maintenance due to frequent API updates
There is little or no standardization on SaaS APIs: IT department expends time dealing with the APIs of each SaaS applications
* Data transmission security
# Data are secure within the hosted environment
# Data transfer between on-premise systems and SaaS applications may not be secure
# Require integration efforts for the diverse, distributed, and decentralized data
# Data integrity, confidentiality, quality, and value have to be preserved

(c) Integration as a service

- Integration as a service (IaaS)
* Migration of the functionality of enterprise applications into the cloud and provide smooth data transport between enterprise and SaaS applications
* Data may be duplicated between on-premise and off-premise applications: need to be completely and compactly synchronized
* Dynamic SaaS APIs worsen the integration problem
* Limited access to the cloud presents another problem
# Accessing local applications is much easier than clouds
# Set local integration points are easy in local applications, but difficult in SaaS applications
# E.g., SalseForce.com's API does not support transactions against multiple records
→ Integration code has to handle that logic
* Service integration

(WWF)

(e) Business-to-business integration services

- Business-to-business integration (B2Bi)
  → Extend business processes to business partners including customers, vendors, suppliers, distributors, and other alliances
  * Pure EAI: only for internal data share
  * B2Bi has the following capabilities
    # Encrypt files for safe passage across the public network
    # Manage large data volumes, transfer batch files, and convert disparate file formats
    # Guarantee data accuracy, integrity, confidentiality, and delivery
  * Provides connectors for major ERP, CRM, SCM, …
- Cloud-based enterprise mashup integration services for B2B scenarios
  * Mashup: combination of different and distributed resources including content, data or application functionality
    # Resources represent the core building blocks for mashups, and can be accessed through APIs
    # Widgets or gadgets put a face on the underlying resources by providing a graphical representation and piping the data received from the resources
    # Piping can include data aggregation, merging, or filtering
  * Enterprise mashups: users integrate heterogeneous digital data and applications from multiple