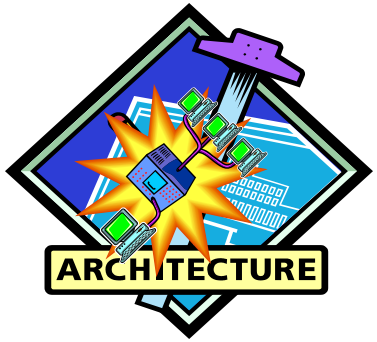
AZORA™

Achieve Efficient Integration©

Service Oriented Architecture Based Integration



Mike Rosen

CTO, AZORA Technologies, Inc.

Mike.Rosen@Azoratech.com

Mike Rosen

CUTTER CONSORTIUM
ACCESS TO THE EXPERTS

- Consultant
  - Chief Enterprise Architect for service and component based systems
    - Finance, Insurance, Telecom
  - SOA, EA, MDA implementation and training
  - 20 years experience in distributed applications
- Author
  - Cutter Consortium
    - “Service Oriented Integration – Aligning SOA with Enterprise Integration”
    - “Implementing SOA on Common Technologies”
  - Implementing SOA Applications, due 2005
  - Developing e-Business Systems and Architecture: A Manager’s Guide, 2000, Morgan-Kaufman
  - Integrating CORBA and COM Applications, 1998, Wiley.

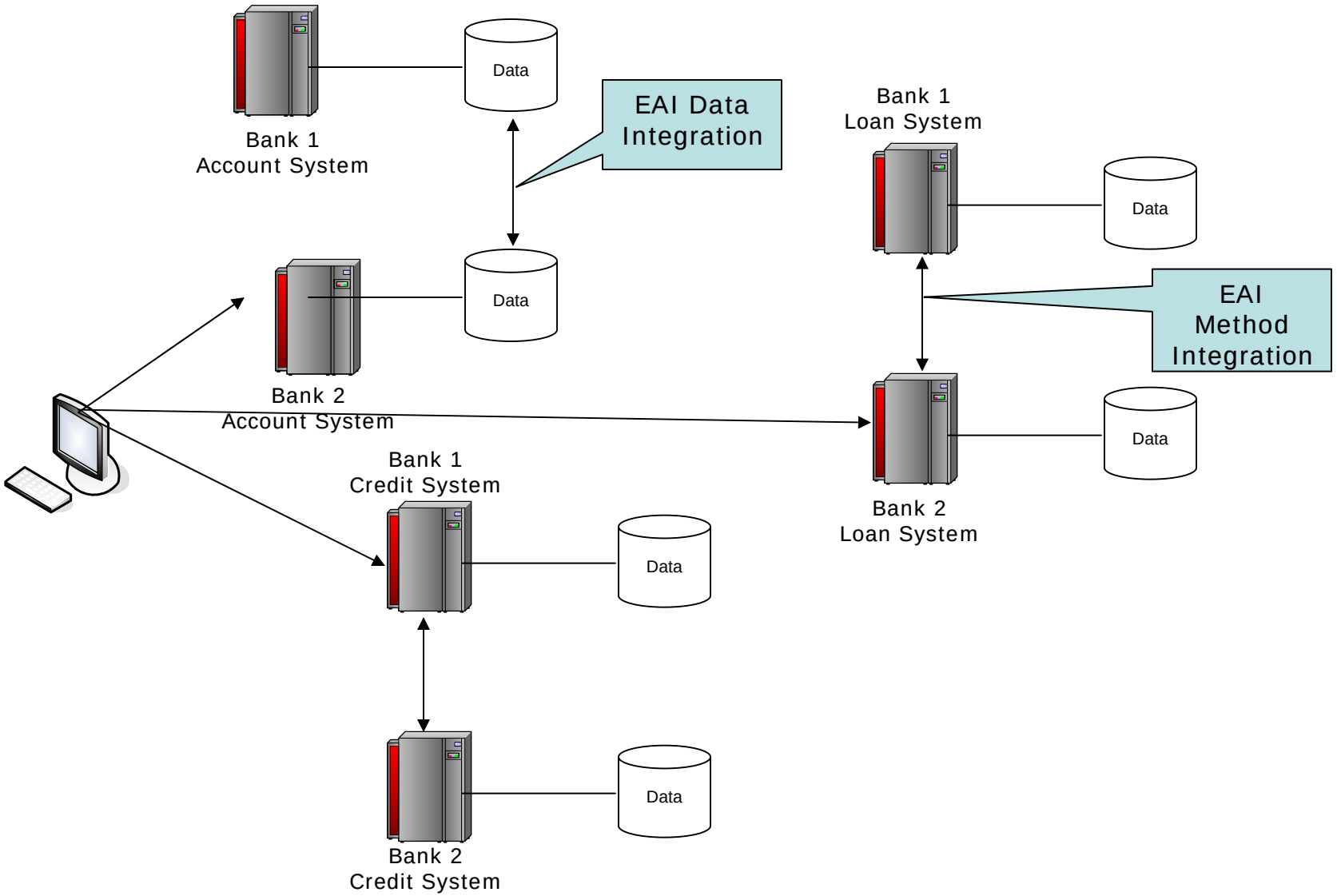
© Michael Rosen 2005

Agenda

- Enterprise Application Integration
- Web Services to the Rescue
- What is Service Oriented Architecture?
- Business and Integration Services
- Enterprise Architecture for SOI
- Conclusion

