

Service Oriented Architecture Principles, Layers, and Benefits

SOA principles

- ♣ Service Encapsulation
- ♣ Service Loose coupling
- ♣ Service Contract
- ♣ Service abstraction
- ♣ Service Documentation
- ♣ Service reusability
- ♣ Service composability
- ♣ Service autonomy
- ♣ Service optimization and Discovery
- ♣ Service statelessness

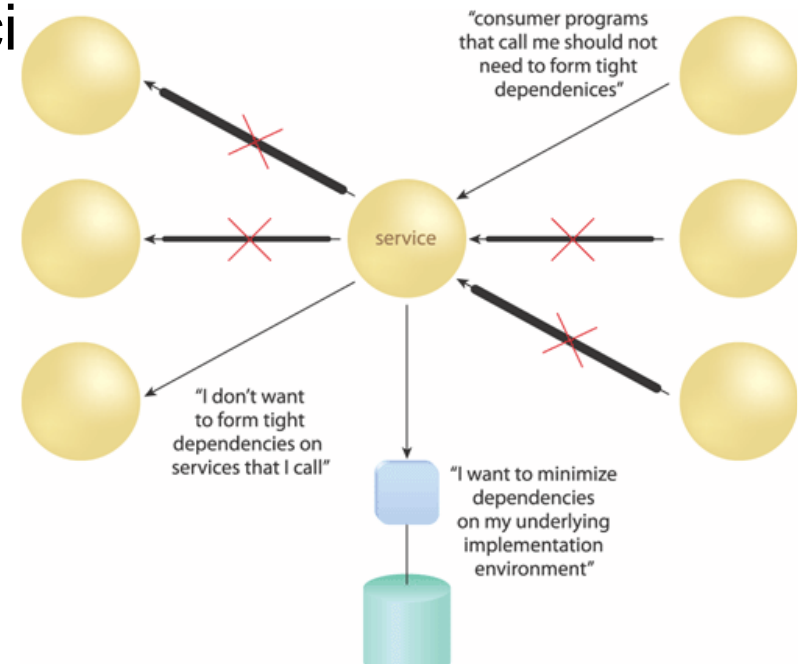
Loose Coupling

“Service contracts impose low consumer coupling requirements and are themselves decoupled from their surrounding environment.”

Create specific types of relationships within and outside of service boundaries with a constant emphasis on reducing (“loosening”) dependencies

between

- Service contract
- Service implementation
- Service consumers

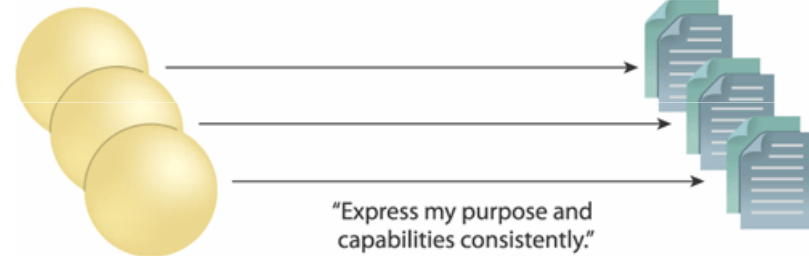


Standardized Service Contracts

▽ “Services within the same service inventory are in compliance with the same contract design standards.”

▽ Services use service contract to

- Express their purpose
- Express their capabilities



▽ Use formal, standardized service contracts

▽ Focus on the areas of

- Functional expression
- Data representation
- Policy

Source: Thomas Erl

Abstraction



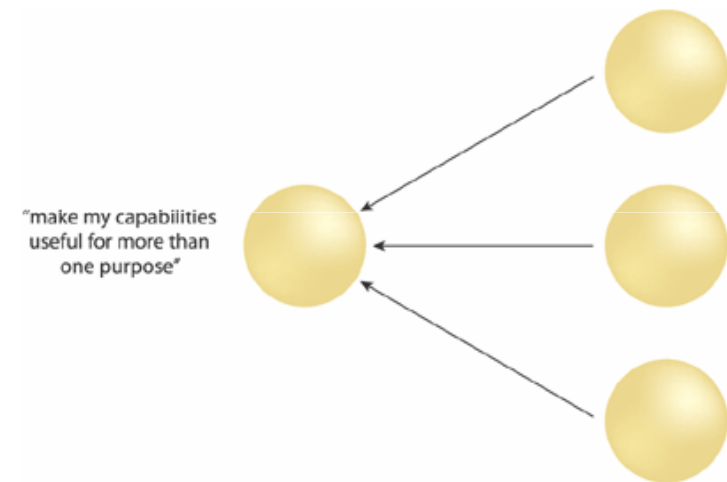
“only publish information
about me that others
absolutely need to know”



