IONA®

Orbix® 6.2

Technical Overview

IONA Technologies PLC

December 2004

Table of Contents

1 Introduction

CORBA makes software work together regardless of where it is located, what it is running on, or what language it is written in. Today, CORBA is mainstream technology even in traditionally conservative IT environments, including banking, insurance, utilities, manufacturing, and government.

IONA Technologies built its business around CORBA. Orbix is the market-leading implementation of the CORBA standard and has been in existence almost as long as CORBA. It enables applications to interoperate across network, language, CPU, and operating system boundaries. Beyond the implementation of the standard, it also provides enterprise-class features that reside at the core of thousands of distributed systems around the world. Orbix routinely handles integration and scalability problems in the most complex and largest systems.

Between the extremes of everyday commercial data processing and hard realtime control applications, Orbix operates across the entire spectrum of computing tasks, from billing systems and multimedia news delivery to airport runway illumination and aircraft radio control. Most of the world's telephone systems, as well as the truly mission-critical systems operated by the world's biggest banks, are built on Orbix.

Orbix has always been attractive to IT organizations in industries that need large, resilient systems to handle enormous peak volumes of data and service requests, while guaranteeing a high level of availability. Financial services companies constitute the most transaction-intensive industry in the world. For this reason, nearly 70% of the financial organizations listed in the Fortune Magazine Global 100 rely on Orbix.

Orbix has also made impressive inroads in aerospace, government, high-tech manufacturing, and several other industries. More developers today have used Orbix than any other ORB. This creates a larger pool of experience on which development organizations can draw, and lowers recruiting and training costs.

IONA's dominance of the CORBA market positions us to provide the best and broadest support for CORBA implementations. IONA now offers five-year support up front. This means that your long-term deployment plans have the backing of IONA's support services. IONA guarantees binary compatibility for future versions of Orbix 6, allowing an easy upgrade.

IONA supports CORBA on more hardware and operating system platforms than any other CORBA vendor. IONA's continuing support for the latest operating systems and compilers allows you to take advantage of the latest performance improvements in hardware.

2 Architecture

Orbix includes an Object Request Broker (ORB) that presents an abstraction layer that relieves the programmer of dealing with many of the complexities of network programming, such as thread pool control, request dispatch, connection management, and so on. The ORB creates the illusion that all objects are local objects and accessible by way of an application's native programming language.

Orbix allows systems to communicate directly across a network, regardless of the programming languages used to create them, and regardless of the operating systems and platforms on which they run. Orbix is used to integrate applications written in languages such as Java, C++, C, Visual Basic, and COBOL, and runs on PCs, UNIX hosts, and mainframes.

Orbix is built on IONA's Adaptive Runtime Technology (ART). ART is a plug-in framework that delivers scalability, high performance, and flexibility to distributed enterprise applications, as illustrated in *Figure 1*.

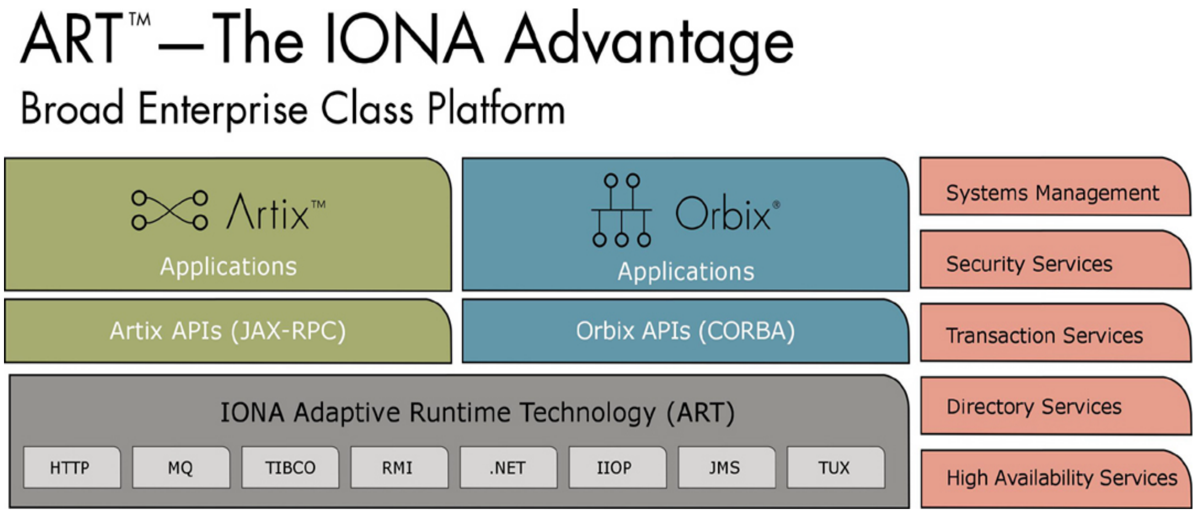


Figure 1: ART Plug-In Architecture

Enterprise qualities of service are provided by way of a systems management, directory service, transaction service, security service and high availability services, all of which are built as ART plug-ins. Plug-ins can be linked directly with an application, loaded when an application starts up, or loaded on demand while the application is running.

Orbix has been designed from the ground up to support enterprise-class distributed systems. Orbix provides major advances in scalability, performance, and deployment flexibility, thanks in part to its plug-in framework, which delivers greater flexibility and scalability to distributed applications than has ever before been possible.

IONA also provides Enterprise Middleware Integration, Web services integration, J2EE integration and .NET integration through its Artix product range, which is based on the same ART plug-in framework as Orbix.

3 Features and Benefits

Orbix is available in Standard and Enterprise editions. This section outlines the features and benefits that are new in Orbix 6.2 (see *Table 1*). It also provides a more detailed description of these features and all other features available in Orbix Standard (see *subsection 3.2*) and in Orbix Enterprise (see both *subsection 3.2* and *3.3*).

3.1 What's New in Orbix 6.2

Features that are new in Orbix 6.2 are outlined in *Table 1*:

Category	***Feature***
Configuration	▪ Supports extensible configuration. Everything does not need to be configured up front when a system is first deployed. Domain functionality can be extended at a later stage by deploying, for example, a trading service, or adding or removing service replicas.
High Availability	▪ Slaves dynamically promoted to master. ▪ Replication through Berkeley DB ▪ Performance enhancements.
Security	Security service clustering: ▪ Multiple security servers remove single point of failure through the automatic failover to backup servers ▪ Load balancing—Improved scalability by spreading the load among a number of clustered security servers ▪ Federation of IONA security servers—Allows single sign-on across different security domains.
	Enhanced ACL: ▪ New option for centralized ACL policy definition. ▪ Orbix Security Service can act as a security authorization policy server. ▪ Existing support for local ACL maintained. ▪ Support for allowing the authorization engine to be replaced with a custom ACL implementation.

