Enterprise Integration Options

Several options exist for integrating applications within the enterprise. This section gives a brief outline and comparison of the major options.

Option 1: Custom Point-to-Point Integrations

Definition: A direct point-to-point link is created between applications for each business function

Attributes:

- Designed and implemented solely for the purpose of directly integrating two specific systems
- Custom code for
  - Data extraction
  - Business rule processing
  - Data loading
- Custom data format
- Inconsistent transmission protocol
- Used for both real time and batch integrations

Advantages:

- No need to invest in expensive tools up front
- No need for developers to learn new skills and packages
- No extended time frame for developing and deploying enterprise integration strategy

Option 2: Messaging or EAI (Enterprise Application Integration) Tools

Definition: Source systems "publish" enterprise messages to a common bus; application "subscribe" to relevant messages and act on them

Attributes:

- "Wraps" each application
- Acts as a broker between applications
- Provides near real-time, guaranteed, once-only delivery
- Stores and forwards messages
- Provides an environment in which to define rules

Advantages:

- Systems are integrated but not coupled
- Business rules are centralized in the message broker and transformation engine
- Allows for near real-time integrations which reduced latency
- Solves the n^2 problem; as the number of systems increases, the integration effort expands linearly

Option 3: Web Services

Definition: Functionality to be integrated is exposed via XML on an open protocol such as SOAP. Other systems can consume this service if needed. Inputs and outputs to the web services are XML

Attributes:

- Common language of communication across heterogeneous systems
- Based on standard Internet technologies
- Self describing and advertising
- Supports dynamic discovery & integration
- Services fit within an overall architectural model
- Widely supported by major vendors

Advantages:

- Solves problems similar to those EAI solves, as well as
- Need for expensive integration tools
- Use of proprietary integration platforms

Option 4: ETL (Extract, Transform, Load) Tools

Definition: A standard set of tools and processes used to extract, transform and load large volumes of data between systems. Very useful in populating a data warehouse

Attributes:

- Provides tools for data cleansing; correcting misspellings, resolving conflicts (city & zip code incompatibilities), missing elements, parsing elements
- Can combine data sources: Matching on key values, fuzzy matches on non-key attributes, textual comparisons to reference tables
- De-duplicate processing: Identifying and eliminating duplicates
- Can create surrogate keys: Operational systems and the data warehouse have different assumptions and data requirements thus the data warehouse requires its own set of primary keys
- Create aggregates to boost performance of common queries in data warehouses and data marts
- Loading and indexing: For large data warehouses specialized bulk loading processes are required

Advantages:

- Extremely efficient for moving large volumes of data in short timeframes
- Applies consistent transformations
- Can provide or integrate with meta-data for the enterprise data model

Comparison of Options

	EAI	Point to point	Web Services	ETL
Concept	• Publish/Subscribe mechanism • Most suitable for real time data needs • Loosely coupled	• Custom code for each integration need • Suitable for complex integration needs • Tightly coupled	• Standards based integration • Most suitable for inter-organization integration • Loosely coupled	• Suitable for large volumes of data • Generally used to move data between two or more databases
Strengths	• Reliability (guaranteed delivery) • Enables real-time business decisions • Out of box adapters for many enterprise systems	• Familiar technologies and processes • Many point to point integrations already exist • No major up front investment required	• Standards based integration • High degree of reuse • Wide tool support including open source • Low up front investment	• Metadata driven approach • GUI tools for most tasks (little coding) • Extremely efficient for large data volumes
Weaknesses	• High upfront cost • Relatively complex design patterns	• Costly over time • Tight coupling • Scalability issues • Opportunities for reuse are slim	• Lack of transaction support • Not a publishing model • Less established technology	• High upfront costs • Complexity of tool • Batch oriented
When to Use	• Real time data is important • High volume, low footprint data exchange • Many consumers of the same data	• Should be rarely used • When defined enterprise strategy cannot work • Proto typing	• Integration model is request/reply • Real time requirements • High volume, moderate data	• In conjunction with a data warehouse