Introducing BizTalk

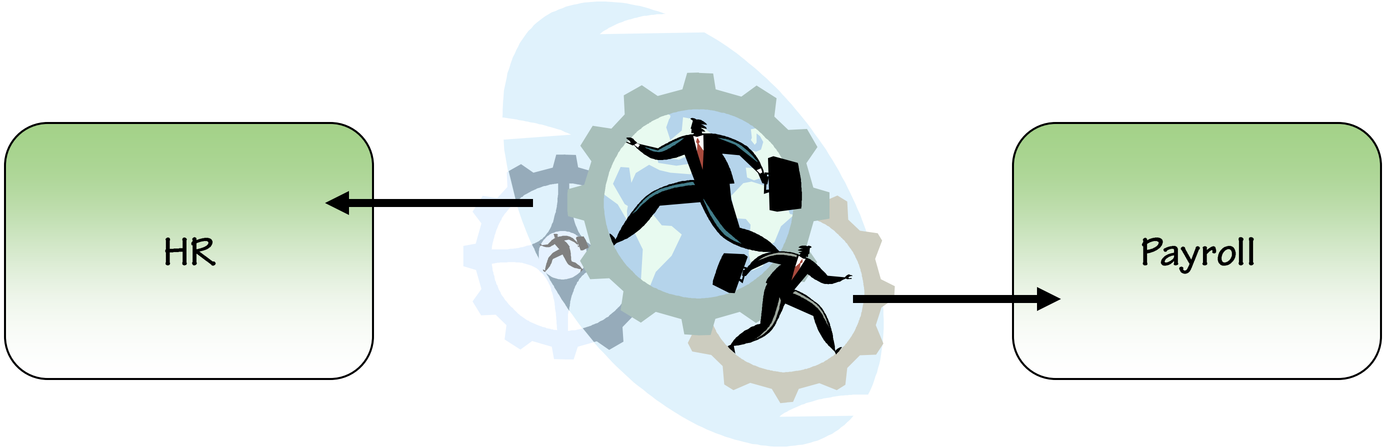
The thinking behind BTS 2006

Outline

- Fundamental integration principles
- Integration architecture patterns
- Introducing BizTalk Server 2006

The need for integration

- **For years companies have purchased and written applications**
  - Each application helps fulfill specific business processes
  - Valuable business data stored within each application
- **Business processes and data have not been easy to share**
  - Humans often needed to broker such sharing

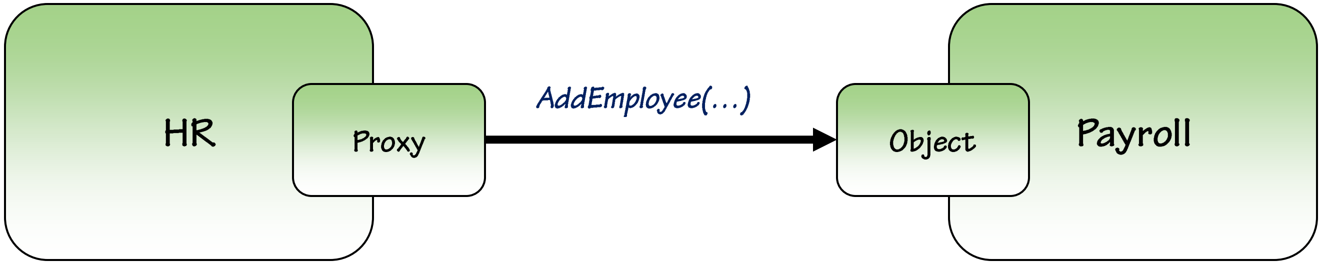


Integration principles

- *Integration* is about sharing *data* and *business processes*
  - Referred to as EAI within an organization
  - Referred to as B2B across organizations & trading partners
  - The issues are similar for both scenarios
- **One of the most costly and troublesome areas of IT**
  - And it's only going to get worse
  - IT explosion constantly compounding the problem

