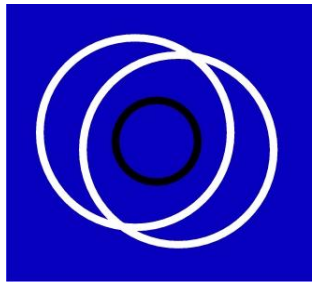
Module 1
Fundamental SOA & Service-Oriented Computing

SOASchool.com™
SOA CERTIFIED
Professional

Module 1 Overview

- Strategic Goals and Benefits
- Understanding SOA
- Understanding Service-Orientation
- Fundamental Terminology and Concepts
- Service Delivery Lifecycle
- Adoption Impacts and Requirements
- Introduction to Common Service Technologies

Strategic Goals and Benefits



The Seven Strategic Goals of Service-Oriented Computing

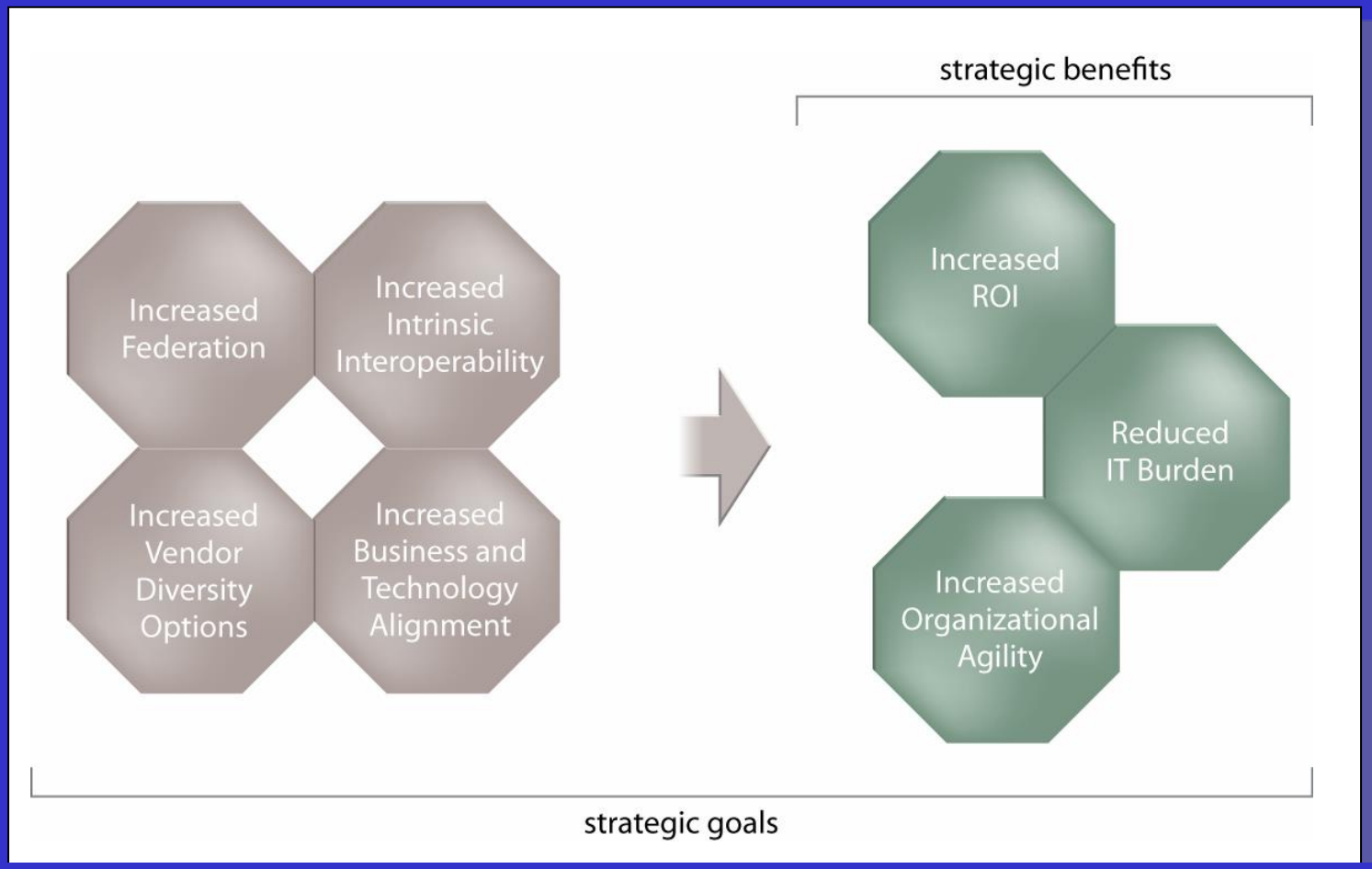
- There are seven strategic goals associated with the adoption of SOA and the overarching field of service-oriented computing.
- It is helpful to begin with an understanding of these goals because it helps provide context to subsequent sections during which SOA-related topics are explained.
- Specifically, it helps us understand why the distinct characteristics, qualities and preferences of SOA adoption exist.
- As you explore these topics throughout this and other courses, you will often be able to associate them with one or more of the upcoming goals.

The Seven Strategic Goals of Service-Oriented Computing

The strategic goals of service-oriented computing are:

- Increased Intrinsic Interoperability
- Increased Federation
- Increased Business and Technology Alignment
- Increased Vendor Diversification Options
- Increased ROI
- Increased Organizational Agility
- Reduced IT Burden

The Seven Strategic Goals of Service-Oriented Computing



The first four goals lead to the attainment of the latter three goals and benefits.

Strategic vs. Tactical

- All goals are strategic in nature, meaning they are for the long-term benefit of the IT enterprise.
- In comparison, tactical goals are focused on fulfilling immediate requirements in the short-term.
- The strategic nature of service-oriented computing is one of its distinguishing characteristics.
- It contrasts the more tactical nature of traditional silo-based application development.

Increased Intrinsic Interoperability

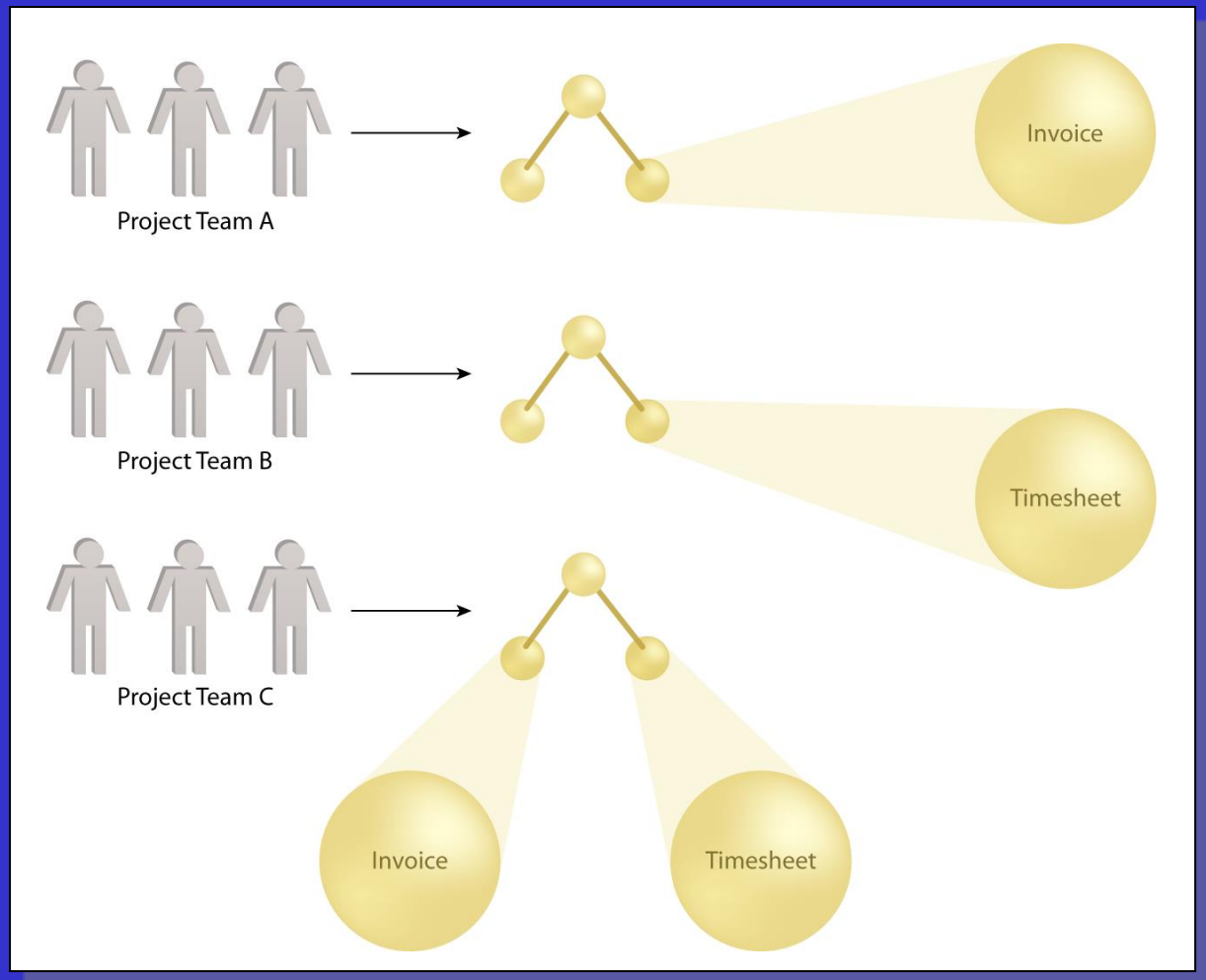
- Interoperability represents the ability of software programs to interact and exchange data.
- Integration represents the effort required to achieve interoperability between software programs.
- Integration effort is usually required when software programs are not compatible and therefore not interoperable.

Increased Intrinsic Interoperability

- A goal of service-orientation is to establish native or *intrinsic* interoperability between services in order to reduce the need for integration.
- This means that services are designed to be compatible and interoperable, regardless of when and by whom they are delivered.
- As a result, integration as a concept begins to fade when service-orientation is broadly applied because intrinsic interoperability between services becomes norm.

Increased Intrinsic Interoperability

Services delivered by Project Teams A and B are intrinsically interoperable and can therefore be reused by Project Team C without the need for a major integration project.



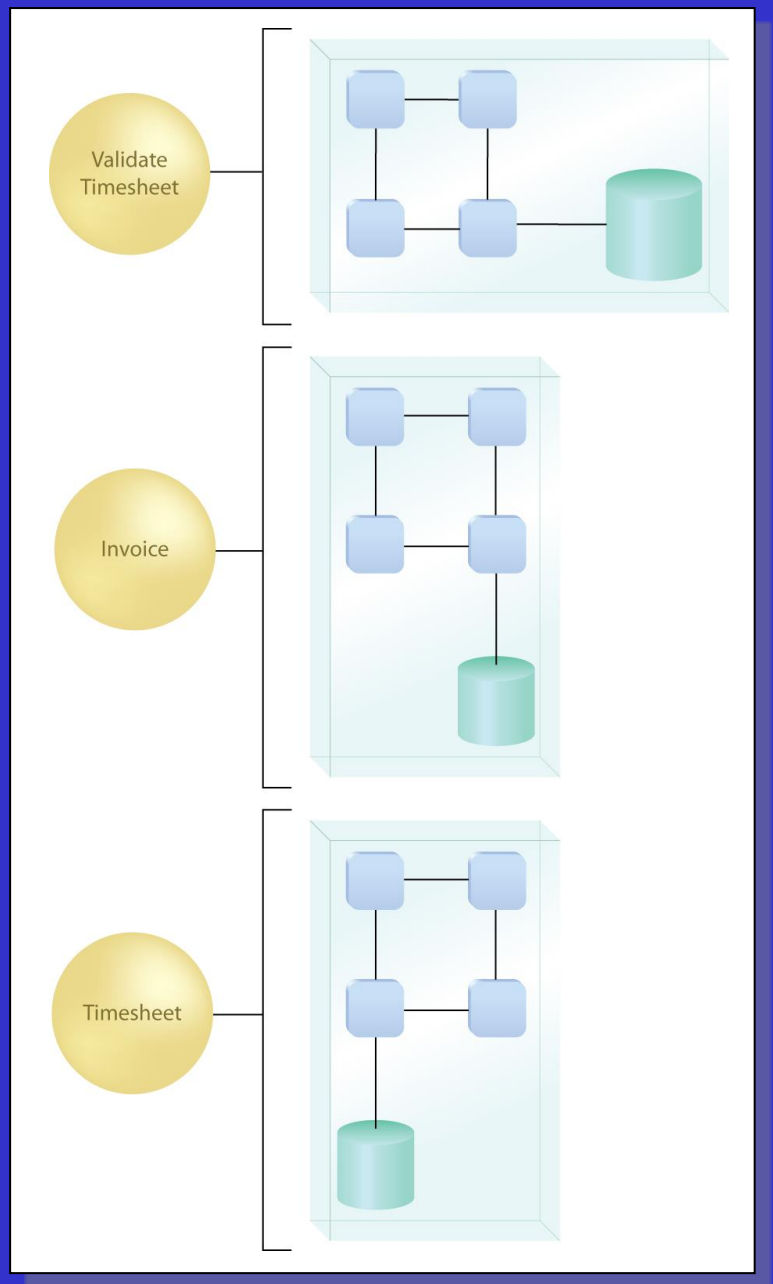
Increased Federation

- Federation is the unification of disparate environments while allowing those environments to be independently governed.
- SOA aims to establish a federated perspective of an enterprise through the wide spread deployment of standardized and composable services.
- Each service establishes a standardized technical interface or *endpoint* that represents a segment of the enterprise expressed in a consistent manner.

Increased Federation

Each of the services encapsulates a different system (right).

From an endpoint perspective (left), the service contracts express a standardized, federated service layer.

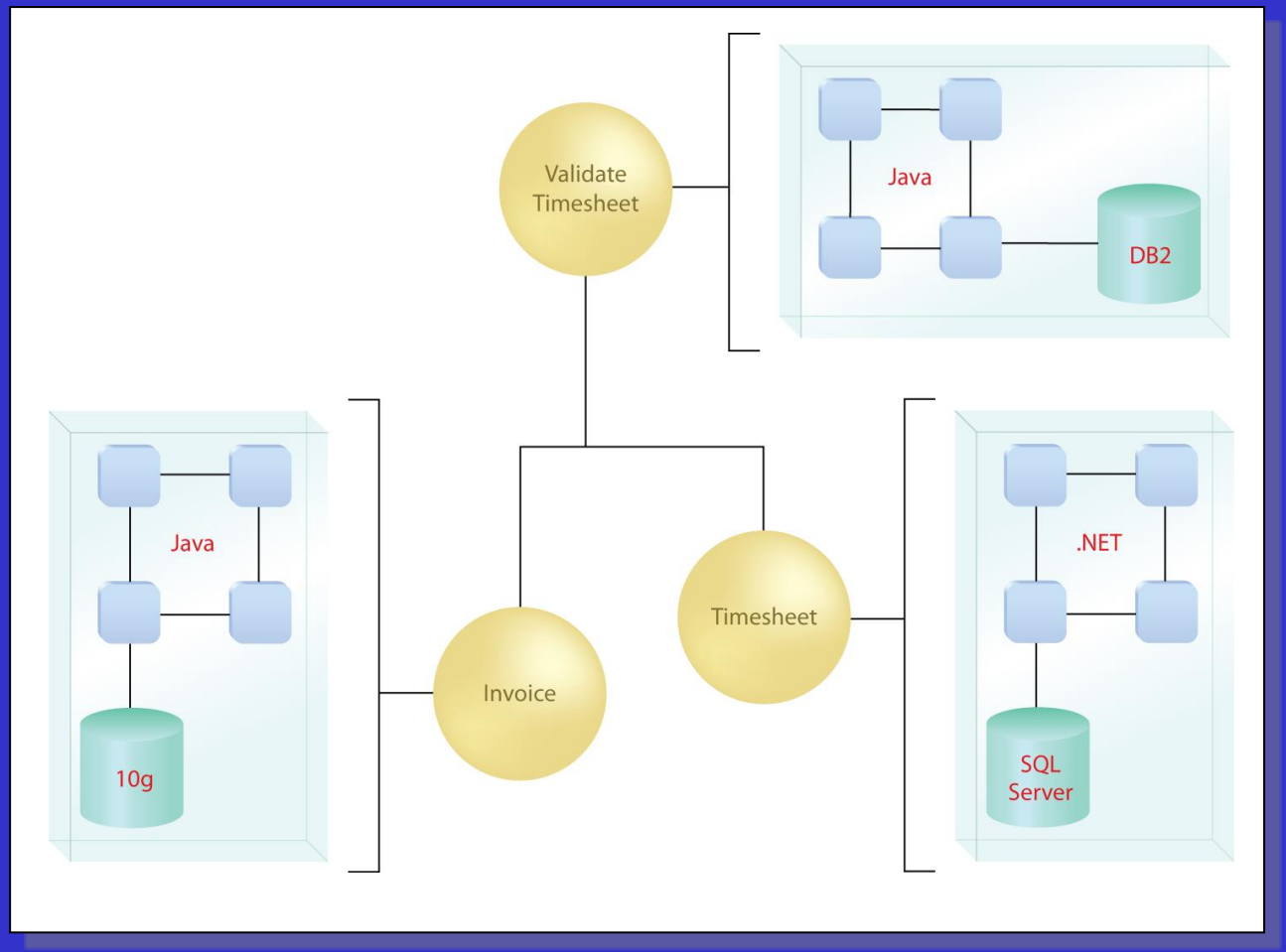


Increased Vendor Diversification Options

- Vendor diversification represents the option to refactor or extend parts of an enterprise with new vendor technologies and products.
- This can be achieved by designing SOA in alignment with but also neutral to vendor platforms and by positioning services as standardized endpoints so that proprietary implementation details can be abstracted.
- Vendor diversification is not always desirable but it is important to have the option to diversify when needed.

Increased Vendor Diversification Options

Even though the three services are built with different vendor technologies, they are able to interoperate.



Increased Business and Technology Alignment

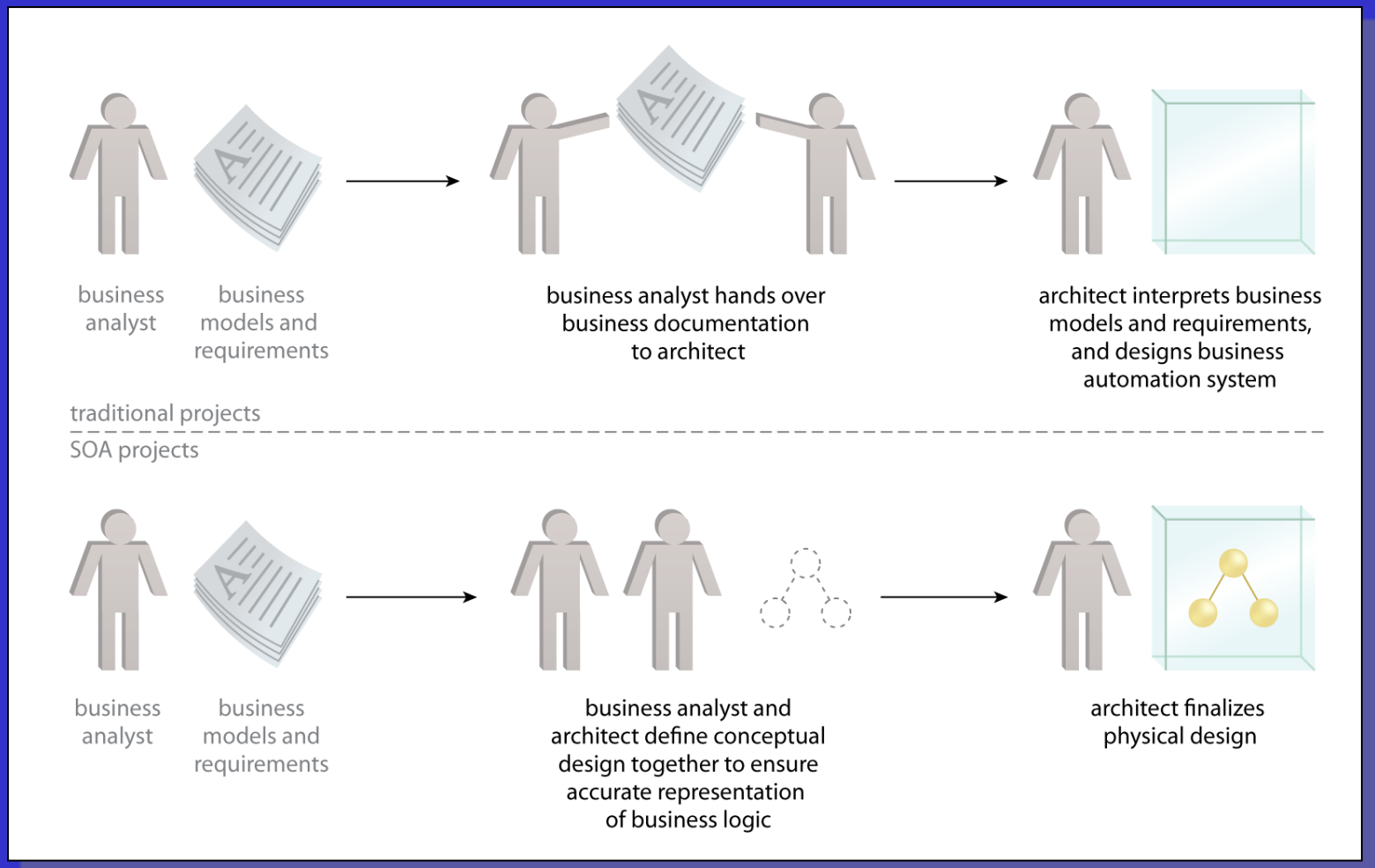
- "Business and technology alignment" represents the extent to which automated systems and the IT enterprise can mirror and evolve with the business.
- Increased business and technology alignment can be achieved via the collaboration of business and technology experts during analysis and modeling phases.
- Service-oriented computing introduces a design paradigm that promotes the abstraction and accurate encapsulation and expression of business logic.

Increased Business and Technology Alignment

- One of the most fundamental means of enabling business and technology alignment is through the definition and creation of business services.
- By following service-oriented analysis and design practices, business logic can be partitioned into flexible business services that can be repeatedly augmented and combined to continually respond to business change.
- Service-oriented analysis and design processes are introduced in the *SOA Project Delivery Cycle* section.
- This strategic goal is also known as *"Increased Business and Technology Domain Alignment"*

Increased Business and Technology Alignment

Business and technology communities are required to collaborate in a hands-on capacity.

