
ENTERPRISE JAVA BEANS™ TECHNOLOGY

A BUSINESS BENEFITS ANALYSIS



By
Zona Research, Inc.
June 1999

Copyright © 1999 Zona Research, Inc. All rights reserved. No portion of this document may be reproduced without prior written consent. The information and statistical data contained herein have been obtained from sources that we believe to be reliable, but are not warranted by us. We do not undertake to advise you as to any changes in the data or our views. Zona Research, Inc. and its affiliates and partners, or members of their families, may perform services for, and/or engage in business with, and/or hold equity positions in one or more of the companies referred to in this document, or their competitors. Zona Research, Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material. Trademarks mentioned are the property of their respective companies.

ENTERPRISE JAVA BEANS TECHNOLOGY

Abstract: Enterprise JavaBeans™ (EJB) technology is an emerging part of a fundamental software architecture that attempts to combine the best of Internet technology, distributed object architecture, and platform-independent coding into a better way of creating and maintaining E-commerce, corporate and personal applications. Announced as a specification effort two years ago and as a specification one year ago, serious EJB-based applications are now making their appearance in corporate Web applications.

This paper highlights the technical differences between EJB architecture and other software approaches, and focuses on the business payoffs that IT managers have seen and expect to see in their first generation of EJB technology-based applications. Zona Research conducted in-depth interviews with IT managers in the finance, telecommunications, enterprise resource planning, and transportation industries in order to extract their assessments of the positive and negative impact of EJB technology on their businesses. These case studies are also presented in this paper.

TABLE OF CONTENTS

Introduction: What is a Business Benefits Analysis?	5
What is EJB Technology?	5
What Difference Does It Make to Real Businesses?	6
What Are the Benefits?	7
Some EJB Efficiencies Calculated by Real Users:	7
Case Studies	8
Instinet Corporation	8
Federal Aviation Authority (Seattle Air Space)	10
SCT Corp.	12
Covad Communications	14
Conclusion: Future Developments and Areas of Improvement	14

Introduction: What is a Business Benefits Analysis?

When a technology has been out in the industry long enough, one can do an analysis of the complete life cycle of the technology and come up with a total cost of ownership (TCO) study. Before that, there is also a time in which a technology can be fit into a formal return on investment (ROI) model, and one can apply formulas to calculate the effect of a technology on a given business process.

Zona Research has done TCO studies and it has done ROI studies, but we found that much, if not most, of the technology that we are asked to assess is brand new, first-generation technology. These technologies are usually too young to have established the basis for a formal ROI model. It is these technologies, such as the Enterprise JavaBeans technology portrayed in this paper, that often represent the potential for largest gain in the shortest time, but they drive IT managers to make difficult, intuitive decisions. If they wait until the technology is established enough for an ROI, their competitors may have already used it, and the window of opportunity can then belong to the competition. If they use it too soon, will they suffer the slings and arrows of non-standardization, and possibly of unforeseen delays as new technologies are worked out?

The answer lies somewhere in between, and we took the direct path to get it. We asked the early-adopting IT managers represented here what rationales they used, how they justified use of the EJB technology to their bosses, and what, if any, tangible parameters could be used for measuring the success or failure of their initiatives.

What is EJB Technology?

There is Java™ technology, there are JavaBeans™ components, there are Java servlets, and there is Enterprise JavaBeans component architecture. Enterprise JavaBeans technologies are the newest outgrowth of the Java technology-based development movement. They are components that allow applications to communicate across multi-tiered client and server environments, and across Internet and Intranet structures. It may well be that Java, JavaBeans, Java servlets, and the EJB technologies together constitute the most fundamental architectural change in software development for at least the next five years

When there is a need for distributed objects to communicate with each other across either the client-server barrier or the server-server barrier, then Enterprise JavaBeans technology comes into play. It enables developers to implement N-tier architectures in their applications, and to derive the benefits of such an architecture. An N-tier architecture is one in which pieces of an application, distributed across the client tier, the middle-tier server, and the enterprise backend, function together to partition tasks in the most logical fashion. We use the term N-tier rather than 3-tier, because both the middle-tier and backend portions may consist of numerous servers calling servers, and servers calling databases, so that each of these partitions may consist of many tiers.

