

eRA J2EE Platform and Tool Recommendations

Version 0.1 (Complete Draft)

March 2, 2001

Contract No:
Task Order No:

Document ID:

Prepared For:

Office of Policy for Extramural Research Administration
Office of Extramural Research
National Institutes of Health
Bethesda, MD

Prepared By:

Logicon Federal Data/ROW Sciences and Silicon Spirit Consulting Group

Silicon Spirit Consulting Group, Inc.
1801 Robert Fulton Drive, Reston, VA 20191
www.siliconspirit.com Phone: 703-453-6849

Logicon Federal Data/R.O.W. Sciences
1700 Research Blvd, Suite 400, Rockville, Maryland 20850-3142.

Table of Contents

1	Background.....	3
2	Detailed Recommendation of Tools and Packages.....	4
2.1	Recommended Deployment (Physical) Architecture.....	4
2.2	User Desktop Machine Platform.....	5
2.3	Web Server Machine Platforms and Tools	5
2.3.1	Hardware/Operating System.....	5
2.3.2	Web Server Process.....	5
2.3.3	Java Virtual Machine (JVM).....	5
2.3.4	JSP/Servlet Engine.....	6
2.3.5	Web Security Packages.....	6
2.4	Application Server Machine Platforms and Tools	6
2.4.1	Hardware/Operating System.....	6
2.4.2	Java Virtual Machine(JVM).....	6
2.4.3	EJB Application Server.....	6
2.4.4	Application Security Package.....	9
2.4.5	Object To Relational (OR) Mapping.....	9
2.4.6	Business Rules Engine.....	10
2.4.7	XML Parser Package.....	10
2.5	Database Server Machine Platforms and Tools.....	10
2.5.1	Hardware/Operating System.....	10
2.5.2	RDBMS Vendor.....	10
2.6	Miscellaneous Tools.....	11
2.6.1	Integrated Development Environment (IDE).....	11
2.6.2	Methodology/Process.....	12
2.6.3	Object Oriented Analysis/Design/Architecture.....	12
3	Summary of Recommendations.....	13
4	Next Steps.....	15
	Appendix A—Acronyms/Abbreviations/Definitions.....	16

List of Figures

Figure 1.	eRA J2EE Deployment (Physical) Architecture.....	4
-----------	--	---

1 Background

This is the second of three planning documents relating to the application of Java 2, Enterprise Edition (J2EE) technology to the Electronic Research Administration (eRA) Commons system. The first document, "eRA J2EE Technology Overview" provides an overview of the recommended J2EE architecture for eRA Commons systems. It explains the concepts and technologies behind a standard J2EE-based implementation as well as how these will map to specifications associated with the eRA environment and requirements.

The purpose of this document is to provide specific product recommendations for the physical implementation of the architecture discussed in the "eRA J2EE Technology Overview." This document should only be reviewed in the context of the "eRA J2EE Technology Overview," as the concepts discussed here are fully explained in the overview document.

This document should be considered a living document that will remain a work in progress for some time. Due to the complex nature and interdependencies of products required (as discussed throughout the document), it is not possible to make one blanket set of recommendations. Instead, an iterative approach needs to be taken, where business, cost, and technical factors are considered for one product category at a time. In that light the initial version of this document provides strong recommendations in some areas, and a 'short list' of products for further consideration in other areas. A 'Next Steps' section at the end of this document summarizes action items remaining to finalize all recommendations. As these steps are executed and additional recommendations are finalized, new revisions of the document will be generated.

2 Detailed Recommendation of Tools and Packages.

2.1 Recommended Deployment (Physical) Architecture.

Figure 1 illustrates the preliminary deployment or physical architecture for the eRA J2EE Commons application. This architecture is based on a logical N-tier architecture described in the "eRA J2EE Technology Overview" document. Figure 1 maps the logical tiers discussed in the overview document to four physical machine platforms: user desktop, web server machine, application server machine, and database server machine. The recommendations of this document are discussed in the context of these platforms. Note that the details of this architecture are subject to change based on the application architecture effort to be conducted once the eRA J2EE Commons software development cycle officially begins.

