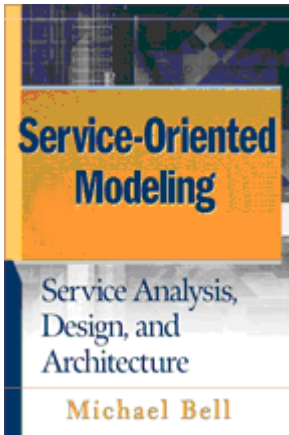


# Streamlining Organizational Software Assets

By Employing Service-Oriented Modeling Practices



### Guiding Questions

- What are the Common Organizational Problems that the Service-Oriented Modeling Practice Addresses?
- How can Service-Oriented Modeling Disciplines Help Resolving Organizational Concerns?
- What is the Proposed Roadmap and Process that Offers Proper Remedies to Enterprise Problems?

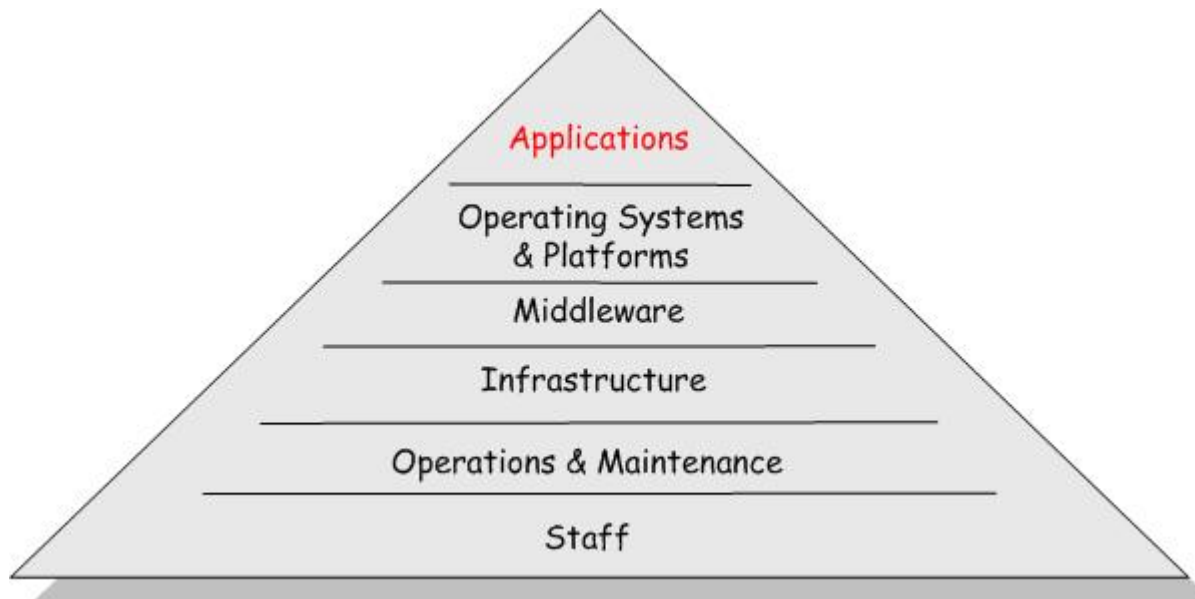
The Service-Oriented Modeling Practice Introduces a Simple and Universal Language for Business and Technology Organizations to Foster:

- Organizational Software Reuse
- Consolidation of Software Assets
- Reduction of IT Expenditure
- Construction of Efficient, Reusable, and Interoperable Software
- Alignment between Business and IT Organizations
- Faster Time-to-Market

## Organizational Expenditure Pyramid

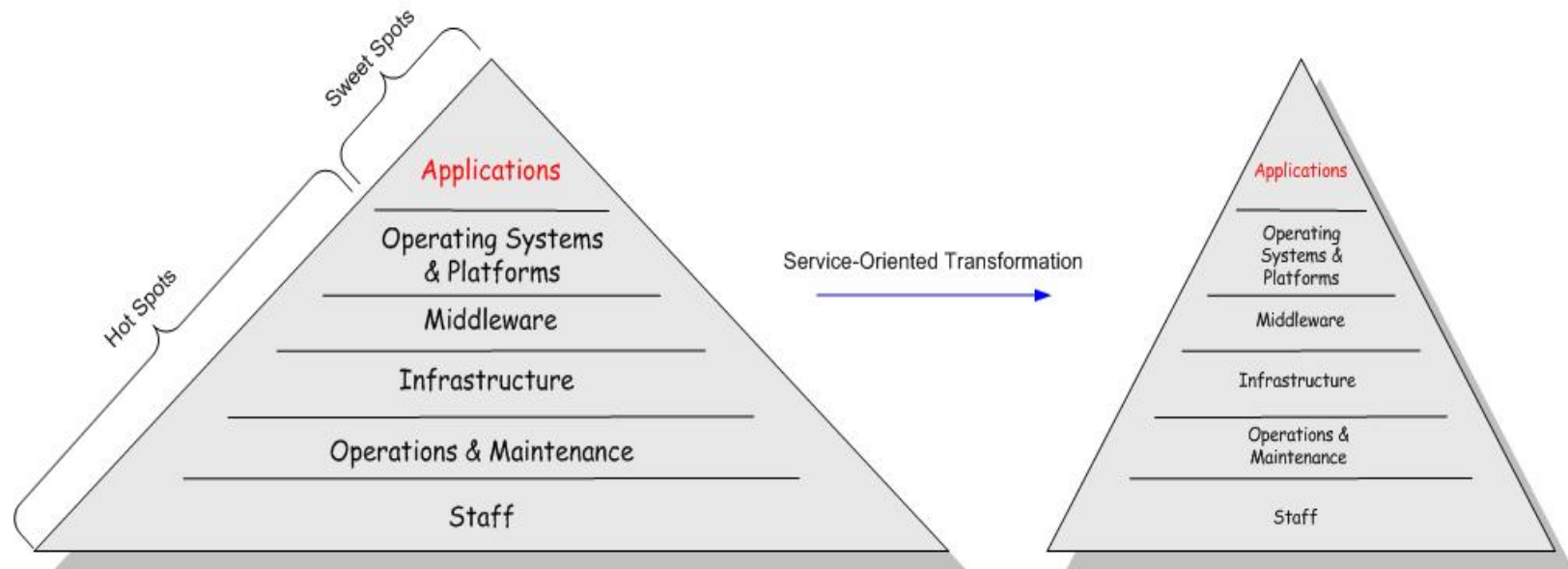
The expenditure pyramid simply identifies a hierarchical dependency between applications and their supporting platforms, infrastructure, and middleware.

Organizations typically allocate the majority of their funding to maintaining and operating the technologies that enable execution of applications in production environments.



Service-oriented modeling disciplines address organizational expenditure reduction by employing **software transformation** mechanisms. These disciplines are applied during construction of new applications or when embarking upon streamlining activities of an existing software legacy environment.

Efficient application-level (sweet spots) modeling activities can immensely reduce the investment in hot spot entities and staff activities.



