

Metaserver 4.1 Product Overview

White Paper

January 2003



Metaserver, Inc.
157 Church Street
New Haven, Connecticut 06510

Copyright

Copyright © 1997-2003 Metaserver, Inc. All rights reserved.

157 Church Street

New Haven, Connecticut 06510 USA

www.metaserver.com

This publication contains confidential, proprietary, and trade secret information. No part of this document may be copied, photocopied, reproduced, translated, or reduced to any machine-readable or electronic format without prior written permission from Metaserver. Information in this document is subject to change without notice.

Metaserver, the Metaserver logo, Metaserver Modeling Environment, Metaserver Process Integration Engine, and Metaserver Management Console are trademarks or registered trademarks in the United States and other countries. All other products or services mentioned in this document are identified by the trademarks, service marks, or product names as designated by the companies who market those products. Inquiries should be made directly to those companies.

Metaserver 4.1

Product Overview

Abstract This high-level paper is written for Metaserver clients, evaluators, and partners who want more functional or conceptual information about the product and its features, without all the technical details. It explains the major functional components of Metaserver 4.1, including the modeling environment, the engine, and the management console, and briefly describes some of the important concepts behind the architecture.

Introduction
Metaserver Modeling Environment
Pluggable Connectivity Framework
Metaserver Process Integration Engine
Metaserver Management Console
Supported Connectivity
Summary

Introduction

Innovative solutions often come from forward thinking companies with a history of thought leadership. Metaserver Inc. is a spin-off from network computing pioneer Scientific Computing Associates. SCA is renowned in the computing industry as the developer of Linda[®], the leading-edge parallel computing system. This technology was originally developed to create network-based supercomputers for computationally-intensive applications. As SCA began to focus on the growing Internet market, they realized that virtual shared memory and distributed parallel processing, already proven in the scientific community, could be used as the technological core of an innovative high-performance solution designed to bring business applications to the web.

Metaserver 4.1 is that high-performance solution. It provides a standards-based architecture that lets businesses create web applications faster and more cost-effectively than ever thought possible. Metaserver provides a rapid application modeling environment, powered by high-performance server software, that allows businesses to bring their existing and emerging applications to the web in an open, scalable, secure, and centrally-managed environment.

Metaserver lets you create *business processes* from your disconnected back-end systems. It runs them on an engine that's extremely fast, completely reliable, and can intelligently balance the processing load among all the distributed resources in your environment. The result is a web-enabled business application with faster throughput, higher availability, and better response time than any solution on the market. Metaserver supports a wide range of systems, applications, and databases, as well as all current and emerging standards. And it can help you build effective and successful web applications that are easy to change when you have to quickly adjust to new business requirements.

Business Process Integration

Metaserver brings Business Process Integration to the web. BPI is a strategy allows organizations to integrate entire business processes within their enterprise as well as with outside suppliers, partners, and customers. Metaserver lets you create a complete business process out of your current computing environment. It is highly scalable, which means it can accommodate any changes to your infrastructure as you update existing systems and add new technologies. Metaserver can support a continually growing base of systems and users as your enterprise expands. Its open architecture does not lock you into any single technology and it provides broad support for current and industry emerging standards.

This platform-independent technology lets you quickly create a web-enabled business process without the time-and-labor-intensive process of building a new web application from scratch. In Metaserver, business processes are completely independent of the platforms, technologies, and applications that participate in the process. And this makes them easier to implement, maintain, evolve, and scale.

Metaserver 4.1 Features

Metaserver 4.1 is designed to meet the technological challenges associated with moving business processes to the web. It provides an integrated environment for graphically modeling the flow of information in a business process to back-end systems. It provides a single graphical user interface that can be used by both business analysts and software developers. The business analyst can focus on defining how the process should work and the developer can focus on connecting the back-end systems. This intuitive modeling environment ultimately provides a unified solution that's web-ready, scalable, and works seamlessly with web servers like Apache, iPlanet, and IIS.

With Metaserver's easy-to-use modeling tools, management console, and high-performance server software, your organization can quickly leverage existing legacy applications to create a comprehensive web-enabled solution. And all this happens without massive rewriting, hiring new talent, or abandoning legacy applications.

Key Components

Metaserver consists of the following key components, each of which is described on the following pages. For more detailed technical information, see the online documentation, contact an account representative, or visit our website.