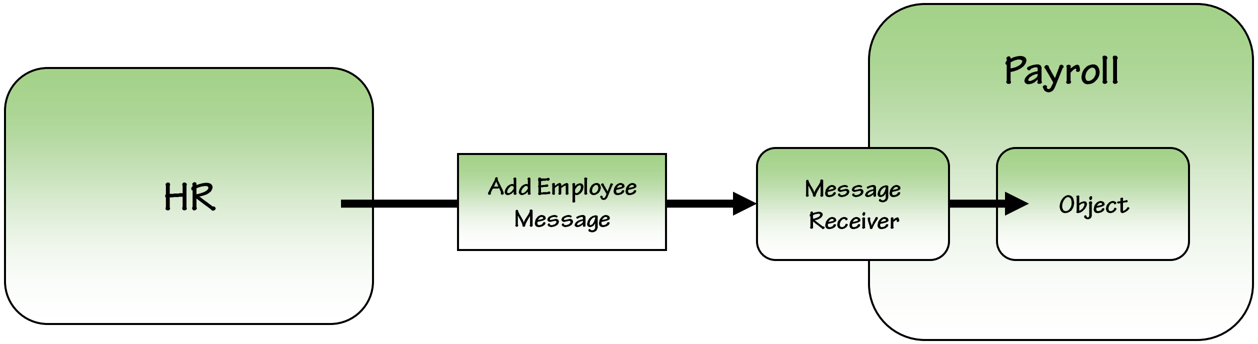
RPC-based integration

- **One approach is to leverage an application-specific API**
  - Applications expose processes and data via objects/rpcs
  - Consumers call those objects directly (DCOM, Corba, RMI)
- **Requires both sides to agree on RPC technology**
  - Requires intimate knowledge of app's inner-workings
  - Easily "breaks" whenever application API changes



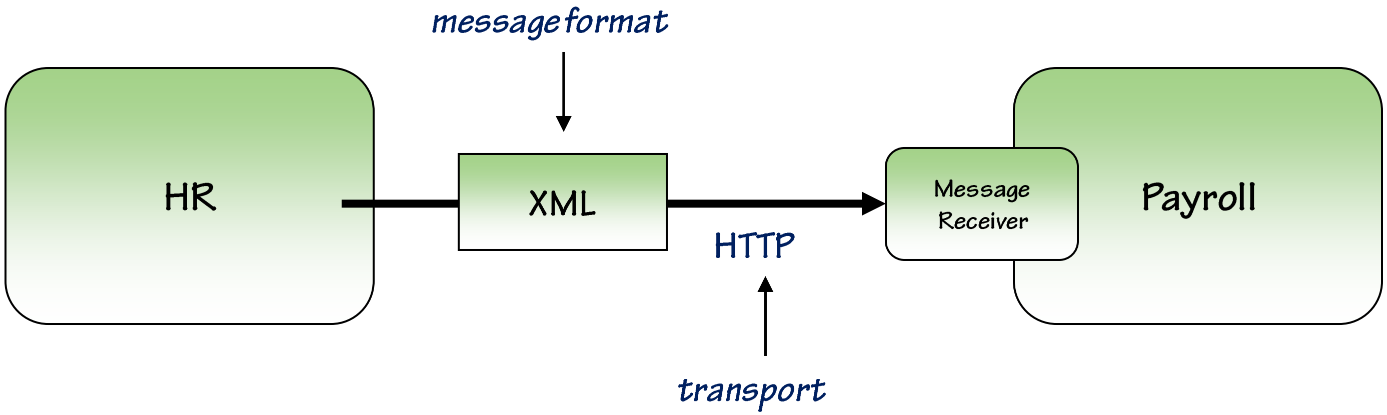
Message-based integration

- **A better approach is to exchange *messages* between apps**
  - Applications expose *locations* for receiving messages
  - Consumers send messages to those locations
- **Loosens coupling, offers technology freedom on each side**
  - Shields consumers from inner-workings and internal change



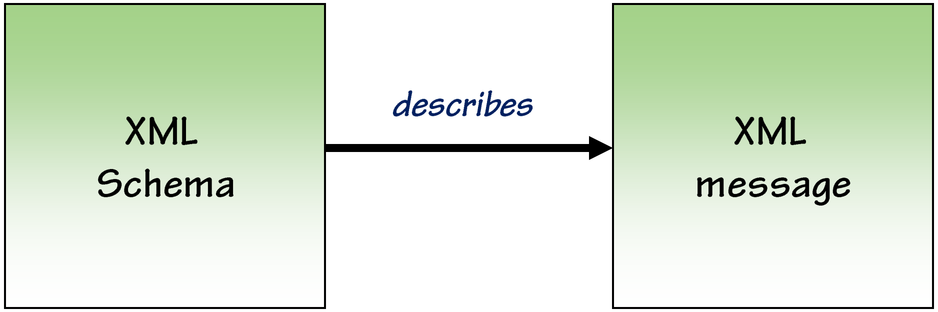
Messaging fundamentals

- **Applications must agree on *message format* and *transport***
  - Format defines message syntax/encoding: XML, EDI, CSV, etc.
  - Transport transmits messages: HTTP, FTP, MSMQ, BAPI, etc.
- **A *schema* defines the content of a particular message type**



Defining message schemas

- **You use a schema language to define the messages in use**
  - Use *XML Schema (XSD)* to define XML message types
  - Other schema languages exist for non-XML formats
- **Schema tells application what to send, and what to expect**

