Leveraging Legacy Systems in a Network Computing Architecture

An Oracle White Paper

July 1997
OVERVIEW

The World Wide Web, Intranet, and Extranet are rapidly evolving to become the infrastructure for the global economy, offering low-cost implementations, ease of use and maintenance, and extensibility. Although organizations are eager to move in this direction, potential obstacles include:

- Limited resources for new development, training, and infrastructure
- Dependence on existing systems and technology
- Previous investment in existing systems, expertise, and data

By integrating legacy systems and client/server computing with Internet and distributed object architectures, Oracle’s Network Computing Architecture (NCA) allows you to leverage previous IS investments and combine the best new and existing technologies.

This paper will discuss several mechanisms available for integrating legacy systems in NCA:

- Direct access to legacy data from all tiers of the Network Computing Architecture through open ODBC, JDBC, and JSQL interfaces
- Complete, seamless integration of legacy data and applications through Oracle Open Gateways working with the Universal Data Server

Oracle’s Network Computing Architecture is an extensible distributed architecture that allows you to incorporate new object and Internet technology with existing legacy systems.
NETWORK COMPUTING ARCHITECTURE

Companies are rapidly embracing the Internet because of its low-cost, high-bandwidth networks, and simple easy-to-deploy Web technology. Internet use is quickly expanding from a platform for document browsing to use in core business applications.

Object technology promises to bring a new level of programming productivity to the IS world by allowing developers to package software capabilities into more manageable and useful pieces. With network facilities in the picture, objects become distributed objects that can operate across different operating systems, networks, languages, and hardware.

Despite the attraction of these new technologies, today most mission-critical systems run on client/server and mainframe systems because of the robust services they provide, including security, transaction, messaging, and data access.

Ideally, companies would be able to leverage the best of each environment -- to deploy robust Internet applications while protecting their significant investments in client/server technology and existing legacy systems.

Oracle’s Network Computing Architecture™ addresses these requirements, providing a unifying, standards-based architecture that encompasses client/server, Web, and distributed objects. This open, multi-tier architecture allows developers to mix and match the best new and existing technologies to create enterprise operational, decision support, Intranet, and electronic commerce solutions.

For the first time, developers will be able to create enterprise distributed and web applications that combine:

- The robustness of the mainframe and client/server worlds
- The ease of use, low cost of deployment, and universal access to applications and data provided through the Web
- The flexibility of object (component) technology

Key Components

The key architectural components of Network Computing Architecture are:

- “Pluggable” objects called cartridges that are manageable and provide extensible functionality at any tier
- Open protocols and standardized interfaces that enable communication among cartridges through a software bus called Inter-Cartridge Exchange (ICX)
- Extensible clients, application servers, and database servers
- Integrated development and management of cartridges
Fully extensible clients, application servers, and database servers make heterogeneous, cross-platform solutions possible. Cartridge technology provides extensibility at all tiers of Network Computing Architecture.

Any Client

Network Computing Architecture supports a broad range of clients including: personal computers, network computers, information kiosks, telephones, and mobile devices. Support of open, standards-based protocols allows developers to build cartridges using their choice of Java, JavaScript, C/C++, Visual Basic, and SQL-based languages while ensuring interoperability across the architecture.

Web Application Server

The Web Application Server plays a central role in applications implemented using the Network Computing Architecture. It acts as a platform for reusable business logic, taking responsibility for application code that might otherwise reside at the client or the database server. Network Computing Architecture clients thus become lightweight and manageable, and application developers may become more specialized: domain experts can code the business rules, while interface designers can create the client.

The Web Application Server is based on an open Object Request Broker (ORB) which provides scalable connectivity and services. Oracle's Web Application Server 3.0 is an application server that supports application cartridges for Java/HTML-based programs.

