ADAPTERS IN THE INTEGRATION SPACE

Prepared By

Goldstone Technologies Limited

Table Of Contents

Introduction 3

Integration Architectures 3

I. Point to Point Integration 3

II. Process/Structural Integration Through Message Brokers 3

III. Integration Using Middleware 4

What are adapters? 4

Types of adapters 5

Feature Set of Adapters 6

Adapter Architecture 6

Adapters in different enterprise scenarios 8

I. EAI Scenario 8

II. B2B Scenario 9

III. Wireless Integration Scenario 10

IV. SCM Scenario 11

Intelligent Adapters 11

Conclusion 13

References 13

Introduction

Today's increasingly networked economy demands sharing a wide range of information between businesses. The onset of eBusiness has necessitated not only extensive integration within the enterprise but also real-time connecting with suppliers, partners and customers. According to DataQuest, future corporate wars will be fought not between the enterprises but between their respective supply-chains and demand-chains. Further, mergers and acquisitions have also led to a growing clamor for integration solutions that deliver quick and reliable integration over a wide variety of platforms and applications. Again, to a greater degree, success of mergers/acquisitions is going to depend upon the effective integrations of IT systems between the enterprises.

In its simplest form, application integration is the encapsulation of an existing application by a software component that acts as a functional interface to and from that application. The creation of an interface allows the other applications in the enterprise/extraprise to interoperate with the wrapped application, increasing its value and long-term usability. It involves a new category of software products that unifies and reuses the existing IT assets and facilitates their integration into a cohesive corporate system framework.

The new interface is variously called a wrapper, a connector, or an adapter. It allows predefined application events to be invoked across the interface. Those events will fetch or store objects and invoke processes native to the wrapped application. The beauty of this approach is that it transforms older legacy systems into components that can be integrated into a network of applications, enabling a distributed computing environment. For purposes of simplicity, the term "adapter" is used throughout this white paper. The fundamental concepts that enable this method of Business Integration are as follows:

- **Objectfying the application interface** — by doing so, we expose the application's internal business logic as a method. We also expose the application's data structures as an object model.

- **Messaging across the interface** — having objectified the interface, every interaction across the interface becomes an event (a message, in object-speak) that transports both data and control, inbound or outbound.

- **Defining a process** — this becomes a series of simple steps that invoke the necessary events on the appropriate components. The process definition and its rules are stored in a central repository, and process execution is controlled from this repository.

Integration Architectures

There are three broad types of architectures or methodologies. They are:

I. Point to Point Integration

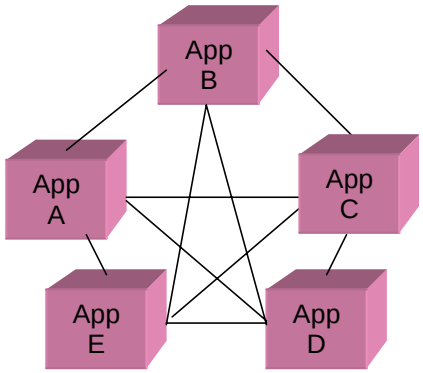

***Figure 1**:Point to Point Integration*

Here each application is connected uniquely to every other application. The disadvantages of this system make this integration approach suitable only for environments consisting of one to three applications. Features of this form of integration include:

- Point to Point Custom Code
- Multiple overlapping of clients, servers and data sources
- Highly customized workflow mechanisms.
- Costly implementation

II. Process/Structural Integration Through Message Brokers

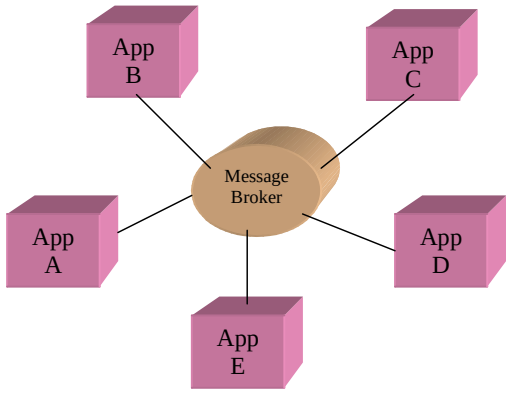

***Figure 2**:Integration Through Message Brokers*

This method of integration involves connecting applications through message brokers. Message broker is an intelligent intermediary that directs the flow of messages between applications, which are now the sources and consumers of information.
Features include:

- Server Centric
- Performance Bottleneck
- Poor scalability

III. Integration Using Middleware

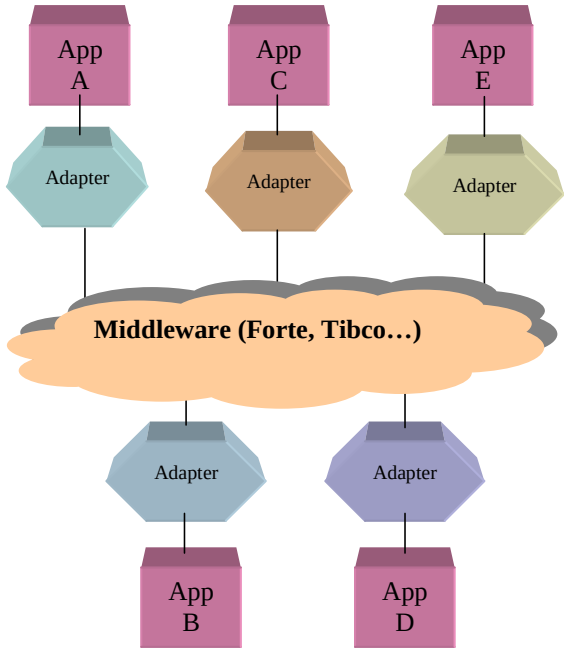

Figure 3: Integration Using Middleware

This method of integration involves connecting applications to middleware such as Forte, Tibco, etc, within which the complexity of integrating applications can be managed. This not only reduces cost, but also increases flexibility and ultimately, solves business problems.
Features of this form of integration include:

- Network-Centric
- Widespread Deployment
- Centrally Managed

What are adapters?

