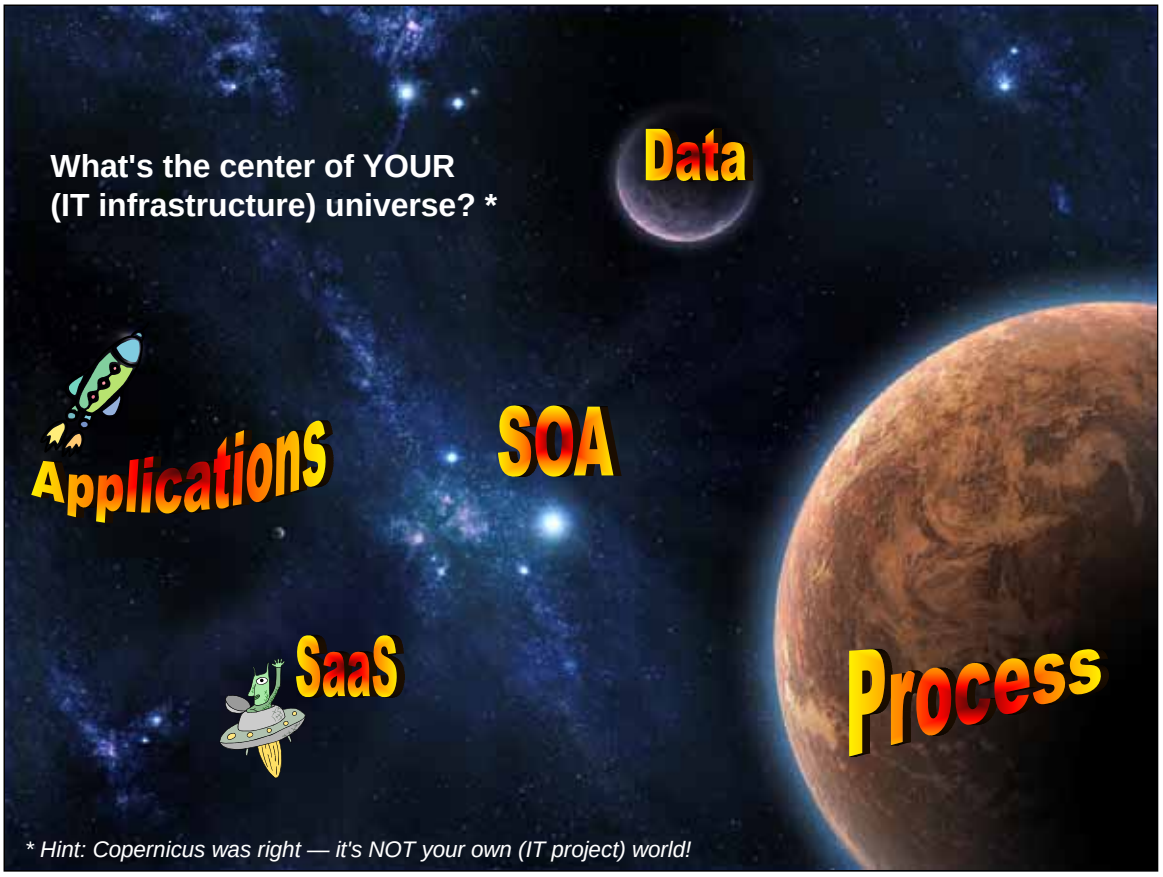
Platform and Integration Middleware: Time to Take IT Convergence Seriously

Massimo Pezzini and Paolo Malinverno

Gartner Symposium/ITxpo

November 4-8, 2007
Palais Des Festivals
Cannes, France

Tactical Guideline: Establish a holistic IT infrastructure strategy that establishes a clear relationship between all high-level IT infrastructure components, including applications, data, process, software as a service (SaaS) and architecture such as service-oriented architecture (SOA).



Ask any IT person what dominates their (IT) world and they're likely to answer "process improvement," "application development," "data quality," "service-oriented architecture," "SaaS," "Web services" or something equally IT-myopic. In most cases this is not a symptom of selfishness — for such people, these IT projects *are* indeed the center of their IT universe.

But CIOs, CTOs, IT Directors and other people in the IT organization with responsibilities across the whole IT organization and a wide range of IT projects know that: IT convergence is an inevitable, emerging IT trend, and that process, data, applications, SaaS and architecture are ever more tightly related; an unbalanced focus only on SOA can lead to un-reconciled data between applications and SaaS; application upgrades by themselves can't improve brittle business process execution across applications.

Those responsible for managing a large portfolio of IT projects need a balanced respect for, and should apply sufficient attention to, all the high-level components of their IT infrastructure, including applications, data, process and architecture. They also should understand that one size does not fit all, and that often different — even if partially overlapping — IT solutions are needed to address different IT project requirements. In this presentation, we put many of the high-level IT infrastructure components, including applications, data, process and architecture, into perspective, showing how they're related and providing a bird's-eye view and comparison of the many IT products and vendors available to implement them.

Key Issues

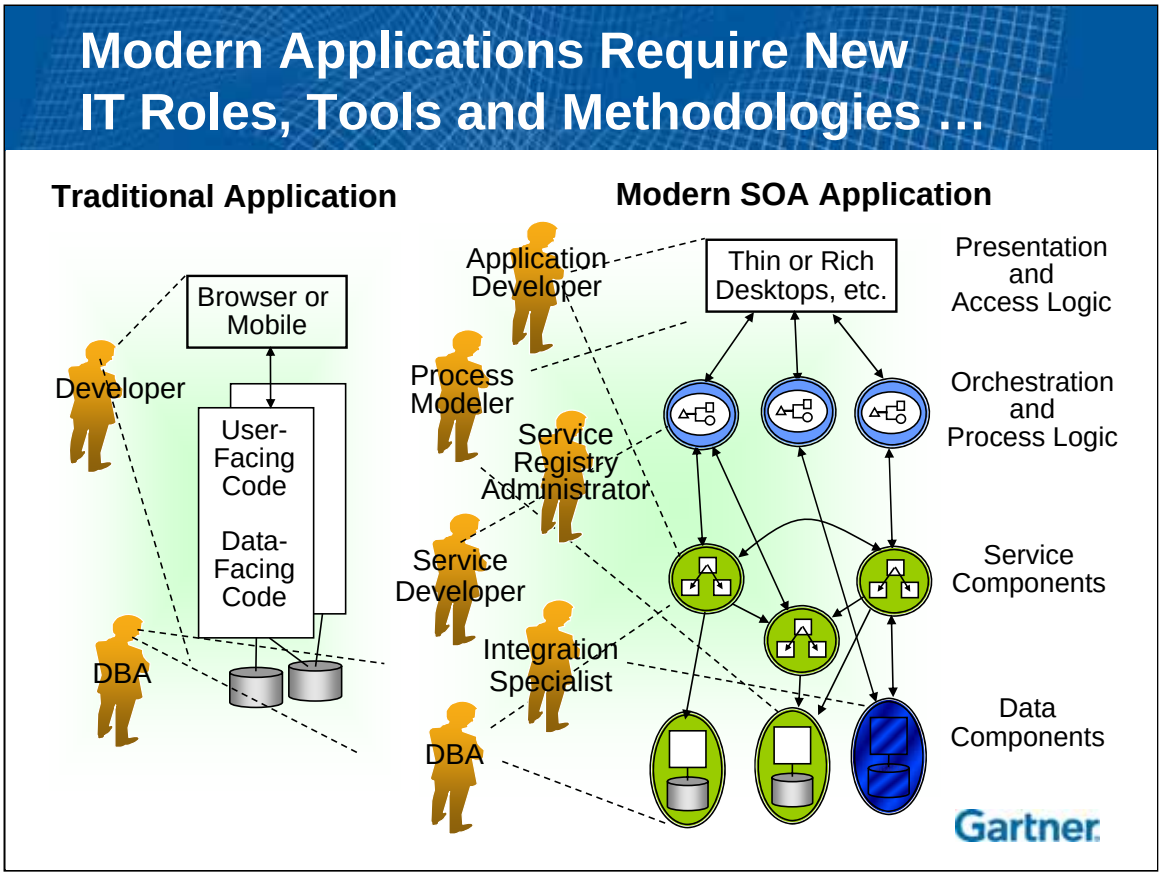
1. What is integration, and what is its relationship to business process execution, technology and infrastructure?
2. What are common and emerging integration problems, and what solutions will be available to solve them?
3. What strategy should your company adopt to guide process integration deployment and infrastructure?

Gartner

IT convergence is confusing application designers and IT architects. Applications now come bundled with development tools and integration features. Development tools are now used to implement applications, processes, components and legacy integration. Service-oriented architecture intersects business process management. This presentation explains the impact of IT convergence on distributed business applications, and will help you choose the right infrastructure for your next distributed application project.

Key Issue: What is integration, and what is its relationship to business process execution, technology and infrastructure?

Strategic Planning Assumption: SOA will be used, in part, in more than 50% of new, mission-critical applications and business processes designed in 2007, and in more than 80% by 2010.