- **Metaserver Modeling Environment** – a graphical, integrated modeling environment. It comprises a set of visual tools that lets you rapidly create, assemble, and deploy business process models. These tools let you model both the execution sequence and the flow of information. The MME includes a set of sophisticated component wizards that create connectors to your existing back-end systems.
- **Metaserver Process Integration Engine** – a reliable, fault-tolerant server that hosts your business processes. The engine stores and runs the business processes created in the modeling environment. It uses parallel processing and provides built-in load balancing and failover. The MMC includes an admin console that manages the run-time environment. The MMC is a web-based tool that provides centralized management of all deployed business processes. It lets you monitor and control applications, via a standard browser, from any remote location.

- **Client SDK** – These are the Metaserver client APIs let you interact with the Metaserver engine from various client applications. These include pre-packaged applications such as Microsoft ASP pages, JavaServer Pages, and Macromedia ColdFusion Server, as well as custom applications written in Java, Visual Basic, and C++.

Metaserver Modeling Environment

In Metaserver, a business application consists of one or more process models. The MME is the visual modeling environment where you create, assemble, and deploy the individual processes that comprise a complete application. In the Metaserver Modeling Environment, you use a structured sequence of steps to:

1. create metalinks. (These are the thin adapters that link back-end business applications to the Metaserver engine.)
2. model your business process by assembling the metalinks into a process model.
3. deploy the process model to the Metaserver engine from where it can be invoked by a client, for example a web browser.

The MME is designed to let business analysts and developers model and develop discrete models that route data from disparate systems into a single, cohesive business application. In its simplest form, a Metaserver process model is a collection of metalinks and subprocesses that have been designed to work together. In the MME the process model, metalinks, and other components are organized into "projects" much like in other integrated development environments. This modeling environment lets you "diagram" the required connectivity logic in a business process without programming.

The MME has component wizards for connecting to back-end applications like EJB, COM, MQSeries, JDBC, etc. and it lets you connect to these applications without writing a single line of code. The MME lets business analysts assemble these individual components into a cohesive process and it has a built-in web server so you can test your finished application right from the modeling environment.

The MME also lets you deploy metalinks and process models to a central repository in the engine. This built-in persistence subsystem stores these elements in a central location from where they can be managed by an administrator and invoked by a client.

Creating a Business Process

The development effort for a business process consists of building and linking the series of processes (representing various business steps) that will execute at run-time. *In Metaserver, a process represents a single business process that consists of metalinks and subprocesses.* You use a variety of wizards to create the metalinks that provide connectivity to legacy applications and then you use the Process Designer to assemble them into a process model. Using an interactive approach you can move from design to development to deployment without ever leaving the modeling environment.

During the design phase, you use the Process Designer to conceptualize the flow of information and the execution sequence for each process. The Process Designer lets you drag in placeholders without having to actually define metalink interfaces. Typically the interfaces will closely match the interface requirements of the business application the metalink

connects to. After you create a metalink, you can simply drag it into the Process Designer, replacing the placeholder.

While working with your process models and metalink interfaces you can also start to prototype the end-user interface. In general, design and development is an iterative process between the client code that invokes the process, the interfaces, the metalink parameters, and ultimately the metalink's interaction with your business application.

The business analyst who designs the process models, the developer who creates the metalinks, and the client developers who create a user interface never need to code for fault tolerance or load balancing; these are built-in features of the Metaserver engine. Thus the development effort is entirely focused on solving business problems rather than managing the data or processes running on the engine. Metaserver simplifies the creation, testing, and deployment of an entire business process.

Metalinks, Process Models, and Wizards

Metalinks and process models are the building blocks of the modeling environment. A metalink is an adapter that lets the engine communicate with a back-end application. Multiple metalinks are wired together in the Process Designer to create a process model. A process model controls the flow of execution and the flow of data between the various steps (metalinks) in a business process.

You create metalinks using a variety of application-specific wizards; you create process models by diagrammatically connecting the metalinks together. This creates a single, cohesive *business process*, that you can deploy and run on the engine. In Metaserver, all of these processes are stored on the engine, and when you develop your own client front-end to interface with them, you are actually creating a complete, web-enabled business application.

Metalinks

Metalinks connect to databases, mainframes, applications, and other system resources. They are adapters containing the embedded lines of code, subroutines, etc. that link your business applications to the engine. metalinks provide the "glue" between the engine and the external applications you access via application-specific protocols. (This glue is automatically generated by the appropriate wizard and no custom coding is required.) By creating metalinks, and arranging them in the order they should execute to perform a process, you model a business process.