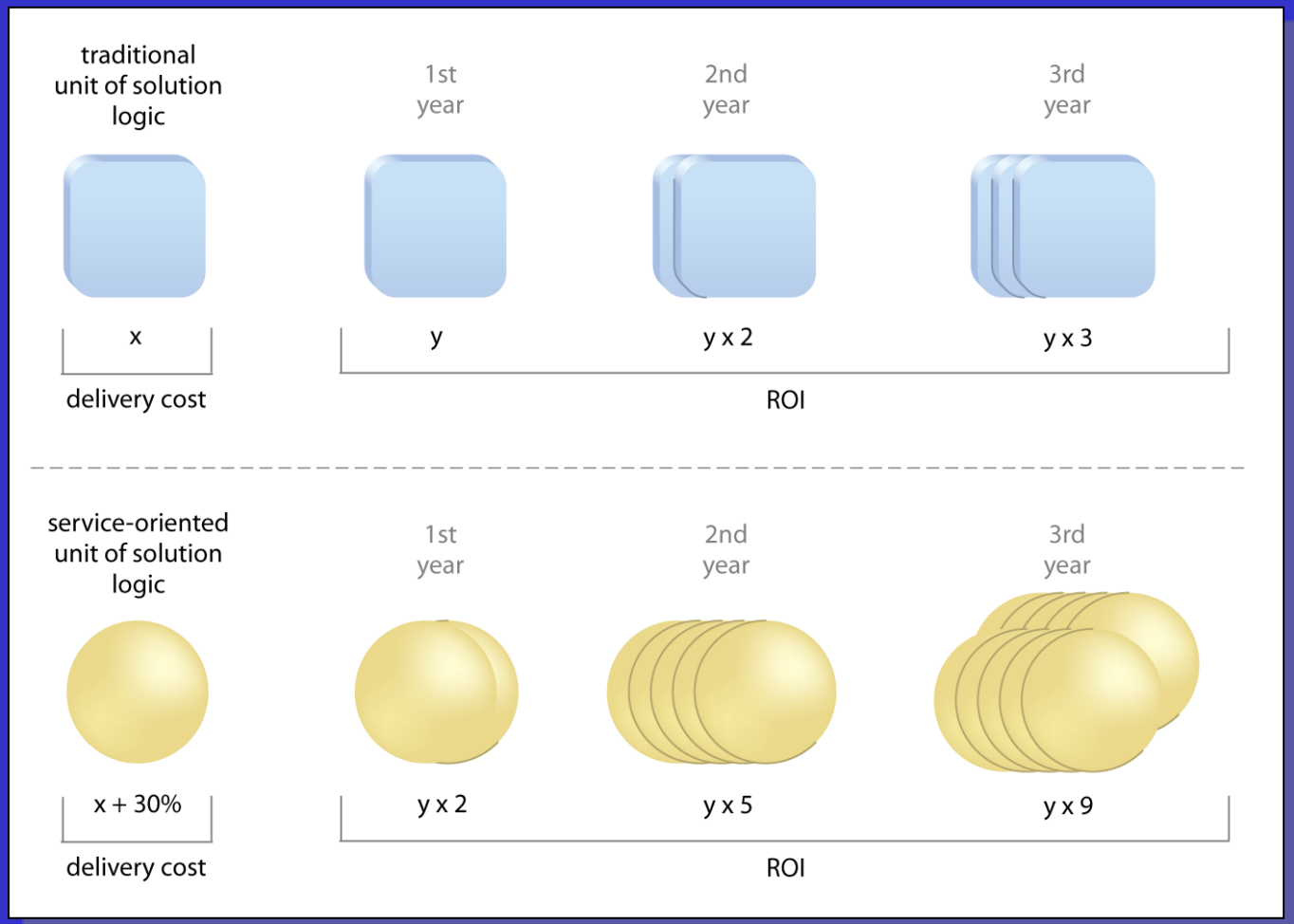


Increased ROI

- ROI (return on investment) represents the tangible value and cost-savings that something provides when compared to the cost of producing and governing it.
- Service-oriented computing fosters the creation of agnostic solution logic – logic that is agnostic to any one purpose and therefore multi-purpose.
- Design techniques that originated with commercial product design are applied to turn agnostic logic into highly reusable logic fully capable of leveraging intrinsic interoperability to provide increased ROI.

Increased ROI

An unprecedented emphasis is placed on getting the most out of every service, although it costs more to deliver each service.

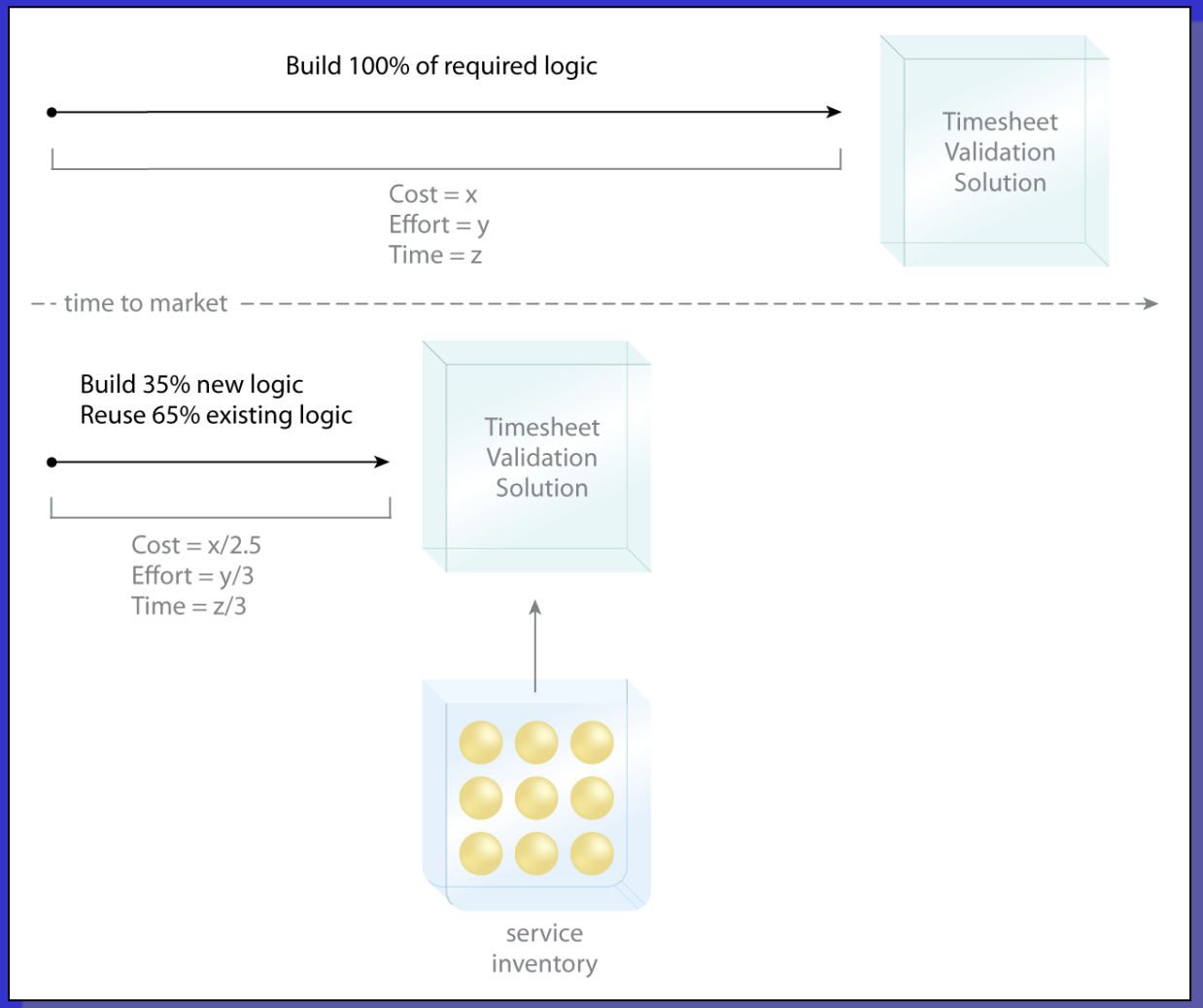


Increased Organizational Agility

- Agility is best compared to responsiveness.
- Something that is more responsive can react and adapt to change more efficiently and effectively.
- Agnostic services become reusable IT assets that can be repeatedly composed into different configurations.
- As a result, once a collection of mature agnostic services is available, the time and effort required to fulfill new or changed business requirements is dramatically reduced.

Increased Organizational Agility

Once a collection of federated, interoperable services is in place, the cost and effort required to build or modify solutions is significantly lowered.



Reduced IT Burden

Consistently applying service-orientation results in an IT enterprise with:

- reduced waste and redundancy
- reduced size and operational cost
- reduced overhead associated with its governance and evolution

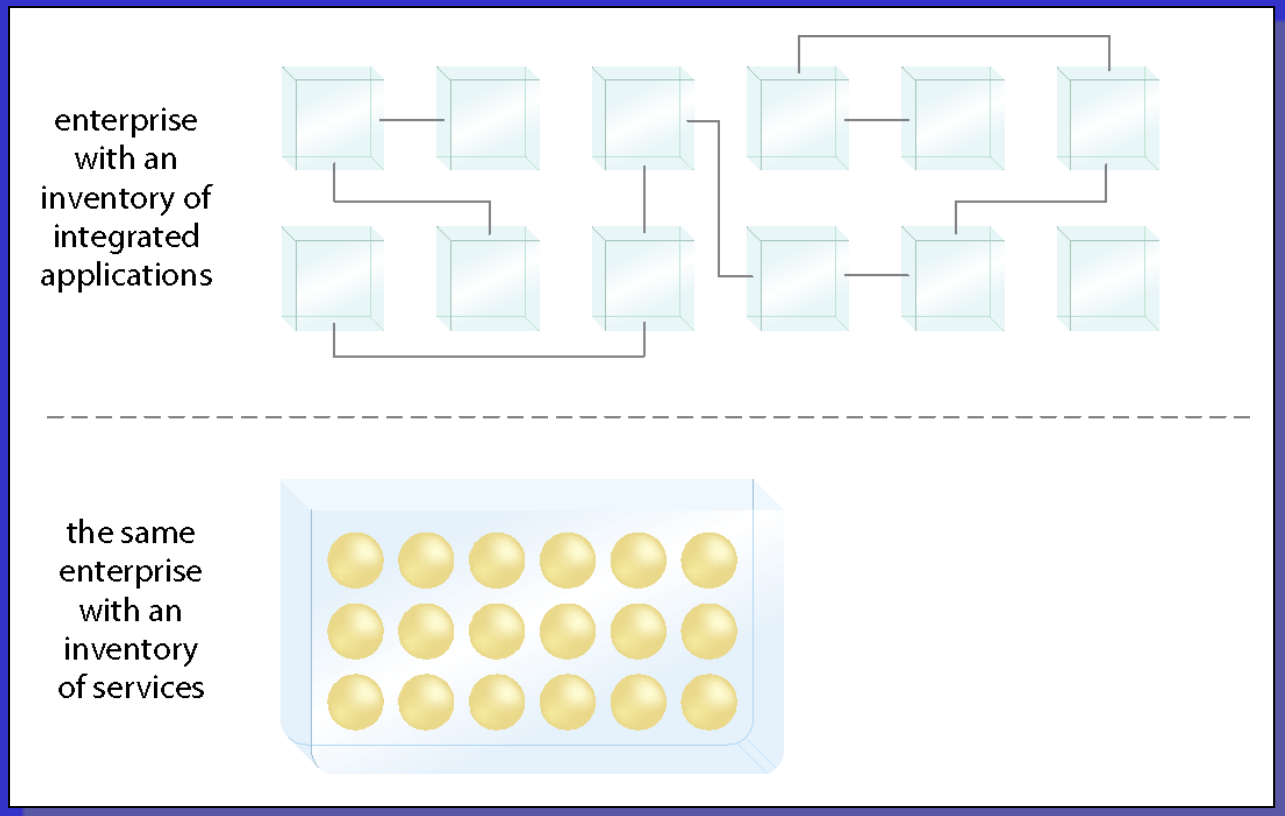
Such an enterprise can benefit an organization through dramatic increases in responsiveness and cost-effectiveness.

Reduced IT Burden

The attainment of the previously described goals can create a leaner, more agile IT department;

…one that is less of a burden on the organization…

…and more of an enabling contributor to its strategic goals.

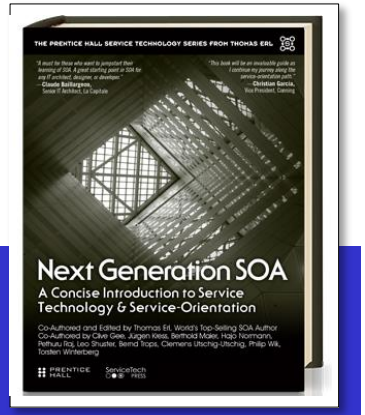


Exercise

Exercise 1.1

Fill in the Blanks

Reading



"Next Generation SOA: A Concise Introduction to Service Technology & Service-Orientation"

Pages 18 - 20

Chapter 2, *The Seven Goals of Applying Service-Orientation*

Understanding SOA

Service-Oriented Architecture

"Service-oriented architecture represents an architectural model that aims to enhance the agility and cost-effectiveness of an enterprise while reducing the overall burden of IT on an organization.

It accomplishes this by positioning services as the primary means through which solution logic is represented.

SOA supports service-orientation in the realization of the strategic goals associated with service-oriented computing."

ServiceOrientation.com

Service-Oriented Architecture

Short Definition:

SOA is a distributed technology architectural model with distinct characteristics in support of realizing service-orientation.

The fundamental characteristics of SOA are:

- business-driven
- vendor-neutral
- enterprise-centric
- composition-centric

These characteristics are often viewed as establishing a distinct context to a technology architecture. This context is ultimately associated with the overarching, aforementioned strategic goals and benefits.

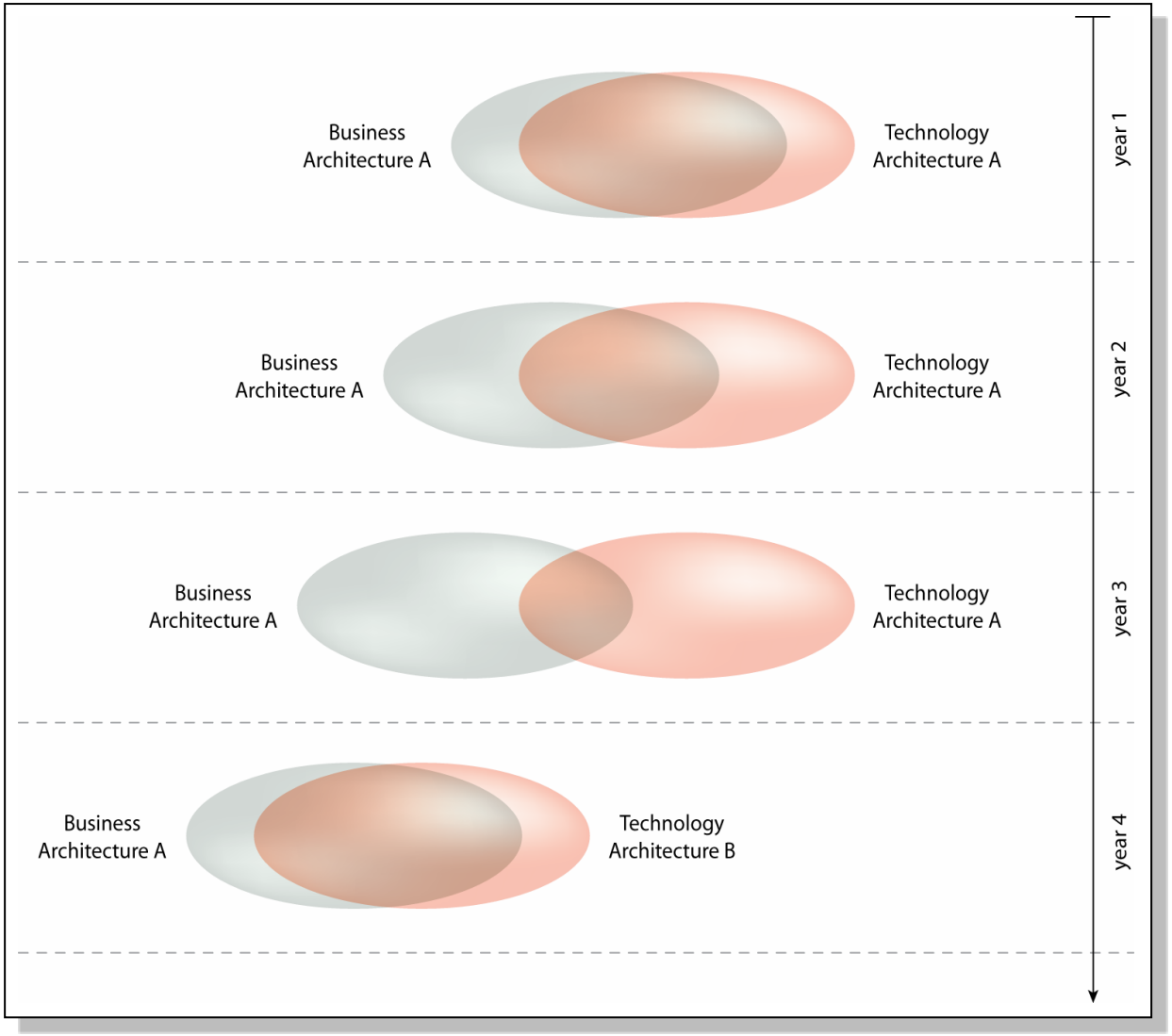
Business-Driven Characteristic

- Traditional technology architectures are often delivered in alignment with the current state of a business, but are incapable of keeping up with how the business evolves.
- As business and technology architectures become increasingly out of synch, business requirement fulfillment decreases.
- Often this occurs to the point that a whole new technology architecture is required.

Business-Driven Characteristic

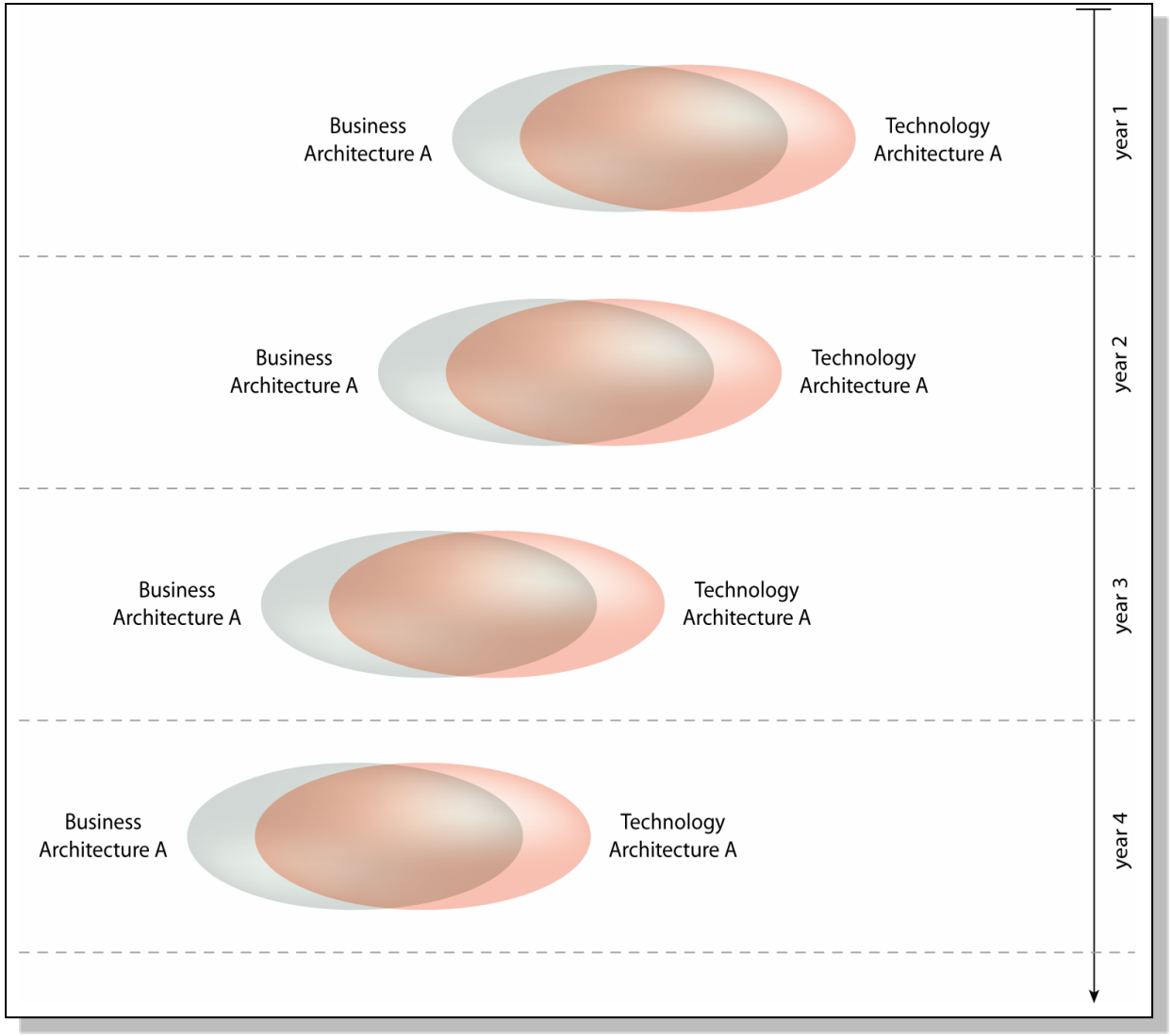
Over time, the scope and context of a technology architecture is outgrown by the business as it evolves in new directions.

This results in the need to eventually replace the architecture.



Business-Driven Characteristic

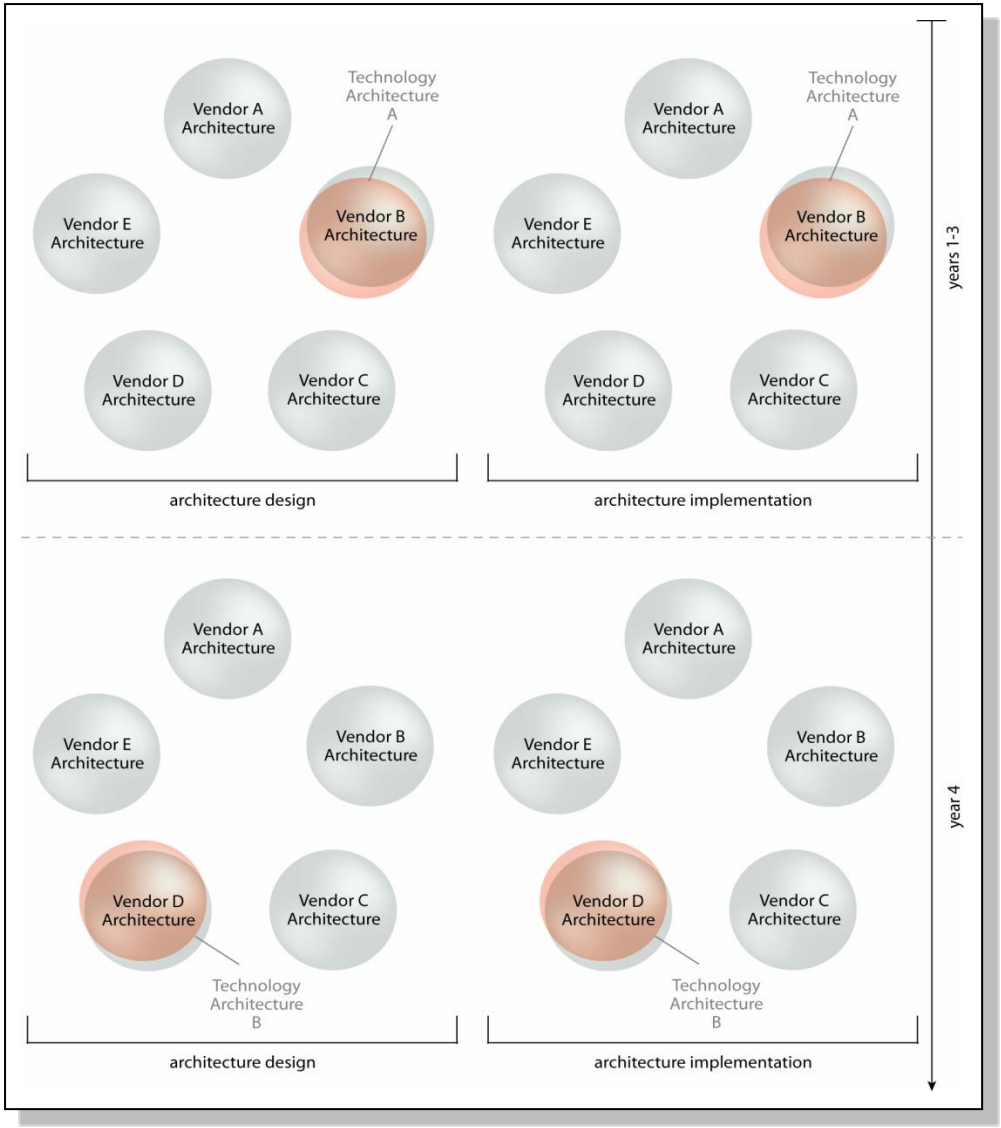
By applying a business-driven strategic scope to the technology architecture, it can be kept in constant synch with how the business evolves over time.