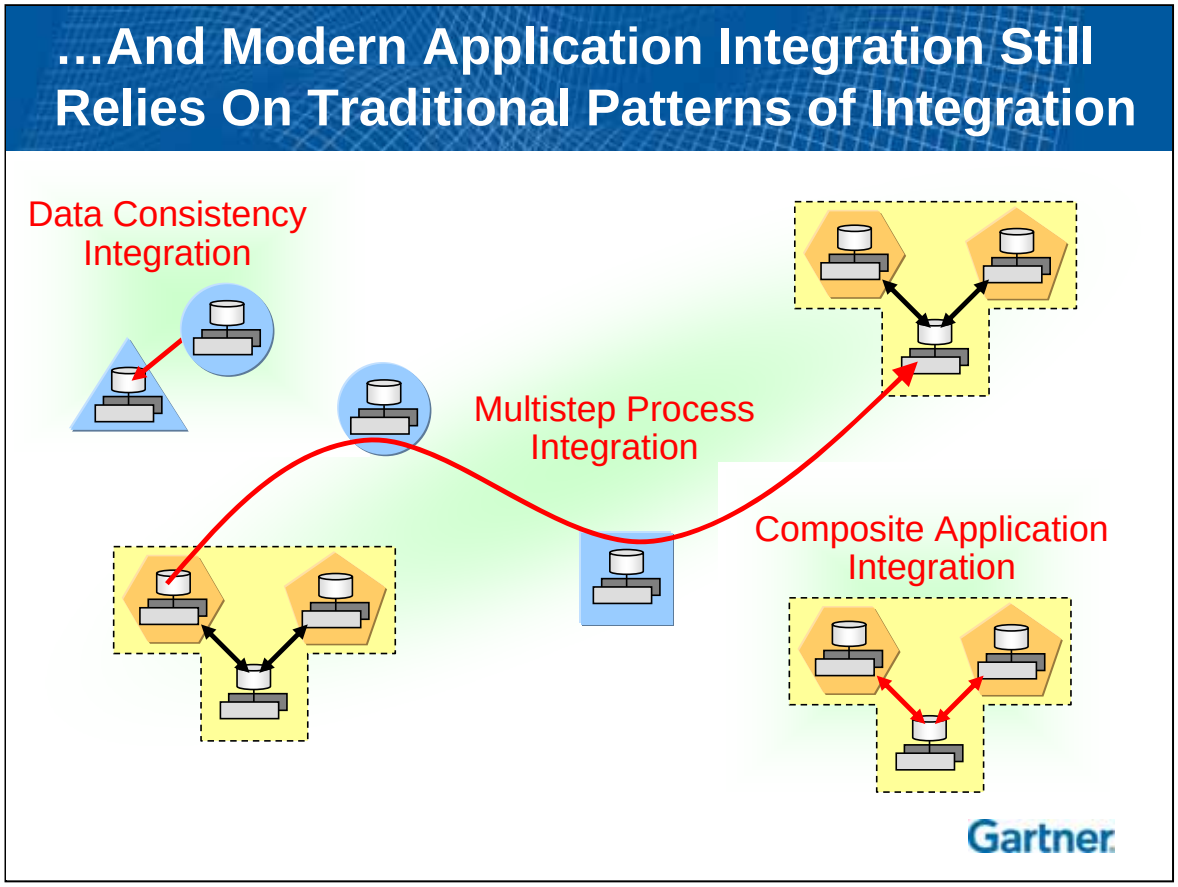


Business applications traditionally consisted of a few large programs that were hard to change. Monolithic applications were a custom-fitted sequence: end-user-facing presentation logic, business logic, data logic and data. In contrast, SOA is an architecture style founded on modularity and layering. Discrete functions are packaged into encapsulated, shareable modules (service components and data components) that can be invoked in a loosely coupled manner by multiple disparate local or remote consumer modules. The metaphor of a "black box" describes how encapsulation hides the internals of each component. An SOA service is the contract between a consumer and a service provider — it's the consumer's view of a provider's capability. SOA applications are developed in separate stages, generally by two or more kinds of developers.

Service and data components in the back-end layer implement sharable ("reusable") functions that encompass data and business logic. Other developers focus on the user-facing (presentation) layer and the middle integration layer, increasingly (but not always) orchestrating services in the context of business process modeling. SOA applications can be developed, extended and maintained in small, easily understood increments, facilitating an incremental approach to business growth and ever-changing business applications. The SOA domain starts small and gradually expands to handle additional tasks and business processes. Each application or business process might use one or many services — process complexity varies greatly.

Action Item: To prepare for modern application development, organizations should become familiar with SOA-enabling technology, service modularization, and relevant SOA best practices as soon as possible.

Strategic Planning Assumption: Through 2011, 75% of Fortune 1000 companies will continue to implement a combination of data consistency, multistep process and composite application integration patterns.



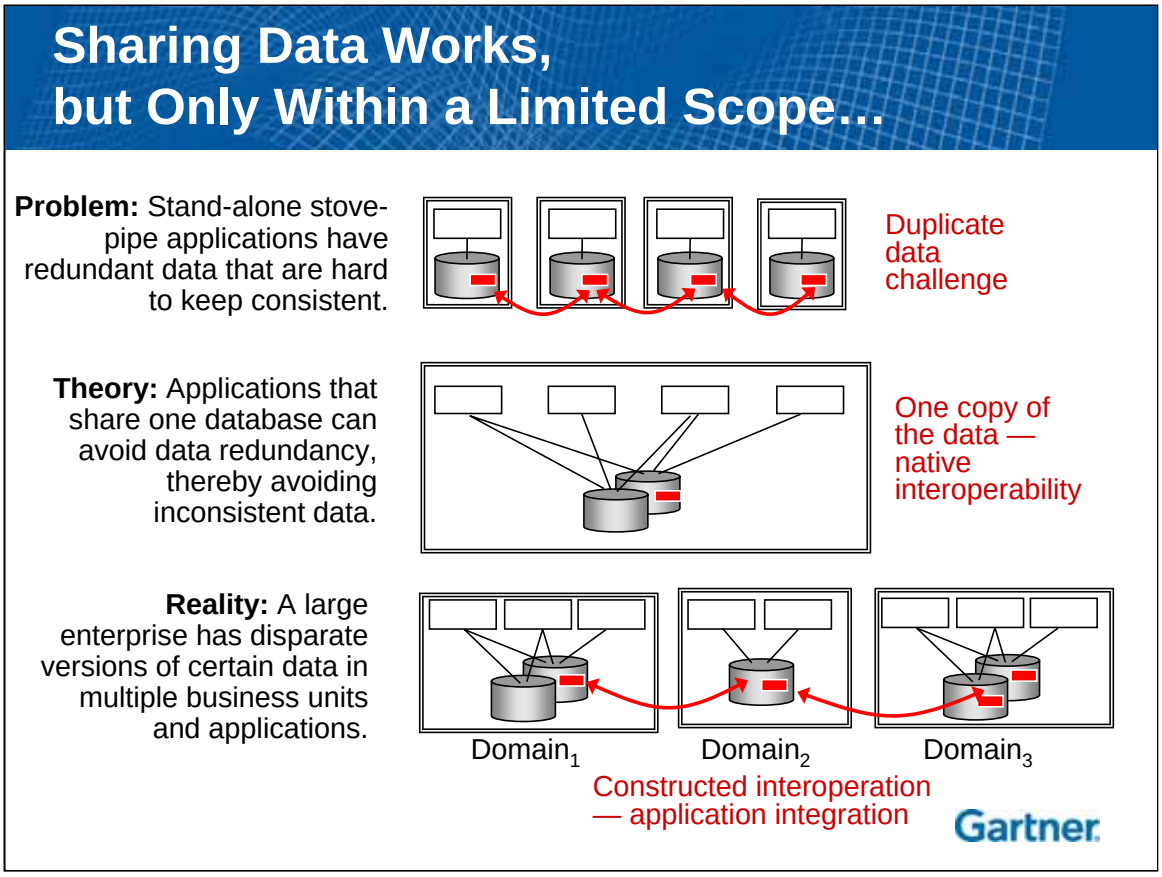
The most serious shortcoming in most companies' integration strategies is "compartmentalization" — focusing on one integration problem and simplistically trying to apply that solution to other integration tasks, which may require a different solution. There are three recurring patterns of integration problems that companies must solve, all which have existed for well over a decade and which companies will continue to have to solve for at least another decade.

The goal of *data consistency* — the most traditional and enduring form of application integration — is to get multiple application systems, often controlled by disparate business units, to agree on the facts. This is necessary when there are two or more databases holding information on the same subject. A *multistep process integration* project involves a sequence of related activities across multiple application systems, each relevant to a process (such as procurement). Each instance of a business process (for example, each order management or claims adjudication) has a life cycle that consists of steps that collectively take seconds, minutes, hours or days. *Composite applications* are closely knit applications that are bound together by a software assembly that implements one business function.

These integration patterns can be implemented discretely or they may overlap — for example, a multistep process may link to composite applications, each which may be part of a data synchronization project.

Action Item: Developers should be familiar with all three integration patterns and choose an appropriate solution for each, depending on the nature of the business problem (no pattern is right or wrong).

Strategic Planning Assumption: Despite improvements in data synchronization and master data management strategies, companies will have more redundant data in 2012 than they have in 2007.



