THE LEADING PRACTICE
DATA REFERENCE CONTENT
#LEAD-ES20012BC

A Data Ontology & Data Semantic Description, Views, Stakeholders, and Concerns

Version Status: LEAD 4.0

Contents

Executive Summary

Numerous studies link project failure to aspects of information modelling, data architecture failure, and poor data standardization and integration. Many of these challenges come from a poor information vision, stakeholders, and requirements not linked to the description of the system, software, and enterprise architectures, leading to weak information blueprinting and execution. This Data Reference Content provides a data ontology supported by data description, specific data semantic relations, and correlations. It is based on a collection of best and leading practice about how to work with data within the enterprise modelling, enterprise engineering, and enterprise architecture disciplines. In following the international standards for architecture descriptions of systems and software of ISO/IEC/IEEE 42010, it organizes an architecture description into multiple architecture views. The LEAD Data Reference Content presents a new system and software engineering and architecture way of thinking, working, and modelling to work with data in a holistic manner and create a comprehensive and integrated view of the modern enterprise. It presents standardized definitions/terminology, a conceptual foundation, and a logical design for use in the architecture description of the enterprise, supported by meta objects, an ontology, and required semantics, relations, artefacts, and templates. These tools include providing the ability to link stakeholders needs, wants, requirements, concerns, and viewpoints, as well as architectural descriptions and an architecture framework to the applicable data objects.

The Data Reference Content is an essential tool for any practitioner working with and around the data aspects of an enterprise. It provides an overview of the key data centric features of the enterprise and how the data assets relate to the various dimensions of the business e.g. to goal, process, measurement and information reporting, and to the data/information channels. The Reference Content therefore provides a standardized, repeatable way of analysing, appraising, approximating, assessing, and capturing data related objects to enable innovation and transformation.

The critical value and reasons to use the Data Reference Content are:

- It provides a uniform and complete data ontology with its specific data descriptions, semantic relations, and correlations.
- It defines how to organize and structure the viewpoints and objects associated with data development and data management.
- It establishes a common practice for creating, interpreting, analysing, and using data objects within a particular domain and/or layers of an enterprise or an organization.
- It provides a set of principles e.g. how and where the data objects are related (and where not).
- It is vendor neutral and agnostic and can therefore be used with most existing frameworks, methods, and/ or approaches that have any of the meta-objects mentioned in this document.
- It has a structured repeatable pattern for data related objects, structures, and artefacts (the basis of our standards). The definition of a pattern used here is the description of the repeatable and mostly used/generic specifications and relations of a topic, not all theoretically possible specifications, or relations.

- It uses Data Standards that increase the level of re-usability and replication.
- It employs a fully integrated and standardized set of data maps, matrices, and models that allow for advanced ways of thinking, working, modelling, implementation, and governance of the strategies for the use of data.

Overview of our Analysis and Research Findings around data as well as information modelling and architecture

Data Modelling, data engineering, and data architecture disciplines are not new. In our analysis of the organizations' Business Information and Technology Information usage, e.g. information and data usage, reporting, integration, standards, etc. we identified some key issues. For example, we identified that less than 25% of organizations succeed with their information efforts, leaving more than 75% of the organizations feeling that they did not succeed with their information automation, data standardization, or data transformation efforts. Our analysis also found that most project failures are not related to the product or software issues but are mostly connected to poor information and data, and derived from these failures, poor software product specification.

The largest failures identified within business and IT enabled information transformation are found in the areas of:

- Poor alignment of Information goals with business requirements and business goals, resulting in complex and redundant roles, functions, processes, and information enabled by overly complex and expensive applications lacking end to end ability to understand organizational performance and risk.
- Poor data requirements management: Primarily by linking business and functional requirements together, and not adequately distinguishing the role and the nature of each, enterprise data is not available to support risk management, compliance, and other strategic needs, leaving executives without ready access to an essential tool for their success.
- Defining Information functions based on the process activities, not on the context in which they are executed or the value they produce. The result of this is critical information assets become trapped inside process stovepipes and are not available to support decisions, or are redundantly created with shadow processes at great expense.
- Not addressing the possible duplication of business functions, services, and tasks at a cost in effectiveness and increased risk due to the complexity of the resulting way the business must work.
- ERP, CRM, SCM, Portal systems are implemented based on current business practice as opposed to designing and building to a future vision; automating the activities that the organization has today without consideration of their value. Alternatively, by automating the process flows without consideration to service flows and information flow, creates a nightmare of the broken and siloed information, process, service, or other flows, which work only within a narrow part of the enterprise.
- Not identifying measurement and reporting needs, or not re-thinking the way reporting is done, thereby automating manual approaches to reporting. The result is that while the full

potential of information technology is not obtained, the cost of development is higher than necessary.

- Applying best practice, out of the box functionality of enabling software to areas where the organization is unique; not realizing that this will destroy the organization's uniqueness and thereby their basis for value creation, perhaps even destroying its ability to achieve its mandate, or not exploiting out of the box functionality. The result to the agency is unnecessarily forcing it to bear the cost of customization when it is not actually warranted.
- Addressing issues related to the duplication of information/data with complex and otherwise unnecessary investments in the integration of these resources without resolving the root cause of the underlying problem creating complex and fragile connections between applications within overly complex and expensive application portfolios.
- Little to no transformation embedded in the program of investment, leading to a failure to exploit the full potential of the opportunities associated with automation.

Our findings were confirmed when cross-referencing the Global University Alliance research findings to similar studies in this area. Among them, the 2012 IBM Global CEO study[1] confirmed our findings, in that it concluded that 86% of executives say that while business and IT innovation is extremely or very important to their organizations' growth strategy, meanwhile only 19% feel they succeeded with any aspects in their value execution. Both the 2011 and 2013 McKinsey study[2] on transformation identified that over 72% of transformation programs failed to deliver their actual targets. This resulted in substantial economic and productivity losses of $3 Billion, which corresponds to 4.7 % of global GDP.

On the other hand, when analysing the organizations that succeeded with their information projects and transformation projects there were two key observations:

1. When considering information aspects the first finding was that those not succeeding with their information initiatives where considering information modelling and architecture in terms of information is equal to data aspects.
2. While Information Modelling, Software Engineering, and Information/ Solution Architecture are considered distinct and different disciplines, they employ the use of common objects e.g. process and service automation, system measures, resources/roles, system reporting, information, and data objects, etc.

To finding point 1): Traditional view - information is equal to data aspects

When considering information aspects one of the findings was that many organizations still are talking about the traditional view of information architecture, which are more about data aspects e.g.:

[1] McKinsey Transformation Study, 2013, McKinsey
[2] Global CEO Study 2012, IBM Institute for Business Value

- **Metadata Domain** – Defined as 'data about the data'. Metadata is the information that describes the characteristics of each piece of corporate data assets
- **Master Data Domain** – Refers to instances of data describing the core business entities, such as customer or product data.
- **Operational Data Domain** – Also referred to as transactional data capturing data, which is derived from business transactions.
- **Unstructured Data Domain** – Also known as content, typically managed by an enterprise content management application.
- **Analytical Data Domain** – Usually derived through transformation from operational systems to address specific requirements of decision support applications.

When considering the relationship between information models and data models, as the former is about the business and the latter relates solely to what exists or is planned for automation, LEAD does not view one as a type of the other. While they are each unique relative to the other, both information models and data models are an abstract, structured representations of concepts including the need to capture the properties, relationships, and operations that can be performed. The things in an information model may be kinds of real-world objects, such as devices in a network, or they may themselves be abstract, such as those items used in a billing system. Typically, they are used to model a constrained domain that can be described by a closed set of data entity types, properties, relationships and operations. Therefore the above data modelling aspects are relevant to information modelling, including both the business information as well as the technology information aspects across the information construct, the information users, its information flow, the information measurements as well as the information analytics and information reporting assets used to capture and store its decisions. The extent an organization is able to exploit its information assets is a driver of its ability to excel at the cost savings and/or improved quality.

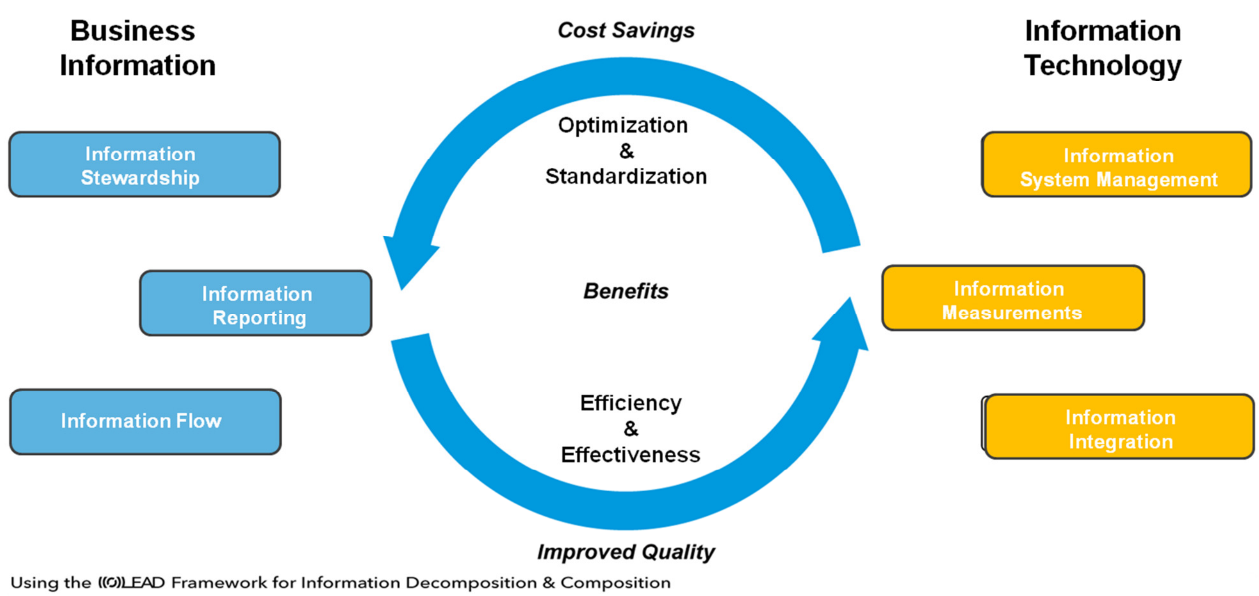

Figure 1 - Business Information versus Information Technology Focus and benefits

To finding point 2): Common cross-objects

While Information Modelling, Software Engineering, and Information/Solution and even data Architecture are considered distinct and different disciplines, they employ the use of common objects e.g. information objects, data objects, service aspects, reporting, system measures, etc.

We also see that that the data models for different information systems are arbitrarily different. The result of this is that complex interfaces are required between systems that share data. These interfaces can account for between 25-70% of the cost of current systems. We would argue that interfaces should be treated as an inherent part of the design as the final product on its own would not be usable without the appropriate interfaces.

Cross-referencing these insights to the reasons identified for the breakdown in the ability of data concerns to be addressed, it became clear that the root cause for the failure was the inability to link the subjects across areas. What was additionally revealing was that the organizations that prevailed actually solved the very same challenges by taking a holistic approach between data engineering, data modelling, and data architecture. With this knowledge, it became clear that one of the biggest challenges faced by most organizations is a lack of understanding of the common objects within an organization and how they relate to each other. Below are some examples that illustrate how multiple aspects relate together in a basic Information System Construct:

- Information System Component.
  - Logical Information System Components may be defined within other Logical Information System or Application Components.
  - The Logical Application Component expresses a Physical Information System or Application Component.
  - The Physical Application Component will be part of a Software Product Version.
  - The Physical Application Component contains one or more Application Modules.
- Information System Module.
  - An Information System or Application Module is contained in an Application Component.
  - An Information System or Application Module consists of one or more Information Application Functions.
- Information System Feature.
  - An Information Feature characterizes the behaviour of an Information Function.
- Information Application Function/Functionality.
  - An Information System Function is part of an Information System Module.
  - An Information System Function is bounded by an Information Rule.
  - An Information System Function is defined within a System Flow.
  - An Information System Function is decomposed into Information System Tasks.
- Information System Task
  - An Information System Task Is part of an Information System Function.
  - An Information System Task orchestrates and is orchestrated by a System Flow.
  - An Information System Task executes an Information Rule.
  - An Information System Task is accessed via one or more Information Channels.
  - An Information System Task is executed by one or more Information System Roles.

    - An Information System Task is a generalized case for an Information Event and an Information Gateway.
    - Information System Tasks are measured by Automated Process Performance Indicators.
    - Information System Tasks call Information Interfaces.
  - Information Events
    - An Information System Task is a generalized case for an Information Event and an Information Gateway.
    - An Information Event, Orchestrates and is Orchestrated by an Information Task.
- <u>Information Service</u> in the form of either an application or Data Service.
    - System Service is delivered through an Information Interface.
- <u>Information Measurements:</u> an information measure typically refers to either - the progression of establishing the performance magnitude of some of the defined attributes of an information object relative to some unit of defined Business Measurements.
    - Information Measurements come and relate to business measurements e.g. Key Performance Indicators (KPIs), Process Performance Indicators (PPIs), and Service Performance Indicators (SPIs).
    - Information System Measurement exposes measurements' result to Object(s) e.g. business, information, and data.
    - System Measurement is captured or presented in a System Report.
- Information Reporting
    - Reporting exposes and portrays through information system functions, tasks, and services. This illustrates and expresses what is happening or what has happened and includes timely collection, analysis, and dissemination of information to operational, tactical and or strategic management level.
    - System Measurements are reported to the System Owner.
    - System Report exposes Data Object.
    - System Flow orchestrates System Report.
- Information Security
    - Information Security is bound by one or more Business as well as Information Security Policies.
    - Information Security Policies are executed within the Business through Business Information Security Rules.
    - Information Security Policies are executed within the Information System through Information System Rules and Data Security Rules.
    - Technology Security is bound by the Information and Data Security Rules.
- Information Integration/Interface
    - An Information Interface exposes an Information Task.
    - The content of an Information Interface is delivered through a System Service in the form of either a Data or Information Service.
- <u>Information Standardization:</u> refer to the existence of a published standard or the development of a standard to increase the level of replication and cost.
    - Information Standardization is based to applying a standard that either exists or is being developed.
    - Standardization is bound to the level of possible replication

- Information Devices: are changing the face of communication, entertainment and organizations and are also influencing how business is conducted and information is managed.
    - Information Devices extend the use and reach of information objects.
    - Information Devices consume application and data services.

While most information and data architects understand the need to structure and relate the multiple entities together, we see that neither data nor information engineering recognize the same need when engaged in their respective work to capture and represent the total picture. The findings revealed the need for a fundamental shift in approach and thus the need to rethink Information modelling, software engineering and Information architecture and the relations among both. While analysing, evaluating and comparing existing software and system engineering and architecture concepts, descriptions and framework, both the Global University Alliance and the LEADing Practice community, realized that the foundation for this reconceptualization of interlinking the disciplines was to understand the objects that link and relate to the data and information system aspects.

Using ontology principles to understand the very nature, the basic categories, as well as using semantic principles to identify which parts relate or should relate, exposed sixteen areas that together provide a starting point that can be used to guide the analysis, decomposition, composition and construction of a software and system architecture description. The Sixteen main areas are presented in Figure 3.

Overview of the Data Reference Content

Introduction

The LEADing Practice Data Reference Content provides the specification of the LEAD Data ontology with its Data descriptions, specific Data semantic relations, and correlations. It is based on a collection of best and leading practice on Data modelling, Data engineering, and Data architecture disciplines. The Data Reference Content is therefore an essential tool for any practitioner working with and around any aspects of data. It provides a structural way of thinking, working, modelling, implementation, and governance, with respect to data, its definitions, and how data centric strategies are applied within business functions and organizational constructs. Providing an overview of the key data aspects of the organisation and how they relate to the various components of the business e.g. goal, process, measurement and information reporting, and data/information channels. It therefore provides a way of analysing, appraising, approximating, assessing, and capturing data related objects to enable innovation and transformation. Below is a summary of why to use the Data Reference Content.

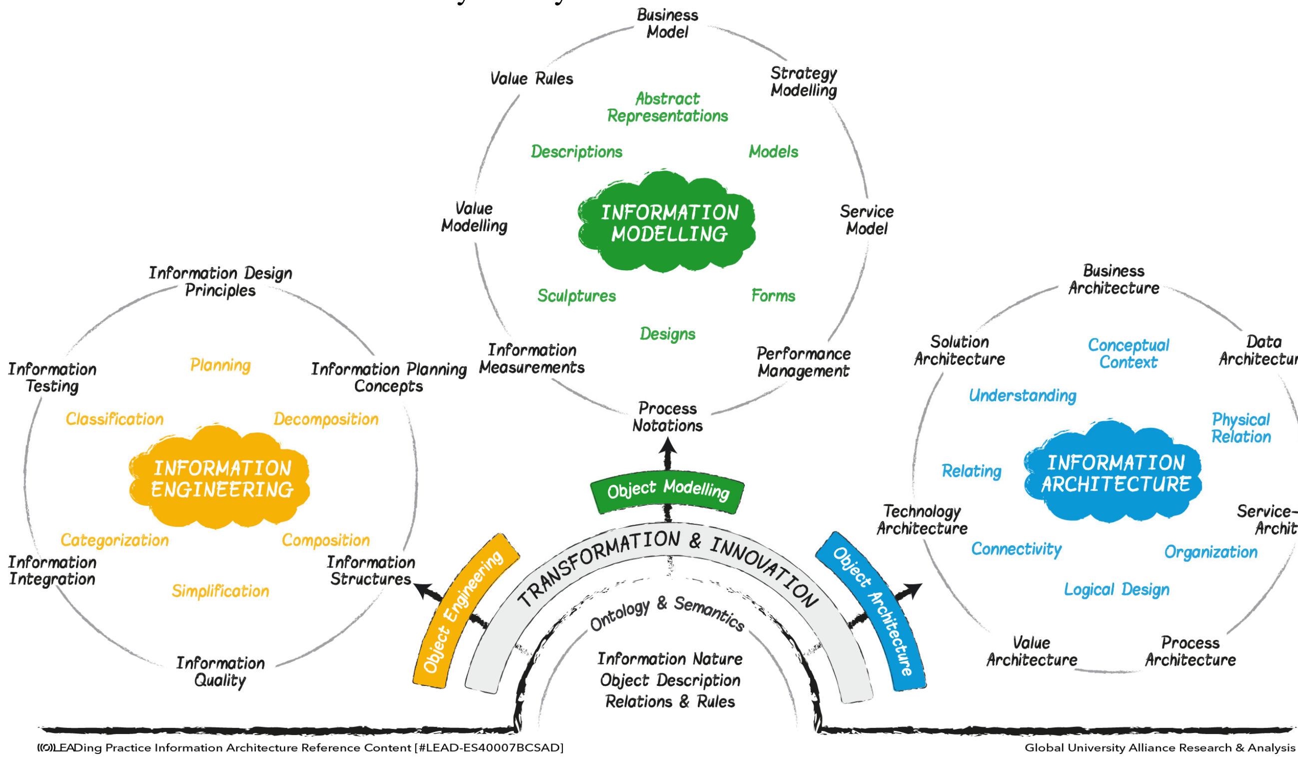

Figure 2 - The Data Objects are part of the many semantic relations between the Data engineering, Data modelling, and Data architecture enabling transformation and innovation.

