THE ART OF B2B INTEGRATION

A STUDY OF THE STATE OF ART B2B INTEGRATION PATTERNS, COMPONENTS AND ARCHITECTURES

Table of Contents

List of Figures

List of Abbreviations and Acronyms

API- Application Programming Interface

ASP – Application Service Provider

RPC- Remote Procedure Calls

SOA- Service Oriented Architecture

XML- Extensible Markup Language

Keywords

Integrations, Business to Business, Service oriented architecture, process, nodes, architecture.

ABSTRACT

In modern business operations, it has become imperative that processes, systems and sometimes organizations integrate. This is for various reasons such as maximizing profits, efficiency, expansion etc. Developing a robust integration plan, architecture and process mapping are key to any B2B integration. The growth and change of software and technology is a very rapid area, thus requiring almost every software system to have constant modification, integration and/or configuration, to provide enhanced integrated solutions to the changing world. Simply put its developing an approach to collaboration, communication, resource sharing, optimization, profitability and efficiency. To define it integration in the

computer industry, is a general term for the software that mediates or joins together two or more separate and usually already existing programs, applications, or systems. Integration tools and applications involves allowing different systems to interoperate and communicate with each other within or beyond a business enterprise, thus allowing complete integration of services and data sources among different applications. Current integration architectures are sufficient on basis of service or product needs. But most B2B integrations leave out the process which contains most if not all stakeholders concerns and quality attributes. In this study, we take a look at the art of integration architecture and patterns, the impact on business processes, services, products as well as challenges hitherto.

1. INTRODUCTION

1.1. Background

The modern economies are digital thus businesses values and competitive advantages are beyond the enterprise boundaries. Its therefore critical to focus on relationships with different partners, suppliers and peers. The business landscape is changed, is rapidly changing and affects the way enterprises conduct business, between its partners as well as the management of the internal business processes. Collaborative e-commerce is the current wave and it requires dynamic creation and management of trading relationships with partners, the public and private business processes automation, increased adaptability and increased flexibility through an integration. Thus business processes are affected heavily by other processes and need for interoperability. When systems integrate, processes are affected, service delivery must change and thus entire planning formula has to be integrated as well.

A value network is a business analysis perspective that describes social and technical resources within and between businesses. The nodes in a value network represent people (or roles). The nodes are connected by interactions that represent tangible and intangible deliverables. These deliverables take the form of knowledge or other intangibles and/or financial value. Value networks exhibit interdependence. They account for the overall worth of products and services. (http://en.wikipedia.org/wiki/Value_network, 2015). This and several other business principles drive firms to integration. One other critical drive for integration is customer needs and value add requirements like 24 hours availability, Out of box transactions, Information sharing and notifications and many more customer satisfaction needs B2B integration is the integration of applications, programs, or systems beyond the walls of an organisation thus extending the organisation in terms of size, service and processes, value and relations. B2B includes heterogeneous infrastructures, data, application software, and business processes integration between two or more businesses. It enhances exchange of data, unify software components, and streamline business processes (Laudon&Traver, 2013). B2B integration strategy should aim to have an integrated, real-time application-to-application, system-to-system interaction with all the partners including the existing ones and new ones. The strategy should also aim to eradicate all the manual steps in business processes, conduct real-time and secure transactions over the Internet, be flexible to accommodate different modes of interaction for each partner, and be able to adapt to change easily and quickly.

It is important to note that based on recommended architectural standards IEEE 1471 (Hilliard, 2007) adopted as ISO/IEC 42010 (Wikipedia, 2011), An Architectural Description (AD) must contain at least the following: Identification of stakeholders and concerns, Selection and declaration of the viewpoints used, Architectural views, each conforming to a viewpoint, Any known inconsistencies and Architectural rationale.

Thus a process stakeholder and concerns are key items in any B2B integrations.

2. LITERATURE REVIEW

1.1. Integration Overview

Sharing information between systems was difficult in the past. The development of many legacy systems used proprietary technology, which created islands of information in different departments of an organization and between the organization's different partners. Often, the development is incremental over time resulting in a haphazard architecture that resists the evolution towards standardizing systems(Laudon&Traver, 2013). B2B integration is an important requirement for businesses that need to share information between its various departments and partners and automate its activities involving multiple business functions, departments, and businesses. More than ever, businesses need efficient management and integration of their information in order to remain viable in the ever increasing competitive market.

B2B integration was defined as the integration of applications, programs, or systems extending the walls of an organisation and it include heterogeneous infrastructures, data, application software, and business processes integration between two or more businesses. B2B integration has been developed to handle information between businesses, business processes, and other components of interest between businesses. It is a top priority for the competitive businesses with an objective of linking data in the diverse information across its partners (Wong, 2013). This report focuses on integration techniques that extend through various businesses.

There is also need to consider integration of information resources and how they work together in the form of data, process, and application etc. An information resource is an object or a service that can handle, process, or produce data in a way that involves communication with external information resources. When information resources work together, either on a network, or like objects within an application, integration is achieved. This is definition cuts across all levels of integration.

