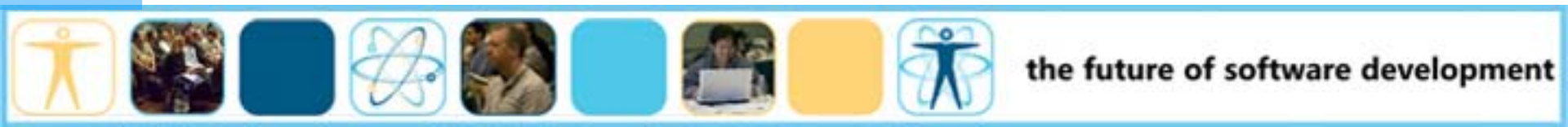
Introduction to Service Oriented Architecture

CSCI-5828 Foundations of Software Engineering

Ming Lian

March 2012



Executive Summary

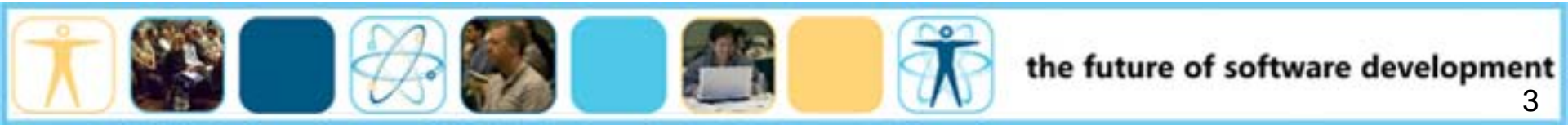
- This Executive Summary gives the straight word to the fresh that have interests in Service Oriented Architecture. It is going to give you the basic knowledge and principles of SOA, why to choose it, how it works and what the differences and benefits it can bring to us.

  The SOA Executive Summary specifically for students and developers who want to have some basic "SOA Literacy". The slides will cover SOA concepts, Service concepts, benefits of SOA, architecture comparison, terminology as well as standards of SOA.

- **What is a Services-Oriented Architecture (SOA)?**
  - When and why do we need a SOA?
  - How does a SOA work?
  - What makes a SOA different?
- **When and why would you use SOA?**
  - How does SOA support business application flexibility, agility, scalability and quality?
- **Who should adopt SOA?**
  - What kind of business and people would take advantages of SOA?

Goals

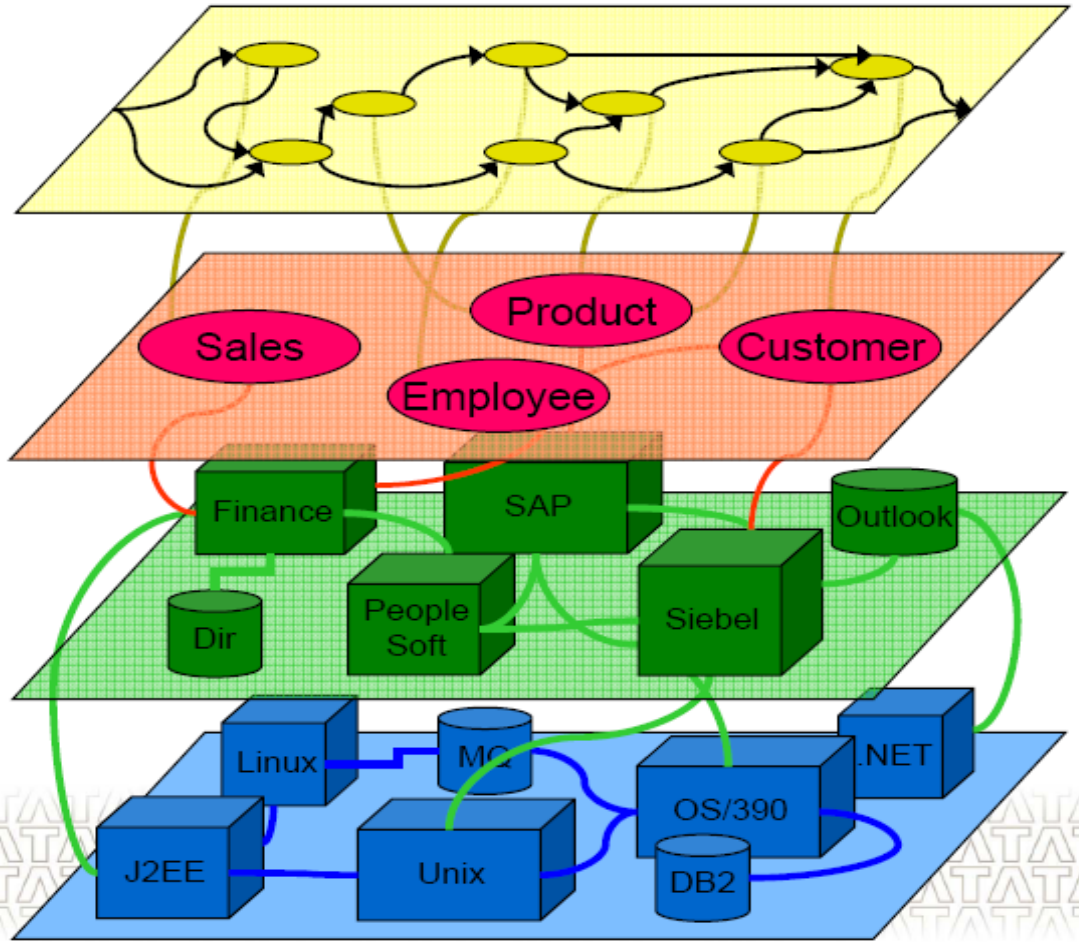
- Present an introduction to the topic of Service Oriented Architecture
  - What is Service?
  - What is SOA?
  - Why SOA?
  - SOA Architecture
  - Traditional Architecture VS SOA Architecture
  - Key standards and technology of SOA
  - Challenges
  - Who adopted SOA?



What is Services?

