Data Models

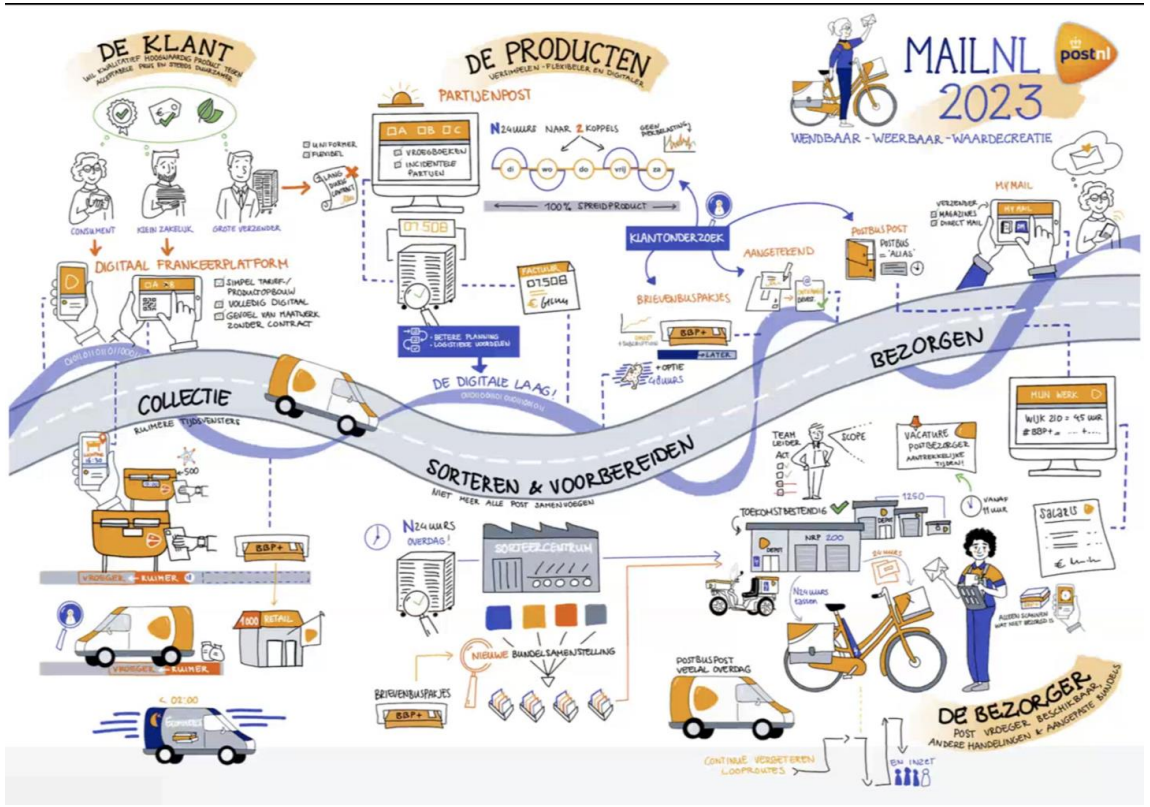
Explanation & examples

Den Haag • january 2023

Business Requirements

Business needed to get a grip on its data:

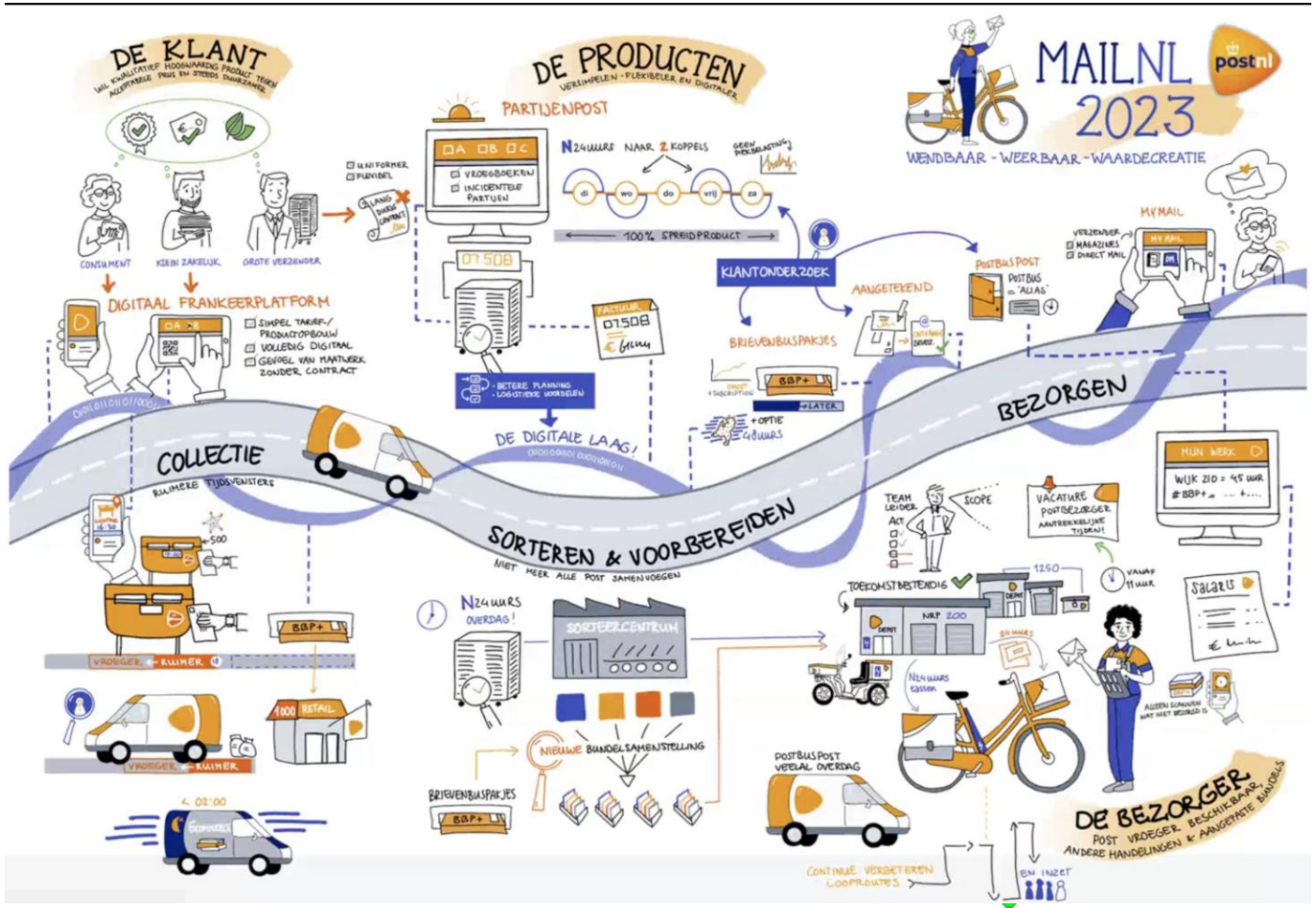
- how does data flow through which systems
- who is responsible for data
- improve data quality
- clear definities, etc



Business goal is to set up some high-level models to:

- Know what data does my landscape consist of
- Have one definition for business/ data objects across the landscape
- Know where is which data used
- Then determine the ABBs/SBBs that have a relationship with these data objects (SoE, SoR, SoU)
- Finally assess the impact of incidents, changes and new functionalities through the chain

Blueprint MailNL – Process Chain & Capabilities



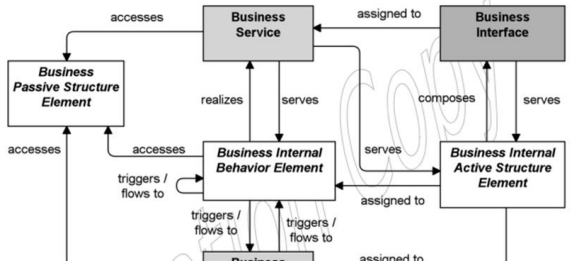
Klant order			Planning		Sourcing		Productie				Bezorgen	Facturatie & nazorg		
Cust mgmt	Order mgmt	Product mgmt	Voor-spellen	Volume planning	Inzet planning	Capaciteit planning	Transport	Adressen soorteer lijsten	Sorteren + voor-bereiden	Opera-tioneel sturing	Bezorgen	Factura-tie	Consu-men mgmt	Helpdesk

Business Object – Data Object

Archimate

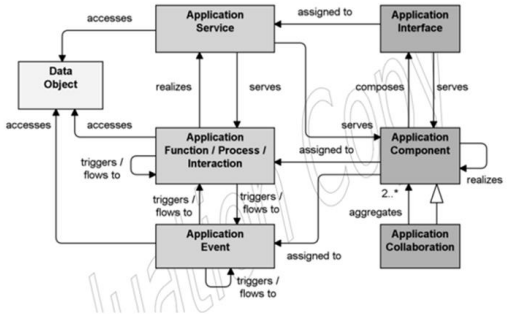
Business Layer Metamodel

Figure 49 gives an overview of the Business Layer elements and their relationships. Business internal active structure element, business internal behavior element, and business passive structure element are abstract elements; only their specializations (as defined in the following sections) are instantiated in models.



