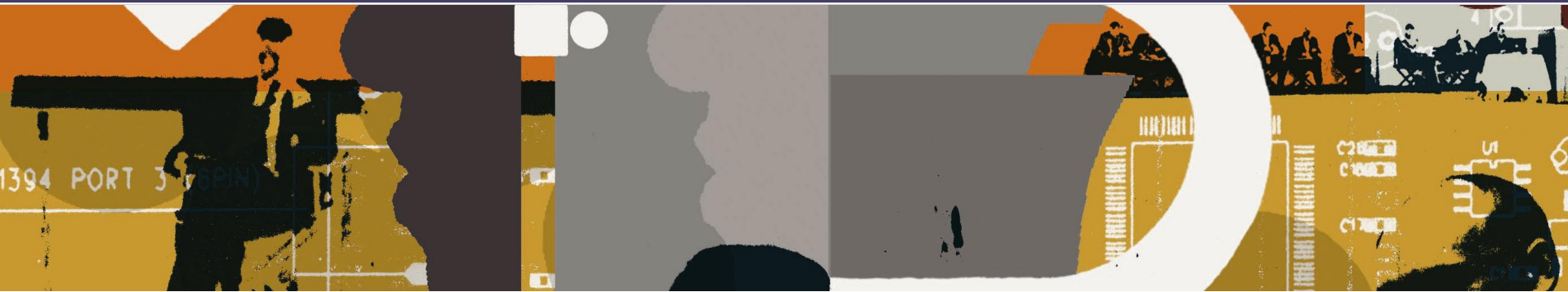


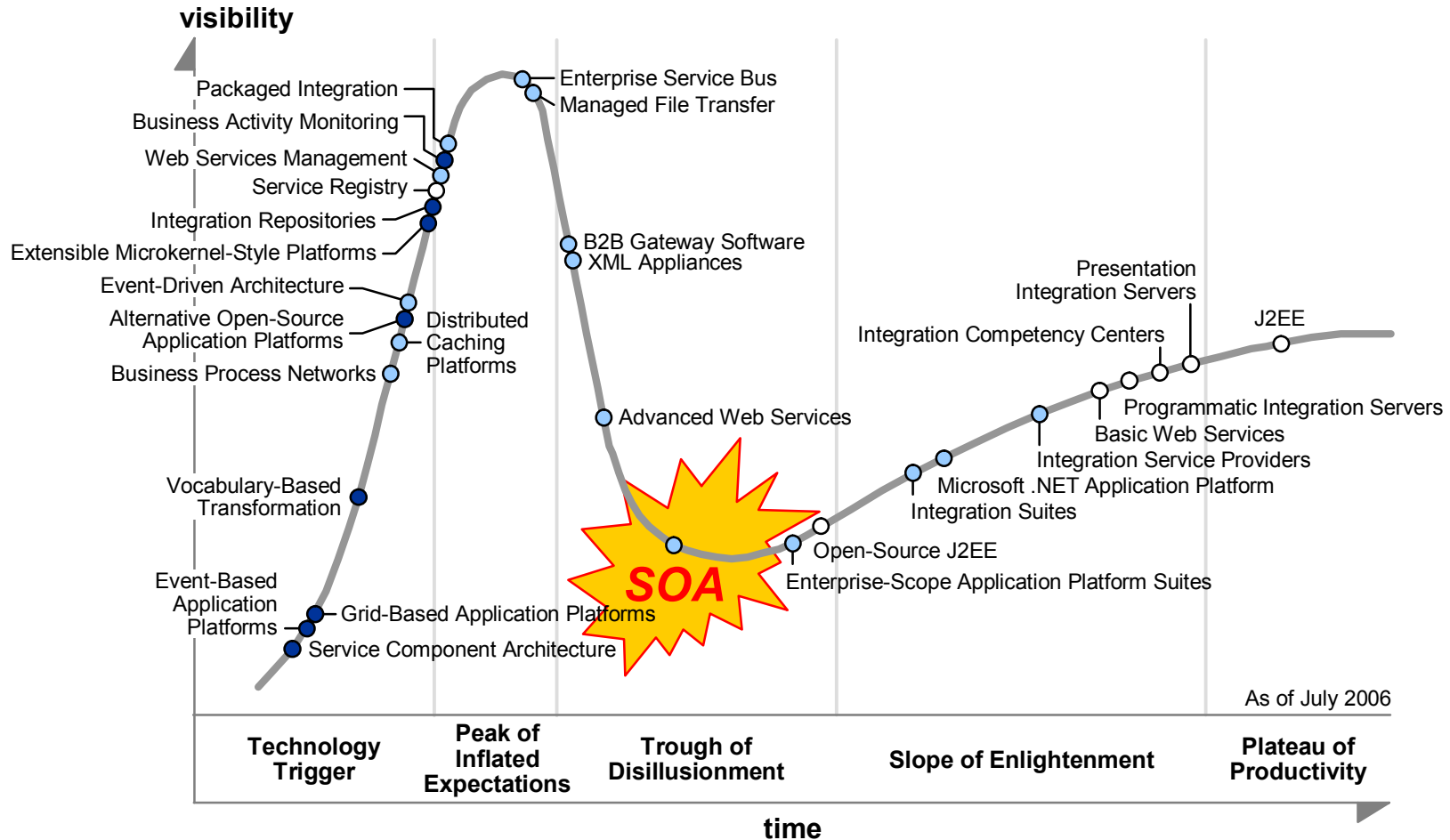
# SOA Down to Earth: What You Should and Shouldn't Do to Make It Happen



