Affordable Integration

s-integrator Service Broker

White Paper

Version 1.2

September, 2001

TABLE OF CONTENTS

1. AFFORDABLE, RAPID INTEGRATION

1.1. Introduction

This paper describes the critical cost, risk and technical issues typically encountered in development projects that require integration. The s-integrator product from Indigo Technology Partners is presented as the optimal solution to these concerns. It introduces Service Broker technologies, the architecture that they implement and how this architecture provides the foundation for an *adaptable enterprise architecture*[1] that exhibits low cost and low risk. The resulting Integration and Software Services Infrastructure delivers *rapid value creation*.

1.1.1. Cost

Current integration solutions are very expensive and are long overdue in becoming competitive and cost effective. This expense is accumulated through the acquisition of consulting services, large integration suites and least of all hardware. And since most existing business logic still resides on the mainframe, the cost of integration can be even higher. High risk and a long time to market commonly accompany this expense.

1.1.2. Integration Server Technology

Integration products usually implement hub and spoke architectures that favor large, centralized integration servers or message brokers. There are a few integration solutions that use bus architectures that favor distributed architectures and provide higher performance. It is valuable for an integration server to accommodate centralized and distributed architectures while not carrying the significant expense associated with current integration solutions that puts budgets and projects at risk.

Web Services[3] technologies provide a new paradigm for integration and are therefore important to be supported architecturally as they mature. Integration and Application Server vendors are attaching these Web Services technologies on top of their current products to capitalize on their marketing value. However, Web Services technologies realize their benefit from using the Software Service as the unit of work. Thus incorporating Web Services technologies requires a fundamentally different architecture than current integration servers have evolved into. The Service Broker is a next generation integration server that realizes the Software Service, not a message, as its core unit of value. A Service Broker implements an architecture that is built from the ground up as a Software Services Platform to fully capitalize on the business value that Web Services technologies provide now and in the future.

Therefore the ideal integration product would be a Service Broker that accommodates centralized and distributed architectures, is low cost and low risk.

1.1.3. Project Integration

Most individual projects cannot wait for an enterprise-wide solution to be in place before they can deliver the business value that project objectives demand. These enterprise integration projects can turn into "big bang" projects that have long timelines. Therefore more cost-effective integration solutions are needed that address the tactical integration needs of projects.

1.1.4. Enterprise-Wide Integration

Modern development methodologies have shown that an incremental, phased approach is often more prudent by limiting risk. The first phase of an enterprise-wide integration project should minimize risk and expense in order to prove the concept, adjust development processes, and accommodate organizational issues and new technologies. This can be achieved through the use of a low cost, low risk integration product.

1.1.5. The Solution

These issues have created a market for simple, scalable, low cost Service Brokers that provide Integration Server Functionality and a Software Services Platform. The ideal solution accommodates both hub and spoke architectures and distributed architectures. It dramatically reduces software cost, facilitates distributed architectures, reduces risk significantly and supports Web Services technologies architecturally. s-integrator is such a solution.

1.2. The Business Need

1.2.1. Project Integration

EAI solutions are a fact of life in e-Business. Unfortunately, most development projects that require such a solution cannot take on the additional cost, time, resources and risk necessary to expand the project scope to include an enterprise level integration server or message broker. Instead, these projects spend considerable time and money creating a throw away integration solution that is difficult to integrate into a larger integration solution and difficult to support. What these types of projects need is an affordable, solid, Integration Server and Software Services Platform that does not force them to get approval for a greatly expanded budget, while easily integrating into a future, enterprise-wide integration system. After all, the project's business case was approved to deliver business value, not an integration server. You do not want your project to become a company-wide integration project. Enterprise level EAI solutions involve rare and expensive consultants, usually costing much more than the already expensive software itself. EAI software packages cost[2] an average of $500,000, while the final cost can easily exceed a $1 million with consulting fees. And developing your own throw away integration solution can take experienced integration developers 6 months to a year at $100,000 to $300,000 to create and test, if all goes smoothly.

s-integrator is a proven, adaptable integration solution costing $10,000 to $60,000.

