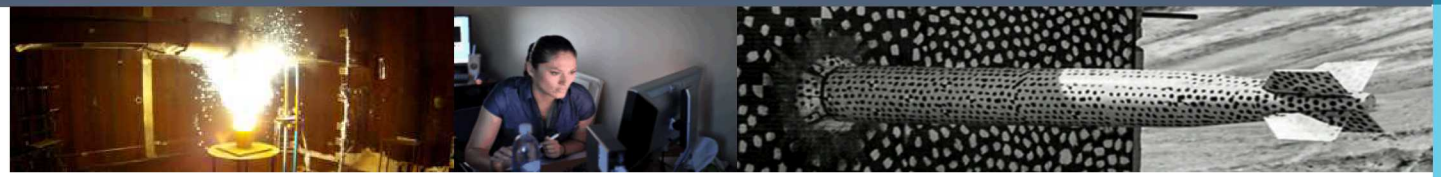
SAND2020-11037C

Enterprise integration

Why can't these applications ever play nice?



PRESENTED BY

Click to edit | Peter Chandler, R&D Systems Architecture, pechand@sandia.gov

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Agenda

Today's Enterprise. "The Challenge"

Integration Principles

Some Best Practices "Integration Manifesto"

Business Patterns

Enterprise Integration Patterns EIP

Integration Technologies, EIP Implementations

Today's Enterprise. "The Challenge"

- **Enterprise integration** is the task of making separate applications work together to produce a unified set of functionalities. (aka. composite application)
- Applications may be custom developed in-house, while others are bought from third-party vendors.
- Applications may be running on multiple systems, with different platforms/OS's, and geographically dispersed.
- Applications may be run outside of the enterprise by business partners or customers.
- Applications may need to be integrated, even though they were not designed for integration and cannot be changed.

Today's Enterprise. "The Challenge"

Large established organizations have an evolution of point-to-point integrations, aka Spaghetti integration, where systems integrate directly with each other.

