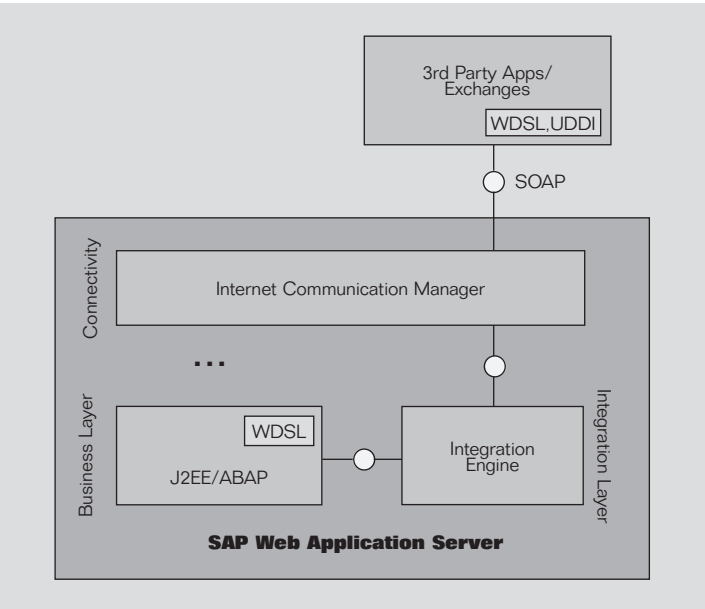


Figure 2: Provision of Web services

Web services that are developed and deployed using SAP Web Application Server are fully integrated into SAP Exchange Infrastructure. This means any component based on SAP Web Application Server integrates out of the box with SAP Exchange Infrastructure and can automatically provide the proxy in ABAP or J2EE according to the interface definition in the integration repository. For more information, see the white paper entitled *SAP Exchange Infrastructure: Process-Centric Collaboration.*

INTEGRATED DEVELOPMENT ENVIRONMENT

SAP Web Application Server is open to Java development environments so that companies can develop and deploy J2EE-based Web applications that operate on SAP Web Application Server. In addition, SAP provides tools and add-ons for specific needs in these development environments. For example, the tools provide the ability to generate Java classes for Web services provision from existing SAP business objects (BAPIs), tool support for tracing configurations, and direct access to the SAP Web Application Server deployment tool.

For the past 10 years, SAP has delivered a highly robust, integrated development environment for developing business applications in ABAP. The ABAP™ Workbench has proven its reliability and scalability in many areas, including traditional graphical user-interface development, Web development, and business logic development on the server side. Many of the built-in features of the ABAP Workbench are also essential to integrated development for Java as well. SAP therefore will reuse some of the underlying concepts for its Java development support in ways that completely conform to the J2EE standard. SAP actively helps drive the evolution of the J2EE platform to enhance its use for mission-critical development and professional user interaction.

SAP recognizes the strong momentum in the market toward open source frameworks for Java IDEs. In a first step, SAP will provide support for the open source framework Eclipse. Companies will then be able to use all the supported functions for J2EE development, plus additional easy-to-use tools built on top of Eclipse to design, develop, deploy, and continuously change J2EE-based business applications. These additional functions include the capabilities of Web Dynpro and comprehensive features for deployment, versioning, and globalization. Furthermore, companies can integrate any tool that their partners or independent vendors use to meet their special needs. Companies can integrate them in an open framework and easily personalize a development environment.

CONNECTIVITY

SAP Web Application Server provides the open and reliable infrastructure for deploying highly scalable Web applications and Web services. Support for industry standards and the open, extensible architecture of SAP Web Application Server ensure that today's investments work with tomorrow's technology. SAP continuously chooses and implements new technologies to meet anticipated needs.

To integrate third-party products, tools, and applications, SAP Web Application Server supports several open connectivity standards, including the J2EE Connector Architecture and Microsoft .NET connectivity. As a result, SAP Web Application Server provides complete technical interoperability across all mySAP.com solutions and third-party components.

A wide range of protocols and formats is supported for communication with SAP, non-SAP, and third-party components. SAP Web Application Server is open to the Common Object Request Broker Architecture (CORBA), Component Object Model+ (COM+), File Transfer Protocol (FTP), and SMTP to connect to non-SAP systems. RFC is an SAP standard protocol for remotely executing functions and remotely invoking methods of business objects. It has been made available in most operating systems and programming languages.