Application Layer Metamodel

Figure 67 gives an overview of the Application Layer elements and their relationships. Whenever applicable, inspiration has been drawn from the analogy with the Business Layer.



Business Object

A business object represents a concept used within a particular business domain.

As explained in Section 3.6, the ArchiMate language in general focuses on the modeling of types, not instances, since this is the most relevant at the Enterprise Architecture level of description. Hence a business object typically models an object type (*cf.* a UML class) of which multiple instances may exist in operations. Only occasionally, business objects represent actual instances of information produced and consumed by behavior elements such as business processes. This is in particular the case for singleton types; i.e., types that have only one instance.

A wide variety of types of business objects can be defined. Business objects are passive in the sense that they do not trigger or perform processes. A business object could be used to represent information assets that are relevant from a business point of view and can be realized by data objects.

Business objects may be accessed (e.g., in the case of information objects, they may be created, read, or written) by a business process, function, business interaction, business event, or business service. A business object may have association, specialization, aggregation, or composition relationships with other business objects. A business object may be realized by a representation or by a data object (or both). The name of a business object should preferably be a noun.

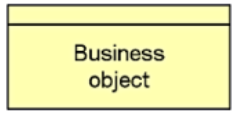


Figure 65: Business Object Notation

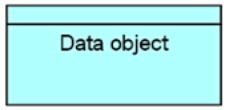
Data Object

A data object represents data structured for automated processing.

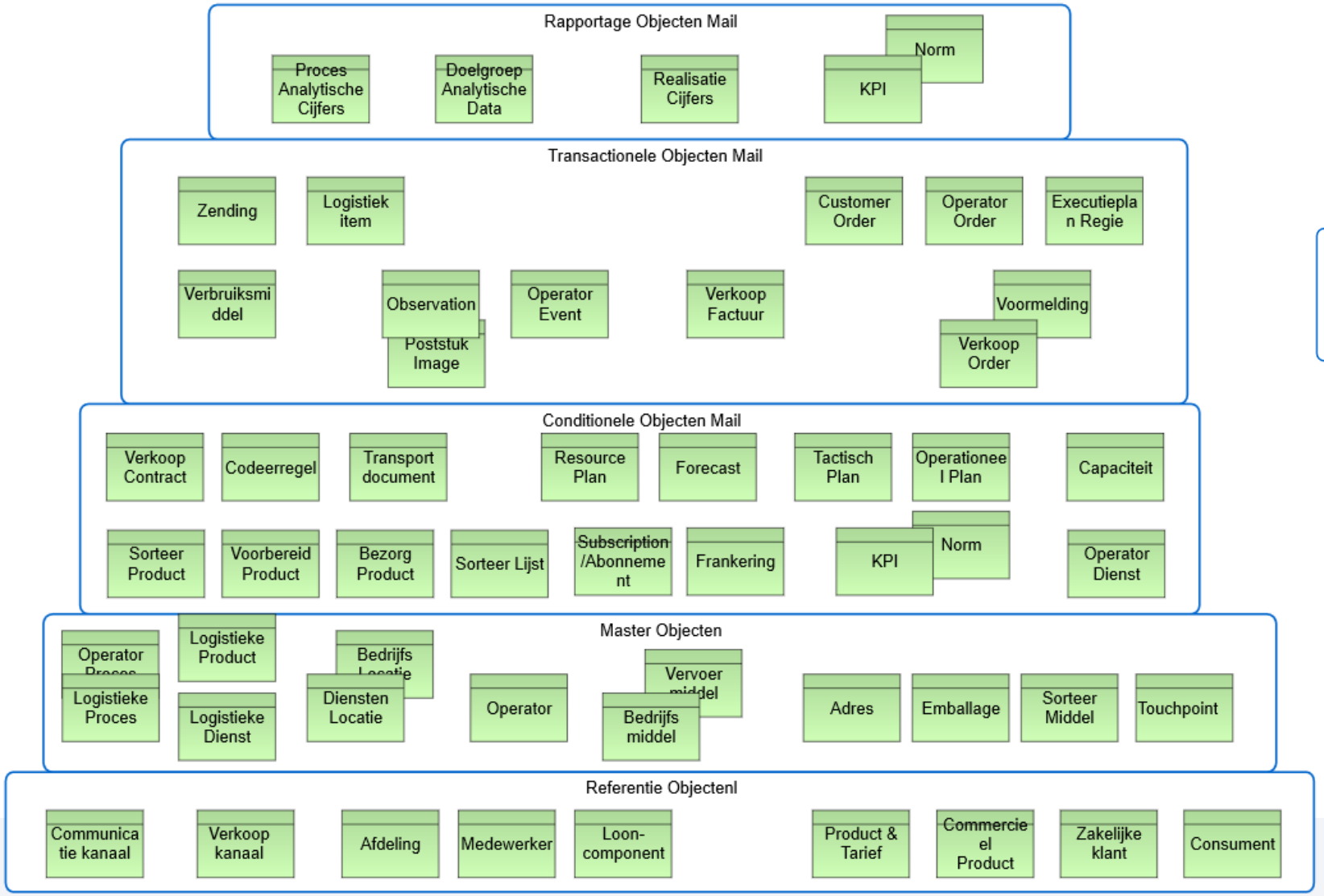
A data object should be a self-contained piece of information with a clear meaning to the business, not just to the application level. Typical examples of data objects are a customer record, a client database, or an insurance claim.

