Gartner

Research

Publication Date: 16 July 2007 **ID Number: G00149380**

Hype Cycle for Application Infrastructure Middleware, Platforms and Architecture, 2007

Paolo Malinverno, Massimo Pezzini, Dale Vecchio, L. Frank Kenney, Jess Thompson, Roy W. Schulte, Yefim V. Natis, Mark Driver, Benoit J. Lheureux, Bill Gassman, Kimihiko Iijima, Michael J. Blechar

The software foundation of business applications is undergoing a transformation that is directly affecting architects, application developers and middleware technologists, and, via service-oriented architecture, indirectly affecting applications and IT services worldwide.

TABLE OF CONTENTS

LIST OF TABLES

Gartner

LIST OF FIGURES

What You Need to Know

Because of the combined effects of increasingly mature integration middleware, service-oriented architectures (SOAs), event-driven architectures (EDAs) and design ideas, as well as the widespread adoption of IT innovation — including Java Platform, Enterprise Edition (Java EE, formerly known as J2EE), Extensible Markup Language (XML), extreme transaction processing (XTP) — and Web standards, such as URLs, HTTP, Web Services Description Language (WSDL) and Simple Object Access Protocol (SOAP), application architecture is undergoing a major paradigm shift. This change in architecture is driving the evolution of the enabling middleware infrastructure and the platforms that contain and host application logic.

Companies should adjust their middleware strategies to implement new communication backbones based on enterprise service buses (ESBs); new types of application servers such as event-driven application platforms or extreme transaction processing (XTP) platforms; and general-purpose, multienterprise business-to-business (B2B) software. However, increasingly distributed and interoperable applications require more than just good design patterns and technologies. Companies must change their IT organization and software management practices through such concepts as integration competency centers or SOA centers of excellence (COEs) to take full advantage of the many new approaches to business computing.

The Hype Cycle

This Hype Cycle addresses the infrastructure for modern business applications built using forms of middleware. Middleware is runtime system software that directly enables application-level interactions among programs in a distributed computing environment. This Hype Cycle involves platform middleware, particularly application servers, that acts as the host container for business application logic. It also covers communication middleware and integration middleware, particularly ESBs and integration suites, that provide the "glue" that enables the wide variety of new, legacy and purchased applications to work together, even as they run on disparate application servers and operating systems and use disparate programming languages within and across company boundaries.

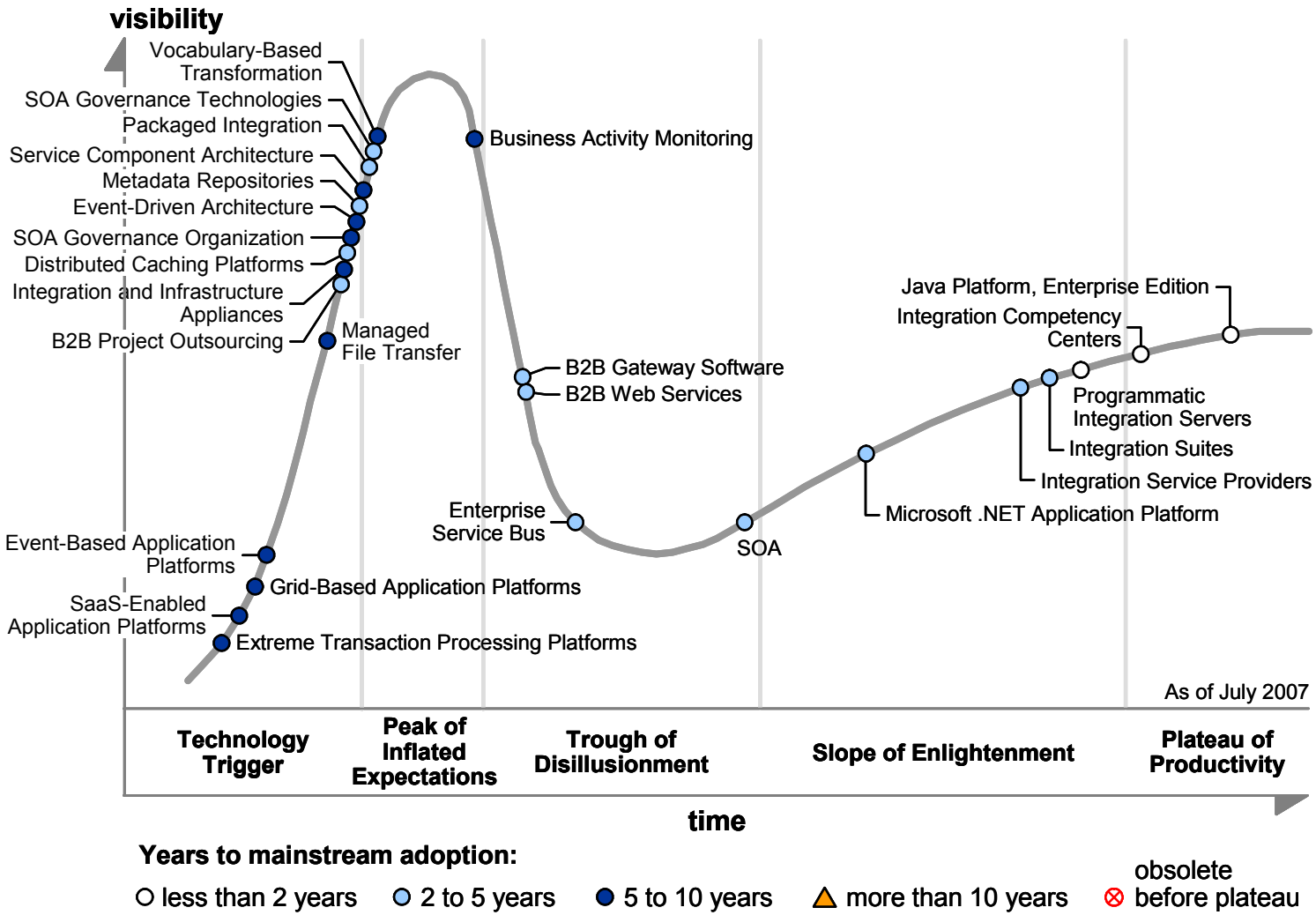
The accelerating pace of business, and the accelerating rate of change in business, is creating the requirement for more-flexible applications. Companies with more-flexible applications have a business advantage vs. those that are not as flexible. The basic idea of what constitutes an application is changing, as monolithic and client/server applications are gradually replaced by applications and processes composed of shareable business components, sometimes sourced outside the enterprise in a software-as-a-service (SaaS) model.

SOA business components are connected in "organic" networks, using loosely coupled and uncoupled EDA relationships. All development projects that will acquire or build more than 20 SOA services or EDA event types should use ESBs as core elements in their communication backbones, rather than point-to-point Web services, plain message-oriented middleware (MOM) or traditional remote procedure calls or object-request brokers. They should also use B2B gateway software to extend ESB infrastructure to link internal applications, processes and SOA to those of external business partners.

Platform middleware technologies, such as application servers, were once thought to be approaching a steady state based on .NET (which is being revitalized by a new, Vista-compatible version) and Java EE. However, new types of application servers — using grid, XTP and event-based technology, or designed for SaaS — are emerging, creating new alternatives for

developers. Simultaneously, integration continues its evolution away from hand-coded, hard-wired, one-to-one connections toward model-driven, "soft wired" connections that run in middleware intermediaries. Such intermediaries — including integration brokers, business process management (BPM) engines, integration suites and ESBs — implement functions such as message transformation, process management, monitoring, intelligent routing and service virtualization. Middleware intermediaries are increasingly extended beyond the boundaries of single enterprises to reach business partners through B2B gateway software.

Figure 1. Hype Cycle for Application Infrastructure Middleware, Platforms and Architecture, 2007



Source: Gartner (July 2007)

The Priority Matrix

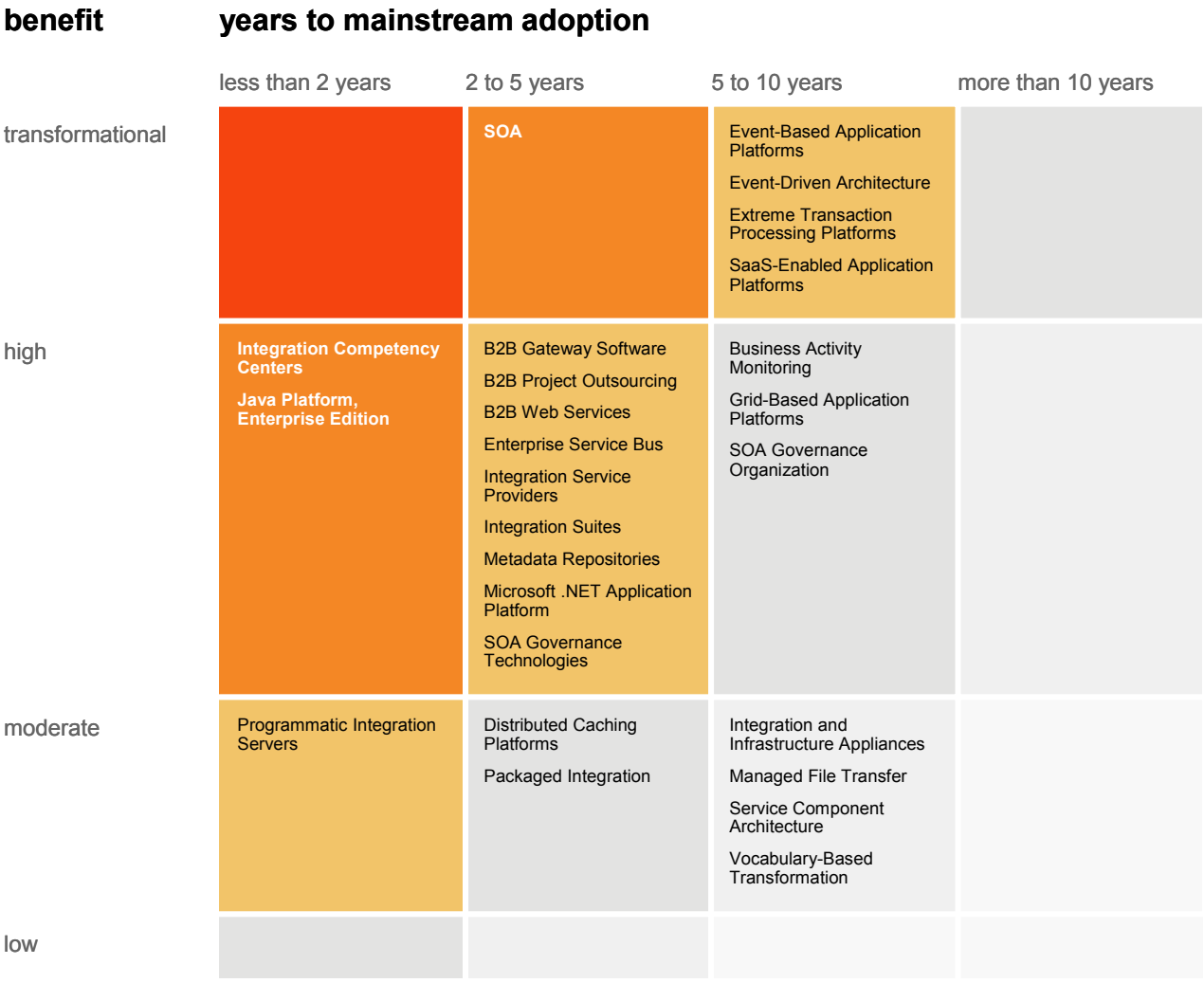
In many respects, SOA applications are inherently more complicated than traditional applications, because of the large number of moving parts (the service components) and the often-complex relationships among those components. To address this challenge, companies must improve the way they organize their application development, integration and maintenance activities by instituting centralized or federated integration competency centers (ICCs), which are sometimes referred to as SOA COEs when they govern SOA projects. ICCs rely on metadata repositories and service registries to help them manage the ever-changing SOA and integration scenarios.

The challenges of governing SOA projects are becoming clear, and vendors have begun to offer technology to support SOA governance processes. Vendors have dedicated large amounts of R&D and marketing funding to their registry and repository products, which grew successfully during 2006.

Mainstream users are increasingly shifting from small, proof-of-concept SOA applications to the large-scale development and deployment of SOA services. This is accelerating the adoption of ESBs and registries/repositories, which provide the runtime infrastructure for such applications. Event-processing concepts are being applied more broadly and on a higher level, pushing the adoption of EDA and the related technology (such as event-driven application servers).

Multienterprise/B2B infrastructures are used to exchange data and link business processes among a company's internal applications and those of its external business partners. Multienterprise infrastructure is available as software (B2B gateways, B2B Web services and managed file transfer) or services (through B2B project outsourcing or integration service providers). Multienterprise offerings are becoming increasingly popular, and their adoption is growing rapidly: Most midsize-to-large companies will implement some combination of multienterprise infrastructure.

Figure 2. Priority Matrix for Application Infrastructure Middleware, Platforms and Architecture, 2007



Source: Gartner (July 2007)

Off The Hype Cycle

Several changes have occurred in the relevant technologies since the previous Hype Cycle:

- Integration repositories have been renamed "metadata repositories."
- Business process networks have been excluded from this year's Hype Cycle, because of lower relative applicability, compared with other technologies included; however, this does not indicate low importance.
- Extensible microkernel-style platforms: Due to extensibility via the Open Services Gateway Initiative (OSGi), JBI, JMX and so forth, this is no longer considered a distinct platform category.

- Event-based application platforms has been slightly renamed to "event-driven application servers" (coverage continues).
- Alternative open-source application platforms: Coverage has been transferred to the Hype Cycle for Open-Source Technology.
- Basic Web services: Coverage has been transferred to the Hype Cycle for Web and User Interaction Technologies.
- Advanced Web services: Coverage has been transferred to the Hype Cycle for Web and User Interaction Technologies.
- Enterprise-scope application platform suites: Coverage has been replaced by composite application platforms, integration suites, business process suites and enterprise application servers.
- Open-source Java EE: Coverage has been transferred to the Hype Cycle for Open-Source Technology.
- Presentation integration servers: The number of vendors providing this kind of products has been reduced significantly due to intense merger and acquisition activity. Little innovation is being introduced into these mature products and adoption is only modestly growing as users turn to SOA as their strategic approach to legacy modernization and integration. Consequently, the functionalities of these products are being incorporated into larger suites addressing composite applications and legacy modernization.
- Web services management: This area of technology has diversified and evolved in several ways — for more information, refer to the Hype Cycle for Web and User Interaction Technologies.