The ability to partition applications into distributed tiers gives developers the flexibility to let each tier do what it does best. Instead of requiring “fat client” applications on the client systems, for example, a developer can allow the middle-tier servers to do the heavy lifting of accessing remote databases, handling transactions, and aggregating information. This model replaces the earlier two-tier client/server model that required the maintenance of numerous drivers on the client system to access enterprise applications.

The middle-tier grounds have also spawned a robust field of Web application servers acting as the application middleman between browser-based clients and corporate backend systems. A long list of vendors, from Sun’s own NetDynamics™ servers to Netscape, Persistence, Bluestone, BEA, IBM, Oracle, Gemstone, Progress Appitivity, Novera, Secant, CompuWare, Silverstream and others, have announced support for Enterprise JavaBeans technology. These Web application servers act as a fertile middleground, extracting data from ERP systems and other enterprise data stores, adding middle-tier business logic processing, load-balancing, caching, and failover capabilities to distributed applications. The idea of using EJB technology fits well with this distributed application model, and we expect to see increasing levels of EJB technology support appear on nearly all Web application servers moving forward.

The Enterprise JavaBeans component model enables the application developer to work on the business aspects of the application without having to focus on transaction management, security, persistence of data or the life cycle management of the object itself.

In the past, programmers themselves have had to write the rollback routines, or at least know the interfaces to complex transactional systems like CICS and Tuxedo. With EJB technology, business logic takes on a more recognizable face. Units of work can be defined in terms that make sense to non-programmers, because they reflect the workflow of the physical business processes that the program is modeling. To the IT manager, this translates into having the ability to assign different people to different tasks that match their skills, and to model the application around recognizable objects of functionality that resemble their corresponding physical world business processes

What Difference Does It Make to Real Businesses?

A recent Zona survey showed that 85% of companies polled either have or are planning to deploy Internet or Intranet applications within 24 months. Only three years ago, that pie chart was reversed, with only 15% having such deployments and plans.

EJB technology brings about the promise of getting new distributed Web applications, and their resulting products and services, to market three to four times faster than with previous means such as C/C++ programming and CORBA objects.

Particularly in the financial services markets, competition has triggered a race to create and modify new software. New financial instruments mean new products, and today that

is synonymous with a new software application. The ability to modify a program quickly so that it works differently and presents a new financial instrument is becoming as important in the generation of new business as the time to market of the original application itself.

In the telecommunications market, the classic example was MCI's Friends and Family marketing program. It took AT&T more than eighteen months to create software with similar associative marketing characteristics, and by that time MCI had scooped up a whopping 15% of AT&T's market. This was done with earlier object-oriented programming, but the same lesson applies as many vertical markets suddenly turn themselves to the opportunities presented by 150 million browser-based customers.

What Are the Benefits?

Across the interviews that Zona conducted, the most popular responses to the question "What are the business benefits of EJB technology?" were time to market, programmer labor savings, manageability of code, reusability of code across multiple platforms, and reduction of "scarce skills" problem.

How can one measure the importance of "Time to Market" to a business? At one extreme, there is the Benjamin Franklin credo that begins with "For want of a nail, the shoe was lost. For want of a shoe, the horse was lost." and ends up with the loss of an entire war. Many corporate managers feel that they are at war with their competitors, and that the missing of identified windows of opportunity are just as crucial to a company's survival as to an army's.

Where we were able to get quantified results from the interviewees on time to market, they were often coupled with programmer man-hour savings and overall project completion time.

Some EJB Efficiencies Calculated by Real Users:

- **Instinet**, a wholly owned subsidiary of **Reuters Group PLC**, estimated that its Fixed Income electronic brokerage service, which has taken less than a year to create using EJB transactional application server technology, would have taken more than twice as long to bring to market with traditional programming environments.
- **Federal Aviation Authority (Seattle Air Space)** estimated that using EJB technology saved them as much as half of the programming time on the entire project to create a set of telecommunications order processing applications.
- **Covad Communications** determined that it could get a new application to market three to four times faster using EJB technology than by using CORBA objects.