Vendor-Neutral Characteristic

Vendor-centric technology architectures are often bound to corresponding vendor platform roadmaps.

This can reduce opportunities to leverage technology innovations provided by other vendor platforms and can result in the need to eventually replace the implementation entirely (which starts the cycle over again).



Vendor-Neutral Characteristic

If the architectural model is designed to be neutral to vendor platforms, it maintains the freedom to diversify its implementation by leveraging multiple vendor technology innovations.

This increases the longevity of the architecture as it is allowed to augment and evolve in response to changing requirements.

Enterprise-Centric Characteristic

- A service-oriented technology architecture is enterprise-centric in that it fosters the creation and reuse of logic across different business tasks and processes.
- The enterprise-centric characteristic contrasts the tasks or process-specific nature of silo-based applications.
- Enterprise-centricity imposes specific architectural requirements upon service-oriented solutions.

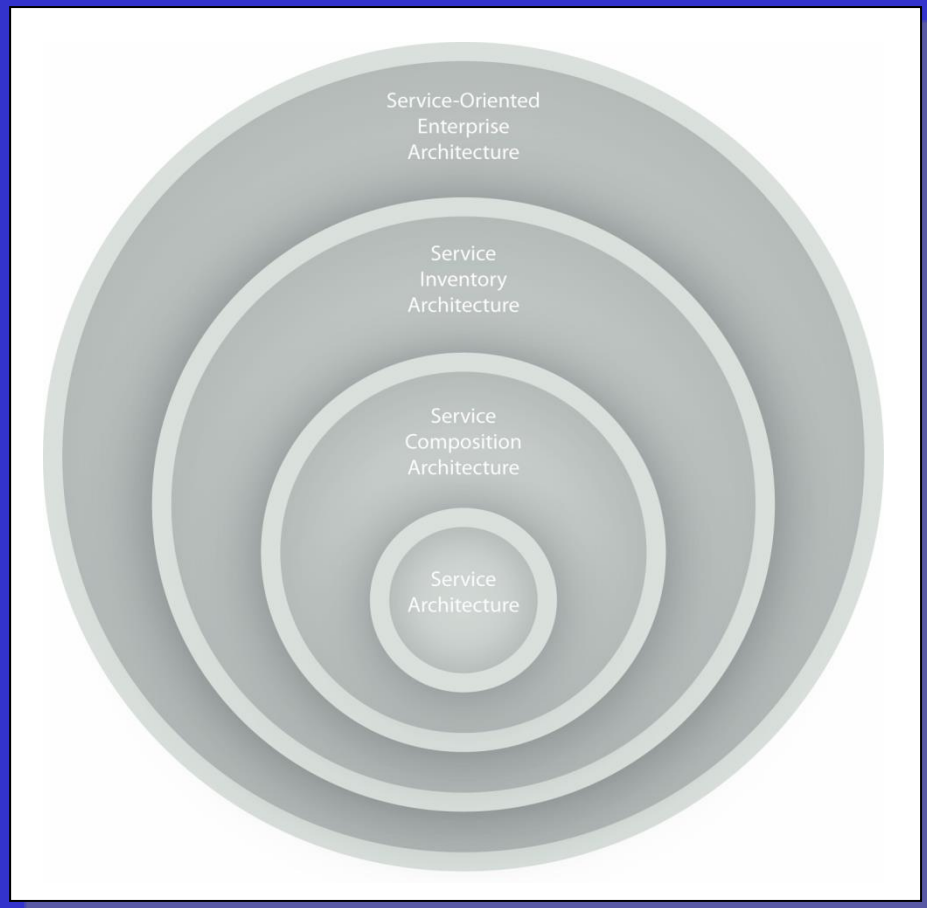
Composition-Centric Characteristic

- Service-oriented solutions are comprised of services that often need to contain reusable logic.
- However, in order for such services to be truly reusable, they must be able to participate in different distributed solution architectures or service compositions.
- A service-oriented technology architecture is must be composition-centric in support of reusable services being pulled into a variety of different service composition designs.

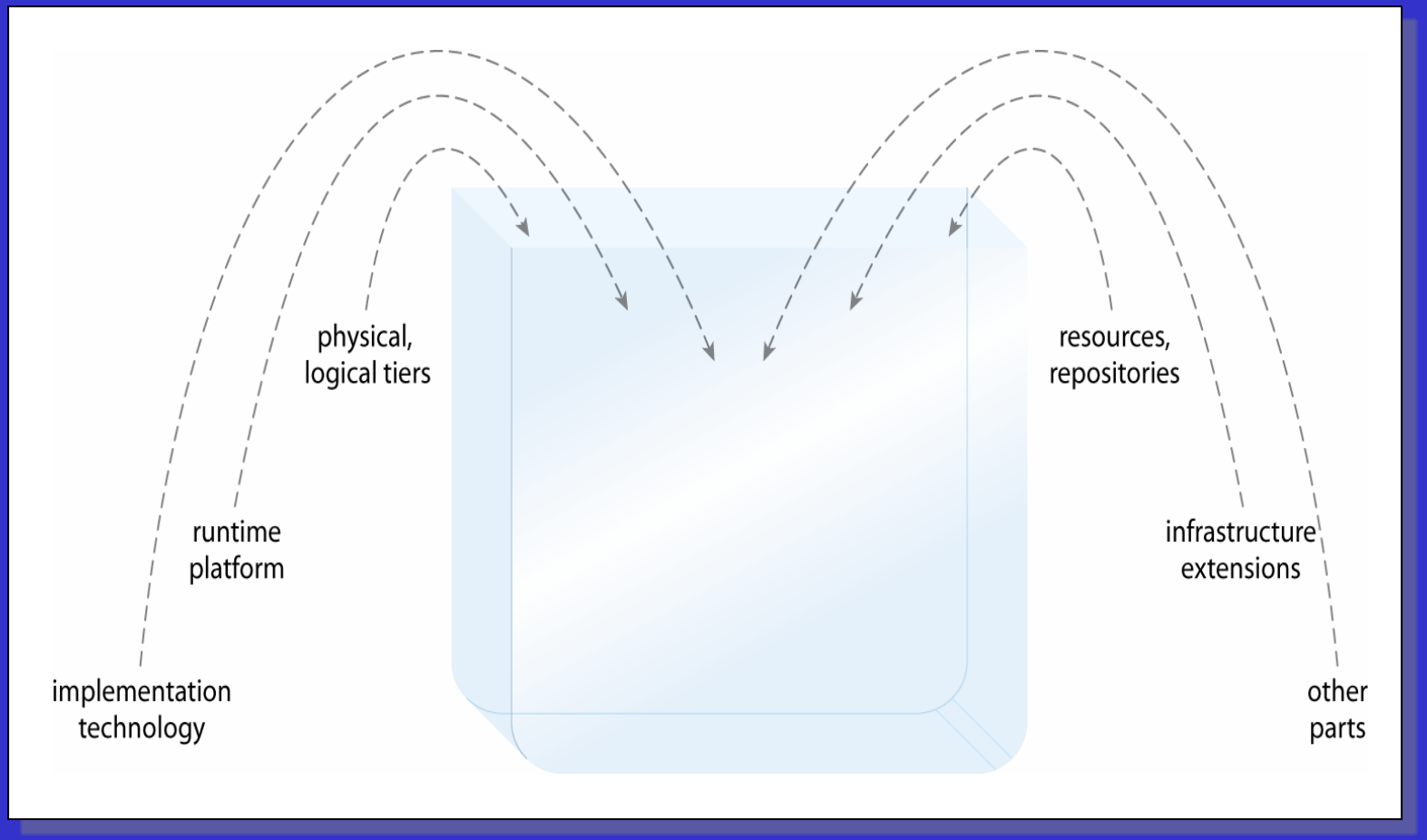
Types of Service-Oriented Architecture

There are four primary types of service-oriented architecture, each associated with a specific scope:

- Service Architecture (inner circle)
- Service Composition Architecture
- Service Inventory Architecture
- Service-Oriented Enterprise Architecture (outer circle)

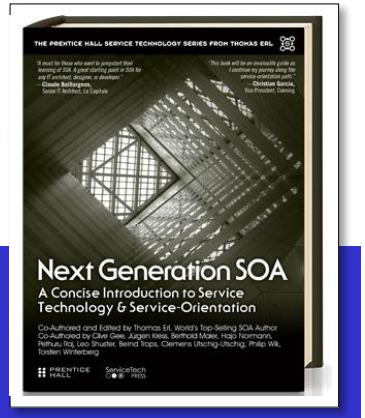


Service-Oriented Architecture



Example considerations for a service inventory architecture.

Reading



"Next Generation SOA: A Concise Introduction to Service Technology & Service-Orientation"

Pages 13 - 17

Chapter 2: *The Four Characteristics of SOA*, *The Four Common Types of SOA*

Understanding Service-Orientation

Service-Orientation

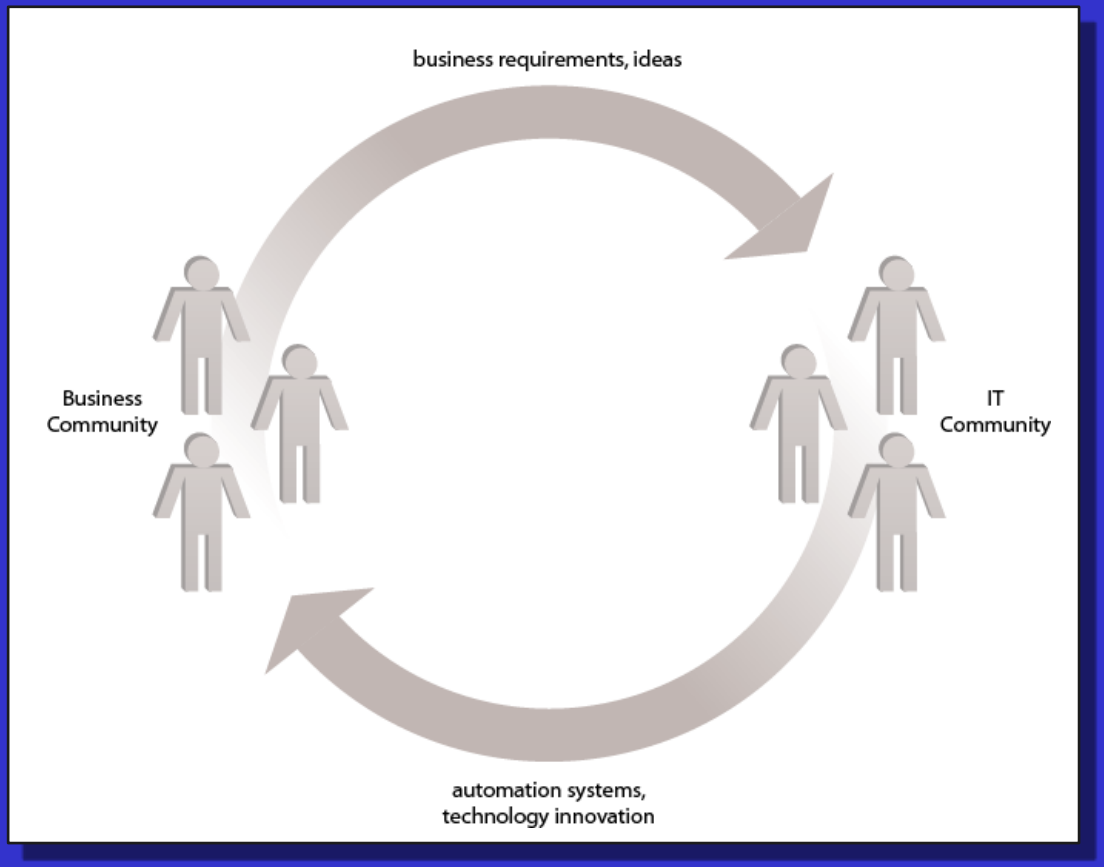
"Service-orientation is a design paradigm intended for the creation of solution logic units that are individually shaped so that they can be collectively and repeatedly utilized in support of the realization of a specific set of strategic goals and benefits associated with SOA and service-oriented computing.

Solution logic designed in accordance with service-orientation can be qualified with 'service-oriented,' and units of service-oriented solution logic are referred to as services."

ServiceOrientation.com

Service-Orientation

The essence of service-orientation is to establish an IT environment inherently capable of accommodating change.



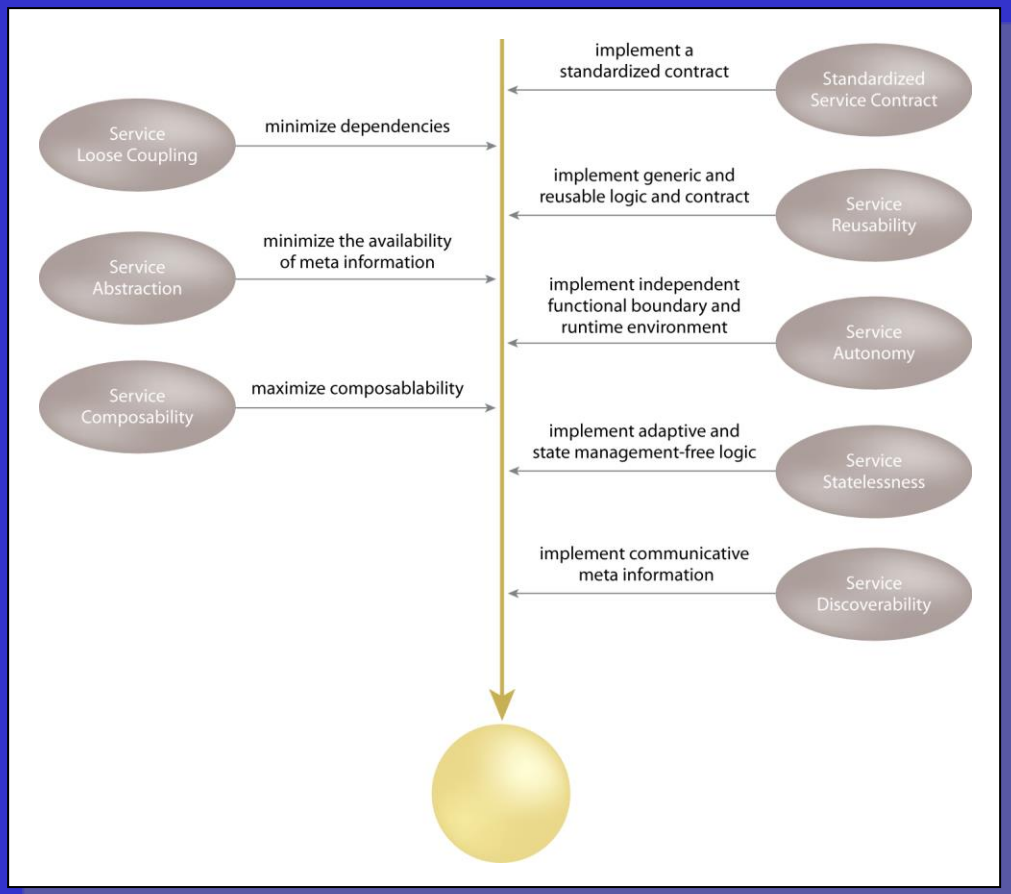
Service-Orientation

- Service-Orientation is the design approach (or design paradigm) used to build service-oriented solutions.
- The application of service-orientation results in the creation of specific design characteristics, all of which foster the attainment of the strategic goals associated with service-oriented computing.
- Service-orientation is comprised of a series of design principles that collectively establish these design characteristics.
- An understanding of service-orientation is required to build service-oriented solutions.

Service-Orientation

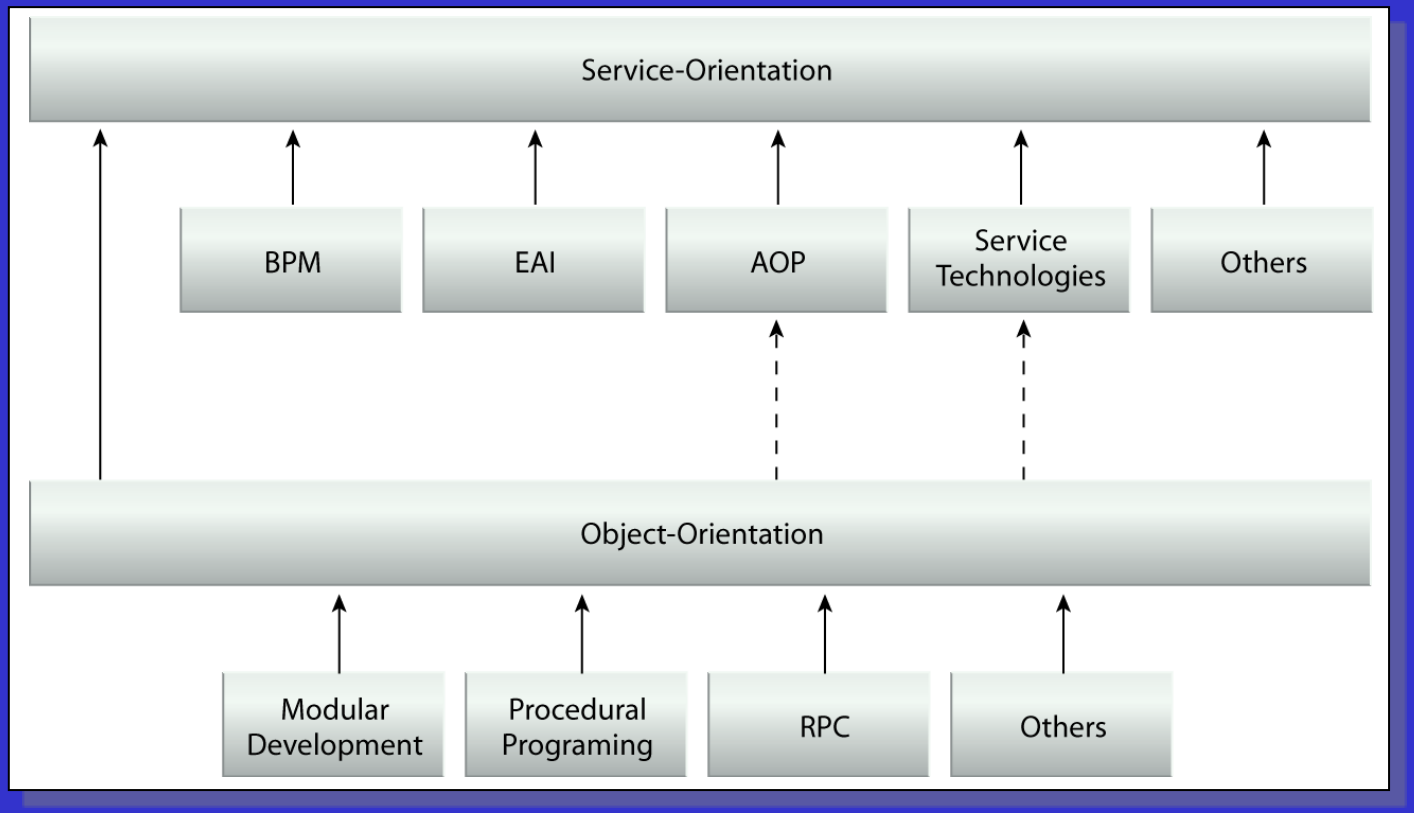
The eight service-orientation design principles:

- Standardized Service Contract
- Service Loose Coupling
- Service Abstraction
- Service Reusability
- Service Autonomy
- Service Statelessness
- Service Discoverability
- Service Composability



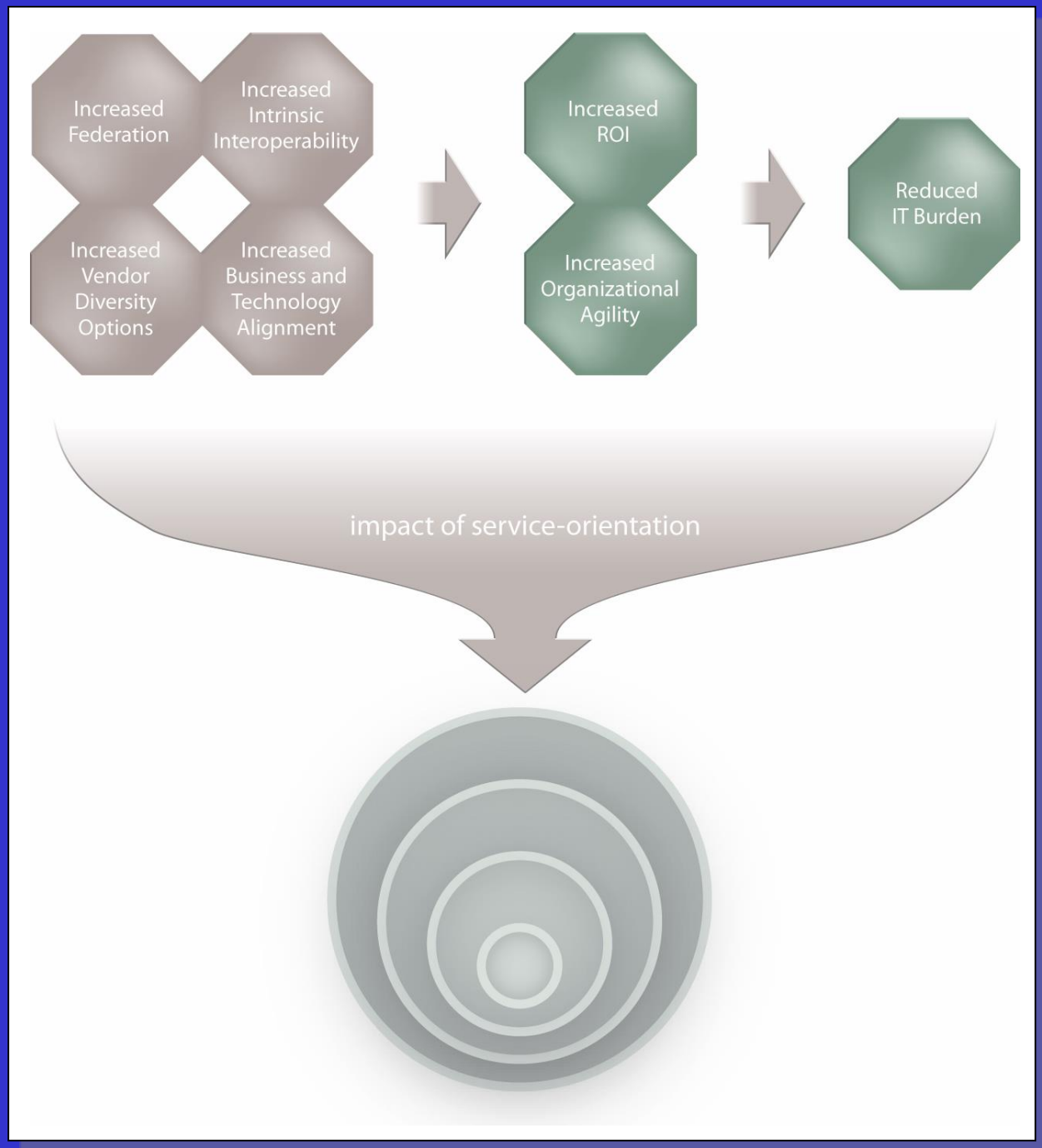
Service-Orientation

The service-orientation design paradigm has been influenced by several established design platforms and technology innovations.

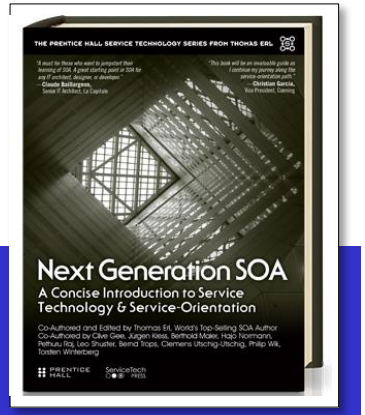


Applying service-orientation is fundamental to achieving the strategic goals and benefits associated with service-oriented computing.

The application of service-orientation impacts all four service-oriented architecture types.