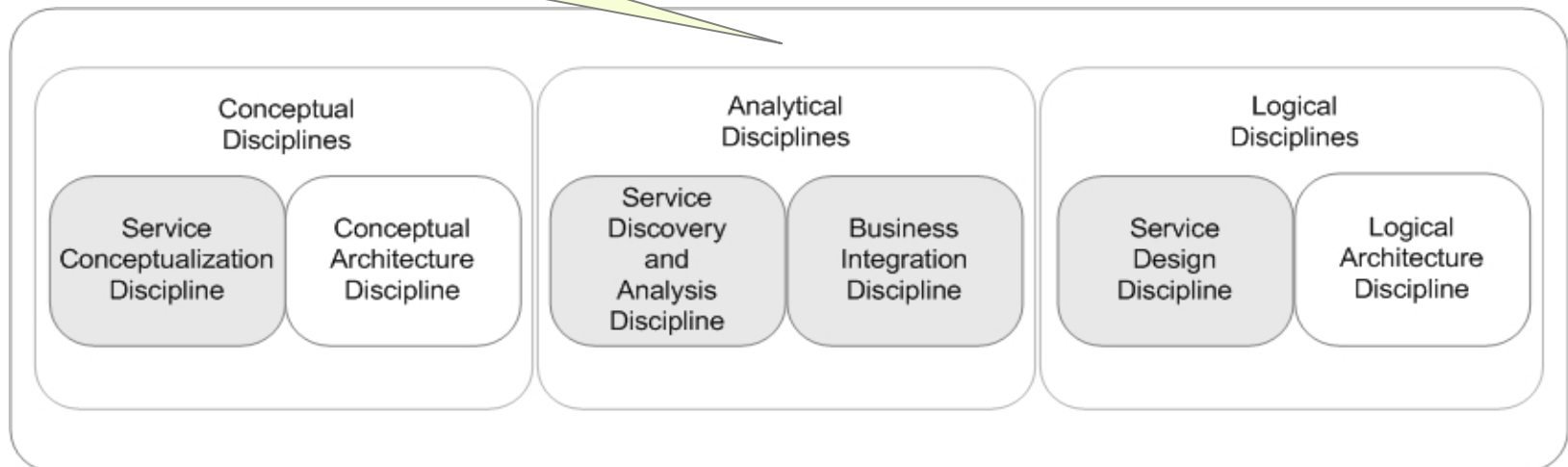
# Service-Oriented Modeling Disciplines Offer Software Life Cycle Management Solutions

The service-oriented modeling disciplines offers three phases to manage a software life cycle, each of which employs corresponding life cycle disciplines:

- 1. Conceptual Phase** – Studying the problems and devising solutions.
- 2. Analytical Phase** – Analyzing and verifying the proposed solutions.
- 3. Logical Phase** – Crafting physical solutions.

Software Life Cycle Phases

Software Life Cycle Disciplines



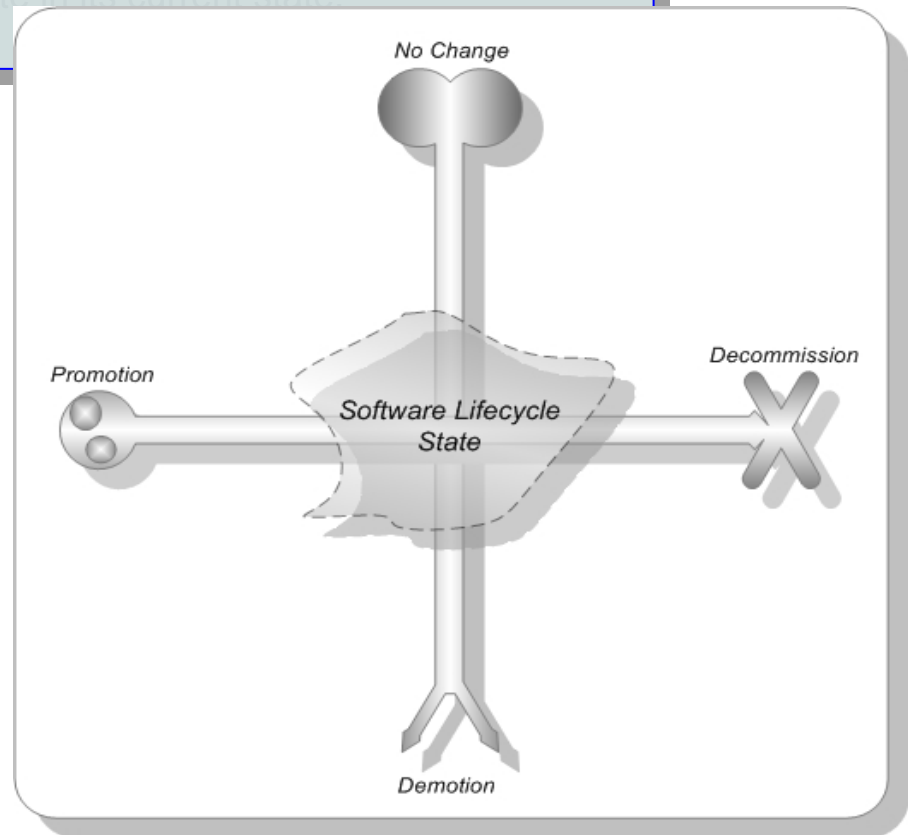
## Service-Oriented Modeling Disciplines Facilitate the Management of Software Lifecycle States

**Promotion** – increasing business functionality, funding, and reuse.

**Decommission** – retiring and terminating execution.

**Demotion** – reducing functionality and reuse, and limiting budget allocation.

**No Change** – Software continues to operate in its current state.



## All Organizational Software Assets are Services

To achieve maximum software reuse and enable effective application consolidation across an organization, the service-oriented modeling process conceives all organizational software assets as Services.

So, what are services? Services are software products, applications, and other artifacts that an organization has been acquiring, analyzing, designing, architecting, constructing, and maintaining for decades—these are legacy applications (Mainframe, J2EE, or .NET), concepts, ideas, business processes and business functionalities, middleware products, language platforms, and even software components and libraries.

## What Does it Mean?

**If you streamlining your legacy environment:** break down your existing legacy applications into services to increase their reuse opportunities across your organization, and encourage consolidation of software assets. A service can constitute a business process, a functionality, or just an idea.

**If you building new applications:** construct services. These smaller software units enable loosely-coupled computing environment and foster asset reusability.