Metalinks provide access to the activities and data that support your business requirements. They can access databases (using JDBC and ODBC), handle messaging (MQSeries), or access legacy applications on mainframe and AS/400 platforms (5250/3270). metalinks can also access proprietary data and resources, and applications that use EJB, COM, CORBA, Java, JavaBeans, ActiveX, C and C++, Perl, and others. Metaserver also provides a library of pre-built metalinks for common functions like access control, FTP, and SMTP, and provides a framework for connecting to third-party adapters.

The MME has specialized wizards to help you create metalinks for different types of back-end systems. For example, you can automatically generate database query or stored-procedure metalinks using a query-generation wizard. These specialized metalinks automatically present data to back-end application in their native format. This means that in object-oriented environments, for example, the appropriate set of objects is automatically created; for shell

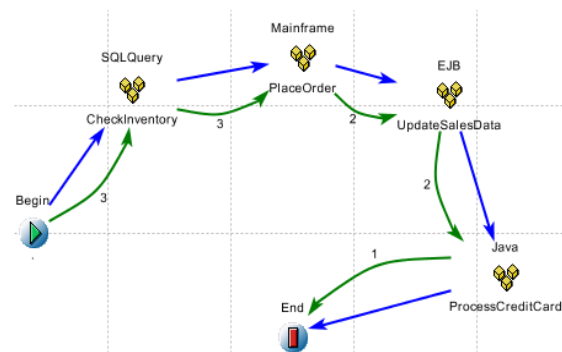
scripts or batch files, the appropriate environment variables or command line arguments are provided.

Process Models

The modeling environment provides a visual and intuitive mechanism for building process models. A process model is a graphical representation of a business process. It specifies the order in which the metalinks are executed and defines the inputs and outputs. For example, you might want to create an order entry process that (sequentially) queries a SQL database, checks inventory on a mainframe, processes a credit card on an application server, etc. This business process you model as a process in Metaserver is actually an event-driven, self-contained application that processes web requests from diverse applications and platforms. And since Metaserver components are re-usable, once you create a process model, you can drag it into other models you are building as a subprocess.

Process models are deployed to the engine's central repository from where they can be invoked via a client request. The process can run from a web application, an HTML or JSP page, or from a variety of other supported client types.

Creating Process Models. In its simplest form, a process model is a collection of metalinks and subprocesses. You create a process model by dragging one or more metalinks (which generally represent a step in a business process) into the Process Designer. Process models contain control flow instructions that specify which activities are processed and in what order. For example, a process might contain instructions to launch a SQL query, place an order against a mainframe database, and then perform a credit check. Each of these items is a metalink, and when you connect them together, you have created a process model that can be deployed to the engine.



A key feature to remember is that since Metaserver process models do not require coding, and are developed completely independently from back-end applications, they are easy to maintain and update when the back-end application changes.

Deploying Process Models. When you create a process model, it can be persisted and deployed to the engine simply by clicking a button in the modeling environment. This copies the necessary information to the central repository on the engine and makes the process available for execution by a client (for example, a web browser or Java application). During deployment, the MME automatically converts your graphical model to the industry-standard XML that the engine uses to interpret and execute processes.

Executing Process Models. Metaserver provides client plug-ins you use in your front-end application to execute processes stored on the engine. Client plug-ins are adapters that let specific types of applications interact with the Metaserver engine.

Client application types (Cold Fusion, Java, VB, JSP, etc.) have specific requirements. Each client plug-in adapts or translates a request from a client application into a call to the engine and translates the response into a format the client can understand. For example, if your client is a browser, you can use the MetaServlet plug-in to translate

HTML requests into engine requests. For example, the following URL will invoke the process named `OrderWidget` from the servlet engine on a web server:

`http://<webserver>/servlet/MetaServlet/?msprocess=OrderWidget`

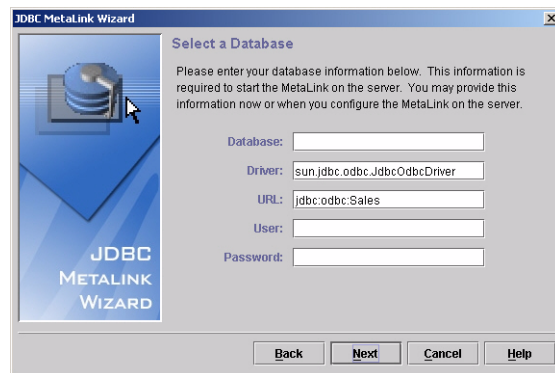
The Metaserver engine does the rest. It invokes the process and coordinates the processing of the widget order from start to finish.

