USING A STANDARDS-BASED INTEGRATION PLATFORM

- Many organizations have introduced electronic means to perform their business-to-business (B2B) transactions. These means include electronic data interchange (EDI) via value-added networks (VANs).
- Although EDI technology has existed for quite some time and is mature, a large number of companies are still using manual phone or fax-based processes to transfer business information (e.g., purchase orders).
- Larger businesses, on the other hand, have widely adopted EDI and benefited from it.
- Recently, the use of Internet in conjunction with technologies such as extensible markup language (XML) and XML Web services is being explored as a possible alternative to traditional EDI.
- As a result, businesses that were unable or reluctant to adopt EDI may now have an opportunity to participate in electronic B2B transactions.

B2B Integration

- Business-to-business (B2B) integration is an automation of business processes and communication between two or more organizations.
- It allows them to work and trade more effectively with their customers, suppliers, and business partners by automating key business processes.
- B2B integration software provides the architecture needed to digitize information and quickly route it through an organization's trading ecosystem.

B2B Integration Platform

- A B2B integration platform helps companies integrate all their complex B2B and EDI processes across their partner communities in a single way.
- The platform collects data from source applications, translates the data into standardized formats, and then sends the documents to the business partner using the appropriate transport protocol.
- B2B integration software is available for on-premises use or integration services that can be accessed through hosted cloud services.

Why is B2B integration important?

- Digitalization is pushing partner and customer expectations. Organizations are finding that slow, inefficient, error-prone manual processing of information is not sustainable in a digitally-connected world.
- Every business has its own mix of systems and applications for exchanging files and messages with partners. Disparate technologies make it difficult to communicate.
- To achieve goals like increasing revenue, speeding time to market, and improving efficiencies, organizations need a successful business network — and that requires a modern B2B integration solution.
- With the right B2B tools, organizations can digitally connect and communicate quickly and reliably. This can reduce the time it takes to get new products and services to market and help companies achieve the nimbleness and agility they need to compete.

Key Capabilities of B2B Integration

- *Communications adapters*
  Never say no to an onboarding request. Supports security-rich internet communications protocols, including AS2, SFTP, MQ, HTTP, Connect:Direct, and others.
- *Application integration*
  Includes adapters to connect to back-end systems, including databases, SAP as well as cloud object storage with AWS S3 Client Adapter.
- *Translation engine*
  Powerful any-to-any transformation. Converts documents from one format to another.
- *Mapping software*
  Enables the visualization of data fields and shows the relationships between documents.
- *Message tracking and reporting*
  Provides event management and reporting, auditing in

centralized dashboards that can enable you to extend real-time activity and status information to your system administrators and your partners.

- *Encryption*
  Encrypt data at rest or in motion. Robust certificate management including support of modern ciphers and algorithms.
- *Data validation*
  Applies business rules with graphical business process modeling to the information being sent to identify any errors.
- *Compression*
  Allows big files to be sent faster. Features include any-size file and message transfer.

TECHNOLOGY OVERVIEW

Services, Workflow, and Messaging

- All business and IT architecture have three (3) critical components mainly services, workflow, and messaging, which come under software architecture. But, not necessarily all business architectures cover all the aspects of services, workflow, and messaging, it actually depends upon the business model and various business drivers used to achieve the goal.
- Architects are expected to be able to recognize all the major components of the process, in defining the service component, designing the workflow, and implementing the various messaging protocols. It is important that the framework must be able to define people, process, technology, and information channels based on the problem in-hand.
- While developing an organization's service model, an architect must define the top-level business functions first. Once the business functions are defined, they are further sectioned into services that represent the processes and activities needed to manage the assets of an organization in their various states.
  - One example is the separation of the business function "Manage Orders" into services such as "Create Order", "Fulfill Order", "Ship Order", "Invoice Order", and "Cancel/Update Order."

Defining Service Description/Specification

- In order to define the service description or specification. It should consist of:
  1. A plain and detailed narrative definition supported by a low-level process model. It must be clearly narrated with the information about the service that facilitates service mediation and consistency checking of an enterprise architecture.
  2. A set of performance indicators that address measures and performance parameters, such as availability (when should members of the organization be able to perform the functions), duration (how long should it take to perform the function), rate (how often will the function be performed over a period of time), etc.
  3. A link to the organization's information model showing what information the service owns (creates, reads, updates, and deletes) and which information it references that is owned by other services.
- State of Services
  - Services can be stateless and stateful. Stateless services can be services like data aggregation services. Stateful services are used for executing business logic.
  - The Service Architecture should be design keeping in mind the below building blocks:
    - Service contract
    - Message Processing Logic (mostly used in web-service related architectures)
    - Service components (core service logic)

Service Component Architecture (SCA)

- Service Component Architecture (SCA) is a set of specifications that describe a model for building applications and systems using a Service-Oriented Architecture (SOA).

