



Enterprise Service Bus (ESB) Solution

*How Cape Clear Business Integration Suite applies SOA and Web Services
principles to deliver a proven ESB solution*



Cape Clear's Enterprise Service Bus (ESB) Solution

Cape Clear's Enterprise Service Bus (ESB) Solution (April 2004)

Copyright © 2004 Cape Clear Software, including this documentation, all demonstrations, and all software. All rights reserved. The document is not intended for production and is furnished "as is" without warranty of any kind. All warranties on this document are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

Trademarks

Cape Clear, Cape Clear Software, Cape Clear Business Integration Suite, Cape Clear Data Interchange, Cape Clear Studio, Cape Clear Server, and Cape Clear Manager are trademarks of Cape Clear Software in the United States and other countries.

All other company, product, and service names mentioned in this document may be trademarks or service marks of others.

CPV-DOC-3040

Contents

Introduction	4
SOA and ESB	5
ESB: A New Tier in the Enterprise Architecture	7
An Aside: Not All Data is XML Data and Not All Applications are XML-Aware	9
Key Capabilities of an ESB	10
Cape Clear's ESB Product Approach	13
Customer Scenarios	16
Summary	17
Resources	17

Introduction

Enterprise service bus (ESB) is a new architecture for lowering the cost of integration. It draws on the disciplines of service-oriented architecture (SOA) and the power of Web Services to radically change the technology and economics of integration projects. Since 1999, Cape Clear has been promoting a new approach to integration: an approach that offers increased flexibility at a lower cost. The Cape Clear Business Integration Suite provides organizations with an ESB that integrates applications and data in less time and cost than traditional integration approaches.

This document provides:

- A brief overview of SOA and ESB concepts and definitions.
- A description of how Cape Clear delivers on these requirements via the Cape Clear Business Integration Suite and how this approach differs from traditional solutions.

For a product suite to be considered an ESB, it must have the following properties¹:

- Deep native support for all relevant XML and Web Services standards.
- Transformation capability.
- Extensive routing support.
- Adapters for existing enterprise applications and infrastructures.
- Offer order-of-magnitude better economics than enterprise application integration (EAI) or customized integration approaches.

¹ Source: Gartner Group

SOA and ESB

The discipline of service-oriented architecture (SOA) has been well understood for a long time. The analyst firm Gartner Group first referred to the concept in 1996, and the idea arguably goes back to the early 1970s.

More recently, Gartner Group has made bullish predictions about the likely prevalence of SOA approaches:²

By 2008, SOA will be a prevailing software-engineering practice, ending the 40-year domination of monolithic software architecture (0.7 probability).

SOA is a set of principles: an approach to developing reusable services. The Web Services standards-based suite is a way to implement SOA in practice. Good Web Services design fosters SOA discipline, and good Web Services tools and platforms inherently mandate SOA approaches.

Enterprise service bus (ESB) is a product category, first identified by Gartner Group, that describes a set of products that simplify integration and broadly implement SOA disciplines using Web Services standards.

SOA principles, as applied to good Business Service and Web Service design, are discussed in more detail in the Cape Clear whitepaper "Principles of SOA Design", which is available from <http://www.capeclear.com>.

Gartner Group defines an ESB as follows³:

Enterprise service buses (ESBs) are a new kind of middleware that combines features from several previous types of middleware into one package. ESBs support Web services by implementing Simple Object Access Protocol (SOAP) and leveraging Web Services Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI). Many ESBs also support other communication styles that involve guaranteed delivery and publish-and-subscribe; those that don't soon will. All ESBs provide some value-added services beyond those found in basic communication middleware, such as message validation, transformation, content-based routing, security, service discovery for a service-oriented architecture (SOA), load balancing, failover, and logging. Some services are built into the ESB core, while others run in "plug-in" modules. ESBs have a distributed architecture wherein some services are executed near the application programs, rather than in a central hub. ESBs support Extensible Markup Language (XML) and often also support other message formats.

² "Service-Oriented Architecture Scenario", Note Number: AV-19-6751, April 16, 2003.

