Abridged from the book

SOA
Right Away!

BRIDGING THE GAPS BETWEEN
BUSINESS & IT

…and modified for presentation
by Al Smith, Jr. www.TheJavaArchitect.com

The Goal of SOA

The Goal of SOA

- **Develop Applications that Provide:**
  - › Standardized Service Contracts
  - › Loosely Coupled Components
  - › Abstracted Components
  - › Reusable Components
  - › Autonomic Components
  - › Stateless Components
  - › Discoverable Components
  - › Composable Components

History of Distributed Computing

- **Distributed computing software components**
  - › CORBA
  - › DCOM
  - › Web Services
- **Interface description languages**
  - › XML-RPC
  - › SOAP

What Makes SOA Different?

- **Service-Orientation**
  - › A design paradigm that provides a means to achieve "separation of concerns" through the use of services
    - • Defined differently between SOA vendors
  - › Derived from
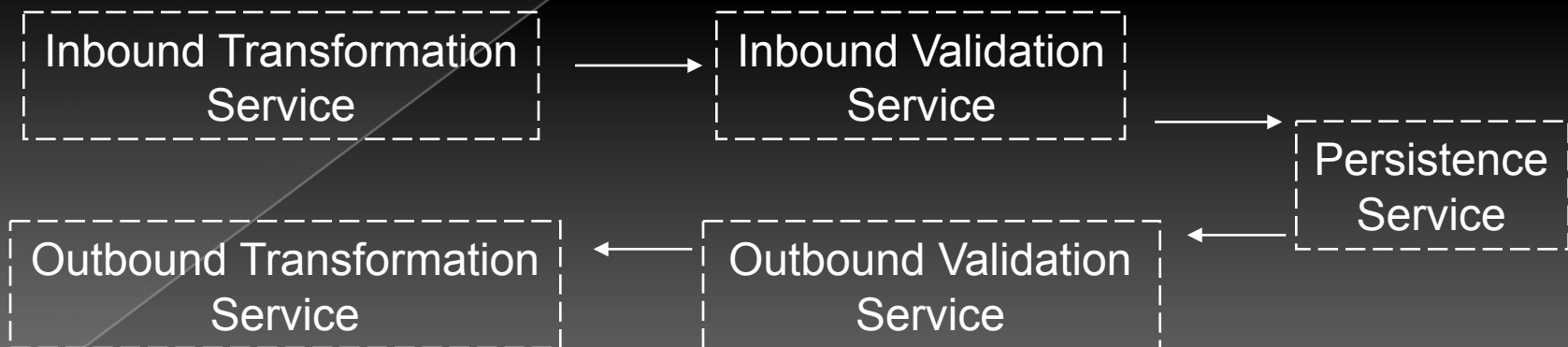   - • Object Orientation
    - • Web Services

> "Ironically this is one of the problems SOA contracts attempt to prevent"

What is Service Oriented Architecture

⊙ **Service**

› An abstract definition of a function to be performed within a business process

- Defines a *contractual* responsibility that a piece of software will perform within the constraints of a given process

```
┌─────────────────────┐      ┌─────────────────────┐
│ Inbound Transformation │ ───→ │  Inbound Validation  │
│      Service         │      │      Service         │
└─────────────────────┘      └─────────────────────┘
```

```
                                              ┌──────────────┐
                                              │ Persistence  │
                                              │   Service    │
                                              └──────────────┘
```

```
┌─────────────────────┐      ┌─────────────────────┐
│ Outbound Transformation │ ←── │  Outbound Validation │
│      Service         │      │      Service         │
└─────────────────────┘      └─────────────────────┘
```

The State of SOA Adoption

- **SOA in a Box**
  - To address many of SOA's new infrastructure requirements, vendors have brought several new product categories to market:

    - SOA repositories
    - Enterprise service buses (ESBs)
    - SOA appliances
    - SOA and Web services management solutions

The State of SOA Adoption

- ◉ **SOA in a Box** (continued)
  - › *Most architects understand how to deploy a tactical, project-by-project approach*
    - • Buying whatever seems to fit the needs