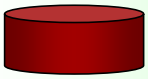
Modern companies have redundant versions of data on customers, products, employees and other important entities. Redundant data has obvious drawbacks, particularly with respect to synchronization. When one copy of the data is updated, the change must be propagated to all the other application systems that hold data on that entity. In theory, an enterprise should consolidate data into a single database to be shared in real time across all application systems. Database management system (DBMS) software enables this, and at least from a technical point of view (transaction atomicity, data integrity, concurrency control and durability of updates) such capabilities have existed since the 1970s. More recently, master data management (MDM) strategies have emerged which enable the consolidation of master data (such as customer or product data) into shared data services.

In practice, companies typically can only share data among a limited set of applications operating within one domain under the control of a single IT group. Many sound business reasons explain why it is often less expensive and more effective to continue to maintain redundant, overlapping copies of business data. For example, a purchased package may be predicated on its own data model and it could cost too much to modify the package to work with another database. Or a particular new application may require a variation on the data model because it needs new attributes. In some cases, data duplication results from poor development practices or irrational needs to control one's own data, but, in many cases, data duplication is pragmatic and sensible.

Action Item: Companies should migrate appropriate data consistency solutions from point-to-point batch jobs to master data management or message-based data synchronization, to improve the speed or quality of business process execution.

Strategic Planning Assumption: By 2012, batch synchronization will be used in only 50% of all IT projects involving master data, down from 80% in 2007.

...So This is How Companies Deal With Overlapping and Inconsistent Data

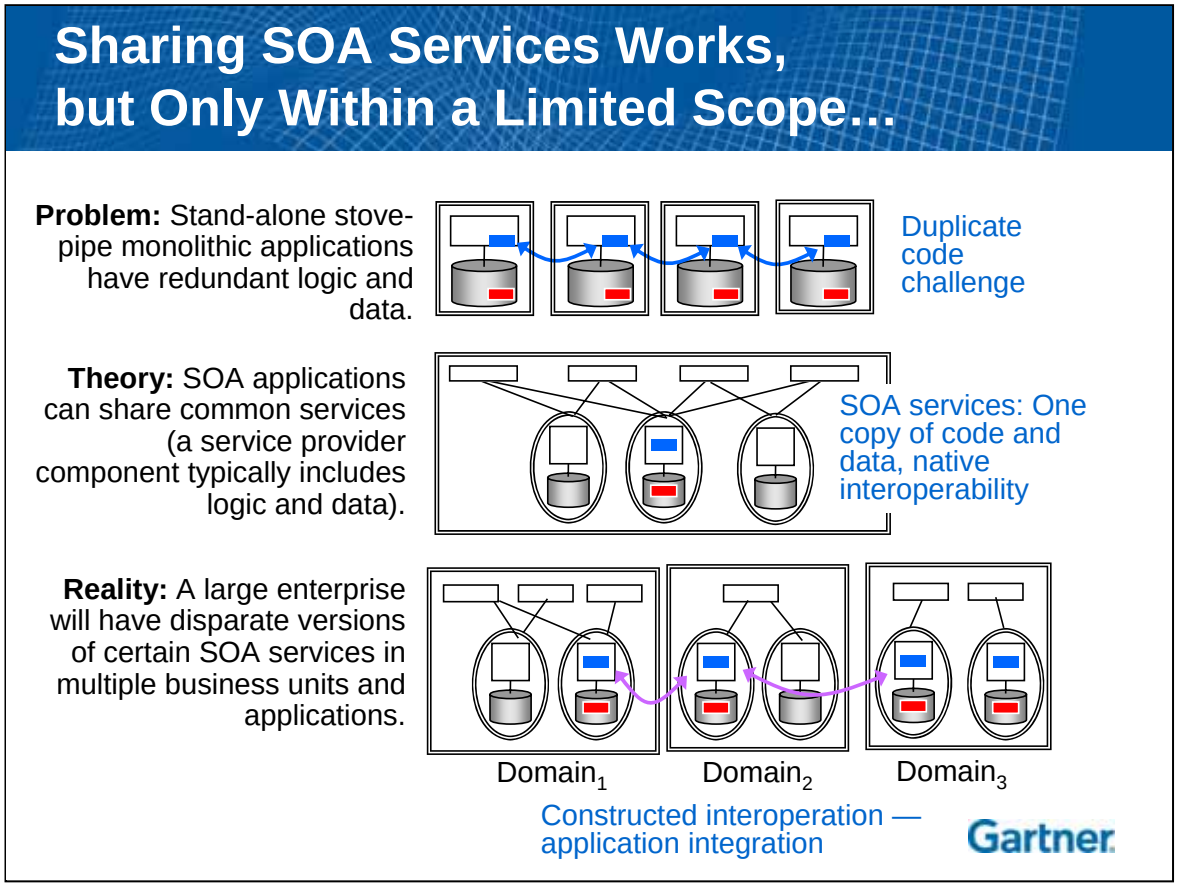
Approach	Using…	Adoption
 *Batch synchronization*	*Custom code, FTP, ETL, etc., used to synchronize data in bulk*	**80% → 50%*** In 2007 In 2012
 *Near real-time updates*	*Middleware, ESB, etc., used to synchronize data at record level*	**15% → 40%**
 *Eliminate the problem via master data management*	*No overlapping or inconsistent data, e.g., MDM as an SOA "data service"*	**5% → 10%**

* Of all IT projects that involve significant amounts of business data

Gartner

Data consistency is one pattern of application integration, and refers to overlapping or inconsistent master data found within different applications and systems. For decades, companies have dealt with data inconsistency only after applications and systems were already deployed, usually doing **batch synchronization,** either by developing custom logic to export, translate and import data, or by utilizing data integration middleware such as extract, transform and load (ETL) tools to accomplish the same thing. In the last 10 years, companies have responded to increasing pressure from business to maintain more **near real-time synchronization** of data. They have deployed a wide range of data integration middleware to implement this, including integration brokers, enterprise service buses, and ETL technology that has been enhanced to support events and near real-time updates in addition to supporting bulk data processing. Although these approaches have improved data accuracy, they have not solved the root master data problem — multiple versions of the facts. An emerging trend is for companies to implement "one version of the truth," i.e., **master data management**. In this scenario, companies centralize master data management, and require all participating applications to utilize the one data source — increasingly via "data services" published to a service-oriented architecture. While this latter approach is substantially better in terms of data architecture, challenges such as how to integrate with existing applications and systems will inhibit the adoption rate, and companies will often have to accept more pragmatic solutions such as a virtual federated database.

Tactical Guideline: Services should be shared within a domain of applications that have common business semantics and compatible service levels, but services cannot be shared where business requirements necessitate significantly different data or behavior in the service.

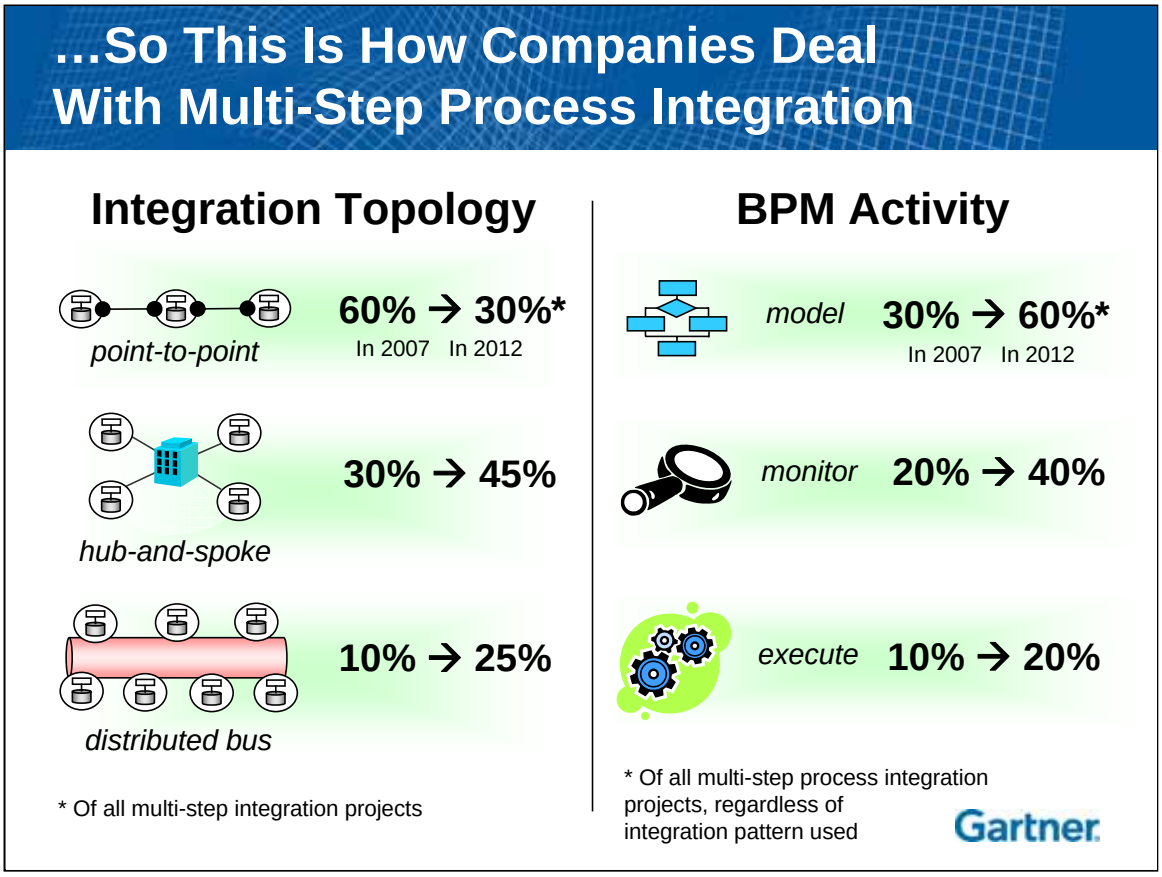


As a general goal, developers should attempt to share SOA services, rather than implement their own new versions of the services. Creating a new service rather than sharing a pre-existing similar service is not usually a reflection of a character defect in the software developers. Many good reasons exist to implement a new version of a service. For example, the effort required to modify a new packaged application to use a previously installed service in a legacy or packaged application may be prohibitive. It is often better to use the new version of the service that comes with the package. Or, when building a new custom application, the unique nature of the business requirements may demand the implementation of a service that has some overlap with older services, but which cannot be fully satisfied by the older service.