Adapters are pieces of software that are used in the integration of component-based applications. They serve as a "wrapper" that mediates access to an application that was not developed with integration in mind, including legacy applications. Adapters offer a "no-coding" approach to integration, providing simple, automated mapping of third party technologies and data formats. Adapters allow new components to be added to the system quickly and easily, without disturbing the existing infrastructure. Each new element only needs to be adapted to the active enterprise model, not to every other application, database and network with which it will interact.

Adapters connect application resources into the integration system so they can interoperate with other connected resources.
Adapters provide services for application integration between heterogeneous environments and resources. Services like:

- Transformation Services – data syntax resolution and validation
- Format Services – schema and semantics of messages
- Interaction Services – such as publish/subscribe, request/reply, send/receive, conversational. Etc
- Connectivity Services – such as, but not limited to TCP, HTTP, SOAP, CORBA, and SNA.

Adapters provide a transformation layer between an application and the integration middleware. Adapters enable packaged applications, databases and networking technologies to become active participants in the enterprise information flow, regardless of their different data formats or models.

Adapters unify complex applications and data using flexible, reusable and modular components over multiple enterprises to support changing e-business initiatives. They ensure minimal impact on network performance with efficient information distribution to multiple endpoints and reduce cost-of-ownership through general n-way rather than point-to-point integration. Adapters provide high-level security and unlock corporate database assets for active sharing with other enterprise resources.

Types of adapters

Static

- A static adapter is one that has been predefined and provides basic integration by establishing connectivity to the target source
- There is very little, if any, data transformation, validation, or filtering

Intelligent

- Implements data manipulation, validation and business rules processing, and it is aware of application metadata
- Provides performance improvements by moving business rule processing from the centralized integration broker to the distributed application adapter, thus reducing network traffic
- Reduces the amount of custom coding and application expertise required

Dynamic

- Actually learns from performing its data manipulation and can change its behavior by detecting changes in application metadata, thereby learning from operational information.

Adapters can also be differentiated based on the type of connectivity that they provide. The various types of connectivity include:

1. Component-level connectivity (interoperability between objects, components, applications).
This usually implies an underlying broker to connect the components. It requires an understanding of the underlying object model of the invoked component. When using this type of adapter, the user must map one data model onto another data model. This sort of adapter is the hardest to build and requires a standard to define the interface. For example, CORBA 2.0 forces the developer to use specific Interface Definition Language (IDL) commands to communicate to an object.

2. API-level connectivity.
This is a predefined window into an application that allows the invoking program to call prepackaged transactions, procedures, or methods. The invoking component must understand the semantics of the application being called.

3. Technology connectivity.
This invokes a low-level technology to pass data. For example, one might use MQSeries to send a message to an application or use SQL directly on a database.

Adapters give access to an encapsulated information resource. The broad range of information resources includes simple interfaces, databases, and predefined transactions of a system. Adapters enable resources to work with events — a formatted message exchanged between the resource and the requestor. An event can trigger some behavior from the resource and capture its response. Additionally, a resource could publish a response via the adapter to broadcast a message to the subscribers of the resource.

The categories of adapters provided by most vendors include:

- **Network Adapters** — enable integration with component or object development models as well as with other messaging technologies. Applications can be integrated based on network technologies into an e-business infrastructure that can expand their functionality to include messaging features, built-in fault tolerance and load balancing.

- **Database Adapters** — these send events that translate into SQL or the query language of the database. The response is the result of the execution of the query on the database. A database adapter can also invoke a remote procedure call (RPC) (on a DBMS) or some procedure in the native format of the database.

- **Interface Emulators** — this broad category includes a variety of green screen emulators, screen scrapers, and terminal emulation. These adapters hide the complexities of acquiring data from nongraphic screens. They transform key entry action and screen display response into events.

- **Middleware Adapters** — these create, receive, and translate event data to/from various middleware software (e.g., CICS transactions, MQSeries message, MTS request, and CORBA object request).

- **Enterprise System Adapters** — these allow connectivity with known enterprise systems, such as SAP, BAAN, PeopleSoft, Siebel, and other popular ERP and CRM packages. The adapter is aware of the semantics of the target system. For instance, an SAP adapter will generate native mode that will be meaningful to SAP.

- **Language Adapters** — these adapters allow the requestor to use the target language of the application to invoke a procedure or acquire some data. A "C" adapter will recognize C data structures and will bind the request to the target environment — for example, converting a database record into an HTML string. By extension, language adapters include translators for EDI (SWIFT, HL7, and ACCORD), HTML, and XML.

