TDWI Data Integration Principles and Practices

Creating Information Unity from Data Disparity

Previews of TDWI course books offer an opportunity to see the quality of our material and help you to select the courses that best fit your needs. The previews cannot be printed.

TDWI strives to provide course books that are content-rich and that serve as useful reference documents after a class has ended.

This preview shows selected pages that are representative of the entire course book; pages are not consecutive. The page numbers shown at the bottom of each page indicate their actual position in the course book. All table-of-contents pages are included to illustrate all of the topics covered by the course.

TABLE OF CONTENTS

COURSE OBJECTIVES

To learn:

- ✓ ***The role, purpose, and issues of data integration strategy***
- ✓ ***Integration patterns and a framework for data integration architecture***
- ✓ ***How to fit unstructured data into integration strategy, architecture, and systems***
- ✓ ***How to use integration architecture and patterns to handle large volume data challenges***
- ✓ ***How to apply architecture and patterns for enterprise, departmental, and local data***
- ✓ ***How to select, mix-and-match, and apply several data integration methods including ETL, federated, service-oriented, and virtualized***
- ✓ ***Techniques to collect and manage data integration requirements***
- ✓ ***Tips and techniques for success throughout the data integration lifecycle—strategy, architecture, systems development, and operations***

Module 1

Data Integration Strategy

What Is Data Integration Strategy?

Strategy Defined

What Is Data Integration Strategy?

Strategy Defined

UNDERSTANDING STRATEGY

Before defining data integration strategy, we need to understand the meaning of the word strategy. Consider this quote from *Strategy Bites Back:* [1]

> *"Strategy can be awfully boring. The consultants can be straighter than we academics, not to mention the planners. Everybody is so serious. If that gets us better strategies, fine. But it often gets us worse ones—standard, generic, uninspiring. Strategy doesn't only have to position, it also has to inspire. So an uninspiring strategy is really no strategy at all."*

Strategy, as described by the definitions on the facing page, is a plan for the future. Good strategy must inspire—inspiration built on a compelling plan for a future that is interesting and desirable.

STRATEGY IN CONTEXT

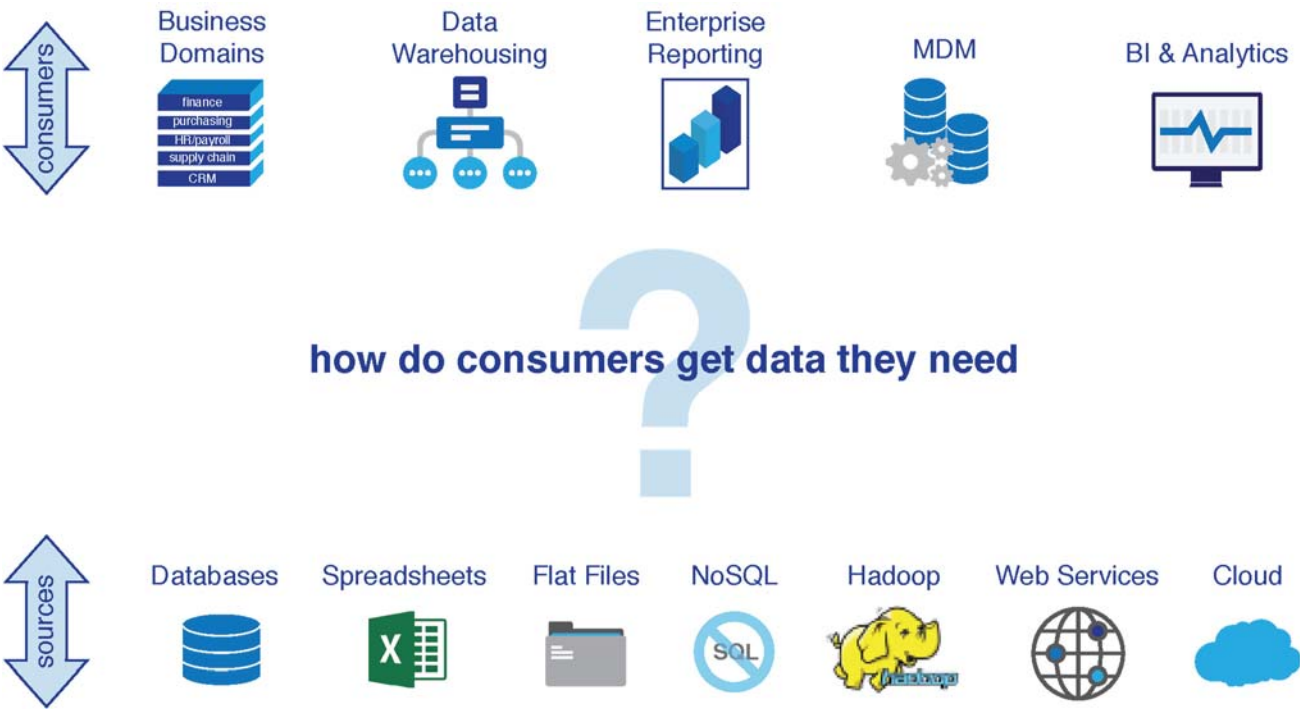
Strategy doesn't work in isolation. It is misdirected unless designed to realize a vision that is driven by mission and values. Furthermore, it is pointless without execution that drives projects and meets objectives.