Massimo Pezzini

# SOA in the Trough of Disillusionment: Bad News or Good News?

## Application Integration & Platform Middleware Hype Cycle, 2006



Years to mainstream adoption:

○ less than 2 years

● 2 to 5 years

● 5 to 10 years

▲ more than 10 years

○ obsolete

⊗ before plateau

# Key Issues

1. Why should organizations adopt SOA?
2. What are the critical phases organizations will go through to get to enterprisewide SOA?
3. Which are the key hurdles on the way to SOA, and how can organizations get over them?

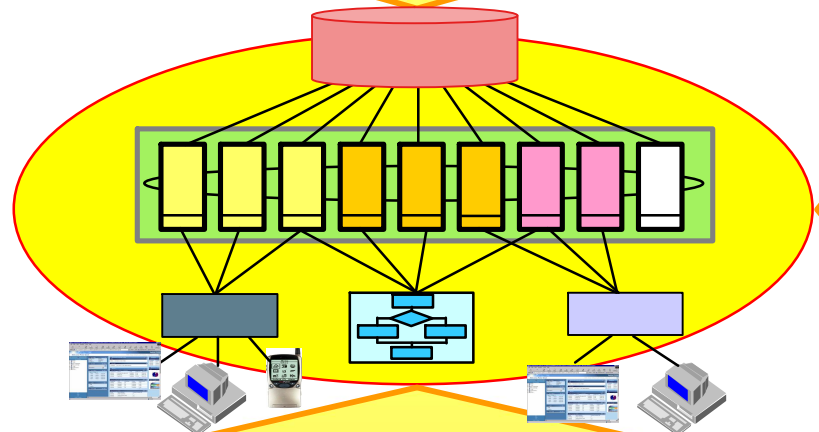
# Why Service-Oriented Architecture?

## Business Drivers Prevail Over IT Drivers

- M & A/Divestitures
- Multichannel sales/support
- Time-to-market
- Continuous innovation
- Process flexibility
- Process visibility

**"Top-Down" Enterprise Drivers**

**SOA**



**"Perennial" IT Challenges**

- "Doing more with less"
- Business/IT alignment
- Data consistency/quality
- Time-to-deployment

- Call center integration
- Single face to clients, suppliers, employees
- Process Integration
- Real-time B2B

**"Bottom Up" Business Unit Drivers**

# Beyond the SOA Hype: What's for Real?

## Benefits

### Architectural Partitioning

- Diverse life cycle "speeds"
- Synergy of different technologies
- Optimal tech skills allocation
- Processes visibility
- Greater maintainability
- Easier outsourcing/offshoring

### Incremental Deployment

- Gradual migration
- Cost "spreading" across projects
- Reduced maintenance cost

### Sharing (Reuse) of Services:

- Faster time to deployment
- Lower development cost
- Greater adaptability

### Common "Language" Between Business and IT

## Implications

### Higher Upfront Costs

- Cultural change
- Infrastructure (SOA backplane)
- More formal methodology
- Longer design time for services
- Testing (unit/end-to-end)

### More Distributed Infrastructure

- Extensive use of middleware
- Transaction management
- Debugging/troubleshooting
- End-to-end management
- More granular security
- Metering/logging