Feature Set of Adapters

Ideally adapters should support the following features:

- Session management
- Transaction Management
- Access control mechanism (User name / Password)
- Error handling
- Exception management
- Help manage the changing metadata and detect changes within an application.
  One of the problems with integrating enterprise applications over the long term is
  the management of application metadata. Every time an application comes up
  with a new version, the adaptor should not be recoded.
- Should be capable of
  - Event routing
  - Queuing
  - Business rule definition
- Graphical user interface for ease of maintaining integration platforms
- Provide an event trigger and be able to aggregate and manage multiple transformations at once
- Ability to manage adapter metadata in a more automated fashion
- IT departments will need adapters that can perform their own software updates and handle node-level data transformation in order to take the load off the integration broker and the network
- Scalability: Adapters need to support Node-level filtering and transformation of data, so that large amounts of raw data do not have to be shipped over the network to the integration broker but, rather, can be handled out at the nodes, where the source application resides. Adapters should have the ability to handle data transformation, reducing demand on the integration broker. Having data transformation performed in the adapter reduces the demand on network bandwidth and increases the scalability of the integration framework.

In addition to the above stated general features, each adapter may contain unique features. For example, e-business adapters do have some special requirements particular to the domain into which they are connected - for instance, they need to be able to handle common EDI (or XML-EDI) protocols.

Adapter Architecture

The figure shown below describes generic adapter architecture.

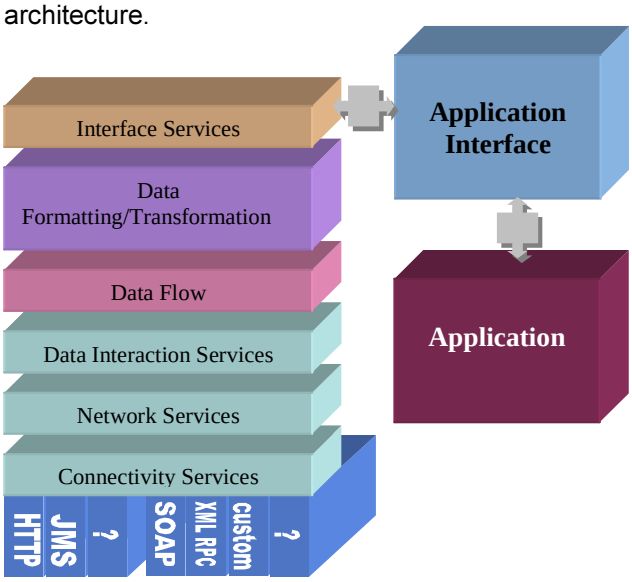

Figure 4: *Generic Logical Adapter Architecture*

Adapters provide complete application integration between heterogeneous environments and resources through a layered architecture, which generally consists of :

- Connectivity services – such as TCP, HTTP, SOAP, CORBA and SNA.
  These services are provided by a collection of software components. Adapters connect application resources into the integration system so they can interoperate with other connected resources
- Network Services – such as queuing, multiplexing, ordering, routing, security, compression and recovery
- Interaction Services – such as publish/subscribe, request/reply, send/receive, conversational, etc
- Data flow services – work and process flow flexibility to reflect business processes
- Format Services – Schema and semantics of messages
- Transformation services – data syntax resolution and validation
- Interface Services – reconciliation and translation of interfaces

However, the approach now a days is towards custom adapters due to the wide range of middleware and proprietary adapter development platforms. Most Enterprise Integration System vendors and application server vendors use vendor-specific architectures to provide connectivity between application servers and enterprise information systems.

With web services, any application can be integrated so long as it is Internet-enabled. The foundation of web services is XML messaging over standard web protocols such as HTTP. This is a very lightweight communication mechanism that any programming language, middleware, or platform can participate in, easing interoperability greatly. These industry standards enjoy widespread industry acceptance, making them very low-risk technologies for corporations to adopt. With web services, you can integrate two businesses, departments, or applications quickly and cost-effectively.

An adapter generally consists of at least two distinct program modules, the Server Module and the Configuration Processing Module.

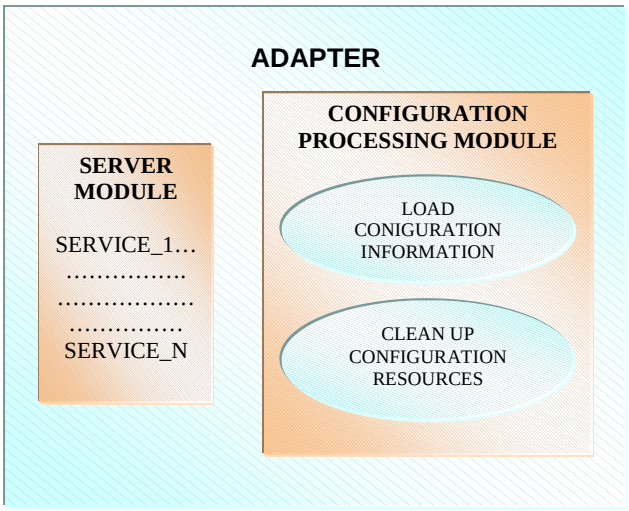

Figure 5: *Standard Adapter Design*

The Server Module

The Server Module contains the code for the functions that perform the services advertised by the server. These functions are called when the server is booted and at server shutdown time. The server side function checks the license with the inbound function in the Adapter Development Kit (ADK) library, opens the messaging catalog, retrieves the name of the configuration file, reads the trace level from the configuration file and Calls functions from the Configuration Processing Module to make use of ADK API functions for parsing, processing, and storing the configuration data.