As a result of these factors, SOA only eliminates duplication of logic within a set of applications that is developed by one team or a group of cooperating teams; it rarely eliminates redundant logic and data across the whole enterprise. A set of SOA services (a "service domain" or "service community") is internally coherent because it grows incrementally, and its developers have knowledge of each other's interface specifications and data models when they design new modules, databases and processes.

Action Item: Developers should share services wherever practical, but architects and managers should recognize that some partial duplication of services is inevitable — often for good business reasons.

Strategic Planning Assumptions:
•An ESB will be used in 25% of all multi-step integration projects in 2012, up from only 10% in 2007.
•Explicit process execution (via a process state engine and rules) will be used in 20% of all multi-step integration projects in 2012, up from only 10% in 2007.



With so many platform and integration middleware products and approaches available, there is a lot of confusion about what is the best way to implement a particular integration project. In particular, for multi-step process integration, companies still use completely different architectural approaches ("integration patterns") and leverage significantly different levels of process implementation for different projects. **Point-to-point** integration is still the most common integration pattern because it's relatively quick and easy to implement, but is also brittle because there's usually no overall process definition. **Hub-and-spoke** integration is an improvement because is provides centralized management and control, but the hub can become a bottleneck for large-scale projects. **An integration bus** (typically an enterprise service bus) uses distributed nodes to support the infrastructure and can support larger-scale SOA projects, but requires more skills and effort than a hub-and-spoke approach. Regardless of what integration pattern is used, companies will leverage different BPM technology — in most cases today, still none at all. But companies are increasingly beginning to **model** processes — the purpose is to have one version of the truth among everyone involved. Companies are also rapidly beginning to layer business activity monitoring and other forms of **process monitoring,** to benefit from real-time visibility and to expose the key performance indicators of critical business processes. And to better control complex and frequently changing processes, companies **execute** business processes via an orchestration engine and business rules — a task that is a challenge to implement across IT systems, but one which increases process flexibility.

Key Issue: What are common and emerging integration problems, and what solutions will be available to solve them?

Strategic Planning Assumptions:

•Through 2009, 75% of large enterprises will have projects to resolve data consistency problems between applications.

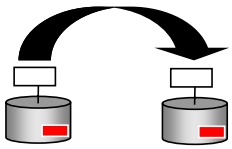
•By 2012, 70% of large organizations with heterogeneous customer and product data environments will be implementing or will have implemented an MDM solution.

Data Integration Tools for Data Consistency Problems

Extract, Transform & Load

Example (not all) ETL vendors:

- Business Objects
- CA
- Cognos
- DataMirror
- IBM (Ascential)*
- Informatica
- iWay
- Microsoft*
- Oracle*



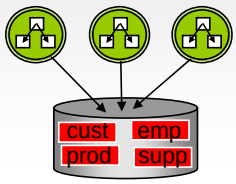
Functionality:
- Bulk data movement
- Transformation
- Adapters
- Task-specific metadata

Typical project scenario:
- Batch-oriented, data-centric integration
- Emerging real-time data integration
- Lot of point-to-point; some hub-n-spoke

Master Data Management

Example (not all) MDM solutions:

- Customer Data Integration
- Business Intelligence
- Data Warehousing
- Data Quality/Management
- Product Catalog
- Product Information Mgmt.
- Product Life Cycle Mgmt.



Functionality:
- Data modeling
- Information quality
- Workflow
- Batch and real-time integration

Typical project scenario:
- "Single customer view"
- Master data (and lots of it)
- SOA data "service"

* Mixed strategy

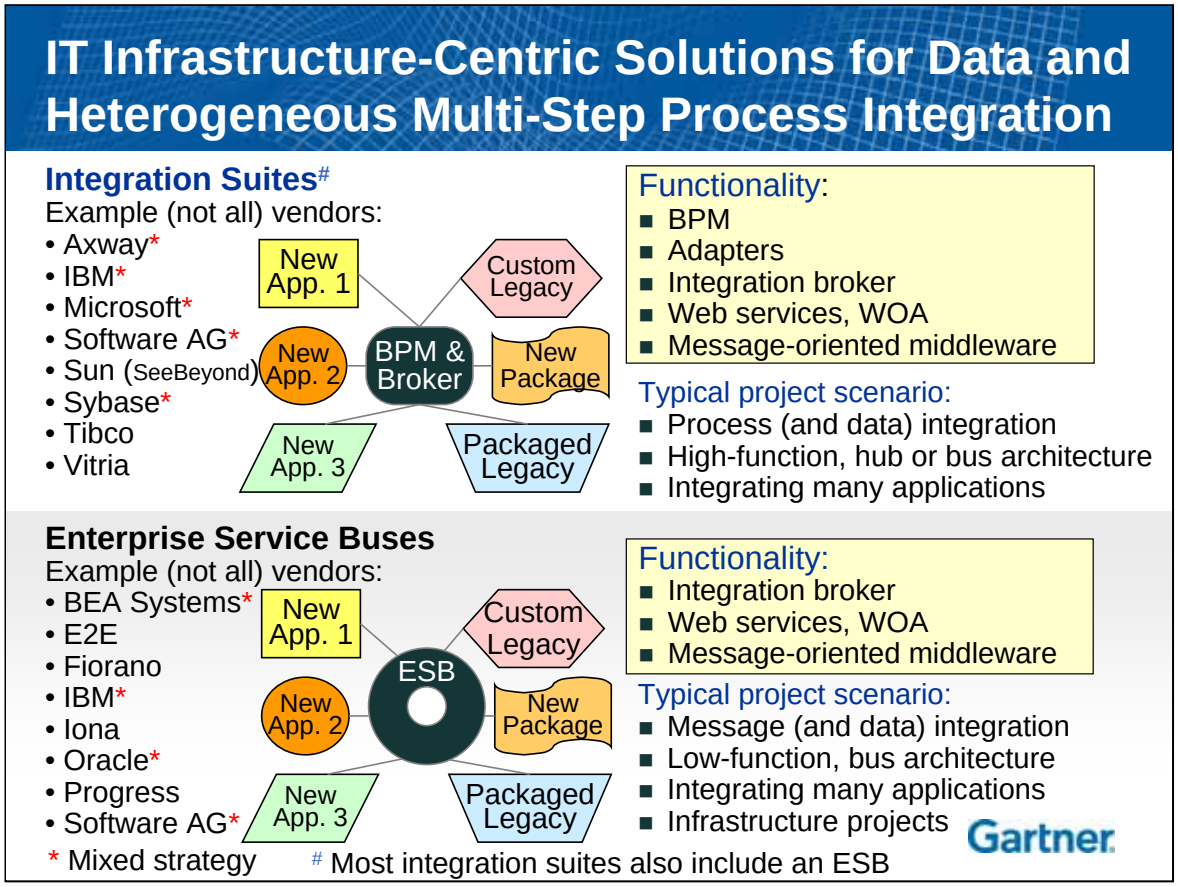
Gartner

Data consistency is a common integration problem involving application data in which two or more databases — typically associated with applications that were not designed to work together — must be kept in sync. Examples include synchronizing purchase orders between order entry and manufacturing applications, or using data from operational systems to populate a data warehouse. Extract, transform and load (ETL) tools have traditionally been used for batch-oriented acquisition, transformation and movement of data. Along with most forms of integration middleware, the ETL tool can be used to synchronize data in a more real-time, transactional mode. ETL tools and integration middleware are converging, such as in Sun's Composite Application Platform Suite (CAPS), which combines ETL with process integration.

Master data management (MDM) is an emerging market with no single MDM product category today, but several formal subsegments for customer and product master data (customer data integration hubs and product information management, respectively). Additional MDM solutions address other kinds of master data (for example, suppliers), or support different MDM functionality or points of entry (such as data governance) or downstream data management (such as data warehousing or product life cycle management). Regardless, MDM is focused on *managing* rather than *integrating* data. When possible, MDM is an architecturally preferred approach, but the challenge is ensuring that all applications and systems can leverage it.

Action Item: Whenever possible, leverage MDM to eliminate data inconsistencies, but be prepared to implement data consistency integration projects when implementing MDM that is not possible or practical.