### Tighter Management/Governance

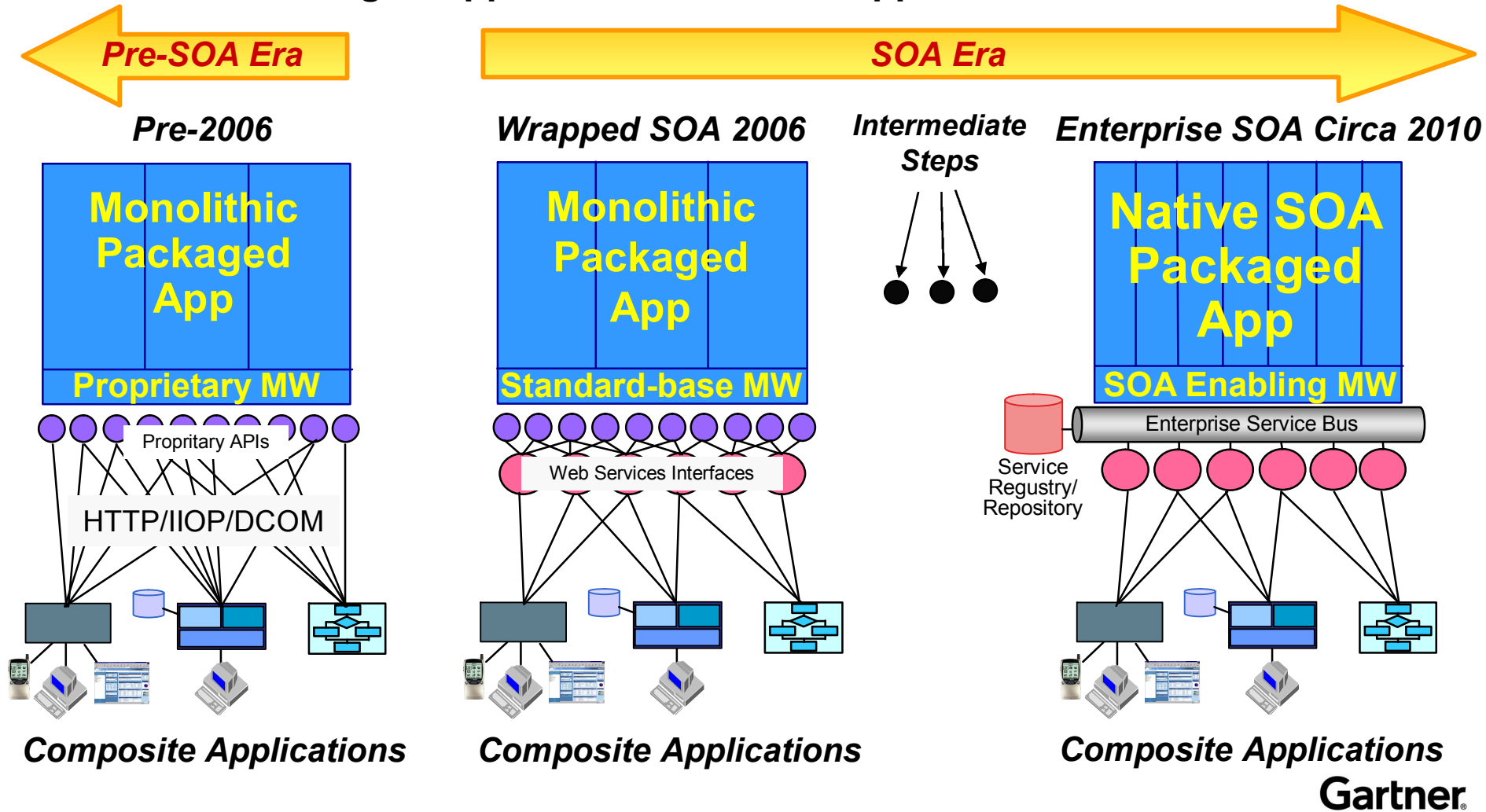
- Ownership/accountability
- Cost allocation
- Prioritization/conflict resolution

# Strategic Planning Assumption

Through 2008, the upfront investment for large-scale service-oriented applications will be justifiable only for projects with a planned lifetime of three years or more (0.8 probability).

# Packaged Business Applications and SOA: An Inevitable Marriage

## Packaged Applications Vendors' Approach to SOA



# Key Issues

1. Why should organizations adopt SOA?
2. What are the critical phases organizations will go through to get to enterprisewide SOA?
3. Which are the key hurdles on the way to SOA, and how can organizations get over them?