Reading

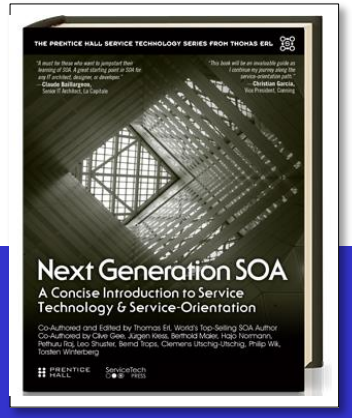


"Next Generation SOA: A Concise Introduction to Service Technology & Service-Orientation"

Pages 8 - 13

Chapter 2: *Services and Service-Orientation, Service-Orientation, Yesterday and Today, The Eight Principles of Service-Orientation*

Optional Reading



Further documents that can help clarify the relationship between service-orientation and SOA are the SOA Manifesto and the Annotated SOA Manifesto. The SOA Manifesto was authored by 17 industry experts from a wide range of industry and academic organizations. Its purpose is to provide a concise declaration of SOA and service-orientation that clearly defines its values and priorities for a broad audience.

The SOA Manifesto and Annotated SOA Manifesto are covered in Chapter 4 of the Next Generation SOA textbook on pages 47-62 and can also be viewed online at www.soa-manifesto.org and www.soa-manifesto.com.

The SOA Manifesto is not covered on Exam S90.01 and the reading of this content is optional.

References

Module 3: SOA Architecture & Design covers the SOA technology architecture model and the service-orientation design paradigm in more detail.

Other SOACP course modules further explore SOA and service-orientation from different perspectives and in relation to different project lifecycle stages.

Fundamental Terminology and Concepts

Service-Oriented Linguistics

"Service-Oriented Architecture"

"Service-Orientation"

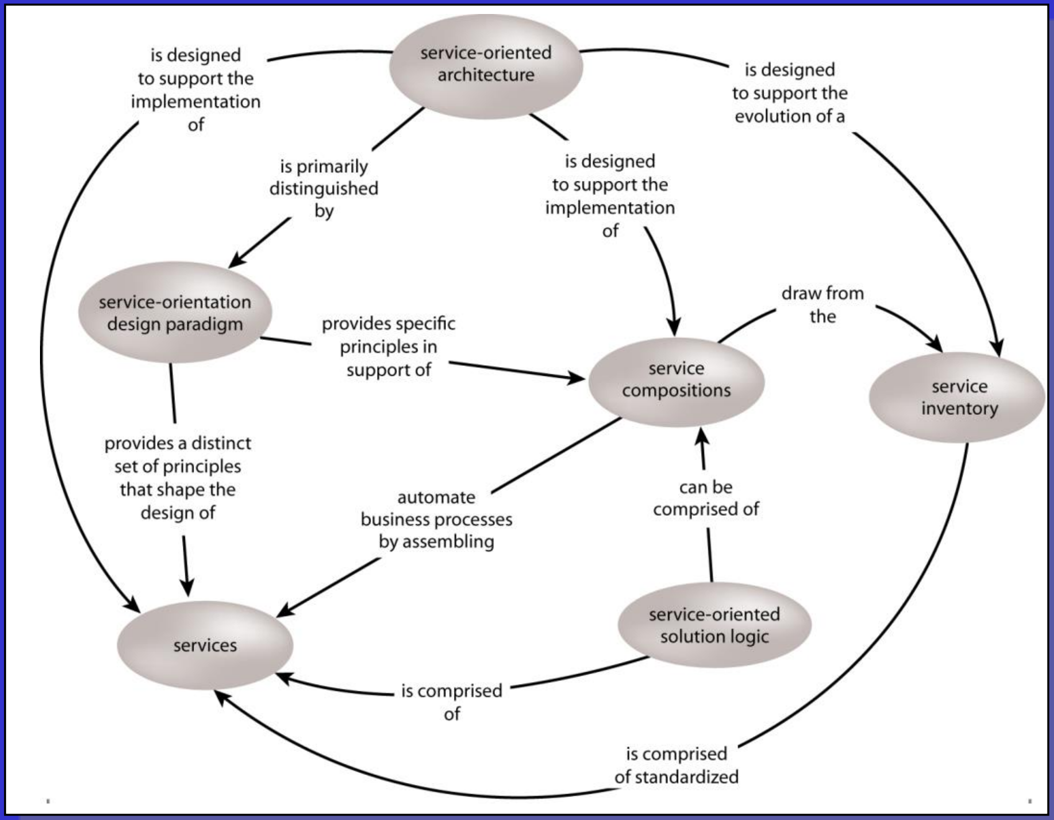
"Service"

"Service Composition"

"Service-Oriented Solution Logic"

"Service Inventory"

"Service-Oriented Computing Platform"



Services

"A service is the most fundamental unit of service-oriented logic.

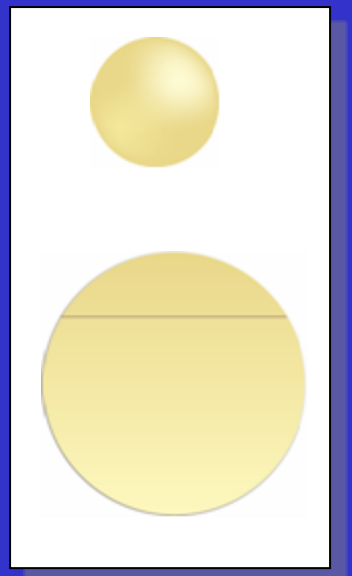
A unit of solution logic is classified as a service when service-orientation has been applied to a meaningful extent.

Service-oriented solutions are typically comprised of multiple services that form a service composition."

ServiceOrientation.com

Services

- Services are the basic building blocks of the service-oriented computing platform.
- Services are designed in a very specific manner, as per the service-orientation design approach.
- Services are delivered via a specific lifecycle that is explained in the *Service Delivery Lifecycle* section.
- Due to the enterprise-centric nature of service-orientation and SOA, services can be classified as enterprise resources.



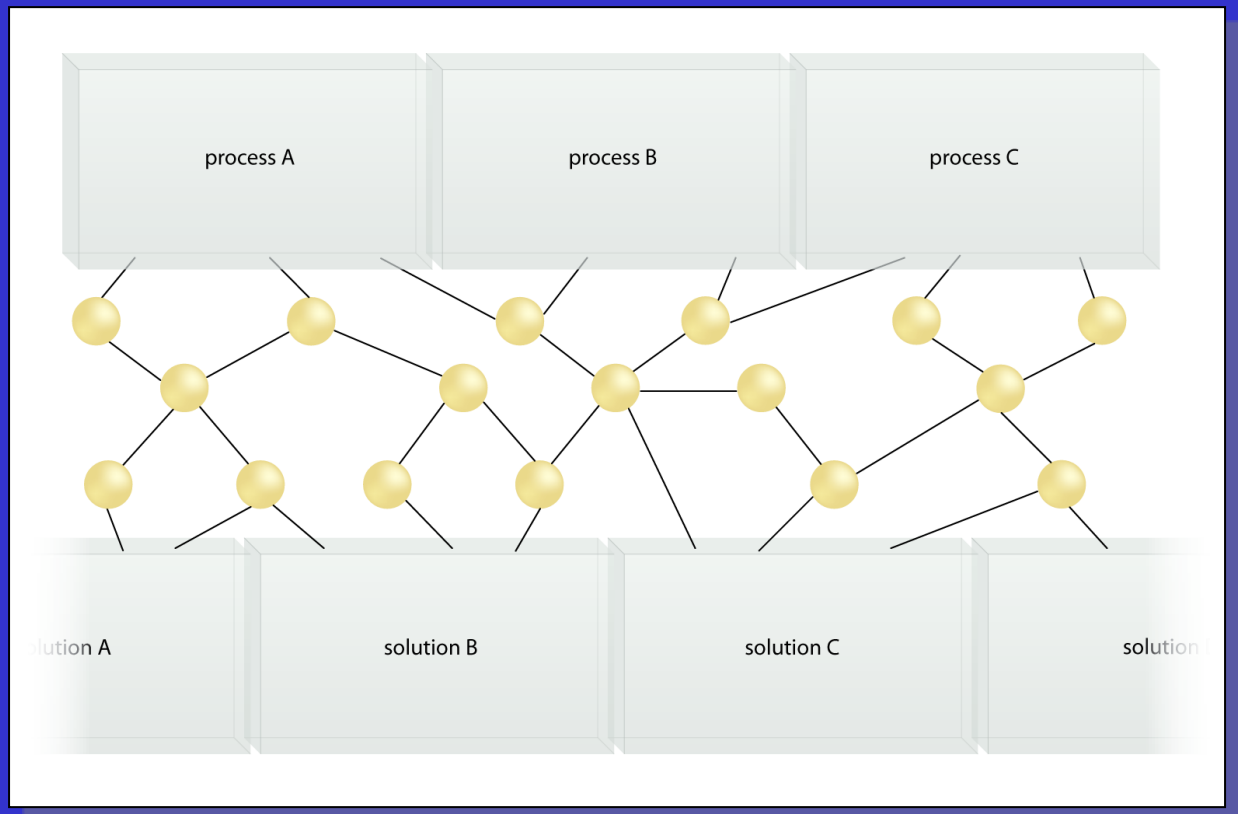
Services

Common service design issues include:

- granularity
- contract design
- agnostic vs. non-agnostic design
- legacy encapsulation
- security
- governance

Agnostic Services

A primary goal of service-orientation is to produce a majority of services with an agnostic functional context.



Service Contracts

- A service contract expresses meta information about a service and sets the terms of engagement (the requirements for invoking and interacting with the service).
- For a service consumer to access and interact with a service, it must comply with the requirements of the service contract.
- The fundamental part of a service contract is its technical interface which forms the technical service contract.

Service Contracts

- A service contract can be further comprised of human-readable documents, such as a Service Level Agreement (SLA) that describes additional quality-of-service features, behaviors, and limitations.
- How service contracts are designed and how they physically exist depends on what implementation technology is used to build the service.
- Service implementation technology options are introduced toward the end of this module.

Service Contracts

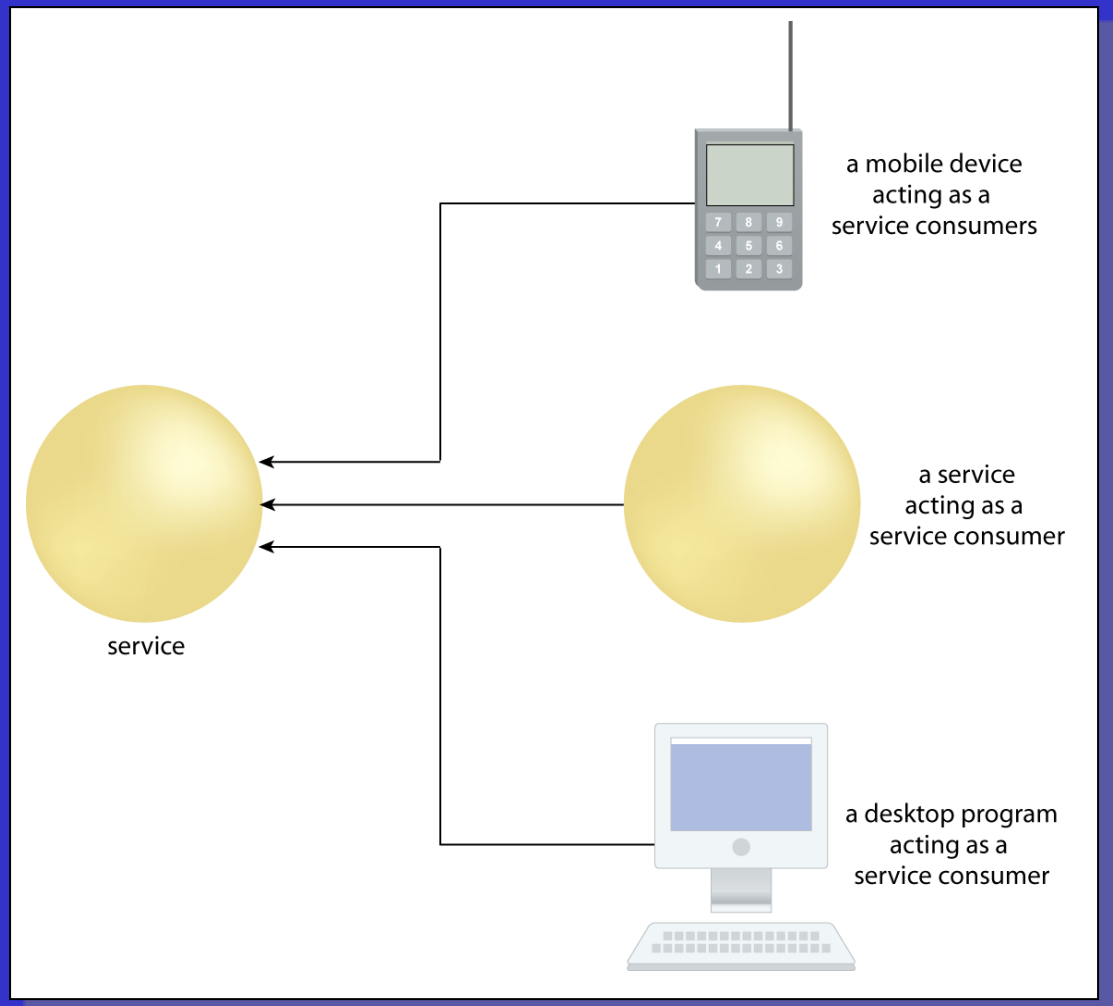
- Within service-orientation, the design and standardization of the service contract is of paramount importance - so much so, that the Standardized Service Contract design principle is dedicated to the standardized definition of service contracts.
- Creating standardized service contracts is a key requirement to achieving the strategic goals of increased federation and intrinsic interoperability.

Service Consumers

- When a program invokes and interacts with a service it is labeled as a service consumer.
- It is important to understand that this term refers to the temporary runtime role assumed by a program at the time it is engaging a service in a data exchange.
- A service consumer may or may not be another service.
- The ability for one service to invoke (consume) another service forms the basis of service composition (as explained shortly).

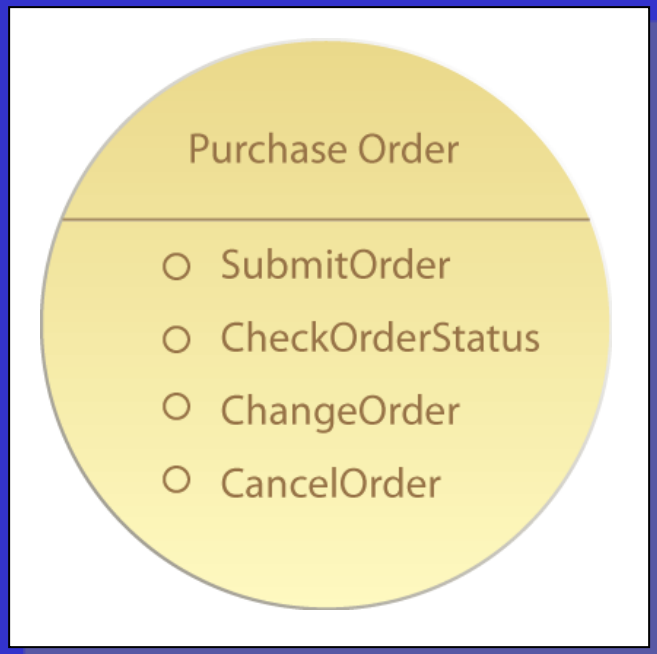
Service Consumers

Any software program capable of invoking and interacting with a service can be considered a service consumer.



Service Capabilities

- Each service is assigned its own distinct functional context and is comprised of a set of functions related to this context.
- These functions are referred to as service capabilities until it is known how a service will be built.



- For example, a service capability may be referred to as a service operation when a service is built as a Web service.

Service Capabilities

- A service can have one or more capabilities.
- Service capabilities can be expressed in a service contract.
- A service consumer will often invoke a specific capability, which means that it invokes only a subset of the functionality a service has to offer.
- The term "service capability" is commonly used when services are first modeled.
- The functionality associated with a given capability can range in scope.

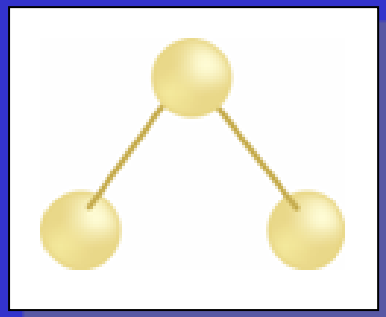
Service Compositions

"A service composition is an aggregate of services collectively composed to automate a particular task or business process.

To qualify as a composition, at least two participating services plus one composition initiator need to be present. Otherwise, the service interaction only represents a point-to-point exchange."

ServiceOrientation.com

Service Compositions

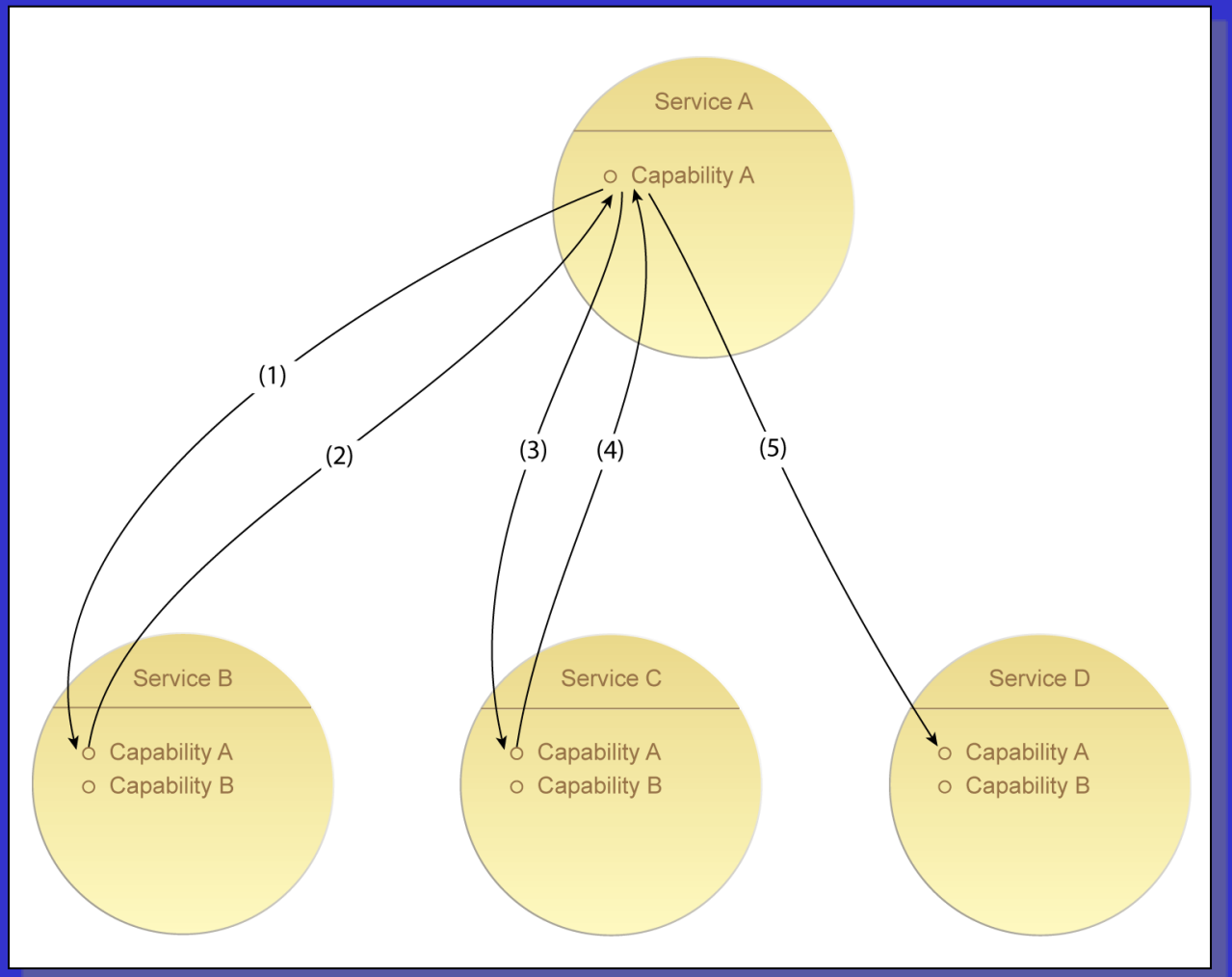


- Service composition is very important to the success of an SOA initiative because the strategic benefits of increased ROI and organizational agility rely on the ability to compose and recompose services together.
- To achieve these goals requires that services be designed with the inherent ability to be effective composition members. (Much of service-orientation is geared toward achieving this design goal.)

Service Compositions

Service composition relies on the ability of services to consume each other.

In this diagram, Service A is acting as a consumer of Services B, C, and D.



Service Compositions

Common service composition design issues include:

- runtime activity state management
- cross-service transactions
- reliable data exchange
- security requirements
- transformation avoidance
- performance demands

Service-Oriented Solution Logic

"Any body of solution logic to which service-orientation has been applied to a meaningful extent is considered 'service-oriented.'

A service represents the most fundamental unit of service-oriented solution logic."

ServiceOrientation.com

Service-Oriented Solution Logic

- There are two basic types of service-oriented solution logic: services and service compositions
- A service-oriented solution can encompass one or more service compositions because it represents a body of logic capable of carrying out one or more related tasks or business processes.
- On the other hand, the simplest of service-oriented solutions may consist of just one service.

Service Inventory

"A service inventory is an independently standardized and governed collection of complementary services within a boundary that represents an enterprise or a meaningful segment of an enterprise.

When an organization has multiple service inventories, this term is further qualified as domain service inventory."

ServiceOrientation.com

Service Inventory

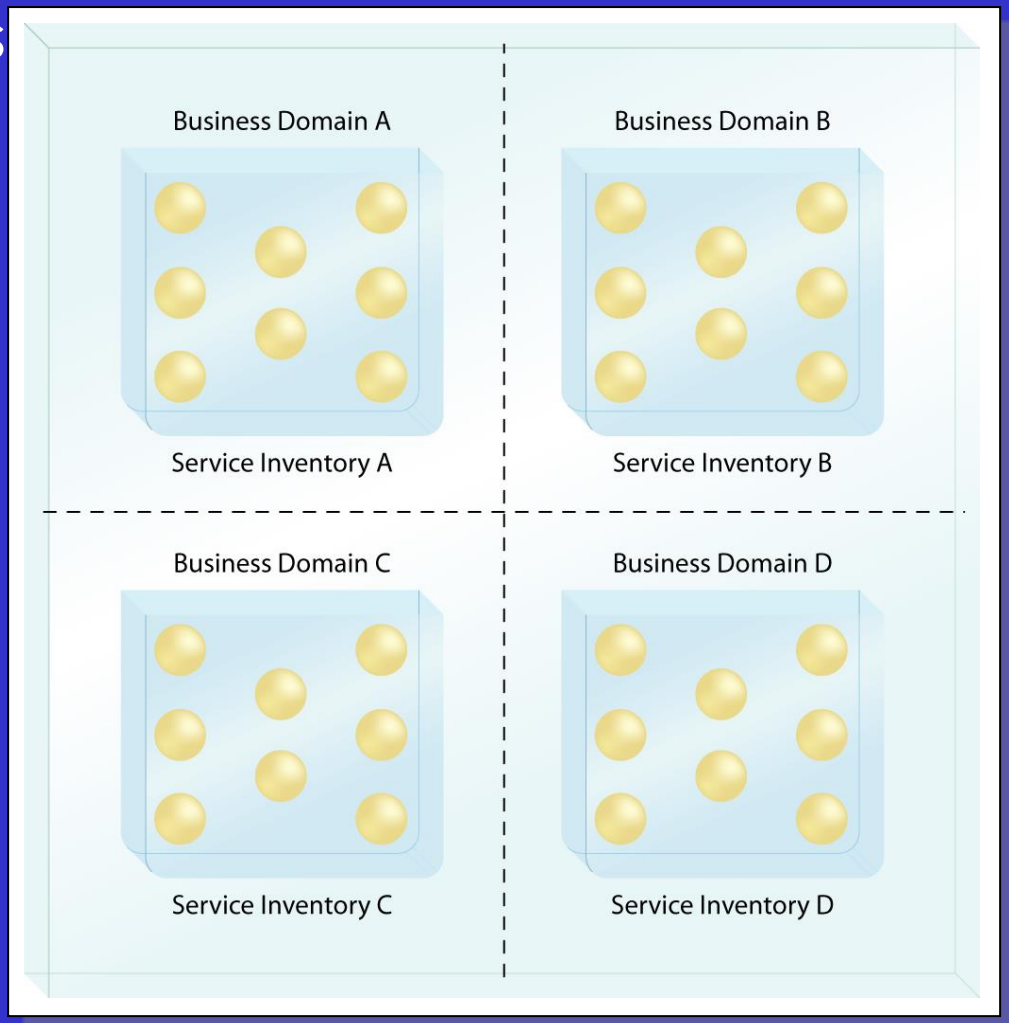
- Ideally, all services are delivered as part of the same enterprise service inventory.
- In larger environments, enterprise-wide standardization may be unattainable. In this case, domain service inventories may need to be formed.
- Service inventory blueprints are typically defined in advance as part of the up-front analysis phase.
- Part of this analysis phase is focused on avoiding overlap among inventory services in an effort to achieve service normalization.

