Enterprise Application Integration

What is EAI?

* Architectural principle to integrate a set of enterprise computer applications.
* Integration framework composed of middleware for enterprise application integration.
* Unrestricted sharing of data and business processes among the connected application or data source in enterprise

Why EAI?

* Makes diverse enterprise application including partner systems to communicate irrespective of platform and geographical location.
* Focuses on integration of both business level processes and data whereas traditional middleware approach is data oriented.
* Solution re-use, application decoupling and information standardization.
* Users need little understanding of the other applications for application integration.
* Decoupled application minimizes impact on the interfaces due to application upgrade, application re-hosting or application replacement.
* Function level integration rather than data level integration enables better use of application logic.
* Higher flexibility to accommodate change.
* Easy monitoring to identify the problem with particular interface.
* Provides common façade for cluster of application

What EAI does?

* Message acceptance
* Message transformation
* Message translation
* Message routing
* Message delivery

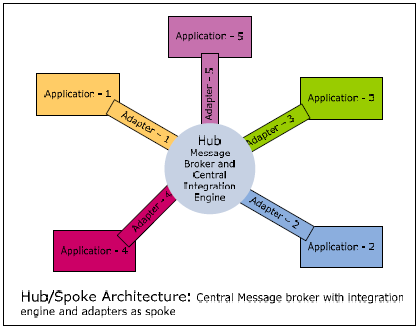
What design patterns are used in EAI?

* Integration pattern
  + Mediation pattern
  + Federation pattern
* Access pattern
  + Synchronous pattern
  + Asynchronous pattern
* Lifetime pattern
  + Short life
  + Long life

What is the architecture of EAI?

* Hub and Spoke
* Bus

EAI architecture: Hub and spoke



* Uses centralized broker (hub) and communicates with different applications through adapters (spoke)
* Adapter (spoke):
  + Connects applications to hub.
  + Converts data format to a format that broker (hub) understands.
  + Publishes message from applications to message broker.
  + Performs request from hub and notifies hub on event of interest.
  + Adapters interact with hub through message queues, web sevices or proprietary protocol.
  + JCA allows adapters to be created in vendor neutral manner.
* Broker (hub) :
  + Brokers all messages, transforms/translate messages to format that destination system understands.
  + Routes messages to subscribing adapters which in turn sends it to destination application.
* Example: Webmethods, IBM Cross worlds, Vitria.

Hub-Spoke :

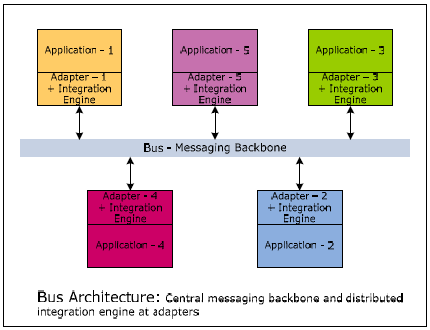
Advantages:

* Easy to manage
* Federated hub solves issues with scalability by having multiple hubs.

Disadvantages:

* Scalability hits as no. of applications increase.
* Because of central point of failure (hub), high availability of the server is very critical.

EAI architecture: Bus



* Uses centralized messaging backbone (bus) for message propagation.
* Applications publish message to bus through adapter
* Subscribing application adapter are responsible for taking the message from bus and transforming it in the format that application understands.
* In Bus architecture the integration engine that performs message transformation and routing is distributed in application adapter.
* Bus architecture requires application adapter to run on same platform as original application.

Bus architecture:

Advantages

* Bus architecture scales well but complex to maintain.
* Offers better performance than hub-spoke model.

Disadvantages

* Implementation of the bus model is more complex.
* Difficult to administer.
* Routing between the nodes is by using proprietary bus protocol, so customers are restricted to use homogeneous broker technology across whole bus.
* Logging or monitoring information across the bus created considerable challenges because it typically involves either broadcasting logging packets to centralized logging server which further clogged bandwidth or browsing the log files across multiple nodes.

EAI Advantages/Disadvantages

Advantages:

* Real time information access among systems.
* Streamline business process and improve organizational efficiency.
* Maintain information integrity across multiple systems.
* Ease of development and maintenance.