# Stages of SOA Adoption

	<b>Stage 1 Introduction</b>	<b>Stage 2 Spreading</b>	<b>Stage 3 Exploitation</b>	<b>Stage 4 Plateau</b>
<b>Business Goals</b>	Address Specific Pain (e.g., Customer Portal)	Process Integration (e.g., B2B)	Process Flexibility (e.g., Time to Market)	Continuous Adaptation & Evolution
<b>IT Goals</b>	Proof of Concept	Establish Technology Platform	Leverage Services Sharing	Enterprise SOA Infrastructure
<b>Scope</b>	Single Application	Multiple Applications (Single BU)	Multiple Applications (Cross BUs)	Virtual Enterprise
<b>No. of Published Services*</b>	<25	<100	<500	>500
<b>No. of Service Consumers*</b>	<5	<25	<50	>50
<b>No. of Service Calls/Day*</b>	<10,000	<100,000	<1,000,000	>1,000,000
<b>No. of Service Developers*</b>	<10	<20	<100	>100
<b>Enabling Technology (cumulative)</b>	Application Server, Portal, Adapters	ESB, WSM Integr. Suite, B2B	SOA Reg/Rep BPM Policy Mgmt.	Enterprise SOA Backplane

*\* =These figures represent typical scenarios, but they may vary considerably depending on the specific organization's requirements.*

# Which Stage of SOA Adoption Are You Ready For?

## Required Management Buy-in

## Required Skills

## Required Organizational Capabilities

	Stage 1 Introduction	Stage 2 Spreading	Stage 3 Exploitation	Stage 4 Plateau
Head of Development or Head of Integration	✓	✓	✓	✓
CTO/Head of Architecture	x	✓	✓	✓
Head of IT Operations		x	✓	✓
CIO/Business Units		x	✓	✓
CEO			x	✓
Basic Middleware	✓	✓	✓	✓
Web Services	✓	✓	✓	✓
Integration Middleware	x	✓	✓	✓
Service-Oriented Development of Applications (SODA)		x	✓	✓
Business Process Management		x	✓	✓
SOA Operations Management		x	✓	✓
SOA Center of Excellence	x	✓	✓	✓
Service Life Cycle Management		x	✓	✓
Service Design Methodology		x	✓	✓
Planning Control and Quality Management		x	✓	✓
Service Reuse Methodolog		x	✓	✓
Operation Management		x	✓	✓
Domains			x	✓
Cost Allocation Schema			x	✓
Consistent Enterprisewide Governance Processes			x	✓
Enterprisewide SOA Backplane			x	✓

✓ = Imperative  
x = Recommended

# Strategic Planning Assumption

Through 2010, fewer than 25% of large companies will have developed the technical and organizational skills needed to deliver enterprisewide SOA (0.8 probability).

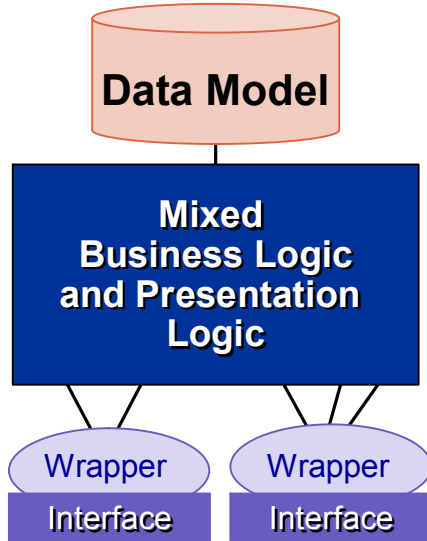
# Key Issues

1. Why should organizations adopt SOA?
2. What are the critical phases organizations will go through to get to enterprisewide SOA?
3. Which are the key hurdles on the way to SOA, and how can organizations get over them?

# SOA-Enabling Established Applications

## Wrapping

Use of screen-scraping to package "pseudo-services"