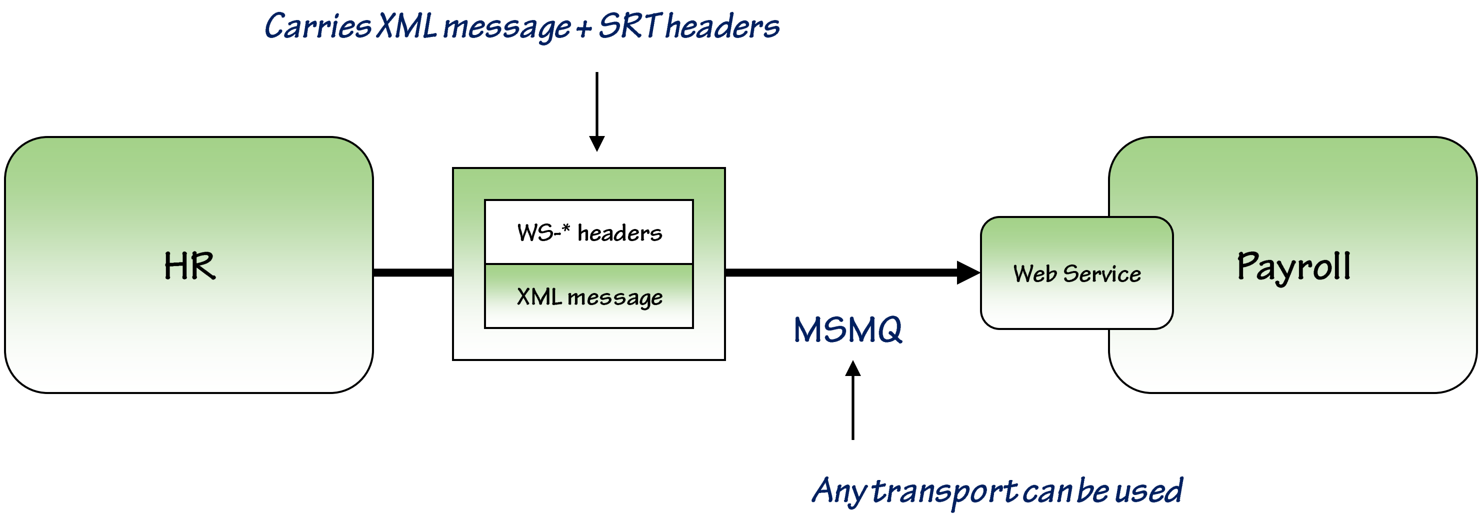


Transports, formats, and schemas

- **Integration is about making it work, no matter what's in use**
  - Architectures require wide support for common choices
  - If what you need isn't provided, you either buy or build it
  - Leveraging standard choices simplifies future integration
- ***XML* and *Web services* have emerged as *standards* today**
  - XML and XSD reduce format complexity
  - Web services reduces transport complexity

Web services

- **SOAP defines a standard framework for XML messaging**
  - SOAP messages carry XML payload + XML headers
  - WS-* defines additional protocols for use in SOAP
- **Web services normalize away transport specific solutions**



Web services are not a silver bullet

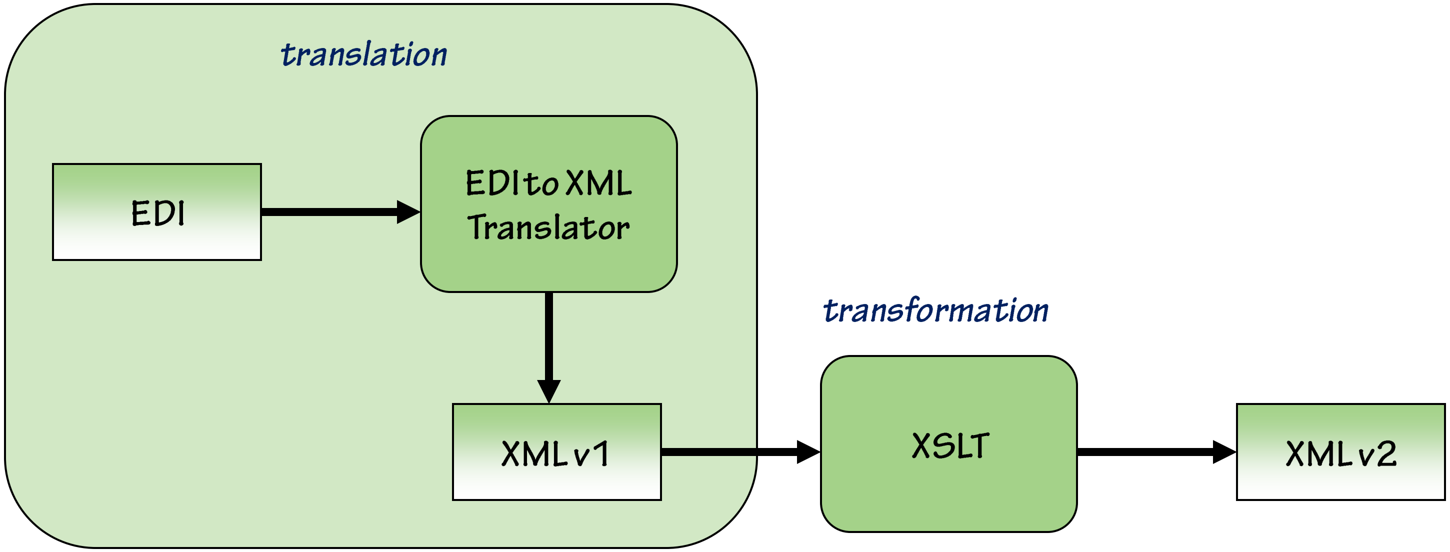
- **Web services are not a silver bullet for integration**
    - Can’t throw away existing investments in EDI and other formats/transports
    - Not everything can use XML, SOAP or WS-*
    - Many stacks only support HTTP today
- **Plus, integration is about more than just format/transport**