  - › Potential Issues:
    - • Duplicate investments
    - • Incompatible infrastructure
    - • Brittle solutions

The State of SOA Adoption

- **Big Bang Adoption**
  - Few organizations can afford to fund a large, upfront, strategic investment approach:
    - Building an SOA platform is best done via incremental steps toward a strategic vision
    - The SOA platform evolves in steps rather than attempting one big jump

The State of SOA Adoption

- **Incremental Adoption**
  - › To successfully execute an incremental SOA platform evolution, you need a coherent approach for:
    - Envisioning
    - Designing
    - Evolving your platform

The State of SOA Adoption

- **The Recommended Approach Consists of:**

  - SOA platform definition
  - A functional planning model
  - An evolutionary manner of building from your existing technologies
  - Integrating SOA specialty products as needed

Build or Buy

- **The Two Extremes are to Provide:**

  - A <u>suite approach</u> (full stack), tightly-integrated, all -embracing technology stack

  - To deliver a set of '<u>best-of-breed</u>' point solutions that the organization assembles to meet their specific requirements

Build or Buy

- **Build or Buy: Suite Approach**
  - Represents:

    - Simple negotiation with a single vendor
    - Less work in setting up the environment
    - Knowledge that all components will interoperate
    - Simpler ongoing management

Build or Buy

- **Build or Buy: Suite Approach**
  - Considerations:
    - It creates a level of lock-in to a 'strategic' vendor that might be unacceptable
    - It limits the ability to extend the functionality (or performance) in a way that is not within the product development plans for that suite

Build or Buy

- **Build or Buy: Best-of-Breed**
  - Represents:
    - Minimal start-up cost
    - Open Source technology
    - Services are acquired as needed
    - Not obligated to fit a square peg into a round hole

Build or Buy

- **Build or Buy: Best-of-Breed**
  - Considerations:

    - Specifications aren't always industry compliant
    - Limited service domain knowledge and technical expertise
    - Requires an in-house development team with SOA expertise

Defining an SOA Strategy

Defining an SOA Strategy

- **Strategic Implementation Paths**
  - Simple Legacy Access
  - Build composite services
  - Achieve mature enterprise SOA delivery

Defining an SOA Strategy

- **Governing SOA Services**
  - › Operation Control
  - › Utilizing Governance to Accelerate Agility
  - › Developing a Smarter SOA with Governance
  - › Enabling Policy Management

Governing SOA Services

- **Operation Control**
  - › SOA needs to be a cooperative venture with buy-in and participation from all the people who will be working on or with the business applications

    - Governance teams overseeing their initiatives
      - Critical in identifying required common services

Governing SOA Services

- **Operation Control** (continued)
  - › SOA needs to start from the beginning with the services and applications that the business *needs*
    - Little value in developing cool services undesired on business side

Governing SOA Services

- **Utilizing Governance to Accelerate Agility**
  - SOA is about business as much as it is about traditional application development
    - Mistake to leave implementation to coders
    - Get business people involved in SOA projects from the start

Governing SOA Services

- **Developing a Smarter SOA with Governance**
  - To get the scope and focus of SOA right it is important to remember that:
    - The A in SOA stands for "architecture"
    - Architects and Project Managers must play a key role

Governing SOA Services

- **Developing a Smarter SOA with Governance** (continued)
  - › Requires investing in management and design to make sure the SOA project fits the business needs
    - *Vendors urge investing in software infrastructure*

Governing SOA Services

- **Enabling Policy Management**
  - › Policy management ensures that policies approved by the governance framework, covering areas such as compliance, conformance, security, etc. are enforced throughout the lifecycle of the SOA initiative

Financial Justification

◉ **Benefits of Implementing SOA:**

- Increased Flexibility

- Increased Extensibility

- Increased Robustness

- Increased Reusability and Productivity

- Increased Business Requirements Fulfillment

Managing SOA Risk

- ## **Risk Identification**
  - Implementations can get bogged down:
    - Project leaders try to do too much too quickly
    - Get overwhelmed with enormity of tasks
    - Taking on too many SOA projects with too many services