- ▽ “*Service contracts only contain essential information and information about services is limited to what is published in service contracts*”
- ▽ Avoid the proliferation of unnecessary service information, meta-data.
- ▽ Hide as much of the underlying details of a service as possible.
 - Enables and preserves the loosely coupled relationships
 - Plays a significant role in the positioning and design of service compositions

Reusability

▽ *“Services contain and express agnostic logic and can be positioned as reusable enterprise resources.”*



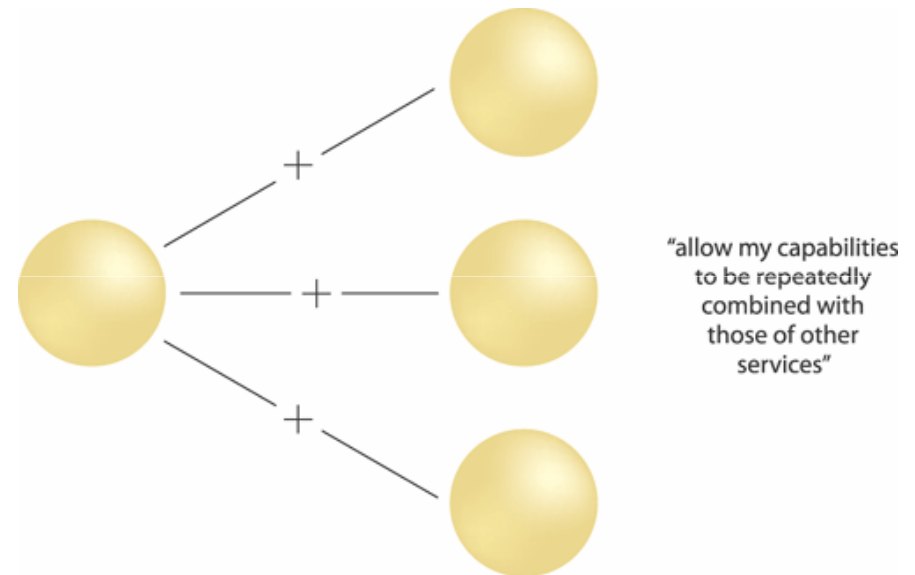
▽ Reusable services have the following characteristics:

- Defined by an agnostic functional context
- Logic is highly generic
- Has a generic and extensible contract
- Can be accessed concurrently

Source: Thomas Erl

Composability

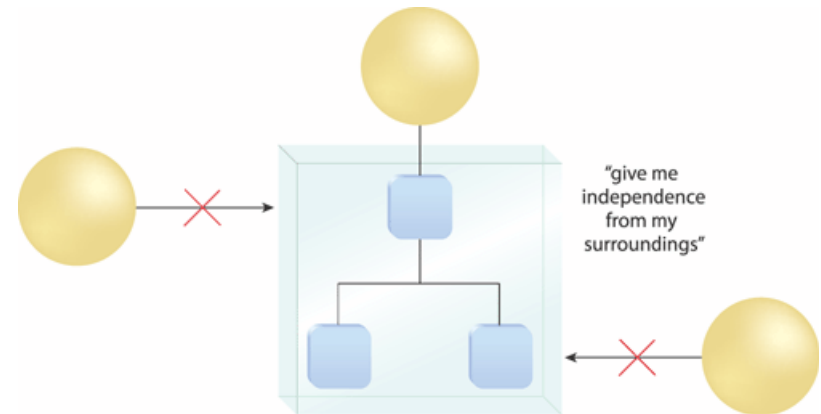
- ▽ *"Services are effective composition participants, regardless of the size and complexity of the composition."*
- ▽ Ensures services are able to participate in multiple compositions to solve multiple larger problems