Web Application Server 3.1 introduces IIOP interoperability and connection services, based on an Object Request Broker and Scalable Cartridge Services. Some services are intrinsic to CORBA 2.0. Others, like the ODBC Application Server cartridge described later in this paper, provide services beyond that specified for CORBA.
Universal Data Server

Oracle Universal Data Server allows robust, scalable data storage and manipulation. In addition to supporting traditional relational data, Oracle Universal Data Server has been extended to manage new types of data, including video, audio, text, and spatial data -- providing powerful new ways to manage, manipulate, and deliver data in the networked economy.

Adding logic to the data stored in the database can provide significant performance and management advantages in many applications. With stored procedures, application developers can place database-intensive code -- such as computing aggregate statistics on sales data -- on the server to leverage powerful database features, such as parallel query execution. And with object-relational database technology, ISVs can extend the core capabilities of the database engine to create new data types with sophisticated functionality, such as pattern matching on image or audio data.

DIRECT ACCESS TO HETEROGENEOUS DATA

Today, many customers have data physically distributed across possibly incompatible networks, operating systems, and storage formats. Customers can directly access heterogeneous information from all tiers of the Network Computing Architecture through open ODBC, JDBC, and JSQL interfaces.

Data Access through Open Database Connectivity (ODBC)

Open Database Connectivity (ODBC) is an industry-standard interface for applications to access data in database management systems using Structured Query Language (SQL). Applications are highly portable because they are written for a vendor-independent API instead of a specific database. Applications call ODBC functions to submit SQL statements and retrieve results, and an ODBC driver manager loads the ODBC driver for that data store. The driver processes the ODBC function calls, submits SQL requests to a specific data source, and returns results to the application.

Any Client

Clients that support ODBC can directly access any datastore for which an ODBC driver exists. Developer2000, Power Objects, and many other Oracle tools provide ODBC support, allowing these clients to directly access legacy data from other vendors. Because each ODBC connection is limited to a single datastore, however, users accessing data through ODBC cannot perform distributed joins or distributed updates across multiple data stores, a key requirement when dealing with the Internet.

Web Application Server

Applications can also access datastores through ODBC at the Web Application Server tier. The Web Application Server includes an ODBC cartridge which allows clients to issue ODBC requests through HTTP statements. Clients can request data returned in an HTML table or as a series of strings. Users can also execute statements such as Data Definition Language (DDL) operations or Data Manipulation Language (DML) inserts and updates.

Universal Data Server

Finally, clients can access datastores through ODBC at the Universal Data Server tier. The Network Computing Architecture allows clients to connect to the Universal Data Server through numerous mechanisms. For example, existing client/server can run unchanged and communicate directly with the Universal Data Server via SQL*Net. Alternatively, “thin” clients can connect to the Universal Data Server directly through JDBC. Finally, a client can connect to the Universal Data Server through the Web Application Server using the ODBC cartridge or PL/SQL cartridge.
Any client that can connect to the Oracle Universal Data Server then has the ability to access virtually any datastore, using database services provided by the Oracle Open Gateway products. One Open Gateway product, the Transparent Gateway for ODBC, provides access to any data store for which an ODBC driver exists.

By connecting through the Oracle Universal Data Server and Transparent Gateway for ODBC, clients achieve connectivity superior to the direct access available through ODBC interfaces at the client and application server tiers. True integration is available because the gateways make heterogeneous data appear to reside within Oracle. Instead of using ODBC functions, applications issue feature-rich Oracle SQL against the ODBC data stores. The ODBC gateway works with Oracle7 to provide distributed joins and other advanced Oracle7 functions—even if the target data stores and ODBC drivers do not support them natively.

See the “Seamless Integration of Legacy Data and Applications” section of this paper for more information on Oracle Open Gateways.
Data Access from Java

Java has emerged as the de facto standard for distributed object programming within the Internet/Intranet arena, making Java ideally suited to the distributed object nature of the Network Computing Architecture. The portability that Java offers across tiers and platforms supports Oracle's commitment to open standards and makes it an excellent choice for cartridge development across all tiers.