The Configuration Processing Module

Parsing and processing of the configuration file is done by the loadConfigurationInformation and cleanupConfigurationResources functions in the Configuration Processing Module. These functions make use of the ADK API functions for parsing, processing, and storing the configuration data.

Nowadays a new type of Architecture is being developed and used by some Integration companies, namely, to provide improved performance by moving network protocol processing from the server CPU to an ASIC chip called an Internet Protocol Processor (IPP). The IPP resides on the server adapter. The concept of offloading applications from a server CPU to an external card is not new. Common examples include adapters for disk processing and graphics adapters for graphics acceleration.
This technology is called Session-Layer Interface Card (SLIC technology). The figure below shows the protocol processing functionality provided by SLIC hardware with Windows NT. The new architecture provides a fast path for many network transactions, offloading protocol processing from the host. Moving protocol processing to the server adapter dramatically improves adapter and server performance. TCP/IP management and control remain on the host. These adapters are deployed in the same manner as legacy adapters. This type of arrangement may be preferred for heavy-duty integration scenarios.

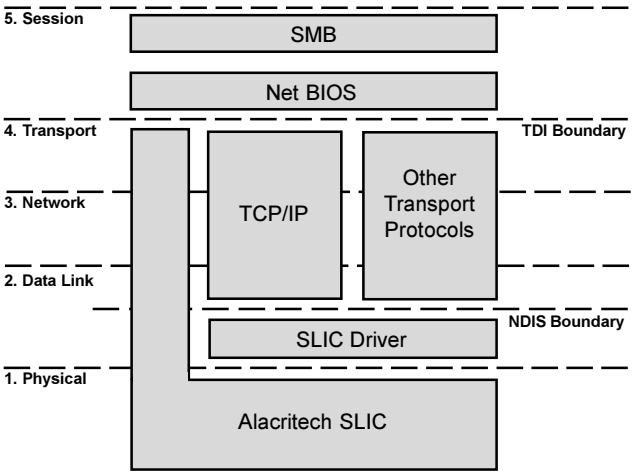

Figure 6: *Protocol stack with SLIC technology*
Source: Alacritech, Inc.

SLIC technology is 100% compatible with IEEE Ethernet standards, including legacy network adapters and other existing LAN infrastructure, PC hardware and software, switches, hubs and routers. These adapters do not use proprietary schemes such as jumbo frames, which are incompatible with existing standards and equipment. Driver updates will allow network managers to update IPP firmware to add new functionality while ensuring compatibility with future changes to standards.

Adapters in different enterprise scenarios

I. EAI Scenario

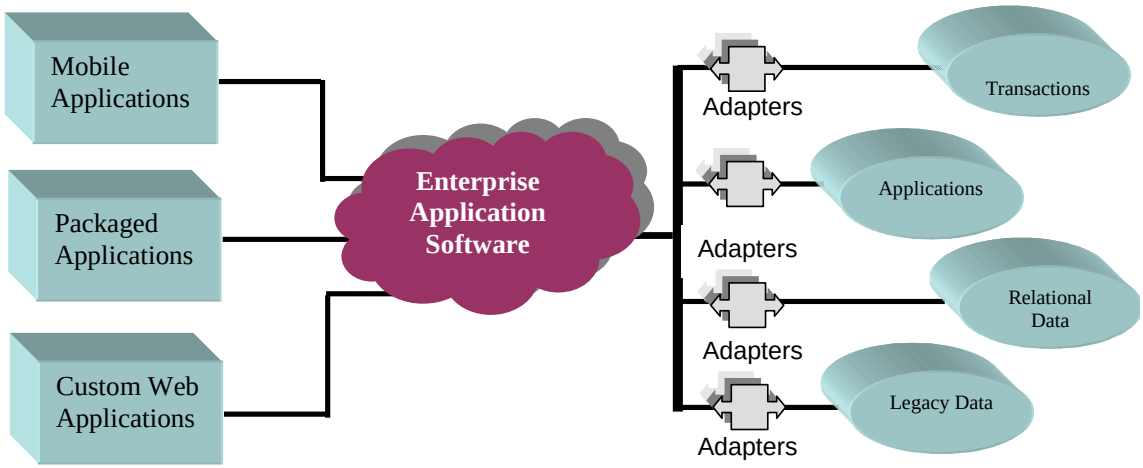

Figure 7: *Adapters fitting in the EAI Scenario*

Enterprise Applications Integration (EAI) connects existing and new systems to enable collaborative operation within an entire organization. EAI provides the ability to specify business rules, which define allowable exceptions and variances for instance based on the content of a purchase requisition or a purchase order, and the profile of the customer. These rules specify information to be communicated to the ERP and custom applications such as processing options, priority decisions, and quality of service requests, as well as the desired acknowledgement method to be used for communicating to the customer. The EAI system typically provides a scripting language to specify the rule as well as the method required to communicate to the application in its own language.