- Service is
  - component of distinctive functional meaning that typically encapsulate a high-level business concept
  - Lego block
- Service contains
  - Contract – message type def, constraint, description (comment)
  - Interface – set of operations
  - Implementation – Logic and data

Type of Services



Business Process

Composed of

Activities/Tasks/Service

Fulfilled by

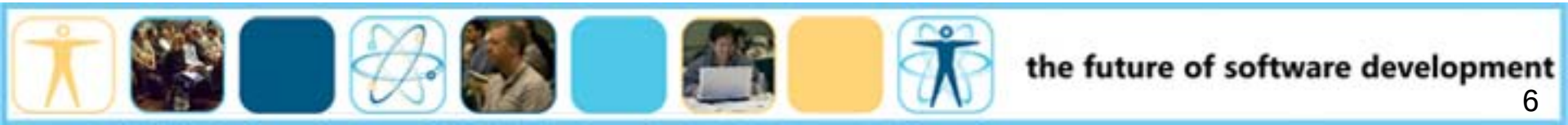
Applications/Packages

Supported by

Infrastructure

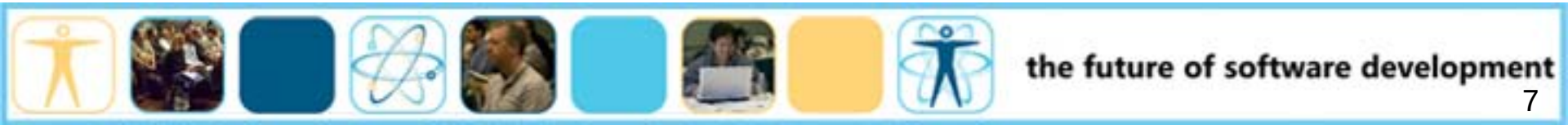
Examples of a Service

- Creating a Purchase Order inside a mainframe application
- Requesting and reserving a room in a hotel
- Applying for a loan by filling out a loan request form
- Search books/music based on keywords



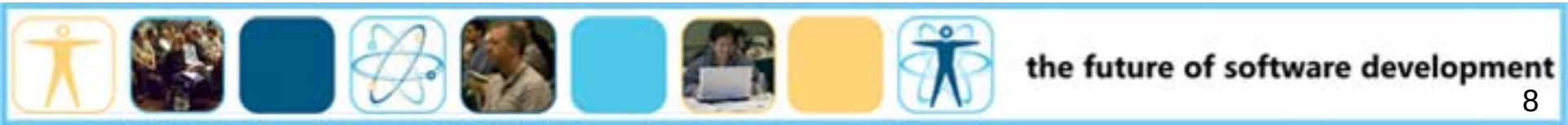
What is SOA?

- A set of *components* which can be invoked, and whose *interface* description can be published and discovered (W3C).
- Service-oriented architecture is a *client/server* design approach in which an application consists of software services and software service consumers (also known as clients or service requesters). SOA differs from the more general client/server model in its definitive emphasis on *loose coupling* between software *components*, and in its use of separately standing *interfaces* (Gartner).



What is SOA?

- Service-Oriented Architecture is a business-driven IT architecture approach that supports integrating your business as linked, repeatable business tasks, or services. SOA helps today's business innovate by ensuring that IT systems can adapt quickly, easily and economically to support rapidly changing business needs. SOA helps customers increase the flexibility of their business processes, strengthen their underlying IT infrastructure and reuse their existing IT investments by creating connections among disparate applications and information sources. (IBM)



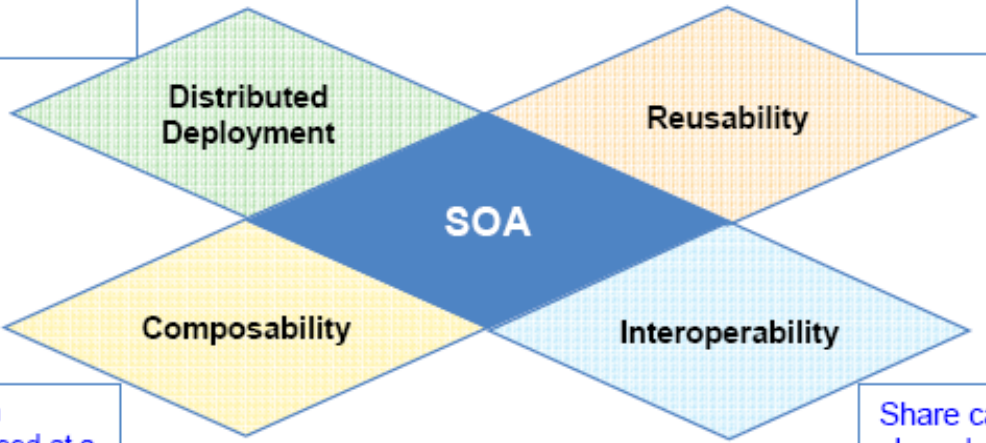
What is SOA?

Definition of SOA

SOA is an architectural approach that allows to:

Expose enterprise data and business logic as loosely coupled, discoverable, structured, standards-based, coarse-grained, stateless units of functionality called services

Choose a services provider and access to existing resources exposed as services

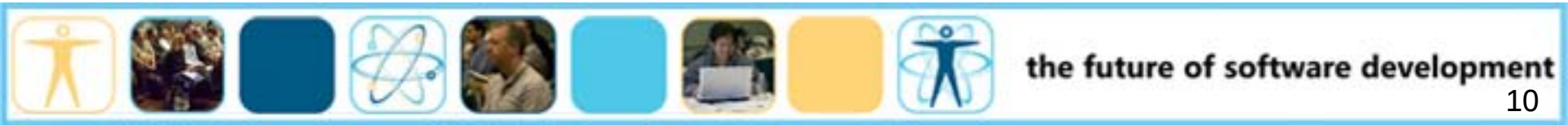


Assemble new processes from existing services that are exposed at a desired granularity through well-defined, published and standards-compliant interfaces

Share capabilities and reuse shared services across a network irrespective of underlying protocols or implementation technology

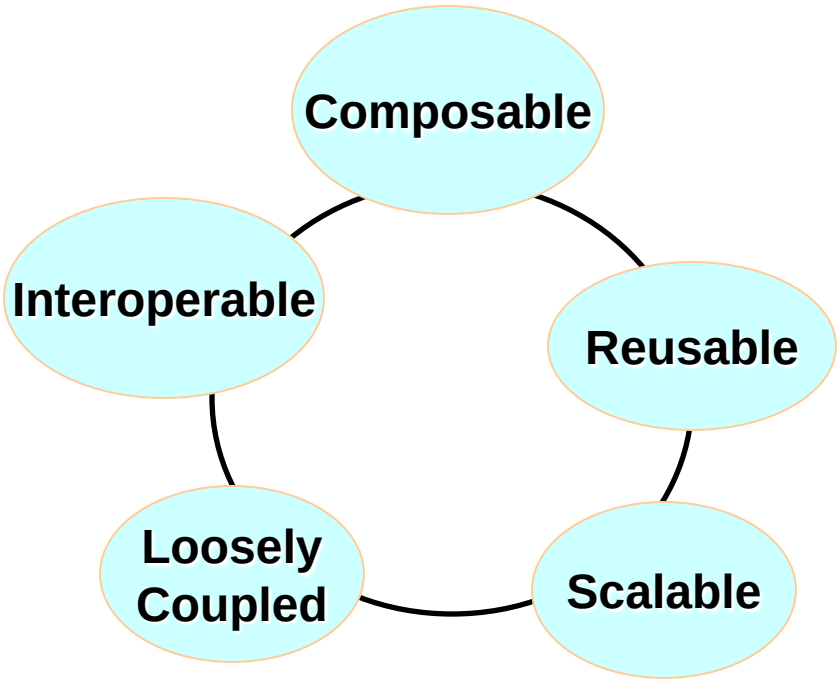
SOA Characteristics

- Based on open standards
- Foster inherent reusability
- Foster intrinsic interoperability
- Emphasizes extensibility
- Fundamentally autonomous
- Promotes dynamic discovery
- Promotes architectural composability
- Promotes loose coupling throughout the enterprise
- Supports incremental implementation