The following sections describe the recommended hardware to be used, all tools/third party packages to be hosted in the hardware, and the communication mechanism to be used between the hardware components. Section 3 summarizes the recommendations in tabular form.

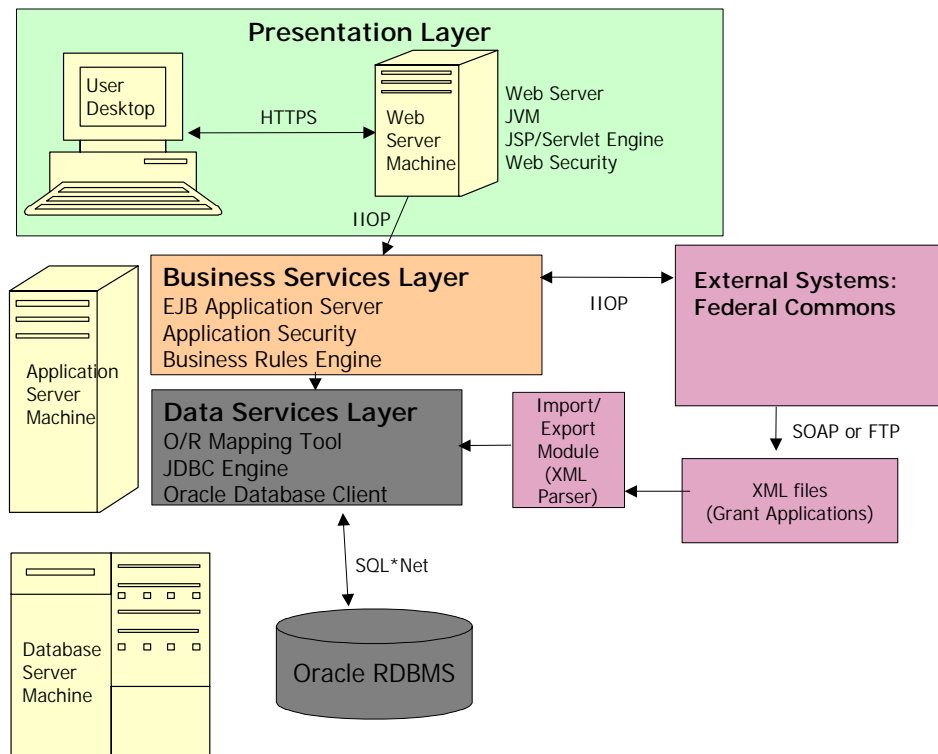


Figure 1. eRA J2EE Deployment (Physical) Architecture

2.2 User Desktop Machine Platform

The User Desktop is considered to be a part of the Presentation Layer. It will host the browser running under any hardware or operating system. The browser will accept html-based screens from the eRA J2EE Commons web server and display them to the user. Netscape Navigator and Microsoft Internet Explorer are the recommended platforms as these two products have the vast majority of the browser market.

2.3 Web Server Machine Platforms and Tools

The Web Server machine is considered part of the logical Presentation Layer, although physically the machine will be part of what is commonly known as the 'middle tier' as it will reside in a centralized location. This machine will have the following recommended software and communication components.