Why use the Data Reference Content?

- It provides a complete data ontology with its specific data descriptions, semantic relations, and correlations.
- It defines how to organize and structure the viewpoints and objects associated with data development and data stewardship.
- It sets out guiding principles to establish a common practice for creating, interpreting, analysing, and using data objects within a particular domain and/or layers of an enterprise or an organization.
- It provides a set of principles of how and where can the data objects be related (and where not).
- It is vendor neutral and agnostic and can therefore be used with most existing frameworks, methods, and /or approaches that have any of the meta-objects mentioned in this document.
- It has a structured repeatable pattern for data related objects, structures, and artefacts (the basis of our standards). The definition of a pattern used here is the description of the repeatable and mostly used/generic specifications and relations of a topic, not all theoretically possible specifications, or relations.
- Its uses a data standard that increase the level of re-usability and replication.

- It has a fully integrated and standardized data maps, matrices, and models that allow for advanced ways of thinking, working, modelling, implementation, and governance of the strategies.

Data related Meta-Object Ontology and Main Characteristics

Using ontology principles to understand the very nature, the basic categories, and semantic principles to identify which categories or relate or should relate exposes sixteen areas that together provide a starting point that can be used to guide the analysis, decomposition, composition, and construction of Data. The sixteen main areas are presented in Figure 3 below.

In order to have a structured way of thinking, working and modelling within the Data Reference Content, the three main properties characterizing the meta-object relevant to modelling and architecture principles are applied:

- Identity: the decomposed Data objects that distinguishes it from other object areas.
- State: describes the purpose of the composed object.
- Behaviour: describes how the decomposed or composed objects can be used with other meta-object's relations across other modelling disciplines and architectural layers

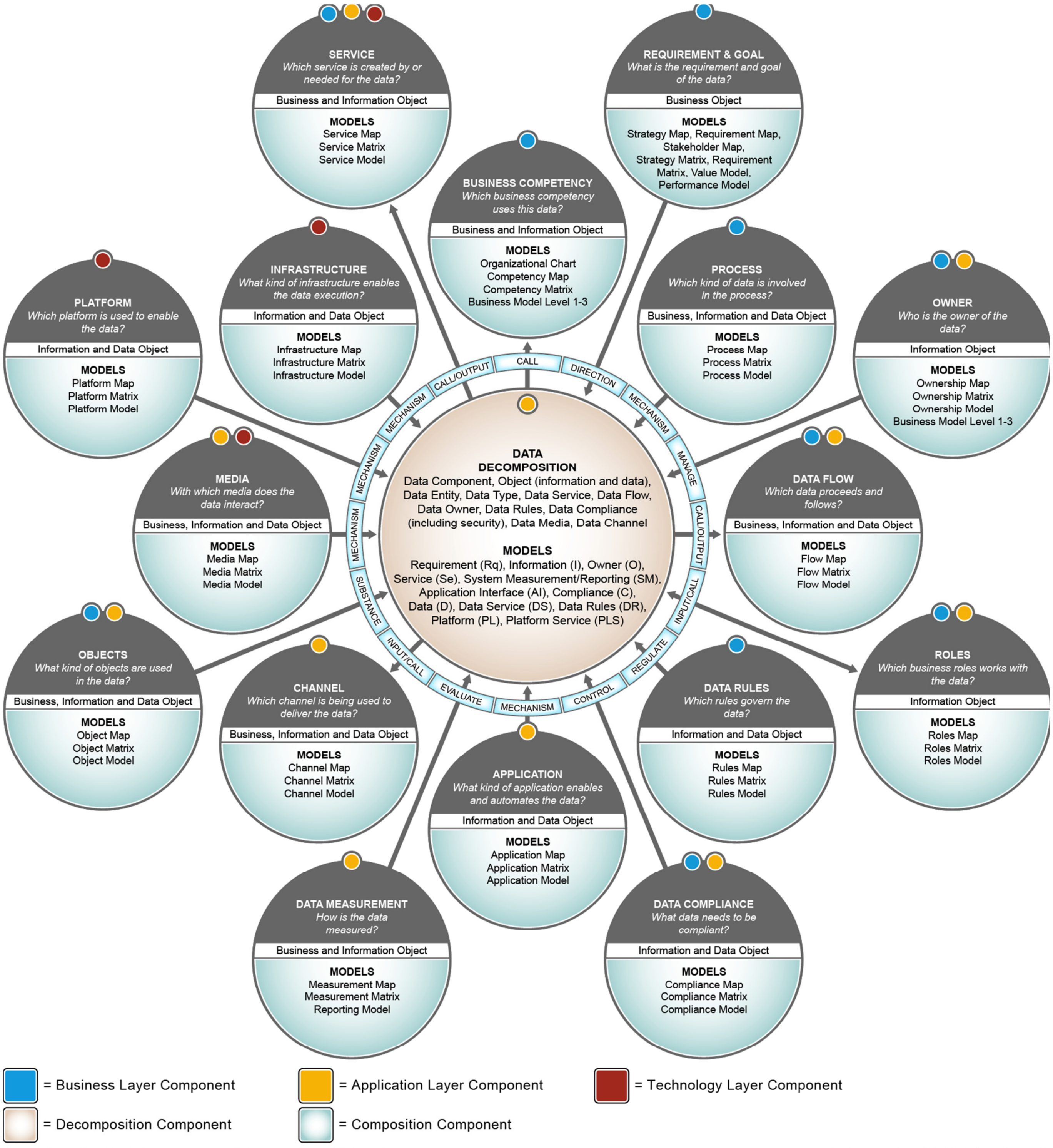

Figure 3 - The 16 LEAD Data Decomposition and Composition objects

1. Through its business competencies the business uses a set of information tasks, data objects and data services. This combination, together with the right business model, will

reduce cost, improve operation in terms of effectiveness and efficiency, and support revenue growth.

2. Business goals and requirements will dictate the goals and requirements for the information while business objectives, performance expectations, and performance indicators can be measured through the executing business and the enabling data.
3. Business services can be (partly) delivered and/or consumed by information system tasks and data services. The mentioned parts of the Information and Data are subject to the relation between the business information service provider and consumer, to the business service construct / delivery and to whether it is a main or a supporting business service. The business service is automated by the information system service, data service, platform service and infrastructure service.
4. Through Information services, business process steps can be automated. Information tasks automate process activities. These can be executed as pre-programmed, as reaction to specific events, as well as business decisions and real time monitoring of processes is done through the data.
5. The data objects are used, created, and stored in several flows in the business such as reporting flow, service flow, information flow, process workflow, information workflow, and the data flow itself.
6. Analytical data is today the source of strategic and real time decision using data in combination of business performance indicators, key performance indicators on the strategic, tactical, and operational level, including for service level agreements, process performance indicators and relate them to the relevant scorecards, dashboards, and cockpits.
7. Information system tasks, and services create, use and/or deliver (parts of) business, information, and data objects. An Information uses, modifies, and/or produces data on several hierarchical levels: information system modules work with data component, information system function with information object, and information task with data service.
8. The business roles use, create, and store through information system functions and tasks various data.
9. When dealing with information and data, different owners can be recognized. All owners have specific responsibilities, which result in different demands and wishes of various aspects of the information and data. There are owners with responsibilities concerning business, process, service, value, performance, information, application, data, platform, infrastructure, security, and compliance, etc.
10. Several rules, which apply to services, processes, applications, data, and security, have to be adhered to and embedded within the information and data constructs.
11. When integrating and or standardizing data objects, the direction is set by strategy, policies, guidelines, standards, regulations, and legislation as well as issues of governance controls, risk management, audit, evaluation, security and monitoring must be taken into account in order to verify their compliance.
12. Data services and information interfaces need to support different business and technology channels. The business channels can either include marketing, sales, distribution, service or other channels; the technology channels can be communication, digital image/screen, programming, broadcasting, I/O, or audio channels.
13. The information system modules and tasks use the data component and services.

14. Data entities and data objects will make use of media e.g. media store, computer media can be data storage devices, information software or other computing media.
15. A platform is used to enable information on several hierarchical levels: platform component enables information system component, platform service enables data services.
16. The data components and data objects reside on infrastructure components. Infrastructure services support the platform services.

As demonstrated, these sixteen data relationship groups not only have relationships to the central concept of data; they have associations and correlations with one another, leading to multiple interaction points of interest. One could ask why this is relevant, especially as in traditional data modelling concepts the focus is solely on data. It is here we believe that the complexity of data and where and how it is related has become so multifaceted that only looking at the data aspect itself will limit the data modelling, data engineering, and data architecture and therefore limit the questions that can be addressed. It is from this that we draw the conclusion that this limit is the root cause of the challenges with more traditional views of data analysis and architecture. At the core of this, is the use of the LEAD composition and decomposition concept applied to the meta objects, enabling the ability to model, engineer and architect things. This is relevant for both data models and data architecture. To illustrate this we have taken an example[3], customers PLACE orders, customers LIVE AT addresses, and line items ARE PART OF orders. Place, live at, and are part of are all terms that define relationships between entities. The relationships between entities are conceptually identical to the relationships (associations) between objects. For example, it is not enough to know that customers place orders. How many orders can a customer place? None, one, or several?

Furthermore, relationships are two-way streets: not only do customers place orders, but also customers place orders. This leads to questions such as, how many customers can be enrolled in any given order and is it possible to have an order with no customer involved? The first thing to notice is the various styles applied to relationship names and roles – different relationships require different approaches. For example, the relationship between Customer and Order has two names, "places" and "is placed by", whereas the relationship between Customer and Address while also has two names, "customer is located at address" and "Address identifies location of Customer" This example reveals the basic relationships of objects, in one direction you use the active verb and in the other its equivalent passive equivalent. The idea is that you want to specify how to read the relationship in each direction, however sometimes it is redundant – you are better off to find a clear wording for a single relationship name, decreasing the clutter on your diagram. Similarly you will often find that by specifying the roles that an entity plays in a relationship will often negate the need to give the relationship a name (although some Computer Assisted Software Engineering (CASE) tools may inadvertently force you to do this). For example the role of billing address and the label billed to are clearly redundant, you really only need one. For example the role part of that Line Item has in its relationship with Order is sufficiently obvious without a relationship name. All of the above is data specific.

---

[3] http://www.agiledata.org/essays/dataModeling101.html#sthash.x3BbRt2u.dpuf

Therefore, to identify and capture all aspects of the relevant data correctly, it was necessary to re-think information as well as data modelling and architecture as it existed. In essence, defining a better and more integrated and standardized Way of Thinking, Working, and Modelling of information and data.

Data Relevant Definitions

In the context of the data reference content the following to definitions are relevant:

- Data Engineering: Data engineering spans aspects of data practice from data planning, data analysis, data design, and data quality concepts, to address data structures aimed at achieving integration. Data Engineering is defined as "An integrated and evolutionary set of tasks and techniques that enhance business communication throughout an enterprise, enabling it to develop people, procedures, and systems to achieve its vision".4 Data engineering is also defined as the generation, distribution, analysis, and use of data in information systems. This latter definition involves the usage of machine learning, data mining, and other computational methods to enhance the presentation and understanding of the high-throughput data that are generated by different systems.
- Data modelling: Data modelling is the act of exploring data-oriented structures. According to Hoberman (2009),5 "A data model is a way finding tool for both business and IT professionals, (who use) a set of symbols and text to precisely explain a subset of real information (with the goal) to improve communication within the organization and lead to a more flexible and stable application environment." It is a representation of information in a structured (graphical, using a specific notation) or formalized (mathematical i.e. using the Z notation) manner suitable for communication, interpretation, or processing by humans or by automatic means. Examples of these structures includes whole models, packages, entities, attributes, classes, domain values, enumeration values, records, tables, rows, columns, and fields.
  A data model explicitly represents the structure of data at rest. The data model is based on the data itself, the data relationships, the data semantics, and data constraints. A data model provides the details of what is to be stored in a persistent or permanent structure, and is of primary use when the final product is the generation of computer software code for an application or the preparation of a functional specification to aid a computer software make-or-buy decision.6 Related to the data model is the class model, which is used in object oriented modelling. In the latter a class is a template for creating objects - In general a class is a construct which defines a set of static properties and its behaviour as "methods', while a Data Object is the actual instance of a class which is created through the execution of software. Fundamentally, the difference between entities within the relational model and objects from within the object model that instances of entities

---

[4] John Hares (1992). "Information engineering for the Advanced Practitioner", Wiley.
[5] Steve Hoberman, "Data Modeling Made Simple 2nd Edition", Technics Publications, LLC 2009
[6] Paul R. Smith & Richard Sarfaty (1993). Creating a strategic plan for configuration management using Computer Aided Software Engineering (CASE) tools. Paper For 1993 National DOE/Contractors and Facilities CAD/CAE User's Group.

are recognized through unique, or primary accession numbers, while objects are identified by a name which instantiates their uniqueness.
Within either the relational or object paradigm, when determining the properties or the entity or class respectively, the data attributes are assigned to entity types just as you would assign attributes and operations to classes. There are associations between entities, similar to the associations between classes – relationships, inheritance, composition, and aggregation are all applicable concepts in data modelling. When working in either of these two approaches the only real difference between the traditional, rational, approach and the objected oriented approach, is that the former focuses solely on data structure – whereas class models allow for the exploration of both the behaviour and structural aspects of the data at the same time. Because of this focus, data modellers have a tendency to be much better at getting the data "right" than object modellers, but when using this approach the linkage to behaviour can be a challenge. That being said, some current data practice allows database methods (stored procedures, stored functions, and triggers) to be addressed within the processes related to the representation of physical database.7

- Data Architecture: In information technology, data architecture is composed of models, policies, rules or standards that govern which data is collected, and how it is stored, arranged, integrated, and put to use in data systems and in organizations.8 Data is usually one of several architecture domains that form the pillars of an enterprise architecture or of a solution architecture.9 In this relation, a Data Architecture is one of several architecture domains that form the pillars of an enterprise architecture or solution architecture. Note that the term "Application Architecture" (without the "s") is commonly used for the internal structure of an application, for its software modularisation. The most notable difference between regular Enterprise Architecture and Layered Enterprise Architecture is the concept of building and structuring objects within a framework of inter-connected layers of Business, Application (Software and Data), and Technology. Applications Architecture (with the "s") describes the parts and relationships of the portfolio of applications. It is the science and art of ensuring the suite of applications being used by an organisation to create the composite application is scalable, reliable, available, and manageable. One not only needs to understand and manage the dynamics of the functionalities the composite application is implementing but also help formulate the deployment strategy and keep an eye out for technological risks that could jeopardize the growth and/or operations of the organisation. Most data architectures come from a set of information silos; therefore, they require highly customized point-to-point transformative data flows between information silos to integrate data. These disparate data architectures are clearly inadequate to support any standards-based data integration since each information silo is not designed to be compatible with any other information silo.

---

[7] See more at: http://www.agiledata.org/essays/dataModeling101.html#sthash.x3BbRt2u.dpuf
[8] Business Dictionary - Data Architecture
[9] What is data architecture GeekInterview, 2008-01-28, accessed 2011-04-28

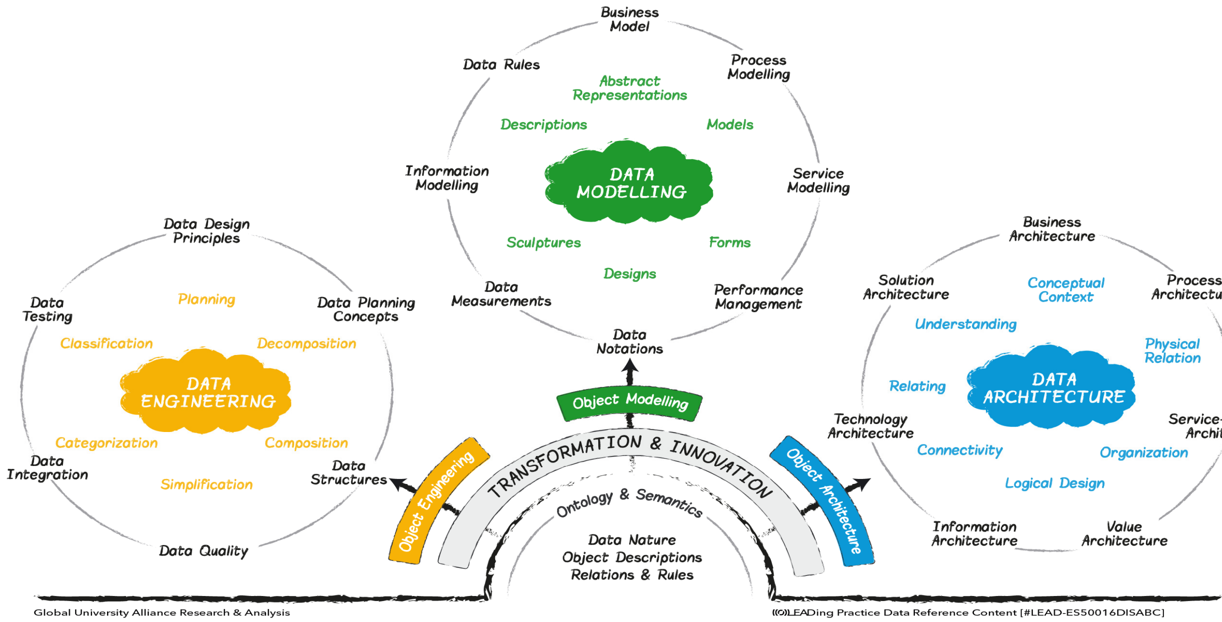

Figure 4 - The relations between the data engineering, data modelling, and data architecture enabling transformation and innovation.

As discussed earlier are their common aspects between the three distinguished data disciplines of data engineering, data modelling, and data architecture. Below is a list of these objects.