³ "Predicts 2004: Enterprise Service Buses are taking off", Gartner Group, December 2003.

IDC Research noted in a March 2003 Executive Brief that ESBs must:

- Be built around industry standards.
- Utilize JMS and MOM (that is, asynchronous) infrastructures.
- Function as both a transport *and* a transformation facilitator.

ESB: A New Tier in the Enterprise Architecture

Application servers are adept at hosting complex back-end business logic. Integration servers are adept at integrating packaged applications (but not via XML). It is Cape Clear's view that the new era of service-oriented enterprise software architectures requires that a new tier emerge in the enterprise architecture, where 'Business Services' are rapidly, opportunistically, and cost-effectively defined and implemented. The associated architecture is shown in Figure 1:

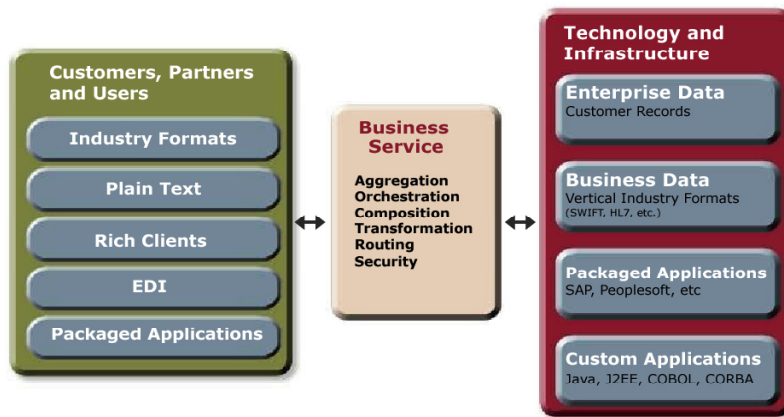


Figure 1: Architecture of service-oriented enterprise software

The 'Business Service' tier is where the ESB resides. It provides a way to define and deliver highly customized services and views of services, close to the client perspective, and without intruding on deep back-end application logic.

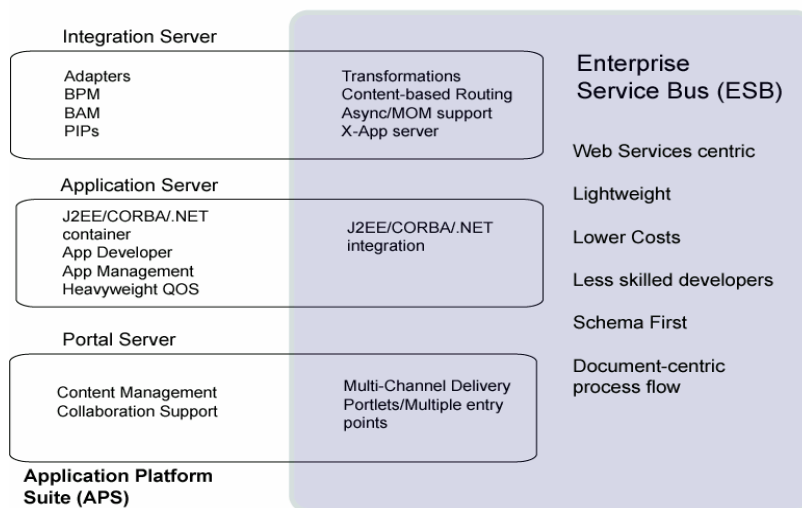


Figure 2: ESB as part of the Business Service tier

The ESB category includes certain elements of the application server, integration broker, and portal server technology and product sets. However, while each of these product categories has deep capabilities within its respective spheres of competence, each category is narrow in terms of its span, for example:

- **Application servers**, which are principally J2EE-based, are widely used to develop and deploy back-end server logic. They are very well suited for this purpose. However, while application servers are focused on supporting new application development, they don't natively support integration. Their approach to Web Services is to auto-enable all Java interfaces as WSDL. This leads to complex and non-interoperable service interfaces—making cross-platform and cross-application integration more complex and costly⁴.
- **Integration brokers**, as realized in enterprise application integration (EAI) suites, have historically been used for the integration of packaged applications via specific and often heavily customized adapters. Vendors like webMethods, Vitria, and SeeBeyond have built their businesses on offering extensive consulting services to make their proprietary adapters work with their customers' specific applications - adding time and cost to integration projects.