### Pros:

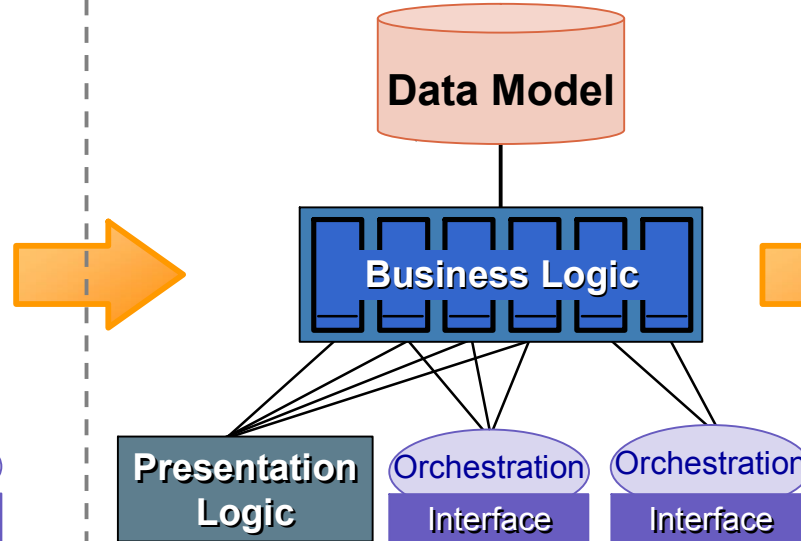
- Noninvasiveness
- Low cost/Fast

### Cons:

- Suboptimal granularity
- Hard to maintain

## Re-engineering

Business logic is modularized and separated from presentation



### Pros:

- Easier to maintain
- Better performance/scalability

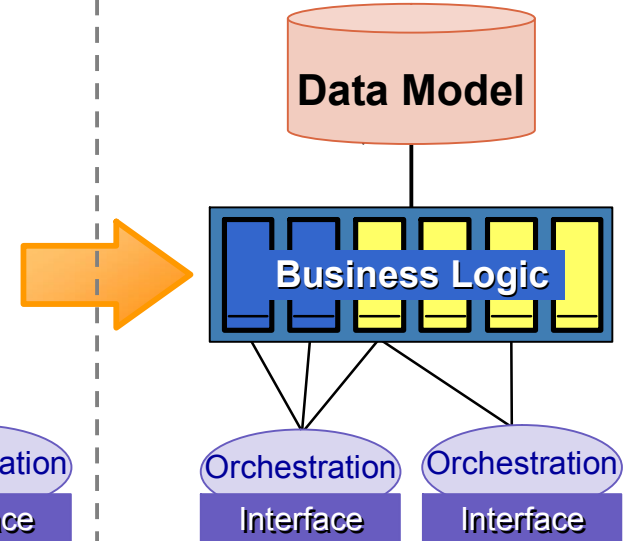
### Cons:

- Invasive/high cost
- Granularity might be suboptimal

## Introduction

## Replacement

Business logic of services is redesigned from scratch or replaces with packages.



### Pros:

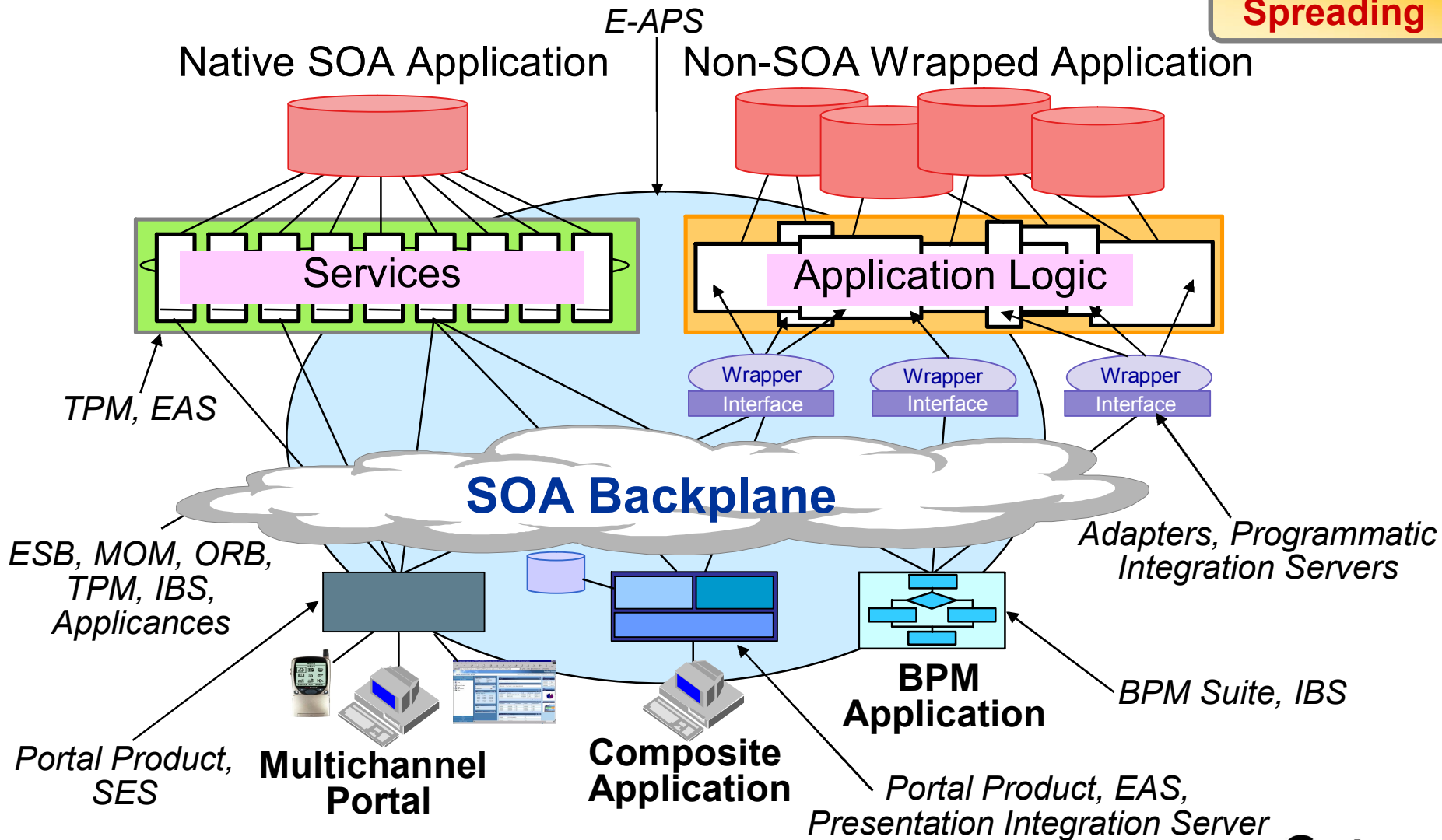
- Optimal granularity/reuse
- Enables technology change