- It all begins with mission and values. Mission is your purpose as an organization and values are the principles that guide organizational behavior and shape organizational culture.
- Mission and values drive vision—your future view of the organization.
- Strategy describes the things that you'll do to make the vision become reality.
- Strategic objectives are the things that the strategy is supposed to accomplish. They should be tangible and measurable things—not abstract, fuzzy, conceptual, or ambiguous.
- Strategic initiatives are sponsored efforts to achieve strategic objectives and they are a primary way that you connect strategy with tactics and day-to-day operations. Strategic initiatives also may be referred to as portfolio projects.

[1] *Strategy Bites Back: It Is Far More, and Less, than You Ever Imagined*, Mintzberg, Ahlstrand, and Lampei

Elements of Data Integration Strategy

Data Consumers and Data Sources



Elements of Data Integration Strategy

Data Consumers and Data Sources

CONNECTING BUSINESS WITH DATA

The purpose of data management, including data integration, is to connect business with data. Data sources and data consumers are the endpoints of data integration strategy—the origins and destinations of data flow through the business. Effective data strategy considers how best to organize and execute data flow—to obtain data from sources and deliver it to consumers.

CONSUMERS

Data consumers include the application types illustrated on the facing page, ranging from domain-specific systems to analytics. However, the ultimate consumers are the business activities that are informed by data—strategic, tactical, and operational—and the executives, managers, and staff who perform those activities.

SOURCES

Data sources include the technical components illustrated—the databases containing data of various types. However, the real sources are the business activities where data is created (planning, management, and day-to-day business functions) and the people (planners, managers, and staff) who perform those activities.