Source: Thomas Erl

Autonomy

- ▽ *"Services exercise a high level of control over their underlying runtime execution environment."*
- ▽ Represents the ability of a service to carry out its logic independently of outside influences
- ▽ To achieve this, services must be more isolated
- ▽ Primary benefits
 - Increased reliability
 - Behavioral predictability



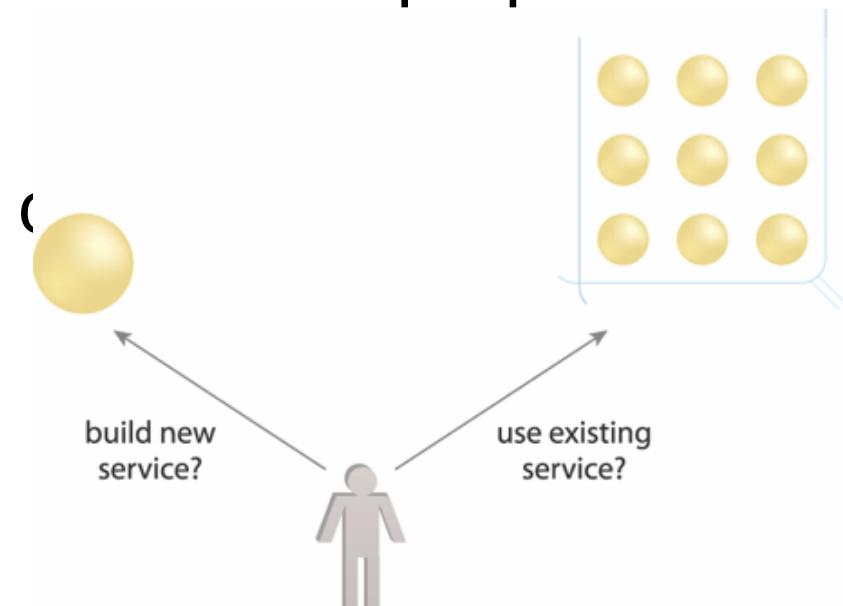
Source: Thomas Erl

Discoverability

- ▽ *"Services are supplemented with communicative meta data by which they can be effectively discovered and interpreted."*
- ▽ Service contracts contain appropriate meta data for discovery which also communicates purpose and capabilities to humans

- ▽ Store service documents meta registry

Source: Thomas Erl



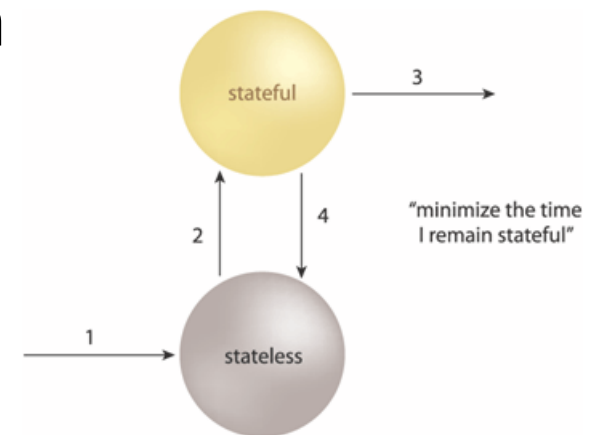
Statelessness

▽ *"Services minimize resource consumption by deferring the management of state information when necessary."*

▽ Incorporate state management deferral extensions within a service design

▽ Goals

- Increase service scalability
- Support design of agnostic logic and improve service reuse



Source: Thomas Erl

Applying SOA - Governance

- Governance is a program that makes sure people do what is 'right'
- In conjunction with software, governance controls the development and operation of software

Goal: Establish SOA organization governance (SOA Board) that governs SOA efforts and breaks down capabilities into non-overlapping services

Applying SOA - Governance

▽ Policies

- Codification of laws, regulations, corporate guidelines and best practices
- Must address all stages of the service lifecycle (technology selection, design, development practices, configuration management, release management, runtime management, etc.)