### Cons:

- Risk is higher
- High cost

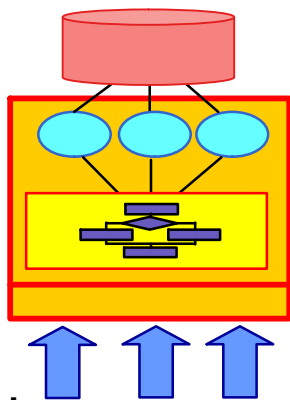
# SOAP and WSDL Are Not Enough: Orient Yourself Through the Middleware Bazaar

Spreading



# Flow Management in SOA: One Size Doesn't Fit All

## Microflows

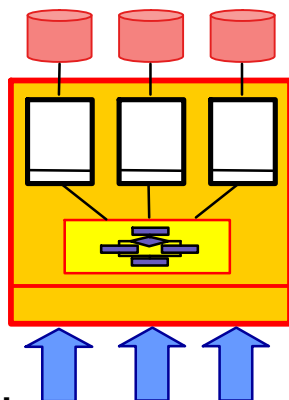


**Goal**  
Implement coarse-grained services by executing flows of "software components"

**Scope**  
Single application

- Key Features**
- Subsecond execution cycle
  - Transaction support
  - Usually nondistributed, single platform
  - State is not persisted

## Orchestration

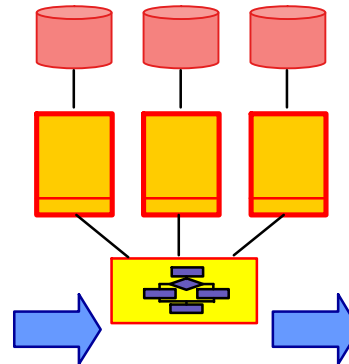


**Goal**  
Implement "coarser-grained" services by orchestrating "finer-grained" services

**Scope**  
Usually single domain

- Key Features**
- Subsecond to minutes execution cycle
  - Compensation flows (automated or manual)
  - Distributed/multiplatform
  - State may or may not be persisted

## Long-Running Process



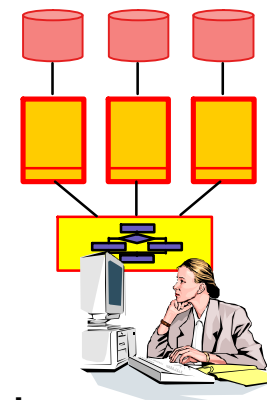
**Goal**  
Automate business processes involving multiple services

**Scope**  
Single/multiple domains

- Key Features**
- Minutes to hours execution cycle
  - Compensation flows (automated or manual)
  - Distributed/multiplatform
  - State must be persisted