Module 2

Data Integration Architecture

Integration Patterns

Methods for Information Sharing

Integration Patterns

Methods for Information Sharing

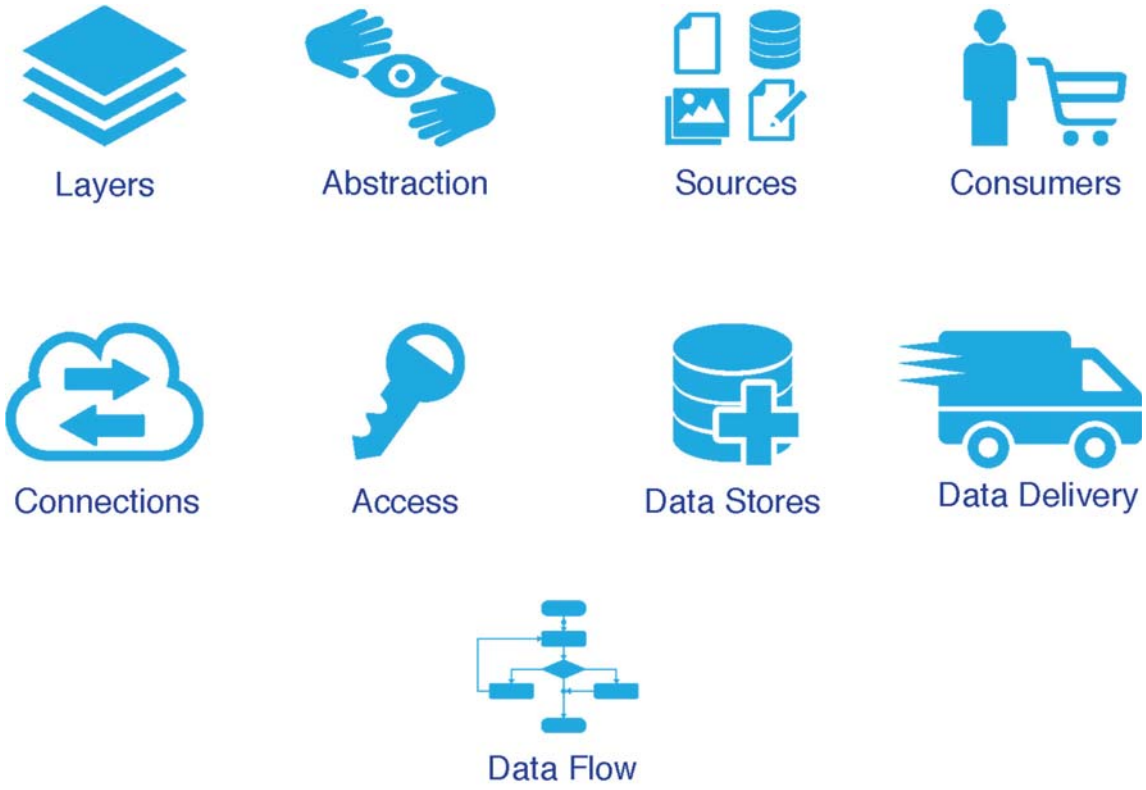
INFORMATION SHARING

One of the primary goals of data integration is to facilitate sharing information across business functions, processes, and organizations. Many different methods of information sharing have been developed through the evolution of information systems. Some of those methods are recognized as integration patterns—either for integration of data or for integration of applications. Common patterns include:

- Shared Databases
- Replication
- Gateways
- Portals
- Entity Aggregation
- Data Services
- Messaging
- File Transfer
- Publish and Subscribe
- Virtualization

Architecture Concepts and Components

Constructs in Data Integration Architecture



Architecture Concepts and Components

Constructs in Data Integration Architecture

COMMON IDEAS

Many vendors and analysts provide reference architectures for data integration. These reference architectures share many common constructs which support the strategy elements previously discussed.