SAP Web Application Server supports technology for parsing and rendering XML documents, providing full Document Object Model (DOM) support for both the ABAP and J2EE environments. It also supports document transformation with Extensible Stylesheet Language Transformations (XSLT) and data access via XML Path Language (XPath).

SAP Web Application Server can integrate with market-leading Web servers like Apache and Microsoft Internet Information Services (IIS) through HTTP proxies. Using this approach, popular Web servers can be used together. Web device servers for mobile devices, such as WAP-enabled cell phones, are also supported. New Web device servers can be easily integrated.

OPEN PLATFORM STRATEGY

SAP Web Application Server is platform independent. It runs on a broad range of platforms for the presentation, communication, and server environment. Companies can choose their favorite platforms from a variety of supported combinations to match their performance requirements and existing IT infrastructure. SAP's strategy is to support any platform that matches SAP's high standards and has a significant and viable market share.

For the presentation layer, the Web browser is the preferred presentation environment. Mobile devices are supported, and special support is provided for the Windows and Java environments. A comprehensive collection of operating system, hardware, and database combinations complete the comprehensive platform support.

SAP works closely with hardware and technology partners to ensure continuous support and to protect current investments.

RELIABLE OPERATIONS

PERFORMANCE AND SCALABILITY

Scalability is the ability to support an increasing number of users. A scalable solution can grow seamlessly with a business and handle an increasing number of incoming requests. In a distributed environment, load balancing is necessary to ensure the scalability of business applications. SAP Web Application Server uses software-based dispatching to distribute requests to all SAP Web Application Server instances running in a clustered environment. The workload of the machines in the cluster is determined from regularly received information. Transparent load balancing with SAP Web Application Server is session based; if one request is sent to SAP Web Application Server, all subsequent requests during the same session are sent to the same SAP Web Application Server instance.

High performance is assured by the proven architecture of SAP Web Application Server, accompanied by SAP's proven transaction capabilities in a distributed environment. Another reason for the excellent performance of SAP Web Application Server is its sophisticated caching of database and HTTP requests.

SAP Web Application Server can be deployed as a very compact solution, with all architecture layers implemented on a single system. Companies can size SAP Web Application Server to meet their current performance requirements, then adapt it over time as requirements change.

HIGH AVAILABILITY

High availability is important not only from a technical standpoint, but also from a business standpoint. Issues to consider include loss of worker productivity in the event of a failure, the impact of a failure on a company's reputation, and the total costs associated with unplanned downtime. After setting up a fail-over strategy and determining an availability level – which describes planned downtime and users' behavior in the event of a failure – companies can define technical requirements for high availability.

In a heterogeneous environment, high availability depends on the availability of all involved components – such as routers, Web servers, gateways, firewalls, databases, and application servers. Depending on the degree of availability required, companies may need a redundant distribution of components. SAP Web Application Server contributes to the fail-over strategy at the application-server level and at the component level. With SAP Web Application Server, requests are automatically distributed to other SAP Web Application Server instances if one server breaks down. This load balancing increases the overall throughput. A fail-safe cluster host provides high availability of the central database. Corresponding network hardware, which selects a suitable server according to the geographical location of the client, supports this load balancing and can be connected with SAP Web Application Server.

SECURITY

SAP Web Application Server supports state-of-the-art Internet security standards such as HTTPS, Secure Sockets Layer (SSL), and Lightweight Directory Access Protocol (LDAP). It provides secure communication between all client and server components, authentication and single sign-on capabilities, central user administration, digital certificates, digital signatures, and auditing capabilities. For detailed information about security, see the white paper entitled *SAP Security: Secure Business in Open Environments*.