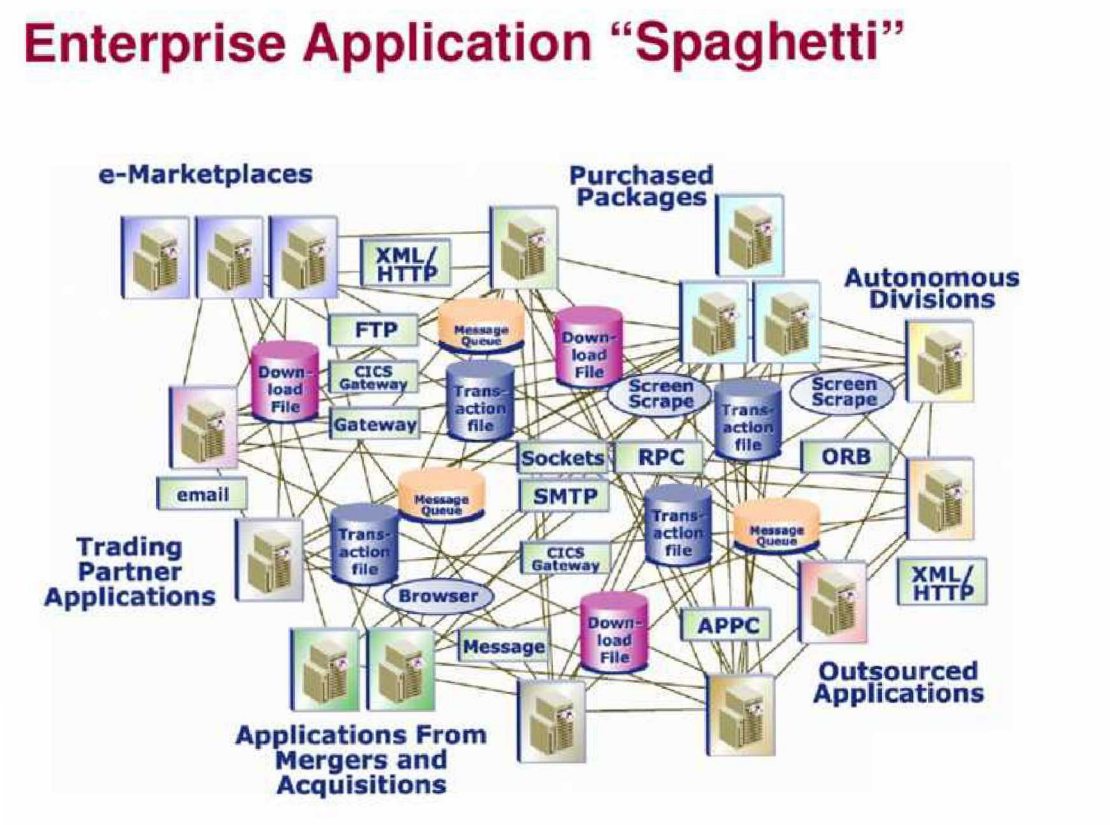
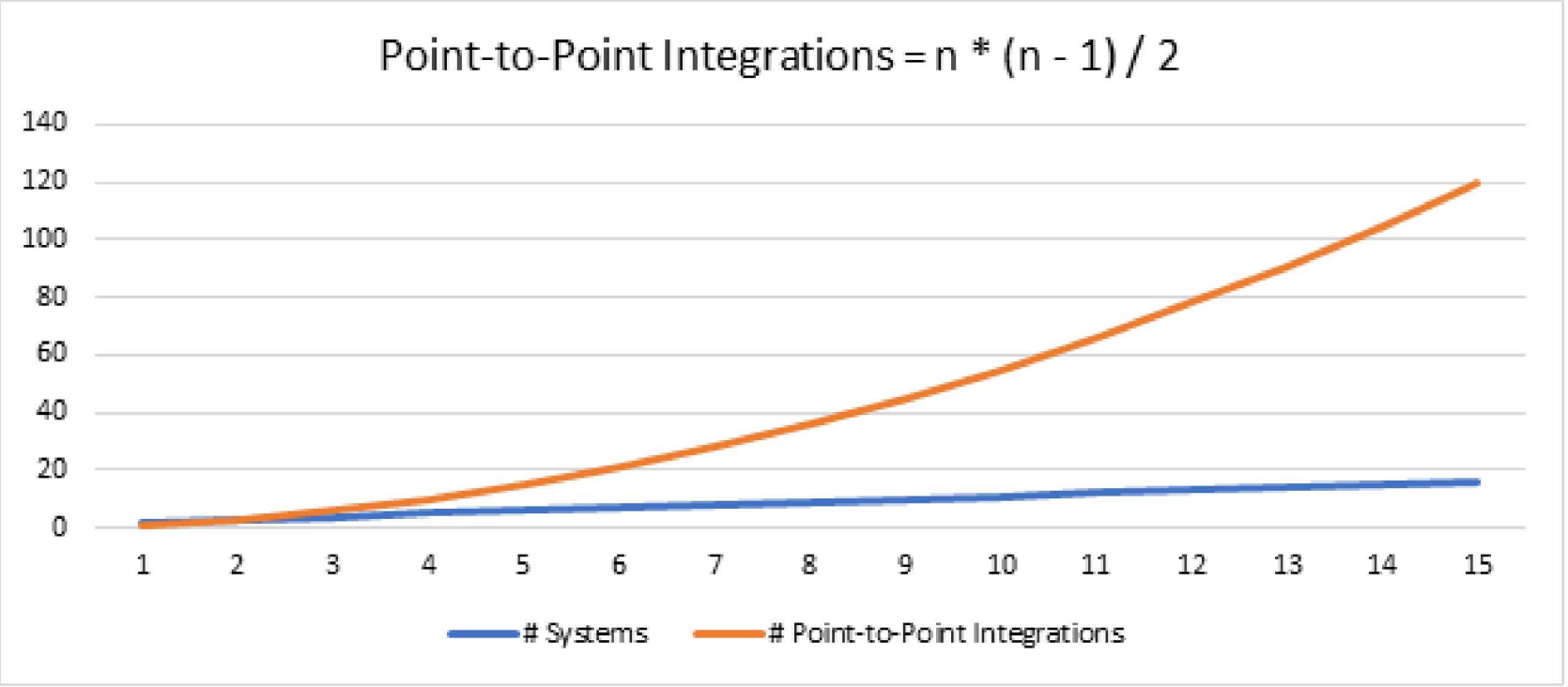


Image Source: Service Oriented Architecture Based Integration, Michael Rosen [Slideshare]

Today's Enterprise. "The Challenge"

The n(n-1) rule for point-to-point integration.