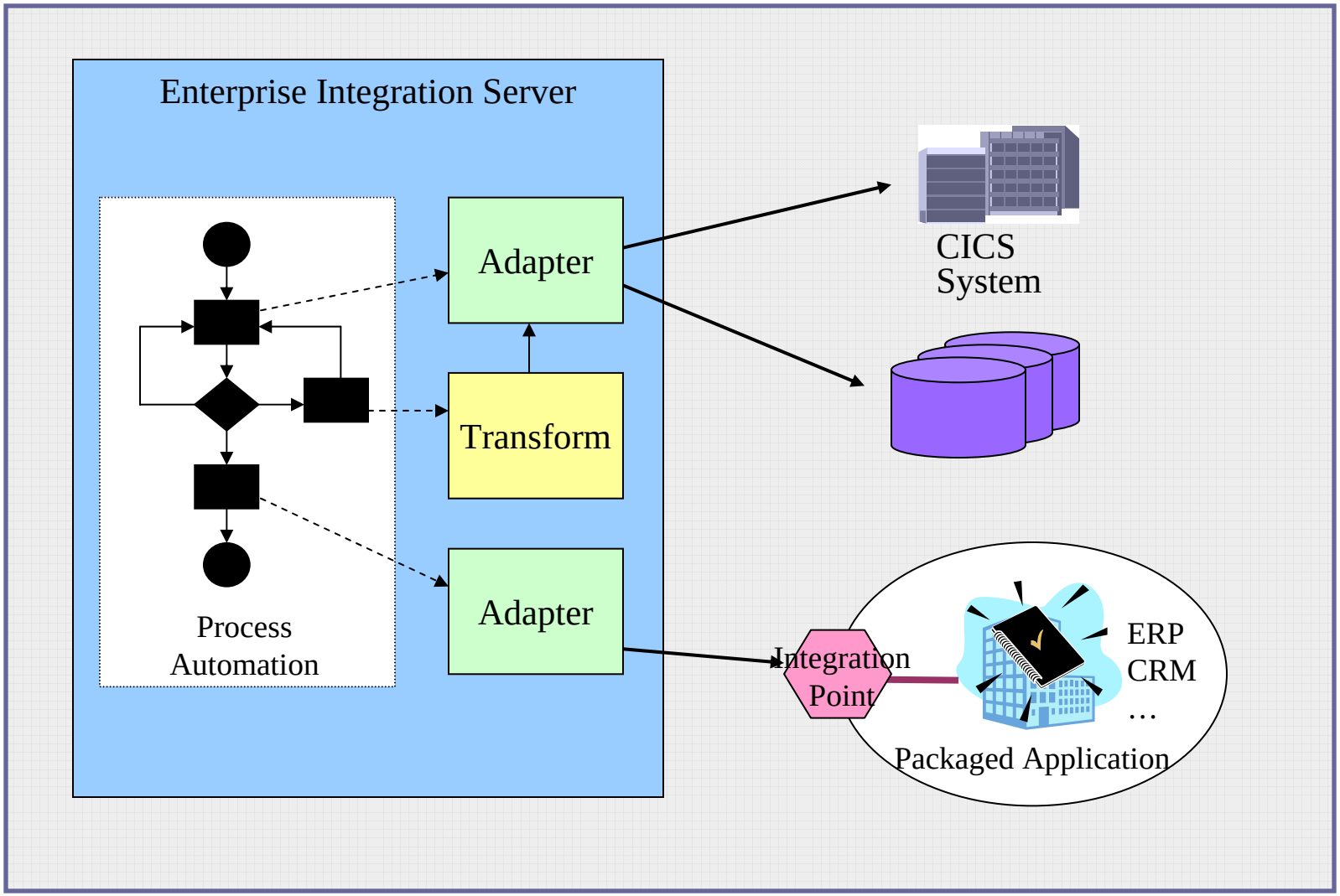
© Michael Rosen 2005

A Typical Integration Scenario



© Michael Rosen 2005

Typical Enterprise Application Integration



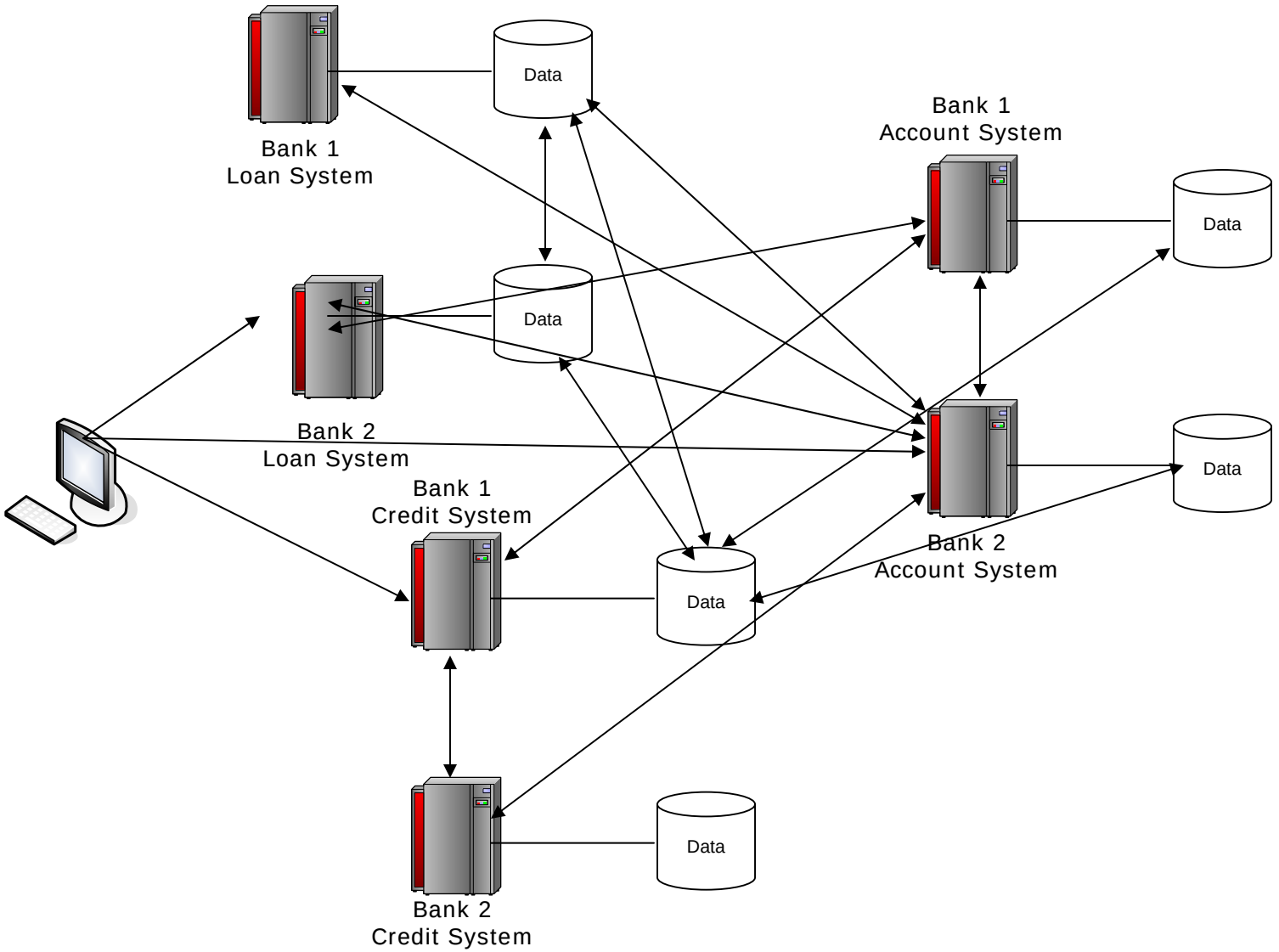
© Michael Rosen 2005

Enterprise Application Integration

- Described by Process Automation/Workflow
- Adapters interface with applications at their integration points
- Transformers change data and/or message format
- Enterprise Integration Server provides runtime
- Good concepts to apply to a bus model, rather than point-to-point