2.3.1 Hardware/Operating System.

Sun Solaris is the recommended hardware and operating system platform for the Web Server. This was the recommendation of the RNSolutions document, "eRA Commons Application Development Technology Analysis." That document provides sufficient justification for the selection, summarized as follows:

- Proven performance, scalability, and reliability as an internet platform.
- Leadership in Java implementation and support
- Successful track record as an internet platform for the eRA project
- General maturity, stability, and technical capability of Sun as a machine vendor

2.3.2 Web Server Process.

The Web Server runs as a process that fields incoming HTTP based requests coming from the user desktop and forwards them to the Java Virtual Machine (JVM). We recommend the Apache Web Server due to its wide deployment base (60% of all Web pages are delivered by Apache), attractive price tag (free), the right balance among performance, portability and extensibility, and a proven history of successful deployment with the product. While 'free' products can traditionally be lacking in support, the Apache Web Server is so popular and such a stable, proven commodity that we do not believe this will cause any problems. The Web Server can easily be swapped out in the future for a higher priced product if Apache proves lacking in some way. A detailed review of the Apache Webserver can be found at

<http://www.webtechniques.com/archives/2001/01/jagielski>.

2.3.3 Java Virtual Machine (JVM).

The JVM is the actual process/program that interprets the Java code containing the application logic. The JVM is contained in the Java Runtime Environment (JRE) package that can be downloaded free from the Sun Microsystems Java website

(www.java.sun.com).

2.3.4 JSP/Servlet Engine.

We recommend the Apache Tomcat product for the JavaServer Pages (JSP)/Servlet Engine due to its maturity (refinement of the original Servlet engine developed by Sun), its separation from the application server or database vendors (enabling vendor neutrality), the price (free), and proven history of successful deployment with the product. While 'free' products can traditionally be lacking in support, Tomcat is so popular and such a stable, proven commodity that we not believe this will cause any problems. The JSP/Servlet Engine can easily be swapped out in the future for a higher priced product if Tomcat proves lacking in some way.

2.3.5 Web Security Packages.

Multiple packages need to be acquired from third parties to perform web security functionality such as authentication, encryption, public key certification, etc. These categories are more in the realm of network architecture rather than application architecture and, therefore, will need to be recommended by the operations group.

2.4 Application Server Machine Platforms and Tools

The Application Server machine hosts software modules that are part of both the Business Services Layer and the Data Services Layer. This machine will have the following software and communication components.

2.4.1 Hardware/Operating System.

Sun Solaris is the recommended hardware and operating system platform for the Application Server. This was the recommendation of the RNSolutions document, eRA Commons Application Development Technology Analysis. This document provides sufficient justification of the selection, which is summarized as follows:

- Proven performance, scalability, and reliability as an internet platform.
- Leadership in Java implementation and support
- Successful track record as an internet platform for the eRA project

2.4.2 Java Virtual Machine(JVM).

The JVM is the actual process/program that interprets the Java code containing the application logic. The JVM is contained in the Java Runtime Environment (JRE) package that can be downloaded free from the Sun Microsystems Java website

(www.java.sun.com).

2.4.3 EJB Application Server.

The Enterprise JavaBeans (EJB) server runs within the context of a JVM and, via its containers, manages the processing of the business process and logic developed within the JavaBeans. There are 20+ vendors supplying the Application Server but the market is converging on three main players: BEA WebLogic, IBM WebSphere, and Sun Iplanet Server. Of these three, we recommend BEA WebLogic due to its maturity, wide deployment base (53% market share), endorsements by industry experts, and proven history of successful deployment with the product. The Turner Consulting Group document, "eRA Commons Application Server Analysis," also recommends BEA WebLogic as the application server.

The WebLogic Application Server product meets the needs of applications that manage transactions involving one database instance. Applications requiring two-phase commits between two database instances would require the Weblogic Enterprise Server product. Currently, we believe this will not be a requirement for the eRA Commons system. If this requirement does arise, the WebLogic Application Server can be upgraded to the Enterprise version.

While BEA WebLogic is the primary recommendation, serious consideration should be given to Oracle's Application Server (Oracle 9iAS)—based on the current relationship with and use of Oracle technology at NIH. Listed below are some high-level pros and cons of considering Oracle 9iAS vs. BEA WebLogic.