Managing SOA Risk

- **Risk Identification** (continued)
  - Great flux in Web services specifications and rapid growth in product functionality
  - Dynamic vendor relationships
  - SOA best practices growing in maturity

Managing SOA Risk

- **Risk Identification** (continued)
  - **Building ahead of your own maturity**
    - Increases risk of issues when building too many services into your platform
    - Requires experience to understand how to use them
    - May not properly prioritize evaluation criteria or may include criteria of non-value

SOA Migration Strategy

- **SOA Migration Strategy**
  - › Extract Information and Put it to Use
  - › Promote Reuse and Eliminate Redundancies
  - › Increase Visibility of Services Across Heterogeneous Platforms

SOA Migration Strategy

⦿ **Extraction of Information and Put it to Use**

> › Requires a carefully thought out data model

  - Traditional applications commonly require data from **external** business partners

  - By contrast, SOA creates cohesive silos which requires a strategy for integrating new and existing technologies

SOA Migration Strategy

⦿ **Promote Reuse and Eliminate Redundancies**

> › Important to move beyond the old concept that new applications require new code
>
> > • Old view defeats purpose of service reuse
> >
> > > • *Discourage people from re-inventing the wheel*

SOA Migration Strategy

- **Increase Visibility of Services Across Heterogeneous Platforms**
  - Requires ability to configure business orchestration and propagate it to the designated components across application deployment
    - Must apply the right operational rules before any communication occurs
  - Provide real-time information about components such as performance and, security

SOA Centric
Analysis and Design

Performing SOA Centric Analysis and Design

- **What Is the Best Solution for Building an SOA Platform?**
  - There isn't a best single sequence or solution for building an SOA platform
    - An SOA platform can start with messaging technologies such as HTTP, SOAP, REST, and message queuing
    - Factors such as diversity among organizations, and existing software infrastructures leads different *flavors* of SOA

Performing SOA Centric Analysis and Design

- **Identify your Existing Infrastructure's SOA Capabilities**
  - › Forms a basis for ensuring that existing capabilities are fully leveraged
    - • Prevents wasteful spending on new products that duplicate existing functions

Performing SOA Centric Analysis and Design

- **Identify your Existing Infrastructure's SOA Capabilities** (continued)
  - › Examine existing application infrastructure and management products against a functional model of an SOA platform
  - › Identify functions provided by existing products

Performing SOA Centric Analysis and Design

- **Identify your SOA Priorities**
  - › Provides a list of potential options for ensuring near-term success with SOA -based solution delivery projects
  - › Work from application road maps and service portfolio plans
  - › Identify types of service implementations required for high-priority services necessary to build over near term

Performing SOA Centric Analysis and Design

- **Identify SOA Priorities (continued)**
  - Determine whether existing infrastructure can fulfill requirements
    - If gaps, investigate SOA specialty product categories, such as ESBs or integration -centric business process management suites (IC-BPMSes)
    - Analysis may highlight ways specialty products may fulfill some requirements more effectively than existing infrastructure

Performing SOA Centric Analysis and Design

- **Match Platform Plans to Organization's Investment Strategy and Risk Profile**
  - Determine where actual investments in an SOA platform will fall along a continuum from:
    - Small
    - Single-project purchases to large,
    - Cross-project infrastructure build-outs

Service-Orientation and Object-Orientation

Service-Orientation and Object-Orientation

- **A Comparison of Goals and Concepts**
  - › Object-oriented analysis and design was responsible for popularizing the vision of applications designed to be:
    - • Reusable
    - • Flexible

Service-Orientation and Object-Orientation

- **A Comparison of Goals and Concepts** (continued)
  - OOAD grew out of a need for:
    - Service federation
      - Attempts to bring order to unstructured development processes
    - Instinct interoperability
      - Emphasizes the creation of code that closely mirrors real world