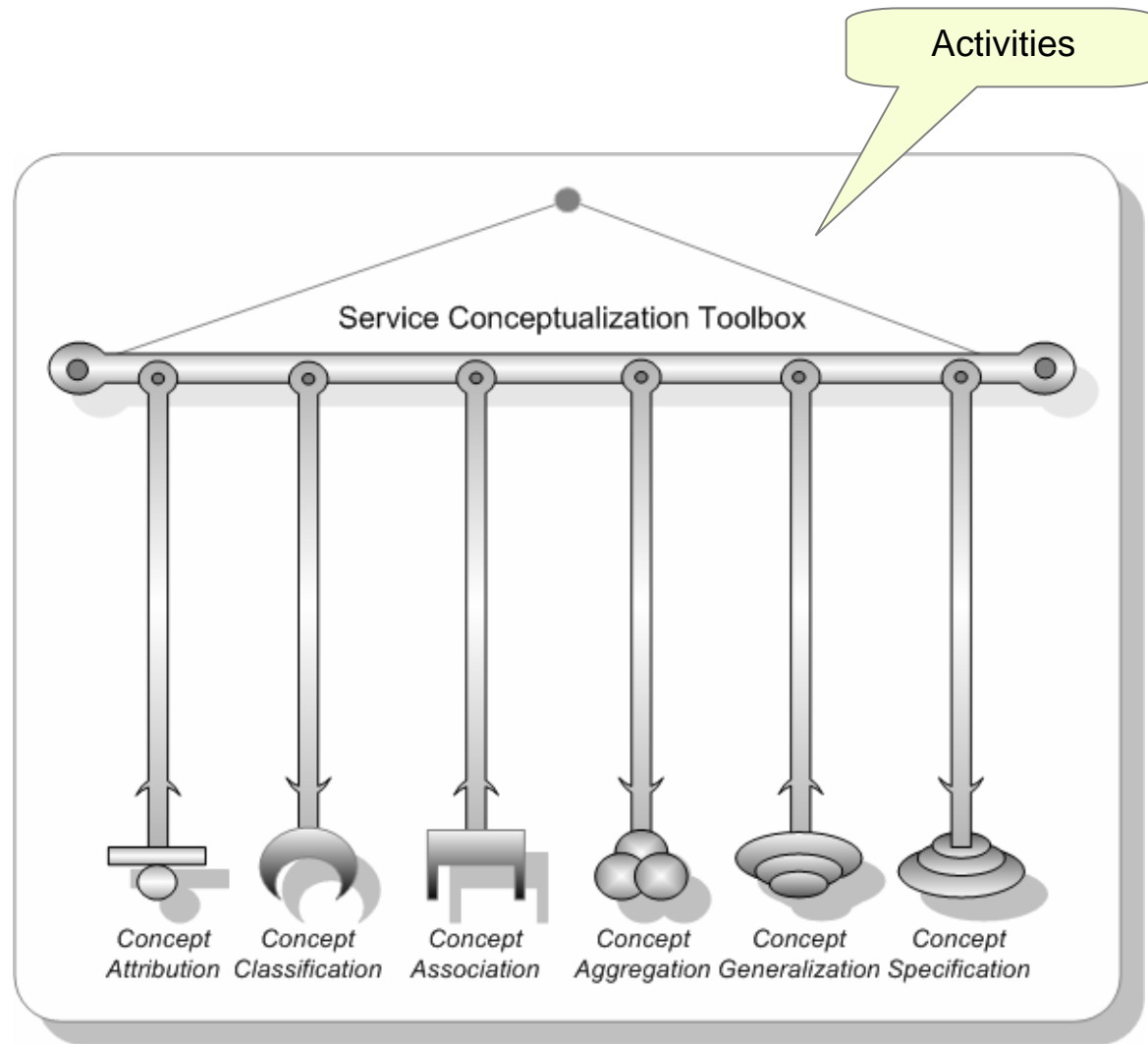


# Conceptual Disciplines: Studying the Problems & Devising Conceptual Solutions!

Activities

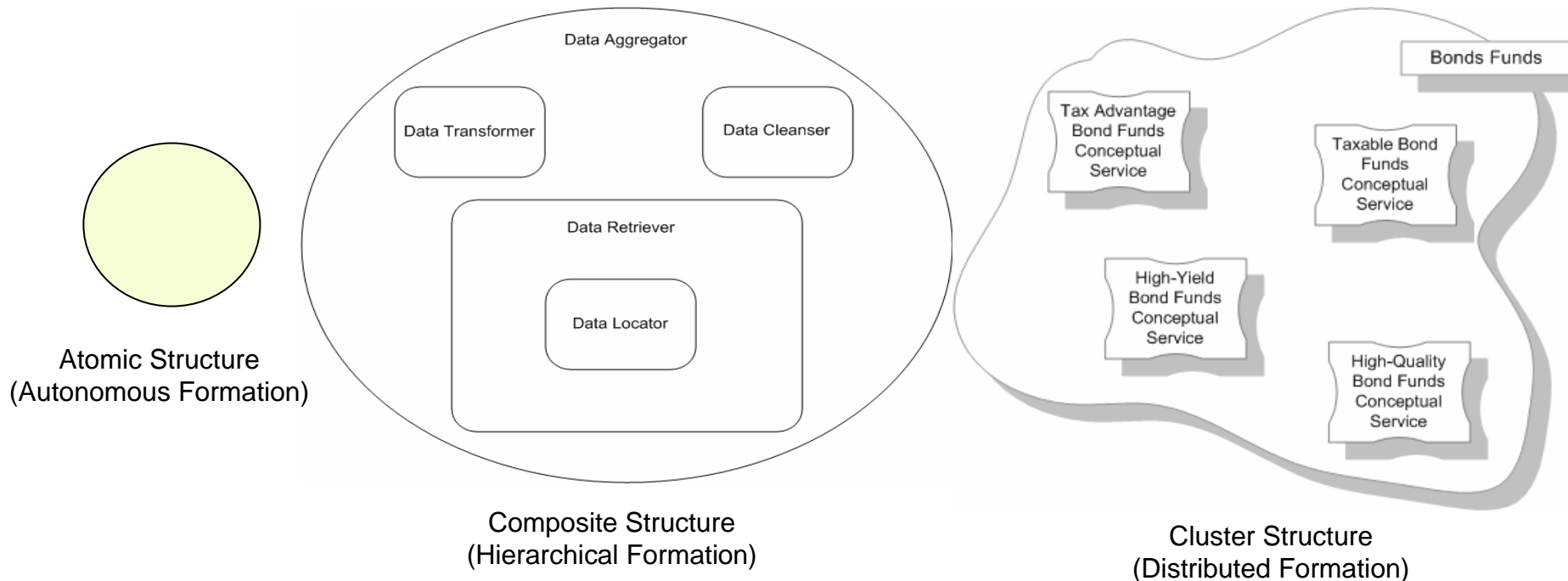
Benefits

- Establishment of **organizational universal concepts** that lead to the development and analysis of software products.
- Second, the establishment of **common language, vocabulary, and taxonomy** that can fill in the communication gaps between business and technology Organizations.
- Institution of a software **asset portfolio**.