SOA Characteristics

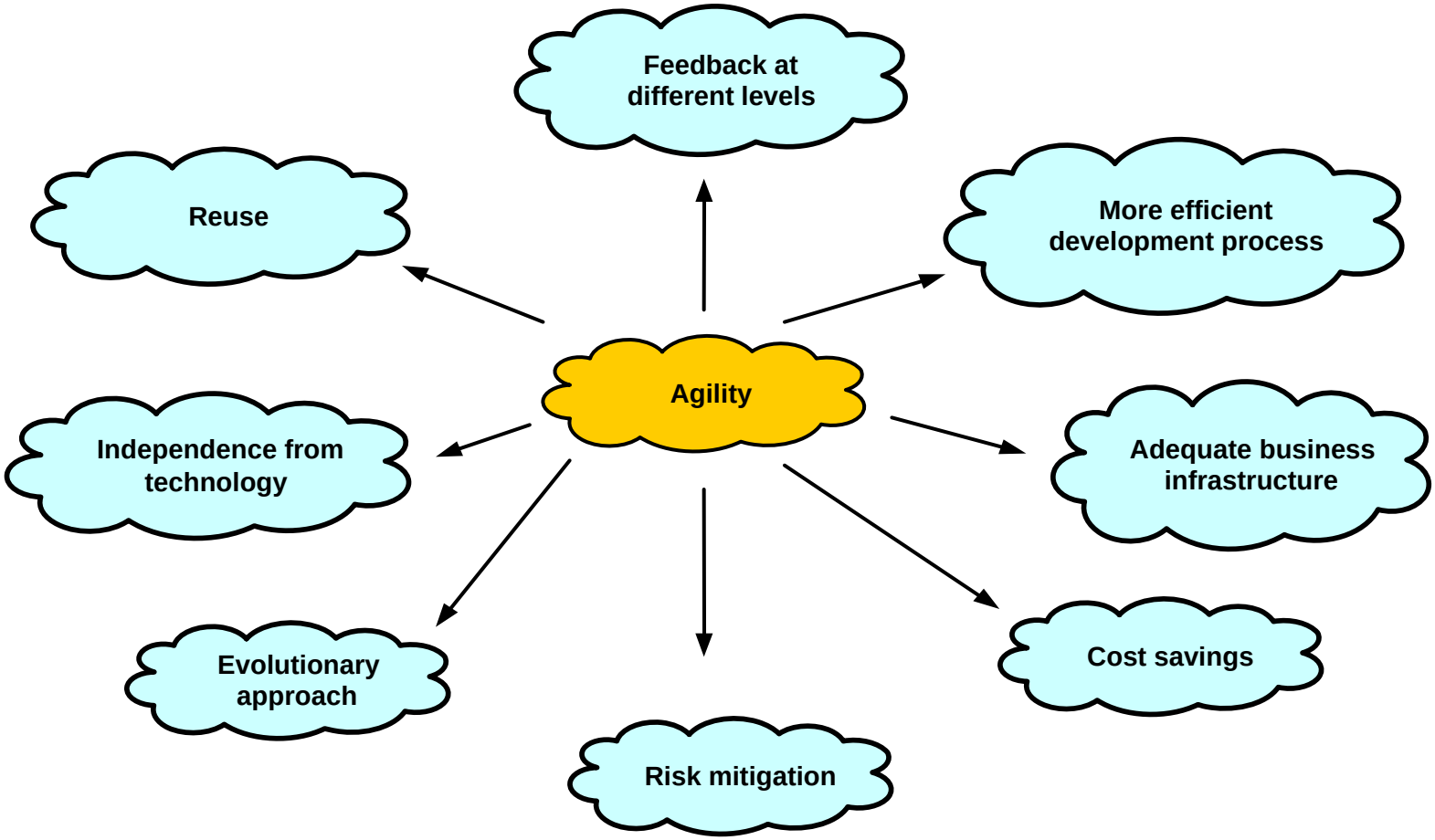
- Services are platform independent, **self describing interfaces** (XML)
- Messages are formally defined
- Services can be discovered
- Services have **quality** of service characteristics defined in policies
- Services can be provided on any platform
- Can be governed



Potential Benefits of SOA

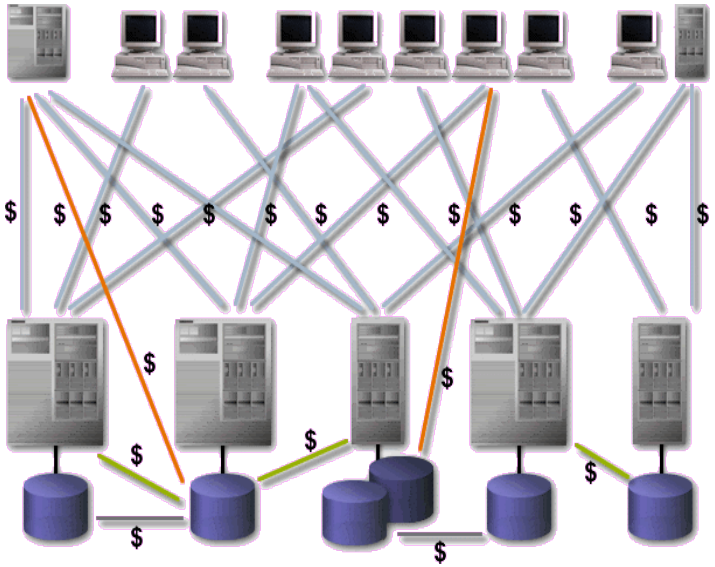
- Efficient and effective usage of ‘Business Services’
- Improved Integration, intrinsic interoperability
- Organizational agility
- Loosely-coupled with reusable assets and services
- Drives business processes closer to end users
- Leverage and integrate existing applications
- Provide standard connections between systems
- Abstract complexity for developers