SOFTWARE LOGISTICS AND LIFE-CYCLE MANAGEMENT

Any business Web application requires comprehensive management for all aspects of the software life cycle. To adapt an application to business needs, companies typically need to change some configuration settings in the application. Sophisticated tools for all configuration settings and ongoing updates to the system landscape help make these tasks more manageable and reduce the cost of ownership. However, changes and updates should not be done directly in the mission-critical productive environment; the new configuration must be tested first to guarantee uninterrupted production operation. To facilitate these tests, SAP recommends that companies separate configuration, quality assurance, and production into separate systems or layers and transfer the configuration settings, as well as any modifications and new developments, between these layers. For this, SAP Web Application Server provides sophisticated change management and transport services that meet mission-critical business needs. In the configuration layer, all changes are reported and transported to the quality assurance layer, where they are tested. After being tested successfully, the settings are moved to the production layer, where they are used for the real business. These services not only support the first implementation of an application, but also support the continuous changes that result from business or organizational changes or the implementation of additional functions. The comprehensive software logistics supported by SAP Web Application Server for development, test, and production landscape are a prerequisite for continuous operations.

MONITORING AND ADMINISTRATION

A comprehensive systems management infrastructure provides easy control of operations and lowers the total cost of ownership. With SAP Web Application Server, companies can achieve all their major monitoring and administration goals – from examining single user interactions to monitoring key business processes within applications from the standpoint of their workflow, impact on performance, and so on. An agent-based architecture acts as a common starting point and gives administrators a view of monitored components and their performance. Quick reaction to various types of alerts and exceptions, performed either automatically or manually by the administrator, reduces costs and helps ensure continuous operation.

GLOBALIZATION

A professional, state-of-the-art Web application must support various languages and business requirements. SAP Web Application Server supports a large library of globalization features, which eases the implementation of typical business processes and improves their quality. The features include currency support, internationalization, time zones, accounting, auditing, and security. A huge set of business knowledge is also available for these features. Best practices that meet the needs of individual countries include specific display options for dates, country-specific holiday and factory calendars, and country-specific format settings such as value-added tax registration numbers. And of course, SAP provides updates regularly. For detailed information about globalization, see the white paper entitled *Globalization: Meeting Local and Global Requirements*.

CONCLUSION

In today's interconnected, collaborative world of e-business, success means providing all members of the business community with the services they need, whenever they need them. The technology infrastructure behind e-business solutions must focus on business goals, and it must not be an administrative burden or an unwieldy, complicated system. A technology solution must be reliable, secure, and always available – so that ultimately, companies can best serve their company's most important resource: their customers.

With SAP Web Application Server as part of mySAP Technology, companies have an open, integrated, and secure infrastructure for Web services that seamlessly connects users and applications. SAP Web Application Server protects the investments a company has made in SAP software. SAP Web Application Server supports the entire life cycle of Web applications – from development through deployment and operation. And developing Web user interfaces for professional business Web applications is easier using Web Dynpro.

With SAP Web Application Server, companies can easily operate and develop reliable Web-based applications and Web services as the basis for a powerful, successful e-business solution.

GLOSSARY

adapter: A component that eases the integration of existing applications with other applications or the infrastructure. An adapter provides technical connectivity and necessary business logic.

ABAP™: SAP's object-oriented programming language and environment for developing, deploying, and operating mySAP.com application components.

application: A set of functions or Web services that is typically delivered as a single component.

application programming interface (API): An interface that applications use to offer Web services and to communicate with each other.

Audit Information System (AIS): An SAP auditing tool that improves the quality of system and business audits. AIS consists of an audit reporting tree and is a structured, preconfigured collection of SAP standard programs.

authentication: The process of identifying a person or system component, usually as a prerequisite for allowing the person or component access to a system.

biometrics: Automated method of verifying or recognizing the identity of users of digital devices on the basis of certain physiological characteristics, such as a fingerprint or iris pattern, or aspects of behavior, such as handwriting or keystroke patterns.

Business Application Programming Interfaces (BAPI®): Open, stable, object-oriented interfaces through which the capabilities of mySAP.com applications can be accessed. BAPIs are independent of the technical realization.

business process: The execution of one or several Web services in a controlled way, driven by one or several individuals or events.

Business Process Modeling Language (BPML): A specification for the management of business processes that span multiple applications, corporate departments, or business partners and that go behind the firewall and over the Internet. For more information, go to http://www.bpmi.org.

business scenario: A business process. In a business scenario, more than one component provides Web services, more than one person is involved, or business process control resides outside a single, service-providing component.

cascading style sheets (CSS): A simple mechanism for adding style (for example, fonts, colors, or spacing) to Web documents, typically in HTML.

channel: Collections of iViews that can be built into role-related work sets defined by the administrator. End users can drag the iViews contained in channels onto portal pages to personalize the portal.

container-managed persistence (CMP): Simplifies the task of writing entity beans because the container generates the code to access the data source and manages the persistent state of the bean.

code page: Defines the mapping between a character set and a sequence of one or more bytes. The code page determines the characters that can be used in programs and that can be displayed on output devices (for example, printers and terminals).

collaboration: Joint work and communication among people and systems of a company – including business partners, suppliers, and customers – to achieve a common business goal.

collaboration knowledge: Describes how collaboration works by detailing process descriptions, business rules, Web services, interfaces, roles, and so on.