	General security: ▪ Automatic warnings on use of SSL certificates near their expiration date or that match user specified criteria. ▪ Support for external bridging to a CSIv2 or Orbix security domain from a non-CORBA technology domain, such as an Artix environment.
Management	▪ Integration with Enterprise Management Systems: BMC Patrol, HP OpenView, and IBM Tivoli. ▪ Client-side performance logging, which gives metrics on server availability and response time. ▪ Orbix work queue instrumented and available in a managed entity (an `MBean`). ▪ `MBean` monitoring plugin that gathers statistics for the log file on whatever is instrumented.
General	▪ Plug-in that compresses data before it is put on the wire, improving throughput and decreasing latency. ▪ High availability, JMS, JMS Notification bridge and the firewall proxy service on IRIX.
Platforms	▪ AIX 5.2 and Visual Age 6.0. ▪ Windows 2003, Visual C++ 6.0 and Visual Studio .NET 2003 7.1. ▪ Full enterprise support for Red Hat Linux Advanced Server (AS) 3.0. ▪ 64-bit JVM support on Sun Solaris 9. ▪ JDK 1.4.2. ▪ Sun Studio 8. ▪ HP-UX 11i on Itanium

Table 1: New Features in Orbix 6.2

3.2 Orbix Standard

Orbix Standard includes the following features, each of which is described in the subsections that follow:

- Portable Object Adapter
- Extensible Configuration
- Secure Sockets Layer/Transport Level Security (SSL/TLS)
- Smart Card Support
- Event Service
- Interoperable Naming Service and Load Balancing Extensions
- Bidirectional GIOP
- Compression Plug-in
- Asynchronous Messaging Interfaces
- CORBA Reflection and Dynamic Type Support
- Configuration and Logging Interfaces
- Persistent State Service
- Code Generation Toolkit

3.2.1 Portable Object Adapter

Orbix servers use the Portable Object Adapter (POA). The POA is the CORBA-standard way to write portable server code. The POA provides a flexible framework for mapping abstract CORBA objects to concrete programming language objects. You can select from a variety of POA policies that control the memory/speed trade-offs in highly scalable servers. For example, using POA policies you can:

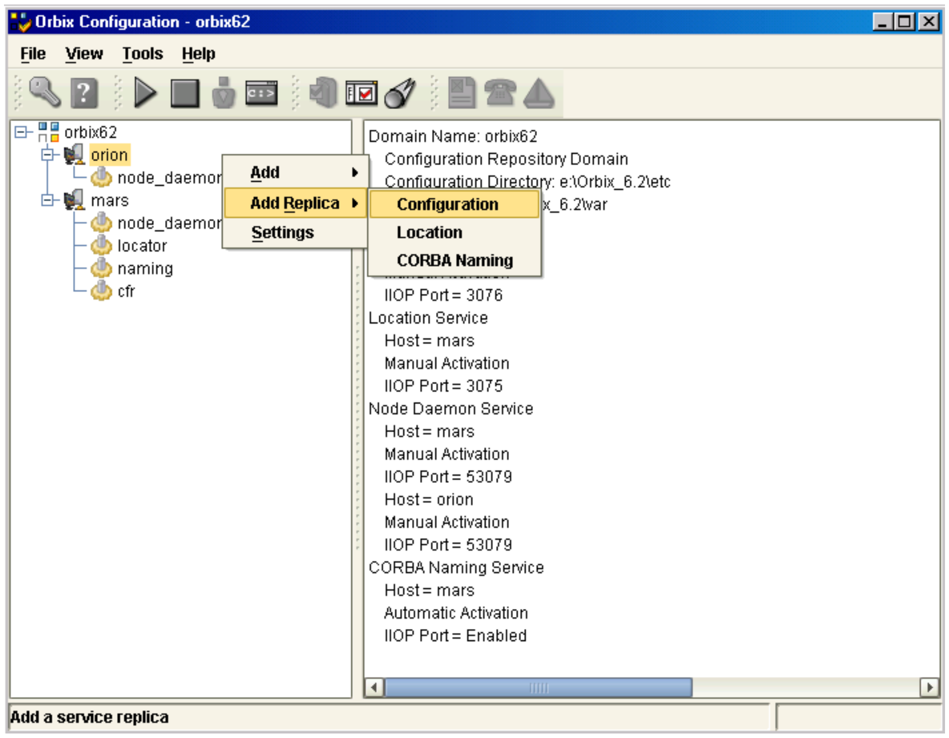
- Create server objects on demand.
- Maintain a bounded cache of most-recently-used server objects.
- Implement many CORBA objects of the same type with a single C++ or Java object in memory.

3.2.2 Extensible Configuration

Orbix is deployed in very large complex production environments. The configuration needs of such environments are different from those needed in a development environment. Production environments are long lived and subject to change. Hardware might need to be added or removed for a variety of reasons. The system might need functional extensions and new Orbix services, such as trader or notification, may need to be added. Higher demands on availability and failover often require the addition of more replicas. Lastly, Orbix administrators need to be able to upgrade from previous versions of Orbix with minimal effort. The time slot available for changing a production system can be as short as two hours in the middle of the night.

Orbix administrators are not CORBA developers and should not need to be familiar with the internals of Orbix configuration. The most frequent changes are small incremental changes to configuration domains. With Orbix 6.2, the Orbix configuration tool, *itconfigure*, supports extensible configuration. Everything does not need to be configured up front when a system is first deployed. Domain functionality can be extended at a later stage by deploying, for example, a trading service, or adding or removing service replicas.

This new functionality is available in both graphical mode (see *Figure 2*) and from the command line. Administrators who manage hundreds or thousands of Orbix servers can use scripting tools to rollout configuration changes on a mass scale.



Domain Name: orbix62
  Configuration Repository Domain
  Configuration Directory: e:\Orbix_6.2\etc
  x_6.2\war

  IIOP Port = 3076
Location Service
  Host = mars
  Manual Activation
  IIOP Port = 3075
Node Daemon Service
  Host = mars
  Manual Activation
  IIOP Port = 53079
  Host = orion
  Manual Activation
  IIOP Port = 53079
CORBA Naming Service
  Host = mars
  Automatic Activation
  IIOP Port = Enabled

Add a service replica

Figure 2: Orbix Configuration GUI

3.2.3 Secure Sockets Layer/Transport Level Security (SSL/TLS)

The Secure Sockets Layer (SSL) provides data security for applications that communicate across networks. SSL/TLS (Transport Layer Security) is a security protocol that sits in between various application protocols and transport level protocol, TCP/IP.

CORBA applications communicate using the CORBA standard Internet Inter-ORB Protocol (IIOP). This application-level protocol is layered above TCP/IP. SSL/TLS applications communicate using IIOP layered above SSL/TLS.

The SSL/TLS protocol provides connection security that has three basic properties:

- **Authentication—**The connection can be authenticated using asymmetric (public key) cryptography. SSL/TLS supports authentication based on RSA and DSS algorithms with X.509 certificates.
- **Privacy—**The connection is private. During the initial handshake, public key encryption is used to define a symmetric secret session key. Symmetric cryptography is used for data encryption.
- **Integrity—**The connection is reliable. Message transport includes a message integrity check, using a keyed Hashed Message Authentication Code (HMAC). Secure hash functions (for example, SHA-1 and MD5) are used for HMAC computations.