Pros:

- NIH already licenses Oracle 9iAS to support existing IMPAC II and Commons applications. There may be a possibility of reusing these existing licenses or receiving significant discounts on new licenses.
- NIH already licenses the Oracle Designer/Developer tools for use on the IMPAC II project. Oracle now offers a combined Internet Developer Suite that includes these tools as well as tools specifically supporting J2EE development. There would presumably be some cost savings inherent in simply upgrading the existing licenses to the Internet Developer Suite rather than purchasing a separate set of development tools from another vendor. For more on this topic, see the Interactive Development Environment section below.
- NIH and contractor staff already have knowledge concerning the configuration, maintenance, and use of Oracle 9iAS. If WebLogic is used, staff will need to obtain and maintain knowledge in both products, as there are no immediate plans to migrate the IMPAC II web-based Oracle Forms applications (that require Oracle 9iAS to execute) to the product selected for the Commons.
- NIH has an existing relationship with Oracle consulting to provide on-site consultants who can facilitate the use of Oracle technologies.
- As Oracle 9iAS is a relatively new product aggressively seeking additional market share, NIH as a major customer may be able to apply leverage to procure additional discounts on products or consulting services when using the product.

- Oracle promises a tightly integrated environment between the application and database servers, with certain advanced features such as automated web/database caching, SQL/J, and Java Stored Procedures that could simplify development and maintenance.

Cons:

- Oracle 9iAS is a new product. As such its stability and maturity cannot be expected to match BEA WebLogic. This could result in NIH serving as a 'semi-beta' site for Oracle as it has done in the past with the original Oracle Application Server product.
- Since it is a new product, it will not be possible to find staff resources (outside of Oracle consulting) who have experience implementing real life J2EE solutions using Oracle 9iAS. Conversely WebLogic has a comparatively longer history and larger user base from which staff resources (either through outside consulting or via internal hires) can be obtained.
- Since Oracle 9iAS does not yet have significant market share, integrated support for it from third party vendors (such as development tools and business rules engines) is currently limited when compared to WebLogic.
- Oracle is still primarily a database company. Although all signs indicate that Oracle will aggressively pursue making 9iAS a market leading product, it has a long way to go to reach the popularity and general acceptance of a product such as WebLogic.
- The tight integration and efficiency Oracle offers may require use of proprietary techniques that would tie the solution to the Oracle platform, and make it difficult to port to another vendor in the future. A good analogy here would be the current use of PL/SQL. Although PL/SQL is a powerful and easy to use language, by taking advantage of this proprietary feature the IMPAC II system has become dependent on the Oracle RDBMS, making ports to another RDBMS vendor very costly.
- Oracle's track record for technical support with NIH has not always been consistently good (especially with the application server product), however there are no guarantees that WebLogic or another vendor would provide better application server support.

Please note one item that is not listed as a pro or con is reuse of existing software. It is our current belief that Oracle 9iAS would not offer any advantage over another J2EE Application Server in this area. As discussed in the Technology Overview document, an EJB can reuse PL/SQL business logic through the use of Bean Managed Persistence (BMP). Any application server that complies with the J2EE specification would support this capability.

Before making a final decision on an application server product, we recommend executing two activities to gather additional information:

1. Coordinate a small meeting with Oracle to focus on the technical capabilities and direction of Oracle 9iAS as a J2EE Application Server as compared to WebLogic. A person familiar with the technical details of J2EE and WebLogic (such as Ani Dutta) should attend this meeting and question Oracle on these topics. This meeting should also be used as an opportunity to review the capabilities of the Oracle Internet Developer Suite as compared to third party Java design/development products.
2. Perform a cost analysis of the alternatives, considering all of the factors above.

2.4.4 Application Security Package.

The Application Server will need an additional third party tool to implement the infrastructure for configuring and controlling secure access to a J2EE application. This area needs further evaluation from an operations perspective, as the identification/authentication scheme used by the Commons should be consistent with membership directories or services provided by other NIH systems.

One product to consider in this evaluation is Netegrity SiteMinder due its compatibility with leading J2EE Application servers such as BEA Weblogic. Compatability with Oracle 9iAS will need to be further evaluated.