As explained in Section 3.6, the ArchiMate language in general focuses on the modeling of types, not instances, since this is the most relevant at the Enterprise Architecture level of description. Hence a data object typically models an object type (*cf.* a UML class) of which multiple instances may exist in operational applications. An important exception is when a data object is used to model a data collection such as a database, of which only one instance exists.

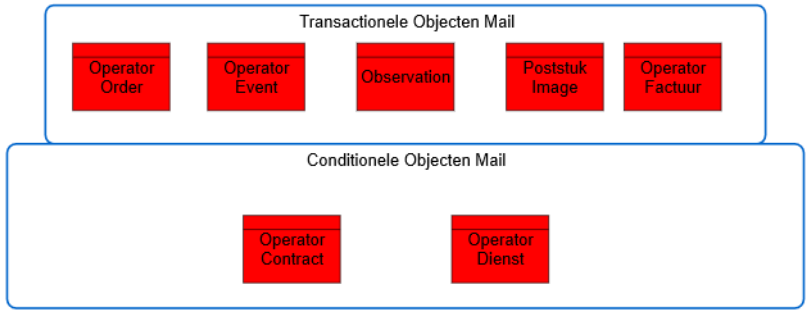
An application function or process can operate on data objects. A data object may be communicated via interactions and used or produced by application services. A data object can be accessed by an application function, application interaction, or application service. A data object may realize a business object and may be realized by an artifact. A data object may have association, specialization, aggregation, or composition relationships with other data objects. The name of a data object should preferably be a noun.