## Conceptual Disciplines - Three Major Software Structures

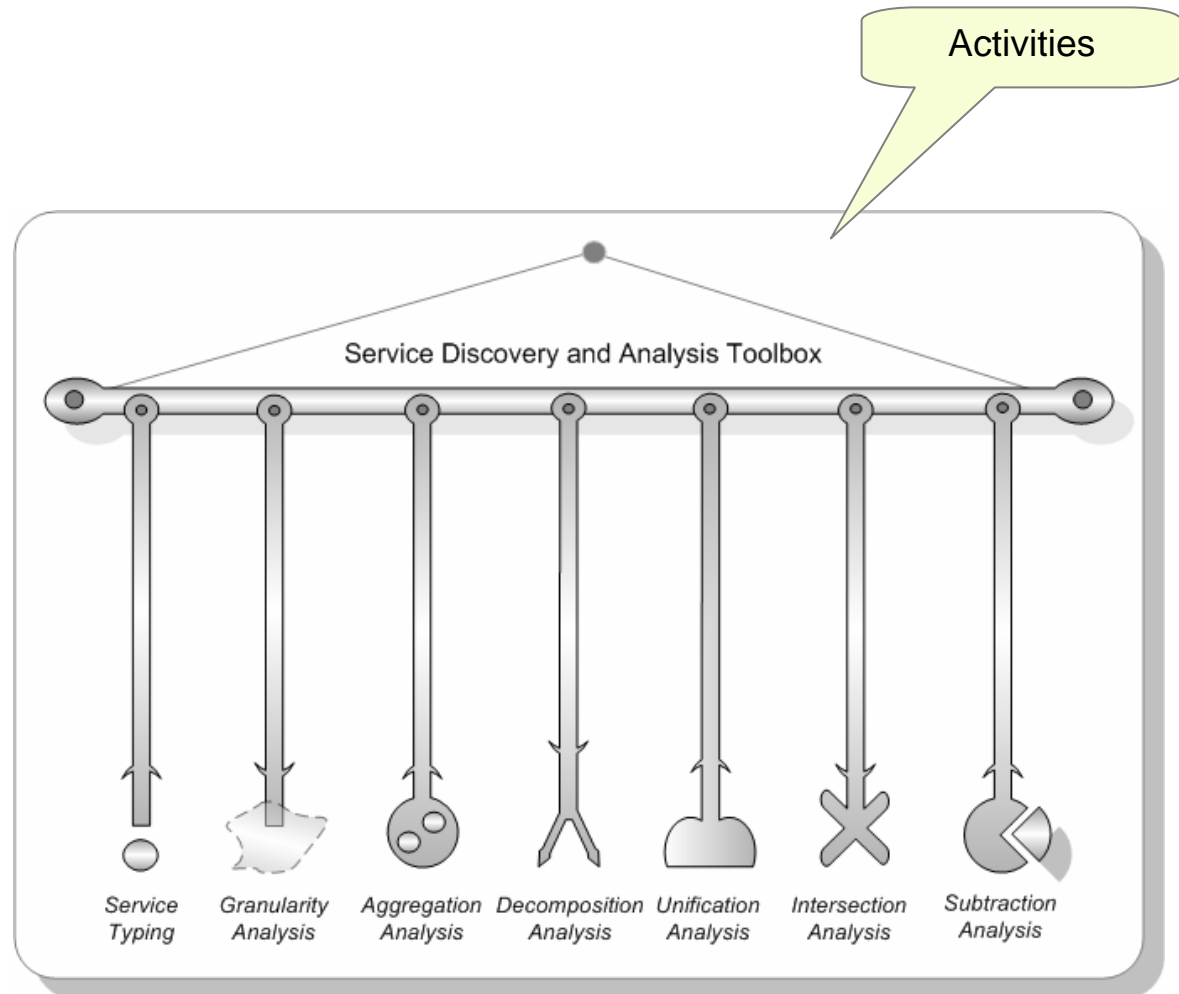
Arranging all organizational software assets in three recognized service structures and establishing their relationship can contribute to software assets reuse, consolidation, and reduction in expenditure. This conceptual process can be applied to either breaking down or aggregating legacy applications, third party products, and even middleware software. The conceptualization process is based on their business functionality and their attributes.



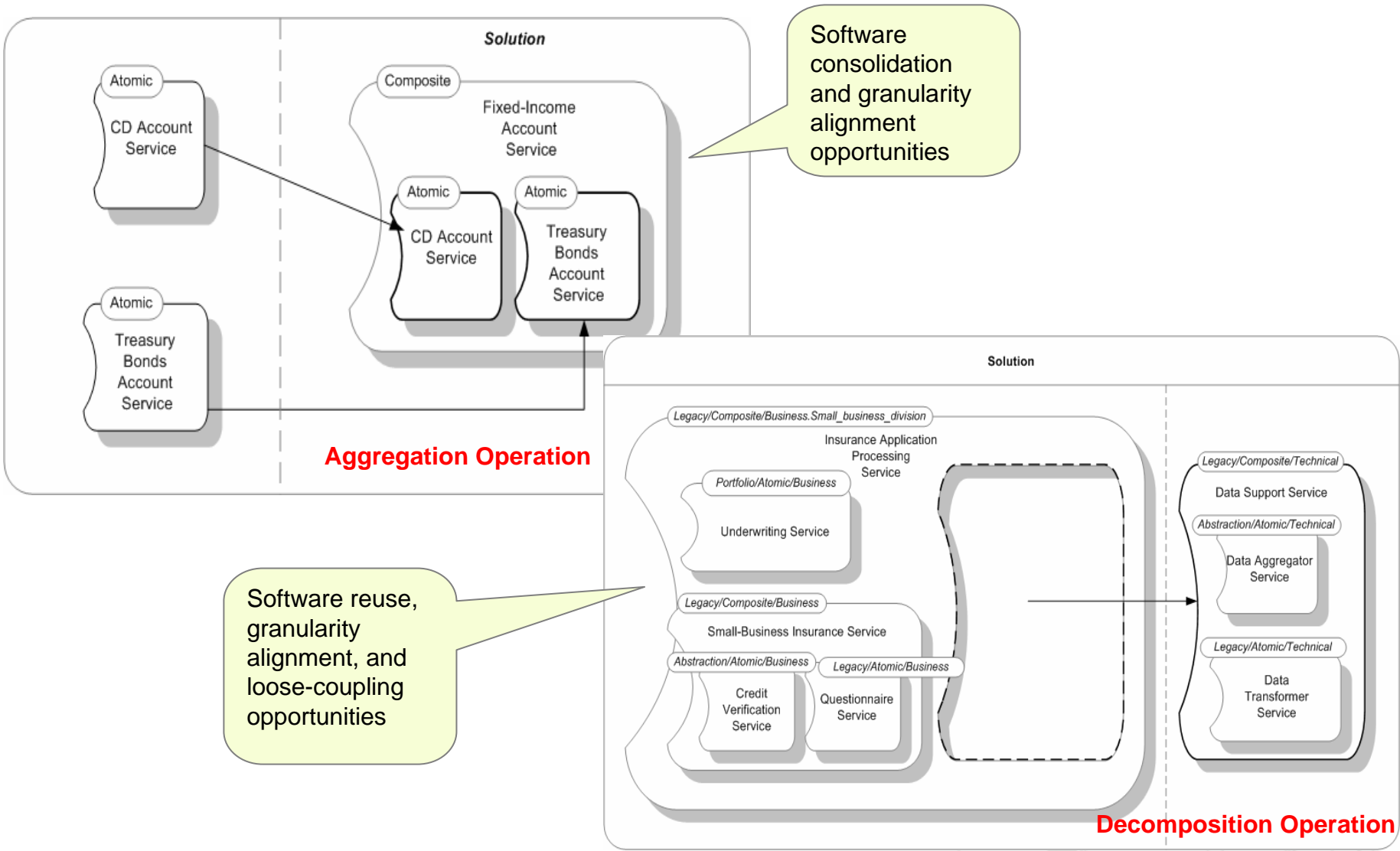
## Benefits

- Analytical disciplines foster software **reusability**
- Encourage application **consolidation**
- Alleviate software **interoperability** challenges
- Encourage **loosely-coupled** computing environment
- Contribute to organizational **expenditure reduction**
- Promote proper **alignment** between business and IT institutions