On the Rise

Extreme Transaction Processing Platforms

Analysis By: Massimo Pezzini

Definition: Extreme transaction processing platforms (XTPPs) are a still-in-the-making form of platform middleware technology meant to provide a comprehensive, cohesive and coherent set of functionality to support XTP-style applications. Gartner defines XTP as an application style aimed at supporting design, development, deployment, management and maintenance of distributed transaction processing (TP) applications, characterized by exceptionally demanding performance, scalability, availability, security, manageability and dependability requirements. Modern TP applications are meant to support applications enabling multichannel access through multiple access devices, including Web browsers, rich clients, portals, mobile gadgets, sensors and other pervasive devices, as well as programmatic, service-oriented and event-driven interfaces. Modern TP applications are heterogeneous, participate in service-oriented and event-driven architectures, are integrated into multistep business processes via business process management tools and serve 24/7 global audiences, generating unpredictable and fluctuating workloads.

XTPPs provide a runtime environment designed to host even the most extreme TP requirements. They will combine, in an optimized fashion, event-processing and service container technologies, flow management and batch frameworks wrapped into a cohesive programming environment based on current software (such as BPEL, CDL, Java or C#) and leveraging a distributed transactional core. This will, in turn, leverage a high-performance ESB and distributed caching

mechanisms to improve performance and minimize latency, and dynamic applications deployment over tera-architectures (that is, large clusters of Linux or Windows commodity servers) to maximize availability and scalability, while keeping deployment costs low. XTPPs will also enable granular, dynamic and optimized deployment of application and system components to allow greater flexibility and low cost of application deployment.

Position and Adoption Speed Justification: Many industry sectors, such as travel, financial services, online betting and gaming, Web commerce, telecommunications and others, are eager to have low-cost transactional power to be able to cope with growth of their business and to minimize the cost of running their business-critical operations. The growing industry investment in software-as-a-service models is also creating demand for software platforms able to dynamically scale up while preserving uptime and response time for an increasingly large set of users (in this particular scenario, there is an additional requirement of support for multitenant deployments). Although current mainstream platforms — such as Java EE application servers and .NET — have proven capable of supporting rather demanding requirements, they are wearing thin when it comes to supporting the most-extreme scenarios. Some users have designed their own, custom XTP platforms by creatively combining open-source technology and specialized XTP products. As users' requirements crystallize around XTP as an appropriate paradigm to address these challenges, demand for comprehensive XTP-enabling products, such as XTPPs, will grow dramatically.

However, at this point, no vendor has released a comprehensive XTPP, but several are under development by large and specialized vendors. Specialized vendors already provide some portions of the picture — such as distributed caching platforms, grid-based application platform vendors and event-driven application servers — and are growing their ambitions. Most recently, large software vendors announced plans to address the XTP opportunity through internal developments or acquisitions. Standards — such as Spring, OSGi, Web services, SCA and others — are increasingly adopted in XTP scenarios. Some vendors released hybrid platforms combining cached messaging and ESB technology, with distributed Java EE, .NET or other forms of container technology to support demanding service-oriented architecture scenarios. Vaguely expressed users' requirements, lack of commonly agreed standards, still-in-the-making offerings from large vendors and immaturity of key technologies continue to be the main obstacles for rapid take off of XTPPs.

User Advice: User organizations facing business issues that can be sorted out with XTP-class applications should evaluate whether traditional transactional platforms, such as TP monitors, or modern mainstream platforms, such as Java EE-based application servers or .NET, can support their requirements. If this is not the case, they should look at point-XTP-enabling technologies, such as distributed caching platforms, event-driven application servers or grid-based application platforms, although they should also bear in mind the limited production readiness of many of these products and the questionable viability of some of the vendors in this space due to their small size and limited financial resources.

When evaluating traditional and specialized XTP-focused providers, consider these vendors' plans for announcing an XTPP strategy within the next 18 to 24 months as an indication to their commitment to the XTP opportunity. Only vendors of integrated XTPPs will have chances of surviving in the long term as users look for low cost but comprehensive offerings to support their requirements for low latency, scalable, high-performance and highly available, business-critical applications.

Business Impact: XTTP will enable mainstream users to implement a class of applications that, so far, only the most deep-pocketed and technically savvy organizations could afford due to the need of procuring expensive, complex and proprietary hardware and platform middleware. This will lower barriers to entry into even the most transactionally demanding business models (for

example, travel reservations or electronic payment processing) will enable users to potentially dramatically reduce costs of their current TP systems (thus significantly improving margins for their companies) and will enable new and creative business models — for example, by providing limitlessly scalable transactional back-ends to consumer-oriented applications leveraging Web 2.0 technologies, mobile phones and other consumer gadgets.

However, XTPPs will also impose new programming models, new best practices for software architecture, and new management and administration burdens on the IT environments, typically in addition to the established system environments they will gradually complement and possibly replace. Thus, IT departments will have to build and maintain in-house the new highly specialized skills or delegate highly demanding XTP projects to specialized service providers.

Benefit Rating: Transformational

Market Penetration: Less than 1% of target audience

Maturity: Emerging

Sample Vendors: BEA Systems; E2E Technologies; GigaSpaces; IBM; Kabira; Majitek; Microgen; Oracle; Paremus; RogueWave Software; Tibco Software; WareLite

Recommended Reading: "The Challenges of Extreme Transaction Processing in a World of Services and Events"

"Extreme Transaction Processing: Technologies to Watch"

"Cool Vendors in Integration and Platforms, 2007"

SaaS-Enabled Application Platforms

Analysis By: Yefim Natis

Definition: A SaaS-enabled application server (SEAS) is platform middleware (an application server) that is specially designed with multitenant capabilities in transaction isolation, data access isolation, and security and performance isolation to support the design and deployment of SaaS-style business applications. The objective is to enable multiple "tenants" (groups of users) to experience the application as dedicated to only them, while allowing the computing resources (such as memory, task scheduling, database access pools and communication channels) to be shared among tenants.

This kind of application server is essential for the effective operation of the SaaS style of application deployment, where one business application environment is used by multiple, independent groups of users. SaaS style may be used by independent third parties (providers; and the tenants are enterprises) or large enterprises (and the tenants are enterprise divisions or departments).

Position and Adoption Speed Justification: The SaaS model of deployment and acquisition of business applications originated from the business applications market, where the enabling technology was built into the solution and has not been isolated or productized. However, with the most recent announcements from WebEx/Cordys (WebEx Connect) and salesforce.com (Apex Code), SEASs are beginning to gain attention. Many leading Java EE application server vendors, as well as Microsoft, are engaged in providing support for SaaS business applications in the future. As these vendors begin to roll out product announcements, we expect this category of SEASs to head for the Peak of Inflated Expectations during the next 12 to 24 months.

Adoption beyond the hype, however, is likely to be gradual, given that users have to evaluate the business as well as the impact of the new relationship on the user enterprise and its technology

provider, as imposed by the SaaS model. Issues of security and preserving competitive differentiation in a SaaS context will require incremental steps to develop an understanding and trust in mainstream enterprises. Vendors will also discover that a fully functional SEAS must be coordinated with multitenant enablement in the database management system (DBMS), which will give a potential advantage to Microsoft, IBM and Oracle, and will make the transition of SEASs from exclusive to mainstream and standard relatively slow. Open source will likely be a significant contributor to the adoption of multitenancy in a standard way during the next three years because there is no open-source provider that has equal strength in DBMS and application servers at the same time.

User Advice: Users and independent software vendors (ISVs) can play multiple roles in a SaaS context. As users of business applications, enterprises should evaluate the SaaS model as a way of acquiring packaged applications (this delegates the underlying technology issues to the provider). Users may also want to offer internal applications to divisions and departments (or partners) as SaaS. In this scenario, users must recognize that SEAS technology is only emerging — and in practice, it is not yet an available purchase option. An alternative to deploying a SEAS is to use a general-purpose application server and to design the application to support multitenancy (the prevailing approach in the industry). Users can also offer a pre-SaaS application as SaaS by deploying multiple instances of that application in isolation — an expensive approach, not suitable for large scale. ISVs might consider joining the ecosystem of a large SaaS provider (such as salesforce.com's ApEx platform).

Some users attempt to separate their business processes between the basic, nondifferentiated "context" and the core processes that provide their businesses with competitive differentiation. They intend to consider only the "context" applications for early SaaS initiatives. We agree that less-critical applications should be the early candidates for experimentation. However, in the long term, we believe that an ability to creatively compose applications of different natures — outsourced, public, contracted, partnered and in-house — may be the key differentiation of the business through the IT organization, enabling the majority of individual business applications to be delegated (as SaaS or otherwise).

Enterprises should also consider their internal software as a candidate to be offered as a service to other businesses. Thus, the enterprise IT organization might turn into a (partial) profit center. To accomplish this, a SaaS-enabled application server capability may be essential, especially if the number of potential "tenants" is expected to be high.

ISVs should consider joining SaaS ecosystems to reduce the pressure for technical expertise in middleware operations, and gain the channel power of an ecosystem community.

Business Impact: Adoption of the SaaS model has a potential transformational business impact on the way enterprise software solutions are acquired, designed, managed and paid for; it alters the relationship between the enterprise IT organization and software providers, as well as internal business relationships in the IT industry.

SEAS technology offers an additional level of virtualization, effectively similar to the hypervisor virtualization of operating systems. In this case, the application operates as if it had exclusive control over the underlying application server, when, in fact, it shares the instance of the physical application server with other applications, and its exclusive "application server" is only virtual.

The emergence of SEASs will enable a widespread availability of the SaaS model in deploying business applications, making SaaS a prominent option in business decisions regarding enterprise software solutions.

Benefit Rating: Transformational

Market Penetration: Less than 1% of target audience

Maturity: Emerging

Sample Vendors: Cordys; salesforce.com; WebEx

Recommended Reading: "Salesforce.com Challenges Conventional Thinking With Web Application Platform"

"WebEx Extends Its Offerings to Provide a Web Platform"

Grid-Based Application Platforms

Analysis By: Massimo Pezzini

Definition: Grid-based application platforms provide an application runtime environment for service-oriented and event-based transactional or analytical applications. Their aim is to support dynamic scalability and maximum availability on server-based "tera architectures" (typically large clusters of Linux or Windows-based commodity servers). These products leverage advanced workload management, peer-to-peer communication (such as Jini technology), and distributed caching (such as JavaSpaces technology) to dynamically and transparently reallocate computing resources — via software deployment platforms, such as the OSGi Framework— across different application components (as needed) to meet application-specific service-level agreements (such as response time, throughput, scalability or availability).

In addition, grid-based application platforms typically enable hot plug-in and removal of servers without stopping application operations or manual reconfigurations. Most grid-based application platforms support a programming model that combines standard component software — such as Microsoft .NET, Plain Old Java Object, Spring, and, in the future, Enterprise JavaBeans (EJB) 3.0 — and microflow technology (such as BPEL or WS-CDL).

Position and Adoption Speed Justification: Grid-based application platforms were released into the market during the 2003 to 2005 time frame by several small startup companies of little viability. Today, must products are available only in some geographic regions (typically U.S. and/or U.K.) but are rapidly evolving. During the past 12 months, vendors restructured their products' architecture to provide support to technologies, such as Spring and OSGi, improve development, administration and management tools, added event-processing capabilities and extended support to more hardware and software platforms. Several proofs of concept, pilots or development projects started in 2005 and 2006 have reached production deployment, and the installed base is growing. Some user organizations also are using these products to run analytical applications, so grid-based application platform vendors are focusing on high-performance computing and mixed transactional/compute-intensive workloads as additional opportunities for their products. Nevertheless, the grid-based application platforms' installed base is still very small.

Recently introduced support for popular component models, such as .NET and Spring (and potentially EJB 3 in the future) and standard flow languages, will likely increase the appeal of grid-based application platforms by reducing the perception of risk and giving users the opportunity to leverage readily available skills. Still limited system integrator/independent software vendor support, lack of commonly accepted standards and limited market awareness, mostly due to the small size of all the vendors in this space, are other key limiting factors for widespread adoption of these products.

User Advice: Leading-edge organizations looking for platform middleware that provides dynamic scalability and nearly 100% availability to run transactional and analytical applications (as well as hybrid combinations of both styles) should consider grid-based application platforms for their

business-critical deployments. However, they should also remember that these products have few proof points in terms of real-life deployments, vendors are of dubious viability, and claimed total cost of ownership benefits vs. traditional application platform vendors still have to be validated.

By now, adoption of grid-based application platforms should be primarily considered as complementary to traditional application platforms — such as transaction processing monitors (TPMs) or classic (that is, Java EE- or .NET-based) enterprise application servers — but relevant only for specific scenarios in which established platforms are manifestly inadequate, or when the adoption of TPMs or classic enterprise application servers would prove too costly to be considered.

Business Impact: Grid-based application platforms can dramatically change the economics of even the most-extreme transaction processing applications by providing mainframe-like availability and scalability using commodity hardware platforms. Their adoption will enable mainstream organizations to address business scenarios characterized by the most-extreme requirements (that is, real-time risk management, global-scale e-commerce applications and online betting). Until now, these business opportunities could only be exploited by relatively few, technically astute organizations that could afford to adopt expensive and mostly proprietary application platforms (such as mainframe-based TPMs).

Benefit Rating: High

Market Penetration: Less than 1% of target audience

Maturity: Adolescent

Sample Vendors: Appistry; Aumega Networks; GigaSpaces; Majitek; Paremus

Recommended Reading: "The Challenges of Extreme Transaction Processing in a World of Services and Events"

"Extreme Transaction Processing: Technologies to Watch"

Event-Based Application Platforms

Analysis By: Yefim Natis; Massimo Pezzini

Definition: Event-driven application servers are designed for applications that implement an event model as the primary programming model for encapsulation and interaction between software components. This is in contrast with the request/reply remote-procedure-call-style programming model that dominates the prevailing application server, such as Remote Method Invocation of Java EE, .NET Remoting of Microsoft application platform and SOAP-based Web services.

In an EDA style of applications, business components do not call each other directly; they merely post or publish event objects (also called messages or events) to the middleware intermediary layer and listen for event alerts from that same layer. In a request/reply style of applications, components call each other directly, by name. Both models are important, but they are not the same, and their platform middleware requirements are different as well. The differences are substantial enough that a specialized application server, designed for event processing internally, typically has a significant performance and management advantage.

Basic event-driven programming is available as part of the Java EE specifications — Java Messaging Service (JMS) and Message-Driven Beans (MDB) — and Microsoft .NET application platform (Windows Communications Foundation [WCF], Biztalk Server [BTS] and Microsoft

Message Queuing). Previous-generation event-driven application servers (such as BEA Tuxedo and IBM IMS) use EDA internally ("EDA Inside") and are proven performance leaders. New, specialized application servers, based on current software and hardware models, are emerging where internal design and the programming model are optimized for events processing as master internal intercomponent communication.