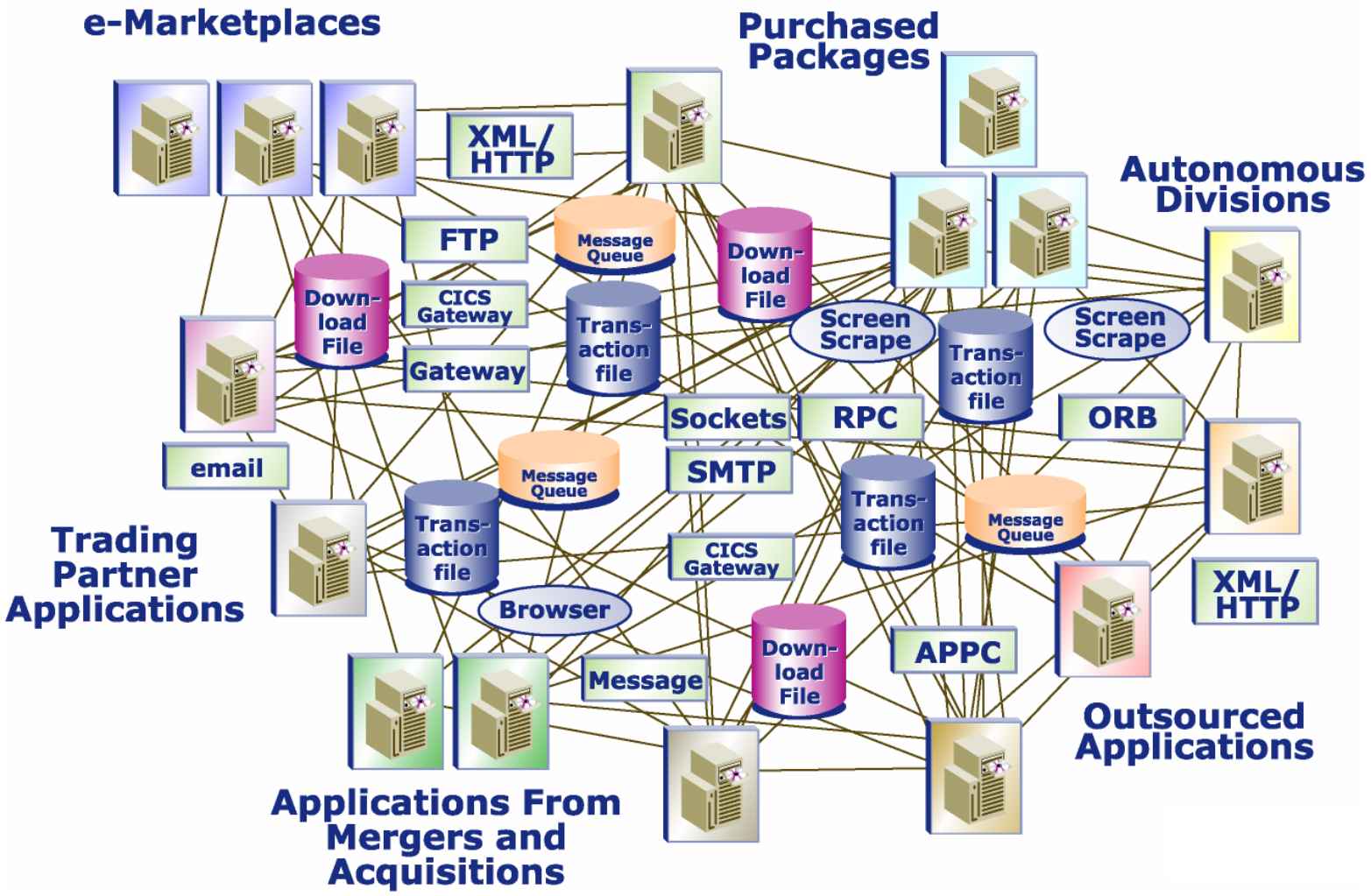
© Michael Rosen 2005

The EAI Solution Doesn't Scale



© Michael Rosen 2005

Enterprise Application “Spaghetti”



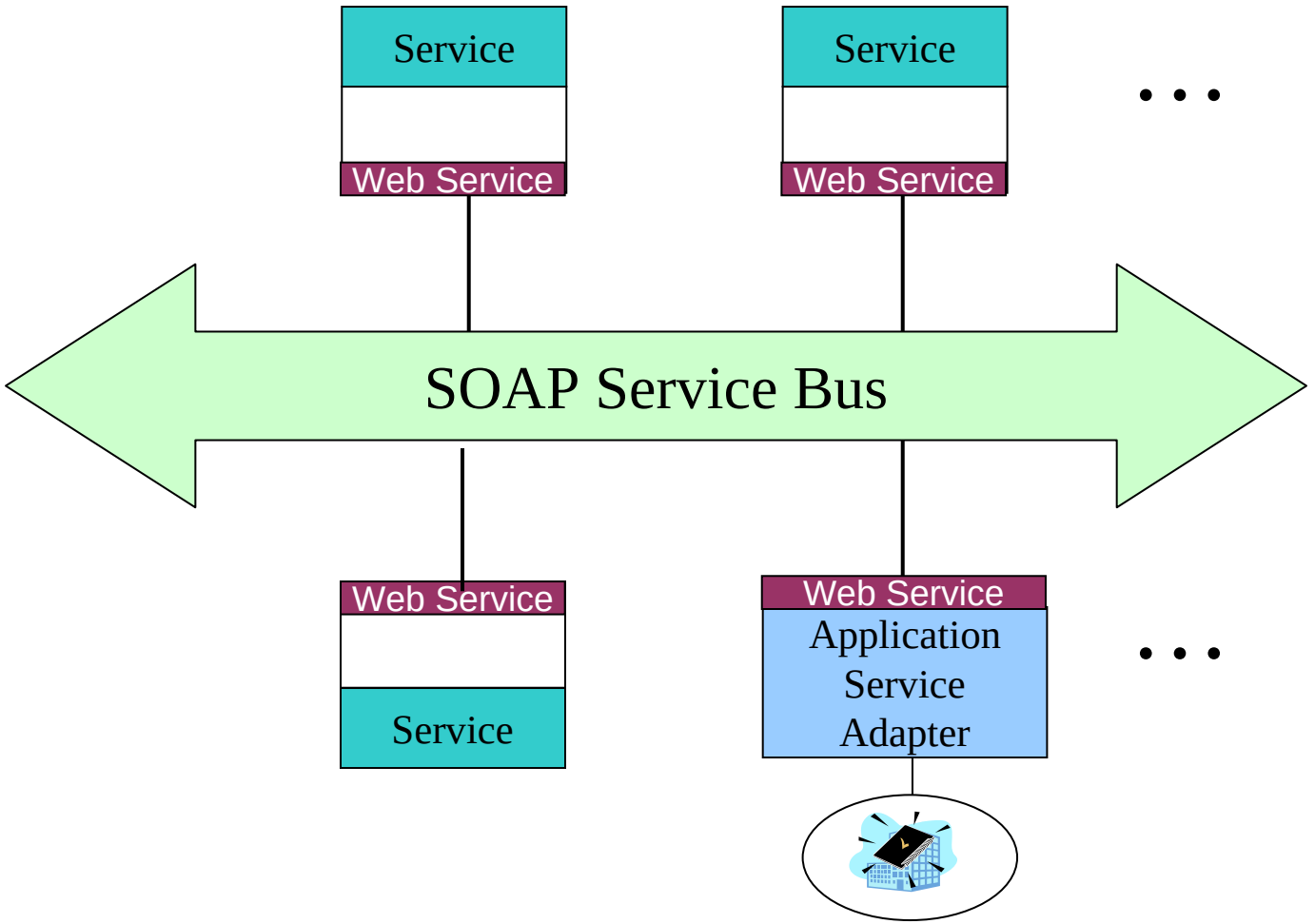
© Michael Rosen 2005

EAI and Web Services

- Then…
  - Previous EAI products used proprietary protocols and techniques
  - EAI applications were too expensive to implement. Many attempts at EAI failed
  - Most EAI implementations created point-to-point connections, essentially new EAI stovepipes
- Now…
  - All EAI vendors are transitioning to Web Services to replace proprietary protocols
  - COTS vendors are also supporting Web Services, thus driving down the cost of connectivity and integration
  - Service Oriented Integration, using a new class of product call the Enterprise Service Bus, is now the target of EAI

© Michael Rosen 2005

Web Services to the Rescue?



© Michael Rosen 2005

SOA History

- Service Oriented Architecture (SOA) is NOT new!
- Many Successful SOA Applications have been built in the past:
  - CORBA (Wells Fargo, Credit Suisse)
  - Tuxedo
- Many, many more attempts at SOA failed
- But, we can learn from what failed, and what succeeded