1.2.2. IT Budgets, The Economy and Mergers & Acquisitions

IT Budgets

As IT budgets shrink and investment decisions become more conservative, affordable integration servers and service brokers will become more attractive. They provide the means to rapidly build business value and open up possibilities for distributed architectures while providing a foundation on which to build strategic systems. This type of solution allows a company to implement a repeatable process for creating an adaptable business and incremental value without sacrificing the future. Costs are reduced from over $1,000,000 to far less than $100,000.

The Economy - Earnings Warnings

As companies continue to post earnings warnings, the stability of new technology companies comes into question. Making large investments in software from young vendors can carry significant risk including source code availability should the vendor go out of business, staff training in obsolete technology and expensive consultants needed for large integration suites. This risk can be managed by

- Using multiple integration vendors
- Using solutions that integrate easily with each other
- Adopting solutions with a small investment, small risk and big payback.

Mergers and Acquisitions

Should the company that you obtained your integration solution from be purchased by a competitor, your company should have a contingency plan. Again, multiple integration solutions and vendors, especially inexpensive ones, can ease this risk.

1.3. Business Scenarios

1.3.1. Distributed Integration Servers and Service Brokers

Some companies run their own software at many locations because it integrates on-site systems and speeds the performance of local software components. Building and maintaining an Integration and Software Services solution in these distributed locations is rarely the business the company is in. Most solutions were developed and priced as large, expensive hub servers and do not address this type of distributed architecture or cost structure. Purchasing an affordable and flexible replacement allows the company to focus on their business.

1.3.2. Tactical Integration Solutions

Companies that realize the value in an enterprise-wide integration platform can test the waters without jumping into a "big bang" integration project that demands extensive up front R&D and significant time. The server software is only part of the risk in such a project. Your systems, their interfaces, your organizational issues, resources, staff, etc. are a major part of such an undertaking at the time of development and during the

operational lifetime of the software and processes supporting it. Taking an existing project as the test bed for a proof of concept is a great start. It is important that the concept being proved is not a product, but the reality of having such an integration solution and supporting it in your organization. It is recommended that this be done with a much more manageable investment in parallel with the research and development that facilitates the larger strategy. Plan strategically and act tactically.

1.3.3. Reducing Project Cost

Systems Integrators take on projects with similar characteristics to those already mentioned. They too need an affordable integration server that provides the required functionality, while bypassing the extensive research, decision, approval and cost of using or building a large integration server or message broker. The savings of this approach multiplies quickly when you consider the number of licenses required for a project multiplied by each CPU on each platform: Development, Test, Production and potentially others. Software services are faster to develop and can be implemented by any level of programmer.

1.4. The Solution

1.4.1. s-integrator

s-integrator is an affordable Integration Server that uses Software Services as its unit of work. It can run these Software Services on the local machine or broker a remote call to a separate machine through a Server Adapter. s-integrator is a sophisticated integration solution that handles complex integration applications where custom protocols and Service Flows must be accommodated. The s-integrator architecture is also a Web Service Broker for exposing existing software, residing on a component host or application server, as Web Services. The Software Services Stack is the foundation for the product (See Figure 1.).

s-integrator can be used to inexpensively expose all internal Software Services (APPC, JDBC, etc.) and Software Components (DCOM, RMI, CORBA) as Web Services. These products empower developers to create scalable Software Services Infrastructures and Applications (See Figure 2.). Most of today's integration solutions are oriented towards business users, but s-integrator is not. Software Services created with s-integrator can participate in Service Flows and can be used from process management or workflow products.