Potential Benefits of SOA

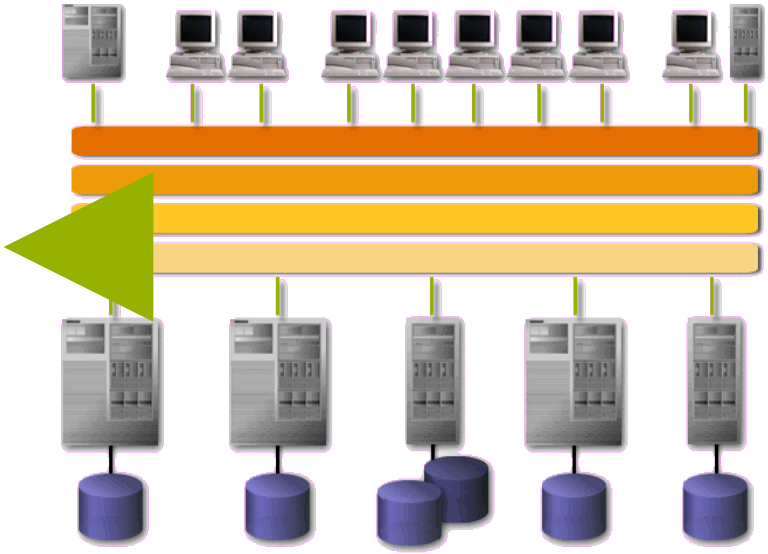


Shift to SOA

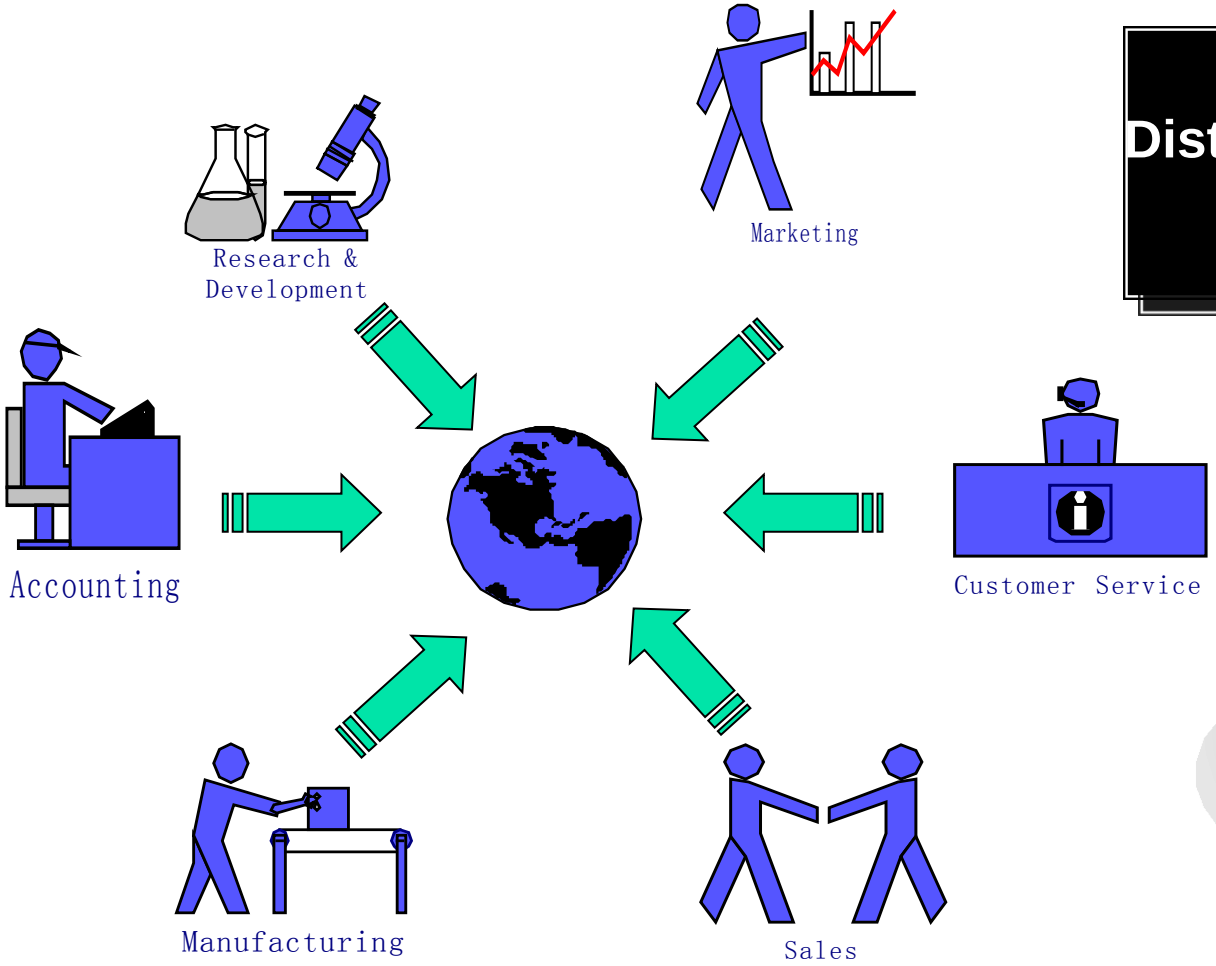
Accidental
Rigid
***Silo*-Oriented**



Layered
Extensible
***Service*-Oriented**



Why SOA?

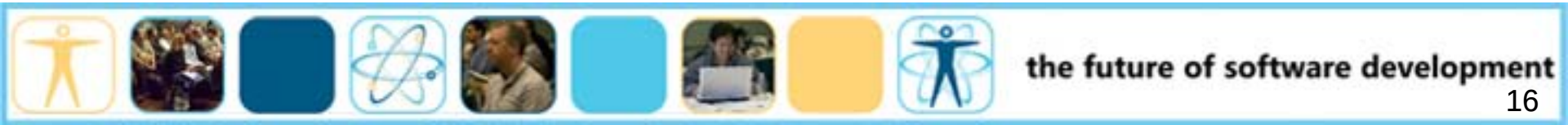


Distributed Data
Distributed Computation
Distributed users
.....