Oracle provides two simple means by which Java programmers (writing code in any tier) may conveniently and efficiently access relational data: JDBC and JSQL. JDBC, a simple call-level interface which cannot exploit RDBMS schema information at compile time, is relatively low-level and can be tedious. JSQL proposes to complement the capabilities of JDBC by providing Java with compile-time access to the RDBMS schema, greatly simplifying code, providing type safety and higher performance. JSQL is specified as a straightforward, seamless integration of SQL and Java.

Because Oracle supports Java at all tiers of its Network Computing Architecture, JDBC and JSQL provide fast, direct access to SQL data from Java clients, applications, and database procedures. Like ODBC, however, the direct connection offered by JDBC and JSQL lacks the powerful integration functionality available for clients who connect to heterogeneous data through Oracle's gateways and the Universal Data Server.

Visit [http://www.oracle.com/nca/java_nca](http://www.oracle.com/nca/java_nca) for more information on Oracle’s support for Java.

JDBC

JDBC is a standard set of Java classes providing vendor-independent access to relational data. JDBC has been specified by JavaSoft. Modeled after ODBC, the JDBC classes provide standard features such as simultaneous connections to several databases, transaction management, simple queries, manipulation of pre-compiled statements with bind variables, calls to stored procedures, streaming access to long column data, access to the database dictionary, and descriptions of cursors. JavaSoft provides a reference implementation of JDBC which relies on an ODBC bridge. The JDBC calls are converted to ODBC calls which in turn call the vendor-specific API.

Being a vendor-independent library, JDBC only provides access to the standard SQL types and does not support the Oracle-specific types from Oracle7™, such as ROWID, or from Oracle 8™, such as different LOB types or user-defined object types and collections. Oracle will provide JDBC extensions to support all Oracle7 and Oracle 8 types, and extensions to improve data throughput. In addition, Oracle will also provide a thin JDBC driver (i.e., an all-Java implementation) suitable for downloading in an applet,
something that cannot be done with a thick, OCI-based driver. This thin driver will also support all Oracle7 and Oracle8 types.

**J/SQL**

J/SQL is a standards initiative developed through close partnership among Oracle, IBM and Tandem. This effort has also been endorsed by JavaSoft. A joint proposal is being made to appropriate ISO/ANSI standards bodies for adoption.

JSQL is an integration of SQL statements in Java programs. It is more concise than JDBC, and more amenable to static analysis and type checking. JSQL draws on Oracle’s experience with its precompilers for embedding SQL in many host languages (C, C++, Ada, FORTRAN, COBOL, Pascal), and in PL/SQL programs stored in the RDBMS. The JSQL preprocessor is itself a Java program.

The JSQL pre-compiler takes as input SQL statements embedded in Java and generates 100% Pure Java with equivalent calls to standard JDBC. The Java type system then assures that objects of those classes are called with correct numbers and types of arguments to pass values to and from SQL statements and Java programs. JSQL can be used to access data from any relational database for which a JDBC driver exists.

**Client**

Java clients can access heterogeneous database management systems directly using JDBC or J/SQL. Oracle is incorporating Java support into all its development and decision support tools, giving developers increased ability to build and deploy new applications and components in Java. Java functionality will be implemented in the following products: Oracle Developer/2000, Oracle Designer/2000, Oracle Power Objects, Oracle Discoverer, and Sedona.

In addition, Oracle provides tools to Web-enable existing client/server applications to be integrated into Network Computing Architecture. All applications built with current and prior versions of Developer/2000 can be deployed on the Web using Java. The Developer/2000 Web Cartridge performs the bulk of the processing on a Web Application Server and interacts with other Network Computing Architecture cartridges. All customers currently using Designer/2000 will thus be able to generate Java applications without writing any code or changing existing models or applications.

**Web Application Server**

Applications can also access datastores through JDBC or JSQL at the Web Application Server tier. The Web Application Server includes a Java cartridge which allows access to server-side Java programs. These Java applications can embed JDBC statements.
Universal Data Server

As mentioned, clients can connect to the Universal Data Server through various mechanisms. Any client that can connect to the Oracle Universal Data Server then has the ability to execute Oracle database procedures. Oracle will support Java stored procedures and database triggers. A specially tuned version of the JDBC driver will run directly inside the database for fast, direct access to relational data from Java stored procedures.