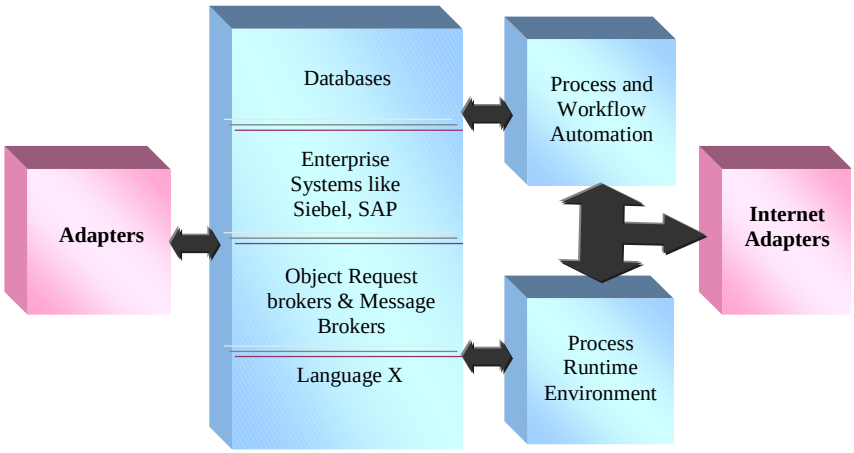

Figure 8: *EAI component architecture*

Adapters provide connectivity to the information resources of the system in the EAI component architecture.

Adapters recognize the different APIs and interfaces presented by various applications and data sources and provide services that are aimed at easing the task of integrating and extending an existing application infrastructure converting it from a custom interface to an open reusable interface.

The EAI component architecture sits on top of the distributed layer, the messaging layer, and the transport layer. Communication with the lower layers can be integrated into the EAI product or can be achieved via an adapter.

To facilitate and automate the development of business rules and applications associated with them, adapters provide the models, semantics and infrastructure to achieve an easy-to-configure communication environment among loosely coupled applications and data, thus leading to the clear industry goal of integrating disparate applications and data sources within and external to an application.

II. B2B Scenario

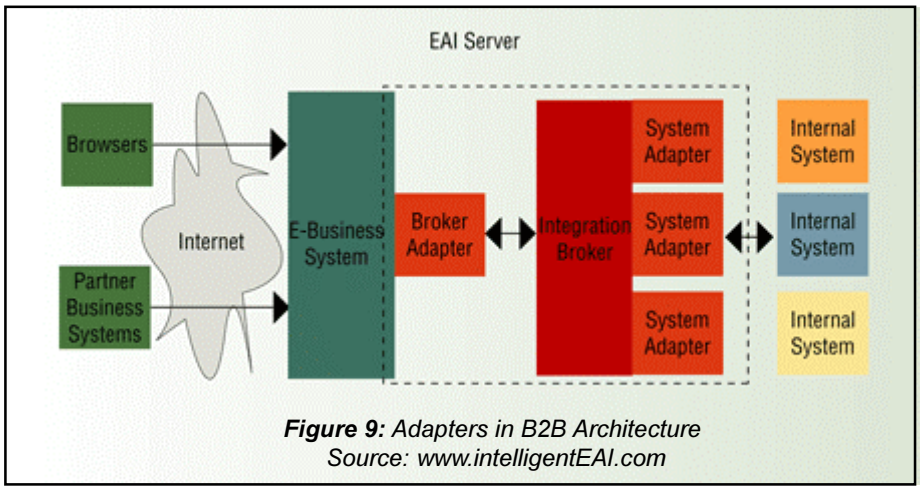

Figure 9: *Adapters in B2B Architecture*
Source: www.intelligentEAI.com

Business-to-Business (B2B) Integration connects applications and business processes from customers, suppliers, and partners across the Internet. When sharing information between businesses, integration problems are compounded. Each enterprise has its own set of applications, technologies and standards. One enterprise rarely has any say over the specific standards and technologies another chooses to implement.