Roles Involved

Various different roles are involved to capture the relevant data aspects and related meta-objects. Each role both within the fields of data modelling, data engineering or data architecture provides a specific view addressing a particular stakeholder concern around data. The mentioned data templates e.g. maps, matrices and models, are specifically designed to capture the relevant stakeholders, their requirements, their concerns, the data aspects involved, the applications relevant and all the architectural rules. Each of these artefacts have the relevant pattern and relationships build into them. Therefore fully integrated and standardized data templates enable the expert/practitioner, or architect to work with the relevant data aspects throughout all the architectural layers (business, application, and technology) to create a complete and traceable design. The following roles are involved in the data reference content:

ENTERPRISE MODELLERS	ENTERPRISE ENGINEERS	ENTERPRISE ARCHITECTS
Business Analyst (P) Process eXpert (S)	Value Engineer (S) Technology Engineer (S)	Business Architect (S) Process Architect (S)

Value eXpert (S) Information eXpert (P) Data eXpert (P) Transformation eXpert (S)	Process Engineer (S) Quality Engineer (S) Software Engineer (P) Data Engineer (P) System Engineer (P) Change Engineer (S)	Value Architect (S) Service Oriented Architect (P) Solution Architect (P) Information Architect (P) Data Architect (P) Technology Architect (P) Enterprise Architect (P)

Table 1: the involved roles in the data reference content

(P) = Primary object/role (S) = Secondary object/role

This ability to conduct advanced role modelling and relating the relevant data aspects and data objects throughout the layers is one of the strengths of the LEAD Reference Content. This is true as not only are the role objects governed by their connection modelling rules, but also how and where the templates link and share common objects is defined and standardized that the connections can be made

Data Object(s) - Decomposition

Data automation is available to different kinds of information centric business activities and interact in several ways with objects in their surroundings, such as business, information, and application objects. To make these relations explicit it is necessary for data to be decomposed in a systematic manner to expose its primitive components e.g. data task, function, services etc. Each component can then be placed in an unambiguous relationship with its surrounding parts and objects. Details about these Objects may be captured or presented or presented in various applicable Maps, Matrices, and/or Models.

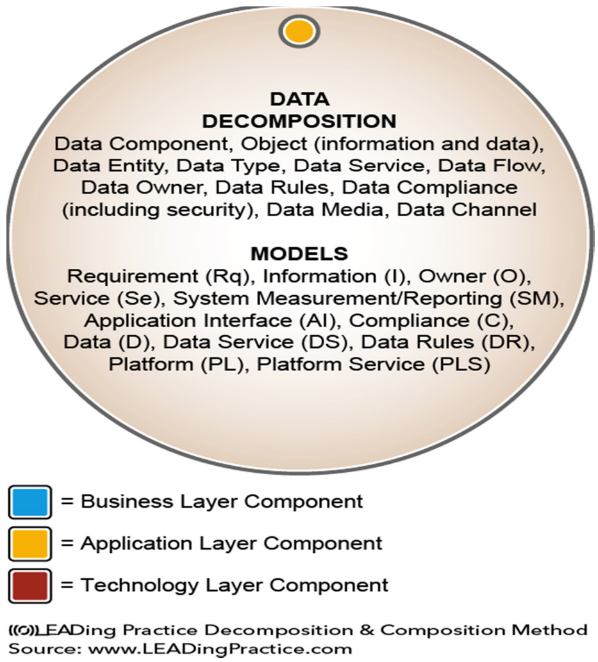

Figure 5 - Components of an Initial Data Decomposition

Summarized; Data can be **decomposed** into the following Objects:

- Data Component
- Object (information and data)
- Data Entity
- Data Type
- Data Service
- Data Flow
- Data Table
- Data Owner
- Data Rules
- Data Compliance (including Security)
- Data Media
- Data Channel

Data Object(s) – Overview All Data related LEAD Objects

The following (decomposition and composition related) LEAD objects are the most relevant to data components within the Data Reference Content and its templates:

Data Meta-object	**Decomposition related meta-objects-**
Data Component	A cohesive collection of data that is part of an application.
Information Object	A real world thing of use by or which exists within the enterprise and information objects reveal only their interface, which consists of a set of clearly defined relations. In the context of the business competency, the relevant objects are only those that relate to the enterprise's means to act.
Data Object	A logical cluster of all sets of related data representing an object view of a business object.
Data Entity	An encapsulation of data. Logical data entities are specification of the organization of the information to store the data as a physically persistence structure e.g. data tied to applications, repositories, and services.
Data Nature	The categorization of the nature of the data into: unstructured or structured.
Data Type	Identifying one of various categories of data types into meta, mater, transaction, operational or analytical.
Data Service	A standards-based, uniform means of accessing information in a form useful to enterprise applications without requiring knowledge of its physical persistence structure.
Data Table	A physical specification of the means of arranging data in rows and columns while be stored in physically persistence structure.
Data Flow	The specification of the sequence in which data moves from one state to another.
Data Owner	A role performed by an actor with the rights, competencies, and capabilities to take decisions about the aspects of data for which stewardship responsibilities have been assigned.
Data Rule	Criteria used in the process of determining or verifying values of data or generalizes certain features of data.
Data Compliance (incl. Security)	The means of adhering to and verifying adherence to policies and decisions about the data.
Data Media	The matter or material used to store physical persistent data
Data Channel	A physical communication path used to requisition, provision, or deliver data.
	Composition related meta-objects
Goal (e.g. business, application, technology)	A desired result considered a part of the direction, aims, targets, and aspirations of the enterprise
Quality	Description of a state of excellence or worth, specifying the essential and distinguishing individual nature and the attributes based on the intended use.

Data Meta-object	Decomposition related meta-objects-
Risk	The combined impact of any condition or events, including that cause by uncertainty, change, hazards, or other factors that can affect the potential for achieving these objectives).
Security	The objects or tools that secure, make safe, and protect through measures that prevent exposure to danger or risk.
Timing	A plan, schedule, or arrangement when (something) should happen or be done or to take place.
Business Role	A part that someone or something has in a particular defined function, activity, or situation. A resource/actor may have a number of roles.
Contract	An agreement between two or more parties that establishes conditions for interaction.
Product	A result and output generated by the business. It has a combination of tangible and intangible attributes (features, functions, usage)
Business Service	The externally visible ("logical") deed or effort performed to satisfy a need or to fulfil a demand, meaningful to the environment.
Service Role	A specific set of prescribed set of expected behaviour and rights (authority to act) that is meant to enable its holder to successfully carry out his or her responsibilities in the delivery of value. Each role represents a set of allowable actions within the organization in terms of the rights that are required for the business to operate.
Process Role	A specific set of prescribed set of expected behaviour and rights (authority to act) that is meant to enable its holder to successfully carry out his or her responsibilities in the performance of work. Each role represents a set of allowable actions within the organization in terms of the rights that are required for the business to operate.
Physical Application Component	A deployable part of a software product, providing identifiable functions and existing within a specific version of the product.
Application Task	The automated behaviour of a process activity performed by an application.
Application Service	An externally visible unit of functionality, provided by one or more components, exposed through well-defined interfaces, and meaningful to the environment.
Application Role	A role performed by an actor with the rights, competencies, and capabilities to take decisions about an application, its behaviour, and properties.
System Measurement	Measures that are defined and implementable within an application.

Figure 6 - The 30 LEAD data meta-objects

Data Templates (engineering) and Artefacts (architecture)

The Data Reference Content templates and artefacts consist of data maps, data matrices, and data models that each capture relevant data meta-objects. Each of these templates is based on a specific view and contains information germane to particular stakeholder concern to enable value identification, creation, and realization. For this, the Data Reference Content templates identify the relevant stakeholders, their requirements, and concerns. Data object descriptions and their modelling and architecture rational, the corresponding rules, architecture views, and viewpoints; each of these artefacts are built as templates to support a particular need and want.

Fully integrated and standardized data templates and artefacts enables the strategist, subject matter experts/practitioner, or architect (value or business architect) to work with the relevant data meta-objects throughout all the architectural layers (business, data, and technology). Advanced data modelling and relating the relevant objects throughout the layers is one of the strengths of the Data Reference Content. Not only are the each of the objects governed by its connection rules, but also by how and where the data templates and artefacts interlink and share common objects.

LEAD Templates & LEAD Meta Object Relations: Data

DATA META-OBJECTS	Requirement (Rq)	Balanced Scorecard (BSC)	Performance (Pe)	Measurement & Reporting (MR)	Information (I)	Role (Ro)	Rule (Ru)	Case (CS)	Process (P)	BPM Notations (BPMN)	Service (Se)	Application (A)	Application Interface (AI)	Application Screen (Asc)	Compliance (C)	Data (D)	Data Service (DS)	Data Rules (DR)	Platform (PL)	Infrastructure (IF)
Goal (e.g. business, application, etc.)	2					2.3			2.3		2	2				2			2	2
Timing	2	2.3	1,2,3	1,2,3	2		2	2,3	2	2	2	2			2	2	2	2	2	2
Quality	2		2	2.3			2	1,2,3							2	2		2		
Risk	2		2	2,3	2		2,3	2	2		2				1,2,3	2			2	2
Security							2,3						2		1,2,3	2		2,3	2	2
Business Roles	1,2,3					1.2		2	2		2					1			1	
Location	1.2	2	1.2	1,2,3	2.3	1,2,3	2.3		1.2	3	2	2			1.2	2	2	2	2	1,2,3
Contract	1.2		2	2.3	2	2						2			2	2				
Product	1.2	2.3	2.3	2.3	2	2					1,2,3	2			2	2				2
Business Service	1.2						2.3	2		2.3	1,2,3					2	1.2			
Service Roles	1.2					1.2		2	2		1.2					1			1	
Process Roles	2.3					1.2			2.3	2.3	1.2					1			1	
Physical Application Component	2.3											1				2.3			2	
Application Task	1,2,3									2.3		1.2		1		2				
Application Service	1,2,3						2.3			2.3	2.3	1,2,3	1,2,3	1.3	2	2	1.2		2	
Application Roles	1.2					1.2			2		1.2			1		2			1	
Information Object	2.3				1,2,3		2.3			2.3			2.3			2				
Data Component	2.3															1.3		1.2	2	
Dat Object	2.3				1,2,3		2.3	2		2.3			2.3			1.2		1.2	2	
Data Entity	2.3				1											1.2		2		
Data Table	2.3				2.3									2		1.2		2		
Data Service	1,2,3									2.3	2.3		1,2,3			1.2	1,2,3	2.3	2.3	
Data Owner	1,2,3		2.3	2.3												2		1,2,3		
Data Rules	2.3						1,2,3		2.3	2.3					1.2	2		1,2,3		
Data Compliance (incl. Security)	2.3				3		2.3		2.3						1.2	2.3		1,2,3		
Data Media	2.3												2			1		2	2	
Data Channel	2.3															1		2	2	

LEADing Practice Data Reference Content ((#LEAD-ES20012BC)   Legend: 1 = Map 2 = Matrix 3 = Model

Figure 7 - The data objects and their Maps, Matrices, & Models

In a map, the information is set out the form of a list and are a representation of the decomposed data objects, while the matrices are the continuity of and interconnection between a map (a representation of decomposed objects) and a representation of interconnected and related objects. Models often show the graphical representation of the relations and connections. The maps, matrices, and models are used in the decomposition and composition work within and throughout the layers. The specific templates and artefacts do not only show which objects are within what template, specifying if it is a map, matrix or model, it furthermore shows where the object of one template can be reused in another template.

A map, which is a simple list and shows a list of objects that is known, may be used within its row, to link to a matrix. The matrix will in turn link the object in the row to the object in the column, allowing the analyst link something that is known, to something that was unknown but now make the connection, and thus allow work to occur on a new map within a new row. It is only through the sequential utilization of the maps and matrices that the objects and therefore the parts of a problem or area or concern or interest, to be connected in a traceable manner.

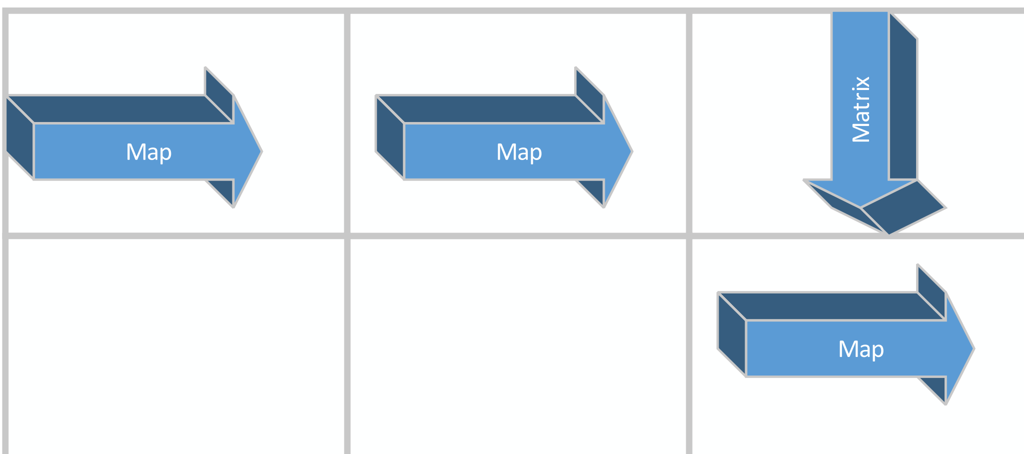

Figure 8 - Navigation through the Reference Framework

Models may logically be built out of sets of objects within the same row or the same column where the reference framework indicates that it is meaningful to relate the objects and show complex connections.

Way of Thinking about Data

As it is the basis of the formation of guiding principles for the data disciplines within the Data Reference Content the Way of Thinking is essential, its core contribution being that it provides structure for the value specification of data definitions e.g. wants, needs, goals, issues, and problems. Furthermore it can be used to postulate what ought to be, including specifying the right data abstraction level. Fundament-ally, the way of thinking does the following; it is used within analysis, appraisal, assessment, and the capture of all relevant aspects of data objects and artefacts; their idea, data design, data plan, data scheme, and data structure. This is done to understand the underlying data concept, thought, view, vision as well as the perspective, philosophy, or belief that is represented.

The purpose of having a common way of thinking with respect to data concepts is to specify how to organize and structure the viewpoints and data objects associated with the various disciplines e.g. business data and "IT"-data, applying the concepts in a common way.

The data reference concept has proven to help organizations with some of the most common and complex advanced data principles, dilemmas, and challenges that have to be confronted.

This includes, but is not limited to:

- What data does the organisation has to manage?
- For what purpose, information, does the data have to be kept?
- Who is the owner of the data, and what should be its quality?
- How to assure the quality of the data?
- How can the data be secured (security)?

What many organizations do not realize is that there is something common within each of these areas. The common things are the data objects. We have, through research and analysis, identified the semantic relations of the various data objects, and how they can be applied within different disciplines. The relations of the data objects are built into the LEAD data templates and artefacts that follow e.g. data maps, data matrices, and/ or data models.

Usage of Data Maps

A Data Map is a list and representation of the decomposed and/or composed Data Objects. The purpose of this list is to identify and decompose an inventory of all the data in the enterprise. This list helps to understand the breadth of functionality provided by the data. It shows the decomposed data aspects (version number, data component, data module, data function, data feature, and data task.)

The Data Reference Content Architecture & Modelling Rules

The data decomposition map should capture the Key Data of the organisation and its specific Data Component, Data Object (Information/Data), Data Entity, Data Type, Data Service, Data Owner, Data User, Data Channel, and Data Media. These are captured in separate maps as described below.

Each Item within the LEAD templates and within the tasks to complete the template is classified according to one of the twelve interrogatives in English; what, which, who/whom, where, where how, why, etc.

Data decomposition Map

	What/which specification:					Who is involved:		Where is it used:	
Data #	**Data Component**	**Data Object (information/data)**	**Data Entity**	**Data Type**	**Data Service**	**Data Owner**	**Data Users**	**Data Channel**	**Data Media**
#									
#									
#									

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 9 - Data with decomposed data objects

While all of the objects are relevant and necessary, this is not sufficient; all required are the tasks and rules to relate them.

The relational data modelling paradigm, the most common structural model for organizing data (others include hierarchic, network, and semantic) the central component of storage is a subject occurrence, or instance, which is a record, within a subject, which is in turn expressed as an entity and stored within a data base management system as a table. In its purest form the subject areas with a relational data structure are sets of attributes that make up a subject and contain no repeating concepts within a particular subject. The strength of the relational data model is that in minimized the use of disk space (a major issue when it was developed), its greatest weakness is that it is fragile and difficult and expensive to change. In the relational model, a data entity of a particular type can represent any one collection of items of interest to the business: people, places, things, events, or concepts. Examples of data entities in an order entry system would typically include such subjects as “Customer”, “Address”, “Order”, “Order Item” (the, otherwise repeating items in the Order, separated to be expressed as a subject onto themselves), and “Tax”. When developing the list it is critical to get its membership correct. During discovery or analysis work Data Entity types are often not identified, or are identified incorrectly. This can lead to replication of data, data structure, and functionality, together with the attendant costs in development and maintenance. Therefore, identifying the right data components, data entities, and the data types should be as explicit and easy to understand as possible to minimize misinterpretation and duplication.

Ideally, an entity in a relational model should be “normalized”, the data modelling world’s version of “cohesive”. A normalized entity depicts exactly one concept and a non-repeating list of properties associated with the concept. Similarly, a cohesive class represents one concept. For example, ‘customer’ and ‘order’ are clearly two different concepts; therefore, it makes sense to treat them as separate entities. This example reveals to us the relevance of ensuring not only that all relevant objects are identified, but also that they are related at classified the right way. Therefore the content of the data map that is captured should be based on enterprise modelling- and architecture rules and is related to tasks the practitioner needs to execute. Therefore for each individual column of the data map their applicable decomposition- (D), primary- (P) and secondary (S) relationship related rules as well as the related tasks are described below:

The 'what/which' specification in terms of which data component.	
Rules	(P) Data relates to Data Component
Tasks	• Identify, classify, and categorize the data components.
The 'what/which' specification in terms of which data/information object.	
Rules	(P) Data relates to Object (Data Object, Information Object)
Tasks	• Identify and categorize the objects: 1. Information objects, and 2. Data objects
The 'what/which' specification in terms of which data entity.	
Rules	(P) Data relates to Data Entity
Tasks	• Identify, classify, and categorize the data entities.
The 'what/which' specification in terms of which data type.	
Rules	(P) Data relates to Data Type
Tasks	• Identify, classify, and categorize the data types.
The 'what/which' specification in terms of which data service.	
Rules	(P) Data relates to Data Service
Tasks	• Identify, list and label the data services.
The 'who' specification in terms of which data owner is involved.	
Rules	(P) Data relates to Owner (Data Owner)
Tasks	• Associate and link the data owner(s) to the data.
The 'who' specification in terms of which data user is involved.	
Rules	(P) Data relates to Data User
Tasks	• Associate and link the data user(s) to the data.
The 'where' specification in terms of which data channel uses the data.	
Rules	(P) Data relates to Channel (Data Channel)
Tasks	• Identify, label, and categorize the data channels.