Types of Integration

There are several different systems that divide integration solutions into different levels. One way is dividing integration techniques into external and internal integration. In more detail, categorization focuses on the scalability and the flexibility of the integration solution. In this later categorization, the various levels include data, platform, component, application, process, and B2B integration.

a. Platform Integration

Platform integration is the solution for connectivity and interfaces between systems having different hardware, operating systems, and applications. Platform integration solutions are individual and different Remote Procedure Calls (RPCs) or Brokers carry out the connection(Wong, 2013). Every integration solution is individual and the workload is therefore, high when new systems need to be added to the present architecture, just as the first integration.

b. Data Integration

Data integration solutions provide tools that extract, transform, and move data. Data integration includes integration of platform and it requires information about the database schemas that underlie the data(Samtani, Healey&Samtani,2002). An example of data

integration is the movement of data between two databases through SQL. In many cases, data integration is possible between two different database systems and platforms, which make use of API and database connectivity drivers in accessing each of the servers of the databases and fetch data into the other.

c. Component Integration

Component integration is the development of data integration where several network features such as the load balance, session management, fault protection, and security are added to the product. At the base of a component integration solution there is a server that handles the network features and thus making it easy to add new logic(Samtani, Healey&Samtani,2002).

d. Application integration

Application integration solutions are those that provide a framework for creating and changing the integration solution such as easy and fast addition of new applications. The application integration framework consists of pre- built adapters for the most common systems, which lower the time for adding new applications. When applications change, only the mapping to the unified form must change and all the other applications remain unchanged. The solution includes platform, data, and component integration. The figure 2.1 below is a description of application integration and the flexibility of the integration topology.

In employing application integration, an organization aims to create a framework that integrates incompatible and distributed systems within an organization thus making it easier and faster to extend its business processes throughout the organization (Nunamaker, Chen &Purdin,1991). Today, every organization must strive to develop a framework, tools, and infrastructure to accomplish this integration. Enterprise integration is the process of making disparate applications work together to produce a unified set of functionality. This process entails more than just integrating the applications but also considering the criteria, integration options, patterns and policies as well as the definition of the best approaches in integrating the applications.

Enterprise application integration is the use of software and computer systems architectural principles to integrate a set of enterprise computer applications. It is the business computing term for the plans, tools, and methods of modernizing, consolidating, and coordination of computer applications(Wong, 2013). Typically, enterprises have existing legacy applications and databases, which they want to continue to use while they add or migrate to new set of

applications that exploit e-commerce, Internet, extranet, and other technologies as they emerge. When new applications are developed, they must fit into the view of the business and applications view of an enterprise. Ways to efficiently reuse the resources that already exist while adding new applications and data must also be devised.

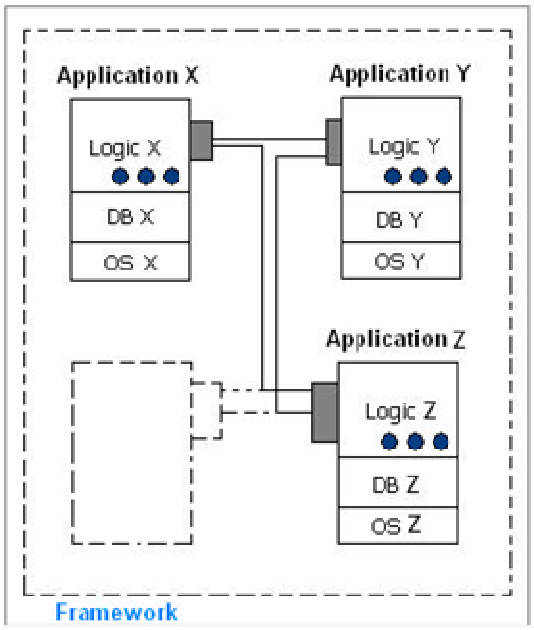

Figure 2.1 Application Integration

e. Process Integration

Process integration is the integration using a graphical modelling interface above an integration server and thus connecting all the applications (Samtani, Healey&Samtani,2002). Therefore, it provides another level of abstraction and adaptability to integration solutions. This is done through a graphical modelling interface above an integration server that connects all application. The solutions of process integration enable business managers to define, change, and monitor business processes. See figure 2.2 below.