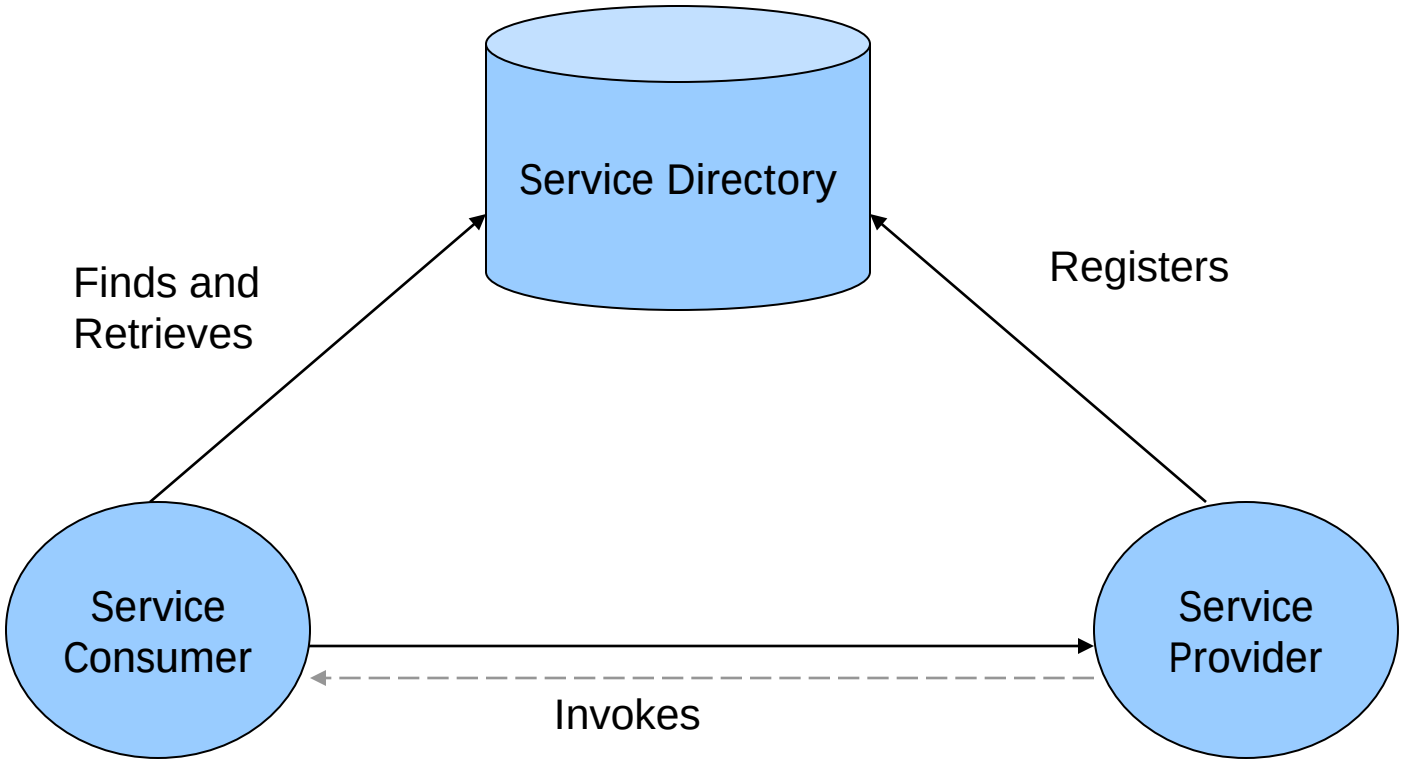
Distributed Business
Requires
Distributed Computing

Why SOA?

- Interoperation issues
  - Heterogeneous network protocols
  - Heterogeneous hardware platforms
  - Heterogeneous operating systems
  - Heterogeneous application formats
  - ……
- Increased Competitions
- Enhancement of Business Capabilities
- There must be consensus On **Interoperability**



SOA architecture



Traditional Architecture Vs Service Oriented Architecture

Traditional Architecture

ARCHITECTURE

- Components are tightly coupled
- Interface between subsystems is explicitly defined in terms a stack of protocols
- Known implementation
- Components are not independent of implementation attributes
- Tends to be closed architecture – Difficult to replace, or reuse components from one system to another
- Commonly, functions are accessible with the help of point-point connections over the network
- Tends to be confined to a single organization
- Based on standard set of layer – presentation, business, data access, Database

Service Oriented Architecture

ARCHITECTURE

- Loose coupling by means of services with standardized interfaces
- Application components communicate only through services and can be plugged in to any infrastructure that implements the standardized service
- Uses abstraction and is based on XML over SOAP
- Largely independent of implementation attributes
- Loosely coupling between interaction software components – leads to re-use of software components
- Designed to follow publically accessible models for consumption
- Meant for enabling participation of multiple organizations
- Requires additional layers
  - Business layer => Service and business model / components
  - Service Bus / Service Facade
  - BPM

Traditional Architecture Vs Service Oriented Architecture

Traditional Architecture

STANDARDS

- Involves only traditional J2EE and Web related standards
- Uses only HTTP
- Uses HTTPS for security
- More or less stable set of standards