The 'where' specification in terms of which data media uses the data.	
Rules	(P) Data relates to Media (Media Channel)
Tasks	• Identify, label, and categorize the data media.

Figure 10 - How data is based on rules and relates to tasks the practitioner needs to execute

Other Data Maps

Other data maps are:

- Data Rule Map
- Data Service Map

These maps, and their specific templates and artefacts, matrices, and tasks are described in individual LEADing Practice Reference Content documents.

Way of Working with Data

The Data Way of Working is critical discipline of translating both strategic planning and effective execution. It structures the arrangement of effort and work by translating the "Way of Thinking" into a structural "Way of Working". The Way of Working organizes, classifies, aligns, arranges, quantifies, recommends, and selects the data objects and with it the relevant data template in a systemized and categorized way they need to be de-composed (broken down) or composed (related) together.

It is within the Way of Working is where one defines the best suitable technique, manner, routine, and method that will help the practitioner to ensure integrity, accuracy, and completeness of each particular task related to the rule that ensures the right data relation. The data way of working is therefore a series of phases with a collection of activities that the user of the data methods needs to follow and undertake in order to reach a specific goal/ outcome. The below specified way of working therefore structures the practitioner's techniques in applying the correct semantic principles, rules, procedures, and practices.

Usage of Data Matrices

The Data Matrices are a representation that expose the relationship between specific decomposed and composed data objects. The core idea of a data matrix is that they each contain a set of objects that have primary and direct natural relations to each other. These are always in a tabular form and the data objects that need to be related to them. The primary object exist in the rows and columns and the objects that relate them are at their intersection. This allows the data matrix to relate the unfamiliar to the familiar data objects in the different layers (composition), which represents the matrix diagram (rows and columns). These ontology and semantic based data relations have been standardized to ensure reusability and replication of success in outlining the right connection points that is actually based on a common relationship pattern of the data objects.

The Data Reference Content Architecture & Modelling Rules

The data matrices should capture the key data of the organisation and its related business service, application service, application task, data requirement, data goal, data rule, and data compliance. These are captured in separate matrixes as described below.

Data-Goal Matrix

This matrix shows the columns of the data map in combination with goal, the 'why' in terms of to what end or purpose or end is the data required.

Goal (which business service does the data service collaborate with?)	**Data #**	**What/which specification:**				**Who is involved:**		**Where is it used:**	
		Physical Application Component	**Data Object**	**Data Entity**	**Data Service**	**Data Owner**	**Data User**	**Channel**	**Media**
Goal 1									
Goal 2									
Goal N									

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 11 - Data with goal objects

When capturing the content of the data-goal matrix, membership should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to goal:

(Data) Goal: A desired result considered a part of the direction, aims, targets, and aspirations of the enterprise.	
Rules	(S) Data relates to Goal (Data Goal).
Tasks	• Associate and relate the goals (business, data, technology) to the data. • Associate and relate the goals (business, data, technology) to the physical application component. • Associate and relate the goals (business, data, technology) to the data objects. • Associate and relate the goals (business, data, technology) to the data entities. • Associate and relate the goals (business, data, technology) to the data services. • Associate and relate the goals (business, data, technology) to the data owner. • Associate and relate the goals (business, data, technology) to the data users. • Associate and relate the goals (business, data, technology) to the data channels. • Associate and relate the goals (business, data, technology) to the data media.

Figure 12 - How data objects relate to goal and the tasks the practitioner needs to execute

Data-Requirement Matrix

This matrix shows the columns of the data map in combination with requirement; the 'what/which' in terms of what requirement (high-level or detailed) does the data have to meet.

Data Requirement (what requirements does the data have to meet?)	**Data #**	**What/which specification:**				**Who is involved:**		**Where is it used:**	
		Physical Application Component	**Data Object**	**Data Entity**	**Data Service**	**Data Owner**	**Data User**	**Channel**	**Media**
Requirement 1									
Requirement 2									
Requirement N									

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 13 - Data with requirement objects.

The items captured within the data-requirement matrix should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to requirement:

Requirement:	
Rules	(S) Data relates to Requirement (Data Requirement).
Tasks	• Link and associate the data requirements (high-level, detailed) to the data. • Link and associate the data requirements (high-level, detailed) to the physical application component. • Link and associate the data requirements (high-level, detailed) to the data objects. • Link and associate the data requirements (high-level, detailed) to the data entities. • Link and associate the data requirements (high-level, detailed) to the data services. • Link and associate the data requirements (high-level, detailed) to the data owner. • Link and associate the data requirements (high-level, detailed) to the data users. • Link and associate the data requirements (high-level, detailed) to the data channels. • Link and associate the data requirements (high-level, detailed) to the data media.

Figure 14 - A table showing that data objects relate to requirement and the tasks associated with it.

Data-Quality Matrix

This matrix shows the columns of the data map in combination with requirement; the 'what/which' in terms of what requirement (high-level or detailed) does the data have to meet.

Quality Requirement (what quality requirements does the data have to meet?)	**Data #**	**What/which specification:**				**Who is involved:**		**Where is it used:**	
		Physical Application Component	**Data Object**	**Data Entity**	**Data Service**	**Data Owner**	**Data User**	**Channel**	**Media**
Quality Requirement 1									
Quality Requirement 2									
Quality Requirement N									

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 15 - Data with quality requirement objects

The methods for populating the data-quality matrix should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to quality:

Quality: Description of a state of excellence or worth, specifying the essential and distinguishing individual nature and the attributes.	
Rules	(S) Data relates to Quality (Quality Requirement).
Tasks	• Link and associate the quality requirements to the data. • Link and associate the quality requirements to the physical application component. • Link and associate the quality requirements to the data objects. • Link and associate the quality requirements to the data entities. • Link and associate the quality requirements to the data services. • Link and associate the quality requirements to the data owner. • Link and associate the quality requirements to the data users. • Link and associate the quality requirements to the data channels. • Link and associate the quality requirements to the data media.

Figure 16 - How data objects relate to quality and the tasks the practitioner needs to execute

Data-Risk Matrix

This matrix shows the columns of the data map in combination with risk; the 'what/which' in terms of what risk is related to the data.

Risk (which risks are relate to the data?)	**Data #**	**What/which specification:**				**Who is involved:**		**Where is it used:**	
		Physical Application Component	**Data Object**	**Data Entity**	**Data Service**	**Data Owner**	**Data User**	**Channel**	**Media**
Risk 1									
Risk 2									
Risk N									

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 17 - Data with risk objects

The content of the data-risk matrix should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to risk:

Risk: The combined impact of any condition or events, including those caused by uncertainty, change, hazards, or other factor that can affect the potential for achieving these objectives.	
Rules	(S) Data relates to Risk.
Tasks	• Link and associate the risks to the data. • Link and associate the risks to the physical application component. • Link and associate the risks to the data objects. • Link and associate the risks to the data entities. • Link and associate the risks to the data services. • Link and associate the risks to the data owner. • Link and associate the risks to the data users. • Link and associate the risks to the data channels. • Link and associate the risks to the data media.

Figure 18 – How data objects relate to risk and the tasks the practitioner needs to execute

Data-Security Matrix

This matrix shows the columns of the data map in combination with security; the 'what/which' in terms of what security measures are in place to secure the data.

Security (what security measures are in place to secure the data?)	**Data #**	**What/which specification:**				**Who is involved:**		**Where is it used:**	
		Physical Application Component	**Data Object**	**Data Entity**	**Data Service**	**Data Owner**	**Data User**	**Channel**	**Media**
Security Measure 1									
Security Measure 2									
Security Measure N									

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 19 - Data with security objects

When populating the data-security matrix, the items included should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to security:

Security: The objects or tools that secure, make safe, and protect through measures that prevent exposure to danger or risk.	
Rules	(S) Data relates to Security.
Tasks	• Link and associate the security measures to the data. • Link and associate the security measures to the physical application component. • Link and associate the security measures to the data objects. • Link and associate the security measures to the data entities. • Link and associate the security measures to the data services. • Link and associate the security measures to the data owner. • Link and associate the security measures to the data users. • Link and associate the security measures to the data channels. • Link and associate the security measures to the data media.

Figure 20 – How data objects relate to quality and the tasks the practitioner needs to execute

Data-Location Matrix

This matrix shows the columns of the data map in combination with location, the 'where' in terms of in which location is the data stored.

Location (in which location is the data stored?)	**Data #**	**What/which specification:**				**Who is involved:**		**Where is it used:**	
		Physical Application Component	**Data Object**	**Data Entity**	**Data Service**	**Data Owner**	**Data User**	**Channel**	**Media**
Location 1									
Location 2									
Location N									

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 21 - Data with location objects

When populating the data-location matrix what is captured should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to location:

Location: A facility, place, or geographic position.	
Rules	(S) Data relates to Location.
Tasks	• Associate and relate the location to the data. • Associate and relate the location to the physical application component. • Associate and relate the location to the data objects. • Associate and relate the location to the data entities. • Associate and relate the location to the data services. • Associate and relate the location to the data owner. • Associate and relate the location to the data users. • Associate and relate the location to the data channels. • Associate and relate the location to the data media.

Figure 22 - How data objects relate to location and the tasks the practitioner needs to execute

Data-Product Matrix

This matrix shows the columns of the data map in combination with product, the ‘what/which’ in terms of to which products can be related to the data.

Product (which products can be related to the data?)	**Data #**	**What/which specification:**				**Who is involved:**		**Where is it used:**	
		Physical Application Component	**Data Object**	**Data Entity**	**Data Service**	**Data Owner**	**Data User**	**Channel**	**Media**
Product 1									
Product 2									
Product N									

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 23 - Data with product objects

When populating the data-product matrix the information captured should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to product:

Product: A result and output generated by the business. It has a combination of tangible and intangible attributes (features, functions, usage).	
Rules	(S) Data relates to Product.
Tasks	• Associate and relate the product(s) to the data. • Associate and relate the product(s) to the physical application component. • Associate and relate the product(s) to the data objects. • Associate and relate the product(s) to the data entities. • Associate and relate the product(s) to the data services. • Associate and relate the product(s) to the data owner. • Associate and relate the product(s) to the data users. • Associate and relate the product(s) to the data channels. • Associate and relate the products to the data media.

Figure 24 – How data objects relate to product and the tasks the practitioner needs to execute

Data-Business Service Matrix

This matrix shows the columns of the data map in combination with business service; the 'what/which' in terms of which business service with which the data collaborates.

Business Service (which business service does the data service collaborate with?)	**Data #**	**What/which specification:**				**Who is involved:**		**Where is it used:**	
		Physical Application Component	**Data Object**	**Data Entity**	**Data Service**	**Data Owner**	**Data User**	**Channel**	**Media**
Business Service 1									
Business Service 2									
Business Service N									

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 25 - Data with business service objects

When populating the data-business service matrix the information collected should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to business service:

Business Service: The externally visible ("logical") deed or effort performed to satisfy a need or to fulfil a demand, meaningful to the environment.	
Rules	(S) Data relates to Service (Business Service).
Tasks	• Associate and relate the business service(s) to the data. • Associate and relate the business service(s) to the physical application component. • Associate and relate the business service(s) to the data objects. • Associate and relate the business service(s) to the data entities. • Associate and relate the business service(s) to the data services. • Associate and relate the business service(s) to the data owner. • Associate and relate the business service(s) to the data users. • Associate and relate the business service(s) to the data channels. • Associate and relate the business service(s) to the data media.

Figure 26 - How data objects relate to business service and the tasks the practitioner needs to execute

Data-Application Service Matrix

This matrix shows the columns of the data map in combination with application service, the 'what/which' in terms of which application service with which the data service collaborates.

Application Service (which application service does the data service collaborate with?)	**Data #**	**What/which specification:**				**Who is involved:**		**Where is it used:**	
		Physical Application Component	**Data Object**	**Data Entity**	**Data Service**	**Data Owner**	**Data User**	**Channel**	**Media**
Application Service 1									
Application Service 2									
Application Service N									

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 27 - Data with application service objects

When populating the data-application service matrix the information captured should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to application service:

Application Service: An externally visible unit of functionality, provided by one or more components, exposed through well-defined interfaces, and meaningful to the environment.	
Rules	(S) Data relates to Application (Application Service)
Tasks	• Associate and relate the application service(s) to the data. • Associate and relate the application service(s) to the physical application component. • Associate and relate the application service(s) to the data objects. • Associate and relate the application service(s) to the data entities. • Associate and relate the application service(s) to the data services. • Associate and relate the application service(s) to the data owner. • Associate and relate the application service(s) to the data users. • Associate and relate the application service(s) to the data channels. • Associate and relate the application service(s) to the data media.

Figure 28 – How data objects relate to application service and the tasks the practitioner needs to execute

While we have illustrated the relationship between data services and business services as well as application services, we realize that date services also can relate to platform services, the overall Service flow and thus to the entire Service Construct. For more information of on the specific relations between data services and service orientation please use the Data Service Reference Content with the ID#LEAD-ES20012BC

Data-Application Task Matrix

This matrix shows the columns of the data map in combination with application task, the 'what/which' in terms of which application task uses the data.

Application Task (which application task uses the data?)	Data #	What/which specification:				Who is involved:		Where is it used:	
		Physical Application Component	**Data Object**	**Data Entity**	**Data Service**	**Data Owner**	**Data User**	**Channel**	**Media**
Application Task 1									
Application Task 2									
Application Task N									

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 29 - Data with application task objects.

When populating the data-application task matrix the information captured should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to application task:

Application Task: The automated behaviour of a process activity performed by an application.	
Rules	(S) Data relates to Application (Application Task).
Tasks	• Associate and relate the application task(s) to the data. • Associate and relate the application task(s) to the physical application component. • Associate and relate the application task(s) to the data objects. • Associate and relate the application task(s) to the data entities. • Associate and relate the application task(s) to the data services. • Associate and relate the application task(s) to the data owner. • Associate and relate the application task(s) to the data users. • Associate and relate the application task(s) to the data channels. • Associate and relate the application task(s) to the data media.

Figure 30 - How data objects relate to application task and the tasks the practitioner needs to execute

Data-Physical Application Component Matrix

This matrix shows the columns of the data map in combination with physical application component, the 'what/which' in terms of which physical application component uses the data.

Physical application component (which physical application component uses the data?)	Data #	What/which specification:				Who is involved:		Where is it used:	
		Physical Application Component	Data Object	Data Entity	Data Service	Data Owner	Data User	Channel	Media
Physical application component 1									
Physical application component 2									
Physical application component N									

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 31 - Data with physical application component objects

When populating the data-physical application component matrix the information captured should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute, as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to physical application component:

Physical application component: A deployable part of a software product, providing identifiable functions and existing within a specific version of the product.	
Rules	(S) Data relates to Application (Physical application component).
Tasks	• Associate and relate the physical application component(s) to the data. • Associate and relate the physical application component(s) to the physical application component. • Associate and relate the physical application component(s) to the data objects. • Associate and relate the physical application component(s) to the data entities. • Associate and relate the physical application component(s) to the data services. • Associate and relate the physical application component(s) to the data owner. • Associate and relate the physical application component(s) to the data users. • Associate and relate the physical application component(s) to the data channels. • Associate and relate the physical application component(s) to the data media.

Figure 32 - How data objects relate to physical application component and the tasks the practitioner needs to execute

Data-Information Object Matrix

This matrix shows the columns of the data map in combination with information object, the 'why' in terms of to what information end or purpose or end is the data required.

Information Object (which information object purpose does the data serve?)	Data #	What/which specification:				Who is involved:		Where is it used:	
		Physical Application Component	**Data Object**	**Data Entity**	**Data Service**	**Data Owner**	**Data User**	**Channel**	**Media**
Information Object 1									
Information Object 2									
Information Object N									

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 33 - Data with 'information object' objects

When populating the data-information object matrix the information captured should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute, as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to information object:

Object (Business & Information): A real world thing of use by or which exists within the enterprise and information objects reveal only their interface, which consists of a set of clearly defined relations. In the context of the business competency, the relevant objects are only those that relate to the enterprise's means to act.	
Rules	(S) Data relates to Objects (Information Object).
Tasks	• Associate and relate the information object(s) to the data. • Associate and relate the information object(s) to the physical application component. • Associate and relate the information object(s) to the data objects. • Associate and relate the information object(s) to the data entities. • Associate and relate the information object(s) to the data services. • Associate and relate the information object(s) to the data owner. • Associate and relate the information object(s) to the data users. • Associate and relate the information object(s) to the data channels. • Associate and relate the information object(s) to the data media.

Figure 34 - How data objects relate to information objects and the tasks the practitioner needs to execute

Data-Rule Matrix

This matrix shows the columns of the data map in combination with rule; the 'what' in terms of what rule governs the data.

Data Rule (what rule governs the data?)	Data #	What/which specification:				Who is involved:		Where is it used:	
		Physical Application Component	**Data Object**	**Data Entity**	**Data Service**	**Data Owner**	**Data User**	**Channel**	**Media**
Data Rule 1									
Data Rule 2									
Data Rule N									

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 35 - Data with rule objects

When populating the data-rule matrix the information captured should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute, as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to rule:

Data Rule: Criteria used in the process of determining or verifying values of data or generalises certain features of data.	
Rules	(S) Data relates to Rule (Data Rule).
Tasks	• Associate and relate the data rule(s) to the data. • Associate and relate the data rule(s) to the physical application component. • Associate and relate the data rule(s) to the data objects. • Associate and relate the data rule(s) to the data entities. • Associate and relate the data rule(s) to the data services. • Associate and relate the data rule(s) to the data owner. • Associate and relate the data rule(s) to the data users. • Associate and relate the data rule(s) to the data channels. • Associate and relate the data rule(s) to the data media.

Figure 36 - How data objects relate to rule and the tasks the practitioner needs to execute

It is important to note that besides the relationship between the data objects and data rules, also the relationship between Data Rules and other rules could be relevant. For example, the data rule has to adhere to specific Business Rules, Service Rules, Application Rules, and Platform Rules. In addition, the data rule has to comply with specific Data Compliance aspects.