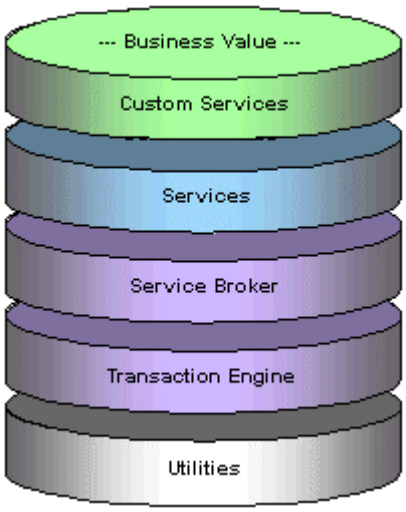

Figure 1. Software Services Stack

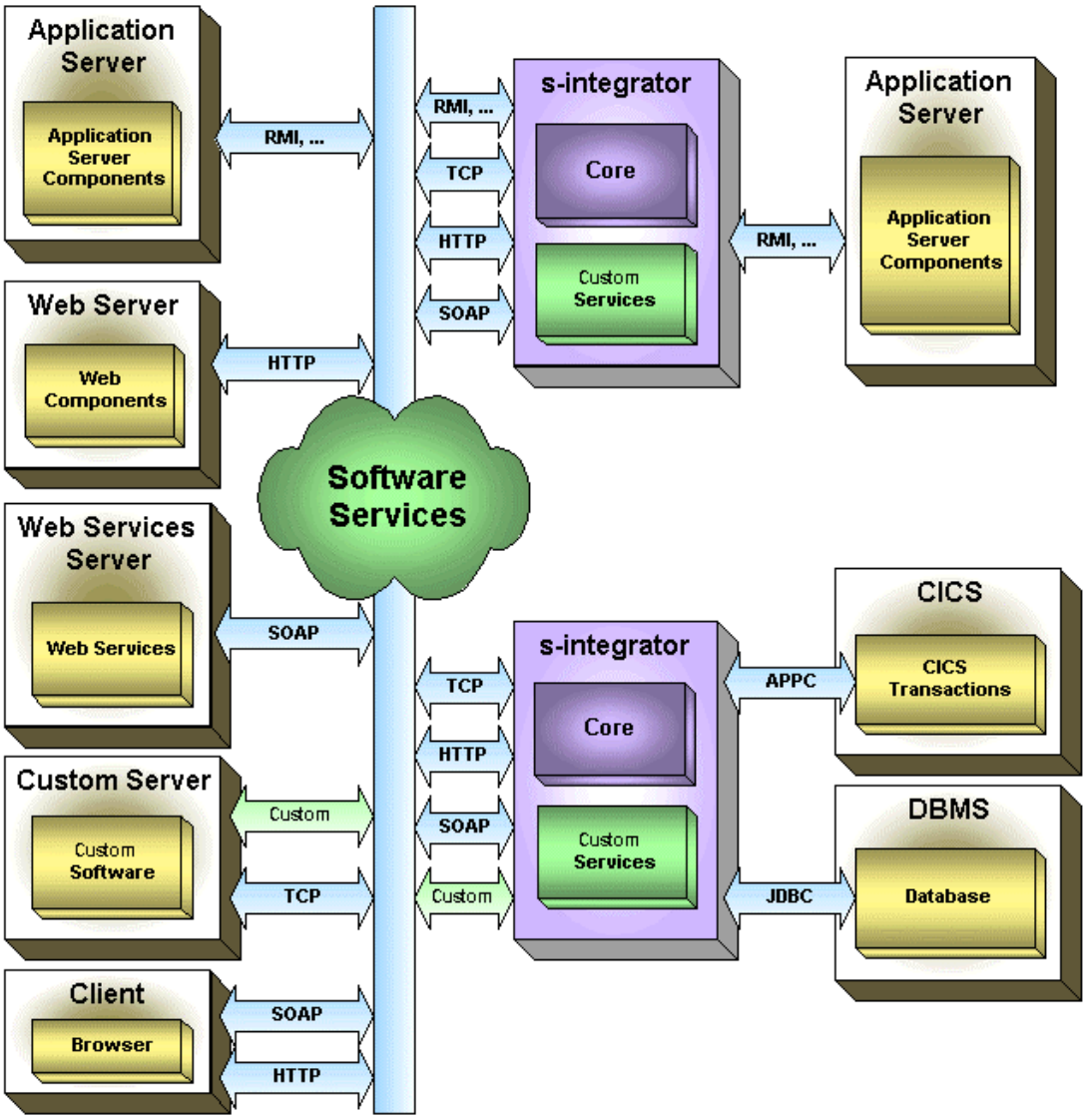

Figure 2. The s-integrator Distributed Architecture

1.4.2. *s-integrator* - A Service Broker

The issues raised in this paper have created significant demand for an affordable Integration Server and Software Services Platform that has all of the important features that one expects from an integration server:

- Transaction Engine
- Routing and Transformation
- "Any" to "Any" Extensible Adapter Framework
- Built-in Adapters for Current & New Technologies

- Software Services Platform including Web Services

s-integrator is a service-integrator that is the synthesis of an Integration Server and a Software Services Platform. It includes a proven Multi-threaded Transaction Engine, Transformation and Routing with an Adapter and Software Services Framework to empower developers to create "Software Services" that deliver immediate value (See Figure 3.). s-integrator is only a fraction of the cost of a large EAI solution, while it reduces risk substantially and accommodates future EAI solutions.

Software Services implement pre- and post-processing for business components, but they can also represent business components themselves, when necessary, to optimize efficiency and performance. This is especially applicable in more distributed architectures. A Software Service is essentially custom code that a developer creates to process a request. Transforms are custom code that a developer creates to convert data between formats and a Service Flow binds a sequence of Software Services and Transforms into a sequential process or flow.

Custom protocol adapters can be developed to receive and/or send requests and responses over additional communications and application protocols.

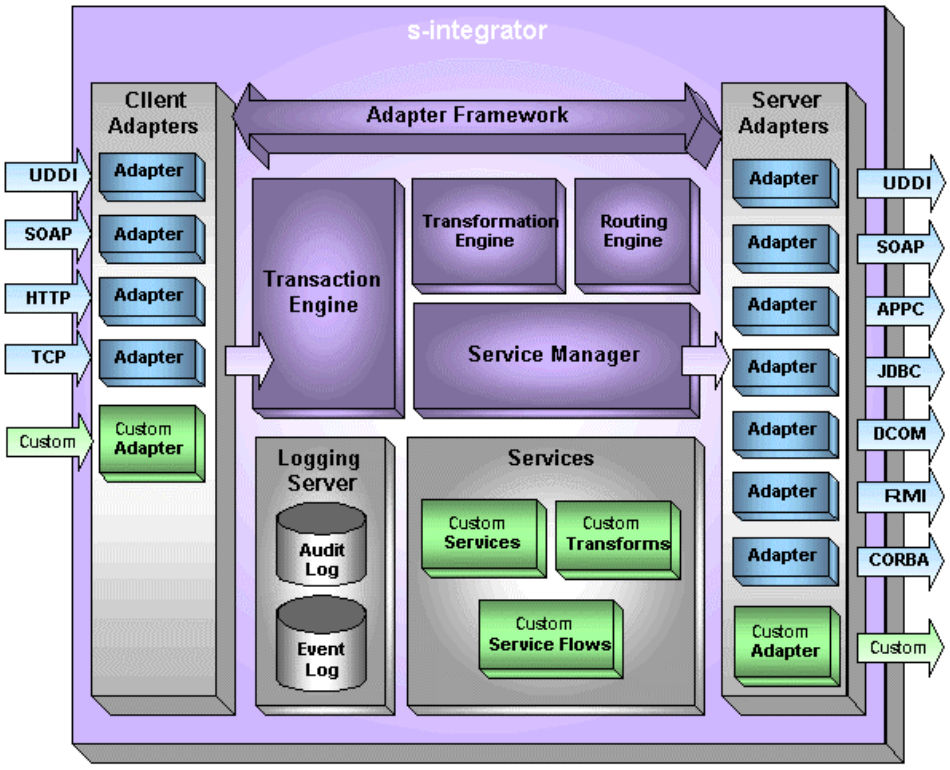

Figure 3. The s-integrator Architecture

1.4.3. s-integrator Features

1.4.3.1. Client Adapters

Currently available:

- TCP
- HTTP
- SOAP
- WSDL
- XML
- APPC

Under Development:

- HTTPS
- UDDI
- FILE

Custom Client Adapters

1.4.3.2. Server Adapters

Currently available:

- APPC
- JDBC

Under Development:

- TCP
- HTTP
- HTTPS
- SOAP
- XML
- MQSeries, MSMQ
- UDDI
- SMTP
- RMI, CORBA, DCOM
- FILE
- PEM

Custom Server Adapters

1.4.3.3. Software Services Framework

- Custom Services
- Custom Transforms
- Custom Service Flows

1.4.4. A Web Service Broker for Distributed Architectures

s-integrator will soon be able to be deployed as an autonomous, smart, service agent that encapsulates and markets Software Services as global e-business value. In this form, it will create a Software Services Storefront that exposes existing software so it can participate in Global Service Markets (for web connected systems). It will wrap existing software into valuable distributed application agents that can be rapidly combined into highly efficient distributed architectures and easily syndicated service markets.

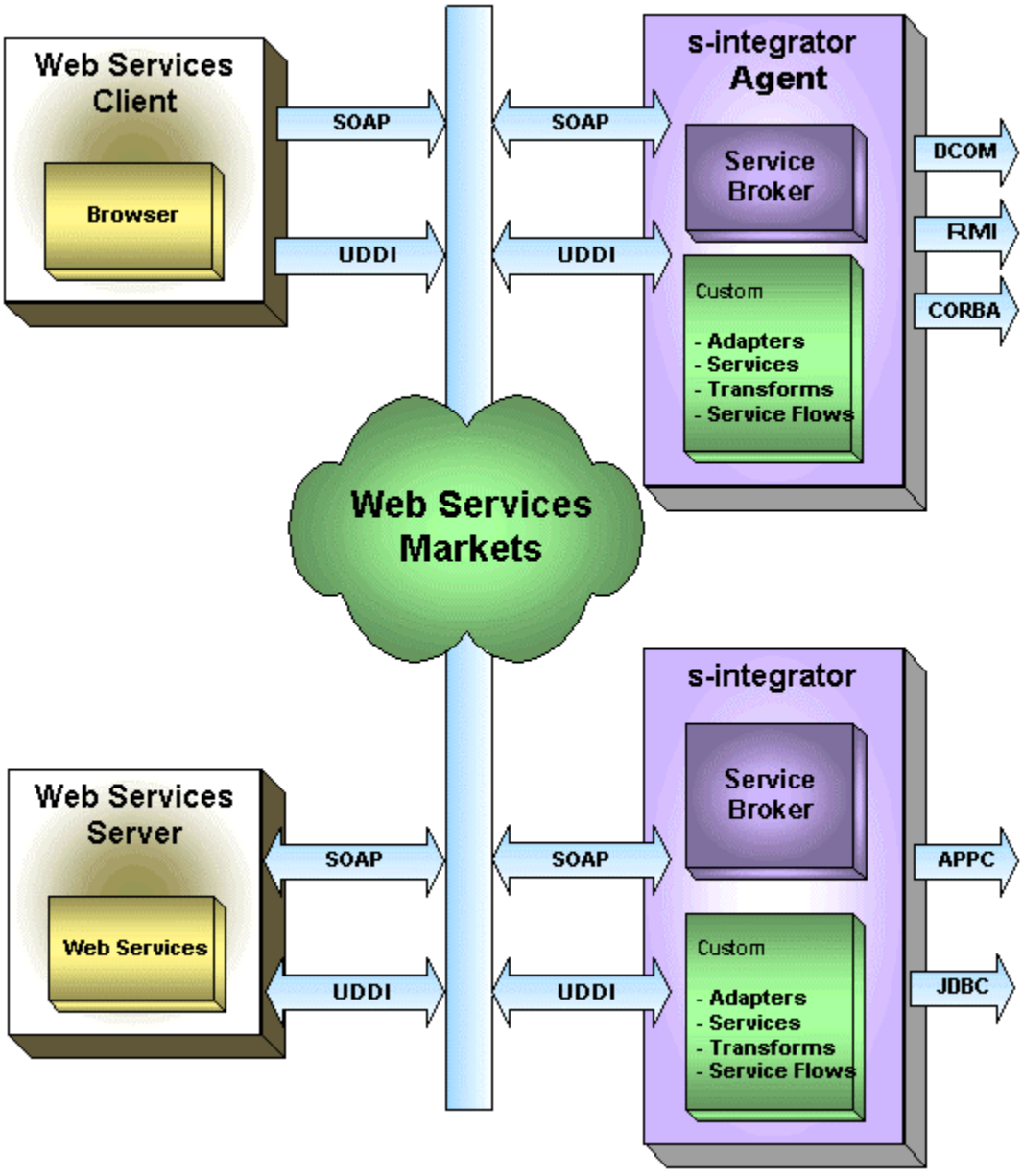

Figure 4. The s-integrator Agent Architecture

The s-integrator Agent implementation will have the following features (See Figure 4.):

- Custom Web Services
- SOAP, WSDL and UDDI Support
- RMI, CORBA and COM/DCOM Support

1.5. The Bottom Line

s-integrator is a highly affordable Integration Server and Software Services Platform that comes with frequently needed adapters including a proven, high speed LU6.2 adapter for mainframe application integration and Web Services integration. The cost is an order of magnitude less than other solutions. Combining Web Services technologies into the equation further reduces investment; opens up the potential for more fully distributed architectures and significantly reduce project risk. This solution does not attempt to remove developers from the equation, but empower them.

One of the top three credit card companies uses s-integrator for its domestic card activation process. It provides integration between external vendor systems and internal mainframe applications while hosting a small amount of critical business logic. This enabled the project to reduce mainframe development, rapidly integrate systems, and focus on the non-integration, business issues of the project.

These solutions are for businesses and systems integrators that need to reduce project risk by purchasing a solid Integration Server and Software Services Platform within their project budget, while providing easily integration with enterprise-wide integration solutions. Companies are seeking rapid integration, lower project costs and increased profits are prime beneficiaries of these solutions.