The manageability of code and the reduction of the scarce skills problem were often lumped together by the IT managers we interviewed. One IT manager pointed out that “creeping featurism”, a factor that all IT departments must deal with, where users ask for more and more functionality, is easier to handle with the reusable component approach of EJB technology. Because the logic is broken down into smaller parts, features can be added in smaller batches, and revised more frequently. This reduces, although does not eliminate, the political battles that must be fought over which department’s features are incorporated into a given release of an application.

The scarce skills problem was particularly noticed by those IT managers who were implementing transaction-enabled applications. Because the EJB server automatically manages transaction behavior, the IT managers could have their programmers concentrate on writing the business logic of the application, rather than spending time writing the transactional code into the application.

Case Studies

Instinet Corporation

Instinet is the global brokerage firm that pioneered the brokerage business model that combines an electronic service with high-quality voice broker support. In early 1999, Instinet announced that it was bringing its agency brokerage model to Fixed Income.

Instinet’s key tenets are neutrality, anonymity, and transparency. Neutrality is not taking a proprietary position in any of the instruments traded, anonymity means that client identity is not revealed, and transparency means that participants see all orders posted and, when they trade, the net price of each transaction.

Today, the vast majority of Fixed Income brokerage business is conducted by voice using specialized telephony – turrets, dedicated lines, and so on. Consequently, brokers spend virtually all their time on the phone handling order flow directly. However, with the drive towards more efficient markets, margins are now becoming wafer thin. At the Bond Markets Association Advanced Technology Conference in New York last Fall, the question on attendees’ minds was not whether Fixed Income brokerage was going electronic but when: The consensus was that within a year we would see the emergence of several key participants in this arena.

As Duncan Johnston-Watt, Head of Development and Systems Architecture for Instinet Fixed Income observed, “It is a one way street. Once we have introduced our electronic brokerage service there can be no going back. Imagine, if you will, trying to persuade an air traffic controller that he or she can do the job without radar! As soon as you give bond traders a tool that gives them the ability to track and access liquidity more effectively, they will leverage this in a hundred different ways.”

Instinet chose to implement its electronic brokerage service using Persistence Software’s PowerTier for EJB transactional application server technology as a key component of its

N-Tier architecture. According to Johnston-Watt, Power Tier for EJB servers deliver zero-latency access to Instinet's data and provide robust transactional support. Using this technology, Instinet can guarantee that every client transaction will be permanently stored in a secure database as soon as it is submitted. When completed, this system may represent the largest distributed server application of EJB technology.

Rachel Moseley, Chief Information Officer of Instinet Fixed Income, mandated that no compromise on application robustness, performance or institutional security and data integrity could be tolerated. This led Johnston-Watt to tackle the problem of synchronizing distributed databases using a combination of Persistence Software's Power Tier products and TIB/Rendezvous message-oriented middleware from TIBCO, another subsidiary of Reuters Group PLC. In the process, his team has been instrumental in helping Persistence develop the EJB technology-based distributed cache synchronization technology announced on April 19, as the PowerSync feature of its new PowerTier 5.0 for EJB. Persistence was an early supporter of EJB entity beans, which provide for the container-managed persistence of data in EJB applications.

Overall, development time for Instinet would need to have been doubled if it had tried to implement the same system using just C++ and CORBA technology.

Security and Data Integrity

"One of the biggest obstacles to establishing our service is proving that it is secure and safeguards the integrity of our clients' data," said Johnston-Watt. "They need to be convinced that their trading activity is not being compromised in any way." Instinet has addressed this issue by presenting technical specifications to the client institutions one by one, taking them through the architecture step by step. Instinet's data never touches the Internet, but is instead carried over a private network that will eventually be capable of handling over 1000 transactions per second.

Both Entity beans and Session Beans

Johnston-Watt's team used both entity beans, the EJB components that hold the data in a persistent fashion, and session beans, the EJB technology that implement the business logic. "We picked Persistence in part because they were the first to offer support of entity beans," he said. "They were able to do this as their EJB application server technology is built upon a mature object-oriented caching product."

Another important consideration was being able to offer their clients a solution that met the 100% Pure Java™ certification standards. "We regard Java 2 technology as a real watershed," he said. "EJB technology and the whole Java Enterprise Computing model has come of age with this release. Whether our clients are accessing our service through our custom GUI or via one of our FIX (Financial Information eXchange) APIs the solution will meet the 100% Pure Java certification standards. We have a Java 2 desktop application and have just announced that we will be embedding Javelin's FIX engine Coppelia in Power Tier for EJB technology."