## Activities



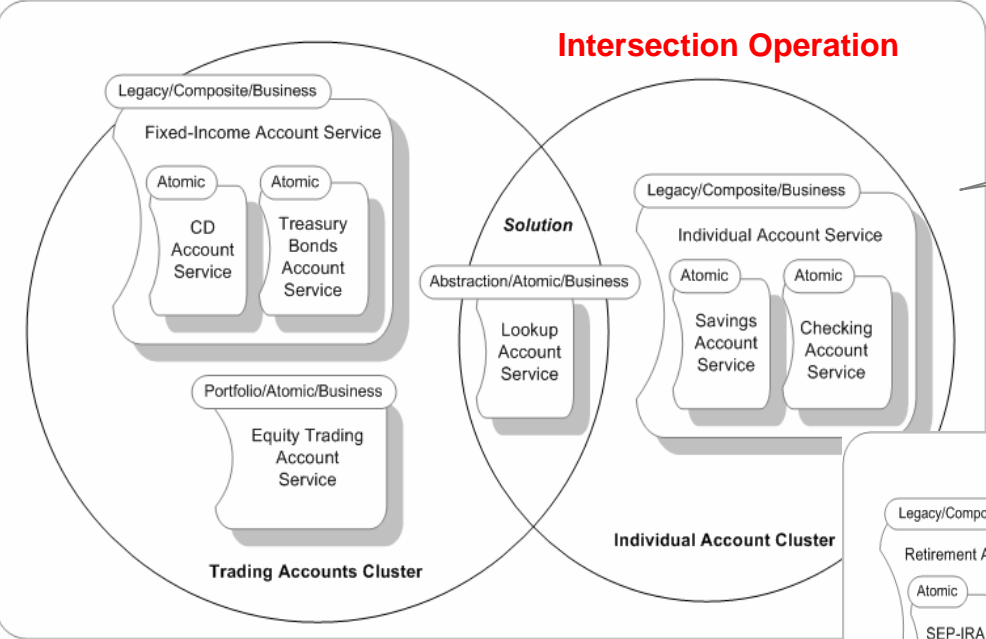
# Analytical Disciplines - Operations on Services I