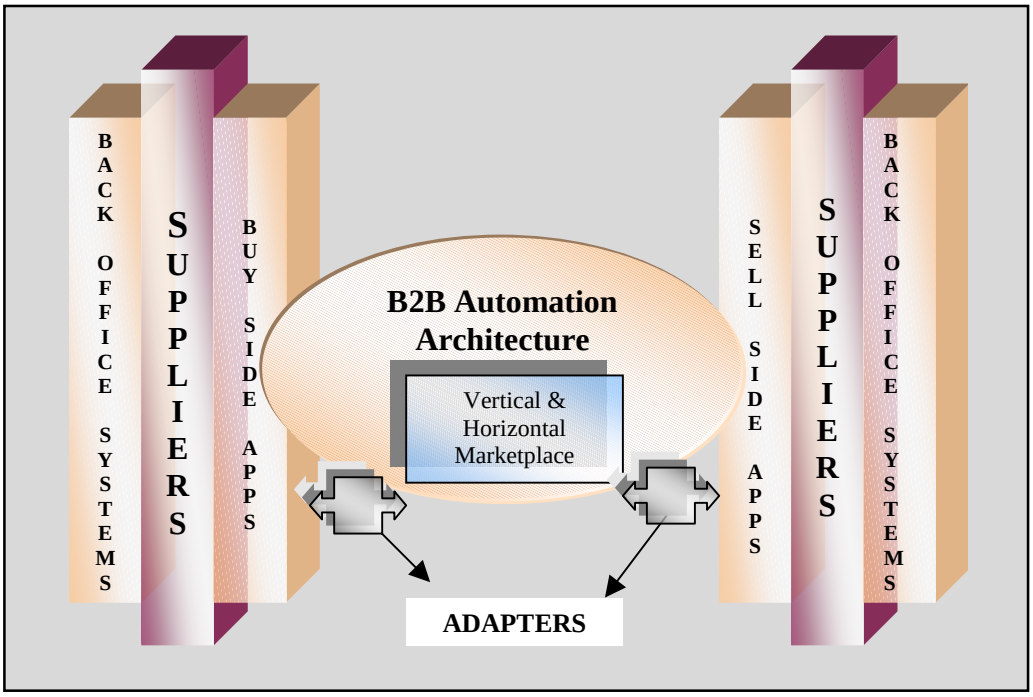

Figure 10: Adapters Fitting In B2B Scenario

B2B Integration technologies enable four ways for interactions to occur with an application: Web/client access, data exchange, direct-application interaction, and process sharing. B2B Integration technologies also provide a variety of methods for communicating over the Internet so that application-to-application communication can be supported no matter what the native environment of any one application happens to be. These forms of communication include TCP/IP, HTTP, SMTP, HTML, EDI, MIME, XML, CORBA, COM, Java and SOAP.B2B Integration also handles data transformation from internal representations to external representations, and it maps information between applications on both syntactic and semantic levels. To coordinate process across company boundaries, B2B Integration uses technologies such as distributed transaction processing, workflow, publish/subscribe message brokers, and distributed object brokering.

III. Wireless Integration Scenario

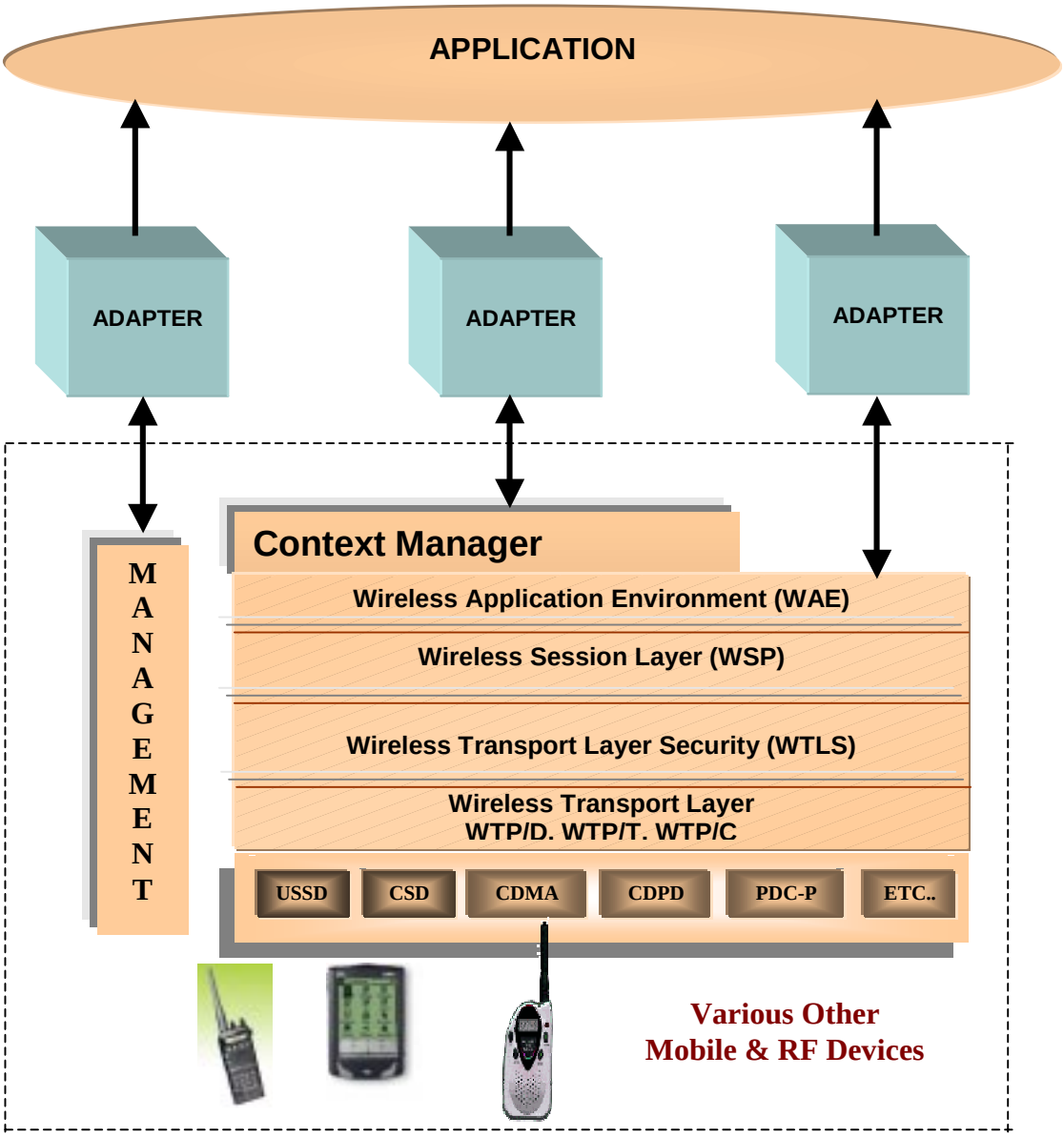

Figure 11: *Adapters Fitting In Wireless Scenario*

With Wireless Application Integration emerging as one of the waves of IT innovation, organizations have been leveraging the web technologies for managing and developing enterprise information systems and collaborative computing environments. The new investment space in WAI is an aggregate blend of wireless technologies and enterprise integration services. The concept of WAI intends to enable access to relevant information, save user's time, increase user productivity and to enable faster decision-making.