Business Reasons for Choosing EJB

Why did he choose to implement this application using EJB technology? In a phrase, Time to Market. "With the help of Java and EJB technologies we can achieve a strategic solution in a tactical timeframe," said Johnston-Watt. PowerTier for EJB technology comes with out-of-the-box support for Rational Software's Rational Rose UML-based object-oriented modeling tool, enabling process model-driven development. "We are able to generate both PowerTier Classic (C++) and PowerTier for EJB implementations of our object model. This radically reduces the risks associated with developing a multi-tier, multi-language application," he said.

Federal Aviation Authority (Seattle Air Space)

Who says Java applications are slow? Not Nick Xidis, telecommunications project manager for the FAA's Seattle Air Space facilities. For Xidis, the real secret to operational speed in a deployed application was the underlying object model upon which Java technology rides, and the ticket to quick development was that EJB system level services allowed him to separate business logic from implementation mechanism.

Xidis' team has completed the first of six telecommunications service order processing applications that will use EJB. For Xidis, the big movement was away from PL/SQL, Oracle's stored procedure language, and toward an object model. In this case, the EJB component model.

Enterprise JavaBeans entered Xidis' picture after the decision to go with an object-oriented application environment. He says that EJB technology provides an abstract browsing process, enabling the FAA to create a common business object layer. "We can then define the general notion of a unit of work, in our own terms," says Xidis.

"The telecom data set doesn't lend itself to relational databases and stored procedures," Xidis says. "There are a large number of relationships among the data, and there are multiple [database] joins involved in the data sets."

Xidis' legacy application includes 30 million record sets stored on Oracle 7 running on the Solaris™ operating environment. He used the Gemstone J/ Java server as the working platform to hold the records. Gemstone's internal object-oriented database maintained the relationships in the data in storing them as objects, and returning "orders of magnitude" better performance than trying to retrieve the same data through PL/SQL calls into Oracle.

The application Xidis' team created with EJB technology allows order submissions for telecom services as units of work defined by the beans. "Units of work ought to be snapped together, like Lego blocks, into larger blocks," said Xidis. "Each unit of work is different, and each has a different transactional content."

That general layer would be the most difficult to write, if EJB technology had not come along to fill in the gap. He says that implementing the same logic using PL/SQL would require a "whole set of gymnastics" that would be "incredibly difficult to understand" as

well as to write. "With EJB technology, we can define objects that are much more complex."

Xidis began his Telecom application project using Gemstone/J 1.1, a version that did not yet support EJB technology. "That meant that we had to write a bunch of logic that now is simply implemented as an EJB component," he said.

For Xidis, the platform-independent aspect of Enterprise JavaBeans architecture was not important, it was the ability to manipulate functionality as objects. The value of EJB containers is that Xidis does not have to write them. The container code is "half the code in the domain model", according to Xidis.

The applications have a potential client base of 55,000 FAA employees, but in the first stage, they are being addressed to a group of 2,500 client systems in the business service ordering and provisioning area. Two of the applications are completed, with the remainder to be completed within the year.

EJB Technology As the Best Object Model:

Xidis went to EJB technology because it was part of an overall strategy to go to an object model. The business benefit of highest priority was in completing the projects in less than a year.

With an object model such as Java and EJB, Xidis felt he could complete the project on time, and without it, he could get bogged down in a morass of stored procedure PL/SQL code.

Xidis evaluated Microsoft DCOM, OMG's CORBA, and Java technologies' RMI as alternative technologies. Accessing CORBA objects via RMI protocols defused the differences between Java technology and CORBA advocates, while the handling of threads steered Xidis away from Windows NT and COM/DCOM.

Scalability in N-Tier Applications:

By going to an object model such as that used in Java and EJB technologies, Xidis' project achieves N-tier application partitioning of the applications and their associated programming problems. This enables the FAA to grow middle tier business logic and middle tier server capability as its client base and application traffic increases.

Enhancing the Manageability of Application Code:

"It is an important skill to be able to break IS into manageable pieces," said Xidis. It allows him to divide the programming tasks along functional lines. He can, for example, let the supply and support area solve the business processes associated with inventory management.