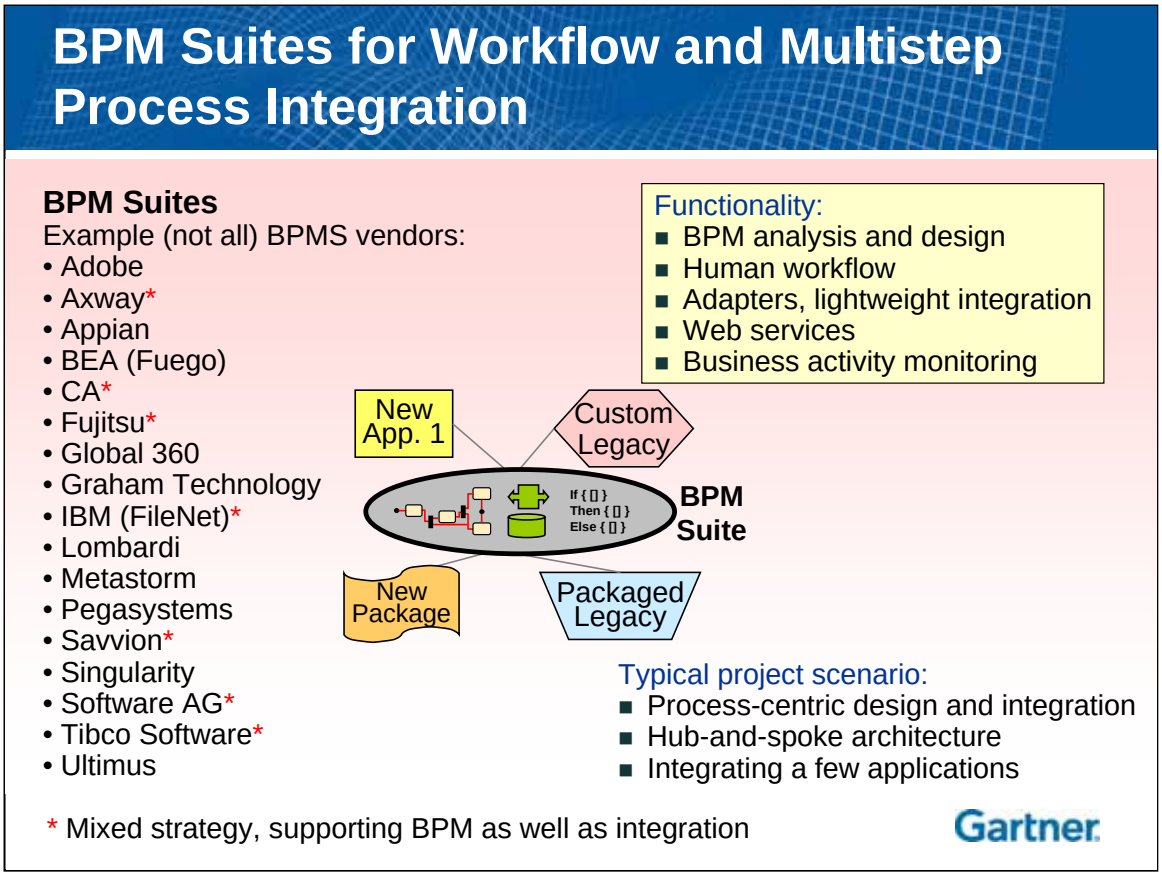
Tactical Guideline: All major integration suites and application platform suites include an ESB core, which can be leveraged when scaling up your integration projects.



Integration suites originally supplied just fundamental services: transformation, reliable messaging and intelligent routing (for example, content-based publish-and-subscribe). When these activities are performed in the broker, the participating application programs may not need to be modified to send or receive messages for integration purposes, which reduces the impact and disruption in the application. Vendors have added features to integration brokers, such as BPM, adapters and adapter development toolkits, Web services support, and better metadata and management facilities. Through this evolution, integration brokers have evolved into feature-rich "integration suites." Enterprise service buses (ESBs) have been available since 2002 and are an enabling technology for integration suites and SOA infrastructure, and also the backbone for event-driven applications and the integration of legacy applications. Although ESBs can be used like an integration suite as a backbone for large-scale integration projects, they tend to focus on large-scale infrastructure features (for example, security, load-balancing, failover, logging, performance and availability monitoring), but don't always include support for BPM and application-level functionality, including metadata management (such functionality is typically on top of the ESB). Integration suites and ESBs are more prominently deployed to support multistep process integration, used to link processes across multiple application systems, but these products are also often used to support data integration, helping to synchronize data to eliminate data inconsistency at a message or record level of granularity.

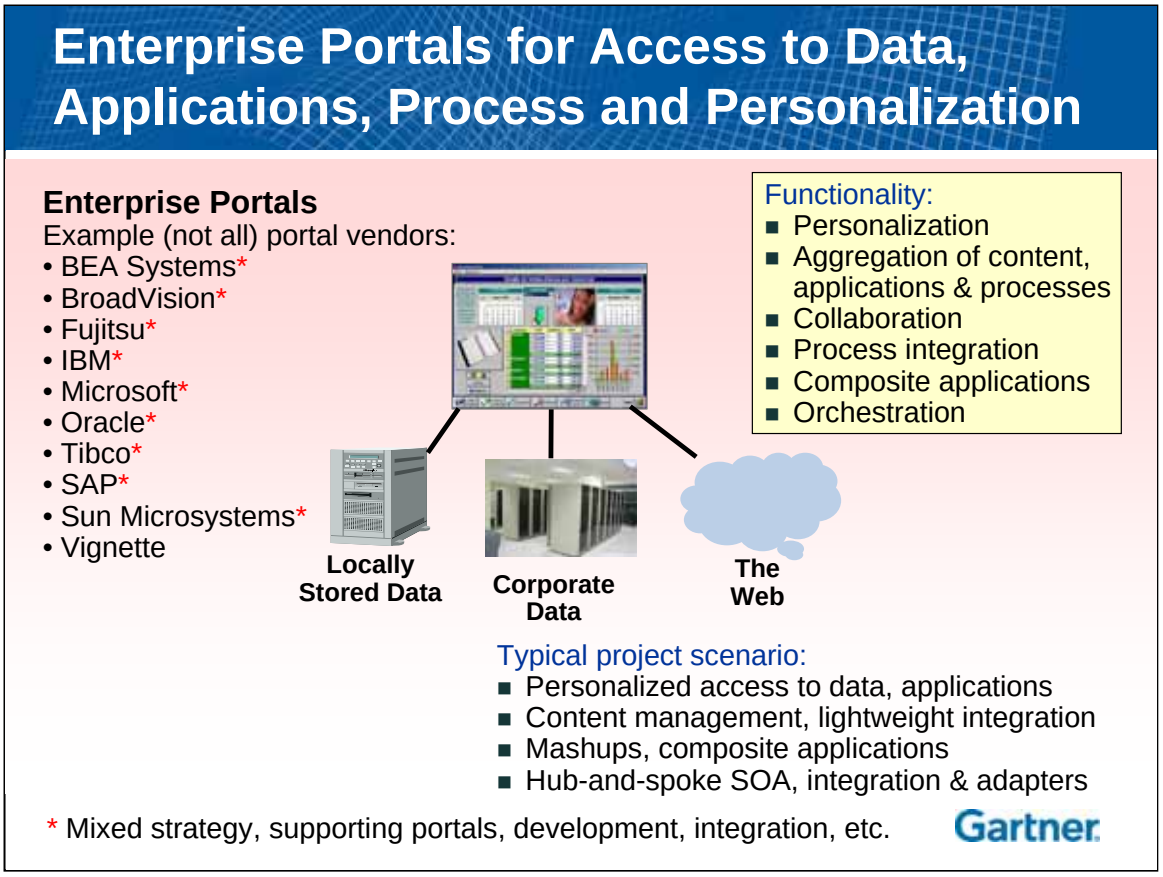
Action Item: IT managers and architects must try to understand the new varieties of integration middleware, because product features and packaging are changing quickly.

Strategic Planning Assumption: No more than 25 of the 150 BPM technology vendors in late 2006 will make the transition to the BPM suite market in 2008.



BPM suites represent the second generation of BPM-enabling technologies that have been packaged into a suite, rather than sold as independent, best-of-breed tools. Traditionally, BPM-enabling technology coordinates process interactions among people, systems and data as equally important aspects of work. In recent times, BPM-enabling technologies have evolved to support business' increasing desire to see and manage processes across organizational boundaries, and to give non-technical managers hands-on control of those processes. Modern BPM suites enhance the control and management of business processes to support collaboration between IT and business users. For years, BPM-enabling technologies have abstracted singular aspects of business processes. For example, human workflow automation products isolated the human interactions of processes by providing tools to better coordinate human activities. By putting controls around human tasks, these tools made these interactions more visible and, thus, more manageable by managers. However, some of these early products tended to ignore the importance of other business content and system-based work steps. On the other hand, traditional integration suites isolated such system interactions within processes by providing tools to better coordinate system interaction, but these ignored the importance of human contributions to the successful completion of work. In 2003, Gartner identified the emerging market for BPM suites that addressed process coordination in which people and systems are treated as equally important aspects of work.

Strategic Planning Assumption: By 2008, more than 75% of portal product vendors will deliver a service-oriented version.