The event-driven programming model offers greater agility and power in application extension, management and integration. This is because it implements a minimally coupled relationship between software components, and all intercomponent and intersystem communication traffic is *intermediated* by the event-enabling middleware (typically, a form of a message queuing system). Intermediation and encapsulation of the event context in event objects produce opportunities for control, monitoring and analytics. They also support gradual change that isn't intrusive, and the extension and integration of applications (all elements of agility in software). An advanced minority of high-end business software projects benefit from using the event-driven model today, but, in most cases, these projects depend on middleware extensions developed by their own engineering efforts.

Position and Adoption Speed Justification: Application infrastructure vendors that concentrate on telecommunications and radio frequency identification (RFID) are the first to offer event-based application servers. In these business contexts, and in some others (such as stock market monitoring and network monitoring), event processing is the nature of the business. The EDA-style software model can be beneficial in other business situations as well, but it has typically been downplayed in favor of the prevailing and conventional request/reply model.

Early implementations of event-driven application servers are driven by proprietary products and some early-stage emerging standards, such as the Java Community Process JSLEE (JSR 22 and JSR 240) and SIP Servlet (JSR 116 and JSR 289) for telecommunication event processing, or University of Cambridge Electronic Product Code Information Systems (EPCIS) specification for RFID event processing. (Although the initial standards are driven by specialized requirements, the more advanced ones are applicable to all event processing use scenarios.)

Most programmers and system designers are not sufficiently familiar with the event-driven programming model. As the overall understanding of the intrinsic powers of the EDA style of software architecture increases, and the complementary powers of the interactive SOA and EDA (sometimes referred to as event-driven SOA) become more-widely appreciated, and the demand for event-driven application servers will grow.

User Advice: In the telecommunications industry and elsewhere, if event processing is essential to business, then evaluate new event-driven application servers based on JSLEE or other programming models. These platforms are dedicated to event processing; optimized for security, performance, management and the user experience; and specially designed for decoupled event-driven applications.

To maximize the agility of your mainstream applications, investigate and familiarize yourself with the current basic support for event-driven programming in Java EE, .NET or the specialized messaging platforms of some integration suites (such as Tibco BusinessWorks). Expect leading software vendors to introduce and grow their EDA-focused platforms (including middleware, development and management tools) in the next two years.

Understand that event-driven processing will be deployed in addition to, not instead of, the current request-reply-style services. The growing mix of programming models and middleware will require an increased level of system administration tools. Continue investing in strengthening your software systems' control tools and capabilities.

Business Impact: Businesses have become increasingly dependent on the availability and quality of their software services and information assets. This increases the demand for more-agile and cost-effective IT solutions and business models. Many business scenarios in the real world are described as sequences of events, rather than request/reply interactions. Event-based applications see the business process as a relationship and sequence of events. In addition to the continuing support of the request/reply SOA model, the adoption of an event-based model enables closer affinity between software design and business process modeling, narrowing the gap between an enterprise's business and its IT department. This, in turn, enables the enterprise to develop greater insight into its information resources, achieve greater control of its processes and increase its competitive agility. However, continuing innovation and renovation will increase the demand for IT budgets.

Benefit Rating: Transformational

Market Penetration: Less than 1% of target audience

Maturity: Emerging

Sample Vendors: BEA Systems; jNetX; Kabira; OpenCloud; Red Hat (Mobicents); Sybase (iAnywhere); WareLite Limited

Recommended Reading: "Extreme Transaction Processing: Technologies to Watch"

"Predicts 2007: Application Platforms on the Verge of Change"

"Cool Vendors in Platform Middleware, Event-Driven Application Servers, 2006"

"Magic Quadrant for Enterprise Application Servers, 2Q06"

Managed File Transfer

Analysis By: Frank Kenney

Definition: Managed file transfer (MFT) software enables companies to automate, compress, restart, secure, log, analyze and audit the transfer of data from one endpoint to another. These applications support the movement of data over connections, such as HTTP, SMTP, FTP, X.25, systems network architecture (SNA), MQSeries, Bisync or Async, securing these connections with proprietary encryption, digital certificates, pretty good privacy (PGP), secure sockets layer (SSL) or virtual private network (VPN). Although files of any size can be managed, MFT solutions are optimized for the movement of large flat files (that is, IDOC, JPEG, TIFF and spreadsheets) or files containing unstructured data, such as comma-delimited files. Compression, encryption and workflow tools and engines that automate processes that push and pull data in and out of the MFT suite are included in most solutions.

Position and Adoption Speed Justification: Although the management and security functionality of most MFT solutions are mature, some end users are unaware of the offerings in this market. As a result, homegrown solutions using command line interfaces (to trigger connectivity), job schedulers (to automate) and extraction, transformation and loading tools are commonly used, although many users often acknowledge the limitations of these tools when applied in an integration file transfer context. Internal and external security, management and integration requirements, and the need to scale up file transfer projects, will drive many companies to scrutinize and consider replacements to their homegrown solutions.

User Advice: Managing file transfer will become a necessity driven by internal and external mandates, increasing requirements for high-performance data integration, compliance initiatives

and intellectual asset protection. Users should look for this functionality to be offered by existing SOA infrastructure, middleware vendors and service providers.

Business Impact: Organizations with data centers where there are high volumes will continue to be impacted by these technologies but will also benefit from added functionality from these technologies. Companies doing multienterprise collaboration will find that MFT will become more important and will be included in B2B gateway technologies; further consolidating a company's external communications.

Benefit Rating: Moderate

Market Penetration: 1% to 5% of target audience

Maturity: Adolescent

Sample Vendors: Accellion; Axway; Biscom; Cyber-Ark Software; Globalscape; Inovis; Ipswitch; Metastorm; nuBridges; Primeur; Proginet; Radiance; Sterling Commerce; Tumbleweed Communications

B2B Project Outsourcing

Analysis By: Benoit Lheureux

Definition: B2B project outsourcing is an extension of integration as a service (IaaS) and a specific category of IT outsourcing. It is the combination of outsourcing your technical, multienterprise (also called B2B) infrastructure and the outsourcing of your B2B project (the sourcing of people and processes to manage your multienterprise infrastructure). No well-established industry label or acronym for these projects exists, but providers of B2B project outsourcing have used labels such as managed electronic data interchange (EDI), outsourced EDI and EDI SaaS.

Companies are beginning to outsource B2B infrastructure projects regardless of the type of project, but specific multienterprise projects include:

- E-commerce (either buy- or sell-side for manufacturing direct materials procurement or retail/consumer packaged goods procurement)
- ERP extension projects (when connecting your ERP or other internal applications to external business partners)
- B2B consolidation projects (to combine multiple B2B projects onto one infrastructure)
- SOA extension (similar to ERP but for connecting your internal SOA infrastructure to external business partners)

B2B project outsourcing generally includes the following high-level components:

- One-time implementation of multienterprise IT infrastructure (on-premise or on-demand) — in many cases, the B2B infrastructure is a multitenant IT stack supporting multiple B2B projects and typically also used by the integration service provider or system integrator to support a separate IaaS offering; if so, then this is a differentiator and gives the vendor some economy of scale
- One-time implementation of functionality (for example, documents, maps, services and processes)
- One-time provisioning of connections to trading partners and external service providers

- Ongoing B2B infrastructure operations, reporting and change management

Although some vendors, such as GXS and Sterling Commerce, offer B2B project outsourcing as a stand-alone offering; others, such as Atos Origin, Capgemini, EDS and IBM, offer B2B project outsourcing more typically in the context of larger outsourcing projects, such as business process outsourcing.

Position and Adoption Speed Justification: IT vendors have offered various forms of B2B project outsourcing for years, but the recent growth of multienterprise projects (see "Multienterprise Integration Projects Will Proliferate and Innovate" 143694) is driving more companies to reconsider their B2B strategies, including whether to implement multienterprise infrastructure themselves or to outsource the task.

Although many companies successfully brought B2B projects "in-house" via B2B software during the decline of the EDI value-added networks in the early 2000s, the need to scale these projects up to do more B2B, combined with increasing pressure on the IT organization to outsource noncritical core competencies, is driving a selective backlash against in-house B2B and increased interest in the outsourced approach.

Also, vendors are doing a better job of creating bundled IT-outsourcing offerings that more consistently and clearly combine the right services (such as multienterprise communications, in-line translation, community ramp-up and ongoing project management) into fixed-price offerings. Within a year or so, expectations for outsourced B2B will peak as companies look for relief, and this will lead to a subsequent slide into the Trough of Disillusionment as the complexity of B2B — and the challenges of outsourcing it, including customization (for example, to meet unique B2B project requirements) — becomes better-understood.

User Advice: Like internal integration, multienterprise integration is a complex task, and much of the intellectual property associated with your B2B projects will often be "sticky" and, thus, will not be easy to transfer to another provider or back in-house, so select your B2B project outsourcing vendors carefully, treating them as strategic technology partners.

Vendor viability will be particularly important for larger projects (involving hundreds or thousands of external business partners) with a five-year or longer life span. Although complex projects may require custom implementations and quotes, prospects should consider vendors that manage costs by leveraging a multitenant B2B infrastructure implementation (rather than custom deployments) and that offer unit pricing for one-time and recurring fees (rather than custom quotes).

Business Impact: Although many companies will choose to implement their own B2B projects, the alternative — B2B project outsourcing — is a potential benefit for nearly all companies small and large across all industries. Multienterprise integration and interoperability — like internal integration and interoperability — are ubiquitously pervasive IT requirements (see "Multienterprise Integration Projects Will Proliferate and Innovate").

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Adolescent

Sample Vendors: Accenture; Atos Origin; crossgate; Capgemini; DIcentral; eBRIDGE Software; E2open; EDS; GXS; Hubspan; IBM; Inovis; Internet Commerce; RedTail Solutions; SPS Commerce; Sterling Commerce; TietoEnator

Recommended Reading: "Taxonomy and Definitions for the Multienterprise/B2B Infrastructure Market"

"Multienterprise Integration Projects Will Proliferate and Innovate"

Integration and Infrastructure Appliances

Analysis By: Frank Kenney

Definition: Integration middleware vendors have become better at shifting the costs associated with complexity by preconfiguring adapters, predefining processes and events, and premapping transformations — and packaged all of this into an easy-to-install, lower-operational total cost of ownership hardware appliance. Although the complexity of integration is not completely eliminated, the user is at least partially shielded from some of its complexity. This approach has led to list-price reduction and a reduced need for professional services for solutions to an increasing number of integration problems. In meeting this challenge of feature/functionality vs. complexity/cost, integration middleware vendors with appliance-based solutions are providing alternative, desirable pricing models and sales channels, sometimes without the added revenue benefit of professional services.

Although there remains a substantial market for "high end" integration technology, vendors cannot ignore the opportunity of revenue generation by non-Fortune 500 enterprises. Potential users of this technology are custom coding integrations, and they will benefit from the packaged integrations and processes offered by integration appliances, even if the functionality is somewhat "hard wired" and lacks some of the flexibility and agility of more-traditional software solutions.

Position and Adoption Speed Justification: Many integration and infrastructure appliance vendors have enhanced their technologies and transformed their message to support one of three emerging scenarios. These scenarios include:

- SOA policy management
- Appliances for packaged integration and processes
- Highly specialized functionality in one specific discipline, such as security or management

Because the adoption of "all things" SOA (for example, SOA infrastructure, service-oriented development of applications, service-oriented business applications and SOA governance), the need for and use of SOA policy management technology will continue to grow. Many of the appliance vendors with SOA policy management technologies have been acquired by larger, more-influential vendors (Cisco/Reactivity, IBM/Datapower and Intel/Sarvega) that can leverage their sales channels and installed bases to increase visibility and usage.

Other vendors offer quick tactical integration with SaaS providers, such as salesforce.com or appliance-based solutions for multienterprise collaboration. Finally, some vendors are extending their performance and security capabilities, targeting service providers and telecommunications vendors offering application and infrastructure hosting.

User Advice: Although integration and infrastructure appliances will peak, go through deflated expectations and eventually reach the Plateau of Productivity, companies should consider the use of these products today to solve specific integration problems, and for small-to-midsize SOA policy management, prepackaged integration and specialized integration requirements (for example, security and management). Appliances can help reduce the complexity of integration projects and more-quickly help companies meet SOA governance needs. These benefits can

overshadow the various challenges of appliances, including the proliferation of point-to-point integration.

Business Impact: Any part of the organization that is looking to provide a set of managed services, such as multienterprise collaboration, expense reporting, logistics or enterprise resource planning, will benefit from the use of some appliances as they allow applications not normally meant for multitennet use to be used that way. Some line managers, such as sales or customer support, will find appliances beneficial, in that integration with SaaS providers is made less complex; thus easier, cheaper and quicker.

Benefit Rating: Moderate

Market Penetration: Less than 1% of target audience

Maturity: Emerging

Sample Vendors: Cast Iron Systems; Cisco Systems; Covast; DataPower; Extol; Forum Systems; Layer 7 Technologies; Solace Systems; Vordel

Recommended Reading: "Key Issues for Integration and Infrastructure Appliances"

"Cisco's Plans to Buy Reactivity Raise Questions About AON"

"No 'Leader' Exists in SOA Governance ... At Least Not Yet"

Distributed Caching Platforms

Analysis By: Massimo Pezzini

Definition: The goal of distributed caching platforms (also called "data fabrics" or "information fabrics" by vendors) is to dramatically improve performance and scalability of data-intensive applications by keeping critical data as close as possible to the application (that is, "in memory"). These platforms provide an in-memory store (the cache) where applications can place and manipulate data (or objects) asynchronously fetched from a variety of data sources, such as relational DBMSs (RDBMSs), real-time data feeds, XML documents, asynchronous messages or application programming interfaces (APIs) via adapters. Typically, application programs manipulate objects in the cache via specific or generic (for example, Java Database Connectivity and JMS) APIs, but some products can transparently plug beneath an application through declarative properties associated to data structures that are meant to be shared through the cache.

The distributed caching platform is responsible for initially loading the cache, synchronizing the cache state with original data sources, locking cache objects, managing transactions and sending cache event notifications. Caches can often be distributed across multiple physical servers to further boost performance or for protecting data integrity from system failures through replication of critical data. Therefore, the platform provides for clustering and failover management, as well as cache mirroring and partitioning, and it supports the necessary security (to protect data for unauthorized access) and management features. Distributed caching platforms plug into Java Platform, Enterprise Edition (Java EE) application servers, .NET or other platforms, and some of them enable sharing of cached objects across multiple platforms (for example, Java EE and .NET) through the distributed cache.