In addition, Orbix SSL/TLS features include:

- Sophisticated, mechanism-neutral API based on CORBASEC Level 2 interfaces.
- Support for the OMG Common Secure Interoperability specification, version 2 Level 0 (CSIv2) includes username/password authentication, identity propagation control fully integrated with the Orbix security server, and a single sign-on CORBA login service.
- Separate Key Distribution Mechanism (KDM) component. The ORB can distribute pass-phrases to automatically launched server applications. The server uses these pass-phrases to decrypt the relevant private key. KDM communications are fully TLS secure (encryption, privacy and integrity are guaranteed).
- An extensive X.509 C++ parsing API is supported, providing a complete IDL wrapping of X.509v3 certificates, and including X.509v3 extension support (note that use of this API is only required for advanced applications).
- PKCS#12 container format support.

3.2.4 Smart Card Support

Smart cards are used to store user credentials more securely than on a user's machine. It is a stronger authentication mechanism because users must be in possession of a smart card, rather than a certificate stored on a computer disk (the smart card plugs into a USB port or is read through a reader attached to the serial port). When employees leave a company, they must return their smart cards, whereas a file-based private key could be copied and transferred to a home machine.

A Java Orbix application can obtain credentials from a smart card as well as from a file. The API used is the standard JDK 1.4 JSSE API with cryptographic support provided in the JDK JCE interface. There are a number of implementations of the JCE modules that support cryptographic hardware devices available from third parties. This is designed for replaceability and allows customers to plug in any smart card that provides a JDK 1.3 or 1.4 JCE API-compliant wrapper.

For C++ Orbix applications, use of the Microsoft Crypto API (MS CAPI) is supported through the Schannel toolkit. Microsoft Windows provides a Security Support Provider Interface (SSPI) for plugging security providers into the system. One of the standard Windows security providers, called Schannel, is a software implementation of the SSL/TLS security protocol.

Schannel uses MS CAPI to implement cryptographic functionality required by the SSL/TLS protocol. Since practically all cryptographic hardware vendors make their devices available as a MS CAPI Cryptographic Service Provider (CSP), the use of Schannel enables access to a great many existing cryptographic devices, smart cards, cryptographic tokens, and so on.

3.2.5 Event Service

Orbix implements the CORBA Event Service specification, which defines an asynchronous model to supplement the synchronous request/response model normally used for communication between CORBA client and server applications.

The CORBA event service introduces the concept of *events* to CORBA communications. An event originates at an event *supplier* and is transferred to any number of event *consumers*. Suppliers and consumers are completely decoupled: a supplier has no knowledge of the number of consumers or their identities, and consumers have no knowledge of which supplier generated a given event.

3.2.6 Interoperable Naming Service and Load Balancing Extensions

Orbix supports the CORBA Interoperable Naming Service specification, a superset of the original CORBA Naming Service specification. The interoperable naming service adds some ease-of-use features and provides a standard URL format for CORBA object references, to simplify configuration and administration of CORBA services.

3.2.7 Bidirectional GIOP

The Orbix GIOP plug-in supports the CORBA standard for connection establishment between client and server. Typically, a client can open a connection to a server through a firewall, but it is not possible for the server to open a new connection back to the client in order to send a callback.

Bidirectional GIOP is a simple and efficient solution to this problem. It allows connections from the client to the server to be reused for callbacks from the server to the client. It is applicable over any connection-oriented transport, such as IIOP, IIOP/TLS or SHMIOP.

Orbix bidirectional GIOP is compatible with GIOP 1.2 (that is, it is not dependent on a GIOP 1.4 `NegotiateSession` message) and supports bidirectional invocations on Orbix 3 callback references.

In bidirectional communications secured by TLS, the connections are validated by mutual authentication.

3.2.8 Compression Plug-in

The Orbix ZIOP compression plug-in provides optional compression of GIOP messages on the wire. Compressed and uncompressed transports can be mixed. This can provide significant performance improvements on low bandwidth networks. The performance improvements depend on the network as well as the message data. If, for example, the requests contain jpeg images, there will be virtually no compression, whereas with repetitive string data, there will be good compression.

Compression can be configured per ORB as well as per binding (via proprietary ORB policies). The compression is done using a configurable compression library. The plug-in comes with support for `gzip`, `pkzip`, and `pzip2` compression algorithms.

3.2.9 Asynchronous Messaging Interfaces

Orbix implements some key features of the CORBA Messaging specification from CORBA 3.0. Asynchronous Messaging Interfaces (AMI) enable clients to make type-safe, asynchronous invocations of normal CORBA operations. The IDL compiler generates an AMI stub from a normal synchronous interface, and clients can use the AMI stub to make a request and then do other work until the reply is ready. The ORB delivers the reply by invoking a client-supplied callback object. The client's use of the AMI does not require any special support by the server that implements the original synchronous interface.

3.2.10 CORBA Reflection and Dynamic Type Support

Orbix 6.2 supports CORBA reflection. This is the ability to discover and dynamically invoke interfaces at runtime. CORBA provides reflection by way of its Dynamic Invocation Interface (DII), Dynamic Skeleton Interface (DSI), `DynAny` interface, CORBA reflection, and the Interface Repository (IFR). The DII and the IFR have been available since the first releases of the CORBA specification. The DSI and `DynAny` interfaces were introduced in CORBA 2.x, whereas CORBA reflection was invented by IONA and will become part of the CORBA specification sometime in 2005.

Orbix, therefore, provides full support for handling data types that are not known at compile time. The IFR stores information about all CORBA types known to the system and can be queried at runtime. Orbix users can also choose to have their objects support runtime introspection by using the appropriate IDL compiler switches, thereby eliminating the need to run an instance of the IFR. Either way, client applications can construct requests based on the available runtime type information using the DII, and with the DSI, servers can implement "universal" objects that can implement any interface at runtime.

The `DynAny` interface allows clients and servers to interpret or construct values based purely on runtime information, without any compiled-in data types. Together, these features are ideal for building generic object browsers, type repositories, or protocol gateways that map CORBA requests into some other middleware protocol. Interfaces discovered by way of the `DynAny` interface can also be dynamically modified; for example, by adding or removing interface methods.

In addition, Orbix provides the CORBA reflection module, which provides IDL interfaces that allow objects to support runtime metadata queries. This module implements an `XMLProvider` interface, which has a single `get_description` operation that returns a string. Using this interface, a CORBA client application can directly query an object for its metadata, and the client can use regular XML parsing tools such as DOM or SAX to parse the returned XML-formatted metadata. A variant of the `XMLProvider` interface is currently being standardized by the OMG for inclusion into CORBA sometime in 2005.

3.2.11 Configuration and Logging Interfaces

Applications can store their own configuration information in configuration domains, and access them using IDL interfaces—thus taking advantage of the infrastructure Orbix provides for ORB configuration.

Applications can also use IDL interfaces to log diagnostic messages to the logging subsystem. These messages are logged to whatever log-stream objects are registered with the ORB. Log streams for local output, file logging, and system logging (UNIX `syslogd` or Windows NT event logging service) are provided with Orbix. Application developers can also implement their own log streams to capture ORB and application diagnostics and send them to any desired destination.