Domain Service Inventories

Multiple service inventories can be created for one enterprise.

The scope of each represents a well-defined enterprise domain often associated with a business domain.

Within each domain, service inventories are standardized and governed independently.

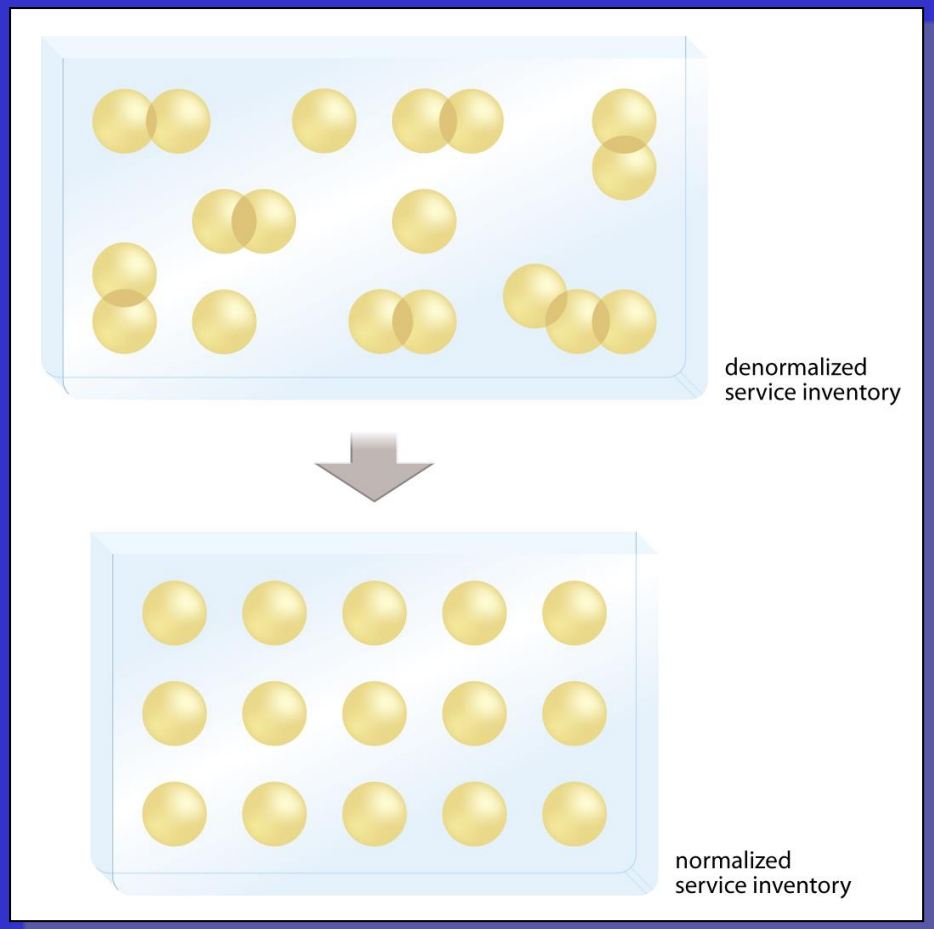


Normalized Service Inventories

A service inventory containing services with overlapping functional boundaries introduces denormalization.

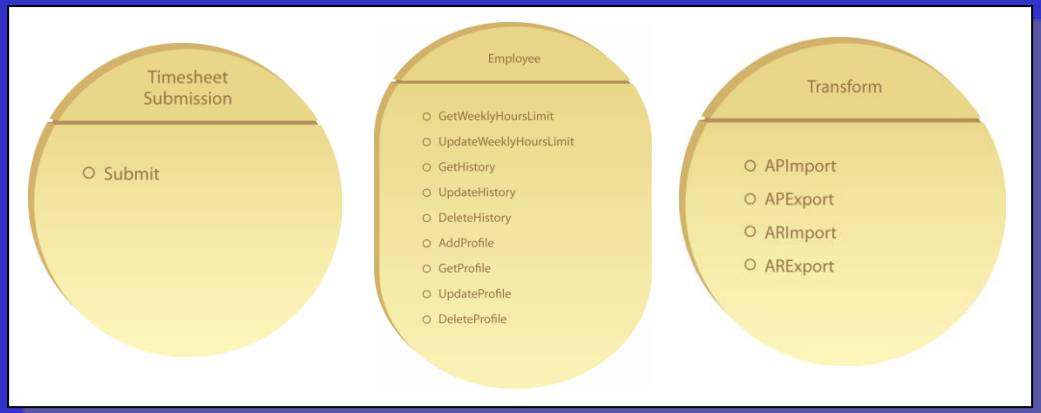
Services can be collectively modeled so that each service boundary is planned out so as to ensure that it does not overlap with other services.

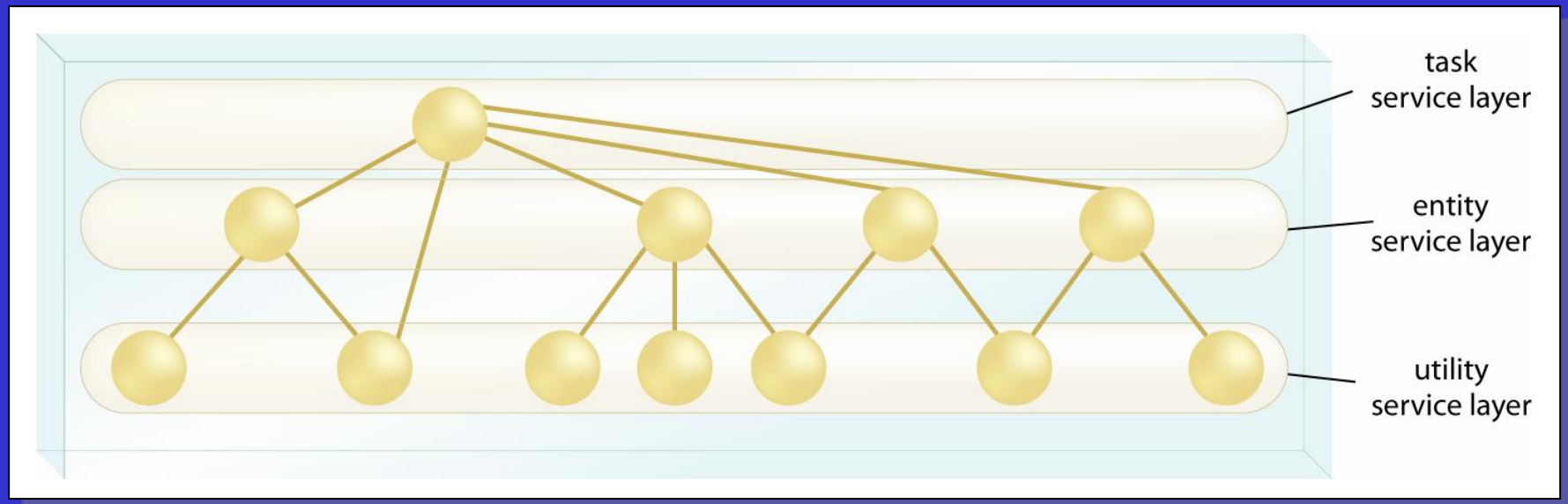
The result is a service inventory with a higher degree of functional normalization.



Service Models & Service Layers

Within an inventory services are classified using service models and organized into logical service layers.





Service Inventory

Common service inventory design issues include:

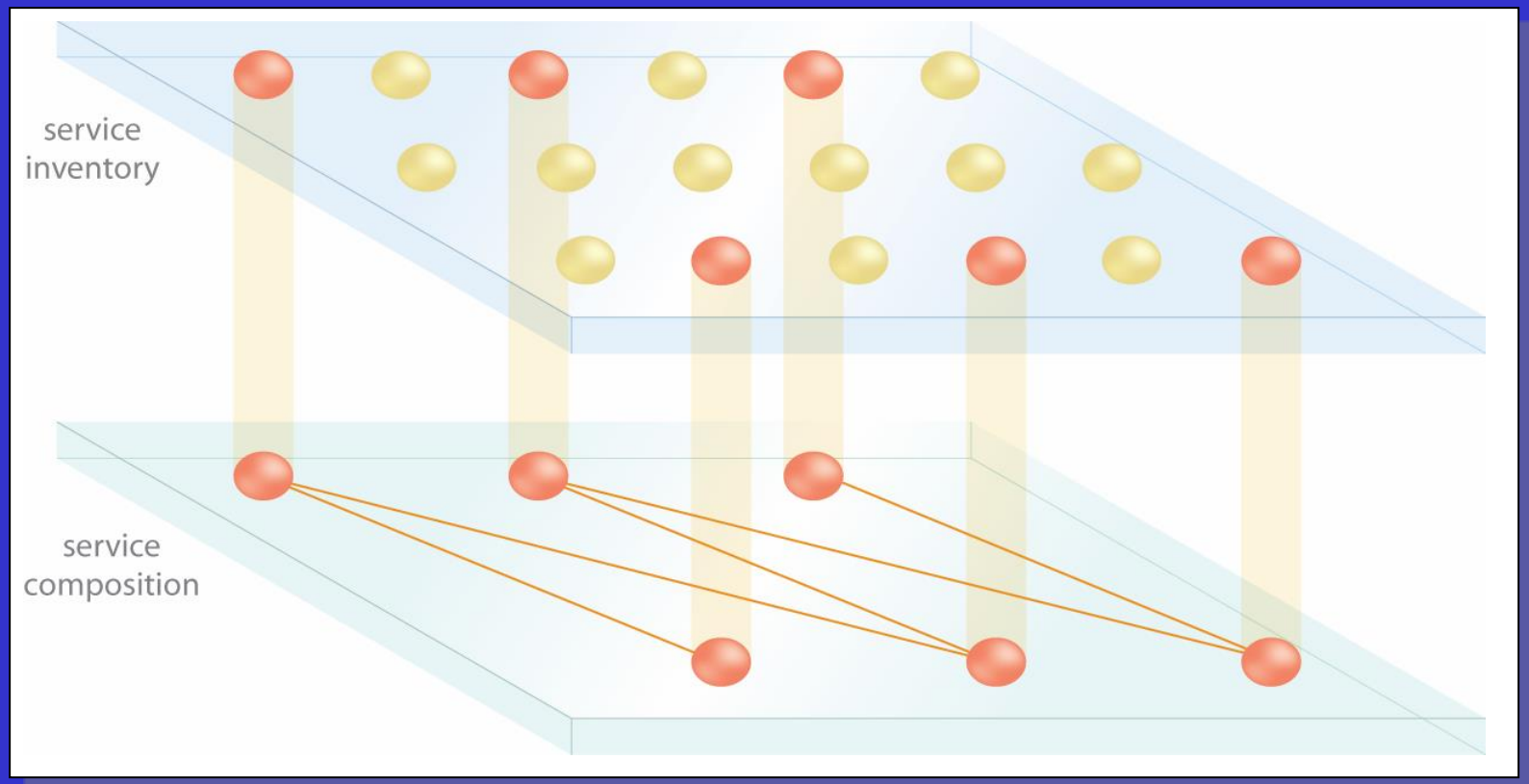
- scope and boundary
- service standardization
- scalability
- runtime platforms
- infrastructure
- governance

Service Inventory

- The strategic goals and benefits of service-oriented computing are realized within the boundaries of service inventories.
- The fundamental dynamic a service inventory must successfully support is the composition and re-composition of services.
- To achieve this the inventory must itself be supported by capable technologies, platforms, and infrastructure.

The Re-Composition Dynamic

Agnostic services within a service inventory are constantly available for repeated composition.



Service-Oriented Computing

"Service-oriented computing is an umbrella term used to represent a distributed computing platform based on service-orientation.

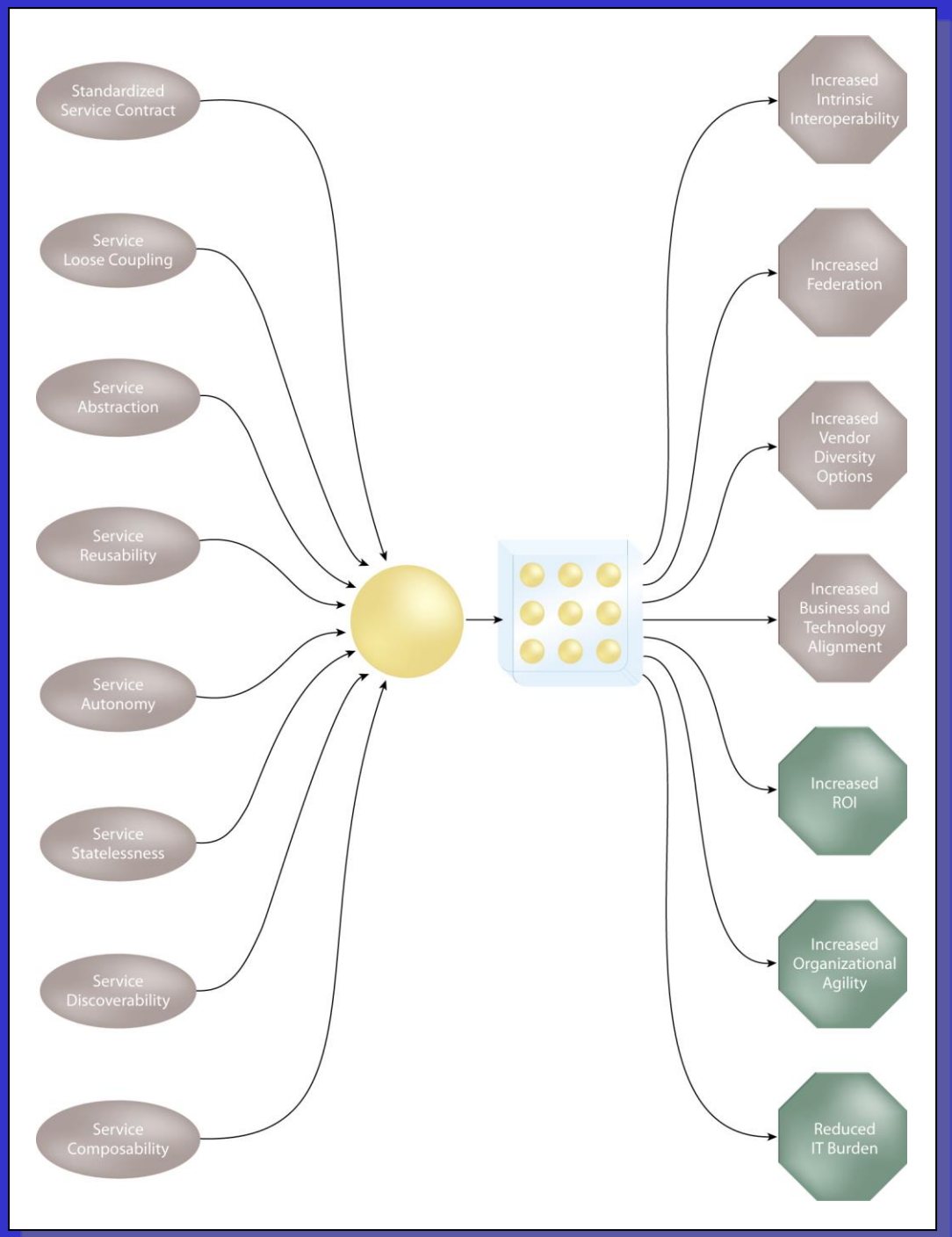
As such, it encompasses many things, including its own design paradigm and design principles, design pattern catalogs, pattern languages, a distinct architectural model, and related concepts, technologies, and frameworks."

ServiceOrientation.com

Service-Oriented Computing

- Everything we've discussed so far is part of the service-oriented computing platform.
- Also included would be all of the relevant vendor products, open source technologies, standards specifications, etc.
- Service-oriented computing essentially represents a specialized type of distributed computing.

Service-orientation (along with the individual elements of service-oriented computing covered in this section) are all collectively applied to the design of each service within a service inventory, in pursuit of the overarching strategic goals and benefits of service-oriented computing.



Exercise

Exercise 1.2

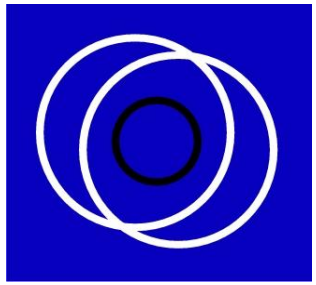
Fill in the Blanks

References

Services, Service Compositions, Service Inventories, Service-Orientation are further covered in:

Module 3: SOA Design & Architecture
Module 8: Advanced SOA Design
Module 9: SOA Design Lab

Service Delivery Lifecycle



SOA Project Delivery

Prior to commencing with an SOA initiative, a project delivery approach needs to be chosen to best organize the overall delivery lifecycle.

Three common approaches exist:

- top-down
- bottom-up
- agile (or meet-in-the-middle)

Of these, some extent of top-down delivery is usually required to incorporate meaningful levels of inventory and service analysis.

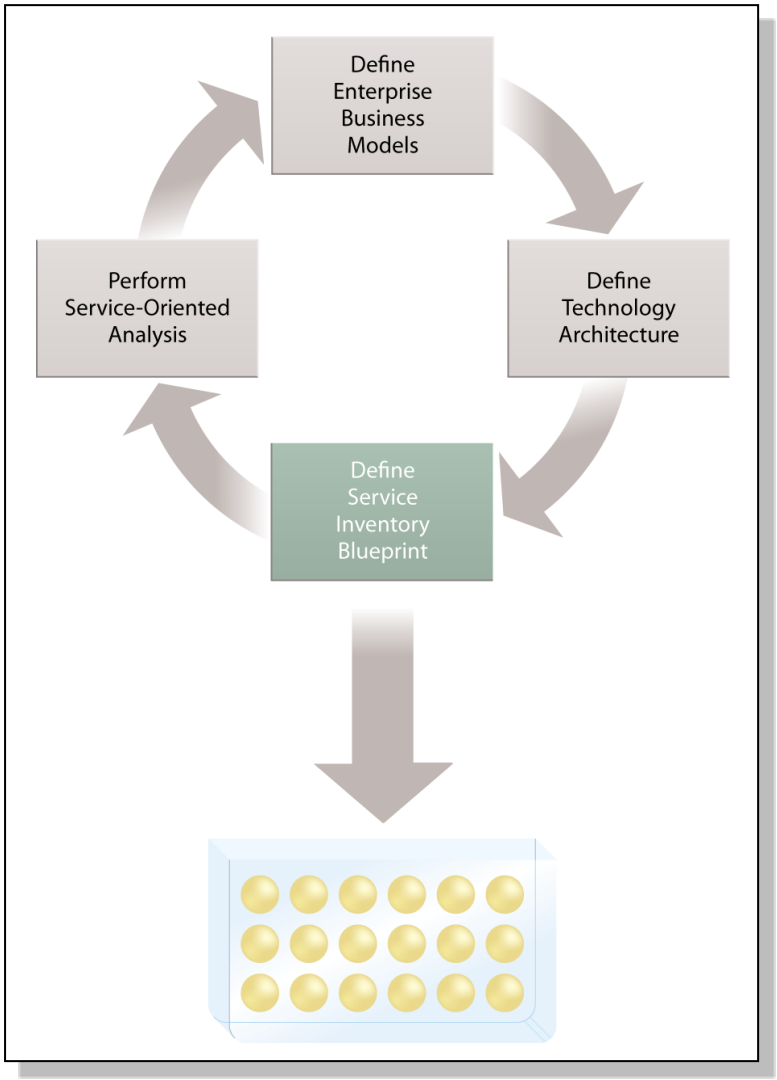
Primary Service Delivery Lifecycle Stages

- Service-Oriented Analysis
- Service Modeling
- Service-Oriented Design
- Service Development
- Service Implementation

Subsequent to these stages, service governance phases take over.

Service Inventory Analysis Cycle

The service inventory analysis as part of the top-down delivery process is comprised of an iterative cycle during which the service inventory blueprint is incrementally defined as a result of repeated iterations of steps that include the service-oriented analysis.



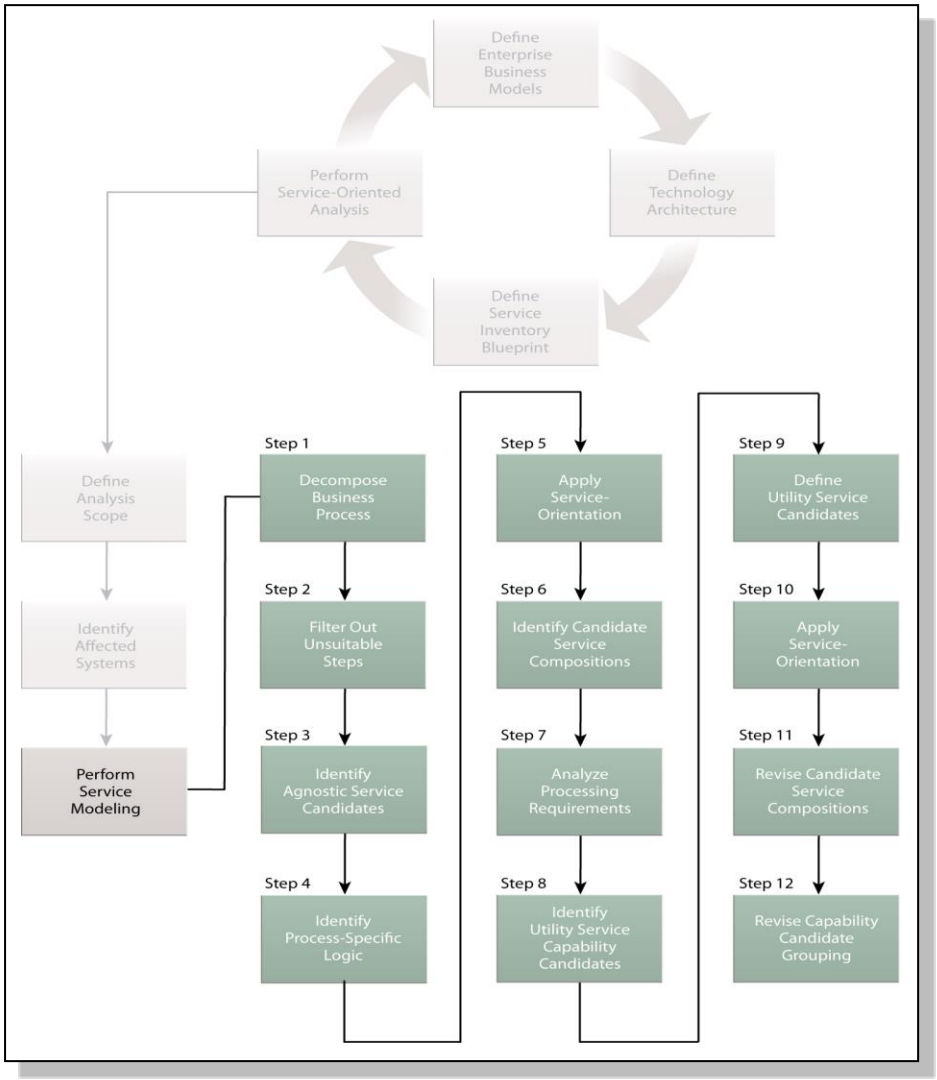
Service-Oriented Analysis

- SOA emphasizes a direct relationship between business analysis intelligence and the services that end up representing and implementing business logic.
- This results in the requirement for a unique form of analysis that needs to be completed prior to the design of individual services.
- Service-oriented analysis establishes a process through which service-orientation is applied to business automation logic and requirements.