Business Object Classification Model Mail

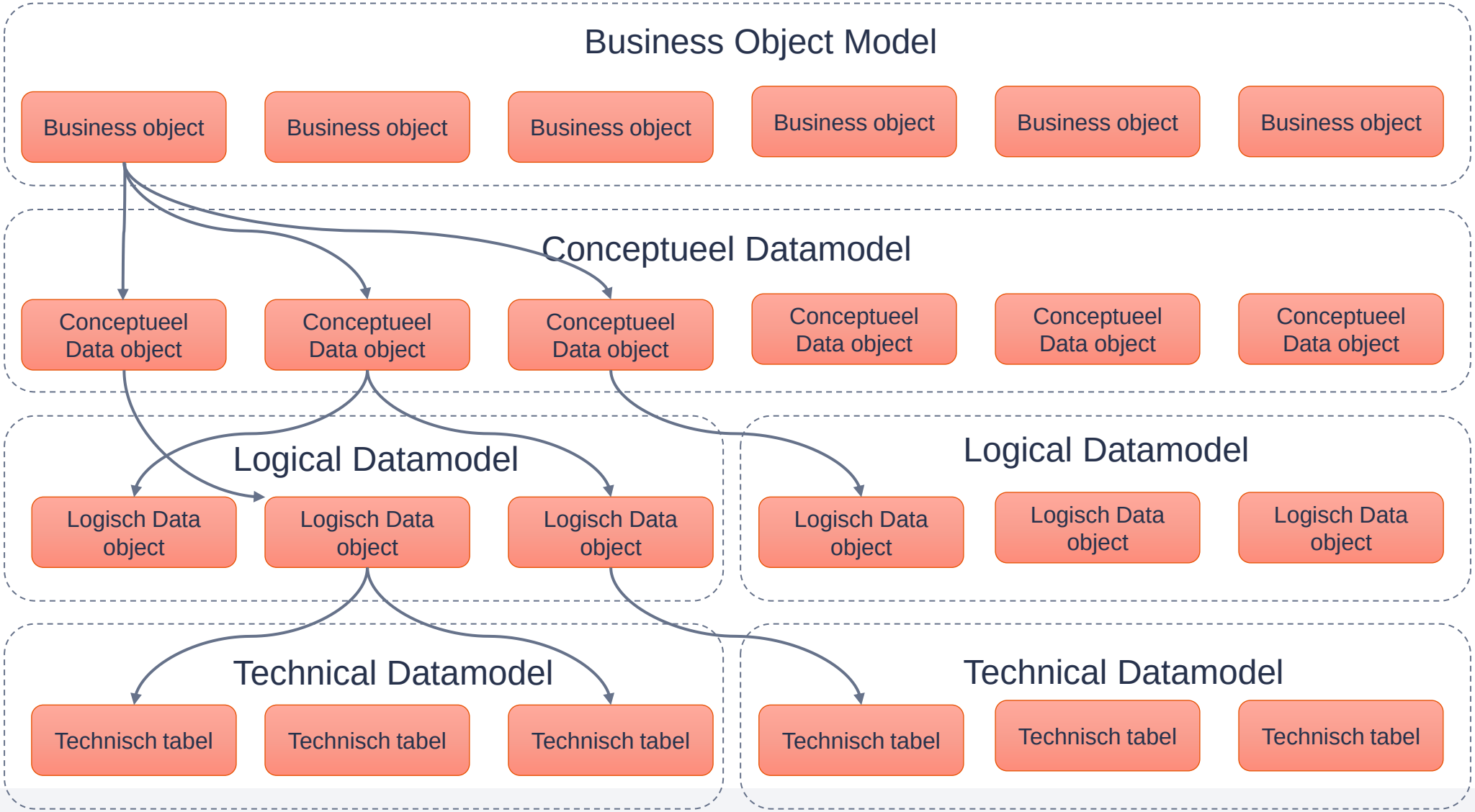


MailNL als operator voor een Regie organisatie

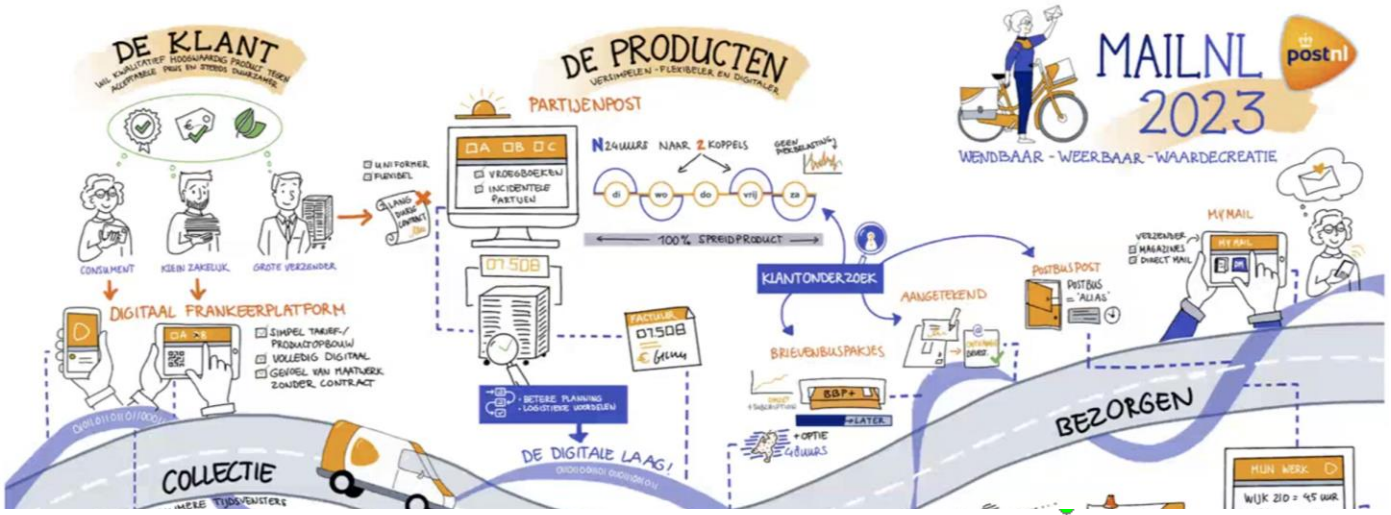


A Business Object Classification Model is a graphical representation of the business objects in the domain, categorized on the layers of the pyramid.
Including the corresponding definitions

Coherence over the model layers



Blueprint MailNL & Datamodels



Klant order			Planning		Sourcing		Productie				Bezorgen	Facturatie & nazorg		
Cust mgmt	Order mgmt	Product mgmt	Voor-spellen	Volume planning	Inzet planning	Capaciteit planning	Transport	Adressen soorteer lijsten	Sorteren + voor-bereiden	Opera-tioneel sturing	Bezorgen	Factura-tie	Consu-men mgmt	Helpdesk