Netegrity provides an Application Server agent that will provide customers with the following capabilities:

- Single sign-on across J2EE applications as well as other applications in the enterprise.
- Authentication management — application server developers can now use SiteMinder to integrate public key infrastructure (PKI) technology, password management, and even biometric capabilities into their applications.
- Entitlement management — policies can be centrally created to manage user access to J2EE applications. SiteMinder will be able to provide very granular security for EJB, JSP, and Java Servlets.

2.4.5 Object To Relational (OR) Mapping.

Some method is required to simplify the mapping of EJB attributes to underlying Oracle database tables and columns. The actual techniques and tools used here will be dependent on both the Application Server and development tool selections. One popular product is the TopLink Foundation Library, which automatically generates database access code as Java objects that can easily be accessed from EJBs. TopLink provides customized integration for WebLogic, and as it is a WebGain product integrates well with WebGain Studio/Visual Café (see IDE section below).

How well TopLink would integrate with other Application Servers and tools requires further evaluation. Oracle offers a different approach to object relational mapping based

on Oracle Business Components for Java (BC4J), which comes packaged with the Oracle Internet Developer Suite.

2.4.6 Business Rules Engine.

Companies using traditional development methods must manually implement the many "business rules" that are the foundation of their applications. Once applications are created and deployed, changing these "business rules" requires manually modifying the code within the applications. To accommodate a relatively simple, but crucial shift in strategy, a business must often make hundreds of programming changes requiring many months of work by skilled programmers familiar with the company's applications. Business Rules Engine products enable non-developer types to specify business rules as they change during the life of the enterprise. Java code for the business rule will be generated automatically by the Business Rules Engine and can be invoked by the Beans at run time. This results in significant flexibility in addition to cost savings in development time.

The business rules engine category has been included in this document for completeness, but we do not feel it is necessary to purchase such a product at this time. There may be some functional areas in the Commons (such as the validation of a grant application) which would benefit from the use of such a tool; however, at this point we feel this to be of secondary importance and is something to be seriously considered only after the more critical product selections have been solidified. When the time comes, two leading vendors to consider in this space are ILOG and Versata (which is recommended in the Turner Consulting Group document).

2.4.7 XML Parser Package.

The XML parser package invoked by the J2EE application will parse and validate the XML data and, upon successful validation, load the data into the corresponding entity beans. At this point there are no urgent requirements for XML parsing by the eRA Commons, as the details and timeframes for Federal Commons XML requirements have not been solidified. However, when these requirements solidify we recommend evaluating the IBM XML4J 3_1_1 and the Sun Microsystem's JAXP 1.1 products.

2.5 Database Server Machine Platforms and Tools.

2.5.1 Hardware/Operating System.

Compaq (DEC) Tru64 is the recommended hardware and operating system platform for the Database Server. This platform has become the NIH standard for enterprise databases.

2.5.2 RDBMS Vendor

Oracle is the recommended RDBMS vendor for the Commons. As the current Commons and IMPAC II systems utilize an Oracle RDBMS, compatibility with these platforms is the primary reason why Oracle should be maintained as the sole RDBMS vendor. In

addition, Oracle is still certainly the leading RDBMS vendor and all current staff knowledge is in the Oracle RDBMS.

2.6 Miscellaneous Tools.

Several miscellaneous tools will be required to facilitate activities in different phases of the software cycle. Some tools are a must for productivity and some tools need to be further evaluated to see if they are worth the investment. In the following sections we will distinguish them where appropriate.

2.6.1 Integrated Development Environment (IDE).

Most high-end development tools today are advertised as an Integrated Development Environment (IDE), as they combine a visual code editor with an integrated set of add on components, compilers, and debuggers. Traditional IDEs have focused on the presentation layer, or user interface development. For Java IDEs this has involved a focus on Java applet and 'fat' client applications, with wizards and other code generation techniques serving to simplify the coding of complex GUI Java classes (such as AWT and Swing) supporting user interface widgets and events. Fortunately, with the growing popularity of J2EE as a standard architecture for multi-tiered development, most popular Java IDEs have now been expanded to 'suites' that include integrated capabilities for back end development, including JSP and EJB support.