3.2.12 Persistent State Service

Orbix includes an implementation of the Persistent State Service (PSS), a CORBA service that interposes a CORBA-based abstraction layer between a server and persistent data. The implementation of PSS is integrated with Berkeley DB, an efficient embedded database that is included with the ORB. Orbix uses PSS extensively in its internal operation. In addition, the PSS interface is available for use by applications.

3.2.13 Code Generation Toolkit

The Orbix developer kit includes a code generation toolkit for rapid application development. At the heart of this toolkit is an IDL compiler integrated with the Tcl scripting language. Out-of-the-box scripts can generate a complete and operational client/server application automatically from an IDL file. The toolkit also provides a useful debugging tool—you can use an auto-generated server to debug your client, and vice-versa. Advanced users can write their own code generation scripts to automate repetitive coding in a large application.

The code generation toolkit is supplied with a set of ready-to-use code generation scripts. By simply running these scripts, you can do common tasks such as creating a complete client-server application from an IDL file, creating functions that print your IDL data types, and so on. The toolkit also includes libraries of Tcl procedures that enable you to create your own customized code-generation scripts.

3.3 Orbix Enterprise

Available in both C++ and Java for a variety of NT and UNIX platforms (including Linux), Orbix Enterprise is the most advanced CORBA product available today. In addition to all the features and benefits described above for Orbix Standard (see subsection 3.2), Orbix Enterprise also offers the following features and benefits:

- Orbix management and administration tools.
- Security.
- High availability of Orbix services through replication clustering.
- Advanced messaging architecture.
- CORBA Object Transaction Service.
- CORBA Trader Service.
- IONA Firewall Proxy Service (FPS).

3.3.1 Orbix Management and Administration Tools

Orbix Enterprise provides easy to use management and administration tools that address the biggest system management problem facing enterprises that run large-scale, mission-critical systems. In such enterprises, operations groups must deal with hundreds or thousands of servers, many different operating systems, and an even greater number of different software vendors and products—while in a “must not fail” system environment.

In addition, many organizations are contractually obliged to provide reports to customers proving compliance with service-level agreements. These reports are based on the data provided by the Orbix enterprise management tools. In addition, because Orbix is pervasive enough to give a system-wide view, these tools can help to identify bottlenecks in the overall system. And, the performance impact of using them is very low.

System management capabilities in Orbix support test and debug, fine-tuning of applications, and operational control and support. Orbix management facilities provide mechanisms to set thresholds on critical system attributes in order to alert devices, system operators, or other software components of problems without requiring human intervention.

The Orbix management framework also provides for integration with the major third-party Enterprise Management Systems (EMS), such as IBM Tivoli, HP OpenView and BMC Patrol. This enables system administrators and production operators to monitor enterprise-critical applications from a single management console.

In addition, Orbix provides management tools that you can be use to define and implement managed entities (`MBean`s) specific to your environment.

All of this adds up to a management service that flexible and extensible enough to enable you to decide what is appropriate for your system. It includes the following major features:

- Performance logging
- Enterprise Management System integration
- Management customization tools

3.3.1.1 Performance Logging

Orbix 6.2 introduces client-side and server-side performance logging. This gives metrics on server availability and response time. It does not require any changes to code. A simple configuration setting is all that is required to set this in action.

Orbix 6.2 also includes a plug-in that monitors managed entities. It gathers statistics on whatever has been instrumented and stores them in the log file. For example, the Orbix work queue has been instrumented and its length can be monitored. In addition, any application-level managed entities can be monitored.

3.3.1.2 Enterprise Management Systems Integration

Orbix can be integrated with several Enterprise Management Systems (EMSs). These include BMC Patrol, HP OpenView, and IBM Tivoli. An integrated management system means that fault reports can be organized and correlated so that operators can find the cause of a problem, rather than being swamped by the symptoms.

Having a single management console reduces the learning curve for the operations staff. In addition, an EMS helps by providing the automatic triggering of recovery actions when problems occur. And, an integrated EMS enables service-level agreement compliance to be monitored and the business impact of system problems to be analysed.

Orbix EMS integration is based on a generic architecture for EMS systems integration. This provides abstract interfaces for the functions provided by the EMS, as shown in *Figure 3*.

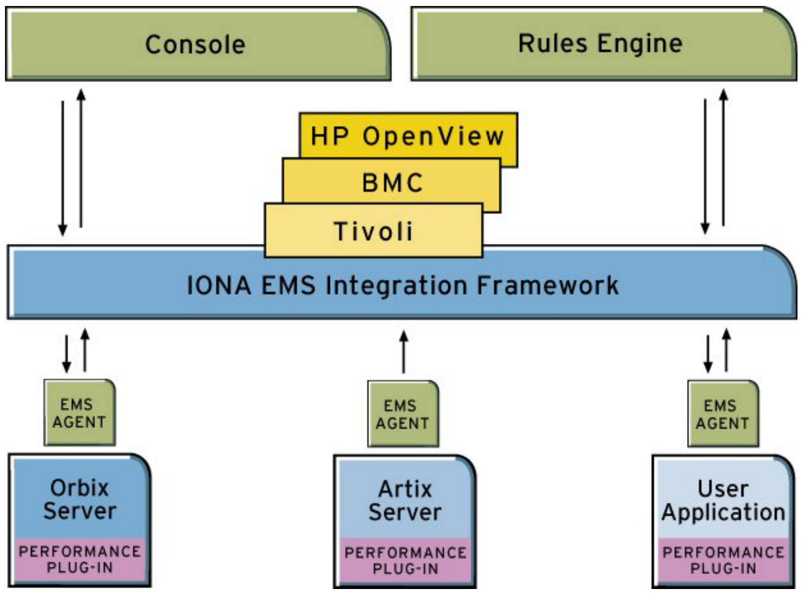


Figure 3: Orbix Management Architecture

In *Figure 3*, EMS components are shown in green. The IONA-supplied plug-ins, shown in purple, use abstract interfaces to monitor and log key system measurements to the log file. This information is then mapped to the requirements of the EMS; for example, Tivoli Resource Models.

System management information for Orbix systems and applications is transferred to, for example, the Tivoli EMS, to inform the operations staff of significant events, including:

- Whether a server is alive or dead (an event is posted when a server becomes inoperative).
- Server metrics, including the number of invocations received, and average, maximum, and minimum response times (events can be generated when any of these parameters go out of specified bounds).
- Perform actions on servers. By default, these actions include start, stop, and restart, but the list of possible actions is extensible.

The capabilities of Orbix management tools integrated with, for example, Tivoli include:

- Setting thresholds.
- Monitoring endpoints.
- Detecting server crashes.
- Detecting response time problems.
- Viewing historical data.
- Manual start and automatic restart of servers.

In addition, IONA provides EMS installation utilities that help users to integrate an Orbix installation with, for example, Tivoli in a matter of hours. These installation utilities:

- Create directories for configuration files, logs, and scripts.
- Generate the configuration “glue” between Orbix and Tivoli.
- Generate a package to install on the Tivoli server.
- Instantiate sample monitoring profiles and task libraries with pre-filled arguments.