While adequate for simple integration, this model is quickly unmanageable for more extensive integration requirements because of the n(n-1) connections rule (also referred to as the n-squared problem).



Integration Principles

- Remote/Programmatic access to the data. What API/Protocol do I use to get at the data? Example: REST.
- Understanding the meaning of the data is critical. How do individual fields "map" from one system to another? Example: Ontology, Semantic mapper, and what makes a "record" unique?

Some Best Practices "Integration Manifesto"

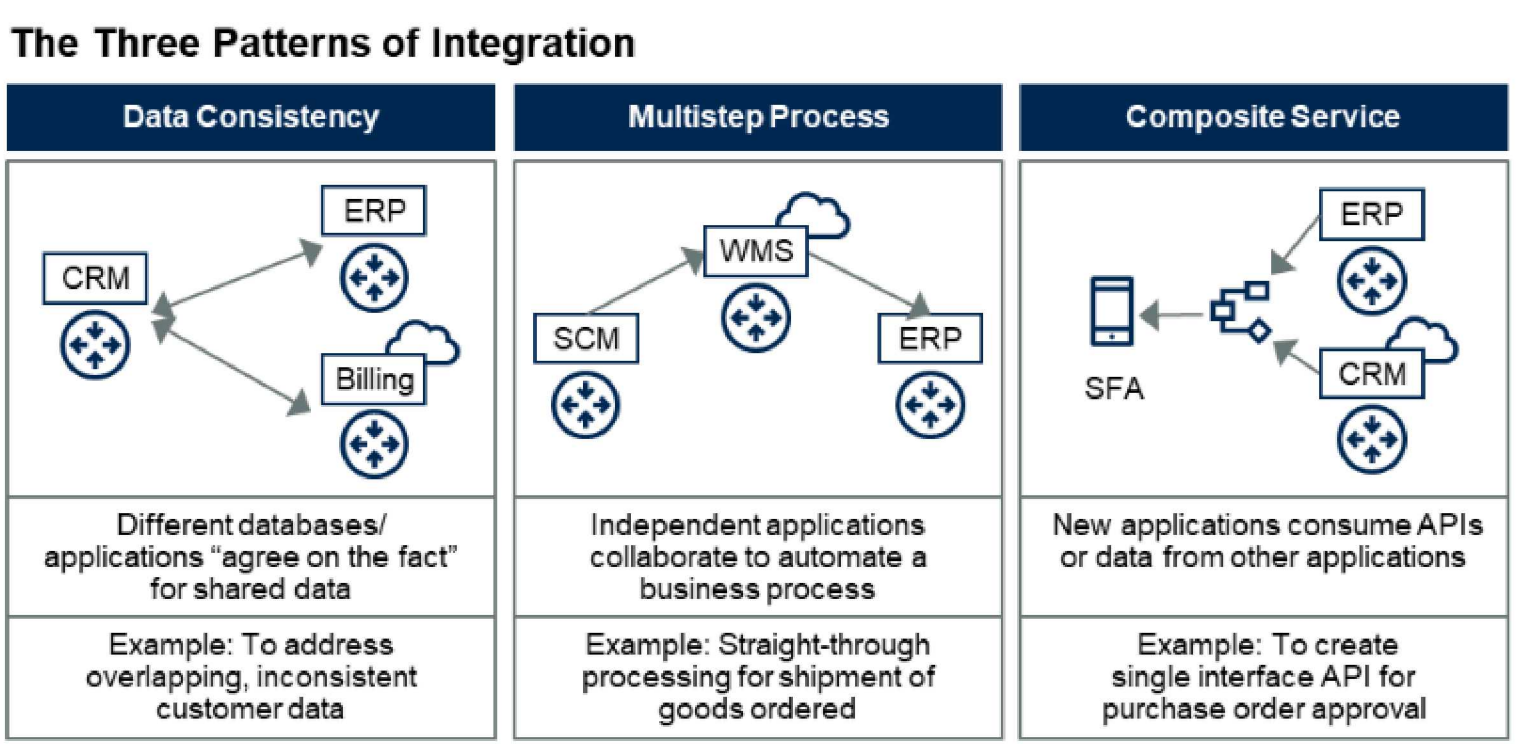
1. No one domain, organization, or application owns data. The Parent Organization is the data owner. i.e., "Free the data." All domains are responsible for sharing data with other need-to-know domains.
2. Avoid creating more static copies of the data. Example: warehouses, repository, or DataMart.
3. Data persists within a Domain and its applications. The tool or application that creates the data should store it when possible.
4. The "system of record," i.e., truth for data (authoritative source) can vary, as it is a function of the product development phase. There should be one and only one "system of record" at a given time.
5. Applications (COTS, GOTS, custom-developed) should provide a standard based external (network) interface. Some standards to be considered are: REST, SOAP, OpenAPI, OSLC, RFD, ISO/IEC JTC 1/SC 7, STEP AP 239, FMI, XMI
6. All new integrations should be M x N, i.e.; multiple different applications can pull data and push data. Consider Publish/Subscribe pattern. Avoid point-to-point solutions.
7. All data should be in a neutral/universal/canonical form that is self-describing. Consider the RDF format for application data.