Adapters provide vital support for Application-to-Application wireless messaging. Adapters give the desktop computer users wireless data exchange directly from mobile computers, PDAs, remote controls and other high power RF devices used in communication systems supporting worldwide communication markets. They also provide wireless data transfer for record synchronization, file back up and network access, directly from PCs and PDAs. Also, adapters let you avoid writing custom-coded integration components for enterprise information assets, saving time, money, and effort on the first application.

IV. SCM Scenario

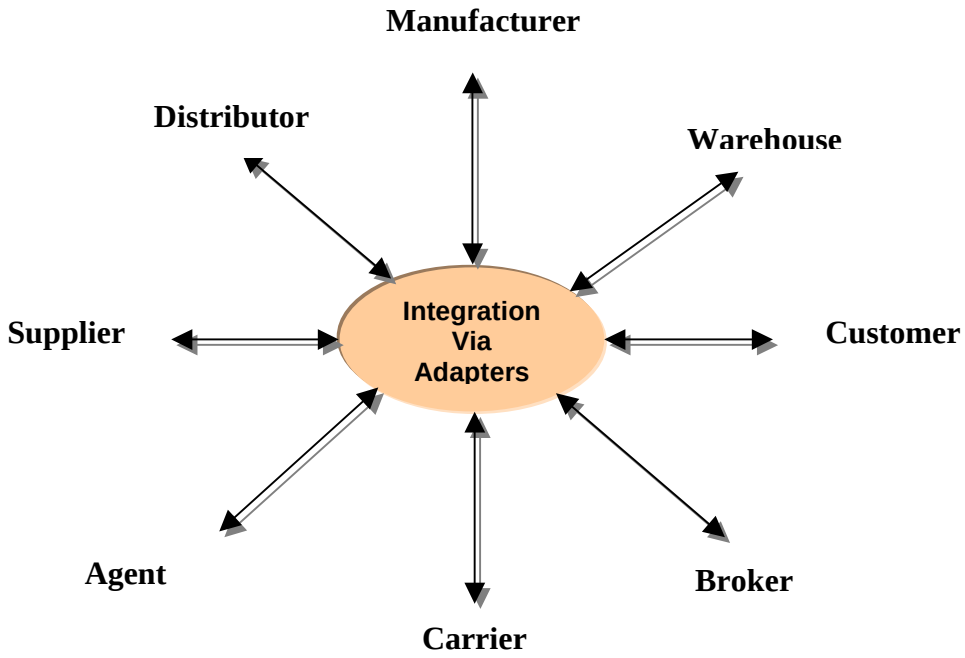

Figure 12: *Supply Chain Collaboration*

Today's networked economy requires that all members of the supply chain act as a part of one seamless fulfillment process. The Web-based supply chain is a responsive, fluid environment that transforms rapidly and requires not just managing multiple processes in an enterprise, but also managing multiple enterprises within one collaborative process.

In an enterprise scenario, excellence in fulfillment operations becomes a critical issue and requires the integration of the entire inter-enterprise supply chain. Adapters form a seamless part of the integration technology that allows companies to connect with every trading partner and service provider in the community cost-effectively and quickly thus providing an approach to connect multiple enterprises managed with one process to make up a trading community. Adapters provide flexibility in supply chain execution allowing applications to change and adapt as quickly as the business around them.

Intelligent Adapters

Standard "static" adapter technology simply links one application to another, or to an integration broker, in an un-intelligent, point-to-point connection. The main benefit of a static adapter is that the programmer doesn't have to work with all the different application program interfaces, since the adapter handles the basic translation behind the scenes. But that's about all such adapters do. They can't transform incompatible data between disparate applications or update themselves when changes occur in the applications.

Intelligent adapters are designed to handle many of the dynamic tasks required in application integration – tasks usually handled either in the integration broker or done by hand by a programmer. Intelligent adapters automate real-time integration by eliminating the need for application expertise, eliminating custom coding and reducing deployment time.

Static adapters provide standard off-the shelf libraries for application interfaces, which have to be modified to fit into the customer's environment. By being aware of metadata, intelligent adapters provide a large library of adapters for each application. As these adapters are aware of application metadata changes, this library need not be modified if the customer has made any modifications to the target system. Intelligent adapters provide ways to decrease the time, cost and expertise needed to install new applications or to modify/upgrade current systems.

Intelligent adapters use static connectors to obtain the input data from the source environment. The intelligent adapter performs data transformation, validation and reformatting and the output data is given to the target static connector to deliver the transformed data to the final target application in the integration suite.