3.3.1.3 Management Customization Tools

Orbix provides APIs that let you define and implement managed entities specific to your environment. You can then register these with the Orbix management service, and monitor and access them through the IONA Administrator Web Console. This is an optional feature for deeper, more advanced management.

The Orbix runtime provides managed entities that you can use without the need for coding; for example, out of the box, you can monitor application status and health, as well as such attributes as Orbix work queue length and server throughput.

3.3.2 Security

The Orbix security architecture, shown in *Figure 4* and *Figure 5*, supports several new Orbix security features that increase the security of the enterprise—and make life easier for developers, integrators, and users. User authentication is easier due to a choice of user/password-based authentication and X.509 certificate-based authentication, single sign-on and smart card or smart token support. Developers can implement application security more easily, thanks to new CORBA security features such parameterized ACLs and CSIv2. Integrators benefit from the Orbix Security Service, which makes it easy to integrate with arbitrary enterprise security solutions.

In addition, Orbix enhances the security of the entire enterprise by providing support for strong authentication within the Orbix infrastructure itself.

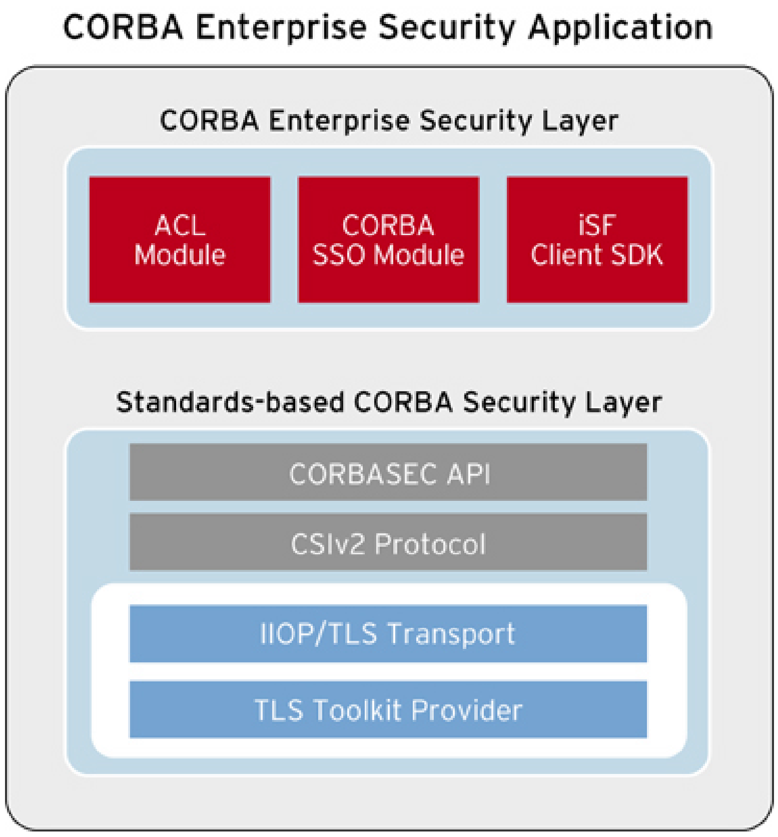

CORBA Enterprise Security Application
CORBA Enterprise Security Layer
ACL Module
CORBA SSO Module
iSF Client SDK
Standards-based CORBA Security Layer
CORBASEC API
CSIv2 Protocol
IIOP/TLS Transport
TLS Toolkit Provider

Figure 4: Architecture of Orbix CORBA server application that has been security enabled.

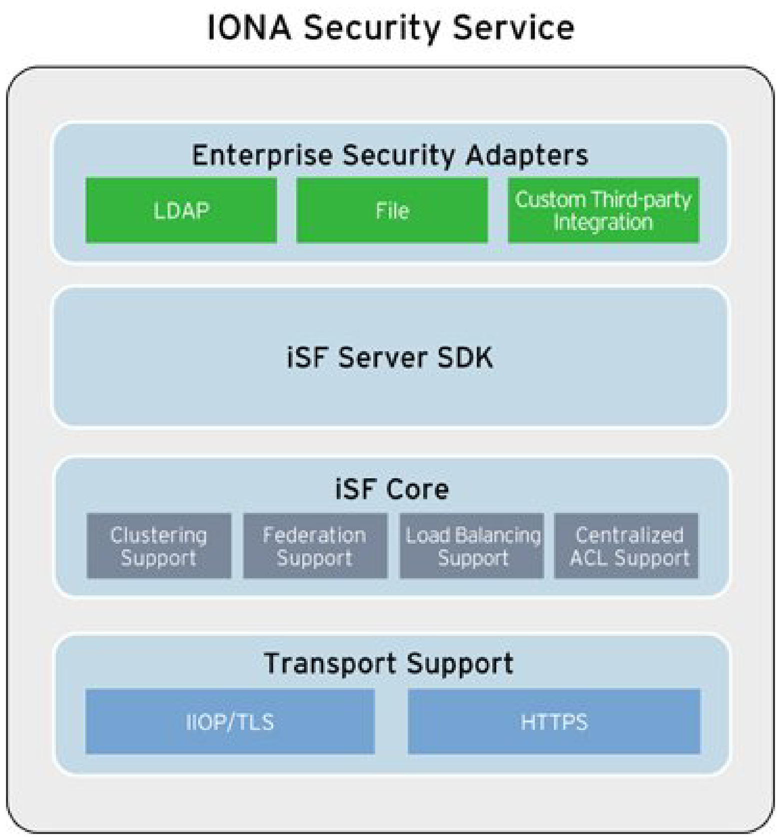


Figure 5: Architecture of IONA's Security Service, which is designed to integrate with your choice of enterprise security solution.

The Orbix Security Service is a scalable, standards-based security implementation with the following features:

- Unified security platform that works across various IONA products such as Orbix and Artix.
- Integration with third-party enterprise security systems via pluggable enterprise security adapters.
- Flat file and LDAP enterprise security adapters are provided out of the box.
- Role-based access control.
- Logging.

The Orbix Security Service provides an authentication service, an authorization service and a repository of user information and credentials. When it is deployed in standalone mode, all application types, including J2EE and CORBA applications, can call it remotely.

The following security standards are supported by the Orbix Security Service:

- Secure Sockets Layer / Transport Layer Security (SSL/TLS).
- CCITT X.509, which governs the form of security certificates based on public (asymmetric) key systems.
- OMG Common Secure Interoperability specification, version 2 (CSIv2).
- Security Assertion Markup Language (SAML).

The Orbix Security Service is designed to be integrated with third-party enterprise security solutions, to support customers who have already made a product choice or created their own authentication and authorization service.

Orbix also provides the IONA Security Framework (iSF) server adapter software development kit (SDK). The iSF server adapter SDK supports the creation of new iSF server adapters to allow integration with arbitrary enterprise security systems.

