Key Differences between ESB, EAI and SOA

Enterprise Application Integration vs Service-Oriented Architecture vs Enterprise Service Bus: Advantages and Disadvantages

Simply put, an ESB is one model of SOA implementation and SOA is one way to achieve EAI. The relationship between these three concepts has been described as a Russian stacking doll, with EAI as the outer doll, SOA as the middle doll and ESB as the inner doll.

In order to further differentiate between these three technology concepts, which promote standardization both within and across industries, it is best to first define them and then understand their functions.

What is EAI?

EAI, short for "enterprise application integration," is a framework that makes use of software and computer systems to enable data integration across applications within a single organization, while also simplifying business processes among connected applications and data sources.

There are three main types of EAI:

1. The Presentation Integration Model
2. The Data Integration Model
3. The Functional Integration Model<

The sharing of data and business processes between applications are its primary purposes. However, EAI also defines a set of principles for integration of multiple systems for communication architectures, such as message-oriented middleware (MOM).

EAI serves as a system that can provide a business service to simplify information data between diverse applications, which makes it possible to easily integrate them when needed.

This discipline of integrating applications and data within the enterprise has been a critical component of today's enterprise strategies. In fact, many vendors offer EAI suites that provide cross-platform, cross-language integration solutions.

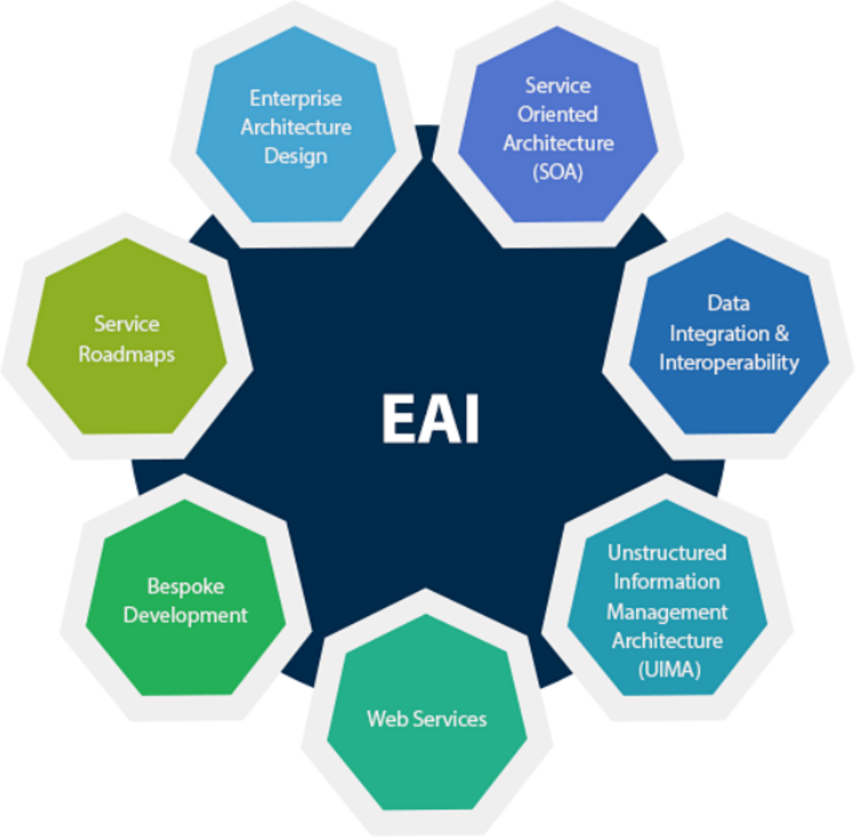
EAI
Enterprise Architecture Design
Service Oriented Architecture (SOA)
Data Integration & Interoperability
Unstructured Information Management Architecture (UIMA)
Web Services
Bespoke Development
Service Roadmaps

Figure 1: EAI Structure

What is SOA?

In contrast to EAI, which deals with linking enterprise applications so they can communicate with one another (by means of an intelligent reasoning engine) and carry out "batch" data transfers, service oriented architecture (SOA) provides "transactional" data transfers, with no third-party software required. SOA is different from the EAI approach in that it does not depend on a third-party solution.

Service oriented architecture (https://www.innovativearchitects.com/EDI-SOA-Consulting.aspx) is an integration paradigm that is based on a fundamental design principle and provides architectural interoperable services - hence its name. It encompasses legacy systems, software components and message processing capabilities.

SOA's primary function is the integration of services and the development of applications. Lately, SOA-based systems have become more common than ESB and EAI because of development technologies and platforms as well as the changes in traditional IT roles and responsibilities. SOA is the preferred solution for business logic and technology.