Integration realities

- **Applications don't usually agree on:**
  - A single message format (not even SOAP)
  - The message schemas
  - A single transport solution
  - How to handle security
- **Integration requires bridging these gaps**

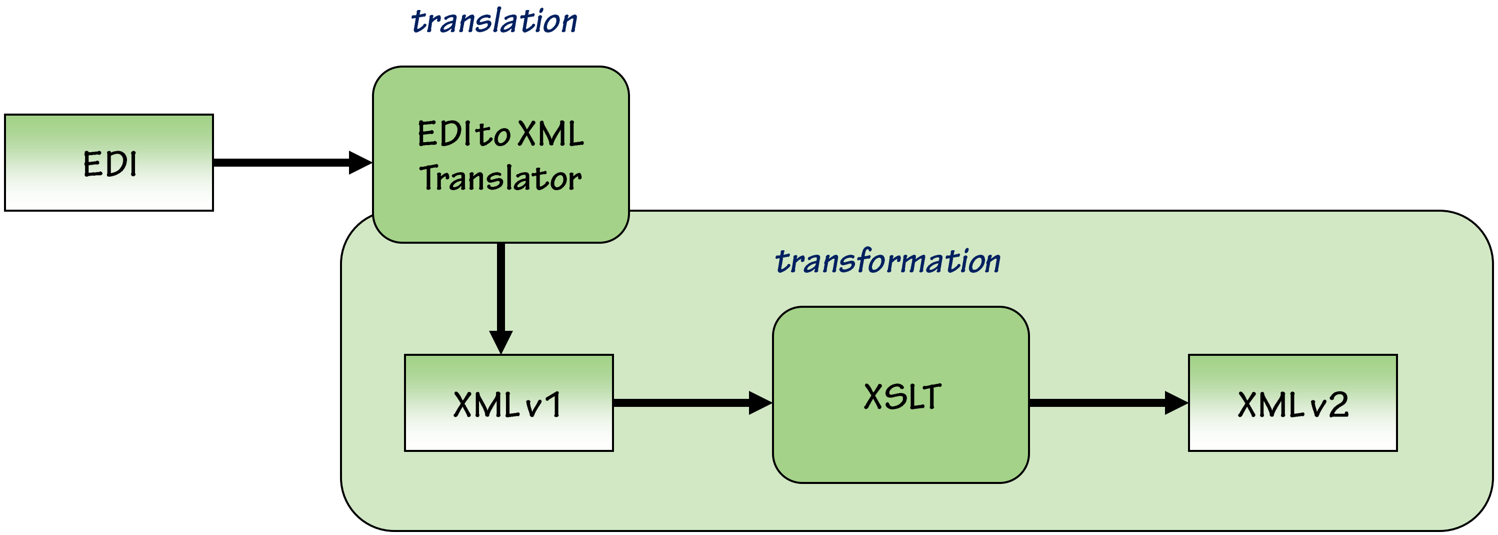
Supporting multiple message formats

- **Integration requires supporting multiple message formats**
  - Some applications are not capable of working with XML
  - Consider applications that deal with EDI or flat files
- **You can *translate* non-XML formats into XML formats**



Supporting multiple message schemas

- **Integration requires supporting multiple formats/schemas**
  - Applications won't always agree on same schemas
- **Message *transformations* move between schemas**
  - *XSLT* makes it easy to define XML transformations