There are currently five leading J2EE IDE suites available in the marketplace today: Borland JBuilder Enterprise, IBM WebSphere Studio, Sun Forte for Java, WebGain Studio, and Oracle Internet Developer. These are the five most popular Java IDEs in the Java Developer's Journal Reader's Choice Awards (see <http://www.sys-con.com/java/readerschoice2001/>) and are also the five Java IDEs covered in the December 8, 2000, PC Magazine review of "Web Development Tools" (<http://www.zdnet.com/pcmag/stories/reviews/0,6755,2660110,00.html>). Of these five, we recommend three for further consideration: Borland JBuilder Enterprise, WebGain Studio, and Oracle Internet Developer, as explained below.

Borland JBuilder Enterprise should be considered, as it is unquestionably the leading vote getter in Java Developer's Journal and the Editor's Choice in the PC Magazine review. Borland has historically been a developer favorite, and its vendor neutral approach would make it a good choice regardless of the Application Server selected.

WebGain Studio is a relatively new packaged suite based on the Visual Café product (previously owned by Symantec). WebGain has expanded the suite to include support for server side development and has custom integration capabilities with WebLogic. Historically, Visual Café has not been as popular as Borland JBuilder, but WebGain's additions have made this a promising product and the integration with WebLogic make this an attractive choice to work with that application server.

Oracle Internet Developer is Oracle's Java IDE offering. The core product, JDeveloper, was based on Borland's JBuilder, with a number of extensions added for middle tier

development and integration with Oracle 9iAS and 9i RDBMS features. If Oracle 9iAS is selected, this product would be a good option as it would provide the best integration with that application server. Oracle has also publicized future plans ('Project Cherokee' is the current code name) to further simplify/automate J2EE development; however, this product will not be available in production form in time for serious consideration for initial eRA J2EE efforts.

We do not believe that further consideration should be given to IBM WebSphere Studio or Sun Forte for Java. The IBM product would be a good selection if we were considering using WebSphere as the application server, but as this is not a consideration we do not see any advantage to the WebSphere Studio product (which includes VisualAge for Java) as compared to the other options. Sun Forte for Java, while a promising product, does not currently include any EJB support which rules it out as an option for Commons development.

Concerning next steps, we recommend finalizing an application server selection before an IDE selection is finalized based on the dependencies discussed above. Once the application server decision is made, further analysis can be performed on the IDE tools in this context.

2.6.2 Methodology/Process.

As will be documented in the "eRA J2EE Management Plan," we will be utilizing the Rational Unified Process (RUP) to provide best practices for Java development. To support this, we recommend licensing the RUP methodology, which is available as a product from Rational containing descriptions of tasks, roles, and templates for various work products (e.g., Scope Document, Use Cases, System Architecture). We will take the extensive RUP process and streamline it to map to our existing eRA methodology framework and fit the needs of eRA Commons development.

2.6.3 Object Oriented Analysis/Design/Architecture

Having a tool to facilitate the analysis and design phase is very essential for a rapid life cycle implementation effort. Specifically, such tools help document requirement artifacts such as use cases, analysis object models, and state models; design artifacts such as design object models and sequence diagrams; and architecture artifacts such as deployment diagrams. Rational Rose is the primary recommendation in this category, as it is the dominant product in this space and the top vote getter for the "Best Java Modeling Tool" in the Java Developer's Journal Reader's Choice Awards (<http://www2.sys-con.com/java/readerschoice2001/liveupdatemodeling2.cfm>).

However, WebGain Studio and the Oracle Internet Developer Suite come packaged with their own modeling tools, which should be considered if one of these is selected. Rational Rose is packaged with the Borland JBuilder Enterprise suite.