"The establishment of boundary layers is the ticket," said Xidis. "The object oriented approach allows us to model the software along recognizable business process lines." The

alternative task, using PL./SQL stored procedure calls, would have been much harder to manage, he said.

Measuring the Time to Market:

The ultimate measure, for Xidis, is that the project applications be delivered on time, and for the money that the group said it would cost. So far with the use of the EJB architecture, that appears to be the case, he said.

Another measure of success is that the use of Java and EJB technology has enabled Xidis' team to more easily deal with the typical creeping increases in requirements and functionality by allowing them to divide the projects into smaller, more quickly-executed revisions. "This makes for short schedules and tiny teams. It means we can really stick to 12 month deliveries and 90-day enhancements," said Xidis. This secures the confidence of his upper managers.

SCT Corp.

SCT Corp. is a producer of ERP software, including supply chain management, planning and scheduling optimization. Its ADAGE, FYGIR Planning and FYGIR Scheduling software competes in the process supply chain management arena with ERP software from Oracle, SAP, PeopleSoft, Baan and JDEdwards.

SCT is currently using the Gemstone/J Web application server, embedded object-oriented database, and business object development environment. Bob Marsilio, general manager of component development for SCT, says the next generation of ADAGE and other products from SCTCorp. will be built using EJB technology.

EJB Technology Chosen Over CORBA for Transactions:

SCT Corp decided to build its future applications around EJB technology after first exploring CORBA. Marsilio wanted to move to an object oriented technology in order to begin to get some reusability of code among SCT's four business divisions.

Besides reusability of code, Marsilio wanted the functional partitioning that an object-oriented development environment could give him.

At the time Marsilio made the decision, (Jan. 1997) CORBA had not progressed as far as he needed. CORBA had naming services and events, but no transactional basis. (CORBA based Object Request Brokers have since implemented OTS, CORBA's Object Transaction Service).

"We decided to grab onto an object database and chose Gemstone," said Marsilio.

SCT developed its own transactions.

With EJB technology, Marsilio says, his team can develop the business process functionality without having to worry about how the distributed commits involved in the backend transactions actually work.

"Our management wanted business applications, and we were showing them how the distributed transactional commits would work. We needed to move it up to a higher level," said Marsilio.

The early stage of EJB 1.0 compliant Web application servers in early 1998 gave Marsilio pause. In such early stages, commitment to one vendor's products was indeed a commitment of his own project to the continuing growth of that vendor.

Even so, the adoption of one of the EJB application servers enabled Marsilio's group to establish one set of application programming interfaces toward which he could build applications.

EJB Components Are Re-Usable:

Right now, SCT Corp's applications are silos, each with its own monolithic structure. The business benefit that EJB technology will bring to this picture is reusable EJB components that can be used by several different SCT applications.

The business benefit is reduced administration effort involved in the migration of functionality, according to Marsilio. "The whole promise of an object-oriented world has been slow in coming, but I think its potential benefits are greater than we even know. I think the payback will be tremendous," he said.

It also allows him to build a layer of software that can be used as an interface for partnering products. "Componentization such as that afforded by EJB technology is an architectural revolution comparable to putting automobile production on an assembly line, or constructing housing out of standardized components," said Marsilio. "The potential impact is that large."

COM/DCOM Considered as Alternative:

Marsilio considered Microsoft's COM/DCOM object model as an alternative, but SCT's products must run on a large variety of platforms, including Windows NT, all of the major Unix variants, mainframes and AS/400 systems.

"EJB technologies hide the UNIX® differences between these systems. We will be doing some work with COM/DCOM, but in general we are looking for openness."

Timeframe for EJB Implementation:

By August 1999, Marsilio plans to have an EJB and Java technology-based Workflow System available in his company's applications.

The key to Marsilio's applications is the use of entity beans, which gives persistence to the session objects his team creates. These beans can be stored in Oracle 8i as abstract data type components.

EJB Technology Makes Security Integration Easier:

Marsilio also believes that EJB technology will make his team's life simpler when it comes to integrating security into their applications. "We are going to rely on EJB technology for that. We don't want to be in the security business, but we do want our own security." EJB components are already available to handle digital certificates and LDAP directory security.