⁴ See <http://www.capescience.com/articles/wsdfirst/index.shtml>.

An Aside: Not All Data is XML Data and Not All Applications are XML-Aware

Many integration solutions and ESBs assume that all the data and applications that need to be incorporated in an integration effort are already magically accessible as Web Services or XML. It is Cape Clear's experience that this is an unrealistic assumption. A large variety of information, including content stored and shared in spreadsheets, databases, comma-separated, and other flat files, is often part of a given integration problem.

A spaghetti-like web of Perl scripts, custom Java code, manual processes, and extensive data re-entry is often the only way to incorporate these different data sources into an integration project. Cape Clear now enables you to rapidly build 100 percent-standard 'on-ramps' and 'off-ramps' to and from the Web Services world for *any* semi-structured data format. This enables you to define, capture, and manipulate the semantics of your business data as it flows through communications pipes, and not just at the programming endpoints. You can thereby include non-XML data as a 'first-class citizen' in the world of ESB and Web Services integration.

Many structured and standard data formats in a variety of industries (for example, EDI, SWIFT, ACORD, and Parlay) either do not have standardized XML representations or, where they do, they are not yet widely used in practice. The need to process these data types gives rise to the need to process and manipulate non-XML data in the context of ESB-based integration efforts.

Another common source of non-standard data occurs when company boundaries are crossed in the course of the execution of a business process. For example, if a financial services company wants to expose a given service (for example, equity trade execution) to a selection of third parties, it is very likely that their systems for trade execution requests will look very different from the service being exposed by the financial institution. Reconciling these incompatibilities is essential for successful integration with third parties. Achieving this requires the ability to map between different representations of the same service at multiple endpoints.

Key Capabilities of an ESB

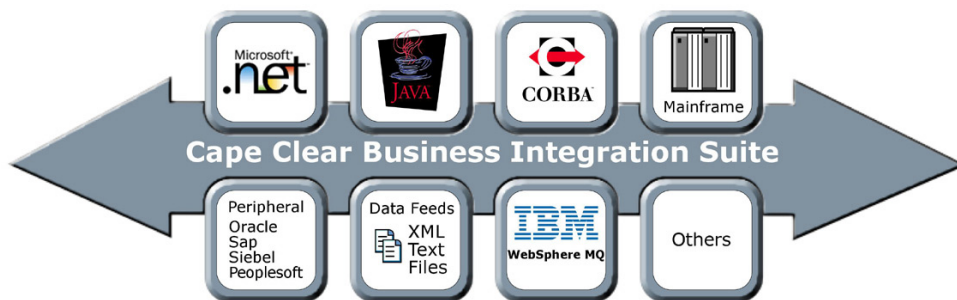


Figure 3: Cape Clear's ESB solution

An ESB, as represented in Figure 3, enables the more efficient value-added integration of a number of different application components, by positioning them behind a service-oriented façade and by applying Web Services technology to the problem. The physical product architectures and detailed functional requirements for an ESB product suite are described in Table 1 below.

Table 1: Key ESB features

Category	ESB Feature	Feature Description
XML/Web Services Standards	WSDL, SOAP, and UDDI support	An ESB product must offer support for the latest Web Services standards, including the WS-I Basic Profile 1.0. UDDI support enables the rapid creation of a repository of reusable services.
	XML Schema processing support	In order to parse, manipulate, and process potentially highly complex XML documents, an ESB must have comprehensive and sophisticated support for all XML schema types and processing modes and technologies, including XSLT, XPath, SAX, and DOM. Doing these tasks efficiently (for a high-performance runtime) is also key for performance and throughput-sensitive applications.
Transforms	On- and off-ramps to and from XML	Most data and applications don't speak XML natively. The ability to map these non-native data formats into and out of XML is critical. Without this capability, most data and applications can't talk to your new ESB, so your project is stillborn.