USAGE

- Process centric
- Known context of usage

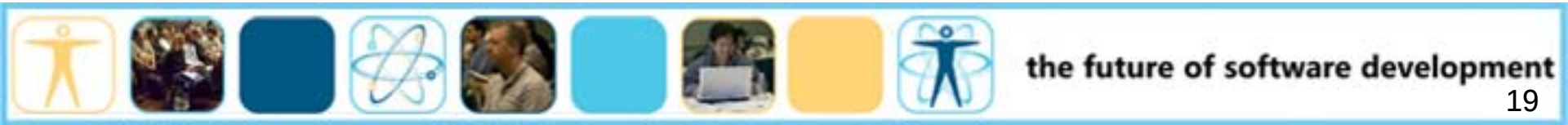
Service Oriented Architecture

STANDARDS

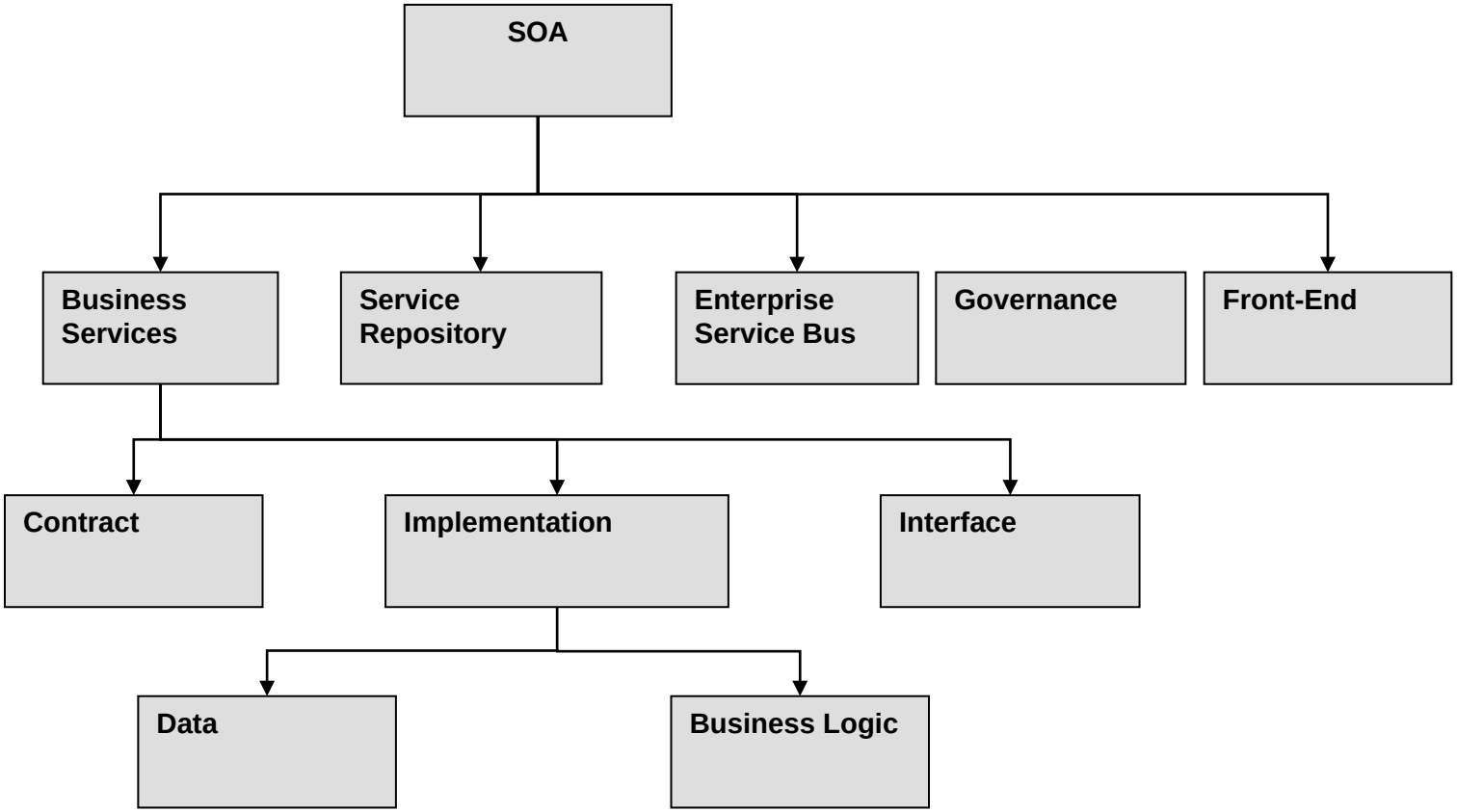
- Includes standards related to Web Service
- Builds a messaging layer above HTTP using SOAP
- Prefer WS-Security for end-to-end security
- Implementations must deal with evolving set of standards

USAGE

- Workflow centric
- To a large extent, future context of usage unknown at the time of design i.e unknown users and usage platforms



Key components of SOA



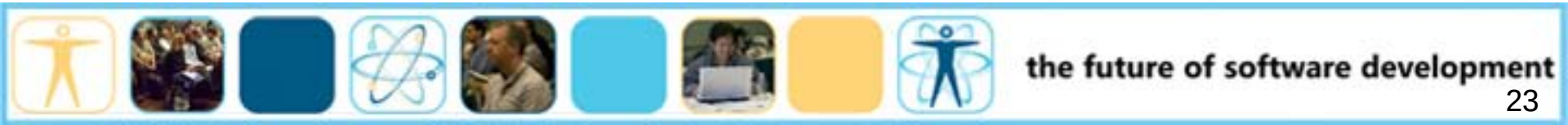
Key components of SOA

- Services (common denominator)
- Service Description
- Advertising and Discovery
- Specification of associated data model
- Service contracts

Associated Terminology

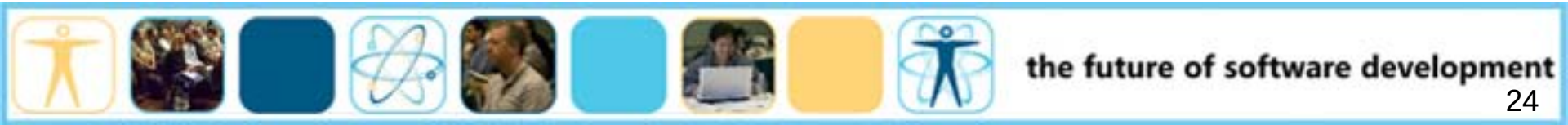
Associated Terminology

BPO	Business Process Outsourcing
BPM	Business Process Management
ESP	Enterprise Service Provider
GDM	Global Delivery Model
SOA	Service Oriented Architecture
SODA	Service Oriented Development of Applications
SOBA	Service Oriented Business Applications
SOE	Service Oriented Enterprise
WS	Web Services