## Spreading

## Workflow



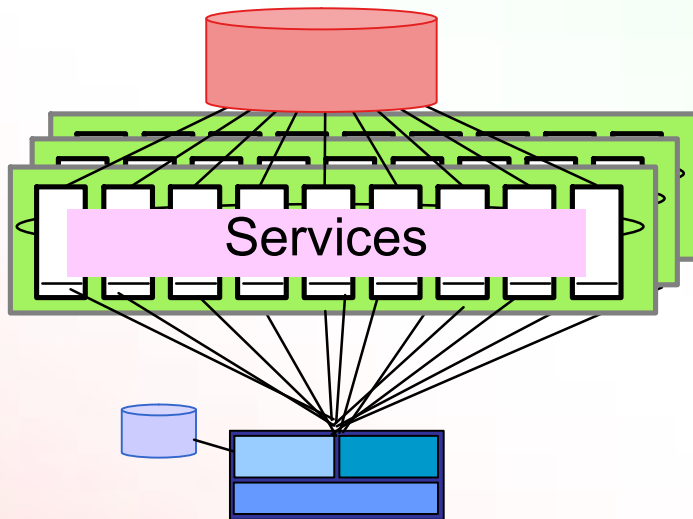
**Goal**  
Automate business processes involving human tasks and services

**Scope**  
Single/multiple domains

- Key Features**
- Minutes to months execution cycle
  - Compensation flows (automated or manual)
  - Distributed/multiplatform
  - State must be persisted