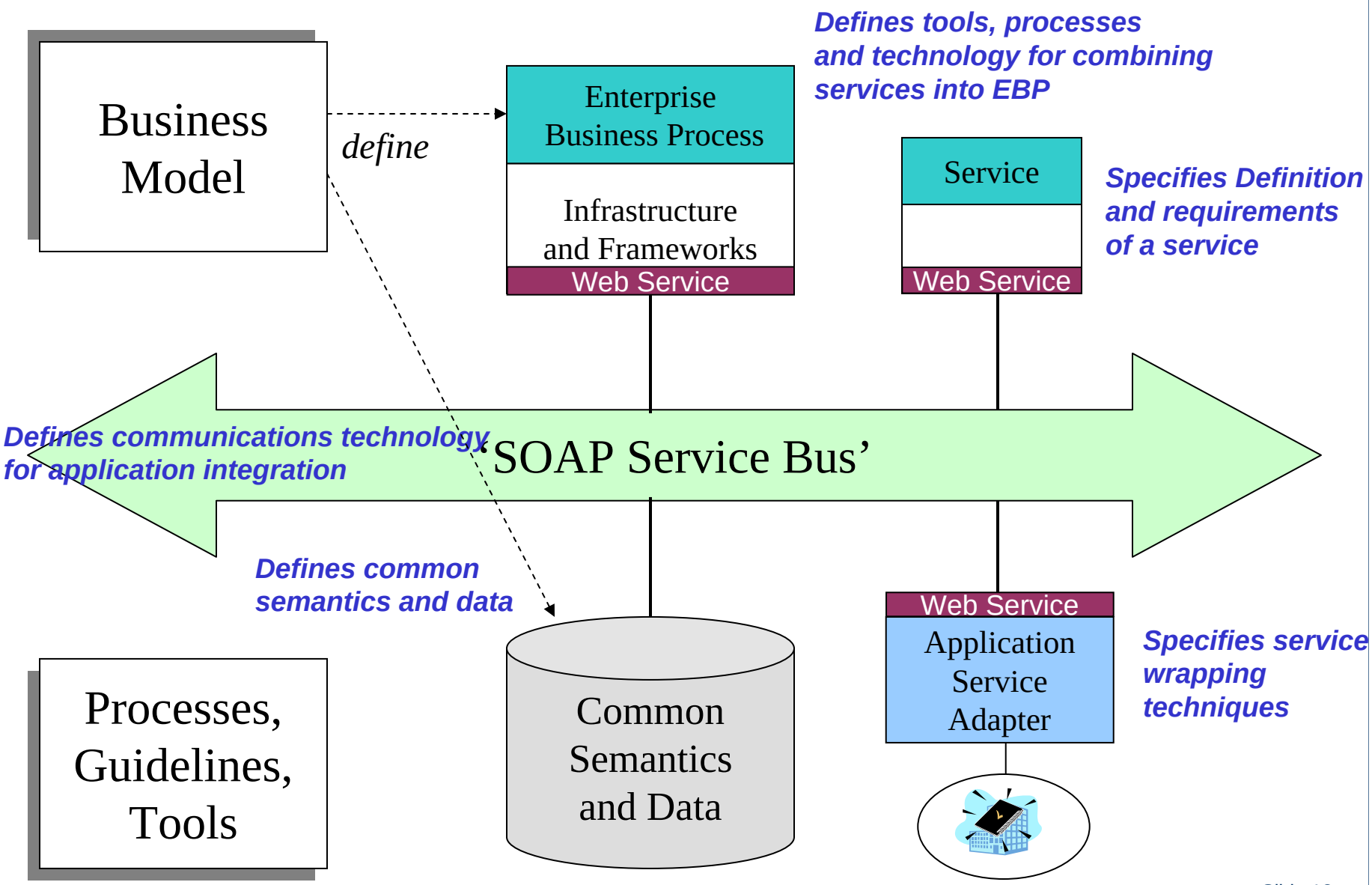
© Michael Rosen 2005

SOA is Hard!

- Previous technical infrastructures were very difficult to master
- We did not adequately understand the characteristics of services and service design
- Requires an understanding of the business and information and a strategic vision
- Requires an architectural based approach
  - But architecture is hard too!
- Requires an appropriate methodology
- Requires a supporting organizational structure

© Michael Rosen 2005

…And Now, the Rest of the Story



© Michael Rosen 2005

What is an SOA?

- SOA is concerned with the *independent* construction of services which can be *combined* into meaningful, higher level business processes within the *context of the enterprise*.

- A Service Oriented Architecture describes several aspects of services within an enterprise:
  - The granularity and types of services
  - How services are constructed
  - How the services communicate at a technical level
  - How the services are combined together (i.e. orchestrated)
  - How the services interoperate at a semantic level (i.e. how they share common meanings)
  - How services contribute to IT and Business Strategy

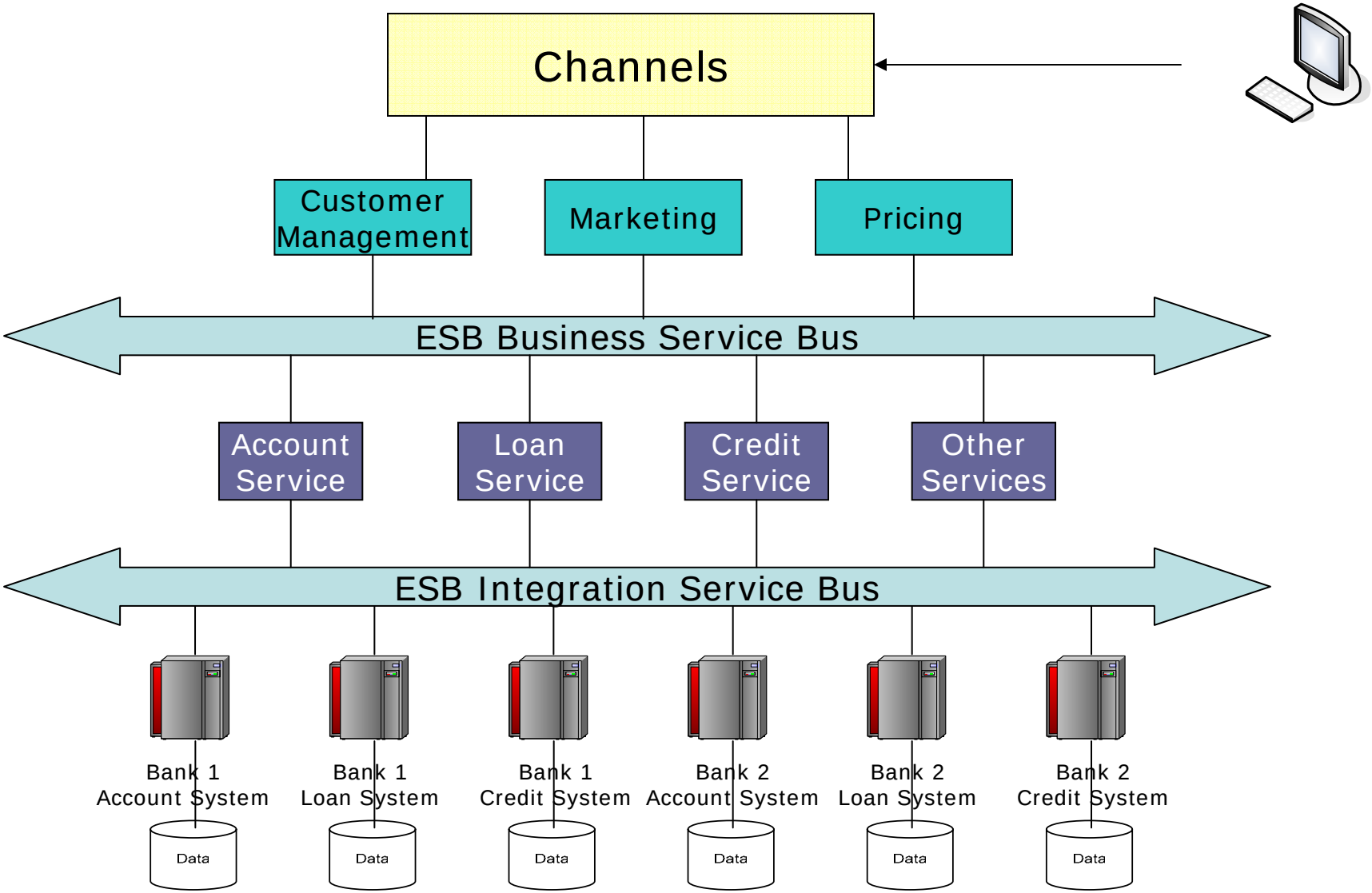
© Michael Rosen 2005

Service Oriented Integration

- SOI
- An architectural and technology based approach to exposing and integrating existing applications as services
- Builds on EAI technology, using new Web services based platforms
- Exposes services to a bus, not point-to-point
- Extends SOA to integration solutions