CM: PartijenPost Keten

CM: Forecasting & Planning

CM: Klantperspectief

CM: Klantperspectief

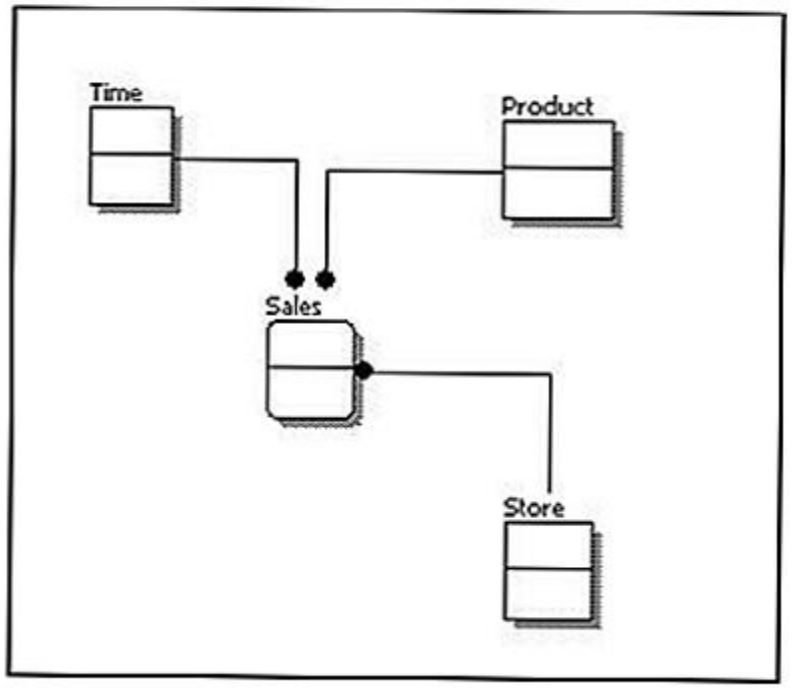
CM: Sustainability

MASTERDATA

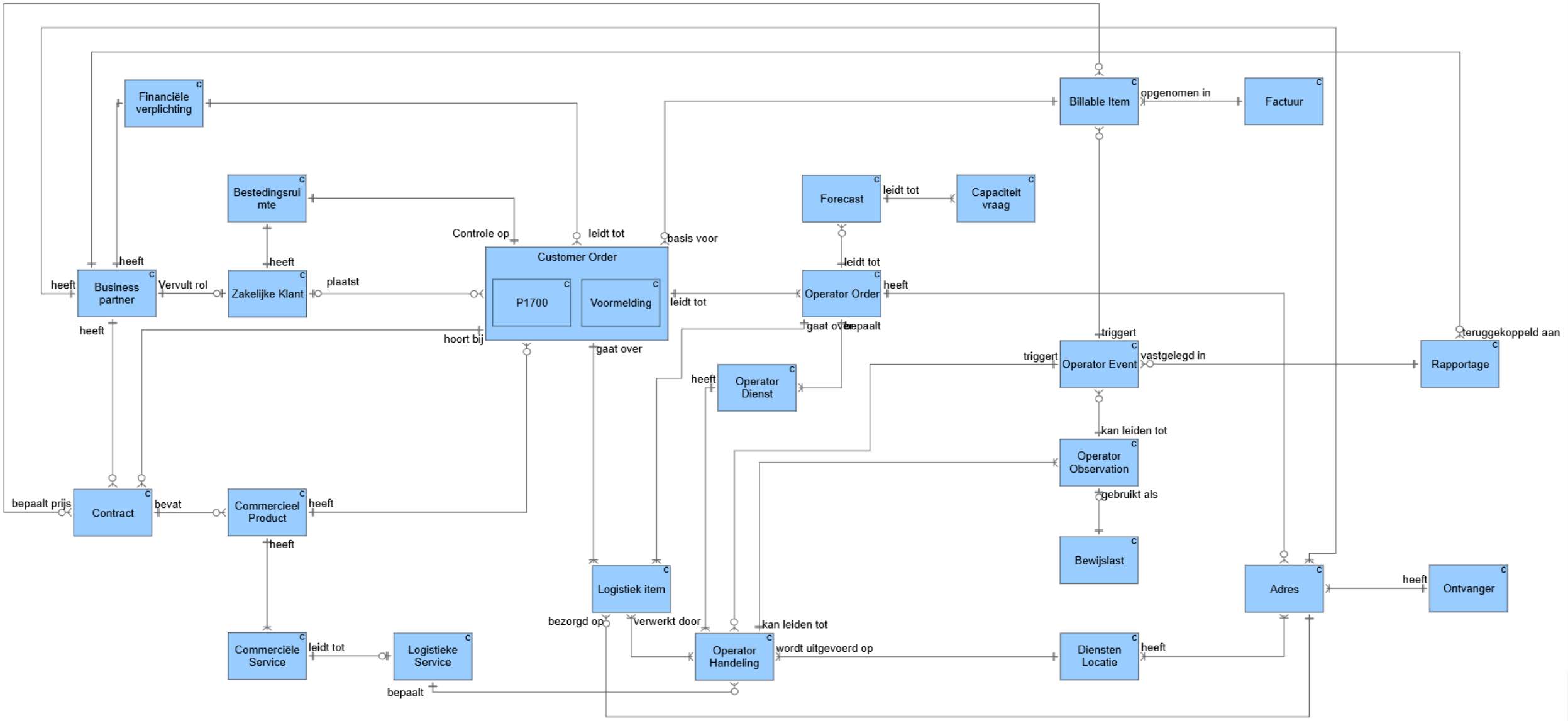
Conceptual Data Model

The conceptual data model shows how the data elements are structured and what the connections are between them.

Feature	Conceptual
Entity names	X
Entity relationships	X
Attributes	