Applying SOA - Governance

∇ Processes

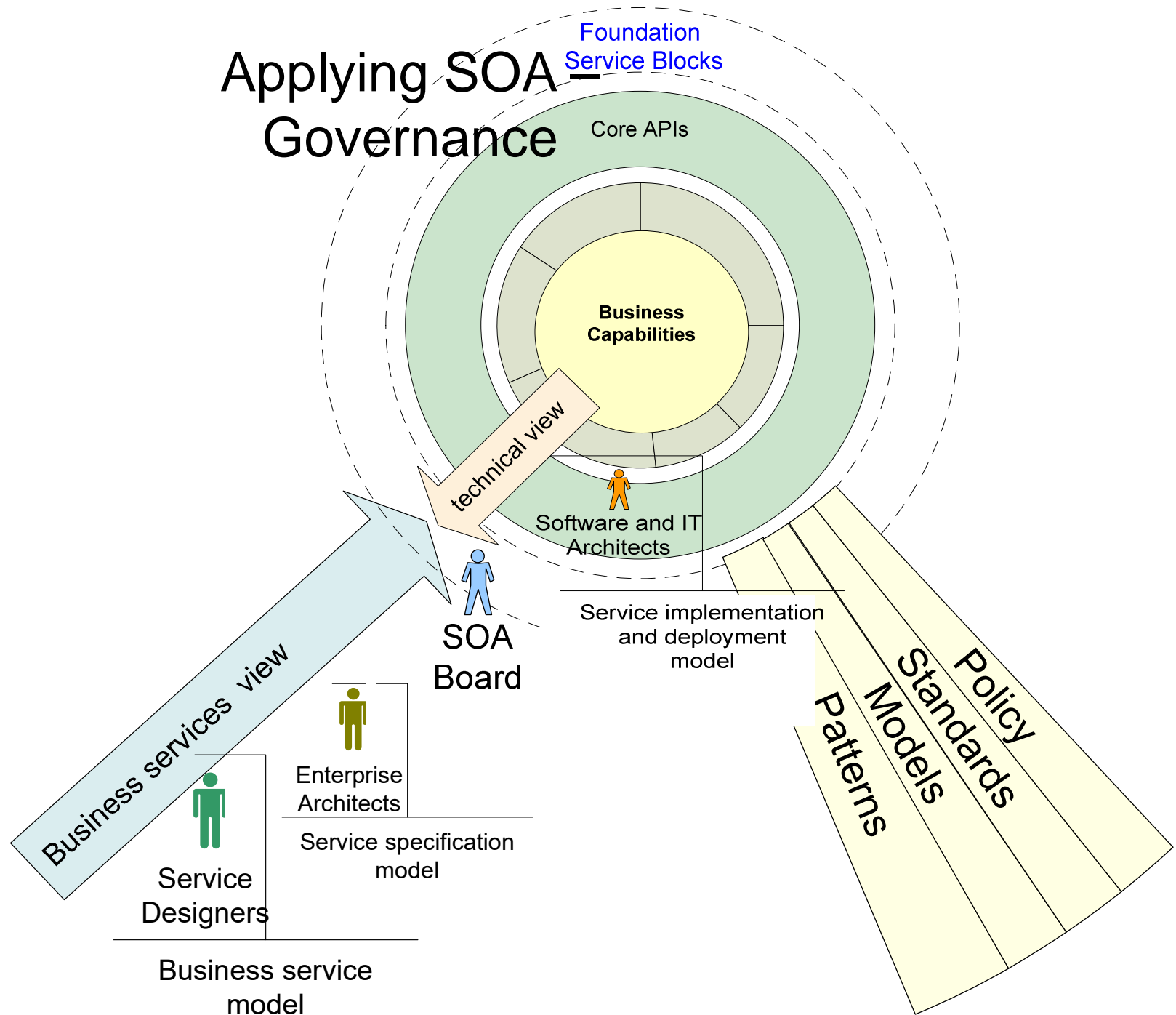
- Enforce policies
- System-driven processes (code check-in, code builds, unit tests)
- Human-driven process (requests, design reviews, code reviews, threat assessment, test case review, release engineering, service registration, etc.)