# How Do You Enforce Sharing of Services?

## "Bad" SOA



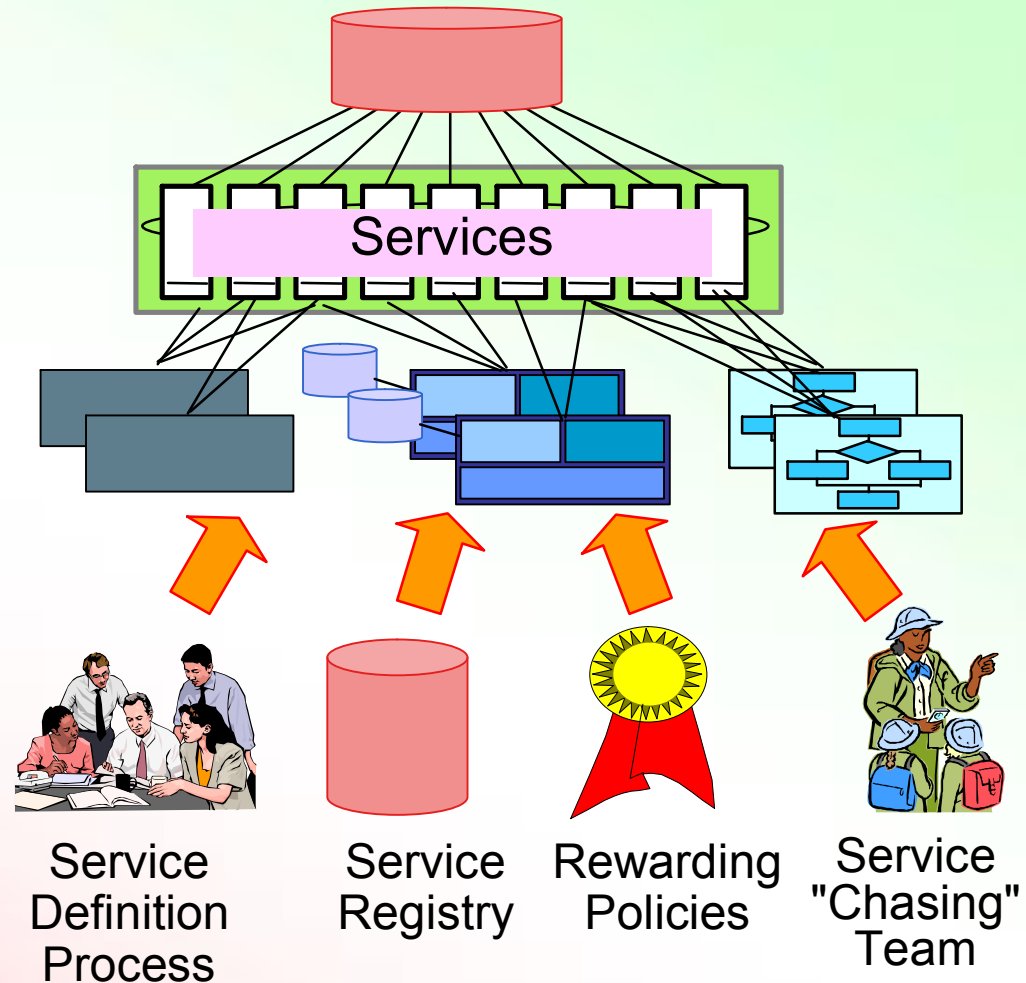
### Metrics

- # Services/# Consumers  $\gg 20$
- # of reused services  $< 10\%$

### Potential Issues:

- Too-fine services granularity
- Duplicated services
- Over-specified SOA

## "Good" SOA



Exploitation

Service  
Definition  
Process

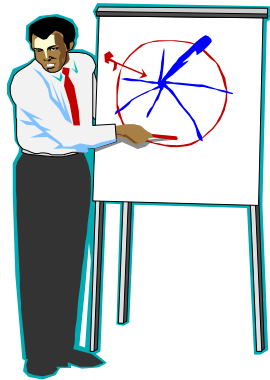
Service  
Registry

Rewarding  
Policies

Service  
'Chasing'  
Team

# Managing Life Cycle in SOA: The Service Registry

Governance and Cost Allocation

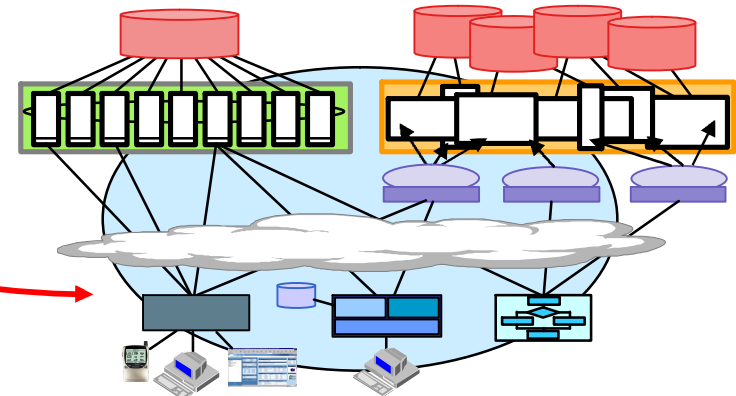
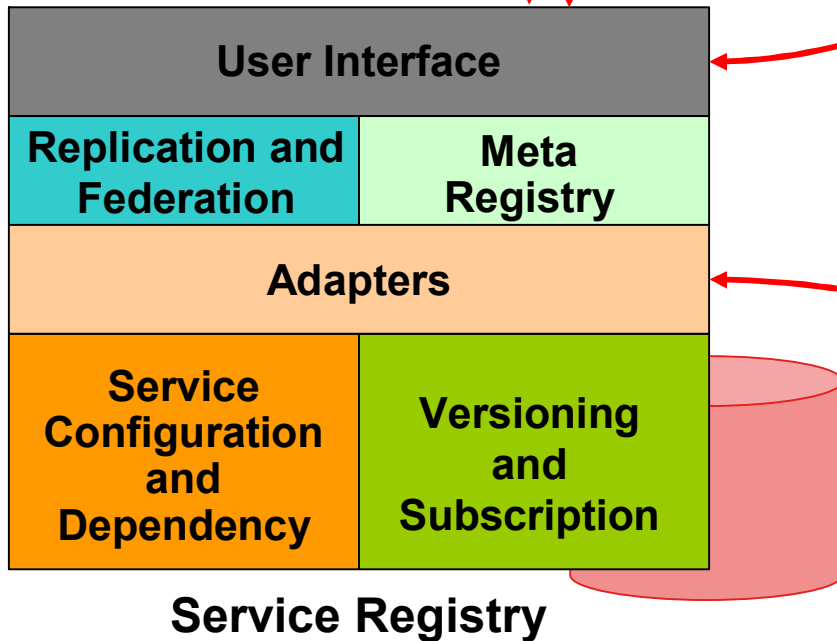
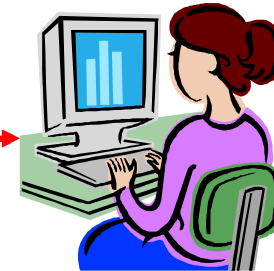


Definition and Validation



**Exploitation**

Development and Maintenance



Discovery, Location,  
Management and Security



# Recommendations

- ✓ Endorse SOA to be able to respond to business and IT challenges.
- ✓ SOA is not only about Web services. Build knowledge about middleware and application integration technologies as well.
- ✓ SOA is a journey. Plan for multiyear incremental implementation steps, but look for short- and medium-term payback.
- ✓ Avoid a "wild" proliferation of services. Establish processes focused on maximizing reuse.
- ✓ Beware of SOA "extremism." SOA principles should be applied with a grain of salt.

# SOA Down to Earth: What You Should and Shouldn't Do to Make It Happen



Massimo Pezzini

# SOA Down to Earth: What You Should and Shouldn't Do to Make It Happen



Massimo Pezzini