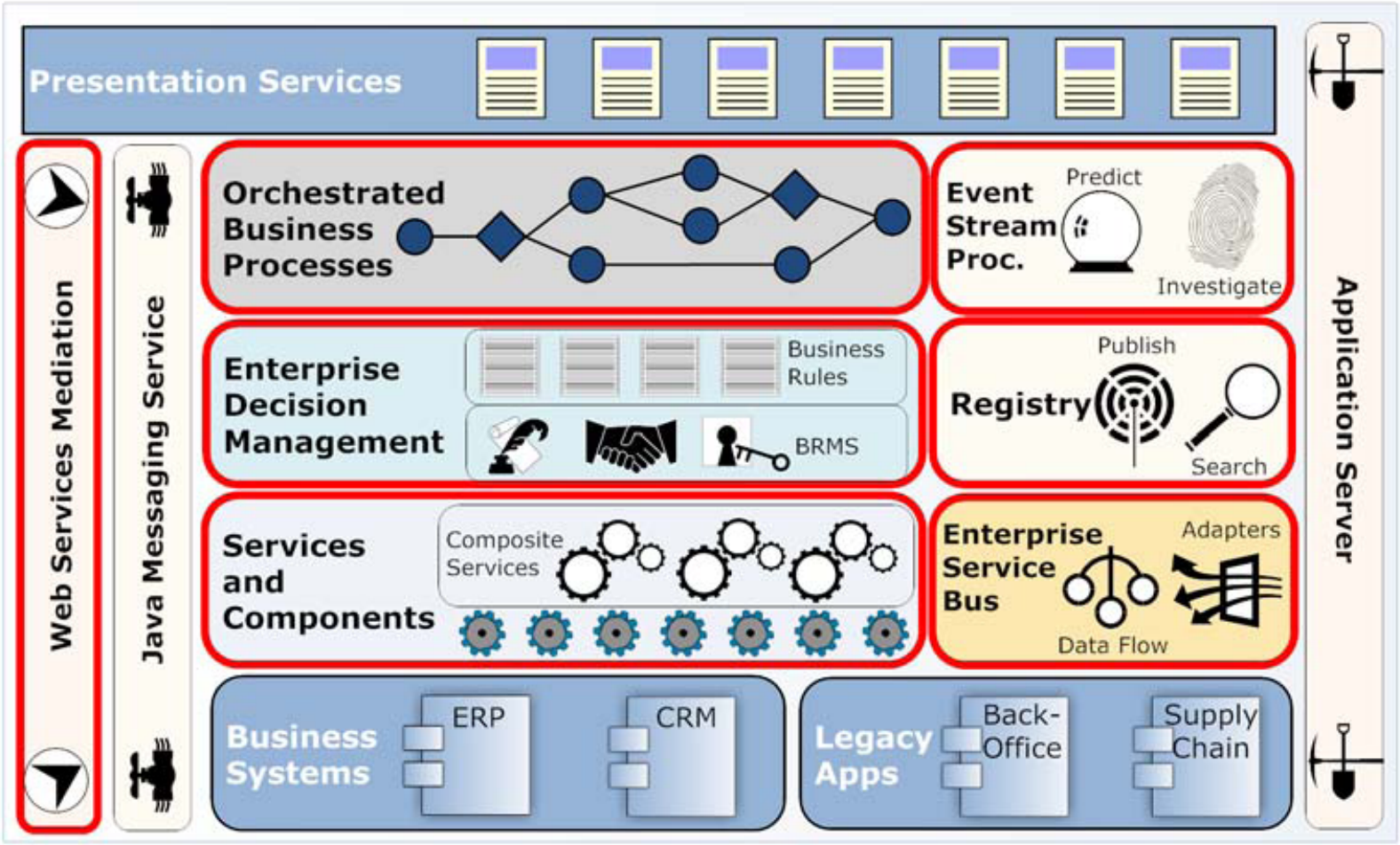


Key Standards of and Technology of SOA

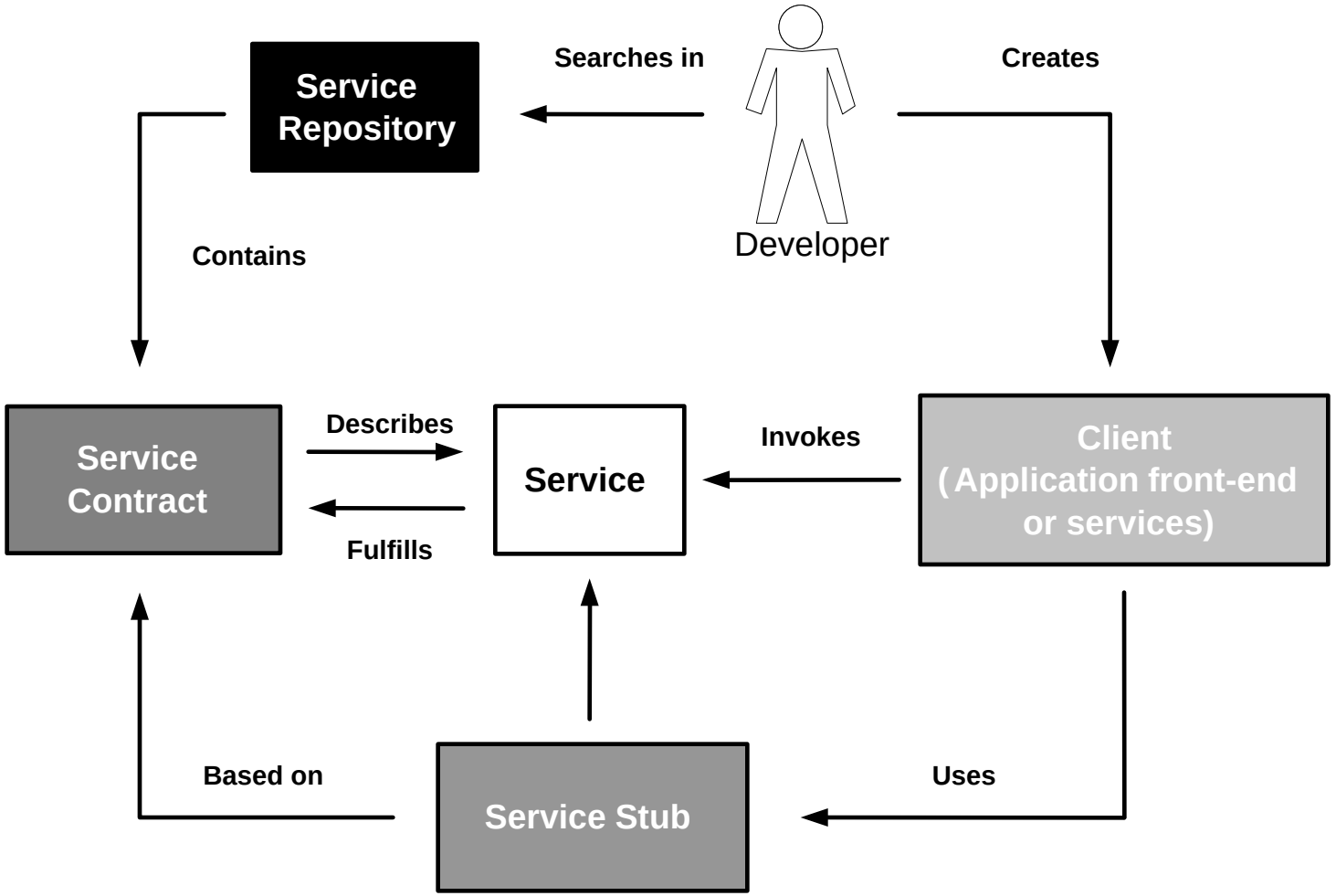
XML	• Markup Language designed to carry/transport data • Structure of the document i.e. the tags can be user defined based on the data being transported
Web Services	• Loosely coupled software components delivered over Internet standard technologies
SOAP	• Message format communication between parties involved in a web service
WSDL	• Mechanism for describing a web service in a platform independent way
UDDI	• Facilitates registration and organization of web service descriptions into a searchable directory



SOA Platform



How Does SOA Work?