Service-Orientation and Object-Orientation

◉ **A Comparison of Goals and Concepts (continued)**

› Provides rules and guidelines that govern careful separation of application logic and data into objects

- Can be individually maintained

- Helps minimize the impact of change on the application as a whole

Service-Orientation and Object-Orientation

⊙ **A Comparison of Goals and Concepts** **(continued)**

> Service-orientation shares similar goals as OOAD

- Seeks to establish flexible design

- Allow for agile accommodation of business requirements

Service-Orientation and Object-Orientation

- **Service-Orientation**
  - › Similarly, service-orientation design is concerned with minimizing impact of change
    - Service Loose Coupling
    - Service Composability

Implementation

Implementation

- **Web Services**
  - › Is an implementation of SOA that enables *service* to be distributed across a network
    - Uses Interface definitions to achieve,
      Abstracted Components, Loosely Couple Components
    - Uses XML to achieve
      Autonomic Components, Stateless Components
    - Uses WSDL to achieve
      Standardized Service Contracts

Implementation

◉ **Web Services** (continued)

› Is an implementation of SOA that enables *service* to be distributed across a network

- Uses UDDI to achieve
  Discoverable Components

- Use Repositories to achieve
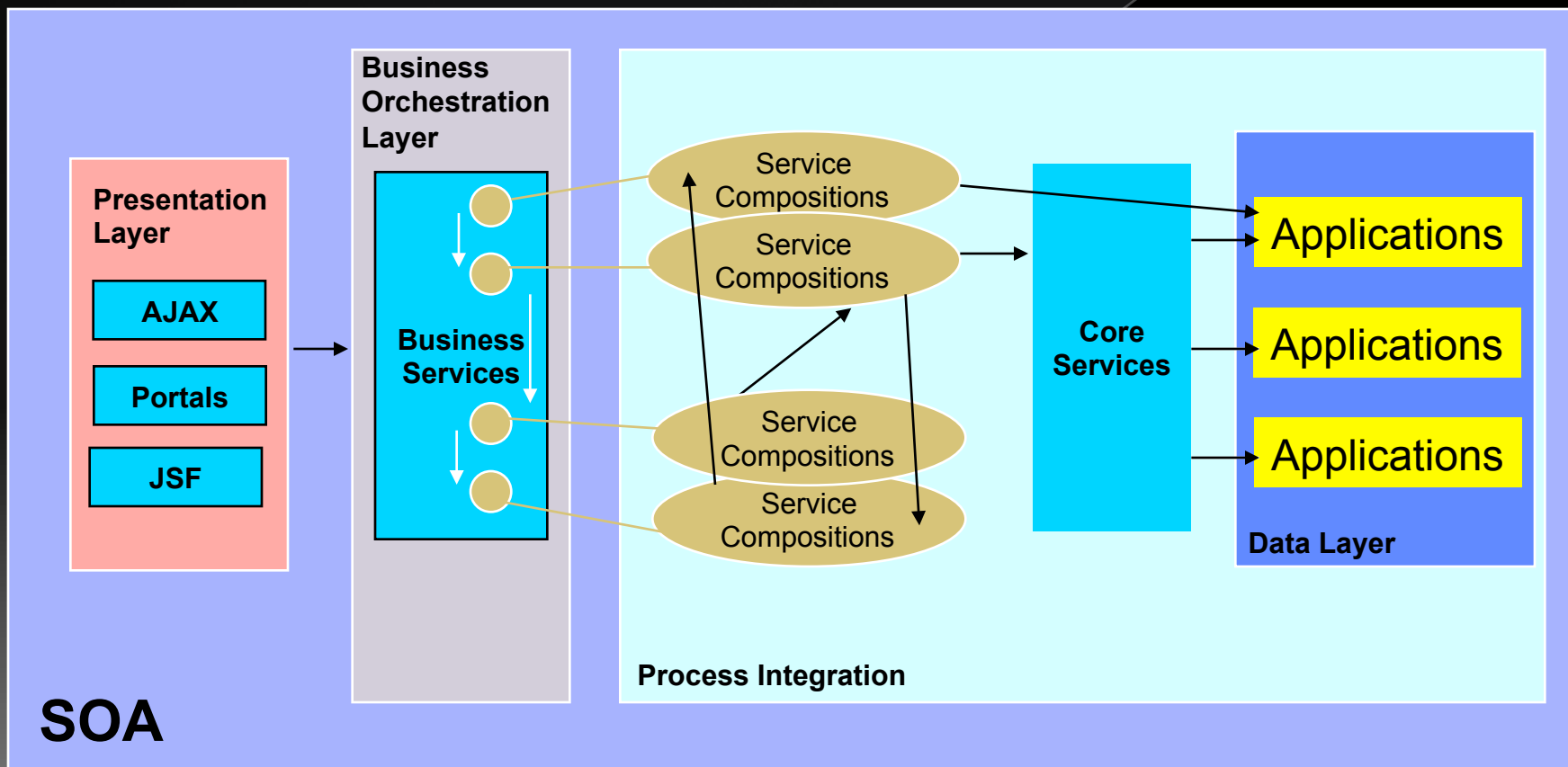 **Reusable Components**