Common Information Model (CIM): A model for describing the overall management information in a network or enterprise environment. For more information, go to http://www.dmtf.org/standards/standard_cim.php.

Common Object Request Broker Architecture (CORBA): An open protocol for communication between distributed objects.

Component Object Model+ (COM+): A component model from Microsoft for communication between distributed objects.

component: Software that provides application functions and Web services. Components can be shipped and deployed independently, and they have their own release cycle.

demilitarized zone (DMZ): Security zone that exists between two networks. It allows connections between the networks while preventing unauthorized access to the systems located within the networks.

digital certificate: Digital document that acts as a user's digital identification card on the Internet. Digital certificates are used for authentication and for verifying digital signatures.

digital signatures: Security mechanism for protecting digital data. The digital signature serves the same function for the processing of digital data as a handwritten signature serves for paper documents. It is based on public-key cryptography.

directory: Used to store and look up shared information. Typically, directories are optimized for read access. In this context, a directory is mainly used for information at configuration time.

double-byte code page: Defines the mapping between a set of characters and a sequence of one or more bytes. Some languages, such as Japanese, use more than 256 characters so the complete character set for these languages cannot be mapped using a single-byte code page. In a double-byte code page, a character can be either one or two bytes long. For example, in the Japanese code page (SJIS), if the first byte has a value between 0x81 and 0x9F, then the second byte is also part of the representation of the current character. Otherwise, only one byte is used to identify a character.

Drag&Relate™: Trademarked name for the user operation that reaps the benefit of unification. Drag&Relate is a user drag-and-drop action.

Dynamic Program (Dynpro): Screens elements of SAP transactions. Dynpros are the combination of the screen and its accompanying flow logic.

Eclipse: An open-source, extensible IDE platform for developing and debugging applications. Written entirely in Java, Eclipse is designed to be a unified development environment, including testing, performance tuning, and debugging in multiple programming languages.

Electronic Business XML (ebXML): Framework for an open, XML-based infrastructure that enables the global use of electronic business information in an interoperable, secure, and consistent manner by all parties. It is sponsored by OASIS and other groups. For more information, go to http://www.ebxml.org.

electronic data interchange (EDI): A family of standards that facilitate the electronic exchange of information among different companies.

engine: Provides the runtime environment for dedicated functions and Web services. Engines are part of components.

enterprise portal: A single point of entry to all information, applications, and services that people need to do their jobs according to their roles. Enterprise portals provide a way for suppliers, customers, partners, and employees to access all relevant content easily and securely and to participate in all types of business processes.

eventing: The automatic passing of parameters from one iView to one or more related iViews.

eXtensible Business Reporting Language (XBRL): An open specification that uses XML-based data tags to describe financial statements for both public and private companies. For more information, go to http://www.xbrl.org.

Extensible Markup Language (XML): The universal format for structured documents and data on the Web, XML is increasingly becoming the general standard document format of structured data. For more information, go to http://www.w3c.org/XML.

Extensible Stylesheet Language Transformations (XSLT): A specification for transforming XML documents into HTML or other types of documents. For more information, go to http://www.w3.org/Style/XSL.

Fast Common Gateway Interface (FastCGI): A standard for interfacing external applications with information servers, such as HTTP or Web servers. FastCGI is a language-independent, scalable, open extension to the CGI specification that provides high performance without the limitations of server-specific APIs.

File Transfer Protocol (FTP): An open protocol for exchanging files.

Generic Security Services-Application Programming Interface (GSS-API): Application-level interface (an API) to network security systems. GSS-API allows the integration of security functions that are available from external security products, such as strong authentication or encryption.

globalization: Strategy that addresses all of the enterprise issues associated with making a company truly global. Globalizing products and service involves integrating all of the internal and external business functions with marketing, sales, and customer support in the world market.