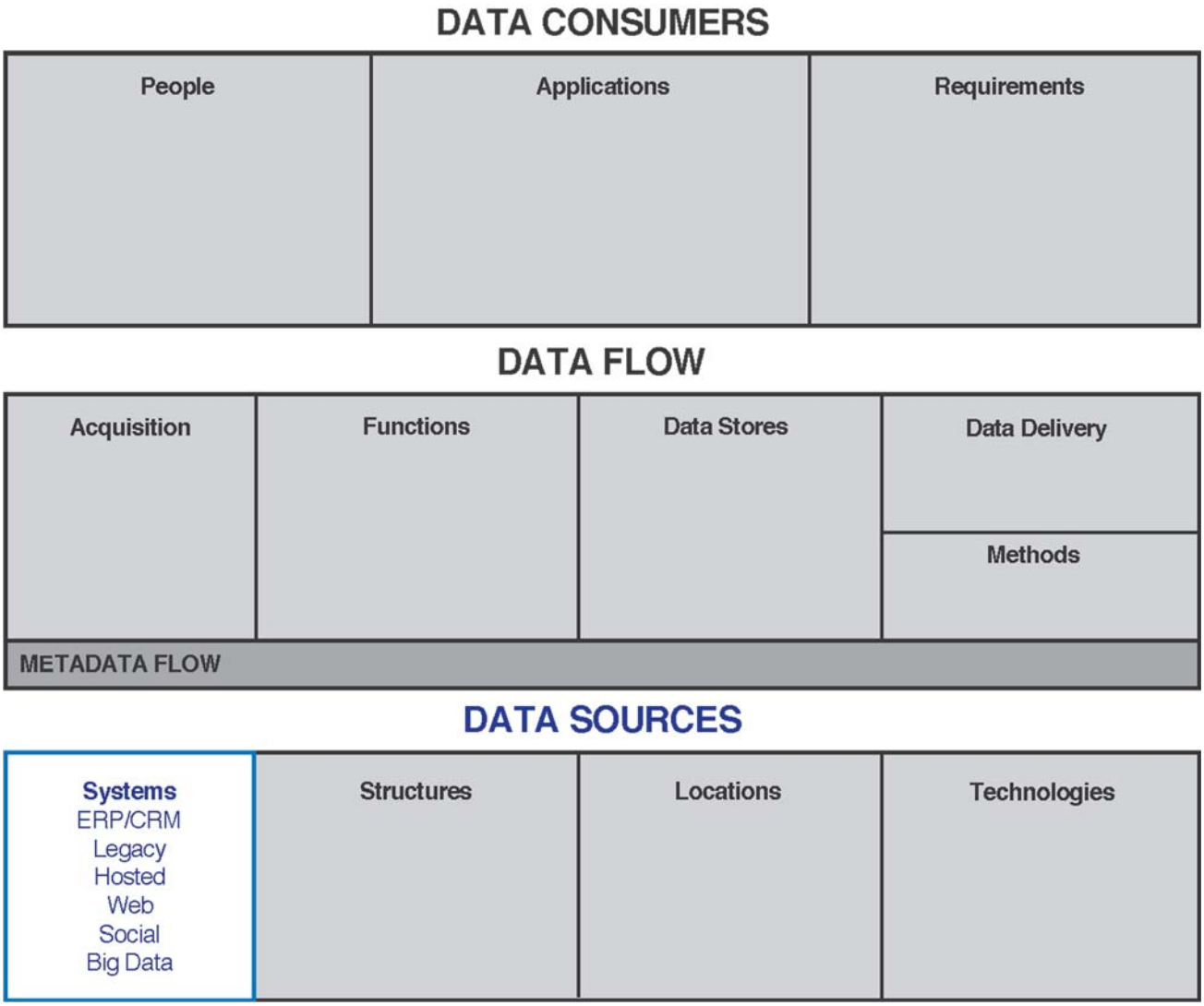
The concepts shown on the facing page appear repeatedly throughout reference architectures, indicating that they are minimally good practices, and perhaps essential components, for every data integration architecture.