© Michael Rosen 2005

SOI: A Better Solution



© Michael Rosen 2005

Benefits of SOA Integration Approach

- Integrate once, connect many
  - Each system is integrated once into the service bus, rather than many time for each point-to-point connection
  - Less cost, consistent access
- Build up higher level business services
  - Combine lower level operations into business services that align with the goals and strategy of the new enterprise, rather than of the old systems
  - Quickly construct high-level, high-value business processes from the business services in response to new initiatives, competitive pressures, regulatory changes, …
- Flexibility
  - Multiple services can be easily constructed from the integration of existing applications
  - New processes can be constructed from the service

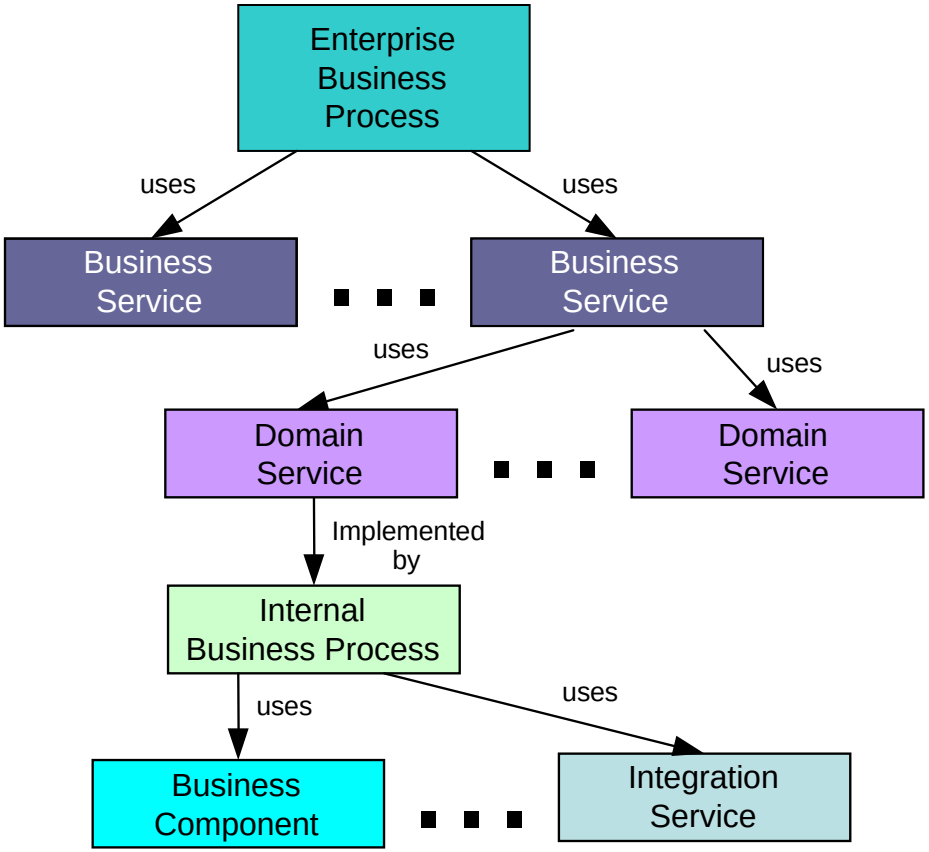
© Michael Rosen 2005

Benefits of SOI Approach (2)

- Adaptability to change
  - Business Processes change quickly
  - Operational Systems are difficult, costly and slow to change
    - Layered SOI approach enables quickly reconfiguring processes or services without needing to change operational systems
  - Operational systems are retired or replaced
    - Layered SOI approach allows operational systems to change without affecting business processes
- Incremental Approach
  - Start small
  - Add new integration services, business services and processes over time, as part of specific projects, in response to specific business needs
  - Flexibility and capabilities increase exponentially with each new service.

© Michael Rosen 2005

Hierarchy of Service Types



© Michael Rosen 2005

Factors Affecting Services

- Granularity – Amount of work performed per invocation
  - Fine
  - Medium
  - Large
- Visibility – Who can see and invoke the service
  - Published
  - Public
  - Private
- Scope – Organization unit or boundary for the service
  - Application
  - Workgroup
  - Line-of-business or division
  - Enterprise

© Michael Rosen 2005

Integration Services

- An architectural and implementation approach to integration using Web services and SOA
- Data Integration Service – provides data integration between multiple applications. Initiated by a legacy system in which data has changed. Frequently implemented as Publish and Subscribe.
- Functional Integration Service – provides shared functionality between multiple applications. Initiated by the application requiring the functionality.
- Not the same as SOA Business Services!!!
  - Although often treated the same.

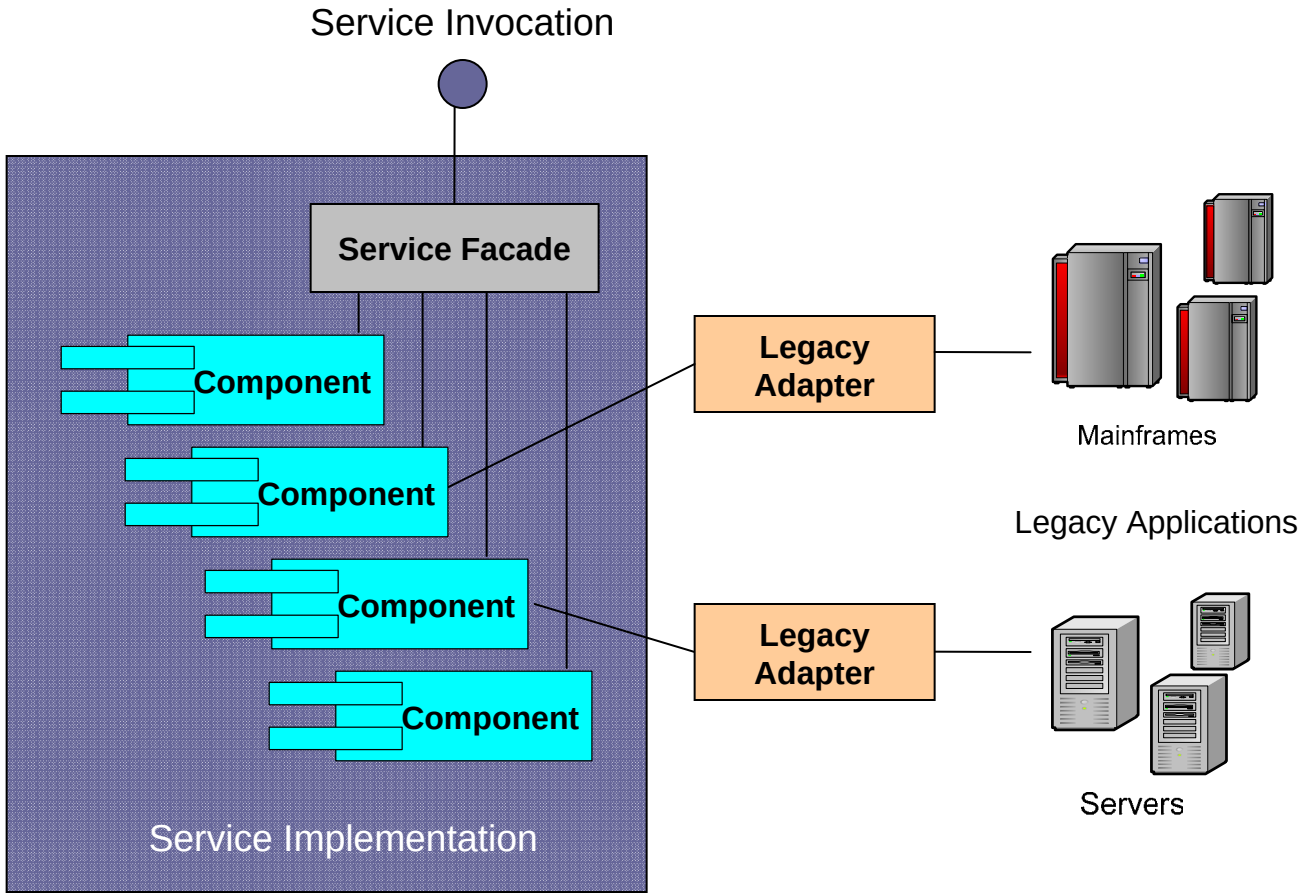
© Michael Rosen 2005

Integration Services (2)