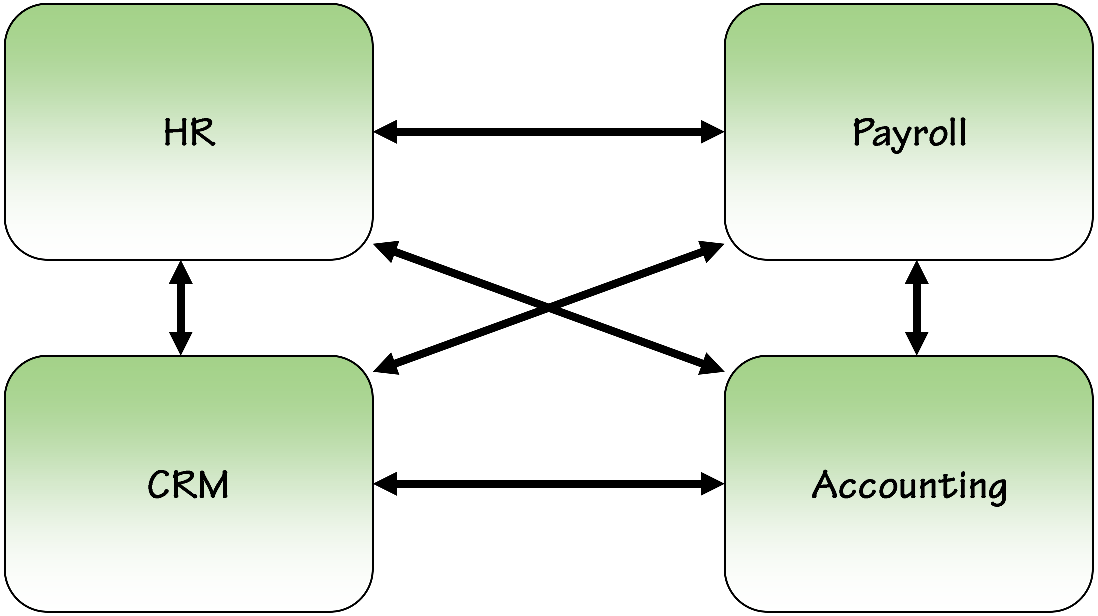


Supporting multiple security models

- **Applications won't always agree on how to handle security**
  - Transport vs. message
  - CIA (confidentiality, integrity, authentication)
  - Authorization scheme
- **Each transport comes with its own security mechanisms**
- **Integration requires managing the differences**
  - Mapping credentials across applications/transports