Service Modeling

Service modeling is the part of the analysis phase during which services and their capabilities are conceptualized prior to their actual physical definition and development.

Conceptualized services are called "service candidates".



Business-Centric Focus

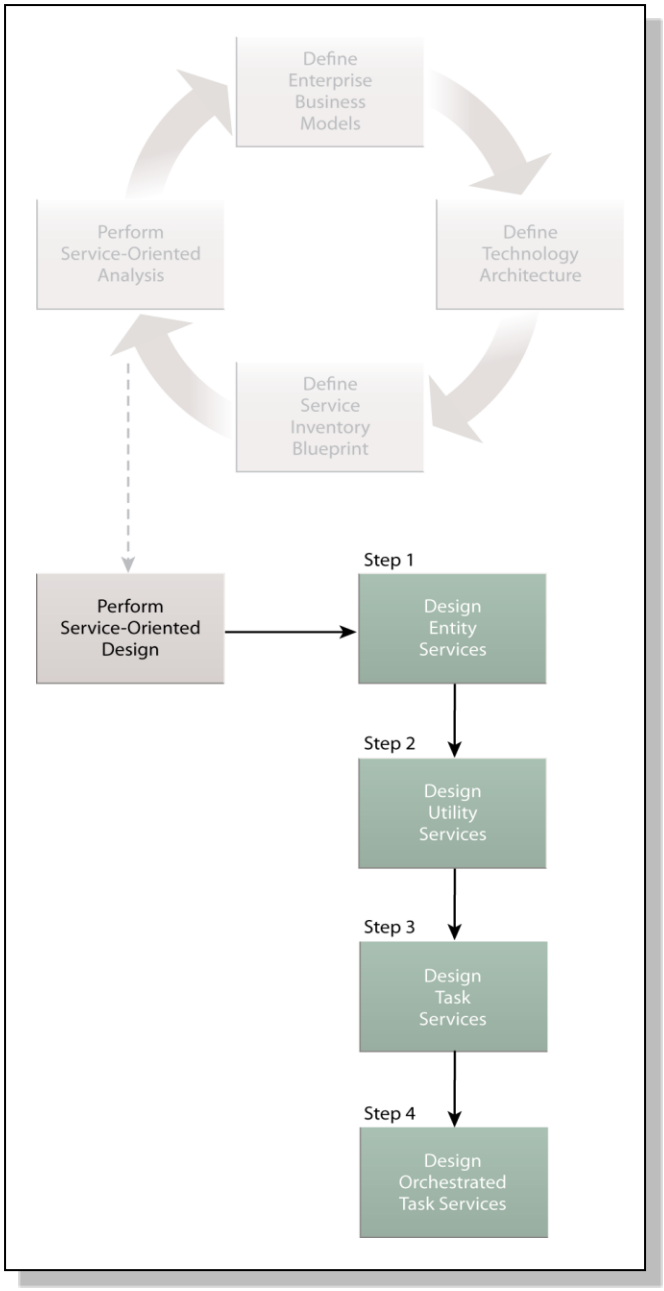
- Much of service modeling is focused on the encapsulation and abstraction of business logic.
- The concept of the "business service" needs to be embraced and requires close collaboration between technology and business professionals.
- The iterative nature of service-oriented analysis allows for business service candidates to be continually refined before moving to design.
- Those who participate in the modeling of services often end up involved in the eventual governance of services.

Service-Oriented Design Processes

Service-oriented design continues from where service-oriented analysis leaves off.

There is a separate service-oriented design process for each service model.

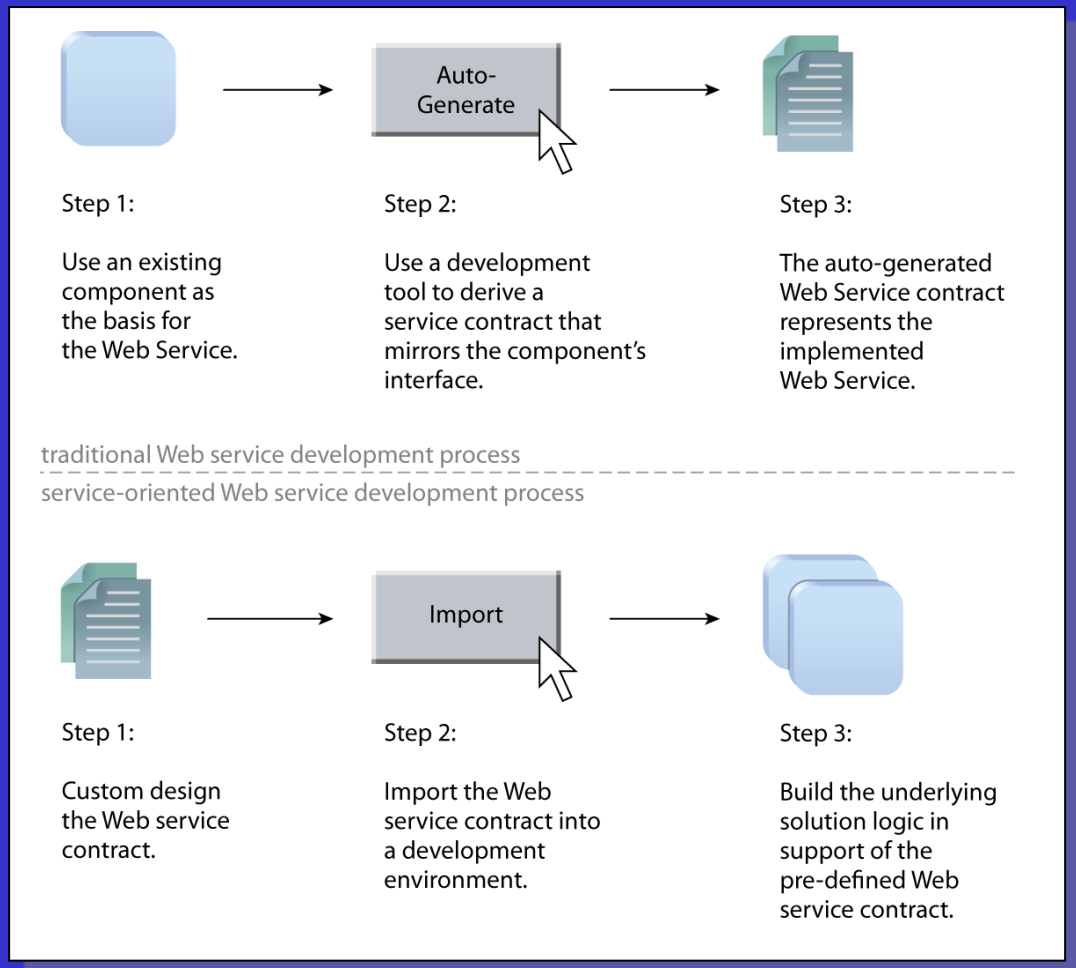
Service candidates become the starting point for these processes.



"Contract First" Design

Service-orientation places an emphasis on standardizing and decoupling the technical contract of each service.

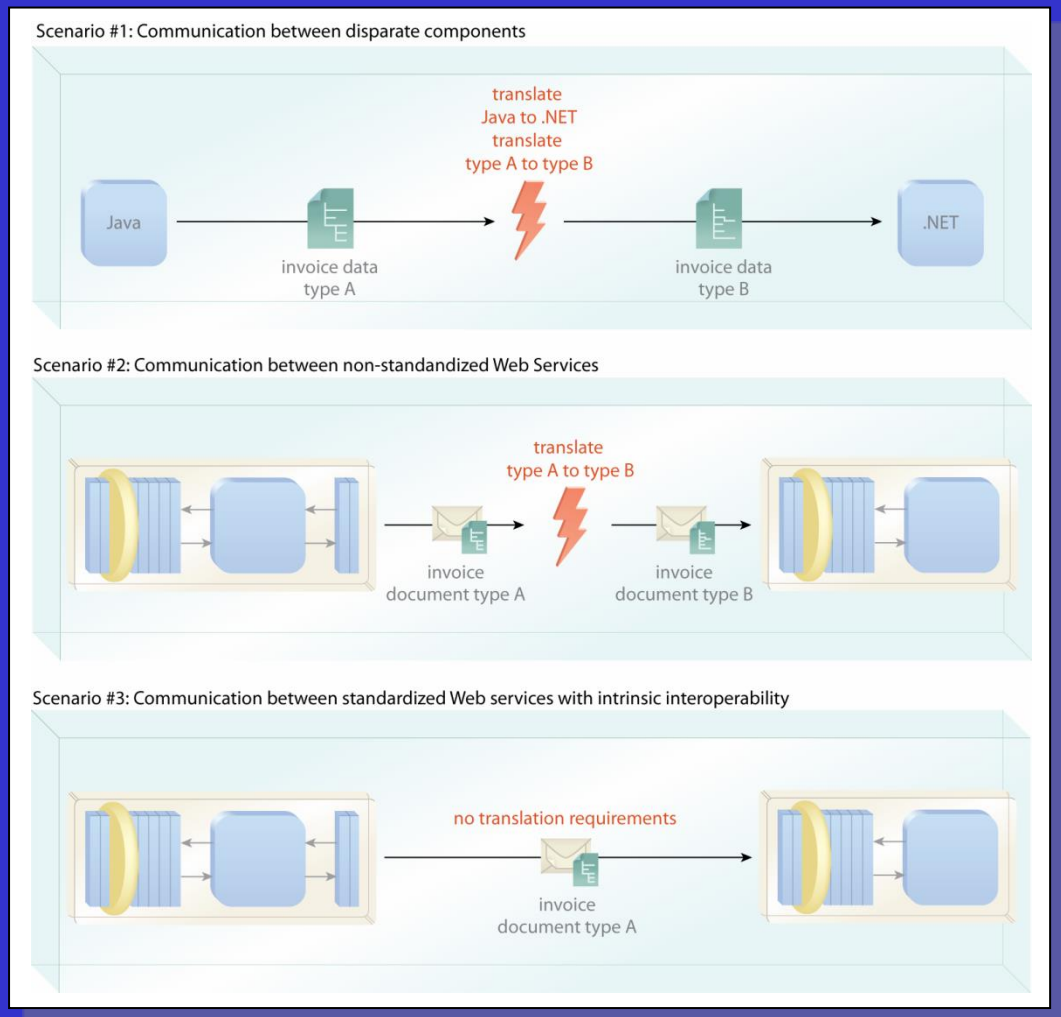
Service-oriented design therefore is based on a contract first approach, requiring developers to avoid the use of auto-generation tools.



Transformation Avoidance

One of the primary goals of service-oriented design is the long-term avoidance of transformation technologies in support of fostering intrinsic interoperability.

This relies heavily on service contract standardization.



Service Inventory Governance

Governance can encompass many things, but from a delivery perspective, the eventual burden of governing services is a primary consideration.

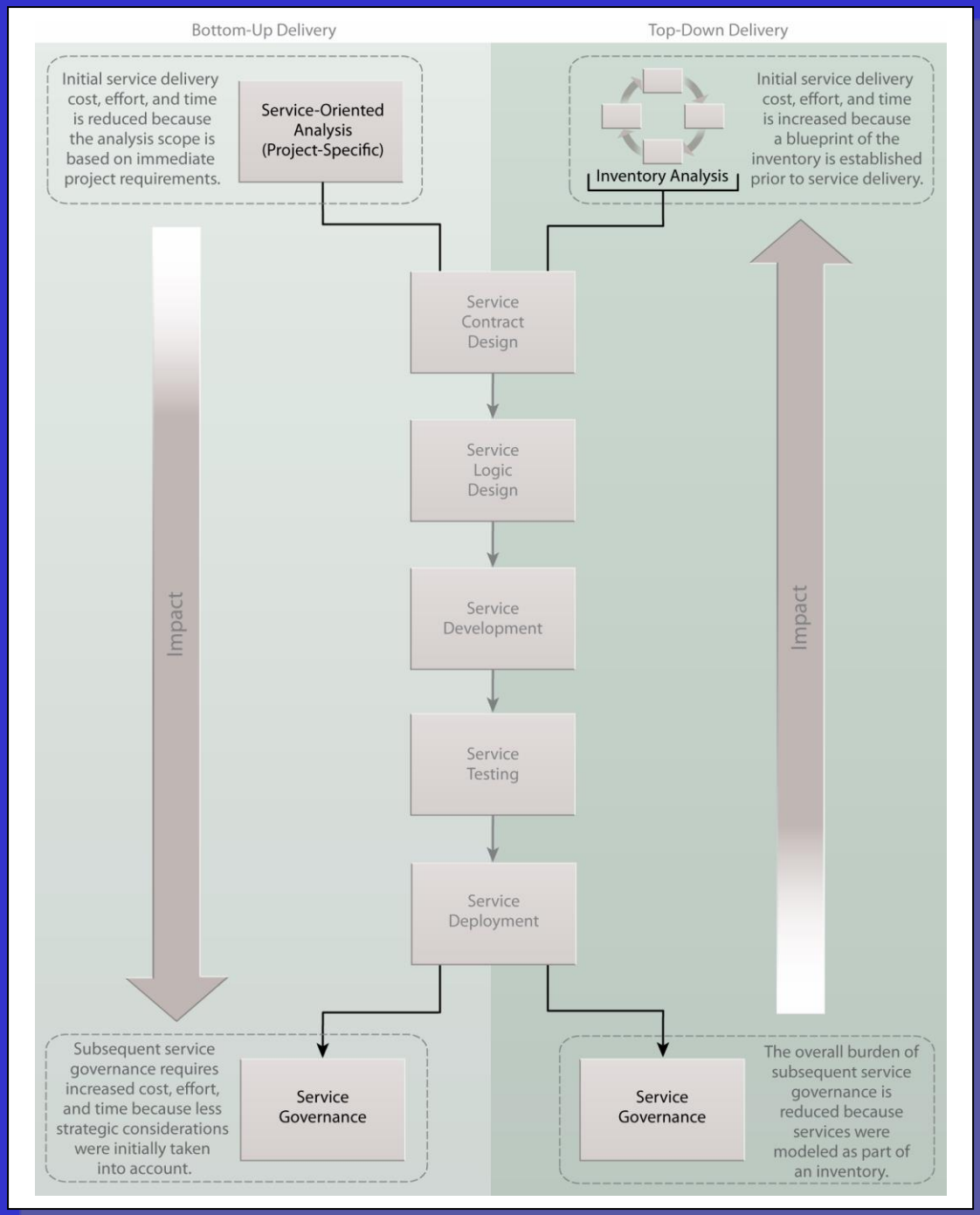
As a rule of thumb, the following applies:

- Up-front analysis as part of a top-down effort reduces the governance burden.
- The bottom-up approach results in less up-front impact, but defers burden to the governance phase.

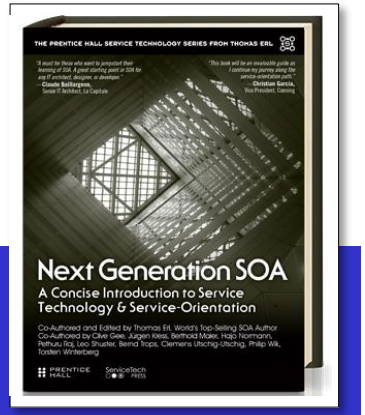
Service Inventory Governance

Each service delivery strategy has its own advantages and trade-offs.

The key is to always take the governance burden into account when deciding on a delivery approach.



Optional Reading



"Next Generation SOA: A Concise Introduction to Service Technology & Service-Orientation"

Pages 26 - 46

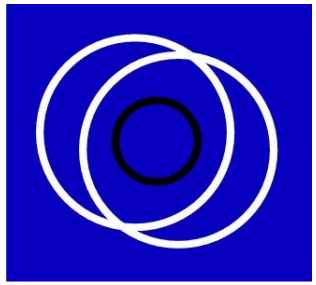
Chapter 3: *A Look at How Services are Defined and Composed*

References

SOA Project Lifecycles, Service-Oriented Analysis, Service Modeling, Service-Oriented Design, and SOA Project Roles are covered in:

Module 4: SOA Project Delivery & Methodology
Module 6: Advanced SOA Analysis & Modeling
Module 7: SOA Analysis & Modeling Lab

Adoption Impacts and Requirements



Adoption Impacts and Requirements

Primary impacts and requirements:

- Standardization Considerations
- Organizational Considerations
- Governance Considerations
- Infrastructure Considerations
- Maturity Considerations
- Migration Considerations

Note that many of these considerations are inter-related.

Standardization Considerations

- Industry standards are generally represented by technology specifications that are produced by standards organizations and developed by technical committees.
- WSDL, SOAP, WS-Addressing and WS-Policy are examples of industry standards relevant to building services as Web services.
- XML and XML Schema are examples of industry standards used to represent and express data exchanged by services.

Standardization Considerations

- Industry standards help standardized technologies in support of vendor-neutral service-oriented architectural models.
- However, the use of industry standards alone does not result in fully standardized SOA implementations.
- Design standards (or custom design standards) are needed to achieve the level of standardization required to support the goals of service-oriented computing.

Standardization Considerations

- Design standards provide custom design conventions specific to an enterprise or specific to a domain within the enterprise.
- For example, design standards will determine which industry standards (and which versions of industry standards) will be used.
- Design standards are produced by an IT department (as opposed to industry standards which are produced by technical committees).

The standardization of data models within service inventory boundaries is a key part of the overall standardization effort.

Standardization Considerations

- Design standardization of service contracts results in the avoidance or reduction of transformation requirements.
- Challenges to achieving design standardization are often more cultural than technical.
- Design standards enforcement is often a key point of conflict.
- Sometimes significant measures are required, such as making standards compliance legally binding.

Organizational Considerations

SOA initiatives can introduce many new considerations that can impact the organizational complexion in a variety of ways.

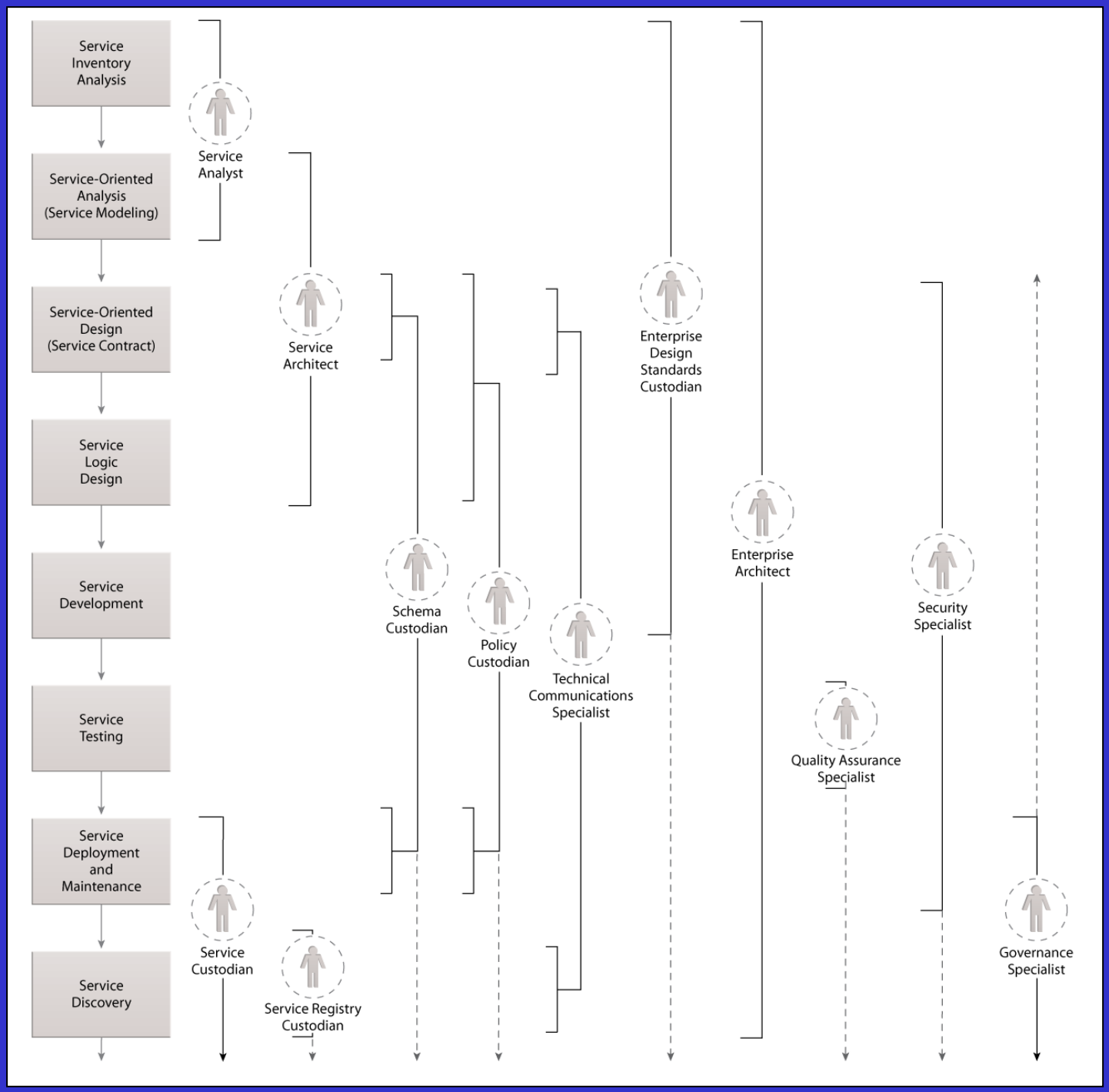
For example:

- new roles and processes
- new skill-set and resourcing requirements
- changes to project team structure & delivery lifecycles
- changes to business analysis and modeling approaches
- the use and enforcement of internal design standards

Organizational Considerations

- One of the most fundamental impacts originates from the emphasis on creating highly agnostic services.
- Once a collection of agnostic services exists, standards, processes, and custodians need to be in place to properly administer and govern their usage.
- Depending on the scope of the SOA initiative, this can lead to a shift in how the IT department itself is structured.
- Often the need for more enterprise-centric IT staff arises, while the demand for project-centric staff decreases.

Common SOA Project Roles



Organizational Considerations

- The required organizational changes of an SOA adoption are often underestimated leaving organizations unprepared for these impacts.
- Furthermore, there is often resistance from parts of the organization to these changes.
- Planning, communication, and education are key factors.
- Project teams need to understand why the major (mostly long-term) benefits are worth the significant (mostly short-term) impacts.

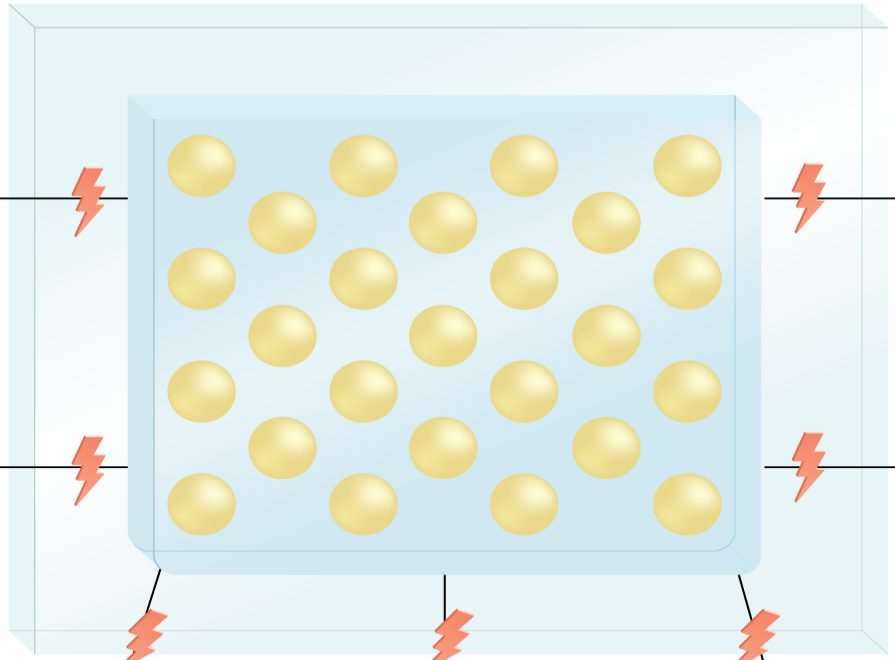
Cultural Impacts

"There is too much up-front work required and too many restrictive standards for us to deliver projects efficiently and within reasonable budgets."