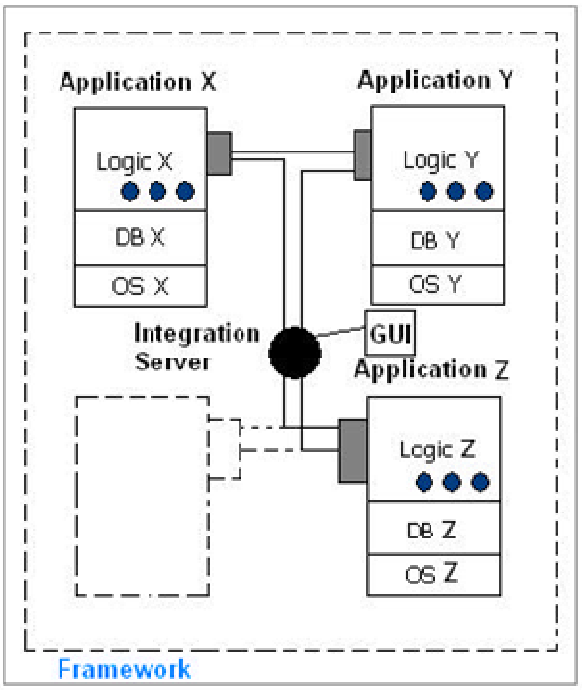

Figure 2.2 Process Integration

2.2. B2B Integration

B2B integration is basically the secured coordination of information among businesses and the businesses' information systems. See figure 2.3 below. B2B integration solutions provide a technology framework for B2B collaborative e-commerce (IBM, 2003). B2B integration promises to transform the way an organization conducts its business with the different partners. Through integration of technical processes and businesses, organizations can strengthen their relationships with service partners and their customers, they can achieve unified integration within and beyond an enterprise, they can gain real time views for the accounts for customers, they can reduce costs, and increase operational efficiency. Therefore, B2B integration is the pervasive enabler of business strategies such as collaborative networks, collaborative e-commerce, supply chain management, and customer relationship in a multichannel delivery system including the Internet and wireless devices.

While application integration is the integration of internal applications in an enterprise, B2B integration is the integration of data and business processes across multiple partners. Both the enterprise application integration and the B2B integration are accomplished by data, application, and process integration and therefore, integration challenges in enterprise

application integration and B2B integration have a lot in common and a single, integrated solution can solve them.

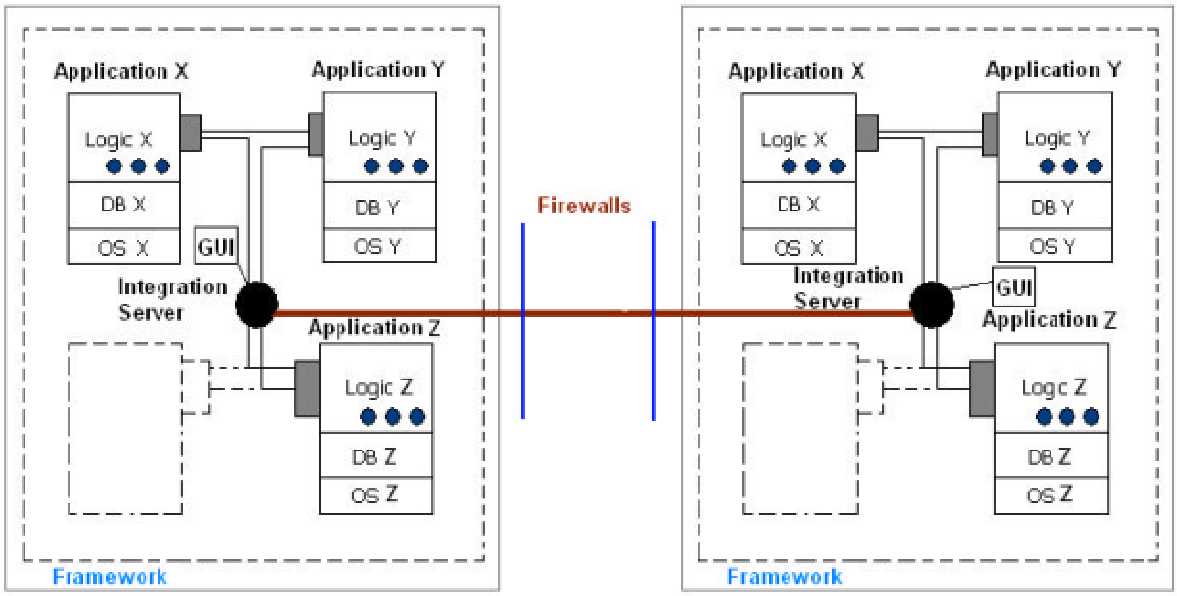

Figure 2.3 B2B Integration

2.2.1. Steps in building a B2B Integration Solution

- Plan a B2B integration strategy: This involves stating clearly the business processes that can be integrated with the partners over the Internet, the benefits the migration strategy will provide and the resources required. The step should also encompass evaluation and benchmarking of how others have implemented b2B integration.
- Determine the short-term and long-term goals:
- Leverage existing systems: Companies should evaluate how they can keep their existing internal systems while extending information systems to its partners.
- Choose the right solution and solution provider.

2.3. B2B Integration Patterns

Businesses can make use of one or more of the integration patterns to achieve B2B integration. They include:-

- **Data oriented integration:** The integration pattern integrates data across organizations through batch processes, replication, data marts, data warehouses, and data federations. Data sources exist independently and their data schema may change

at any given time. The sources run on different platforms and provide proprietary APIs(Samtani, Healey&Samtani,2002). The integration pattern attempts to eradicate the use of proprietary APIs in accessing data. Data from disparate internal data sources can be accessed over an intranet and external data from various partners through a standard interface.

Data oriented integration allows businesses to publish and share corporate data over the internet regardless of their networking platforms, operating systems, or database systems. The integration also supports unified information views to all the partners and update across their systems.