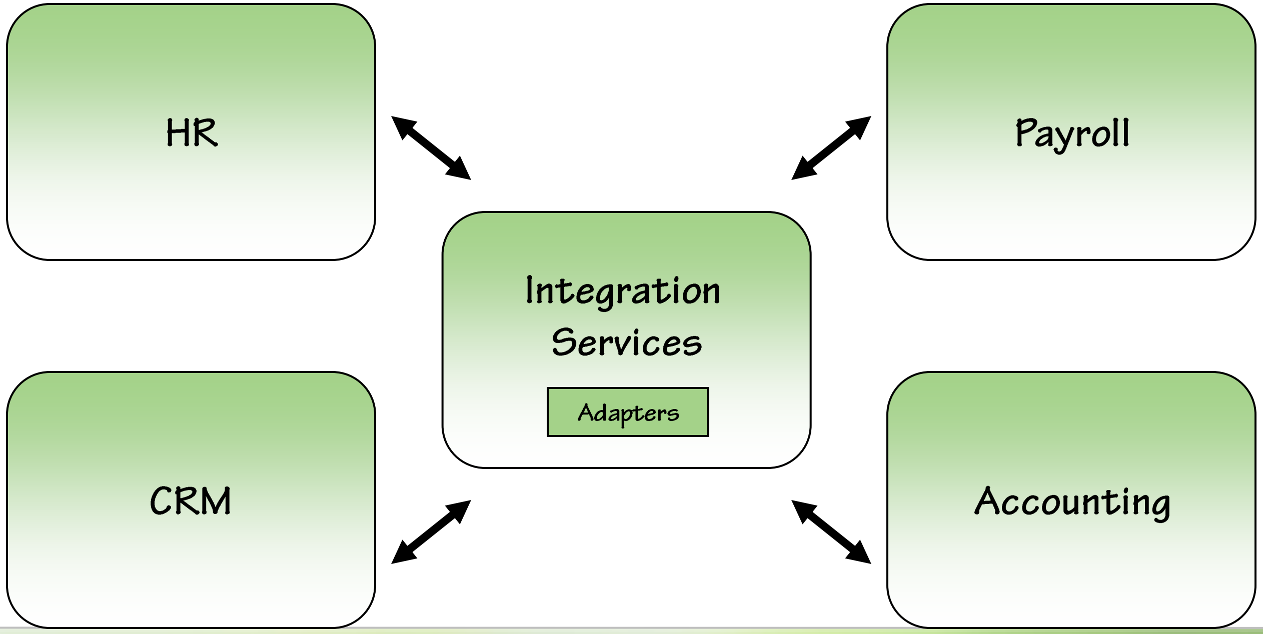
Integrating multiple applications

- **The challenge gets worse the more applications you integrate**
  - Each application has to deal with all variations
  - Transport, format, schema, and security permutations



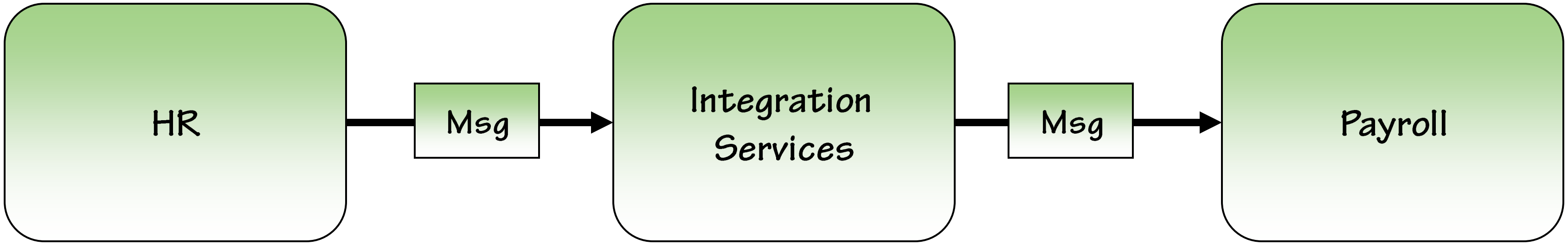
Centralized integration services

- **Centralized *integration services* reduces overall complexity**
  - Adds a level of indirection between applications
  - *Adapters* connect applications to integration services



Message routing

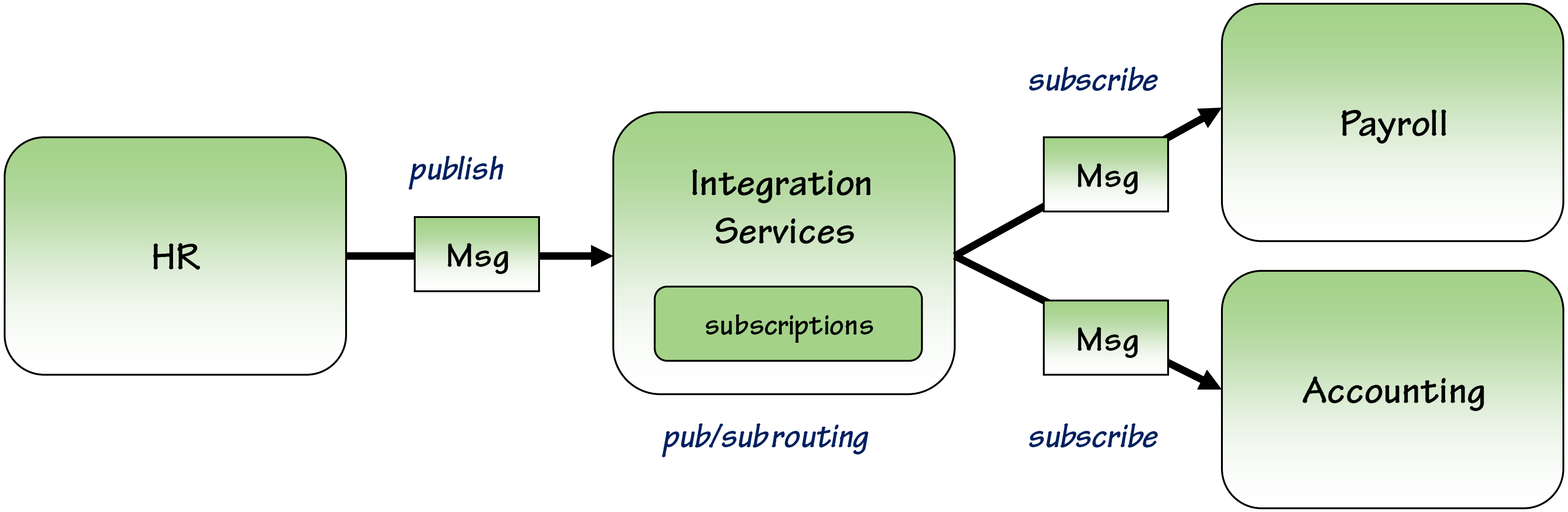
- **Centralized integration services require *message routing***
  - □ In order to route messages from one application to another
- **Numerous message routing implementation patterns exist**
  - □ Simple rules based on application names
  - □ Content-based routing provides more flexibility