Applying SOA - Governance

▽ Metrics

- Measurements of service reuse, compliancy with policy, etc.
- Organization
- Governance program should be run by SOA Board, which should have cross-functional representatives

Applying SOA Governance



Applying SOA - Challenges

▽ Service Orientation

Business functionality has to be made available as services.
Service contracts must be fixed

▽ Reuse

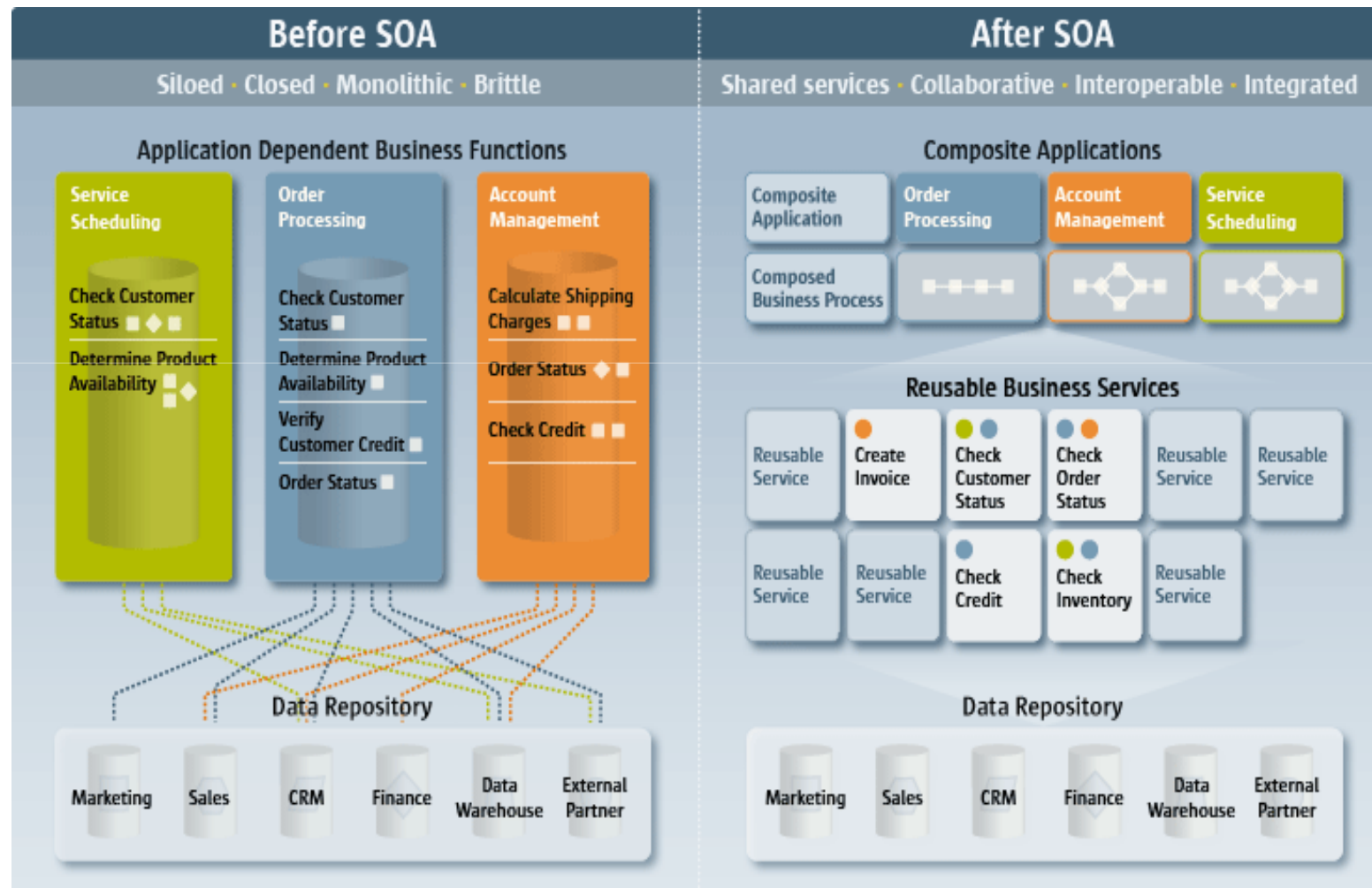
Implemented services must be designed with reuse in mind.
This creates some overhead.

▽ Sharing of Responsibilities

▽ Increased complexity!