HR-XML: An independent, nonprofit organization dedicated to developing and promoting standardized XML vocabularies for human resources. For more information, go to http://www.hr-xml.org.

HyperRelational Navigation Protocol (HRNP): A naming protocol tunneled through HTTP. HRNP manifests itself as a link behind data represented to users. This HRNP link contains metadata about the source it is being dragged from and how it relates to the target it is being related to by the user.

HyperRelational technology: Technology that leverages unification to pass data and context to the portal through HRNP.

Hypertext Markup Language (HTML): The standard document format for display in Web browsers.

Hypertext Transfer Protocol (HTTP): The open Internet standard protocol that is used to exchange documents.

Hypertext Transfer Protocol with Secure Sockets Layer (HTTPS): The secure variant of HTTP using Secure Sockets Layer (SSL).

integrated development environment (IDE): A development environment with easy integration into the development and deployment process for applications. The Java IDE is used for Java development.

Intermediate Document (IDoc): SAP document standard for electronic document exchange.

impersonation: Acting on behalf of a user. A component accesses another component, and the accessed component assumes the access comes from a specific user.

Interactive Financial Exchange (IFX) Forum: A family of global business requirements and specifications that result in an open and interoperable foundation for online financial services. For more information, go to http://www.ifxforum.org.

interface: An abstract definition of Web services. Interfaces allow Web services that comply with these interfaces to be accessed. The interface defines which information and data must be provided to use a Web service and how the result of the Web service will be made available.

internationalization: Provides the technical foundation to enable programs to support multiple scripts and languages without redesign or modification. Once a user has selected a language environment, all programs transparently alter their runtime behavior to meet the expectations of the user.

Internet standards: A common set of open standards used for communication and integration over the Internet. Examples of Internet standards are HTTP, XML, and WSDL.

iView: A self-contained, XML-based presentation element. A well-defined set of interfaces displays content and the personalization of the content elements presented as part of a portal page.

Java 2 Platform, Enterprise Edition (J2EE): The standard for developing multitier enterprise applications based on Java. This standard has been defined by an open community, including SAP, and is driven by Sun Microsystems Inc. For more information, go to http://java.sun.com.

J2EE Connector Architecture (JCA): Defines standard Java interfaces for simplifying the integration of enterprise applications with J2EE-based Java applications. With these interfaces, Java developers can access existing databases, e-business applications, and legacy systems.

Java Message Service (JMS): Provides a consistent set of APIs that gives developers access to the common features of different messaging system products.

Java Management Extensions (JMX): A universal, open technology for managing the adapted legacy systems, implementing new management solutions, and plugging into those of the future. JMX provides the tools for building distributed, Web-based, modular, and dynamic solutions for managing Java-based devices, applications, and service-driven networks.

JavaScript: A basic scripting language that allows Web authors to create dynamic pages that react to user interaction.

Java Database Connectivity (JDBC): Provides uniform access to relational databases like DB2, Oracle, Microsoft SQL Server, and SAP DB.

Java Data Objects (JDO): An API for transparent database access. Developers can write code in the Java programming language that transparently accesses the underlying data store without using database-specific code.

JavaServer Pages (JSP): Allows Web developers and designers to rapidly develop and easily maintain information-rich, dynamic Web pages. JSP technology separates the user interface from content generation, enabling designers to change the overall page layout without altering the underlying dynamic content.

Kerberos: An authentication system that uses symmetric cryptography to provide protection. For more information, go to http://web.mit.edu/kerberos/www/.

Lightweight Directory Access Protocol (LDAP): A standard protocol for accessing directory services. It is typically used to retrieve organizational and user data, as well as other resources, such as files and devices, in both the public Internet and corporate intranets.

localization: Making a product linguistically and culturally appropriate to the target locale (country or region and language) where it will be used and sold.

marketplace: Connect business communities and enhance business processes by providing collaborative functions, streamlining operations, and improving efficiencies. Private marketplaces are also called private exchanges.

messaging: Exchanging documents. Messaging is the transfer of information among Web services that are provided by separate components.