Implementation

- ◉ **Services Layers**

  - Consequence of deploying SOA:

    - Heterogeneity across legacy systems is likely to increase

  - Benefit of deploying SOA:

    - Easier to manage heterogeneity and focus leveraging existing infrastructure instead of replacing it

  - *Wrapper Services* enable legacy applications to be reused and integrated into new SOA platforms, homogenously
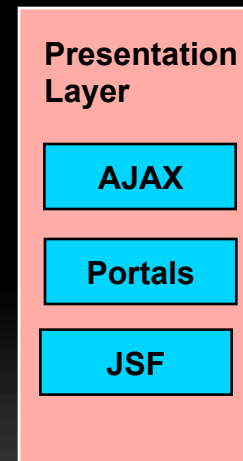
Implementation

- **Service Layers** (continued)

Presentation Layer:
- AJAX
- Portals
- JSF

Business Orchestration Layer:
- Business Services

Process Integration:
- Service Compositions
- Service Compositions
- Service Compositions
- Service Compositions
- Core Services

Data Layer:
- Applications
- Applications
- Applications

SOA

Implementation

- **Presentation Layer Wrapping**

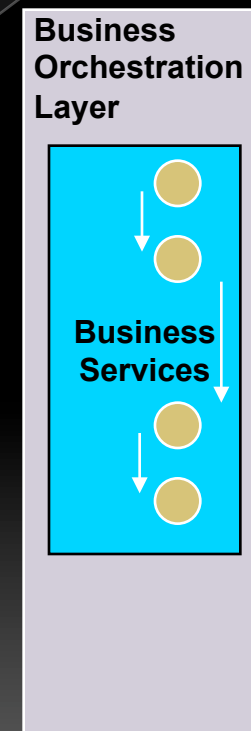 - Technologies such as AJAX, Portals and JSF make it possible for the UI to interact directly with backend services

Presentation Layer

AJAX

Portals

JSF

Implementation

- **Business Layer Wrapping**

  - Wrap business logic as
    services to communicate
    with other internal and
    external business services

  **Business
  Orchestration
  Layer**

  **Business
  Services**

Implementation

- **Data Layer Wrapping**

  - Wrap databases and legacy applications behind services so they can be exposed to other services

Applications

Applications

Applications

Implementation

- **Enhancing Connectivity and Flexibility with an Enterprise Service Bus (ESB)**

  - Enterprise Service Bus (ESB): essential run -time infrastructure for SOA

  - Primary functions of an ESB:

    - Ensure messages sent to and from services

    - Ensure messages arrive reliably at the right endpoint and in the right format

    - The business logic to be performed and messaged to be delivered in the correct sequence.