Wizards

Much of the business process development effort is automated through the use of metalink wizards. A wizard is a sequence of dialog windows that automatically builds a metalink that encapsulates the business logic contained in a back-end process. For example, using a step-by-step wizard, you can build a metalink to validate a user or add registration information to a database. In most cases, the wizards eliminate nearly all of the custom coding needed to connect to a back-end application. When done, the metalink is automatically added to the Project Explorer and you can drag it into the Process Designer and use it to build a process model. There are wizards for custom components and for specialized components.

- **Custom component wizards** – provide full control over the logic used to access a back-end application. Metaserver has custom wizards to help build components for Java, Java/COM, Visual Basic, and C++.
- **Specialized component wizards** – support specific, pre-defined functions. Metaserver has specialized wizards for SQL, Stored Procedures, IBM MQSeries Send/Receive, and Jacada (3270/5250) components.

When creating a process model, you can drag re-usable, wizard-created metalinks into the Process Designer from the Project Explorer. For example, you can use the JDBC metalink wizard to construct a general-purpose SQL query. This metalink is automatically added to the Project Explorer and you can drag it into any process that needs to query a JDBC-compliant database. Metaserver also has wizards that connect to application servers like WebSphere and WebLogic. This makes it easy to integrate the business logic already available on your application server.



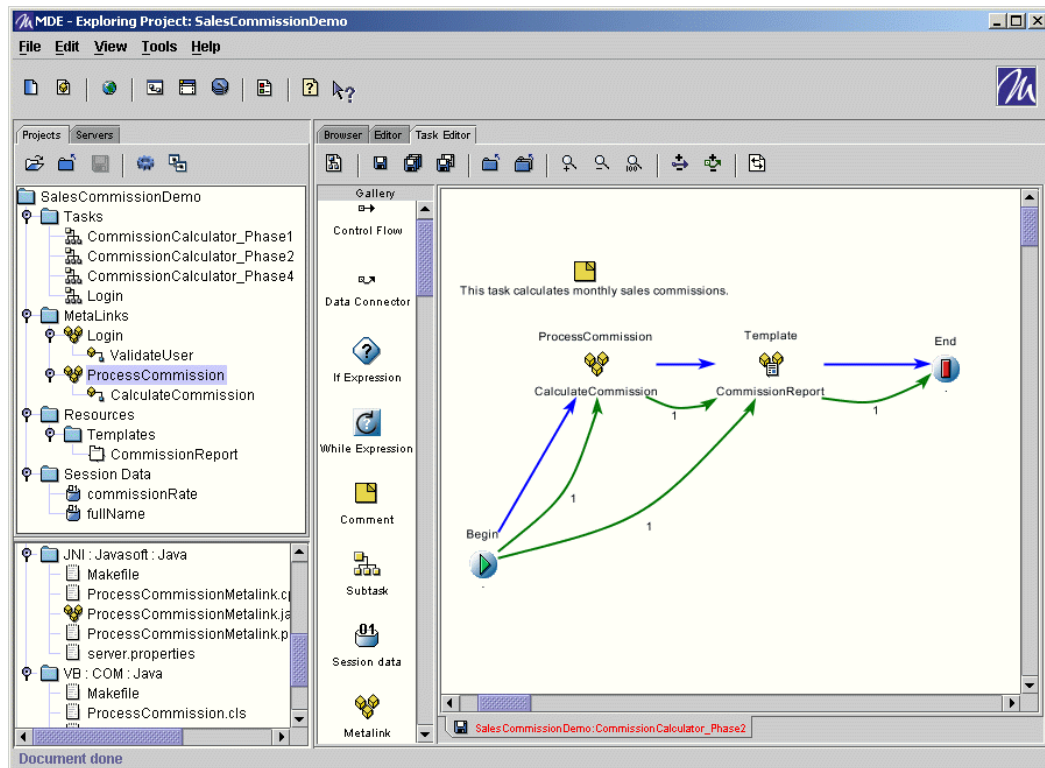
Metalinks are easy to create or change. To edit an existing metalink, simply re-open it in the modeling environment and edit the definition; you can change the initialization properties, methods, and any associated inputs or outputs.