Microsoft .NET: A platform from Microsoft for XML-based Web services. It includes tools to develop and deploy Web-based applications. For more information, go to http://www.microsoft.com/net.

model-view-controller (MVC): A design pattern for successfully and efficiently relating the user interface to underlying data models. It is widely used by developers as a useful pattern to reuse object code. It significantly reduces the time required to develop applications with user interfaces.

nonrepudiation: The inability to deny having performed an action. For example, a nonrepudiation service can prove that a person sent a message to another person. In electronic business, this can be achieved by using public-key technology.

NT LAN Manager (NTLM): The authentication protocol that Windows NT uses to pass authentication information between the client and server when logging on.

Object Linking and Embedding (OLE) Database (DB): A low-level application programming interface for access to different data sources. OLE DB provides SQL-based access and other types of access to different data sources.

online analytical processing (OLAP): Enables users to easily and selectively extract and view data from different points of view. To facilitate this kind of analysis, OLAP data is stored in a multidimensional database.

Online Certificate Status Protocol (OCSP): Standard used by client applications to check whether digital certificates are valid at the time of a given transaction. OCSP enables rapid verification of the revocation status of digital certificates.

Organization for the Advancement of Structured Information Standards (OASIS): A nonprofit, international consortium that creates interoperable industry specifications based on public standards like XML and SGML. For more information, go to http://www.oasis-open.org.

personal digital assistant (PDA): A handheld information appliance that offers multiple communication features, including address book, personal information manager, cellular phone, calendar, and networking capabilities. PDAs use several different types of input technologies, including stylus, voice, and keyboard.

public-key infrastructure (PKI): System that manages the trust relationships involved with using public-key technology. The role of the public-key infrastructure is to make sure that digital certificates and certification authorities can be validated and trusted.

Remote Function Call (RFC): An SAP standard protocol for remote execution of functions and remote invocation of methods of business objects. RFC is available on most operating systems and programming languages.

repository: A storage area for shared metadata and information. In this context, a repository is mainly used for information at design time.

role: A collection of content that users need to access to do their jobs. Roles are specific to individual groups of internal and external users, and they match their specific tasks with information or service needs.

RosettaNet: A consortium of high-tech companies working to create and implement industrywide, open, e-business process standards. For more information, go to http://www.rosettanet.org.

SAP® GUI: A universal client for accessing SAP functionality. SAP GUI works like a browser. It gets information from the SAP server like "what, where, when, and how" to display content in its window.

SAP® GUI for HTML: SAP client that dynamically emulates the screen elements of SAP transactions in a Web browser by automatically mapping them to HTML.

Secure Network Communication (SNC): SAP interface that links SAP systems to third-party security products for authentication and encryption.

Secure Sockets Layer (SSL): A standard protocol for transmitting secure messages over the Internet using public-key and private-key encryption.

Security Assertion Markup Language (SAML): An XML security standard for exchanging authentication and authorization information. For more information, go to http://www.oasis-open.org/cover/saml.html.

Simple Mail Transfer Protocol (SMTP): An open protocol for exchanging electronic mail.

single-byte code page: Defines the mapping between a set of characters and a sequence of one or more bytes. In a single-byte code page (for example, ISO 8859x), each character is mapped to a single byte. Therefore, a single-byte code page can only contain a maximum of 255 characters. In ASCII-based code pages, the first part of a code page table (0x20 Ox7F) contains printable characters of the 7-bit ASCII character set (English alphabet). The second half contains language-specific characters (for example, German umlauts or Greek characters).

single sign-on (SSO): Mechanism that eliminates the need for users to enter passwords for every system that they log on to. Single sign-on allows users to authenticate themselves once and then log on to all the systems that operate in the single sign-on environment without further intervention.

SOAP: A lightweight protocol for exchanging information in a decentralized, distributed environment. It is an XML-based protocol that is typically used with HTTP. SOAP includes conventions to represent method calls of objects or function calls and the respective responses, as well as conventions to represent standardized data types. For more information, go to http://www.w3.org/TR/SOAP.

SQL-J: A technology that enables a Java program to access a database using embedded Structured Query Language (SQL) statements.

time stamp: Time and date of an event converted from local time to Coordinated Universal Time (UTC). UTC corresponds to Greenwich mean time (GMT).

time zone: Set of rules that dictate the offset of the user's local time from Coordinated Universal Time. The local time of a particular user depends on the user's location. The offset of a location from UTC depends on geography, region, or country and the use of daylight saving time (DST).