Some Best Practices "Integration Manifesto"

8. Enterprise Integration Patterns EIPs should be used whenever possible.
9. When data is published or consumed, it should be validated (cleaned).
10. Data transfer patterns should be defined: on-demand (synchronous), scheduled, and event-based (asynchronous).
11. Distributed transactions require integrations compensating transactions or XA support.
12. Information transfer should be monitored, have tracing, and failure detection.
13. Avoid database-level integration. Use the vendor's API.
14. Integrations should be versioned.

Business Patterns

- Data Synchronization – Keep the data in two or more systems the same.
- Workflow/orchestration – Multiple Systems/Applications collaborate to perform an automated multi-step business process.
- Data Aggregation – A system consumes data from multiple other systems (data sources).

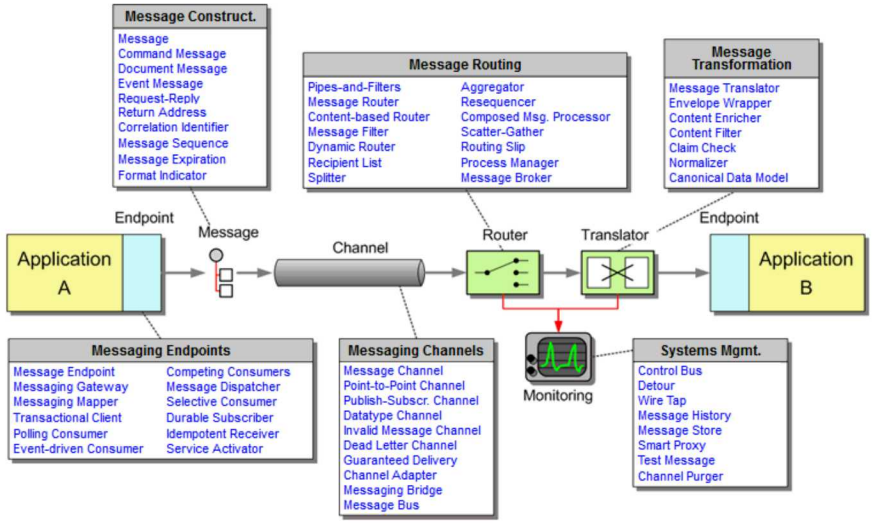


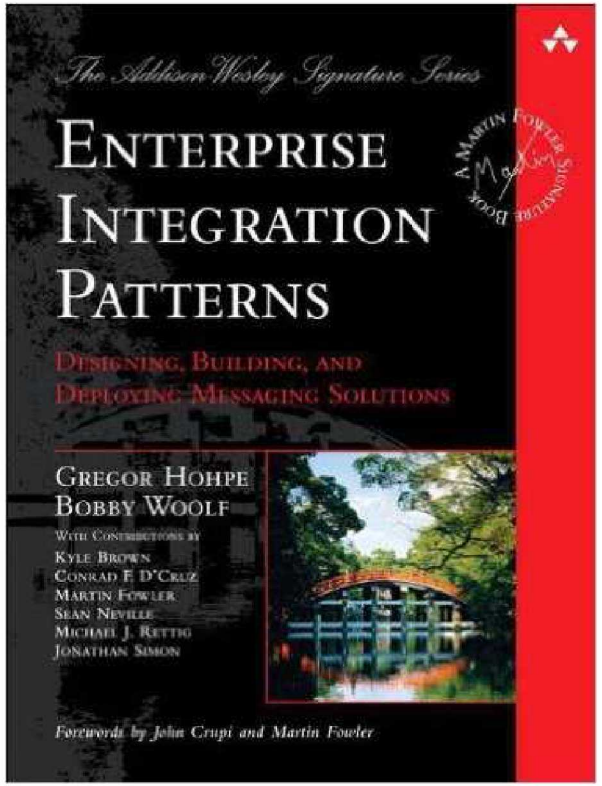
Source Gartner: Choose the Best Integration Tool for Your Needs Based on the Three Basic Patterns of Integration Published: 20 September 2019