The following security features are also available in Orbix:

- Clustered security service.
- Federation of security server instances across different technology and security policy domains.
- Centralized ACLs.
- X.509-based authentication/authorization/SSO.
- Username/password or token-based authentication/authorization/SSO.
- Configuration repository authorization support.
- Sophisticated parameterized ACL support.
- CSIv2—providing industry standard secure interoperability and identity propagation.
- Securing of additional Orbix infrastructure (CFR, information repository, and node daemon, as well as the activator, locator, security, management, and naming services).
- Pluggable TLS toolkit provider support.

3.3.2.1 Clustered Security Service

New in Orbix 6.2, multiple security servers can be deployed to remove any single points of failure through automatic failover to backup servers, over IIOP/TLS secured connections (see *Figure 6*). Orbix security supports load balancing across security server instances in a security service cluster.

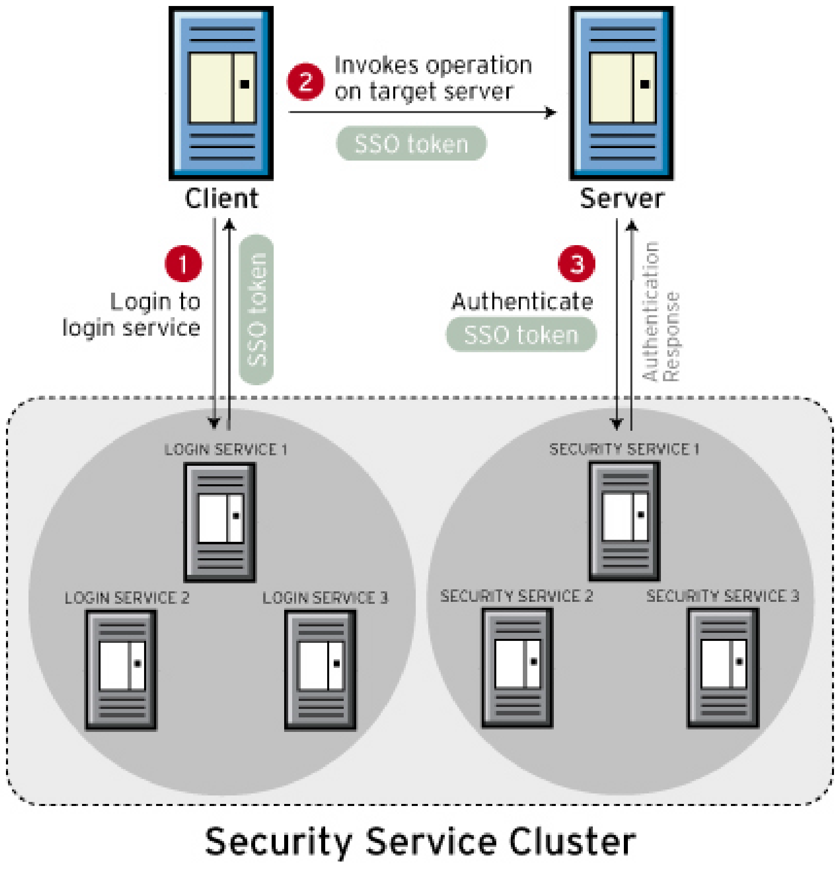


Figure 6: Orbix Security Service Cluster

The example scenario shown in *Figure 6* can be described as follows:

1. The client contacts the login service to obtain a single sign-on (SSO) token. Because the client is configured to perform random load balancing, it chooses one of the login services at random and opens a connection to that service. The login service sends back an SSO token.
2. The client invokes an operation on the server and sends the SSO token with the request.

3. The server authenticates the SSO token received from the client before allowing the client invocation to proceed. To authenticate the SSO token the server invokes on the security service cluster. If the server has an existing connection to a security service in the cluster, it reuses that connection. Otherwise, the server randomly selects a security service.

The login service is co-located with security service instances by default, but can be deployed separately as shown in *Figure 6*, if additional flexibility is required.

Security servers can be federated so that you only need to sign on once to have access to multiple security domains.

3.3.2.2 Centralized ACLs

Orbix 6.2 introduces a new option for centralized access control list (ACL) policy definition. By default, a secure Orbix application is configured to store its ACL locally. In large deployments, therefore, ACL files might be scattered over many hosts. From an administration point of view, however, it is often more convenient to gather ACL files onto a central host. Orbix 6.2 enables you to do just that. You can configure your secure applications to use such a centralized ACL repository. As a result, you can administer all of the ACL data in one place, making it easier to update and maintain.

3.3.2.3 Replaceable ACL Engine

The authorization engine can be replaced with a custom ACL implementation or with a third-party ACL engine such as Netegrity or Oblix. This custom ACL implementation can access IONA Security Service authentication information to make its access decisions. Support for replacing the ACL engine is available in for both local and centralized ACL modes.

3.3.2.4 Single Sign-On

Single sign-on (SSO) means that an application user needs to authenticate himself only once, even if the application spans security technology and security policy domains. SSO significantly increases the security of password deployment and usage, because the username and password information is visible only to the CORBA login service.

Orbix SSO support is layered above CSIv2, and supports both username and password, and TLS X.509 client authentication. Clients can use SSO tokens to authentication with server applications, and TLS X.509 clients can authenticate themselves to servers that demand CSIv2 username and password authentication by obtaining an SSO token from the CORBA login service.

CORBA clients can simply be configured to use SSO—no code changes are required. In addition, SSO token expiry and automatic refreshing of SSO tokens is handled in a manner transparent to the application, which avoids the need to write any application exception handling code.

3.3.2.5 Configuration Repository Authorization

Configuration repository authorization support allows administrators to define ACLs for configuration scopes, thus preventing unauthorized users from reading and writing sensitive scopes. The hierarchical nature of the Orbix configuration scoping mechanism makes it easy to define ACLs for groups of related applications that share a parent configuration scope.

CFR authorization is supported in a replicated CFR environment.

3.3.2.6 ACL Extensions

Orbix provides sophisticated parameterized ACL support. This means that access control can be based on the values of a parameter in a method call. Take, for example, a file server application that controls access to data on a machine. The user can ask for the data in a file by passing over the fully qualified name of the file. Access to the data can be controlled by first examining the directory in which the file resides. If the file resides in a restricted area, the call can be prohibited.

Parameterized ACL is achieved by defining your own “action analyser” plug-ins. These plug-in objects determine what resources are being accessed and by what attempted actions. The ACL enforcement mechanism ensures that the user has the required access rights. This is a two-step process:

1. Determine the logical action(s) being carried out.

2. Apply the ACL engine to the logical action(s).

Orbix ACLs also provide extensive wildcarding support for server names, interfaces, and operations. Applications can use the Orbix enhanced ACL support without changing any code.

3.3.2.7 Shared Credentials Support

Orbix supports shared credentials across multiple ORBs. This simplifies credential gathering for applications that use multiple ORBs or that use plug-ins that create internal ORBs. An application’s credentials can be shared across all ORBs that are configured to support credential sharing.

3.3.2.8 CSIv2 Support

The Common Secure Interoperability specification, version 2 (CSIv2) level 0 enables interoperable TLS over IIOP/TLS authentication and delegation in the service context of General Inter-ORB Protocol (GIOP) request/reply messages, over a connection-based transport. CSIv2 is normally used in conjunction with SSL/TLS. It is possible to use it over insecure connections. This would not, however, normally be recommended for secure deployments.