Caching is a technique extensively used in DBMS, Web servers, application servers and other system software. But, in these cases, it is typically transparent for application developers, is specialized to supporting only certain types of data (for example, Web pages or HTTP sessions) and doesn't implement the clustering, replication, partitioning and integrity features provided by

distributed caching platforms. Although these products are frequently compared to in-memory DBMSs, they do not offer the full support for relational semantic provided by products such as Oracle's TimesTen and other similar in-memory RDBMSs.

Position and Adoption Speed Justification: The 64-bit hardware architecture and the availability of inexpensive memory are paving the way for mainstream use of distributed caching systems. Numerous commercial and open-source products are available in the market (some have been available for several years), and vendors' revenue and installed bases are growing fast. During the past 12 months, a few vendors exited the market, whereas the more-established products have matured in functionality by adding support for heterogeneous platforms, "stored procedure like" capabilities, and by improving manageability and high availability. Large application infrastructure vendors have turned their attention to this technology through internal developments, partnerships or acquisitions (Oracle bought Tangosol). Several independent software vendors bundle these products in tools and packaged applications.

A growing number of organizations are using this technology for production systems. In many cases, they are supporting business-critical applications in industries, such as telecommunications, financial services, automotive, e-commerce and defense. Distributed caching platform adoption will continue to grow because of push from large middleware vendors, the emergence of open-source products and through bundling into larger software stacks, such as platform middleware (for example, event-driven or grid-based application platforms), enterprise service buses, BPM products, business intelligence tools and packaged applications. Technology is rapidly maturing, but usage is still limited to leading-edge user organizations. The main limiting factors for more-widespread adoption of these products include:

- The small size of some of the leading vendors
- Lack of commonly agreed standards
- Limited system integrators' support and complexity in deployment, configuration and management

User Advice: Mainstream organizations looking for minimally invasive solutions to boost performance and scalability of data-intensive .NET or Java EE-based applications, as well as to implement loosely coupled, high-performance application interoperability, should consider distributed caching platforms. Despite their technical complexity, they are proven products with significant production deployments. They can be used as complementary extensions of mainstream platform middleware to improve established application performance and as platform foundations for new deployments.

However, users should also take into account that only a few mature and proven products are available from large software vendors and from small, albeit fast-growing, specialized companies. Most products, including from large, established software vendors, are immature and/or only partially proven.

Business Impact: Distributed caching platforms enable organizations to retrofit, with minimal impact, established applications to increase their performance and scalability. Thus, organizations can enhance the productivity of their business processes, improve customer/supplier/employee satisfaction and extend application reach to larger user constituencies while protecting investments in application assets.

In combination with Java EE or .NET platforms, distributed cache systems enable organizations to develop distributed, ultra-high-end transactional applications (for example, global-class e-commerce, real-time access to large databases and online trading) that cannot be otherwise

supported by traditional platforms alone, thus enabling organizations to explore new business opportunities while protecting users' platform middleware purchases and skills.

Benefit Rating: Moderate

Market Penetration: 1% to 5% of target audience

Maturity: Early mainstream

Sample Vendors: Alachisoft; GemStone Systems; GigaSpaces; IBM; JBoss; Oracle; Terracotta

Recommended Reading: "The Challenges of Extreme Transaction Processing in a World of Services and Events"

"Extreme Transaction Processing: Technologies to Watch"

"Oracle to Acquire a Lead in Extreme Transaction Processing"

SOA Governance Organization

Analysis By: Paolo Malinverno

Definition: SOA governance identifies responsibility and decision-making authority for:

- Defining or modifying the business processes that will be supported with SOA techniques
- Defining, designing, accessing, executing, operating and maintaining reusable services
- Identifying the service levels (including performance) required and the access rights
- Determining service ownership and cost allocation
- Setting policies, responsibilities and rules for all the above
- Measuring use and compliance, and specifying incentives to promote the adoption of those rules

Position and Adoption Speed Justification: SOA is becoming increasingly widespread and will form a major part of application portfolios by 2010. As organizations progressively grow their post-pilot SOA projects, they face two major challenges:

- Extending the initial pilot technology infrastructure to support the SOA (often starting with Web services, but extending to guarantee the required levels of performance, scalability, reliability and connection into legacy, non-Web-services-enabled applications, dynamic discovery of services, routing of service invocations, mediation or transformation of service interfaces, load balancing, failover management, security and more).
- Growing their SOAs with discipline by enforcing service reuse and avoiding service duplication. This is only possible through carefully designed and consistently enforced governance procedures.

In this context, SOA governance focuses on the second challenge: underlining the importance of governance in a midsize to large SOA project, and the role an ICC (see the Integration Competency Centers section) plays in it. The various components of an ICC play a fundamental role in the definition of high-level, highly reusable services, and in the governance processes associated with making a successful SOA project.

User Advice:

- Make the complexity of governance arrangements proportional to company size. There is no need to define complex processes when it's easy and effective to get the main stakeholders all together in a single face-to-face meeting.
- Ensure that you have appropriate executive-level buy-in and a dependable executive sponsor, mainly by communicating the importance of mitigating project risks due to the lack of SOA governance.
- You need "just enough" governance: too little governance or too much governance will kill your SOA project.
- Identify where you need to go and be realistic: focus on a few goals, desirable behaviors, metrics and mechanisms.
- There's no one-size-fits-all governance mechanism set: different mixtures of mechanisms work best in different companies.

Business Impact: With the widespread adoption of SOA, the challenges associated with SOA projects are emerging. Through 2010, the main barriers to SOA adoption will be nontechnical issues of inadequate governance, lack of clear value metrics, poorly defined requirements and scope, and insufficient business involvement in project prioritization and service identification. The bigger the SOA, the more governance it needs, and the more complex the governance roles and mechanisms must be. Governance arrangements take a long time to design and install, and are difficult to enforce; but without them, every SOA project out of the pilot phase is at risk (see "Gartner Research Index on SOA Governance"). Service reuse is only possible through carefully designed and consistently enforced governance procedures.

Consequently, SOA governance offerings have proliferated in the market in two forms:

- *SOA governance technology* (see the SOA Governance Technologies section) built on policy management — be aware that most of the self-defining SOA governance technology offerings address only a narrow set of design policies, missing the wider SOA governance organizational picture.
- *SOA governance consulting services,* built on general service-identification methodologies and around the concept of an SOA center of excellence — an ICC empowered by strong governance processes (see the Integration Competency Centers section).

Strategic Planning Assumption: Through 2010, lack of working SOA governance arrangements will be the most common reason for SOA failure (0.8 probability).

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Emerging

Sample Vendors: HP Services; IBM Global Business Services; Oracle Consulting

Recommended Reading: "Service-Oriented Architecture Craves Governance"

"Gartner Research Index on SOA Governance"

"Sample Governance Mechanisms for a Service-Oriented Architecture"

Event-Driven Architecture

Analysis By: Roy Schulte; Yefim Natis

Definition: EDA is a subset of the more general topic of event processing. EDA is an architectural style in which some of the elements of the application execute in response to the arrival of event objects. An element decides whether to act and how to act based on the incoming event objects. In EDA, the event objects are delivered in messages that do not specify any method name (such messages are called event notifications).The event source does not tell the event receiver what operation to perform. An event is something that happens (or does not happen, but was expected or thought possible). Examples include a stock trade, customer order, address change, and a shipment arriving or failing to arrive (under specified conditions). An event may be documented in software by creating an event object (sometimes called plain "event," which then is a second meaning for the term). An event (object) represents or records a happening ("ordinary") event. Examples of event objects include a message from a financial data feed (a stock tick), an XML document containing an order or a database row. In casual discussion, programmers often call the message that conveys an event object an "event."

Position and Adoption Speed Justification: Computer systems have used event processing in many different ways for decades. Event processing is moving through the Hype Cycle now because its concepts are being applied more broadly and on a higher level. Business events, such as purchase orders, address changes, payments, credit card transactions or Web "clicks" are being used as a focus in application design. This contrasts to past treatments of events where business applications addressed events more indirectly, and event modeling was considered to be secondary to data modeling, object modeling and process modeling. Businesses have always been real-time, event-driven systems, but now more aspects of their application systems are also real-time systems. EDA concepts are also used on a technical level to make application servers and other software more-efficient and scalable. The spread of other types of service-oriented architecture (SOA; conventional, request/reply SOA) is also helping to pave the way for EDA because some of the concepts, middleware tools and organizational strategies are the same.

User Advice: In an era of accelerating business processes, pervasive computing and exploding data volumes, companies must master event processing if they are to thrive. Companies should use event processing in two ways: 1) to engineer more-flexible application software through the use of message-driven processing, and 2) to gain better insight into current business conditions through complex-event processing (CEP). Architects can use available methodologies and tools to build good EDA applications, but must consciously impose an explicit focus on events because standard methodologies and tools do not yet make events first-class citizens in the development process. Companies should implement EDA as part of their SOA strategy because many of the same middleware tools and organizational techniques (such as using an SOA center of excellence for EDA and for other kinds of SOA) apply. Companies should not implement request/reply SOA now and wait for one or two years to implement EDA SOA because a request/reply-only SOA strategy will not be able to support some business requirements well.

Business Impact: EDA is relevant in every industry. Large companies experience literally trillions of ordinary business events every day, although only a minority of these are represented as event objects, and only a tiny minority of those event objects are fully exploited for their maximum information value. The number and size of event streams are growing as the cost of computing and networking continues to drop. Companies now generate data on events that were never reported in the past. The CEP type of business EDA was first used in financial trading, energy trading, supply chain management, fraud detection, homeland security, telecommunications, customer contact centers, logistics and sensor networks, such as those based on RFID. Event processing is a key enabler in business activity monitoring, which makes business operations more visible to end users.

Benefit Rating: Transformational

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Sample Vendors: Actimize; Agent Logic; Agentis Software; Aleri; Avaya; Axeda; BEA Systems; coral8; Cordys; Event Zero; Exegy; firstRain; IBM; jNetX; Kabira; Kx Systems; open cloud; Oracle; Progress Software/Apama; Red Hat (Mobicents); Rhysome; SAP; SeeWhy; StreamBase Systems; Sun; Sybase; Syndera; Synthean; Systar SA; Tibco Software; Truviso; Vayusphere; Vhayu; Vitria Technology; WareLite

Metadata Repositories

Analysis By: Michael Blechar; Jess Thompson

Definition: Metadata is an abstracted level of information about the characteristics of an artifact, such as its name, location, perceived importance, quality or value to the organization, and relationship to other artifacts. Technologies called "metadata repositories" are used to document, manage and perform analysis (such as change impact analysis and gap analysis) on metadata in the form of artifacts representing assets that the enterprise wants to manage. Repositories cover a wide spectrum of metadata/artifacts, such as those related to business processes, components, data/information, frameworks, hardware, organizational structure, services and software in support of focus areas like application development, data architecture, data warehousing and enterprise architecture.

Position and Adoption Speed Justification: Most organizations that have tried to implement a single enterprise metadata repository have failed to meet the expected return on investment. Community-based repositories supporting business process modeling and analysis, SOA and data integration have shown benefits in improved quality and productivity through an improved understanding of the artifacts, impact queries and the reuse of assets such as services and components. For the near future, there will be no proved, viable solution that federates multiple metadata repositories (or federates repositories with other technologies that contain metadata, like service registries holding runtime metadata artifacts) sufficiently to satisfy the needs of organizations.

Mainstream IT organizations will find that the most pragmatic approach to metadata management and reporting is to have multiple, community-based repositories, which have some degree of federation and synchronization. Although it is possible to create federated queries across multiple repositories, many organizations still may want to consolidate and aggregate selected metadata information from disparate sources into a "metadata warehouse" for ease of reporting and for ad hoc query purposes. Leading metadata repository vendors are well-positioned to meet this need, but competitors will emerge, including large ISVs that will look to provide these capabilities in their tool suites. Large vendors, such as IBM, Oracle and SAP, are adding repositories — or are improving their repository support for design-time and runtime platforms — to enhance metadata management support for their development and deployment environment. As a result, Gartner expects to see a broader degree of acceptance by customers, along with a consolidation in this market during the next few years. We position metadata repositories as being two to five years from plateau, because most Global 1000 companies have purchased metadata repositories and are not yet aggressively seeking replacements, and because most new buyers are less-sophisticated IT organizations looking to large ISVs to improve their federation capabilities before committing to the new tools. As a result, most repository purchases will be tactical in nature based on the needs of specific communities, such as data warehousing and SOAs.

User Advice: Owing to the diversification and consolidation of metadata management solutions, the enterprise uber-repository market no longer exists. Consider the acquisition or extension of using a metadata repository as part of moving to SOAs, or consider implementing business process management, data architecture, data warehousing and enterprise architecture initiatives. Most organizations will be best-served by living with metadata in multiple tools or by using different repositories based on communities of interest, with some limited bridging or synchronization to promote the reuse and leveraging of knowledge and effort. Organizations that need to approximate the capabilities of an enterprise metadata repository are still best-served by solutions from leading repository vendors.

Business Impact: Metadata repository technology can be applied to aspects of business, enterprise, information and technical architectures, including the portfolio management and cataloging of software services and components; business models; data-warehousing extraction, transformation and loading rules; business intelligence transformations and queries; data architecture; electronic data interchange; and outsourcing engagements.

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Sample Vendors: Allen Systems Group; BEA Systems; LogicLibrary

Recommended Reading: "Are Federated Metadata Approaches to Business Service Repositories Valid?"

"Best Practices for Metadata Management"

"Metadata Management Technology Integration Cautions and Considerations"

"Metadata Repositories Address Disparate Sets of Needs"

"The Evolving Metadata Repository Market"

Service Component Architecture

Analysis By: Jess Thompson

Definition: Service component architecture (SCA) is a collection of specifications (proposed and published) by a consortium of vendors. The specifications use a model-based approach for capturing a wide set of metadata that will simplify the deployment and management of services. The specifications address two major aspects of SOA:

- The assembly of sets of components to build business applications through the wiring of service references to services
- The implementation of components that provide and consume services

SCA references Service Data Objects (SDO), which defines core infrastructure application programming interfaces (APIs) for heterogeneous data access that supports common application design patterns and higher-level tools and frameworks (such as SCA). SDO is an approved Java Specification Request (JSR) that is being worked on in the Java community, and where a few early implementations already exist. Information about the JSR can be found at www.jcp.org/en/jsr/detail?id=235, and more information about the SCA proposal can be found at www-128.ibm.com/developerworks/library/specification/ws-sca/.

On 17 April 2007, the consortium driving SCA/SDO turned over the reins to the newly formed OASIS Open Composite Services Architecture (Open CSA) Member section. Open CSA's goal is to advance the SCA and SDO families of specifications, which will simplify the development of applications having SOA.