Disadvantages:

* High initial development cost.
* Difficult to provide business context for any technical failure.
* Flexibility required in business process required changes in business application and hence slowing down the entire process.
* Mostly focussed on IT benefits hence missed business and IT alignment.

EAI Tools

* TIBCO
* MQSeries
* WebMethods
* Vitria
* iPlanet
* BizTalk
* Weblogic

EAI and ESB

* EAI same as ESB, differs only from the emphasis of ESB on webservices
* Both share the idea of centralized point of control.

EAI and SOA comparison: ANALYIS

EAI

* Analyze application portfolio.
* Analyze information exchange b/w application as per technical business flow.

SOA

* Identify “To Be” business process which could be monitored from both business and IT perspective.
* Identification of user roles and system usecases which form the basis for service rationalization.

EAI and SOA comparison: DESIGN

EAI

* Identify integration points (database level, message level or API level).
* Define common message model to achieve source and target decoupling.

SOA

* Define service contract.
* Define SLA’s and map them to business KPI’s
* Change focus from technical monitoring to business monitoring for defining the SLA’s.

EAI and SOA comparison: IMPLEMENT

EAI

* Implementation logic for source to canonical message and canonical to target message format translation and validation.
* Implement MOM and broker platform to achieve loosely coupled integration
* Message reprocessing in case of broker failure.

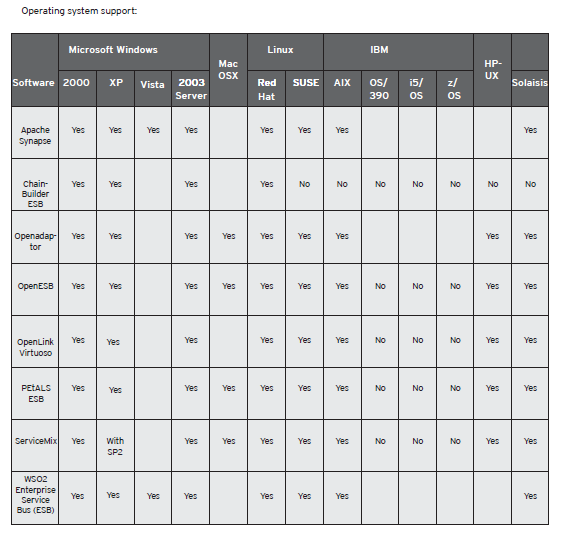
SOA

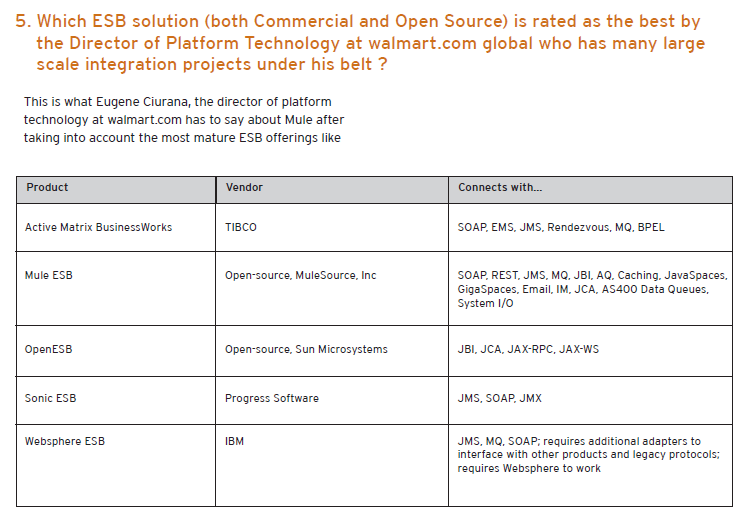
* Implement services and business KPI’s
* Use business rules for creation of services based on condition based service interaction.
* Use BPM for services which are based on process flow.

Comparitive analysis of some Open Source product offerings in the integration space



Operating System Support





**REFERENCES**

http://www.mphasis.com/pdfs/OpenSourceEAI\_SQATools.pdf

<http://www.infosys.com/offerings/it-services/packaged-application-services/white-papers/documents/eai-soa.pdf>