The following rules modelling aspects could be relevant for data rules:

1. Business Rules
2. Service Rules
3. Application Rules
4. Platform Rules
5. Data Components
6. Data Services
7. Data Owners
8. Data Compliance
9. Data Type
10. Data Flows

Each of these is based on a specific rule view, with particular stakeholder concern to enable their specific way of evaluating if the specific rule has been meet or not. Below are examples of how they can be related together:[10]

Data Rule-Business Rule Matrix

This matrix shows the data rules in combination with business rules, which business rule does the data rule have to adhere to:

Business Rule (which business rule does the data rule have to adhere to?)	What/which specification:						
	Data Rule #	**Data Rule #**	**Data Rule #**	**Data Rule #**	**Data Rule #**	**Data Rule #**	**Data Rule #**
Business Rule 1							
Business Rule 2							
Business Rule N							

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 37 - Data rule with business rule objects

[10] For more informaiton see the Data Rule Reference Content

When populating data-goal matrix the information captured should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute, as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to business rule:

Business Rule: A statement that defines or constrains some aspect of behaviour within the business and always resolves to either true or false.	
Rules	(S) Data rule relates to Business Rule.
Tasks	• Link and relates business rules to data rules.

Figure 38 - How data rule relates to business rule and the tasks the practitioner needs to execute

Data Rule-Service Rule Matrix

This matrix shows the columns of the data map in combination with service rules to indicate which the data rule the service rule has to adhere.

Service Rule (which service rule does the data rule have to adhere to?)	**What/which specification:**						
	Data Rule #	**Data Rule #**	**Data Rule #**	**Data Rule #**	**Data Rule #**	**Data Rule #**	**Data Rule #**
Service Rule 1							
Service Rule 2							
Service Rule N							

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 39 - Data rule with service rule objects.

The process to capture of the content of data-goal matrix should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute, as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to service rule:

Service Rule: A statement that defines or constrains some aspect of the creation of value within the enterprise.	
Rules	(S) Data Rule relates to Service Rule.
Tasks	• Link and relate service rules to data rules.

Figure 40 - How data rule objects relate to service rule and the tasks the practitioner needs to execute

Data Rule-Application Rule Matrix

This matrix shows the columns of the data map in combination with application rule, to indicate which application rule the data rule has to adhere to:

Application Rule (which application rule does the data rule have to adhere to?)	**What/which specification:**						
	Data Rule #	**Data Rule #**	**Data Rule #**	**Data Rule #**	**Data Rule #**	**Data Rule #**	**Data Rule #**
Application Rule 1							
Application Rule 2							
Application Rule N							

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 41 - Data rule with application rule objects

When populating the data-requirement matrix the information captured should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute, as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to application rule:

Application Rule: A business rule implemented within and able to be executed by an application.	
Rules	(S) Data rule relates to Application Rule.
Tasks	• Link and relate application rules to data rules.

Figure 42 – How data objects relate to application rule and the tasks the practitioner needs to execute

Data Rule-Platform Rule Matrix

This matrix shows the columns of the data map in combination with platform rule, to indicate which platform rule the data rule has to adhere.

Platform Rule (which platform rule does the data rule have to adhere to?)	What/which specification:						
	Data Rule #	Data Rule #	Data Rule #	Data Rule #	Data Rule #	Data Rule #	Data Rule #
Platform Rule 1							
Platform Rule 2							
Platform Rule N							

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 43 - Data rule with platform rule objects

When populating the data-rule matrix the information captured should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute, as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to platform rule:

Platform Rule: Criteria used in the process of determining the behaviour of the platform.	
Rules	(S) Data Rule relates to Platform Rule.
Tasks	• Link and relate platform rules to the data rules.

Figure 44 - A table showing that data objects relate to platform rule and the tasks associated with it

Data Rule-Data Compliance Matrix

This matrix shows the columns of the data map in combination with data compliance show that data rule with which it must comply.

Data Compliance (what does the data rule have to comply with)	**What/which specification:**						
	Data Rule #	**Data Rule #**	**Data Rule #**	**Data Rule #**	**Data Rule #**	**Data Rule #**	**Data Rule #**
Data Compliance 1							
Data Compliance 2							
Data Compliance N							

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 45 - Data rule with data compliance objects

When populating the data-goal matrix the information captured should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute, as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to data flow:

Data Compliance: The means of adhering to and verifying adherence to policies and decisions about the platform.	
Rules	(S) Data rule relates to Data Compliance.
Tasks	• Associate and connect data compliance to data rules.

Figure 46 – How data rule relates to data compliance and the tasks the practitioner needs to execute

Data Rule-Data Entity Matrix

This matrix shows the columns of the data map in combination with data entity, which data entities are related to the data rule. With this relation the other data related relations can easily be derived as well, e.g. data component, data table, data owner.

Data Entity (which data entities are related to the data rule)	**What/which specification:**						
	Data Rule #	**Data Rule #**	**Data Rule #**	**Data Rule #**	**Data Rule #**	**Data Rule #**	**Data Rule #**
Data Entity 1							
Data Entity 2							
Data Entity N							

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 47 - Data rule with data entity objects

When populating the data-entity matrix the information captured should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute, as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to data entity:

Data Entity: An encapsulation of data. Logical data entities are specification of the organization of the information to store the data as a physically persistence structure e.g. data tied to applications, repositories, and services.	
Rules	(S) Data rule relates to Data Entity.
Tasks	• Associate and connect data entities to data rules.

Figure 48 - A table showing that data rule relates to data entity and the tasks associated with it.

If required in a specific situation the following matrices can be derived directly from this Data Rule-Data Entity Matrix:

- Data Rule-Data Component Matrix
- Data Rule-Data Object Matrix
- Data Rule-Data Table Matrix
- Data Rule-Data Type Matrix
- Data Rule-Data Nature
- Data Rule-Data Service Matrix
- Data Rule-Data Flow Matrix
- Data Rule-Data Owner Matrix
- Data-Rule-Data Media Matrix
- Data Rule-Data Channel Matrix

Data-Compliance Matrix

This matrix shows the columns of the data map in combination with compliance, the 'how' in terms of in what way does the data has to comply.

Data Compliance (which data compliance does the data has to adhere to)	Data #	What/which specification:				Who is involved:		Where is it used:	
		Physical Application Component	Data Object	Data Entity	Data Service	Data Owner	Data User	Channel	Media
Data Compliance 1									
Data Compliance 2									
Data Compliance N									

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 49 - Data with compliance objects

When populating the data-compliance matrix the information captured should be based on enterprise modelling- and architecture rules and is related to the tasks the practitioner needs to execute, as described under the data map. In addition to those rules and tasks, the following rules and tasks are related to compliance (including security):

Data Compliance (including Security): The means of adhering to and verifying adherence to policies and decisions about the data.	
Rules	(S) Data relates to Compliance (Data Compliance, including Security).
Tasks	• Associate and relate the data compliance (including security) to the data. • Associate and relate the data compliance (including security) to the physical application component. • Associate and relate the data compliance (including security) to the data objects. • Associate and relate the data compliance (including security) to the data entities. • Associate and relate the data compliance (including security) to the data services. • Associate and relate the data compliance (including security) to the data owner. • Associate and relate the data compliance (including security) to the data users. • Associate and relate the data compliance (including security) to the data channels. • Associate and relate the data compliance (including security) to the data media.

Figure 50 – How data objects relate to compliance (including security) and the tasks the practitioner needs to execute

Other Data Matrices

The following other data matrices are described, together with their specific templates and artefacts, maps, and tasks in their individual LEADing Practice Reference Content documents as listed:

Data Rule Matrices	LEADing Practice Data Rule REFERENCE CONTENT
Data Service Matrices	LEADing Practice Data Service REFERENCE CONTENT

Way of Modelling for Data

The Data Way of Modelling provides the means for the various practitioners working with many aspects of data to assist them through the use of the modelling principles so as to make assist in making an objective assessment of the possible data object relationships with other objects. One of the key features of the Data Way of Modelling is that it provides a uniform and strict description of the models where the data objects and artefacts within one or more different types of models can be portrayed. These models are a representation that graphically show the data relationships and the interconnection of specific composed objects that complies with a specific set of rules for what the graphical components mean, and how they are connected to the rest of the business. The key idea of a data model is that it is a representation, an illustration, of a composition of information intended to represent an aspect of an enterprise (e.g. business, data and/or technology), using a specific set of rules, which express a specific logic or grammar.

Each practitioner working with data aspects has to be able to translate the "Way of Working" into a "Way of Modelling", which for the most part include the following:

- **Expressiveness:** the degree to which a given modelling technique is able to denote the models of any number and kinds of layered domains (business, data, and technology).
- **Arbitrariness:** the degree of freedom one has when decomposing and composing different models on the same domain.
- **Suitability:** the degree to which a given modelling technique is specifically tailored for a specific kind of wanted output/result.
- **Comprehensibility:** the ease of how the way of working and way of modelling techniques are understood by participants.
- **Coherence:** the degree to which the individual sub-models of a way of modelling constitute a whole.
- **Completeness:** the degree to which all-necessary concepts of the data domains are represented in the way of modelling.
- **Efficiency:** the degree to which the modelling steps (e.g. practitioner steps) use resources such as time and people.
- **Effectiveness:** the degree to which the modelling principles achieve its goals.
- **Audit:** the degree to which the results of the models achieve its goals.

As we described earlier the data matrix provides the continuity of and interconnection between a data map (a representation of decomposed and/or composed objects) and a data model (a representation of interconnected and related objects).

Based on already acquired information from the data maps and/or a data matrices (or both), a data model may be crafted to enable complex information to be used in different disciplines and to be communicated more easily to stakeholders, management, and leadership. The fully integrated and standardized data templates and artefacts enable the practitioner to work and model with the data objects throughout all the aspects of the enterprise (business, data, and technology). In this way, not only is each object governed by its semantic relations and connections, but also by the relevant data modelling rules and tasks, ensuring how and where the data templates and artefacts interlink and share common data objects is defined and standardized.

Using the LEAD data templates and artefacts to manage the different kinds of highly connected information and relations ensures the structured formation of data into its larger context. The data map (which lists the various related objects in order to capture the decomposed unrelated objects) is vital. This is also true of the data matrix (which composes in terms of relating specific objects together) and the data model (which graphically represents the decomposed and composed objects). Both are critical in integrating and standardizing the data templates and artefacts and tools of the practitioner. Collectively these are an essential part of supporting, integrating, and standardizing the practitioner's Way of Thinking, Working, and Modelling. The result then for an organization that uses the data way of modelling templates and artefacts is that it is brought to the highest maturity possible of working not only documented (level 3) or managed (level 4) but enabling optimization, sound governance, and continuous improvement (level 5).

Data Decomposition and Composition Model

The Data Decomposition and Composition Model seen previously in shows the sixteen main areas that provide a starting point that can be used the analysis, decomposition, composition, and construction of a business architecture and/ or the description of is software and system architecture.

Data modelling is the act of exploring data-oriented structures. Like other modelling artefacts, data models can be range from high-level conceptual view of the data to physical, implementation-oriented specification of data structure. The models may be used for a variety of purposes, indeed, there are many types of data models, too many to describe them all in this document. Today the most common type of data model is based on the relational paradigm with it associated normalization[11]. These models show the tables, or there logical or conceptual equivalent, the entities and may include their keys (their unique identifiers) and their data fields or attributes. **Error! Reference source not found.** below shows an example of such a data model.

[12]. According to Whitten,[13] data modelling may be performed during various types of information projects and in multiple phases of information projects. He determined two major types of data modelling:

- Strategic data modelling: This is part of the creation of an information strategy, which defines an overall vision and architecture for information systems is defined. Information engineering, modelling, and information architecture is therefore relevant to consider within conceptual data modelling.

---

[11] http://web.archive.org/web/20080805014412/http://www.datamodel.org/NormalizationRules.html
[12] Matthew West and Julian Fowler (1999). Developing High Quality Data Models. The European Process Industries STEP Technical Liaison Executive (EPISTLE).
[13] Whitten, Jeffrey L.; Lonnie D. Bentley, Kevin C. Dittman. (2004). Systems Analysis and Design Methods. 6th edition. ISBN 0-256-19906-X.

- Data modelling during information systems analysis: In information systems analysis, logical data models are created as part of the development of new databases.

Data modelling is also used as a technique for detailing information requirements for specific databases.

From the perspective of an object-oriented developer data modelling is conceptually similar to class modelling. With data modelling, you identify data entity types whereas with class modelling you identify data classes. Data attributes are assigned to entity types just as you would assign attributes and operations to classes. There are associations between entities, similar to the associations between classes – relationships, inheritance, composition, and aggregation are all applicable concepts in data modelling.

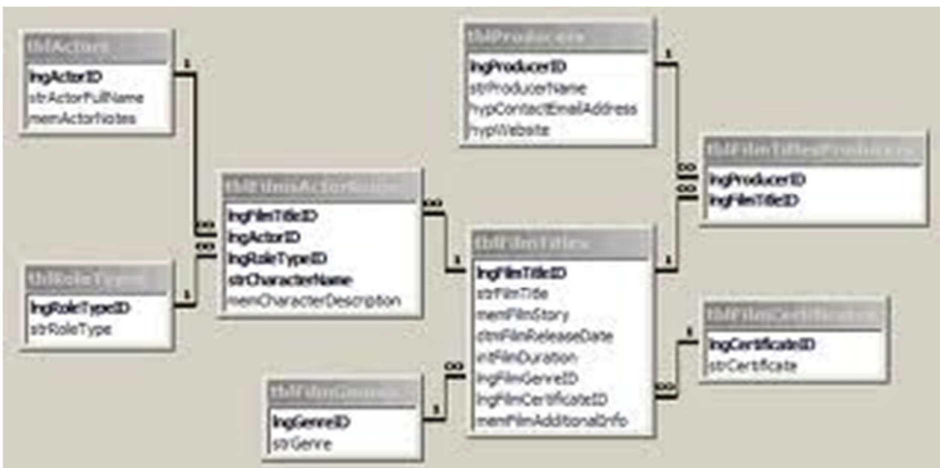

Figure 51 - Data Model example

Where traditional data modelling is different from class modelling is that the former focuses solely on data, whereas class models allow for the exploration of both the behaviour and data aspects of the domain or problem space. With a data model, it is only possible explore data issues. Because of this focus, data modellers have a tendency to be much better at getting the data “right" than object modellers. Somewhere in the middle ground some practitioners will model database methods (stored procedures, stored functions, and triggers) when they are physical data modelling. Whatever the approach, in any sort of data modelling, engineering, or data architecture it is relevant to remember that data models represent a description of the form of the data used within information systems by providing specific structure, definition, and format. If a semantically sound and coherent view of the data is created through data model that is then used consistently across information systems then integration of the data resources can be achieved. For example, if the same data structures are used to store and access data then different information systems can share data seamlessly

In data modelling it is important to consider the information model that will be using and consuming the data. In the following, we will therefore explore how data modelling is used within this context. The Information context shows the relationship between the several data models, data services, and the data flow. It shows information components and interfaces between components. Interfaces will be associated with data objects and the data entities.

Data Sequence Diagram

The typical way of illustrating the flows associated with data in a network representation would be in a Data Sequence Diagram. As they show how groups of objects work together as part of a larger system these models are one of the cornerstones for structured systems analysis and design.

These models may be used to represent the sequence of tasks within the business layer, or they may be used to show how components of a complex automated application behave. In both cases the focus of the model is on representing how actors[14] will pass information in flows between themselves in a particular order or sequence to carry out a task. To do this the diagram will show the context to the task, the relevant objects, the packages of information passed between those objects and through some indication of time, show the order in which the message packages are passed.

To place the task in context and to show do this the typical Data Sequence Diagram will use four symbols to capture how activities collaborate. The three entities that must be represented are:

- Data flows - movement of packages of data (the messages passed between a caller and sender)
- Data stores – data repositories for data that is not moving
- Actors – transforms of incoming data flow(s) to outgoing data flow(s)

The diagram may further contain:

- Sequence synchronization -
- External entities – sources or destinations outside the specified system boundary and provide context as to the system boundary

Within the business layer, the actors will be individuals or organizations, whereas within the application layer these actors may be automated processes.

An example of a Data Sequence Diagram within the business layer is shown in Figure 52, which shows a simplistic example the driver provides the police with information; the police then use this to check the Police Information System (PIS), which provides a response and the police then close off the engagement with the driver

---

[14] A specific person, system, or organization that initiates or interacts with the defined functions and activities. Actors may be internal or external to an organization.

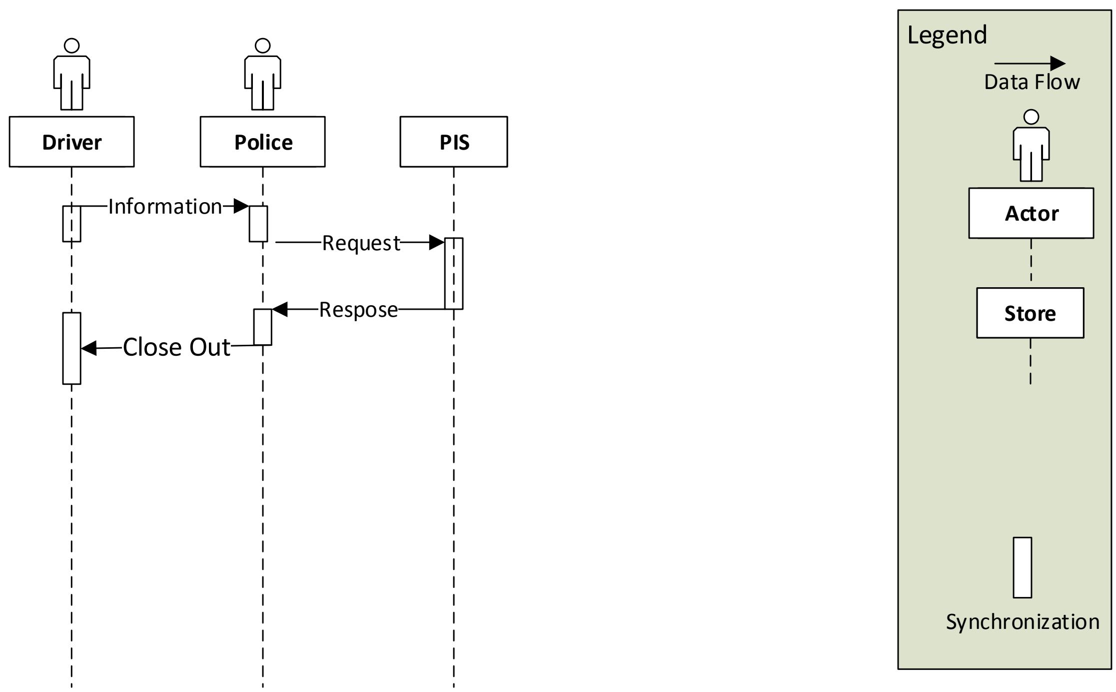

Figure 52 – Example of LEAD Data Sequence Diagram

Within the application layer, a Data Sequence Diagram can similarly show data flows. Figure 53 demonstrates such a data flow example, where it summarizes the basic data flow rules, and how to apply them in a data flow diagram. In this example the following activities collaborate:

1. The Client (or servlet, which is really just a representation of a client on the server) calls the controller (and never calls data factory directly).
2. Tnhe Controller calls the Data Factory, preferably from its own subsystem or calls other Controllers (if it needs to access data from other subsystems).
3. The Data Factory calls other Data Factories. Again, this occurs by preference from its own subsystem, but it will never call a Controller.