simple message routing

Publish and subscribe

- **Content-based routing facilitates *publish/subscribe* pattern**
  - □ Receivers *subscribe* to certain messages, matching criteria
  - □ Senders *publish* messages to the integration services
  - □ Integration services route based on *subscriptions*
- **Decouples senders from receivers, adds significant flexibility**



Business processes

- **Publish and subscribe is about a single point of integration**
  - A single message interaction across applications
- **Most *business processes* consist of multiple interactions**
  - Across applications, trading partners, or people
  - The overall process is often *long-lived*

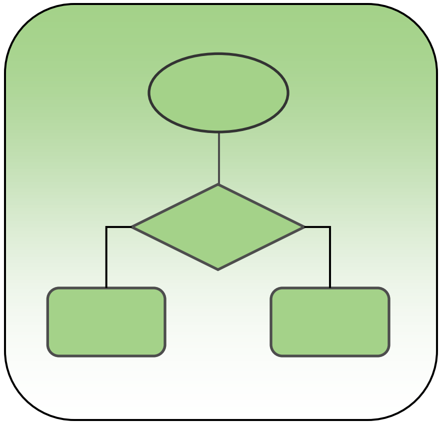
Sample business process

1. *Receive invoice message at FTP site*
2. *Transform invoice to the schema required by internal invoicing application*
3. *Send new invoice message to invoicing app*
4. *Wait for response message at queue*
5. *If Amount < 1000 and Status = Approved*
   - *Create message for accounting and send*
6. *Create email message with results and send*

Business process integration

- **Automating business processes across applications is referred to as *business process integration* (BPI)**
  - Requires a language (designers) for defining processes
  - Requires a *runtime* that manages process execution
- **Key runtime features:**
  - A scalable *state management* strategy
  - Message *correlation*
  - Long-running and *compensating* transactions