Primary keys	
Foreign keys	
Table names	
Column names	
Column data types	



CM PartijenPost Keten - example

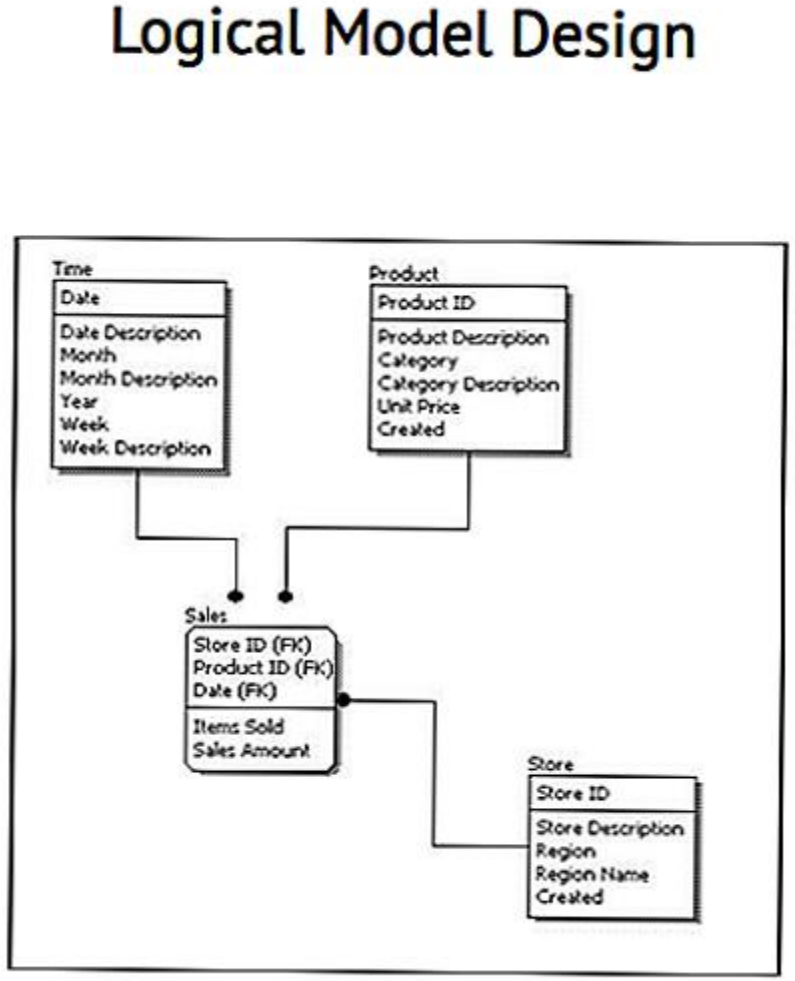


Logical Data Model

This model is more detailed than the conceptual one, without considering how the information will be physically applied in the database.

- all entities and their relationships.
- attributes for each entity.
- primary and foreign key for each entity.