- Integration services provide interface to existing application
- Interface granularity influenced by existing applications
- Synchronous Invocation is common
- ACID Transactions may be required
- Not exposed directly to business services
  - Wrapped by business components or other services
  - Hide internal API's, data models and application topology
  - Enhance, modify or combine existing functionality

© Michael Rosen 2005

Integration Service Pattern



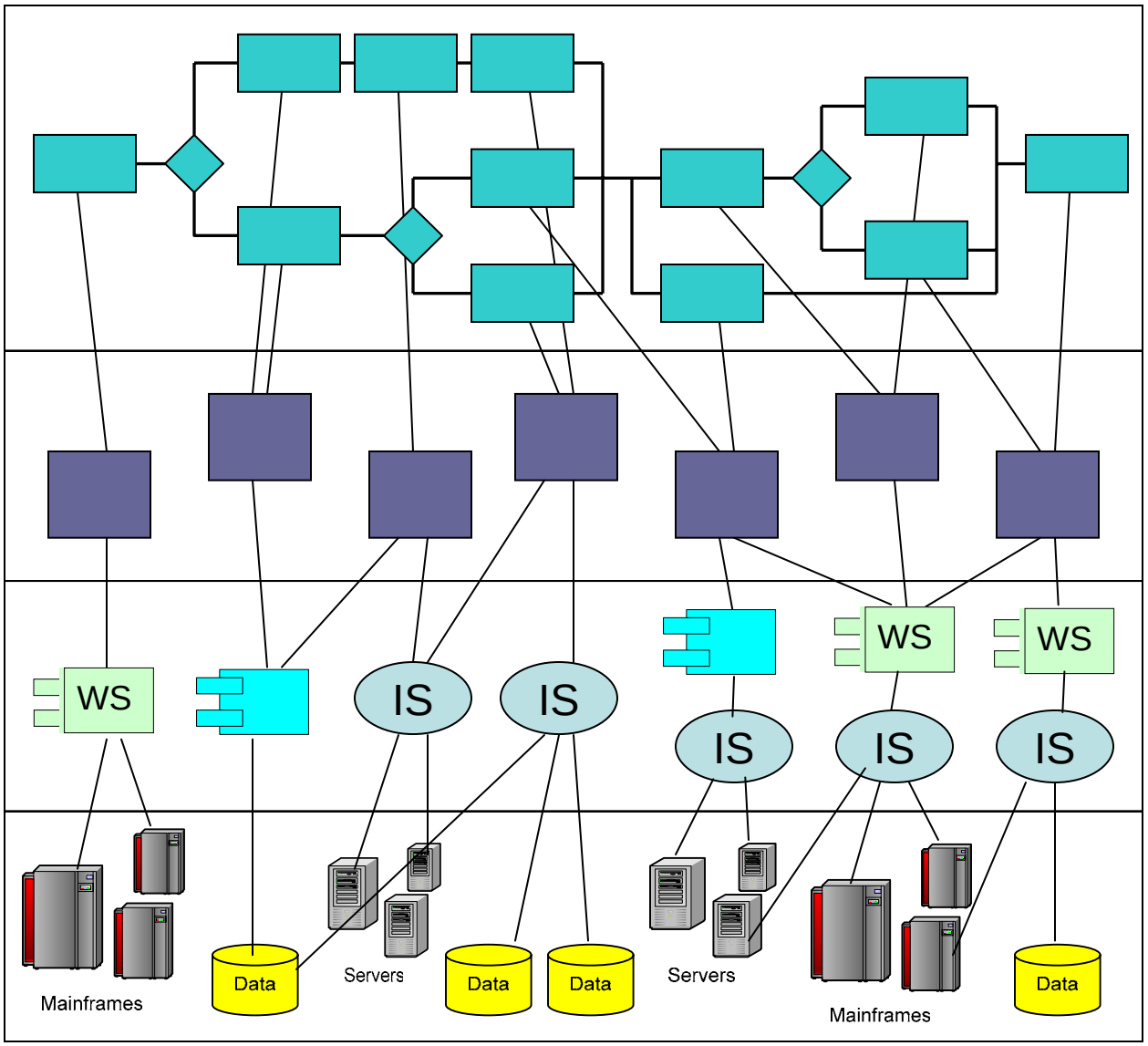
© Michael Rosen 2005

Common ‘Bottom Up’ Approach

- Start with existing application functionality
  - Expose functionality as a ‘service’ using existing API
  - Expose data model in the service interface
  - Creates dependency between existing application and new service interface
  - API and data model dependencies create functionally and semantically incompatible service interactions, essentially limiting service to point-to-point integration
  - Not driven by enterprise requirements
- Promoted by tool vendors
  - Service interface can be generated
  - Makes a great demo
- EAI all over again
  - New Web services technologies
  - Same old architectural problems

© Michael Rosen 2005

Layered Enterprise SOI Architecture



Business Processes

Business Services

Components and Integration Services

Enterprise Resources

© Michael Rosen 2005

SOI Enterprise Architecture Layers

- Layer 1 – Enterprise Resources and Operational Systems
  - Consist of existing applications, legacy and COTS systems, CRM and ERP applications, and older OO implementations
  - Provide ‘business operations’ – transactions implementing single units of work within the operational systems
  - Typically access or modify data in a ‘System of Record’
- Layer 2 – Components and Integration Services
  - Integration Services provide access to the resources and systems of Layer 1
  - Components wrap integration services
  - Components provide a ‘single point of contact’ for integration services, preventing a proliferation of cut-and-paste code
  - Components are typically implemented with EJB or .NET
  - Integration services are increasingly being implemented with Web services

© Michael Rosen 2005

SOI Architecture Layers (2)

- Layer 3 – Business Services
  - Provide high level business functionality throughout the enterprise
  - Provide a ‘service interface’ layer of abstraction to the functionality of layer 2
  - Services are managed, governed enterprise assets with SLAs
  - Represent a logical grouping of component, integration services and operations
- Layer 4 – Business Processes
  - Processes are a series of activities which are executed in an ordered sequence according to a set of business rules (called a choreography or business process model)
  - Executed in response to business events
  - Provide long-running sets of activities
  - Composed of multiple services and typically involving multiple service invocations