Orbix provides powerful control over all aspects of CSIv2, including policies that control the acceptance and transmission of propagated identities.

3.3.3 High Availability of Orbix Services through Replication Clustering

One of the most powerful features in Orbix is its server clustering architecture. With server clustering, it is possible to group together multiple physical servers—each of which may be running on a different machine—into a single logical server. To clients using the server, the appearance is that of a single server process, but the Orbix infrastructure distributes invocations across the set of server processes in the cluster.

Server clustering greatly improves the reliability of the system because the failure of one server in a cluster does not result in a loss of service—there is always another server available to take its place. The clustering mechanism is transparent to the application; when one of the servers in the cluster fails, the infrastructure detects the failure and automatically reroutes clients to another functioning server. As long as there are no transactions in progress on the server when it fails, the entire failover process is invisible to the application.

Server clustering also dramatically improves the overall performance of the system because it provides *load balancing*. This ensures that no single server becomes a bottleneck in the system, and lets you take advantage of the processing power of multiple machines.

Orbix provides several out-of-the box strategies that can be used to distribute client requests across the different servers in the cluster, including round robin, random distribution, and a feedback-driven method that routes new clients to the least-loaded server. These strategies can be applied on a per-service basis, which lets you employ different strategies for different applications.

In addition to supporting the development of clustered applications, the ORB infrastructure itself uses clustering to achieve high availability. Each of the CORBA services, including the locator, naming service, trading service, and configuration repository, can be grouped into a server cluster.

In Orbix 6.2, changes have been made at the Berkeley DB level. Berkeley DB has the ability to propagate replication data between different instances of the database. Orbix inherits this ability to replicate and propagates the data across the network through the PSS layer (see *Figure 7*). For the user this provides a dramatic performance improvement when slaves are being promoted to master. Unlike in previous Orbix releases, the database does not need to be opened, closed and recovered with each replication update on a slave replica.

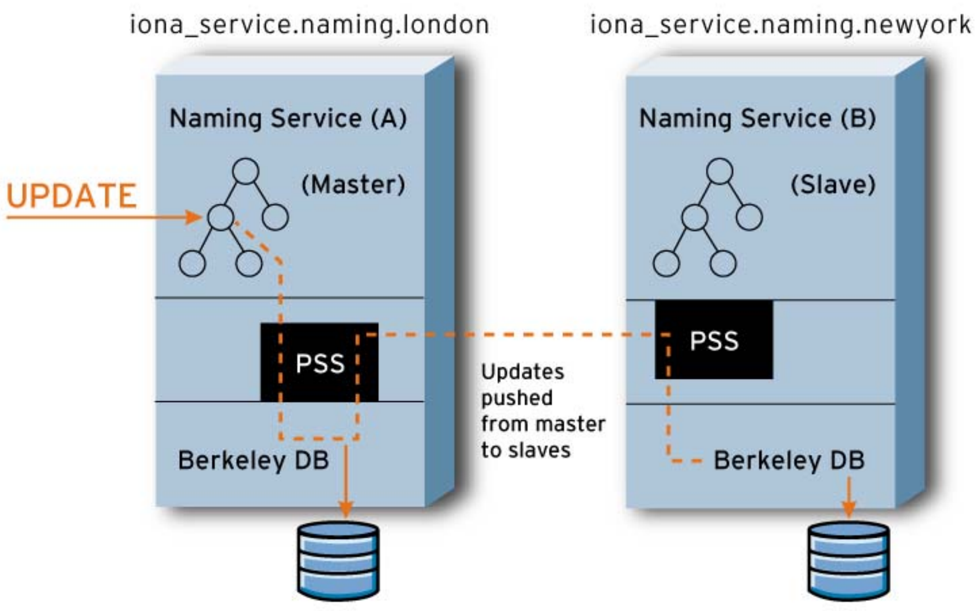

iona_service.naming.london
iona_service.naming.newyork
Naming Service (A)
(Master)
UPDATE
PSS
Berkeley DB
Naming Service (B)
(Slave)
PSS
Berkeley DB
Updates pushed from master to slaves

Figure 7: Orbix Replication

If the master fails, a slave is automatically promoted without the need to restart any services or make any changes to configuration. During the promotion period, write operations are blocked until a new master is chosen or until a configurable timeout occurs. Berkeley DB has an election protocol that guarantees that the most appropriate slave is promoted when the master fails. Berkeley DB can check which slave is most up to date. The most up-to-date slave is always elected first. If all slaves are at the same level, then they are promoted according to a priority setting. If no priorities are assigned, slaves are promoted randomly.

3.3.4 Advanced Messaging Architecture

Orbix Enterprise includes implementations of both the CORBA Event Service and the CORBA Notification Service specifications. IONA's implementation of these services provides both connection and event reliability. In addition, Orbix provides an implementation of the CORBA Telecom Log Service specification, and a bridge between the Java Messaging Service (JMS) and the CORBA Notification Service. These features are described in the following subsections.

3.3.4.1 CORBA Notification Service

The Orbix notification service supports event reliability and connection reliability, thus enabling channels to be configured to recover their state after process restart, and to provide highly reliable event delivery. The notification service has a built-in database (Berkeley DB), enabling persistence of the notification messages. Open Data Base Connectivity (ODBC) is not required. The Orbix notification service implementation provides twice the throughput of the ASP 6.0 version.

- **Connection Reliability—**Specifies whether or not the channel retains persistent information about all of its clients, including their filters and configured quality-of-service properties. By retaining this information, the channel's state can be dynamically recreated upon restart.
- **Event Reliability—**Specifies whether or not the channel stores a persistent copy of the event, so that it can guarantee delivery to all consumers even if the channel server fails.

3.3.4.2 Multicast

The notification service supports User Datagram Protocol (UDP) based IP multicast as an alternate plug-in transport layer (thus subsuming the functionality of OrbixTalk).

A multicast group is the set of hosts that receive messages on a particular IP multicast address. A multicast group can span multiple networks. Hosts can join or leave multicast groups at any time. Adding hosts to a multicast group does not affect the number of messages sent over the network—a single message is sent regardless of the number of hosts in the multicast group. Thus the multicast transport service reduces the load on network resources and scales easily to large numbers of receivers. Because Orbix implements multicast as a plug-in, a single channel can support both IIOP and multicast messages.

3.3.4.3 CORBA Telecom Log Service

IONA's implementation of the CORBA Telecom Log Service is a high-performance, distributed logging system that provides support for long-term event storage and playback of historical events (for example, by clients that connect to the service after events they are interested in have already been emitted). Telecom logging provides a mechanism for creating a persistent log of events in a distributed computing environment and allows for the recovery of events in the face of a catastrophic failure.

When combined with the notification service, the telecom log service enables you to simply exchange an existing event or notification channel with a fully compatible logging version of the channel.

3.3.4.4 Java Messaging Service and CORBA Notification Bridge

The Java Message Service (JMS) is an enterprise-capable middleware component that provides the fundamental functions of message-oriented middleware (MOM). JMS support is built into Orbix, giving developers the ability to write applications that deliver high volumes of messages in a secure and reliable manner.