Covad Communications

Covad needed a number of telecommunications operations support systems, including a customer care system, an order status system, a performance monitoring system, and an online backup system. Covad Development Manager Vinu Sunaresan decided to do them all using EJB technology, after first considering Microsoft's COM/DCOM model.

"We knew we needed a component model, but we were very nervous about the scalability of Windows NT. We chose to go with Oracle on the Solaris operating environment, using Java technology in order to function on a large scale," said Sunaresan. He uses WebLogic application servers, drawing the needed data from Oracle data stores on a Solaris-based system.

The applications on his list include a customer care system, order status, network performance monitoring, inventory, and service provisioning. Sunaresan needed these applications to communicate with clients using HTML, Java based applets, and XML. A Web application server using EJB technology seemed like a natural center for the operations. He chose EJB architecture to give himself the ability to model the process at a higher business process level. In the future, he hopes to be able to buy third party EJB components that function as truth tables, charts, state machines, and data collectors.

Rapid application development and quick time to market were important criteria, but perhaps even more so was the facility to change the application very quickly every three months or so. "We started off thinking about using the CORBA model, but we calculated that it would have cost us three to four times the go to market time to create them versus using EJB and Java technologies," said Sunaresan.

In this case, Sunaresan was able to save himself a good deal of time by buying third party beans and containers. These included Nightfire, Vitria's business process automation technology, Inprise's JBuilder library, and a set of plain JavaBeans from the KL Group.

Conclusion: Future Developments and Areas of Improvement

To look more than five years down the road, one needs either a curved telescope or a ouija board. Given that even horizon, it appears that with EJB technology and the Java technology movement we are looking at the most fundamental architectural change in software development for at least the next five years. The EJB deployment stage is beginning now because Java technology has had sufficient time to reach a second level of maturity, and because Web application servers that support the EJB 1.0 specification

began to become available last summer. As we head into the summer of 1999, we are seeing the first wave of serious, enterprise-wide, business-critical applications appear from the in-house efforts of major commercial institutions.

This development stands on the shoulders of some very important other developments, including most importantly the emergence of object-oriented programming techniques and the emergence of three-tier and N-tier application partition in the creation of distributed network applications.

To our knowledge, there is not a single piece of functionality that one could create with Java and EJB technologies that one could not create with C or C++, including transaction handling. The difference is that one can create the functionality considerably more quickly using Java and EJB technologies, and have far fewer worries about memory leakage and pointer allocation errors that make C/C++ programmers toil away their days. Moreover, because of the platform independence of Java and Enterprise JavaBeans technology, this code can be used in a multi-vendor, heterogeneous environment.

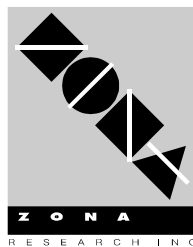
The IT managers whose experience is detailed in this report have articulated the benefits that they have seen and expect to see in the usage of EJB technology. These include, most importantly

- Reduction in the time that it takes to bring a new application to market, and at a lower cost
- Shortening of the software revision cycle so that smaller, more incremental changes can be brought to market more quickly
- Simplifying the tasks of creating and modeling business eliminating the need to implement backend functions like transaction monitoring, messaging, and security maintenance.

All these are on the very bright side, and, in our judgment, assure both Java and EJB technologies place in future distributed Web applications.

In the past, we have heard and written about many complaints about Java technology execution speed, and we have written about Sun's efforts to apply "hotspot" technology to speed up the execution of Java technology itself. We find that in the server domain, this does not seem to be an issue with IT managers. We believe this is because on the Internet server side of the application partition, it is network I/O latencies, rather than CPU processing functions, which assume the dominant role.

Moving forward, we see a geometric progression of EJB technology adoption as deployed applications move down the experience curve to the point where top management will feel comfortable committing more and more applications based upon success with current applications and further quantifications of the business benefits derived from their usage.



Zona Research, Inc.
A Subsidiary of IntelliQuest, Inc.
<http://zonaresearch.com>
Email: info@zonaresearch.com

Zona Research, Inc.
900 Veterans Blvd, Suite 500
Redwood City, CA 94063
V: 650.298.4000
F: 650.306.2420

IntelliQuest, Inc.
1250 Capital of Texas Highway South
Austin, TX 78746
V: 512.347.2932
F: 512-347-2909