Process Designer

The Process Designer is the workspace for the modeling environment. It is a visual work area into which you drag and drop components from the Project Explorer. (The Project Explorer is the resource window to the left of the Process Designer. It groups related items and their associated files into a tree structure and makes it easy to design and manage multiple projects concurrently.)

You create a process model by connecting the metalinks that have been dragged into the Process Designer. When you drop a metalink in the Process Designer, its visual appearance reflects its current characteristics. For example, if the metalink has three inputs and two outputs, its appearance visually reflects that signature. Similarly, if you mouseover one of the input/output stubs, a floating textbox displays its name, type, and any default values.

You connect metalinks with control flow connectors and data flow connectors. These connectors are visually represented by arrows from the output to the input. The straight control flow connectors describe the execution sequence, that is, the order in which the components will execute to perform a business process; the curved data connectors describe how the data outputs from one activity are mapped to the data inputs of another activity.



Also, since the activities in a process generally have different signatures, mapping dialogs are provided so that connected activities with different signatures can be easily mapped to each other. When done, the input and output stubs visually change to indicate that a mapping has been created between the elements. Once you create a process, you simply give it a name, and click Deploy. This copies the necessary information to a central repository where it is available to all potential users and client types connected to the engine. The processes are saved in persistent storage and is executed, for example, by simply accessing the appropriate URL on a web server from your browser. If you need to change a process model, you simply re-open it in the MME, make any changes, and then re-deploy.

Many enhancements to the modeling environment are currently in progress. A future release, for example, will let you graphically represent complex processes like the parallel execution of several components. In this scenario, if two system activities within a process can be processed in parallel, you simply draw that into the diagram and the processes will automatically run in parallel.

Pluggable Connectivity Framework

Metaserver provides a flexible, multi-layered connectivity framework with data mapping and transformation at each level. In a typical development environment, business analysts model their business processes by wiring together activities without regard for the underlying implementation. This means that: (1) each activity can have its own name and high-level data signature; (2) each of these activities can also have multiple implementations, e.g. for EJB and JDBC; and (3) each implementation can access the native business application which has its *own* data structures and constraints.

Metaserver enables connectivity across all of these application requirements by performing the data mapping and data transformation required at each level for connecting to databases, legacy systems, and packaged applications. The pluggable connectivity framework consists of a variety of wizards, tools, and language-level bindings for connecting to business systems on a wide range of platforms, applications, and technologies. For example, there are wizards for creating metalinks that connect to mainframe applications (via CICS, MQSeries, and 3270), Windows applications accessed through COM, EJB components accessed through Java, and many other systems. You can build metalinks in Java, Visual Basic, or C++.

Metaserver also provides connectivity to legacy mainframes and AS/400 systems. A variety of wizards, mapping dialogs, and importers provide easy, graphical connectivity (without coding) to CICS, 3270, 5250, and COBOL applications deployed on these platforms. Metaserver allows each system resource to run in its native environment, providing maximum flexibility, leveraging existing investments in applications, and reusing the business logic.

Metaserver Process Integration Engine

The Metaserver engine is a reliable, scalable, fault-tolerant server that hosts the business processes created with Metaserver. It runs in a distributed environment and is designed from the ground up for lightning-fast, parallel processing. With parallel processing, Metaserver can intelligently distribute work and take maximum advantage of the widely dispersed resources typically found in a distributed computing environment.

For example, if an online order for books requires a credit check and an inventory check, both requests can be executed at the same time (in parallel) on different machines. You can configure your server environment in a variety of ways depending on your own unique requirements. It generally consists of one primary server, additional processing servers, and a hot spare to back up the primary server and provide built-in high availability.

The Process Integration Engine manages the parallel processing. The core technology is based on Linda, the industry-leading platform for distributed, parallel processing. Linda provides the processing functions that manage data transfers between running business processes. Linda is the basic communication mechanism and the key enabler for load balancing and scalability. The engine is easy to scale because it uses memory capacity, not CPU cycles, to execute processes. (If you need to add power, you don't start building a server farm, you just add memory.) Linda also provides state-smart application coordination by swapping data in and out of virtual shared memory. Virtual shared memory is the key to automated scheduling, process synchronization, and optimum resource utilization (load balancing).