Orbix provides a bridging mechanism between the CORBA notification service clients and JMS clients. This bridge allows notification service clients and JMS clients to exchange messages. Using the bridge, JMS publishers can forward messages to CORBA notification consumers and CORBA notification suppliers can forward messages to JMS subscribers.

3.3.5 CORBA Object Transaction Service

Transactions are an important programming paradigm for simplifying the construction of reliable and available applications, especially those that require concurrent access to shared data. Today it is widely accepted that transactions are the key to constructing reliable distributed applications.

Orbix includes an implementation of the Object Transaction Service (OTS). IONA provides two varieties of the OTS, supporting single and multiple resources respectively. The single-resource OTS version supports transactional semantics for transactions that involve only one resource, for example, a single database. This version of OTS is bundled with Orbix Standard and Orbix Enterprise. The multiple-resource version supports transactions that span multiple resources and therefore require two-phase commit. The multiple-resource version of OTS is a separately-orderable component.

3.3.6 CORBA Trading Object Service

Orbix Enterprise provides an efficient, robust, and complete implementation of the CORBA Trading Object Service specification. Orbix trader is the mechanism by which instances of a particular kind of service advertise themselves, and by which other objects can discover these services. In this way, Orbix trader supports the dynamic discovery of services as well as the ability to bind to these services at runtime.

3.3.7 IONA Firewall Proxy Service (FPS)

J2EE application servers, and CORBA servers in the middle tier, use IIOP to communicate with services and resources, which means that IIOP-based messages need to traverse firewalls securely. Unfortunately, most TCP firewalls do not support IIOP traffic at the protocol proxy level. IONA's Firewall Proxy Service (FPS) is a configurable proxy that is placed on a "bastion" host between the server and its clients, as shown in *Figure 8*.

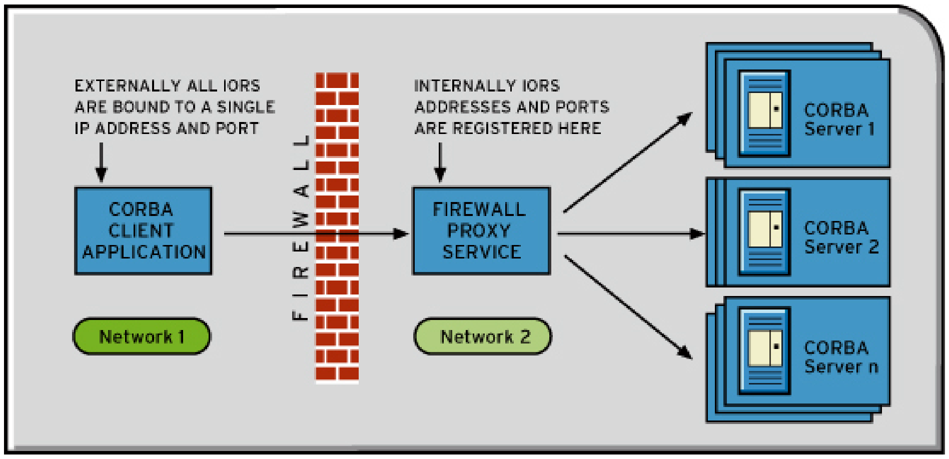

EXTERNALLY ALL IORS ARE BOUND TO A SINGLE IP ADDRESS AND PORT
CORBA CLIENT APPLICATION
Network 1
FIREWALL
INTERNALLY IORS ADDRESSES AND PORTS ARE REGISTERED HERE
FIREWALL PROXY SERVICE
Network 2
CORBA Server 1
CORBA Server 2
CORBA Server n

Figure 8: IONA Firewall Proxy Service

Rather than opening a large range of dynamically allocated ports, the FPS allows an administrator to open a limited number of specific ports.

FPS works by mapping interoperable object references (IORs) exposed to the external clients to those of the real CORBA servers. Only Portable Object Adapter (POA) based servers can be accessed through FPS. A server that uses FPS exchanges IOR template information with the service during a registration process that is initiated by the creation of a POA.

When a server has registered with FPS, all IORs generated by that service point clients to proxies managed by FPS. FPS maintains a persistent store of registration information, such that when FPS initializes, it recreates the bindings for any server that registered with it during a previous execution. This assures that server registration is persistent across many executions of FPS.

For performance reasons, the Firewall Proxy Service does not attempt any authorization or filtering of the messages. The Firewall Proxy Service supports IIOP traffic only. It does not support IIOP over SSL/TLS.

4 Support from IONA Professional Services and Partners

IONA offers up-front support contracts with a duration ranging from one to five years for Orbix 6.2. With more customers and more systems in production than any other CORBA vendor, IONA offers the highest quality CORBA support in the industry. We provide technical support for Orbix 6.2 at three levels:

- **Standard—**Expert technical help from IONA's Technical Support Center, eight hours a day, five days a week.
- **Silver—**Expert technical help from IONA's Technical Support Center, 24 hours a day, five days a week.
- **Gold—**Expert technical help from IONA's Technical Support Center, 24 hours a day, seven days a week.

In addition, IONA's Professional Services and our Consulting Partners provide experienced developers, architects and project managers to assist with the architecture, design, development, integration, rollout, support and optimization of your Orbix 6.2 applications. No matter what your integration challenges are, IONA consultants can play any number of roles, on both short and long-term engagements, to help you get the most out of Orbix 6.2.

Whether you are building a new application from the ground up, or working to improve an existing application, IONA consultants will keep your project on track by sharing their world-class expertise in areas such as:

- Architecture.
- Security.
- Scalability/performance/throughput.
- Systems management.
- High availability and fault tolerance.
- Server consolidation.
- Interoperability with other middleware such as .NET, COM and J2EE.
- Integration with the mainframe.
- Migration from earlier versions of Orbix and from third-party ORBs.

For an immediate impact on your project, we offer one-week assessments in any of the areas above that provide you with a specific set of steps to improve your application.

5 Conclusion

Orbix provides all of these capabilities through modern, open, and entirely standards-based technologies. Orbix 6.2 continues to excel at meeting the needs of integration and development teams that need to integrate CORBA with other technologies, while meeting the most demanding scalability, reliability, and performance requirements.

See www.iona.com for the current list of platforms supported by Orbix 6.2.

Contact Details

IONA Technologies PLC
The IONA Building
Shelbourne Road
Dublin 4
Ireland
Phone: ....+353 1 637 2000
Fax: ....+353 1 637 2888

IONA Technologies Inc.
200 West St
Waltham, MA 02451
USA
Phone: ....+1 781 902 8000
Fax: ....+1 781 902 8001

IONA Technologies Japan Ltd
Akasaka Sanchome Bldg 7/F
3-21-16 Akasaka
Minato-ku, Tokyo
Japan 107-0052
Phone: ....+813 3560 5611
Fax: ....+813 3560 5612

Support: ....support@iona.com
Training: ....training@iona.com
Orbix Sales: ....sales@iona.com
IONA's FTP site ....ftp.iona.com

World Wide Web: www.iona.com