Enterprise Integration Patterns EIP

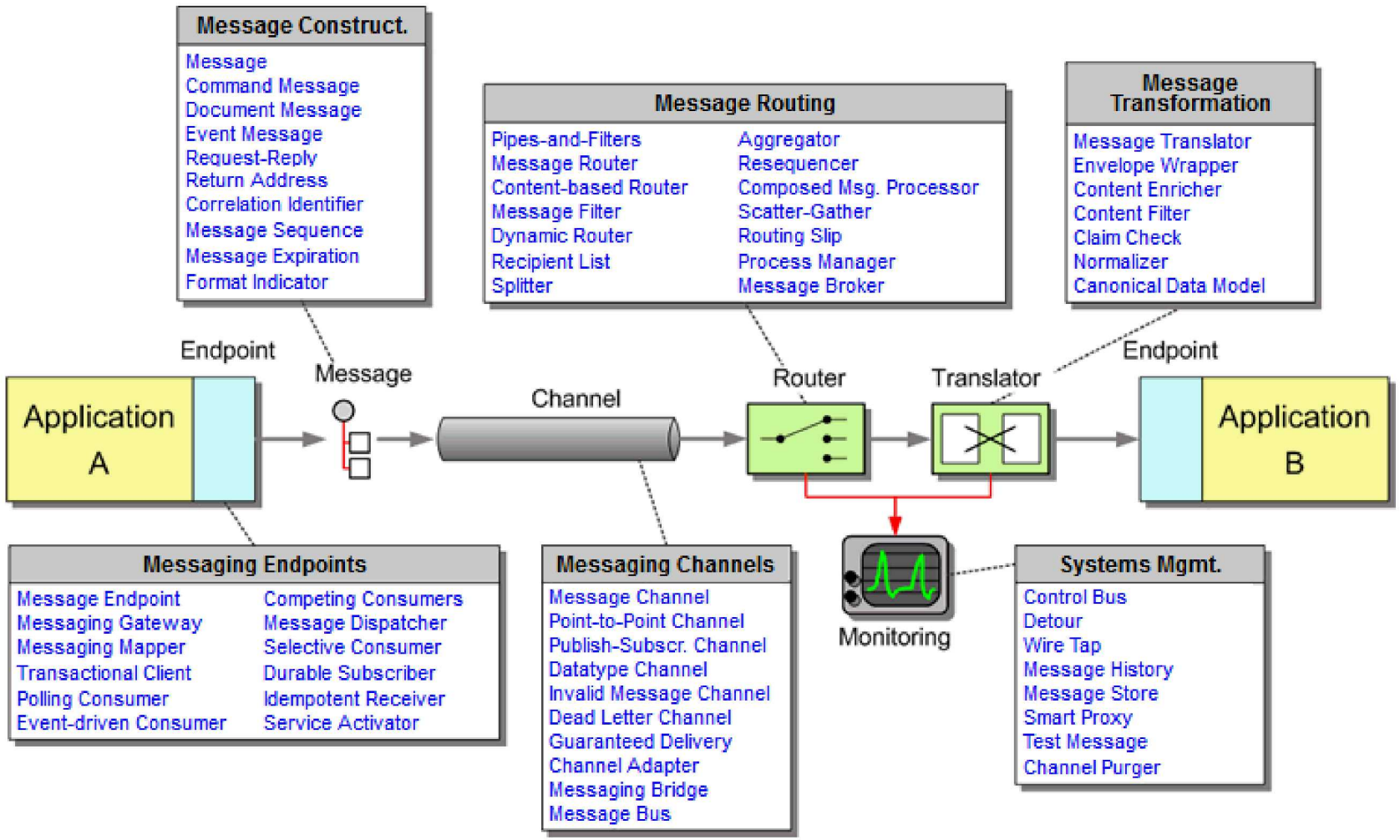
What is it?