The power of the engine derives from the fact that in a Metaserver environment, the business process does not run or process requests on a web server. Instead, all steps in the business

process are executed by the various distributed resources themselves. The applications are running in their own native environments on an independent network of linked computers. The Metaserver engine *coordinates* the processing; it does not actually run the processes.

The engine is specifically designed to quickly and reliably execute a process that requires extracting data from multiple, incompatible systems. Metaserver executes all processes in parallel: the instructions are performed simultaneously, on different machines, to optimize performance and throughput. Metaserver also balances the computing load over multiple physical machines so that requests for service are automatically routed to lightly-loaded servers.

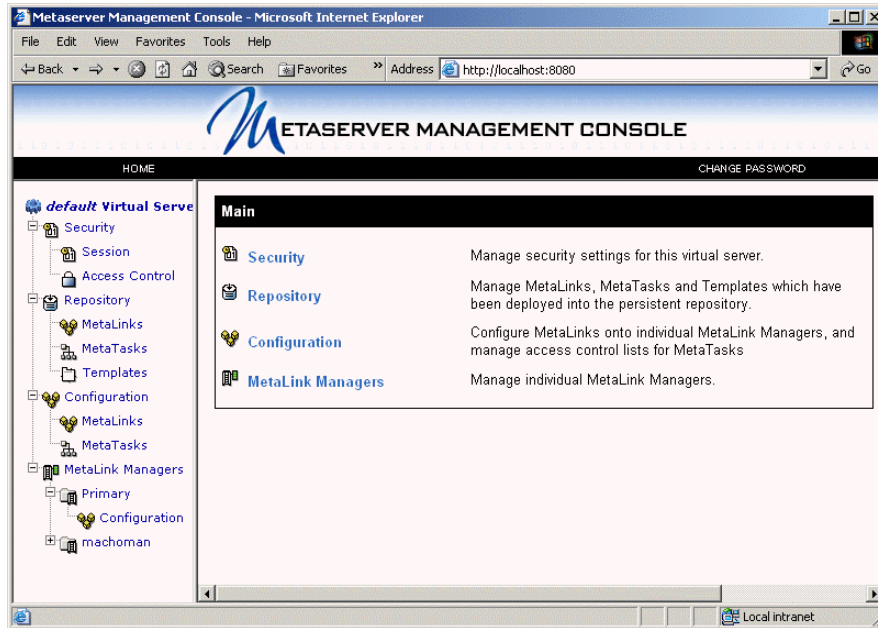
The Process Integration Engine has true plug-and-play: you can add or remove hardware, or reconfigure metalinks, without restarting the engine. For example, you can dynamically reconfigure the system and allocate resources to optimize performance and increase throughput as needed, on demand. You can connect additional servers, on-the-fly, to shorten the response time for a popular application or system resource. Or you can swap back-end systems in or out without affecting the availability or performance of a running business process.

Similarly, since parts of a process may be executing in parallel on different machines, if one system is removed, another can be dynamically re-configured to replace it with minimal impact and no loss of service.

Metaserver Management Console

The Metaserver Management Console (MMC) is the graphical interface that lets you access and manage processes configured to run on the engine. You run the MMC in a standard browser either locally or from any remote location. Perhaps its single most important feature is that it lets you configure metalinks to run across multiple machines in a truly distributed environment. This increases throughput, eliminates bottlenecks, and gives you complete control over how your system is configured to run.

The MMC is a browser-based application written completely in Java. It's built entirely with Metaserver 4.1 technology and lets you monitor and control the engine from a centralized admin console. You can start, stop, and configure metalinks, as well as set and modify security policies. With the MMC, if you need to start additional instances of a metalink to handle heavier than normal traffic, you simply specify how many metalinks you need, and click Start.



The MMC also lets you create and manage "virtual servers" that segment, or partition, the engine. A virtual server is a way to create a unique run-time environment for a specific set of metalinks and processes. This lets each process execute in a dedicated environment and gives you additional options for backup, failover, and load balancing.

The management console is the centralized admin tool for the entire system. It is automatically installed when you install the Process Integration Engine, and is also installed on a secondary server if you configure a hot spare. The MMC is also used to manage the central repository of metalinks and processes. *Once created and deployed, these re-usable components are available to any developer who can access the engine.*

Integrated MMC Features

The Metaserver engine incorporates the built-in advantages of parallel processing and provides real-time execution of business processes in a server environment. This environment also provides for centralized control of key housekeeping features like user authentication, access control, and security, directly from the management console. This makes the MMC a powerful, versatile tool for managing your enterprise.