Position and Adoption Speed Justification: SCA is important because of the spectrum of powerful vendors that have contributed to and endorsed it. However, because it's a new specification and not yet a standard, vendors have not provided tooling and SCA is not yet being used. Several of the vendors in the initial SCA consortium have disclosed plans to release initial tooling during 2007.

User Advice: The implementation and management of SOA will become more systematic than it is today. SCA does not eliminate the need to design and develop service components. However, it does introduce a set of concepts and terminology that captures vital metadata, such as interface definitions and dependencies among services (created when one service accesses another service), which organizations must begin to use during implementation and management processes. We believe that SCA will become an approved OASIS standard within 18 to 24 months and will be used across the majority of toolsets for building SOAs.

As SCA-based tools become available, users should employ these tools, whenever practical, as mechanisms to develop what will become (in about two years) requisite skill sets and processes for building, deploying and managing composite applications.

Business Impact: Although vendors are interested in the potential for interoperability that comes from the SCA model, the total tool profile will require vendors to coordinate many things, some of which are outside the scope of SCA. User expectation should be that there will be export/import capability for metadata, and the artifacts from one vendor's tooling will interact with artifacts produced by another vendor's tooling. However, users should not expect complete artifact portability — that is, users may not be able to manipulate one tool's artifacts using tools from another vendor. SCA will simplify the creation and management of services and composite applications, which are critical to the overall success of SOAs. The commonality of skill sets and best practices associated with SCA will become pervasive during 2009.

Benefit Rating: Moderate

Market Penetration: Less than 1% of target audience

Maturity: Embryonic

Sample Vendors: BEA Systems; IBM; Iona; Oracle; SAP; Siebel Systems; Sybase

At the Peak

Packaged Integration

Analysis By: Benoit Lheureux; Frank Kenney

Definition: Packaged integration refers to any form of integration product (regular licensed software, open-source software or appliance-based solution) that includes all the technical artifacts (such as adapters, transformation maps, routing rules, business process definitions and business object schemas) needed to support a specific, well-defined and recurring integration scenario between two or more application systems, within or among companies (for example, synchronizing orders between an order entry and a fulfillment system, or synchronizing customer data between sales automation functionality hosted by salesforce.com and your internal CRM systems).

Packaged integration logically consists of two forms: packaged integrating processes (PIPs), which mainly provide integration logic, and packaged composite applications (PCAs), which include business logic covering a specific application need. PIPs and PCAs can be considered "integrating applications" that are pre-built to solve a typical problem using an embedded combination of integration and platform middleware. Vendors use labels for packaged integration, which include:

- Packaged integration
- Integrating applications
- Industry solutions (or sometimes just "solutions")
- Integration accelerators (or sometimes just "accelerators")
- Process templates (or sometimes just "templates")
- Other analogous terms

Position and Adoption Speed Justification: For years, integration vendors and system integrators have leveraged PIPs and PCAs primarily as accelerators for custom projects, but not in the form of packaged, licensed software products. However, in the past few years, IT vendors have begun marketing and selling packaged integration for internal and multienterprise scenarios. Packaged integration solutions are available from a range of IT vendors, including those with PIPs for internal use (such as Impress Software, Oracle, SAP and Sierra Atlantic); those with PCAs for internal use (such as Cordys, Portrait Software, Oracle, Optura, SAP and Vitria); those with PIPs licensed as regular software for multienterprise scenarios (such as eBridge Software, Extol and Traxian); and integration service providers that offer PIPs and PCAs as a hosted service (such as ADX, GXS, Hubspan, Inovis, RedTail Solutions, SPS Commerce and Sterling Commerce, which each offer some form of pre-built order process visibility PCA, as well as PIPs to accelerate the integration of companies into networks running specific applications).

In multienterprise scenarios, PIPs and PCAs often solve specific process integration problems in industries — for example, PIPs for data synchronization and PCAs for order visibility in retail/consumer packaged goods, or PIPs for vendor-managed inventory and PCAs for supply chain visibility in high-technology manufacturing. Although packaged integration is beginning to proliferate, in most cases (more than 75% of the time), no packaged solution is available to address the complete integration project requirements. Even in cases where packaged integration *is* available, inconsistent nomenclature and general unfamiliarity conspire to confuse the IT public or, worse, render companies unaware that a solution may already exist to solve their integration problems.

However, packaged integration solutions will continue to proliferate because:

- Packaged integration can be developed using almost any combination of platform and integration middleware.
- Many IT vendors are likely to discover recurring integration problems as they service their respective customer bases.
- Independent software vendors will use service-oriented-architecture composite application frameworks from leading IT vendors (such as IBM, Microsoft, Oracle and SAP) and others (such as iWay Software, Persistence Software and salesforce.com) to build even more PIPs and PCAs.

The IT public will become increasingly aware of this option, and packaged integration will increase in popularity. Soon after, it will experience a peak of inflated expectations, tempered shortly afterward by the realization of some of the inherent challenges associated with the approach, including the need for constant maintenance as associated applications continue to change and evolve.

User Advice: For new process integration or composite application projects, companies should determine whether packaged integration is available and, if so, at least consider it as an alternative to implementing the project from scratch. Although the availability of packaged integration doesn't guarantee that it's the correct approach for any project, we believe that, in many cases, packaged integration can lower overall project costs, simplify deployment and reduce time-to-market times for IT projects. Some pertinent issues to discuss with vendors include:

- Licensing and maintenance costs
- The completeness of the packaged integration solution (which can range from 100% complete, which is rare, to 70% to 80% solutions (common), to something less than 50% complete — often positioned as templates or starter applications — as in the case of applications than need extensive customization); generally, we consider anything less than 70% complete to fall short of the spirit implied in the notion of a package for "packaged integration process" or "packaged composite application"
- The latency for the availability of, and the procedure for, installing packaged integration upgrades to correspond with upgrades as they're released for associated target applications

Business Impact: Packaged integration will continue to proliferate, and often reduce the cost and time of deploying standardized integration applications and composite applications. Although some customization is typically required, PIPs and PCAs can make process integration and composite applications more cost-effective and acceptable to companies with specific projects (when buying vs. building is an option). PIPs and PCAs will also favor the adoption of modern application integration, as well as the associated business benefits, by midsize enterprises that don't have the skills to carry out custom integration developments.

Benefit Rating: Moderate

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Sample Vendors: Cordys; eBridge Software; Extol; GXS; Hubspan; iWay Software; IBM; Impress Software; Optura (Open Text); Oracle; Passport; Persistence Software; RedTail Solutions; SAP; SEEC; Sierra Atlantic; SigmaTrak; SPS Commerce; Traxian; Vitria

Recommended Reading: "Packaged Integrating Processes Accelerate Integration"

"Packaged Composite Applications: Applications That Integrate"

"Microsoft, GXS and Covast Will Automate E-Commerce Integration"

SOA Governance Technologies

Analysis By: Frank Kenney

Definition: The key to being successful with your SOA projects is to understand and control your SOA artifacts. SOA artifacts can include services, SOA policies — that is, service-level

agreements (SLAs) — business processes and profiles of consumers and providers. The key to understanding and controlling these artifacts is SOA governance. Various technologies can help you control how your artifacts are being used, managed, secured and tested, as well as how visible they are. These technologies include:

- SOA policy management provides the technology to create, discover, reference and sometimes enforce policies related to SOA artifacts, such as access control, performance and service levels.
- SOA registries and repositories help manage metadata related to SOA artifacts (for example, services, policies, processes and profiles) and have recently evolved to include the creation and documentation of the relationships (that is, configurations and dependencies) between various metadata and artifacts.
- SOA quality assurance and validation technologies validate the individual SOA artifacts, and determine the relationships to each other within the context of an SOA deployment. For example, these technologies will test and validate a composite service that executes specific processes, while having specific policies enforced on it.
- Monitoring is present throughout the individual technical domains and enables companies to study an SOA and its environment and provide deeper, real-time business intelligence and analytics applications. It also helps them checking that the various governance processes are actually followed. Business activity monitoring (BAM; see "MarketScope for Business Activity Monitoring Platforms, 3Q06") plays a key role in the evolution and agility of an SOA and is the foundation for future complex event processing scenarios as the SOA life cycle (a cycle of developing, testing, deploying, monitoring, analyzing and refining).
- Adapters, interfaces, application program interfaces and interoperability standards enable all the technical domains to communicate and share information, as well as enable the governance suite to be integrated with existing infrastructure applications, such as business applications, integration middleware or operating systems for optimal policy definition and executions.

Position and Adoption Speed Justification: SOA governance technologies, specifically the service registry, and SOA policy enforcement (service management and service security) have been hyped by vendors and end users; many end users are deploying these technologies without credible SOA governance organizational processes and strategies. (As a result, service registries and policy enforcement tools are often underused today (only for cataloging and XML security). With more vendors entering into OEM agreements and partnerships with best-of-breed vendors, these technologies will reach the Peak of Inflated Expectations within 12 months. However, because most SOA deployments will likely fail without proper governance, companies will eventually move to better leverage SOA governance technologies to provide visibility, manageability, monitoring security and quality assurance.

User Advice: Regardless of the overhyping of SOA governance, companies deploying SOAs need to first develop a strategy and process for SOA governance that encompass technologies and organizations. Deploying a service registry for reuse and developing some policies around the development of services is a good start, but companies should plan on using that registry for SOA life cycle management and for visibility into various SOA artifacts.

Business Impact: Any company or division deploying an SOA will be impacted by SOA governance. Entities providing software as a service, integration as a service, business-to-business services or hosting applications should take advantage of SOA governance

technologies to enhance their offerings, better manage their SOA artifacts and obtain competitive differentiation.

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Sample Vendors: Actional; AmberPoint; BEA; HP-Mercury; iTKO; IBM; Layer 7 Technologies; LogicLibrary; Oracle; Reactivity; Software AG/webMethods; SOA Software; Tibco Software; Vordel; WebLayers

Recommended Reading: "Criteria for Evaluating a Vendor's SOA Governance Strategy"

"No 'Leader' Exists in SOA Governance…At Least Not Yet"

Vocabulary-Based Transformation

Analysis By: Jess Thompson

Definition: Vocabulary-based transformation is an integration feature that leverages a conceptual vocabulary that establishes a standard terminology for enterprise data assets. The vocabulary contains specifications of business entities (for example, customer) that consist of a set of elementary and composite terms, relationships and constraints and is sometimes implemented in the form of a taxonomy" or "ontology." The advantage of vocabulary-based transformation is that it reduces the number of transformations required to move data among databases.

Position and Adoption Speed Justification: Vocabulary-based transformation is offered in commercial products, but it is used in fewer than 5% of integration scenarios. It is particularly useful when reusable canonical messages organized around business entities, such as purchase orders, are defined to convey information to multiple underlying application systems. This enables companies to eliminate point-to-point translation between pairwise application interfaces resulting in an overall reduction of development and maintenance efforts. This number will increase as organizations recognize the importance of semantic reconciliation for creating base-level services for SOAs. Standards for semantic reconciliation — including the World Wide Web Consortium's Web Ontology Language (OWL) and Resource Description Framework, and the Object Management Group's Semantics of Business Vocabulary and Business Rules (SVBR) — have limited adoption, but they offer potential benefits. Some vocabularies exist for specific vertical industries — for example, Health Level Seven (HL7), the Agency for Co-operation and Research in Development (ACORD), and the Society for Worldwide Interbank Financial Telecommunication (SWIFT). The applicability and benefit of these standards are focused on their respective vertical industries, but substantial benefits are possible when used in those areas.

User Advice: Organizations in vertical industries where standard vocabularies exist are strongly advised to adopt those vocabularies. This will enable organizations to align the concepts and terms used in conjunction with integration and SOA projects with those occurring in other organizations within their vertical industries. Organizations for which no standard vocabularies exist should consider the use of horizontal standards, such as that provided by the Open Applications Group, as a starting point to simplify the transformation that is a part of integration and SOA projects. Use vocabulary-based transformation when information about business entities must be conveyed among multiple application systems. Use point-to-point translation (not vocabulary-based transformation) when information conveyed by the interface is unique to the two participating applications.

Business Impact: Vocabulary-based semantic transformation can reduce the development time needed to implement the majority of application integration patterns, as well as the time required to establish foundational services for SOA initiatives.

Benefit Rating: Moderate

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Sample Vendors: Business Objects; Contivo; enLeague Systems; IBM; Informatica; Sun; Tibco Software; Unicorn; Vitria; webMethods

Business Activity Monitoring

Analysis By: Bill Gassman

Definition: BAM describes the processes and technologies that provide event-driven, real-time access to and analysis of critical business performance indicators. BAM is used to improve the speed and effectiveness of business operations by raising awareness about issues as soon as they can be detected. BAM applications issue alerts about a business opportunity or problem and, in most cases, drive a dashboard with metrics, historical information, an event log and drill-down features to help the business operations staff process them. The processing logic of a BAM system may use simple stream or complex event processing .

Position and Adoption Speed Justification: BAM has crossed over the hump of the Hype Cycle, and although growth in the market is steady, marketing will be done with less fanfare than before. There is no single BAM market. The hype is distributed across multiple areas, which makes it appear less exciting than it really is. Looking across all avenues of deployment, there are signs that there will be more deployments of BAM applications this year than last. Application vendors, including those selling SaaS, are building real-time metrics into their products. Business process management vendors continue to partner with BAM vendors or build process-monitoring features, a form of BAM, into their products. Enterprise service bus vendors continue to enhance and promote BAM. Business intelligence (BI) vendors are waking up to the topic. For example, in January 2007, Cognos acquired Celequest, one of the pure-play BAM vendors. The remaining pure-play vendors are growing, although most business will multifaceted vendors. There is awareness of BAM in the IT community, but business users need to drive the projects and many are still unaware of the possibilities or value of BAM. In addition, there is terminology confusion. Other marketing terms, such as operational BI and real-time BI, are blending with the perception of what BAM offers. The vendors building complex event-processing engines are enjoying market growth and will likely add dashboards and human-alerting features, overlapping with and becoming part of the BAM market. In addition, we are seeing enterprises build BAM applications out of BI tools and other piece parts. We believe that three to five years is a realistic, although perhaps a bit optimistic, expectation for the market to reach the plateau, which is defined as more than 20% adoption by the target market.

User Advice: Begin adoption now to gain experience. Start with simple projects, or expand the use of BAM if early projects have been successful. Allow time for resources to learn to trust the system and to take action based on real-time alerts. Match current and future system performance requirements with product capacity. Depending on specific industry or process needs, look for BAM products that include specialized knowledge, such as supply chain, check clearing, compliance monitoring or fraud detection. Promote success and share best practices with other groups in the enterprise. Products will come and go, but it takes time to build a culture that can understand how to use real-time alerts and information in its processes.