s-Integrator pricing reflects the desire to remove price as a major obstacle to rapid service integration. Prices start at $10,000 per machine with unlimited CPUs.

1.6. Contacts

s-integrator is available now from:

Indigo Technology Partners, Inc.
Skillman, NJ
Email: info@indigotp.com
Company WebSite: http://www.indigotp.com
Product Web Site: http://www.s-integrator.com.

2. APPENDIX

2.1. References

[1] "TRIMing During Economic Downturns", by Al Passori, META Group Research, April 2001.

[2] "Value of EAI Grows As Integration Needs Expand", by Antone Gonsalves, Information Week, May 2001.

[3] "The Service Broker", by Billy Newport, TheServerSide.com, July 2001.

[4] "Important Growth in Software Application Infrastructure and Architecture", by Mike Gilpin, Giga Information Group, July 2001.

2.2. Glossary of Terms

s-integrator - s-integrator is an affordable and customizable Service Broker that provides "Any" to "Any" integration and exposes software components as Software Services and Web Services.

s-integrator Agent - The s-integrator Agent architecture. A cost-effective Web Service Broker that acts as an agent for software components, exposing them as Web Services. It makes DCOM, CORBA and RMI accessible components available as Web Services, ready for execution via SOAP and listed in UDDI directories.

Service Broker - A server that brokers "Software Services". A Software Service is often simply a request for a transaction, but it can represent a business function or a Service Flow. A Web Service is one type of Software Service.

Service Flow - A Software Service implementing a sequence of other Software Services and Transformations as steps in a process. Service Flows combines existing Software Services and other Service Flows to build automated processes.

Software Service - A set of software operations that provide business value either directly itself, or indirectly by invoking a separate software component or system. These services can be exposed through any protocol that an adapter is present for such as TCP, HTTP and Web Services technologies (such as SOAP, WSDL and UDDI).

Transform - A special Software Service that converts and transforms data for reuse by subsequent Software Services in a Service Flow.

Web Service - A representation of software that provides business value either directly itself or indirectly by invoking a separate software component or system. Web Services are exposed through Web Services technologies such as SOAP, WSDL and UDDI.