Users can connect to SQL data stores at the client, Web Application Server, or Universal Data Server tier of Oracle’s Network Computing Architecture. Each connection allows access to a single data store.
SEAMLESS INTEGRATION OF LEGACY DATA AND APPLICATIONS

Oracle Open Gateways is a family of products designed to integrate heterogeneous systems and includes the following categories of products:

Oracle Transparent Gateways give applications SQL access to legacy data stores. Operating at the Universal Data Server tier, these gateways exploit the facilities of the Oracle server to deliver greater integration than available through ODBC, JDBC, or JSQL access alone.

Oracle Procedural Gateways allow you to leverage non Oracle applications and data. Integrated with the Universal Data Server, this gateway allows Oracle applications to issue Remote Procedure Calls to access OLTP and messaging applications.

Oracle Access Managers allow you to leverage existing mainframe applications by providing them with SQL access to data in the Oracle Universal Data Server.

Oracle Replication Services allow custom, automated data replication between the Oracle Universal Data Server and any data store accessible through the Transparent Gateways, allowing disparate systems to coexist so you can maximize your investment.

Visit the Oracle Web site http://www.oracle.com/products/gateways or contact your Oracle representative for more information on these Open Gateway products.
Oracle Transparent Gateways

As mentioned, three-tier NCA clients and traditional two-tier clients can connect to the Universal Data Server through various mechanisms. Any client that can connect to the Oracle Universal Data Server can use the Oracle Transparent Gateways to access heterogeneous data. Oracle Transparent Gateways provide read/write access to virtually any data store. In addition to specialized gateways for over 30 relational and non-relational data stores, Oracle offers a Transparent Gateway for ODBC which enables access to any data store for which an ODBC driver can be written.

Transparent Integration

The data's physical location and storage characteristics are “transparent” to applications and end-users because the gateways create the illusion that all your data appears within a single, local relational database. This transparency allows you to integrate heterogeneous data seamlessly, and eases your application development and maintenance.
The Oracle Transparent Gateways provide transparency at every level:

- **Location Transparency**
  End-users can access tables by name, without having to know the physical location of any table.

- **Network Transparency**
  The gateways exploit Oracle's SQL*Net technology to allow users to access data across multiple networks, without concern for the network architecture or protocols. Multiple protocols are supported, including: TCP/IP, DECnet, SPX/IPX, and APPC/LU6.2.

- **Operating System Transparency**
  Users can access data stored under multiple operating systems without having to interact with the operating systems that host the data.

- **Data Storage Transparency**
  Data can be accessed regardless of its database or file format.

- **Access Method Transparency**
  Users can utilize a single dialect of SQL for any data store, eliminating the need to code for database-specific access methods or SQL implementations.

The gateways achieve this transparency without any manual dictionary mapping. Before an application can access any information, the application must be told the structure of that data, such as the columns of a table and their lengths. Most vendors’ middleware require administrators to manually define and maintain that information in a special data dictionary. None of this inefficient duplication is necessary with Oracle Transparent Gateways. Because the gateways use the existing, native dictionaries of each database, no redundant dictionary need ever be created or maintained.

**Extending Your Oracle Distributed Environment**

The gateways provide these unique advantages through tight integration with the Oracle Universal Data Server. In a Oracle distributed environment, multiple Oracle databases interoperate seamlessly, simulating a single-database environment through: two-phase commit transaction protection; maintenance-free access to remote data; and uniform support for feature-rich SQL functions, Oracle stored procedures, and Oracle database triggers. By exploiting the facilities of the Universal Data Server, the gateways extend these and other benefits to your non-Oracle data. Now all your data can participate in your Oracle distributed environment.
Below is an overview of some sophisticated Oracle Universal Data Server services available through the gateways.