- SCA extends and complements prior approaches to implementing services, and SCA builds on open standards such as Web services (Figure 1).

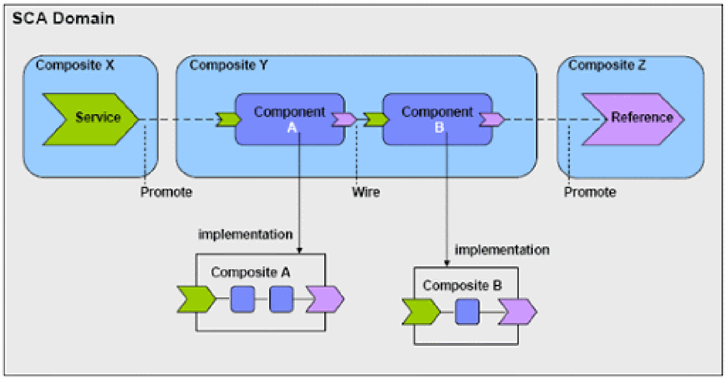

Figure 1. Service component architecture

- SCA is based on the idea that business function is provided as a series of services, which are assembled together to create solutions that serve a particular business need. These composite applications can contain both new services created specifically for the application and also business function from existing systems and applications, reused as part of the composition.
- SCA provides a model both for the composition of services and for the creation of service components, including the reuse of existing application function within SCA compositions.

Designing Workflow

- Workflow is fundamentally about the organization of work. It is a set of activities that coordinate people and/or software.
- Communicating the organization to humans and automated processes is the value-add that workflow provides to the solutions.
- A workflow may consist of other workflows (each of which may consist of aggregated services). The workflow model encourages reuse and agility, leading to more flexible business processes (Figure 2).
- Workflow can basically be defined by three (3) parameters:
  1. Input description - the information, material, and energy required to complete the step
  2. Transformation rules, algorithms, which may be carried out by associated human roles or machines, or a combination
  3. Output description - the information, material, and energy produced by the step and provided as input to downstream steps

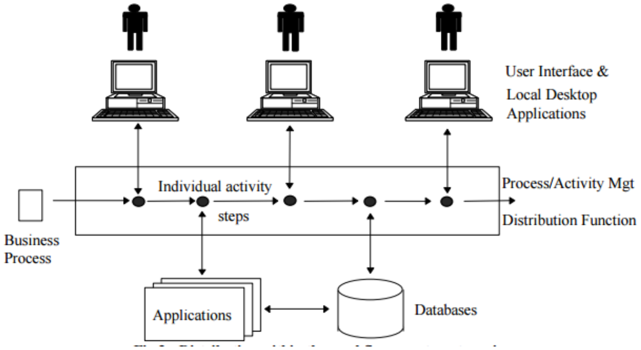

Figure 2. Workflow reference model

- **Workflow Lifecycle**
  - An individual business process may have a life cycle ranging from minutes to days (or even months), depending upon its complexity and the duration of the various constituent activities.
  - Such systems may be implemented in a wide variety of IT and communications infrastructure and operate in an environment ranging from small local workgroup to inter-enterprise (Figure 3).

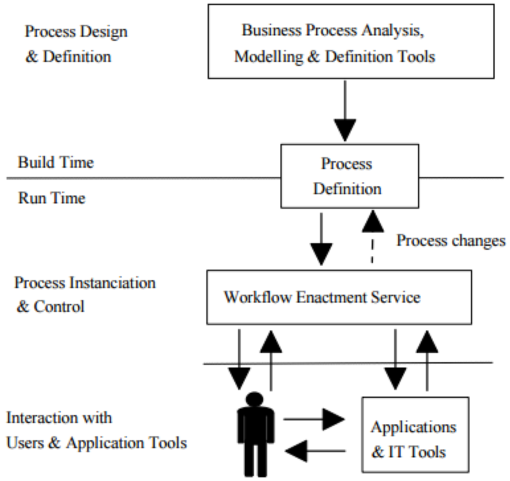

Figure 3. Workflow Lifecycle

Building the Messaging Architecture

- It is foremost important that the design of Messaging Architecture is such that the messaging framework must be capable of supporting the publish-and-subscribe message exchange pattern (MEP) and associated complex event processing and tracking.
- One of the problems with the distributed systems built today is that they are fragile. As one part of the system slows down, the effect tends to ripple out and cripple the entire system. One of the primary design goals of enterprise service bus (ESB) should be to eliminate those problems.
- The guideline should be available to developers to write code that is robust in production environments. That robustness prevents data loss under failure conditions.
- The communications pattern that enables robustness is one-way messaging, also known as “fire and forget”. Since the amount of time it can take to communicate with another machine across the network is both unknown and unbounded, communications are based on a store-and-forward model (Figure 4).