Business Impact: BAM provides real-time situational awareness and detects anomalies in the processes of supply chain operations, event-based marketing, B2B value-added networks, compliance activities and orchestrated business processes. Anywhere an enterprise has a time-sensitive business process, automated or manual, it can deploy BAM to better understand, monitor and generate alerts when problems or opportunities arise.

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Sample Vendors: Cognos; Information Builders; Microsoft; Oracle; Progress Software/Apama; SeeBeyond; SeeWhy; Software AG; Syndera; Systar SA; Tibco Software

Recommended Reading: "MarketScope for Business Activity Monitoring Platforms, 3Q06"

"Selection Requirements for Business Activity Monitoring Tools"

Sliding Into the Trough

B2B Gateway Software

Analysis By: Benoit Lheureux; Paolo Malinverno

Definition: Multienterprise/B2B gateway software is integration middleware that is used to consolidate and centralize data, process and SOA integration and interoperability between a company's internal applications and systems and those of its external business partners. Centrally managing B2B interactions:

Gives companies economies of scale, deeper insight into the technical aspects of data integration, transaction delivery, process integration and SOA interoperability, such as consolidating, tracking, storing and auditing files, messages, process events, acknowledgments, receipts, and errors and exceptions.

Provides a central, reusable repository for external business partner profiles and processes. This is particularly valuable when dealing with a large number of external business partners and when multiple business units interact with the same partners or partners have similar processes.

Provides support for the myriad of data formats, transport and communication protocols, and security standards.

The multienterprise/B2B gateway software market is composed of middleware technology from multiple disciplines, such as the integration backbone (also called integration brokers), ESBs, general SOA/EDA infrastructure, application servers, application platform suites (APSs), EDI translators and the B2B-enabled integration middleware that is now increasingly available with packaged applications such as Oracle's Fusion and SAP's NetWeaver. As integration disciplines and technology mature, many companies that have invested in integration middleware will likely change, upgrade or augment that middleware with B2B capabilities. Because automating B2B interactions is essential to bottom- and top-line revenue, companies are increasingly seeking to implement an abstract layer of functionality that enables commerce and other forms of process execution and data exchange and reconciliation on multiple layers, including batch and bulk, message-oriented and service-centric.

Position and Adoption Speed Justification: Most multienterprise projects have been successful. Companies using B2B gateway software since the late 1990s have achieved their initial project objectives such as connecting securely and reliably to external business partners,

whether to a few, to dozens, to hundreds or, in a few cases, to thousands. Although many companies implemented multienterprise integration projects on a case-by-case basis, this is changing as they begin applying the same discipline and approaches to multienterprise integration as have been applied to internal integration projects in the last decade: better linkage between integration and business strategy, formation of an integration competency center, and implementation of a common integration infrastructure. Although such projects will continue to proliferate — and will do so even more when B2B Web services achieve critical mass — the B2B gateway software renaissance period is over. Companies are beginning to discover that multienterprise integration consumes significant resources and is not as simple as originally assumed. Challenges that will drive many users of B2B gateway software into the Trough of Disillusionment include the cost of maintaining internal skills for B2B protocols, the cost of upgrading entry-level B2B gateway software, B2B project consolidation, business continuity and the need to implement B2B Web services support within the gateway to extend the internal SOA infrastructure to external business partners. In many cases, companies will continue to implement their own B2B gateway software, making fresh investments in hardware, software and strategy. Despite that trend, in some cases — we've already seen a few — companies will reverse their decision to unilaterally deploy and manage their own multienterprise infrastructure, and shift responsibility for this important task to external service providers, such as integration service providers and system integrators, saving valuable internal resources for other IT tasks.

User Advice: Every company that wants to transact business electronically with external business partners should have a multienterprise strategy and a corresponding multienterprise infrastructure, leveraging some form of B2B gateway software. The majority of companies will implement their own B2B infrastructure using some form of B2B gateway software. Regardless, every company should make a relevant build-or-buy decision regarding a multienterprise infrastructure, possibly outsourcing if other IT priorities are more important. Stop implementing multienterprise projects strictly on a case-by-case basis, and begin implementing such projects within the context of an overall multienterprise strategy that is aligned with your internal integration and business strategies. In addition, look for opportunities to consolidate multiple B2B projects onto one B2B gateway infrastructure, onto two at the most.

Business Impact: A well-implemented multienterprise strategy, including a B2B gateway infrastructure, can directly impact a company's existing business requirements and help it to flexibly implement changes as the business strategy changes. For example, many companies are still automating transactions in trading supply chains, which can lower costs and increase supply chain efficiency, thus continuing to bottom-line revenue. More importantly, using modern B2B gateway software enables companies to handle a wide range of integration approaches such as Applicability Statement 2 (AS2), file transfer and Web services, as well as batch and near-real-time interactions. In addition, it enables new multienterprise capabilities, such as process visibility, data validation and business activity monitoring to, as an example, provide earlier detection and resolution of supply chain exceptions. Such capabilities increase multienterprise process flexibility, improve supply chain execution and data quality, and make automation more appealing and valuable to your external business partners, thus driving top-line revenue growth. Although to date companies have focused on automating their buy-side e-commerce, there is less automation using modern B2B gateway software on the sell side — in part because companies generally have less control over what IT approaches their customers will use. Despite such challenges, in the next few years, more companies will focus on the sell side of their e-commerce initiatives to reduce downstream supply chain friction, better respond to competitive pressure, become more flexible and easy to interact with, and drive additional top-line revenue.

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Sample Vendors: Axway; BEA; Cleo Communications; Click Commerce; Extol; EZCom Software; GXS; iWay Software; IBM; Inovis; Kewill Systems; Microsoft; nuBridges; Oracle; Proginet; Seeburger; Software AG/webMethods; Sterling Commerce; Sun; Tibco

Recommended Reading: "Magic Quadrant for B2B Gateway Providers, 1Q06"

"When to Use (or Not to Use) an Integration Service Provider"

"Findings: B2B Dynamics in Europe"

B2B Web Services

Analysis By: Benoit Lheureux; Frank Kenney

Definition: Web services — whether implemented via SOAP, WS-* or representational state transfer (REST) — are increasingly deployed in multienterprise, B2B application and process integration projects. Examples include the Web-services-based exchange of information and events in support of vendor-managed inventory for complex high-tech manufacturing supply chains; to supply near-real-time integration of business information via Web services between companies (see research on StrikeIron in "Cool Vendors in B2B Integration and Managed File Transfer, 2006"); by Amazon.com to allow third-party companies to leverage knowledge about its buyer's preferences (see "Implement Four Key Business Criteria to Achieve Widely Adopted Web Services"); and by salesforce.com to implement its Apex Connect APIs. B2B Web services are also used to access business information from internal and external applications (run by external business partners) to support portal-based applications and are beginning to be used by integration service providers (for example, GXS and Hubspan) as an interface into their hosted integration services; for example, to automate the provisioning of new users on their network. In these contexts, Web services are being used as a multienterprise extension of internal SOAs or in lieu of more-traditional approaches to multienterprise integration, such as AS2, EDI and FTP.

Position and Adoption Speed Justification: Expectations for B2B Web services have passed their initial hype and are approaching the Trough of Disillusionment as companies realize that, despite their internal emerging success with Web services, several factors have conspired to inhibit more widespread adoption of Web services in multienterprise projects. These factors include the long-overdue availability of an approved Web services reliable-messaging specification (which has, in part, helped fuel widespread adoption of AS2), the still relatively new solutions to support multienterprise Web services (for example, Amberpoint, Infravio — acquired by Software AG/webMethods, Reactivity and others), and user's overall immature understanding of the architectural framework necessary to support Web services (see "Gain Control and Integration Functionality With a Web Services Framework"). These inhibitors will disappoint companies as they continue to expand their use of Web services internally and realize that better governance will be required, particularly when extending internal Web services to external business partners.

User Advice: Watch for an important bellwether event: the long-overdue approval of a Web-services-reliable messaging specification that is broadly implemented by vendors from all parts of the IT industry, including pure-play Web services infrastructure vendors (listed above), integration suite vendors (such as Tibco Software and webMethods), B2B gateway software vendors (such as Axway and Sterling Commerce), packaged application vendors (such as Oracle and SAP), integration service providers (E2open and GXS) and others. Other drivers will include the increasing maturity of Web services infrastructure in mainstream business, the proliferation of outsourcing (which forces more application logic outside your company) and the decomposition of application logic into smaller components (which require a more flexible infrastructure to integrate

them). But, as Web services proliferate, in part fueled by "Web 2.0," look to vendors for registry implementations to make it easier to discover and implement Web services, and also look for management and governance tools that enable you to audit and manage Web services as well as to identify dependencies between them.

Business Impact: This will impact e-commerce, supply chain and demand chain, as well as outsourcing, including all forms of software as a service and integration as a service.

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Sample Vendors: BEA Systems; GXS; IBM; Microsoft; Software AG; SOA Software; Sun; Tibco Software; webMethods

Recommended Reading: "Gain Control and Integration Functionality With a Web Services Framework"

Enterprise Service Bus

Analysis By: Roy Schulte; Jess Thompson; Kimihiko Iijima

Definition: An ESB is a Web-services-capable middleware infrastructure that supports program-to-program communication and related mediation services, such as service virtualization, transformation, security, intelligent routing and guaranteed delivery. All ESBs support SOAP and HTTP. Nearly all of them support alternative MOM communication protocols, publish-and-subscribe and optional adapters. Many also support load balancing, failover, service monitoring, service orchestration, message validation and logging.

Position and Adoption Speed Justification: Most major software vendors and SOA infrastructure specialists offer ESB technology. Sometimes it is offered in a product called an ESB, but often the technology is bundled into other products, such as operating systems, integration suites, application platform suites or packaged applications. ESBs are more functional than plain Web services stacks, and they are more standards-compliant than MOM. They cost much less than a full integration suite or application platform suite, so they appeal to a broad middle ground of SOA applications.

Vendors are generally enthusiastic about the benefits of ESBs, but their product offerings vary widely in maturity and quality (some are mature and proved, while others are incomplete). User companies increased their buying and use of ESB technology significantly during 2006 and 1H07, largely in support of their increasing use of SOA. However, some user companies remain uncertain about the need for an ESB in their environment, so adoption is far from universal.

User Advice: All development projects that will acquire or build more than 10 to 30 SOA services or EDA event types should use ESB technology as the core element in their SOA infrastructure, rather than using basic Web services stacks, plain MOM, remote procedure calls or object request brokers. Many companies will acquire ESB technology without making a separate purchase decision because it is bundled into a packaged application or a larger SOA infrastructure product.

Business Impact: ESBs improve the quality, manageability and scalability of program-to-program communication, make it easier to share SOA services, and make service versioning and changes to SOA interfaces easier to implement. ESBs are often used to present portions of legacy and purchased non-SOA applications as SOA services. However, ESBs are also relevant

in large-scale or long-living SOA service domains where there are no legacy non-SOA applications.

Eighty percent of large companies will have ESBs or similar SOA infrastructure products from three or more vendors by 2009 (0.7 probability). The SOA and integration market is dynamic, with a relatively high rate of mergers and acquisitions. Therefore, companies buying an ESB product or another SOA middleware infrastructure product that contains an ESB product should consider the product viability and vendor viability, as well as the ease of development, standards compliance, quality of local support, administrative tools and feature richness.

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Sample Vendors: Apache Software Foundation; Axway; BEA Systems; Cape Clear; Cordys; E2E; Fiorano; Fujitsu; IBM; Iona; Magic Software Enterprises; Microsoft; MuleSource; Oracle; PolarLake; Progress Software; Red Hat; Software AG; SOA Software; Sun; Tibco Software; Vitria

Recommended Reading: "The Enterprise Service Bus: Communication Backbone for SOA"

"Where to Use an Enterprise Service Bus and Why"

SOA

Analysis By: Roy Schulte; Yefim Natis

Definition: SOA is a style of application architecture. An application is an SOA application if it is modular; the modules are distributable; software developers have written or generated interface metadata that specifies an explicit contract so that another developer can find and use the service; the interface is separate from the implementation (code and data) of the service provider; and the services are shareable — that is, designed and deployed in a manner that enables them to be invoked successively by disparate consumers. Unlike some other types of distributed computing, services in SOA can be shared across applications running on disparate platforms and are inherently easier to integrate with software from other development teams.

Position and Adoption Speed Justification: The use of SOA is accelerating in response to escalating business requirements, the emergence of Web and Web services standards (such as WSDL and SOAP) and the improving availability of SOA-capable development tools and applications. Competition, globalization and technology advances are driving companies to change their products, business processes and prices more frequently than they did before the mid-1990s. The growing use of BPM and BAM is also causing companies to use more SOA, because BPM and BAM are more-effective and easier to develop when using SOA. Vendors of middleware, development tools and packaged applications have committed to moving to SOA, and their product lines are well into the transition. User companies are moving more slowly, on average, and they are experiencing varying degrees of difficulty in ramping up their use of SOA. These difficulties hinder, but will not prevent, the spread of SOA throughout the application portfolios of large companies. The growing, if limited, practical experience with SOA has demonstrated the real costs and benefits of the transition to SOA. SOA skepticism is gradually giving way to a realistic anticipation of costs and benefits. Development and management best practices for SOA are still not fully mature, but companies are largely satisfied with their experience with it.

User Advice: Use SOA when designing new business applications, particularly those whose life spans are expected to be more than three years and that will undergo continuous refinement,

maintenance or enlargement. SOA is well-suited especially for building composite applications. When buying packaged applications, rate those that implement SOA more highly than those that do not. Also, use SOA in application integration scenarios that involve composite applications that tie new logic to purchased packages, legacy applications or services offered by other business units. However, do not discard non-SOA applications in favor of SOA applications just on the basis of architecture. Discard non-SOA applications only if there are compelling business reasons why the non-SOA application has become unsatisfactory. Continue to use non-SOA architecture styles for some new, tactical applications of limited size and complexity, and for minor changes to installed non-SOA applications. Recognize that there are multiple patterns within SOA (such as multichannel applications, composite applications, multistep process flows and event-driven SOA), and each of these has its own best practices for design, deployment and management.

Business Impact: SOA is a durable change in application architecture, like the relational data model and the graphical user interface. The main benefit of SOA is that it reduces the effort and time needed to change application systems to support changes in the business. The implementation of the first SOA application in a business domain will generally be as difficult as, or more difficult than, building the same application using non-SOA designs. However, subsequent applications and changes to the initial SOA application are easier, faster and less expensive because they leverage the SOA infrastructure and previously built services. SOA is an essential ingredient in strategies that seek to enhance the agility of a company. SOA also reduces the cost of application integration, especially after enough applications have been converted or modernized to support an SOA model.