Intelligent adapters for applications offer prebuilt transformation and load templates that contain schema and transformation information specific to the applications. They offer a robust graphical user interface (GUI) that allows users to capture the data/metadata and to represent relationships between source and target data and applications. Intelligent adapters are designed to meet new challenges of data and content integration introduced by the mobile internet.

A new genre of intelligent adapters called intelligent agent adapters has emerged to solve complex integration problems. Agent-adapters extend the concept of an application adapter beyond simple API abstractions to components with scalable facilities for complex application patterns. Intelligent agent adapters expand integration potential by delivering sophisticated resource management, localized caching and conversational mode processing and eliminate up to 80% of the custom coding usually required in integration projects.Agent adapters are a combination of agent containers and adapter components. The adapter components consist of the actual code for communicating with the enterprise application. The agent container handles services such as session management, error handling and exception management.

These agent adapters provide node-level filtering and transformation of data, so that large amounts of raw data do not have to be shipped over the network to the integration broker but, rather, can be handled out at the nodes, where the source application resides. While data transformation can occur elsewhere, such as in the target application or in the integration broker, having it performed in the adapter reduces the demand on network bandwidth and increases the scalability of the integration framework. Agent adapters sit at the outer edges of the integration solution - at the end points where the enterprise application resides – and can therefore filter data before passing it along to the integration broker at the hub. That prevents the hub from becoming a bottleneck. By virtue of node level filtering, these adapters don't just blindly accept data from the application. So, if there's any erroneous data, it is filtered out before it is put on the wire.

One of the problems with integrating enterprise applications over the long term is the management of application metadata. Every time an application comes up with a new version, the adapter may need to be modified, sometimes heavily or may need to be recoded all over again. Intelligent agent adapters help manage the changing metadata and detect changes within an application.

It is this capability, along with the ability to manage adapter metadata in a more automated fashion that will make intelligent adapter technology more critical to integration projects in the future. As organizations connect greater numbers of applications over internal and external networks, IT departments will need adapters that can perform their own software updates and handle node-level data transformation in order to take the load off the integration broker and the network.

Agent-adapters typically function as a coordinated federation linked together by the EAI framework. The integration framework consists of a loosely coupled set of services for integrating application data. Facilities such as security, transformation, validation, and routing are available to each agent-adapter. At startup, the agent-adapter would register for services of interest. A common repository service manages discrete information on each application and the corresponding agent-adapter components. The repository also contains the integration objects that encapsulate the relationship of process or information flow between individual agent-adapters.

Because agent-adapters can perform high-level functions such as content validation and transformation, the integration architect has the option to design a system that distributes processing across the individual agent-adapter nodes.

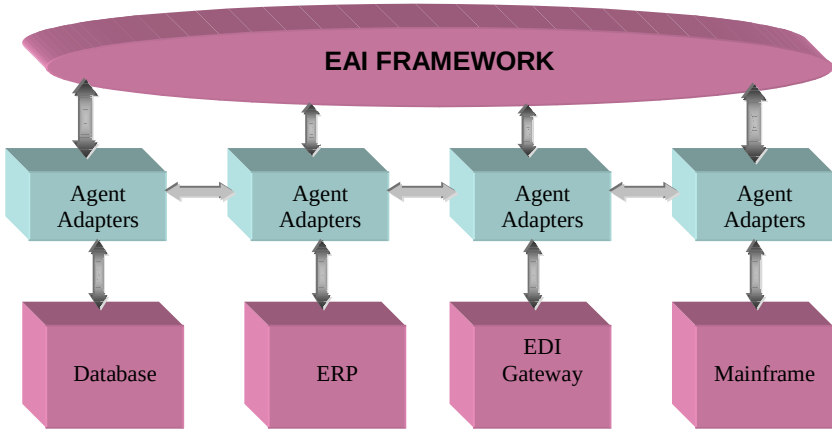

Figure 13:Using Agent Adapters

This distributed model will scale and provide necessary flexibility to integrate various application architectures. The essential advantage is efficiency. By applying qualifying rules and filters at the node level, it minimizes the need to place unqualified data "on the wire," only to have it resolved at a central point. If application events do not qualify, they are filtered out by the agent-adapter.

Conclusion

The increasingly collaborative nature of business in the global range today has necessitated not only extensive integration within the enterprise but also real-time connections to suppliers, partners and customers. Mergers and Acquisitions have come to depend, to a great extent, on the speed of integrating their respective diverse IT systems.

Adapters form a seamless part in the integration scenario. They decrease the need for application expertise, eliminate custom coding, reduce deployment time and promise to deliver cost-effective integration solutions for systems of all types including enterprise systems and e-business solutions.

The *White Paper* has attempted to explore the use of adapters to reduce the overall effort of integrating disparate systems. The paper highlights on the architecture and methodologies involved in adapters and also tries to feature the way adapters fit into various enterprise scenarios like EAI, B2B, SCM and Wireless Integration. The paper has made an attempt to discuss different kinds of adapters including intelligent adapters that automate real-time integration to meet new upcoming challenges in data and content integration.

References

1 www.eaijournal.com

2 www.iwaysoftware.com

3 www.ebizq.net

4 www.tibco.com

5 www.ittoolbox.com

6 www.intelligentEAI.com; Article: ” Shaking Hands”

7 www.sdmagazine.com; Article: “Agent Adapters: Beyond APIs”

8 Cutter Consortium, Article: ”Integrating The Enterprise”