Implementation

- **Enhancing Connectivity and Flexibility with an ESB** (continued)
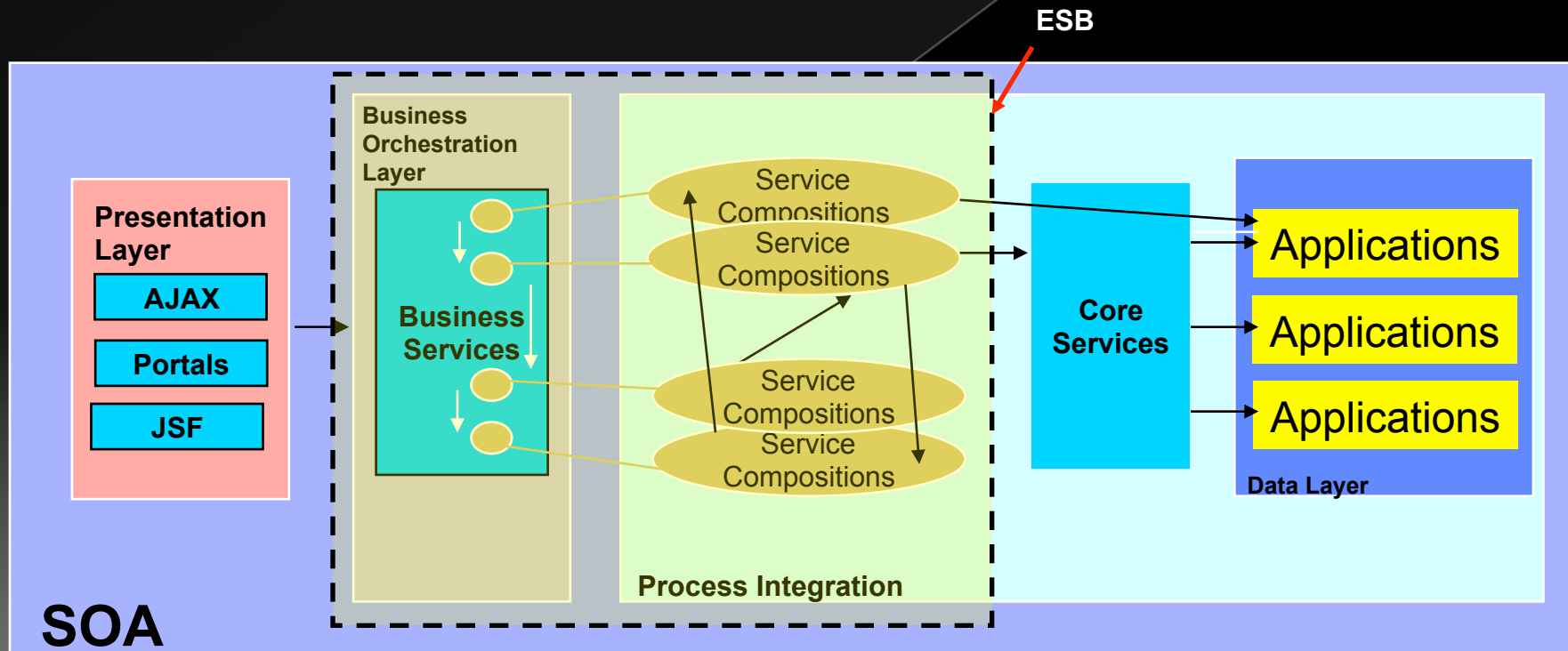
Implementation

- **Enhancing Connectivity and Flexibility with an ESB**
  - Provided in a highly-distributed and dynamically-changing environment
  - Support the performance, scalability and fault tolerance requirements
  - Vendors bundle different combinations of capabilities to orchestrate
    - Sequencing of business rules
    - Execution of business rules
    - Security

Conclusion

- **Invest in management and design to make sure the SOA project fits the business needs.**

- **Important to move beyond the old concept that new applications require new code.**

- **SOA needs to start from the beginning with the services and applications that business needs.**

This presentation and the complete book,
"SOA Right Away!"

Are available at **SoaRightAway.com**