Getting started with Enterprise Application Integration (EAI) using Enterprise Service Bus (ESB)

ESB – The Next Step in EAI

- ESB - An attempt to move away from problems caused by brokered hub and spoke EAI approach.
- Bus architecture lessens the burden of functionality placed on the single hub by distributing the integration tasks to other part of the networks.

ESB – The Next Step in EAI

- –The granular loosely coupled components can be grouped in various configurations to handle different integration scenario. Can be hosted anywhere within the infrastructure or duplicated for scalability.
- –Componentize necessary functionalities like security, transaction processing, error handling

Core ESB features

- There are a number of different ESB products available on the market today. Some, such as WebSphere Message Broker or TIBCO Business Works, are traditional EAI products that have been re-factored to offer ESB-like functionality, but still function in a broker-like manner.

Core ESB features

- **Core features**
- Location transparency
- Transformation – usable formats for all consumers
- Protocol conversion – Accept all protocols for consumption
- Routing – determine appropriate end consumer based on preconfigured rules or dynamic created requests.

Core ESB features

- Monitoring/Administration
- Security – ESB security involves tow main components
  - Ensure secure handing of messages
  - Ensure secure transport of messages

Core ESB features

- Connecting Anything to Anything

  Transports: HTTP, HTTPS, POP, IMAP, SMTP, JMS, AMQP, FIX, TCP, UDP, FTPS, SFTP, SMS

  Formats & protocols: JSON, XML, SOAP 1.1, SOAP 1.2, WS-*, HTML, EDI, HL7
- Routing, Mediation & Transformation

  Routing: Header based, content based, rule-based and priority-based routing

Core ESB features

Mediation: EIPs (including scatter/gather, message filters, recipient list, dead-letter channels, guaranteed delivery and message enrichment), database integration, event publishing, logging & auditing, validation

Transformation: XSLT 1.0/2.0, XPath, XQuery, Smooks

Core ESB features

- Message, Service, API & Security Gateway

  Expose existing applications & services over different protocols & message formats