- **SQL Functions**
  Your applications can access all your data using feature-rich Oracle SQL. Advanced Oracle server functions such as outer joins are available, even if the target data stores do not support them natively. The gateways' integration with the Oracle database ensures that the newest features of each database release are always immediately available to the gateway.

  Users can also access target data stores using the native SQL for those data stores.

- **Distributed Capabilities**
  Heterogeneous data can be integrated seamlessly, because Oracle distributed capabilities such as JOINS and UNIONS can be applied against non-Oracle data without any special programming or mapping.

- **Distributed Query Optimization**
  The Oracle database can utilize its advanced query optimization techniques to ensure that SQL statements are executed efficiently against any of your data. The data distribution and storage characteristics of local and remote data are equally considered.

- **Two-Phase Commit Protection**
  Oracle's two-phase commit mechanism provides consistency across data stores by ensuring that a transaction that spans data stores is still treated as a single-unit-of-work. Changes are not committed, or permanently stored, in any data store unless the changes can be committed in all the data stores affected.

- **Oracle Stored Procedures and Database Triggers**
  The same Oracle stored procedures and database triggers can be used to access all your data, ensuring uniform enforcement of your business rules across the enterprise.

- **Non-Oracle Stored Procedures**
  Through the gateway, Oracle users can execute stored procedures in the target data store as easily as if they were Oracle stored procedures.
Oracle Procedural Gateways

As mentioned, three-tier NCA clients and traditional two-tier clients can connect to the Universal Data Server through various mechanisms. Any client that can connect to the Oracle Universal Data Server can access non Oracle applications through the Oracle Procedural Gateways.

Transparent Integration

Oracle Procedural Gateways allow your Oracle applications to be integrated seamlessly with non-Oracle OLTP and messaging applications. The Oracle Procedural Gateways deliver this transparency by exploiting the Oracle Universal Data Server and its advanced stored procedure capabilities.

Your application simply issues a Remote Procedure Call to an Oracle PL/SQL stored procedure, and it accesses the non-Oracle application for you through the gateway. The procedure can execute a mainframe transaction directly or communicate with another application through a message queue.

Users only need to know the name of the stored procedure—not the physical location, transaction monitor or message queue, or programming logic of the remote application. Similarly, users do not need to be aware that the data accessed through these remote applications is physically located in another vendors’ database or file system, such as DB2, IMS, VSAM, CA-IDMS or ADABAS. Transactions or messaging operations invoked through the gateway are even automatically protected by Oracle’s standard two-phase commit mechanism.
You can leverage non-Oracle applications and data through Oracle Procedural Gateways without awareness of the transaction’s physical location, transaction monitor or message queue, programming logic, and data accessed.

Procedural Gateway for APPC
Using the Procedural Gateway for APPC, Oracle users can execute mainframe transactions through any transaction manager with built-in APPC support, including IBM CICS, IBM IMS/TM, or CA-IDMS/DC. A single gateway can access transactions on MVS, VM, AS/400, and any other system connected to your SNA network.

The Oracle Procedural Gateway for APPC communicates with the transactions using the industry-standard APPC and CPI-C call interfaces, not proprietary APIs. This enables any transaction to be used unchanged by both Oracle applications and any other CPI-C and APPC program.

You can leverage mainframe transactions to read and update databases and file systems, perform complex calculations, or launch processes involving peripheral devices. For example, a company can use the gateway to leverage existing, complex mainframe transactions for payroll, tax, or currency conversion calculations. When the calculation is complete, the transactions can pass the results to the Oracle calling application through the gateway, generate predefined reports, or print payment checks.

Procedural Gateway for MQSeries
The Oracle Procedural Gateway for IBM MQSeries allows Oracle applications to exchange messages with IBM MQSeries applications through queues, areas of storage where messages are held. By using queues, programs can communicate across a number of IBM and non-IBM platforms—without the complexities of direct communication across incompatible protocols, platforms, and networks. And because this communication is asynchronous, a message sent by one program does not interrupt the target program or rely on the target program to be available. The target can retrieve messages from the queue whenever it suits the business need.
Message flow can take many shapes, allowing one-to-one, one-to-many, or many-to-many program relationships. This flexible method of communication is ideal for a wide range of inter-related applications, including: e-mail, workflow, data replication and mobile applications.