Defining a business process



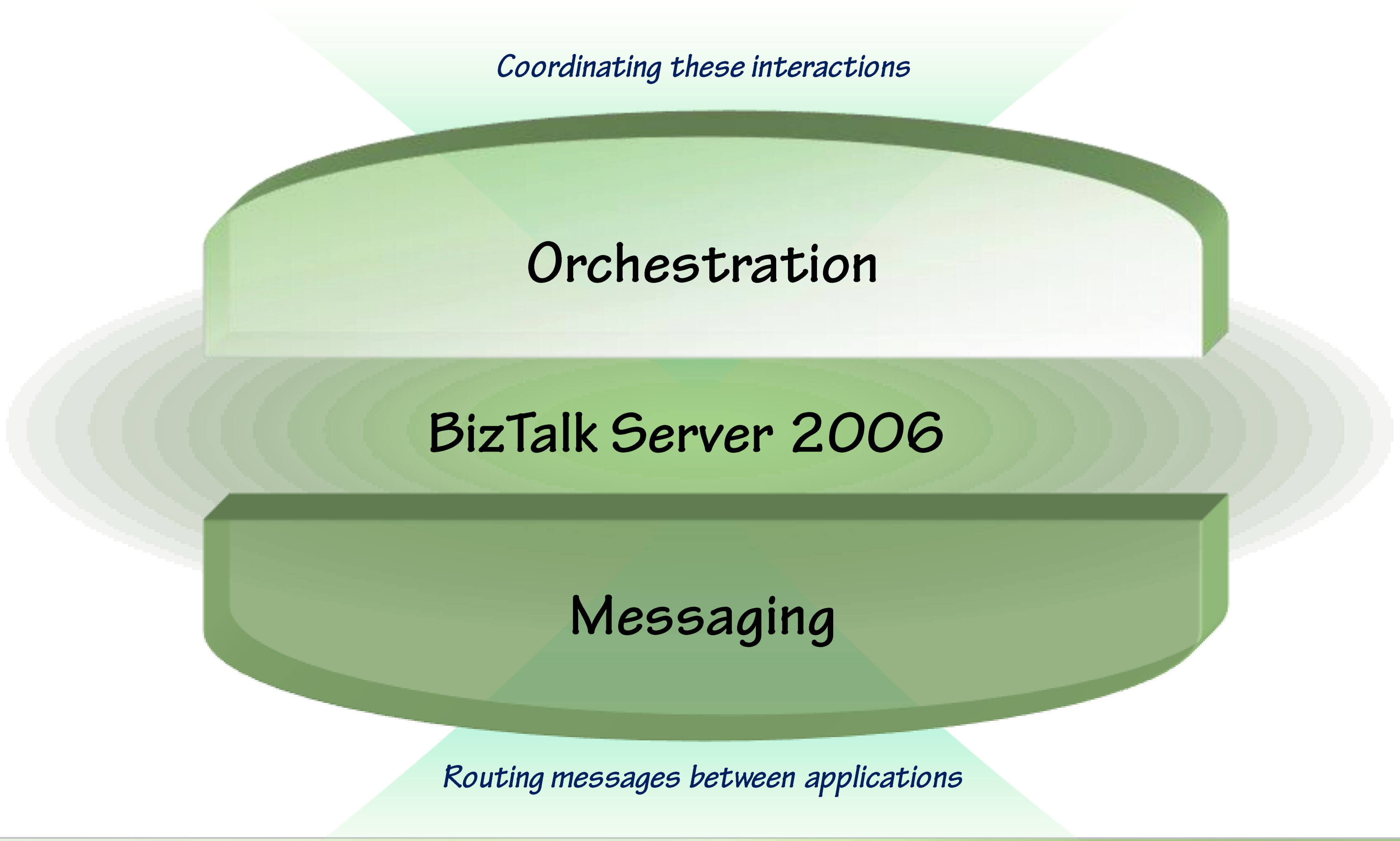
Business process management

- **Simplifying *business process management* (BPM) is a key goal**
  - Business rules, tracking, and monitoring are central areas
- **Business process definitions consist of *business rules***
  - Business rules determine what happens during the process
  - Analysts define the rules and adjust them over time
- **Tracking data allows analysts to monitor how rules are working**
  - Commonly referred to as *Business Activity Monitoring* (BAM)

Introducing BizTalk Server 2006

- ***BizTalk Server 2006*** **is Microsoft's integration server product**
  - Provides *all* of the integration services described herein
- **Architecture is divided into two general areas:**
  - *Messaging*: provides core message integration services
  - *Orchestration*: provides layered BPI-related services

BTS architecture



Coordinating these interactions

Orchestration

BizTalk Server 2006

Messaging

Routing messages between applications

BTS messaging layer

- **The BTS messaging layer is built on an XML foundation**
  - XML is the primary format, XSD the primary type system
  - Supports numerous transports through *adapters*
  - Supports Web services via SOAP, WSE, and WCF adapters
  - Facilitates industry standard schemas via *accelerators*
  - Provides integrated support for EDI and AS2
  - Supports XML to non-XML translations via *pipelines*
  - Supports message transformations via *maps* (XSLT transforms)
  - Automates security mappings through *single sign on* (SSO)
- **Message routing provided by central pub-sub engine**
  - Message *subscriptions* stored in the *Message Box* (MB)

BTS orchestration layer

- **The BTS orchestration layer is built on the messaging layer**
  - Allows coordination of interactions via *orchestrations*
  - Provides a graphical *orchestration designer*
  - Orchestration represented in *XLANG* internally
  - Orchestration can be shared with others via *BPEL4WS*
  - *Orchestration engine* (OE) executes and manages instances
- **OE provides sophisticated state & transaction management**
  - Long-running processes, orchestration *(re)hydration*
  - Atomic and *long-running transactions* w/compensation
- **BTS provides sophisticated BPM features and services**

BizTalk RFID

- **Device independent RFID management**
- **Centralized management of devices and business processes**
- **Tag read event processing**
  - Utilize business rules engine
  - Extensible processing stages
  - Data stored in SQL Server database
- **Marketed under the BizTalk brand, but a separate install**
  - BizTalk can receive data from the SQL database
  - Custom sink can send data directly to BizTalk

Summary

- **BTS 2006 provides a complete integration services architecture**
  - Implemented with a flexible & scalable hub-bus architecture
  - The messaging layer is built on an XML foundation
  - Message routing provided by central pub-sub engine
  - The orchestration layer is built on the messaging layer
  - OE provides sophisticated state & transaction management
  - BTS provides sophisticated BPM features and services

References

- **Understanding BizTalk Server 2006**
  - http://www.microsoft.com/biztalk/techinfo/whitepapers/understanding.mspx
- **Business Process and Integration Developer Center**
  - http://msdn.microsoft.com/bpi/
- **BizTalk Server 2006 Whitepapers**
  - http://www.microsoft.com/biztalk/2006/prodinfo/whitepapers.mspx
- **Pluralsight's BizTalk Wiki**
  - http://pluralsight.com/wiki/default.aspx/Aaron/TheBiztalkWiki.html