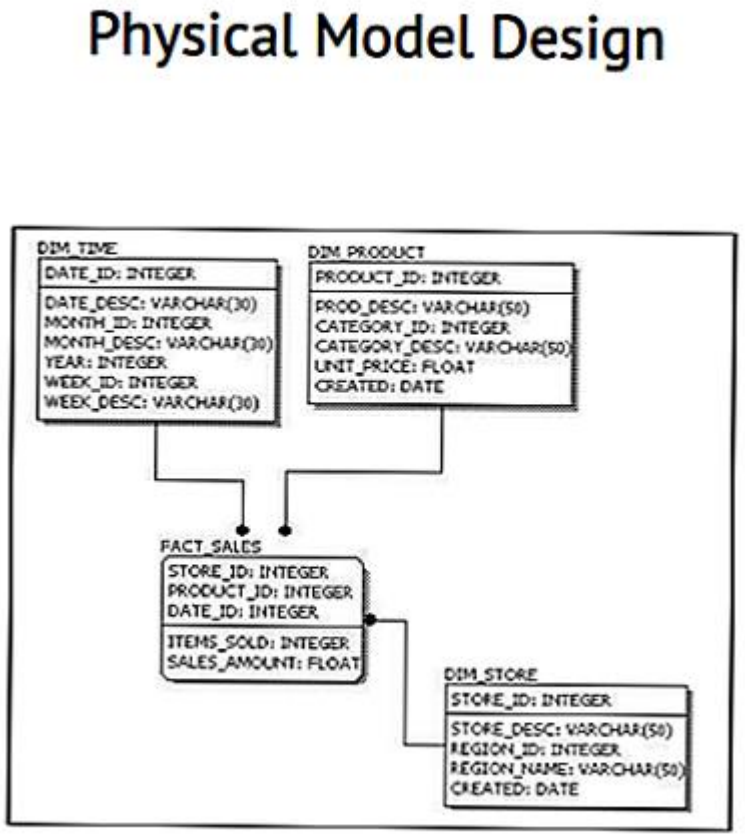
Feature	Logical
Entity names	X
Entity relationships	X
Attributes	X
Primary keys	X
Foreign keys	X
Table names	
Column names	
Column data types	



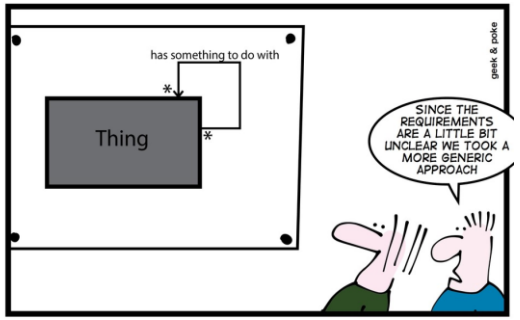
Technical / Physical Data Model

The technical data model shows the final implementation in a system including table names, attributes and data types.

Feature	Physical
Entity names	
Entity relationships	
Attributes	
Primary keys	X
Foreign keys	X
Table names	X
Column names	X
Column data types	X



Summary



HOW TO CREATE A STABLE DATA MODEL

Data Modellen

Met een datamodel, of gegevensmodel, wordt beschreven hoe de gegevens in een informatiesysteem gestructureerd zijn. Onderscheid wordt gemaakt tussen het semantisch model, conceptueel datamodel, het logisch datamodel en het fysiek datamodel.

Het **Semantisch model** is een praatplaat in bijv. PowerPoint. Doel is om samen met de business de scope te bepalen en het begrippenkader helder te maken zonder dat kennis nodig is van de werking van een datamodel.

Het **Conceptuele datamodel** beschrijft de structuur van en de relaties tussen de conceptuele gegevensobjecten, genaamd entiteiten. Tevens toont het de context van het gemodelleerde model. De grafische vastlegging is in de vorm van een ERD, Entiteit Relatie Diagram.

Het **Logisch datamodel** beschrijft de structuur van en de referenties tussen de logische gegevensobjecten, genaamd tabellen. Het logisch datamodel wordt vastgelegd in een ERD en voldoet aan de 3e normaalvorm van Codd (zie Cookbook datamodelleren voor uitleg hierover).

Het **Fysieke datamodel** beschrijft de manier waarop gegevens in een individuele database zijn opgeslagen.

Conceptuele en logische datamodellen kunnen worden gebruikt als handig middel voor het in kaart brengen van een data gerelateerd business vraagstuk en de erbij behorende informatiebehoefte. Het datamodel wordt dan gebruikt als een handig te bewerken praatplaat die houvast biedt tijdens brainstormsessies en andere gedachten vormden discussies. Delen van het model die niet relevant zijn, worden verwijderd en dummy entiteiten die nodig zijn om het verhaal vanuit business perspectief te verduidelijken mogen worden toegevoegd.