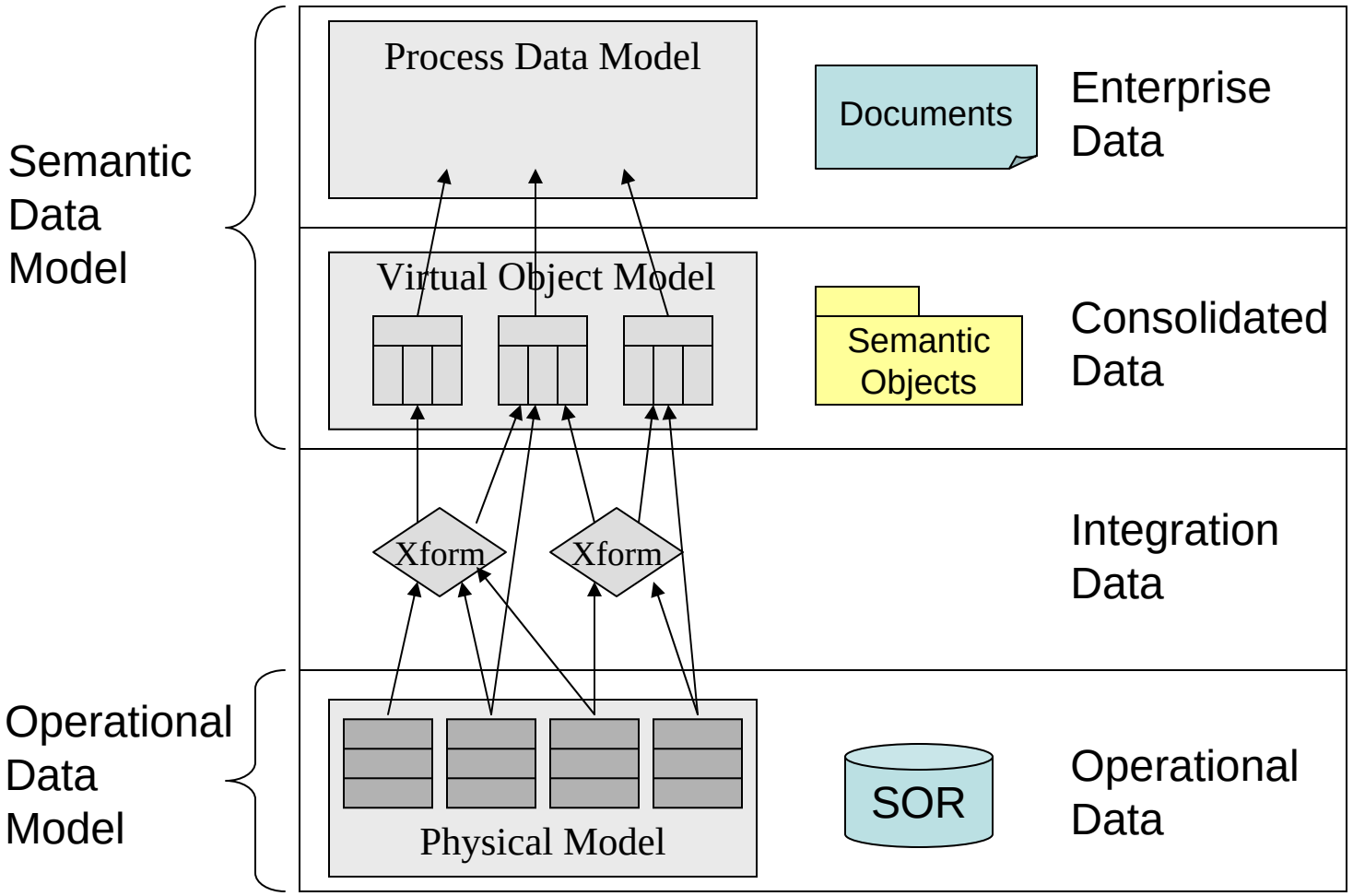
© Michael Rosen 2005

Enterprise Information Design

- Equally important to enterprise flexibility and agility
- Defines the business semantics needed to support the enterprise processes and services
- Critical to ‘ad-hoc’ combination of services into business processes
- Coordinated with enterprise process design

© Michael Rosen 2005

Enterprise Information Layers



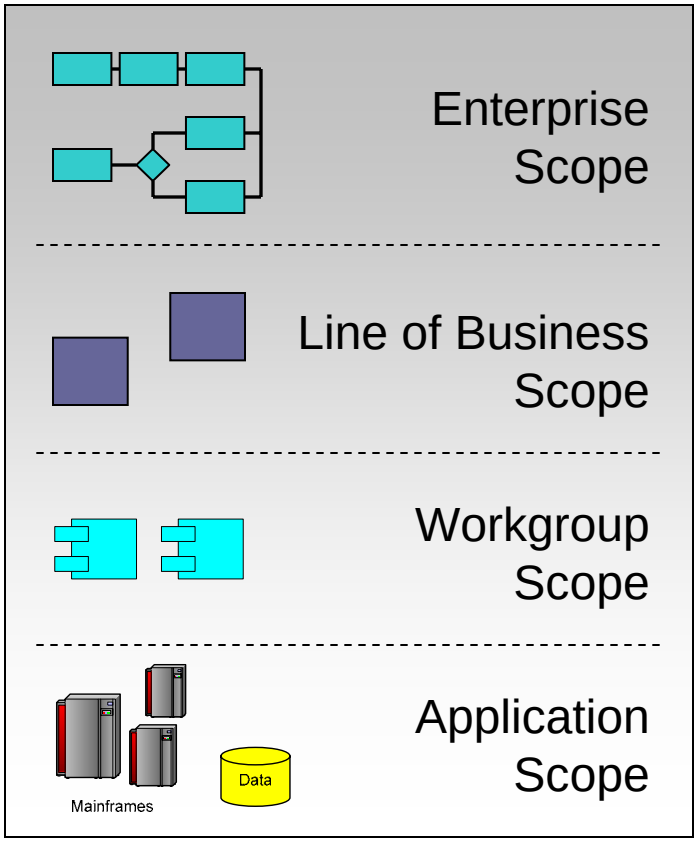
© Michael Rosen 2005

Enterprise Information Layers (2)

- Operational Data
  - Resides in systems of record and other operational data stores
  - Described by the ‘Physical Data Model’
- Integration Data
  - Operational data is exchanged between operational systems
  - Data is extracted from one system, transformed, and the loaded to another
- Semantic (Consolidated) Data
  - Provides the data required by the service interfaces
  - Described by the ‘Virtual Data Model’
  - Implemented by a mapping/transformation of the operational data
- Enterprise Data
  - Business Documents provide a composite of semantic business objects to support enterprise processes