  Virtualize services for loose coupling & SOA governance

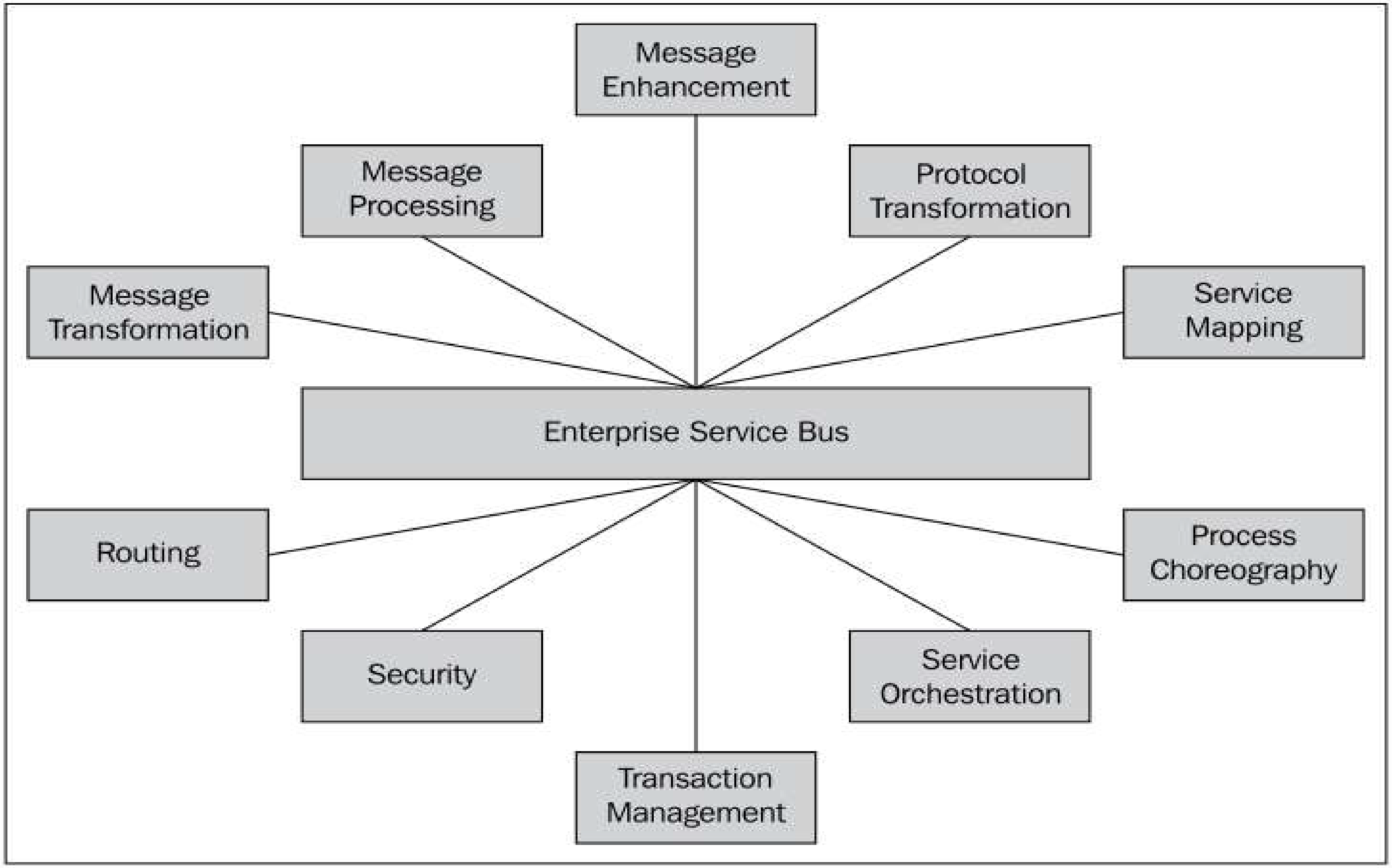
  Load balancing for scalability and fail-over for high availability of business endpoints

  Create service facades for legacy / non-standard services

Core ESB features

- SSL tunneling and SSL profiles support for inbound and outbound scenarios
- CRL/OCSP Certificate revocation verification

Core ESB features



ESB Key Components

1) Message Oriented Middleware (MOM)
   - Robust, reliable transport
   - Efficient movement of data across abstract data channels
   - End-to-end reliability
2) Service Container and Abstract Endpoints
   - Endpoints
     - Logical abstraction, representing remote services in various implementations
   - Container
     - The physical manifestation of the endpoints
     - Distributed and lightweight

ESB Key Components

3) Intelligent routing

- Message routing based on content and context
- Message routing based on business process rules
- Business process orchestration based on a rules language such BPEL4WS

Advantages of ESB

- **Lightweight:** because an ESB is made up of many interoperating services, rather than a single hub that contains ever possible service, ESBs can be as heavy or light as an organization needs them to be, making them the most efficient integration solution available.
- **Easy to expand:** If an organization knows that they will need to connect additional applications or systems to their architecture in the future, an ESB allows them to integrate their systems right away, instead of worrying about whether or not a new system will not work with their existing infrastructure. When the new application is ready, all they need to do to get it working with the rest of their infrastructure is hook it up to the bus.

Advantages of ESB

- **Scalable and Distributable:** Unlike broker architectures, ESB functionality can easily be dispersed across a geographically distributed network as needed. Additionally, because individual components are used to offer each feature, it is much simpler and cost-effective to ensure high availability and scalability for critical parts of the architecture when using an ESB solution.
- **SOA-Friendly:** ESBs are built with Service Oriented Architecture in mind. This means that an organization seeking to migrate towards an SOA can do so incrementally, continuing to use their existing systems while plugging in re-usable services as they implement them.