- **Application oriented integration:** The integration pattern integrates distributed applications by invoking the exposed interface of each of the organizations through standard mechanisms such as APIs and RPCs. The pattern involves direct application-to-application integration across organizations and across platforms over a network. Application to application integration requires new and old technologies to integrate thus realizing a common architecture thus allowing applications among businesses to connect and coordinate data and events(Samtani, Healey&Samtani,2002).

  Application oriented integration is based on request/reply interactions between clients and the server. The middleware, therefore, must provide the adapters supporting synchronous program invocation.

  For a successful implementation of this pattern, organizations must give away autonomy and develop applications cooperating with their partners closely. To make any changes, informing and coordinating with the other partners is unavoidable. This makes the pattern least inflexible rendering the pattern very expensive. The pattern also requires authenticating strictly before executing functions and methods.

- **Portal oriented integration:** This integration pattern presents a consistent web interface or presentation layer for major applications and major business processes to a user in any organization in a personalized and a secured way. Portals provide web front-end that supports key business functions that involve business transactions and sharing of information between different partners, service providers, departments, employees, and customers(Samtani, Healey&Samtani,2002). Through portals, secure and single point interaction with diverse information, processes, and people is possible. Users use a web browser to access and interact with applications that a company hosts. The content and the data that is accessed reside virtually anywhere.

In B2B integration, web portals provide a quick and efficient way of establishing and maintaining the relationships in a business and all interactions are done through personalized services thus adding value to them. Partners can collaborate through portals by engaging tools that help cross-organizational teams to communicate and work together. The portals allow for partners to get information regardless of the distance between partners and at any time.

However, in most cases, portal oriented integration does not achieve full B2B integration as most portals do not have sophisticated tools for management of business processes. Most of them lack tools that allow data to be transferred based on business events automation.

- **Business process oriented integration:** The integration pattern integrating internal and external business processes that spread across multiple organizations. The integration pattern aims to implement integration based on rules of the business in order to improve business efficiency.

  The pattern includes logic and reasoning of doing business. It removes the limitations of data oriented and oriented integration patterns by focusing on business processes and not only the data(Samtani, Healey&Samtani,2002). To achieve this pattern, organizations must involve design, model, automation, execution, and monitoring of the business processes. An instance of business process integration may involve multiple application and data integration at different levels and involving multiple steps.

2.4. Benefits of B2B Integration

a) Dynamic business relationships: B2B integration allows quick and flexible response of dynamic business relationships and processes as business models and customer demands change. Through B2B integration, business can bring into their range new associates and automate business processes across enterprises in an easier, faster, and safer way.

b) Real-time information across partners: Through B2B integration, businesses can exchange real-time, task-specific, and partner-specific information over the Internet in a secure way. The organizations benefit from the' power of now'. It gives companies the 'power of now'.

c) Lower transaction costs: B2B integration enables businesses to reduce transactional costs and the complexity associated with manual transactions while they maintain complete business logic.

d) Participation in online marketplaces: Through B2B integration, companies can build digital market places and participate in multiple horizontal and vertical market places. This cuts transactional costs, cuts production and inventory costs, and reduces exceptions and other errors.

e) Streamlining of business operations: Through B2B integration, companies can automate, reshape, and improve their business processes' efficiency through business process management. The integration automates inter personal, inter-functional, and inter-organizational activities of the business processes.

2.5. Challenges of B2B Integration

a) Internal application integration: A major challenge in B2B integration is the integration of information among internal enterprise applications. Since B2B integration is accomplished through multiple application systems, organizations must first integrate their internal applications before implementing the B2B integration plan.

b) Disparate internal corporate data: When application data in organizations is scattered across several databases, internal application integration becomes very complex.

c) System heterogeneity: For large companies conducting business with various partners, the internal and external system heterogeneity is a big task. The organization's IT infrastructure comprises of multiple applications running on multiple platforms. Additionally, each of the partners is in a similar situation. This provides a complexity and difficulty in integrating the technologies for heterogeneous and distributed system for B2B integration.

d) Data security: Businesses need to find a complete B2B integration solution to link their systems with those of their partners quickly, bring guaranteed delivery, transaction traceability, and secure communications over the Internet. The transactions mostly involve high value and sensitive data and therefore, there is a lot at stake making the issue of security is extremely important.

e) Transaction integrity: Transaction integrity is the degree to which a transaction flowing through a network reaches its destination without diminishing of its meaning,

content, or function. Since B2B applications are distributed across many organizations, maintaining transactional integrity is a major challenge.

f) Process differences. In organizations different processes are handled differently even f in the same industry.