Module 3

Creating Data Integration Architecture

Data Sources

Source Systems



Data Sources

Source Systems

SYSTEM PERSPECTIVE

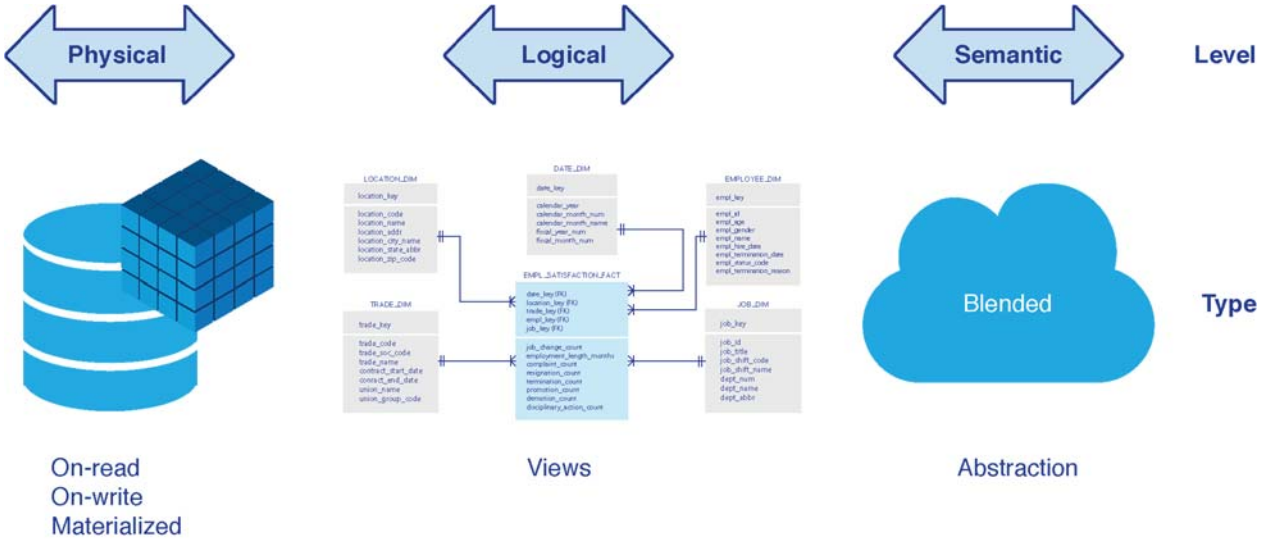
Data sources are clearly one of the critical components of data integration architecture. It is necessary to understand those sources from more than a simple database point of view. The architecture should include components for all of the various types of systems from which you acquire (or will acquire) data.

Common types of systems include:

- Legacy systems—all the older systems still operating in your information systems environment
- Hosted systems—cloud, application service provider (ASP), and outsourced
- Web systems—both internal and external
- Social media systems—Twitter, Facebook, and other sites from which you can gain much customer and brand insight
- Big data systems—the increasing volume and variety of data that is available and accessible today

Data Integration Approaches

Many Types and Levels of Integration



Data Integration Approaches

Many Types and Levels of Integration

LEVELS OF INTEGRATION

Data integration involves the application of processing rules to transform data sets. Data can be integrated at physical, logical, or semantic levels.

- *Physical integration* occurs when processing rules are applied to one or more data sets, and the results are stored in a database.
- *Logical integration* occurs when the processing rules are defined but not necessarily applied in advance. Database views and virtualization solutions provide logical integration.
- *Semantic integration* occurs when a business view of information is mapped to technical structures that are either logical or physical.

These different levels of integration can be combined to meet the needs of an integration architecture. For example, a team may decide that logical integration will be used to abstract each data source, thereby shielding other systems from underlying changes. Physical integration, in turn, will be used to combine the data from these logical interfaces into a single data warehouse repository. Lastly, sematic integration provides a unified business glossary of data in the warehouse tables.

TYPES OF INTEGRATION

There are several types of data integration. As with levels, these types of integration may be mixed and matched according to business needs:

- *Data relationships* refers to integration activities that are based on key to key correspondence. Key-based relationships do not necessarily require a DBMS. For example, it is possible to link customer IDs from two or more text files.
- *Data unification* refers to integration activities that consolidate or merge data sets.
- *Data blending* refers to synthesis of data from multiple sources where clear key relationships do not exist. Blending may involve the use of fuzzy matches, probabilities, or alternate keys.

TIMING OF INTEGRATION

The different types of integration can be performed at different levels and on a variety of schedules. Data may be integrated in batch, as transactions occur, or on demand.