For example, an agent using an Oracle client/server Travel Reservation application can request reservations for a customer's airline, automobile, and hotel arrangements. Using the Oracle Gateway, the application can place three messages in a queue—one for each type of reservation. Non-Oracle applications for Airline, Automobile, and Hotel Booking can take their relevant messages from the queue, process the reservations, and place confirmation messages in the queue. Upon receiving all three confirmations, the Oracle Travel Reservation application can print a complete, confirmed itinerary for the customer.
**Oracle Access Managers**

Oracle Access Managers allow you to exploit existing MVS and AS/400 applications by providing them with SQL access to data in the Oracle Universal Data Server. Used in combination with the Transparent Gateways, Access Managers allow your mainframe applications to access virtually any data store.

**Access Manager for AS/400**

The Access Manager for AS/400 allows AS/400 applications written in RPG, COBOL, and C to access data stored in the Oracle Universal Data Server. The same applications can access Oracle and DB2/400 data. Access to the non-DB2/400 data is completely transparent. Users can read and write information without even realizing that the data is stored in a different database, on a different machine. Ad-hoc SQL can also be issued as easily as if the data were stored in DB2/400.

**Access Manager for CICS and IMS/TM**

The Oracle Access Managers for CICS and IMS/TM are part of the Oracle for MVS Client Solution. The Oracle for MVS Client Solution allows you to access data in the Oracle Universal Data Server from TSO, Batch, CICS, and IMS/TM programs:

- Users can start a TSO session and have direct interactive access to any Oracle database server anywhere in your enterprise.

- *Batch* processes can update remote Oracle servers directly, eliminating the need to set up complicated file transfer mechanisms.

- *CICS and IMS/TM* transactions can update Oracle databases and other recoverable resources, and have the transaction protected with two-phase commit using SYNCPOINT processing.
Oracle Replication Services

Using Oracle Replication Services with the Transparent Gateways, customers can automatically maintain consistency between data stored in the Oracle Universal Data Server and virtually any other data store. This consistency gives you the freedom to move your data wherever it makes the most business sense, so you can leverage existing data and systems across the enterprise.

Oracle Replication Services provides automated, heterogeneous replication so you can maintain consistency between the Oracle Universal Data Server and other data sources. With this bidirectional replication capability, customers can automatically update Oracle tables with current data from Sybase, DB2/400, or any other data store accessible through the Oracle Transparent Gateways -- AND update those data stores with the latest changes in Oracle. For example, Replication Services can be used to replicate DB2/MVS data to an Oracle warehouse or Oracle departmental workgroups, copying either incremental changes or full refreshes of the tables or subsets of tables that you specify.
Automated Data Movement

Instead of manually modifying your applications to update multiple data stores, you just define your requirements with an easy-to-use, graphical administration utility. Oracle Replication Services will generate the necessary routines and perform the replication automatically. Information can flow seamlessly between the target and source, without requiring any changes to your applications. When the source system has a different table structure from the target, simply define with the click of a mouse how the data should be re-arranged when it is moved.
CONCLUSION

By combining legacy systems and client/server computing with Internet and distributed object architectures, Oracle's Network Computing Architecture allows developers to mix and match the best of new and existing technologies.

This flexible architecture provides direct access to legacy data from all tiers of the Network Computing Architecture through open ODBC, JDBC, and JSQL interfaces. Only accessing data through Oracle Open Gateways at the Universal Data Server tier, however, allows customers to seamlessly integrate legacy data and applications into their Oracle environments.

Allowing any client to connect to any data through a standard interface, the Oracle Universal Data Server and Oracle Open Gateways move information access one step closer to the new Information Age. Companies can make better business decisions through consolidated access to information, and share information effortlessly among employees, customers, and business partners. Step into the future by leveraging the past, with Oracle Open Gateways.