2.6. Approaches to B2B Integration

Manual B2B Integration

Manual B2B integration is the integration process involving personnel (e.g. employees or customers) to act as the interfaces between business to business applications and thus facilitate their integration (Jones, 2011). Manual integration could be in the form of the collection of information and entry of the information in multiple systems and consequently reading information from those systems to respond to different business needs. In other cases, an employee may have to get information from the database of one business, and then provide it to an employee of the other business to enter it into the other business' database so that the business can use the information. This form of integration requires very little technology investment. However, manual integration has the following drawbacks:

- The manual integration approach is highly inconsistent and lacks scalability which poses a high probability of inaccuracies in data
- When there are many businesses involved depending on the same data, the integration becomes more complex
- As the complexity and the amount of data needed by a business and its partners increases, or as the number of applications using the same data from different organisations increases, organizations will require more and more people to maintain such manual integration environment.

An environment relying heavily on manual integration is very inefficient, and restricts growth as opposed to more automated techniques.

Partially Automated Interconnection B2B Integration

Partially automated B2B interconnection integration combines both the human effort and some automation (Jones, 2011). The integration may involve a person in an area where the

corresponding automated solution is too expensive or difficult, or where the businesses require a human decision. In other parts, the systems between different businesses are usually interconnected through a network.

Partially automated technique requires more technology investment as compared to manual integration. However, once the businesses make the initial investment, the number of people involved in integrating the different businesses' applications is often reduced. This reduction on human involvement in this manner usually reduces costs and increases reliability.

Partially automated B2B integration is still not efficient enough, since it still requires the intervention of some people in the middle of the process, for instance the need for human intervention to transform data that is required in another system.

Fully Automated interconnection B2B Integration

Fully automated B2B interconnection integration removes people from the businesses process entirely, although the process requires them to maintain the solution. Fully automated interconnection is a type of integration consisting of applications and systems between different organisations communicating through a series of interfaces and adapters (Jones, 2011). For example, two databases from different organisations might share data, which is then automatically transformed and committed to the second database from the first without human interaction. Fully automated B2B integration removes the dependency on people. However, the systems can be more expensive to implement and may not be practical for some business problems. In many cases, organisations will still require people to make business decisions, and often a person controlling a technical process as well is more efficient. The praised 'fully' automated B2B integration technique is arguably the reliable technique due to scalability, consistency and efficiency derived in the process of sharing and exchanging business data.

2.7. B2B Integration Components

B2B integration has multiple components with differing priorities depending on the involved organizations. Whereas it might be hard to incorporate all the B2B integration components in the first phases of implementation, including majority of them at least in the long term B2B

goal is important. The most important B2B components include XML, the Middleware, integration brokers, and wireless technologies.

XML

Extensible Markup Language is a technology that is based on open standards and aims to standardize and simplify processes communication among organizations(Samtani, Healey&Samtani,2002). It provides an open platform for organizations to communicate over a network and thus it enables the dynamic 'anywhere-to-anywhere' interactions of businesses. XML is the universal language, which makes data over a network meaningful and thus enables different systems to recognize the data easily and allows for easy exchange of contents and business documents. XML in B2B applications allow trading partners to avoid the cumbersome process of mapping the other businesses' process into complex electronic data interchange messages. Businesses incorporating B2B integration must have XML-centric business processes and their structure must support XML standards, enable process transformations, provide applications links.

Middleware

Middleware is the software layer between the client and server applications providing a uniform channel for the applications to communicate with each other(Samtani, Healey&Samtani,2002). Middleware provides the platform that allows applications developed in different languages and installed on different operating systems and data sources from different vendors to integrate.

Integration Brokers

Integration brokers are interface engines between two different third-party systems that provide adapters for converting data to and from various third-party formats(Samtani, Healey&Samtani,2002). Integration brokers contain message brokers, which are built on top of messaging middleware and provide intelligent translation and routing message from the source application to the target application. Message brokers enable content based, conditional, and sequential routing of messages. Message brokers also function as message warehouse that stores raw unprocessed messages for retrieval and archiving.

In B2B integration, integration brokers will enable the businesses achieve reliable and faster integration. The brokers provide adapters for servers, packaged applications, legacy applications, and databases. They also support all open standards and enable applications to communicate irrespective of the protocol for communication. Integration brokers link applications both within an enterprise and across business to business networks.

Wireless Technologies

Organizations need to provide their employee's and partners access to computing resources anywhere they might be. The access is not completed if it is limited to basic features such as emails. It should include hookup into applications enabling them be productive anywhere at any time. Therefore, there is need for a solution that extends internet, intranet, and extranet applications wirelessly to handheld devices and phones. Integrating mobile devices with the internal and trading partners' application is an important part of B2B integration.

3. B2B INTEGRATION ARCHITECTURES

B2B integration can be supported by different architectures. One of the factors that determine the architecture to be adopted is the number of connections to be made, that is the number of businesses to be interconnected, the size of those businesses, and the number business processes that need integration in those businesses (Samtani, Healey&Samtani,2002). Some architectures work efficiently when there are a few processes or applications to be integrated. However, as the number of processes/applications increase, alternative architecture is necessary. That is why we propose the process to node architecture.