# Analytical Disciplines - Operations on Services II

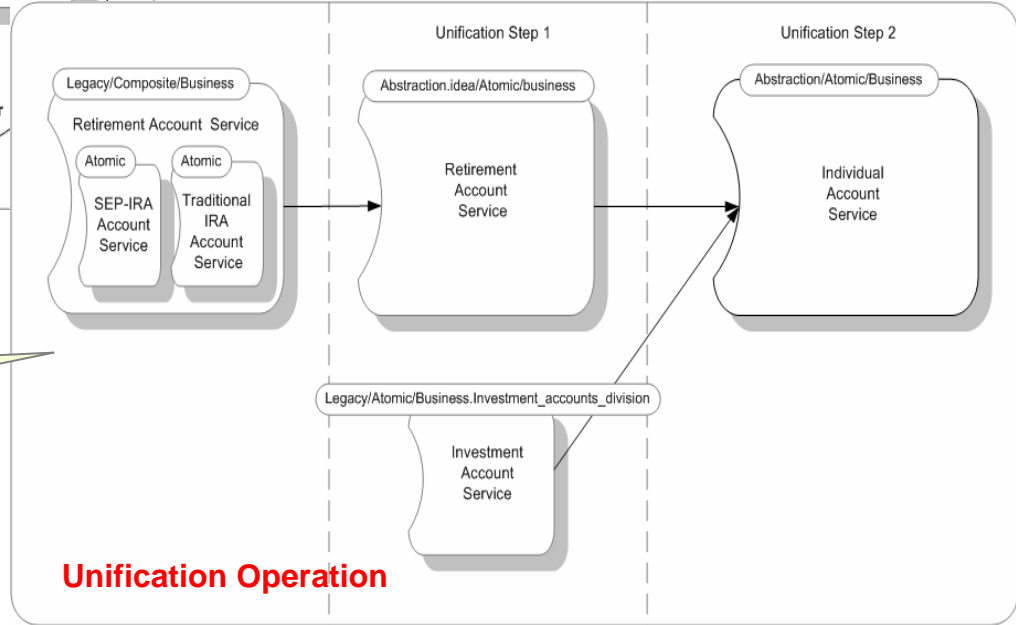
## Intersection Operation

Software reuse, and consolidation opportunities



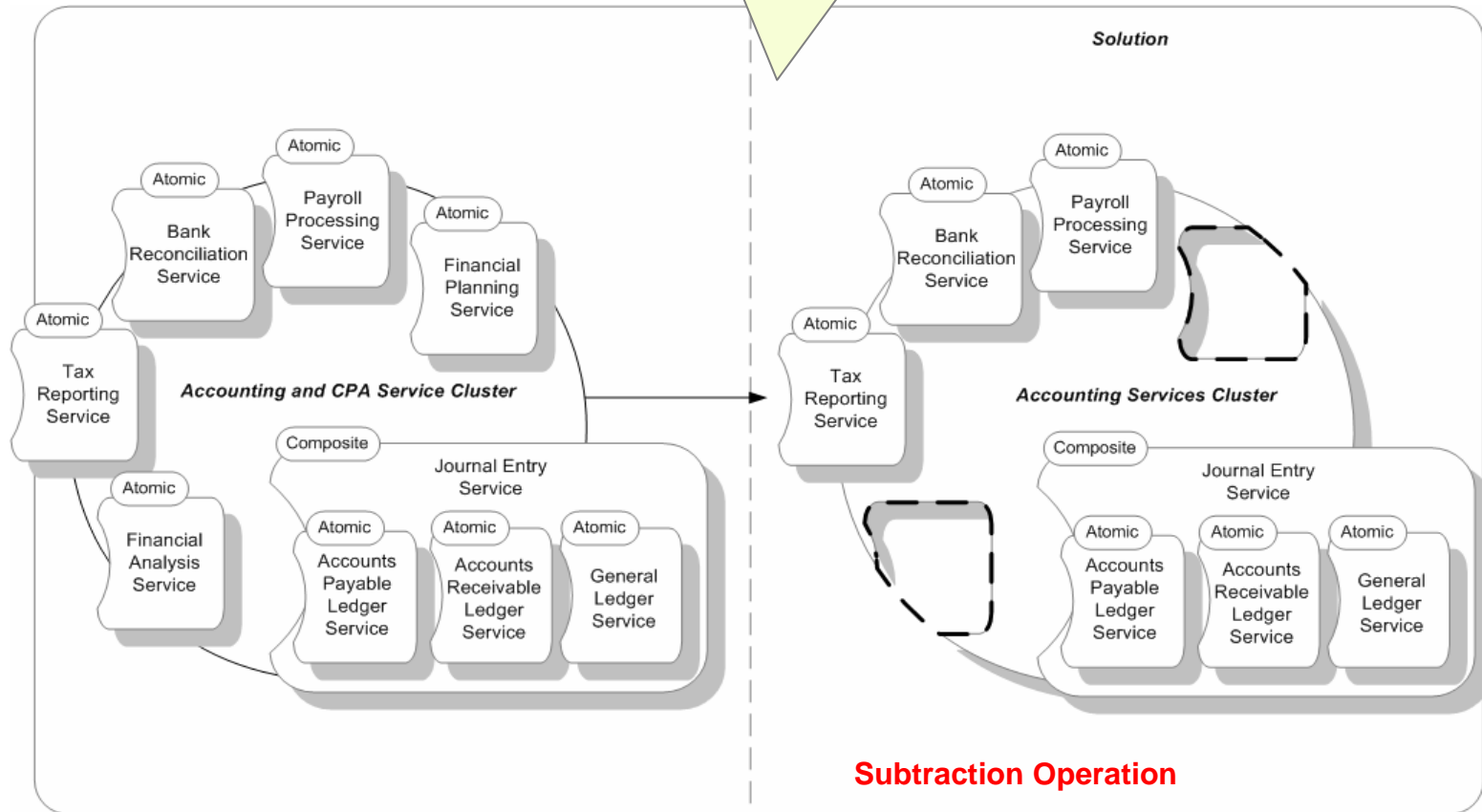
Software consolidation opportunities

## Unification Operation



# Analytical Disciplines - Operations on Services III

Software decommission and functionality reduction opportunities

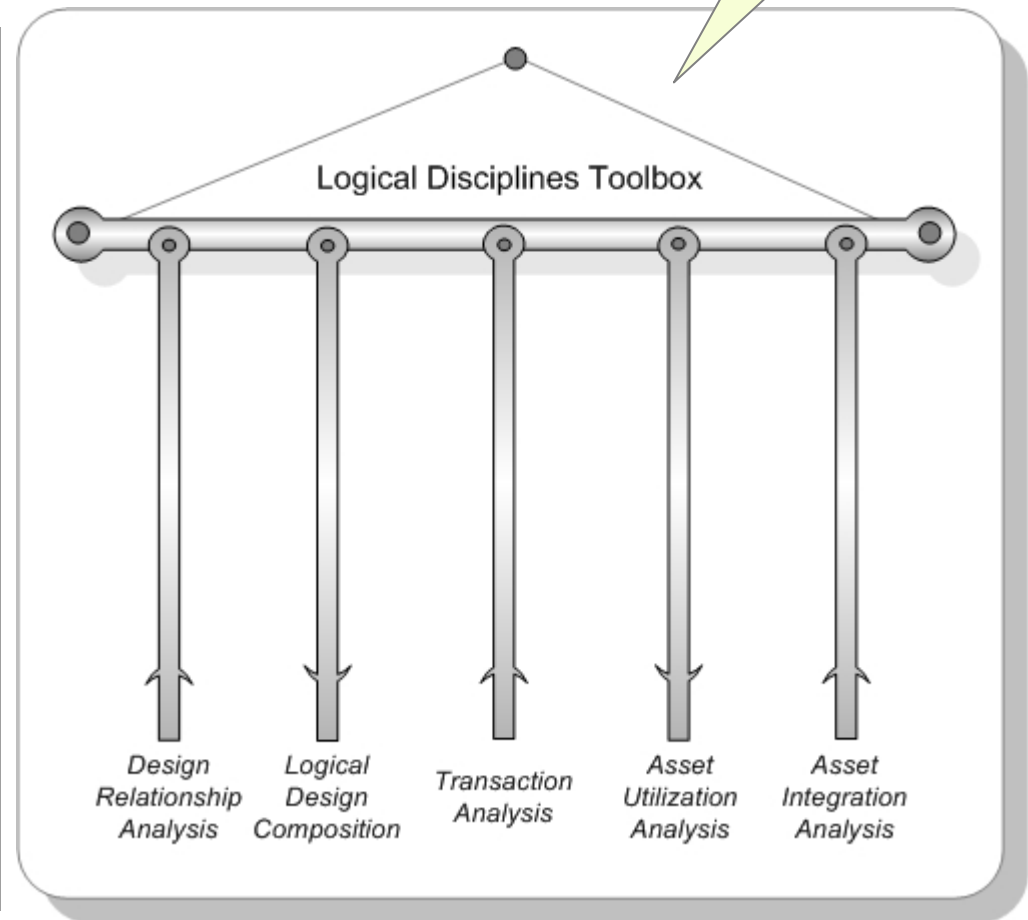


### Benefits

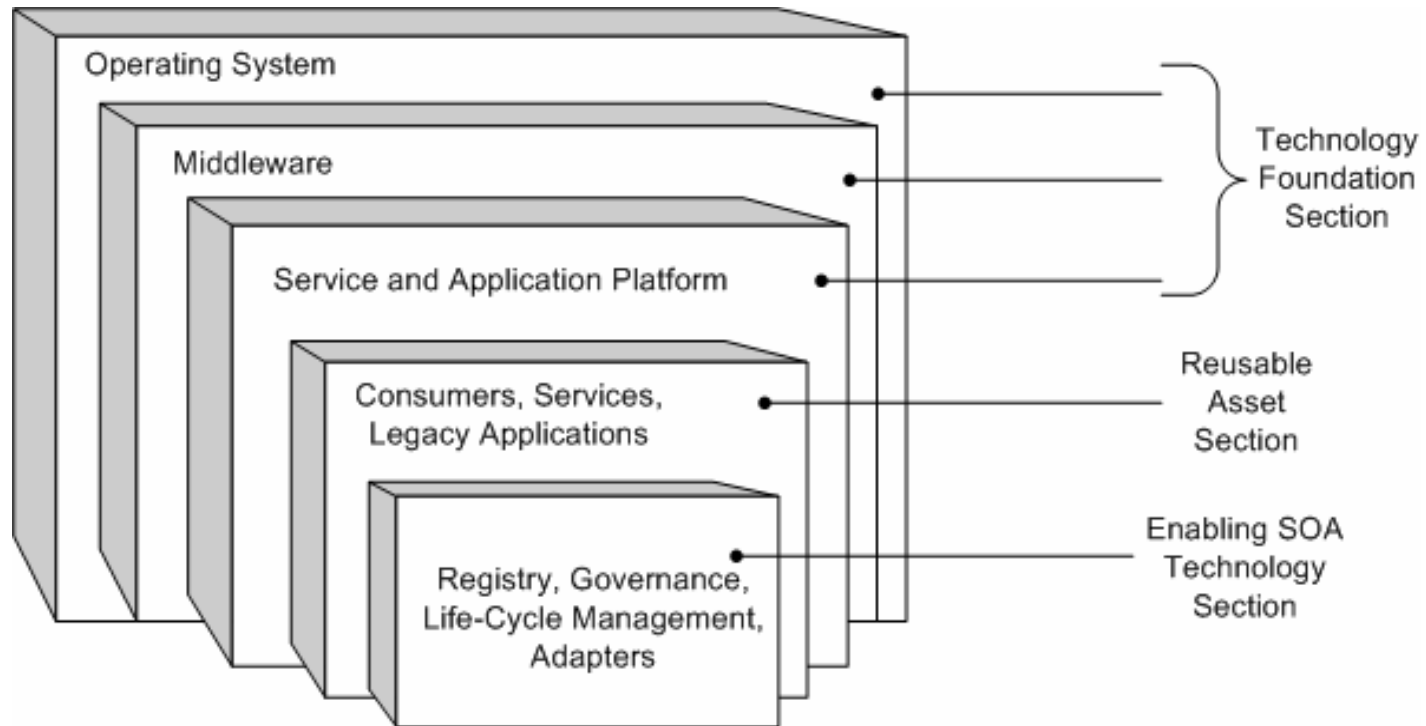
The logical disciplines facilitate the establishment of a service-oriented **ecosystem** that enables software entities to collaborate, exchange messages, and execute transactions. It is all about **integration** of various applications in heterogeneous loosely-coupled production environments.