3 Summary of Recommendations.

This section summarizes the detailed recommendations in tabular form. Recommendations that are considered 'finalized' are bolded.

Table 1: Platform & Tools Recommendation Summary

User Desktop Machine Platform.

<i>Function</i>	<i>Description/ Comments</i>	<i>Primary Recommendation</i>				<i>Other Alternative/s</i>
		<i>Vendor</i>	<i>Package Name</i>	<i>Version</i>	<i>Cost</i>	
Web browser	Browser that displays HTML pages	Netscape, Microsoft	Navigator, Internet Explorer	Navigator 4.0+, Internet Explorer 3.0+	Free	None

Web Server Machine Platforms and Tools.

<i>Function</i>	<i>Description/ Comments</i>	<i>Primary Recommendation</i>				<i>Other Alternative/s</i>
		<i>Vendor</i>	<i>Package Name</i>	<i>Version</i>	<i>Cost</i>	
HW/OS	Machine hardware and operating system	Sun	Solaris	2.6	?	None
Web Server	Serves up HTML pages.	The Apache Software Foundation	Apache Web Server	Apache 1.3.12	Free	Iplanet Enterprise, Microsoft Internet Server
Web Container	JSP/ Servlet Engine	The Apache Software Foundation	Tomcat	3.1 (JSP 1.1, Servlet 2.2)	Free	Allaire, BEA
Application Security	Application Authentication, Password Management, integration with Bean Capabilities	Netegrity	Netminder	TBD	TBD	TBD

Application Server Machine Platform and Tools.

<i>Function</i>	<i>Description/ Comments</i>	<i>Primary Recommendation</i>				<i>Other Alternative/s</i>
		<i>Vendor</i>	<i>Package Name</i>	<i>Version</i>	<i>Cost</i>	
HW/OS	Machine hardware and operating system	Sun	Solaris	2.6	?	None
Application Server	Package that provides J2EE certified container services.	BEA	WebLogic	6.0	\$10K/CPU	Oracle 9iAS
Business Rules Engine	Package that provides non developers capability to specify business rules.	ILOG	JRule	3.1	TBD	Versata Logic Server
Validating XML Parser	Package that parses and validates XML files and converts into an object oriented representation.	IBM/Apache?	XML4J	3_1_1	Free?	Sun JAXP 1.1
Object-To-Relational Mapper	Dependent on application server vendor selected.	WebGain	TopLink Foundation Library Weblogic	TBD (should work with Weblogic 6.0)	TBD	Oracle BC4J

Database Server Machine Platform and Tools.

<i>Function</i>	<i>Description/ Comments</i>	<i>Primary Recommendation</i>				<i>Other Alternative/s</i>
		<i>Vendor</i>	<i>Package Name</i>	<i>Version</i>	<i>Cost</i>	
HW/OS	Machine hardware and operating system	Dec/Compaq	Tru64	?	?	None
RDBMS	Relational Database Management Server	Oracle	Database	9i	?	None

Miscellaneous Tools

<i>Function</i>	<i>Description/ Comments</i>	<i>Primary Recommendation</i>				<i>Other Alternative/s</i>
		<i>Vendor</i>	<i>Package Name</i>	<i>Version</i>	<i>Cost</i>	
Methodology/ Process	Contains task description & document templates	Rational	RUP	?	TBD	None
Analysis & Design Tool	Facilitates analysis, architecture, and design	Rational	ROSE	?	\$2,394/license	Oracle Internet Developer Suite
IDE	Facilitates end to end J2EE development	WebGain	Studio	4.1	\$8,995/license (includes TopLink)	Borland JBuilder, Oracle Internet Developer

4 Next Steps

The following is a summary of the next steps to be executed to finalize these recommendations, in general order of priority:

1. **Finalize Application Server selection.** As cited above, further discussions with Oracle and a cost analysis should be conducted to determine if BEA WebLogic or Oracle 9iAS should be used as the Application Server.
2. **Finalize IDE selection.** Based on the application server selected, further analysis should be performed between Borland JBuilder, WebGain Studio, and Oracle Internet Developer to finalize this selection.
3. **Finalize Analysis & Design Tool(s) selection.** Based on the application server and IDE selected, further analysis should be performed between Rational Rose and the design components of WebGain or Oracle Internet Developer to finalize this selection.
4. **Finalize Object to Relational Mapping approach.** Based on the application server and IDE selection, further analysis should be performed between TopLink and any built-in mapping capability of the IDE tool selected (if other than WebGain).
5. **Finalize Security Architecture.** Additional investigation on current NIH wide security mechanisms and products needs to be performed to finalize these selections.
6. **Evaluate Business Rules Engine.** As functional requirements are refined and the application server platform is finalized, further investigation into the applicability of a Business Rules Engine should be performed.
7. **Evaluate XML Parser.** As XML requirements are refined further, investigation into an XML parser should be performed.

Appendix A—Acronyms/Abbreviations/Definitions

API	Application Program Interface
Artifact	Work products of the methodology and process used for software development
BC4J	Oracle Business Components for Java
BMP	Bean Managed Persistence. Data transfer between an entity bean's variables and a resource manager managed by the entity bean.
Commons	NIH/eRA Commons System
Container	Same as an EJB Container
EJB™	Enterprise Java Beans. A component architecture for the development and deployment of object-oriented, distributed, enterprise-level applications. Applications written using the Enterprise JavaBeans architecture are scalable, transactional, and secure.
EJB Container	A container that implements the EJB component contract of the J2EE architecture. This contract specifies a runtime environment for enterprise beans that includes security, concurrency, life cycle management, transaction, deployment, naming, and other services. An EJB container is provided by an EJB or J2EE server.
eRA	Electronic Research Administration
HTML	Hypertext Markup Language. A markup language for hypertext documents on the Internet. HTML enables the embedding of images, sounds, video streams, form fields, references to other objects with URLs and basic text formatting.
IDE	Integrated Development Environment
IIOP	Internet Inter-ORB Protocol. A protocol used for communication between CORBA object request brokers.
J2EE™	Java 2, Enterprise Edition
JDBC™	An API for database-independent connectivity between the J2EE platform and a wide range of data sources
JRE	Java Runtime Environment
JSP	JavaServer Pages. An extensible Web technology that uses template data, custom elements, scripting languages, and server-side Java objects to return dynamic content to a client. Typically the template data is HTML or XML elements, and in many cases the client is a Web browser.
JVM	Java Virtual Machine
NIH	National Institutes of Health

OER	Office of Extramural Research
PKI	Public Key Infrastructure
RDBMS	Relational Database Management System
RMI	Remote Method Invocation. A technology that allows an object running in one Java virtual machine to invoke methods on an object running in a different Java virtual machine.
RUP	Rational Unified Process
Servlet	A Java program that extends the functionality of a Web server, generating dynamic content and interacting with Web clients using a request-response paradigm
SQL	Structured Query Language
SQL/J	A set of standards that includes specifications for embedding SQL statements in methods in the Java programming language and specifications for calling Java static methods as SQL stored procedures and user-defined functions. An SQL checker can detect errors in static SQL statements at program development time, rather than at execution time as with a JDBC driver.
TCG	Turner Consulting Group
UML	Unified Modeling Language
Web Component	A component that provides services in response to requests; either a servlet or a JSP page
Web Container	A container that implements the Web component contract of the J2EE architecture. This contract specifies a runtime environment for Web components that includes security, concurrency, life cycle management, transaction, deployment, and other services. A Web container provides the same services as a JSP container and a federated view of the J2EE platform APIs. A Web container is provided by a Web or J2EE server.
XML	eXtensible Markup Language. A markup language that allows you to define the tags (markup) needed to identify the data and text in XML documents. J2EE deployment descriptors are expressed in XML.