- EIP is a set (65) of technology-independent/agnostic design patterns for implementing system/application Integration.
- EIP is intended to orchestrate business transactions.
- Data mediation propagates changes, or new information flows from one application to another without manual intervention.
- The design patterns provide a well known/documented methodology for Integration.
- The patterns are reusable and solve common integration problems.
- Using patterns, one creates a common approach to all integrations and avoids "one-off" or "point-to-point" solutions.
- The patterns support near real-time data exchange.





Enterprise Integration Patterns EIP



Source: https://www.enterpriseintegrationpatterns.com/patterns/messaging/

Enterprise Integration Patterns EIP

How do EIPs compare to other Technologies? 1/2

Data virtualization tools.

- Generally, data is pulled from different data sources and aggregated together
- Data Virtualization is not commonly used for:
  - Data source updates
  - Near real-time application updates
  - Event-Driven (asynchronous) updates
  - To orchestrate transactions (XA) across multiple systems

Extract, Transform, Load (ETL), Federated database system, Data Warehouse

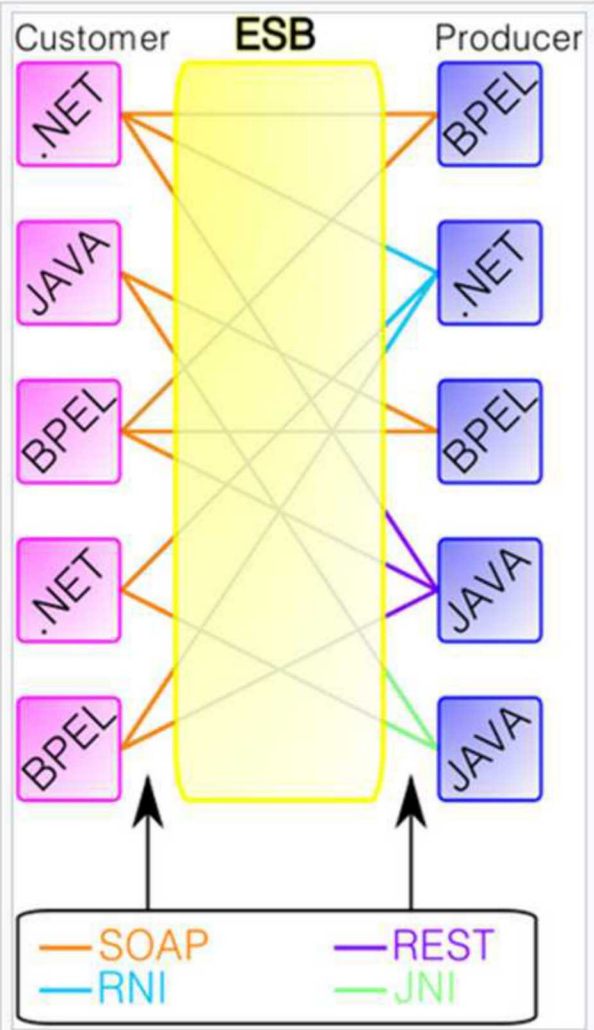
- Collect and store information, which results in duplication of data.

Enterprise Integration Patterns EIP

How do EIPs compare to other Technologies? 2/2

Enterprise Service Bus (ESB)

- Heavyweight platform.
- Proprietary, vendor-specific API/implementation.
- A significant learning curve and is still challenging to use.
- Vendor lock-in.
  - Usually requires significant Vendor services to implement and use
- Commercial versions cost big $$$ to implement and have a high Total Cost of Ownership.



All customer services communicate in the same way with the ESB: the ESB translates a message to the correct message type and sends the message to the correct consumer service.

Integration Technologies, EIP Implementations

Notable EIP implementations include:

- **Spring Integration**
- **Apache Camel**
- Red Hat Fuse
- Mule ESB
- Guaraná DSL

Enterprise integration
Why can’t these applications ever play nice?