Logical disciplines help resolve software **integration, consumption, and utilization of software** challenges. They facilitate **streamlining** activities of production environments, and help with the **consolidation** of business functionality and software assets. Furthermore, these disciplines contribute to the **collaboration** and **reuse of deployed** software assets by utilizing service-oriented products.

### Activities



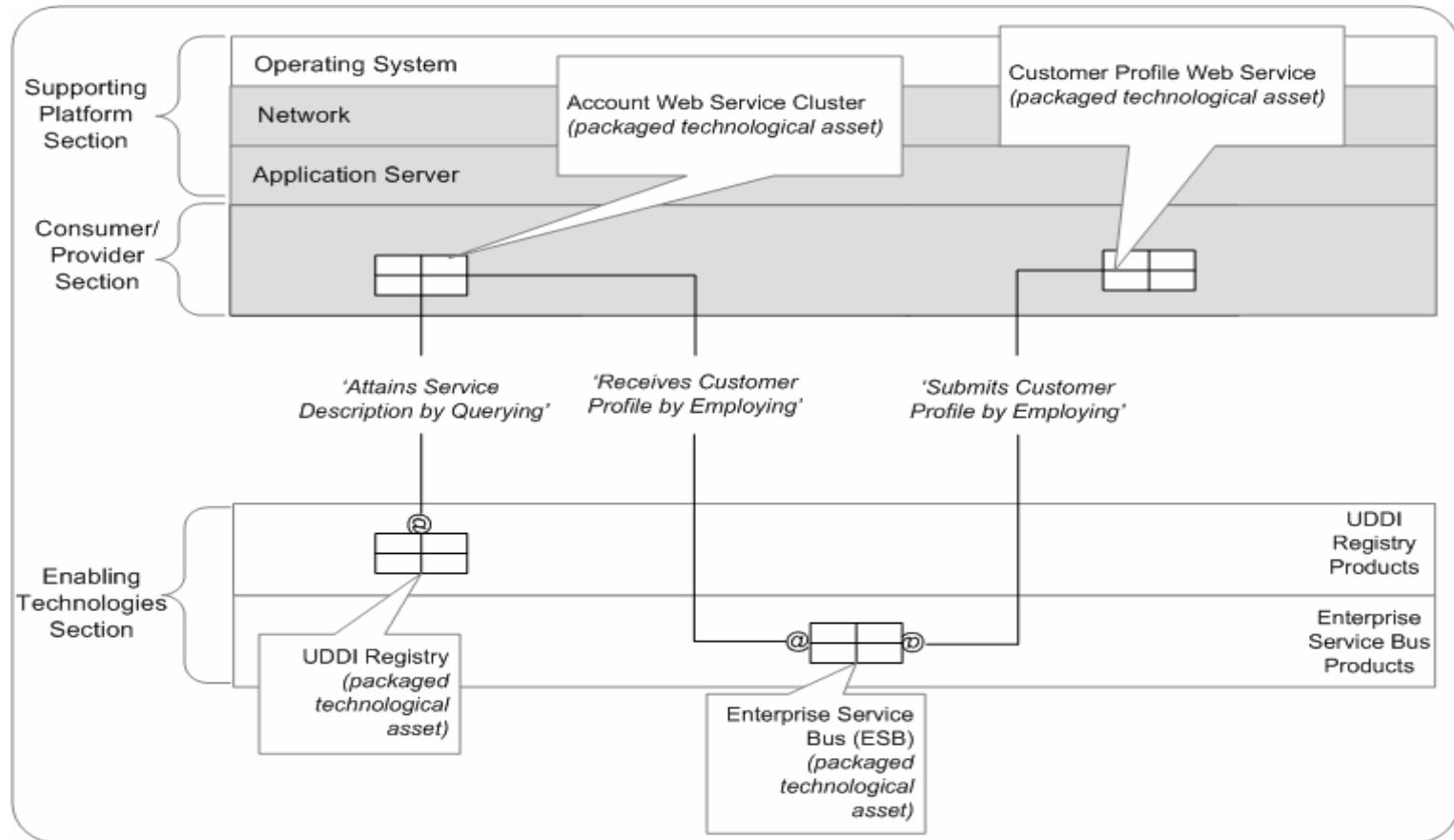
Logical disciplines facilitate the **integration** of software in a modularized formation. We typically arrange our applications and their supporting technologies in a hierarchical structure to encourage **reuse** and **expenditure reduction** of resources.