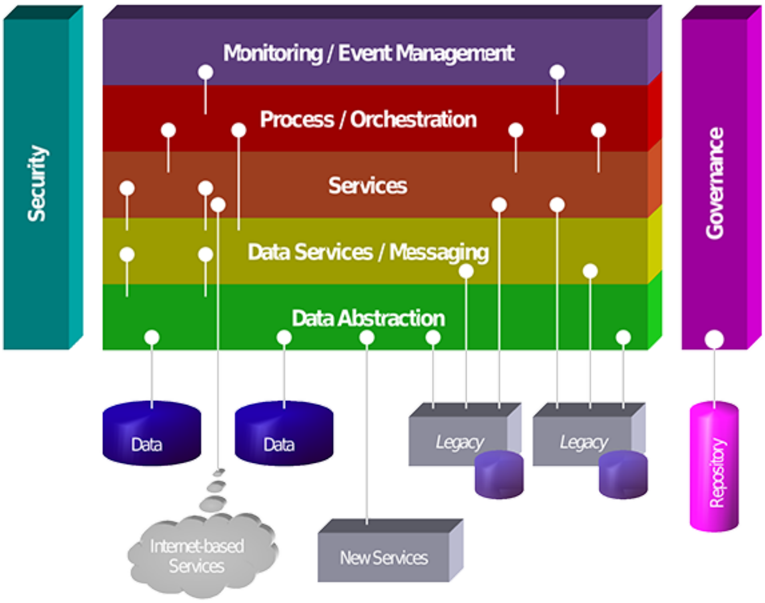

Figure 2: SOA Process

SOA represents distinct principles that foster service-orientation in loosely coupled systems (like ESB where it acts as a broker between services), network change management governance and has become an important aspect of Web 2.0 applications and services. It is a framework that can automate business processes while specifying service-oriented modeling business systems within an architectural application. It is used as a design scheme of application-level and enterprise-level solutions for modeling a computing environment.

SOA is able to help overcome all sorts of shortages in EAI and ESB technologies and provide a means for reusable services. It supports Distributed Enterprise Application Integration and “enterprise service bus” integration environments. The latter is the new software architecture (called “SOA-based enterprise service bus,” or simply ESBsoa) which is a Web services-based application supporting platform ideal for transport adaptation, service adaptation and common services.

What is ESB?

ESB, which is the abbreviation for “enterprise service bus,” is a software architecture that provides integration of enterprise applications and services for complex architectures, such as middleware infrastructure platforms.

An ESB's primary function is to provide the connections between communicating applications - acting much like a router to control the data. It is commonly used in enterprise application integration (EAI) or service-oriented architecture (SOA) principles. The interaction and communication between components are across the bus, which has a similar function as a physical computer bus to handle data transfer or message exchange between services without writing any actual code.

ESB as an infrastructure software service-oriented model works as a managed message system that provides routing, data transformation, translation upon a client's request and event-interpretation. It is often needed to transform messages into a format that the application can interpret. ESB is also used to change data content or execute services via a rule engine.