Project Managers

"We simply cannot give up the amount of control we'd be required to in order to comply with all of these new standards and processes."

IT Managers



Architects

"The enterprise architecture is too complex and half of the IT department is ignoring our design standards."

Business Analysts

"We don't have accurate and current enterprise-wide business models, and creating these will turn into a huge analysis project."

Data Analysts

"There is so much existing data disparity that establishing a global data model would take years."

Developers

"There are so many concurrent projects that we're never sure that what we're building isn't already being built by another team."

End-Users

"It's taking far too long for the planning stage to be completed before anything real is actually being built."

Governance Considerations

Governance is the act of governing or administrating something. It is a means by which an organization makes decisions about decision-making.

Within IT, a governance system is responsible for organizing, directing, and guiding the creation and evolution of IT assets and resources.

A governance system:

- establishes constraints and parameters that control, guide, or influence decisions
- determines who has responsibility and authority to make decisions
- prescribes consequences for non-compliance to required constraints

Governance Considerations

- IT governance represents the responsibility of regulating and evolving IT ecosystems throughout their lifecycles.
- An SOA governance system can relate to all parts of an SOA ecosystem, including:
  - services
  - service compositions
  - service-oriented solutions
  - service-oriented architectures
  - supporting infrastructure
  - supporting business models and specifications

Governance Considerations

SOA governance requirements can impose organizational challenges that affect:

- policies and regulations
- allocation of staff
- creation of new processes
- internal design standards
- compliance checking and enforcement
- service ownership

etc…

Governance Considerations

SOA governance can also impose technology challenges related to:

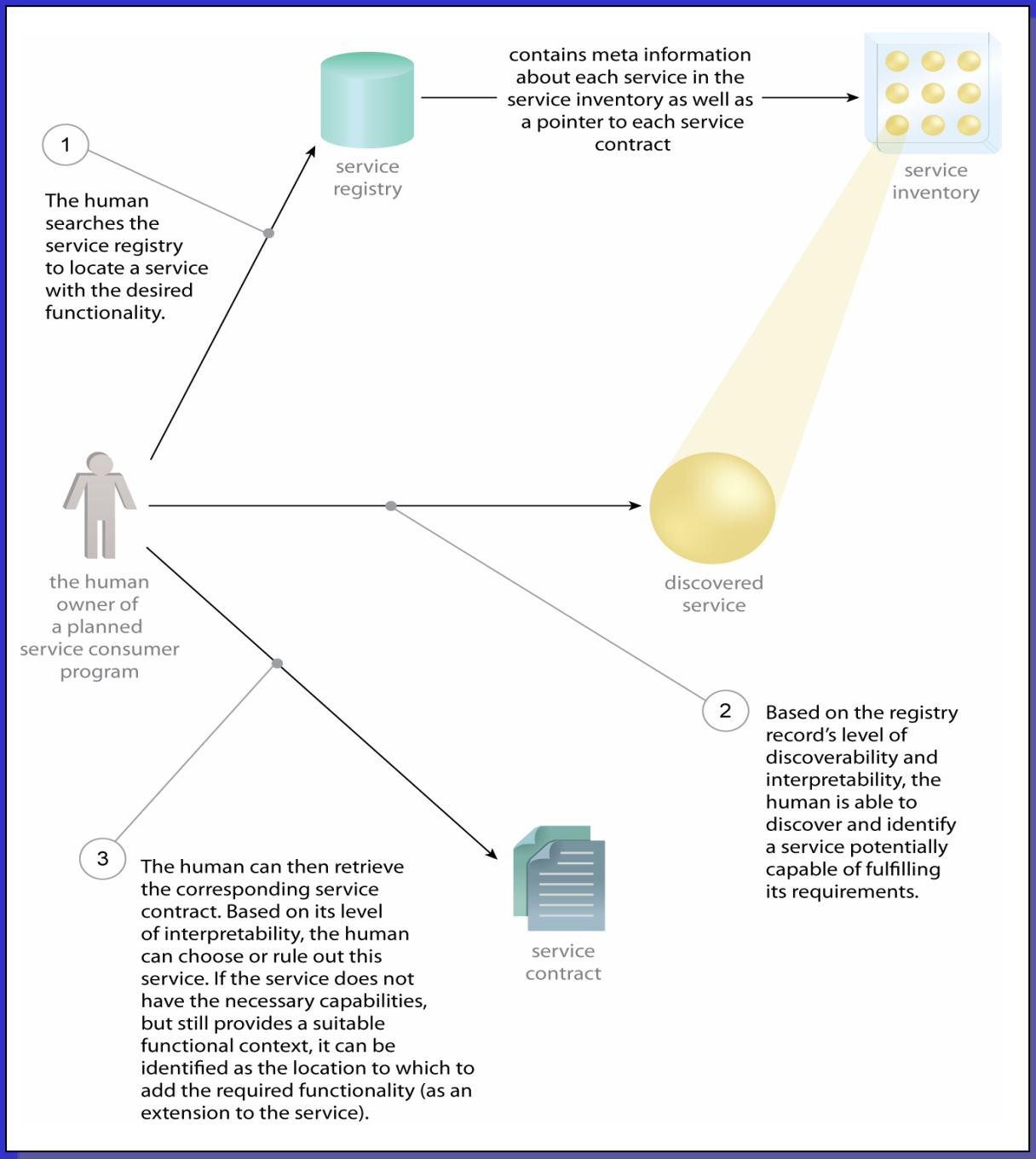
- service monitoring
- service vitality
- service versioning
- accommodation of changing service consumer requirements
- the need for modern middleware

etc…

Governance Considerations

- An example of a governance consideration especially emphasized by SOA initiatives is the introduction of a discovery process.
- A service registry is established so as to allow public service metadata to be centrally registered and maintained and discovered by potential consumer program designers.
- One of the eight service-orientation design principles (Service Discoverability) is actually dedicated to facilitating this process.

A common service discovery process revolving around the introduction of a central service registry.



Governance Considerations

- Governance is often the least understood and least planned for aspect of SOA.
- Absence of a governance plan (and budget) can jeopardize the success of SOA initiatives because of their focus on long-term strategic goals.
- SOA governance impacts are closely related to organizational and infrastructure impacts.

References

SOA Governance is covered in:

Module 15: Fundamental SOA Governance
Module 16: Advanced SOA Governance
Module 17: SOA Governance Lab

...as part of the SOA Governance Specialist Certification program.

Performance Considerations

Performance challenges are a primary design concern when building complex service-oriented solutions because of the following:

- dependence on XML processing
- high reuse of agnostic services and related scalability demands
- the need for larger, more complex service compositions

and others…

Performance Considerations

- Because modern service-oriented solutions are based upon layers of XML document processing, they are subject to the associated performance overhead imposed by these layers.
- Service-oriented solutions built with Web services deepen the dependency on XML data representation, which can magnify XML processing-related performance challenges.
- Many technology advances address this concern, including advanced XML processors, hardware accelerators, and even increased XML parsing capabilities within microprocessors.

Performance Considerations

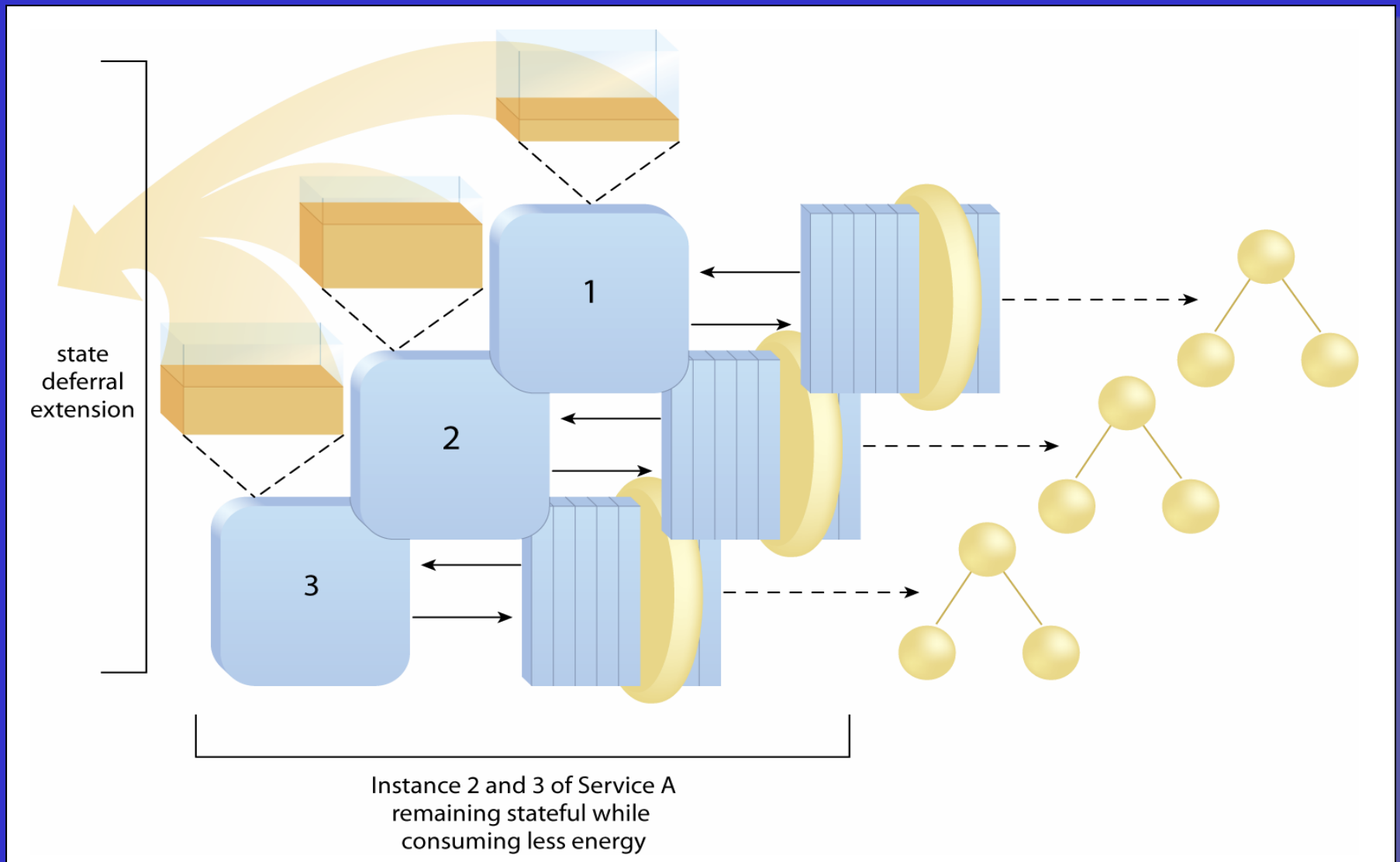
- Well designed agnostic services are reusable and therefore subject to increased concurrent usage.
- Reuse applies both to multiple instances of the same service for the same parent task and reuse of the service in support of multiple parent processes.
- Furthermore, agnostic services contain generic processing logic that can naturally consume more runtime cycles.
- Clustering, grid technologies, and redundant implementations are used to address this concern so as to guarantee predictable performance during periods of high concurrent usage.

Performance Considerations

- Each data exchange between services introduces a level of processing overhead.
- When combining multiple services into compositions, this overhead is accumulated, yet it still needs to be reasonable so as to not introduce latency.
- The creation of agnostic services tends to result in more finer grained services that increase the size of compositions.
- Modern runtime environments need to accommodate this to allow services to be continually composed and recomposed.

Performance Considerations

- Every environment will have finite processing limits that set performance boundaries that set constraints upon SOA projects.
- A key goal in any SOA transition plan is to identify these performance limitations in advance and then plan to accommodate them.
- These constraints can even be taken into account when defining a service inventory blueprint. They help determine the appropriate level of service granularity because compositions are modeled out ahead of time.

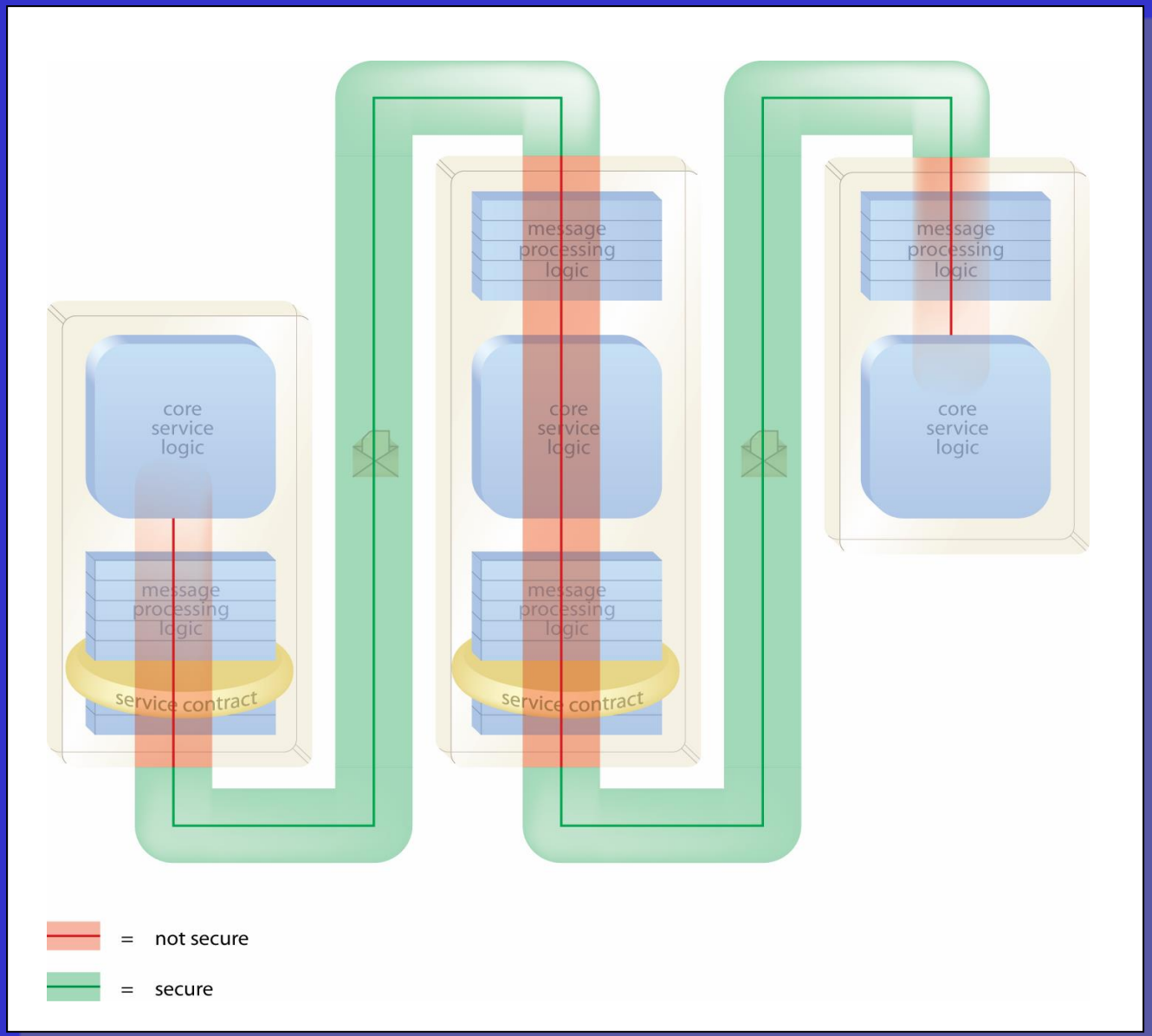


An example of how service scalability is accommodated is via the runtime deferral of state data.

Security Considerations

- Once services begin to take on greater amounts of processing responsibility, the need for message-layer security measures as well as security controls that apply to shared services begins to arise.
- The WS-Security framework (and related extensions) enable message-level security for services built as Web services.
- Other security concerns include single sign-on/ federated identity management.

Common transport-layer encryption (such as SSL) does not protect message contents when they are being processed within service boundaries.



References

SOA Security is covered in:

Module 18: Fundamental SOA Security
Module 19: Advanced SOA Security
Module 20: SOA Security Lab

...as part of the SOA Security Specialist Certification program.

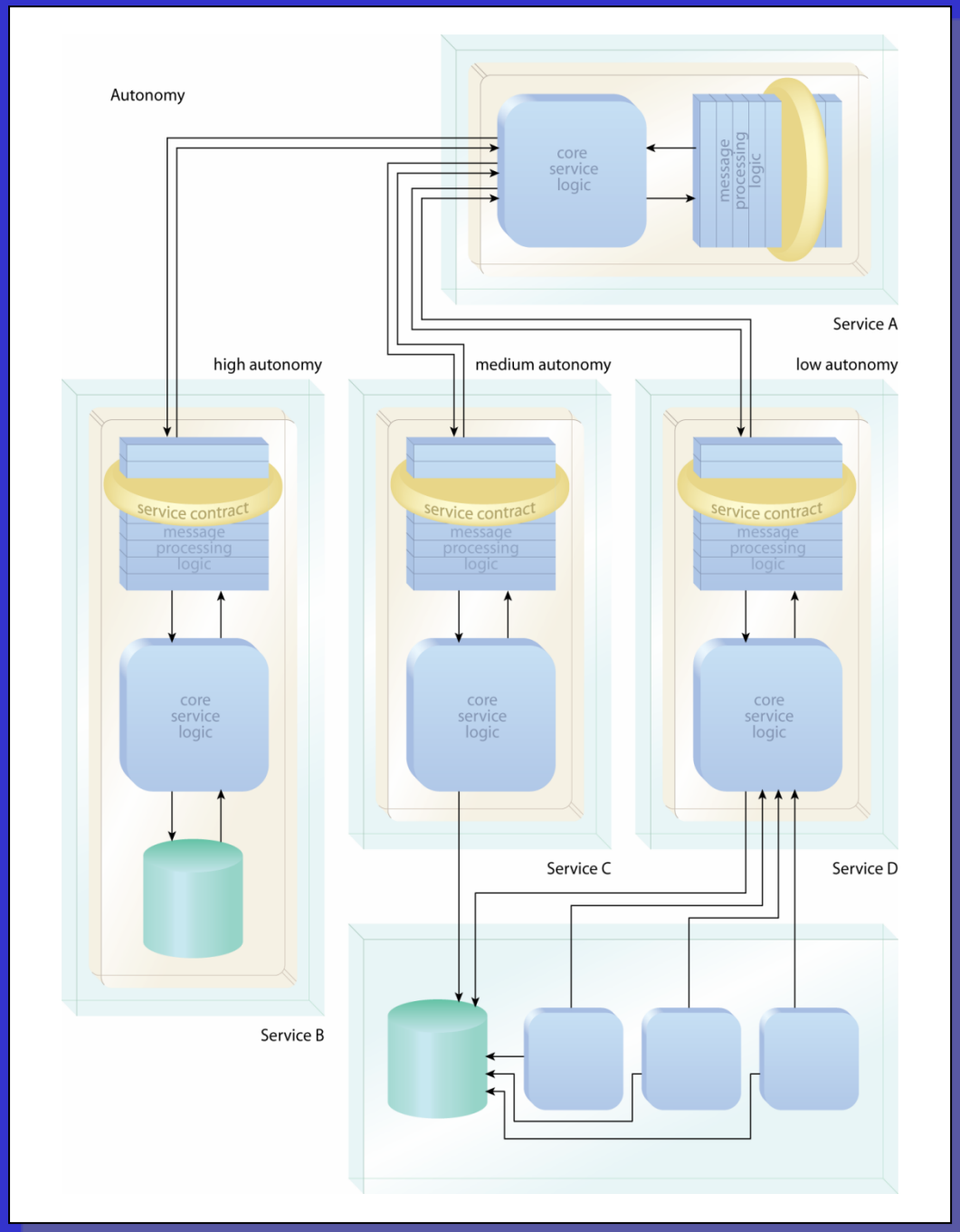
Other Infrastructure Considerations

All of the considerations discussed so far affect the infrastructure required for service-oriented architecture implementations.

Additional areas of impact include:

- legacy encapsulation
- orchestration
- reliability
- failover
- availability

When studying the implementation architectures of service compositions, infrastructure-related impacts become more evident, especially when legacy encapsulation is involved.



Maturity Considerations

- Industry-level SOA is not defined by any one vendor. In support of vendor diversification options and organizational agility, it is therefore important to maintain a clear separation of architectural model, design paradigm, and implementation technology options.
- Any current SOA planning efforts need to take the state and direction of the marketplace into consideration.
- Current areas that continue to mature include governance platforms, security, and high volume processing.

Maturity Considerations

- Vendor marketing campaigns have been responsible for some confusion.
- Marketing and branding of "SOA support" by products and technologies is often questionable.
- To best assess the suitability and maturity of available products and technologies the "SOA" branding needs to be ignored.
- Service-orientation and enterprise requirements are the fundamental criteria.

Migration Considerations

- An SOA transition plan allows you to coordinate a controlled phasing in of service-orientation so that the migration can be planned on a technological, architectural, and organizational level.
- Each plan is unique to an organization's requirements, constraints, and goals.
- A central repository needs to be established for all related documents.
- Transition plans attempt to balance tactical requirements with the ultimate attainment of the strategic benefits of service-oriented computing.

Migration Considerations

A typical SOA transition plan will include:

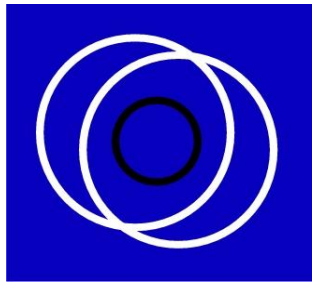
- organizational business vision
- impact analysis
- dynamic transition architectures and related milestones
- phasing strategies
- estimated inventory scopes

more…

Summary

- An SOA initiative does not need to be enterprise-wide to provide benefit, nor does it require the creation of a global data model.
- Most SOA benefits and ROI are long-term, most investment and impact is up-front.
- The more up-front effort you put into each service, the less of a governance burden the service becomes.
- Focusing on concepts, principles and requirements enables you to view the SOA industry with increased clarity.