Benefit Rating: Transformational

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Recommended Reading: "Five Principles of SOA in Business and IT"

"SOA: Where Do I Start?"

"Applied SOA: Transforming Fundamental Principles Into Best Practices"

Climbing the Slope

Microsoft .NET Application Platform

Analysis By: Yefim Natis; Mark Driver

Definition: Microsoft does not identify a product that is an application server. However, its Microsoft Windows offering contains all the functional elements of an enterprise application server (EAS). Despite the lack of a named EAS product, Microsoft clearly (and effectively) competes against the leading EAS vendors. We refer to the subset of Microsoft Windows, with the other Microsoft offerings that amount to an EAS and a larger application platform suite, as the Microsoft Application Platform (MSAP). Today, MSAP is based on the .NET architecture. MSAP includes the .NET framework (which includes ASP.NET), Internet Information Server, enterprise services (such as COM+), Microsoft Message Queuing, BizTalk Server, Office SharePoint and more. The release of Windows Server 2008 (formerly known as Longhorn) will likely make some changes and additions to this list, including, most notably, Windows Communications Foundation. It is already clear that the .NET Framework v.3 adds significant enrichment to the previous versions of MSAP — in the areas of productivity, support of SOA, innovations in support of modern user experience and multiprotocol integration.

MSAP competes against Java EE and other enterprise platform architectures in the high-end enterprise projects. MSAP also competes against PHP platform, Ruby on Rails, ColdFusion and open-source Java frameworks, such as Spring and Struts, in lower-end productivity-oriented enterprise projects.

Position and Adoption Speed Justification: The technical quality and quality of service of .NET are suitable for the majority of enterprise projects. However, Microsoft's business strategy for mission-critical projects, including support, account management, long-term continuity in the relationship and its software offerings, lags behind leading enterprise vendors' strategies. Microsoft does not have a good reputation of seeing its customers through long-term software architecture endeavors. Its exclusive commitment to Windows as the only operating system platform reduces the appeal of MSAP in large-scale enterprise settings. The impending replacement of Windows Server 2003 in the 2008 time frame and its anticipated fundamental platform changes continue to remind high-end enterprise platform users that MSAP has not yet reached dependable maturity levels, despite its increasing technical strengths.

However, in lower-end enterprise settings, where MSAP dominates with its productivity, ubiquity of skills and relatively low cost, MSAP remains a strong and growing option for the majority of projects. Yet, MSAP's market share has not been increasing dramatically lately because of the growing number of Java and other high-productivity alternatives, including open source.

MSAP is a technically strong platform option. It is used widely for small and midsize software projects and, on occasion, for very large enterprise projects. Because of business strategy shortcomings, MSAP is not considered for high-end projects as much as it could technically handle. MSAP's market share is stable and will likely remain at present levels until the next generation of MSAP is available (and proven) with the release of Windows Server 2008 and until Microsoft's business strategy in high-end enterprise settings better reflects the requirements of that environment.

User Advice: If you choose to use MSAP, then you have to use the Windows OS platform. If that is acceptable, then consider MSAP for small and midsize software projects without limitations. For large-scale projects, recognize that the technical quality of the MSAP technology is likely sufficient for the requirements of the project. However, to achieve the "whole product" experience, you will need to rely on third-party system integrators.

Microsoft excels at providing strong developer productivity at the cost of long-term flexibility. Consider MSAP as one element in a larger IT strategy (for example, that might include Java EE) or as the principle strategic platform when a Microsoft-centric strategy is acceptable or preferable.

Business Impact: MSAP frees IT organizations from the lock-in into a single programming language (such as Java or PHP), but it locks the project into the Windows operating system and only the hardware options that are available for Windows. MSAP can be a lower-cost option for enterprise platform technologies. However, in larger settings, this assumption is not always true and must be verified with real numbers. The history of significant discontinuities as Microsoft releases major new versions of its MSAP technologies have also increased costs of long-term use of MSAP. Above all, however, users who are happy to use Windows infrastructure for all their development needs find MSAP and its development environment easy to learn, easy to use and easy to staff —major factors reducing the costs and improving the time-to-market of software projects.

Benefit Rating: High

Market Penetration: More than 50% of target audience

Maturity: Early mainstream

Sample Vendors: Microsoft

Recommended Reading: "Magic Quadrant for Enterprise Application Servers, 2Q06"

Integration Service Providers

Analysis By: Benoit Lheureux; Paolo Malinverno

Definition: A market exists for B2B integration capabilities that are hosted in a multitenant environment and delivered as a service rather than as software. Traditionally known as EDI value-added networks (VANs), we now call these hosted offerings IaaS, and we call vendors that offer such services (usually in one role relative to other roles) "integration service providers" (see "Taxonomy and Definitions for the Multienterprise/B2B Infrastructure Market").

To be considered an integration service provider, a vendor must offer hosted multienterprise integration and interoperability services. These include some combination of four major categories of IaaS functionality, such as:

- Communication services, including multiprotocol support for protocols, such as EDI, AS2, RosettaNet and Web services
- Trading-partner management services, such as tools to provision connections and manage certificates for security
- Integration services, such as in-line translation and back-end system integration
- Application services, such as order visibility or compliance management

In many cases, IaaS is only one component of these vendors' overall IT solution portfolios. Rather than simply categorizing them as integration service providers, it is more accurate to say these vendors offer IaaS in one role, while often offering IT products and services in another. To be an IaaS provider, a vendor must separately market, sell and support its hosted integration services from its other IT products and services. Vendors that bundle hosted integration services into their overall IT portfolios may be doing what they need to address their customers' requirements, but we do not consider them to be direct competitors in the IaaS market segment.

Position and Adoption Speed Justification: Many integration service providers that were primarily only supporting EDI mailbox-based B2B projects have recently substantially expanded their services. For example, they now support modern data formats (such as XML) and modern Internet protocols (such as AS2 and Web services), rather than legacy approaches, such as async dial up, bisync dial up or FTP (still widely used). They also have implemented new features to address users' increasing needs for holistic monitoring of multienterprise processes, such as order visibility. The more-advanced vendors also leverage new approaches, such as service-oriented architecture, to perform integration within the firewall and to facilitate new functionality, including data validation and supply chain execution, as well as to make it easier to link such services as automated provisioning to external partners. Although the Trough of Disillusionment is several years past, vendors still have several years of work to finish modernizing their services, including providing more-programmatic interfaces to their services, implementing Web services support and a registry, and expanding their application portfolios in top industries or in Europe to have a solid cross-national offering.

User Advice: To outsource multienterprise integration projects, consider integration service providers. However, the modernization of traditional EDI VANs (such as EasyLink, Global eXchange Services [GXS], Sterling Commerce and Inovis) into full-featured integration service providers with modern, reliable IT platforms and new hosted applications to solve problems in specific industries will continue for the next few years. Although some of the more-modern

integration service providers (such as E2open, Hubspan and Internet Commerce Corporation [ICC]) entered the market more recently with modern IT infrastructures and solutions, these vendors are still struggling to expand their market share, sometimes in new industry segments. These forces are collectively driving consolidation in this market segment (for example, in 1Q07, ICC.net acquired EasyLink), so pay attention to a prospective vendor's long-term strategy and viability.

In Europe, the cross-national offerings of integration service providers are maturing, but prospects should always verify whether a potential provider of IaaS can meet their particular regions for international B2B projects. When negotiating an agreement with providers, look for transparent and predictable pricing. Customers are increasingly signing deals with "bundled" B2B integration features, such as tiered number of external business partners and volume, fixed-price in-line translation, and process visibility that associates relevant B2B documents (for example, purchase orders, advanced shipment notices and invoices for order to cash).

Business Impact: Companies of all sizes and across all industries have the option to outsource their B2B infrastructure, rather than licensing and deploying B2B gateway software in-house. Multienterprise projects are almost always mission-critical, but the increased modernization, reliability and scale provided by integration service providers mean that companies often have a choice: buy or build. This means they can decide whether doing B2B integration is a required internal core competency, or whether the project is better outsourced to free up internal IT resources for other business priorities.

Benefit Rating: High

Market Penetration: More than 50% of target audience

Maturity: Mature mainstream

Sample Vendors: Advanced Data Exchange; BT Group; CrossGate; DIcentral; E2open; EasyLink Services; Global eXchange Services; Hubspan; Inovis; Internet Commerce Corp.; Mincom; nuBridges; Perfect Commerce; Railinc; RedTail Solutions; Seeburger; SPS Commerce; Sterling Commerce; TietoEnator; True Commerce

Recommended Reading: "Taxonomy and Definitions for the Multienterprise/B2B Infrastructure Market"

"Findings: B2B Dynamics in Europe"

"Magic Quadrant for Integration Service Providers, 1Q06"

Integration Suites

Analysis By: Jess Thompson

Definition: An integration suite is a comprehensive set of infrastructure for making independently developed application systems work together. It is comprised of a set of features hosted on a messaging platform. The messaging platform can range from MOM to an ESB. The features typically include transformation, intelligent routing, BPM, adapters, administration, management and monitoring tools, and B2B features, such as trading-partner management.

Position and Adoption Speed Justification: Integration suites continue to evolve technically, as vendors address growing user requirements and as standards (such as Web services and Java) become more applicable to the challenges of this market. SOA, BPM, BAM and EDA are all drivers of integration suite adoption. Moreover, packaged application vendors bundle integration technology into their products, which is making the integration suite adoption cycle shorter for

"bundled solution-oriented" customers. Vendors are beginning to break apart large integration suites by providing basic integration features via an ESB, making integration platforms more standards-based and priced in a manner that makes these products affordable for small and midsize businesses.

User Advice: Companies with a broad range of demanding business requirements in areas such as BPM, connectivity to packaged and legacy applications, BAM; high-volume workloads, frequent changes to business processes, and heterogeneity of technology — including disparate operating systems, application servers and DBMSs — should consider full-blown integration suites. Although relatively expensive in their license and maintenance costs, these products can deliver high business benefits and low total cost of ownership (TCO) during the life of an enterprise application. However, these products are overkill for business scenarios with only moderate volumes of work, with less-technical heterogeneity or when simpler, less-expensive integration tools, such as ESBs, batch file transfer tools, data integration tools or programmatic integration servers, can handle the requirements.

Business Impact: Because they're now embedded in APSs from all the major vendors (BEA Systems, IBM, Microsoft, Oracle and SAP), as well as in packaged applications (for example, from SAP and Oracle), integration suites will be adopted more widely.

Vendors, especially pure-play integration providers that are looking to maintain a functional edge over the larger generalists, try to anticipate user requirements by offering new features, in some cases, before leading-edge buyers are ready to use them. Capabilities such as BAM and complex event processing give some vendors a competitive differentiation, although the market for those features is still limited to leading-edge implementers, because most mainstream projects are not yet using those capabilities.

Benefit Rating: High

Market Penetration: More than 50% of target audience

Maturity: Mature mainstream

Sample Vendors: BEA; Fujitsu; IBM; Microsoft; Oracle; SAP; Sun; Tibco Software; Vitria; webMethods

Recommended Reading: "Predicts 2006: The Evolution of Integration Products Will Continue to Affect How Integration Is Pursued"

Programmatic Integration Servers

Analysis By: Massimo Pezzini; Dale Vecchio

Definition: Programmatic integration servers are relatively low-cost products — for example, as compared with integration suites — that are focused on enabling legacy (mainframe or midrange-based) integration. They integrate established applications developed in COBOL, PL/1, RPG or other programming languages with other applications, using a combination of screen scraping (3270, 5250, VT and other terminal data streams), programmatic access, transformation, routing and microflow technologies.

Programmatic integration servers enable developers to orchestrate the execution of multiple transactions (possibly belonging to different applications on different platforms). These "transaction flows" can then be published in the form of Java, .NET, XML or Web-services-based components that can be consumed (or invoked), typically by user-facing applications deployed on Java EE servers, .NET, portal products, business process managers and other platforms.

Architecturally, a programmatic integration server can also be considered a specialized, functional subset of an ESB, providing transformation, routing, microflow-based orchestration, service publishing, metadata management, and Web services and messaging support. Specifically, it is focused on exposing business function in established applications running on mainframes or proprietary midrange computers as Web services for consumption in new BPM or custom-made applications.

Position and Adoption Speed Justification: Programmatic integration servers are rapidly maturing, proven products. In most cases, they are based on technology (such as screen scrapers) that has been in the market since the late 1990s. Moreover, through incessant mergers and acquisitions, vendors are extending the capabilities of these products to cover even the most exotic legacy platforms to enable multiple deployment scenarios (for example, on or off mainframe) and to support various interoperability techniques: Web services, XML, JMS, Java Connector Architecture (JCA) and so forth. The growing popularity of SOAs encourages users to define strategies for leveraging their established mainframe or midrange applications to build reusable business services.

In many cases, these activities will foster legacy transformation and re-engineering projects; however, in other situations, simply wrapping established applications will often prove to be the most-cost-effective and low-risk strategy. At times, programmatic integration servers are used in combination with ESBs or integration suites to provide a sound infrastructure for service-enabling pre-SOA applications through integration and wrapping. Their relatively low cost, ease of use and rapid return on investment are propelling their adoption by mainstream enterprises.

User Advice: Organizations with SOA projects that include integrating and reusing established mainframe or legacy midrange-based applications should evaluate the adoption of programmatic integration servers as a key SOA-enabling technology. They can be used as the primary infrastructure for legacy-centric integration projects or in combination with ESBs to provide "last mile" connectivity with legacy platforms.

The programmatic integration server space is under competitive pressure from IBM and, to a lesser extent, Microsoft. Both provide a variety of legacy integration products. Therefore, this market is rapidly consolidating, as demonstrated by the number of acquisitions during the past 12 months (Rocket Software acquired Seagull Software, Progress Software/DataDirect acquired Neon Systems, and Attachmate and WRQ merged). When selecting a programmatic integration server, be aware that this market is stable but represents a contrast of large vendors (IBM, Microsoft, Progress Software, Software AG and others) with much smaller, albeit focused vendors (GT Software, AttachmateWRQ and Seagull Software).

Business Impact: Programmatic integration servers enable users to rapidly wrap legacy applications into business services used to implement new composite applications — such as call center front ends, and e-commerce and self-service portals — thus significantly reducing development cost and time to deployment. By adopting these products, organizations can gain a competitive advantage by rapidly implementing new business processes (or by extending established ones), maximizing the reuse of established application assets and introducing minimal disruptions in the application environment.