Gartner defines a portal as a "Web software infrastructure that provides access to, and interaction with, relevant information assets (for example, information/content, applications and business processes), knowledge assets and human assets by select targeted audiences, delivered in a highly personalized manner." Enterprise portals may face different audiences, including:

• Employees — business-to-employee (B2E)

• Customers — business-to-consumer (B2C)

• Business partners — business-to-business (B2B)

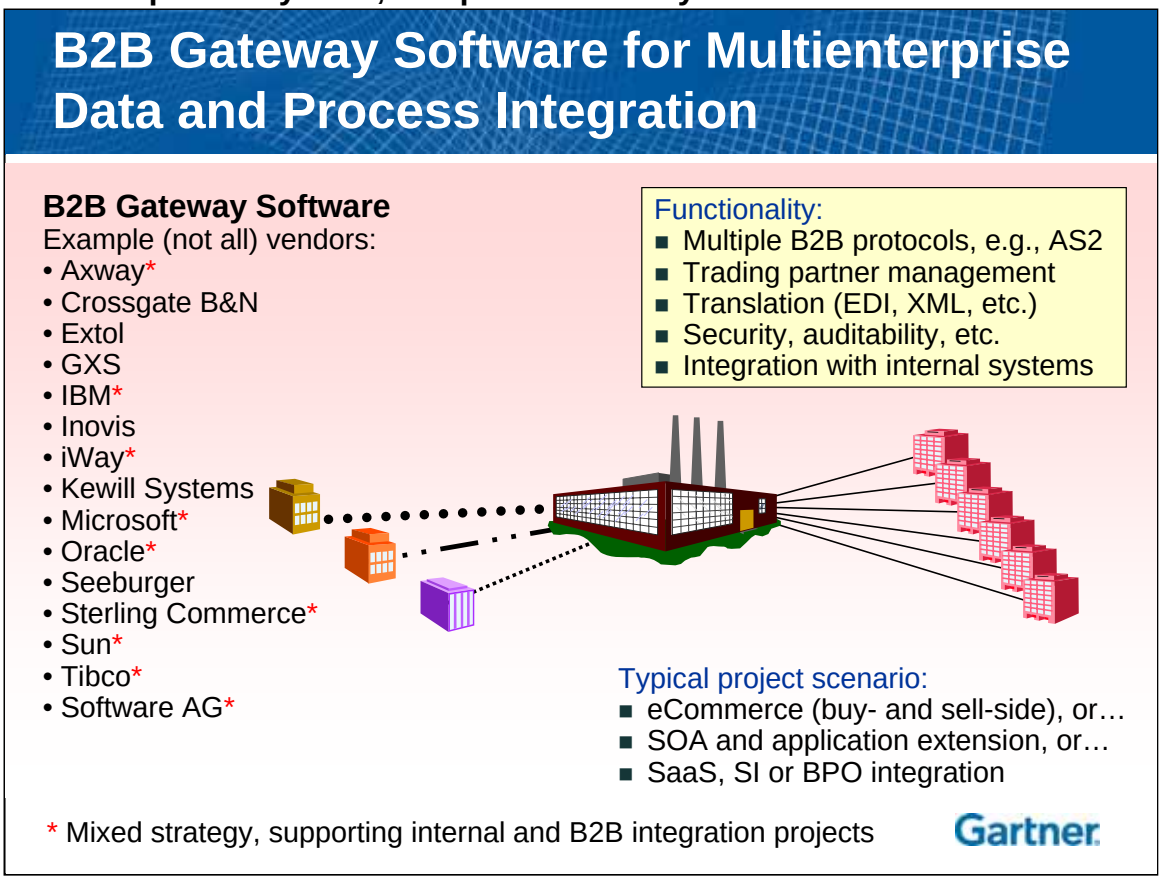
A portal product is a packaged software application that is used to create and maintain enterprise portals. These products can be used to design vertical or horizontal portals. **Vertical portals** focus on accessing specific applications or business functions. **Horizontal portals** seek to integrate and aggregate information from multiple cross-enterprise applications, as well as specific line-of-business tools and applications.

Most companies looking for portal functionality will deploy horizontal portal products, although SESs and APSs also contain vertical portal frameworks. Portal services are increasingly embedded in multiple product types, combined with packaged applications, and an increasing number of enterprises will use these other product packages to obtain portal functionality. Portals are often the hub in SOA and integration projects, but are not generally used as the "SOA Backplane" for large projects such as an ESB.

Strategic Planning Assumptions:

•By 2011, midsize-to-large companies will at least double the number of multienterprise integration and interoperability projects they're managing, and will spend at least 50% more on B2B projects, compared with 2006.

•Web services will be part of a multienterprise integration and interoperability strategy in 75% of Global 2000 companies by 2011, compared with only 20% in 2006.



B2B gateway software is used to implement multi-enterprise infrastructure so companies can exchange business data (for example, customer or item data) or link and automate business processes (for example, order-to-cash or procure-to-pay) between two or more companies in a way that is easier to manage, faster, more affordable and more accurate than manual approaches or custom coding. Multienterprise projects at midsize-to-large businesses are proliferating, rapidly:

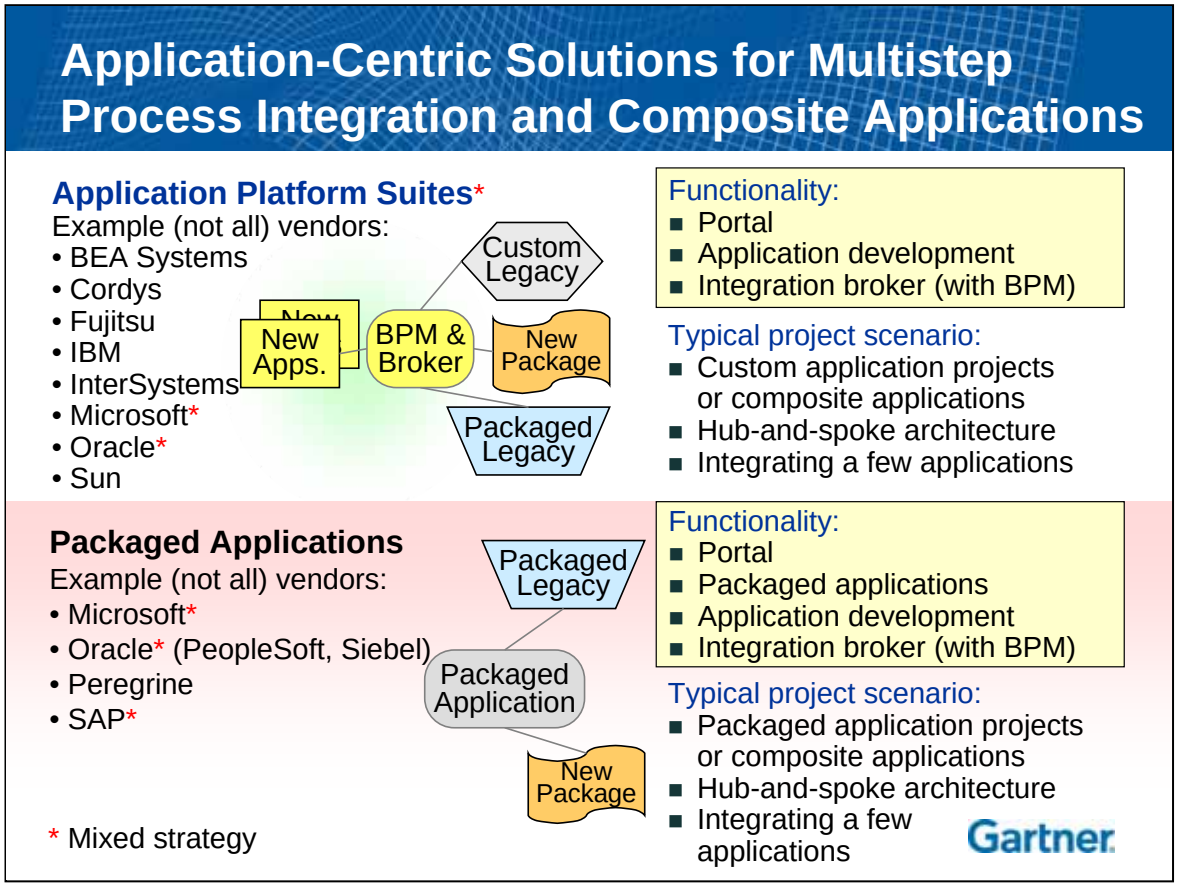
E-commerce projects are the most traditional form of multienterprise infrastructure project and focus on automating and managing buy- and sell-side trading partner networks. *ERP extension* is a frequently occurring form of multienterprise infrastructure project in which a company links its internal ERP or other applications to its external business partners using SOA or some other approach to integration, such as AS2 or batch file transfer. *SOA extension* is a rapidly emerging form of multi-enterprise infrastructure project in which a company links its internal SOA to its external business partners' SOA, typically using Web services. *B2B consolidation* is a project where a company consolidates multiple stand-alone B2B projects onto a shared (common) B2B infrastructure. All of these forms of multienterprise project may involve suppliers, customers, vendors, system integrators, providers of SaaS or BPO, or any other external business.

Action Item: Midsize to large companies should plan to implement twice the number of B2B projects and spend 50% more on B2B in 2012 than in 2007.

Strategic Planning Assumptions:

• By 2008, packaged composite applications will account for at least 50% of the systematically designed composite applications deployed.