Introduction to Common Service Technologies



Overview

- SOA and XML
- SOA and XML Schema
- Services as Components
- Services as Web Services
- Services as REST Services
- API Management *
- Social Network Technologies *
- Mobile Computing *
- Big Data *
- Service Virtualization *
- Cloud Computing *

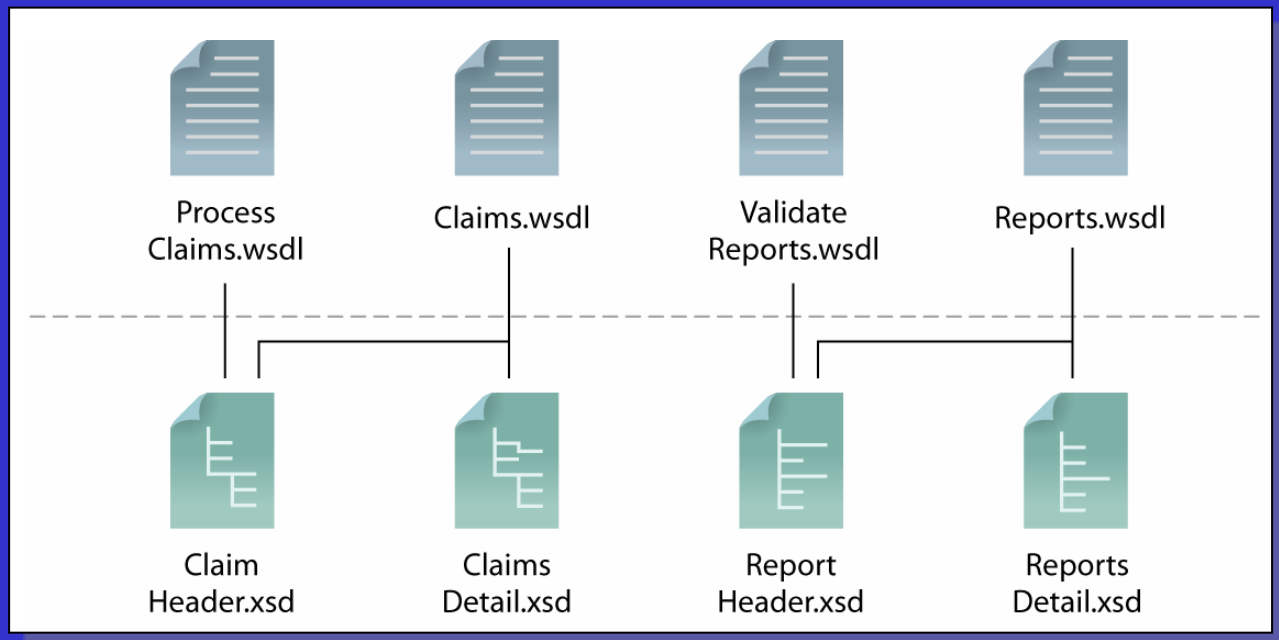
** Topics not covered on Exam S90.01*

SOA and XML

- XML represents a foundation technology platform for just about any modern service-oriented architecture implementation.
- XML and XML Schema establish the format and structure of messages traveling throughout services and service compositions.
- As a result, XML Schema can become a fundamental part of service contract design.
- The data representation layer relies on the creation of standardized XML schemas.

SOA and XML Schema

- A service layer will typically be built upon a separate data representation layer.
- The service layer establishes a series of federated endpoints (service contracts).
- The data representation layer establishes a series of standardized data models that are shared by the services and typically exist as XML schemas.



Service Implementation Mediums

- SOA as an architectural model is neutral to any one technology platform.
- Essentially any implementation technology that can be used to create distributed systems may be suitable for service-orientation.
- Current technologies for building services:
  - components
  - Web services (or SOAP-based Web services)
  - REST services

Services as Components

- A component is a software program designed to be part of a distributed system.
- A component exposes a technical interface with capabilities called methods, allowing it be explicitly invoked by other programs.
- Components are typically proprietary to a specific platform (i.e. Java or .NET).
- Service-orientation can be applied to components in order to design them as services.

Services as Web Services

- The Web services technology platform is based on a set of established industry standards.
- Core technologies are WSDL, SOAP and UDDI, while second generation technologies are collectively referred to as WS-*.
- Web services are also called SOAP-based Web services to distinguish them from other types of services that use Web technology.
- Service-orientation can be applied to SOAP-based Web services in order to design them as services.
- The distinction between a service and a Web service can be confusing because both terms use the word “service”.

Services as Web Services

- Probably the number one obstacle organizations have faced when carrying out SOA projects is trying to build traditional distributed solutions under the pretense that they are building actual service-oriented solutions.
- This often comes from the incorrect assumption that the mere use of Web services constitutes a service-oriented solution.
- This trend has led to many failed projects and has also skewed many analysis reports regarding SOA in general.

Services as REST Services

- REST stands for Representational State Transfer, a term established in Roy Fielding's Ph.D. dissertation.
- REST services do not have individual service contracts, but instead share a uniform contract via the standard HTTP protocol.
- REST services typically communicate with each other using HTTP methods, such as GET and PUT.
- Service-orientation can be applied to REST services.
- As with SOAP-based Web services, the distinction between a service and a REST service can be confusing because both terms use the word "service".

API Management

- Shared services are commonly used in public domains to provide crossover functions, such as adding content from multiple social media accounts or public data services, mashing content with maps and embedding videos.
- To facilitate these types of high-traffic interactions, commercial-grade APIs are created, typically based on mainstream Web-based service technologies, such as SOAP and REST.
- A number of vendors offer API management solutions for these types of shared services that include API definition, billing, intermediation, security, analytics, and auditing.
- Web-based API management tools and platforms can assist with the management of both in-house and external third-party services that are part of service-oriented solutions.

Social Network Technologies

- The advent of social media has led to the emergence of Web culture communities and hosted services such as social networking sites, video-sharing sites, wikis, and blogs.
- These types of services can be designed to seek interconnections and consume and remix data from multiple sources.
- The extent to which services that encapsulate this type of functionality can make their way into service-oriented architectures depends on how relevant the data that they process is to serving an organization's business needs.
- Services as part of service-oriented solutions can be designed to receive data streams for social networking services or they can be designed to fully interface with social networks.

Mobile Computing

- Mobile devices provide alternative channels for exposing existing services to new potential service consumers.
- Consequently, mobile devices have introduced user-interfaces with different resolutions and screen sizes, as well as different interaction mechanisms
- While existing service logic can be shared and extended to support the use of mobile devices, user-interface and communication-related limitations typically impose the need for further processing and presentation logic customization.

Mobile Computing

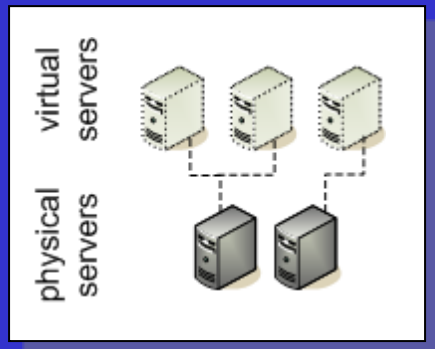
- On the server side, communicating with mobile apps often requires a new channel-specific technical layer on top of existing service layers.
- Business-centric mobile applications further require users to provide direct input through mobile applications as part of the completion of business process activities.
- Business-centric mobile applications can introduce the need for dedicated services that abstract the logic required to cope with the distinct requirements and demands of mobile computing.

Big Data

- Big Data is defined as a field dedicated to the analysis, processing, and storage of large collections of data that frequently originate from disparate sources.
- The heterogeneous nature of Big Data requires that data be interpreted and handled in a consistent manner, which can be performed by services that are specifically built to deliver data to a variety of destinations.
- Consumers of these services are able to process data regardless of its source, syntax, or semantic meaning.
- Big Data and SOA are a natural fit for one another in how Big Data solutions can provide an enterprise with valuable insights that can be fed into and responded by service logic within service-oriented solutions.

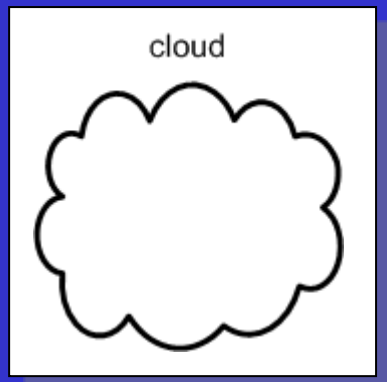
Service Virtualization

- Virtualization is an established technology that has enabled hardware owners to repeatedly leverage physical servers for wide, concurrent usage.
- Service virtualization separates the core service logic from the details of its invocation as a service.
- Virtualization allows physical IT resources to provide multiple virtual images of themselves so that their underlying processing capabilities can be shared individually by multiple consumers.
- A virtual server is a form of virtualization software that emulates a physical computer (a physical server). Each physical server can host multiple virtual servers



Cloud Computing

- Cloud computing is a specialized form of distributed computing that introduces utilization models for remotely provisioning scalable and measured IT resources.
- The term "cloud" originated with the cloud symbol used to represent the Internet.



- The terms "on-premise" or "on-premises" are used to qualify an IT resource that is not remotely accessible via a cloud, but instead resides within an internal IT enterprise environment.

Cloud Computing

- An IT resource is a physical or virtual IT-related artifact (software or hardware), such as a physical server, a virtual server, a storage device, or a software program designed as a service.
- A key business driver of cloud computing is the availability of IT resources for lease using a pay-for-use model by third-party cloud providers.
- This allows organizations to leverage vast cloud platform infrastructure resources without having to invest in establishing and maintaining this infrastructure on-premise.

Cloud Computing

- Behind the scenes, a cloud provider will typically rely heavily on virtualization technology to make shared resources available to many cloud consumers.
- Further related technology innovation has allowed clouds to provide dynamically scalable IT resources that can automatically scale in response to usage demands.
- Service-oriented solutions can compose services provided by clouds or they can be entirely hosted in cloud environments.

Cloud Computing

- Cloud computing (including service virtualization) is a separate topic area covered by an independent curriculum of 15 course modules as part of the Cloud Certified Professional (CCP) program from CloudSchool.com

- The next set of pages highlight some of the fundamental topics from the first two CCP course modules. The content on these pages is not part of the scope of this course module.

- As explained shortly, a separate Cloud Computing & SOA Connection Points supplement is available as part of the SOACP and CCP Module 1 Self-Study Kits. This supplement provides concrete mapping between SOA and cloud computing.

Cloud Computing Origins

Cloud computing emerged from a combination of business drivers and technology innovations.

Business Drivers:

- Capacity Planning
- Operating Overhead
- Organizational Agility

Technology Innovations:

- Grid Computing Technology
- Clustering Technology
- Virtualization Technology

Cloud Computing Benefits and Challenges

Benefits:

- Reduced Investment and Proportional Costs
- Increased Scalability
- Increased Availability and Reliability

Challenges:

- Increased Security Vulnerabilities
- Reduced Operational Governance Control
- Limited Portability Between Cloud Providers
- Multi-regional Compliance and Legal Issues

For descriptions of these benefits and challenges, visit www.whatiscloud.com.

Cloud Characteristics, Delivery & Deployment Models

Characteristics of a Cloud:

- On-Demand Usage
- Ubiquitous Access
- Multitenancy
- Elasticity
- Measured Usage
- Resiliency

Delivery Models:

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

Deployment Models:

- Public Cloud
- Community Cloud
- Private Cloud
- Hybrid Cloud

For descriptions of these characteristics and models, visit www.whatiscloud.com.

Cloud Computing Mechanisms

Cloud computing mechanisms represent the common architectural building blocks of cloud platforms:

- Virtual Server
- Ready-Made Environment
- Automated Scaling Listener
- Failover System
- Multi-Device Broker
- Pay-for-Use Monitor
- State Management Database
- Resource Replication
- Logical Network Perimeter
- Cloud Storage Device
- Hypervisor
- Load Balancer
- Resource Cluster
- Audit Monitor
- SLA Monitor
- SLA Management System
- Billing Management System
- Resource Management System
- Remote Administration System
- Cloud Usage Monitor

For descriptions of these mechanisms, visit www.cloudpatterns.org.

Cloud Computing & SOA: Connection Points Supplement

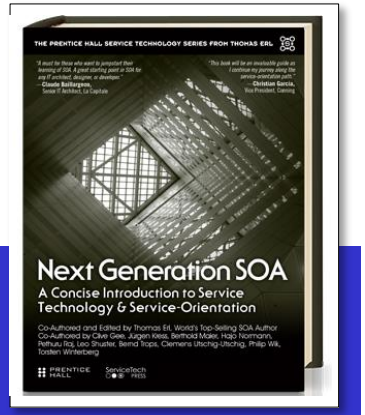
- The CCP curriculum exists independently from the SOACP curriculum, but preserves consistency in terminology, conventions, and notation.
- CCP course modules introduce new benefits, challenges, concepts, mechanisms, and characteristics that have been mapped to the SOACP course modules in order to establish a set of connection points.
- These connections points are helpful for SOA professionals that want to understand how SOA and service-orientation can be applied to cloud-based solutions – and to cloud computing professionals wanting to understand how cloud environments can benefit from SOA and service-orientation.
- The Cloud Computing & SOA Connection Points Official Course Supplement and DVD provides further coverage. These materials are included in the Module 1 Self-Study Kit.
- Also see www.youtube.com/arcitura/ for the multi-part online Cloud Computing & SOA Connection Points video.

Exercise

Exercise 1.3

Challenge the Myths

Optional Reading



"Next Generation SOA: A Concise Introduction to Service Technology & Service-Orientation"

Pages 64 - 78

Chapter 5: *An Overview of Service Technology* covers several additional service-related technologies.

Conclusion and Q&A

Recommended Reading:

- "SOA Principles of Service Design" Chapters 4, 6, 16, Appendix B
- "SOA Design Patterns" Chapters 3, 4
- www.serviceorientation.com
- www.servicetechspecs.com
- www.whatiscloud.com

Prometric Exam S90.01

The course you just completed corresponds to Exam S90.01, which is an official exam that is part of the SOA Certified Professional (SOACP) program.

For more information, visit www.soaschool.com/exams/

Exam S90.01 can be taken world-wide at Prometric testing centers.

For more information, visit www.prometric.com/arcitura/

Note that if you received this course booklet as part of an instructor-led workshop, you may be entitled to purchase discounted Prometric exam vouchers. For details, contact info@soaschool.com.

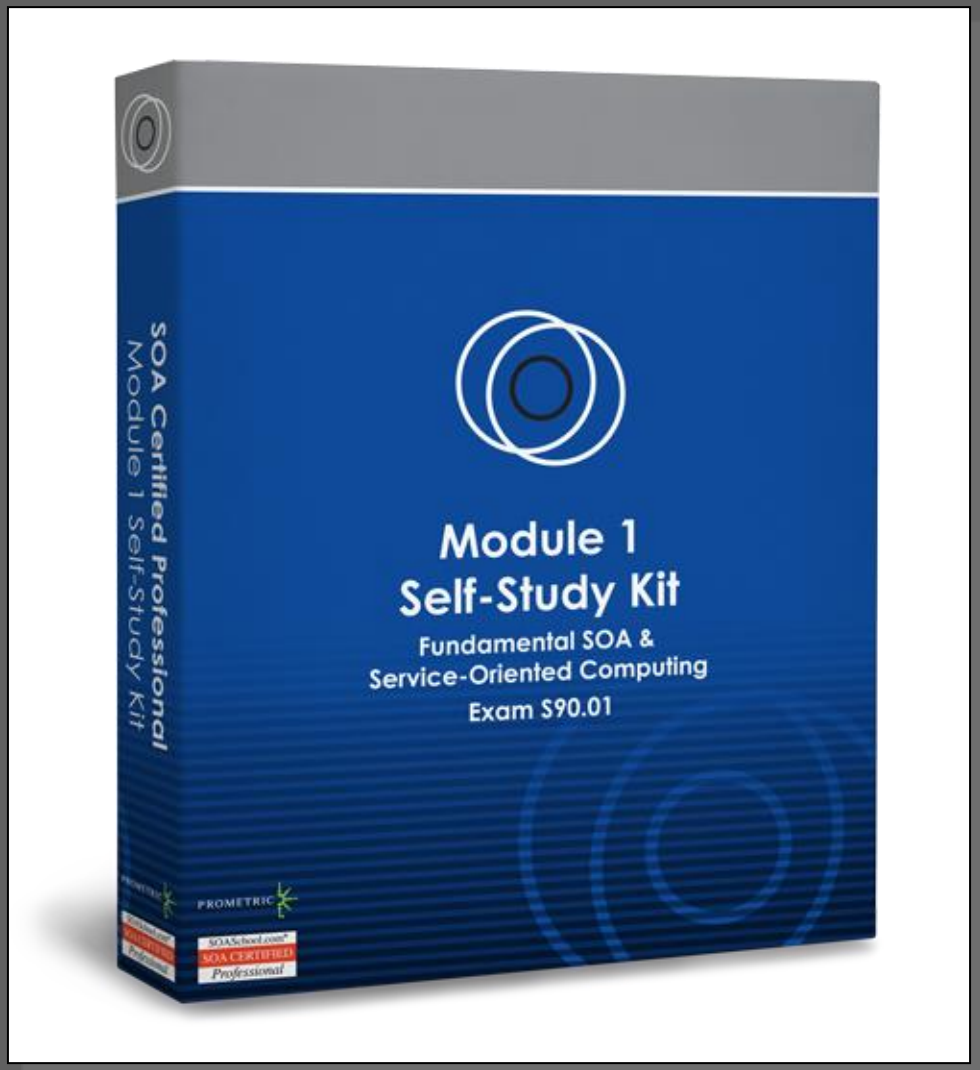
Module 1 Self-Study Kit

An official SOACP Self-Study Kit is available for this module, providing additional study aids and resources, including a separate self-study guide, audio tutor CDs, mind map poster, and flash cards.

If you received this course booklet as part of an instructor-led workshop you may be entitled to a 50% discount on the purchase of this self-study kit.

Note that versions of this self-study kit are available with and without a Prometric exam voucher for Exam S90.01.

For more information, visit:
www.soaselfstudy.com



Service Technology Magazine

The Service Technology Magazine is a monthly online journal published by Arcitura Education and dedicated to providing specialized articles, case studies, and papers by industry experts and practitioners.

SOA professionals are welcome to contribute.

For more information, visit:
www.servicetechmag.com

AITCP Community

Join the growing international Arcitura IT Certified Professional (AITCP) community by connecting with official social media channels available via LinkedIn, Twitter, Facebook and YouTube.

Social media and community links are accessible at:

- www.arcitura.com/community
- www.servicetechbooks.com/community

Contact and Resources

Arcitura Education	www.arcitura.com
SOA School	www.soaschool.com
Workshops	www.soaworkshops.com
Self-Study Kits	www.soaselfstudy.com
Service Technology Magazine	www.servicetechmag.com
Service Technology Standards	www.servicetechspecs.com
SOA Basics	www.serviceorientation.com
SOA Patterns	www.soapatterns.org
General Inquiries	info@soaschool.com
Automatic Updates	notify@arcitura.com
Becoming a Trainer/Partner	info@arcitura.com

Big Data Science Certified Professional Program

Big Data Science School
CERTIFIED BIG DATA
Professional

The Big Data Science Certified Professional (BDSCP) program from Big Data Science School provides a comprehensive vendor-neutral curriculum of 15 course modules and exams for a series of industry certifications dedicated to areas of specialization in the fields of Big Data science.

PROMETRIC

www.bigdatascienceschool.com • www.bigdataselfstudy.com • www.bigdataworkshops.com

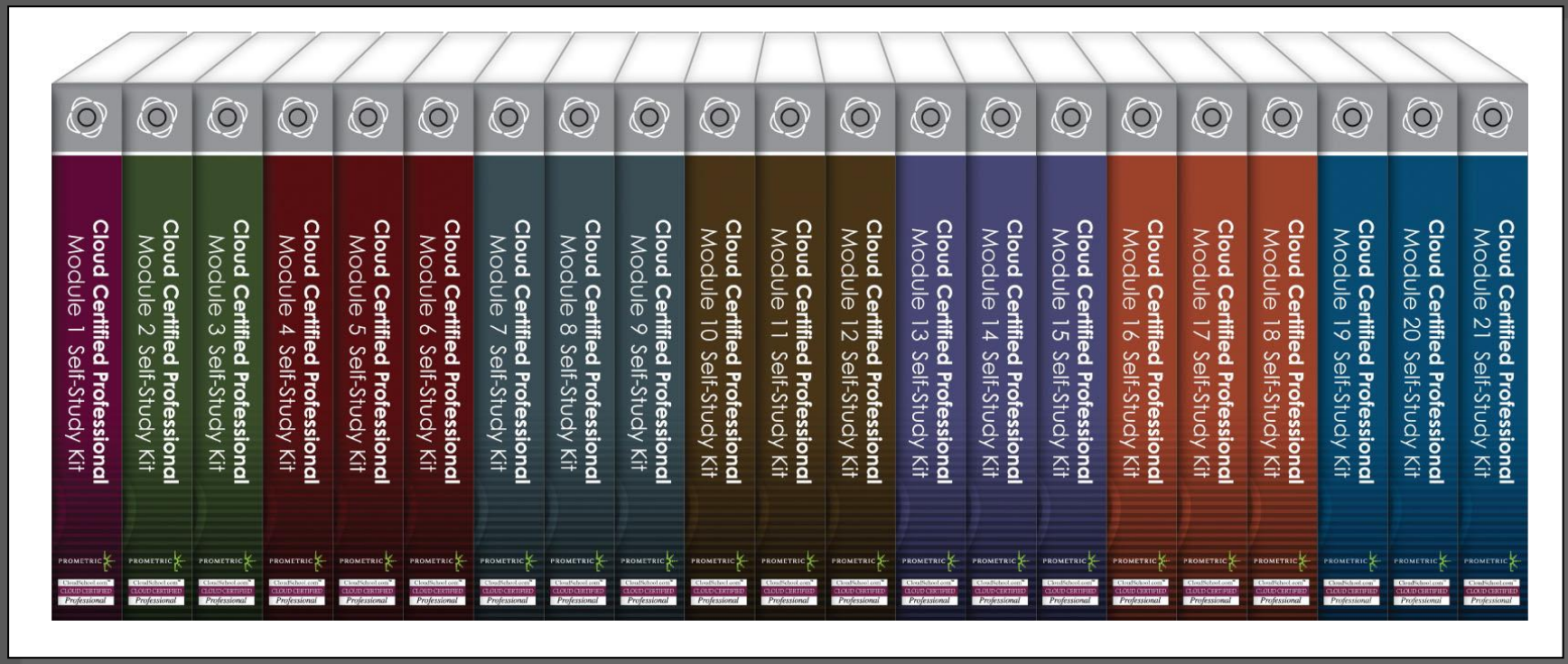
Cloud Certified Professional Program

CloudSchool.com™
CLOUD CERTIFIED
Professional

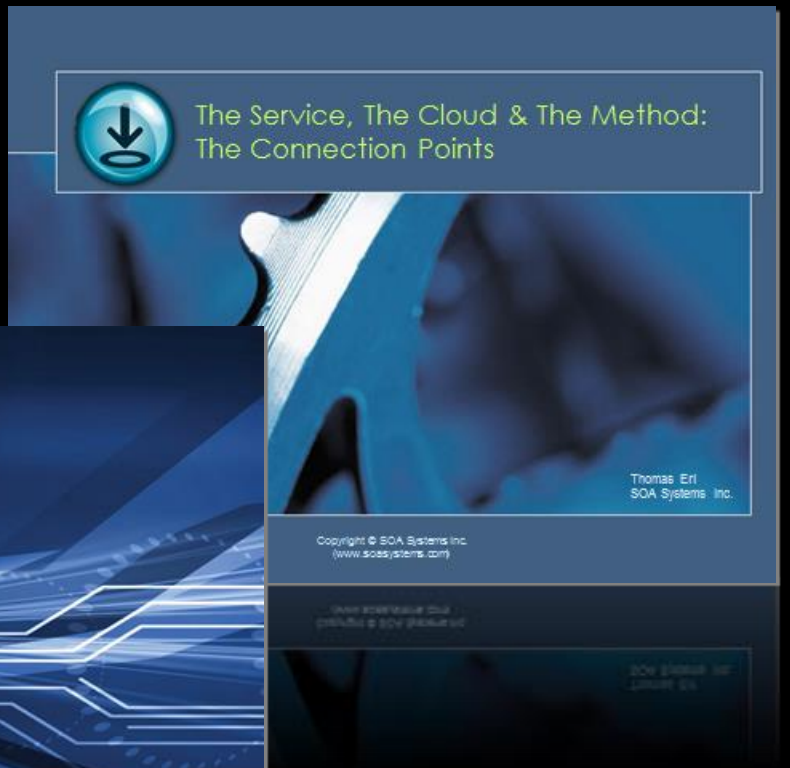
The Cloud Certified Professional (CCP) program from CloudSchool.com provides a comprehensive vendor-neutral curriculum of 21 course modules and exams for a series of industry certifications dedicated to areas of specialization in the field of cloud computing.

PROMETRIC

www.cloudschool.com • www.cloudselfstudy.com • www.cloudworkshops.com



Learn more about SOA and Cloud Computing at the Arcitura YouTube Channel
www.youtube/arcitura



Arcitura Education Inc. is a leading global provider of progressive, vendor-neutral training and certification programs. With a worldwide network of certified instructors, licensed training partners, and testing centers, Arcitura schools and accreditation programs have become internationally established.

If you don't already have an online AITCP profile, visit ww.arcitura.com/aitcp for more information.

Arcitura

the IT education company

www.arcitura.com