An important part of making B2B integration scalable is to increase the level of automation and reduce human intervention. This has made fully automated B2B integration more viable, scalable, and efficient as compared to the manual and semi-automated integrations. This generally involves creating interfaces between businesses' systems along with predefined logic that replaces the involvement of human beings. The choices for automated B2B integration discussed in this paper are:

a. Point-to-Point Integration

Point-to-point integration is the pair-wise integration of different organisation's back-end application systems. Every back-end application system in this architecture has an established

direct data transfer to transport messages between two connected back-end application systems(Ramblings, 2012). See figure 3.1 below. An integration software mines data form one back-end application system and transports it to the other back end application in which it is connected to.

Point-to-point integration may involve synchronous communication or asynchronous communication. In synchronous communication, the software system extracts data from a back-end application and invokes the other application synchronously to insert the data in it. Data transformation on the data occurs after the extraction but before being taken to the receiving application. In asynchronous communication, data is stored in an intermediary storage after extraction where the integration software retrieves it from and inserts it to the receiving application. The intermediary storage in asynchronous communication acts as an integration buffer that mediates extraction and insertion speed and isolates application systems from each other enabling application independence. Instead of intermediary storage for asynchronous communication, persistent queues may be used. Where persistent queues are used, de-queuing of messages is in the order in which they were queued.

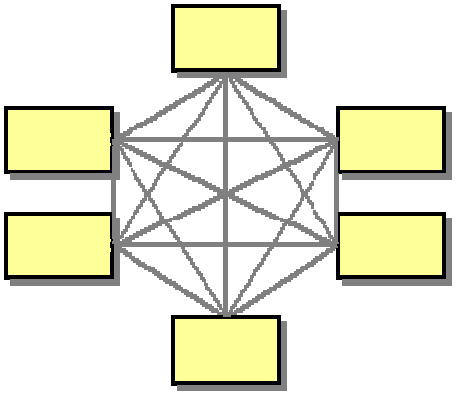

Figure 3.1 Point-to-Point Integration

b. Service Oriented Architecture (SOA)

SOA is an architectural style that guides creation and use of business processes, which are packaged as web services, throughout their lifecycle. It defines and provides the IT infrastructure allowing different application systems to exchange data and participate in business processes regardless of the programming languages or the operating systems underlying the systems (Durvasula, 2006). SOA represents a model in which functionality is decomposed into small, distinct unit known as services. These services are distributed over a

network and their combination is reused to form business applications. By passing data from one service to another, the services are able to communicate. See figure 3.2 below. They can also communicate by coordinating an activity between the services (Bussler, 2003). SOA consists of service and event interfaces to both legacy systems and newer applications that are built on top of platforms.

SOA architecture encourages creating loosely coupled business services by integrating disparate applications regardless of their underlying technologies. SOA's one key goal is to maximization of the reuse of application-neutral services in order to increase business and IT efficiency and adaptability. SOA incorporates a style that supports loosely coupled services efficiently thus enabling business flexibility in an interoperable, technology skeptical manner (Bussler, 2003).

Also SOA consists of a composite set of business services that support a flexible delivery model with dynamically re-configurable end-to-end business process realization using event-driven, interface-based service descriptions (Erl,2005).It provides the ability to invoke remote business services and install them as local components in a different application, all without writing a single line of low-level code.

SOA contains three main components, namely: Service provider, service broker, and a subscriber. The service provider is responsible for creation and publishing of the service interfaces, provision of the actual services, and provision of responses to any use of the services. Service brokers are responsible for the registration and categorization of the public services by service providers. Service brokers also offer search services for users to search for the registered service providers and what they offer. Subscribers or the service requester are the actual users of the services and they subscribe the services from the service providers (Durvasula, 2006)[2].

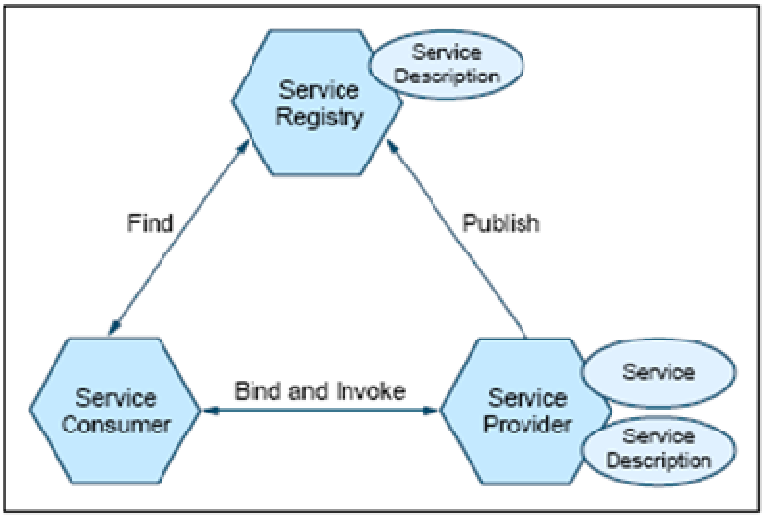

Figure 3.2 SOA components

SOA defines how to integrate widely disparate applications for a Web-based environment and uses multiple implementation platforms. Rather than defining an API, SOA defines the interface in terms of protocols and functionality. An endpoint is the entry point for such a SOA implementation.