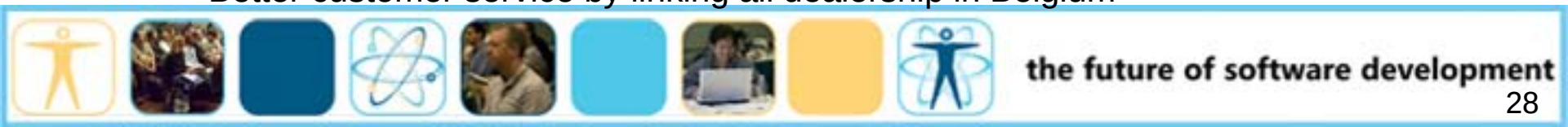


Challenges of SOA

- Technical Challenges
  - Security challenges - loosely coupled environment
  - Performance - XML brings robustness not speed
  - Optimization
  - Organizing the services – registry & repository
  - Finding the right services and right interfaces
  - Transaction management is complex in interactions between logically separate system

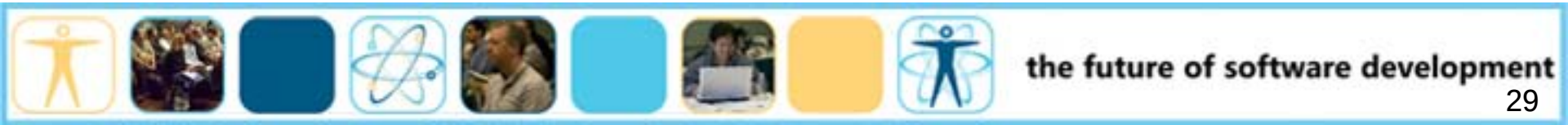
Where SOA made a difference

- **eBay**
  - Abstracting enterprise information
  - Helped to mange more than 2 perabytes of data
- **IBM**
  - 77 shareable and reusable services in production
  - Reduced application inventories
- **Hewlett Packard**
  - Reuse across services
  - Cutting operational costs
- **Amazon.com**
  - Handle 60 million customers and one million partners
  - Handle growing Transactional load
- **Citi Group**
  - Governace
  - Enable “separation of powers” among corporate, divisions, departments
- **DreamWorks**
  - Simplify and consolidate key business operations
  - Use SOA to make movies a easier process
- **Volvo**
  - Better customer service by linking all dealership in Belgium



Wrapping Up

- Adopting SOA is essential to deliver the business agility and IT flexibility promised by Web Services.
- SOA enables dynamic collaboration among loosely coupled, reusable software components through standard Internet protocols.
- SOA not only has many potential benefits to business and IT model but also holds several challenges that need to be solved in future improvement.
- SOA made differences to many entrepreneurs including IT, Sale, Financial and manufacture.
- SOA needs a bunch of standards and technologies to support that are widely deployed and acceptable



Resources

Books:

SOA : Using Java Web Services
- by Mark D. Hansen

Service-Oriented Architecture (Concepts, Technology and Design) - by Thomas Erl

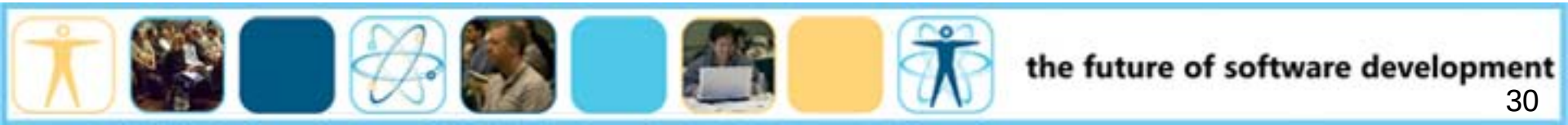
Web resources:

Amazon Web service:
http://en.wikipedia.org/wiki/Amazon_Web_Services

Oracle SOA
http://en.wikipedia.org/wiki/Oracle_SOA_Suite

Service component architecture
http://en.wikipedia.org/wiki/Service_component_architecture

Service-oriented analysis and design
http://en.wikipedia.org/wiki/Service-oriented_analysis_and_design



Resources

Open ESB

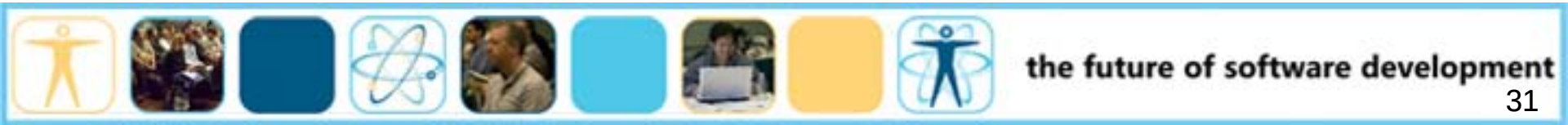
http://en.wikipedia.org/wiki/Open_ESB

Service-Oriented modeling

http://en.wikipedia.org/wiki/Service-oriented_modeling

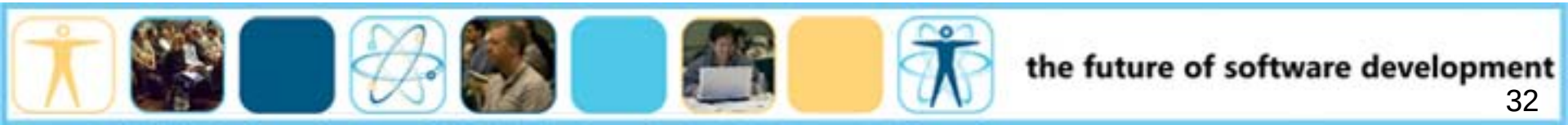
http://www.tibco.com/solutions/soa/default.jsp

http://www.microsoft.com/biztalk/solutions/soa/overview.mspx



References

- Douglas K. Barry, Web Services and Service-Oriented Architectures: the savvy manager's guide.
- Thomas Erl, Service-Oriented Architecture: concepts, technology and design.
- Thomas Erl, Service-Oriented Architecture: a field guide to integrating XML and web services.
- ObjectWeb, http://middleware.objectweb.org/
- OMG, http://www.omg.org/
- Doug Schmidt's CORBA page, http://www.cs.wustl.edu/~schmidt/corba.html



References

- Alan Mateus and Danilo Vieira, SOA Platform Technologies
- IBM SOA glossary, http://www-306.ibm.com/software/solutions/soa/glossary/index.html
- http://www.slideshare.net/Zubin67/soappt-3988559
- http://www.slideshare.net/Byungwook/soa-overview
- http://www.slideshare.net/datainc/introduction-to-service-oriented-architecture