Potential service users must be involved in the design process and will have influence on the service design

Before SOA – After SOA



source:IBM

Service Architecture Organized by Layers

Reasons for Layering

1. **Flexible composition.**
2. **Reuse.**
3. **Functional standardization** in lower levels
4. **Customization** in higher layers
5. **Separation of Concerns.**
6. **Policies may vary by Layer**

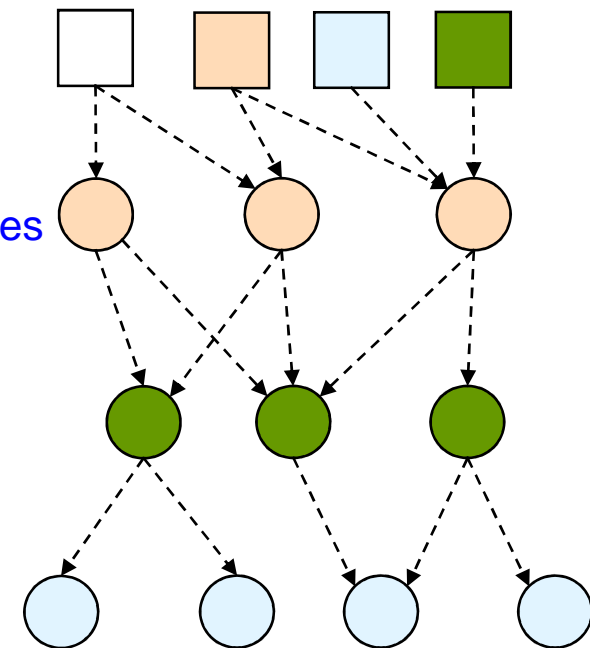
Example Layers

Presentation
& workflow

Composed Services

Basic Services

Underlying
API



*according to: TietoEnator AB,
Kurtis Bilder*

Different layers of SOA

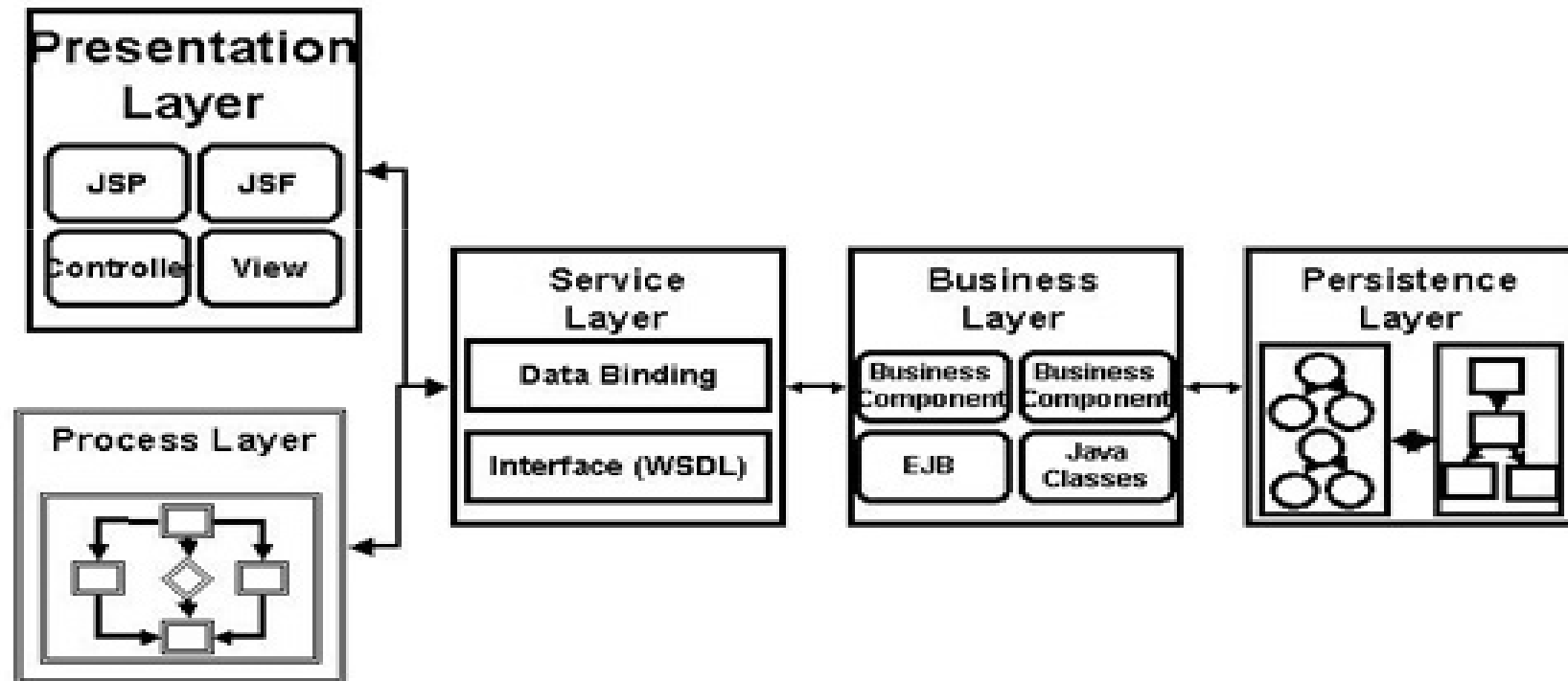
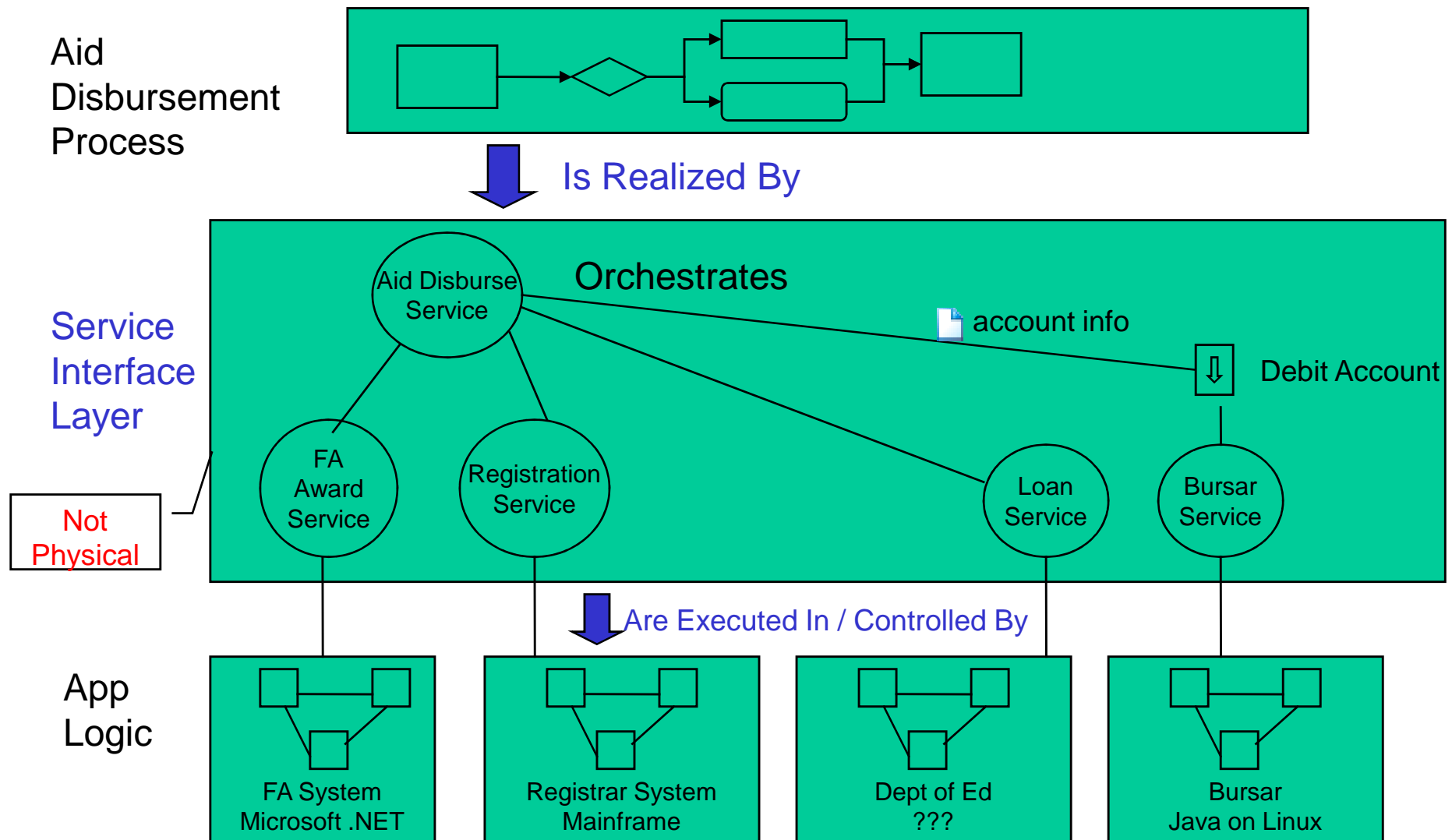