© Michael Rosen 2005

Scope of SOA Constructs



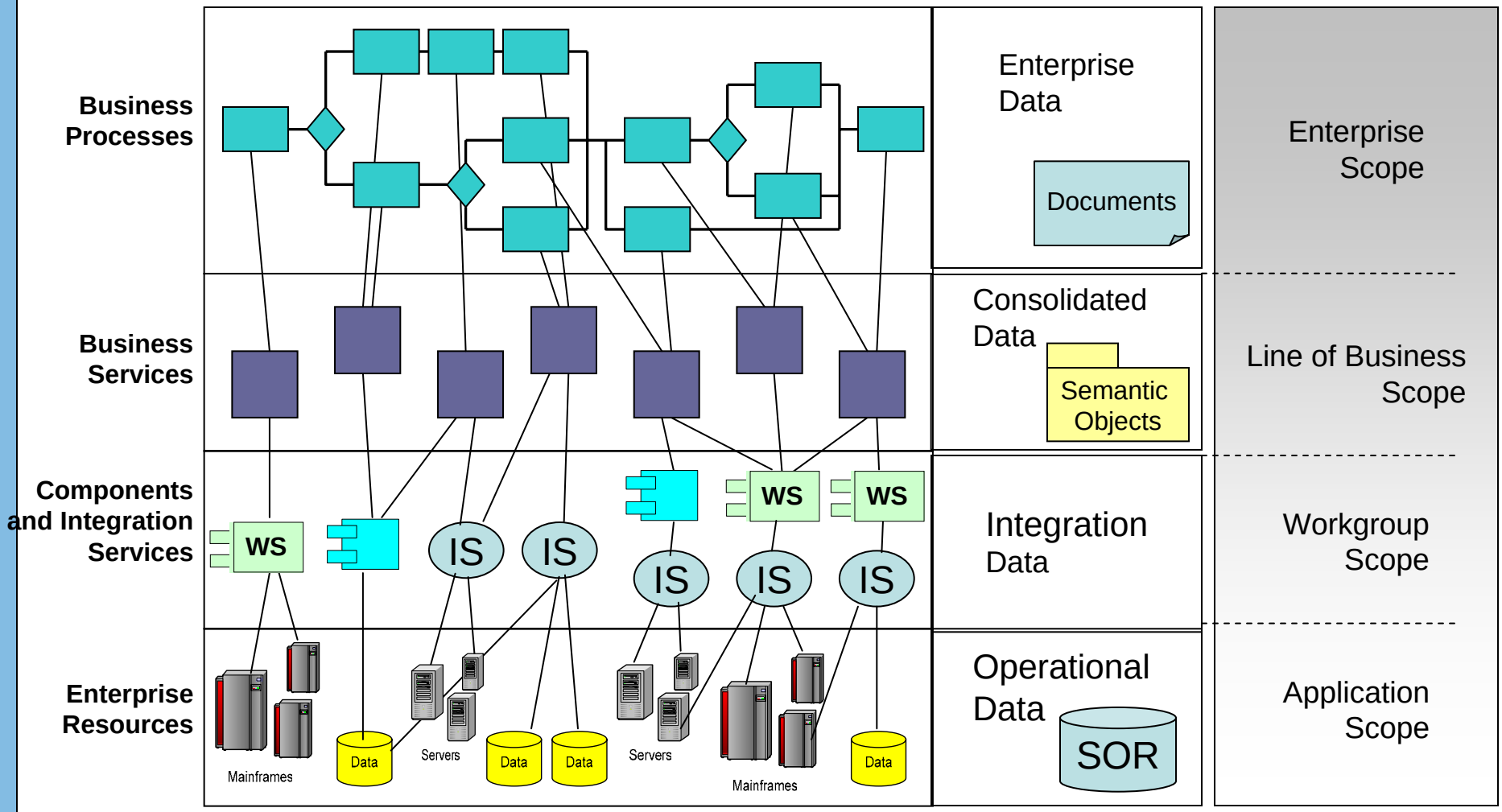
Enterprise Concerns

Scope

Organizational independence

© Michael Rosen 2005

Putting it all Together



© Michael Rosen 2005

The Enterprise Service Bus

- Provides an ideal platform for SOI Applications
- Integration Infrastructure – exposing existing applications as services
- Service Infrastructure – defining, implementing, invoking and combining services
- Enterprise Strength – enterprise class management, scalability, performance, reliability
- Full Disclosure Notice: The author does not represent or promote any specific ESB products (although he has used several and has his own opinions (incidentally not shared by the International House of Architecture).
  He does not own stock or stand to profit in anyway should you decide to follow his recommendations (actually, I still have a pile of BEA, but that's another story…)

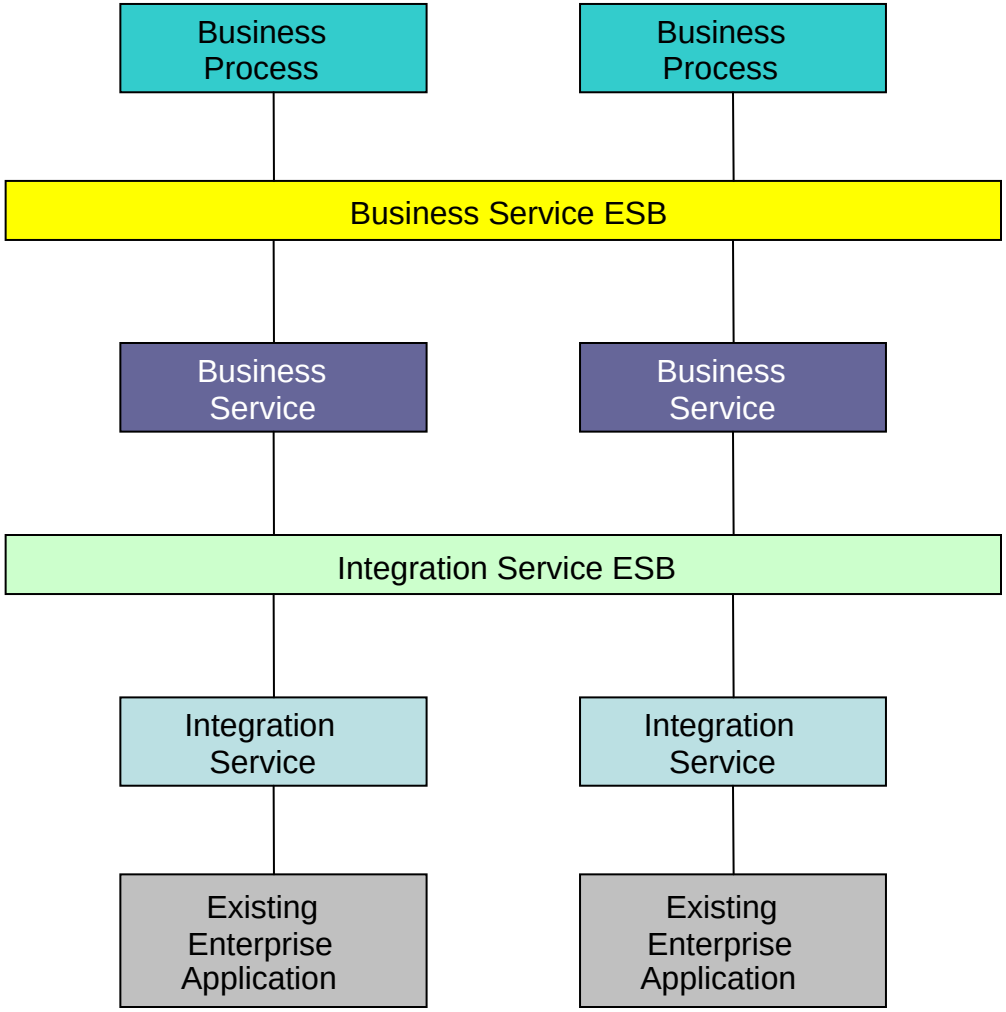
© Michael Rosen 2005

Requirements of Different Service Types

Requirement	Integration Service	Business Service
Design	Expose existing application functionality and data	Provide services aligned with Enterprise Business Model
Semantics and Data	Based on operational data models, required significant transformation services	Based on enterprise semantic model. Some aggregation, minimal transformation
Invocation	Event driven, or invoked by other services: Synchronous request/reply, events, fire-and-forget	Invoke by a business process Asynchronous request/reply or fire-and-forget
Development	Requires specialized adapters for connection to legacy systems	New development using ESB APIs directly
Important Features	• Synchronous request/reply, events, publish and subscribe • Integration • Message Processing • QoS • Security and Management	• Asynchronous request/reply • Service Interaction • Process Orchestration • SLA • Security and Management

© Michael Rosen 2005

We’re all Bozos on this Bus



© Michael Rosen 2005

Summary

- SOI combines Web service, EAI and SOA
- Driven top down by business requirements and model
- Driven bottom up by existing applications
- SOI provides an level of indirection between operational system's API and data and Enterprise defined semantics, services and processes
- This leads to a more flexible, agile enterprise
- Integration Services are fundamentally different than Business Services

© Michael Rosen 2005

Integration Service Summary

- Provides service access to existing systems
- Responsible for mapping between Enterprise business / service model and existing application functionality
- Responsible for mapping between Enterprise Semantic model and existing application data model
- Simultaneously insulates and integrates

© Michael Rosen 2005

Implementing SOI Integration Requires…

- Architecture
  - Describe the essential context to enable cooperating services
  - But phase in the details as needed
- Business Model
  - Describe processes, services, interfaces, enterprise data and semantics
- Tools
  - Support service construction
  - Incorporate architectural concepts
  - Implement Service composition
  - Provide platform independence of business logic

© Michael Rosen 2005

AZORA™

Achieve Efficient Integration©

Questions