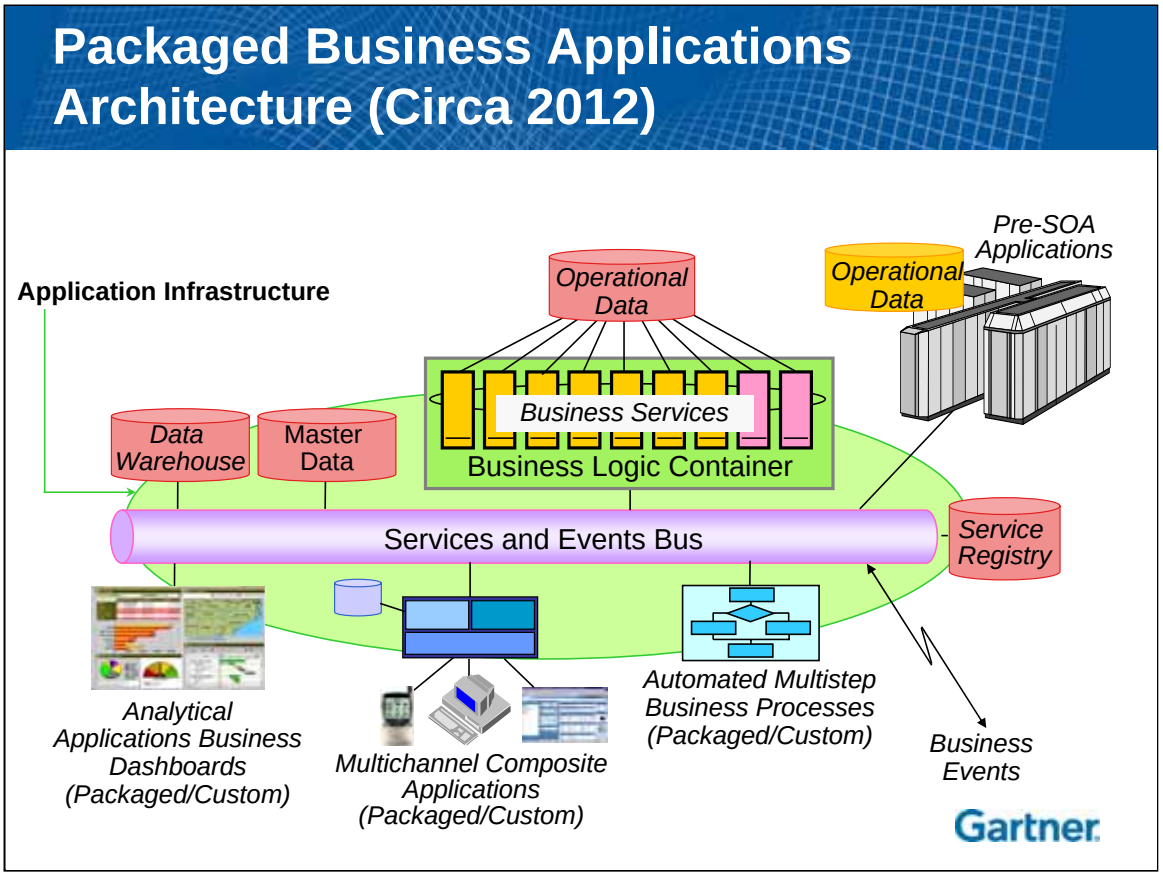
• By 2008, users will leverage infrastructure from business application vendors to build (or assemble) more than 35% of their composition applications.



Application platform suites (APSs) combine an execution platform (such as an application server), an integration platform (such as an integration suite) and a user interaction platform (such as a portal product). However, a full-function APS, even while serving its three core functions, really consists of more than three technology stacks. Most APSs now include an integrative layer of common development, metadata, security and system management. Many vendors (for example, SAP, Oracle, Cordys and IBM) extend their APS with a library of pre-built, reusable business components. These range from semi-technical application building blocks (Cordys) to complete applications (SAP).

Business applications are almost always complemented by some custom-developed functionality because no packaged application (even from multiple vendors) provides everything needed. The combination of packaged application logic, along with the emerging integrated set of composition technologies and a business service repository, gives users a much more flexible way to extend packaged business applications and "blurs" the lines between the buy and build approaches to application portfolio development. Modern packaged applications come with associated application development and integration infrastructure (application development, application integration, portal technology and metadata repository) that make it possible to accommodate rapid, but controlled, business process change through the use of integrated process composition technologies and reusable business process components, in addition to package application logic.

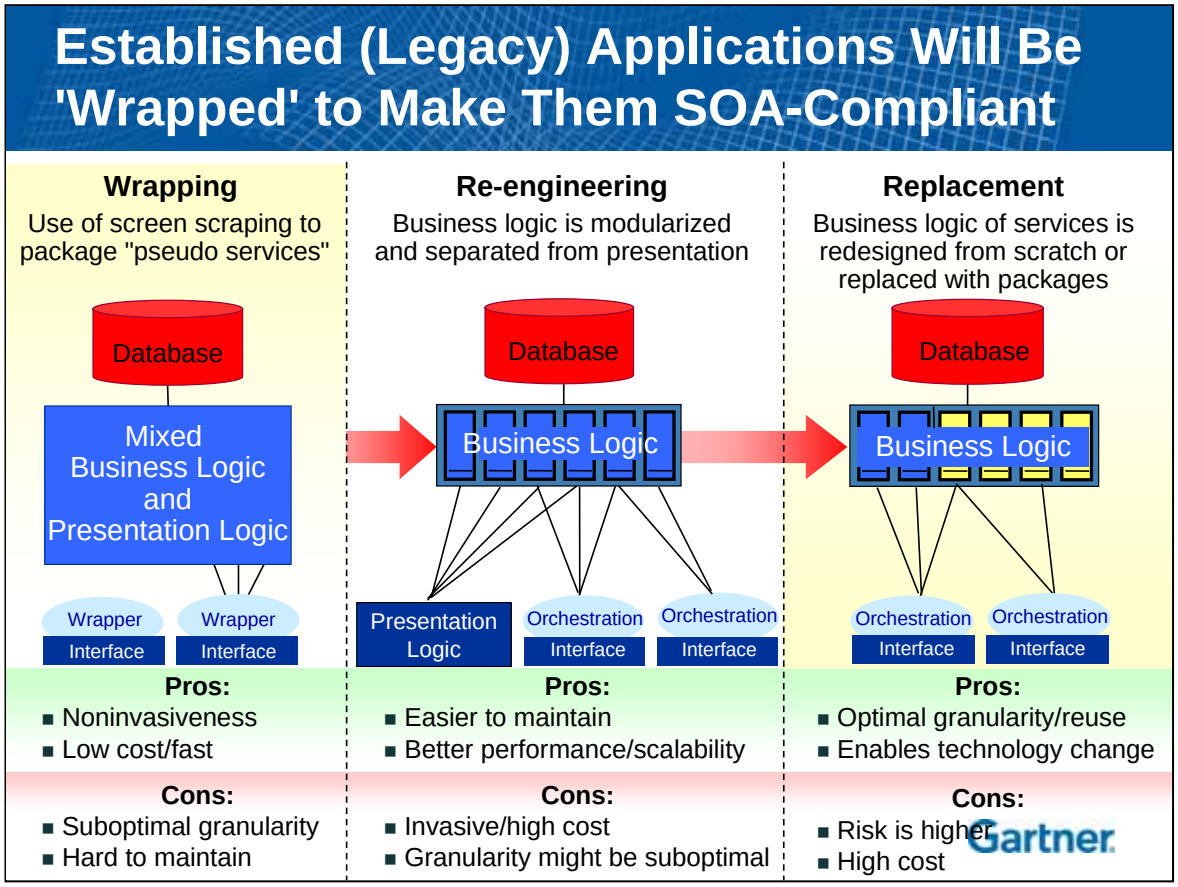
Strategic Planning Assumption: By 2011, more than 65% of packaged applications' users will adopt "SOA-enabled" versions of their core business application products, but more than two-thirds of them will be motivated by the need to access new application functionalities, not by the desire to enjoy SOA IT and business benefits.



SOA benefits span business and IT, thus pushing users to adoption, despite of significant and hard to overcome obstacles. Therefore, in 2007 organizations adopt SOA by carefully balancing benefits is terms of greater business agility and costs associated with cultural change and greater technical complexity. But SOA adoption will be less often an autonomous users' decision. Increasingly vendor-specific SOA renditions will be introduced in a "stealth" mode as a consequence of SOA adoption by packaged-application providers. As new, SOA-enabled releases of popular packaged business applications hit the market, users of these products will gradually move to the new versions, mostly to take advantage of new add-on application functionalities provided by the vendor and its partners. As a consequence, users will necessarily have to adopt their particular vendor's (or vendors') rendition of the SOA principles and enabling application infrastructures. Vendors' SOA-enabling strategies will have profound implications in terms of packaging, go to market, partnership and delivery model. Large user organizations will likely have to support multiple, partially incompatible SOA infrastructures, each provided by a different packaged-application vendor. The role of technologies capable of federating these multiple SOA renditions will be paramount. In particular, federated service registries and metadata management technology will be extensively adopted as "packaged services" will be most of in-place, registered services for many users.

Action Item: Packaged-application users should familiarize themselves with SOA-enabling technology and plan for filling their SOA technical, skills and organizational gaps in preparation for a migration to new, SOA-enabled versions of their packaged applications, if they want to enjoy the benefits of SOA.

Strategic Planning Assumption: Through 2008, at least 65% of custom-developed services for new SOA projects will be implemented via the wrapping or re-engineering of established applications.