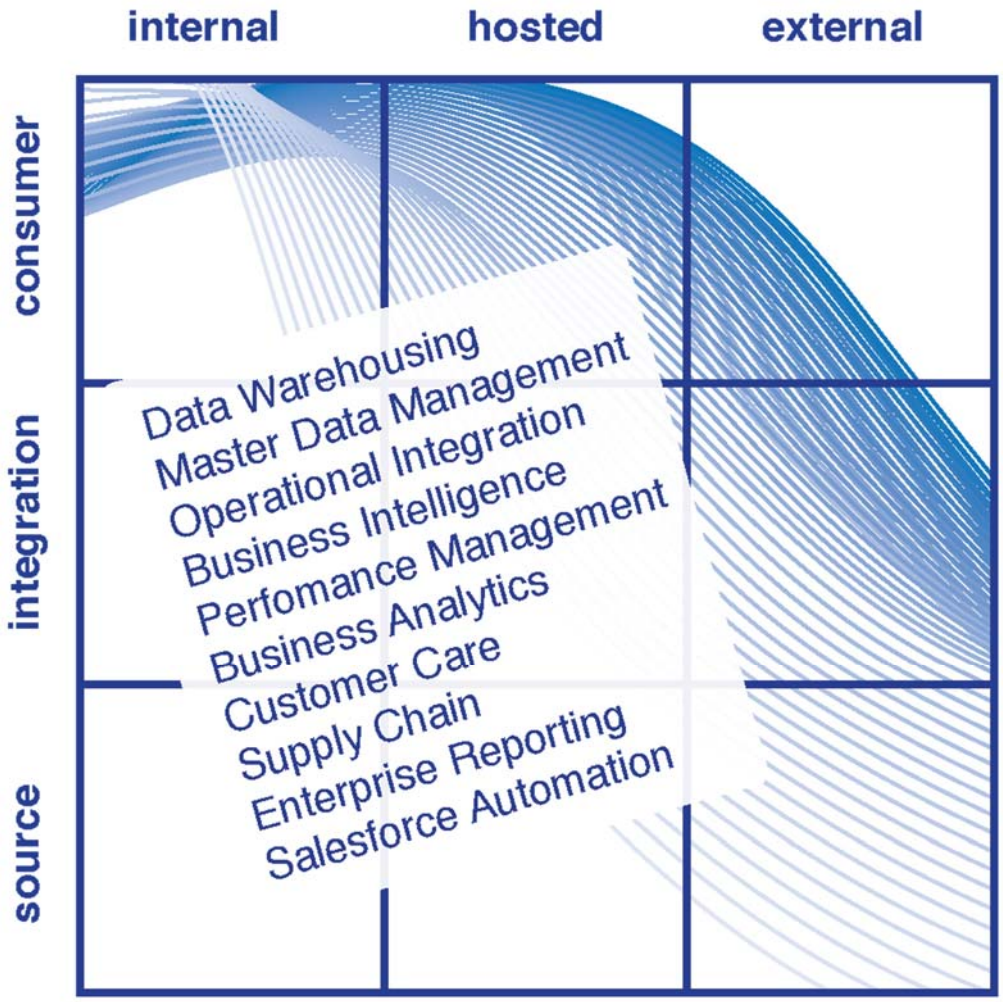
Transforming Data With Intelligence™

Module 4

Using Data Integration Architecture

Systems and Data Integration Architecture

Existing Systems



Systems and Data Integration Architecture

Existing Systems

ARCHITECTURAL COMPLIANCE

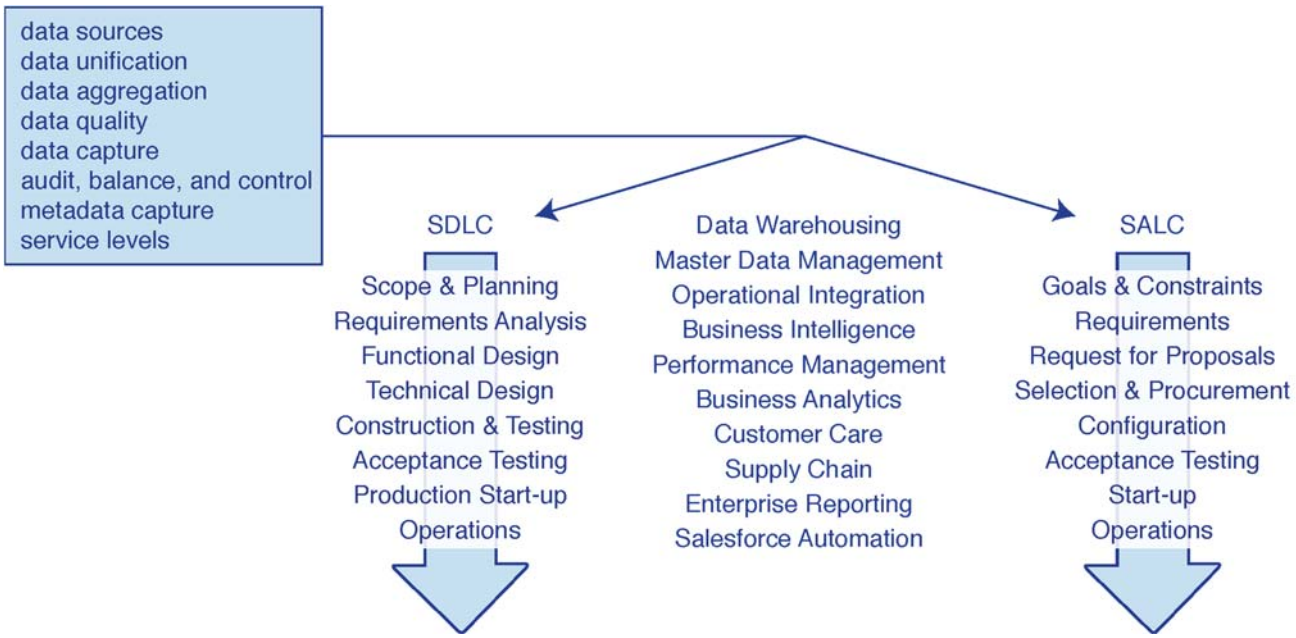
You already operate data integration systems—some that simply consume integrated data, some which provide integrated data, and some that do both. When a new architecture is defined or an existing architecture updated, several architectural compliance questions arise:

- To what degree are existing systems required to comply with architectural guidelines and constraints?
- How and by whom should architectural compliance reviews be conducted?
- What kinds of existing system changes should be driven by architectural compliance? What priority do these changes have when competing with other system maintenance items?

Architectural compliance will be influenced by how architecture is managed in your organization. For instance, do you have an enterprise architecture organization? Enterprise architecture practices may influence what defines compliance in your organization. If enterprise architecture does not exist in your organization, is compliance defined by the data integration team? These are other considerations in defining architectural compliance.

Development and Data Integration Architecture

Requirements Analysis



Development and Data Integration Architecture

Requirements Analysis

ARCHITECTURE-DRIVEN REQUIREMENTS

Data integration architecture is a useful tool to guide requirements analysis for data integration systems. Review architectural capabilities, constraints, and guidelines when gathering requirements for each of the following.

- Data sources—Which kinds of data sources are readily supported in the architecture? Which may push the limits of your data integration architecture?
- Data unification and aggregation—What kinds of data consolidation are needed? Which are readily supported in your architecture? Which methods and data integration functions map to the requirements? At which layers are unification requirements met?
- Data quality—What requirements do you have for data cleansing and quality improvement? How are they supported by integration methods and functions? At which layers should data cleansing be performed?
- Data capture—What data needs to be captured in data acquisition processing? What timing, frequency, and latency are required? Is data acquisition triggered by scheduling or by consumer demand? What architectural constructs support the data capture requirements?
- ABCs—What audit, balance, and control requirements do you have? How are they supported in data integration architecture? Where should audit and control features be placed in data flow?
- Metadata capture—What metadata requirements do you have? How are they supported with architectural components for metadata repositories and metadata rationalization?
- Service levels—What service level requirements do you have? What architecture guidelines and capabilities are best suited to meeting various requirements for performance, security, scalability, etc.?