Authentication

User authentication is provided by a built-in authenticator component. Metaserver includes an LDAP-based authenticator that performs user authentication by binding to an LDAP server using a name and password pair. The default authenticator can be replaced with any custom component that authenticates using a different mechanism as long as it conforms to the signature of the built-in authenticator. This lets you deploy custom authentication schemes but keeps the rest of the Metaserver access control system intact.

Access Control

The management console also lets you build access control lists (ACLs) for process models. For example, you can create specific permissions for processes that may restrict the activities each user, or group of users, can perform. When the engine receives a request to execute a specific process, it checks the ACL information first. If there is no ACL, it uses a predefined system policy that either always allows or always denies access to processes with no defined ACL. If an ACL object is found, the engine checks for authentication to determine whether the user or group making the request has the appropriate permissions. If not, access is blocked and an exception is sent to the requestor.

Load Balancing

The Metaserver engine is powerful, reliable, and effective. It uses virtual shared memory to completely automate load balancing across multiple back-end servers so there is never a single point of failure. You can also use the MMC to configure multiple instances of a metalink on different machines. This means that if one instance of a metalink fails in mid-stream, the engine automatically routes the processing request to another instance of the same metalink running on a different machine. Also, in a fault-tolerant configuration, the engine starts on a primary server and has a hot spare backup running on a different machine. If the primary server fails for any reason, all processing automatically fails over to the backup. No information is lost and the entire failover process is transparent to end-users.

Security

Metaserver provides inbound and outbound encryption via the Secure Sockets Layer protocol. (SSL is the industry standard for secure Internet communications.) Metaserver also supports custom encryption and you can use any pluggable encryption framework. Metaserver also supports digital certificates and all inbound and outbound requests can be digitally signed to guarantee authenticity.

Supported Connectivity

Metaserver is platform-independent and connects to a wide variety of current and emerging technologies. Connectivity is supported, but not limited, to all of the technologies shown below. Contact a Metaserver account representative if you have specific questions about your own environment or technology.

Servers

- Integrates with any LDAP server
 - Integrates with EJB servers, e.g. BEA WebLogic, IBM WebSphere, etc.
 - Compatible with ColdFusion Server 4.1 via the ColdFusion Custom Java API
 - Integrates with all standard web servers including Apache, iPlanet, and IIS
-

Operating Systems

AIX, Linux, Solaris, Windows NT, Windows 2000, and any systems with a JVM 3.1 or higher installed

Client Support

HTML clients running in browsers, XML/XSL clients, Java Server Pages, Active Server Pages, WAP-enabled phones, PDAs, fat clients written in Visual Basic or Java, and Microsoft Excel macro clients

Databases

- Sybase, Oracle, and Microsoft SQL Server using native drivers
 - IBM DB2, Informix, and any JDBC-compliant database
-

Legacy Systems

- IBM-compatible using Jacada to access data and transactions
 - Metaserver components can access 3270 and 5250 applications running on S/390 and AS/400 systems
 - Compatible with IBM MQSeries via the Sun Java Messaging Service (JMS) API
-

Components

You can mix and match the following component that access data using the metalink API: NT Authenticator, LDAP Authenticator, XML/XSL, SMTP, FTP, ICVerify (CCP)

Connectors

SAP, PeopleSoft, Oracle Financials, Siebel, ISO, Sungard, Acord, i2

Standards Compliance

Java 2 (EJB, JCE, JMS), XML/XSL, COM, CORBA, SMTP, SSL and digital signatures

Languages

Java, Visual Basic, C++

Summary

Metaserver lets organizations use the technologies and applications they currently have in place to rapidly build the new web-enabled applications they actually need. It completely automates critical requirements like load balancing and failover and provides an agile and scalable framework for growth. It supports a wide range of industry standards like COM, EJB, LDAP, and HTML and can easily leverage off-the-shelf application servers, directory servers, web servers, etc. that implement those standards. And perhaps most importantly, Metaserver lets organizations use the existing skills of their people most effectively.

Metaserver means business. It has a strong set of features that meet clear business objectives. It's an integrated modeling environment that lets you define business processes graphically. And it executes those processes reliably and securely over widely distributed resources. Metaserver is a rapid process modeling environment and it can bring your business to the web faster than you ever thought possible.