	Transformations within XML	Once you've got everything you need into the XML domain, you need to transform among various incompatible XML representations. This can be done in order to erase semantic differences, or to reconcile different client and server views of the same Business Service. This capability to create XSLT should be graphical, intuitive, and capable of handling arbitrarily complex transforms. The suggestion that you should hand-code XSL is ludicrous: the format of XSL is as inscrutable as PostScript.
Routing	Addressability	Simple routing looks just at headers or static addresses. Complex and flexible routing strategies require the ability to 'look into' message contents and make routing decisions accordingly. Deep (and easy to use) XPath support makes this possible.
	Transport	Provide support for multiple synchronous and asynchronous transports, via JMS.
	Content-based routing	Having access to content enables dynamic, content-based routing strategies, based on parameters and values on a per-message, per-client, or per-instance basis. Support for WS-Addressing and WS-Routing are a must for .NET interoperability and future-proofing.
	Complex (clustered or context-based, multi-hop) routing strategies, with extensibility	Not all required routes can be as simple as 'if A, then route to B'. Realistic scenarios can involve: <ul style="list-style-type: none"> ▪ Routes that involve multiple hops. ▪ Context-dependent routing to any one of a number of replicated or clustered service implementations. ▪ The ability to easily add new or extended routing strategies. ▪ Error handling and correction. ▪ Version control can also be implemented using customized router strategies.
Adapters	CORBA, J2EE,	Many enterprise applications, both packaged and

Cape Clear's Enterprise Service Bus (ESB) Solution

	.NET, mainframe, and MOM interoperability	custom-developed, are written using these venerable technologies. ESBs must be capable of flexibly and transparently reusing these components. If not, then they are only useful for green-field applications—an extremely small percentage of overall development work.
Price	Price	ESBs provide a simpler and cheaper solution than current approaches to solving complex integration problems. They should also be shrink-wrapped products and should not depend on heavy up-front or ongoing expert consulting assistance.
Quality of Service (QoS)	Security and robustness	Your ESB is going to offer business-critical services to potentially millions of entities, some of which may be external or hostile to your company. Therefore, the ESB infrastructure must be bullet-proof in terms of security, robustness, performance, and manageability.

Cape Clear's ESB Product Approach

Cape Clear has a straightforward approach to product delivery. Our principles include:

- **Radical simplicity.** In order to achieve the productivity benefits promised by an ESB strategy, products must enable less-skilled developers to be more productive at solving complex integration problems in a new way. This imposes requirements throughout the product set in terms of ease of installation, learning curve, complexity, and the user interaction model. The Cape Clear Business Integration Suite offers a highly graphical, fourth-generation set of products, the primary design goal of which is to drive complexity out of the integration process.
- **SOA and Web Services standards from the ground up.** Our WSDL-first/schema-first approach is evidenced by the fact that we have invested heavily in the development of tools for WSDL and schema design, as well as the generation of services from these views. This fosters inherently good SOA discipline—something that a 'code-first' approach cannot achieve.
- **Enterprise-class products.** Cape Clear has over 100 customers that are fully deployed in mission-critical environments, which require 24x7 uptime and support.

The Cape Clear Business Integration Suite offers a comprehensive and proven implementation of the ESB product category. It consists of four product elements, as shown in Figure 4:

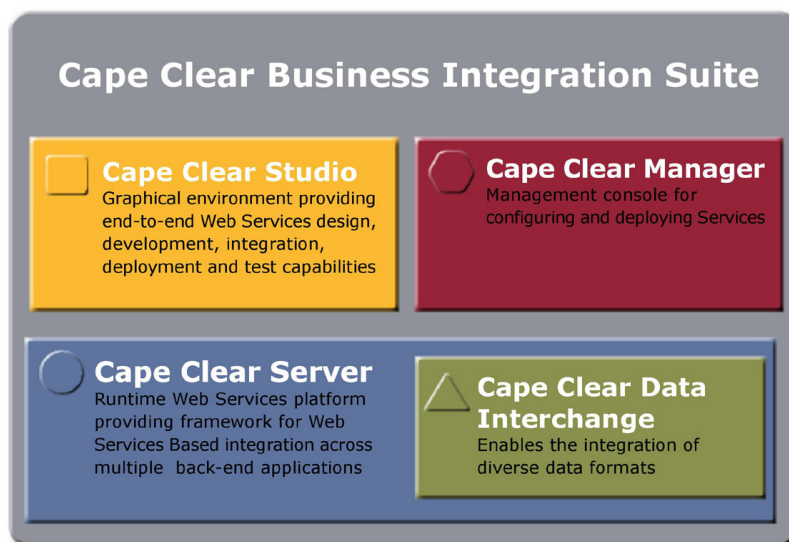


Figure 4: The Cape Clear Business Integration Suite

These four components implement key ESB requirements, as noted in Table 2:

Table 2: How Cape Clear meets ESB requirements

Category	ESB Feature				
XML/Web Services Standards	WSDL, SOAP, and UDDI support	✓	✓	✓	✓
	XML Schema processing support	✓	✓	✓	✓
Transforms	On- and off-ramps to and from XML			✓	
	Transformations within XML			✓	
Routing	Addressability		✓	✓	
	Transport support (IIOP, RMI, HTTP, SMTP, FTP, JMS)		✓	✓	
	Content-based routing		✓	✓	
	Complex (clustered or context-based, multi-hop) routing strategies, with extensibility		✓	✓	
Adapters	CORBA, J2EE, .NET, mainframe, and MOM interoperability		✓		
Quality of Service (QoS)	Manageability, security robustness, and performance	✓	✓	✓	✓

Cape Clear's Enterprise Service Bus (ESB) Solution

The schematic architecture of the product suite is shown in Figure 5. In addition to the Cape Clear Server and Cape Clear Data Interchange products, Cape Clear Studio is the design-time tool that facilitates the creation of services and integrations, and Cape Clear Manager is a browser-based (but JMX- and SNMP-integrated) management console that offers comprehensive control over deployed servers and services.

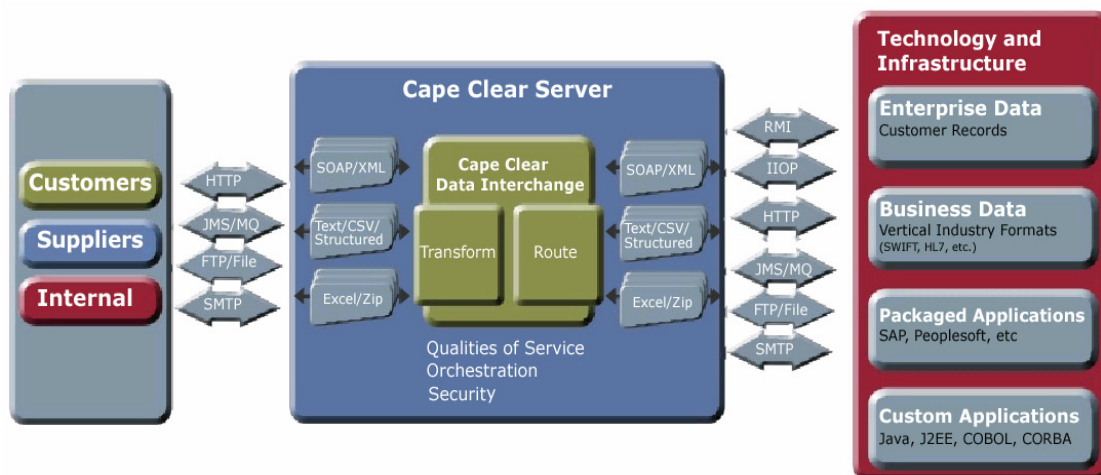


Figure 5: Architecture of the Cape Clear Business Integration Suite

The overall Cape Clear Business Integration Suite enables the rapid delivery of an ESB strategy, with minimal fuss, maximum reuse of existing applications and data assets, and demonstrably superior return on investment (ROI).