Benefit Rating: Moderate

Market Penetration: 20% to 50% of target audience

Maturity: Mature mainstream

Sample Vendors: Attachmate; GT Software; IBM; Jacada; Microsoft; Progress Software; Seagull Software; Software AG

Entering the Plateau

Integration Competency Centers

Analysis By: Paolo Malinverno

Definition: An application ICC is an organizational unit that routinely addresses several integration planning and control issues through a common set of defined technologies, methodologies and policies. This leads to reduced costs and risks, better manageability and significant economies of scale in terms of technology and skills.

ICCs are frequently used to drive SOA projects, and thus are renamed to SOA COEs. This typically adds quite a lot of SOA governance duties to the ICC's work.

Position and Adoption Speed Justification: ICCs are mostly found in large and very-large companies, which frequently have more than one, and federate them through a careful split of the areas of influence. Most midsize companies haven't yet created an ICC, but many will as they implement more-sophisticated integration tools, Web services and SOA projects. The ICC is a good start for an SOA project, but it needs to be reinforced with more-structured governance processes and additional policies to support an effective SOA.

The ICC carries out projects, such as the single customer view, and sets general guidelines and standards for integration work. These projects have high business value, have to put together different and diverse portions of an IT infrastructure, and have to reuse the integration interfaces across different integrated applications. In most ICCs, reuse of all the integration work is essentially automatic because decisions on how to perform integration work are taken in the ICC, and reuse is part of the "good housekeeping rules" by which all effective ICCs work and deliver value.

Typically, in SOA projects, implementation decisions are made in separate development groups. Unless strict governance provides the necessary coordination, different groups (or even individual programmers) will take different and uncoordinated decisions, multiplying the diversity and driving the reuse down. Imposing decision rules and localizing them into the ICC seems to be the closest option at hand, but it's not that simple. (For more information, see "The ICC and SOA Governance.")

SOA projects will be easier in companies that have an ICC. However, those skills are not enough to make the associated decisions or to ensure those decisions are endorsed companywide. SOA requires more-formal discipline and governance than the ordinary structured integration work. To extend its responsibility into driving the growth of an SOA, an ICC needs to adopt new structured procedures, in addition to the ones already in place, for the design, reuse, operation and maintenance of services.

User Advice: Every IT department of any size should have an ICC. Its benefits quickly pay off the organizational challenges that must be addressed while an ICC is created. ICCs/SOA COEs are absolutely vital in every SOA project out of the pilot stage. Growing an SOA will form ICCs in companies that don't have one, but will cause significant organizational turbulence because most SOA decisions will involve several constituencies in the company (for example, process architects and developers) that might not be used to working together.

Business Impact: ICCs affect software development project teams, IT architects, IT management and business unit management by centralizing some of the work items and decisions that would normally be carried out by developers. ICCs identify and manage integration technology, manage the development of reusable services in SOA projects, maintain integration-related metadata, and work with development teams to ensure the consistent and effective

development of business applications, and the implementation of business processes. By 2010, more than 85% of very large enterprises and more than 60% of large enterprises will have one or more ICCs (0.7 probability).

Benefit Rating: High

Market Penetration: More than 50% of target audience

Maturity: Early mainstream

Sample Vendors: Acando; BEA Systems; HP Services; IBM Global Services; Informatica; Oracle Consulting

Recommended Reading: "The Four Steps to Building an Integration Competency Center"

"Integration Competency Centers Demand a Wide Set of Skills"

"The ICC and SOA Governance"

"Sample Governance Mechanisms for a Service-Oriented Architecture"

Java Platform, Enterprise Edition

Analysis By: Yefim Natis

Definition: Java EE is a Sun-sponsored and Java Community Process (JCP)-managed architecture and programming model for multiplatform Java business applications. (Prior to this version, the architecture was named Java 2 Platform, Enterprise Edition [J2EE]). Java EE is implemented as Java EE application servers by many commercial and some open-source, community-based vendors. JCP provides compliance test cases and reference implementations for Java EE components. Many vendors claim that their Java EE-compliant application servers pass all the test cases. The test cases are available to licensees at a substantial fee. Some smaller vendors avoid the cost and claim compatibility without having passed the tests.

A Java EE application server offers several container models — for example, Servlets, session EJB, entity EJB, JCA and MDBs — and other application programming interfaces, such as Java Management eXtensions (JMX), JMS and Java application programming interface for XML-Web services (JAX-WS). It is a comprehensive platform architecture for modern application designs that:

- Provides unprecedented application portability
- Can offer near-mainframe-level quality of service (by some more-advanced implementations)
- Is backed by a reasonably independent and large consortium of vendors — members of JCP
- Is too complex for many software projects (although Java EE 5 is an improvement) and supports only one programming language (Java)
- In its complete rendition, offers too much functionality for many business applications
- Provides basic support for event-driven architecture via MDBs and JMS programming models
- Is designed well for distributed homogeneous applications, but lacks completeness for full support of heterogeneous service-oriented applications

Position and Adoption Speed Justification: The Java EE programming model dominates high-end enterprise software projects. Thousands of mainstream enterprises use Java EE as the platform for their mission-critical applications. Leading application independent software vendors use Java EE for their new software development. The platform implementations range from high-cost, high-end versions to low-cost, mass-market versions. Closed-source and open-source options compete for new projects.

The Java EE specification changes have become gradual, incremental and infrequent. Discontinuous innovation is rare and typically addresses isolated specification shortcomings. The leading products are proven, dependable and interchangeable for most applications. The use practices and the development methodologies are well-established, and the skilled resources are broadly available and steadily increasing. The product is clearly near its plateau of productivity, although not quite there yet. The new Java EE 5 has introduced a mild disruption to the installed base. Introduction of EJB v.3 is technically incremental, but, in effect, discontinuous. It is a major re-architecture of the Java EE component model and, although the previous versions are required to be supported by compliant implementations, no new development should invest in anything but the more-efficient and easier-to-use EJB v.3, which, in turn, requires new skills and design practices.

User Advice: Most mainstream business applications do not need the full power of a high-end Java EE platform. Considering the high degree of compatibility between implementations of Java EE, use the proven low-cost offerings for less-demanding parts of the application, and invest in the high-end alternative platforms only for select high-demand parts of the application environment.

Consider the open-source option as a viable alternative to the more-established, closed-source implementations.

Expect price decreases in the low end of Java EE and price increases in the less-standard, high-end and extended Java EE arena.

Expect continuing emergence of new easy-to-use programming model frameworks for Java (such as Interface 21 Spring Framework) and other core languages (such as PHP, Ruby and Groovy), which are likely to confine the full and native Java EE implementations to mostly high-end application projects.

Business Impact: In the mainstream enterprise practices, outside Windows-dedicated software engineering, Java is approaching the status of COBOL of the 1970s and 1980s. The significant degree of portability and the well-established interoperability of the licensed Java EE implementations reduce costs of software engineering by making vendor implementations and skilled engineering resources readily available in-house, through system integrators, or through off-shore outsourcing.

Benefit Rating: High

Market Penetration: More than 50% of target audience

Maturity: Mature mainstream

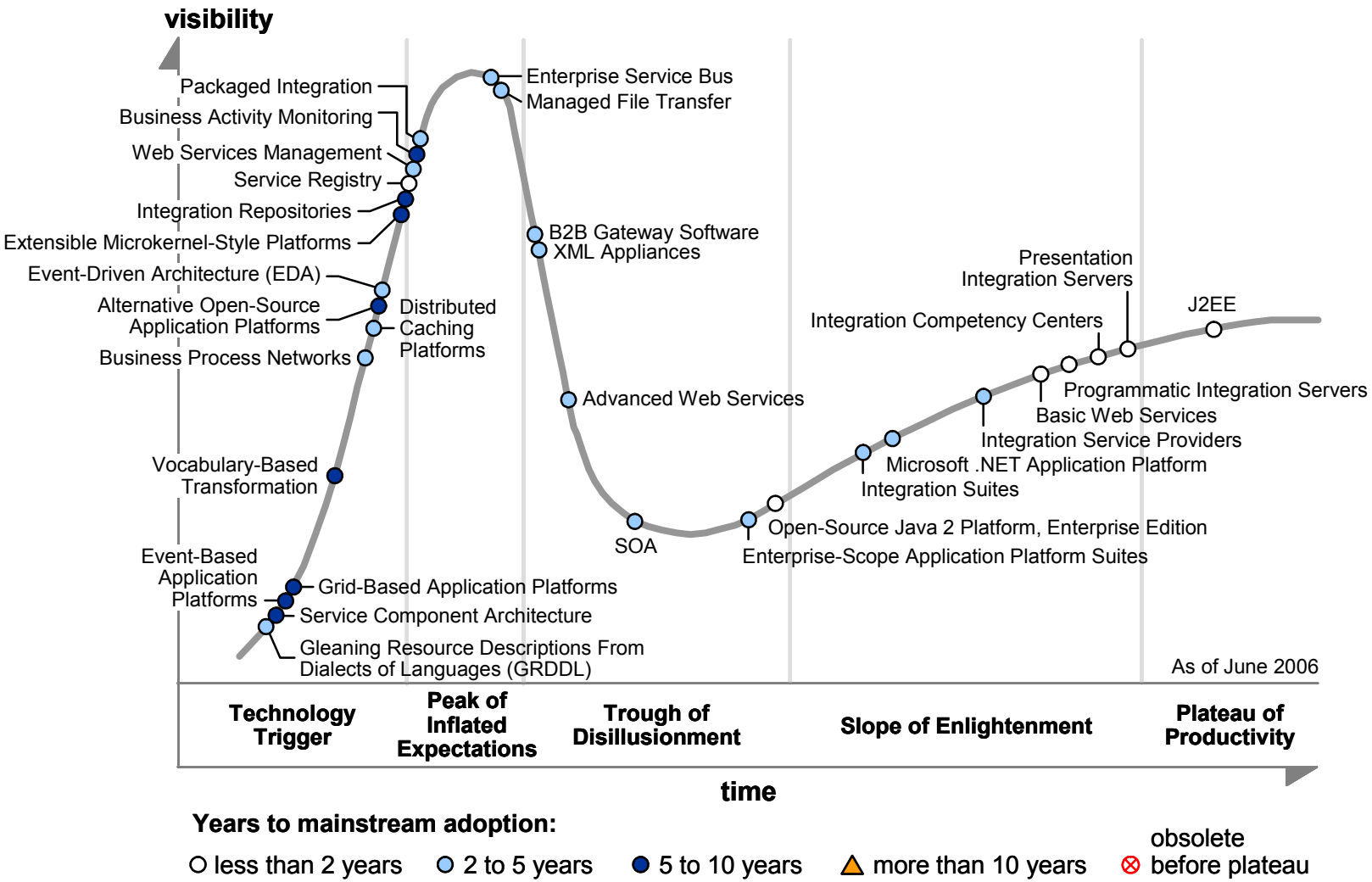
Sample Vendors: Apache Software Foundation; BEA Systems; IBM; Oracle; Red Hat (JBoss); SAP; Sun

Recommended Reading: "Key Issues for Platform Middleware"

"Magic Quadrant for Enterprise Application Servers, 2Q06"

Appendices

Figure 3. Hype Cycle for Application Integration and Platform Middleware, 2006



Source: Gartner (July 2007)

Gartner

Hype Cycle Phases, Benefit Ratings and Maturity Levels

Table 1. Hype Cycle Phases

Phase	Definition
Technology Trigger	A breakthrough, public demonstration, product launch or other event generates significant press and industry interest.
Peak of Inflated Expectations	During this phase of overenthusiasm and unrealistic projections, a flurry of well-publicized activity by technology leaders results in some successes, but more failures, as the technology is pushed to its limits. The only enterprises making money are conference organizers and magazine publishers.
Trough of Disillusionment	Because the technology does not live up to its overinflated expectations, it rapidly becomes unfashionable. Media interest wanes, except for a few cautionary tales.
Slope of Enlightenment	Focused experimentation and solid hard work by an increasingly diverse range of organizations lead to a true understanding of the technology's applicability, risks and benefits. Commercial, off-the-shelf methodologies and tools ease the development process.
Plateau of Productivity	The real-world benefits of the technology are demonstrated and accepted. Tools and methodologies are increasingly stable as they enter their second and third generations. Growing numbers of organizations feel comfortable with the reduced level of risk; the rapid growth phase of adoption begins. Approximately 20% of the technology's target audience has adopted or is adopting the technology as it enters the Plateau.
Years to Mainstream Adoption	The time required for the technology to reach the Plateau of Productivity.

Source: Gartner (January 2007)

Table 2. Benefit Ratings

Benefit Rating	Definition
Transformational	Enables new ways of doing business across industries that will result in major shifts in industry dynamics
High	Enables new ways of performing horizontal or vertical processes that will result in significantly increased revenue or cost savings for an enterprise
Moderate	Provides incremental improvements to established processes that will result in increased revenue or cost savings for an enterprise

Benefit Rating	Definition
Low	Slightly improves processes (for example, improved user experience) that will be difficult to translate into increased revenue or cost savings

Source: Gartner (January 2007)

Table 3. Maturity Levels

Maturity Level	Status	Products/Vendors
Embryonic	In labs	None
Emerging	Commercialization by vendors Pilots and deployments by industry leaders	First generation High price Much customization
Adolescent	Maturing technology capabilities and process understanding Uptake beyond early adopters	Second generation Less customization
Early mainstream	Proven technology Vendors, technology and adoption rapidly evolving	Third generation More out of box Methodologies
Mature mainstream	Robust technology Not much evolution in vendors or technology	Several dominant vendors
Legacy	Not appropriate for new developments Cost of migration constrains replacement	Maintenance revenue focus
Obsolete	Rarely used	Used/resale market only

Source: Gartner (January 2007)

RECOMMENDED READING

"Understanding Gartner's Hype Cycles, 2007"

This research is part of a set of related research pieces. See "Gartner's Hype Cycle Special Report for 2007" for an overview.

REGIONAL HEADQUARTERS

Corporate Headquarters
56 Top Gallant Road
Stamford, CT 06902-7700
U.S.A.
+1 203 964 0096

European Headquarters
Tamesis
The Glanty
Egham
Surrey, TW20 9AW
UNITED KINGDOM
+44 1784 431611

Asia/Pacific Headquarters
Gartner Australasia Pty. Ltd.
Level 9, 141 Walker Street
North Sydney
New South Wales 2060
AUSTRALIA
+61 2 9459 4600

Japan Headquarters
Gartner Japan Ltd.
Aobadai Hills, 6F
7-7, Aobadai, 4-chome
Meguro-ku, Tokyo 153-0042
JAPAN
+81 3 3481 3670

Latin America Headquarters
Gartner do Brazil
Av. das Nações Unidas, 12551
9° andar—World Trade Center
04578-903—São Paulo SP
BRAZIL
+55 11 3443 1509

Gartner