Unicode: An international standard (see ISO/IEC 10646) that assigns characters from virtually every language and script a unique Unicode scalar value. Unicode currently defines more than 90,000 characters, with room for more than one million characters. Unicode includes all characters used in business-relevant languages, as well as other symbols and icons. For more information, go to http://www.unicode.org.

unification: Permits user-centric integration of data sources. Using Drag&Relate, users can understand how information in one unified source correlates to information in another unified source.

unifiers: Leverages the application architecture, user interfaces, security, and all customization inherent in an application while surfacing the application with HyperRelational technology.

Universal Description, Discovery, and Integration (UDDI): A sweeping industry initiative to create a platform-independent, open framework for describing Web services, discovering businesses, and integrating business services using the Internet. Its purpose is to create an operational registry, which is available today. For more information, go to http://www.uddi.org.

virtual private network (VPN): A private data network that uses the public telecommunication infrastructure while maintaining privacy through security protocols.

Web-Based Distributed Authoring and Versioning (WebDAV): An extension of the HTTP protocol that allows file sharing over the Internet. It provides locking, property management, and remote file management capabilities, and it makes Web resources work like standard, local file sharing.

Web-Based Enterprise Management (WBEM): A set of management and Internet standard technologies developed to unify the management of enterprise computing environments. For more information, go to http://www.dmtf.org/standards/standard_wbem.php.

Web service: A self-contained, modularized function, that can be published, discovered, and accessed across a network using open standards. It is the implementation of an interface by a component, as well as an executable entity. To the caller or sender, a Web service is a black box that may require input and delivers a result. Web services provide integration within and across enterprises on top of any communication technology stack (asynchronous or synchronous) and in any format.

Web Service Choreography Interface (WSCI): XML-based interface to deliver automated, application-to-application collaboration by describing the flow of messages exchanged by a Web service in a particular process.

Web Services Description Language (WSDL): A specification for describing Web services as a set of end points operating on messages. For more information, go to http://www.w3.org/TR/wsdl.

Web Services Interoperability Organization (WS-I): Open industry organization chartered to promote Web services interoperability across platforms, operating systems, and programming languages.

Windows Message Instrumentation (WMI): An implementation of the DMTF WBEM initiative for Microsoft Windows.

Wireless Application Protocol (WAP): A standard defined by the WAP Forum, an industry consortium with such members as Openwave, Ericsson, Nokia, Motorola, and Siemens. The purpose of the standard is to enable wireless access to the Internet and advanced telephony services. For more information, go to http://www.wapforum.org.

Wireless Markup Language (WML): Part of the application development environment of WAP. It is a markup language, similar to HTML, for wireless devices like mobile phones.

work set: A collection of application screens and iViews on a page or series of pages that allows users to better perform tasks defined in their roles.

XML Common Business Library (xCBL): A cross-industry XML component library for business-to-business e-commerce. It was originally developed by Commerce One Inc. For more information, go to http://www.xcbl.org.

XML Schema Definition Language (XSDL): A language to express a shared vocabulary. It provides a way to define the structure, content, and semantics of XML documents.

X.509: Widely used standard for digital certificates. The format requires certain standard information (for example, an e-mail address) to identify the owner and the issuer of the certificate.

DOCUMENT HISTORY

VERSION 1.2

- **Changes in the general structure of the white paper series**
  - Web Dynpro was broken out as a separate white paper.
- **Executive Summary**
  - The key benefits of SAP Web Application Server were added: reliability, openness, and low cost of ownership.
  - This section explains that J2EE is now part of SAP Web Application Server.
- **SAP Web Application Server Architecture**
  - The overview architecture illustration has been updated.
  - The first architecture illustration has been removed.
  - The architecture description describes the separate functional layers and main technical components.
  - This section now describes the new structure of architecture.
- **Openness Based on Technology Standards**
  - A new section was included in the main part on Web services.
  - Information on Eclipse was added.
- **Reliable Operations**
  - The section summarizes runtime requirements.

THE BEST-RUN E-BUSINESSES RUN SAP

SAP AG
Neurottstraße 16
69190 Walldorf
Germany
T +49/1805 /34 34 24*
F +49/1805 /34 34 20*
* Subject to charge
www.sap.com

50 052 038 (02/07/13) Printed on environmentally friendly paper.