Customer Scenarios

There are three main scenarios in which Cape Clear's customers apply ESB approaches. While finance and telecommunications (operators and equipment vendors) are currently the dominant sectors, others such as manufacturing, federal government, and consumer goods are also strongly adopting the ESB approach.

- **Internal operations.** As a first step in the process, applying ESB and Web Services techniques offers a 'faster, cheaper, better EAI', enabling organizations to integrate and re-purpose existing application and data assets.
- **Customer connection.** Having defined, integrated, and deployed a number of 'internal' services, the next step is to make these services available to selected third parties, typically customers. Conservative approaches are to initially offer 'read-only' access to services, to be followed over time by 'update' services. It is important to note that extending internal services to be externally accessible is a simple process—unlike using a traditional approach such as application servers or EAI, where an entirely different logical, physical, and security architecture is required. With Cape Clear's ESB approach, externalizing a service is as simple as publishing a new URI.
- **Supplier linkage.** Analogous to customer connection, a natural extension of internally available services is to publish them to trusted suppliers so that respective applications and datasets can be viewed as part of the extended supply chain, again taking advantage of the simplicity offered by an ESB.

Figure 6 illustrates a number of other Cape Clear customers who have had similar success. For more details on these customers' experiences, go to

<http://www.capeclear.com/customers>.

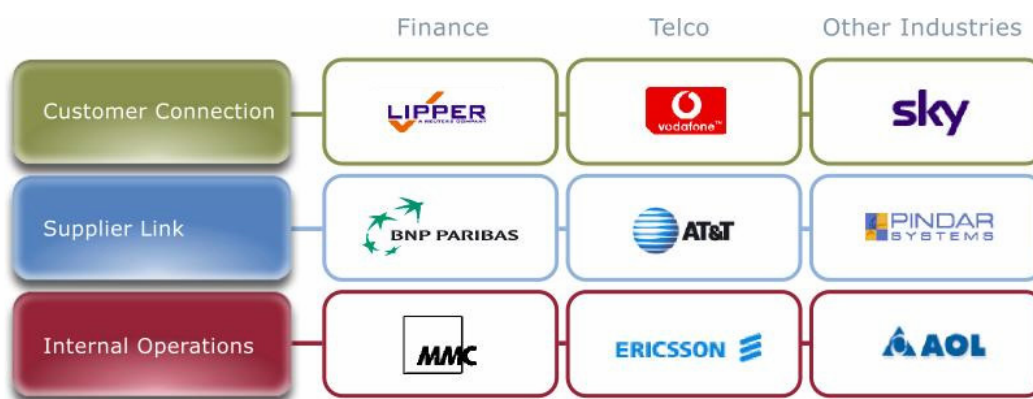


Figure 6: Some Cape Clear customer successes

Summary

ESB is a meaningful and rapidly emerging product category that changes the economics of integration. ESB enables the rapid introduction of SOA discipline to integration projects, which offers the potential for significant technical and economic benefits, but only if a proper, fully functional ESB is chosen. Cape Clear has repeatedly demonstrated that it has the technology, products, and solution skills to help you deliver a successful ESB project.

For more information, visit <http://www.capeclear.com/esb>.

Resources

Download a free trial of the Cape Clear Business Integration Suite at:

<http://www.capeclear.com/products/download/downloadenterprise.php>

"Principles of SOA Design" - This whitepaper discusses how Web Service design principles can be used to support a service-oriented architecture (SOA):

<http://www.capeclear.com/products/whitepapers/index.shtml>

"The Big Tour" - A look at how one company used Web Services to integrate their suppliers, customers, and internal systems:

<http://www.capeclear.com/products/tour/index.shtml>

"Clear Thinking" - Insights into the changing world of integration:

http://www.capeclear.com/clear_thinking.shtml

Cape Clear's Web Services Developer Community:

<http://www.capescience.com/>

Cape Clear Software Inc.

Web site: <http://www.capeclear.com>

U.S. toll-free: (888) 227 3439

Europe: +44 20 8899 6653

E-mail: sales@capeclear.com