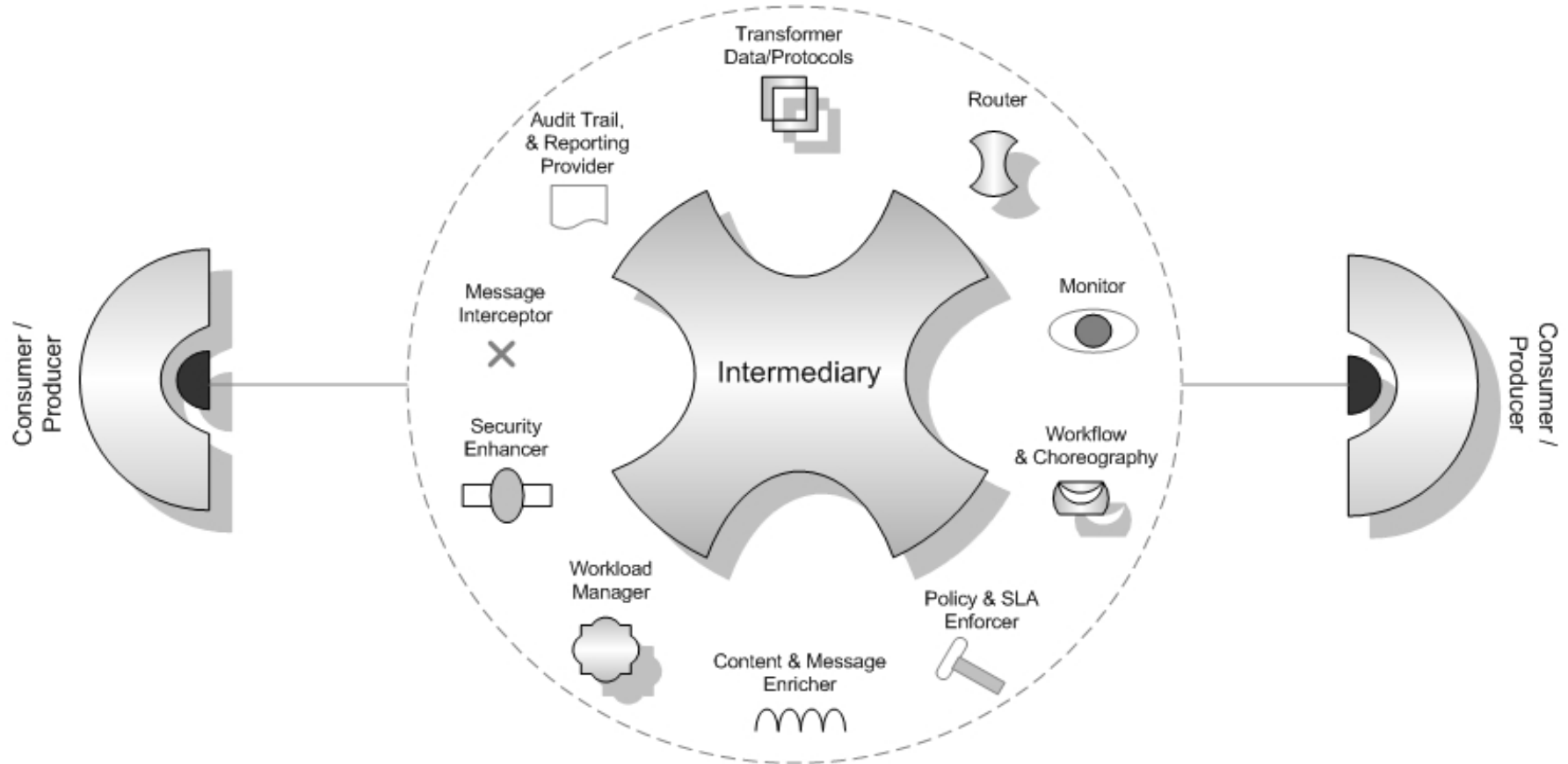
## Logical Disciplines: The Ecosystem - Asset Utilization Diagram

The logical view of an organization can be illustrated by an asset utilization diagram that depicts a tangible solution. It typically illustrates a deployment environment that employs enabling products and message exchange routes.



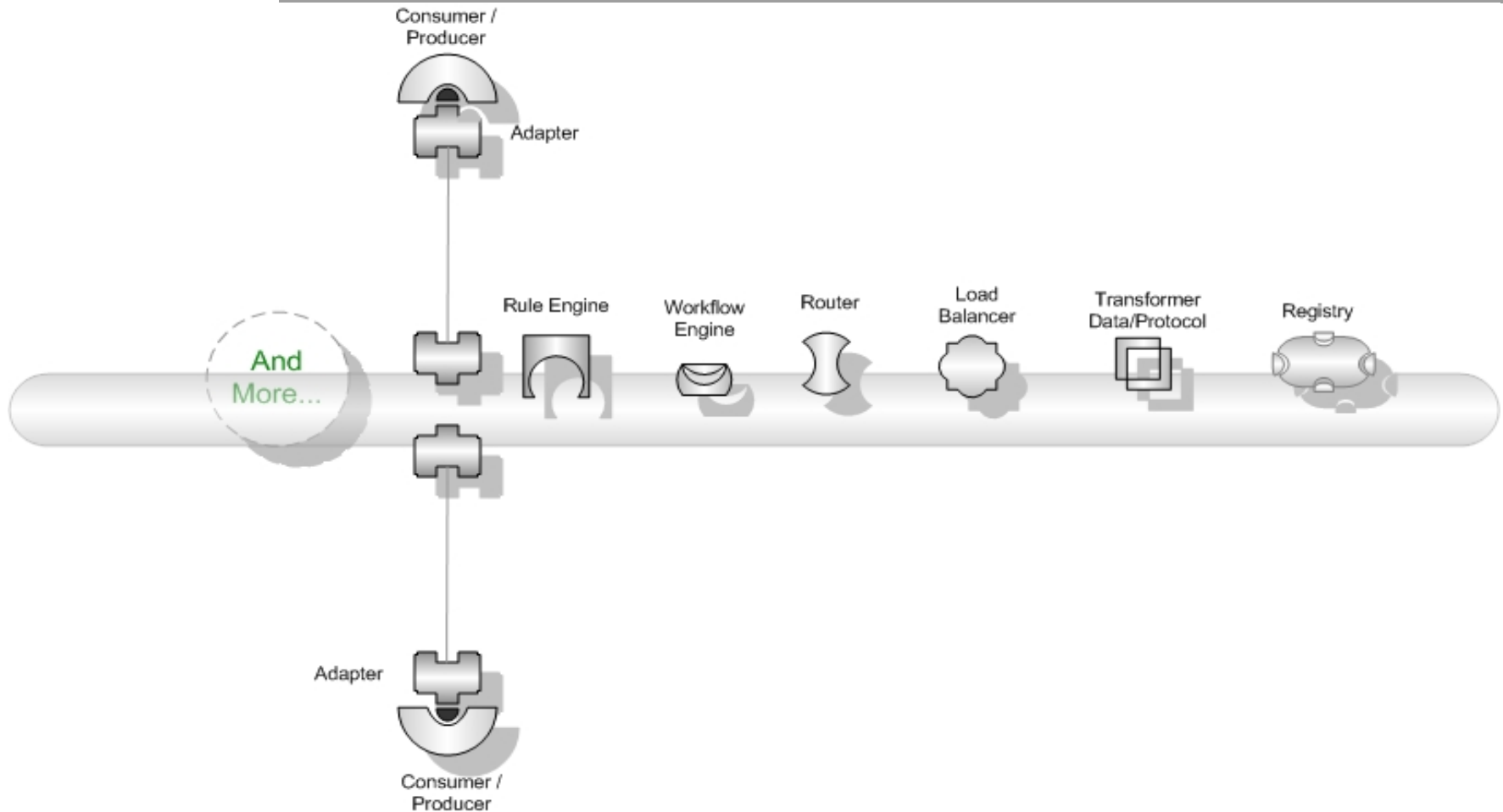
## Logical Disciplines: Enabling Technologies are the Key to Integration Success!

Software intermediaries offer reuse opportunities of disbursed applications. Logical solutions often utilize intermediary middleware products to bridge the gap between services across an organization. They provide data transformation mechanisms, security enhancement, and even monitoring capabilities.



## Logical Disciplines: Enterprise Service Bus - Software Reuse and Interoperability Solution.

An enterprise service bus is another middleware product that not only connects disbursed applications but also enable reuse of software functionality across an organization.



## A Roadmap to Success

To provide efficient solutions devise three major steps: identification, solution proposition, and physical solution. Employ the top-down and the bottom-up approaches and iterate until satisfactory results have been achieved.