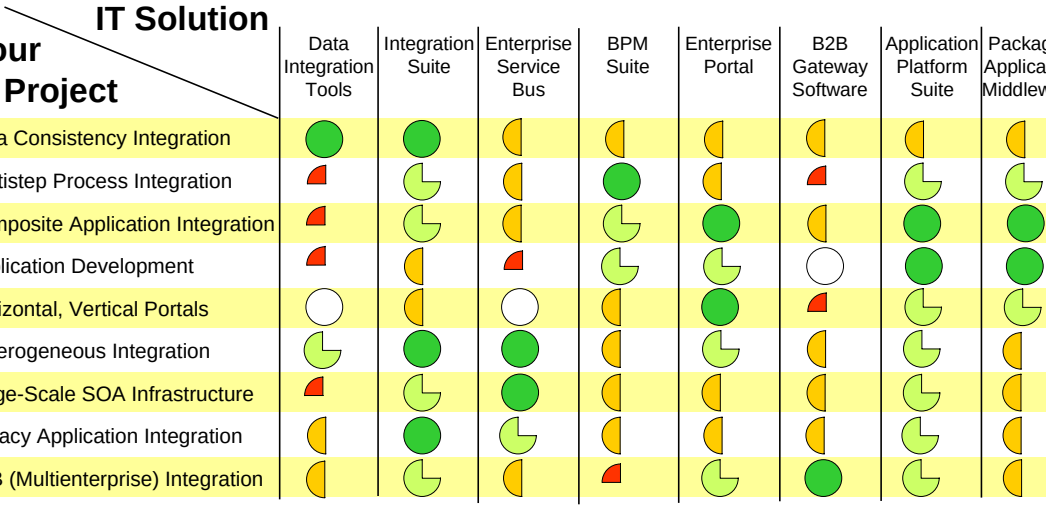
SOA applications are rarely designed from scratch. Most SOA projects are implemented by combining established applications and new services. SOA's ability to combine old and new is part of its power and one of its virtues. Many organizations embrace SOA to breathe new life into legacy applications by making them more flexible and easier to integrate. This often requires re-engineering of the legacy application to separate out its presentation logic from the business logic and to partition the latter into modules. This approach is clearly expensive and can be risky, but usually leads to greater maintainability and paves the way for future redesign of services (when and if needed) as an example atop a different application platform, such as Java EE or .NET. The cheaper, faster and less-invasive way to service-enable established applications is by using integration technology (for example, portals or programmatic integration servers) to wrap applications' business logic into "pseudo services." Often, this is achieved by "orchestrating" multiple legacy transactions to implement services of proper granularity. Wrapping has limitations in terms of performance, scalability and maintainability, but it is effective in the "introduction" stage of SOA, when organizations must prove the concept and demonstrate value. In many cases, wrapping is the only practical approach (for example, if the application is a package users cannot modify), and often a first step toward more substantial re-engineering or even redevelopment/replacement of the application.

Action Item: To effectively implement SOA, organizations should familiarize themselves with application integration technologies and methodologies.

Key Issue: What strategy should your company adopt to guide process integration deployment and infrastructure?

Strategic Imperative: Midsize to large companies should plan to implement at least three different types of platform and integration middleware to meet their IT project requirements for deploying and integrating business applications.

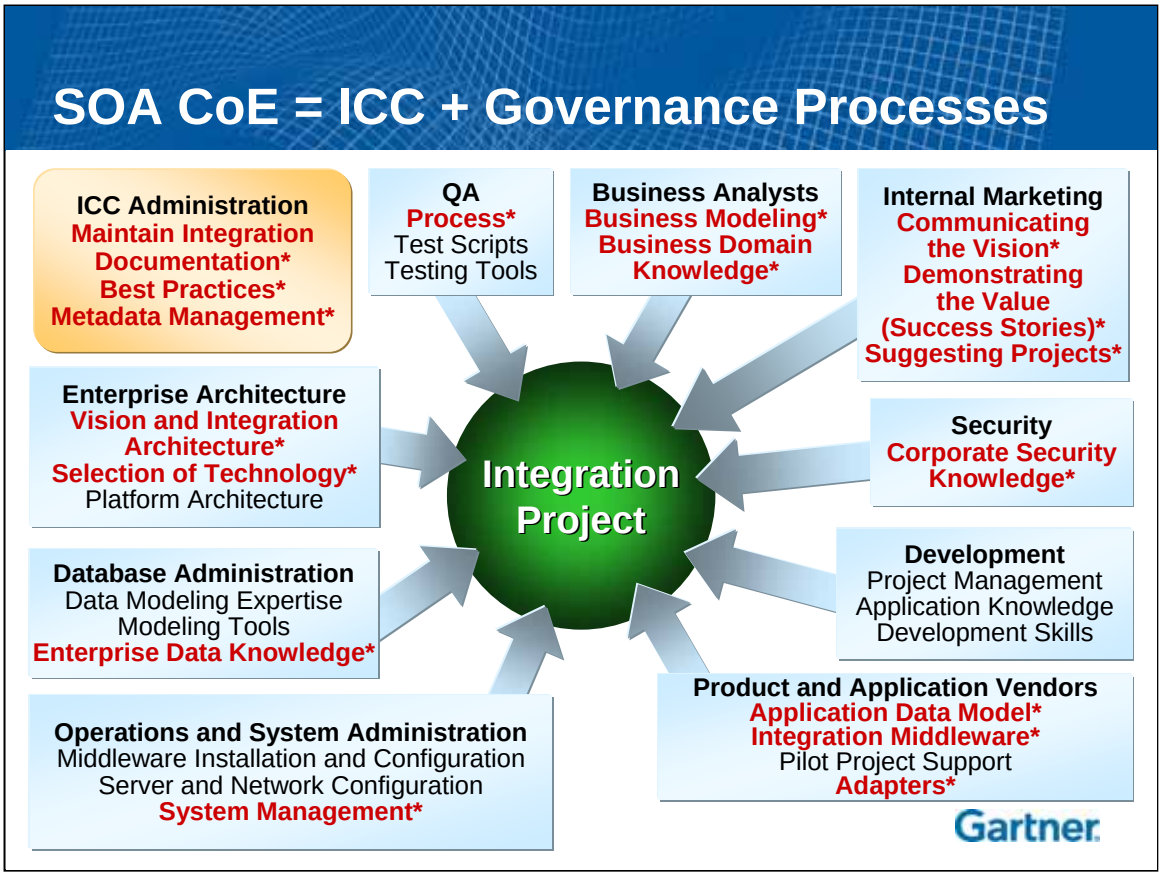
One Size Does Not Fit All — Match Your IT Project Requirements to Best-Fit Solution



It should come as no surprise that midsize to large companies have so many different application deployment and integration requirements that they need to combine several best-of-breed solutions. Yet, it is still tempting for system architectures to mandate "one-size-fits-all" IT infrastructure to meet diverse project requirements. The reality is that although some IT solutions, such as integration suites, are well-rounded and can support a lot of IT projects, specialized solutions — such as ETL tools, portals or B2B gateway software — are required to best address specific IT project requirements. Therefore, companies need to balance the appeal of a common IT infrastructure (to benefit from economy of scale, for example) with the recognition that — sometimes — a project, such as data synchronization, will not be optimized on the common infrastructure, so a different, best-of-breed, solution will be the better choice.

Tactical Guideline: Many decision types (architectural, organizational, technical and strategic) are best taken by different groups of people, all linked to the Integration Competency Center (ICC).

Fact: The ICC already contains the roles and responsibilities necessary to drive a successful SOA.



The right IT organizational structure and design practices can ameliorate problems arising from diverse applications/platforms. Large companies rarely avoid having different enterprise service buses (ESBs), integration suites and application platform suites, even if they try to standardize on one, but they can simplify cross-domain interactions by centrally documenting and managing integration metadata. Well-run enterprises implement centralized or federated ICCs, generally consisting of six to 12 full-time employees (but may be 0.5 to 150 people). The development side of each ICC provides expertise to the application development (AD) teams for integration. It also maintains common integration standards and metadata and WSDLs for Web services messages. The operational side installs/maintains integration infrastructure. Most young ICCs centralize integration work for the organization, using governance policies. As ICCs mature, the integration work needed and time pressures make them grow. In strong governance organizations, demand for application integration can be satisfied by internal growth of ICC resources. A central corporate ICC controls (and delivers) integration work into maturity. Usually, the evolving, consolidated ICC simply can't cope with the workload and time constraints of the projects it has to deliver. At that point, the ICC gets leaner, maintaining key training, governing and mentoring skills, and offloading skills to other teams in the IT department or outside the company.

Action Item: Create a full-time, centralized or federated ICC for the enterprise nervous system, including middleware technology, SOA service definitions, and event-driven architecture message schemas.

Recommendations

- ✓ Deploy a competency center that holistically develops your application, data, BPM, SOA, SaaS and integration skills.
- ✓ Choose middleware that is optimized for your IT infrastructure project's "center of gravity": application, data or process.
- ✓ For your largest IT infrastructure projects, combine best-of-breed BPM technology with best-of-breed SOA and integration middleware.
- ✓ Data synchronization or master data management – pick one!
- ✓ Use integration middleware to "connect" BPM and SOA projects.
- ✓ Use an enterprise service bus to scale up SOA infrastructure.