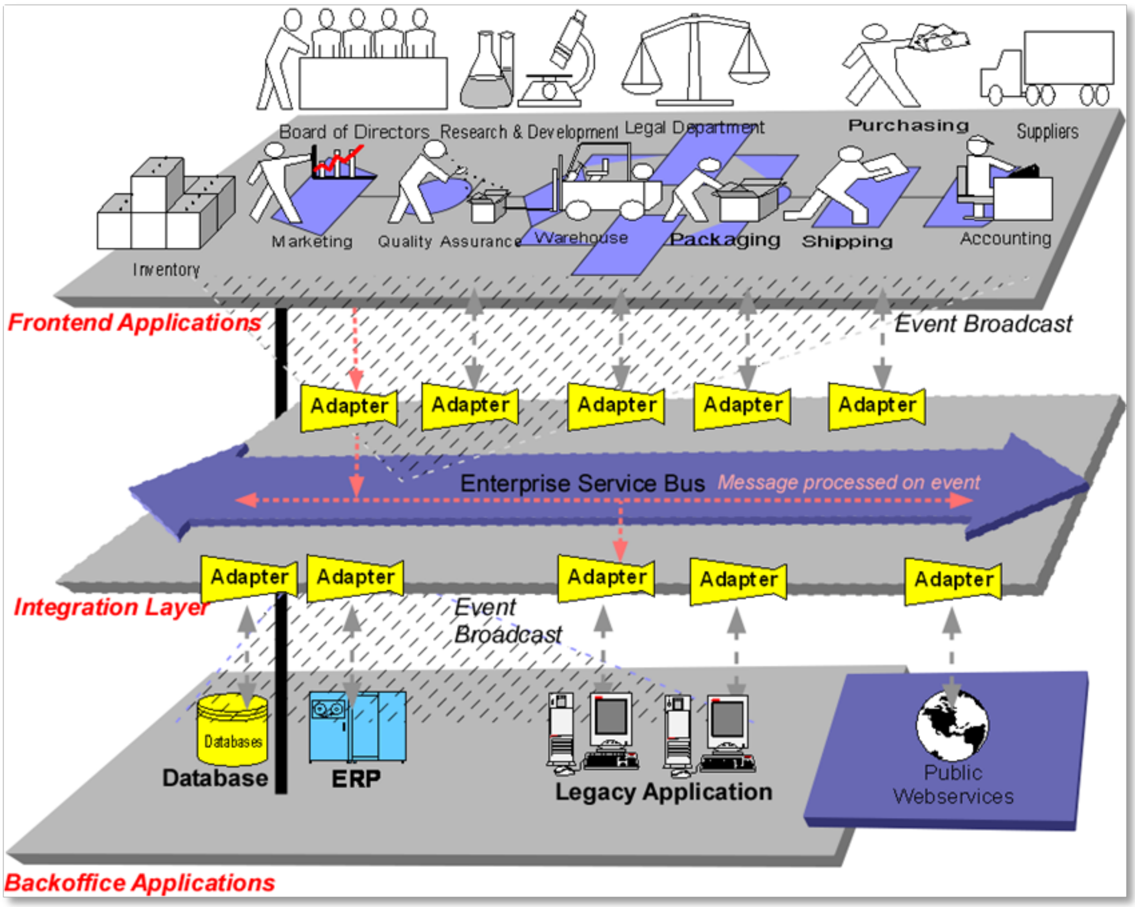

Figure 3: ESB Architecture

EAI, SOA or ESB: Which is Best?

By now, you should be able to differentiate the three concepts: EAI, SOA and ESB.

In deciding which one to implement, you should consider the pros and cons of each option. For instance, ESB offers a low cost as it is often used by those with limited IT resources. And even though some consider EAI to be a thing of the past, with SOA there are now packaged EAI solutions that might be appropriate for large scale integrations. SOA, on the other hand, offers better flexibility to adapt to changing requirements and technologies.

EAI **Enterprise Application Integration**	SOA **Service Oriented Architecture**	ESB **Enterprise Service Bus**

✓ Pros	- Easy to quickly accomplish goals - Guaranteed delivery - Simplifies distributed asynchronous computing and access to data sources - Greater flexibility - Allows reuse of data across other applications	- Reusability of services in other applications - Easy to update and maintain - Greater reliability and fewer software errors - Location independent - Scalable and available - Platform independence	- Less customization required - Consistency and good practice in application integration - Boost operational efficiency and agility by consolidating apps - Wide selection of tools - Easier operational support due to transparency
✗ Cons	- More complex architecture - Steeper learning curve - Performance bottlenecks possible - Difficulty accessing or maintaining business logic	- Increased overall costs due to higher response time and machine load - Complex service management - Significant upfront investment	- Risk of projects "bottlenecks" requires significant planning - Difficult to find developers with extensive ESB skills - Risk of regression - Single point of failure

To determine which option is best for you, discuss your individual business' needs with one of our experienced SOA consultants (https://www.innovativearchitects.com/soa-services.aspx) today. Contact us (https://www.innovativearchitects.com/ContactUs.aspx) for more information.

About

- Getting Started (/GettingStarted.aspx)

- Clients (/Clients.aspx)
- Client Locations (/ClientLocations.aspx)
- Partnerships (/Partner.aspx)

- News (/LatestNews.aspx)
- Press Releases (/PressReleases.aspx)
- Community Support (/CommunitySupport.aspx)

- Knowledge Center (/KnowledgeCenter/Default.aspx)
- Case Studies (/CaseStudies.aspx)
- White Papers (/WhitePapers.aspx)

Top ^

We partner with a diverse list of industry-leading providers so you always get the right solution

Microsoft GOLD CERTIFIED Partner

Microsoft Consulting & Professional Services

EMC² where information lives

Biml™

azure

SAP®

INFORMATICA®

DOTNETNUKE

Innovative Architects (http://local.innovativearchitects.com/duluth-ga/)

Atlanta Office

3122 Hill St.
Duluth, GA 30096 USA

Innovative Architects (http://local.innovativearchitects.com/washington-dc/)

D.C. Office

2101 L St NW Ofc 8106 Ste 800
Washington, DC 20037-1657 USA

Innovative Architects (http://local.innovativearchitects.com/west-palm-beach-fl/)

Florida Office

324 Datura St. Ste 300
West Palm Beach, FL 33401 USA