SOA Framework

SOA-based solutions endeavour to enable business objectives while building an enterprise-quality system.

SOA architecture is viewed as five horizontal layers:

1. Consumer Interface Layer – These are GUI for end users or apps accessing a pps/service interfaces.
2. Business Process Layer – These are choreographed services representing business use-cases in terms of applications.
3. Services – Services are consolidated together for whole-enterprise in-service inventory.
4. Service Components – The components used to build the services, such as functional and technical libraries, technological interfaces etc.

5. Operational Systems – This layer contains the data models, enterprise data repository, technological platforms etc.

There are four cross-cutting vertical layers, each of which are applied to and supported by each of the following horizontal layers:

1. Integration Layer – starts with platform integration (protocols support), data integration, service integration, application integration, leading to enterprise application integration supporting B2B and B2C.

2. Quality of Service – Security, availability, performance etc. constitute the quality of service parameters which are configured based on required SLAs, OLAs.

3. Informational – provide business information.

4. Governance – IT strategy is governed to each horizontal layer to achieve required operating and capability model.

Principles

There are no industry standards relating to the exact composition of a service-oriented architecture, although many industry sources have published their own principles. Some of these include the following:

1. Standardized service contract: Services adhere to a communications agreement, as defined collectively by one or more service-description documents.

2. Service loose coupling: Services maintain a relationship that minimizes dependencies and only requires that they maintain an awareness of each other.

3. Service abstraction: Beyond descriptions in the service contract, services hide logic from the outside world.

4. Service reusability: Logic is divided into services with the intention of promoting reuse.

5. Service autonomy: Services have control over the logic they encapsulate, from a Design-time and a Run-time perspective.

6. Service statelessness: Services minimize resource consumption by deferring the management of state information when necessary.

7. Service discoverability: Services are supplemented with communicative meta data by which they can be effectively discovered and interpreted.

8. Service composability: Services are effective composition participants, regardless of the size and complexity of the composition.

9. Service granularity: A design consideration to provide optimal scope and right granular level of the business functionality in a service operation.

10. Service normalization: Services are decomposed and/or consolidated to a level of normal form to minimize redundancy. In some cases, services are denormalized for specific purposes, such as performance optimization, access, and aggregation.

11. Service optimization: All else being equal, high-quality services are generally preferable to low-quality ones.

12. Service relevance: Functionality is presented at a granularity recognized by the user as a meaningful service.

13. Service encapsulation: Many services are consolidated for use under the SOA. Often such services service.

14. Service encapsulation: Many services are consolidated for use under the SOA. Often such services were not planned to be under SOA.

15. Service location transparency: This refers to the ability of a service consumer to invoke a service regardless of its actual location in the network. This also recognizes the discoverability property (one of the core principle of SOA) and the right of a consumer to access the service. Often, the idea of service virtualization also relates to location transparency. This is where the consumer simply calls a logical service while a suitable SOA-enabling runtime infrastructure component, commonly a service bus, maps this logical service call to a physical service.

SOA BENEFITS AND LIMITATIONS

Its technical characteristics translate directly into real bottom-line benefits.

The main characteristics include:

- Loosely coupled architecture
- Modular approach
- Non-intrusive nature
- Universality of standards.

Issues, inherent due to the very nature of service orientation, can be summarized as follows:

1) Coarse granularity: This may mean that
    a) testing and validating every combination of every condition in a complex service may well become humanly impossible;
    b) one service trying to serve a dozen masters may lead to spaghetti code and therefore introduce massive inefficiency and
    c) a generic service, because of its coarse granularity, cannot be easily optimized for efficiency

- Loose coupling: It is an architect's dream but making a system distributed adds a new level of complexity and therefore, as Fowler (Fowler, 2002) puts it, it can become a developer's nightmare
- Integration of services: This can be a complex task especially when there is a lack of skilled people to work in a SOA based environment
- Service interoperability: When web services are used to exchange SOAP messages over HTTP, encapsulating XML data, integration of services in heterogeneous environment can become a serious issue
- Evolutionary development: Building and updating services is fine. However, if applications continually require additional functionality, and these requests are granted, the entire system may become unstable.

- Process visibility. One may not whole figure out the business process end to end though the integration process is visualized.

4. ANALYSIS AND SUGGESTIONS

Traditional integration methods are great and have worked wonderfully especially messaging. Most common integration practice is to achieve short-term ad-hoc objectives by creating dedicated point-to-point links between the subsystems everywhere it is necessary. Most of the links are established between components of the process and control layer, where supervisory control systems receive data from plant-floor devices and subsystems.