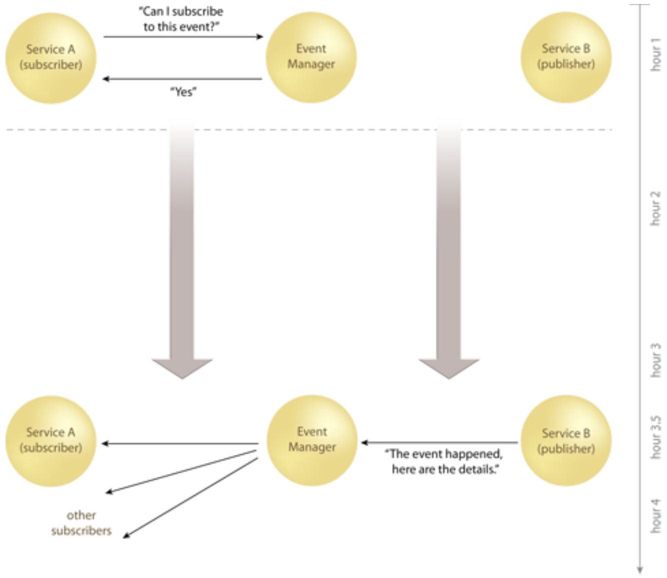

Figure 4. Store-and-forward model of communication

Message Exchange Pattern

- Messaging pattern is an important part of the enterprise service bus (ESB) in the service-oriented architecture (SOA).
- An ESB, being a model, is used for designing and implementing communication between mutually interacting software applications in an SOA.

  - How can the monitoring and control routing of message be exchanged between services?
  - How to resolve contention between communicating service components?

- How to control deployment and versioning of services?
- How to marshal the use of redundant services?
- How to cater for commodity services like event handling, data transformation and mapping, message and event queuing and sequencing, security or exception handling, and protocol conversion?
- How to enforce proper quality of communication service?

- The three basic message exchange patterns mostly followed are datagram, request-response, and duplex (Figure 5).

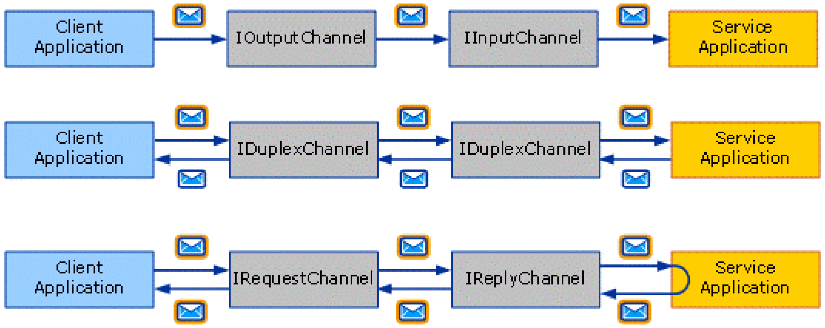

Figure 5. The three basic message exchange patterns

REFERENCES:

Fastrack IT Academy (2019). *SAP business one – Basic.* Fastrack IT Academy.

Kale, V. (2016). *Enhancing enterprise intelligence: Leveraging ERP, CRM, SCM, PLM, BPM, and BI.* CRC Press.

Lam, W. (2017).*Enterprise architecture and integration: Methods, implementation, and technologies.* Information Science Reference.

B2B Integration. Retrieved from: https://www.ibm.com/topics/b2b-integration?p1=Search&p4=43700054903266195&p5=b&gclid=Cj0KCQiAs5eCBhCBARIsAEhk4r539xArO3Hwy_b__iVaMGbYSEWr0iWuraZgx3XJa1boypWQOno4O1caAqpzEALw_wcB&gclsrc=aw.ds

Service, workflow and messaging. Retrieved from: https://itabok.iasaglobal.org/itabok3_0/capability-taxonomy-and-descriptions-3-0/specializations/software-architecture-specialization/services-workflow-and-messaging/