Fig 2. Different Layers of Service Oriented Applications

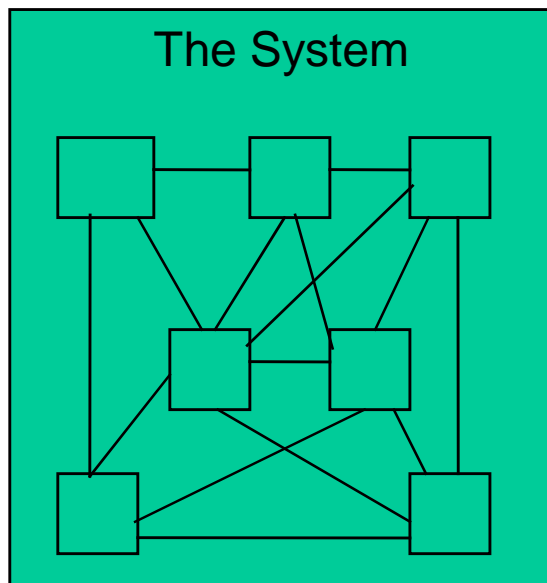
Service Composition Example



Applying services to the problem

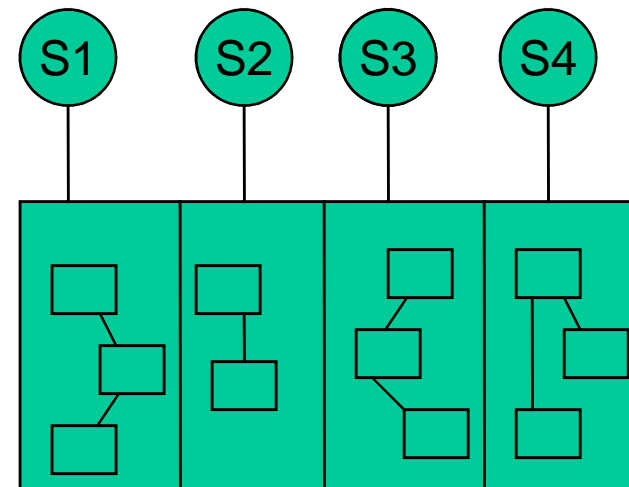
Monolithic

Before



System replacement is a total process
System modules are tightly interdependent
making change difficult

After



System composed of many logical service units (decomposition)

Underlying business logic decoupled as much as possible from other services (autonomy and loose coupling)

SOA Benefits

- ♣ Allow us to execute complex business processes by composing systems from small, less complex building blocks
- ♣ Fosters collaborative business and technical environment through sharing and coordination of services
- ♣ Create outward facing self-service applications not constrained by organizational boundaries
- ♣ Enables creating adaptive, agile systems that are resilient to changes in the business environment

Summary

- ♣ SOA represents a fundamental change to the way information systems will be designed in the future
- ♣ Long term impact on IT portfolio management is dramatic
- ♣ Adds a significant dimension to system evaluation process
- ♣ Undertaking SOA requires commitment from all levels of the organization and significant investments (people, process, and tools)

Summary

- ▽ If done correct, SOA is “not just another architectural ”
- ▽ SOA seeks to bridge the gap between business and technology promoting business agility (its all about managing change)
- ▽ SOA
 - Is complex
 - Requires governance
 - Requires executive management buy-in
 - Requires commitment with resources

References

- ♣ Coyle, “XML, Web Services and Data Revolution”, Pearson Education, 2002.
- ♣ Chatterjee and Webber, “Developing Enterprise Web Services – An Architect’s Guide”, Pearson Education, 2004.
- ♣ Liu, “Distributed Computing – Principles and Applications”, Pearson Education, 2004.
- ♣ <http://www.microsoft.comarchitecture/soa>
- ♣ <http://www.ibm.com/soa>
- ♣ <http://www.sun.com/products/soa>

Thank You.