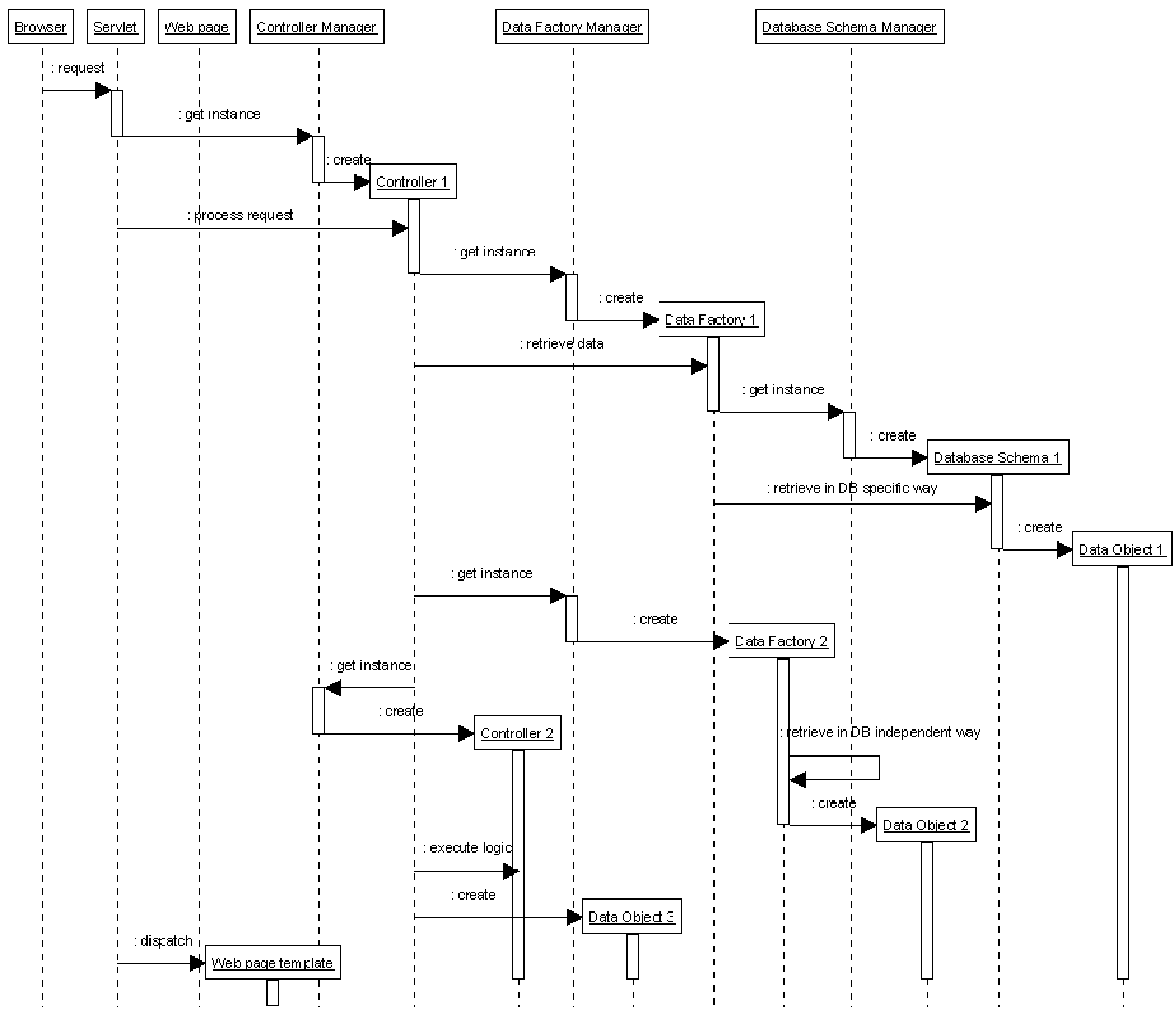

Figure 53 - Example of a Data Flow diagram of the systems development process

This example shows a general case of a three-tier application architecture whereby, by employing some basic rules to separate responsibility on these three tiers (client, controller, and data factory), the flow of the data and responsibilities of each layer in the application will be clear and straightforward to implement.

Data Sequence Diagrams do not show decisions or specific timing of events. Their function is to illustrate data sources, destinations, flows, stores, and transformations. The capabilities of data sequence diagramming align directly with general definitions of systems, whether automated or manual. Data sequence diagrams are an implementation of a method for representing systems concepts including boundaries, input/outputs, processes/sub processes, etc. The data sequence diagram is analogous to a road map. It is a network model of all possibilities with different detail shown on different hierarchical levels.

Within Data Sequence Diagrams, the level of detail required to achieve the necessary specificity may vary. It is quite appropriate to representing different levels of detail and different partitions depending on the problem being addressed.

Unfortunately, past practice has led to data structures for different information systems that are often arbitrarily different. The result of this is that complex interfaces are required between systems that share data. These interfaces can account for between 25-70% of the cost of current systems. For this reason required interfaces should be considered inherently while designing a data model, as a data model on its own would not be usable without interfaces within different information systems. Applying a standard Data Sequence Diagram, working at the same level can reduce the complexity of the interfaces, reduce issues associated with application interoperability, and improve information flow.

Information Distribution model

The Information Distribution Model is developed by applying the following architectural modelling principles:

- Logical Application Components may be defined within other Logical Applications.
- The Physical Application Component will be part of a Software Product Version.
- The Physical Application Component contains one or more Application Modules.
- An Application Module consists of one or more Application Functions.
- An Application Function is decomposed into Application Tasks.
- Application Tasks call Information Interfaces.
- An Application Service is delivered through an Information Interface.
- Information Measurements come and relate to business measurements e.g. Key Performance Indicators (KPIs), Process Performance Indicators (PPIs), and Service Performance Indicators (SPIs).
- Application System Measurement exposes measurements' result to Object(s) e.g. business, information, and data.
- System Measurement is captured or presented in a System Report.
- System Report exposes Data Object.
- System Flow orchestrates System Report.
- An Application Interface exposes an Application Task.
- The content of an Application Interface is delivered through a Application Service in the form of either a Data or Information Services.
- An Information Interface consists of data services

As seen in Figure 53, the Information Distribution Model therefore captures the Application Components, Application Modules such as CRM, Finance (FI), Logistics, etc., as well as the reporting flows as well as the various interfaces (numbering them).

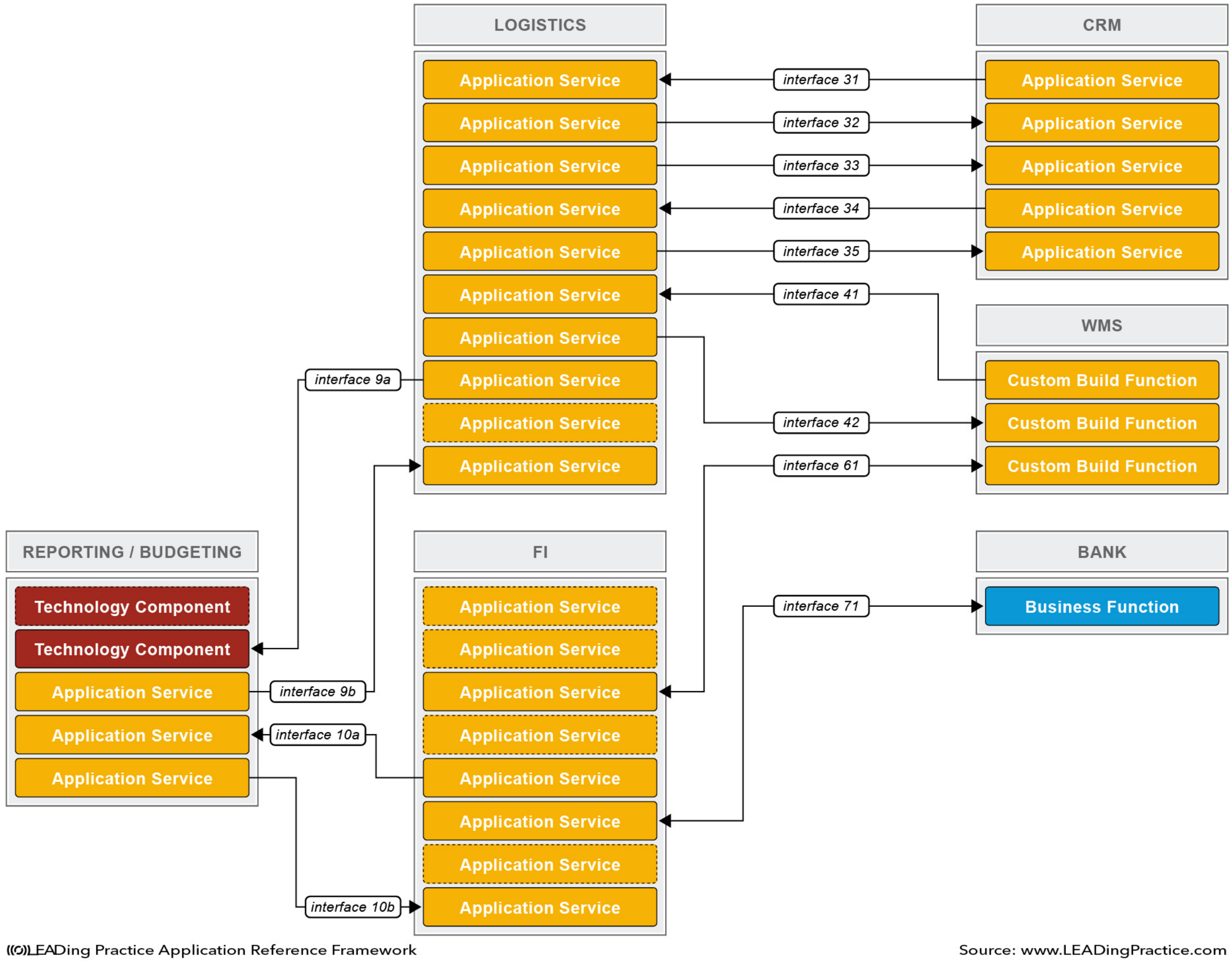

Figure 54 - Example of an Information distribution Model

Information Service Model

The Information Service Model shows the relationship between the Information Services, and the different data flows, based on the Process Flow.

Usage: The purpose of the Information Service Model is to depict clearly the system flow sequence of events when multiple information services are involved in executing a business process flow. It enhances the Information Model by augmenting it with any sequencing constraints, and hand-off points between batch and real-time processing. It would identify complex sequences and identify possible points in the architecture in order to provide information to business users.

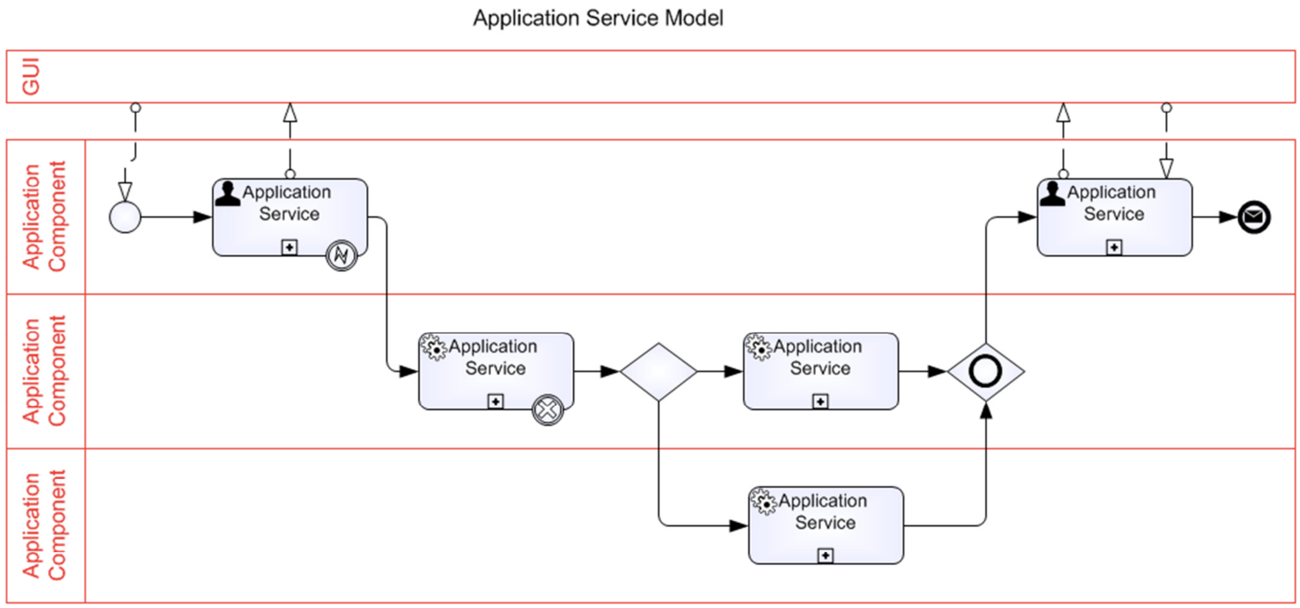

Figure 55 - Example Information Service Model

The Information Service Model is developed by applying the following architectural modelling principles:

- An Information Service calls Information Task(s)
- An Information Service exposes Information Function(s)
- An Information Function is part of Information Module (Logical Information Component)
- An Information Function contains System Flow(s)
- System flow orchestrates Information Service(s)
- System flow adheres to Information Rules
- System flow matches Process Flow (including Gateways & Events)
- Information Role consumes Information Service

Information Interface Model

The Information Interface Model shows in a UML sequence diagram the relationship between Information Service, Data Object, Data and Information Channel, Data, and Information Media.

Usage: This model depicts the information data exchange sequence through several interfaces.

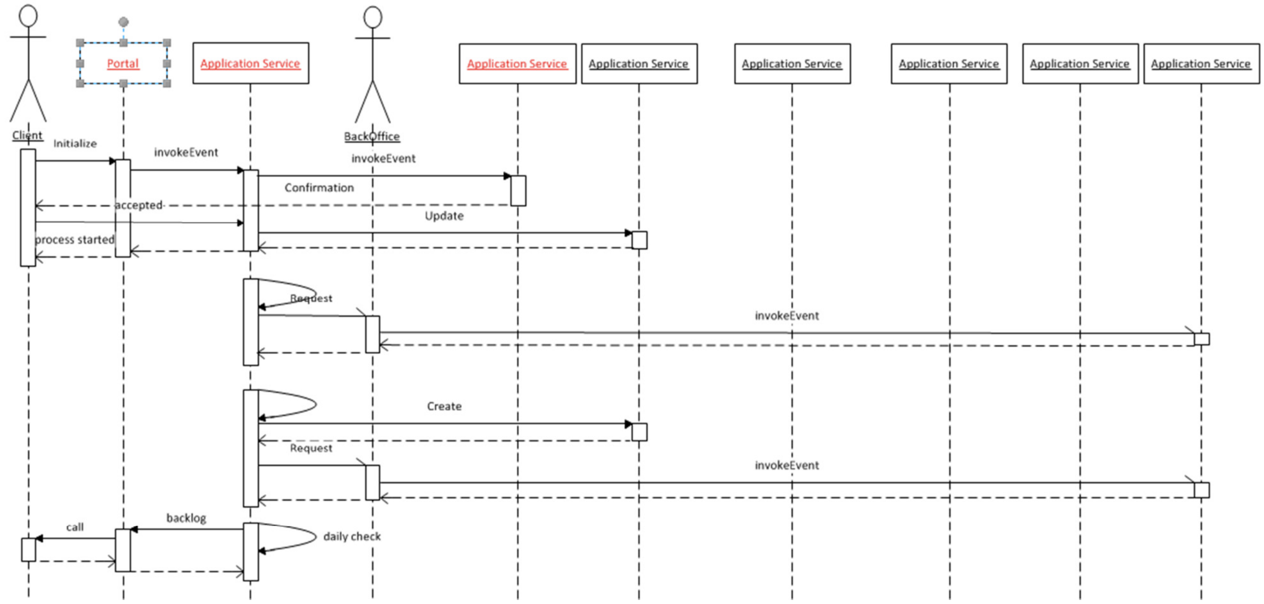

Figure 56 - Example of LEAD Information Interface Model

The Information Interface Model is created by applying the following architectural modelling principles:

- An Information Interface consists of Information Service(s)
- An Information Interface transports Information Object(s)
- An Information Interface aggregates Data Service(s)
- An Information Interface is a logical connection between Logical Information Components (Modules)
- An Information Interface is orchestrated by System flow(s
- An Information Interface adheres to Information Rules
- An Information Interface contains System Flow(s))

Information Interface Map

The Information Interface Map shows the content of the interface and the interaction (input and output) of an Information service through Information channel and/or Information Media mapped against the calling information task. Some of the most common interfaces an application user will experience are application screens and reports, but these may also be machine-to-machine interfaces or even within a purely manual environment.

Usage: The purpose of the Information Interface Map is to scope and document the interface and the external behaviour that exists outside the system of concern (input and output) of an Information Service.

The Information Interface Map completed by applying the following architectural modelling principles:

- An Information Interface (user) interfaces Information Service(s)
- An Information Interface is part of Information Media(s)
- An Information Interface is provided by Information Channel(s)
- An Information Interface is assigned to Information Role(s)
- An Information Interface channels input and/or output by calling Information Task(s)

Interface #	What/Which specification:					Who/Whom specification:
	Interface Name	Interface Description	Information Service	Information Task	Input or Output	Information Role

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 57 - Example of an LEAD Information Interface Map

Information/Platform Service Matrix

The Information/Platform Service Matrix shows the relationship of Platform Service with Information Function, Information Service, Data Service, and Infrastructure Service.

Usage: The purpose of the Information/Platform Service Matrix is to depict the relationship between Platform Services with Information Function, Information Service, Data Service, and Infrastructure Service within the enterprise.