After years of establishing ad hoc links, the interconnection network becomes very complicated and chaotic causing that we have to deal with the following problems:

1. Difficult modification and maintenance – each new system will require new links making the structure more and more complex and deepening chaos of communication.
2. Inefficiency – a lot of the associations are based on a common communication medium, sometimes of a low quality, e.g. enterprise field controller network – the complex structure will necessitate transferring the same data over the network many times and, finally, cause a waste of the bandwidth.
3. Costs – if communication is based on the third-party toll infrastructure, the fee for the data transfer may be significant. Additionally, a strong dependence on the independent operator increases.
4. Partial interoperability –the data presentation and description at the link sink is precisely suited to the systems it services and, therefore, data cannot be accessed directly by other systems.
5. Mess – in a real world enterprise there can be tens, hundreds or even thousands of subsystems in every layer (in the figure above there are only a few components). Maintenance and documentation of such a complex architecture is a real challenge for administrators.
6. Anarchy – if subsystems have different methods of authorization, authentication and user rights management, it is almost impossible to keep an appropriate level of security. As a consequence, some data can be lost because the rules are too weak or communication cannot be established because the rules are too strong.

There are strong advantages provided by a process to node integration architecture. Simply map a process to service, process, point or product as a node and develop an expected web considering the business process factors. The key to B2B integration architecture is the process. The need maybe initiated by any other requirement such as a function or a service but these too belong to a process that is within a business entity. Thus the strategy is to get all the business processes to be integrated, extract the stakeholders and concerns and map them to a node for integration.

A node may be defined as an endpoint, product, interface, a system or a database.

If optimum integration is implemented, then the following outputs are expected:

- Efficiency of inter-organizational business operations will improve.
- Transaction costs will reduce.
- Time to get information from a partner will reduce by 70%
- Wholesome business knowledge will increase.
- Learned and leading model of integration.

5. CONCLUSION

B2B integration has many advantages. But it's all about the business undertaking that translates to margins. For a holistic B2B integration taking care of both present needs and expandable into future, I suggest enhancing Service Oriented Architecture using process to node model. With business needs evolving rapidly a process to node architecture will guide based on the four angles of a business The Customer, The product/service, The delivery and The Margins and their tradeoffs. The node can be a product, a service point, a resource or even a deliverable in a business process. Thus the integrating business have a clear view of the requirement on basis of a holistic picture not at some point of need. Implementation can then take a combination of technologies but most importantly having already identified the perfect architecture of integration - a process to node map.

Works Cited

1. Fowler, M. (2002). Patterns of enterprise application architecture. *Patterns of enterprise application architecture* .
2. Hilliard, R. (2007). *All About IEEE Std 1471*. Retrieved from http://www.iso-architecture.org/: http://www.iso-architecture.org/ieee-1471/docs/all-about-ieee-1471.pdf
3. http://en.wikipedia.org/wiki/Value_network. (2015). *Value Network.*
4. Wikipedia. (2011). *ISO/IEC 42010.* Retrieved from Wikipedia: http://en.wikipedia.org/wiki/ISO/IEC_42010

References

1. Archer, L. B. (1981). A View of the Nature of the Design Research, in Design: Science: Method, R. Jacques, J. A. Powell, eds. (Guilford, Surrey: IPC Business Press Ltd., 1981), 30–47. L. Bruce Archer gave this definition at the Portsmouth DRS conference.
2. Bussler, C. (2003). B2B integration: Concepts and architecture: with 4 tables. Berlin [u.a.: Springer.
3. Dan, A., & ICSOC, Chicago, Ill.>. (2006). Service oriented computing: 4th international conference ; proceedings. Berlin: Springer.
4. Durvasula, S. (2006) SOA Practitioners' Guide, Part 1: Why Services-Oriented Architecture?
5. Durvasula, S. (2006). SOA Practitioners' Guide Part 2: SOA Reference Architecture
6. Erl, T. (2005).Service-Oriented Architecture: Concepts, Technology, and Design (Prentice Hall Service-Oriented Computing Series. Prentice Hall
7. IBM (2003). Business Integration Solutions (WebSphere Software), IBM
8. Jones, R. (2011). B2B integration.Manufacturing Engineer, 80(4), 165.
9. Krishnaprem, B. A. P. H. P. (2013). Getting Started with Oracle SOA B2B Integration: A Hands-On Tutorial.Packt Publishing.
10. Laudon, K. C., &Traver, C. G. (2013).E-commerce: business, technology, society (Ninth ed.). Prentice Hall.

11. Nunamaker, J. F., Chen, M., &Purdin, T. D. (1991). Systems Development in Information System Research: Journal of Management Information Systems 7 (3): 89 – 106.
12. Ramblings, G. (2012).Hub and Spoke [or] Zen and the Art of Message Broker Maintenance.Integration Patterns. Retrieved October 24, 2014, from http://www.eaipatterns.com/ramblings/03_hubandspoke.html
13. Runeson, P. (2012). Case study research in software engineering guidelines and examples. Hoboken: John Wiley & Sons.
14. Samtani, G., Healey, M., &Samtani, S. (2002).B2B integration: A practical guide to collaborative e-commerce. London: Imperial College Press.
15. Scott, M. S. (1984). The state of art of research.In the Information System Research Challenge, F. W. McFarlan, ed. Cambridge, M.a: Havard Business School Press: pg. 13 – 41
16. Wong, W. E. (2013). Emerging technologies for information systems, computing, and management. New York, NY: Springer.
17. M. Fowler, "Patterns of enterprise application architecture", Addison Wesley, 2002.