	What specification:			
	Platform Service #	Platform Service #	Platform Service #	Platform Service #
Business Service (which business service uses the platform service)				
Information Service (which information service uses the platform service)				
Information Task (which information task uses the platform service)				
Data Service (which data service is supported by the platform service)				

A part of the LEADing Practice Modelling and Architecture Principles and Templates

Figure 58 - Template of LEAD Platform Service Matrix

The Platform Service Matrix completed by applying the following architectural modelling principles:

- An Information Function is deployed by Platform Service(s)
- An Information Service is supported by Platform Service(s)
- A Platform Service enables Data Service(s)
- A Platform Service is supported by Infrastructure Service(s)

(System) Measurement/Reporting Map

The (System) Measurement/Reporting Map captures a list of System Measurements related to Information Functions or System Reports.

Usage: A list of system measurements incorporated in Information or System Reports and/or defined measurements to be realized in Information or System Reports.

Measurement #	What/Which specification:				
	Measurement Name	Measurement Definition	Rationale	Dimension	Information Function/System Report

Figure 59 - Template of LEAD System Measurement/Reporting Map (Information) Role Map

The (Information) Role Map captures a list of Information Roles with description, to be related to Information Tasks or Information Services.

Usage: An inventory of available or required Information roles with linked authorizations, admissions, and permissions.

Role #	Who/Whom specification:	What/Which specification:	
	Information Role	Role Description	Rationale
#			
#			
#			
#			
#			

Figure 60 - Template of LEAD Role Map

(Information) Rule Map

The (Information) Rule Map captures a list of Information Rules with description, rationale, nature etc.

Usage: Provides an overview of the relevant rules for a specific view of information from a particular view.

Rule #	What/Which specification:			
	Information Rule	**Rule Description**	**Rationale**	**Nature**
#				
#				
#				
#				
#				
#				

Figure 61 - Template of LEAD Rule Map

(Information) Compliance Map

The (Information) Compliance Map captures a list Information Rules, which must be compliant.

Usage: Provides an overview of all compliancy related Information Rules.

Compliance #	What/Which specification:			
	Information Compliance	**Information Rule**	**Compliance Description**	**Rationale**
#				
#				
#				
#				
#				

Figure 62 - Template of LEAD Compliance Map

(Information) Role Matrix

The (Information) Role Matrix shows the relationship of Information Roles with Business Roles, Information Services, and/or Tasks.

Usage: Gives the association of Information Roles to Business Roles, and vice versa to execute Information Tasks and Services.

Information Role, #	**Role #**	**Who/Whom specification:**	**What/Which specification:**		**Who/Whom specification:**	**What/Which specification:**	
		<Business Role, #>	**<Information Task, #>**	**<Information Service, #>**	**<Business Role, #>**	**<Information Task, #>**	**<Information Service, #>**
<Information Role Name>	#						
<Information Role Name>	#						
<Information Role Name>	#						

Figure 63: Template of LEAD Role Matrix

The (Information) Role Matrix is completed by applying the following architectural modelling principles:

- An Information Role is related to Business Role(s)
- An Information Role consumes Information Service(s)
- An Information Role executes Information Task(s)

(Information) Rule Matrix

The (Information) Rule Matrix shows the relationship of Information Rules with Information Functions and System Flow.

Usage: Defines the implementation of Information Rules within Information Functions and System Flow.

Information Rule, #	Rule #	What/Which specification:			
		<Business Rule, #>	**<Information Function, #>**	**<Business Rule, #>**	**<Information Function, #>**
<Information Rule Name>	#				
<Information Rule Name>	#				
<Information Rule Name>	#				
<Information Rule Name>	#				

Figure 64 -Template of LEAD Rules Matrix

The (Information) Rules Matrix is completed by applying the following architectural modelling principles:

- An Information Rule bounds an Information Function
- An Information Rule is subject to Business Rule(s)

(Information) Compliance Matrix

The (Information) Compliance Matrix shows the relationship of Information Compliance with Information Service(s), Information Rules, System Measurements, and/or System Owners.

Usage: Provides insight of implemented Information Compliancy, measurement of relevant Information Rules and reporting to System Owners.

Information Compliance, #	Compliance #	What/Which specification:			Who/Whom specification:
		<Information Rule, #>	**<Information Service, #>**	**<System Measurement, #>**	**<System Owner, #>**
<Information Compliance Name>	#				

Figure 65 - Example of LEAD Compliance Matrix

The (Information) Compliance Matrix is completed by applying the following architectural modelling principles:

- Information Compliance shows the conformation to the rules of Information Service(s)
- Information Compliance is source to Information Rule(s)
- Information Compliance can be measured by System Measurement(s)

- Information Compliance reports to System Owner(s)
- System Measurement is captured or presented in a System Report

(System) Measurement/Reporting Matrix

The (System) Measurement/Reporting Matrix shows the relationship of System Measurements with Performance indicators, Data Objects, Information Tasks, and System Owner.

Usage: Provides a list of all System Measurements, its relevant Information Tasks, its Data Objects, and System Owners.

The System Measurement/Reporting Matrix is completed by applying the following architectural modelling principles:

- System Measurement is measured against PPI
- System Measurement exposes measurement result to Data Object(s)
- System Report exposes Data Object
- System Measurement is reported to System owner
- System Measurement defines the measurement for Information Task(s)
- Information Compliance can be measured by System Measurement(s)
- Information Compliance reports to System Owner(s)
- System Measurement is captured or presented in a System Report

System Measurement, #	**System Measurement #**	**What/Which specification:**				**Who/Whose specification:**
		<Process Performance Indicator, #>	**<System Report Name, #>**	**<Information Task, #>**	**<Data Object, #>**	**<System Owner, #>**
<System Measurement>	#					
<System Measurement>	#					
<System Measurement>	#					
<System Measurement>	#					
<System Measurement>	#					
<System Measurement>	#					

Figure 66 - Template of LEAD System Measurement/Reporting Matrix.

(System) Measurement/Reporting Model

The (System) Measurement/Reporting Model shows the System Measurements and Measurement Reporting to the (System) Owner(s).

Usage: Illustrates the System Measurement captured or presented in System Reporting Flow to (System) Owner(s).

The System Measurement/Reporting Model is completed by applying the following architectural modelling principles:

- System Measurement is reported to the System Owner
- System Measurement exposes measurements' result to Data Object(s).
- System Report exposes Data Object
- System Measurement is captured or presented in a System Report.
- System Flow orchestrates System Report

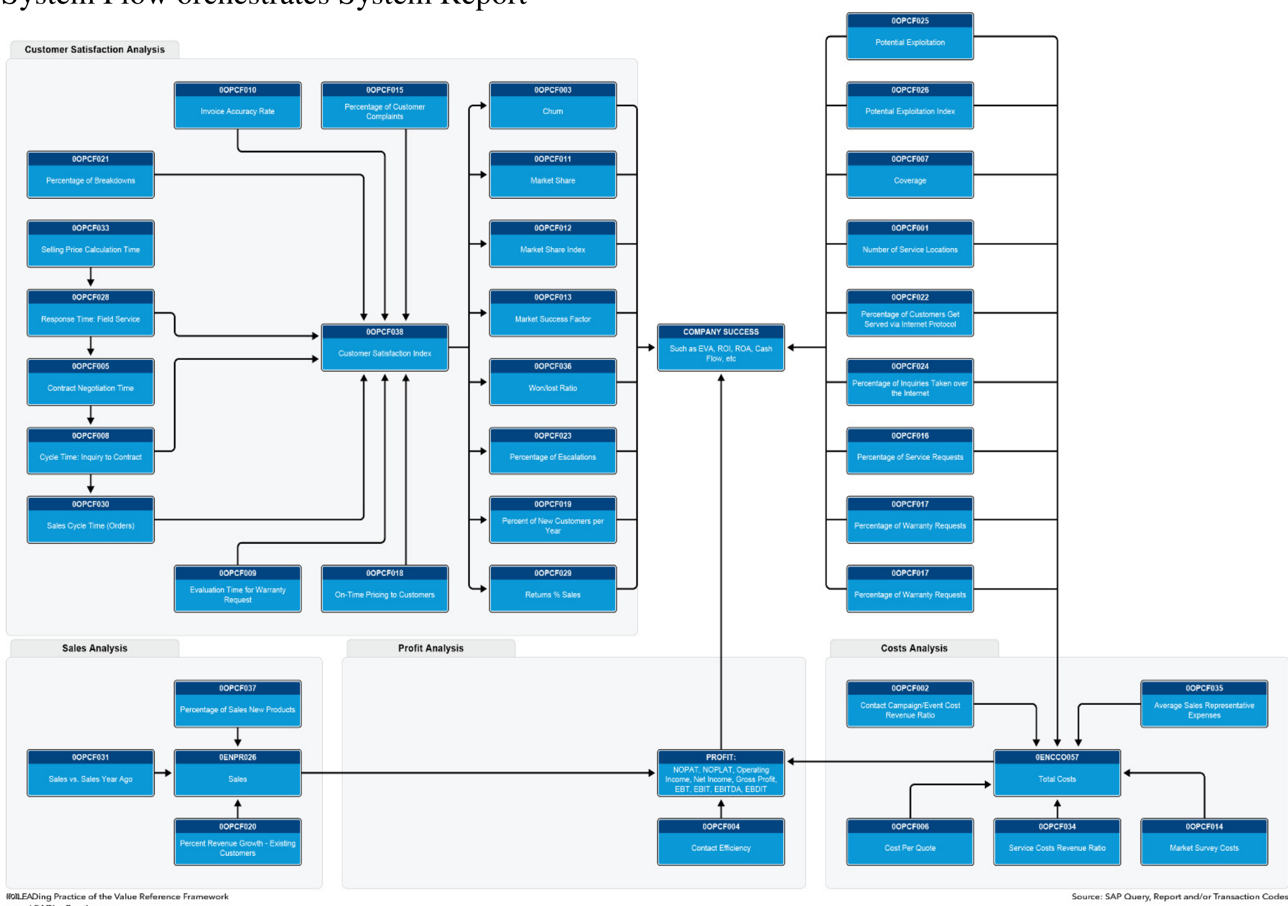

Figure 67 - Example of System Measurement/Reporting Model with the relevant areas/owners, data object queries, transaction codes and reporting relations.

(Information) Goal and Requirement Map

The Information Goal and Requirements Map captures assumptions and expectations that the Information Functions and Information Services need to meet the Business' Critical Success Factors (objectives). Information (Goals and) Requirements are typically generated from Business Process and Business Service. Requirements can also be used as a quality assurance tool to ensure that a particular architecture is fit-for-purpose (i.e., can the architecture meet all identified requirements).

The Requirements Map contains the following entities:

- Requirement (needs, wants)
- Assumption
- Constraint (specifications, information goal, etc.)
- Gap (pain, goal)

Usage: Inventory of strategic, tactical, and operational requirements to be addressed by Information Functions and delivered by Information Services

Goal/ Requirement #	What/Which specification:							Who/Whom specification:
	Objective	Expectation	Requirement, Specification or Assumption	Details	Business Process	Information Component	Information Function or Information Service	Stakeholder
#								
#								
#								
#								

Figure 68 - Template of LEAD Information Goal and Requirement Map.

The Information Goal and Requirement Map is completed by applying the following architectural modelling principles:

- Objective and Expectation sets the requirements for Information Function(s)
- Objective and Expectation sets the requirements for Information Service(s)
- Objective and Expectation sets the requirements for Information Feature(s)
- Business Process sets requirement for Information Function(s)
- Objective and Expectation is owned by a Stakeholder

Depending on the organisation the following Data Models, they can have different views and granularity. Below is an example:

- A Data Model that illustrates the connection and relationship of **data components** to data.v
- A Data Model that illustrates the connection and relationship between **physical application component(s)** and data.
- A Data Model that illustrates the connection and relationship between **data compliance** and data rules.

As we see on the above examples, data models can have different purpose and thus different granularity and views. The most common ways to view data models in practice is by three basic styles/views:

- **Conceptual data models**. These models, sometimes called domain models, are typically used to explore domain concepts with project stakeholders.
- **Logical data models** (LDMs). LDMs are used to explore the domain concepts, and their relationships, of your problem domain. This could be done for the scope of a single project or for your entire enterprise. LDMs depict the logical entity types, typically referred to simply as entity types, the data attributes describing those entities, and the relationships between the entities.
- **Physical data models** (PDMs). PDMs are used to design the internal schema of a database, depicting the data tables, the data columns of those tables, and the relationships between the tables.

Way of Governing

The Way of Governance - the word governance is derived from the Greek verb “κυβερνάω” [kubernáo], which means to steer. Plato documented its use in a metaphorical sense for the first time. It then passed on to Latin and then on to many languages. In Enterprise Architecture, governance is the system of rules, practices and processes by which an enterprise directs and controls the description and form it takes or is in the process of developing or deploying. In the Way of Governance, it relates to decisions and guidance that define expectations and direction, grant power, or verify and ensure value identification and creation. It consists of either a separate project governance process in terms of data analysis, design, implement and run/monitor/optimize or a part of the enterprise architecture continuum governance or even the LEAD Architect leadership processes.

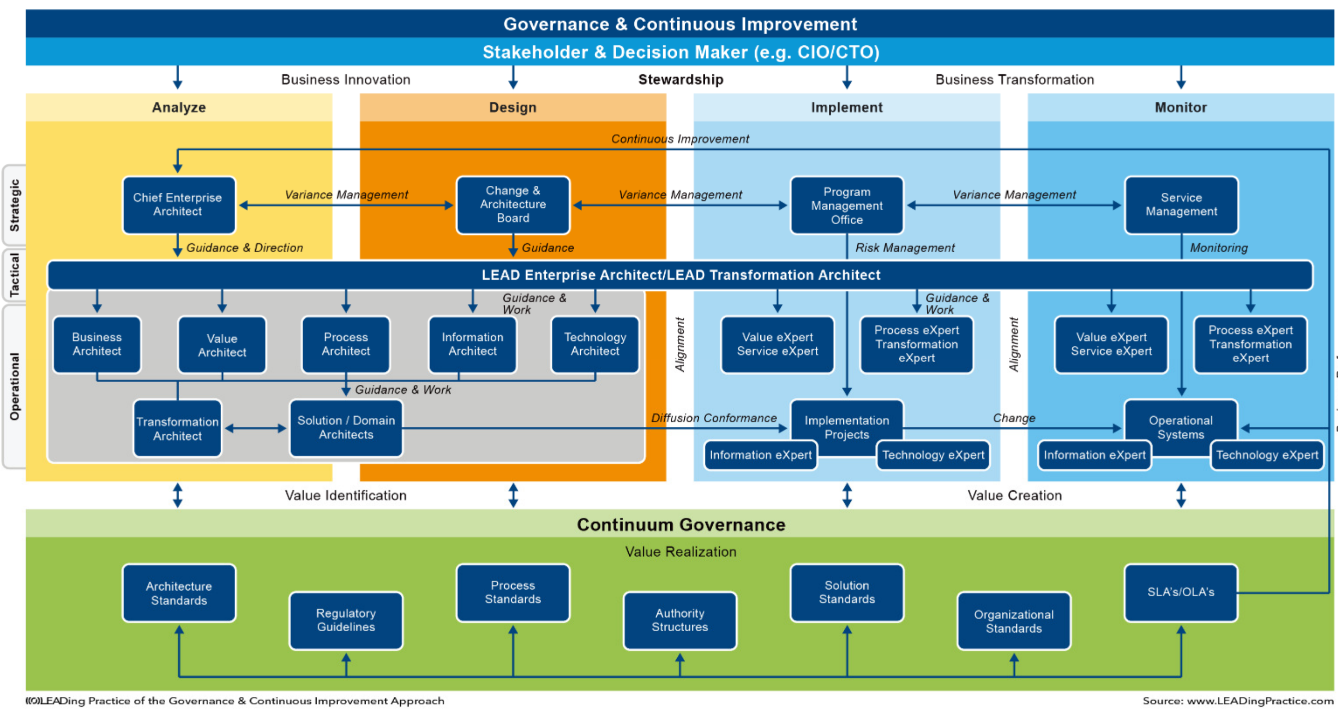

Figure 69 - The LEAD Way of Governance

A reasonable or rational purpose of governance might aim to assure, (sometimes on behalf of others in terms of stakeholders) that an organization produces a worthwhile data standard or pattern of good results while avoiding an undesirable pattern of bad circumstances. Therefore, a governance body typically administers the data architecture governance and continuous data improvement route and systems. According to Whitten[15] data modelling may be performed during various types of projects and in multiple phases of projects. Data models are progressive;

---

[15] Whitten, Jeffrey L.; Lonnie D. Bentley, Kevin C. Dittman. (2004). Systems Analysis and Design Methods. 6th edition. ISBN 0-256-19906-X.

there is no such thing as the final data model for a business or application. Instead, a data model should be considered a living document that will change in response to a changing business. The data models should ideally be stored in a repository so that they can be retrieved, expanded, and edited over time. This is why data governance is such a relevant aspect within both data engineering, modelling, and data architecture. The daily data governance consists of assuring, on behalf of those governed, the desired business innovation, transformation, and value creation while avoiding an undesirable pattern of high cost, ineffectiveness, and inefficiency (low performance). The data governance consists of the set of modelling rules, connections and architecture, regulatory requirements, processes, authority structure, solutions, organizational standards and guidelines as well as service level/operating level agreements, affecting the way people direct, administer or control a corporation.

Data governance also includes the relationships among the many players involved (the stakeholders) and the business goals. The principal players include the CIO, CTO, shareholders, data users, and the solution/ data architecture governance board. Other stakeholders might include the employees, suppliers, end-consumers and the IT community at large. The LEAD Way of Governance therefore applies to the entire lifecycle, as it represents the course of developmental changes through which the information evolves in terms of innovation and/or transformation as it passes during its lifetime. From data analysis, data strategy, data component, data tasks and data service definition, data operations, data improvements, and changes. The Data Lifecycle consists of a set of steps/phases in which each phase of the data aspects interact with other vital aspects e.g. requirements, value, process and testing. The input of one phase is the results of the previous one. It provides a sequence of phases and activities for Data Experts and Data/Solution Architects to ensure value investigation, identification, creation, realization, and governance.

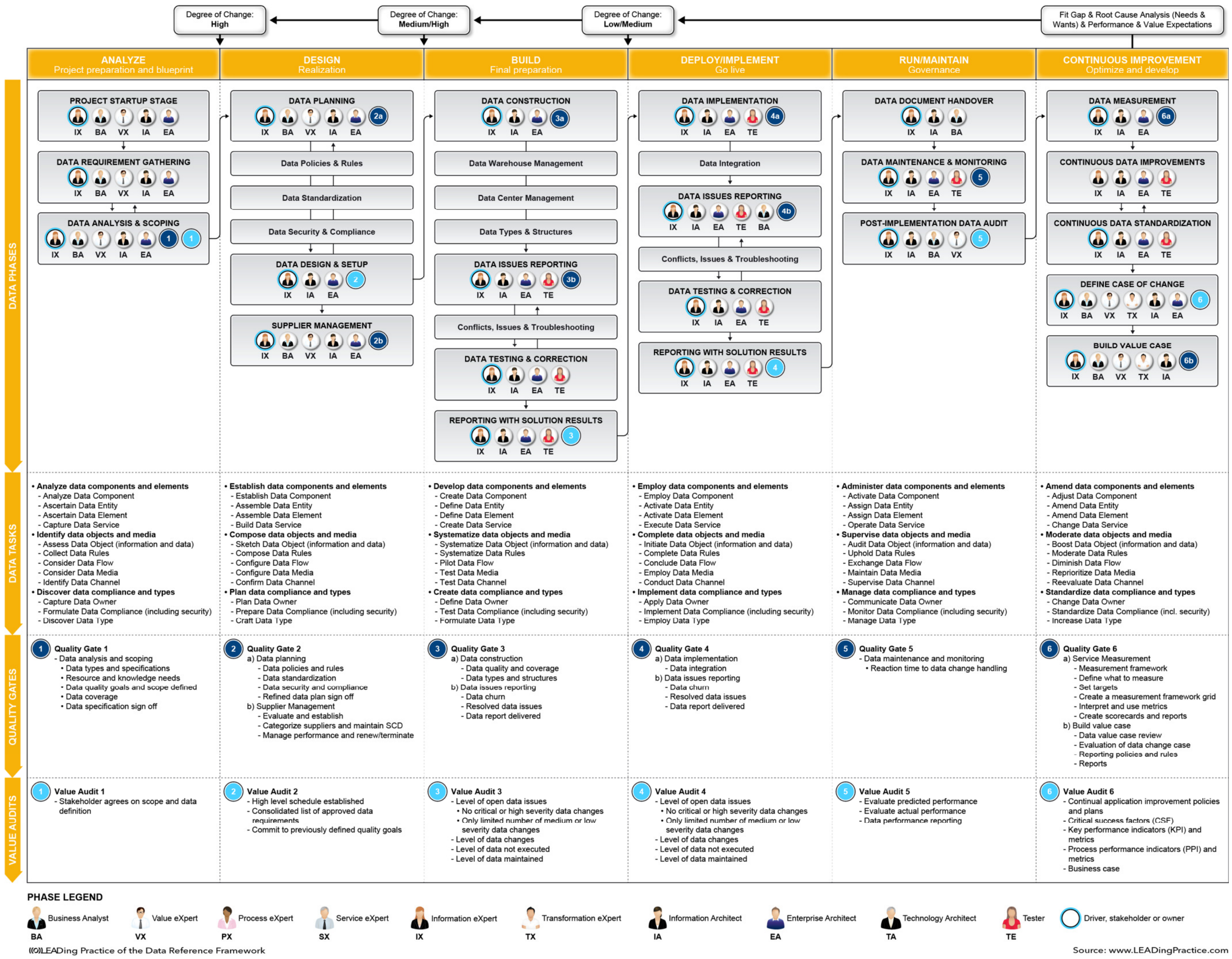

Figure 70 - The LEAD Data Lifecycle

The LEAD Data Lifecycle adheres to important phases that are essential for the various roles, such as data analysis, data strategy/planning, data design, and data implementation resources as well as those involved in continuous data improvements. Focusing on all data aspects from requirements to architecture the LEAD Data Lifecycle covers:

- **Data Analysis & Strategy**: The phase where ones data strategy is defined based on the business and data requirements e.g. business needs and wants, as well as business and data demands. Then data goals and detailed requirements are defined, data choice clarified, through blueprinting the data maps, matrix and models are developed.
- **Data Design**: The phase where one initiates, aligns, arranges, categorizes, charts, defines, determines, quantifies, drafts, outlines, and designs the <data concept>. The data design phase considers the identified business requirements and the specific design considerations for data components, data tables, data nature, data types, data entities and data services.

- **Data Build & Test**: The phase where one creates, sets up, builds, integrates, standardizes, harmonizes, consolidates, and tests the data solutions. Furthermore, data standardization as well as data integration & interface e.g. API are considered.
- **Data Implementation and/or Deployment**: The phase where one launches, implements, executes, deploys, activates, completes, concludes, and transitions the data to execution (go live).
- **Data Operation**: The phase where the data is managed in terms of its components, services, incidents/ issues and data change request fulfilments, etc.
- **Continuous Data Improvement**: The phase where one improves the existing data operation, evaluates, adjusts, alters, amends, changes, corrects, eliminates, enhances, increases, modifies, optimizes and/or excludes specific data parts.

Way of Implementing

The Data Reference Content's Way of Implementation combines the enterprise engineering, enterprise modelling, and enterprise architecture principles to apply the way of data thinking, data working, and data modelling into a physical vie of data and therefore the ability to carry out the execution of data.

Because they automate the existing Way of Working around data concepts, most implementations fall short of transforming the business and creating real data. Because of this these implementation actually tend to reinforce a siloed and ineffective way of automation. >>It is about the possibility to rethink the data flow within the information flow, the data flow, the process flow as well as the measurement and reporting flow. It can fundamentally rethink and transform the different ways of working within an organization.

The Way of Data Implementation has been developed as a fully integrated part of a Blueprinting and Implementation concept. In this way, the data aspects can be integrated to any other engineering, modelling or architecture discipline e.g. process, data, data/software, data etc. With this, the Way of Implementation provides a uniform and formal implementation concept of where the Data meta-objects and artefacts can be used. By using decomposition and composition modelling techniques within the 40 steps of the Way of Implementation, the data objects within the templates and artefacts can be applied to the relevant subjects within the different layers (business, data, or technology).

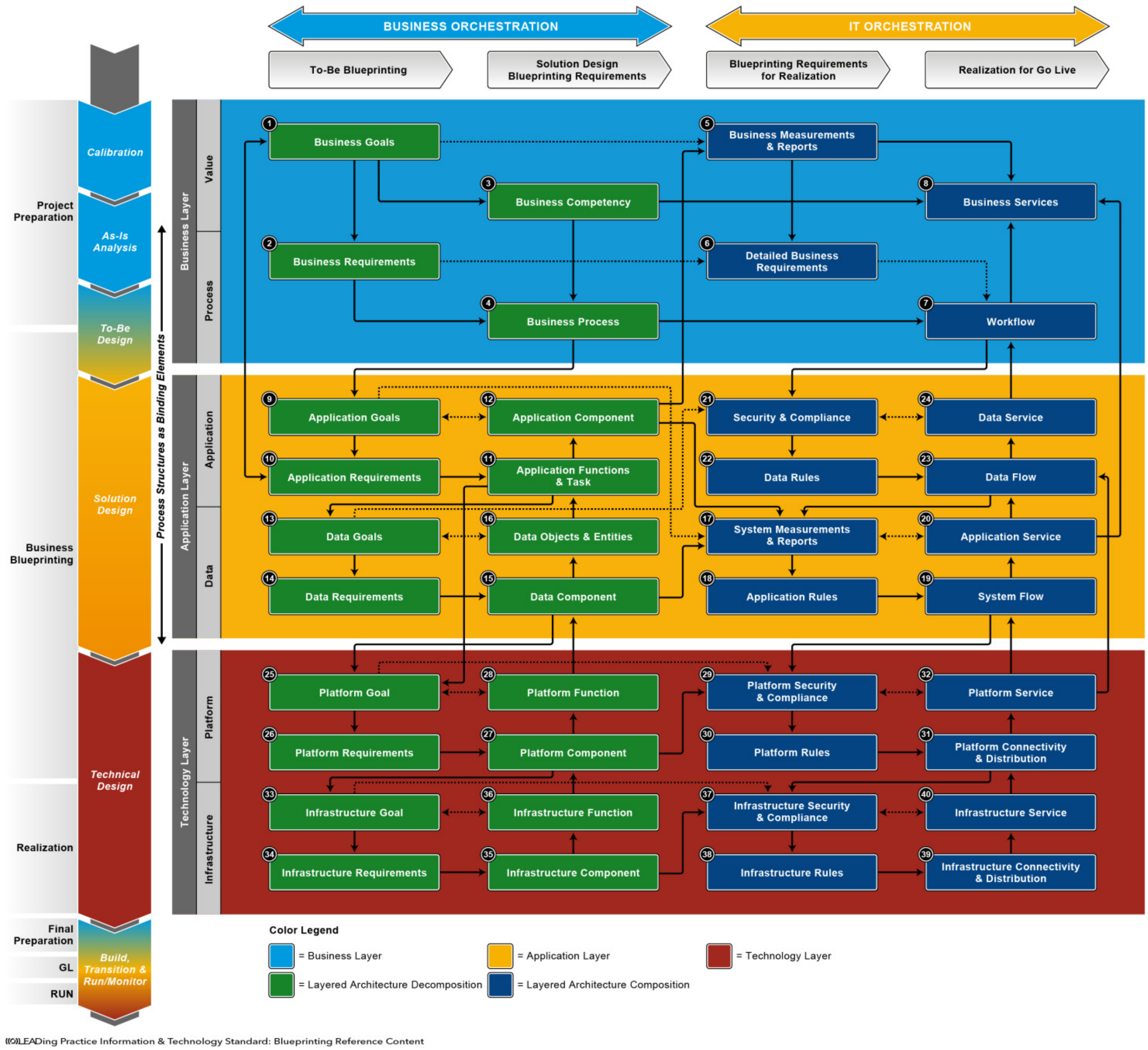

Figure 71 - A model showing the 40 Blueprinting & Implementation steps across the Business, Data, and Technology Layer.

The steps across the LEAD Blueprinting and Implementation model where the Data Objects are applied from the model shown in Figure 71 are (not all steps are relevant to data):

<u>Step 13: Data Objects and the tasks to apply them within the Data Goals step:</u>

- ✓ Matrix: Associate and relate the **goals (business, data, technology)** to the: 1. Data, 2. Data Objects, 3. Data Entities, 4. Data Services, 5. Data Owner, and 6. Data Users (Figure 13).

- ✓ Matrix: Associate and relate the **product** to the: 1. Data, 2. Data Objects, 3. Data Entities, 4. Data Services, 5. Data Owner, and 6. Data Users (Figure 24).

<u>Step 14: Data Objects and the tasks to apply them within the Data Requirements step:</u>

- ✓ Matrix: Associate and relate the **data requirements (high-level)** to the: 1. Data, 2. Data Objects, 3. Data Entities, 4. Data Services, 5. Data Owner, and 6. Data Users (Figure 14).
- ✓ Matrix: Associate and relate the **quality requirements (high-level)** to the: 1. Data, 2. Data Objects, 3. Data Entities, 4. Data Services, 5. Data Owner, and 6. Data Users (Figure 16).
- ✓ Matrix: Associate and relate the **location** to the: 1. Data, 2. Data Objects, 3. Data Entities, 4. Data Services, 5. Data Owner, and 6. Data Users (Figure 22).

<u>Step 15: Data Objects and the tasks to apply them within the Data Components step:</u>

- ✓ Map: Identify and list the **roles** working with data: 1. Business Roles, 2. Service Roles, 3. Application Roles.
- ✓ Map: Identify and categorize the **objects:** 1. Information objects, and 2. Data objects (Figure 10).
- ✓ Map: Identify, classify, and categorize the **data components** (Figure 10).
- ✓ Model: Construct a Data Model that illustrates the connection and relationship of **data components** to data.
- ✓ Map: Identify, classify, and categorize the **data entities** (Figure 10).
- ✓ Matrix: Link and associate each **data entity** to the data.
- ✓ Map: Identify, classify, and categorize the **data tables.**
- ✓ Matrix: Link and associate each **data table** to the data.
- ✓ Map: Identify, classify, and categorize the **data types** (Figure 10).
- ✓ Matrix: Associate and connect data-to-**data types**: 1. Metadata, 2. Master data, 3. Transactional data.
- ✓ Map: Identify, label, and categorize the **data media** (Figure 10).
- ✓ Map: Identify, label, and categorize the **data channels** (Figure 10).
- ✓ Matrix: Associate and relate **business services** to data (Figure 13).
- ✓ Matrix: Associate and relate the **information objects** to the: 1. Data, 2. Data Objects, 3. Data Entities, 4. Data Services, 5. Data Owner, and 6. Data Users (Figure 31).
- ✓ Matrix: Associate and link the **data owner(s)** to the data (Figure 10).
- ✓ Matrix: Associate and link the **data user(s)** to the data (Figure 10).
- ✓ Matrix: Associate and relate the **physical application components** to the: 1. Data, 2. Data Objects, 3. Data Entities, 4. Data Services, 5. Data Owner, and 6. Data Users (Figure 32).
- ✓ Model: Construct a Data Model that illustrates the connection and relationship between **physical application component(s)** and data.
- ✓ Matrix: Associate and relate the **application task(s)** to the: 1. Data, 2. Data Objects, 3. Data Entities, 4. Data Services, 5. Data Owner, and 6. Data Users (Figure 30).

<u>Step 16: Data Objects and the tasks to apply them within the Data Objects step:</u>

- ✓ Map: Identify and categorize the **objects**: 1. Information and 2. Data objects.

- ✓ Matrix: Associate and connect the data values to the **data objects**.
- ✓ Map: Identify, classify, and categorize the **data entities**.

Step 21: Data Objects and the tasks to apply them within the Application & Data Security & Compliance step:

- ✓ Matrix: Associate and relate the **risks** to the: 1. Data, 2. Data Objects, 3. Data Entities, 4. Data Services, 5. Data Owner, and 6. Data Users (Figure 18).
- ✓ Matrix: Associate and relate the **security measures** to the: 1. Data, 2. Data Objects, 3. Data Entities, 4. Data Services, 5. Data Owner, and 6. Data Users (Figure 20).
- ✓ Matrix: Associate and relate the **data compliance** to the: 1. Data, 2. Data Objects, 3. Data Entities, 4. Data Services, 5. Data Owner, and 6. Data Users (Figure 38).
- ✓ Model: Construct a Data Model that illustrates the connection and relationship between **data compliance** and data rules.

Step 22: Data Objects and the tasks to apply them within the Data Rules step:

- ✓ Matrix: Associate and relate the **data rules** to the: 1. Data, 2. Data Objects, 3. Data Entities, 4. Data Services, 5. Data Owner, and 6. Data Users (Figure 36).

Step 24: Data Objects and the tasks to apply them within the Data Services step:

- ✓ Map: Identify, list and label the **data services** (Figure 10).
- ✓ Matrix: Associate and relate the **data services** to the data.
- ✓ Matrix: Associate and relate the **business services** to the: 1. Data, 2. Data Objects, 3. Data Entities, 4. Data Services, 5. Data Owner, and 6. Data Users (Figure 26).
- ✓ Matrix: Associate and relate the **application services** to the: 1. Data, 2. Data Objects, 3. Data Entities, 4. Data Services, 5. Data Owner, and 6. Data Users (*Figure 29*).

.

Conclusion

While this document should be seen and used as a detailed description of how the LEAD data reference content can be used, it does not address all aspects of the data engineering, modelling, and architecture. It attempted to build a basis of a structured way of thinking, working, modelling, and implementation of data objects. It endeavoured to provide a standardized terminology, build common understanding and make available the standardized and integrated data templates and artefacts. Enabling practitioners to use the data reference content to:

- Identify the relevant data objects.
- Decompose the data objects into the smallest parts that can, should and needs to be modelled, and then compose the data objects entities before building them (through mapping, simulation and scenarios).
- Visualize and clarify data object relationships with the data artefacts by using maps, matrices, and models (alternative representation of information).
- Reduce and/or enhance complexity of data modelling, data engineering and data architecture principles applying the data decomposition and composition standard (see Decomposition and Composition Reference Content)
- Model the relevant data objects through the architectural layers (see Layered Architecture Reference Content).
- Adding Data Requirements (see Requirement Reference Content)
- Provide a structured Data Blueprinting and Implementation (see Blueprint & Implementation Reference Content).

For further learning around semantic object relations, decomposition and composition, layered modelling, engineering and architecture or how the data reference content can be used within the other LEADing Practice Reference Contents, we refer to both the Body of Knowledge document as well as the other Enterprise Standards and their Reference Content on www.LEADingPractice.com.

Please be aware that while we are a Community Open Source with absolutely no cost associated around our Enterprise Standards and Reference Content our material is copyright, intellectual capital and trademark protected (WIPO- World Intellectual Property Organization where we have a global patent license). Limitations apply to the use/re-use as we only provide our community with open access to all material. If you are interested to use our material or want to join our community, please feel free to contact us. We in the LEADing Practice community with +3000 organizations as well as the Global University Alliance (our founders) with +300 Universities look forward, to you joining a new way of thinking, working, modelling, and implementing data objects.

Standard it belongs to) and possibly indicate the LEAD work product family, such as "Part of the LEAD Process Reference Content".

- Cannot be systematically "given away" – do not download all our content and simply hand it over to other colleagues or clients that are not trained and certified.

To ensure correct usage, any organization usage of the LEAD material e.g. templates and tools has to be tailored and agreed upon by LEADing Practice ApS. LEADing Practice ApS may, in appropriate circumstances, and at its discretion, terminate the access/accounts of users who infringe the intellectual property rights and pursue legal action.

Guidelines for non-LEAD community members using the IPR material

The following conditions apply to use of the LEAD Intellectual Property for non-community members:

- Can be used free of charge for lecturing and research at any University and Business School

Material available at www.LEADingpractice.com can be used in a non-commercial way for knowledge sharing. When using any materials, it must include a source should be specified the following way: "Source: A part of the LEAD Reference Content" (with an indication of which type of Enterprise or Industry Standard it belongs to) and possibly indicate the LEAD work product family, such as "Part of the LEAD Process Reference Content".

General guidelines that apply for all LEAD IPR material

- Any use of original texts, graphics, images, screen shots, and other materials from LEAD sources must be approved by LEADing Practice ApS.
- Any material cannot be generally distributed to colleagues, clients and or an undefined audience without written permission from LEADing Practice ApS.
- Cannot be altered or changed (the using organization) in any way without explicit written permission from LEADing Practice ApS.

In most cases, the LEADing Practice ApS acts as a distribution channel for the Publisher and Authors of the material provided. LEADing Practice ApS may, in appropriate circumstances of infringement of the intellectual property rights pursue legal action. For questions, please get in touch with us at info@LEADingPractice.com.